Application manual

Force Control for Machining

Controller software IRC5
RobotWare 5.12

Application manual

Force Control for Machining

RobotWare 5.12

Document ID: 3HAC027595-001

Revision: E

ABB AB
Robotics Products
SE-721 68 Västerås
Sweden

Overview

About this manual

This manual contains information about the option RobotWare Machining FC (Force Control for Machining).

Usage

This manual can be used to find out what Force Control for machining is and how to use it. It provides information about system parameters and RAPID components related to Force Control for Machining, and examples of how to use them.

Limitations

- The total load, i.e. the sum of gravitational forces (sensor, gripper and part) and external contact forces, must not exceed limits as specified in the load diagrams for the specific robot used.
- Force Control can not be activated when the robot is running in MultiMove Coordinated mode.

Who should read this manual?

This manual is mainly intended for robot programmers.

Prerequisites

The reader should...

- be familiar with industrial robots and their terminology.
- be familiar with the RAPID programming language.
- be familiar with system parameters and how to configure them.

Organization of chapters

The manual is organized in the following chapters:

Chapter	Contents
1. Introduction	Introduction to Force Control for Machining
2. Installation	Information on how to get started.
3. Navigate and handle the graphical user interface	Description and navigation of the RW Machining FC interface
4. Configuration	Description of system parameters used in Force Control for Machining including configuration example.
5. Programming	Information on how to program Force Control for Machining. RAPID instructions, functions and data types, including code examples.
6. Execution behavior	Description of conflicting reference values and special cases.
7. Troubleshooting	Information on what do when the robot’s behavior is faulty.
8. Rapid reference information	RAPID components used in Force Control for Machining.

Continues on next page

Continued

Chapter	Contents
9. System parameters reference information	All parameters used in Force Control for Machining.
10. Further references	Document references and more detailed information

References

Reference	Document ID
Application manual - MultiMove	3HAC021272-001
Operating manual - IRC5 with FlexPendant	3HAC16590-1
Operating manual - RobotStudio	3HAC032104-001
Product manual - IRC5	3HAC021313-001
Technical reference manual - RAPID overview	3HAC16580-1
Technical reference manual - RAPID Instructions, Functions and Data types	3HAC16581-1
Technical reference manual - System parameters	3HAC17076-1

Revisions

Revision	Description
-	First edition
A	Added instructions and system parameters
B	Added new chapter which describes the user interface of RobotWare Machining FC. A new system parameter, *Sensor serial number*, is added.
C	The following new system parameters added: *Disable check of saturation*, *Time in saturation before error*, and *Time in saturation before warning*.
D	Added new section *6.1.3 Programming in Path coordinates.* Minor corrections made.
E	Chapter *3 Navigate and handle the graphical user interface* is updated with the following: • New section, *Improving pressure process quality*, is added • Added tips for teaching, learning, and exporting Structure changed in chapter *6 Execution behavior*, and illustration added to section *6.1 Damping and LP-filter*. New optional argument added to section *8.1.2 FCPress1LStart*.. New data type added to section *8.3 Data types.* Added signals in section *10 Test Signal Viewer*. Minor corrections made.

1 Introduction

1.1. About Machining FC

Purpose

Robot machining with force control includes different kinds of material removal processes, surface finishing and surface processing. Where the robot can hold the tool and work on a fixed part or hold the part and work on a fixed tool.

Examples of process where Machining FC is useful:

- Grinding
- Milling
- Polishing
- Deburring
- Deflashing
- etc.

The package Machining FC consist of two parts, FC Pressure and FC SpeedChange. For more information about these, see *About FC Pressure on page 10* and *About FC SpeedChange on page 11*.

What is included

The option RobotWare Machining FC gives you access to:

- FC Pressure
- FC SpeedChange
- Instructions to setup gravity compensation and sensor offset calibration.
- Instruction for supervision
- Instructions for defining reference values (i.e. desired force, SpeedChange parameters, movement or sensor interface).

1.2. About FC Pressure

Purpose

The purpose of FC Pressure is to make the robot sensitive to contact forces. The robot can "feel" its surroundings and follow the surface of the processed part to obtain a certain pressure against an object. This means that the robot will change its position in order to apply a constant force/pressure on a surface, even if the exact position of the surface is not known.

Since pressure is obtained by moving the robot path, this function is suited for polishing, grinding and cleaning, where a surface should be made even and smooth. The material that is removed and the changes of the surface topology / dimensions depends on the process parameters like tooling, applied pressure, robot speed etc.

Here are some examples from foundry and metal fabrication where FC Pressure is useful:

- Grinding of faucets
- Polishing of kitchen sinks
- Deflashing, grinding and cleaning of castings
- Deburring of castings
- etc

What is included

The function FC Pressure gives you access to:

- Instructions for programming FC Pressure start, movements and stop.

What is needed

FC Pressure requires a sensor input from the measured process forces to adjust the behavior of the robot. Depending of the application and required flexibility different force/torque sensors can be used.

For applications with the function FC Pressure, use

- 1 DOF (Degree Of Freedom), if the direction of the force is constant and the sensor can be integrated in the tooling.
- 6 DOF, for more flexible solutions.

Basic approach

1. Identify the load and calibrate the system
2. Move to a point close to contact
3. Set up desired force and start movement towards the surface
4. Move linear or circular performing the process with contact.
5. Leave surface and deactivate force control.

1.3. About FC SpeedChange

Purpose

In processes where path accuracy is important and where the finished result shall comply with specific dimensions, FC SpeedChange is recommended. This function will be useful combined with force sensor or other input indicating excessive process forces, which can deteriorate the finished result. When SpeedChange is active and if machining forces exceed a certain value, then the path speed will automatically be reduced, thus decreasing forces, minimizing changed dimensions due to deflections of the robot arm and most probably avoid damaging the part/tool due to stress and heat. This will guarantee path accuracy even if much material shall be removed.

See below some examples from foundry and metal fabrication where FC SpeedChange is useful

- Grinding unevenly distributed material on casted surfaces
- Milling along the edge of a work piece
- Deburring along contour of a work piece
- Deflashing unevenly distributed burr along a part line on castings
- Deburring of castings
- etc.

What is included

The function FC SpeedChange gives you access to

- Instructions for programming FC SpeedChange.
- Instructions for defining, a recover function for FC SpeedChange

What is needed

FC SpeedChange requires a sensor input from the measured process forces to adjust the behavior of the robot. Depending of the application and required flexibility different sensors can be used.

For applications with the function FC SpeedChange, use

- built in (analog, voltage) signal from the spindle representing the current or the torque of the motor. Spindle motors used for milling or grinding normaly has a built in signal.
- 6 DOF force/torque, for more flexible solutions.

Basic aproach

1. Configure the parameters for SpeedChange controller
2. Identify the load and calibrate the sensor (if 6DOF sensor is used)
3. Define recover function
4. Activate FC SpeedChange
5. Execute the machining task; move along the path performing the process.
6. Deactivate FC SpeedChange.

2 Installation

2.1. Getting Started

Overview

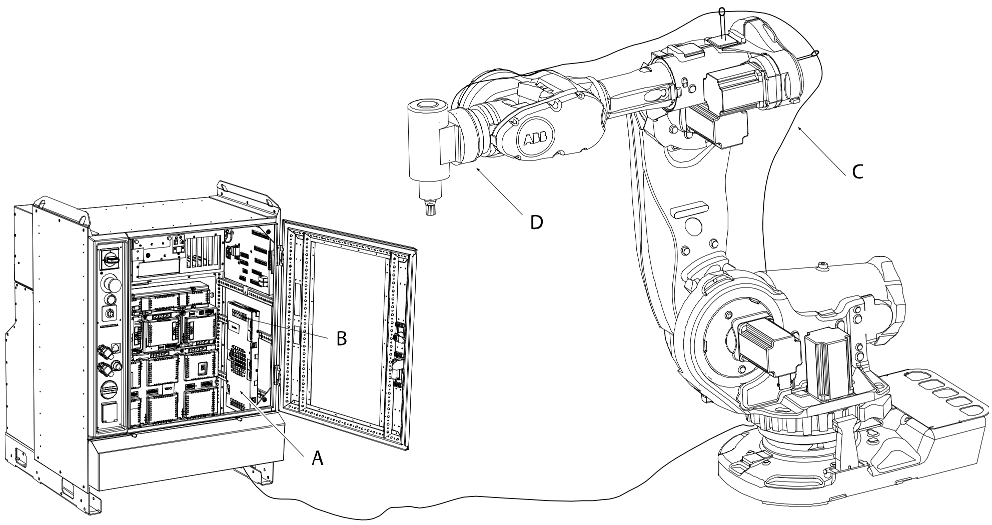
This chapter describes the basic steps to get started with Force Control, from mounting the sensor to writing the first program. This manual only describes what is specific for a Force Control installation. If Force Control is used in a MultiMove installation, that is, several robots with a single robot controller, see also *Application manual - MultiMove*. For more information about installation and commissioning of the controller, see *Product manual - IRC5*.

Hardware using a force sensor

The following hardware items are needed for Force Control:

A. DSQC635, axis computer with support for PMC
B. PMC card (PCI Mezzanine Card)
C. Cable between force sensor and PMC card
D. Robot mounted or room fixed force sensor

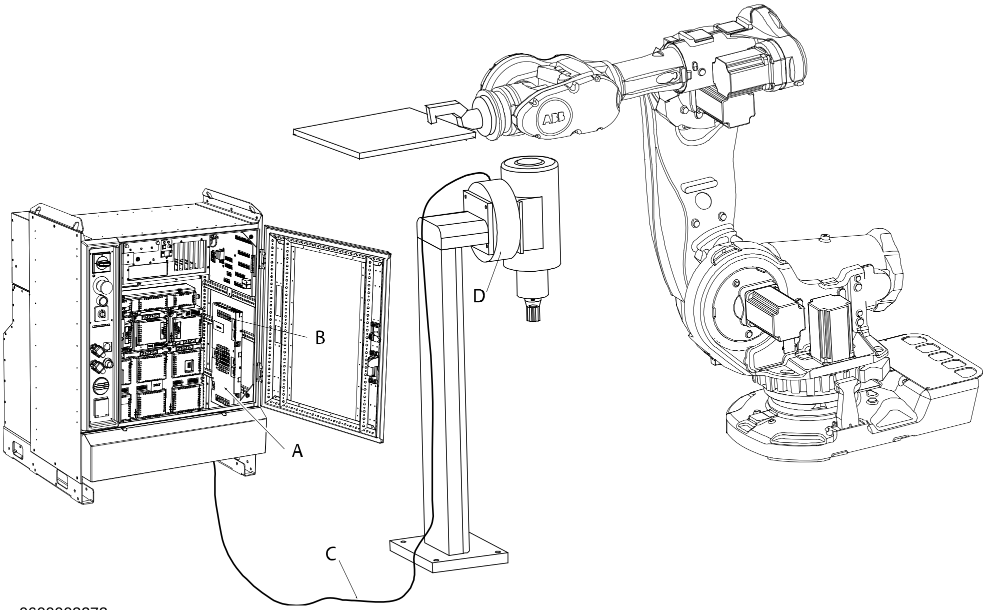
Robot Mounted Sensor



xx0600003272

Continues on next page

Room fixed sensor



xx0600003273

PMC card, cable and force sensor

The PMC card, cable and force sensor are extra hardware needed for the Force Control option. These parts can be bought from any sensor supplier. ABB has integrated a support using Force sensors from ATI Industrial Automation which includes adapter plate and calibration parameter file for easy integration, but other sensor supplier can also be used. For further details see *About the Force Sensor interface on page 203*. The supported PMC card is manufactured by Acromag and called PMC330 (16bit A/D interface). The PMC card enables the axis computer to read sensor values. It is mounted on the axis computer and connected to the sensor by a cable. For information on how to mount the PMC card see *Product manual - IRC5*, section *Installation of PMC card* for Force Control Function.

NOTE!

For recommended 6 DOF force sensor see *Product specification - Controller Software IRC5, RobotWare 5*.0

Hardware using spindle signal for SpeedChange

The following hardware items are needed for Force Control:

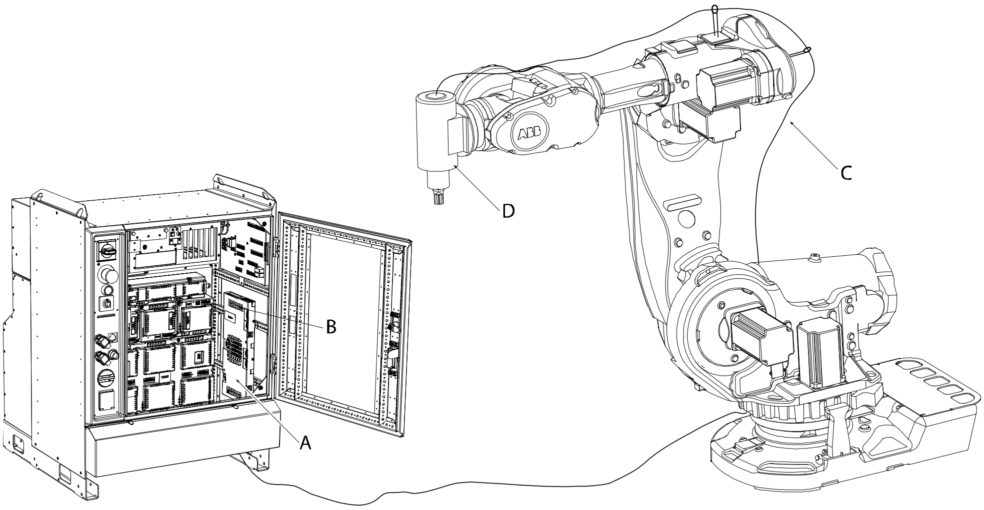
A. DSQC635, axis computer with support for PMC

B. PMC card (PCI Mezzanine Card)

C. Cable between Spindle motor and PMC card

D. Spindle with analog voltage signal output

Continues on next page



xx060000327

PMC card, cable and spindle signal

The PMC card and cable are extra hardware needed for the Force Control option FC SpeedChange. The supported PMC card is manufactured by ACROMAG and called PMC 330 (16 bit A/D interface). The PMC card enables the axis computer to read the sensor value. It is mounted on the axis computer and connected to the spindle signal output by a cable. The signal output (representing motor current or motor torque) must be integrated with the spindle used for machining. The cable shall connect the spindle motor and PMC card. For more information on how to mount the PMC card see *Product manual - IRC5*, section *Installation of PMC* card for Force Control Function. For more information regarding the PMC card interface see *Acromag DAQ board on page 204*.As an alternative to the figure above, the spindle may be mounted stationary while the robot holds the work piece.

NOTE!

For recommended 6 DOF force sensor see *Product specification - Controller Software IRC5, RobotWare 5.0*

Procedure

	Action	Note/illustration
1.	Mount the force sensor fixed in the room or on the robot's mounting flange. Sometimes adapter plates are needed between robot and sensor.	Read the sensor supplier manual and be careful so that the plates are not rotated by mistake.
2.	Locate the signal output on the spindle motor	Read Spindle supplier manual for more data. Make sure the signal amplitude is between 5V and 10V.
3.	Mount the PMC card on the axis computer.	See *Product manual - IRC5 - Installation of PMC card for Force Control Function.*

Continues on next page

Continued

	Action	Note/illustration
4.	Connect the cable from the PMC card to the sensor/ spindle motor.	• Make sure to position the cable on the robot so that it is not damaged by the movements of the robot. • If a 1DOF sensor is going to be used, make sure that the sensor signal is connected to correct channel in combination with the settings of the calibration matrix in order to activate Fx, Fy or Fz according to the reference settings in the program/application. • If a spindle signal is used for SpeedChange a channel to connect the cable needs to be set. The channel is set by the parameter *DAC channel on page 182*.
5.	Install RobotStudio and RobotWare on a PC.	See *Operating manual - RobotStudio*.
6.	Create a new system in RobotStudio. In the **Modify Options** dialog box, select the RobotWare option *Machining FC*.	See *Operating manual - RobotStudio*
7.	Configure the system using RobotStudio. Force Control parameters are set in the configuration topic Motion.	Some of the Force Control parameters are already set. However, make sure they work for your application. When using MultiMove it is necessary to set a few parameters, see chapter *Programming in MultiMove System*.
8.	If the sensor manufacturer is ATI Automation, sensor calibration data can be loaded from a CD supplied with the sensor. The file can be found on the CD in the directory **calibration/ ABB_FTxxxx.cfg**, where xxxx is the serial number of the sensor. For any other sensor/signal configuration, sensor calibration data needs to be supplied by the manufacturer. Right click the **Configuration** node in RobotStudio, select **Load Parameters** and then **Load parameters and replace duplicates**.	See also *System parameters overview on page 63* and *Configuration example on page 67*. In a MultiMove system use the file in the right directory according to the selected robot: • ATI_ACROMAG1 for Robot1 • ATI_ACROMAG2 for Robot2 • ATI_ACROMAG3 for Robot3 • ATI_ACROMAG4 for Robot4
9.	Now that the system is configured the last step is to program the application. To get started more easily there are some basic code examples on how to use Force control. There is also a Rapid component overview for easy usage.	*RAPID Components on page 71* *Code examples on page 76*

Continues on next page

General guidelines for Force Control

These guidelines can be useful even if you are an advanced programmer in Force Control and it is recommended to go through these steps from time to time.

1. Force sensor calibration is required prior to any operation with force control enabled.
2. Jogging the robot in force control is possible but all teaching and jogging in force control mode should be done with extreme caution.
3. Avoid programming and jogging through singularities (see *Technical reference manual - RAPID overview*, section *Singularities*).
4. Always change damping parameters carefully and in small steps.
5. Always change bandwidth of force filter carefully and in small steps.
6. If you want to be sure to limit reference forces and movement speed, change the maximum parameters in the configuration to a desired value to avoid mistakes in the program.
7. Avoid deactivating Force Control while in contact. Remember that force control is deactivated when the program pointer is moved.
8. In case the machining program is interrupted, always start the program from the beginning.

Programming guidelines for Force Control

These guidelines can be useful even if you are an advanced programmer in Force Control and it is recommended to go through these steps from time to time.

1. Even if Force Control will change the path to obtain the contact reference, it's always best to have an initial programming of the path as close to the correct surface as possible.
2. Always try your new FC Pressure instructions with small reference without contact to any object to verify the movements.
3. Make sure that the FC start and end positions are not in contact with the work piece, but as close as possible.
4. Always use a smaller reference than intended when trying out the new movements.
5. Log the test signal for the spindle current/torque at normal operation in order to identify the correct, desirable signal level to be use in the SpeedChange controller.

3 Navigate and handle the graphical user interface

3.1 FlexPendant interface

3.1.1. RobotWare Machining FC main menu

RobotWare Machining FC interface

The RobotWare Machining FC interface is available from the ABB menu on the FlexPendant.

NOTE!

The graphical user interface of RobotWare Machining FC can only be used with a robot mounted 6 DOF sensor.

Opening the interface of RobotWare Machining FC

Use this procedure to start RobotWare Machining FC:

1. Tap the **ABB** button to display the ABB menu.

   RobotWare Machining FC is listed as a menu item in this menu.

pic105-000

2. In the ABB menu tap **RobotWare Machining FC**.

   The main menu of RoboWare Machining FC opens.

pic1051-001

Continues on next page

Part	Description
FC Setup	Access to Force Control LoadID, Activate Force Control and Deactivate Force Control. See *FC Setup on page 22*.
Load	Load an existing project to the system. See *Loading a project on page 25*.
Create	Open the program wizard to create a new project. See *Creating a project on page 27*.
Modify	Modify the current project with the program wizard.
Execute	Execute all exported path modules in the current project one by one.

NOTE! When the main menu of RobotWare Machining FC opens, only **FC Setup**, **Load** and **Create** are active. To activate **Modify** and **Execute**, you should load an existing project or create a new one to the current system.

3.1.2. Version and compatibility

Version information

The version information of RobotWare Machining FC is displayed at the lower-left corner of the main menu.

Compatibility

When you load a project to the system, check the RobotWare Machining FC version and the project version first.

Unmatched versions may cause the following problems:

Project file Version	**Product Version**	
	Previous Version	Current Version
Previous Version	OK	Save function does not work.
Current Version	Version incompatible message reported.	OK

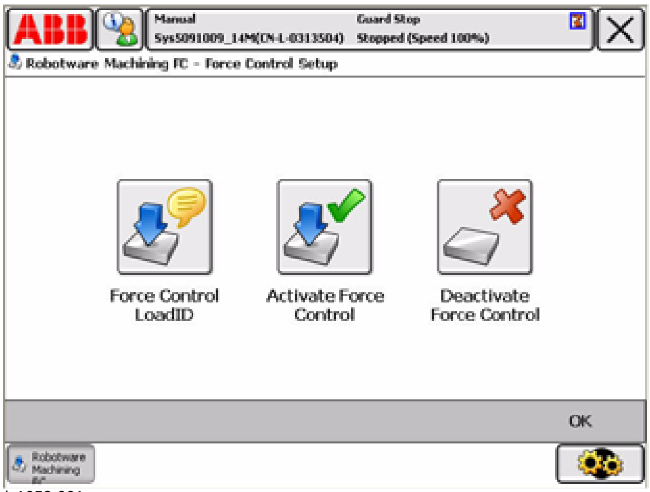
NOTE! Save function works only when the project version match with the product version.

3.2 Workflow for handling the graphical user interface

3.2.1. FC Setup

Force Control Setup page

In the main menu of RobotWare Machining FC, tap **FC Setup**, the **Force Control Setup** page opens.



pic1052-001

The Force Control Setup page consists of three parts:

- Force Control LoadID
- Activate Force Control
- Deactivate Force Control

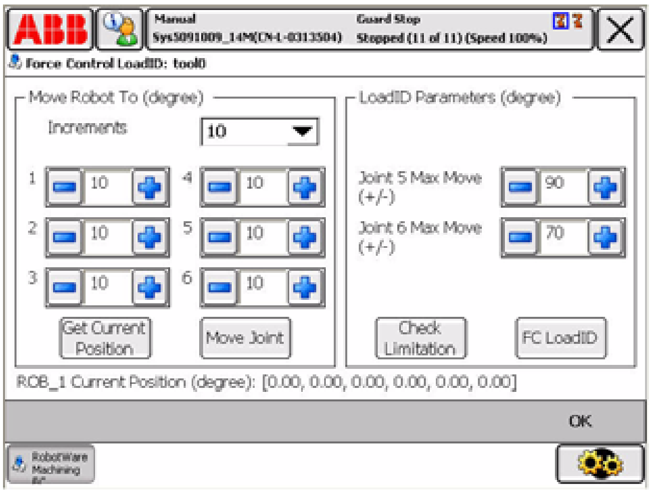
Continues on next page

Force Control LoadID

Force Control LoadID is a routine used to automatically identify payload of a tool using force sensor signal as input.

To open the LoadID views:

- In the Force Control Setup page, tap **Force Control LoadID**.



pic1052-002

Items	Action
Move Robot To	• To copy the current position value to the text box of each axis, tap **Get Current Position**. To set value for each axis, tap plus or minus. • To make the robot to go to the specified axis angles, tap **Move Joint**.
LoadID Parameters	To automatically calculate the maximum range of joints 5 and 6, tap **Check Limitation**. If the mounted tool or fixture is too large and there is the possibility of a collision, reduce the range of joints 5 and 6 by tapping plus or minus. To move and identify the loaddata for a specified tooldata, tap **FC LoadID**. **NOTE!** • Only joints 5 and 6 will move during the execution. • The result of Force Control LoadID will be stored in a loaddata named *Tool name_LD*.

NOTE!

Force Control LoadID is a very important step for all tools. It works in parallel with the standard LoadID, and will not replace the standard LoadID. If accurate tooldata is required for the standard robot motion, a standard LoadID is also required.

NOTE!

When set the movement range, make sure that:

- collision free,
- The movement range is reachable for the robot.

Continues on next page

Activate/Deactivate Force Control

Use this procedure to activate/deactivate force control to make sure that the force sensor works well:

1. In the **Force Control Setup** page, tap **Activate Force Control**.

   Force Control with tool0 and wobj0 is active, and the robot should respond to the external force applied to tools, fixtures, or parts mounted on the sensor.

2. In the **Force Control Setup** page, tap **Deactivate Force Control**.

   Force Control is inactive.

3.2.2. Loading a project

Overview

To work with a project, you should first have a project in the current system, then you can modify, execute or save the project.

You can choose to load an existing project or create a new project.

This section describes how to load a project to the current system.

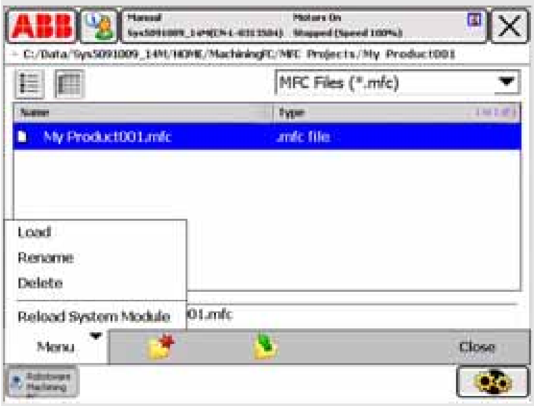
NOTE!

As default setting, the .mfc project and corresponding RAPID modules are loaded without deleting the current loaded RAPID program. The user modules (with different module names as the modules related to loaded .mfc project) in the program keeps still after loading .mfc project.

During loading, it is possible that errors are reported by RAPID syntax checking. The reason is that RAPID modules are loaded in sequence; some RAPID references can be invalid during loading process. In case that error occurs, check the program manually after loading, solve all errors and perform *PP movement*. Thus, the loaded program is ready to be modified or executed.

How to load a project

Use this procedure to load a project:

	Action	Note/illustration
1.	In the maim menu of RobotWare Machining FC, tap **Load**. The load project view opens.	The projects are saved in *\Home\Machining\MFC Projects* directory by default. **NOTE!** RobotWare Machining FC does not support to save /load a project to a folder not under system *\Home\Machining\MFC Projects* directory.
2.	Select a project from the list and tap **Menu**, and then tap **Load**.	 pic1051-002 The selected project will be loaded and set to be the current project. It is also possible to **Rename** or **Delete** a project from this view.

Continues on next page

How to restore system module

If the system modules are removed by mistake, it is possible to restore them.

Use this procedures:

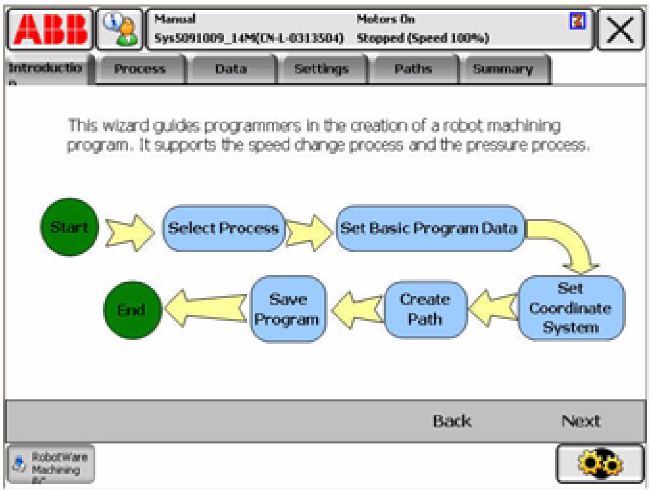
1. In the main menu of RobotWare Machining FC, tap **Load**.
2. Tap **Menu**, then tap **Reload system module**.

   System modules are stored in *Home\Machining\BaseLib* by default.

3.2.3. Creating a project

Introduction to the program wizard

In the main menu of RobotWare Machining FC, tap **Create** to open the program wizard.



pic1053-001

Wizard tab overview	Information
Introduction	Provides the wizard description.
Process	Two processes are supported in RW Machining FC: • Force Controlled Speed Change Process, for example, deburring. • Force Controlled Pressure Process, for example, grinding and polishing.
Data	Specifies the general data.
Settings	Defines tool data and work object data that will be used in the project.
Paths	Contains the following operations: • Add paths • Teach paths • Automatic learn • Export RAPID module • Test
Summary	All the paths are listed here.

Browse the program wizard

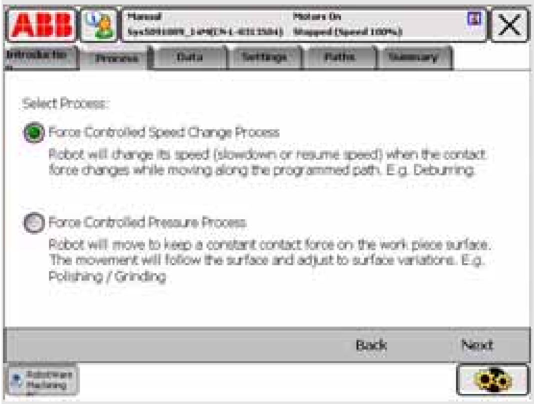
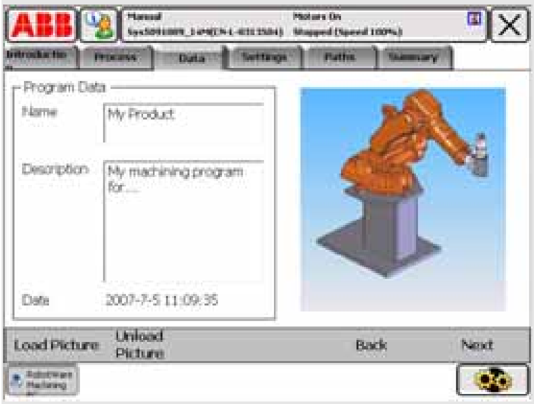
There are two ways to browse the wizard:

- Tap **Next** and **Back** that are displayed in the bottom of the wizard to browse the wizard step by step.
- Tap each tab on the top of the window to browse the wizard.

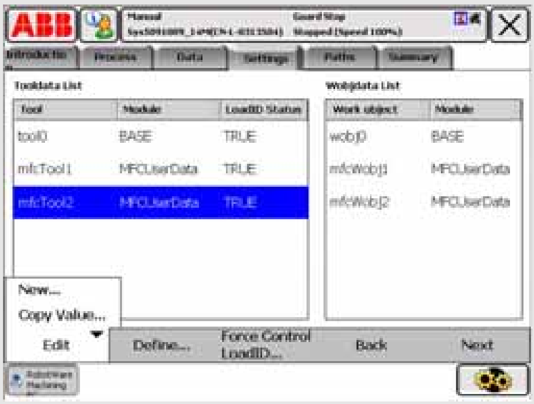
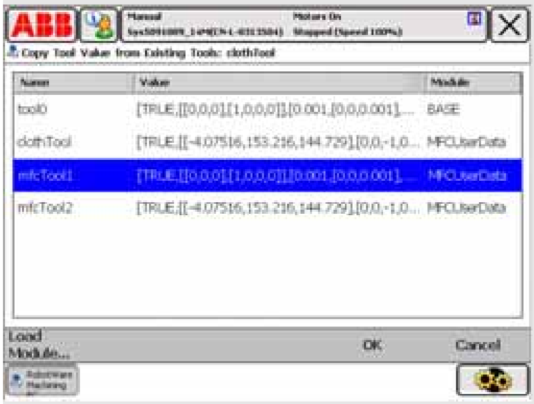
Continues on next page

How to create a project

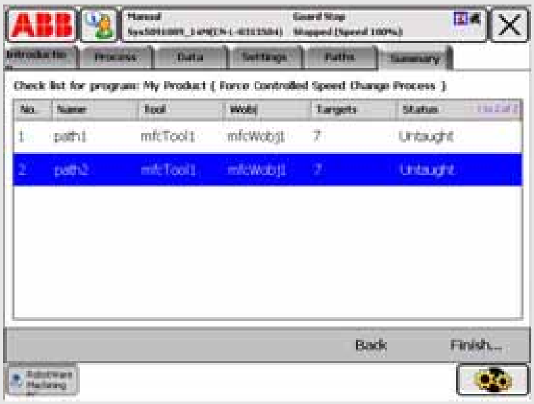
The table below details how to create a project.

	Action	Note/illustration
1.	To open the program wizard (which contains six tabs), tap **Create** in the main menu.	
2.	**Introduction** tab Read the description to familiarize yourself with the wizard. To enter the **Process** tab, tap **Next**.	
3.	**Process** tab Read the explanation of each process and select the relevant process according to the application. Tap **Next**.	 pic1053-002
4.	**Data** tab The general information of the project is set here. • Tap letters, numbers and special characters to type the project name and description using the soft keyboard, • Tap **Load Picture** and select a desired picture, or tap **Unload Picture** to use the default one. Tap **Next**.	 pic1053-003

Continues on next page

	Action	Note/illustration
5.	**Settings** tab - how to create new data. Define tooldata and wobjdata from this tab. • To add a set of new tooldata or wobjdata, tap **Edit** and then **New**. • To switch to define the tool or workobject coordinate system, tap **Define**. See *Operating manual - IRC5 with FlexPendant* for how to define tool or wobj frame. • To identify payload, tap **Force Control LoadID**. See *FC Setup on page 22*. **NOTE!** This step is important. Select a set of tooldata and tap **Edit**.	 pic1053-004
6.	**Settings** tab - how to use predefined data. It is also possible to reuse the existing data. • To add a set of new tooldata or wobjdata, tap **Edit** and then **New**. • To select data from the default data list, tap **Edit** and then **Copy value**. Or to load a module (which contains the desired data) to the system, tap **Load Module**. To confirm, tap **OK**. • To identify payload, tap **Force Control LoadID**. See *FC Setup on page 22*. To enter the Paths tab, tap **Next**.	 pic105-039

Continues on next page

	Action	Note/illustration
7.	**Paths** tab Create and edit paths from this tab. See *Paths on page 33* for detailed description.	
8.	**Summary** tab Check all the paths, and then tap **Finish**. A message box appears asking *Do you want to save the project?*. • To save the project and set it as the current, tap **Save/ Save as**. • To go back to the main menu without loading any project to the system and to discard all data, tap **Discard**. • To continue editing the project with the program wizard, tap **Cancel**.	 pic1053-006 **Note!** • If the project has been modified, the **Finish** command appears. • If the project has not been modified, the **Quit** command appears. To quit the program wizard without any changes, tap **Quit**. • If the project has been modified and the program wizard is closed by mistake before saving the currently editing project, the project will by default be saved as *Restore.mfc*. The project can be loaded from the load project view.

3.2.4. Modifying a project with the program wizard

Overview

Modifying a project is similar to creating a project since the program wizard is used in both cases.

How to modify a project

The table below describes how to modify a project.

	Action	Note/illustration
1.	Load a project to the system.	Save the current project before loading a new project. Otherwise all the changes to the current project will be lost.
2.	To open the program wizard, tap **Modify** in the main menu.	
3.	Modify the current project in the program wizard.	See *Creating a project on page 27* for detailed information.

TIP!

RobotWare Machining FC does not support to delete tool or wobj. If you want to delete a tool or work object, use Program Data.

See section *Tools* and section *Work objects* in *Operating manual - IRC5 with FlexPendant* for detailed description.

3.2.5. Executing

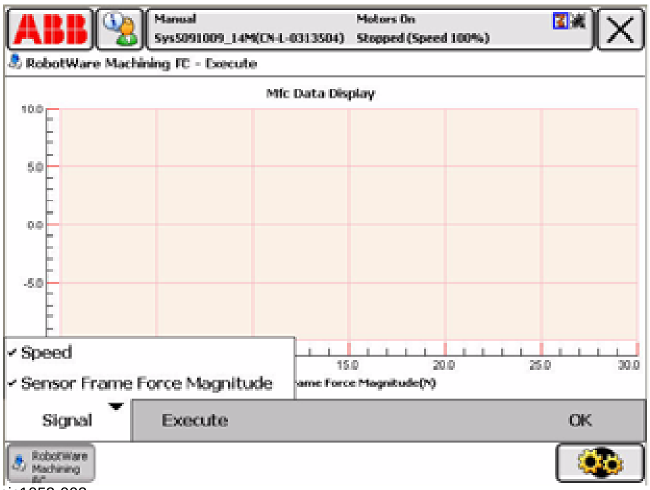
Overview

After exporting a path in the current project, the **Execute** command is available.

To open the Execute view:

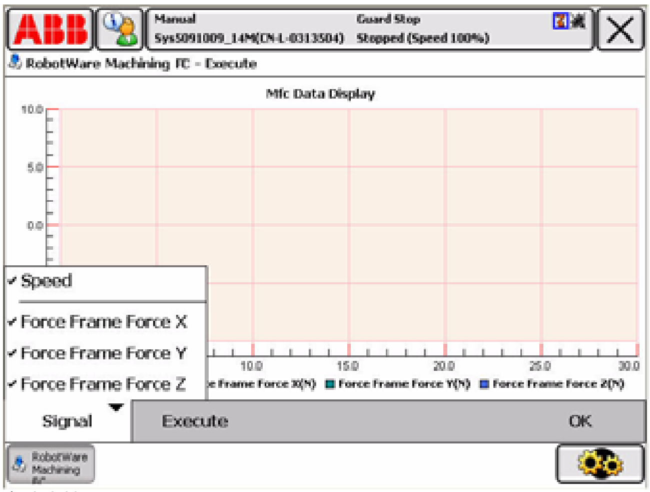
- In the RobotWare Machining FC main menu, tap **Execute**.

Force Control Speed Change Process



pic1052-003

Force Control Pressure Process



pic1052-004

How to proceed

All exported paths in the current project run one by one.

- Tap **Signal** and choose which signals to monitor during execution,
- To start the project, tap **Execute**.

During the execution the monitored signals are shown as dynamic curves on the FlexPendant.

3.3 Paths

3.3.1. Creating

Overview

Path describes the actual robot path to be programmed. The complete application defined in the project may consist of different paths which allow the process to be divided into modules (paths) in order to support more advanced programming use, for example, the use of tool changes.

How to create a new path

The table below describes how to create new paths.

	Action	Note/illustration
1.	From the **Paths** tab page, tap **Add**. A new path is added to the Path list.	 pic105-001 The status of the new path is *Untaught*. **NOTE!** When a path is created, seven targets are added to the path by default. They are: • 3 approach targets • 2 process targets • 2 withdraw targets
2.	To rename the new path, tap the name of the new path and then type the new name using the soft keyboard.	A path name can be used as an ID in the relevant modules and routines. **NOTE!** Every path must have a unique name in the current project.
3.	Tap **Tool** or **Wobj**, select the tool or wobj from the drop-down list.	The **Teach** button will not be available until the tool coordinate and the work object coordinate are set. **NOTE!** Only the tooldata that meets the following requirements can be seen from the drop-down list: • The tooldata is defined in BASE.sys module or MFCUserData.sys data module. • The tooldata has been identified with **Force Control LoadID**.
4.	To complete the operation, tap **Teach**, **Learn**, **Export** and **Test** in sequence.	See the following sections for detailed information: *Teaching on page 35*, *Learning on page 46*, *Exporting on page 50* and *Testing on page 53*.

Continues on next page

	Action	Note/illustration
5.	It is possible to create multi-paths. To create a new path, tap **Add**. Or select an existing path and then tap **Copy**. All values will be copied from the selected path.	The new path status will be degraded to *Taught* if the status of the path being copied is *Learned*, *Generated* or *Tested*.
6.	To delete the selected path, tap **Delete**.	

3.3.2. Teaching

Overview

This section contains the introduction to the teach view and the description of how to teach a path.

Warning

DANGER!

Never activate force control when the robot is in a collision or the tool is in contact with the surrounding environment. Otherwise activating force control may cause the robot to move or jump unpredictably. There is a high risk of injury or damage in this circumstance.

DANGER!

For safety reasons, follow the safety rules below when force control is activated:

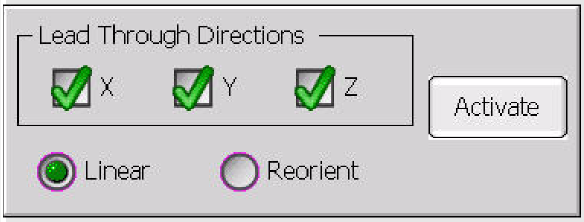
- Only the person holding the FlexPendant and activating the enabling switch is allowed to lead the robot by hand.
- The robot tool needs to be equipped with a handle if there is no normal way of grasping the tool when leading the robot by hand.
- The free space between the person and any fixed installation (wall, fence etc.) needs to be more than 0.5 m.
- The robot tool needs to be equipped with a dead-man's handle (including an enabling switch connected to the controller) if there is a risk of primary injury by grasping the tool directly (hot surfaces, sharp edges, chemicals etc.).

Teach view

The table below describes items in the teach view.

Items	Description
Insert Before	Create a new target before the currently selected target. The newly created target will be set as the selected.
Insert After	Create a new target after the currently selected target. The newly created target will be set as the selected.
Delete	Delete the currently selected target. Then the selected target will be set as none to avoid continuous deletion.
Modify Position	Modify the target with the current robot position. • Drag or jog the robot to the desired position. • Select a target and tap **Modify Position**. The current robot position is recorded and set as the target value. The next target will be set as the selected.
Go To	The robot will directly go to the selected target. **NOTE!** Make sure that it is collision free between the current target and the desired target. **NOTE!** Go To will automatically deactivate force control.

Continues on next page

Items	Description
Activate Force Control	 pic105-020 To enter a quick set panel from which the force control direction and motion mode can be set, tap **Activate Force Control**. To activate force control, tap **Activate**. Then it is possible to drag the robot. **NOTE!** The effort needed to move the robot in force control mode is determined by the damping parameters, see *Damping and LP-filter* and *Use FC Kinematics*. **NOTE!** It is also possible to jog the robot when force control is active. It is recommended to jog the robot for reorientation and use force control for leading the robot linearly.
Deactivate Force Control	Disable force control and set robot back to position control. In this state only the joystick can be used to move the robot.
Settings	Tap **Settings**, there will be: • **New Target Parameters**. Presents the target parameters page, which contains Set speed and zone for Speed Change Process, or the speed, zone, force, and damping to the FCPress1LStart instruction (see *FCPress1LStart*) for Pressure Process. • **Speed Change Process Parameters**/**Pressure Process Parameters**. These parameters will be used when exporting modules. The data type will be different according to the process type. See *How to set parameters on page 51*. • **Save 3D Path Picture**. This item appears only in the 3D view.

NOTE!

Do not touch the tool or work object mounted on the sensor immediately after tapping the **Activate** button and before the activation is properly executed. The activation will last for about 2 seconds. Otherwise the robot will drift towards the reverse direction of the applied force and the force control loaddata cannot be calibrated correctly.

DANGER!

If the robot drifts, release the enabling device immediately! **Activate Force Control** can be restarted later on by again pressing the enabling device and then tapping **Activate**.

Continues on next page

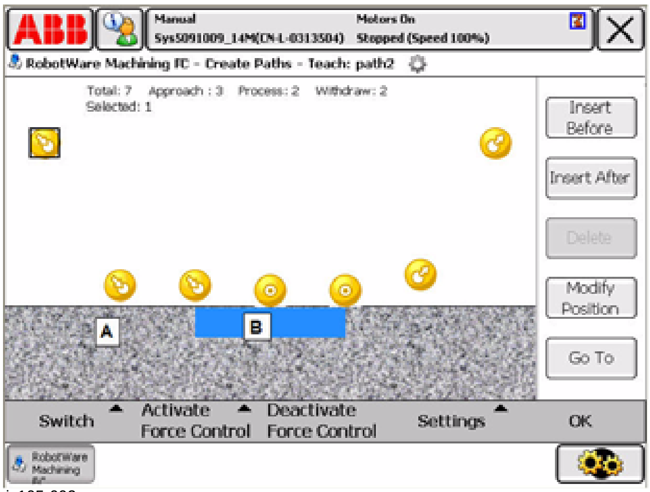
TIP!

One machining path can be showed or modified with any of the following three views, conceptual view, 3D view and target list view illustrated in the following sections. While the three views are synchronized based on the machining path, there are differences in accessed parameters with the three views.

- Conceptual view is defined to illustrate the idea of a machining path mode.
- 3D view is used to show all targets in a 3D environment which help you to check and view the path.
- With target list view, all parameters of individual target can be accessed and modified. The parameters includes: position to be fine tuned, speed, zone and force (if a pressure process is selected).

Conceptual view

The figure illustrates the Machining conceptual view.



pic105-002

A	The work piece.
B	The physical contact area.

Continues on next page

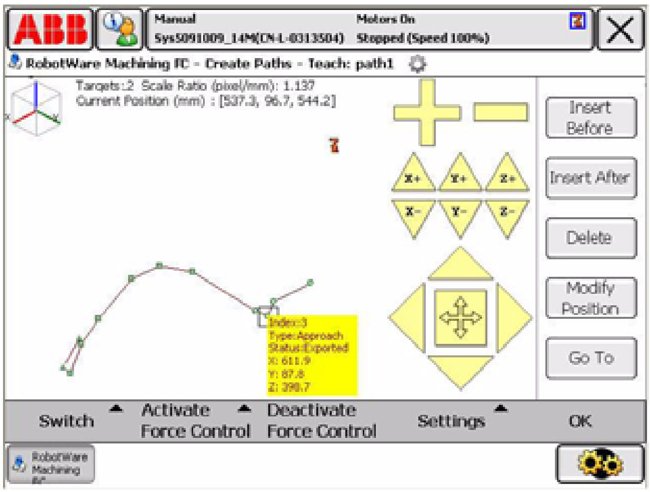
The following table contains different target types and status.

Target type	 pic105-007 A. Approach target B. Process target C. Withdraw target
Target status	 pic105-008 A. Untaught target B. Taught target C. Learned target D. Exported target E. Tested target

3D view

All targets are displayed in a 3D space with the work object coordinate.

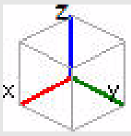
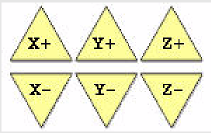
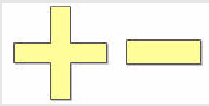
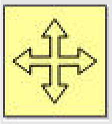
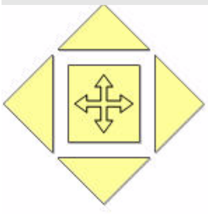
Tap a target from the view, and detailed information of the target appears as shown in the figure.



pic105-003

Continues on next page

The following table describes items in the 3D view.

Items	Description
 pic105-019	Tap this icon to switch to the desired view (X-Y view Y-Z view, X-Z view and ISO view in work object coordinate.
 pic105-016	Rotate the 3D view axis by axis. X, Y and Z are the axes shown above.
 pic105-015	Plus(+) and minus (-) symbols are used to zoom in and zoom out the path.
 pic105-017	This button is used to make the current picture fit the window.
 pic105-018	Arrows around the fit window button are used to pan the view display in related directions.

Continues on next page

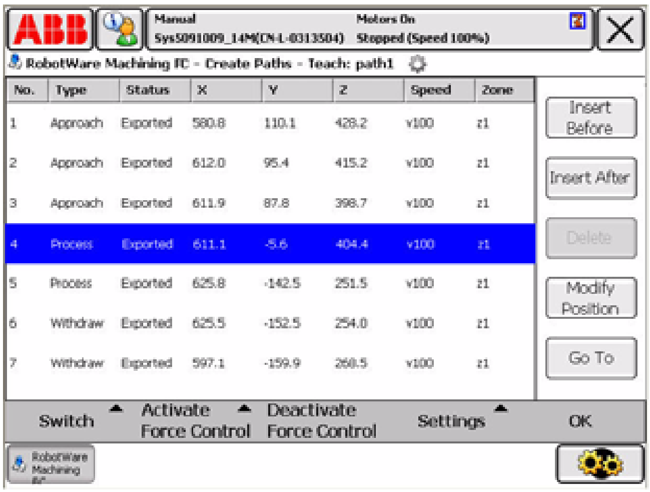
Continued

Target list view

All the detailed values of the targets are listed here.

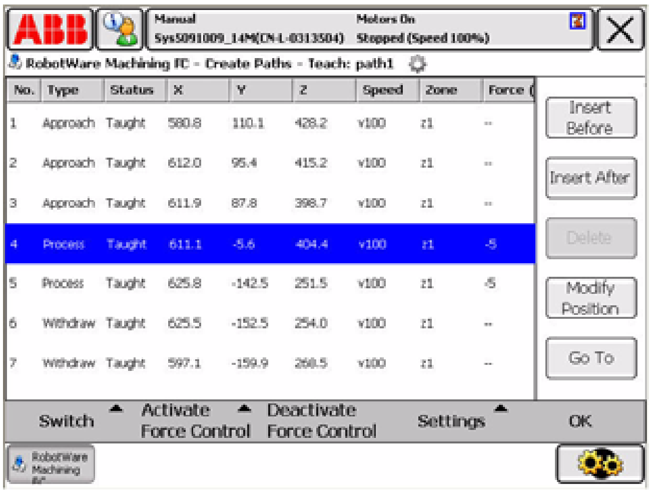
This view provides an easy way to tune the target coordinates (X, Y, Z), speed and zone. (If Pressure Process is the selected process, there will be one more data type: force.)

The following figure illustrates the point list view for the Speed Change Process.



pic105-004

The following figure illustrates the target list view for the Pressure Process.



pic105-030

Continues on next page

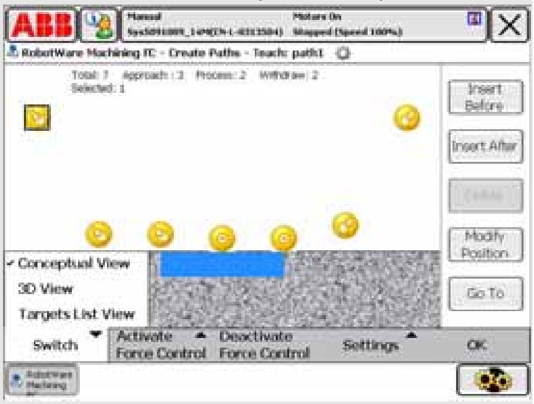
How to teach a path

Use this procedure to teach a path:

TIP!

Guidelines for teaching:

- Three approach targets and two withdraw targets.
- Less targets for straight line.
- More targets for the area with large curvature.
- A new target is always needed when the path direction is changed, for example, in a corner.
- No going back.
- No movement only in Z direction.
- Change orientation gradually.
- More targets do not necessarily mean better.

	Action	Note/illustration
1.	In the **Path** tab page, select a untaught path from the path list, then tap **Teach**. The Teach view opens.	Before teaching, make sure that you select the correct tool and work object for the path.  pic105-037 The conceptual view is the default view. To change to the other two views, tap **Switch**, then select the desired view.
2.	Tap a target. The selected target is highlighted in the view.	**Tips!** When modifying position for a target, we recommend to switch to the **Target List View** with which all parameters of individual target can be accessed and modified.
3.	Drag or jog the robot to the desired position. Or tap +/- to modify the target position from the target list view.	**NOTE!** Before dragging the robot, force control must be activated. See *Teach view on page 35* for how to activate or deactivate force control.
4.	Tap the **Modify Position** button.	The Robot's current position is recorded as the selected target's position, and the status of this target turns to **Taught**.
5.	Repeat step 2-4 for other targets.	

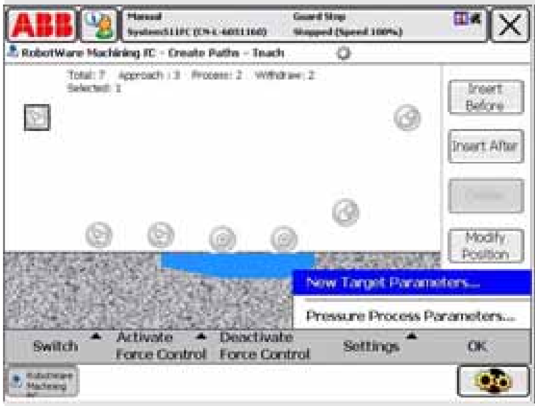
Continues on next page

	Action	Note/illustration
6.	If you want to add more targets to the path, tap a target then tap **Insert Before** or **Insert After**. A new target is added before or after the selected target.	The new added target includes default speed, zone and force values. The following section details how to set the default parameters for the new added target by using **Settings**. Tap **Settings**, • New target parameters, see the following section. • Process parameters, see *How to set parameters on page 51*. These default parameters for individual target can also be modified from the target list view.
7.	Repeat step 5 until you add enough targets. Then repeat step 2-4 to teach target.	If you want to delete a target, you can tap a target then tap **Delete**.
8.	When all the targets in the path are taught, tap **OK** to go back to the path list.	The status of the path is changed to **Taught**. When the paths are taught, **Learn** and **Export** are activated.

How to define default parameters for a new added target

New target parameters are the default parameters for new added targets. The default parameters include speed, zone, and force.

Use this procedure:

	Action	Note/illustration
1.	In **Teach** view, tap **Settings** then **New Target Parameters...**.	 pic105-049

Continues on next page

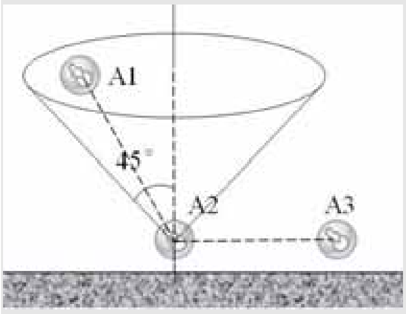
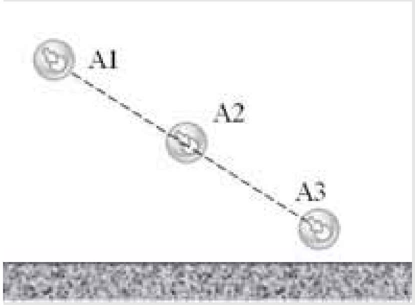
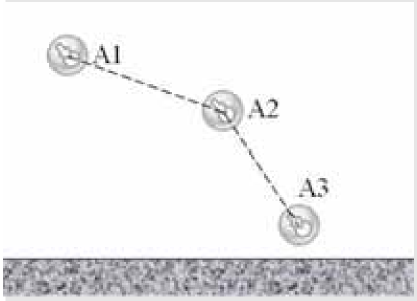
	Action	Note/illustration
2.	In this view: • Tap a value from the drop down list for **Speed**. • Tap a value from the drop down list for **Zone**. • Tap +/- to set a value for **Force (N)**	 pic105-050 **NOTE!** When the new added targets are placed in either approach or withdraw sub-paths, or the **Speed Change Process** is selected, the force parameter does not take effect for the target.
3.	Tap **OK** to make the changes take effect and go back to the teach view.	The default parameters of new added target change to the values you defined in step 2.

Continues on next page

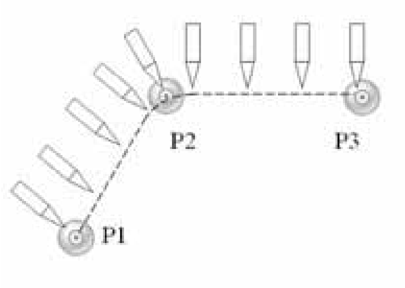
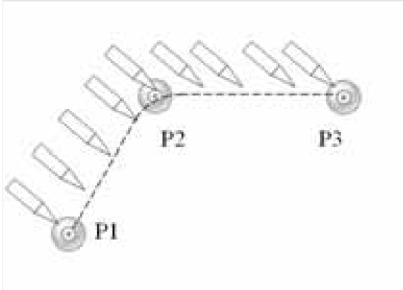
Key targets of teaching

There are some special targets illustrated below. Teach these targets according to the recommended patterns in order to support automatic path learning. See *Learning on page 46*.

Key targets of teaching are described below.

Key target	Recommended patterns	Incorrect patterns
Approach targets	The last three approach targets are illustrated below.  pic105-014 Follow the rules when teaching these targets. 1. The angle between A1A2 and the normal direction of the surface should be less than 45º. It is recommended to make A1A2 perpendicular to the surface. 2. The line A2A3 must be almost parallel to the work piece surface. 3. A2 and A3 must be very close to the surface but without contact. **NOTE!** In case where the above conditions cannot be satisfied for practical reasons, the default force direction may be incorrect.	 pic105-042  pic105-043

Continues on next page

Key target	Recommended patterns	Incorrect patterns
Process targets	 pic105-036 When teaching a path as shown in the above picture, the natural force frame is tool frame during exporting. It is recommended to use tool frame as reference frame and do the following during teaching: • At target P2, rotate the robot by hands or use the joystick to make sure that the force direction is perpendicular to the path.	 pic105-044

3.3.3. Learning

Overview

Learning is an automatic process. During learning:

- The robot will follow the previously taught path.
- The robot will move along the work object with specified force and speed.
- A series of targets will be recorded from the first contact target to the last process target. The first contact target is a target which is between the last *approach target* and the first *process target*.

After learning, an accurate path which closely fits the contour will be created.

There are three types of targets in a learned path:

- Approach targets
- Learned targets
- Withdraw targets

NOTE!

Do not stop learning in the middle. Otherwise you should restarted learning from the beginning.

How to learn a path

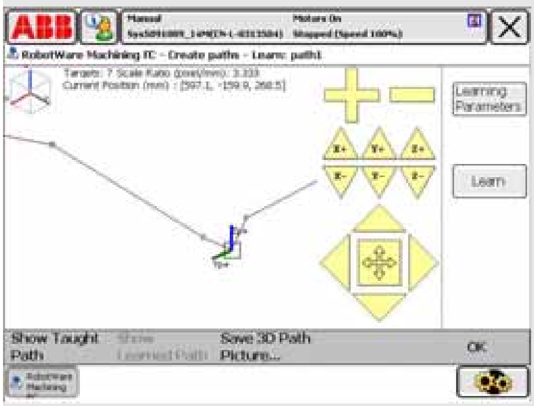
Use this procedure to learn a path:

TIP!

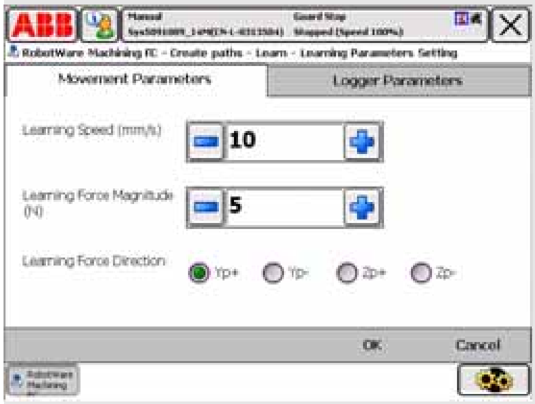
During learning process, the tool held by robot starts to approach to the work piece immediately after passing the last approach point. Make sure that the last approach point is taught above the work piece surface, otherwise the learning can not be correctly performed.

NOTE!

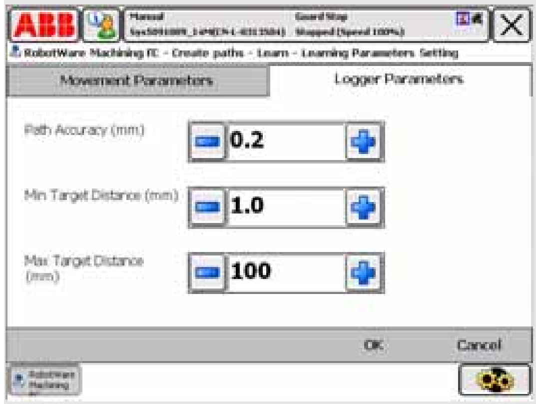
When learning a path, make sure that the z direction of the tooldata is *not* parallel with the forward direction of the path. Especially for learning a circle path the z direction of the tooldata must be parallel with the normal direction of circle surface, that is, perpendicular to the circle surface to avoid that the tooldata z direction is parallel with the path.

	Action	Note/illustration
1.	To open the Learn view, tap a path from the paths list and then tap **Learn**.	 pic105-005

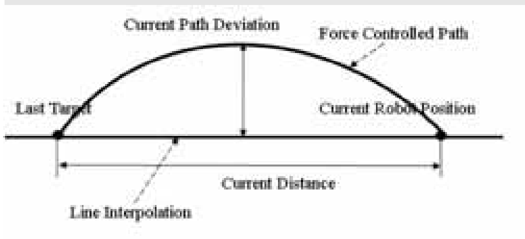
Continues on next page

<table>
<tr><th></th><th>Action</th><th>Note/illustration</th></tr>
<tr><td>2.</td><td>Set Learning Parameters.</td><td>Both the Moving Parameters and Logger Parameters should be set.</td></tr>
<tr><td>3.</td><td>The positive learning force direction can be seen from the 3D View. To choose the direction to apply force, select Yp+, Yp-, Zp+ or Zp-.</td><td>


pic105-021

If the approach targets are chosen correctly as described in Key targets of teaching on page 44, the Learning Force Direction will be correct by default.</td></tr>
<tr><td>4.</td><td>Set logger parameters from this tab.</td><td>


pic105-022



pic105-033

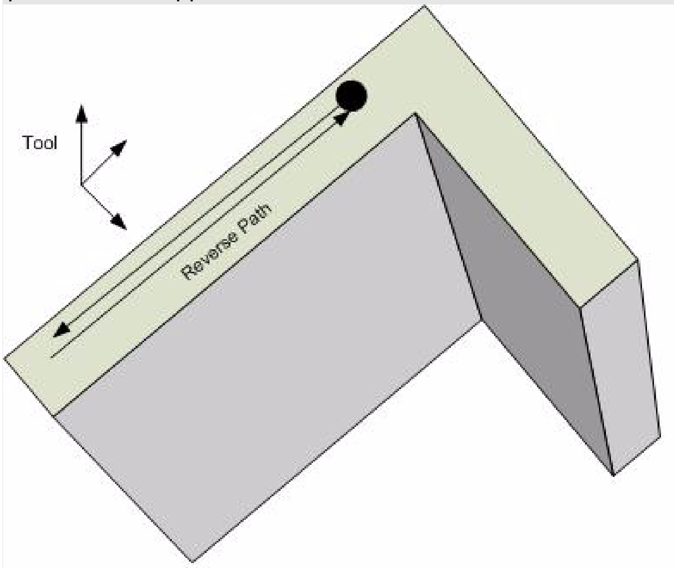
The current robot position will be recorded as a new target when:

Current path deviation is greater than or equal to the Path Accuracy,
Current distance is greater than or equal to Min Target Distance,
Current distance is equal to the Max Target Distance.
</td></tr>
</table>

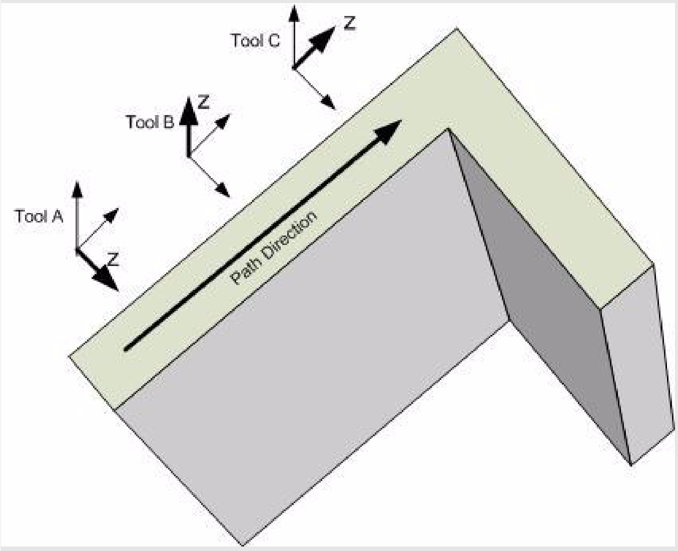
Continues on next page

	Action	Note/illustration
5.	To generate a learned path, tap **Learn**.	
6.	To compare the two paths, tap **Show Taught Path** or **Show Learned Path**	
7.	To go back to the path list, Tap **OK**.	

Special learning

	Description
Learning a reverse direction path	Learning a reverse direction path as shown in the following picture is not supported.  pic105-047

Continues on next page

	Description
Learning along tool Z direction	When creating a project, you need to teach and learn a path with the defined direction tool. In the learning scenario shown in the following picture, the system can learn in the case where with the Z direction of Tool A and Tool B, but the system can not learn in the case where with the Z direction of Tool C.  pic105-048

3.3.4. Exporting

Overview

The path can be generated as a RAPID module by using **Export**.

NOTE!

RobotWare Machining FC does not support automatically renaming an exported path module according to the new input path.You should manually delete the exported path module if a new export operation of the path needs to be performed, otherwise the ambiguous reference name error is reported due to some RAPID data are named same in the previous and new exported path module.

TIP!

If a machining path of pressure process is exported, the tool held by robot starts to approach to the work piece immediately after passing the last approach point. Make sure that the last approach point is located above the work piece surface, otherwise the production with this path can not be correctly performed.

How to export a taught path

The following table describes how to export a taught path.

	Action	Note/illustration
1.	To open the Export view, select a taught path and then tap **Export**.	 pic105-027
2.	Set parameters before export.	See *How to set parameters on page 51*.
3.	To generate the RAPID module, Tap **Export**. (Name the module *mfcPath?.mod*. "?" represents the sequence number of the path).	If a path is taught but not learned it will output a module with all the taught targets. If a path is learned it will output a module with the learned targets.

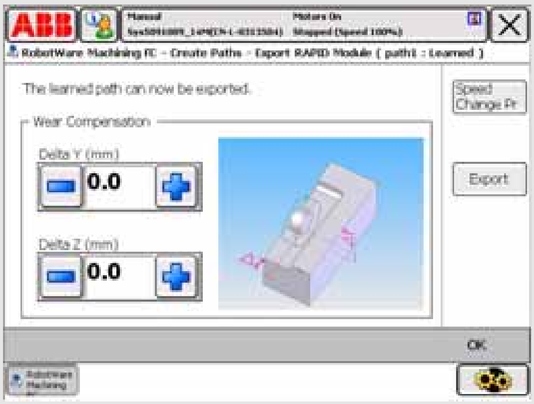
NOTE!

You should not modify the exported module file itself. The RAPID module will be overwritten next time the taught or learned path is exported.

Continues on next page

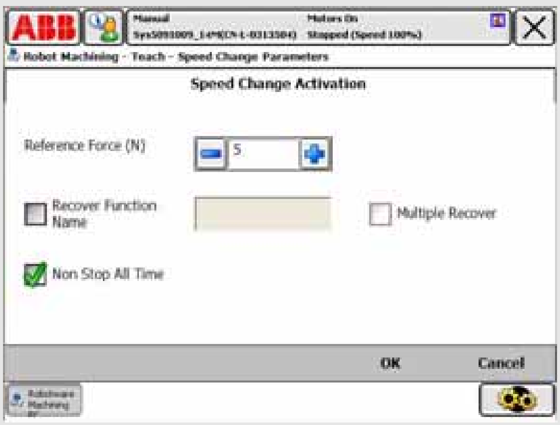
How to export a learned path

The following table describes how to export a learned path.

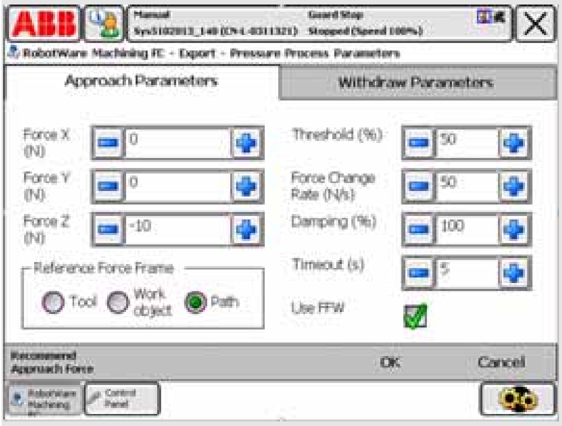
	Action	Note/illustration
1.	To open the Export view, select a learned path and then tap **Export**.	 pic105-006
2.	Set parameters before export.	See *How to set parameters on page 51*.
3.	A path offset may be included before exporting for wear compensation.	A learned path can be offset in Y or Z direction of Path Frame to compensate the wear of abrasive.
4.	To generate the RAPID module, Tap **Export**.	This module can be viewed from the Program Editor.

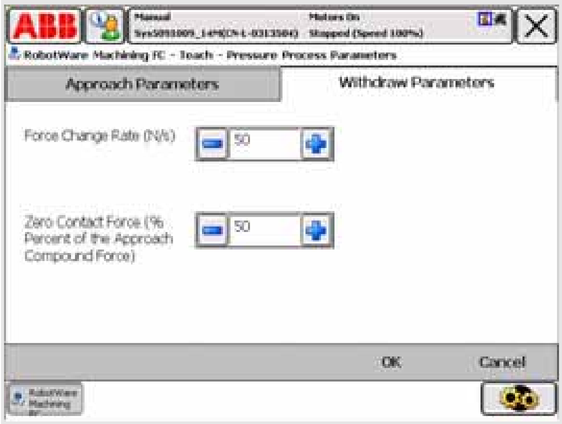
How to set parameters

For the different processes there are the following parameter windows:

Process	Note/illustration
Speed Change Process See *FCSpdChgAct on page 105* for reference.	 pic105-023

Continues on next page

<table>
<tr><th>Process</th><th>Note/illustration</th></tr>
<tr><td>Pressure Process
See *FCPress1LStart on page 93* and *FCPressEnd on page 102* for reference.
To get the recommended ratio among components of the approach force, tap **Recommend Approach Force**. A message box will appear.</td><td>

pic105-024
Exporting module with path coordinate system is supported.


pic105-025
NOTE!
The recommended solution will achieve the most desirable effects if the last third and the last second targets are set normal to the surface.</td></tr>
</table>

3.3.5. Testing

Overview

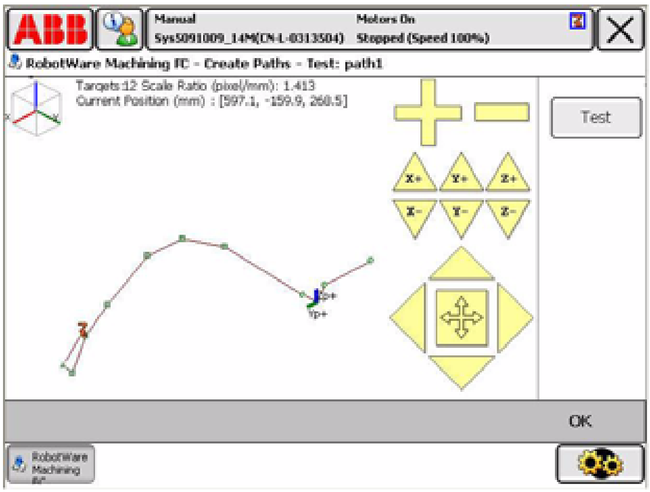
In the test view the exported path can be run and viewed.

To open the test view:

- In the **Paths** tab, tap **Test**.

If this path is not satisfactory, go to the teaching view to tune the targets again. Before running, implement the two RAPID routines *SpindleOn* and *SpindleOff* in MFCUserData.sys.

If the process is a Speed Change Process and a user-defined recovery function has been selected, the recovery function also needs to be implemented.



pic105-040

NOTE!

Before running the **Speed Change** project define **Feedback Type** correctly (see below). Otherwise the robot will not give any response when the force increases over the defined speed change force.

- When using 6DOF force sensor the Feedback Type should be Calib. Force Magn.
- When using 1DOF force sensor the Feedback Type should be UnCalib. Force Magn.
- When using no force sensor the Feedback Type should be Single DAC Input.

How to proceed

The following procedure details how to define **Feedback Type**:

1. On the ABB main menu, tap **Control Panel**.
2. Tap **Configuration**.
3. From the **Topics** list, select **Motion**.
4. To show detailed information, tap **FC Speed Change** twice.
5. To edit parameters, either tap *fc_speed_change1* twice or tap *fc_speed_change1* and then tap **Edit**.
6. Tap **Feedback Type** twice and select a desired type from the drop-down list. To confirm, tap **OK**. A warm start is required.

After warm start the changes will take effect.

Continues on next page

NOTE!

If an event message *TCP speed too high* appears during the testing, a probable cause is that the supervision speed is set too low. Decrease the programmed speed or increase the maximum reference value for the speed. To come to the view where to modify the speed, tap in the following order from the ABB menu: **Control Panel**, **Configuration**, **Topics**, **Motion**, **FC Speed Change**, *fc_speed change*, and **Maximum TCP Speed**.

3.4 Hints for handling the graphical user interface

3.4.1. Summary - differences between Speed Change Process and Pressure Process

Overview

There are two processes supported in RobotWare Machining FC, Force Control speed change process and Pressure Process.

The two processes share the same program wizard but there are still some differences:

- Target list view, see *Target list view on page 40*.
- New target parameters, see *How to define default parameters for a new added target on page 42*.
- Process parameters, see *How to set parameters on page 51*.

3.4.2. Improving pressure process quality

Contact between tool and work piece

To keep tool in continuous contact with work piece is vital important to good learning result and good production quality with the pressure process. With the force control technology to apply pressure, the tool follows the changes of work piece surface and try to keep constant pressure during the movement.

However, in real force control applications, when using pressure process with relative high movement speed on a highly curved work piece surface, it is possible that the contact between tool and work piece is not constantly kept.

Use the following two methods to ensure constantly contact:

- Reduce the movement speed during pressure process. To reduce speed is not a good solution in most cases, because the cycle time is a key factor to high productivity.
- Tune damping parameters. See *Damping in Force z Direction on page 143* for the definition of damping parameters. In practice, when higher damping value is used, the less sensitivity robot responses to force change. While it is *NOT* always a good choice to use lower damping settings, the reason is that robot could to be too sensitive to keep stable contact pressure. To choose a good damping parameter for a specific application requires experiences and sometimes experiments.

Set damping value

See *How to set damping value in force control process on page 58* for the detailed description.

3.4.3. Damping

Description

Damping is a definition of how large contact force is required for the robot to move at a certain speed. Defines how many Newtons are required to make the robot move 1 m/s. The higher the value, the less responsive the robot gets.

In Force Control, a contact force will make the TCP move with a speed proportional to the contact force. A contact torque will make the tool reorient with a speed proportion to the contact torque. The damping variable defines the proportions between a force and the resulting speed, and a torque and the resulting reorientation speed, in the direction x, y and z. The values are given as a percentage of the of the system parameter value defined in the type FC Kinematics.

You can set different damping values for direction x y and z. But for Force Controlled Pressure Process, always use Damping in Z direction even if the force is set to X or Y direction.

Components

To configure the damping value, the following need to be considered:

	Description
Damping in the x direction	Force damping (relation between force and TCP speed) in the x direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.
Damping in the y direction	Force damping (relation between force and TCP speed) in the y direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.
Damping in the z direction	Force damping (relation between force and TCP speed) in the y direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.
Damping in the rotational x direction	Torque damping (relation between torque and tool reorientation speed) around the x direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.
Damping in the rotational y direction	Torque damping (relation between torque and tool reorientation speed) around the y direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.
Damping in the rotational z direction	Torque damping (relation between torque and tool reorientation speed) around the z direction. A smaller value means that the robot is more sensitive. By default the value is 100% (of the system parameter value), and allowed values are between 50% and infinity.

After setting the new damping value, warm start the FlexPendant to make the setting take effect.

Continues on next page

DampingTune

Tuning of force and torque damping.

The DampingTune value can be used to modify the relation between the sensed force and the generated velocity in each direction. By default the value is 100% (of system parameter values) in all direction, but it can be between 50% and infinity. Smaller value means that the robot is more sensitive to external forces.

See *FCPress1LStart on page 93* for reference when setting the DampingTune value.

After the DampingTune value is modified, when the robot is in force control mode and no external force except gravity are present, it should not move. If the robot drifts away by itself or it vibrates, increase the damping value.

How to set damping value in force control process

The damping parameters used in a specific pressure process can be tuned with both setting damping parameter in system configuration, and tuning DampingTune argument with RAPID instructions.

See *Damping in Force z Direction on page 143* for how to set damping parameters with system parameters.

See *FCPress1LStart on page 93* for how to tune damping parameters with RAPID instructions.

3.4.4. Programming in MultiMove System

Overview

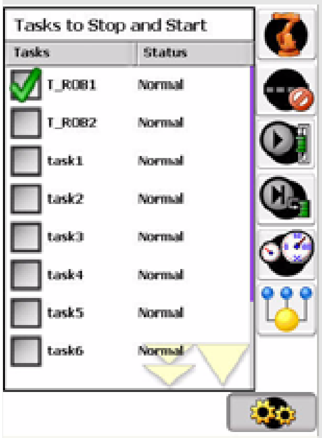
This section describes how to configure RW Machining FC to support Single Robot FC in the MultiMove System.

NOTE!

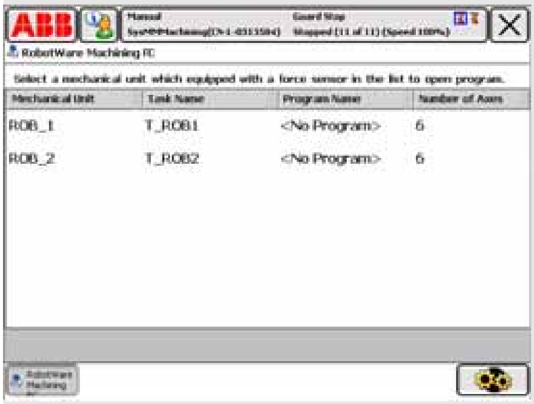
To work in a MultiMove System, make sure the controller key contains the option MultiMove Independent.

How to program in a MultiMove System

Download a system to the controller, then follow the following procedure.

	Action	Note/illustration
1.	Load force sensor parameters. Use RobotStudio.	See *Operating manual - RobotStudio* for how to load parameters.
2.	Configure the force sensor parameters: • Use FC Master • Use PMC Master.	These two parameters can be configured. From the ABB main menu tap **Control Panel**, **Configuration**, **Topics**, **Motion**, *ROB_1* (or *ROB_2*,...), and then select **Use FC Master** or **Use PMC Master**.
3.	Make a warm start.	
4.	Disable all other tasks: • Tap **QuickSet** and then **Multitasking**. • Select or clear the check boxes to enable or disable tasks.	 pic105-046 **NOTE!** In a MultiMove System make sure that the program pointers are available (Move PP to main or other routine) for all enabled tasks except the task that is relative to the selected mechanical unit. Otherwise some functions cannot be run.

Continues on next page

	Action	Note/illustration
5.	On the ABB main menu, tap **RobotWare Machining FC**. The mechanical unit selection view appears.	 pic105-045
6.	Tap a mechanical unit. The RobotWare Machining FC main menu automatically appears and the selected unit will be used in the later operation.	
7.	Do the normal operation as in the single robot system.	Programming in a MultiMove System is almost the same as programming in a single robot system. See detailed information previously described.

3.4.5. Working with RobotStudio Machining PowerPac

Overview

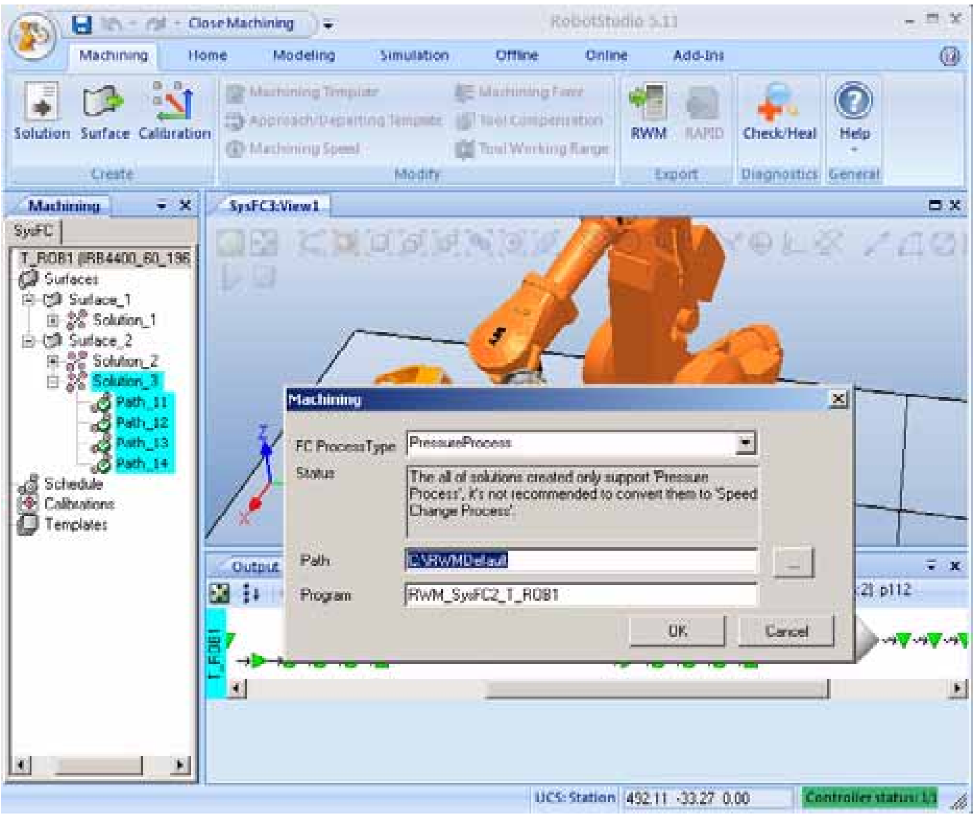
The RobotWare Machining FC can use the project file generated from **RobotStudio Machining PowerPac**, which is an add-in for machining based on RobotStudio. The RobotStudio Machining PowerPac can generate machining paths based on 3D model of the real part. The generated paths can be exported into RobotWare Machining FC project file format.

RobotStudio Machining PowerPac

RobotStudio Machining PowerPac is a dedicated off-line programming tool for machining as a complement to online programming for RobotWare Machining FC and the related graphical user interface used for online programming of force controlled machining applications.

You can create paths on a complex surface of a part directly by using several patterns in RobotStudio Machining PowerPac. Also you are able to adjust the generated targets in this PowerPac. Then you can export all of the targets as a file with specific format to RobotWare Machining FC. Two types of FC process are supported, *Pressure Process* and *Speed Change Process*.

The graphical user interface of RobotStudio Machining PowerPac is shown in the following picture.



pic1055-001

Continues on next page

How to work with RobotStudio Machining PowerPac

	Action	Note/illustration
1.	Export project file from the RobotStudio Machining PowerPac.	See *Operating manual - RobotStudio Machining PowerPac* for detailed description. **NOTE!** All generated targets from RobotStudio Machining PowerPac are taught targets.
2.	Copy the generated project file into controller MFC project folder.	The generated project folder should be copied into *\HOME\MachiningFC\MFC Projects* directory of the current robot system folder.
3.	Load the project file to RobotWare Machining FC.	A learning process based on generated paths can be performed when 3D model of the real part is not accurate enough. All the RobotWare Machining FC supported function can be performed.

4 Configuration

4.1. System parameters overview

About this section

This is an overview of the system parameters used in Force Control. For more information, see the respective parameter in the chapter *System parameter reference information*.

NOTE!

Parameters not listed here but visible in RobotStudio, are only used in the option Assembly FC.

Robot

These parameters belong to the type *Robot* in the topic *Motion*.

Parameter	Description
Use FC Master	Specifies which *FC Master* to use.
Use PMC Sensor	Specifies which *PMC Sensor* to use.

FC Master

These parameters belong to the type *FC Master* in the topic *Motion*.

Parameter	Description
Name	The name of the *FC Master*.
Use FC Sensor	Specifies which *FC Sensor* to use.
Use FC Kinematics	Specifies which *FC Kinematics* to use.
Use FC Application	Specifies which *FC Application* to use.
Use FC Speed Change	Specifies which *FC Speed Change* to use.

FC Sensor

These parameters belong to the type *FC Sensor* in the topic *Motion*.

Parameter	Description
Name	The name of the *FC Sensor*.
Force Sensor Mount Unit Name	The name of the mechanical unit the sensor is mounted on. Most often ROB_1. If the sensor is room fixed the value should be empty.
Force Sensor Type	Specifies if the sensor should measure both force and torque or only force.
Noise level	The highest noise level at which a force sensor calibration is allowed. Used to check that the robot is standing still.
Force Sensor Frame x - z	The origin of the sensor coordinate system, expressed in the tool0 coordinate system when the sensor is mounted on the robot, or expressed in the world coordinate system when the sensor is room fixed. The unit is meter.

Continues on next page

Continued

Parameter	Description
Force Sensor Frame q1 - q4	The orientation of the sensor coordinate system in relation to the tool0 coordinate system when the sensor is mounted on the robot, or to the world coordinate system when the sensor is room fixed.

FC Kinematics

These parameters belong to the type *FC Kinematics* in the topic *Motion*.

Parameter	Description
Name	The name of the *FC Kinematics*.
Damping in Force z Direction	Specifies how the robot speed depends on the contact force. A higher value makes the robot less sensitive to contact forces. The unit is Ns/m.
Bandwidth of force loop filter	Specifies the behavior of the force control loop. A higher value makes the robot more compliant but can cause instability. The unit is Hz.
Bandwidth of force frame filter	The force measured in "force frame" will be low pass filtered with this bandwidth. The unit is Hz.

FC Application

These parameters belong to the type *FC Application* in the topic *Motion*.

Parameter	Description
Name	The name of the *FC Application*.
Max Ref Force	Maximum allowed reference force. The unit is N.
Max Ref Force Change	Maximum allowed change in reference force. The unit is N/s.
Largest measured contact force	If measured contact force is larger than this value it is set to this value.
Lowest measured contact force	A measured contact force lower than this value will be set to zero [dead band]. The unit is N.
Max Press TCP Speed	Maximum lin speed in press movements. The unit is m/s.
Max Press Rot Speed	Maximum rot speed in press movements. The unit is rad/s.

NOTE!

The values of the *Max Ref xxx* parameters define the ramping step of the reference movement. If the parameter value is set too high, the ramping will produce jerks and trig the speed supervision.

NOTE!

If the parameters *Lowest measured contact force* and *Lowest measured contact torque* are set too low, there is a risk that the robot will drift when in force control mode.

Continues on next page

FC Speed Change

These parameters belong to the type *FC Speed Change* in the topic *Motion*.

Parameter	Description
Name	Defines the name of the *FC Speed Change*.
Speed ratio min	Specifies the minimum allowed speed ratio.
No of speed levels	Defines the number of speed levels.
Speed ratio delta	Limits acceleration/deceleration due to the SpeedChange functionality. A low value will give slower but smoother speed changes. Too high value of Speed_ratio_delta will result in jerky behavior.
Speed max update period	Specifies the minimum time in seconds between speed changes.
Feedback type	Defines the type of feedback to be used for speed change control.
Feedback offset	Specifies the offset to be subtracted from the measured feedback before the value is used in speed change control.
Use Fdb LP filter	Defines whether feedback low pass filter should be active.
Fdb LP filter bandwidth	Defines the bandwidth of the feedback low pass filter (Hz).
Maximum TCP speed	Defines the maximum original TCP speed for speed change. The unit is m/s.
Recover rule fdb ratio	A feedback to reference ratio larger than this while having reduced speed to lowest level will trig recover behavior or stop robot.
Decrease rule safety fdb ratio	Defines the maximum feedback to reference ratio.
Decrease rule safety fdb time	Define the maximum time in seconds that the feedback to reference ratio can be continuously over "Decrease rule safety fdb ratio" before reducing robot speed.
Fdb trend step size	Defines the minimum difference between two consecutive feedback values to count as a change in feedback.
Decrease rule 1 fdb ratio	For ABB internal use only.
Decrease rule 1 fdb trend	For ABB internal use only.
Decrease rule 2 fdb ratio	For ABB internal use only.
Decrease rule 2 fdb trend	For ABB internal use only.
Increase rule 1 fdb ratio	For ABB internal use only.
Increase rule 1 fdb trend	For ABB internal use only.
Increase rule 2 fdb ratio	For ABB internal use only.
Increase rule 2 fdb trend	For ABB internal use only.

PMC Sensor

These parameters belong to the type *PMC Sensor* in the topic *Motion*.

Parameter	Description
Name	The name of the *PMC Sensor*.
Use PMC Sensor Setup	Specifies which *PMC Sensor Setup* to use.

Continues on next page

Continued

NOTE!

PMC Sensor *Name* must be the same as FC Sensor *Name* when Force Control is used.

PMC Sensor Setup

These parameters belong to the type *PMC Sensor Setup* in the topic *Motion* and are supplied by the sensor manufacturer.

Parameter	Description
Name	The name of the *PMC Sensor Setup*.
fx 1 - fx 6	Specifies the contribution from each of the sensor’s input voltages to the force measurement in the x direction.
fy 1 - fy 6	Specifies the contribution from each of the sensor’s input voltages to the force measurement in the y direction.
fz 1 - fz 6	Specifies the contribution from each of the sensor’s input voltages to the force measurement in the z direction.
tx 1 - tx 6	Specifies the contribution from each of the sensor’s input voltages to the torque measurement in the x direction.
ty 1 - ty 6	Specifies the contribution from each of the sensor’s input voltages to the torque measurement in the y direction.
tz 1 - tz 6	Specifies the contribution from each of the sensor’s input voltages to the torque measurement in the z direction.
fx scale - tz scale	Scaling factors to transform the values of the input voltages into force and torque values.
fx max - tz max	The sensor’s specified maximum force or torque in the respective direction.
Max voltage for external AD card	The value should be set to the voltage working range of the A/D board. The range is assumed to be + - this value.
Disable force sensor cable check	Some force sensors have a separate safety channel, which delivers a voltage above a certain level when the sensor is working, thus enabling system supervision. If the force sensor in use does not have a safety channel this supervision must be disabled.
Safety channel voltage	Safety channel voltage level.
Last node	Specifies the last node used on the PMC sensor.

NOTE!

It is very important that the *PMC Sensor Setup* values are correct, otherwise load identification and calibration will be incorrect. When running the function `FcLoadId`, the argument `LoadidErr` will give a strong indication it the values are set correctly.

NOTE!

The +/- sign of the calibration parameters needs to be chosen so that the resulting force and torque is the force/torque by which the surrounding effects the robot, not the other way round.

4.2. Configuration example

Overview

This section shows a real configuration example, intended to facilitate the setup of the configuration parameters. Some parameters are sensor specific, but others can be copied, like maximum allowed values.

NOTE!

Parameters not listed here but visible in RobotStudio, are only used in the option Assembly FC.

Robot

Parameter	Value	Unit / Note
Use FC Master	fc_master1	-
Use PMC Sensor	fc_sensor1	-

FC Master

Parameter	Value	Unit / Note
Name	fc_master1	-
Use FC Sensor	fc_sensor1	-
Use FC Kinematics	fc_kinematics1	-
Use FC Application	fc_application1	-
Use FC Speed Change	fc_speed_change1	-

FC Sensor

Parameter	Value	Unit / Note
Name	fc_sensor1	-
Force Sensor Type	Force and Torque	6DOF
Mount Unit Name	ROB_1	Sensor mounted on robot 1
Noise level	25	-
Force Sensor Frame x	0	meter
Force Sensor Frame y	0	meter
Force Sensor Frame z	0.05	meter
Force Sensor Frame q1	1	-
Force Sensor Frame q2	0	-
Force Sensor Frame q3	0	-
Force Sensor Frame q4	0	-

Continues on next page

Continued

FC Kinematics

Parameter	Value	Unit / Note
Name	fc_kinematics1	-
Bandwidth of force frame filter	25	Hz
Bandwidth of force loop filter	3	Hz
Damping in Force z Direction	3000	Ns/m

FC Application

Parameter	Value	Unit / Note
Name	fc_application1	-
Max Ref Force	1000	N
Max Ref Force Change	1000	N/s
Largest measured contact force	1000	N
Lowest measured contact force	3	N
Max Press TCP Speed	5	m/s
Max Press Rot speed	5	rad/s

FC Speed Change

Parameter	Value	Unit / Note
Name	fc_speed_change1	-
Speed ratio min	0.1	-
No of speed levels	2	-
Speed ratio delta	0.07	-
Speed max update period	0.08	second
Feedback type	Single DAC input	Single dimension external sensor
Feedback offset	0	-
Use Fdb LP filter	Yes	-
Fdb LP filter Bandwidth	30	Hz
Maximum TCP speed	0.3	m/s
Recover rule fdb ratio	1.3	-
Decrease rule safety fdb ratio	1.5	-
Decrease rule safety fdb time	0.1	second
Fdb trend step size	8	-
Decrease rule1 fdb ration	0.7	-
...	...	...
Increase rule 2 fdb trend	10	-

Continues on next page

PMC Sensor

Parameter	Value	Unit / Note
Name	fc_sensor1	-
Use PMC Sensor Setup	ATI_ACROMAG1	ATI_ACROMAG1 is used for robot 1 (ROB_1), ATI_ACROMAG2 is used for robot 2(ROB_2) etc.

PMC Sensor Setup

Parameter	Value	Unit / Note
Name	ATI_ACROMAG1	-
fx 1	0.25479	-
fx 2	0.02958	-
.....	.....	.....
ty max	400	Nm
tz max	400	Nm

NOTE!

The PMC Sensor Setup values are supplied by the sensor manufacturer. This file needs to be modified if it is used in a MultiMove system.

5 Programming

5.1 RAPID Components

5.1.1. Force Controlled Pressure applications

Overview

These are the instructions used to start stop and run force controlled pressure applications.

Between the start and End instructions any combination and number of FCPressL and FCPressC can be used. The Speed, force and zone may be changed for a new instruction allowing the process to be changed along with the properties of the application"

Pressure instructions

Instruction	Description
FCPress1LStart	Activates Force Control, starts movement and defines the data needed for the process below such as: • Movement: ToPoint, Speed zonedata Zone, Tool WObj • Force settings: direction of force, Force Threshold to start movementForceFrameRef i.e Wobj
FCPressL	Moves linear to robtarget with a force in the direction setup by FCPress1LStart.Magnitude of the force can be changed for every FCPressL
FCPressC	Moves circular to robtarget with a force in the direction setup by FCPress1LStart.Magnitude of the force can be changed for every FCPressC.
FCPressEnd	Leaves surface and moves to robtarget.

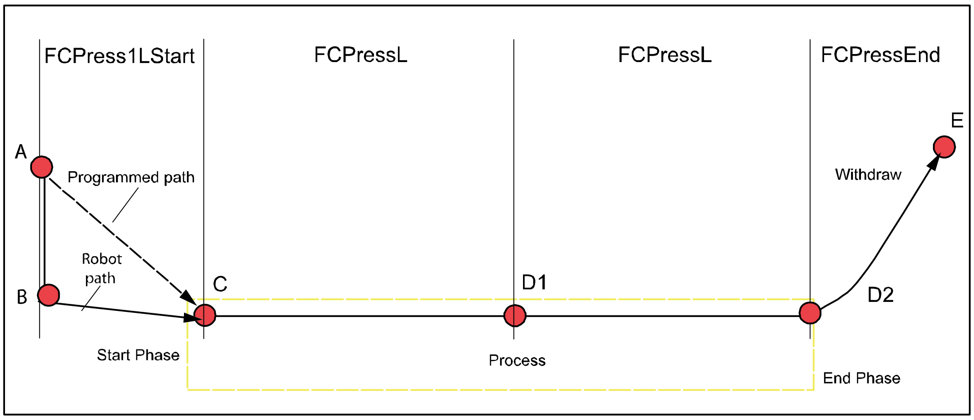
Calibrate the sensor

Instruction	Description
FCLoadID	Identifies the load measured by the force sensor. The identified load is used to calibrate the force sensor.
FCCalib	Calibrates the force sensor to remove sensor offset and compensate for gravity. Note that the calibration requires a precise definition of the load. Therefore, use the function FCLoadID before FCCalib

Continues on next page

The Phases

The figure below describes the phases for the FC pressure application.



xx0600003195

1. During the Start Phase the robot will switch to force control mode and move in the direction of the reference force in order to search contact with the work piece (B). Once contact is achieved the robot will start the movement towards the programmed position (C).
2. During the Process any number and combination of FCPressL and FCPressC may be used in order to run the application process (D1) (D2)…(Dx).
3. After the last process movement the robot will retract from the work piece in the opposite direction of the reference force until zero contact force is measured. At this time the robot will switch to position control and move to the end position (E).

5.1.2. Force Controlled SpeedChange applications

Overview

The Force Controlled SpeedChange function will automatically slow down the robot speed based on the process force information (measured by force sensor, spindle motor current, etc.). The robot will slow down when the process force level raises above a defined threshold. After the process force reduces below a certain level, the robot will automatically regain its programmed speed. Between activation and deactivation any standard move instruction can be used.

SpeedChange instructions

These are the instructions used to activate and deactivate Force Controlled SpeedChange:

Instruction	Description
FCSpdChgAct	Activates Force Controlled SpeedChange function with following parameters: • Reference (force, spindle motor current, etc.), reduce robot speed when the measured signal is greater than the reference. • Recover function name (RAPID routine with the specified name will be called when certain condition satisfies. • Different recover behaviors (e.g., MultipleRecover, Non-StopAllTime).
FCSpdChgDeact	Deactivates Force Controlled SpeedChange function.

These are instructions used to tune on-line configuration parameters for speed change function

Instruction	Description
FCSpdChgTunSet	Change configuration parameter to a new value, with following input arguments • Configuration parameter type • New valid value
FCSpdChgTunReset	Restore configuration parameter to its original value that stored in configuration file • Configuration parameter type

Calibrate the sensor

This instruction is used to calculate the gravity and centre of gravity for the current load.FCLoadId and FCCalib are required before the sensor can be used for Force Control.

Instruction	Description
FCCalib	Calibrates the force sensor to remove sensor offset and compensate for gravity. Note that the calibration requires a precise definition of the load. Therefore, use the function FCLoadID before FCCalib

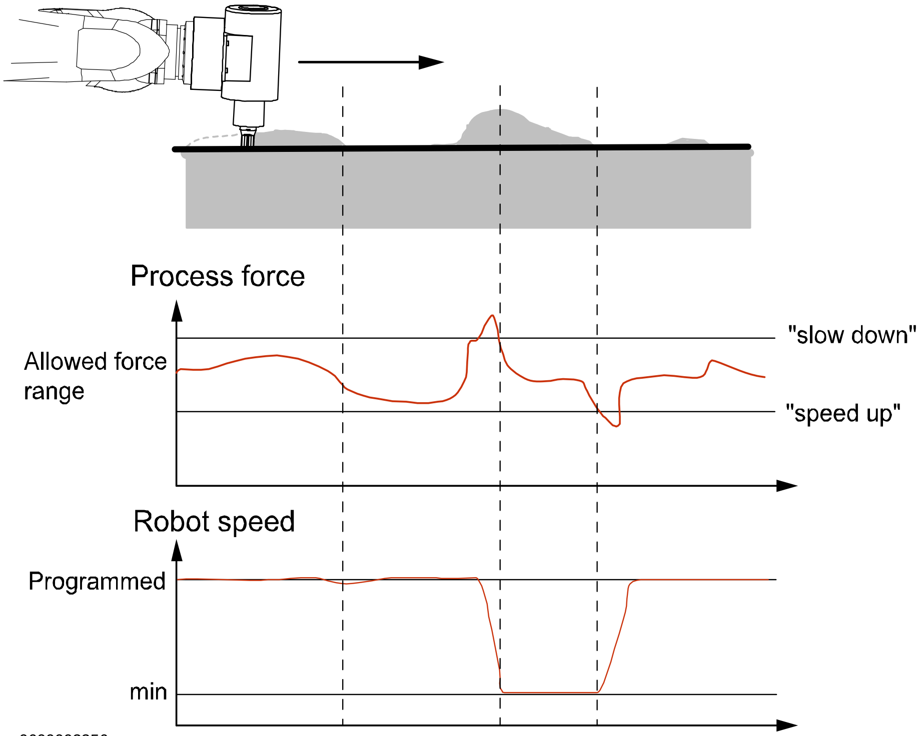
Continues on next page

NOTE!

FCCalib is only used for SpeedChange with force/ torque sensor.

Example

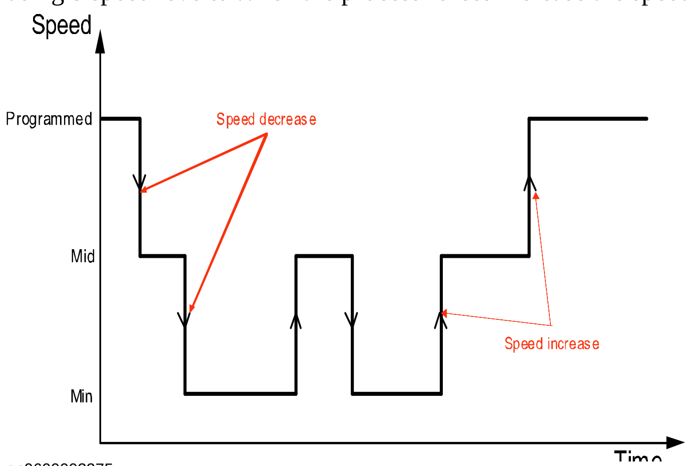
The figure below illustrates how the Robot speed is adapting to keep the process force within allowed force range. For more information, see *How does it work? on page 74*.



en0600003356

How does it work?

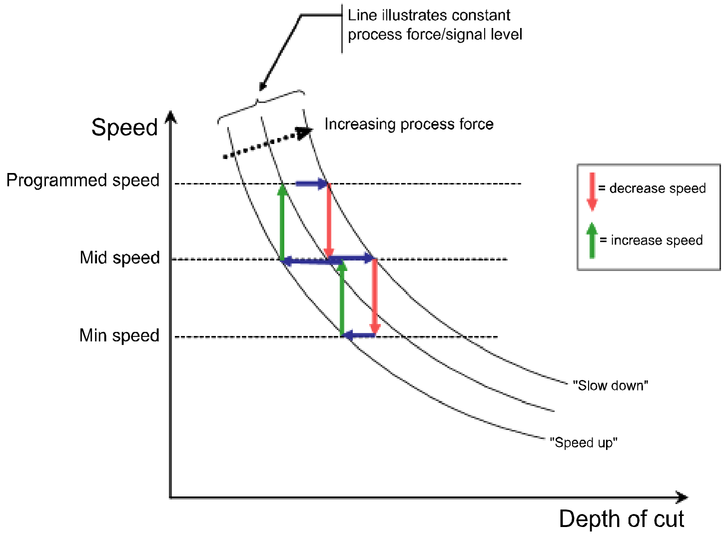
The control function for change the speed of the robot is "rule based" and include discrete speed levels in-between which the robot speed is changed. The number of speed levels can be defined using the parameter "No of speed levels". Below is an example showing a process using 3 speed levels. When the process forces increase the speed is reduced and vice a verse:



en0600003275

Continues on next page

The controller function is illustrated in the picture below:

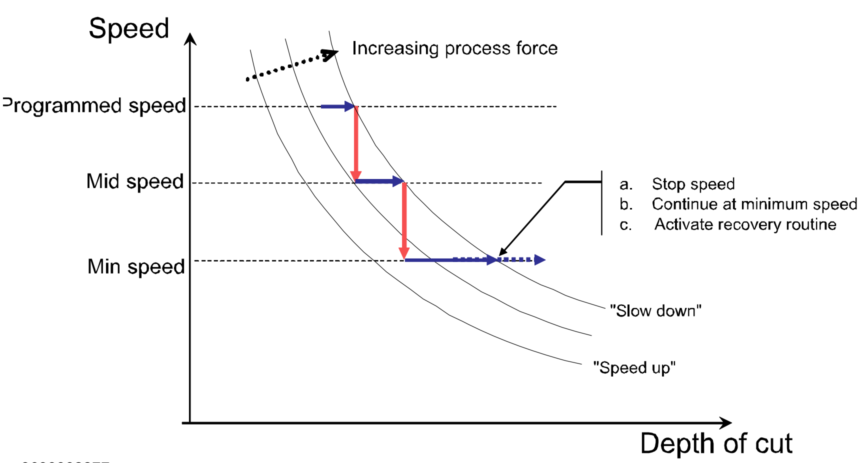


en0600003276

If changes in process will appear suddenly and a short response time for SpeedChange is urgently required one shall consider to use a 2 speed level solution. This will result in the quickest speed reduction. Applications with slowly changing process forces will gain cycle time using a multiple speed level solution, but number of speed levels above 4 should be used with care.

For situations where not even the minimum speed will reduce the process forces below the "slow down" level there are 3 optional behaviors to choose between.

1. When reaching the "slow down" level simply stop the robot speed.
2. When reaching the "slow down" level continue with minimum speed.
3. When reaching the "slow down" level activate a recovery routine. Recovery routine to be defined by the program designer and shall include a back up procedure to eliminate the cause of the exceeded process forces. After executing the recovery routine the robot will continue on the original path. If the "slow down" level is reached during the recovery routine the robot will stop.



en0600003277

5.2 Code examples

5.2.1. FC Press

Overview

This section provides examples on how to program the press instructions in Force Control for Machining. The basic approach for creating a RAPID program using Force Controlled press instruction is as follows:

1. Identify the load
2. Move to a point close to contact but not in contact
3. Calibrate
4. Setup Force control directions and start movement
5. Move linear or circular with contact
6. Leave surface

Example

Example below is with force Z with a movement in x-direction

```
PROC press1()
  PERS loaddata TestLoad:=[0.001,[0,0,0.001],[1,0,0,0],0,0,0];

  ! Identify the load using the sensor
   TestLoad:=FCLoadID();

  ! Move close to contact
  MoveJ  offs(B,0,0,2) , v100, fine, tool0;  !! start 2mm above
        contactpoint

  ! Calibrate the force sensor
  FCCalib TestLoad;

  ! Approach surface and start move to robtarget C  at 50% of
  ! 60 N i.e 30 N
   FCPress1LStart   C, v100, \Fz:=60,  50, z30,  myTool;

  ! Move Linear from C to D1 with a Force of 50 N in the z-direction
  FCPressL D1,v100,50,z30,myTool;

  ! Move Linear from D1 to D2 with a Force of 70 N in the
  ! z-direction
  FCPressL D2,v100,70,z30,myTool;

  ! Leave surface and move to robtarget E, Force control
  ! is disabled after this instruction
  FCPressEnd E, v100,myTool;
  ENDPROC
```

5.2.2. FC SpeedChange

Overview

This section provides examples on how to program the Force Controlled SpeedChange function. The basic approach for creating a RAPID program using Force Controlled SpeedChange is as follows:

1. Configure FC SpeedChange parameters such as, Feedback type, LP filter, etc.
2. If 6DOF force sensor is used for feedback, identify load and calibrate sensor. If an analogue signal is used from e.g. a spindle motor, define the nominal signal level in the parameter "Feedback offset".
3. Active FC SpeedChange with reference and desired recover behavior.
4. Perform machining task.
5. Deactivate FC SpeedChange.

NOTE!

Don't enable force control (FCAct in Assembly FC), when using Force Controlled SpeedChange function.

Example with force sensor

This example shows how to use Force Controlled SpeedChange function with force sensor. Before running the RAPID program, make sure to choose "Calib Force Magn" option for FC SpeedChange system parameter "Feedback type".

```
PERS robtarget myHome := ...
VAR tooldata myTool := ...
VAR wobjdata myWobj := ...
PERS loaddata myLoad := ...

myLoad := FCLoadID();
FCCalib myLoad;

! move to home position
MoveL myHome, v200, fine, myTool\WObj:=myWobj;

! turn on spindle motor before machining
TurnOnMotor();

! activate SpeedChange with reference force = 200 (N)
FCSpdChgAct 200;

! conduct machining task along path
MoveL ...
...

! deactivate SpeedChange function
FCSpdChgDeact;
```


Continues on next page

```
! turn off spindle motor after machining
TurnOffMotor();
```

Example with spindle motor current

This example shows how to use Force Controlled SpeedChange function with spindle motor current.

Before running the RAPID program:

1. Select "Single DAC Input" option for "Feedback type".
2. Set up "Feedback offset" value, e.g., idle current of spindle motor.
3. If measurement is noisy, set up "Fdb LP filter bandwidth (Hz)" and select "Yes" for "Use Fdb LP filter"

```
PERS robtarget myHome := ...
VAR tooldata myTool := ...
VAR wobjdata myWobj := ...

! move to home position
MoveL myHome, v200, fine, myTool\WObj:=myWobj;

! turn on spindle motor before machining
TurnOnMotor();

! activate SpeedChange with reference_motor_current = 2(A)
! use specified recover routine, "my_routine"
FCSpdChgAct 2 \RecoverFunName:="my_routine";

! conduct machining task along path
MoveL ...
...

! deactivate SpeedChange function
FCSpdChgDeact;

! turn off spindle motor after machining
TurnOffMotor();

! user specified recover routine
PROC my_routine()
VAR robtarget current_rbtrgt;

! get starting robot target
current_rbtrgt := CRobT(\Tool:=myTool \WObj:=myWobj);

! local cutting relative to current_rbtrgt
MoveL RelTool(current_rbtrgt,dx,dy,dz), v50, z0, myTool
     \WObj:=myWobj;
```


Continues on next page

```
MoveL RelTool() ...;
...

! move back to starting point and prepare for restoring
! the original planned path
MoveL current_rbtrgt, v50, z0, myTool\WObj:=myWobj;

ENDPROC
```

NOTE!

Parameter "Disable check of saturation" can be used if it is likely that the power output will reach saturation level.

6 Execution behavior

6.1. Damping and LP-filter

Damping

Damping is a definition of how large force is required for the robot to move at a certain speed. The damping parameters define how many Newtons are required to make the robot move at 1 m/s. The higher the value, the less responsive the robot gets.

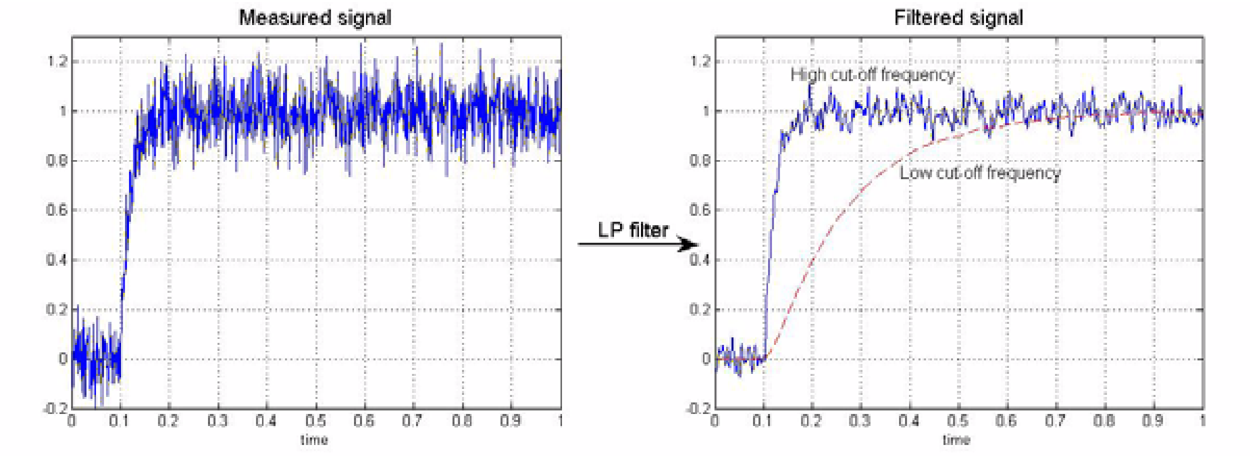
In Force Control, a contact force will make the TCP move with a speed proportional to the contact force. A contact torque will make the tool reorient with a speed proportional to the contact torque. The damping variable defines the proportions between a force and the resulting speed, and a torque and the resulting reorientation speed, in the direction x, y and z. The values are given as a percentage of the of the system parameter values defined in the type FC Kinematics, see *The FC Kinematics type on page 141*.

LP-filter

A Low-Pass filter lets the amplitude of low frequency signals pass through, and the amplitude of frequencies higher than the cut-off frequency are attenuated. If the signal is changing rapidly, a high cut-off frequency is needed. On the other hand, if the measured force is noisy, a low cut-off frequency may be required in order to remove the noise.

Illustration

The figure illustrates an LP-filter.



en0900000128

Continues on next page

Force controller structure and tuning

The picture shows a simplified picture of the force control loop.

In a force controlled direction the measured forces are subtracted from the corresponding reference forces.

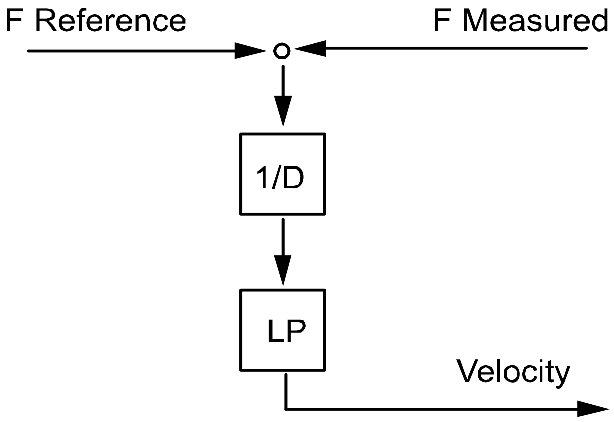
This difference is divided by D (=Damping). Damping is a force to speed factor and thus a speed reference is generated. This speed reference is low pass filtered with a cut off frequency that should be chosen depending on your robot model and process etc.

Default it is set to 3Hz which is a suitable value when the contact is really stiff (metal to metal).

For the large robot it is not possible to increase this value much but for a small robot with some compliance in the tool a filter frequency up to 25Hz can be used.

Since tuning both damping and low pass filter depends on compliance of tool, robot model, robot configuration etc. there is a unique set of these parameters for each process.

Both the damping and the LP filter cut-off frequency strongly affect how quickly and accurately the system is able to control the pressure force. If the robot reacts slowly when the force changes, or loses contact with the workpiece for periods of one or several seconds also for accurately programmed paths and low speeds, it is often possible to improve performance by decreasing the damping and/or increasing the LP filter cut-off frequency. On the other hand, if the robot bounces or vibrates rapidly with constant or increasing amplitude when pressing against the surface, this indicates that the damping should be increased and/or the filter frequency should be decreased.



en0600003351

NOTE!

Changing the parameters of the damping or Low pass filter might make the robot unstable.

Damping

The damping can easily be changed by an argument `in FCPress1LStart`. The argument represent a percentage of the configured damping value. (down to 50%, no upper limit)

For bigger changes the damping value needs to be re configured under, Motion->FC Kinematics-> Damping in Force Z direction.

For FCPressure always use Damping in z-direction even if your force is set in x or y.

6.2 FC Press optimization

6.2.1. Use Spd FFW

Overview

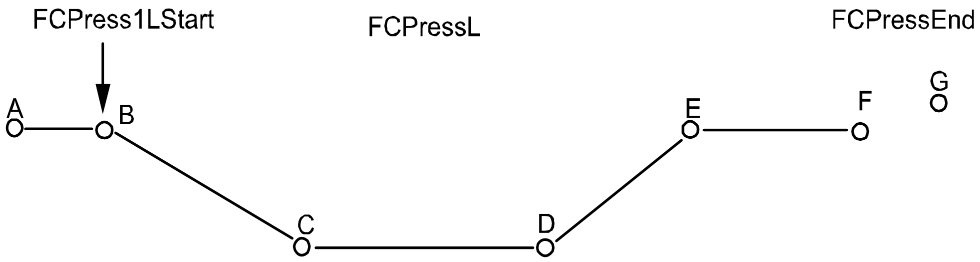
If the path is complex and the programmed path is accurate the performance is going to be enhanced by adding optional argument `UseSpdFFW` (use speed feed forward). FCPressure may be also used for temporary leaving the surface without deactivating.

Example 1

This example illustrates how to increase the performance.

```
FCPress1LStart;
```

The force in this example is directed down, see picture below. It would be possible to run directly from robtarget B to robtarget E but the performance will increase by adding robtarget C and D and using optional argument `\UseSpdFFW`.



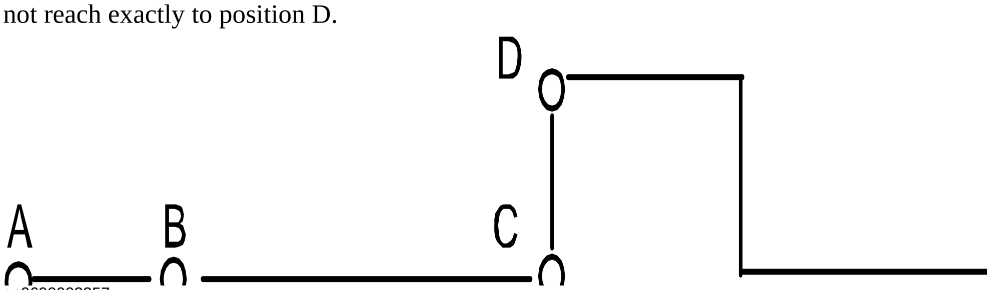
en0600003350

Example 2

This example illustrates how to leave the surface without deactivating.

```
FCPress1LStart B, v100, 70, z30 \UseSpdFFW, tool1;
FCPressL D, v100, 0, z30, tool1;
...
FCPressL E, v100, 70, z30, tool1;
```

The reference force is temporarily switched off (set to 0) together with the optional argument UseSpdFFW. The robot will leave the surface and follow the path to D. Note that the robot is still force controlled and will not behave 100% like position controlled robot. The robot will not reach exactly to position D.



en0600003357

6.3 FC SpeedChange control design

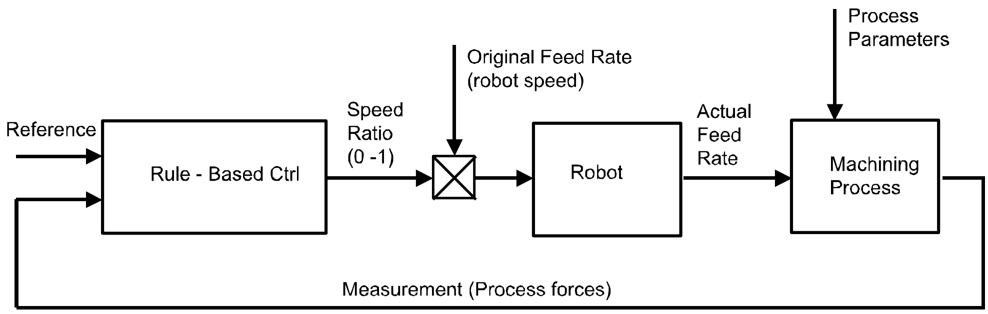
6.3.1. Controller Scheme

Overview

The FC SpeedChange controller for reducing/increasing the robot speed is a rule-based logic controller. The design details are described in following sections.

Scheme

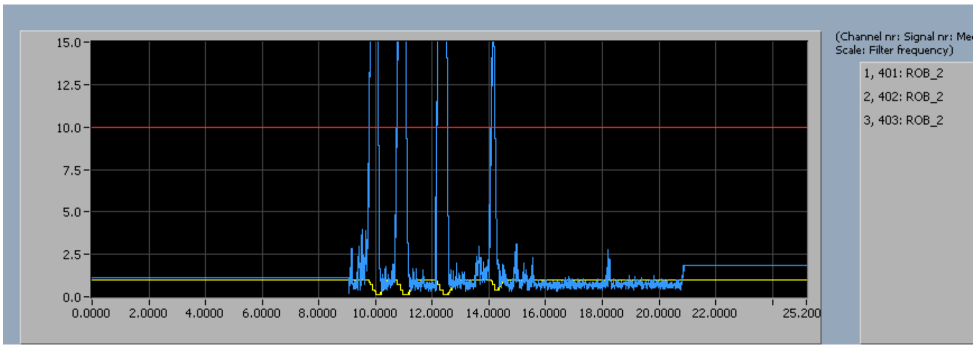
The force controlled speed control scheme is shown in the figure below. The maximum force is specified for the machining process, and the actual process force is monitored and controlled to be less than the maximum force by adjusting the machining feed rate (robot speed). The output of the rule-based logic control is the percentage (between 0% and 100%) of the original feed rate.



en0600003278

Available test signals to tune the process

The following picture illustrates the three test signals that appear from a recording made with the Test Signal Viewer. In this example the reference signal is set to value 10. When the difference between the reference signal (401) and the measurement signal (402) fulfills the criterias for speed decrease the speed ratio signal (403) drops, in this case four levels.



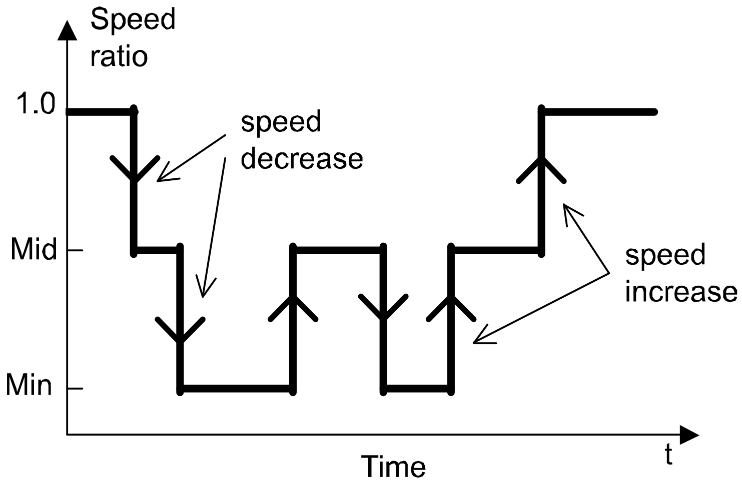
xx0700000533

Signal 401 (red color)	Reference test signal
Signal 402 (blue color)	Measurement (Process force) test signal
Signal 403 (yellow color)	Speed ratio signal

6.3.2. Rule based logical control

Increase/Decrease

The controller output, speed ratio, is generated by certain rules based on the measured process force information. A sample speed ratio output of 3-step rule-based logic controller is shown in Figure below.



en0600003279

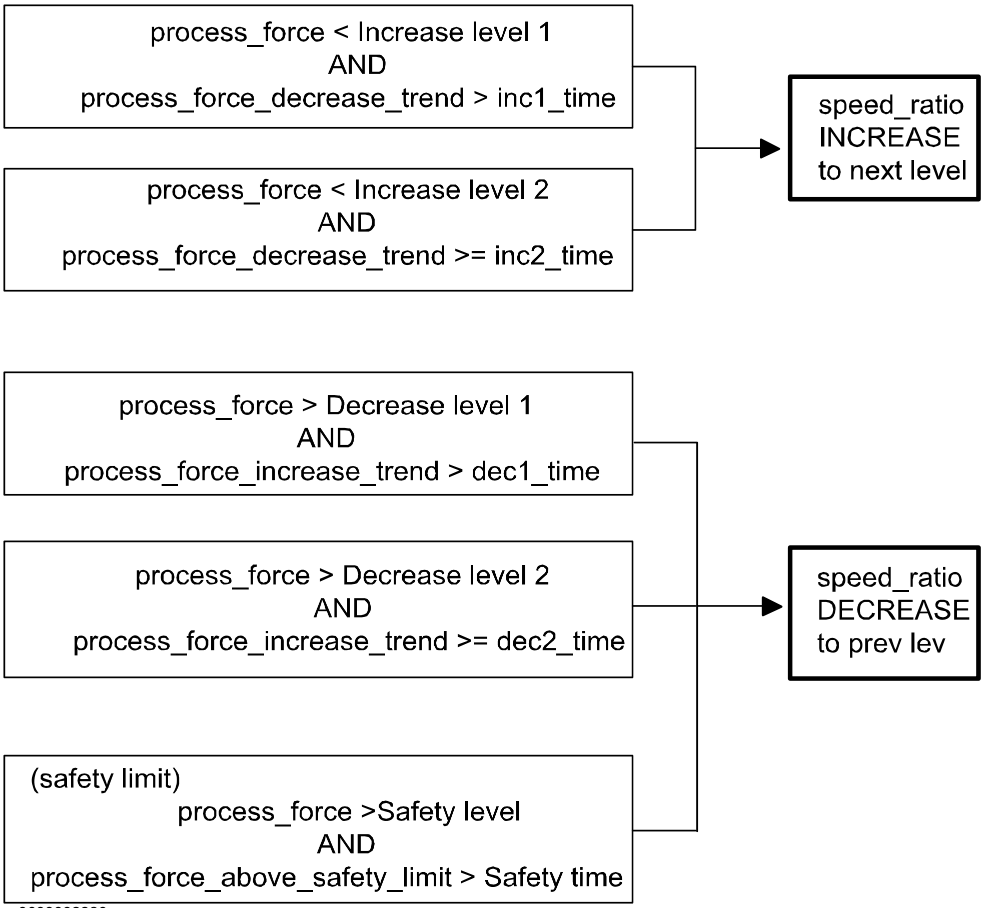
Instead of changing continuously as in normal PID control, speed ratio in rule-based logic control is divided into several discrete steps. The logic rules will decide when the speed will decrease or increase to the next stage or remain at current stage. The goal of SpeedChange is to keep process force below a pre-specified maximum force as fast as possible.

Although ideally more steps means more control accuracy, 3 steps would be enough for most applications. Too many steps will increase the response time when cutting large-size material and make speed reduction less responsive. 2-step speed control would be the most used control setup.

Continues on next page

Rules

The following rules decide when to increase or decrease the speed ratio. In all other conditions, controller maintains the previous speed ratio.



en0600003280

6.3.3. Programming in Path Coordinates

Overview

Defining the force control coordinate system relative to the path coordinate system gives the possibility to define the force control action relative to the programmed motion trajectories of the robot. This is particularly useful in situations where it is desired to apply a force against a surface with a varying normal direction, without reorienting the tool.

For a description of the path coordinate system see *Technical reference manual - RAPID Instructions, Functions and Data types*, instruction *CorrCon*.

Usage

It is important to note that all directions in path coordinates except the force control direction will be position-controlled. If the robot TCP position has drifted away from the programmed path and the path frame undergoes a quick rotation, this will cause the position reference to change quickly. This will introduce a very rapid corrective motion or "jerk" in the position-controlled directions. Built-in supervision is present in the system in order to stop the robot if the corrective motion becomes too fast. If this occurs, modify the program in one or several of the following ways:

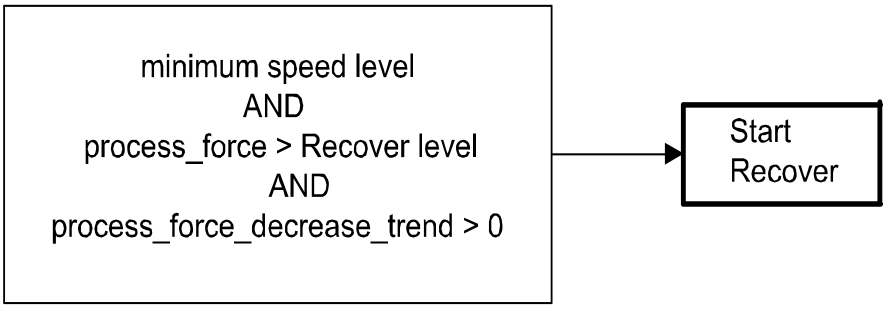
1. Program the path closer to the surface.
2. Decrease the path speed, especially near sharp corners in the path.
3. Reprogram the path to avoid sharp path corners, for example, by increasing the size of the corner zones.

A rule of thumb is that the force control path deviation, the distance from the programmed TCP position to the true TCP position, should be shorter than the effective radius of curvature of the programmed path.

6.3.4. Recover routine

Overview

If process force is still higher than the reference force when the feed rate is already at its lowest possible speed (usually this condition happens if larger volume of material than expected is encountered during machining process), a recover process will start in order to avoid tool damage. The following diagram is the rule to enter the recover process.



en0600003282

The recover process could simply stop the robot, or more a advanced solution would be to perform local cutting in layers to get rid of the large material, and then continue the original path after local cutting. The local cutting process needs to be implemented by user as a RAPID routine, no recovery routine is defined by default. A sample recover routine is shown as below:

Recover Example

```
PROC user_recover_routine()
VAR robtarget current_rbtrgt;

! get starting robot target
current_rbtrgt := CRobT(\Tool:=UserTool \WObj:=UserWobj);

! local cutting relative to current_rbtrgt
MoveL RelTool(current_rbtrgt,dx,dy,dz), v50, z0, UserTool
\WObj:=UserWobj;
MoveL RelTool() …… ;
……
! move back to starting point
MoveL current_rbtrgt, v50, z0, UserTool\WObj:=UserWobj;

ENDPROC
```

The user-defined recover routine will be automatically called during recover process when specified

7 Troubleshooting

7.1. What to do when...

...the robot drifts

When the robot is in force control mode (after executing `FCAct`) and no external force except gravity are present, it should not move (as long as no reference force, torque or movement is applied). If the robot drifts away with a slow movement anyway, check the following:

Step	Action
1.	The robot should be near its working position when calibrated with the instruction FCCalib.
2.	Verify that the load is identified with the instruction `FCLoadID` and that the LoadIdErr is smaller than 0.1 for an optimal load identification. If the movement of axes 5 and 6 are too limited during this load identification, the result may be poor.
3.	Verify that the system parameters are correctly defined. E.g. too low damping in the type *FC Kinematics* may cause the robot to drift.
4.	Verify that the PMC Sensor setup matrix is correctly typed in.
5.	Verify the orientation of gravity with respect to the base frame. If necessary update the system parameter Gravity Alpha and Gravity Beta in Motion->Robot. For more information, see *Technical reference manual - System parameters*.
6.	Increase the value of thesystem parameter "Lowest mesured contact force, in type FCApplication.

8 Rapid reference information

8.1 Instructions

8.1.1. FCCalib

Usage

`FCCalib` is used to calibrate the force sensor. Before this instruction is executed it is not possible to switch to force control. It is necessary to specify the data for the used load as an argument to this instruction. Load data can easily be retrieved by first performing a load identification using the function `FCLoadID`. By using the load data the system can do an internal calibration to compensate for sensor offset and prepare gravity force compensation. It is important to understand that the function `FCCalib` needs to be run every time the load is changed.

TIP!

It is also recommended to do the calibration close to the position where the robot will be doing most work.

NOTE!

The Calibration should always be done when no contact forces are present. The only exception is when using the optional parameter `Recovery` which might be used for example after an emergency stop.

Basic example

```
PERS loaddata my_load:=[0.001,[0,0,0.001],[1,0,0,0],0,0,0];
my_load:= FCLoadID();
FCCalib my_load;
```

The example above shows how to use `FCCalib`. It is very important for force control performance to have a good load definition. It is therefore strongly recommended to identify the load using the function `FCLoadID`.

NOTE!

If sensor is room fixed, any load can be used as argument.

Arguments

```
FCCalib Load [\Recovery]
```

`Load`

Data type: `loaddata`

The load used to calibrate the sensor. Only mass and centre of gravity is presently used, so inertia does not have to be specified. Note that this load is the load the sensor feels. It is normal that this load is **not** zero even if only the sensor itself is mounted on the robot. Use the function `FCLoadID` to identify the load.

Continues on next page

[\Recovery]

Data type: `switch`

Specifies whether to use the previous calibration offset, which was read the last time `FCCalib` was called without this argument. Makes it possible to activate force control when in contact. The argument can be needed for example after an emergency stop.

Program execution

Before the sensor is calibrated with `FCCalib`, most other Force Control instructions are not allowed.

Syntax

```
FCCalib
  [ Load ´:=´ ] < expression (IN) of loaddata > [´\´ Recovery ] ´;´
```

Related information

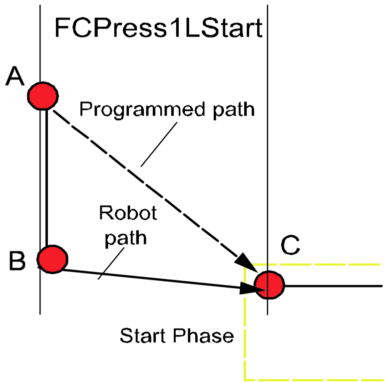
For information about	See
Identifying the load	*FCLoadID on page 117*.

8.1.2. FCPress1LStart

Usage

`FCPress1LStart` is used to make contact to a surface and move the tool centre point (TCP) linearly to a first given destination. The following contact movement should be done with the `FCPressL` instruction. Instruction starts a sequence for regulation in one direction on force. If you want to follow a corner or otherwise follow in more than one dimension this instruction should not be used.

Move close to the contact point (NOT IN CONTACT) The point `ToPoint` is where the first move in contact from the contact point will go. ForceThreshold is the parameter that will have to be tuned for best result.



xx0600003285

(A) is the point close to contact. The force reference will move us to point (B) which is a point when we have contact. The movement to point C (ToPoint) will start when the force has reached a certain level defined by the parameter ForceThreshold (in % of ordered force)

Instruction sets up some data that is true for a sequence. The force is set in each coordinate direction, this will result in one force in one direction calculated from these settings. This direction of force definition is true for all following FCPress-move instructions and will be until `FCPressEnd`. All parameters in `FCPress1LStart` except `ToPoint`, `Speed`, `Force` and `Zone` is true until `FCPressEnd`.

NOTE!

The distance B to C must be more than 100 ms, or else C will become a fine point.

Basic example

Basic example of the instruction `FCPress1LStart` is illustrated below.

See also *More Examples*.

Example

```
FCPress1LStart p10, v100 \Fz:=200, 57, z30, tool1
```

Move in positive force direction (= z) until 57% (= 114N) of force is reached and then start toward p10 while force builds up to 100% (=200N).

Continues on next page

Arguments

```
FCPress1LStart ToPoint [\ToNextPoint] Speed [\Fx] [\Fy] [\Fz]
     ForceThreshold [\ForceFrameRef] [\ForceChange]
     [\DampingTune] [\TimeOut] [\UseSpdFFW] [\PosSupvDist] Zone
     Tool [\WObj]
```

`ToPoint`

Data type: `robtarget`

The destination point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction).

`[\ToNextPoint]`

Data type: `robtarget`

The destination point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction). This should only be used if the robtarget in the first `FCPressL` is to close and therefore becomes a corner path failure.

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool centre point, the tool reorientation and external axes.

`[\Fx]`

Reference force in x direction

Data type: `num`

Defines the constant reference force in the x direction of the force control coordinate system.If this argument is omitted there will be zero contact force in the x direction.

`[\Fy]`

Reference force in y direction

Data type: `num`

Defines the constant reference force in the y direction of the force control coordinate system.If this argument is omitted there will be zero contact force in the y direction.

`[\Fz]`

Reference force in z direction

Data type: `num`

Defines the constant reference force in the z direction of the force control coordinate system.If this argument is omitted there will be zero contact force in the z direction.

`ForceThreshold`

Data type: `num`

Percentage of the contact force that should be reached before move toward the robtarget starts. TCP moves in force direction until this percentage is reached. When percentage of force is reached, the movement toward target starts.

Continues on next page

`[\ForceFrameRef]`

Data type: `fcframe`

`ForceFrameRef` here defines which coordinate system the force control coordinate system is related to. The parameter can be set to either the work object coordinate system, the tool coordinate system or the path coordinate system described in section *Programming in Path Coordinates on page 87*.

FC_REFFRAME_WOBJ, FC_REFFRAME_TOOL or FC_REFFRAME_PATH.

The default value is the tool coordinate system.

`[\ForceChange]`

Data type: `num`

Tuning parameter to ramp up force. Unit is [N/s]. This argument overrides configured value.

`[\DampingTune]`

Data type: `num`

`DampingTune` is the relation value between the measured force and the applied resulting force. By default the value are 100% (of system parameter values), but it can be between 50% and infinity. Smaller values than 100% means that the robot is more sensitive to external force

`[\TimeOut]`

Data type: `num`

If force hasn't build up before this time is reached then continue with next instruction. Unit is [s].

`[\UseSpdFFW]`

Data type: `switch`

If this argument is used then feed forward regulation is used. If argument isn't used then regulation in force direction is done only with force control without help from programmed path. Use this argument if path is complex and programmed path is close to actual path.

`[\PosSupvDist]`

Data type: `num`

The robot will stop if it has moved more than the distance `PosSupvDist` away from the programmed path. Default value is 20mm. Unit is [mm].

`Zone`

Data type: `zonedata`

`Zone` data for the movement. `Zone` data describes the size of the generated corner path.

`Tool`

Data type: `tooldata`

The tool used during Force Control. It is the center point of this tool that is used for all calculations. Note that the dimensions of the sensor and any interface plates need to be included in the tool definition. To change tool force control has to be deactivated.

Continues on next page

`[\WObj]`

Data type: `wobjdata`

The work object (coordinate system) to which the robot position in the instruction is related. This argument can be omitted, and if it is, the position is related to the world coordinate system. If, on the other hand, a stationary TCP or coordinated external axes are used, this argument must be specified. To change work object force control has to be deactivated.

NOTE!

The coordinate systems mentioned in the arguments list are described in *The Coordinate systems on page 201*.

Program execution

Execution behavior:

- `FCPress1LStart` activates Force Control.

More examples

More examples of how to use the instruction `FCPress1LStart` are illustrated below.

Example 1

```
VAR tooldata tool1:=[TRUE,[[97.4,0,223],[1,0,0,0]],
      [5,[23,0,75],[1,0,0,0],0,0,0]];
FCPress1LStart p10, v100 \Fy:=-100, 57
      \ForceFrameRef:=FC_REFFRAME_TOOL, z30, tool1;
```

Activates Force Control and defines a force control coordinate system based on the tool1 coordinate system with force on negative y axis.

Example 2

```
VAR tooldata tool1:=...
VAR wobjdata my_wobj :=
      [FALSE,TRUE,"",[[0,0,0],[,0,0,0]],[[0,0,0],[0.07071,0,0.707
      1,0]]];
FCPress1LStart p10, v100 \Fz:=200, 57,
      \ForceFrameRef:=FC_REFFRAME_WOBJ \ForceChange:=200, z30,
      tool1 \WObj:=my_wobj;
```

Activates Force Control with:

- tool tool1
- force control coordinate system orientation equal to the orientation of the work object my_wobj.
- maximum force change to 200 Newton per second

Example 3

```
VAR tooldata tool1:=...

VAR wobjdata my_wobj :=
[FALSE,TRUE,"",[[0,0,0],[,0,0,0]],[[0,0,0],[0.07071,0,0.707
      1,0]]];
```


Continues on next page

```
FCPress1LStart p10, v100 \Fz:=200, 57,
     \ForceFrameRef:=FC_REFFRAME_WOBJ \ForceChange:=200
     \PosSupvDist:=100 , z30, tool1 \WObj:=my_wobj;
```

This example will behave like example 2 unless the robot would deviate from the programmed path. In example 2 the robot will stop if it deviates more then 20 mm, and in this example the robot may drift 100 mm before it stops.

Limitations

The Force Control will only behave correctly if the load is identified with `FCLoadID` and the sensor is calibrated with `FCCalib` before activating Force Control with `FCPress1LStart` FCPress1-move instructions can only be used between `FCPress1LStart` and `FCPressEnd`.

Syntax

```
FCPress1LStart
  [ ToPoint ´:=´ ] < expression (IN) of robtarget >
  [ ´\´ ToNextPoint ´:=´ < expression (IN) of robtarget > ] ','
  [ Speed ´:=´ ] < expression (IN) of speeddata >
  [ ´\´ Fx ´:=´ < expression (IN) of num > ]
  [ ´\´ Fy ´:=´ < expression (IN) of num > ]
  [ ´\´ Fz ´:=´ < expression (IN) of num > ] ´,´
  [ ForceThreshold ´:=´ ] < expression (IN) of num >
  [ ´\´ ForceFrameRef ´:=´ < expression (IN) of fcframe > ]
  [ ´\´ ForceChange ´:=´ < expression (IN) of num > ]
  [ ´\´ DampingTune ´:=´ < expression (IN) of num > ]
  [ ´\´ TimeOut ´:=´ < expression (IN) of num > ]
  [ ´\´ UseSpdFFW ] ´,´
  [ ´\´ PosSupvDist ´:=´ < expression (IN) of num > ]
  [ Zone ´:=´ ] < expression (IN) of zonedata > ´,´
  [ Tool ´:=´ ] < persistent (PERS) of tooldata >
  [ ´\´ WObj ´:=´ < persistent (PERS) of wobjdata > ] ´;´
```

Related information

For information about	See
Linear one dimensional press instruction	*FCPressL on page 98*
Circular one dimensional press instruction	*FCPressC on page 100*
End the press instruction	*FCPressEnd on page 102*

8.1.3. FCPressL

Usage

`FCPressL` is used to move the tool centre point (TCP) linearly to a given destination and during this movement a contact force can be maintained to a surface.

Basic examples

Basic examples of the instruction `FCPressL` are illustrated below.

Example

```
VAR num Force=60;
FCPressL p10, v100, Force, z30, tool0;
```

Move to point `p10` with speed `v100` and a contact force of 60 N in the direction decided by the `FCPress1LStart` instruction.

Arguments

```
FCPressL ToPoint Speed Force Zone Tool [\Wobj]
```

`ToPoint`

Data type: `robtarget`

The destination point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction).

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool centre point, the tool reorientation and external axes.

`Force`

Data type: `num`

The force size in the direction defined in the coordinate system chosen in `FCPress1LStart`.

`Zone`

Data type: `zonedata`

Zone data for the movement. Zone data describes the size of the generated corner path.

`Tool`

Data type: `tooldata`

The tool is used during Force Control. Note that this must be the same tool that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

`[\Wobj]`

Data type: `wobjdata`

The work object (coordinate system) to which the robot position in the instruction is related to. Note that this must be the same work object that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

Continues on next page

Program execution

Execution behavior:

- `FCPressL` moves toward target in contact with surface at the specified force. Movement will follow the surface and as a result the path will not be completely linear

Limitations

Instruction can only be used between a `FCPress1LStart` and a `FCPressEnd`.

Tool and WObj cannot be changed while force control is active. Tool and work object are set in `FCPress1LStart` and cannot be changed until after `FCPressEnd` and a new `FCPress1LStart`.

Syntax

```
Instruction
  [ ToPoint´:=´ ] < expression (IN) of robtarget> ´,´
  [ Speed ´:=´ ] < expression (IN) of speeddata > ´,´
  [ Force ´:=´ ] < expression (IN) of num > ´,´
  [ Zone ´:=´ ] < expression (IN) of zonedata > ´,´
  [ Tool ´:=´] < persistent (PERS) of tooldata >
  [´\´ Wobj ´:=´ < persistent (PERS) of wobjdata > ] ´;´
```

Related information

For information about	See
Start press instruction	*FCPress1LStart on page 93*
Circular one dimensional press instruction.	*FCPressC on page 100*
End press instruction.	*FCPressEnd on page 102*

8.1.4. FCPressC

Usage

`FCPressC` is used to move the tool centre point (TCP) circular to a given destination and during this movement a contact force can be maintained to a surface

Basic examples

Example

```
VAR num Force=60;
FCPressC p10, p20, v100, Force, z30, tool0;
```

Move circularly to point `p20` with speed `v100` and a contact force of 60 N in the direction decided by the `FCPress1LStart` instruction. The Circle is defined by the start position, the circle point `p10` and the destination point `p20`.

Arguments

```
FCPressC CirPoint ToPoint Speed Force Zone Tool [\Wobj]
```

CirPoint

Circle point

Data type: `robtarget`

The circle point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction).

`ToPoint`

Destination point

Data type: `robtarget`

The destination point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction).

`Speed`

The speed of the TCP

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool centre point, the tool reorientation and external axes.

`Force`

Data type: `num`

The force size in the direction defined in the coordinate system chosen in `FCPress1LStart`.

`Zone`

Data type: `zonedata`

Zone data for the movement. Zone data describes the size of the generated corner path.

Continues on next page

`Tool`

Data type: `tooldata`

The tool is used during Force Control. Note that this must be the same tool that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

`[\Wobj]`

Data type: `wobjdata`

The work object (coordinate system) to which the robot position in the instruction is related to. Note that this must be the same work object that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

Program execution

Execution behavior:

- `FCPressC` moves toward target in contact with surface at the specified force. Movement will follow the surface and as a result the path will not be completely circular.

Limitations

Instruction can only be used between a `FCPress1LStart` and a `FCPressEnd`. Tool and WObj cannot be changed while force control is active. Tool and work object are set in `FCPress1LStart` and cannot be changed until after `FCPressEnd` and a new `FCPress1LStart`.

Syntax

```
FCPressC
  [ CirPoint´:=´ ] < expression (IN) of robtarget> ´,´
  [ ToPoint´:=´ ] < expression (IN) of robtarget> ´,´
  [ Speed ´:=´ ] < expression (IN) of speeddata > ´,´
  [ Force´:=´ ] < expression (IN) of num > ´,´
  [ Zone ´:=´ ] < expression (IN) of zonedata > ´,´
  [ Tool ´:=´] < persistent (PERS) of tooldata >
  [´\´ Wobj ´:=´ < persistent (PERS) of wobjdata > ] ´;´
```

Related information

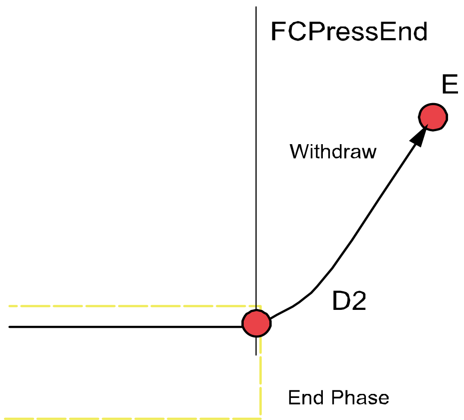
For information about	See
Start press instruction.	*FCPress1LStart on page 93*
Linear one dimensional press instruction.	*FCPressL on page 98*
End press instruction.	*FCPressEnd on page 102*

8.1.5. FCPressEnd

Usage

`FCPressEnd` is used to release the contact from the `FCPress1LStart` and `FCPressL`

When calling this function the position is D2 which is a point where a contact force is present. The user specifies a point E which should be close to contact but NOT IN contact



xx0600003286

The same force reference as was used in `FCPress1LStart` (but with different sign) will start to reduce the contact force giving an upward movement. At the same time the move instruction to point E will give a horizontal movement

When the horizontal movement is finished, force control is switch off and position control will move to point E.

NOTE!

The horizontal (orthogonal to force direction) distance between point (D2) and (E) should be short.

Basic examples

Basic example of the instruction `FCPressEnd` is illustrated below.

See also *More examples on page 103*.

Example

```
FCPressEnd p10, v100, tool0;
```

Move to p10 with 100mm/s and on the way when force is zero turn off force control.

Arguments

```
FCPressEnd ToPoint Speed [\ForceChange] [\ZeroContactValue] Tool
     [\Wobj]
```

`ToPoint`

Data type: `robtarget`

The destination point of the robot and external axes. It is defined as a named position or stored directly in the instruction (marked with an * in the instruction).

Continues on next page

`Speed`

Data type: `speeddata`

The speed data that applies to movements. Speed data defines the velocity for the tool centre point, the tool reorientation and external axes.

`[\ForceChange]`

Data type: `num`

Tuning parameter to ramp up force. Value in argument is given in [N/s].

`[\ZeroContactValue]`

Data type: `num`

When the force is less than this argument, force control is deactivated. If the argument is not used the default valuedefined in the system parameters will be used.

`Tool`

Data type: `tooldata`

The tool is used during Force Control. Note that this must be the same tool that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

`[\Wobj]`

Data type: `wobjdata`

The work object (coordinate system) to which the robot position in the instruction is related to. Note that this must be the same work object that is used in the `FCPress1LStart` instruction.

This argument is only present due to offline programming purposes.

Program execution

Execution behavior:

- `FCPressEnd` deactivates force control and switch to position control when force becomes less then `ZeroContactValue`.

More examples

More examples of how to use the instruction `FCPressEnd` are illustrated below.

Example 1

```
FCPressEnd p10, v100 \ForceChange:=100;
```

Set max release speed of force to 100N/s and then deactivate Force Control.

Example 2

```
FCPressEnd p10, v100 \ZeroContactValue:=2.5;
```

When force is less than 2.5N then deactivate force control and continue move to p10.

Limitations

Tool and work object are set in `FCPress1LStart` and cannot be changed until after `FCPressEnd` in a new `FCPress1LStart`.

Continues on next page

Syntax

```
FCPressEnd
  [ ToPoint´:=´ ] < expression (IN) of robtarget> ´,´
  [ Speed ´:=´ ] < expression (IN) of speeddata >
  [ ´\´ ForceChange´:=´ ] < expression (IN) of num>
  [ ´\´ ZeroContactValue´:=´ ] < expression (IN) of num> ´,´
  [ Tool ´:=´] < persistent (PERS) of tooldata >
  [´\´ Wobj ´:=´ < persistent (PERS) of wobjdata > ] ´;´
```

Related information

For information about	See
Start the press instruction	*FCPress1LStart on page 93*

8.1.6. FCSpdChgAct

Usage

The `FCSpdChgAct` is used to activate FC SpeedChange function with desired reference and recover behavior. When FC SpeedChange function is active, the robot speed will be reduced/ increased in order to keep the measured signal close to the reference.

Basic examples

Basic example of the instruction `FCSpdChgAct` is illustrated below.

See also *More Examples*.

Example

```
FCSpdChgAct 200 /RecoverFunName:="local_grind";
```

Activate FC SpeedChange with user-specified recover routine "local_grind". The measured process signal will be controlled to be 200 by slowing down TCP speed when required.

Arguments

```
FCSpdChgAct  Reference  [\RecoverFunName]  [\NonStopAllTime]
      [\MultipleRecover]
```

`Reference`

Data type: `num`

The reference value for the process force. (Process force defined by input, such as Force sensor, current, torque etc.) The measurement will be controlled below this reference value. The value of the reference must be identified in tests during normal conditions.

`[\RecoverFunName]`

Data type: `string`

This parameter specifies the name of user-defined recovery routine. The recovery routine will be executed, if the process force is still is too large after the TCP speed already is reduced to the minimum speed. The recovery routine needs to be implemented by the user in order to have desired recover behavior. If no recover routine is specified, the robot will stop immediately when the above recover condition met.

`[\NonStopAllTime]`

Data type: `switch`

This option can only be used when `RecoverFunName` argument is NOT used. The robot will at most slow down to minimum feed rate (speed), which means that the robot will not stop for any overload occurring at minimum speed. USE THIS OPTION WITH CAUTION.

`[\MultipleRecover]`

Data type: `switch`

This option can only be used when `RecoverFunName` argument is used with this option, the user-specified recover procedure will be called multiple times along the path whenever overload happens at minimum feed speed. If this option is not specified, the user-specified recover procedure will be called the first time when recover condition met. If the recover condition is met again along the path, the robot will stop immediately

Continues on next page

Program execution

Execution behavior:

- Configure FC SpeedChange system parameters before calling FCSpdChgAct.
- If 6DOF force sensor is used for feedback, FCCalib must be called before FCSpdChgAct.
- The RobotWare option "Path Recovery" must be installed in order to use FcSpdChgAct instruction with recover function. The only exception is to use the FcSpdChgAct instruction with "NonStopAllTime".
- User-specified recover routine will not be called recursively. Which means, if the recover condition met when controller is executing user-specified recover routine, the robot will stop immediately instead of calling user-specified recover routine from itself.
- If the RAPID program pointer is moved manually, FC SpeedChange function will be deactivated automatically.
- If the RAPID program stops, jogs away from current position, then restarts without regain the path, FC SpeedChange function will be deactivated automatically.

Limitations

- Do NOT change tool and work object frame in RAPID program between `FCSpdChgAct` and `FCSpdChgDeact`.

More examples

More examples of how to use the instruction `FCSpdChgAct` are illustrated below.

Example 1

```
FCSpdChgAct 200;
```

Activate FC SpeedChange function with reference 200. No user-specified recover behavior is defined. The robot will stop immediately when recover condition met.

Example 2

```
FCSpdChgAct 200 \RecoverFunName:="local_grind";
```

Activate FC SpeedChange function with reference 200 and user-specified recover routine, "local_grind". Local_grind will be executed when recover condition met, but will be called only once.

Syntax

```
Instruction
  [ Reference ´:=´ ] < expression (IN) of num > ´,´
  [ RecoverFunName ´:=´ ] < expression (IN) of string > ´,´
  [ ´\´ NonStopAllTime ] ´,´
  [ ´\´ MultipelRecover ] ´;´
```

Related information

For information about	See
Deactivate SpeedChange	*FCSpdChgDeact on page 107*

8.1.7. FCSpdChgDeact

Usage

Deactivate FC SpeedChange function.

Basic examples

Basic example of the instruction `FCSpdChgDeact` is illustrated below.

Example

```
FCSpdChgDeact;
```

Deactivates SpeedChange function.

Arguments

```
FCSpdChgDeact
```

There are no arguments to the instruction.

Syntax

```
FCSpdChgDeact
```

Related information

For information about	See
Activate SpeedChange.	*FCSpdChgAct on page 105*

8.1.8. FCSpdChgTunSet

Usage

FCSpdChgTunSet is used to set FC SpeedChange system parameters to a new value.

Basic examples

Basic example of the instruction `FCSpdChgTunSet` is illustrated below.

See also *More Examples*.

Example

```
FCSpdChgTunSet 0.2, FC_SPEED_RATIO_MIN;
```

Set FC SpeedChange system parameter "Speed ratio min" to 0.2.

Arguments

```
FCSpdChgTunSet value, type;
```

`value`

Data type: `num`

Value to be set for the FC SpeedChange system parameter.

`type`

Data type: `fcspdchgtunetype`

The FC SpeedChange system parameter whose value is to be set (FC_SPEED_RATIO_MIN, FC_NO_OF_SPEED_LEVELS). Only two FC SpeedChange system parameters can be tuned by this instruction, as shown in the following table:

Parameter	Type
Speed ratio min	FC_SPEED_RATIO_MIN
No of speed levels	FC_NO_OF_SPEED_LEVELS

Program execution

Execution behaviour:

- Set new values to tunable FC SpeedChange system parameters.

More examples

More examples of how to use the instruction `FCSpdChgTunSet` are illustrated below.

Example 1

```
FCSpdChgTunSet 3, FC_NO_OF_SPEED_LEVELS;
```

Set FC SpeedChange system parameter "No of speed levels" to 3.

Continues on next page

Limitations

FCSpdChgTunSet will not set system parameter to the new value if called inside FCSpdChgAct - FCSpdChgDeact instruction block. It must be called before activating FC Speed Change. The valid value for the system parameters are shown in the following table:

Parameter	Type
Speed ratio min	0.02 - 1.0
No of speed levels	2 - 10

Syntax

```
FCSpdChgTunSet
  [ value ´:=´ ] < expression (IN) of num> ´,´
  [ type ´:=´ ] < expression (IN) of fcspdchgtunetype> ´;´
```

Related information

For information about	See
Set tune parameters to original.	*FCSpdChgTunReset on page 110*

8.1.9. FCSpdChgTunReset

Usage

`FCSpdChgTunReset` reset tuneable FC SpeedChange system parameters to original value stored in configuration.

Basic examples

Basic examples of the instruction `FCSpdChgTunReset` are illustrated below.

See also *More Examples*.

Example

```
FCSpdChgTunReset FC_SPEED_RATIO_MIN
```

Reset FC SpeedChange system parameter "Speed ratio min" to its original value.

Arguments

```
FCSpdChgTunReset type;
```

`type`

Data type: `fcspdchgtunetype`

The FC SpeedChange system parameter whose value is to be reset (FC_SPEED_RATIO_MIN, FC_NO_OF_SPEED_LEVELS). Only two FC SpeedChange system parameters can be reset by this instruction, as shown in the following table:

Parameter	Type
Speed ratio min	FC_SPEED_RATIO_MIN
No of speed levels	FC_NO_OF_SPEED_LEVELS

Program execution

Execution behaviour:

- Reset tunable FC SpeedChange system parameters.

More examples

More examples of how to use the instruction `FCSpdChgTunReset` are illustrated below.

Example

```
FCSpdChgTunReset FC_NO_OF_SPEED_LEVELS;
```

Reset FC SpeedChange system parameter "No of speed levels"

Limitations

FCSpdChgTunReset will not reset system parameter if called inside FCSpdChgAct - FCSpdChgDeact instruction block. It must be called outside the FCSpdChgAct - FCSpdChgDeact instruction block.

Syntax

```
FCSpdChgTunReset
  [ type ':=' ] < expression (IN) of fcspdchgtunetype >';'
```


Continues on next page

Related information

For information about	See
Set tune parameters.	*FCSpdChgTunSet on page 108*

8.1.10. FCSetLPFilterTune

Usage

`FCSetLPFilterTune` is used change the response of force loop according to description in *Damping and LP-filter on page 81*.

Basic examples

Example 1

```
FCSetLPFilterTune  2;
```

Set the force loop cut off frequency to 2 Hz. A low value will make the force control less compliant but more stable.

Arguments

```
FCSetLPFilterTune  CutOffFreq;
```

`CutOffFreq`

Cut off frequency

Data type: `num`

Cut off frequency

Program execution

Execution behaviour:

- Set cut off frequency.

Limitations

Instruction cannot be executed when force control is active.

Syntax

```
FCSetLPFilterTune
  [CutOffFreq ´:=´ ] < expression (IN) of num> ';'
```

Related information

For information about	See
Setting the parameter for low pass filter.	*Bandwidth of force frame filter on page 144*
Instruction how to reset the low pass filter.	*FCResetLPFilterTune on page 113*

CAUTION!

The cut off frequency effects the force loop stability.

8.1.11. FCResetLPFilterTune

Usage

`FCResetLPFilterTune` is used to reset the low pass filter cut off frequency to the configured value. This will change the response of force loop according to description in *Damping and LP-filter on page 81*.

Basic examples

Example

```
FCResetLPFilterTune
```

Resets the low pass filter to configured value.

Arguments

```
FCResetLPFilterTune
```

Program execution

Execution behavior:

- Resets the force loop to the configured cut off frequency value.

Syntax

```
FCResetLPFilterTune;
```

Related information

For information about	See
Setting the parameter for the low pass filter.	*Bandwidth of force loop filter on page 145*
Instruction how to set low pass filter	*FCSetLPFilterTune on page 112*

8.2 Functions

8.2.1. FCGetForce

Usage

The function `FCGetForce` is used to retrieve the force sensor readings. The measured force and torque is returned in a force vector. It is possible to transform the measured force and torque from the force sensor coordinate system to either the tool coordinate system or the work object coordinate system. If the system has been calibrated, i.e. the instruction `FCCalib` has been executed, it is possible to return the force and torque without any offset. In a calibrated system it is also possible to remove the force and torque due to gravity from the sensor readings and only show contact force

Basic example

```
VAR fcforcevector myForceVector;
myForceVector:= FCGetForce();
```

In this example `FCGetForce` gets the values from the sensor and saves it in the variable `myForceVector`. If the system has not been calibrated, using the instruction `FCCalib`, raw measurement data will be returned. That means the sensor offset will be included in the result. If the system has been calibrated, only the force and torque corresponding to the gravity and contact forces will be shown.

See also *More examples on page 115*.

Return value

Data type: `fcforcevector`

The function returns a value of the data type `fcforcevector,` whose components are force and torque in three dimensions (i.e. x, y and z).

Arguments

```
FCGetForce ([\Tool] [\WObj] | [\ContactForce])
```

`[\Tool]`

Data type: `tooldata`

If a tool is specified the returned force will be transformed to the coordinate system of this tool.

`[\WObj]`

Data type: `wobjdata`

If a work object is specified the returned force will be transformed to the coordinate system

`[\ContactForce]`

Data type: `switch`

This option will remove the present gravity force from the result, displaying only contact forces. Note that this option is only allowed if the system has been calibrated before using the function `FCGetForce`.

Continues on next page

Program execution

Execution behavior:

- If the sensor has not been calibrated the returned force is the same as if the sensor had not yet been mounted on the robot, and depends on the sensor manufacturer. Some sensors are offset compensated at startup, which means that the result will be the same as if the sensor had been calibrated, but some are not.
- The resulting torque is in the origin of the new coordinate system. When a transformation is done it is assumed that the contact is in the TCP.

More examples

Example 1

```
VAR fcforcevector myForceVector;
myForceVector:=FCGetForce(\ContactForce);
```

In this example the force and torque due to gravity is removed, meaning that what we see is only contact forces.

Example 2

```
VAR fcforcevector myForceVector;
myForceVector:=FCGetForce(\WObj:=wobj2);
```

In this example the force readings are transformed to the work object coordinate system before returned.

Example 3

```
VAR fcforcevector myForceVector;
myForceVector:=FCGetForce(\Tool:=tool2);
```

In this example the force readings are transformed to the tool coordinate system before returned. It is necessary that the sensor is calibrated or else the function will return an error.

Syntax

```
FCGetForce ´(´
  [´\´ Tool ´:=´ ] < persistent (PERS) of tooldata >
  | [ ´\´ WObj ´:=´ ] < persistent (PERS) of wobjdata >
  [ ´\´ ContactForce ] ´)´
```

A function with a return value of the data type `fcforcevector`.

Related information

For information about	See
The data type fcforcevector	*fcforcevector on page 121*.
Identifying the load	*FCLoadID on page 117*.

8.2.2. FCIsForceMode

Usage

The function `FCIsForceMode` is used to retrieve information whether or not the system is in force control mode.

Basic example

```
VAR bool fc_mode;
fc_mode := FCIsForceMode();
```

In this example FCIsForceMode returns TRUE if force mode is activated.

Return value

Data type: `bool`

The function returns TRUE when force control is activated, FALSE when it is deactivated.

Syntax

```
FCIsForceMode
```

A function with no arguments and a return value of the data type `bool`.

8.2.3. FCLoadID

Usage

The function `FCLoadID` is used to identify the load the sensor feels. Robot axes 5 and 6 move the load while the force sensor is used to detect the mass and center of gravity of the load.

The identified load is returned in a `loaddata` variable. At the present only the mass and center of gravity is identified. Inertias are set to zero. The identified load is used to calibrate the force sensor.

NOTE!

It is important to get an accurate definition of the load in order to get a correct calibration of the sensor. If `loadidErr` is higher than 0.1, the load identification is not optimal and it is recommended to check the sensor coordinate system, see *FC Sensor on page 63*, and the PMC matrix, see *PMC Sensor Setup on page 66*.

NOTE!

`FCLoadID` shall not be used when the sensor is room fixed.

Basic example

```
VAR loaddata my_load;
my_load := FCLoadID();
```

In this example the load in the sensor coordinate system is identified and the value is saved in the variable `my_load`.

See also *More examples on page 119*.

Return value

Data type: `loaddata`

The function returns a variable of type `loaddata` with the identified mass and center of gravity expressed in the force sensor coordinate system.

Arguments

```
FCLoadID ([\MaxMoveAx5] [\MaxMoveAx6]  [\ReadingsPerPoint]
      [\PointsPerAxis] [\loadidErr] [\WarningsOff])
```

`[\MaxMoveAx5]`

Maximum movement of axis 5

Data type: `num`

This parameter decides the maximum movement of robot axis 5 during the load identification procedure. Based on the present position of the robot axis 5, it will move at the most `MaxMoveAx5` degrees in both directions.

The unit is degrees. The default value is 180 degrees.

Continues on next page

`[\MaxMoveAx6]`

Maximum movement of axis 6

Data type: `num`

This parameter decides the maximum movement of robot axis 6 during the load identification procedure. Based on the present position of the robot axis 6, it will move at the most `MaxMoveAx6` degrees in both directions.

The unit is degrees. The default value is 180 degrees.

`[\ReadingsPerPoint]`

Data type: `num`

The number of readings in every point on the axis. The default value is 6 and an average of the readings is calculated.

`[\PointsPerAxis]`

Data type: `num`

The number of points on each axis to make the readings on. The default value is 6. A larger value slows down the identification but may improve the result.

`[\LoadidErr]`

Data type: `num`

`LoadidErr` is an INOUT parameter that returns a value between 0 and 1 depending on the result of the load identification. A value higher than 0.1 indicates that the identification is not optimal.

`[\WarningsOff]`

Data type: `switch`

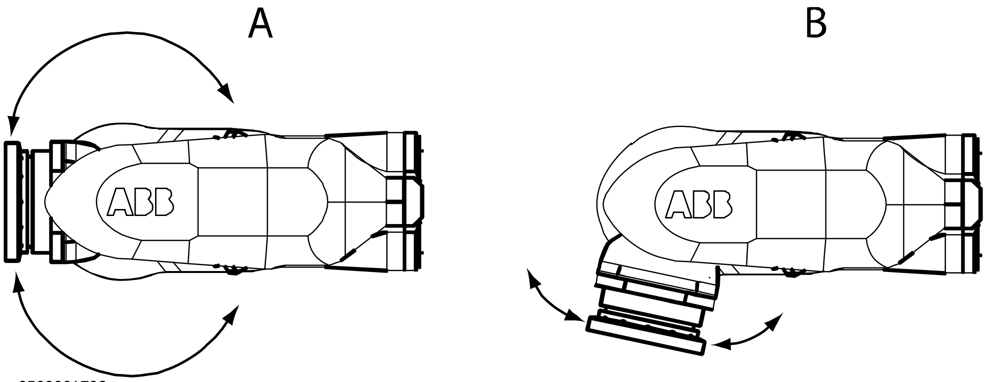
`WarningsOff` is used to disable displaying a message that axis 5 and axis 6 will move outside the specified maximum degrees. The warning message may be useful when running in manual mode, but in automatic mode it is better switched off.

Continues on next page

Program execution

Axis 5 is moved the same angle in both directions from the current position. The same is the case for axis 6.

Even if the movement range is not limited by `MaxMoveAx5` the movement is limited by the robot itself. If the axis is near its end position, the movement will be small and affect the accuracy of the load identification.



xx0500001793

A	The position of axis 5 allows large movements which result in a better load identification.
B	The position of axis 5 allows only a small movement in one direction.The movement in the other direction will be equally small. This results in a poor load identification.

More examples

Example 1

```
VAR num my_status;
VAR loaddata my_sensor_load;
my_sensor_load:=  FCLoadID(\MaxMoveAx5:=30 \MaxMoveAx6:=90
      \ReadingsPerPoint:= 6 \PointsPerAxis:= 5 \LoadIDErr:=
      my_status);
```

The identified load is returned defined in the sensor coordinate system. The robot will move a maximum of 30 degrees in the direction of axis 5 and 90 degrees in direction of axis 6 based on the present robot position. The identification will be based on 6 readings in every point and 5 points on each axis. The variable `my_status` will show the status of the accomplished load identification.

Syntax

```
FCLoadID ´(´
  [ ´\´ MaxMoveAx5 ´:=´ < expression (IN) of num > ]
  [ ´\´ MaxMoveAx6 ´:=´ < expression (IN) of num > ]
  [ ´\´ ReadingsPerPoint ´:=´ < expression (IN) of num > ]
  [ ´\´ PointsPerAxis ´:=´ < expression (IN) of num > ]
  [ ´\´ LoadidErr ´:=´ < expression (INOUT) of num> ]
  [´\´ WarningsOff ] ´)´
```

A function with a return value of the data type `loaddata`.

Continues on next page

Related information

For information about	See
Calibrating the force sensor	*FCCalib on page 91*.

8.3 Data types

8.3.1. fcforcevector

Usage

`fcforcevector` is used by the instruction FCGetForce to return force and torque in three dimensions (i.e. x, y and z).

Components

`xforce`

Data type: `num`

The force in x direction. The unit is N.

`yforce`

Data type: `num`

The force in y direction. The unit is N.

`zforce`

Data type: `num`

The force in z direction. The unit is N.

`xtorque`

Data type: `num`

The torque in x direction. The unit is Nm.

`ytorque`

Data type: `num`

The torque in y direction. The unit is Nm.

`ztorque`

Data type: `num`

The torque in z direction. The unit is Nm.

Examples

Example

```
VAR fcforcevector myForceVector;
myforceVector := FCGetForce();
```

In this example `FCGetForce()` get the values from the sensor and saves it in a variable `myForceVector`.

Continues on next page

Structure

```
< dataobject of fcforcevector >
   < xforce of num >
   < yforce of num >
   < zforce of num >
   < xtorque of num >
   < ytorque of num >
   < ztorque of num >
```

Related information

For information about	See
FCGetForce	*FCGetForce on page 114*

8.3.2. fcframe

Usage

`fcframe` is used by the instruction `FCPress1LStart`, to select which coordinate system should be the reference for the force control coordinate system and the reference movement coordinate system. The alternatives are:

- work object coordinate system
- tool coordinate system
- path coordinate system

Example

```
VAR tooldata tool1:=[TRUE,[[97.4,0,223],[1,0,0,0]], [5,[23,0,75],
      [1,0,0,0],0,0,0]];
VAR fcframe refcoordsys:=FC_REFFRAME_TOOL;
FCPress1LStart tool1 \ForceFrameRef:=refcoordsys;
```

In this example `FCAct` defines a force control coordinate system. This is based on the tool coordinate system, but rotated 180 degrees around the y axis.

Predefined data

Constant	Comment
FC_REFFRAME_TOOL	Tool coordinate system
FC_REFFRAME_WOBJ	Work object coordinate system
FC_REFFRAME_PATH	Path coordinate system

Characteristics

`fcframe` is an alias data type for `num` and consequently inherits its characteristics.

Related information

For information about	See
Starts the press application	*FCPress1LStart on page 93*

8.3.3. fcspdchgtunetype

Usage

`fcspdchgtunetype` is used by the instructions `FCSpdChgTunSet` and `FCSpdChgTunReset` to select which system parameter should be changed.

The alternatives are:

- Speed ratio min
- No of speed levels

Examples

In this example the FC SpeedChange system parameter *Speed ratio min* is set to to 0.2.

```
FCSpdChgTunSet 0.2, FC_SPEED_RATIO_MIN;
```

Predefined data

Constant	Comment
FC_SPEED_RATIO_MIN	Speed ratio min
FC_NO_OF_SPEED_LEVELS	No of speed levels

Characteristics

`fcspdchgtunetype` is an alias data type for num and consequently inherits its characteristics.

Related information

For information about	See
Setting FC SpeedChange system parameter to a new value	*FCSpdChgTunSet on page 108*
Reset FC SpeedChange system parameter to its original value	*FCSpdChgTunReset on page 110*

9 System parameter reference information

9.1 Type Robot

9.1.1. Use FC Master

Parent

Use FC Master belongs to the type *Robot*, in the topic *Motion*.

Cfg name

use_fc_master

Description

Defines which *FC Master* to use for the robot.

Usage

Use FC Master is given the same value as the parameter *Name* of the *FC Master* to use.

Prerequisite

An *FC Master* must be defined.

Allowed values

A string with maximum 32 characters.

Related information

The FC Master type on page 127.

9.1.2. Use PMC Sensor

Parent

Use PMC Sensor belongs to the type *Robot*, in the topic *Motion*.

Cfg name

use_pmc_sensor

Description

Defines which *PMC Sensor* should be used for the robot.

Usage

Use PMC Sensor is given the same value as the parameter *Name* of the *PMC Sensor* to use.

Prerequisites

A *PMC Sensor* must be defined.

Allowed values

A string with maximum 32 characters.

Related information

The PMC Sensor type on page 155.

9.2 Type FC Master

9.2.1. The FC Master type

Overview

This section describes the type *FC Master* which belongs to the topic *Motion*. Each parameter of this type is described in a separate information topic in this section.

Cfg name

FC_MASTER

Type description

The type *FC Master* specifies which *FC Sensor*, *FC Kinematics* and *FC Application* to use for Force Control.

9.2.2. Name

Parent

Name belongs to the type *FC Master*, in the topic *Motion*.

Cfg name

name

Description

The name of the *FC Master*.

Usage

Used to reference the *FC Master* from the parameter *Use FC Master* in the type *Robot*.

Allowed values

A string with maximum 32 characters

9.2.3. Use FC Sensor

Parent

Use FC Sensor belongs to the type *FC Master*, in the topic *Motion*.

Cfg name

use_fc_sensor

Description

Defines which *FC Sensor* to use.

Usage

Use FC Sensor is given the same value as the parameter *Name* of the *FC Sensor* to use.

Prerequisites

An FC sensor must be defined.

Allowed values

A string with maximum 32 characters.

Related information

The FC Sensor type on page 133.

9.2.4. Use FC Kinematics

Parent

Use FC Kinematics belongs to the type *FC Master*, in the topic *Motion*.

Cfg name

use_fc_kinematics

Description

Defines which *FC Kinematics* to use.

Usage

Use FC Kinematics is given the same value as the parameter *Name* of the *FC Kinematics* to use.

Allowed values

A string with maximum 32 characters.

Related information

The FC Kinematics type on page 141

9.2.5. Use FC Application

Parent

Use FC Application belongs to the type *FC Master*, in the topic *Motion*.

Cfg name

use_fc_application

Description

Defines which *FC Application* to use.

Usage

Use FC Application is given the same value as the parameter *Name* in the *FC Application* to use.

Allowed values

A string with maximum 32 characters.

Related information

The FC Application type on page 146

9.2.6. Use FC Speed Change

Parent

Use FC Speed Change belongs to the type *FC Master*, in the topic *Motion*.

Cfg name

use_fc_speed_change

Description

Defines which *FC Speed Change* to use.

Usage

Use FC Speed Change is given the same value as the parameter *Name* in the *FC Speed Change* to use.

Allowed values

A string with maximum 32 characters.

Related information

The FC Application type on page 146.

9.3 Type FC Sensor

9.3.1. The FC Sensor type

Overview

This section describes the type *FC Sensor*, which belongs to the topic *Motion*. Each parameter of this type is described in a separate information topic in this section.

Cfg name

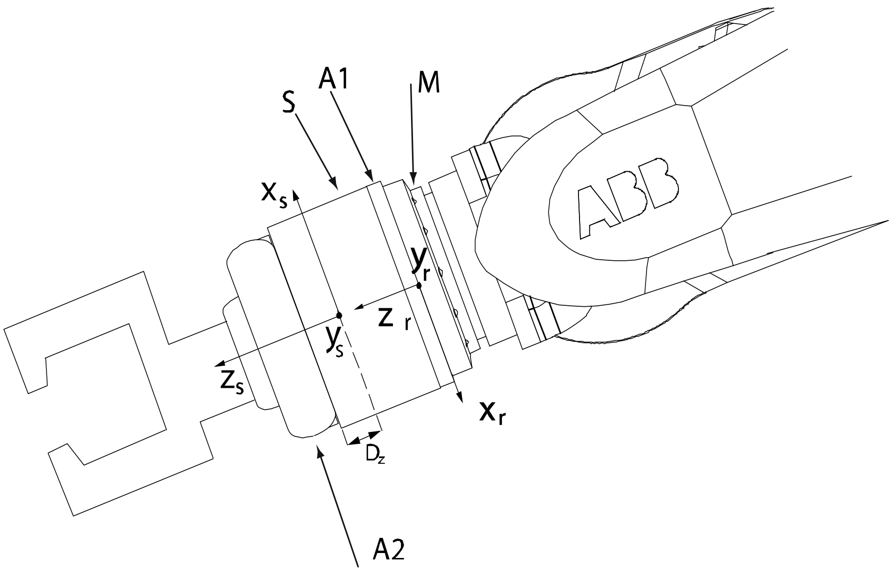
FC_SENSOR

Type description

The type *FC Sensor* is used to define the Force Control sensor. The sensor can be either fixed in the room or mounted on a robot, as specified by the parameter *Force Sensor Mount Unit Name*. It can be a full-fledged sensor measuring both force and torque (6 DOF) or a sensor measuring only force, which is specified by the parameter *Force Sensor Type*.

The sensor has a built in coordinate system measuring forces in x, y and z directions. In order to translate the measured values to other coordinate systems, the sensor coordinate system must be defined. *Force Sensor Frame x - z* defines the position of the sensor coordinate system relative the robot's tool0 coordinate system (robot mounted sensor) or the world coordinate system (room fixed sensor). *Force Sensor Frame q1 - q4* defines the orientation of the sensor coordinate system, relative the robot's tool0 coordinate system (robot mounted sensor) or the world coordinate system (room fixed sensor).

Illustration - robot mounted sensor



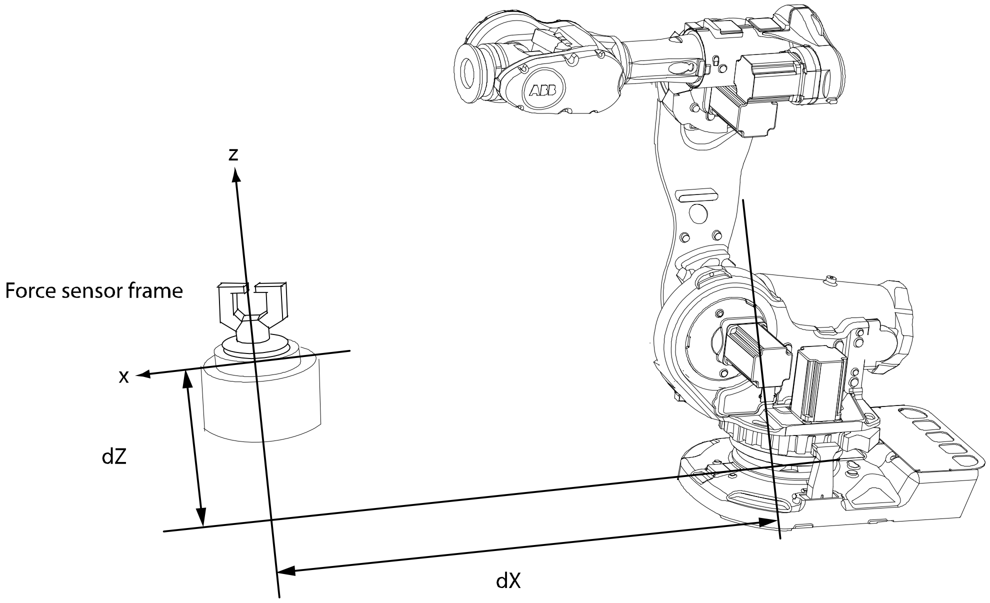
xx0500001554

x_r - z_r	Coordinate system for the robot's tool0.
x_s - z_s	Coordinate system for the sensor.
M	Robot tool flange.
A1	Adaptor plate on the inside (not always used).

Continues on next page

A2	Adaptor plate on the outside (not always used).
D_z	"Basic transform Dz" is the distance between the sensor front and the coordinate system of the sensor.
S	Force sensor.

Illustration - room fixed sensor



xx0600003077

Force Sensor Frame = (dX,0,dZ), (1, 0, 0, 0)

9.3.2. Name

Parent

Name belongs to the type *FC Sensor*, in the topic *Motion*.

Cfg name

name

Description

Defines the name of the *FC Sensor*.

Usage

Name is used to reference the *FC Sensor* from the parameter *Use FC Sensor* in the type *FC Master*.

Allowed values

A string with maximum 32 characters.

NOTE!

The name of the FC sensor must be the same as for the PMC sensor (e.g. fc_sensor1), see also *Name on page 156*

9.3.3. Force Sensor Mount Unit Name

Parent

Force Sensor Mount Unit Name belongs to the type *FC Sensor*, in the topic *Motion*.

Cfg name

force_sensor_mount_unit_name

Description

Defines on which mechanical unit the force sensor is mounted.

Usage

The value should be ROB_1, ROB_2, ROB_3 or ROB_4 when the sensor is mounted on a robot. When the sensor is room fixed the value should be left empty.

Allowed values

ROB_1, ROB_2, ROB_3, ROB_4 or empty.

9.3.4. Force Sensor Type

Parent

Force Sensor Type belongs to the type *FC Sensor*, in the topic *Motion*.

Cfg name

force_sensor_type

Description

Defines the type of sensor.

Usage

If it is a 6 degree of freedom sensor measuring both force and torque the value should be *Force and Torque*. If it is a pure force sensor the value should be *Only Force*.

Allowed values

Force and Torque

Only Force

9.3.5. Force Sensor Frame x - z

Parent

Force Sensor Frame x - z belongs to the type *FC Sensor*, in the topic *Motion*.

Cfg name

force_sensor_frame_pos_x

force_sensor_frame_pos_y

force_sensor_frame_pos_z

Description

Defines the position of the force sensor frame in relation to tool0 (robot mounted sensor) or the world frame (room fixed sensor).

Usage

If the sensor is mounted on a robot the sensor frame is specified with regard to the robot's tool0 coordinate system. *Force Sensor Frame x - z* defines the distance from the center of the robot's mounting plate to the center of the sensor's coordinate system. Note that this coordinate system is not on the outside of the sensor, but normally a few centimeters in from the front of the sensor.

Normally *Force Sensor Frame x* and *Force Sensor Frame y* are set to zero. *Force Sensor Frame z* specifies the thickness of the sensor including the adaptor between robot and sensor if any, minus the distance "Basic Transform D_z" found in the sensor calibration file.

Consult the sensor supplier for detailed data. If the sensor manufacturer cannot provide D_z, it can be identified with a simple experiment. Run the robot to a position where the tool flange is vertical (e.g. robot home position). Use a known weight and hang it at a known distance from the sensor flange. Read signal 205 in test signal viewer, the torque and known weight give a distance (torque arm). D_z = torque_signal 205/(m*g) - known_distance.

If the sensor is fixed in the room the sensor frame is defined in relation to the world frame (robot base in normal cases). Assuming that the sensor z direction is facing the robot 2 meters away from the robot base in the x-direction at 1.5 m height, *Force Sensor Frame x* should be set to 2, Force Sensor Frame y to 0 and Force Sensor Frame to 1.5.

Allowed values

A value between -10 and 10 meters.

TIP!

If the sensor manufactor is ATI, the value is loaded with the sensor parameters from the supplied calibration file.

9.3.6. Force Sensor Frame q1 - q4

Parent

Force Sensor Frame q1 - q4 belong to the type *FC Sensor*, in the topic *Motion*.

Cfg name

force_sensor_frame_orient_u0

force_sensor_frame_orient_u1

force_sensor_frame_orient_u2

force_sensor_frame_orient_u3

Description

Defines the orientation of the force sensor coordinate system with respect to the robot's tool0 coordinate system (robot mounted sensor) or the world coordinate system (room fixed sensor). The orientation is specified as four quaternion values.

Allowed values

A value between -1 and 1.

Related information

For more information on how to calculate quaternions, see the section about the data type `orient` in *Technical reference manual - RAPID Instructions, Functions and Data types*.

9.3.7. Noise level

Parent

Noise level belongs to the type *FC Sensor*, in the topic *Motion*.

Cfg name

force_sensor_noise

Description

Defines the highest noise level at which a force sensor calibration should be allowed. Used to check that the robot is standing still at a force sensor calibration for example.

Usage

If the process is noisy and FCCalib or FC LoadId fails the value can be increased.

Allowed values

A value between 1 and 100.

9.4 Type FC Kinematics

9.4.1. The FC Kinematics type

Overview

This section describes the type *FC Kinematics* which belongs to the topic *Motion*. Each parameter of this type is described in a separate information topic in this section.

Cfg name

FC_KINEMATICS

Type overview

The type *FC Kinematics* is used to define Force Control damping. Damping is a definition of how large contact force is required for the robot to move at a certain speed.

9.4.2. Name

Parent

Name belongs to the type *FC Kinematics*, in the topic *Motion*.

Cfg name

name

Description

Name defines the name of the *FC Kinematics*.

Usage

Name is used to reference the *FC Kinematics* from the parameter *Use FC Kinematics* in the type *FC Master*.

Allowed values

A string with maximum 32 characters.

9.4.3. Damping in Force z Direction

Parent

Damping in Force z Direction belong to the type FC Kinematics, in the topic Motion.

Cfg name

damping_fz

Description

Damping in Force z Direction defines the damping of forces in kinematics in the z direction.

Usage

Defines how many Newtons are required to make the robot move 1 m/s. The higher the value, the less responsive the robot gets.

WARNING!

A too low damping value can make the robot unstable.

Make sure the damping is not too low, even if the tuning level in the `FCAct` instruction is 50%. If the robot drifts away by itself, or if it vibrates, increase the damping value.

Allowed values

A value between min and 10,000,000 Ns/m.

NOTE!

For each robot type there exists minimum allowed values of the damping. It is not possible to set the damping lower than these values.

9.4.4. Bandwidth of force frame filter

Parent

Bandwidth of force frame filter belongs to the type *FC Kinematics*, in the topic *Motion*.

Cfg name

force_frame_ filter_bandwidth

Description

Bandwidth of force frame filter defines the bandwidth in Hz of a low pass filter used to filter measured forces, used for example in force conditions.

Usage

In applications where the measured force/torque are too noisy this parameter can be used to filter the signals in order to eliminate false triggering and errors.

Allowed values

A value between 0 and 125. A value larger than 100 will switch the filter off.

9.4.5. Bandwidth of force loop filter

Parent

Bandwidth of force loop filter belongs to the type *FC Kinematics*, in the topic *Motion*.

Cfg name

force_loop_filter_bandwidth

Description

Bandwidth of force loop filter defines the bandwidth in Hz of a low pass filter used in the force control loop.

Usage

If the robot reacts too slowly for changes in contact force an increase of this parameter will make the robot more adaptable. Too high value will cause instability.

Allowed values

A value between 0.1 and 250. Default value is 3 Hz. A higher value will make the robot move compliant but may cause instability.

9.5 Type FC Application

9.5.1. The FC Application type

Overview

This section describes the type *FC Application* which belongs to the topic *Motion*. Each parameter of this type is described in a separate information topic in this section.

Cfg name

FC_APPLICATION

Type description

The type *FC Application* defines a number of limits for the reference values used in Force Control.

9.5.2. Name

Parent

Name belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

name

Description

Name defines the name of the *FC Application*.

Usage

Name is used to reference the *FC Application* from the parameter *Use FC Application* in the type *FC Master*.

Allowed values

A string with maximum 32 characters.

9.5.3. Max Ref Force

Parent

Max Ref Force belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

max_force

Description

Max Ref Force defines the maximum allowed reference force.

Usage

The reference force, specified by the instruction `FCRefForce`, cannot be larger than *Max Ref Force*. If a larger value is used in `FCRefForce` the execution will continue with a reference force equal to *Max Ref Force* and a warning is shown in the event log.

Allowed values

A value between 0 and max.

NOTE!

There is a maximum allowed reference that depends on the robots default load.

This maximum force is equal to the robots type default load, times the gravity constant.

For a 60kg default load the highest allowed force is 60*9,81=589N.

If a higher value is set in *Max Ref Force* than the maximum allowed the higher value will be ignored.

9.5.4. Max Ref Force Change

Parent

Max Ref Force Change belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

max_force_change

Description

Max Ref Force Change defines the maximum allowed change in the reference force.

Usage

When the instruction `FCRefStart` is executed, the force is ramped up to the desired reference force specified in `FCRefForce`. This ramping is determined by *Max Ref Force Change*.

If a very large oscillating reference force is specified in `FCRefForce`, the oscillations are limited so that the change in force never exceeds *Max Ref Force Change*.

Allowed values

A value between 0 and 10,000 N/s.

9.5.5. Speed superv override

Parent

Speed superv override belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

spd_superv_override_factor

Description

Speed superv override is a parameter used to modify the default speed supervision. This parameter modifies the speed superv in force control mode by a factor from 1 to 20.

Usage

When the robot is in force control mode the speed supervision might trig even during normal usage. If this happens it can be adjusted by defining a higher value of the parameter `speed superv override`

Allowed values

A value between 1and 20.

9.5.6. Largest measured contact force

Parent

Largest measured contact force belongs to the type FC Application, in the topic Motion.

Cfg name

axc_force_max

Description

If measured contact force is larger than this value it is set to this value. The unit is N. The default value is 100 000 meaning this functionality is not active.

Usage

The parameter Largest measured contact force defines a truncation of measured force. If a measured force is larger than this value it is still assumed to be equal to this value. This can be useful in certain lead-tech applians but should otherwise not be used.

Allowed values

A value between 0 and 100 000.

9.5.7. Lowest measured contact force

Parent

Lowest measured contact force belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

axc_force_min

Description

If measured contact force is less than this value, it is set to zero. The unit is N.

Usage

Lowest measured contact force defines a threshold for the force, which needs to be exceeded in order to effect the robot position/speed. Too low a value might cause the robot to drift.

Allowed values

A value between 0 and 1000.

9.5.8. Max Press TCP Speed

Parent

Max Press TCP Speed belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

max_lin_speed_press

Description

Defines the maximum allowed TCP speed in FC Press instructions.

Usage

This parameter defines the highest TCP speed that can be used in FC Press instructions.

Allowed values

A value between 0 and 10 (m/s).

9.5.9. Max Press Rot Speed

Parent

Max Press Rot Speed belongs to the type *FC Application*, in the topic *Motion*.

Cfg name

max_rot_speed_press

Description

Defines the maximum allowed reorient speed in FC Press instructions.

Usage

This parameter defines the highest reorient speed that can be used in FC Press instructions.

Allowed values

A value between 0.01 and 1 (rad/s).

9.6 Type PMC Sensor

9.6.1. The PMC Sensor type

Overview

This section describes the type *PMC Sensor*, which belongs to the topic *Motion*. Each parameter of the type is described in a separate information topic in this section.

Cfg name

PMC_SENSOR

Type description

The type *PMC Sensor* describes a sensor connected to the robot controller via a PCI Mezzanine Card.

9.6.2. Name

Parent

Name belongs to the type *PMC Sensor*, in the topic *Motion*.

Cfg name

name

Description

Defines the name of the *PMC Sensor*.

Usage

Used to reference the *PMC Sensor* from the parameter *Use PMC Sensor* in the type *Robot*.

Allowed values

A string with maximum 32 characters.

NOTE!

The name of the PMC sensor must be the same as for the FC sensor (e.g. fc_sensor1), see also *Name on page 135*.

9.6.3. Use PMC Sensor Setup

Parent

Use PMC Sensor Setup belongs to the type *PMC Sensor*, in the topic *Motion*.

Cfg name

use_pmc_sensor_setup

Description

Defines which *PMC Sensor Setup* to use.

Usage

Use PMC Sensor Setup is given the same value as the parameter *Name* of the *PMC Sensor Setup* to use.

Prerequisites

A *PMC Sensor Setup* must be defined.

Allowed values

A string with maximum 32 characters defining a PMC sensor setup.

Related information

The PMC Sensor Setup type on page 158

9.7 Type PMC Sensor Setup

9.7.1. The PMC Sensor Setup type

Overview

This section describes the type *PMC Sensor Setup*, which belongs to the topic *Motion*. Each parameter of the type is described in a separate information topic in this section.

Cfg name

PMC_SENSOR_SETUP

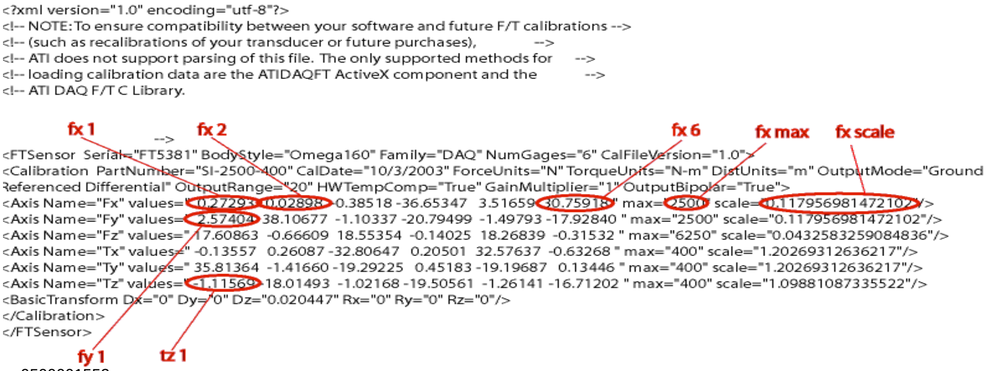
Type description

The type *PMC Sensor Setup* is used to specify sensor calibration data.

The sensor calibration data is delivered from the sensor supplier in a file.

Illustration

This is an example of a sensor calibration file from ATI, showing where to find different values. Set these values for the respective parameter.


```
<?xml version="1.0" encoding="utf-8"?>
<!-- NOTE: To ensure compatibility between your software and future F/T calibrations -->
<!-- (such as recalibrations of your transducer or future purchases),              -->
<!-- ATI does not support parsing of this file. The only supported methods for     -->
<!-- loading calibration data are the ATIDAQFT ActiveX component and the           -->
<!-- ATI DAQ F/T C Library.
                                     -->
<FTSensor  Serial="FT5381" BodyStyle="Omega160" Family="DAQ" NumGages="6" CalFileVersion="1.0">
<Calibration  PartNumber="SI-2500-400" CalDate="10/3/2003" ForceUnits="N" TorqueUnits="N-m" DistUnits="m" OutputMode="Ground Referenced Differential" OutputRange="20" HWTempComp="True" GainMultiplier="1" OutputBipolar="True">
<Axis Name="Fx" values=" 0.27293  0.02898 -0.38518 -36.65347  3.51659  30.75918 " max="2500" scale="0.117956981472102"/>
<Axis Name="Fy" values=" 2.57404  38.10677  -1.10337 -20.79499  -1.49793 -17.92840 " max="2500" scale="0.117956981472102"/>
<Axis Name="Fz" values=" 17.60863  -0.66609  18.55354  -0.14025  18.26839  -0.31532 " max="6250" scale="0.0432583259084836"/>
<Axis Name="Tx" values=" -0.13557  0.26087 -32.80647  0.20501  32.57637  -0.63268 " max="400" scale="1.20269312636217"/>
<Axis Name="Ty" values=" 35.81364  -1.41660 -19.29225  0.45183 -19.19687  0.13446 " max="400" scale="1.20269312636217"/>
<Axis Name="Tz" values=" -1.11569  -18.01493  -1.02168 -19.50561  -1.26141 -16.71202 " max="400" scale="1.09881087335522"/>
<BasicTransform Dx="0" Dy="0" Dz="0.020447" Rx="0" Ry="0" Rz="0"/>
</Calibration>
</FTSensor>
```


en0500001553

TIP!

If the sensor manufacturer is ATI Automation it is possible to load the sensor parameters from the supplied CD instead of manually typing them in RobotStudio. For information on how to load the sensor parameters, see *Procedure on page 15*.

CAUTION!

Remember to always double-check your sensor calibration before running anything the first time.

9.7.2. Name

Parent

Name belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

name

Description

Defines the name of the *PMC Sensor Setup*.

Usage

Used to reference the *PMC Sensor Setup* from the parameter *Use PMC Sensor Setup* in the type *PMC Sensor*.

Allowed values

A string with maximum 32 characters.

9.7.3. Sensor serial number

Parent

Sensor serial number belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

serial_number

Description

Defines the serial number of the sensor.

Usage

Sensor serial number is used to compare the serial number on the physical sensor with the serial number in the configuration file.

Allowed values

A string with maximum 32 characters.

9.7.4. Disable check of saturation

Parent

Disable check of saturation belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

saturation_check_disabled

Description

By default there is a saturation check of voltage input from force/torque sensor. This parameter can disable or enable this check.

Usage

If the force sensor in use does not work correctly with the saturation check, set the value to TRUE to disable the supervision.

Allowed values

TRUE/FALSE.

9.7.5. Time in saturation before error

Parent

Time in saturation before error belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

max_time_in_sat_err

Description

Defines the time in seconds that output levels from force/torque sensor should be in saturation before error message appears.

Usage

Set the desired time sensor gauges that should saturate before error message appears.

Allowed values

A numeric value.

Default: 1 s

9.7.6. Time in saturation before warning

Parent

Time in saturation before warning belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

max_time_in_sat_warn

Description

Defines the time in seconds that output levels from force/torque sensor should be in saturation before warning appears.

Usage

Set the desired time sensor gauges that should saturate before warning appears.

Allowed values

A numeric value.

Default: 0.5 s

9.7.7. fx 1 - fx 6

Parent

fx 1 - *fx 6* belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

fx_value1

fx_value2

fx_value3

fx_value4

fx_value5

fx_value6

Description

Defines how the six input voltages are scaled to create a measured force in the x direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the force in the x direction, while others are more dependent on forces in other directions. The contribution from the first input voltage to the force measurement in the x direction is specified in *fx 1*. The contribution from the second signal is specified in *fx 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured force in the x direction is calculated:

$$Fx = \frac{fx1}{fxScale}U_0 + \frac{fx2}{fxScale}U_1 + \frac{fx3}{fxScale}U_2 + \frac{fx4}{fxScale}U_3 + \frac{fx5}{fxScale}U_4 + \frac{fx6}{fxScale}U_5$$

xx0600003087

Allowed values

A numeric value.

9.7.8. fy 1 - fy 6

Parent

fy 1 - fy 6 belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

fy_value1

fy_value2

fy_value3

fy_value4

fy_value5

fy_value6

Description

Defines how the six input voltages are scaled to create a measured force in the y direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the force in the y direction, while others are more dependent on forces in other directions. The contribution from the first input voltage to the force measurement in the y direction is specified in *fy 1*. The contribution from the second signal is specified in *fy 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured force in the y direction is calculated:

$$Fy = \frac{fy1}{fyScale}U_0 + \frac{fy2}{fyScale}U_1 + \frac{fy3}{fyScale}U_2 + \frac{fy4}{fyScale}U_3 + \frac{fy5}{fyScale}U_4 + \frac{fy6}{fyScale}U_5$$

xx0600003088

Allowed values

A numeric value.

9.7.9. fz 1 - fz 6

Parent

fz 1 - fz 6 belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

fz_value1

fz_value2

fz_value3

fz_value4

fz_value5

fz_value6

Description

Defines how the six input voltages are scaled to create a measured force in the z direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the force in the z direction, while others are more dependent on forces in other directions. The contribution from the first input voltage to the force measurement in the z direction is specified in *fz 1*. The contribution from the second signal is specified in *fz 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured force in the z direction is calculated:

$$Fz = \frac{fz1}{fzScale}U_0 + \frac{fz2}{fzScale}U_1 + \frac{fz3}{fzScale}U_2 + \frac{fz4}{fzScale}U_3 + \frac{fz5}{fzScale}U_4 + \frac{fz6}{fzScale}U_5$$

xx0600003089

Allowed values

A numeric value.

9.7.10. tx 1 - tx 6

Parent

tx 1 - tx 6 belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

tx_value1

tx_value2

tx_value3

tx_value4

tx_value5

tx_value6

Description

Defines how the six input voltages are scaled to create a measured torque in the x direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the torque in the x direction, while others are more dependent on torques in other directions. The contribution from the first input voltage to the torque measurement in the x direction is specified in *tx 1*. The contribution from the second signal is specified in *tx 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured torque in the x direction is calculated:

$$Tx = \frac{tx1}{txScale}U_0 + \frac{tx2}{txScale}U_1 + \frac{tx3}{txScale}U_2 + \frac{tx4}{txScale}U_3 + \frac{tx5}{txScale}U_4 + \frac{tx6}{txScale}U_5$$

xx0600003090

Allowed values

A numeric value.

9.7.11. ty 1 - ty 6

Parent

ty 1 - ty 6 belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

ty_value1

ty_value2

ty_value3

ty_value4

ty_value5

ty_value6

Description

Defines how the six input voltages are scaled to create a measured torque in the y direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the torque in the y direction, while others are more dependent on torques in other directions. The contribution from the first input voltage to the torque measurement in the y direction is specified in *ty 1*. The contribution from the second signal is specified in *ty 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured torque in the y direction is calculated:

$$Ty = \frac{ty1}{tyScale}U_0 + \frac{ty2}{tyScale}U_1 + \frac{ty3}{tyScale}U_2 + \frac{ty4}{tyScale}U_3 + \frac{ty5}{tyScale}U_4 + \frac{ty6}{tyScale}U_5$$

xx0600003091

Allowed values

A numeric value.

9.7.12. tz 1 - tz 6

Parent

tz 1 - tz 6 belong to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

tz_value1

tz_value2

tz_value3

tz_value4

tz_value5

tz_value6

Description

Defines how the six input voltages are scaled to create a measured torque in the z direction.

Usage

The force sensor sends six input voltages to the DAQ board (U_0, U_1, ... U_5). Some of these are very dependent on the torque in the z direction, while others are more dependent on torques in other directions. The contribution from the first input voltage to the torque measurement in the z direction is specified in *tz 1*. The contribution from the second signal is specified in *tz 2*, and so on.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Example

This is how the measured torque in the z direction is calculated:

$$Tz = \frac{tz1}{tzScale}U_0 + \frac{tz2}{tzScale}U_1 + \frac{tz3}{tzScale}U_2 + \frac{tz4}{tzScale}U_3 + \frac{tz5}{tzScale}U_4 + \frac{tz6}{tzScale}U_5$$

xx0600003092

Allowed values

A numeric value.

9.7.13. fx scale - tz scale

Parent

fx scale - *tz scale* belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

fx_scale

fy_scale

fz_scale

tx_scale

ty_scale

tz_scale

Description

Scaling factor for the weighted output from the sensor.

Usage

The force sensor sends sixinput voltages to the DAQ board (U_0, U_1, ... U_5). The voltage of each signal is multiplied by a factor (e.g. *fx 1* to *fx 6*). The results are summed up to a weighted output from the sensor (e.g. of the force in the x direction). By dividing this weighted sensor output with a scaling factor, the corresponding force (or torque) is calculated.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Allowed values

A numerical value.

Unit: V/N (for *fx scale* - *fz scale*) or V/Nm (for *tx scale* - *tz scale*)

9.7.14. fx max - tz max

Parent

fx max - tz max belongs to the type *PMC Sensor Setup*, in the topic *Motion*.

Cfg name

fx_value_max

fy_value_max

fz_value_max

tx_value_max

ty_value_max

tz_value_max

Description

The sensor's specified maximum force or torque in the respective direction.

Usage

fx max specifies the maximum force in the x direction that the sensor can measure reliably. In the same way, *fy max* and *fz max* specify the maximum force in the y and z direction, respectively. *tx max*, *ty max* and *tz max* specify the maximum torque in the x, y and z direction, respectively.

These values are specified by the sensor supplier, see *The PMC Sensor Setup type on page 158*.

Allowed values

A numeric value.

Unit: N (for *fx max* - *fz max*) or Nm (for *tx max* - *tz max*)

9.7.15. Max voltage for external AD card

Parent

Max voltage for external AD card belongs to the type *PMC Sensor*, in the topic *Motion*.

Cfg Name

max_voltage

Description

The value should be set to the voltage working range of the AD board. The range is assumed to be + - this value.

Usage

If the AD board hardware is configured for + - 10V as working range, set the value to 10.

Allowed values

0-100

9.7.16. Disable Force Sensor Cable Check

Parent

Disable Force Sensor Cable Check belongs to the type *PMC Sensor Setup* in the topic *Motion*.

Cfg name

safety_channel_disabled

Description

Some force sensors have a separate safety channel, which delivers a voltage above a certain level when the sensor is working. Safety channel is always connected to input U6 on the DAQ board.

Usage

If the force sensor in use does not have the safety channel feature, set this value to TRUE and disable the supervision.

Allowed values

TRUE/FALSE

9.7.17. Safety Channel Voltage

Parent

Safety Channel Voltage belongs to the type *PMC Sensor Setup* in the topic *Motion*.

Cfg name

safety_channel_level

Description

Defines the minimum allowed voltage from the safety channel. The absolute value of the safety channel input is compared with this value.

Usage

If a separate safety channel is used with the force sensor, set this value to the minimum allowed voltage before the system is stopped.

Allowed values

0-100

9.8 Type FC Speed Change

9.8.1. The FC Speed Change Type

Overview

This section describes the type *FC Speed Change*, which belongs to the topic *Motion*. Each parameter of the type is described in a separate information topic in this section.

Cfg name

FC_SPEED_CHANGE

Type description

The type *FC Speed Change* has a number of parameters used for the SpeedChange functionality available with the RobotWare option Machining FC.

9.8.2. Name

Parent

Name belongs to the type *FC Speed Change*, in the topic *Motion*.

Cfg name

name

Description

Defines the name of the *FC Speed Change*.

Usage

Name is used to reference the *FC Speed Change* from the parameter *Use FC Speed Change* in the type *FC Master.*

Allowed values

A string with maximum 32 characters.

9.8.3. Speed ratio min

Parent

Speed ratio min belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

speed_ratio_min

Description

Defines the minimum allowed speed ratio.

Usage

This parameter defines the lowest robot speed to be used (speed_ratio_min * programmed_speed).

Allowed values

A value between 0.02 and 1.

9.8.4. No of speed levels

Parent

No of speed levels belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

no_of_speed_levels

Description

Defines the number of discrete speed levels.

Usage

This parameter defines the number of discrete speed levels to be used.

Allowed values

A value between 2 and 10.

9.8.5. Speed ratio delta

Parent

Speed ratio delta belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

speed_ratio_delta

Description

Limits the speed change when ramping the speed ratio to a new value.

Usage

The parameter limits acceleration/deceleration due to the SpeedChange functionality. A low value will give slower but smoother speed changes.

Allowed values

A value between 0.0001 and 1.

9.8.6. Speed max update period

Parent

Speed max update period belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

speed_max_update_period

Description

Defines the minimum time in seconds between speed changes.

Usage

This parameter defines the period of time after a speed change, during which the sensor signal will be disregarded. A short time will give faster reactions to overload but may cause the speed to vary too frequently.

Allowed values

A value between 0 and 1.

9.8.7. Feedback type

Parent

Feedback type belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

fdb_type

Description

Defines the type of feedback to be used.

Usage

This parameter is used to decide which type of feedback should control the speed, i e which sensor input is to be used for speed change control.

NOTE! Parameter "Disable check of saturation" can be used if it is likely that the poer output will reach saturation level.

Allowed values

Single DAC Input

Calib. Force Magn.

Uncalib. Force Magn.

9.8.8. DAC channel

Parent

DAC channel belongs to the type *FC speed change* in topic *Motion*.

Cfg name

fdb_dac_channel

Description

Defines the channel used on AD board when running Force Controlled SpeedChange in Single DAC Input mode.

Usage

Chose channel to have as analog input from AD board.

NOTE: In the ATI case channel 6 and 7 are reserved .

Allowed values

A value between 0 and 8 with a default value of 8.

9.8.9. Feedback offset

Parent

Feedback offset belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

fdb_offset

Description

Defines the offset to be removed from the feedback.

Usage

The offset is removed from the measured feedback before the value is used in speed change control. Default value is 0.

Allowed values

A value between -10000 and 10000.

9.8.10. Use Fdb LP filter

Parent

Use Fdb LP filter belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

fdb_lp_active

Description

Defines whether feedback low pass filter should be active.

Usage

If set to true, the feedback values are low pass filtered before used. May be used to filter noisy signals.

Allowed values

TRUE/FALSE.

9.8.11. Fdb LP filter bandwidth

Parent

Fdb LP filter bandwidth belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

fdb_lp_bandwidth

Description

Defines the bandwidth for feedback low pass filter (Hz).

Usage

This parameter is used to filter the feedback values used in the speed change control. Setting it lower will slow down the reaction time for the speed change control.

Allowed values

A value between 1 and 250.

9.8.12. Maximum TCP speed

Parent

Maximum TCP speed belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

maximum_tcp_speed

Description

Defines the maximum original TCP speed for speed change (m/s).

Usage

If the user programs a speed above this value, the system will stop.

Allowed values

A value between 0.01 and 10.

9.8.13. Recover rule fdb ratio

Parent

Recover rule fbd ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

recover_rule_fdb ratio

Description

Defines the maximum allowed feedback (fdb) to reference ratio when at lowest possible speed.

Usage

A feedback to reference ratio larger than this while having reduced speed to lowest level will trig recover behavior or stop robot. The recover function will be activated when the feedback signal is still too high when running at the lowest speed.

Allowed values

A value between 0.01 and 1000.

9.8.14. Decrease rule safety fdb ratio

Parent

Decrease rule safety fdb ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_ rule_safety_fdb_to_ref_ratio

Description

Defines the maximum feedback to reference ratio.

Usage

Speed will be reduced if the feedback to reference ratio is above this value for *Decrease rule safety fdb time* regardless of trends and changes of the feedback.

Allowed values

A value between 0.001 and 1000.

9.8.15. Decrease rule safety fdb time

Parent

Decrease rule safety fdb time belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_ rule_safety_fdb_time

Description

Define the maximum time in seconds that the feedback to reference ratio can be continuously over Decrease rule safety fdb ratio before reducing robot speed

Usage

Speed will be reduced if the feedback to reference ratio is above Decrease rule safety fdb ratio for this time regardless the trend of the feedback

Allowed values

A value between 0.001 and 1000.

9.8.16. Fdb trend step size

Parent

Fdb trend step size belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

fdb_trend_step_size

Description

Defines the minimum difference between consecutive fdb values to count as a change in feedback.

Usage

Used for deciding trends in the feedback that is needed for the SpeedChange rules. This parameter is used to compensate the effects of measurement noise on the trend calculation. Usually the value can be set 2 times the noise level

Allowed values

A value between 0 and 1000.

9.8.17. Decrease rule 1 fdb ratio

Parent

Decrease rule 1 fdb ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_rule1_fdb_to_ref_ratio

Description

Part of condition 1 to decrease speed.

Usage

For ABB internal use only.

Allowed values

A value between 0.001 and 1000.

9.8.18. Decrease rule 1 fdb trend

Parent

Decrease rule 1 fdb trend belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_rule1_fdb_trend

Description

Part of condition 1to decrease speed.

Usage

For ABB internal use only.

Allowed values

A value between -10 and 10.

9.8.19. Decrease rule 2 fdb ratio

Parent

Decrease rule 2 fdb ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_rule2_fdb_to_ref_ratio

Description

Part of condition 2 to decrease speed.

Usage

For ABB internal use only.

Allowed values

A value between 0.001 and 1000.

9.8.20. Decrease rule 2 fdb trend

Parent

Decrease rule 2 fdb trend belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

dec_rule2_fdb_trend

Description

Part of condition 2to decrease speed.

Usage

For ABB internal use only.

Allowed values

A value between -10 and 10.

9.8.21. Increase rule 1 fdb ratio

Parent

Increase rule 1 fdb ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

inc_rule1_fdb_to_ref_ratio

Description

Part of condition 1to increase speed.

Usage

For ABB internal use only.

Allowed values

A value between 0.001 and 1000.

9.8.22. Increase rule 1 fdb trend

Parent

Increase rule 1 fdb trend belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

inc_rule1_fdb_trend

Description

Part of condition 1to increase speed.

Usage

For ABB internal use only.

Allowed values

A value between -10 and 10.

9.8.23. Increase rule 2 fdb ratio

Parent

Increase rule 2 fdb ratio belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

inc_rule2_fdb_to_ref_ratio

Description

Part of condition 2 to increase speed.

Usage

For ABB internal use only.

Allowed values

A value between 0.001 and 1000.

9.8.24. Increase rule 2 fdb trend

Parent

Increase rule 2 fdb trend belongs to the type *FC Speed Change* in topic *Motion*.

Cfg name

inc_rule2_fdb_trend

Description

Part of condition 2 to increase speed.

Usage

For ABB internal use only.

Allowed values

A value between -10 and 10.

10 Further references

10.1. Test Signal Viewer

Overview

Test signal Viewer can be used to monitor forces in Force Control. Forces and torques in both force control coordinate system and sensor coordinate system can be viewed at the same time monitoring 6 signals each (e.g. x., y, z, wx, wy, wz).

Installation

This section only describes what is specific for setting up the Test signal viewer for Force control. For more information see *Test signal manual*.

Procedure

Follow these step to get started with the program.

Step	Action	Reference
1.	Install the Test Signal Viewer.	Included on the RobotWare CD. Browse to *CD contents/tools/Testsignal viewer/setup.exe*.
2.	Start using the program	Test signal viewer manual.
3.	Define Test Signals.	Specified in section *Test Signal number on page 199*.

Test Signal number

To view the forces a specific number needs to be entered in “Signal ident. man” of every channel in the test signal window.

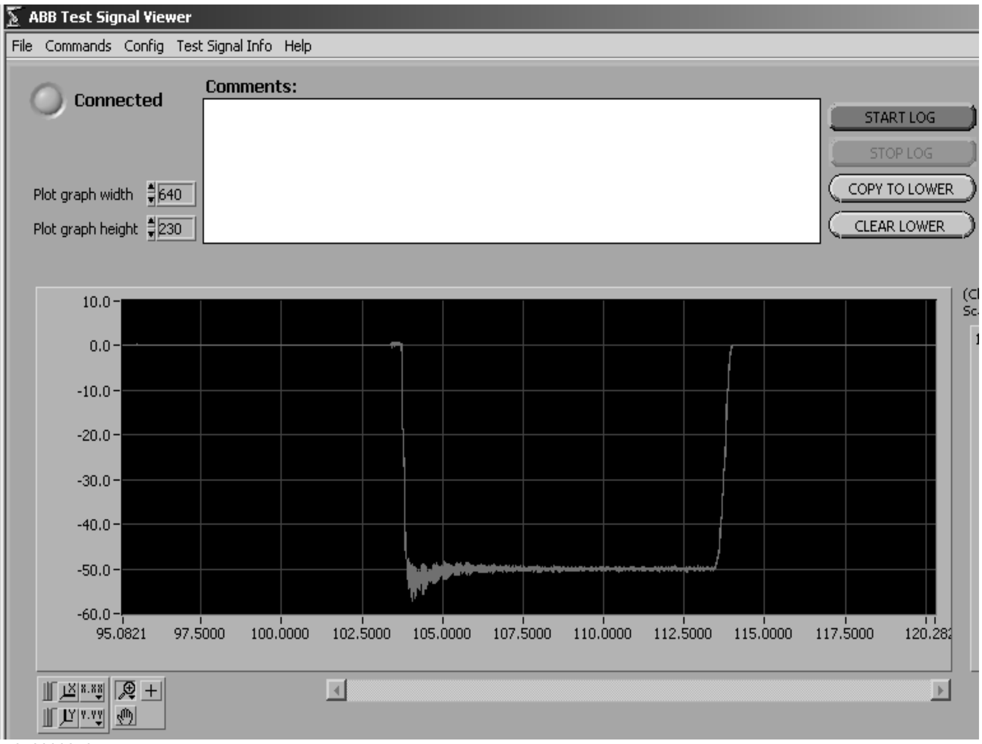
If the signals are not mapped to a specific axis always use axis 1.

It is important to note that in FC Pressure, the force control coordinate system is automatically rotated in such way that the z-axis of the force control coordinate system is always aligned with the pressure direction specified by the arguments `\Fx`, `\Fy` and `\Fz` to `FCPress1LStart`. This means that the test signal 209 should always be used for monitoring of the pressure force.

Signal	Content (force component)
201	Sensor frame, x-direction.
202	Sensor frame, y-direction.
203	Sensor frame, z-direction.
204	Sensor frame, wx-direction.
205	Sensor frame, wy-direction.
206	Sensor frame, wz-direction.
207	Force frame, x-direction.
208	Force frame, y-direction.
209	Force frame, z-direction.
210	Force frame, wx-direction.

Continues on next page

Signal	Content (force component)
211	Force frame, wy-direction.
212	Force frame, wz-direction.



xx0500002465

To view the raw sensor signals specific numbers must be entered in the “Signal ident. Man” using the channels in the test signal viewer window.

Signal number	Content
1001	U0
1002	U1
1003	U2
1004	U3
1005	U4
1006	U5
1007	Safety Channel
1008	Always zero
1009	Single channel for SpeedChange

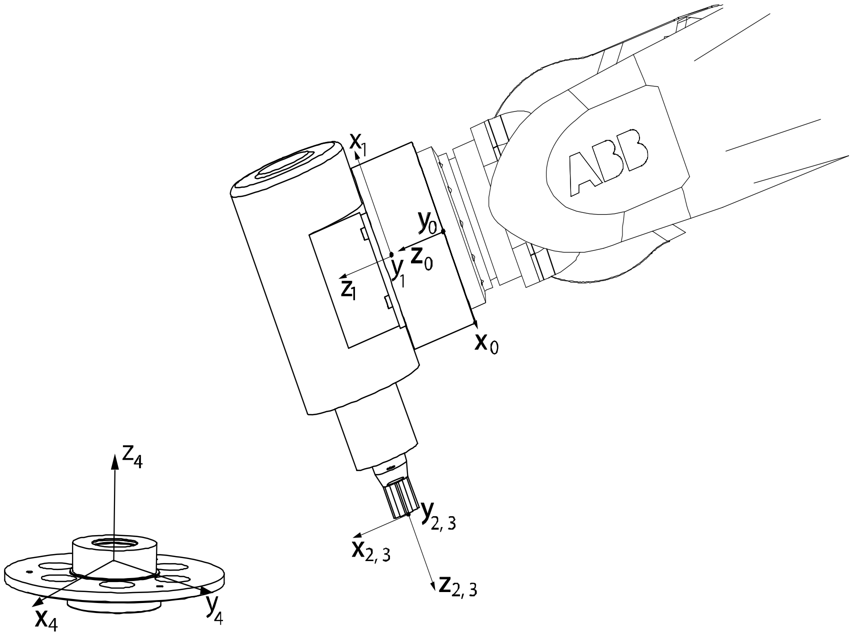
This table specifies the test signals for FCSpeedChange tuning.

Signal number	Content
401	Reference
402	Measurement (process force)
403	Speed ratio signal

10.2. The Coordinate systems

About the coordinate systems

This is an overview over the new coordinate systems created for Force Control. For more information about the basic coordination system see related information.



xx0600003309

X0, Y0, Z0	Tool 0 coordinate system
X1, Y1, Z1	Sensor coordinate system
X2, Y2, Z2	Tool coordinate system
X3, Y3, Z3	Force control coordinate system
X4, Y4, Z4	Work object coordinate system

Force control coordinate system

The origin of the force control coordinate system is in the tool center point (TCP). The orientation is defined by the user in relation to the tool coordinate system, the work object coordinate system, or the path coordinate system.

Sensor coordinate system

The origin and orientation of the sensor coordinate system depends on the manufacture and how it is mounted.

Tool 0 coordinate system

Tool 0 or the wrist coordinate system cannot be changed and is always the same as the mounting flange of the robot.

Tool coordinate system

The tool coordinate system is defined by the user.

Continues on next page

Continued

Related information

For information about	See
the coordinate systems	*Technical reference manual - RAPID overview*

10.3 Force Sensor interface

10.3.1. About the Force Sensor interface

Overview

The force control software has been designed with a generic interface for a 6 degree of freedom (6DOF) force / torque sensor using an Acromag Data Acquisition board (Analog to digital data collection board). As a primary supplier ATI Industrial Automation has been involved and ATI have also prepared sensors with correct settings, adapter plates and calibration files in order to meet the ABB interface requirements. ATI also supplies an Acromag DAQ with correct settings to meet both ABB and ATI requirements.This section describes the interface as such and how to adapt a sensor to the requirements.

It is also possible to use a sensor with less than 6 DOF, which means that the robot is force controlled in less directions.

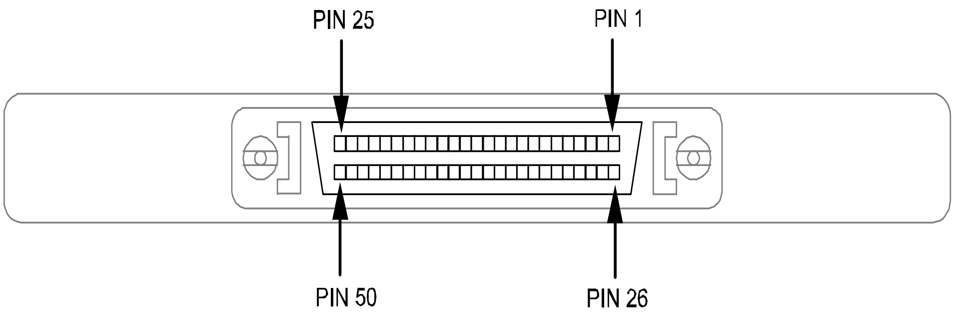
10.3.2. Acromag DAQ board

Acromag mezzanine Card PMC330

The ABB software supports a PMC card from Acromag (Series PMC330).6 channels are used to read the force/torque values of a 6 DOF sensor and force from a 1 DOF sensor. One channel is used for a sensor safety signal and one channel is used for the spindle sensor signal used for SpeedChange. There are DIP switches on the card to modify the input range of the card. The maximum range is +- 10V. Also the supply voltage to the sensor may be changed (pin 45 and 48). See the manufacturer manual for information on how to set it up.

I/O pin configuration

This is a description of the Acromag DAQ board sensor connector. The cable connector should be a scsi-2 (AMP) male connector to match the board connector.



xx0600003073

Acromag DAQ board sensor connector

The cable connector pin assignment is:

Pin	Description	Pin	Description
1	Signal U0 +	15	COMMON
2	Signal U0 -	16	Signal U5 +
3	COMMON	17	Signal U5 -
4	Signal U1 +	18	COMMON
5	Signal U1 -	19	Safety + (u6)
6	COMMON	20	Safety - (u6)
7	Signal U2 +	21	COMMON
8	Signal U2 -	25	SpeedChange single channel + (u8)
9	COMMON	26	SpeedChange single channel - (u9)
10	Signal U3 +	27	COMMON
11	Signal U3 -	45	+12 V
12	COMMON	48	-12 V
13	Signal U4 +	50	SHIELD
14	Signal U4 -		

10.3.3. Sensor configuration

Procedure

In order to make the sensor work configuration of a few system parameters is needed. Use Robot Studio and follow these steps. For more information about the parameters see *The PMC Sensor Setup type on page 158*.

	Action
1.	In the *Motion* configuration topic select the type *PMC Sensor Setup*.
2.	Set *Max Voltage*, defining the input range of the Acromag card, to a value corresponding to the DIP switch settings on the card, normally 5 or 10 volt.
3.	Set up *Disable force sensor cable check*. There is a safety supervision of the sensor connected to PIN 19&20, which enforces an emergency stop when the signal goes below the value set in *Safety Channel Voltage*. It is strongly recommended to use a sensor that has a safety channel. If not, *Disable force sensor cable check* must be set to true.
4.	Set the calibration values (*fx 1 - tz 6*) and scale factors (*fx scale - tz scale*) in order to achieve calibrated forces and torques. These are calculated from the input voltages as follows: $Fx = \frac{fx1}{fxScale}U_0 + \frac{fx2}{fxScale}U_1 + \frac{fx3}{fxScale}U_2 + \frac{fx4}{fxScale}U_3 + \frac{fx5}{fxScale}U_4 + \frac{fx6}{fxScale}U_5$ $Fy = \frac{fy1}{fyScale}U_0 + \frac{fy2}{fyScale}U_1 + \frac{fy3}{fyScale}U_2 + \frac{fy4}{fyScale}U_3 + \frac{fy5}{fyScale}U_4 + \frac{fy6}{fyScale}U_5$ $Fz = \frac{fz1}{fzScale}U_0 + \frac{fz2}{fzScale}U_1 + \frac{fz3}{fzScale}U_2 + \frac{fz4}{fzScale}U_3 + \frac{fz5}{fzScale}U_4 + \frac{fz6}{fzScale}U_5$ $Ty = \frac{ty1}{tyScale}U_0 + \frac{ty2}{tyScale}U_1 + \frac{ty3}{tyScale}U_2 + \frac{ty4}{tyScale}U_3 + \frac{ty5}{tyScale}U_4 + \frac{ty6}{tyScale}U_5$ $Tx = \frac{tx1}{txScale}U_0 + \frac{tx2}{txScale}U_1 + \frac{tx3}{txScale}U_2 + \frac{tx4}{txScale}U_3 + \frac{tx5}{txScale}U_4 + \frac{tx6}{txScale}U_5$ $Tz = \frac{tz1}{tzScale}U_0 + \frac{tz2}{tzScale}U_1 + \frac{tz3}{tzScale}U_2 + \frac{tz4}{tzScale}U_3 + \frac{tz5}{tzScale}U_4 + \frac{tz6}{tzScale}U_5$ xx0600003074
5.	Set *fx max* , *fy max, fz max, tx max, ty max* and *tz max* according to the sensor specifications on maximum load. These parameters specify the max range of the sensor.

10.3.4. Sensors with less than six degrees of freedom

About this section

This section gives information on how to configure a sensor with less than six degrees of freedom.

Pure force sensor

If the system should use a pure force sensor it is important to define that the sensor only measures force. This is done under type *FC Sensor* in the topic *Motion*, by setting the parameter *Force Sensor Type* to Only Force.

NOTE!

All channels on the Acromag DAQ board that is not used must be grounded, connected to shield.

One DOF sensor

If a sensor with a single direction should be used, connect the sensor positive and negative input to either Pin 1&2 (U_0), Pin 4&5 (U_1), Pin 7&8 (U_2), Pin 10&11 (U_3), Pin 13&14 (U_4) or Pin 16&17 (U_5). Assuming the choice was Pin1&2, all values in the calibration matrix should be set to zero except fx 1. Set *fx 1* and *fx scale* so that *(fx 1 / fx scale) = (fx max/ max voltage)*.

10.3.5. Room fixed sensor

Overview

The normal case is to have the sensor mounted on a robot, but there is an option to have the sensor mounted at a stationary position in the work cell. The sensor must then be mounted on a tool or fixture in such a way that the contact force between the robot and the tool is registered.

Configuration

If the sensor is room fixed this must be defined under type *FC Sensor* in the topic *Motion*, by leaving the parameter *Force Sensor Mount Unit Name* empty.

The origin and the direction of the force sensor coordinate system are specified in relation to the world coordinate system.

RAPID

The RAPID function `FCLoadID` should not be used with a room fixed sensor.

`Loaddata` is not relevant to the RAPID instruction `FCCalib`, which is only used in order to find the offset for the sensor.

NOTE!

The calibration convention is identifying the sensed force given from the sensor, which should be equal to the force with which the surrounding effects the robot. This means that there has to be different signs (+/-) in the calibration configuration if the sensor is room fixed or mounted on a robot.

Numerics

C

D

F

I

K

M

P

R

S

T

3HAC027595-001, Revision E, en

ABB AB
Robotics Products
S-721 68 VÄSTERÅS
SWEDEN
Telephone: +46 (0) 21 344000
Telefax: +46 (0) 21 132592