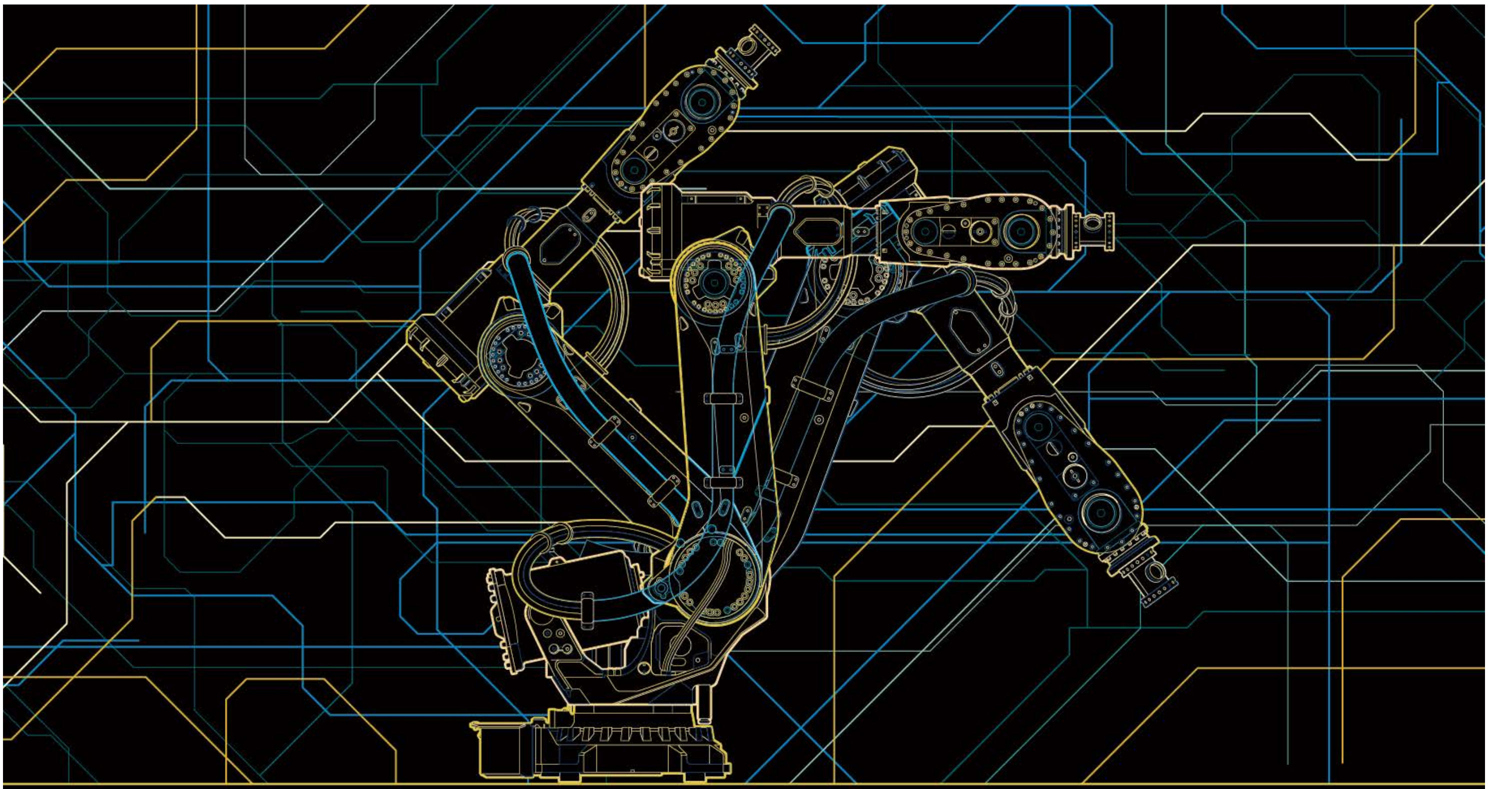


Application manual
Navigator

Power and productivity
for a better world™

Trace back information:
Workspace RW 6-0 version a3
Checked in 2014-11-11
Skribenta version 4.1.349

Application manual
Navigator

Document ID: 3HAC050962-001
Revision: -

ABB AB

Robotics Products

Se-721 68 Västerås

Sweden

Table of contents

Overview of this manual

About this manual

This manual explains the basics of when and how to use the option Navigator:

- Product overview
- Operation overview
- Requirements overview
- Software set-up
- Software reference, RAPID

Usage

This manual can be used either as a reference to find out if an option is the right choice for solving a problem, or as a description of how to use an option. Detailed information regarding syntax for RAPID routines, and similar, is not described here, but can be found in the respective reference manual.

Who should read this manual?

This manual is intended for:

- installation personnel
- 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.

Reference documents

References	Document ID
Technical reference manual - RAPID overview	*3HAC050947-001*
Technical reference manual - RAPID Instructions, Functions and Data types	*3HAC050917-001*
Operating manual - Getting started, IRC5 and RobotStudio	*3HAC027097-001*
Operating manual - IRC5 with FlexPendant	*3HAC050941-001*
Technical reference manual - System parameters	*3HAC050948-001*
Operating manual - RobotStudio	*3HAC032104-001*
Application manual - Production Manager	*3HAC052855-001*

Revisions

Revision	Description
-	Released with RobotWare 6.0.

Product documentation, IRC5

Categories for user documentation from ABB Robotics

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

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

Product manuals

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

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

Technical reference manuals

The technical reference manuals describe reference information for robotics products.

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

Continues on next page

Application manuals

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

An application manual generally contains information about:

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

Operating manuals

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

The group of manuals includes (among others):

- *Operating manual - Emergency safety information*
- *Operating manual - General safety information*
- *Operating manual - Getting started, IRC5 and RobotStudio*
- *Operating manual - Introduction to RAPID*
- *Operating manual - IRC5 with FlexPendant*
- *Operating manual - RobotStudio*
- *Operating manual - Trouble shooting IRC5*, for the controller and manipulator.

Safety

Safety of personnel

When working inside the robot controller it is necessary to be aware of voltage-related risks.

A danger of high voltage is associated with the following parts:

- Devices inside the controller, for example I/O devices, can be supplied with power from an external source.
- The mains supply/mains switch.
- The power unit.
- The power supply unit for the computer system (230 VAC).
- The rectifier unit (400-480 VAC and 700 VDC). Capacitors!
- The drive unit (700 VDC).
- The service outlets (115/230 VAC).
- The power supply unit for tools, or special power supply units for the machining process.
- The external voltage connected to the controller remains live even when the robot is disconnected from the mains.
- Additional connections.

Therefore, it is important that all safety regulations are followed when doing mechanical and electrical installation work.

Safety regulations

Before beginning mechanical and/or electrical installations, ensure you are familiar with the safety regulations described in *Operating manual - General safety information*[1].

1 This manual contains all safety instructions from the product manuals for the manipulators and the controllers.

1 Introduction

1.1 Product overview

1.1.1 General

Navigator is a set of functions for defining calibration points automatically and for coordinate measurements.

The available services consist of:

- Automatically localize tooling balls.
- Automatically perform measurement of coordinates on planes.

1.2 Operation overview

1.2.1 General

Navigator consists of a suite of RAPID Move instructions. The instructions are programmed in traditional RAPID programming manner. Each instruction moves the measurement probe to the search start position and starts a search pattern.

1.3 Requirements overview

1.3.1 System prerequisites

This Navigator version is intended for use in arc welding systems incorporating IRB 1400, 1600, 2400, etc. robots.

- BaseWare requirements: 5.07 or later
- Controller requirements: IRC5

The Navigator package includes software that is loaded into all arc welding motion tasks. Process configuration parameters are used to connect real I/O signals and to modify the default settings.

1.3.2 Hardware prerequisites

The robot should be equipped with a tactile measurement system such as SmarTac or any other system that sets an I/O signal upon contact with a surface.

The TCP should be a well-defined machine tool probe (ABB option BullsEye is preferred for defining the accurate TCP).

1.3.3 User's qualifications

Any competent robot programmer (RAPID language) may be self-taught to program and use Navigator.

This page is intentionally left blank

2 Installation

2.1 Software set-up

2.1.1 System parameters

Navigator I/O connections together with additional settings for navigator are configured in the process configuration database (`PROC`). Actual I/O assignments to real I/O boards are not made by the Navigator installation. The user or system designer must add these definitions to the EIO configuration database.

Below is the default `proc` configuration loaded by Navigator. There is a separate configuration for each task.

```
PROC:CFG_1.0::
# TCMC proc.cfg file

NAVIGATOR_PROP:
  -name "NAVIG_1" -probe_radius 3.175 -course_speed 15 -fine_speed
       3 \
  -detect_input " " -act_sensor_proc "ActSmarTac" \
  -deact_sensor_proc "DeActSmarTac"
```

The search sensitivity can be adjusted by using by modifying the `Max error coarse search` and `Max error fine search` parameters. To change these settings, RobotStudio is preferably used. Default values are 1 mm for coarse search and 0.2 mm for fine search. It is also possible to load an altered `proc.cfg`, the "Add or Replace" feature in RobotStudio must then be used to override the existing fields with the new settings.

Note

Note that Navigator does not install any I/O signals in the EIO configuration database. It provides only a mechanism to connect to existing signals in the system. If the robotic system is not a turnkey system, I/O signals will need to be installed in the system.

2.1.2 Loading software

Software is loaded by purchasing the Navigator baseware option.

3 User scenarios

3.1 Navigator for calibration

3.1.1 General

At installation of robot cells a manual calibration of the relationships between the robot and the external axes, fixtures etc must be performed.

CAUTION

It is very critical to do this step as accurate as possible due to that all robot programs will depend on it.

3.1.2 The calibration process

The calibration process is usually done by localizing calibration points using the robot as measurement system.

Navigator is an automated, very accurate and user independent technique that replaces these manual steps.

The standard idea of localizing calibration points is used by Navigator as well. However, instead of letting the user manually point out positions the robot is equipped with a tool equipped with a tooling ball in the end (called a probe) and the robot cell is prepared with mounting holes on the fixture for tooling balls.

The calibration is performed by letting the robot locate tooling balls placed in the holes on the fixture. Thus, the manual step of pointing out reference positions for calibration is replaced by automated search methods. The calibration points nominal positions are prepared in advance in an off-line programming environment and are not updated on the real the system.

The sensor mechanism is tactile, i.e. the touch between objects is determined and causes the robot to stop.

3.2 Examples

3.2.1 Rotational axis calibration

The basic method to calibrate a rotational external axis is to localize the same position at the turntable or other fixed structure on the external axis at different rotation angles.

The steps to perform the calibration with Navigator consist of:

	Action
1.	Prepare off-line calibration points, i.e. give rough estimates where the points are located. This is easily done in an off-line programming environment such as RobotStudio but can also be done online. Three to ten targets are required. The same tooling ball should be localized, only the external axis angle should be changed between the points.
2.	Attach the probe to robot and calibrate its TCP (BullsEye™ preferred).
3.	Run calibration procedure, i.e. use the Navigator instruction '`SearchSpL`' with the prepared targets as arguments.
4.	Calculate the base frame by using the RAPID instruction '`CalcRotAxisFrame`'.
5.	Restart Controller.

3.2.2 Fixture/workobject calibration

A fixture/workobject can be calibrated in multiple ways. Either use points describing the origin, the x and the y-axis of the coordinate system or more flexible, use any known points in the local object coordinate system. Theses points are usually derived from CAD.

The latter case is calibrated with Navigator in the following steps:

	Action
1.	Prepare off-line calibration points, i.e. give rough estimates where the points are located. This is easily done in an off-line programming environment such as RobotStudio but can also be done online. Three to ten targets are required.
2.	Attach the probe to robot and calibrate its TCP (BullsEye^{TM} preferred).
3.	Run calibration procedure, i.e. use the Navigator instruction '`SearchSpL`' with the off-line prepared targets as arguments.
4.	Calculate the base frame by using the RAPID instruction '`DefAccFrame`'.

3.2.3 Navigator for coordinate measurements

Navigator instruction Measure_1D can be used for various coordinate measurement tasks. Below are two examples.

Fixture cloning

Assume that several identical robot cells are to be manufactured. Each cell has a fixture that needs to be mechanically adjusted.

The first fixture called the master fixture is created and manually adjusted. The robot is then used to measure a number of reference points (surfaces). The points measured on the master fixture are called nominal points.

After the nominal points have been defined the master fixture is removed and a clone fixture is mounted in the same robot cell.

The robot is then ordered to measure the same coordinates on the new not mechanically adjusted fixture. The reference points are compared to the nominal points and a report describing how the clone fixture should be mechanically adjusted is created.

This procedure can be repeated for unlimited number of clone fixtures.

Production batch monitoring

To supervise trends of manufactured parts a number of coordinates are measured on a master part. Subsequent parts/batches are thereafter checked periodically for deviation from the master part by letting Navigator measure the same coordinates and compare them to the master.

The result is logged and production can be stopped if the deviation is too large.

Continues on next page

Work-flow methodology

The workflow to run coordinate measurement using Navigator comprises of the following steps:

	Action

1.	Create robot program.
	Below is an example with five measurement points.
	`MODULE Fixture_LineUp`
	`! GAP declarations`
	`CONST menudata mdCMMSetNominal := ["Setup Nominal Fixture Points", "", "SetNominal", 3, "T_ROB1", GAP_SHOW_ALWAYS, TRUE, GAP_SETUP_TYPE, 255, FALSE];`
	`CONST menudata mdCMMRunBatch := ["Measure Fixture", "", "Run-Batch", 3, "T_ROB1", GAP_SHOW_ALWAYS, TRUE, GAP_SER-VICE_TYPE, 255, FALSE];`
	
	`! Position declarations`
	`CONST robtarget ViaPoint1;`
	`CONST robtarget ViaPoint2;`
	`CONST robtarget ViaPoint3;`
	`CONST robtarget ViaPoint4;`
	`CONST robtarget ViaPoint5;`
	`PERS robtarget rtAppPoint1;`
	`PERS robtarget rtAppPoint2;`
	`PERS robtarget rtAppPoint3;`
	`PERS robtarget rtAppPoint4;`
	`PERS robtarget rtAppPoint5;`
	`PERS robtarget rtNomPoint1;`
	`PERS robtarget rtNomPoint2;`
	`PERS robtarget rtNomPoint3;`
	`PERS robtarget rtNomPoint4;`
	`PERS robtarget rtNomPoint5;`
	
	`! Program data declarations`
	`PERS tooldata tProbe;`
	`PROC RunRCMM()`
	`MoveJ ViaPoint1,v200,fine,tProbe;`
	`Measure_1D rtAppPoint1, rtNomPoint1, v100, tProbe, 50, Measure-Name:="CMM1";`
	`MoveJ ViaPoint2,v200,fine,tProbe;`
	`Measure_1D rtAppPoint2, rtNomPoint2, v100, tProbe, 50, Measure-Name:="CMM2";`
	`MoveJ ViaPoint3,v200,fine,tProbe;`
	`Measure_1D rtAppPoint3, rtNomPoint3, v100, tProbe, 50, Measure-Name:="CMM3";`
	`MoveJ ViaPoint4,v200,fine,tProbe;`
	`Measure_1D rtAppPoint4, rtNomPoint4, v100, tProbe, 50, Measure-Name:="CMM4";`
	`MoveJ ViaPoint5,v200,fine,tProbe;`
	`Measure_1D rtAppPoint5, rtNomPoint5, v100, tProbe, 50, Measure-`

Continues on next page

	Action

	`Name:="CMM5";`
	`ENDPROC`
	`ENDMODULE`

2.	Create nominal points. Execute `RunRCMM`, accessed from GAP Setup dialog.
3.	Run batches. Execute `RunRCMM`, accessed from GAP Service dialog.
4.	Evaluate result.

A log in WebWare will be created. Result will also be displayed on the FlexPendant if running in manual mode.

This page is intentionally left blank

4 Software reference

4.1 Instructions

4.1.1 navgeodata, Navigator geometrical data

Introduction

`Navgeodata` defines the radius of the tooling ball to search for and the search radius.

Description

Navigator is a set of functions for defining calibration points automatically and for quality control measurements. Navgeodata is used to describe the calibration point feature.

Components

`SphereRad`

Data type: `num`

Radius of the cell mounted sphere.

`SearchOffset`

Data type: `num`

Maximum error of the nominal position to search for. If the nominal position in relatively accurately known a small SearchOffset will give faster cycle time.

Example

```
VAR navgeodata ToolBall:=[6.35, 20.0];
```

The exact position of a tooling ball should be defined using Navigator. The tooling ball has a radius of 6.35mm and its position is known within +/- 20mm.

Structure

```
<dataobject of navgeodata >
<SphereRad of num >
<SearchOffset of num >
```

Related information

	Described in:
Linear sphere search	*SearchSpL, Move Linear Search Sphere on page 28*
Joint sphere search	*SearchSpJ, Move Joint Search Sphere on page 32*

4.1.2 SearchSpL, Move Linear Search Sphere

About

Spiral search to localize the exact position of a sphere center given the nominal/CAD position. The movement to reach the approach position is performed linearly.

Example

```
SearchSpL nDist, rtResult, rtToPoint,v500, fine,tTool, spData;
```

The robot moves linearly to the distance nDist above rtToPoint. At the position the robot starts spiral searches with relatively high search speed until four successful hits of the sphere. The high-speed searches are followed by four accurate searches with low speed to determine the exact location of the sphere center. The robtarget rtResult is updated with the search result.

Arguments

```
ApprDist SearchPoint ToPoint Speed Zone Tool [\WObj] SphereData
     [\MeanError] [\MaxError] [\UpdateToPoint] [\SearchName]
     [\TLoad]
```

`ApprDist`

Data type: `num`

The distance above `ToPoint` where the searches will start. The direction of the distance is defined so that the tool orientation coincide with the normal to the surface where the sphere is mounted.

`SearchPoint`

Data type: `robtarget`

The result of the sphere localization, i.e. the exact physical location of the sphere expressed in `Wobj`.

`ToPoint`

Data type: `robtarget`

The nominal position of the sphere center that should be localized oriented such as the tool orientation coincide with the normal to the surface where the sphere is mounted.

`Speed`

Data type: `speeddata`

The speed data used when moving to `ToPoint`. The velocity of the search motion is unaffected.

`Zone`

Data type: `zonedata`

The zone data used when moving to `ToPoint`.

`Tool`

Data type: `tooldata`

The tool used during the search. Both TCP and tool orientation must be valid.

Continues on next page

`[\WObj]`

Data type: `wobjdata`

The work object used during the search. If not selected, `wobj0` is used.

`SphereData`

Data type: `navgeodata`

The definition of radius of the sphere and the search radius.

`[\MeanError]`

Data type: `num`

The mean error of the different searches compared to the final (localized) position.

`[\MaxError]`

Data type: `num`

The max error of the different searches compared to the final (localized) position.

`[\UpdateToPoint]`

Data type: `switch`

If on, the `ToPoint` will be updated with the search result `SearchPoint`.

`[\SearchName]`

Data type: `string`

The name of the search that will be logged if the option Production Monitoring is used.

`[\TLoad]`

Data type: `loaddata`

The `\TLoad` argument describes the total load used in the movement. The total load is the tool load together with the payload that the tool is carrying. If the `\TLoad` argument is used, then the loaddata in the current tooldata is not considered.

If the `\TLoad` argument is set to `load0`, then the `\TLoad` argument is not considered and the loaddata in the current `tooldata` is used instead. For a complete description of the `TLoad` argument, see **MoveL - Moves the robot linearly.**

Program execution

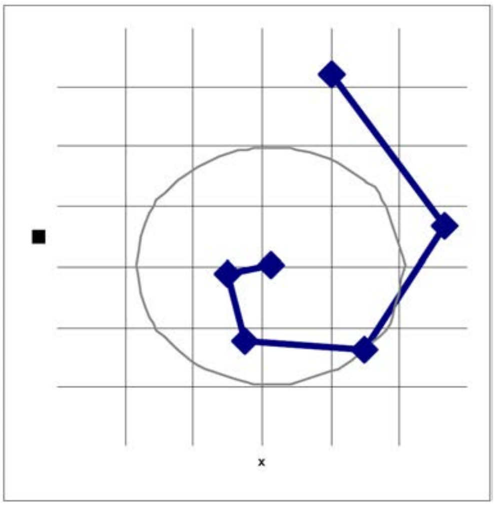
The robot moves linearly to the distance nDist above rtToPoint. The nominal position rtToPoint must have an orientation so that the tool orientation coincides with the normal to the surface where the sphere is mounted. Thus, it is important to not only have a correct TCP but also tool orientation.

At the position the robot starts spiral searches. First a number of coarse searches with relatively high speed.

The coarse search pattern is a segmented spiral search, i.e. a spiral with linear movements. The height above the surface is constant but gradually decreased if no object is found. When the object is found, the robot regains and a new start position is calculated randomly ensuring that not the same contact position is used

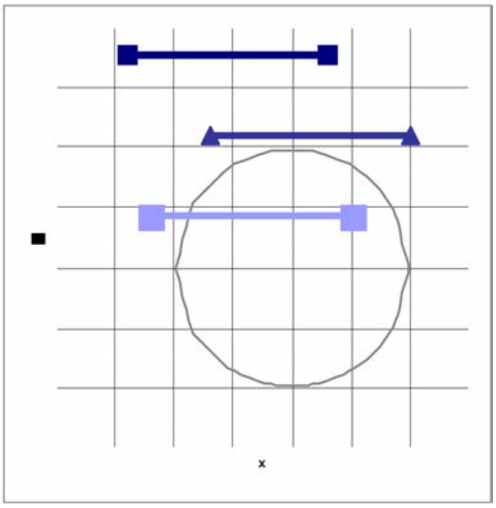
Continues on next page

twice. When four hits are established the coarse location of the sphere is set and four fine searches with low speed follows.



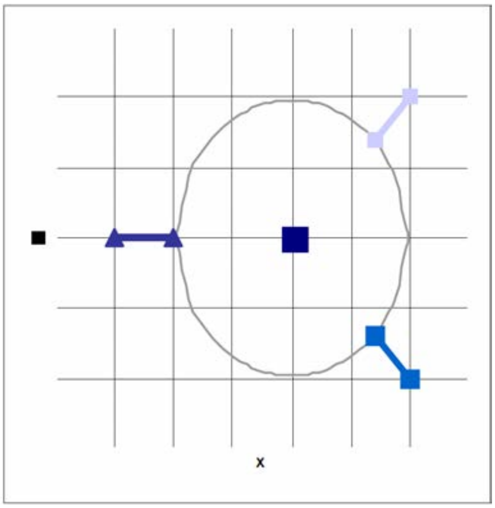
xx1400001414

Figure 4.1: Coarse search pattern - side view



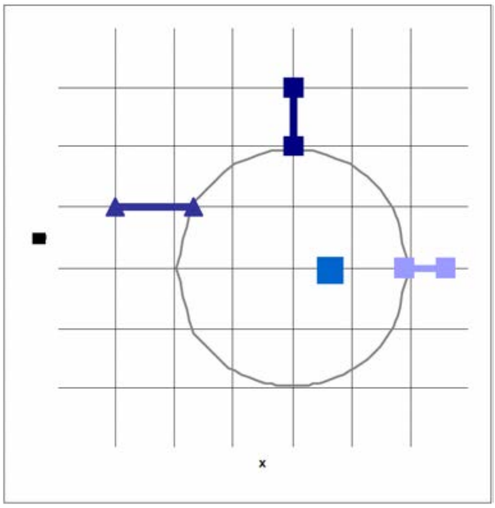
xx1400001415

Figure 4.2: Coarse search pattern - top view



xx1400001416

Figure 4.3: Fine search pattern - top view



xx1400001417

Figure 4.4: Fine search pattern - side view

Examples

```
! Data declarations
PERS robtarget ViaPoint1;
PERS robtarget ViaPoint2;
PERS robtarget rtNomPoint1;
PERS robtarget rtNomPoint2;
PERS robtarget rtResult1;
PERS robtarget rtResult2;
CONST navgeodata spData:=[6.35, 20.0];
VAR num nDist:=20;
PERS tooldata tProbe;
```

Continues on next page

```
PROC CreateCalPoints ()
  MoveJ ViaPoint1,v200,fine,tProbe;
  SearchSpL nDist, rtResult1, rtNomPoint1, v500, fine, tProbe,
       spData \SearchName:="CalPoint1";
  MoveJ ViaPoint2,v200,fine,tProbe;
  SearchSpL nDist, rtResult2, rtNomPoint2, v500, fine, tProbe,
       spData \SearchName:="CalPoint2";
ENDPROC
```

Syntax

```
SearchSpL
  [ ApprDist ':=' ] < expression (IN) of num > ','
  [ SearchPoint ':=' ] < expression (INOUT) of robtarget > ','
  [ ToPoint ':=' ] < expression (PERS) of robtarget > ','
  [ Speed ':=' ] < persistent (IN) of speeddata > ','
  [ Zone ':=' ] < persistent (IN) of zonedata > ','
  [ Tool ':=' ] < persistent (PERS) of tooldata >
  [ '\' WObj ':=' < persistent (PERS) of wobjdata > ]
  [ SphereData ':=' < expression (IN) of navgeodata > ]
  [ '\' MeanError ':=' < expression (INOUT) of num > ]
  [ '\' MaxError ':=' < expression (INOUT) of num > ]
  [ '\' UpdateToPoint]
  [ '\' SearchName ':=' < expression (IN) of string > ]
  [ '\' TLoad':=' ] < persistent (PERS) of loaddata > ] ';'
```

Related information

	Described in:
CalcRotAxisFrame	Technical reference manual - RAPID Instructions, Functions and data types
DefAccFrame	Technical reference manual - RAPID Instructions, Functions and data types
MoveL	Technical reference manual - RAPID Instructions, functions and data types - MoveL
Definition of loaddata	Technical reference manual - RAPID Instructions, functions and data types - loaddata

4.1.3 SearchSpJ, Move Joint Search Sphere

About

Spiral search to localize the exact position of a sphere center given the nominal/CAD position. The movement to reach the approach position does not have to be in a straight line.

Example

```
SearchSpJ nDist, rtResult, rtToPoint,v500, fine,tTool, spData;
```

The robot moves to the distance `nDist` above `rtToPoint`. At the position the robot starts spiral searches with relatively high search speed until four successful hits of the sphere. The `highspeed` searches are followed by four accurate searches with low speed to determine the exact location of the sphere center. The `robtarget rtResult` is updated with the search result.

Arguments

```
ApprDist SearchPoint ToPoint Speed Zone Tool [\WObj] SphereData
     [\MeanError] [\MaxError] [\UpdateToPoint] [\SearchName]
     [\TLoad]
```

`ApprDist`

Data type: `num`

The distance above `ToPoint` where the searches will start. The direction of the distance is defined so that the tool orientation coincide with the normal to the surface where the sphere is mounted.

`SearchPoint`

Data type: `robtarget`

The result of the sphere localization, i.e. the exact physical location of the sphere expressed in `Wobj`.

`ToPoint`

Data type: `robtarget`

The nominal position of the sphere center that should be localized oriented such as the tool orientation coincide with the normal to the surface where the sphere is mounted.

`Speed`

Data type: `speeddata`

The speed data used when moving to `ToPoint`. The velocity of the search motion is unaffected.

`Zone`

Data type: `zonedata`

The zone data used when moving to `ToPoint`.

`Tool`

Data type: `tooldata`

Continues on next page

The tool used during the search. Both TCP and tool orientation must be valid.

`[\WObj]`

Data type: `wobjdata`

The work object used during the search. If not selected, `wobj0` is used.

`SphereData`

Data type: `navgeodata`

The definition of radius of the sphere and the search radius.

`[\MeanError]`

Data type: `num`

The mean error of the different searches compared to the final (localized) position.

`[\MaxError]`

Data type: `num`

The max error of the different searches compared to the final (localized) position.

`[\UpdateToPoint]`

Data type: `switch`

If on, the `ToPoint` will be updated with the search result `SearchPoint`.

`[\SearchName]`

Data type: `string`

The name of the search that will be logged if the option Production Monitoring is used.

`[\TLoad]`

Data type: `loaddata`

The `\TLoad` argument describes the total load used in the movement. The total load is the tool load together with the payload that the tool is carrying. If the `\TLoad` argument is used, then the loaddata in the current tooldata is not considered.

If the `\TLoad` argument is set to `load0`, then the `\TLoad` argument is not considered and the loaddata in the current `tooldata` is used instead. For a complete description of the `TLoad` argument, see MoveL - Moves the robot linearly.

Program execution

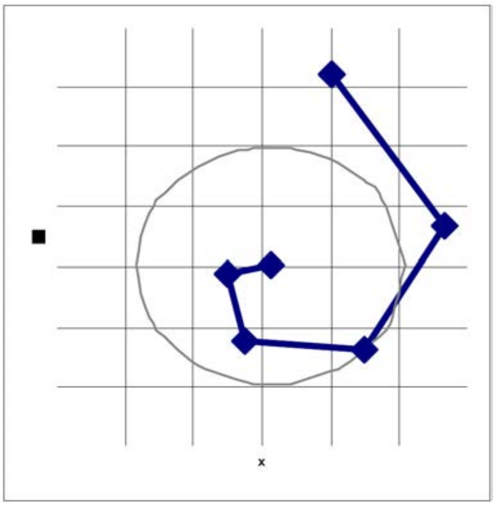
The robot moves to the distance `nDist` above `rtToPoint`. The nominal position `rtToPoint` must have an orientation so that the tool orientation coincides with the normal to the surface where the sphere is mounted. Thus, it is important to not only have a correct TCP but also tool orientation.

At the position the robot starts spiral searches. First a number of coarse searches with relatively high speed.

The coarse search pattern is a segmented spiral search, i.e. a spiral with linear movements. The height above the surface is constant but gradually decreased if no object is found. When the object is found, the robot regains and a new start position is calculated randomly ensuring that not the same contact position is used

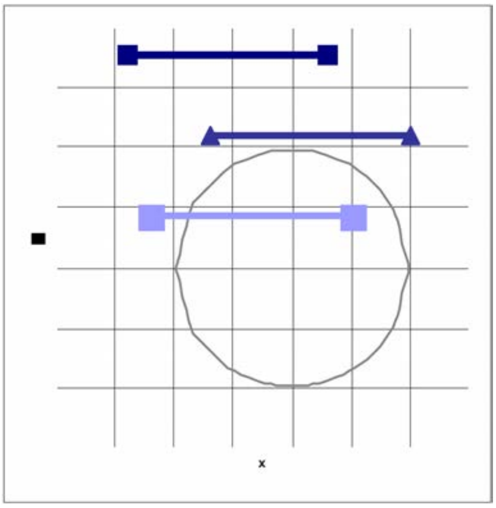
Continues on next page

twice. When four hits are established the coarse location of the sphere is set and four fine searches with low speed follows.



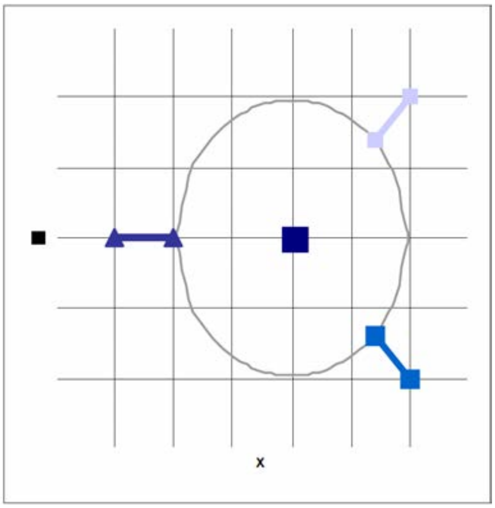
xx1400001414

Figure 4.5: Coarse search pattern - side view



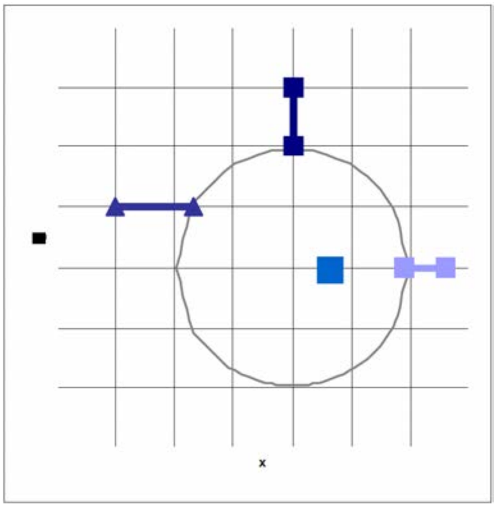
xx1400001415

Figure 4.6: Coarse search pattern - top view



xx1400001416

Figure 4.7: Fine search pattern - top view



xx1400001417

Figure 4.8: Fine search pattern - side view

Examples

```
! Data declarations
PERS robtarget ViaPoint1;
PERS robtarget ViaPoint2;
PERS robtarget rtNomPoint1;
PERS robtarget rtNomPoint2;
PERS robtarget rtResult1;
PERS robtarget rtResult2;
CONST navgeodata spData:=[6.35, 20.0];
VAR num nDist:=20;
nPERS tooldata tProbe;
```

Continues on next page

```
PROC CreateCalPoints ()
  MoveJ ViaPoint1,v200,fine,tProbe;
  SearchSpJ nDist, rtResult1, rtNomPoint1, v500, fine, tProbe,
       spData \SearchName:="CalPoint1";
  MoveJ ViaPoint2,v200,fine,tProbe;
  SearchSpJ nDist, rtResult2, rtNomPoint2, v500, fine, tProbe,
       spData \SearchName:="CalPoint2";
ENDPROC
```

Syntax

```
SearchSpJ
  [ ApprDist ':=' ] < expression (IN) of num > ','
  [ SearchPoint ':=' ] < expression (INOUT) of robtarget > ','
  [ ToPoint ':=' ] < expression (PERS) of robtarget > ','
  [ Speed ':=' ] < persistent (IN) of speeddata > ','
  [ Zone ':=' ] < persistent (IN) of zonedata > ','
  [ Tool ':=' ] < persistent (PERS) of tooldata >
  [ '\' WObj ':=' < persistent (PERS) of wobjdata > ]
  [ SphereData ':=' < expression (IN) of navgeodata > ]
  [ '\' MeanError ':=' < expression (INOUT) of num > ]
  [ '\' MaxError ':=' < expression (INOUT) of num > ]
  [ '\' UpdateToPoint]
  [ '\' SearchName ':=' < expression (IN) of string > ]
  [ '\' TLoad':=' ] < persistent (PERS) of loaddata > ] ';'
```

Related information

	Described in:
CalcRotAxisFrame	Technical reference manual - RAPID Instructions, Functions and data types
DefAccFrame	Technical reference manual - RAPID Instructions, Functions and data types

4.1.4 Measure_1D, One-dimensional measurement

About

Measure_1D is an instruction used for tactilely measuring a coordinate perpendicular to its plane. The result is logged to database and the FlexPendant output window.

Example

```
Measure_1D p1, p2, v20, tProbe, 50, "ExampleMeasure";
```

The robot moves linear to p1 followed by a linear search towards the point p2. When contact is made the difference between p2 and the search position is logged and displayed.

Arguments

```
Measure_1D ApprPoint NominalPoint Speed Tool [\WObj] SetupRadius
    MeasureName [\Diff] [\Setup] [\TLoad]
```

`ApprPoint`

Data type: `robtarget`

The approach point that should be as perpendicular as possible to the NominalPoint. If the argument \SetupRadius is used a search algorithm will be performed to ensure that the approach point is perpendicular.

`NominalPoint`

Data type: `robtarget`

The point where the robot expects to touch the feature. This robtarget is programmed so that the measurement tool is touching the surface of the feature.

`Speed`

Data type: `speeddata`

The speed data used when moving to the approach point. The velocity of the search motion is unaffected.

`Tool`

Data type: `tooldata`

The tool used during the measurement.

`[\WObj]`

Data type: `wobjdata`

The work object used during the measurement. If not selected, `wobj0` is used.

`SetupRadius`

Data type: `num`

Only used if procedure is called from Production Manager Setup or switch Setup is set. Three additional measurements will be performed to define the plane the nominal point belongs to. The first measurement will be the `NominalPoint` and the other two will be at the given radius from `NominalPoint`. The plane defined

Continues on next page

from the measurements will be used to move the approach point `ApprPoint` so it becomes perfectly perpendicular to `NominalPoint`.

`MeasurementName`

Data type: `string`

The measurement will be assigned this identifying name. The name will accompany logs and user dialogs.

`[\Diff]`

Data type: `num`

The result of the search will be available in this optional argument, which corresponds to the absolute distance between the `NominalPoint` and the actual search point.

`[\Setup]`

Data type: `switch`

Used to flag for setup if Production Manager is not used. If Production Manager is used this flag overrides the Production Manager state.

`[\TLoad]`

Data type: `loaddata`

The `\TLoad` argument describes the total load used in the movement. The total load is the tool load together with the payload that the tool is carrying. If the `\TLoad` argument is used, then the `loaddata` in the current `tooldata` is not considered.

If the `\TLoad` argument is set to `load0`, then the `\TLoad` argument is not considered and the `loaddata` in the current `tooldata` is used instead. For a complete description of the `TLoad` argument, see MoveL - Moves the robot linearly.

Program execution

Normal execution

The robot will make a linear movement to the position `ApprPoint`. The search equipment will be activated and motion will start towards the point to measure, `NominalPoint`. The robot will continue past the search point for a maximum total search distance twice the distance between `ApprPoint` and `NominalPoint`. Once the feature is sensed, motion stops, and the distance between `NominalPoint` and search position is logged and displayed on the FlexPendant. If the argument `\UpdateNominal` is used the `NominalPoint` will get the value of the search.

Setup execution, i.e. called from GAP Setup

The robot will make a linear movement to the position `ApprPoint`. The search equipment will be activated and three searches to define the plane the `NominalPoint` belongs to will be made.

1. Search towards `NominalPoint`.
2. Search towards a point at the distance `SetupRadius` from `NominalPoint`.
3. Search towards another point at the distance `SetupRadius` from `NominalPoint`.

Continues on next page

Common for the three searches are that the robot will continue past the search point for a maximum total search distance twice the distance between `ApprPoint` and `NominalPoint`. Once the feature is sensed, motion stops, robot backs up to start position.

After the plane has been defined the `NominalPoint` will be updated. `ApprPoint` will also be updated so it is perpendicular to `NominalPoint`. Thereafter the robot will perform the steps described above in “Normal Execution”.

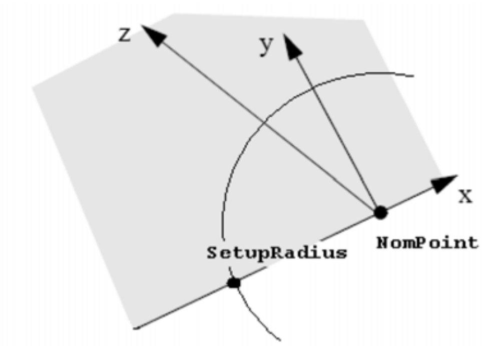
Examples

Study the figure below where the grey area visualizes a plane.

An accurate measurement of the point `NomPoint` is to be made. It is desired to perform the measurement perpendicular to the plane due to that the search will be repeated and there exist a possibility that the plane will move between the repeated searches.

```
PERS tooldata tProbe;
PERS robtarget rtAppPoint;
PERS robtarget rtNomPoint;

PROC Measure()
  Measure_1D rtAppPoint, rtNomPoint, v10, tProbe, 50,
       MeasureName:="Example";
ENDPROC
```



xx1400001418

Figure 4.9: Visualized plane

When the program is executed from GAP Setup four different searches will be performed.

1. First search towards `rtNomPoint`.
2. Search towards point 50mm from `rtNomPoint`.
3. Another search towards point 50mm from `rtNomPoint`.
4. Final search for `rtNomPoint`. The search will be performed perfectly perpendicular to the plane `rtNomPoint` belongs to

Syntax

```
Measure_1D
  [ ApprPoint ':=' ] < expression (INOUT) of robtarget > ','
  [ NominalPoint ':=' ] < expression (INOUT) of robtarget > ','
```

Continues on next page

```
[ Speed ':=' ] < expression (IN) of speeddata > ','
[ Tool ':=' ] < persistent (PERS) of tooldata >
[ '\' WObj ':=' < persistent (PERS) of wobjdata > ] ','
[ SetupRadius expression (IN) of num] ','
[ MeasureName ':=' < expression (IN) of string > ] ','
[ '\' Diff ':=' < expression (INOUT) of num > ]
[ '\' TLoad':=' ] < persistent (PERS) of loaddata > ] ';'
```

Related information

	Described in:
CalcRotAxisFrame	Technical reference manual - RAPID Instructions, Functions and data types
DefAccFrame	Technical reference manual - RAPID Instructions, Functions and data types
MoveL	Technical reference manual - RAPID Instructions, functions and data types - MoveL
Definition of loaddata	Technical reference manual - RAPID Instructions, functions and data types - loaddata

This page is intentionally left blank

Index

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.
5 Lane 369, ChuangYe Road
KangQiao Town, PuDong District
SHANGHAI 201319, China
Telephone: +86 21 6105 6666

www.abb.com/robotics

3HAC050962-001, Rev -, en