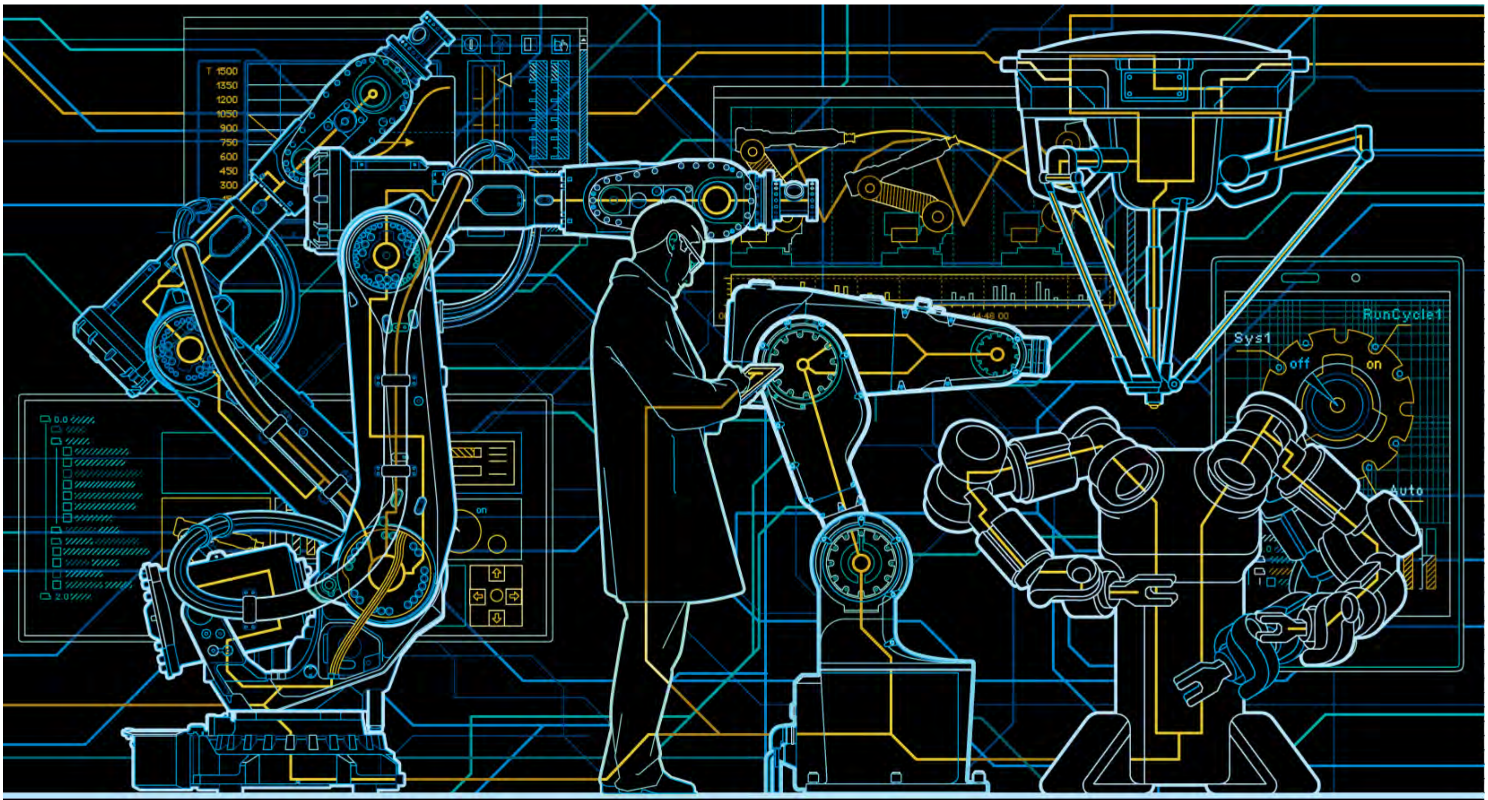


Application manual
Laser Tracker Calibration Interface

Power and productivity
for a better world™

Trace back information:
Workspace R16-2 version a5
Checked in 2016-09-08
Skribenta version 4.6.318

Application manual

Laser Tracker Calibration Interface

RobotWare 6.04

Document ID: 3HAC050957-001

Revision: A

© Copyright 2008-2016 ABB. All rights reserved.

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

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

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

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

Additional copies of this manual may be obtained from ABB.

The original language for this publication is English. Any other languages that are supplied have been translated from English.

© Copyright 2008-2016 ABB. All rights reserved.

ABB AB
Robotics Products
Se-721 68 Västerås
Sweden

Table of contents

© Copyright 2008-2016 ABB. All rights reserved.

This page is intentionally left blank

Overview of this manual

About this manual

This manual contains information about the *Laser Tracker Calibration* (LTC) interface built upon the RobotWare options *Sensor Interface* and *Optical Tracking* used with *ArcWare* or *Continuous Application Platform* (CAP) only. Unless something else is specified, *Laser Tracker Calibration* refers to the interface.

Usage

This manual is meant to be used as a description of how to use the *Laser Tracker Calibration* interface. It provides information about system parameters and RAPID components related to the use of laser trackers, as well as examples of how to calibrate them and verify the obtained precision.

Who should read this manual?

This manual is mainly intended for arc-welding robot operators having to deal with laser trackers.

Prerequisites

The reader should...

- be familiar with the IRC5 robot controller's FlexPendant
- be familiar with the RAPID programming language
- be familiar with system parameters and how to configure them.

References

References	Document ID
Operating manual - IRC5 with FlexPendant	*3HAC050941-001*
Operating manual - RobotStudio	*3HAC032104-001*
Operating manual - Seam tracking with Weldguide III and MultiPass	*3HEA802921-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Technical reference manual - RAPID overview	*3HAC050947-001*
Technical reference manual - System parameters	*3HAC050948-001*
Application manual - Controller software IRC5	*3HAC050798-001*
Application manual - MultiMove	*3HAC050961-001*
Application manual - Arc and Arc Sensor	*3HAC050988-001*
Application manual - Continuous Application Platform	*3HAC050990-001*

Revisions

Revision	Description
-	Released with RobotWare 6.0.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Revision	Description
A	Released with RobotWare 6.04. • Added error handling for the RAPID instructions, see *Instructions on page 42*.

© Copyright 2008-2016 ABB. All rights reserved.

1 Introduction to Laser Tracker Calibration

1.1 Principles of Laser Tracker Calibration

Principles

- Maximum 3 calibrations per sensor: A sensor may have more than one calibration (used in need of good accuracy in different places in space). The maximum allowed number is 3 calibrations per sensor.
- Calibration plate: A plate with a well-known geometry referenced by process parameters defined in PROC.cfg database and by a dedicated work object.
- No simultaneous calibrations: It is not possible to perform simultaneous sensor calibrations from the FlexPendant. Only one sensor per controller can be handled from the FlexPendant at a time. This includes calibration setup and verification.
- MultiMove: The program is designed to handle MultiMove systems. No restriction in the total amount of sensors that can be defined in a robot controller.
- Execution from RAPID: Calibrations and verifications can both be executed from the FlexPendant or using RAPID instructions. See *RAPID interface on page 31* for more information. The LTC__Manager.sys module allows several calibrations and/or verifications to be run simultaneously from RAPID in a MultiMove system comprising several sensors. But only one sensor can be handled at a time per motion task.

© Copyright 2008-2016 ABB. All rights reserved.

1.2 Prerequisites

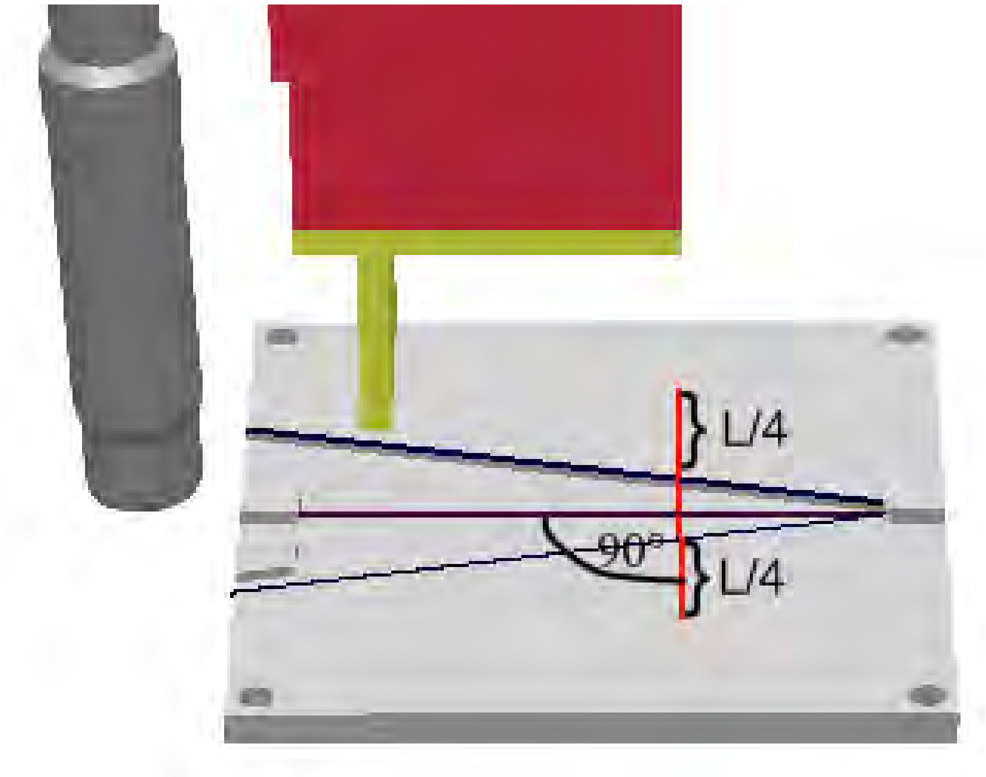
Sensor prerequisites

- The sensor is physically connected to the controller
- The sensor controller communication protocol is ltapp
- The sensor types are correctly defined in the process configuration database PROC.cfg

Note

Within the limitations indicated by the sensor manufacturer, the xxx angle between the sensor head and the torch is free. So is also the angle of the torch in the (xz) plane of the calibration plate. The y-axis of the welding gun should however be aligned with that of the plate. Other angles should be such that the sensor head is aligned with the welding gun both in the (xy) and (yz) planes of the sensor or the plate.

Illustration of angles



Angles

Plate prerequisites

- At least one calibration plate of type ABB or SCOUT is mounted within the robots work area. The plate can be laid with any orientation including vertically or under a ceiling, but the robot work area around the plate shall be free of

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

risk of collision. Possible errors may occur if the orientation of the plate does not allow reorientation of the robot tool during calibration and verification.

- The plate geometry is defined in the system parameter type *Calib_plate_type* in the topic *Process* (PROC.cfg).
- For each calibration plate geometry, one left, one right lap, and one butt joint must be defined in the sensor controller.
- The tracking point for these joints have to be located on the upper edge of the defined joints. The joint values should differ from those used during the welding process.

Note

The orientation of the tracker TCP's coordinate system always verifies:

- y>0 refers to the left-lap joint
- x>0 refers to the welding direction
- z is the result of the cross-product of the x and y vectors.

© Copyright 2008-2016 ABB. All rights reserved.

1.3 Using RW Arc

Required options

The following options must be included in the RobotWare key for the robot system:

Description	Option number
Arc	633-1
Optical Tracking Arc	660-1

LTC__Tmp_CalPose and *LTC__Tmp_CalPos* must be defined as *Calib Variable Name* (*Pose* and *Pos*) in *Optical Sensor Properties* in the topic *Process* (system parameters), see *Application manual - Arc and Arc Sensor*.

Note

To add *Laser tracker Calibration* as a test view on the FlexPendant, open the **Control Panel**, **Additional text view**, and select the checkbox.

© Copyright 2008-2016 ABB. All rights reserved.

1.4 Using Continuous Application Platform (CAP)

Required options

The following options must be included in the RobotWare key for the robot system:

Description	Option number
Continuous Application Platform	624-1
Optical Tracking	813-1

LTC__Tmp_CalPose and *LTC__Tmp_CalPos* must be defined when running *CapLATRSetup* as *\CalibFrame* and *\CalibPos*, see *Application manual - Continuous Application Platform*.

Note

To add *Laser tracker Calibration* as a test view on the FlexPendant, open the **Control Panel**, **Additional text view**, and select the checkbox.

© Copyright 2008-2016 ABB. All rights reserved.

This page is intentionally left blank

2 FlexPendant application

2.1 Overview

General

The Laser Tracker Calibration application on the FlexPendant is used to manage laser tracking sensors.

- *Calibration setup on page 16*.
- *Calibrating on page 21*.
- *Verifying on page 25*.
- *Information views on page 27*.

© Copyright 2008-2016 ABB. All rights reserved.

2.2 Calibration setup

2.2.1 Introduction to calibration setup

Overview

By creating your sensor system in LTC the system parameter configuration (topic *Process*) is updated. A RAPID data file for storing the calibration results is also created and loaded to the controller memory in the motion task specified for the used sensor.

Setting up a system

Follow these steps to set up a system:

1. Define a sensor, see *Defining a sensor on page 17*.
2. Select a CalibID for that sensor, see *Defining a CalibID on page 18*. Select a plate geometry and a work object for the plate.
3. Go through the calibration setup wizard, see *Calibration setup wizard on page 19*.

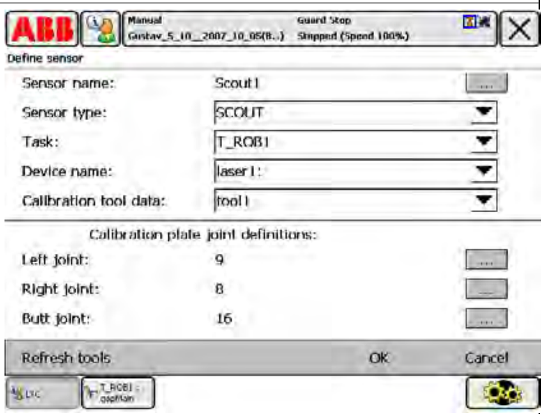
In a more advanced system you might have more than one sensor defined and also more than one calibration for each sensor.

© Copyright 2008-2016 ABB. All rights reserved.

2.2.2 Defining a sensor

Defining a sensor

Use this procedure to define a sensor.

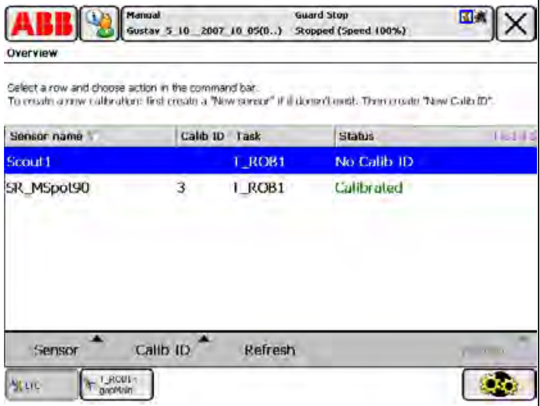
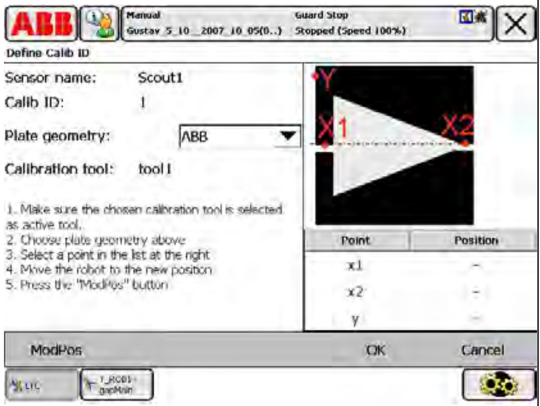
	Action	Note/illustration
1	On the FlexPendant ABB menu, tap **Laser Tracker Calibration**.	
2	On the menu **Sensor**, tap **New**.	
3	Enter values in all fields. 1 The sensor name is used by the program. 2 Sensor type defines the type of sensor. If your sensor type is not in the list, see *My sensor type is not in the sensor type list when defining a sensor on page 47*). 3 Select the motion task the sensor is used with. 4 Device name defines the communication device to be used with the sensor. To create a new communication device, see *Communication devices on page 48*. 5 Select the robot tool that the sensor should be used with. The sensor must be calibrated with this tool. 6 Set the joint definition number for each of the joint types to be used with the calibration plate. The number selected here should match the number defined within the sensor controller.	 Figure 6-1
4	Tap **OK** when finished.	

© Copyright 2008-2016 ABB. All rights reserved.

2.2.3 Defining a CalibID

Defining a CalibID

Use this procedure to define a CalibID.

	Action	Note/illustration
1	On the FlexPendant, use the **Laser Tracker Calibration** application.	
2	Select the sensor for which you would like to create a CalibID.	Figure 6-2
3	Tap the **Calib ID** menu and select a number. Up to 3 CalibID can be defined for each sensor.	
4	In the list **Plate geometry**, select the type of plate for this CalibID. An illustration of the selected plate is displayed to the right of the list.	
5	Define the work object of the plate: 1 Make sure that the correct tool is mounted on the robot. It must be the same tool as when defining the sensor, (calibration tool data). 2 In the **Jogging** window, activate the robot tool and Wobj0. *Laser Tracker Calibration* will show an error message if incorrect jogging parameters are selected before doing ModPos on the x1, x2, and y-positions on the calibration plate. 3 Select one of the positions in the list on the right. 4 Jog the robot to that position. 5 Tap **ModPos** to store the position. 6 Repeat for the other 2 positions.	Figure 6-3 **Note** Make sure that you are very accurate in your positioning of the robot. Any misplacement or displacement of the plate will affect the calibration result.
6	Tap **OK** when finished.	The program will calculate a work object for the plate. Any errors are displayed with a message box.

© Copyright 2008-2016 ABB. All rights reserved.

2.2.4 Calibration setup wizard

Introduction to the calibration wizard

Calibration is set up with a wizard with seven steps. The wizard has three purposes:

- Step 1-3, defining robtargets for approach points and a starting position for calibration.
- Step 4-5, verifying that the calibration plate work object is correct.
- Step 6-7, verifying that the joint settings are good.

Defining approach points and a starting position for calibration

Three positions must be defined.

- Two approach positions, that is how the robot will approach the plate.
- One start position for calibration.

Selecting starting position

The following requirements apply for to choosing the starting position, see figure below.

- The sensor should be positioned so that the laser beam partly falls over the calibration area of the plate.

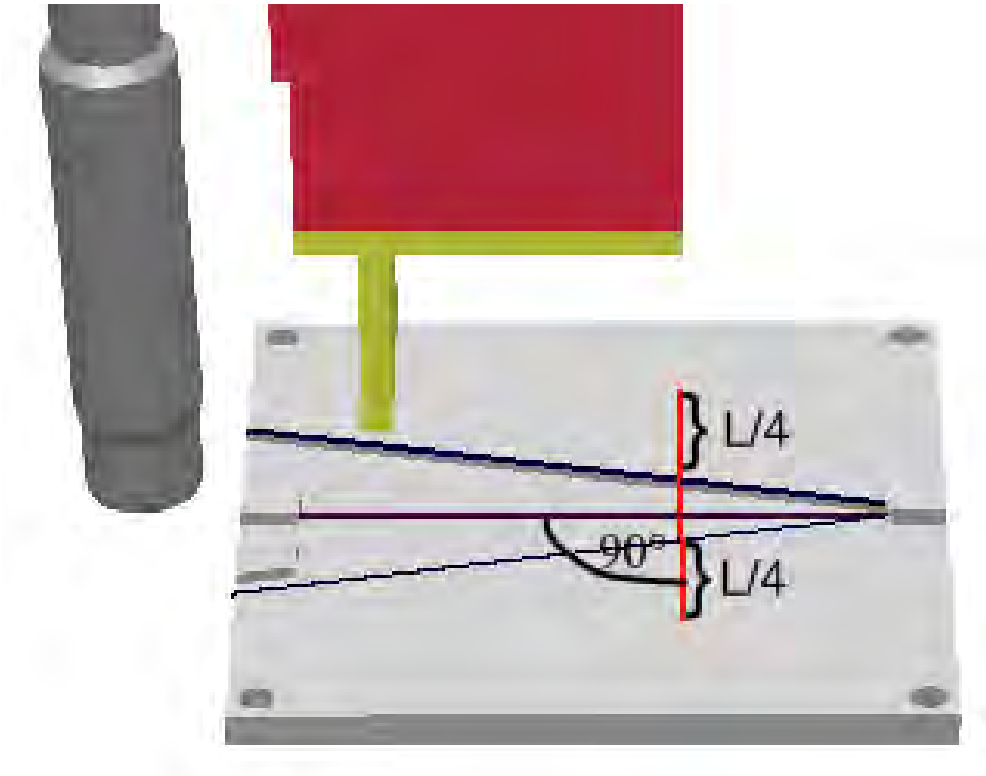
  Note

  About half of the field of view should fall outside the area!

- The y-axis of the sensor should be aligned with the y-axis of the plate.
- The robot tool used for the calibration setup (for example welding gun) should have the same distance from and orientation relative the calibration plate as when running in production.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.



Angles

Verifying the origin of the plate work object

A RAPID module moving the robot to the origin of the work object is loaded to the controller in the specified motion task. Then the program pointer is set to that module.

Verify that the robot is at the real origin of the work object, within a vertical safety margin of a few millimeters (to avoid direct contact of the robot calibration tool with the plate).

Verify the joint settings

The last steps in the wizard will verify that the joint settings in the sensor controller are compatible with the plate and that the sensor works fine.

A RAPID module moving the robot looking for a joint of each type on the plate is loaded to the controller in the motion task specified for the used sensor and the program pointer is set to that module. Start its execution and supervise the action. The laser tracker beam is successively positioned above a right-lap joint, a left-lap joint and, if compatible with the calibration plate used, above a butt-joint. A reading of the respective joints is attempted before moving to the next joint.

The result of the action is summarized in the last step of the wizard.

© Copyright 2008-2016 ABB. All rights reserved.

2.3 Calibrating

2.3.1 Calibrating

Prerequisites

A system must be defined and the calibration setup wizard must be completed.

Calibrating

Use this procedure to calibrate.

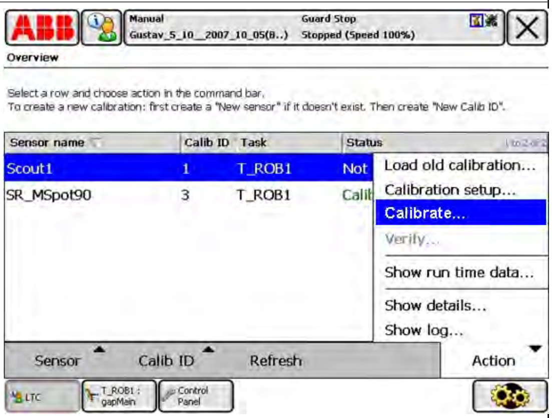
<table>
<tr><th></th><th>Action</th><th>Note/illustration</th></tr>
<tr><td>1</td><td>On the FlexPendant, start the Laser Tracker Calibration application.</td><td></td></tr>
<tr><td>2</td><td>Select the CalibID and sensor name that you want to calibrate and tap Calibrate on the Action menu.</td><td>

Figure 6-5</td></tr>
<tr><td>3</td><td>Select the precision and number of measurements.
The desired precision depends on how the sensor will be used. Some applications need very high precision. The selected number will be used to decide if the calibration is accepted.

Note

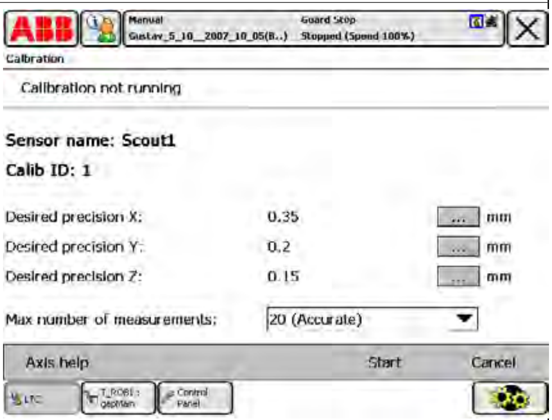
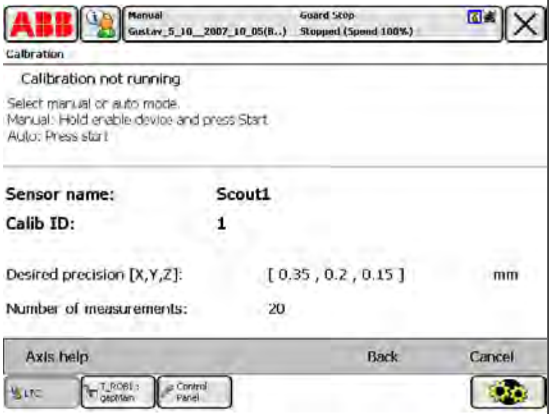
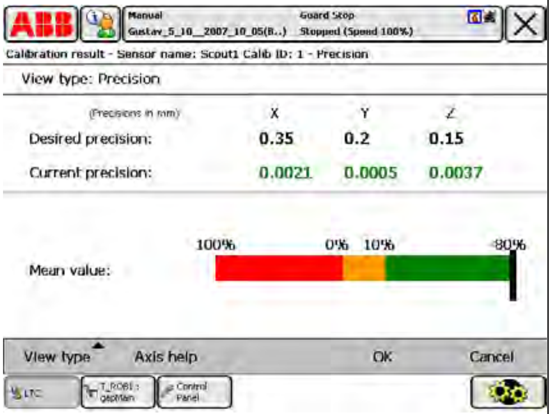
The precision obtained during calibration is rather independent of the number of measurements. But the higher that number, usually the better the obtained precision is achieved during verification. See Calibrating on page 21.</td><td>

Figure 6-6</td></tr>
<tr><td>4</td><td>Tap Start.</td><td></td></tr>
</table>

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

	Action	Note/illustration
5	The necessary RAPID modules are loaded to the controller in the motion task specified for the used sensor and the program pointer is set. Press the Start button on the FlexPendant.	 Figure 6-7
6	Tap **View type** to display the result of the calibration. There are four views. The default view shows the precision, that defines if a calibration is accepted.	 Figure 6-8

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

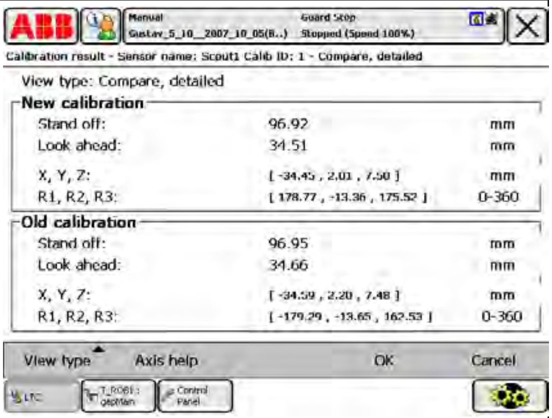
<table>
<tr><th></th><th>Action</th><th>Note/illustration</th></tr>
<tr><td>7</td><td>Use the other views to check the differences or to compare with the new calibration result with the latest one saved.</td><td>

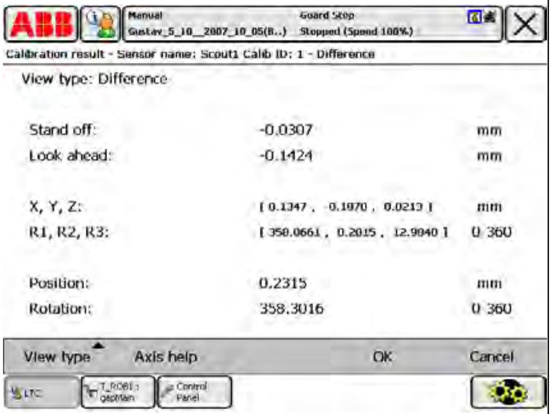
Figure 6-9


Figure 6-10</td></tr>
<tr><td>8</td><td>If the new calibration is approved, tap OK. The new calibration values will then be stored in RAPID variables.

Note

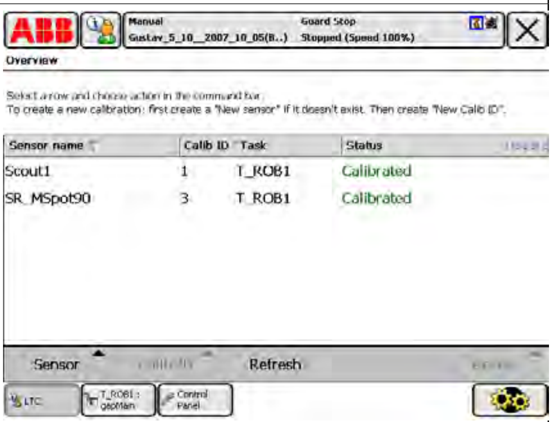
Before tapping OK, the values are only stored in temporary variables. When tapping OK the log will also be updated.</td><td></td></tr>
<tr><td>9</td><td>The calibration is completed. The status of the corresponding CalibID and sensor name should be Calibrated. The sensor is now ready for use in production.</td><td>

Figure 6-11</td></tr>
</table>

© Copyright 2008-2016 ABB. All rights reserved.

2.3.2 Loading old calibration

Introduction

A previous calibration result that has been saved can be loaded to the robot controller for reuse.

Loading old calibration

Use this procedure to load a previous calibration.

	Action	Note/illustration
1	On the **Action** menu, tap **Load old calibration**.	
2	Select a calibration in the list and tap **Show details** to get more information.	Saved previous calibrations are listed chronologically.
3	Select a calibration in the list and tap **Load** to load the calibration results.	

© Copyright 2008-2016 ABB. All rights reserved.

2.4 Verifying

2.4.1 Verifying the calibration

Introduction

The purpose of verification is to check if the calibration data is accepted. A RAPID module executes the verification of the calibration for the used sensor. No parameters need to be defined for the verification. The precision and number of measurements are read from the last performed calibration.

The verification uses three measurement sequences above both lap joints, respectively:

- laser beam aligned with Y axis of calibration plate
- laser beam rotated around its Z axis by an angle of +10 degrees with respect to the Y axis of calibration plate
- laser beam rotated around its Z axis by an angle of -10 degrees with respect to the Y axis of calibration plate

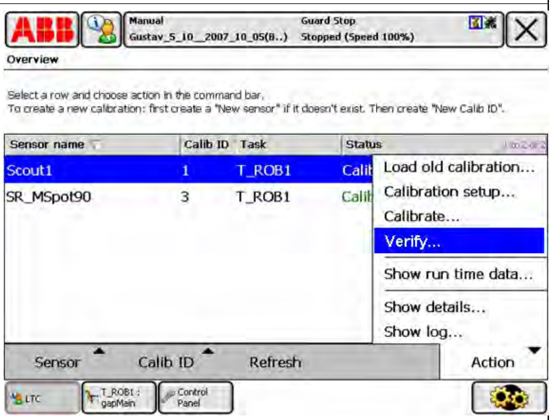
This sequence is repeated by moving the TCP of the sensor tool by 0.5 mm (SCOUT plate) or 1 mm (ABB plate) along the X axis of the plate as many times as the desired number of measurements input before calibration start.

Prerequisites

A calibration must have been done.

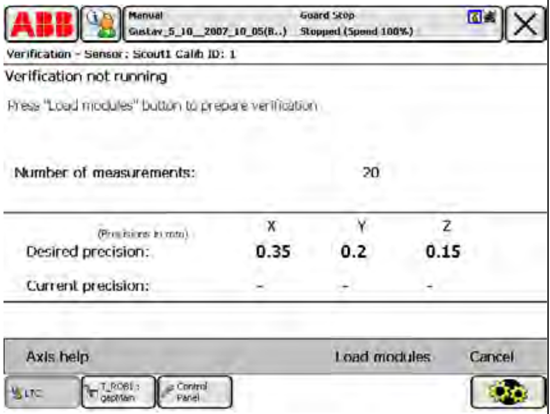
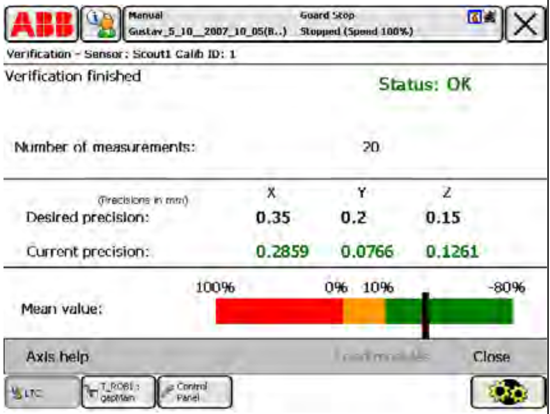
Verifying the calibration

Use this procedure to check the calibration data.

	Action	Note/illustration
1	On the FlexPendant, start the *Laser Tracker Calibration* application.	
2	Select the CalibID and sensor name to be verified.	
3	On the **Action** menu, tap **Verify**.	 Figure 6-12

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

	Action	Note/illustration
4	Press the start button on the FlexPendant to run the verification.	 Figure 6-13
5	The results are displayed when the verification is finished. The log file is updated and the new precision values are written to the configuration (system parameters). If the precision is not acceptable, then do a new calibration. See *Calibrating on page 21*.	 Figure 6-14
6	Tap **Close**.	

© Copyright 2008-2016 ABB. All rights reserved.

2.5 Information views

2.5.1 Introduction to information views

Introduction

The logs are stored in the folder HOME:/LTC in the system. There is one log for each CalibID.

The log can be viewed on the FlexPendant or downloaded to a pc from the controller.

Log structure

The log header shows the name of the sensor, the CalibID used, and creation date. The log file is structured in as a table. Only the last 100 posts are saved in each log.

Runtime interface

The runtime interface gives real time control over the sensor in manual mode. Measurement values in y and z directions of the sensor tool are displayed continuously. The values are updated at the reading frequency of the sensor for a joint number.

It is possible to change the state of the sensor (on or off).

© Copyright 2008-2016 ABB. All rights reserved.

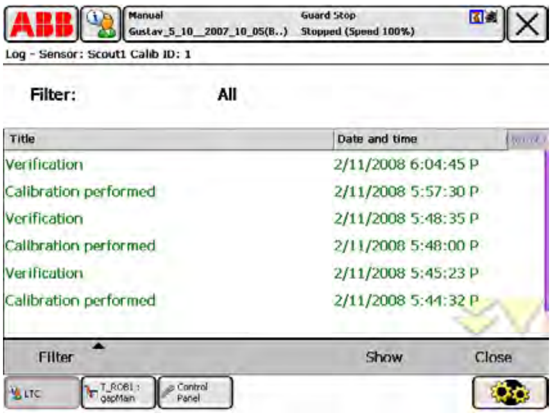
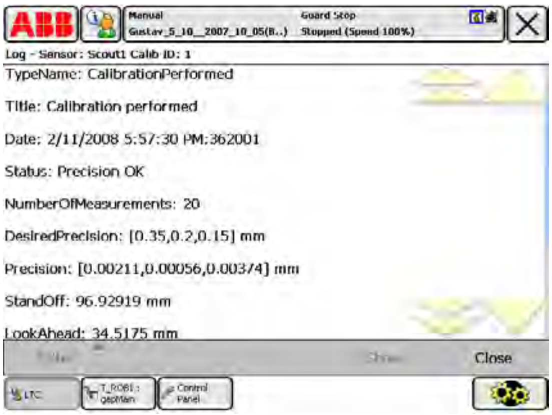
2.5.2 Viewing the log

Introduction to the log

The log view shows all events. Filters can be used to only show calibrations or verifications.

Viewing the log

Use this procedure to view the log.

	Action	Note/illustration
1	On the FlexPendant, start the *Laser Tracker Calibration* application.	
2	Select the CalibID and sensor name.	
3	On the **Action** menu, tap **Show log**. The list of messages is displayed.	 Figure 6-15
4	Select a log and tap **Show** to see details.	 Figure 6-16

© Copyright 2008-2016 ABB. All rights reserved.

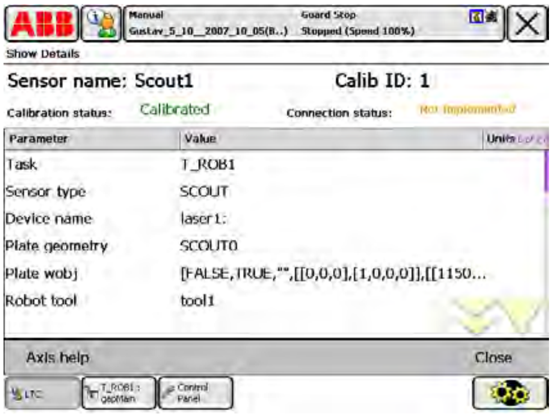
2.5.3 Viewing log details

Introduction

The details view shows all data and parameters for a CalibID.

Viewing log details

Use this procedure to select and view the details for a CalibID.

	Action	Note/illustration
1	On the FlexPendant, start the *Laser Tracker Calibration* application.	
2	Select the CalibID and sensor name.	
3	On the Action menu, tap **Show details.**	 Figure 6-17

© Copyright 2008-2016 ABB. All rights reserved.

This page is intentionally left blank

3 RAPID interface

Introduction

Laser Tracker Calibration has a RAPID interface, that makes it possible to calibrate and verify the calibration on a regular basis during production. The module LTC__TemplateData.sys is used for generating specific data files when creating a new sensor.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

3.1 Using the RAPID modules without the FlexPendant interface

Introduction

This section describes how to use the RAPID modules without the *Laser Tracker Calibration* application on the FlexPendant. This is useful when calibrating the sensor or verifying a calibration using RAPID.

The calibration setup is not described here. It should be done in advance from the FlexPendant application.

Since calibration and verification are similar, they are described together.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

3.1.1 Preparations

Prerequisites

Before running calibration or calibration verification using RAPID, the following must be done on the FlexPendant application:

1. Define the sensor. See *Defining a sensor on page 17*.
2. Define a CalibID for that sensor. See *Defining a CalibID on page 18*.
3. Set up a calibration. See *Calibrating on page 21*.
4. Run a calibration. See *Calibrating on page 21*.

System parameter, topic Process (PROC)

All parameters in SENSOR_TYPE, SENSOR_PHYSICAL, SENSOR_CALIB_ID, and CALIB_PLATE_TYPE must be defined correctly. This is done when the sensor is created, set up, and calibrated using the *Laser Tracker Calibration* application on the FlexPendant.

The system parameters can also be defined using FlexPendant **Control Panel** or RobotStudio **Configuration Editor**. The parameter settings can be exported as a file, PROC.cfg.

For more information about the parameters, see *Topic Process on page 38*.

Sensor data file

A RAPID module named LTC__MySensorData, where *MySensor* is the sensor name, was created and loaded during the first setup of the sensor. That module must exist in the controller's memory before a calibration or a verification can be run without using the FlexPendant application. See *Variables on page 36* for information on the content of the sensor data file.

All the variable names include:

- *MySensor* is the sensor name.
- *X* is the CalibID number.

The following variables must be stored in LTC__MySensorData.sys for calibration or verification purposes:

Data type	Name	Required value
robtarget	LTC__MySensor_X_fast2plate	Yes
robtarget	LTC__MySensor_X_slow2plate	Yes
robtarget	LTC__MySensor_X_cal_Start	Yes
robtarget	LTC__MySensor_X_x1	No
robtarget	LTC__MySensor_X_x2	No
robtarget	LTC__MySensor_X_y	No
wobjdata	LTC__MySensor_X_Wobj	Yes
tooldata	LTC__MySensor_X_SenTool	No for calibration Yes for verification
pose	LTC__MySensor_X_CalPose	No

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Data type	Name	Required value
pos	LTC__MySensor_X_CalPos	No for calibration Yes for verification
pos	LTC__MySensor_X_Precision	No

© Copyright 2008-2016 ABB. All rights reserved.

3.1.2 Performing a calibration or verification

Introduction

Tracker calibration and verification can be initiated from customized RAPID modules by using the LTC__Manager system module.

Performing a calibration or verification

Use this procedure to calibrate and verify the sensor with the RAPID modules *LTC__Manager*.

	Action	Note/illustration
1	Make sure that the preparations are done, see *Preparations on page 33*.	
2	Load the LTC__Manager.sys module to the motion task with the sensor to be calibrated or verified. During its execution, the module will automatically upload the system modules needed for calibration.	
3	Insert the instruction `CalibInstance` where the robot can reach the approach position *Fast2Plate*, to initiate a tracker calibration from a customized RAPID module. `CalibInstance SensorName, \CalibID, \DesiredPrecision;`	The execution of the instruction verifies that the obtained calibration precision is within the desired range. If not, RAPID execution will stop and an error message is displayed. A successful calibration is always followed by an automatic verification of the calibration. That is, as a result of at least one additional measurement it will be verified that the obtained average precision is within the desired range. See *LTCcalibrate on page 42*.
4	To verify a tracker calibration (that is, no new calibration) from a customized RAPID module, insert the instruction `VerifInstance` where the robot can go to the approach position *Fast2Plate*. `VerifInstance SensorName, \CalibID, \DesiredPrecision;`	As mentioned above, at least one additional measurement it will be executed and it will be checked if the obtained precision is within the desired one. If not, RAPID execution will stop and an error message will be displayed. See *LTCverify on page 44*.
5	The calibration/verification is now ready to be started.	

© Copyright 2008-2016 ABB. All rights reserved.

3.2 Variables

3.2.1 Variables in LTC_MySensordata (LTC_TemplateData)

Introduction

The table below describes all the variables defined in LTC__TemplateData.

MySensor refers to the name of the sensor to which a dedicated data file has been created.

1 means that CalibID = 1. In comparison, CalibID 2 suggests that the sensor name is followed by a 2 in each variable name instead.

Data type	Name	Used for
robtarget	LTC__MySensor_1_fast2plate	A remote position from where the robot can reach the calibration plate without collision. Defined in the *Wobj LTC__ MySensor_1_Wobj*
robtarget	LTC__ MySensor _1_slow2plate	A closer position from where the robot can reach the calibration plate without collision. Defined in the Wobj *LTC__ MySensor _1_Wobj*
robtarget	LTC__ MySensor _1_cal_Start	The start position of the calibration. • The sensor should be positioned so that the laser beam partly falls over the calibration area of the plate. Note! About half of the field of view should fall outside the area. • The Y-axis of the sensor should be aligned with the Y-axis of the plate. • The welding gun or the chosen calibration tool (that is, the selected robot tool) should have the same distance and orientation to the plate as in normal production. Defined in the Wobj *LTC__ MySensor _1_Wobj*
robtarget	LTC__ MySensor _1_x1	A position on the X axis of the calibration plate, used for calculating the work object *LTC__ MySensor_1_Wobj*. This robtarget is defined in Wobj0.
robtarget	LTC__ MySensor _1_x2	A position on the X axis of the calibration plate, used for calculating the work object *LTC__ MySensor_1_Wobj*. This robtarget is defined in Wobj0.
robtarget	LTC__ MySensor _1_y	A position on the Y axis of the calibration plate, used for calculating the work object *LTC__ MySensor_1_Wobj*. This robtarget is defined in Wobj0.
wobjdata	LTC__ MySensor _1_Wobj	The Wobj of the calibration plate.
tooldata	LTC__ MySensor _1_SenTool	The tooldata of the sensor, used internally in LTC during verification.
pose	LTC__ MySensor _1_CalPose	The translation from the sensor tool TCP to the robot tool TCP.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Data type	Name	Used for
pos	LTC__ MySensor _1_CalPos	The offset from sensor tool to point of normal production.
pos	LTC__ MySensor _1_Precision	The precision acquired during the latest calibration/verification.

© Copyright 2008-2016 ABB. All rights reserved.

3.3 System parameters

3.3.1 Topic Process

Introduction

The following tables describe the system parameters relevant for *Laser Tracker Interface*. The types belong to the topic *Process*.

SENSOR_TYPE

Variable	Description
Name	The name of the sensor type (string).
Dimensions	The number of dimensions the sensor has, which is 2 or 3 (integer).
AngleCameraToLaser	The angle between the laser beam of the sensor and the camera axis (float).
Frequency	The reading frequency in Hz of the sensor, that is, the number of measurements the sensor captures every second (float).
WidthOfFieldClose	The width of the field-of-view at shortest distance (float).
WidthOfFieldFar	The width of the field-of-view at longest distance (float).
CloseStandOff	The stand-off distance from the z-ref of the sensor to its Close Field-of-View (num).
DepthOfField	The distance between shortest and longest range of view of the sensor (num).
OptTCPStandOff	The optimal TCP stand-off distance from the z-reference of the sensor to the optimal measuring point of the sensor (num).
FarStandOff	The stand-off distance from the z-ref of the sensor to its Far Field-of-View (num).
FrameAlignment	The alignment of the sensor measurement frame (string): • laser: aligned with the laser beam • camera: aligned with the camera housing, not the laser beam (string)
FrameZOrientation	The orientation of the z component of the sensor measurement frame (num): • +1 up, that is, "into" the camera • -1: down, that is, "out of" the camera
Brand	The brand of the sensor (string): • ServoRobot • MetaScout

SENSOR_PHYSICAL

Variable	Description
Name	The name of the sensor for the purpose of calibration (string).
SensorType	The type of the sensor, see *SENSOR_TYPE on page 38*, (string).
Task	The name of the motion task that uses this sensor (string).

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Variable	Description
DeviceName	The name of the device used for communication between the sensor and the robot controller (string).
RobTool	The name of the robot tool used with the sensor (string).
LeftLapJoint	The sensor controller’s ID of the left lap joint for the calibration plate (integer).
RightLapJoint	The sensor controller's ID of the left right joint for the calibration plate (integer).
ButtJoint	The sensor controller's ID of the butt joint for the calibration plate (integer).
UsedForTracking	If the sensor is mounted as usual for tracking, that is, near the TCP of the robot tool that is used in a process application (for example arc welding), this parameter shall be set to *true*. There will be no difference compared to earlier RobotWare releases. If the sensor is not mounted for tracking, and points in an other direction than or is placed far from the robot tool TCP, then the parameter shall be set to *false*. The difference during calibration is, that no visual verification of the position of the calibration plate origo (x=0, y=0, z=0 in the calibration plate work object) will be performed. This step is skipped during the calibration setup to avoid collision with the sensor.

SENSOR_CALIB_ID

Variable	Description
Name	The name of the calibration instance, constituted by the sensor name for the purpose of calibration and the calibID (string).
SensorName	The name of the sensor, used for calibration. See *SENSOR_PHYSICAL on page 38*, (string).
PlateGeometry	The name of the calibration plate used for the calibration instance (string).
LookAhead	The distance between the robot tool TCP and the sensor TCP at the CalPos position (float). Note! The sensor TCP is sometimes located within the unit, but the CapPos position corresponds to the location of the joint in normal production.
MaxSpeed	The maximum speed at which the sensor still can read the seam at a point and give a data feedback to the controller before the point is reached by the robot tool (float).
NumberOfMeasurements	The number of measurements carried out during the last calibration (integer).
DesiredPrecision	The precision required during the last calibration (string, in vector format: `[0.532, 0.435, 0.345]`).
LastCalibrated	The date and time of the last calibration (string).
LastVerified	The date and time of the last verification (string).
HasBeenSetup	True if the sensor has been setup, otherwise false (string, `TRUE` or `FALSE`).

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

CALIB_PLATE_TYPE

Variable	Description
Name	An abbreviation describing the name of the calibration plate available. For ease of use, the abbreviation can be terminated with a 0 or a 1 indicating the type of the calibration plate (string).
CalType	The type of the calibration plate (integer, 0 or 1).
Vertice_x	The x position of the calibration vertex in the coordinate system of the plate (float).
(Vertice_y)	The y position of the calibration vertex in the coordinate system of the plate (float)
CalAngle	The value of the calibration angle on the calibration plate (float)

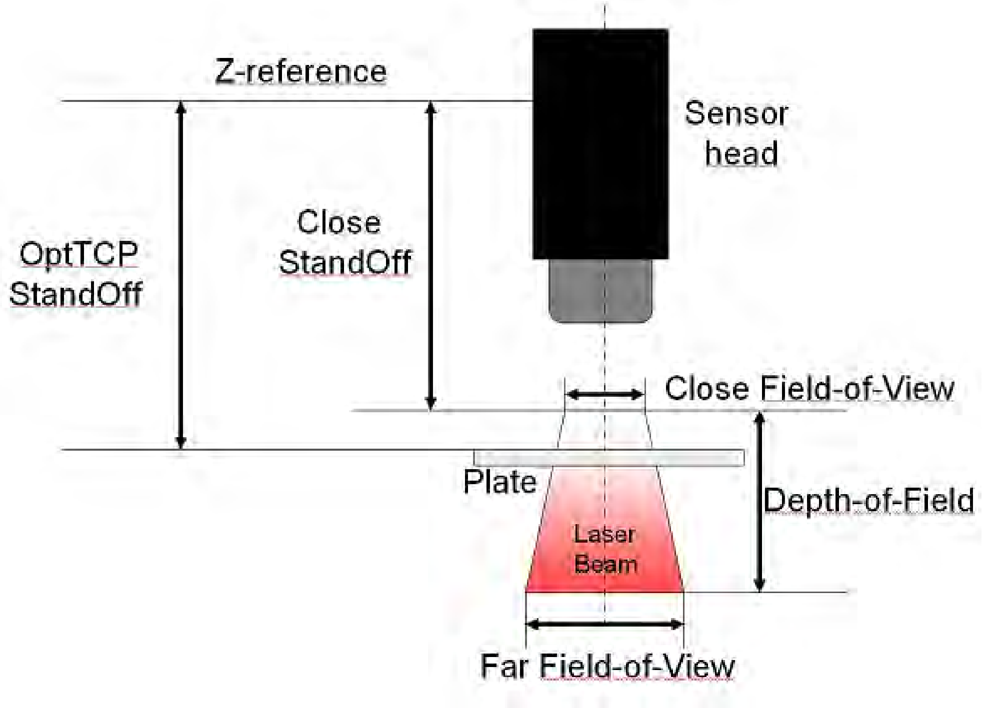


Figure 8-1

Parameters described in SENSOR_TYPE (see *SENSOR_TYPE on page 38*) are commonly used for all sensor optical configurations in *Laser Tracker Calibration* and illustrated above. As sensor specifications vary between manufacturers, some parameter values may have to be deduced from others.

Field-of-view

By default, *Laser Tracker Calibration* will use the value entered as *FieldOfViewClose* in the type *SENSOR_TYPE*. If sensor specifications only refer to a nominal field-of-view value, store that value both as *FieldOfViewClose* and *FieldOfViewFar*, unless values can be measured.

Depth-of-Field

If sensor specifications refer to a useful depth-of-field at nominal stand-off in the form of an interval [+zmax, -zmin] around the nominal stand-off position, calculate the depth-of-field as: `zmax - (-zmin)` and store that value as *DepthOfField*.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Close Stand-off

If sensor specifications refer to a nominal stand-off distance, calculate the close stand-off distance as: `nominal stand-off - Abs(-zmin)` and store that value as *CloseStandOff*.

Example 1: Metavision sensor head MLP2/30

Data provided by the sensor manufacturer:

Field-of -view (horiz.) at nominal stand-off	30 mm
Nominal stand-off	75 mm
Useful depth-of-field at nominal stand-off	-30 to +45 mm

These data should appear in the SENSOR_TYPE field of PROC as:

WidthOfFieldClose	30
WidthOfFieldFar	30
CloseStandOff	45
DepthOfField	75
OptTCPStandOff	75

Example 2: ServoRobot sensor head SF-80/D

Data provided by the sensor manufacturer:

Depth-of-Field	85 mm
Width (close plane)	24 mm
Width (far plane)	36 mm
Close stand-off	97 mm

These data should appear in the SENSOR_TYPE field of PROC as:

WidthOfFieldClose	30
WidthOfFieldFar	30
CloseStandOff	45
DepthOfField	75
OptTCPStandOff	75

© Copyright 2008-2016 ABB. All rights reserved.

3.4 Instructions

3.4.1 LTCcalibrate

Usage

`LTCcalibrate` is used to enable creation of the instance of a tracker calibration from a customized RAPID module.

Basic example

```
LTCcalibrate "MSPOT90_PL ", \CalibID:=2\precision:=,[0.5,0.4, 0.8];
```

The tracker of name MSPOT90_PL with calibration identity 2 is selected to run a calibration with precision [0.5, 0.4, and 0.8] respectively in the x, y, and z directions of the tracker's coordinate system.

Arguments

```
LTCcalibrate SenName, \CalibID, \precision;
```

`SenName`

Data type: `string`

The name of the sensor to be calibrated as declared in the system parameters (PROC.cfg).

`\CalibID`

Data type `num`, default value: 1

The calibration instance number as defined in PROC.cfg.

`\precision`

Data type `pos`, default values from LTC.

The desired precision respectively in the x, y, and z directions of the sensor tool expressed in millimeters as a vector. For example, [0.3, 0.2, 0.3]. When using [0,0,0] as desired precision, the values stored in PROC.cfg during the original calibration are reused automatically.

Program execution

`LTCcalibrate` is the name of a routine in the LTC__Manager.sys module. When executed, the routine first unloads data and calibration/verification system modules that may have existed in the controller's memory for the motion task used by the sensor, so that no error will happen when loading the system modules needed for a calibration in the motion task.

A data initialization is done where the `LTCcalibrate` arguments are used to read parameters from PROC.cfg and store their values in temporary variables defined in LTC__TmpData.sys. After a calibration, the obtained precision is compared to the desired one. If the obtained precision is good enough, the LTC__Calibration.sys module is unloaded and replaced with the LTC__Verification.sys module to run a verification of the newly performed calibration. Again, the desired precision is compared to the obtained precision.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

If the obtained precision differs from the desired, a recoverable error message is displayed on FlexPendant and RAPID execution is redirected to the error handler. Otherwise the uploaded system modules are unloaded from the motion task used by the sensor and RAPID execution resumes in the customized RAPID module.

Error handling

The following recoverable errors are generated and can be handled in an error handler. The system variable `ERRNO` will be set to:

Name	Cause of error
`LTC_CALIB_ERR`	This error is raised if the calibration of the sensor does not meet the imposed precision requirement.

Limitations

In a MultiMove system, the LTC__Manager.sys module has to be loaded in the respective motion tasks that own a laser tracker in order for `CalibInstance` commands to be executed from the respective motion tasks at the same time.

© Copyright 2008-2016 ABB. All rights reserved.

3.4.2 LTCverify

Usage

`LTCverify` is used to enable the verification of a tracker calibration from a customized RAPID module.

Basic examples

```
LTCverify "MSPOT90_PL ", \CalibID:=2\precision:=,[0.5,0.4, 0.6];
```

The tracker of name MSPOT90_PL with calibration identity 2 is selected to run calibration verification with the desired precision [0.5, 0.4 and 0.6] respectively in the x, y, and z directions of the tracker's coordinate system.

Arguments

```
LTCverify SenName, \CalibID, \precision;
```

`SenName`

Data type `string`

The name of the sensor to be calibrated as declared in the system parameters (PROC.cfg).

`\CalibID`

Data type `num`, default: 1

The calibration instance number as defined in PROC.cfg.

`\precision`

Data type `pos`, default: values from LTC

The desired precision respectively in the x, y, and z directions of the sensor tool expressed in millimeters as a vector. For example, [0.3, 0.2, 0.3]. When using [0,0,0] as desired precision, the values stored in PROC.cfg during the original calibration are reused automatically.

Program execution

`LTCverify` is the name of a routine in the LTC__Manager.sys module. When executed, the routine first unloads data and calibration/verification system modules that may have existed in the controller's memory for the motion task used by the sensor, so that no error will happen when loading the system modules needed for calibration verification in the motion task.

A data initialization is done where the `LTCverify` arguments are used to read parameters from PROC.cfg and store their values in temporary variables defined in LTC__TmpData.sys. After verification is done, the obtained precision is compared to the desired one.

If the obtained precision is good enough, the uploaded system modules are unloaded from the motion task used by the sensor and RAPID execution resumes in the customized RAPID module. If the obtained precision departs from the desired one, an error message is displayed on FlexPendant and RAPID execution is redirected to the error handler.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

Error handling

The following recoverable errors are generated and can be handled in an error handler. The system variable `ERRNO` will be set to:

Name	Cause of error
`LTC_VERIFY_ERR`	This error is raised if the verification of the sensor does not meet the imposed precision requirement.

Limitations

In a MultiMove system, the LTC__Manager.sys module has to be loaded in the respective motion tasks that own a laser tracker in order for `CalibInstance` commands to be executed from the respective motion tasks possibly at the same time.

© Copyright 2008-2016 ABB. All rights reserved.

This page is intentionally left blank

4 Trouble shooting

4.1 Sensor types

4.1.1 My sensor type is not in the sensor type list when defining a sensor

Available sensor types

The list of available sensor types is created by reading the system parameter configuration for the type SENSOR_TYPE (topic *Process*). That is, the defined instances will be available in the sensor type list.

As a rule, most laser trackers for robotic arc welding applications available commercially from ServoRobot and Metavision-Scout work fine with the *Laser Tracker Calibration* interface. Off-lens sensors with circular beams produced by *Oxford Sensor Technology* are not supported by the interface.

Defining a sensor

To create new instances, modify the system parameters using RobotStudio or the FlexPendant, or by editing PROC.cfg. To use a new sensor type, make sure that the RAPID calibration modules can handle the sensor. Otherwise create new such modules.

© Copyright 2008-2016 ABB. All rights reserved.

4.2 Communication devices

4.2.1 Creating a new communication device

Communication devices

Several communication devices can be necessary when using more than one sensor in a robot system. By default, laser1: is the only communication device predefined in the system input/output configuration file (SIO.cfg).

Creating a new communication device

To create a new communication device, modify the system parameters using RobotStudio or the FlexPendant, or by editing the SIO.cfg file. Add a new device name in the type *Transmission Protocol*, in the topic *Communication*.

Do a warm restart to load the modified system parameter configuration.

Note

Make sure the SIO.cfg file is updated with the correct baud rates for each COM port affected in the laser tracker systems.

© Copyright 2008-2016 ABB. All rights reserved.

4.3 The precision of the calibration/verification is bad

4.3.1 The precision of the calibration/verification

Poor calibration positions

Poor calibration precision can be caused by many reasons.

- The location of the calibration plate might have inadvertently been changed since the last calibration, or the plate might be damaged. Make sure that the plate is still at its original place and does not have any major scratches or other defects.
- The robot tool might be badly calibrated. Before using ModPos for the Wobj of the plate or calibrating the sensor, make sure that the robot tool itself is calibrated with good precision.
- The robot might be badly calibrated. Before using ModPos for the Wobj of the plate or calibrating the sensor, make sure that the robot is correctly calibrated. Update the robot revolution counters before continuing using your laser tracker system.
- The sensor might be loosely mounted. Make sure that the sensor is fastened well and not loose.

Note

As the robot stands still during calibration, in average the obtained precision during a calibration does not get much better by selecting a higher number of measurements. During verification of a calibration, the obtained precision can decrease slightly (that is, discrepancies increase a little) with the number of measurements performed but it is usually stable as worst measurements are rejected by mean deviation calculation and averaging is run over the remaining number of measurement points.

Inaccurate Wobj

One reason for poor calibration can be that the workobject (Wobj) for the plate is inaccurate. Make sure that the Wobj of the calibration plate is very accurately defined. Define the Wobj from the *Laser Tracker Calibration* graphical user interface in the view **Define CalibID**, or store the data to the correct RAPID variable in LTC__MySensorData.sys. The Wobj of a calibration is named *LTC__MySensor_X_Wobj*, where *MySensor* is the name of the sensor for the calibration instance, and X is the ID of the instantiation, referred to as *CalibID* (for example *LTC__Mspot_90_left_1_X_Wobj*).

Verifying the Wobj

Robot positions that define the plate Wobj can be verified quickly by using the **GOTO** function in the **Jogging** window.

The function moves the TCP straight from its current position to the selected robtarget. Such movements can damage both a tool and the calibration plate if not with a safety distance between the TCP and the plate's surface.

Continues on next page

© Copyright 2008-2016 ABB. All rights reserved.

To use the **GOTO** function without any risk, first use a small RAPID program containing the following `MoveJ` instructions to verify that the approach positions are programmed correctly:

```
MoveJ approch_x1, v30, z10, tool1;
MoveLx1,v30,fine, tool1;
MoveL approch _x1, v30, z10, tool1;
MoveJ approch_x2, v30,z10,tool1;
MoveL x1, v30, fine, tool1;
MoveL approch_x2,v30,z10, tool1;
MoveJ approch _y, v30, z10, tool1;
MoveLx1, v30, fine, tool1;
MoveL approch _y, v30, z10, tool1;
```

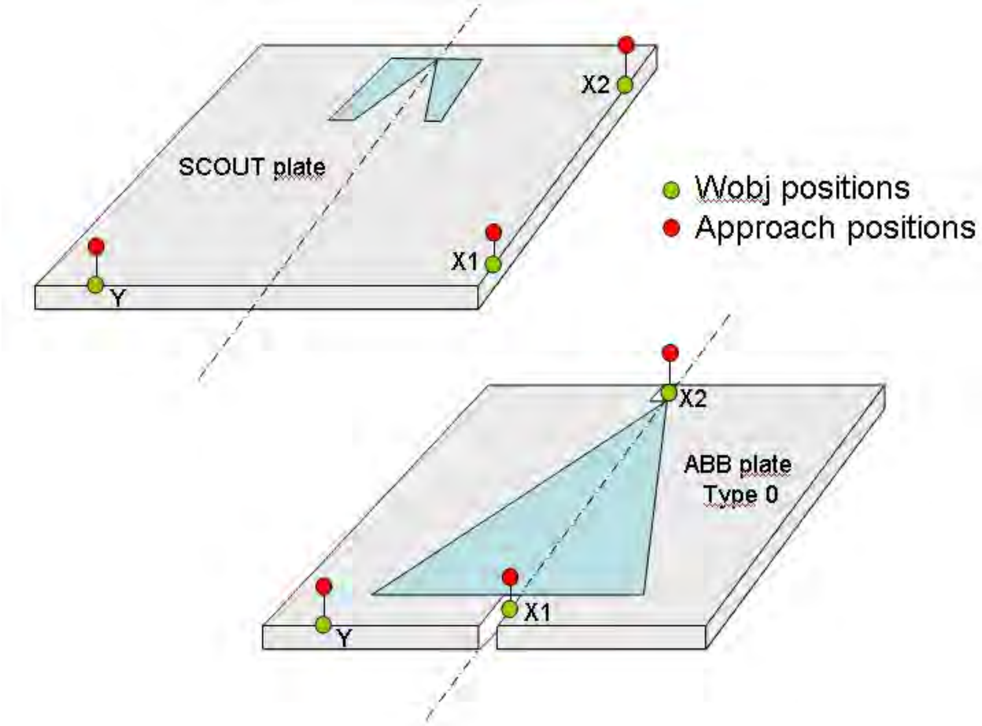


Figure 11-1

Using a 3-dimensional tracker

When using a 3-dimensional tracker with a short field of view, select a Cal_start position as close as possible to the calibration vertex of the plate so that all stripes of the tracker beam always cross both lap-joints of the plate at any time.

Note

In practice, the fifth stripe of a 3-dimensional tracker beam can sometimes lose contact with a lap-joint on the calibration plate, but measurement is still possible with four stripes only, without much effect on the obtained precision.

© Copyright 2008-2016 ABB. All rights reserved.

4.3.2 Joint position error

Joint position errors

The robot can have difficulties to reach programmed positions around the calibration plate due to their location relative to each other.

The *Laser Tracker Calibration* interface makes the robot movement as automatic as possible to minimize manual efforts from an operator.

Solving joint position errors

To solve joint position errors, consider the current orientation of the robot and see if another robot position could be used (ModPos) to calibrate the laser tracker.

Or, try changing the position of the calibration plate relative to the robot.

© Copyright 2008-2016 ABB. All rights reserved.

4.3.3 General error from sensor

Solving sensor errors

Due to communication errors between sensor and controller, the background task *tLtcBgTsk* can stop. If so:

- Check that the baud rates are identical. Use the **Control Panel** on the FlexPendant or the **Configuration Editor** in RobotStudio. The type *Physical Channel* is in the topic *Communication*.
- Verify that the correct COM ports on the sensor's interface and on the controller are connected with each other.
- Check the joints definition in the sensor's PC interface.
- Restart the sensor's computer when all settings on the controller are correct.

© Copyright 2008-2016 ABB. All rights reserved.

4.3.4 Backup

Creating a backup

Making a backup of your robot system from FlexPendant is a cautious step to save your calibration result together with other process and sensor data. In case of problems affecting the general settings of the robot system, the saved calibration can be retrieved anytime by loading the backup to the robot controller’s memory.

© Copyright 2008-2016 ABB. All rights reserved.

This page is intentionally left blank

Index

© Copyright 2008-2016 ABB. All rights reserved.

Contact us

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

ABB AS, Robotics
Discrete Automation and Motion
Nordlysvegen 7, N-4340 BRYNE, Norway
Box 265, N-4349 BRYNE, Norway
Telephone: +47 51489000

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

www.abb.com/robotics

3HAC050957-001, Rev A, en

Power and productivity
for a better world™