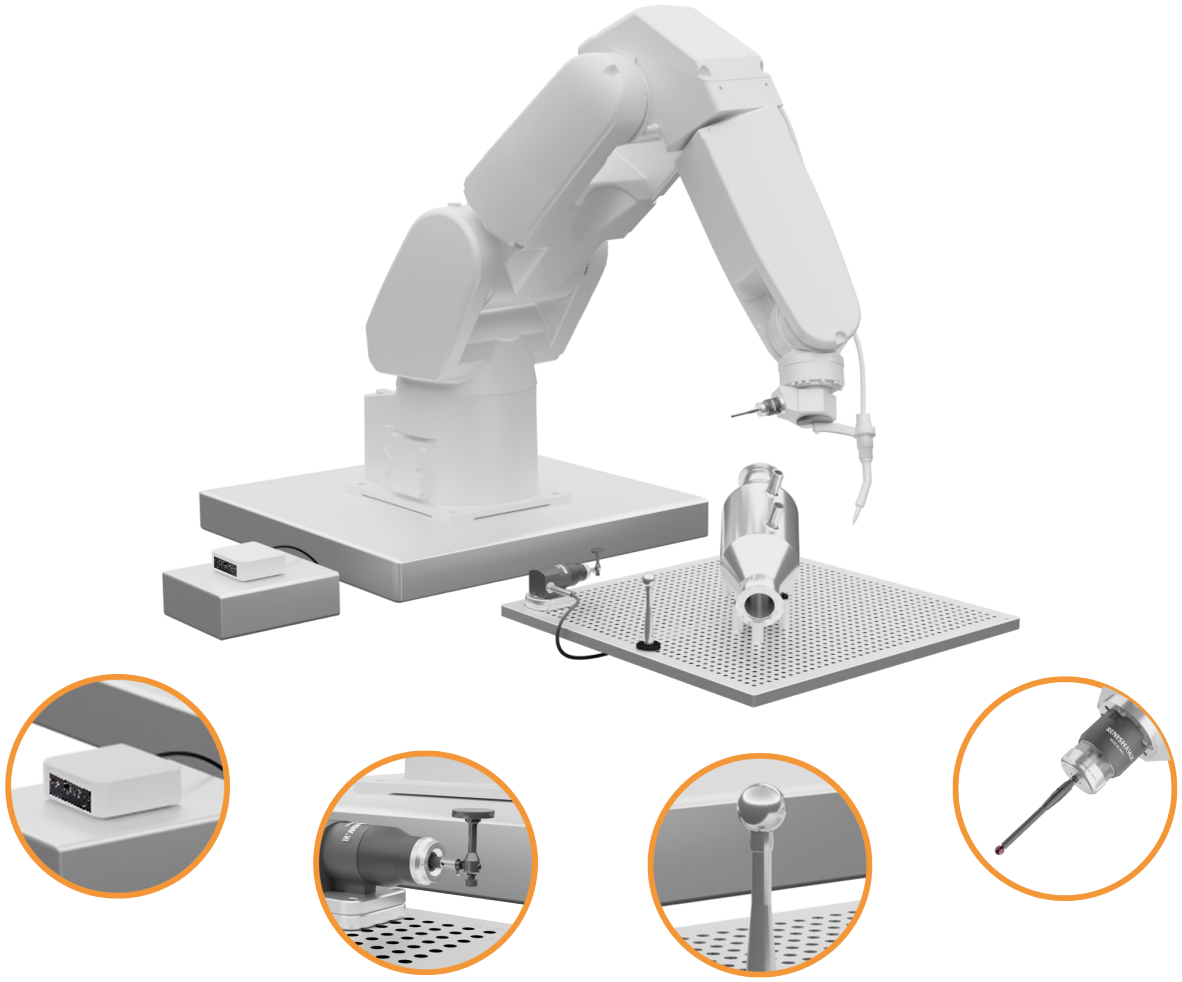
User guide

RENISHAW
apply innovation™

RCS P-series



www.renishaw.com/rcs-probing

#renishaw

Contents

Legal information 4
Warranty 4
Packing material 4
Renishaw software EULA 4
Regulations and conformance 5
Intended use 5
Safety 5
Transmission systems 6
Cleaning and maintenance 7
Datum balls 7
Probes 7
RCS RPU 7
Set up your RCS P-series system 8
Overview 8
Install hardware 9
Install software 10
RCS probing server licensing 12
Use your RCS P-series system 13
Overview 13
Collect, store, and analyse data 14
Program structure 15
Touch parameters 15
Touch points 17
Defining directions 20
Touch moves 25
Calculations 33
Clearing and exporting data 64
Glossary of terms 65
Active 65
Alignment 65
Apparent diameter 65
Approach position 65
Argument 65
Back off distance 65
Best fit 65
Co-ordinate system, frame 66
Datum 66

Feature . . . 67
Move speed . . . 67
Nominal . . . 67
Normal . . . 67
Orthogonal . . . 67
Overtravel . . . 67
Parameter . . . 67
Perpendicular . . . 67
Part setter . . . 67
Residual error . . . 68
RCS probing server, Server . . . 68
Right-hand rules . . . 69
Search distance . . . 70
Signed distance . . . 70
Stem direction . . . 70
Surface . . . 70
Tool setter . . . 71
Touch point . . . 71
Touch speed . . . 71
Vector . . . 71
RCS P-series components . . . 72
Further information . . . 73

Legal information

Warranty

Unless you and Renishaw have agreed and signed a separate written agreement, the equipment and/or software are sold subject to the Renishaw Standard Terms and Conditions supplied with such equipment and/or software, or available on request from your local Renishaw office.

Renishaw warrants its equipment and software for a limited period (as set out in the Standard Terms and Conditions), provided that they are installed and used exactly as defined in associated Renishaw documentation. You should consult these Standard Terms and Conditions to find out the full details of your warranty.

Equipment and/or software purchased by you from a third-party supplier is subject to separate terms and conditions supplied with such equipment and/or software. You should contact your third-party supplier for details. If you purchased the equipment from any other supplier, you should contact them to find out what repairs are covered by their warranty.

Packing material

Packaging component	Material	94/62/EC code	94/62/EC number
Outer box	Non-corrugated fibreboard	PAP	21
Bag	Low-density polyethylene	LDPE	4

Renishaw software EULA

Renishaw software is licensed in accordance with the Renishaw licence at: **www.renishaw.com/legal/en/software-licence-agreement--47112**.

This product also contains third-party software from dotnet, which is subject to the following licence:

The MIT License (MIT)

Copyright (c) .NET Foundation and Contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Regulations and conformance

For regulatory information of RCS P-series hardware products, refer to the individual product installation guides at: **www.renishaw.com/rcs-support**.

Intended use

The intended use of this equipment and accompanying software is to improve the accuracy and repeatability of industrial robots and any associated equipment.

There are risks associated with installation and activities undertaken within the hazard zone of robotic machinery and other hazardous equipment. Consider these risks and manage them to an acceptable level.

Safety

Ensure that you read and understand this user guide and any accompanying product installation guides before using the RCS P-series equipment.

WARNING: To ensure the safety of all users and personnel in the vicinity, we recommend conducting a comprehensive risk assessment of the robot system under test before starting.

A qualified user with machine competency and technical knowledge must conduct the risk assessment. They should take advice from a trained risk assessor and consider the safety of all personnel. We recommend for risk assessors to pay particular attention to robot system movement (in both manual and automatic modes), manual handling, and electrical safety.

Mitigate identified risks before using the systems.

CAUTION: Refer to individual installation guides for a list of user-serviceable parts.

NOTE: Wear appropriate Personal Protective Equipment (PPE) while installing and operating the equipment within a robot cell.

Electrical and power safety

- Do not allow fluids to come into contact with the RCS Robot Processing Unit (RPU) at any time.
- You can power the RPU by the robot's power supply or an external power supply. See the installation guide for your RPU model for further details.
- Never connect any part of the system to devices that are not part of RCS P-series.
- Before connecting wired devices to an electric panel, ensure that it is safe to do so. If unsure, contact a trained electrician.

Mechanical safety

- Manage all cables to avoid or reduce the risk of trip hazards.
- Exercise caution if mounting the system to an external axis that moves or rotates. Beware of cables becoming entangled.
- If operating the robot system with guards or any safety features removed or disabled, it is the responsibility of the operator to take alternative safety measures in line with the operating instructions for the robot system or relevant codes of practice.
- If operating the robot system within a guarded cell, it is the responsibility of the operator to follow safe practice.

Transmission systems

Radio

Some RCS P-series devices use radio transmission systems.

RCS P-series devices comply with FCC regulations and operate in the 2.4 GHz band. They deliver interference-free transmission by using hybrid FHSS (frequency-hopping spread spectrum) technology, which allows multiple systems to operate in the same machine shop without risk of cross-interference.

The robot and its control system will be exposed to these radio transmissions. Ensure that the use of this Renishaw equipment, when close to the industrial robot and other equipment, does not cause hazards, particularly from unintended motion.

Optical

Some RCS P-series devices use forms of optical transmission.

These products contain LEDs that emit both visible and invisible light. We recommend that you do not stare at or look directly into any LED device.

Cleaning and maintenance

Datum balls

Before use, check that all datum balls are clean and undamaged. Use a soft lint-free cloth to sweep around the entire datum ball.

To order replacement balls, contact your local Renishaw representative.

Probes

Clean stylus tips regularly to capture reliable data. Contact your local Renishaw office for help or advice regarding the cleaning and care of your stylus.

Wipe probes and interfaces with a clean cloth to reduce dirt build-up. Liquids or dust accumulating on the RCS R-PK1, RCS O-PK1, RCS RMI-QE, or RCS OMI-2 may affect transmission performance.

RCS RPU

Do not clean RPUs with liquids. Use a dry cloth to remove dust regularly.

Service personnel must check the RPU in the following situations:

- The power cord is damaged.
- Liquid penetrates the equipment, or it is exposed to moisture.
- The equipment is malfunctioning or does not operate according to the user guide.
- The RPU is dropped or damaged.
- The RPU shows obvious signs of breakage.

NOTE: Dismantling and repair of Renishaw equipment is a highly specialised operation, which must be carried out at an authorised Renishaw Service Centre. Return equipment that requires repair or attention under warranty to your supplier.

Set up your RCS P-series system

Overview

An RCS P-series system comprises of a combination of touch-trigger probes, software, and processing units.

Each RCS P-series set-up is individual to your specific robot cell. **Figure 1** shows a typical set-up, including:

1. Part-setting probe (hard-wired, or wireless radio/optical with receiver)
2. Tool-setting probe (hard-wired, or wireless radio with receiver)
3. RPU running a licensed RCS probing server application
4. Datum ball

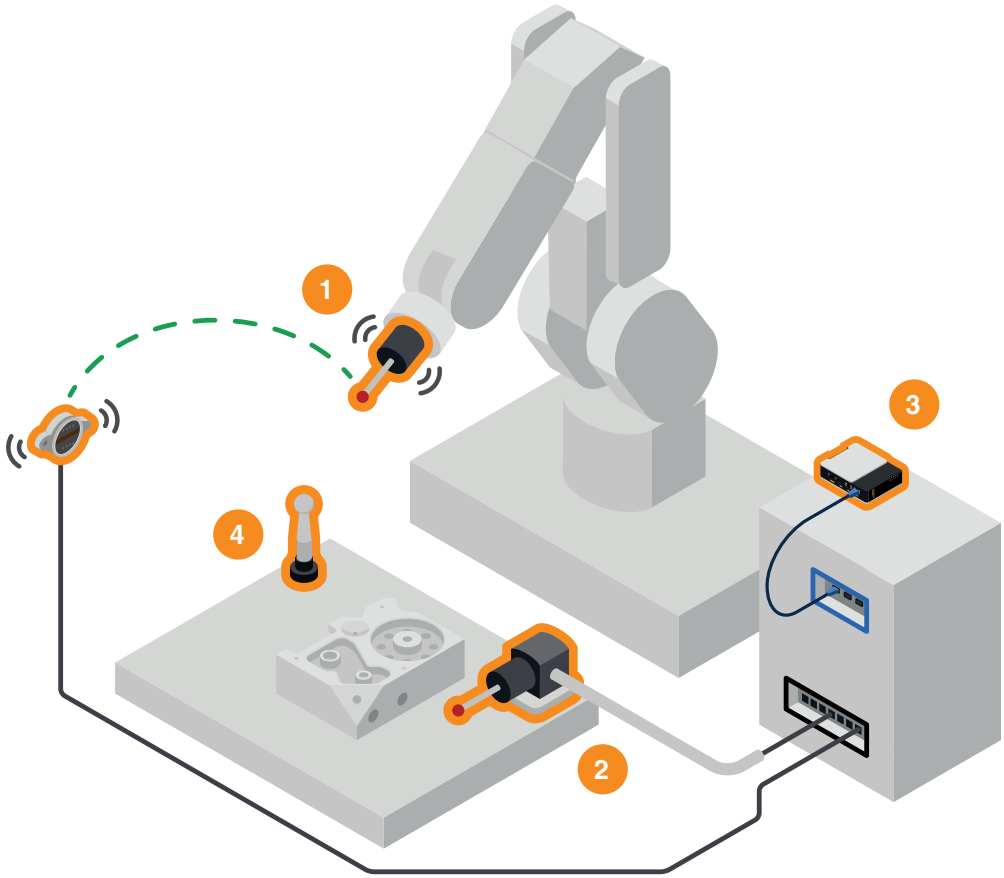


Figure 1 RCS P-series system example

NOTE: All interfaces connect via digital Input/Output (I/O), but wireless probes communicate with their interface via radio or optical transmission.

Install hardware

To install your RCS P-series hardware devices, refer to the installation guide for your chosen products at **www.renishaw.com/rcs-support**.

For datum ball installation instructions, see “Mount your datum ball”, on page **37**.

Positioning products in your robot cell

Mount tool-setting probes and datum spheres within the robot’s work envelope. Make sure that there is enough clearance for the robot to move without interfering with other components in the cell.

Mount part-setting probes and datum balls on your robot so that they do not interfere with other end-of-arm tooling (EOAT). The EOAT must not interfere with the operation of the part setter or datum ball.

Tools required

You need the following tools to install your RCS P-series system:

- Hex keys
- Bootlace ferrule crimps
- Small side cutters
- Small flat head screwdriver
- Cross head screwdriver
- Tape measure

Renishaw part setters and tool setters packaging include any required specialist tools.

Install software

RCS P-series features two software applications to enable robot probing.

RCS probing server

The system requires RCS probing server (the Server) to function. The Server runs on your RPU. It stores, analyses, and computes all data collected using probes. The Server exchanges data with your robot to perform the processes associated with an RCS P-series system.

We have developed custom parameters and macro instructions to support RCS P-series:

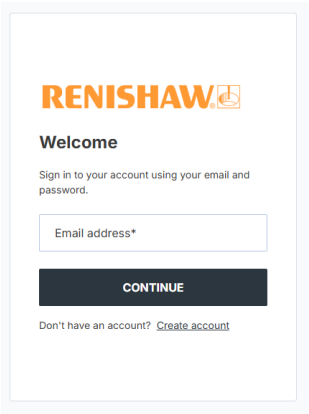
- Communication parameters enable system communication.
- Touch parameters control the robot's behaviour.
- Macros manage system communication and Renishaw probes, as well as collect and analyse data.

RCS Software Suite

RCS Software Suite supports the function of other RCS products, but is an optional application to use with RCS P-series. It features the 'Probing Alignment' datum builder, which creates program templates for simple and complex alignments using CAD or nominal surface data. RCS Software Suite helps to reduce time spent writing RCS P-series programs. See "Align tool and part frames", on page **42**.

Download software

1. On your PC, go to **www.renishaw.com/softwarelicensing/en/connect/RCSSoftwareSuite**.
2. Sign in to your Renishaw.com account.
3. If you do not have a Renishaw.com account, click **Create account** and follow the instructions.



4.
5. In 'Your Downloads', click **RCSSoftwareSuite**.
6. In the 'New Versions' tab, select the software package you want to install.
7. Take the steps in the following sections to install RCS probing server or RCS Software Suite.

NOTE: If the software packages do not appear in your downloads list, contact your local Renishaw representative.

Install RCS probing server

1. Download the file ‘RPU_[ONE/TWO]_[version number].tar”.

NOTE: The .tar file name differs for each software version. For example, the file name of the 1.9.0 release of RCS probing server for the RCS RPU-1 device is RPU_ONE_1.9.0_tar.

2. Access the RPU browser page from your PC.
   a. Connect your PC via Ethernet to the RPU service port (LAN 2, or ).
   b. Set your PC IP address to 10.0.0.2.
   c. In your web browser, type the static IP address 10.0.0.1 into the web address field.

NOTE: For further instruction, refer to the installation guide for your RPU (Renishaw part no. H-6852-8041 for RCS RPU-1, H-6852-8031 for RCS RPU-2)

3. On the browser page, go to the ‘Update’ tab.
4. Click **Choose File**, and upload the ‘RPU_[ONE/TWO]_[version number].tar’ file from your PC.
5. Click **Submit**.

Install RCS macros and parameters

1. Download the following compressed folders:
   a. The folder named after your robot’s manufacturer
   b. The ‘Macro syntax and descriptions’ folder
2. Extract the compressed folders to a location on your computer.
3. Open the ‘Macro syntax and descriptions’ HTML file for your robot manufacturer.
4. Follow the instructions in the ‘Install RCS macros and parameters’ section of the HTML file.

NOTE: Install the macros on your robot controller to use them in your robot program. Macro names differ across manufacturers to account for their respective programming language requirements.

Many macros use arguments, which you need to enter for the robot to run the command. The data types for each macro argument can also differ across robot manufacturers.

The ‘Macro syntax and descriptions’ file for your robot manufacturer provides the syntax and data types required to use the macros in your robot’s programming language.

Install RCS Software Suite

1. Download the file ‘rcs_suite_[version number]_x64.exe’.

NOTE: The .exe file name differs for each software version. For example, the file name of the 1.19.0 release of RCS Software Suite is rcs_suite_19.0.0_x64.exe.

2. Open the .exe file to view the set-up wizard.

3. Follow the steps in the set-up wizard to install RCS Software Suite.

RCS probing server licensing

The Server application runs on the RCS RPU. Server licences are contained on a USB dongle that inserts into the RPU.

To run a given calculation (see “Calculations”, on page **33**), you need the licence required for that calculation. RCS core probing (Renishaw part no. CS-SOF-SW-19-7202) is the standard licence that you need to run basic calculations, such as probe calibration, simple alignments, and statistical analysis.

Expand the capabilities of your RCS P-series system by purchasing the following add-on licences:

- **RCS advanced probing** (Renishaw part no. CS-SOF-SW-19-7211). For computing complex alignments, best fit features, and analyse relationships between features.
- **RCS spindle calibration** (Renishaw part no. CS-SOF-SW-19-7221). For calibrating spindle cutting tools in five degrees of freedom (DOF).

Install and activate probing licences

NOTE: USB dongles purchased with RCS RPU-1 devices come pre-installed with a core probing licence. Use Renishaw Licence Manager to install advanced probing or spindle calibration add-on licences.

NOTE: RCS RPU-2 can connect with up to five robots. Contact your local Renishaw representative to discuss licences for multiple robot connections. Purchase RCS probing licences at **store.renishaw.com.**

Use your RCS P-series system

Overview

RCS P-series offers an in-process probing solution for robotic automation cells.

RCS P-series enables you to:

- Precisely locate tool and part frames.
- Recover your cell after interruptions.
- Best fit basic geometrical features, such as planes, cylinders, or spheres.
- Inspect the relative orientation and position between features.
- Calibrate a spindle cutting tool axis, length, and diameter.

Required knowledge

To use an RCS P-series system, you must have working knowledge of robot operation and programming. You need to:

- Structure and write robot programs in your robot manufacturer's programming language.
- Call and combine macros within a robot program.

Linking all components of a robot cell

All positions of a robot are located relative to its base co-ordinate system. The robot determines the position of its flange relative to the base using its kinematic model. RCS P-series enables the robot to locate the positions of all other components of the cell using touch-trigger probes.

Calibrating a part-setting probe locates its tool centre point (TCP) relative to the robot’s flange co-ordinate system. The part setter can then locate all parts in the cell relative to the robot base.

Calibrating a tool-setting probe locates its TCP relative to the robot base. The tool setter can then locate all tools relative to the robot flange.

This process links all components of the robot cell together, constraining up to six degrees of freedom (DOF).

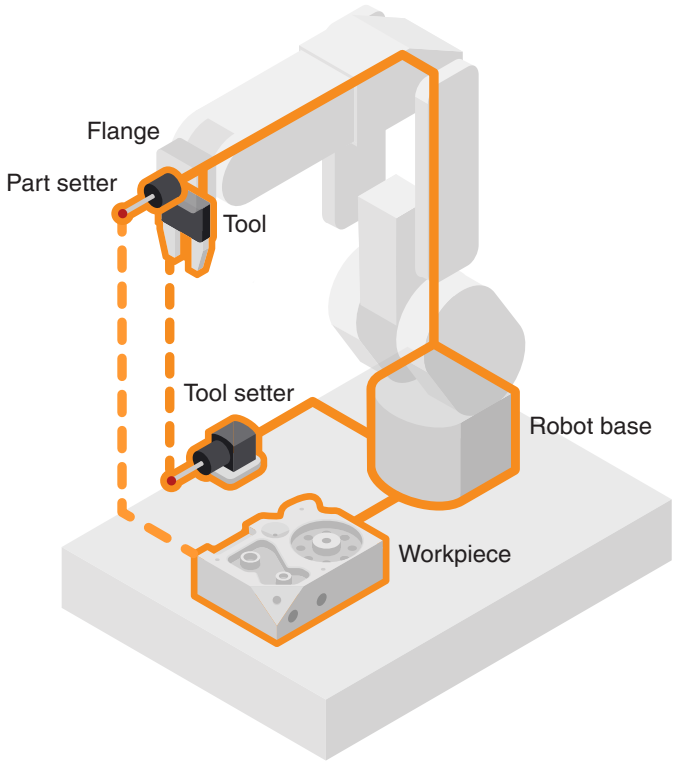


Figure 2 Probes connect all components of a robot cell so that their positions and orientations are known relative to each other.

Collect, store, and analyse data

RCS P-series uses custom-developed macros to collect, store, and analyse data.

Touch move macros tell the system to collect and store position data of tool and part surfaces using Renishaw touch-trigger probes. See “Touch moves”, on page **25**.

The Server stores data under ID numbers.

- Surface IDs store the position data that a probe collects.
- Feature IDs store best fitted feature data, which the Server calculates from surface ID data.
- DatumConfig IDs are files that define nominal position, orientation, and surface IDs of features used in complex alignment calculations.

Calculation macros tell the Server to analyse data stored in its IDs. See “Calculations”, on page **33**.

When the Server calculates data for the robot to use, such as tool and part frames, the robot stores the data in its system.

Program structure

Most RCS P-series programs follow the 'MICE' structure (Manage, Initialise, Capture, Evaluate). All steps in the following structure have accompanying RCS macros, unless otherwise stated.

1. **Manage**

    a. Activate the probe that you intend to capture data with.

    b. Power on the probe.

    c. Optionally, adjust any variable touch parameters (see "Variable touch parameters", on page **17**).

    d. If applicable for your robot manufacturer, declare any variables required for your program. This does not require RCS macros. Use your robot's convention to declare variables.

2. **Initialise**

    a. Initialise the system for the calculation you intend to do (see "Calculations", on page **33**). This clears existing data from the surface IDs you intend to use, ready to store new data. Where required, initialisation macros also set the tool and part frame for the robot to use in the program.

3. **Capture**

    a. Program the robot to move to suitable approach positions for probing. This does not require RCS macros. Use your robot's convention to program the positions.

    b. Probe surfaces and collect position data to use in the intended calculation (see "Touch moves", on page **25**).

4. **Evaluate**

    a. Instruct the Server to do the calculation you initialised for in step 2 (see "Calculations", on page **33**).

Touch parameters

Touch parameters control the robot's behaviour when running touch moves. Your robot stores default constant values for each touch parameter. However, you can modify some touch parameters within your robot program.

CAUTION: Modifying some touch parameter values without recalibrating the probe can negatively affect the accuracy and repeatability of results.

Refer to the parameter tables on page **16** and page **17** for information about the touch parameters.

NOTE: Parameters marked with an asterisk (*) are only applicable to select robot manufacturers. Refer to your touch parameters file for the parameters required for your robot manufacturer.

Constant touch parameters

Edit the values shown in the following table from the touch parameters file in a text editor, or your robot programming software, if available.

Touch parameter	Definition
Part setter – datum ball diameter	The diameter of the datum ball used to calibrate the part setter in millimetres.
Part setter – nominal tip diameter	The diameter of the part setter's stylus tip in millimetres. When the Server runs a probe calibration, it calculates and stores the exact tip diameter to use in subsequent calculations.
*** Part setter – input type**	The input signal type for the part setter.
*** Part setter – input index**	The robot's digital input for the part setter. The robot checks this input for trigger signals when the part setter is the active probe.
Part setter – transmission type	Indicates whether the part setter is a hard-wired or wireless probe.
*** Part setter – output index (wireless probes only)**	The robot's digital output that connects the part setter to the machine start input in the wireless probe receiver.
Tool setter – datum ball diameter	The diameter of the datum ball used to calibrate the tool setter in millimetres.
Tool setter – nominal tip diameter	The diameter of the tool setter's stylus tip in millimetres. When the Server runs a probe calibration, it calculates and stores the exact tip diameter to use in subsequent calculations.
Tool setter – disc stylus thickness	The thickness of the disc stylus that the tool setter uses during spindle calibration, in millimetres.
*** Tool setter – input type**	The input signal type for the tool setter.
*** Tool setter – input index**	The robot's digital input for the tool setter. The robot checks this input for trigger signals when the tool setter is the active probe.
Tool setter – transmission type	Indicates whether the tool setter is a hard-wired or wireless probe.
*** Tool setter – output index (wireless probes only)**	The robot's digital output that connects the tool setter to the machine start input in the wireless probe receiver.
*** Communication error – input type (wireless probes only)**	The input signal type for communication errors in wireless probes.
*** Communication error – output index (wireless probes only)**	The robot's digital output that connects to the probe receiver to detect communication errors with wireless probes.
*** Low battery – input type (wireless probes only)**	The input signal type for low battery in wireless probes.
*** Low battery – output index (wireless probes only)**	The robot's digital output that connects to the probe receiver to detect low battery in wireless probes.
Active probe – default	The default probe whose input the robot checks for trigger signals. Use macros to change the active probe in your robot program.
Back off distance – default	The default value of the back off distance. Use macros to change the back off distance, or reset it to its default value, in your robot program.
Move speed – default	The default value of the move speed. Use macros to change the move speed, or reset it to its default value, in your robot program.
Search distance – default	The default value of the search distance. Use macros to change the search distance, or reset it to its default value, in your robot program.
Touch speed	The robot's speed when approaching a surface to collect position data (touch points).
Double touch behaviour – On/Off default	The default state of the double touch behaviour (on or off). Use macros to turn the double touch behaviour on or off, or reset it to its default value, in your robot program.
Double touch – touch speed	The robot's speed when approaching a surface to take the first of a double touch. The robot does not collect position data when approaching surfaces at this speed.
Double touch – back off distance	The distance that the robot moves away from a surface after taking the first touch of a double touch.
Spindle calibration stem – stem diameter	The diameter of the spindle calibration stem in millimetres.
Spindle calibration stem – tip diameter	The diameter of the spindle calibration stem's ball-ended tip in millimetres.

Variable touch parameters

Use macros in your robot program to change variable touch parameters from their default value, or reset them to their default value.

Touch parameter	Definition	Associated macros
Active probe	The probe whose input the robot checks for trigger signals. The 'Reset all' macro does not reset the active probe to its default.	Activate part setter
		Activate tool setter
Back off distance	The distance that the robot moves away from a surface after touching it. Back off distance is between the touched surface and the surface of the probe stylus tip.	Set new back off distance
		Reset to default value
		Reset all
Double touch behaviour	When enabled, the robot touches a surface twice to collect data for a single touch point.	Enable/disable behaviour
		Reset to default behaviour
		Reset all
Move speed	The speed that the robot moves at when running any moves apart from collecting touch point data.	Set new move speed
		Reset to default value
		Reset all
Search distance	The maximum distance that the robot travels when searching for a surface. If the robot does not detect a probe trigger after travelling the search distance, it will stop and display an error in the teach pendant.	Set new search distance
		Reset to default value
		Reset all

Touch points

When a probe touches a surface, it sends a trigger signal – this is a 'touch point'. The robot records the position of the touch point and sends it to the Server. To record the accurate position of a touch point, the robot must approach it slowly. The robot can record the position using either a single touch, or a double touch.

Choose whether to enable single or double touch behaviour depending on your robot's capability.

If the robot has the option to use a rapid input to detect probe triggers, it is likely suitable to enable single touch behaviour.

If the robot's probe trigger input synchronisation delay is high, it will overtravel before detecting a trigger signal. This will record an incorrect and highly variable position for the touch. To prevent this, enable double touch behaviour.

Single touch

When taking a single touch, the robot moves at the **touch speed** until it detects a probe trigger, or until it has moved the **search distance**. After a probe triggers, the robot stops, records the probe position, and sends it to the Server. Then, the robot moves in the opposite direction at the **move speed** until it has moved the **back off distance**.

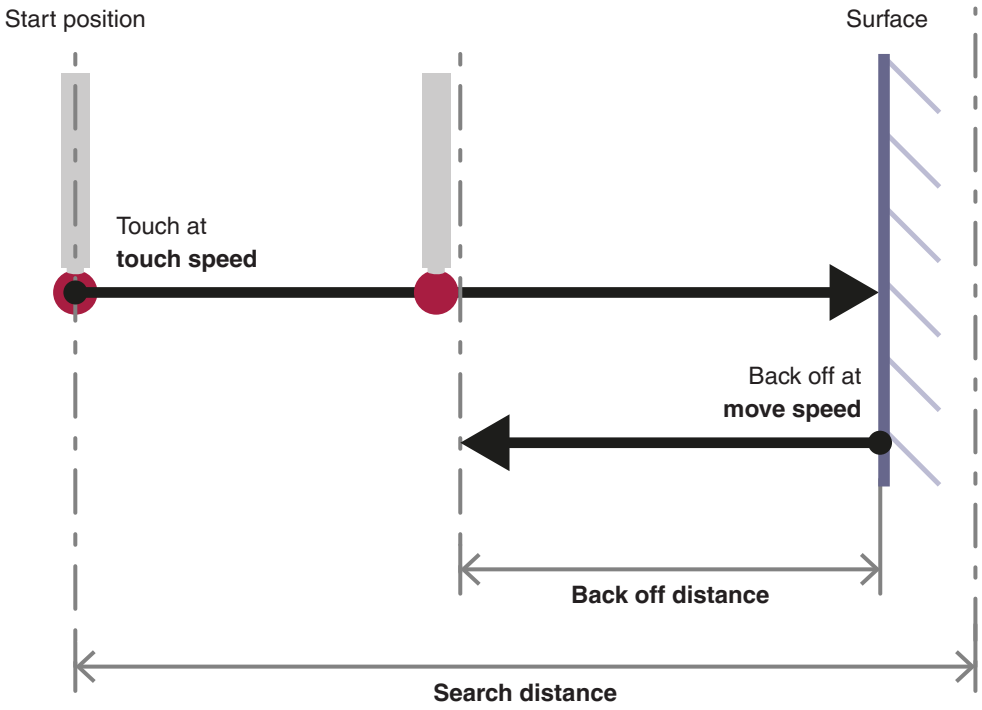


Figure 3 Single touch behaviour of a robot-mounted part setter.

NOTE: Only use single touch behaviour when using a rapid input to detect probe triggers. If your robot does not have a rapid input, use double touch behaviour instead.

Double touch

When taking a double touch, the robot moves at the **double touch speed** until it detects a probe trigger, or until it has moved the **search distance**. After a probe triggers, the robot stops and moves in the opposite direction at the **move speed** until it has travelled the **double touch back off distance**. The robot then moves in the original direction at the **touch speed** until it detects a probe trigger. After the second probe trigger, the robot stops, records the probe position, and sends it to the Server. Then, the robot moves in the opposite direction at the **move speed** until it has travelled the **back off distance**.

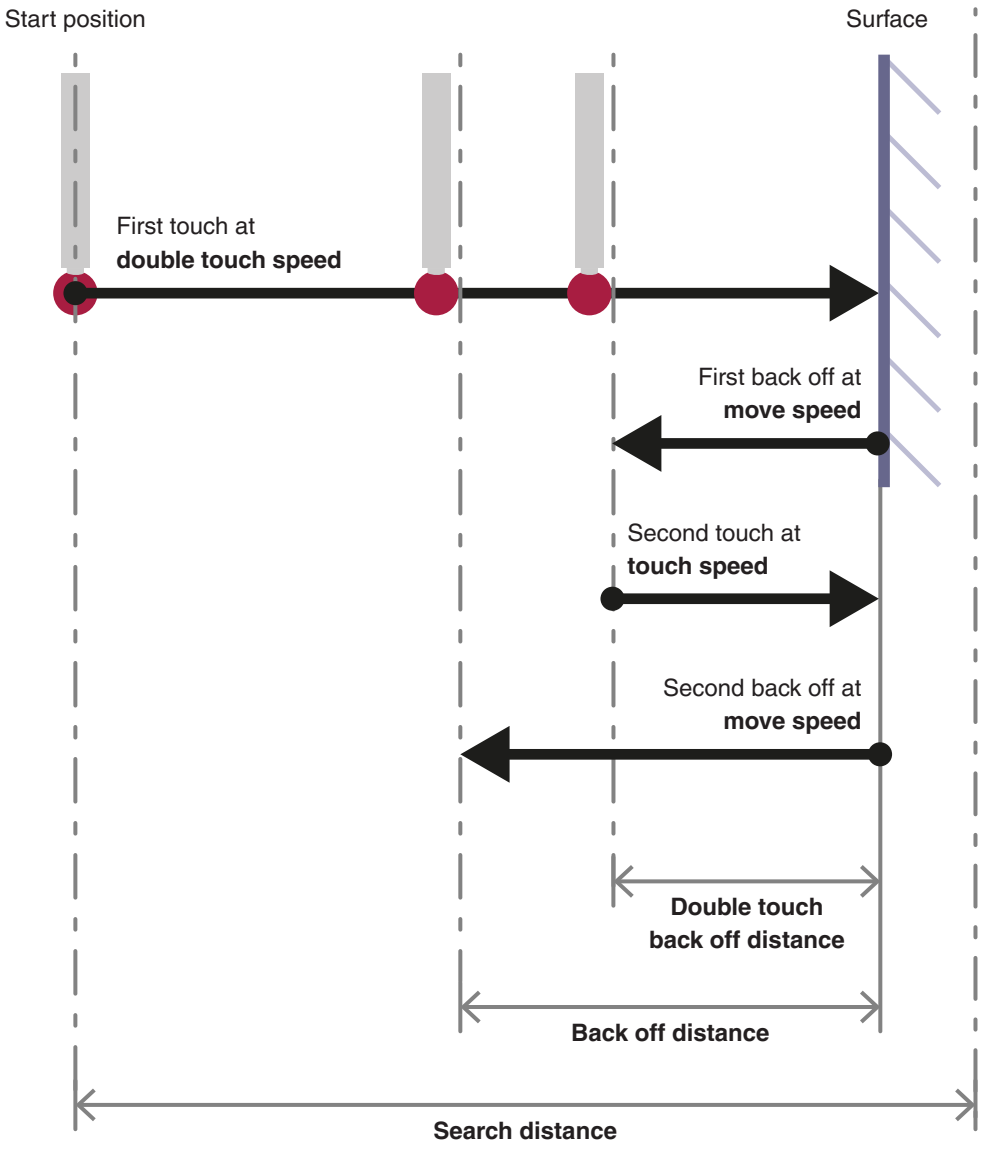


Figure 4 Double touch behaviour of a robot-mounted part setter.

Defining directions

RCS macros use unit vectors to define directions. Examples of these unit vectors include the following:

- Touch direction – The direction that the robot moves in to collect a touch point.
- Normal direction – The unit vector that is perpendicular to a surface (see “Normal directions”, on page **24**).
- Sphere stem direction – A stem is the supporting structure of a sphere. Its direction is the line pointing from the stem base towards the sphere centre.

Unit vectors

A unit vector has a length of 1, so it can define the direction of a vector without its magnitude.

Unit vectors have an X, Y, and Z component. The unit vectors along the X, Y, and Z axes are called $\vec{\mathrm{i}}$, $\vec{\mathrm{j}}$, and $\vec{\mathrm{k}}$ respectively (**Figure 5** and **Figure 6**).

$$\vec{\mathrm{i}} = (1,\ 0,\ 0)$$

$$\vec{\mathrm{j}} = (0,\ 1,\ 0)$$

$$\vec{\mathrm{k}} = (0,\ 0,\ 1)$$

Figure 5 Unit vectors $\vec{\mathrm{i}}$, $\vec{\mathrm{j}}$, and, $\vec{\mathrm{k}}$ expressed by their X, Y, and Z components.

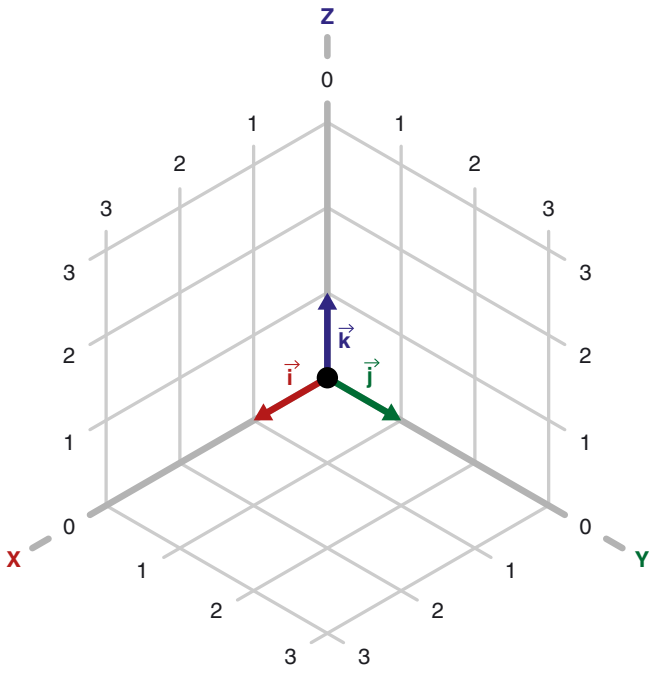


Figure 6 Unit vectors $\vec{\mathrm{i}}$, $\vec{\mathrm{j}}$, and $\vec{\mathrm{k}}$ represented in a 3D space.

A vector, $\vec{V}$, which has both a direction and a magnitude, can be represented as shown in **Figure 7**. The unit vectors $\vec{i}$, $\vec{j}$, and $\vec{k}$ act as multipliers with their corresponding x, y, and z co-ordinate to determine the direction and magnitude of the X, Y, and Z components of the vector.

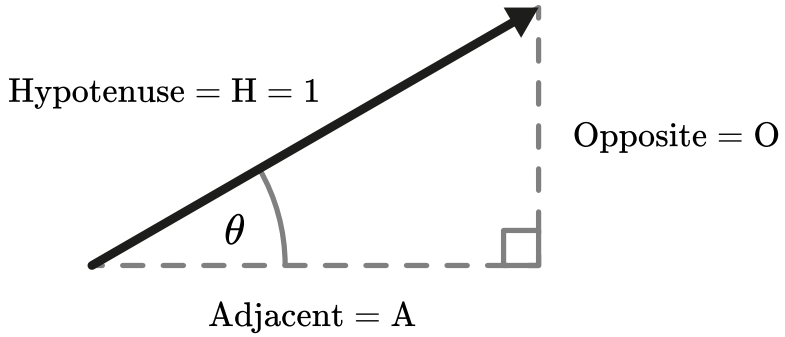
$$\vec{V} = x\,\vec{i} + y\,\vec{j} + z\,\vec{k}$$

Figure 7 The x, y, and z values denote the magnitude of the X, Y, and Z components. The $\vec{i}$, $\vec{j}$, and $\vec{k}$ values denote the directions of the X, Y, and Z components.

NOTE: When a macro requires a direction definition, you need to provide numeric values for x, y, and z as arguments in the macro parameters, unless otherwise stated.

Calculating the components of a unit vector

The angle (θ) of a unit vector dictates its direction. Use trigonometric functions to calculate the values required to define the angle, where the hypotenuse (H) of the triangle acts as the unit vector (**Figure 8** and **Figure 9**). Refer to **Figure 10** for the X, Y, and Z values of common 2D vectors.



$$\sin\theta = \frac{O}{H} \qquad \cos\theta = \frac{A}{H}$$

$$O = H \times \sin\theta \qquad A = H \times \cos\theta$$

Figure 8 Define the direction of unit vector 'H' using sine and cosine functions to calculate the lengths of the opposite and adjacent vector components.

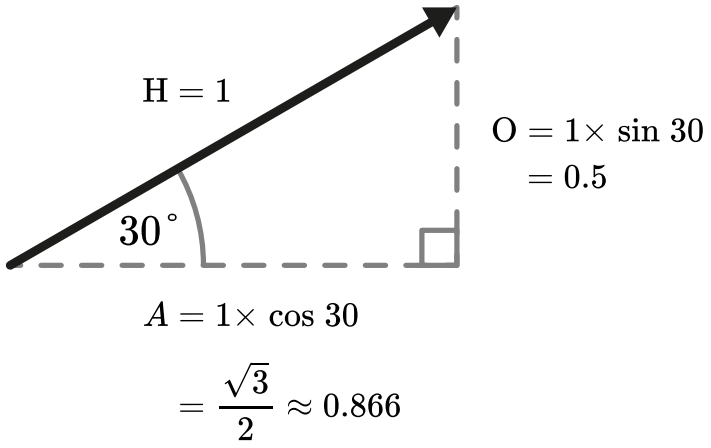


Figure 9 Shows the required lengths of the opposite and adjacent vectors to result in unit vector 'H' having a 30° angle.

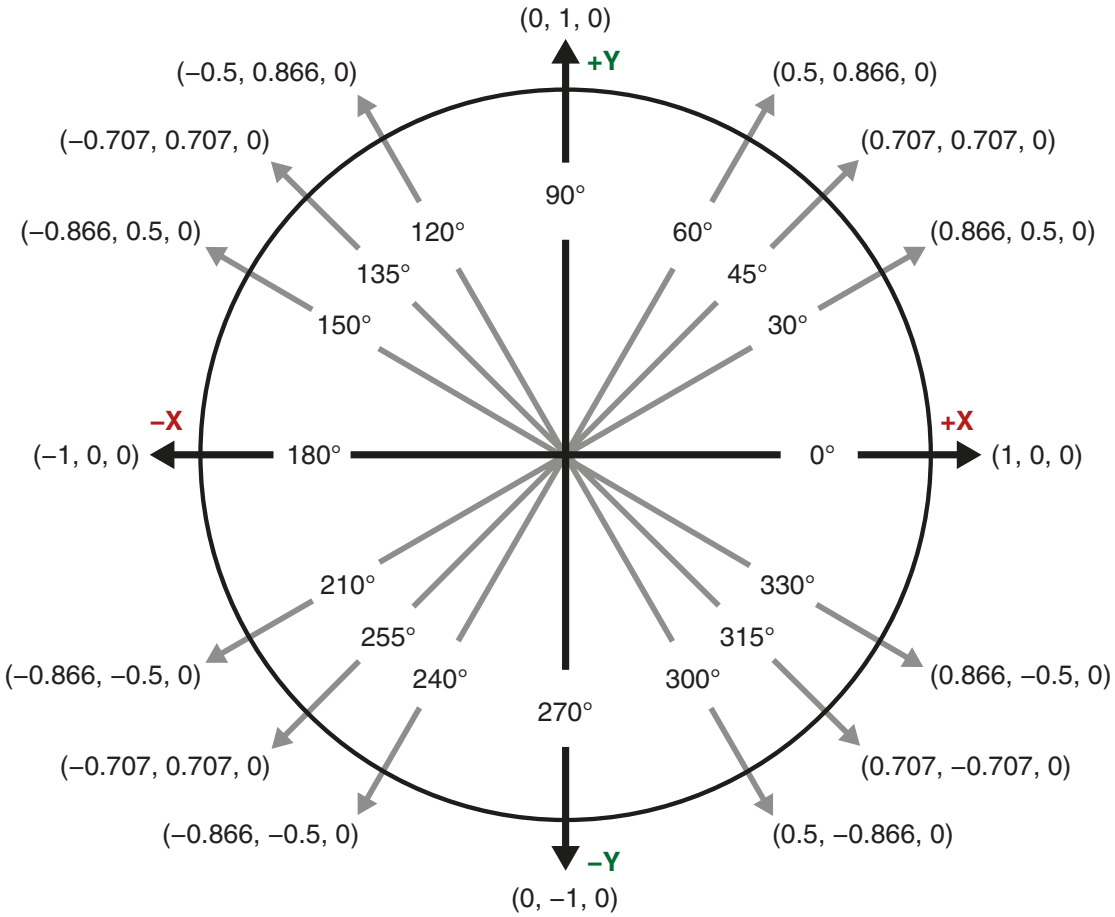


Figure 10 Common 2D vectors.

Normalising vectors

When defining a direction in RCS macros, you can also use values greater than 1 for x, y, and z. In this case, the Server 'normalises' the vector, converting it into a unit vector while preserving its direction.

For example, by mapping the vector (3, 3, 0) onto a grid, you can calculate its angle as 45°. Normalising the vector keeps the 45° direction, but adjusts its magnitude to 1 (**Figure 11**).

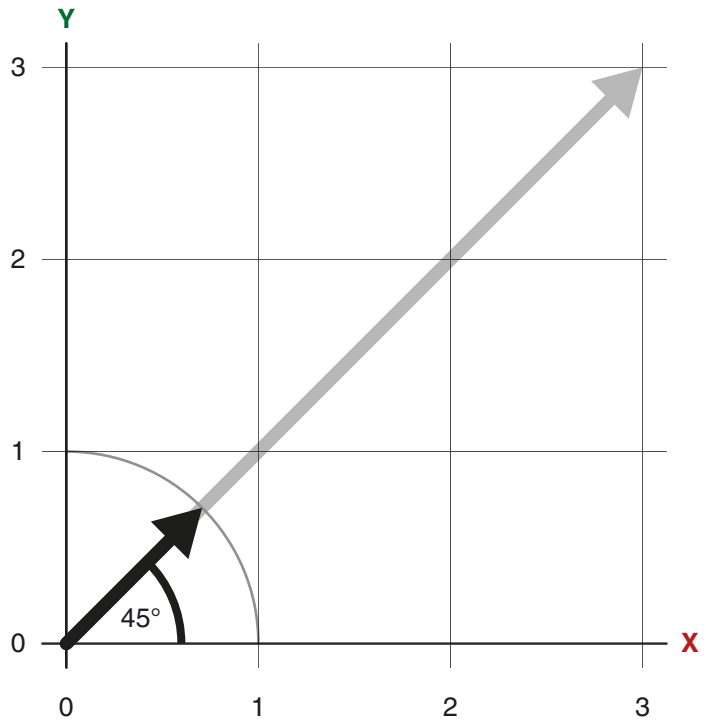


Figure 11 Normalisation of a vector (3, 3, 0) to calculate its direction while setting its magnitude as 1.

Normal directions

A normal vector is a vector that is perpendicular to a surface. A normal vector with a length of 1 (a unit vector), is the normal direction of the surface.

The normal direction indicates which direction the surface or feature is ‘facing’ within a given frame, which the RCS P-series system uses for some calculations. The normal direction points ‘out’ of the feature. The example in **Figure 12** shows a probe touching a plane on a part. The Server calculates the plane’s normal direction as pointing towards the side that the probe approached from, perpendicular to the plane. In this example, the normal direction points in the +Z direction relative to the active part frame.

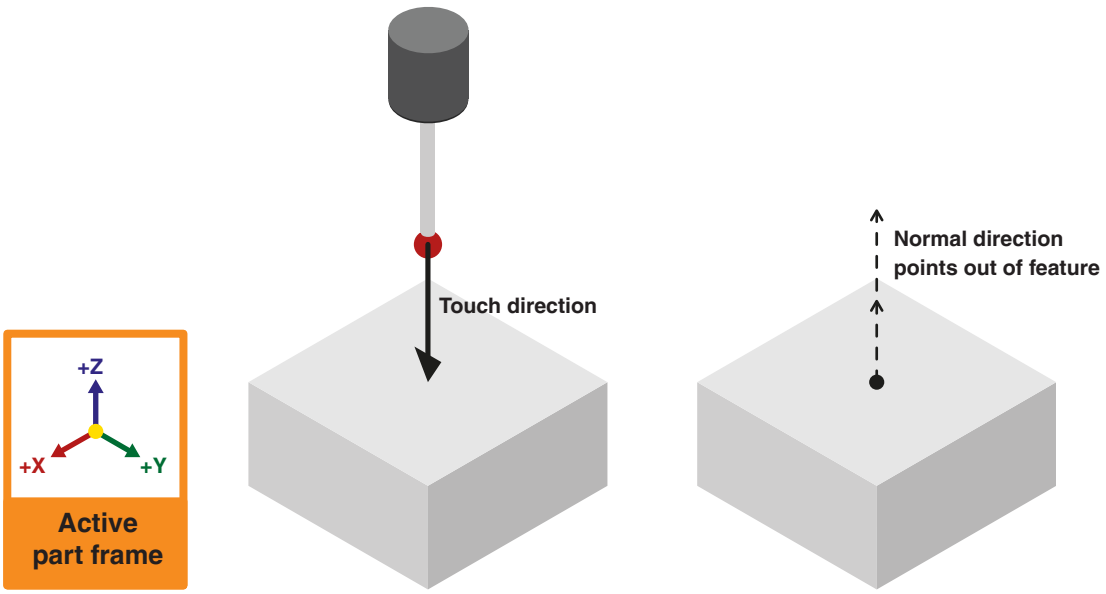


Figure 12 The normal direction of a surface.

Touch moves

The catalogue of touch macros enables you to collect position data from various types of surfaces. This section describes the macros that you can use to collect touch point data.

Touch macros instruct the robot to move the active tool in a specified touch direction until it detects a trigger from the active probe, or until the robot has moved the maximum search distance. After the probe triggers, the robot backs off from the surface and sends its position data to the Server. The Server stores the position data in the surface ID you provide.

A surface ID can store unlimited position data. The Server assigns an index to each touch point stored in a surface ID, starting in collection order from zero. For three touch points saved in a surface ID, the first measured touch point has index 0, the second index 1, and the third index 2.

Some illustrated examples in this user guide show touch points. Example touch points are labelled with the letter ‘P’, and differentiated by surface ID and index. For example, touch point P3.4 is stored in surface ID 3 and has an index of 4.

After collecting position data, you can reference the surface IDs in the calculation you intend to conduct (see “Calculations”, on page **33**).

When collecting a touch point, the robot either takes a single or double touch, depending on which behaviour you enable (see “Single touch”, on page **18** and “Double touch”, on page **19**).

Some touch move macros collect one touch point. Others collect multiple touch points during a single command.

Touch in an active frame direction

These macros instruct the robot to take one touch point by moving in a specified direction within the active part frame or tool frame. The robot can move:

- In the positive or negative direction along an axis (X, Y, or Z) within the given frame type.
- A direction which combines an X, Y, and Z component within the given frame type (see “Defining directions”, on page **20**).

Figure 13 shows a robot with a part-setting probe mounted on its flange. The robot’s active tool frame is the tip of the part setter stylus, and the robot’s base is the active part frame.

In this example, the robot moves in the +Y direction of the active frame. The +Y direction in the active tool frame is different to the +Y direction in the active part frame. So the robot will move in a different direction, depending on which frame type the macro tells it to move in.

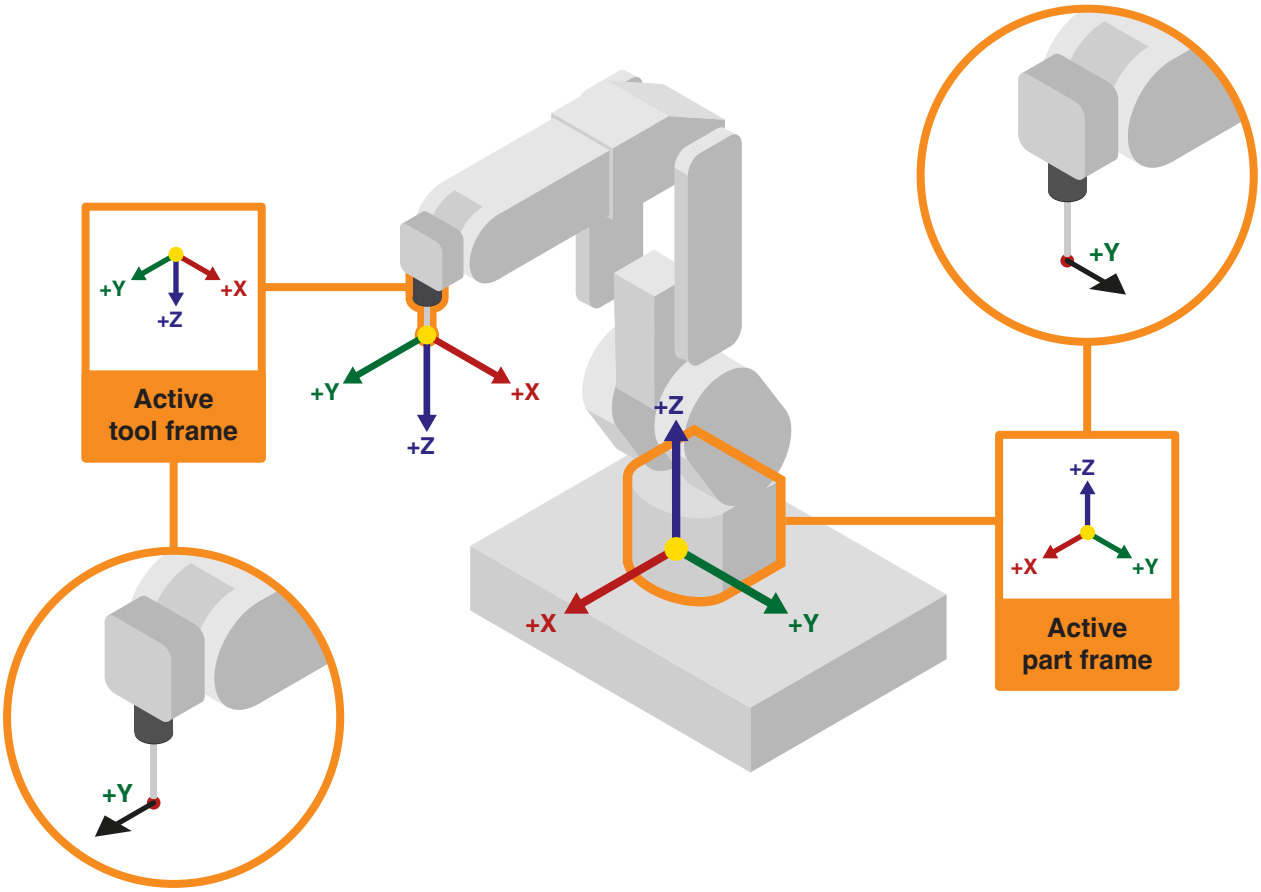


Figure 13 A robot moves a part setter in the +Y direction of the active tool frame and the active part frame.

Touch a target point within the active part frame

This macro instructs the robot to probe a nominal position, which you define, on a surface.

You must define a touch direction (see “Defining directions”, on page **20**) and approach distance. The robot calculates an approach position (“Approach position”, on page **65**) using these values.

The robot moves from its current position to the approach position, then moves in the touch direction within the active part frame to collect the touch point.

This is useful when programming touch points using CAD files, because the CAD software assigns a point position and surface ID which you can enter into the macro.

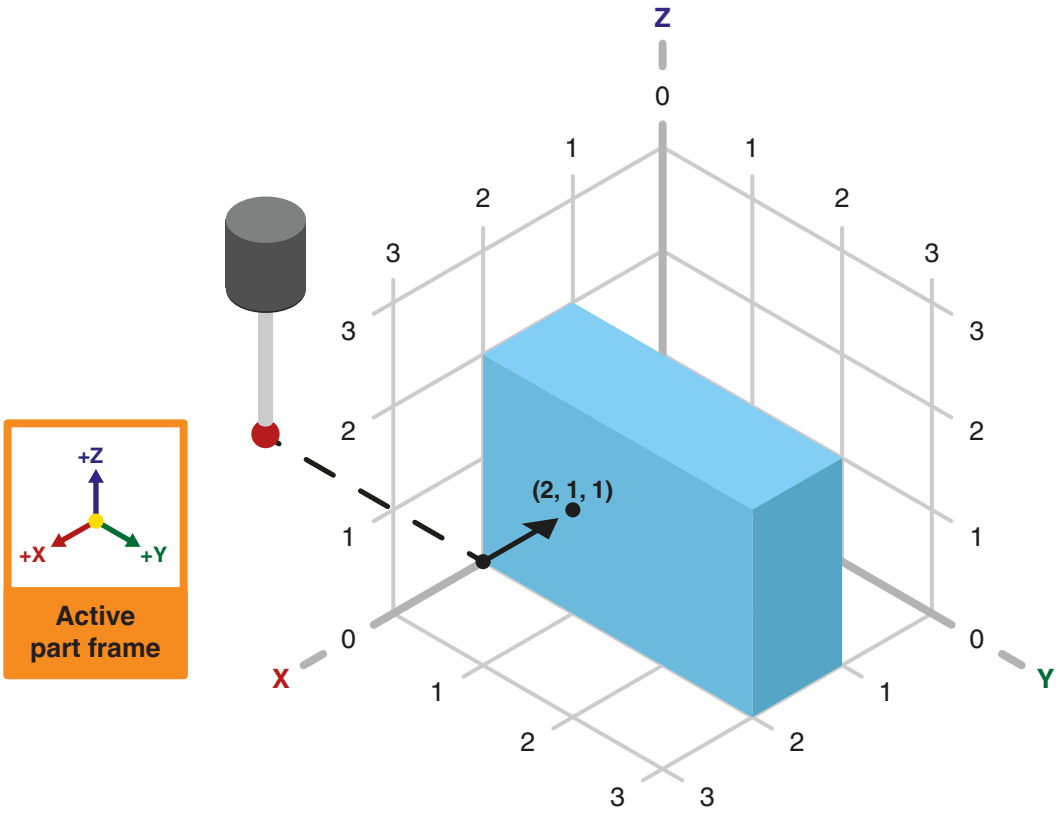


Figure 14 A part setter taking a touch point at a user-defined position (2, 1, 1). The robot moves the part setter to an approach position, which it calculates based on a touch direction of –X, and an approach distance of 1.

Touch circles

Use one of two macros to tell the robot to probe either an internal (bore/hole) or external (boss) circle by taking four equally-spaced touch points.

The robot starts moving from its current position relative to the active part frame, in the direction that you set, to take the first touch point.

The robot uses the diameter you supply in the macro to calculate the centre of a circular path from the first touch point. It moves around this path, stopping at calculated approach positions to collect the other touch points. The circle's normal direction, relative to the active part frame, determines the direction that the robot travels along its path. Use the right-hand rule for axial vectors (see “Axial vector rule”, on page **70**) to work out the normal direction needed to make the robot move in the direction you want.

After taking all four touch points, the robot backs off from the fourth touch point. It does not return to the start position.

Use one touch circle macro to evaluate a 2D circle in alignments or best fit calculations. To evaluate cylinders, use two touch circle macros at different positions along the cylinder axis so that the Server can identify its direction. See “Evaluate features”, on page **33**.

NOTE: The Server requires three touch points to evaluate a circle. Touch circle macros take four touch points to allow for up to one anomalous touch point.

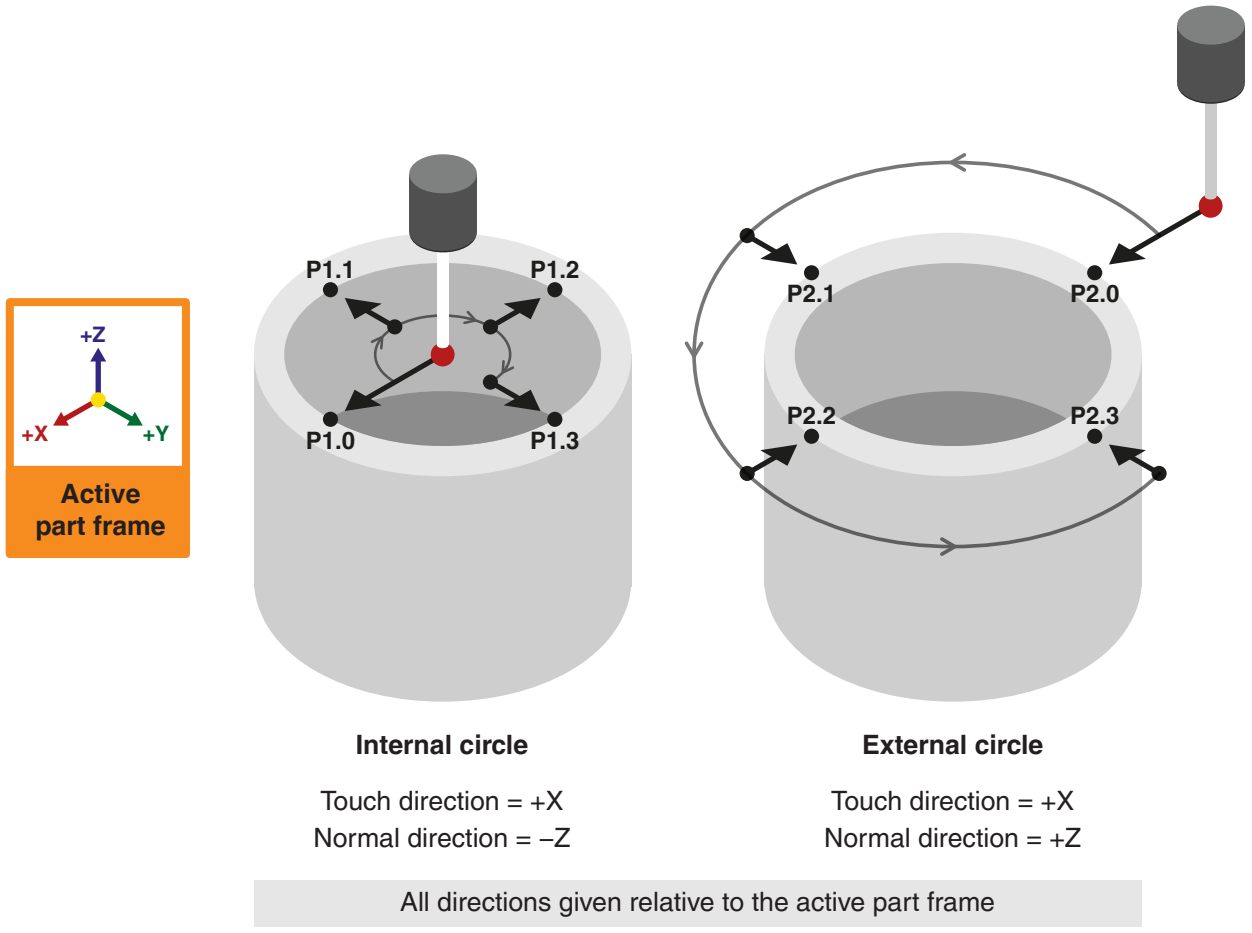


Figure 15 A part setter collecting touch points for two circles (P1.0–P1.3, and P2.0–P2.3).

Touch sphere

This macro tells the robot to probe a sphere by taking five touch points (P1.0, P1.1, P1.2, P1.3, P1.4).

Before calling this macro:

- Set a nominal frame for the active probe where the Z axis aligns with the stylus stem. For a part setter, this is a tool frame. For a tool setter, this is a part frame.
- Move your robot so that the probe points towards the centre of the sphere.

The robot starts moving from its current position to take the first touch point (P1.0). If the active probe is a part setter, the robot moves in the +Z direction of the active tool frame. If the active probe is a tool setter, the robot moves in the –Z direction of the active part frame.

The robot considers P1.0 as the sphere 'pole'. For the remaining touches, the robot uses the pole and diameter to find the centre of a circular path around the equator of the sphere. It moves around this path, stopping at calculated approach positions to collect the other touch points.

After taking all five touch points, the robot returns to the pole at the approach distance away from P1.0.

NOTE: The Server requires four touch points to evaluate a circle. Touch sphere macros take five touch points to allow for up to one anomalous touch point.

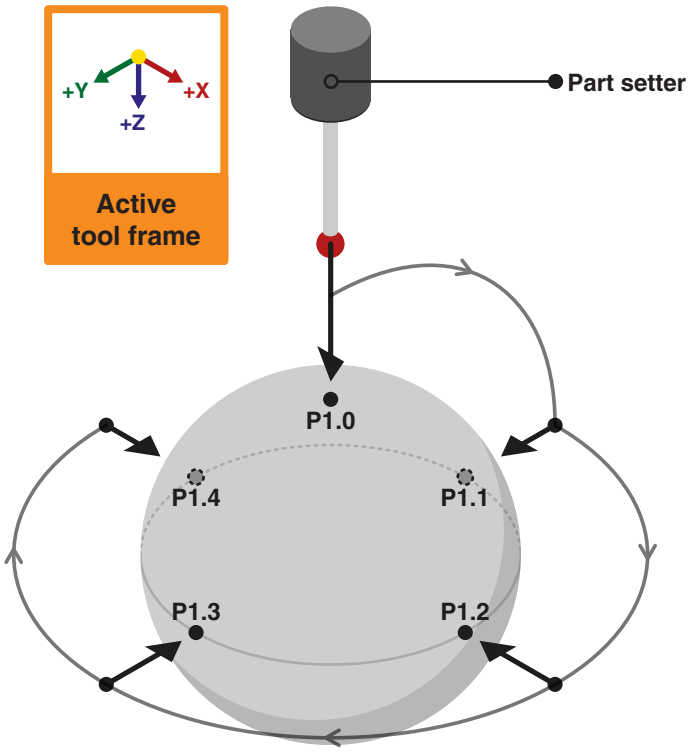


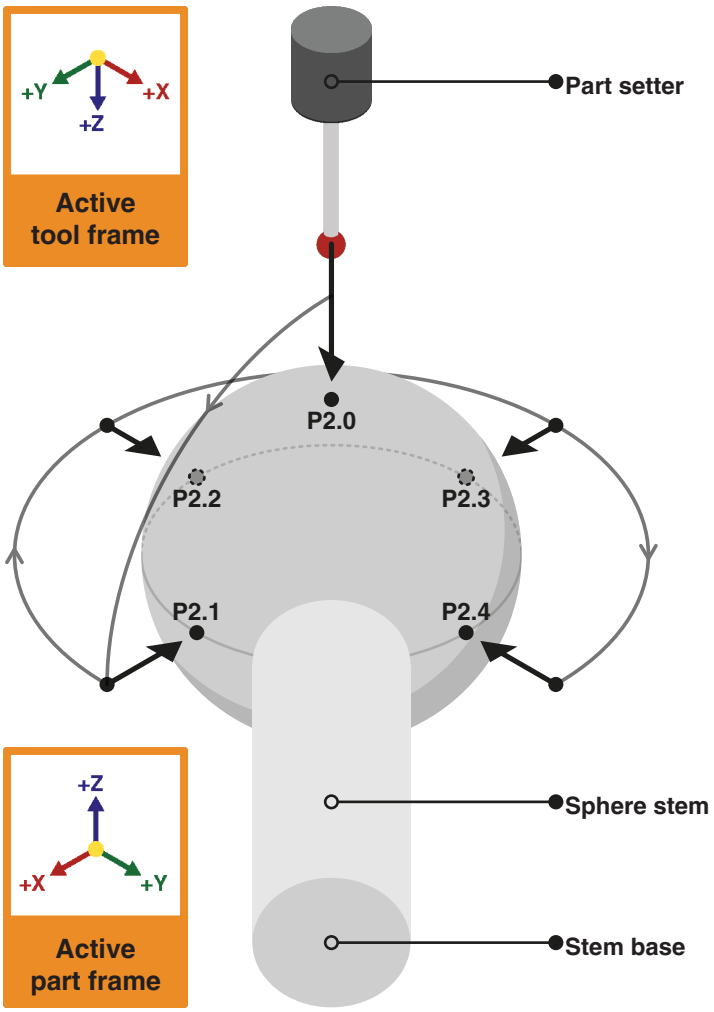
Figure 16 Running a touch sphere macro with an activated part setter.

Touch sphere with stem avoidance

This variant of the touch sphere macro tells the robot to avoid hitting the sphere's supporting structure (stem) whilst collecting touch points. This is useful for calibrating probes, where you need to probe the spherical surface of a datum ball at multiple orientations.

Before calling this macro:

- Set a nominal frame for the active probe where the Z axis aligns with the stylus stem. For a part setter, this is a tool frame. For a tool setter, this is a part frame.
- Move your robot so that the probe points towards the centre of the sphere.
- Define the sphere stem direction. See "Record the sphere stem direction", on page **31** and "Set the sphere stem direction", on page **32**.



Active probe = Part setter
Touch direction = +Z in the active tool frame
Stem direction = –X–Y in the active part frame

Figure 17 Running a touch sphere macro with an active part-setting probe, whilst avoiding the sphere stem.

Record the sphere stem direction

This macro records directions in the active frames to define the stem direction.

Before calling this macro:

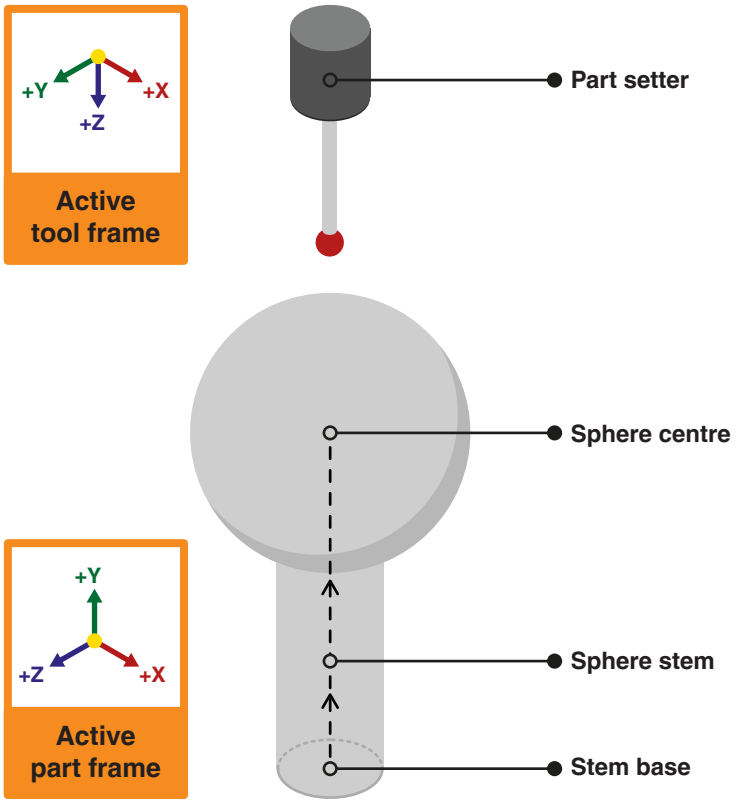
- Set a nominal frame for the active probe where the Z axis aligns with the stylus stem. For a part setter, this is a tool frame. For a tool setter, this is a part frame.
- Move your robot so that the probe points through the sphere centre and towards the stem base.

When called, this macro records the –Z direction of the:

- Active tool frame if the part setter is active.
- Active part frame if the tool setter is active.

After recording the –Z direction, the macro sets the equivalent direction in the other active frame as the stem direction.

For the example in **Figure 18**, the part setter is the active probe. Running this macro records the –Z direction of the active tool frame. The equivalent direction in the active part frame is the +Y direction, which the macro sets as the stem direction.



Active probe = Part setter
Stem direction = +Y in the active part frame

Figure 18 Recording the sphere stem direction using the active part setter's tool frame.

Set the sphere stem direction

This macro sets the stem direction with a unit vector (see “Defining directions”, on page **20**).

The unit vector defines the stem direction within the:

- Active part frame if the part setter active.
- Active tool frame if the tool setter is active.

For the example in **Figure 19**, to set the stem direction as the –Z direction in the active tool frame:

- The active probe is the tool setter.
- The unit vector given to the macro is (0, 0, –1).

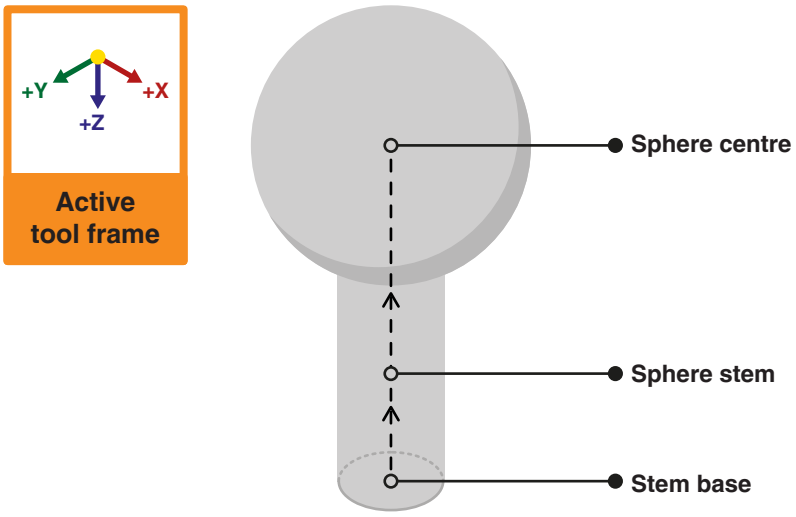


Figure 19 Setting the sphere stem direction using a vector relative to the active tool frame.

Calculations

After collecting position data, run calculation macros to analyse it for your application. This section describes the calculations that you can do with RCS P-series.

Evaluate features

A best fit is the 'best' model to represent a set of data.

After collecting touch points on a surface, the Server can best fit a feature. This uses the collected data to create a feature model for the best approximation of the surface's position and orientation in space.

Figure 20 shows a line of best fit to represent a set of data points on a graph. The Server uses the same principle when calculating the best fit of a feature to represent the position data collected from a surface.

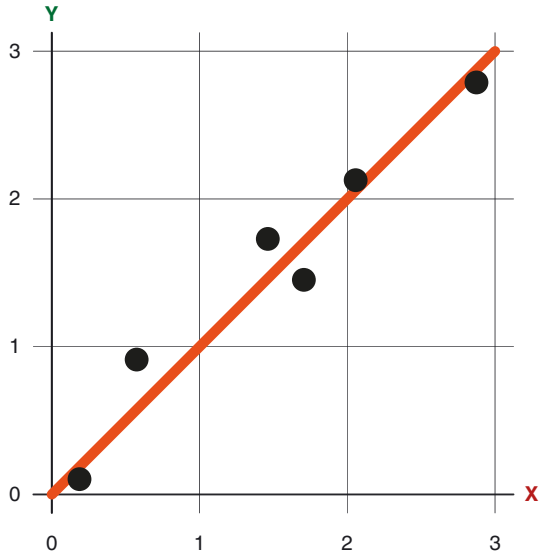


Figure 20 A line of best fit to represent a collection of data points..

The Server can calculate best fit models of the following feature types: planes, circles, spheres, cylinders, and cones. **Figure 21** shows how the Server could use the same four touch points to calculate a best fit model of two different features, depending on which macro you call in your robot program.

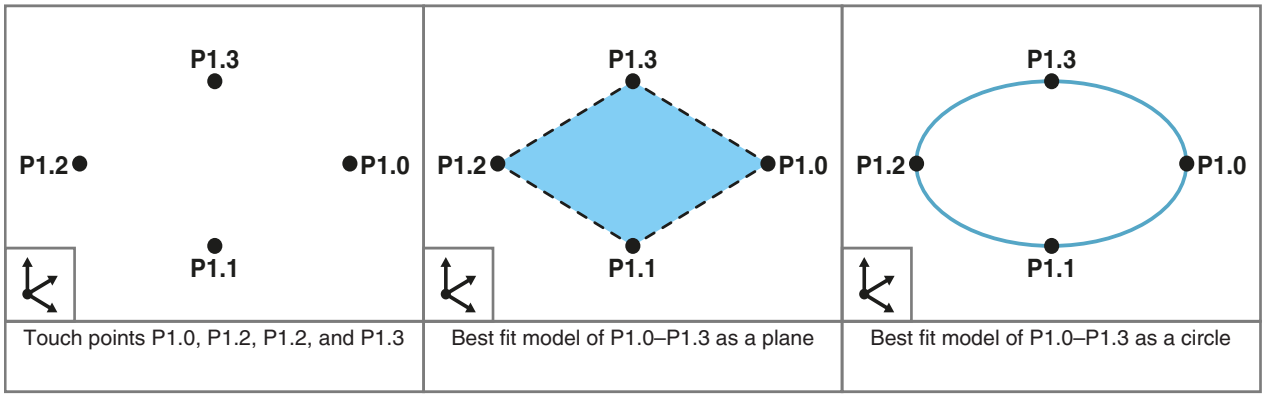


Figure 21 Shows how the same four touch points can be used to create a best fit model of a plane or a circle.

The Server calculates best fits to understand feature characteristics when completing the following analyses:

- Probe calibration, where the Server best fits a sphere using datum ball position data to calculate the stylus TCP. See “Calibrate your probe”, on page **36**.
- Tool and part frame alignments, where the Server best fits any feature used in the alignment. See “Align tool and part frames”, on page **42** and “Spindle tool calibration”, on page **61**.

For calculation purposes, the Server considers a feature to be either a point, line, or plane.

Circles and spheres are considered as points. They have a position (or centre) and diameter (Ø). To best fit a circle, the probe needs to take at least three touch points for the Server to calculate the circle's centre (see “Touch circles”, on page **28**). To best fit a sphere, the probe needs to take at least four touch points for the Server to calculate the sphere's centre (see “Touch sphere”, on page **29**).

Cylinders and cones are considered as lines. They have a position, axis direction, and diameter. To best fit a cylinder or cone, the Server needs to connect the centres of two circles along the same axis to define the axis direction.

A plane has a position and a normal direction. To best fit a plane, the probe needs to collect at least three touch points on a flat surface for the Server to calculate the normal direction. The touch points need to spread as far around the plane as possible, and not be in a straight line, to best fit the most accurate model of the plane.

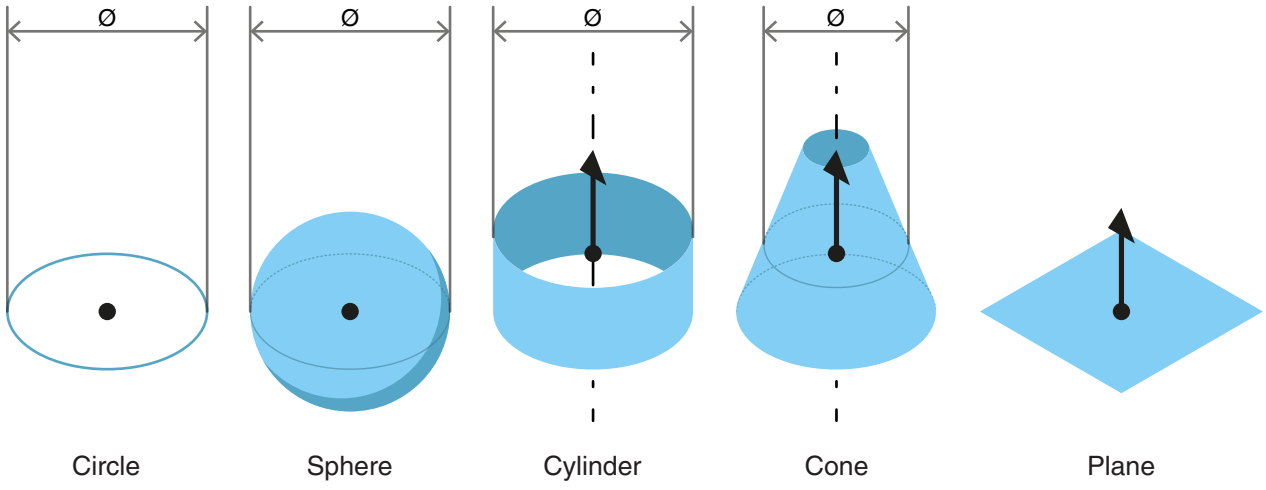


Figure 22 Shows the position and, where applicable, the direction and diameter of different feature models.

NOTE: Since cones have a variable diameter, the Server outputs a calculated diameter at the midpoint along the cone axis between the two recorded circles.

If you have an RCS advanced probing licence, you can calculate best fit feature models separately from an alignment calculation. The Server saves best fit results in Feature IDs, so that you can reference them in other calculations. This enables you to analyse the feature's relationship to other features, or co-ordinate systems. See “Analyse feature relationships”, on page **57** and “Baseline and recovery”, on page **55**.

Define nominal features

RCS P-series includes macros which enable you to define the following nominal features:

- Internal or external sphere, which needs user-defined values for a position and diameter.
- Plane, which needs user-defined values for a position, direction and size (width or length).
- Internal or external cylinder, which needs user-defined values for a position, direction, diameter, and length.

You can use the nominal features to compare their relationship with evaluated features (see “Analyse feature relationships”, on page **57**). This may be useful to determine if a feature is located in a desired position, for example.

The Server saves the nominal data to Feature IDs for future reference.

NOTE: This function requires the RCS advanced probing licence.

Calibrate your probe

To collect accurate data with a probe, you must calibrate it. This determines the position of the probe stylus tip's centre and apparent diameter.

Use touch move macros to collect data for probe calibration. After collecting position data, call the appropriate macro for your chosen calibration method.

To calibrate a part setter, mount a datum ball to a fixed surface in your robot's work envelope, ready to take touch points on it with the part setter (**Figure 23**).

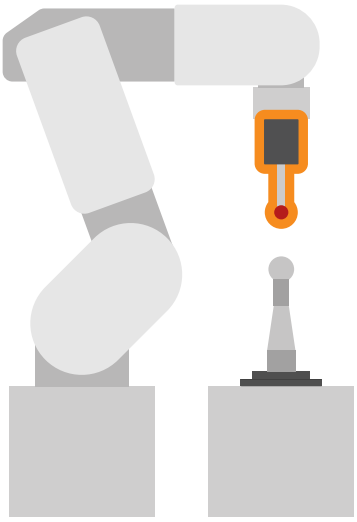


Figure 23 Set-up for part setter calibration.

To calibrate a tool setter, use one of the following methods:

- Mount a datum ball on your robot arm, ready to take touch points on it with the tool setter (**Figure 24**).
- Calibrate a robot-mounted part setter first, ready to take touch points on it with the tool setter (**Figure 25**).

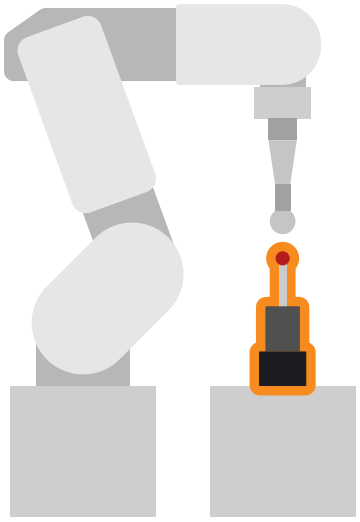


Figure 24 Set-up for tool setter calibration with a datum ball.

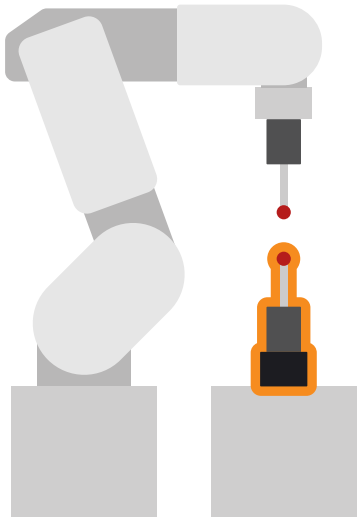


Figure 25 Set-up for tool setter calibration with a calibrated part setter.

Mount your datum ball

Mount the datum ball kit (**Figure 26**) as required for the probe calibration method you are going to do.

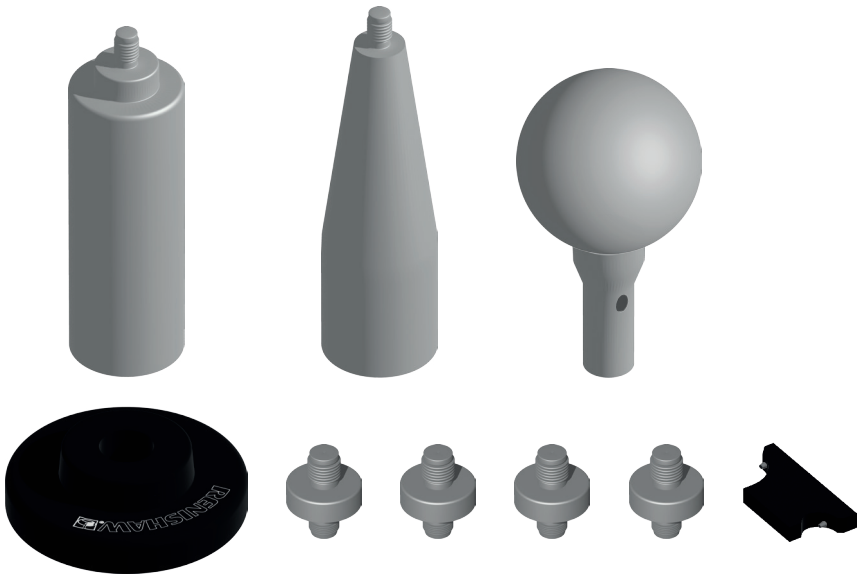
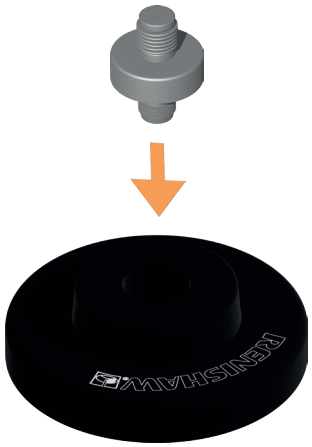


Figure 26 Datum ball kit components, featuring extensions and adaptors to tailor the kit to your cell set-up .

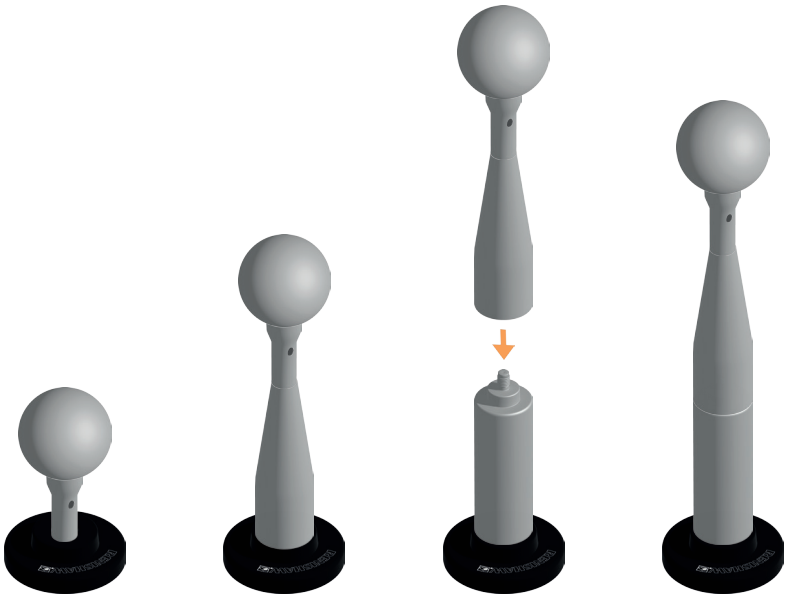
For part setter calibration



1. Choose the appropriate sized thread adaptor for your set-up. Screw the datum ball base mount into a threaded hole on a fixed surface within your robot's work envelope.



2. Use the supplied spanner to tighten the thread adaptor.



3. The datum ball kit includes extensions to adjust the pillar height. There must be enough space for the robot-mounted part setter to take touch points around the whole datum ball. Attach extensions as required to enable this.

For tool setter calibration

Use a thread adaptor to attach the datum sphere pillar to a threaded hole on the robot end-effector. Use the pillar extensions if needed to adjust the datum sphere's offset from the robot.

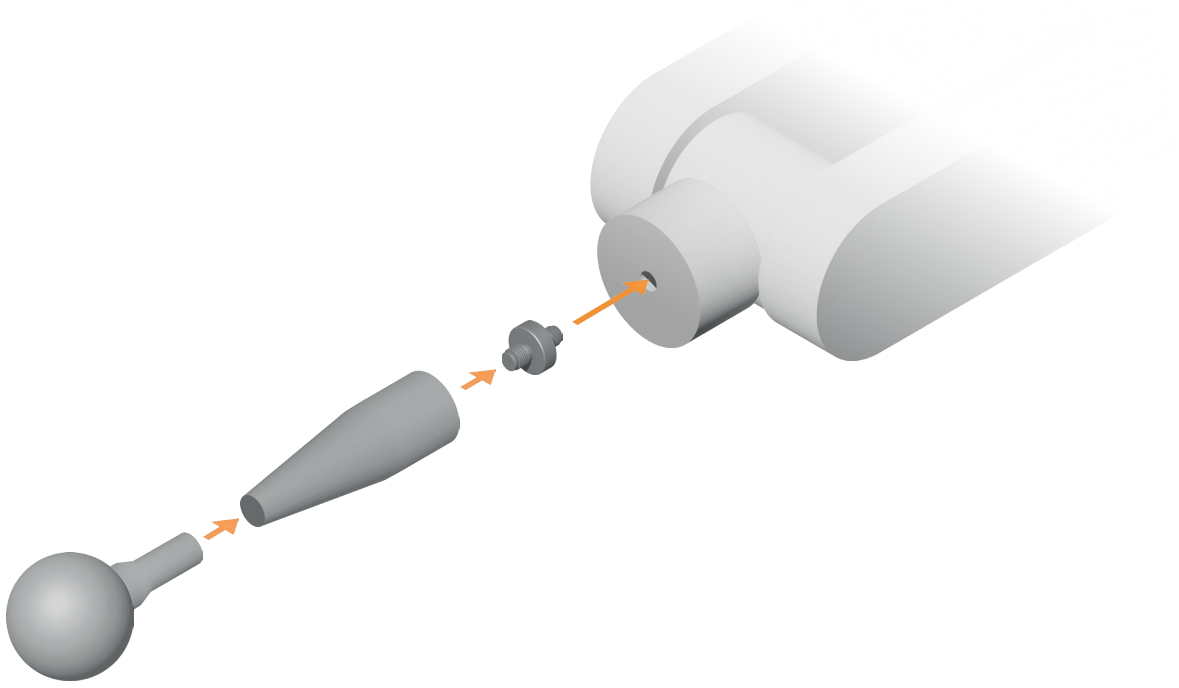


Figure 27 The datum ball assembly attaches to the robot arm via the thread adaptor.

NOTE: If using a calibrated part setter to calibrate a tool setter, you do not need to attach a datum sphere to your robot arm.

Calibrate a probe with a datum ball

Use the “Touch sphere with stem avoidance” macro (see page **30**) to collect touch points on the datum ball. Call the macro five times, from different start positions and orientations. This collects 25 total touch points, which is the recommended number of points required for an optimal calibration. From the first start position, each subsequent start position should be ±45°–90° around the datum ball. For all start positions, the stylus tip must point towards the datum ball centre.

For part setter calibration, change start positions by rotating and translating the robot-mounted probe around the fixed datum ball (**Figure 28**).

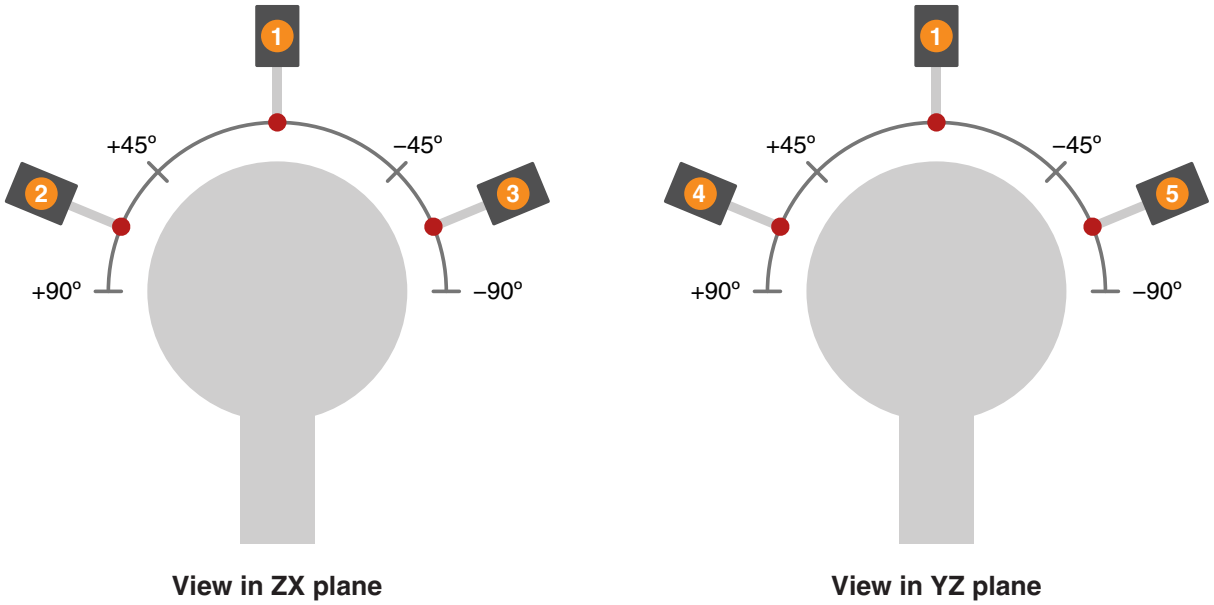


Figure 28 Example of five different start positions around a datum ball for calibrating a part setter.

For tool setter calibration, change start positions by rotating the robot-mounted datum ball in place. The datum ball does not move in translation (**Figure 29**).

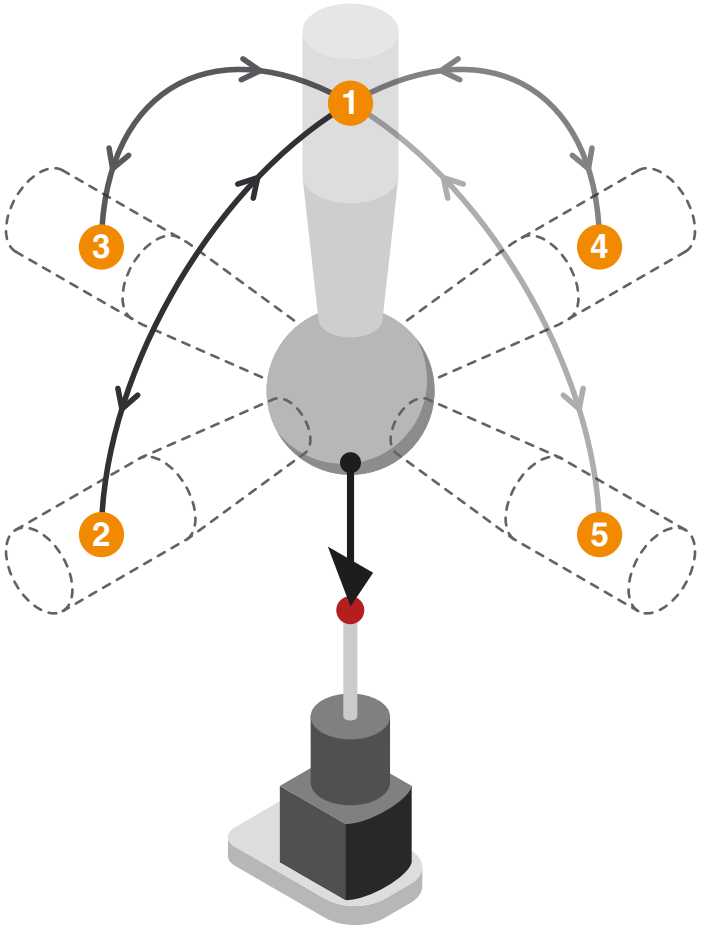


Figure 29 Example of five different start positions of a datum ball when calibrating a tool setter.

Calibrate a tool-setting probe with a calibrated part setter

First, calibrate the part setter using a datum ball.

Take five touch points on the part setter from varying start positions. From the first start position, each subsequent start position should be ±45°–90° around the tool setter stylus tip.

The part setter's stylus must always point towards the tool setter's tip centre, and be at a non-zero angle relative to the tool setter stylus axis. Use macros to touch in the +Z direction of the part setter's tool frame (see "Touch in an active frame direction", on page **26**). This is the highest force direction of the part setter. Taking touches in this direction forces the tool setter to trigger before the part setter, to prevent an incorrect calibration result. **Figure 30** shows an example of five different start positions around the tool setter tip.

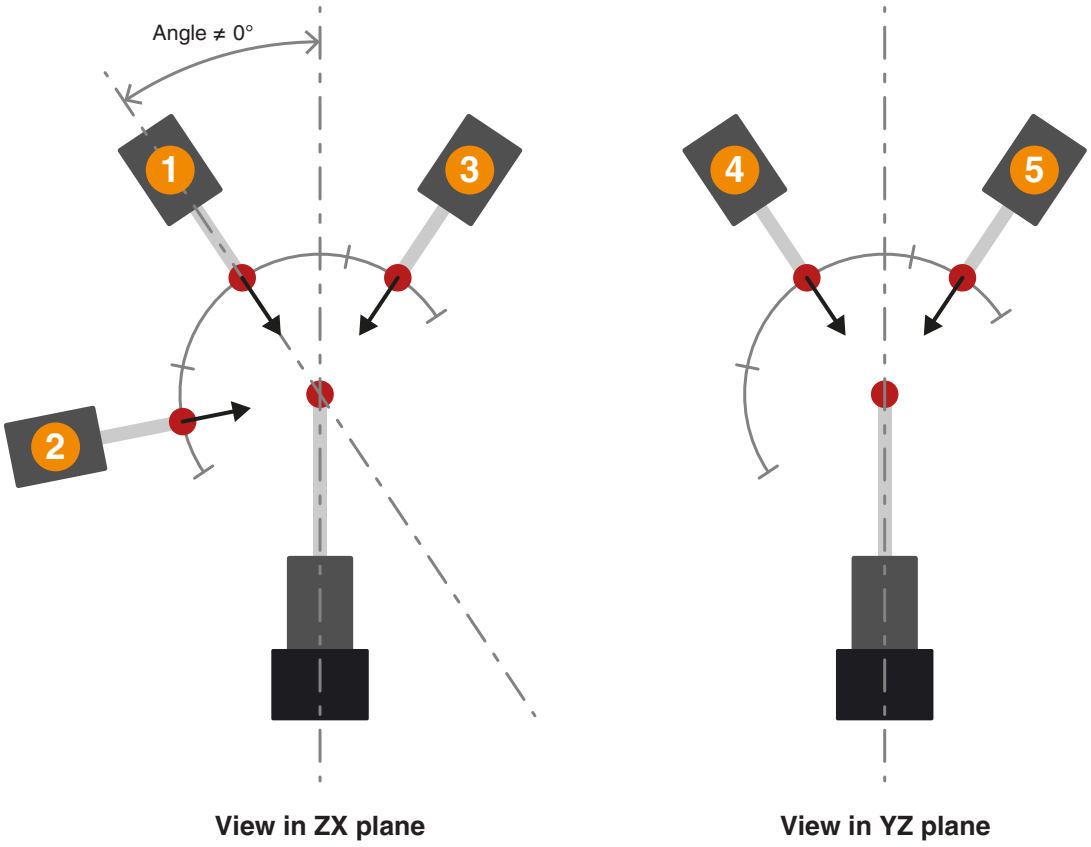


Figure 30 Example of five different start positions of a calibrated part setter when calibrating a tool setter.

For SCARA robots

Calibrate all SCARA (Selective Compliance Assembly Robot Arm) probes using a datum ball.

1. Position the robot so that the part or tool-setting probe stylus points towards the centre of the datum ball.
2. Call the “Touch sphere with stem avoidance” macro (see page **30**).
3. Rotate ±90°–180° about the Z axis.
4. Call the “Touch sphere with stem avoidance” macro (see page **30**).
5. Run the SCARA robot part or tool-setting probe calibration macro.

Align tool and part frames

RCS P-series uses feature position data to set tool and part frames in alignment macros. There are various types of alignment you can do. Each alignment type has a macro to align a tool frame, and another to align a part frame.

Use touch move macros to probe surfaces and assign the position data to surface IDs. Reference these surface IDs in the alignment macros.

The Server best fits reference features when calculating a frame. It adjusts its calculations based on the calibrated diameter of the active probes stylus tip. You must calibrate the active probe before conducting an alignment.

Simple alignments

Simple alignments use three or fewer features to lock up to six degrees of freedom. Features can be a plane, line (cylinders, or two touch points on the same surface), or a point (spheres, circles, or one touch point on a surface). See “Evaluate features”, on page **33**.

In the macro arguments, surface IDs are referenced in a given order. Editing the order in which the surface IDs are referenced in the macro affects the frame result.

Write simple alignment programs in your teach pendant or programming software.

Alternatively, use the Probing Alignment session in RCS Software Suite to create template programs. These contain all macro instructions needed to conduct the chosen alignment. You need to edit the template to add robot movements and positions to probe the features before running the program.

3-2-1

A 3-2-1 alignment sets the origins and directions of the X, Y and Z axes. It needs three flat surfaces (planes) which are nominally orthogonal to each other.

The first plane needs three touch points (P1.0, P1.1, P1.2).

The second plane needs two touch points (P2.0, P2.1) to create a line.

The third plane needs one touch point (P3.0).

Figure 31 shows how the Server calculates the frame.

The first plane sets the Z origin and direction. The Z origin lies on the plane. The +Z direction is perpendicular to the plane, pointing out of the feature (see “Normal directions”, on page **24**).

The second plane sets the X origin and Y direction. The X origin lies on the plane. The line from P2.0 to P2.1 sets the +Y direction perpendicular to the Z direction.

The third plane sets the Y origin on the plane.

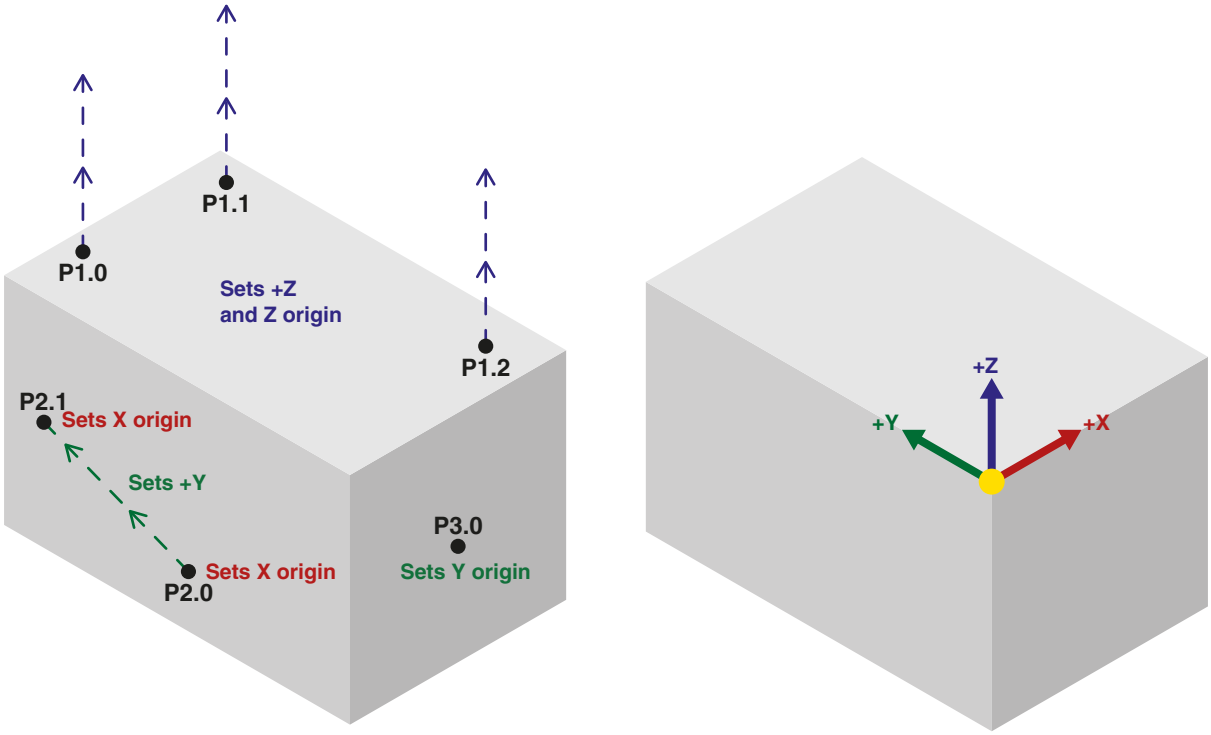


Figure 31 Shows how the Server uses three orthogonal planes to create a plane, line, and point when aligning a new frame.

Plane-plane-plane

A plane-plane-plane alignment sets the origins and directions of the X, Y and Z axes. It needs three flat surfaces (planes) which are nominally orthogonal to each other.

All planes need three touch points.

Figure 32 shows how the Server calculates the frame result.

The first plane (P1.0, P1.1, P1.2) sets the Z origin and direction. The Z origin lies on the plane. The +Z direction is perpendicular to the plane, pointing out of the feature (see “Normal directions”, on page **24**).

The second plane (P2.0, P2.1, P2.2) sets the X origin and direction. The X origin lies on the plane. The +X direction is perpendicular to the plane.

The Server best fits the third plane (P3.0, P3.1, P3.2) to be orthogonal to the first two planes to set the Y origin.

The Server follows the right-hand rule for cross products to calculate the +Y direction, using the Z and X directions. See “Cross product rule”, on page **69**.

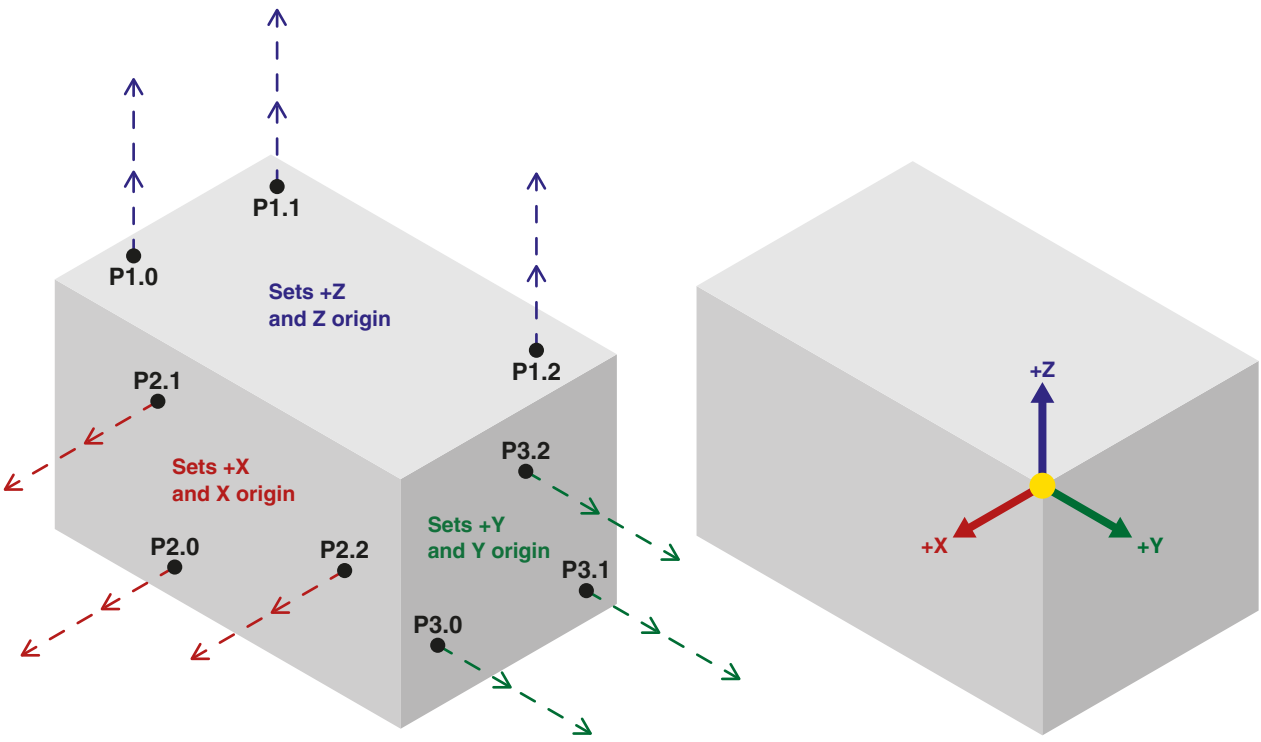


Figure 32 Shows how the Server uses three orthogonal planes to align a new frame.

Plane-circle-circle

A plane-circle-circle alignment sets the origins and directions of the X, Y and Z axes.

This alignment needs the following features:

1. A plane, where the probe has taken three touch points (P1.0, P1.1, P1.2).
2. Two circular surfaces, such as holes, which are normal to the plane (see “Normal directions”, on page **24**).

Figure 33 shows how the Server calculates the frame.

The first surface is the plane, which sets the Z origin and direction. The Z origin lies on the plane. The +Z direction is perpendicular to the plane, pointing out of the feature (see “Normal directions”, on page **24**).

The second surface is the first circle. Its centre (C2) sets the X and Y origins.

The third surface is the second circle. The line from C2 to the centre of the second circle (C3) sets the +Y direction.

The Server follows the right-hand rule for cross products to calculate the +X direction, using the Y and Z directions. See “Cross product rule”, on page **69**.

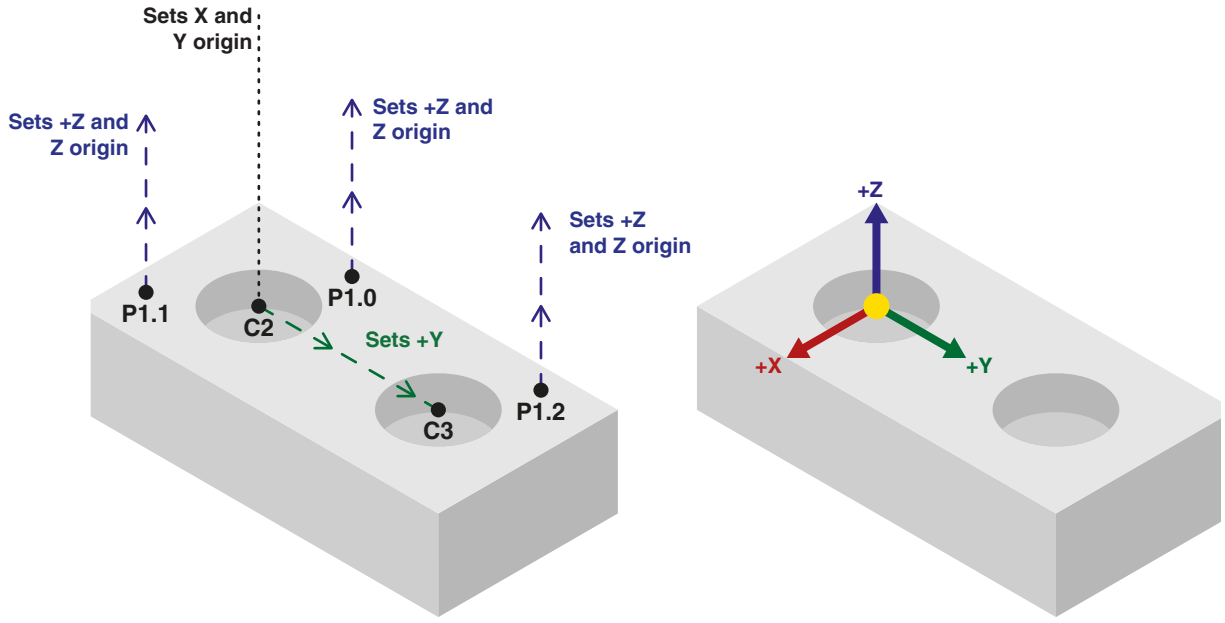


Figure 33 Shows how the Server uses two circles which are normal to a plane to align a new frame.

Plane-plane-point

A plane-plane-point alignment sets the origins and directions of the X, Y and Z axes. It creates the frame at the intersection of two planes. It can be useful for welding applications.

This alignment needs the following features:

1. Two planes which intersect each other, and where the probe has taken at least three touch points on both planes.

2. A third plane, which is nominally orthogonal to the intersecting line of the first two planes, where the probe has taken one touch point.

Figure 34 shows how the Server calculates the frame.

The first surface is Plane 1, the second surface is Plane 2.

The X axis lies along the intersecting line between Planes 1 and 2. The right-hand rule for axial vectors determines the +X direction, where your curled fingers point into Plane 2, through the feature, towards Plane 1. Your thumb points in the +X direction. To reverse the +X direction, swap the order of surface IDs for Plane 1 and Plane 2 in the macro. See “Axial vector rule”, on page **70**.

The intersection of the planes also sets the Y and Z origins.

The +Z direction is orthogonal to the intersection, pointing out of the feature at a half angle between Plane 1 and Plane 2.

The point (P3.0) on the third plane sets the X origin.

The Server follows the right-hand rule for cross products to calculate the +Y direction, using the Z and X directions. See “Cross product rule”, on page **69**.

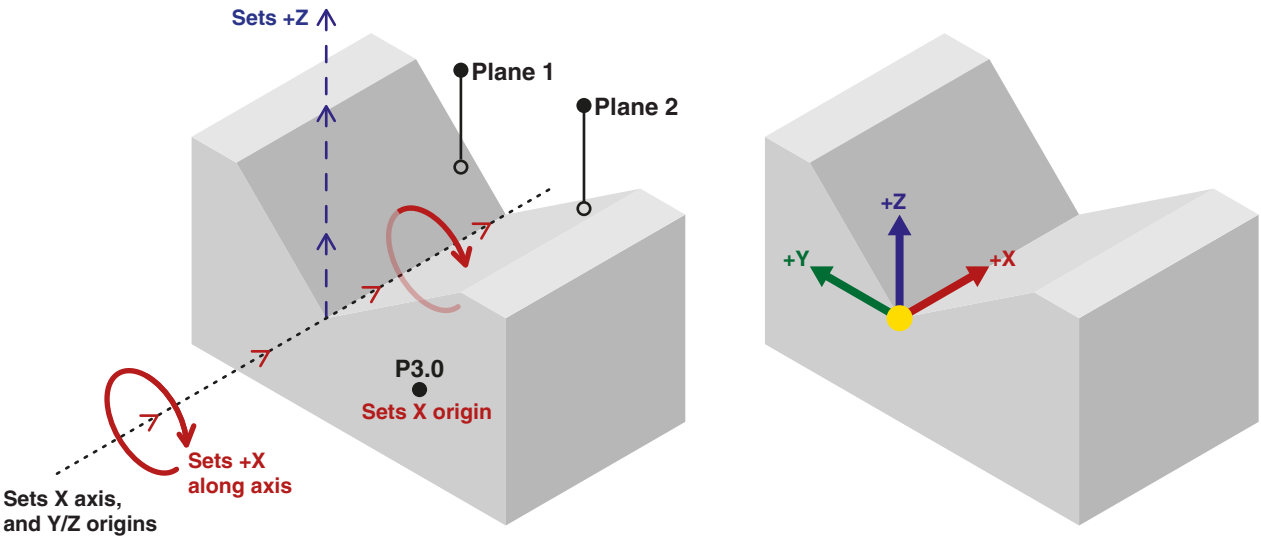


Figure 34 Shows how the Server uses two intersecting planes, and a point, to align a new frame.

Sphere-sphere-sphere

A sphere-sphere-sphere alignment sets the origins and directions of the X, Y and Z axes.

This alignment needs three spherical surfaces, such as datum balls.

There must be space around each sphere for the robot to take at least the five touch points of a sphere touch macro, avoiding the sphere stems if needed (see “Touch sphere with stem avoidance”, on page **30**).

Figure 35 shows how the Server calculates the frame result.

The centre of the first sphere (S1) sets the X, Y, and Z axis origins.

The line (L1) from S1 to the centre of the second sphere (S2) sets the +Y direction.

The line (L2) is calculated from the centre of the third sphere (S3) to a perpendicular intersection with L1. This sets the +X direction.

The Server follows the right-hand rule for cross products to calculate the +Z direction, using the X and Y directions. See “Cross product rule”, on page **69**.

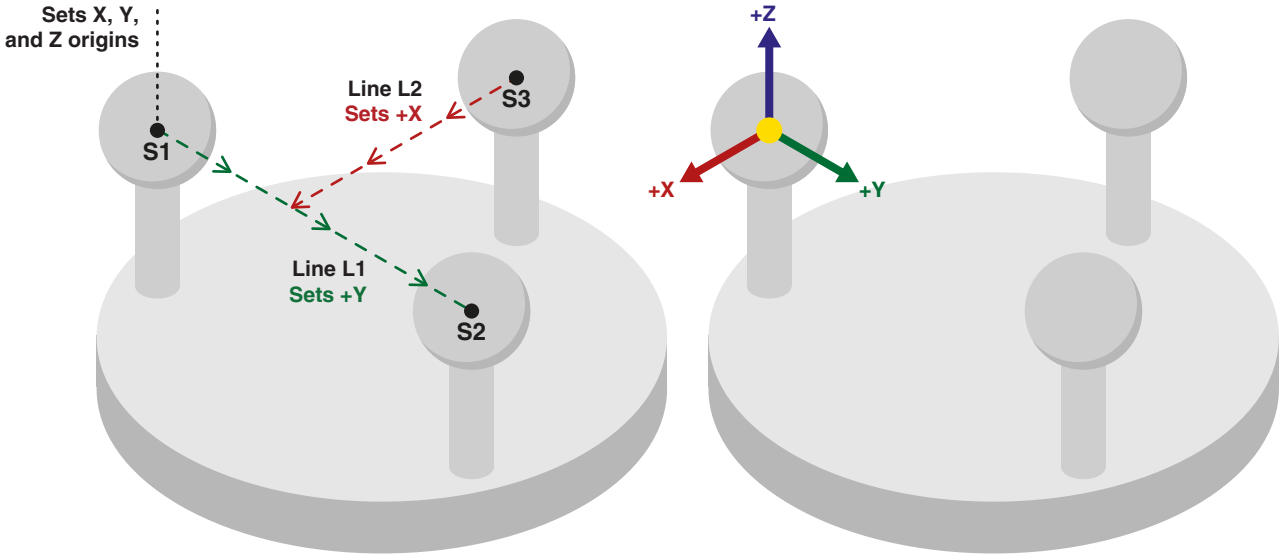


Figure 35 Shows how the Server uses the centres of three spheres to align a new frame.

Plane-circle

A plane-circle alignment updates a frame to set the positive or negative direction and origin of the axis you choose. It also sets the origins of the remaining axes. It does not constrain the rotation about the remaining axes.

This alignment needs:

1. A plane, where the probe has taken three touch points (P1.0, P1.1, P1.2).

2. A circular surface, such as a hole, which is normal to the plane (see “Normal directions”, on page **24**).

Figure 36 shows how the Server calculates the frame result when updating the +X direction.

The first surface is the plane, which sets the chosen axis origin and direction. In this example, the X origin lies on the plane. The +X direction is perpendicular to the plane, pointing out of the feature.

The second surface is the circle. Its centre (C2) sets the origins of the remaining axes. In this example, the Y and Z origins.

The existing co-ordinate system rotates to set the new +X direction. The Y and Z directions update as a result of this rotation.

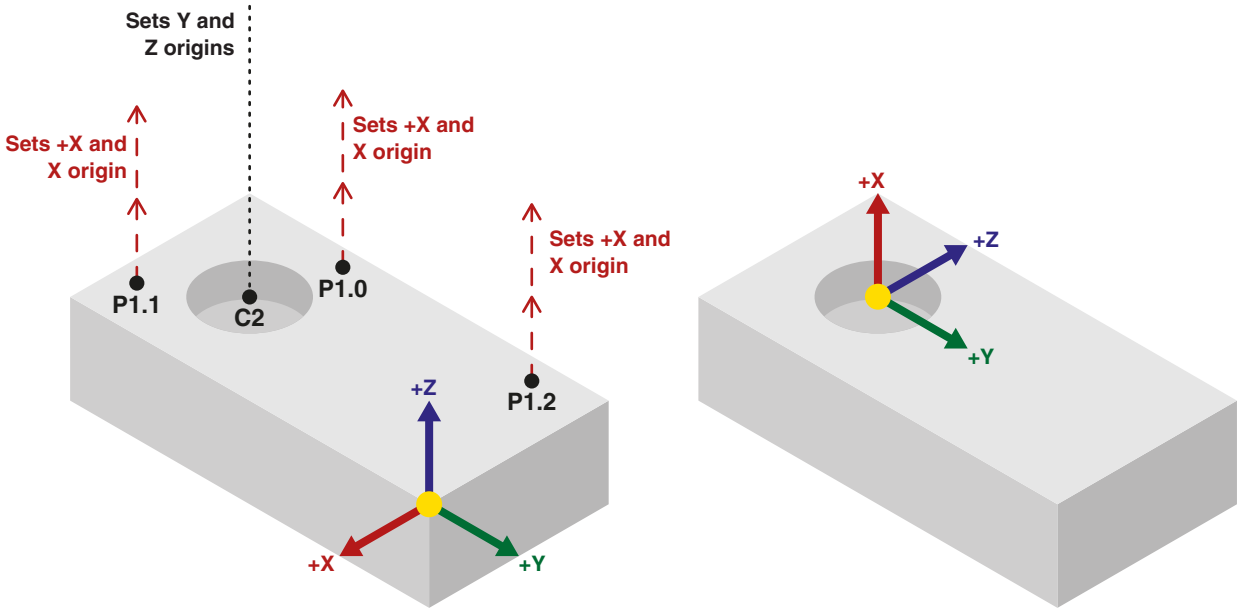


Figure 36 Shows how the Server updates a frame in a plane-circle alignment, when using the plane normal to change the +X direction. The Y and Z directions of the frame rotate with the X axis to set the new +X direction.

Cylinder-plane

A cylinder-plane alignment updates a frame to set the origin and positive or negative direction of the axis you choose. It also sets the origins of the remaining axes. It does not constrain the rotation about the remaining axes.

This alignment needs:

1. A cylinder (C1–C2), which is normal to a plane (see “Normal directions”, on page **24**).

2. A plane, where the probe has taken at least one touch point.

Figure 37 shows how the Server calculates the frame result when updating the –Y direction.

The first surface is the cylinder, which sets the chosen axis direction. The Server calculates a best fit line through the centres of the measured circles along the cylinder to set the orientation of the Y axis. The vector from the first recorded touch point (P1.0) to the last recorded touch point (P1.7) sets the new –Y direction along the axis.

The best fit line also sets the origins of the remaining axes. In this example, the Z and X origins.

The second surface is Plane 1. It sets the origin of the chosen axis, Y, along the cylinder.

The existing co-ordinate system rotates to set the new –Y direction. The Z and X directions update as a result of this rotation.

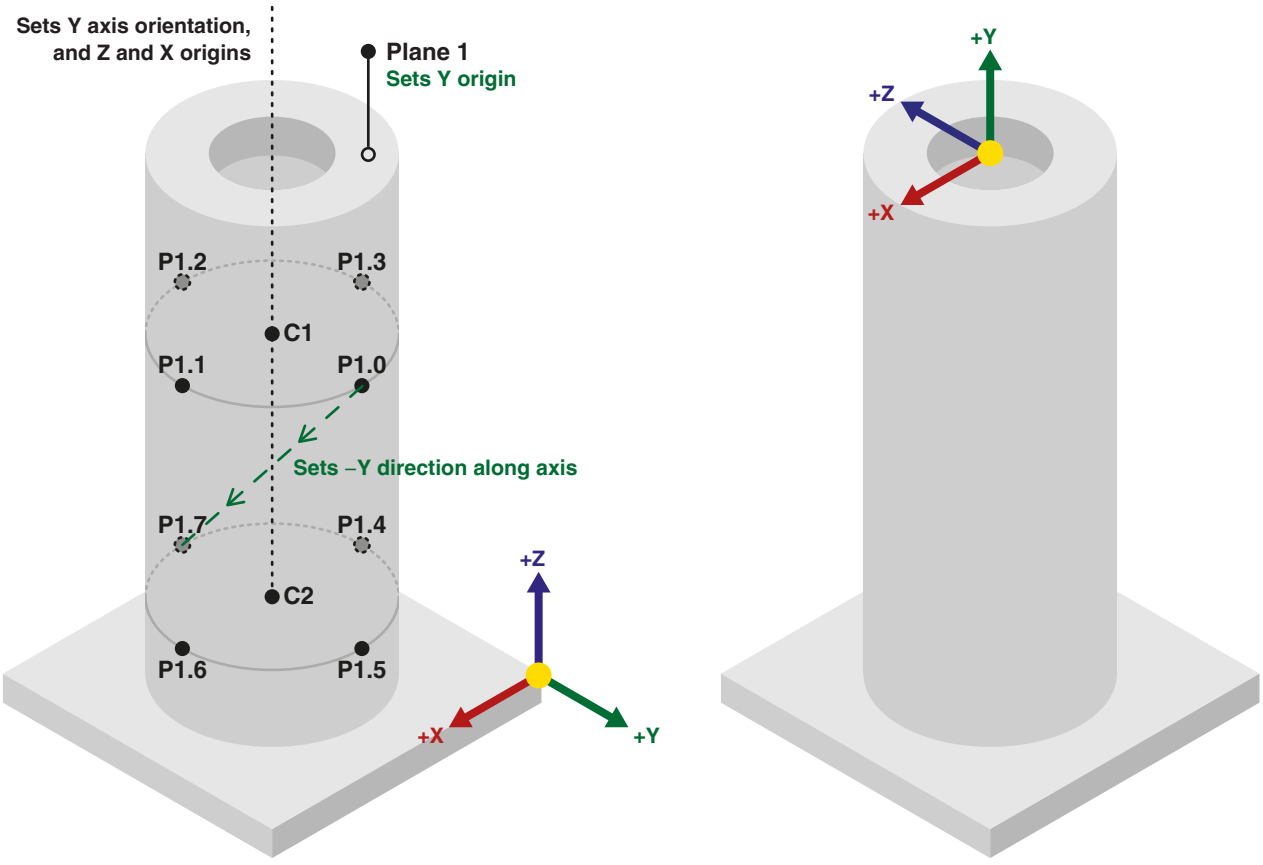


Figure 37 Shows how a frame updates in a cylinder-plane alignment, when using the cylinder to change the –Y direction. The X and Z directions of the frame rotate with the Y axis to set the new –Y direction.

Plane

A plane alignment updates a frame to set the origin and positive or negative direction of the axis you choose.

This alignment needs a plane, where the probe has taken at least three touch points (P1.0, P1.1, P1.2).

Figure 38 shows how the Server calculates the frame result when updating the Z origin and positive direction.

The plane sets the chosen axis origin and direction. In this example, the Z origin lies on the plane. The +Z direction is perpendicular to the plane, pointing out of the feature.

The existing co-ordinate system rotates to set the new +Z direction. The X and Y directions update as a result of this rotation. The X and Y origins do not change.

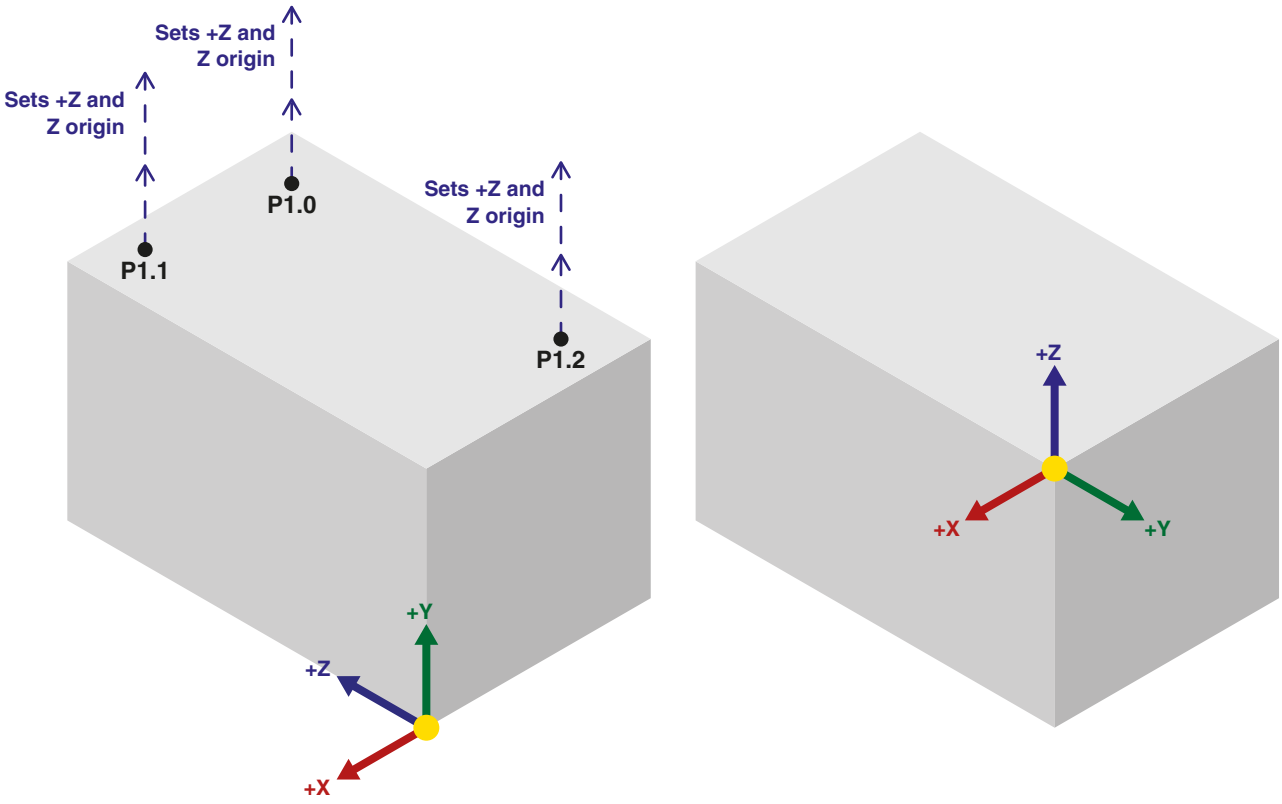


Figure 38 Shows how the Server uses a plane to set the Z origin and positive direction when updating a frame. The X and Y directions of the frame rotate with the Z axis to set the new +Z direction.

Cylinder

A cylinder alignment updates a frame to set the positive or negative direction of the axis you choose. It also sets the origins of the X, Y and Z axes.

This alignment needs a cylinder (C1–C2). **Figure 39** shows how the Server calculates the frame result when updating the –X direction.

The Server calculates a best fit line through the centres of the measured circles along the cylinder to set the orientation of the X axis. The vector from the first recorded touch point (P1.0) to the last recorded touch point (P1.7) sets the new –X direction along the axis of the fitted cylinder.

The existing co-ordinate system rotates to set the new –X direction. The Y and Z directions update as a result of this rotation.

The X, Y and Z origins lie at the midpoint of the best fit line through the circles.

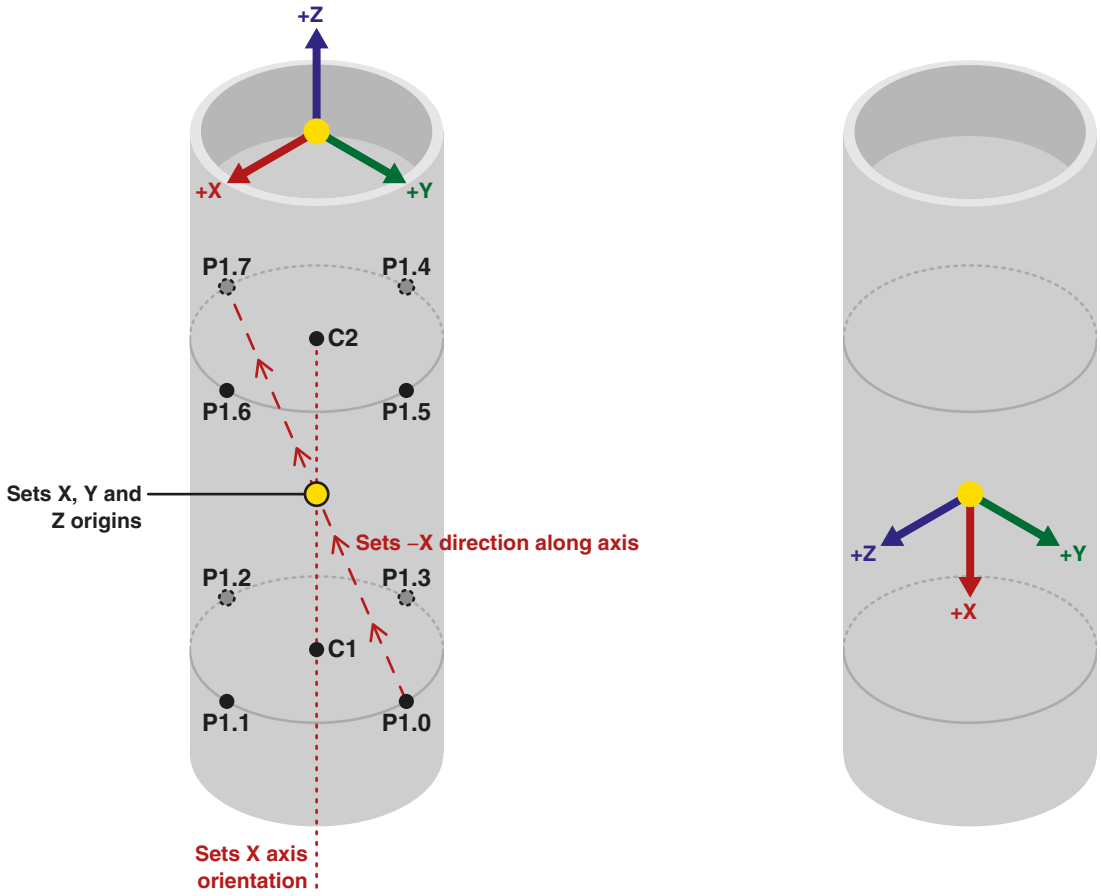


Figure 39 Shows how the Server uses a cylinder to set the X origin and negative direction when updating a frame. The Y and Z directions of the frame rotate with the X axis to set the new –X direction.

Sphere

A sphere alignment updates a frame to set the origins of the X, Y and Z axes.

This alignment needs a spherical surface. **Figure 40** shows how the Server calculates the frame result.

The Server calculates the centre of the sphere (S1). The existing co-ordinate system moves to set S1 as the origin for the X, Y, and Z axes. The co-ordinate system does not rotate, and so there is no change to the X, Y, or Z directions.

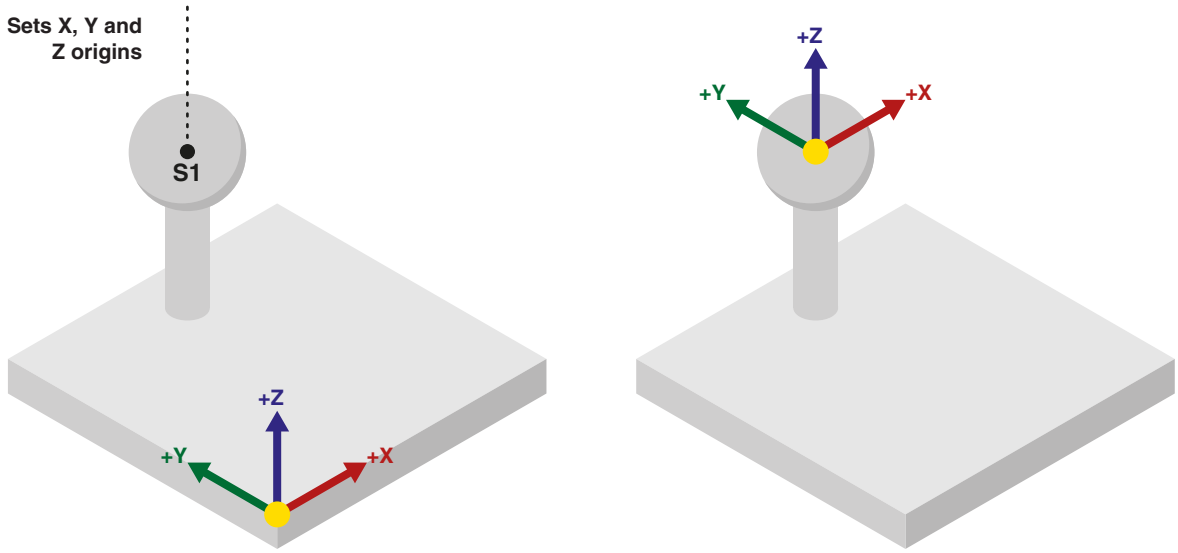


Figure 40 Shows how the Server uses the centre of a sphere to set a new X, Y, and Z origin when updating a frame. The X, Y and Z origins move, but their directions do not change.

Complex alignments

Complex alignments lock up to six degrees of freedom, and have one or more of the following characteristics:

- An alignment using constructed features which are based on real features. For example, two measured planes extended to construct a line where they would theoretically intersect in space.
- An alignment which has a datum offset from real features (see “Datum”, on page **66**).
- An alignment based on a freeform shape.

NOTE: This function requires the RCS advanced probing licence.

Complex alignments require a datum configuration file. This stores the position and orientation of nominal features relative to the co-ordinate system that is being located, which enables the Server to calculate the alignment after probing the real features.

In a datum configuration file, surface IDs can have first, second or third priority. The priority affects how the Server calculates the frame result. If multiple features have the same priority, the Server averages the error of both features and fits the alignment to minimise these errors.

Use the Probing Alignment session in RCS Software Suite to create a datum configuration file and program template for the complex alignment. It supports using CAD models or nominal features for the alignment. Generated program templates contain all macro instructions needed to conduct the chosen alignment. You need to edit the template to add robot movements and positions to probe the features before running the program.

Alternatively, create a datum configuration file within a robot program as part of a baseline and recovery process. See “Baseline and recovery”, on page **55**.

Figure 41 shows an example of a complex alignment on a gripper tool. The alignment is complex because it uses five real features, two constructed features, and the resulting datum is offset from the real features.

The first priority surface is Plane 1. A plane sets an origin and direction. In this example, Plane 1 sets the Z origin and the +Z direction.

The second priority surfaces are Planes 2 and 3. The Server constructs a plane at the midpoint between them, which sets another origin and direction. In this example, the constructed plane sets the X origin and +X direction.

The third priority surfaces are Planes 4 and 5. The Server constructs a plane at the midpoint between them, which sets the final axis origin, Y. The Server follows the right-hand rule for cross products to calculate the +Y direction, using the X and Z directions. See “Cross product rule”, on page **69**.

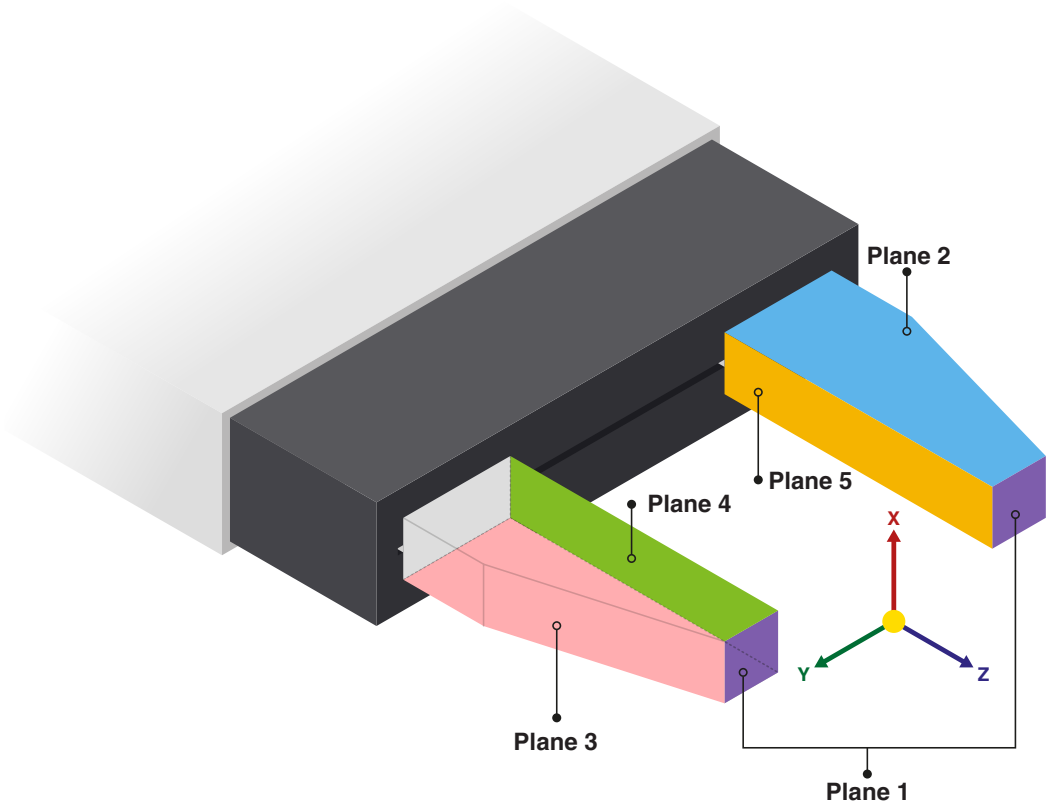


Figure 41 Alignment of a gripper tool frame. It is a complex alignment because it uses five real features, some constructed features based on the real features, and the resulting datum is offset from the real features.

Baseline and recovery

Use macros to record a complex alignment of the active co-ordinate system, so that you can recover it at a later stage. This is useful if relocating your cell, or if a part moves position. It means your robot programs still work, and you avoid spending time reteaching.

NOTE: This function requires the RCS advanced probing licence.

The process involves creating a 'baseline' program and a 'recovery' program.

The baseline program probes the position and orientation of work piece features relative to the active co-ordinate system. The Server stores this information in a datum configuration file.

The recovery program probes the same work piece features again. The Server calculates a co-ordinate system that correlates to the feature positions in the datum configuration file.

The baseline and recovery robot programs need the following structures:

Baseline program

1. Activate co-ordinate systems
2. Activate probe
3. Initialise best fit
4. Collect feature position data
5. Run best fit calculations
6. Create datum configuration file
7. Add feature IDs to datum configuration file
8. Save datum configuration file

Recovery program

1. Activate co-ordinate systems
2. Activate probe
3. Initialise datum builder
4. Collect feature position data
5. Run datum builder calculation

NOTE: In the baseline program, run best fit calculations of all features before adding them to the datum configuration file.

In the recovery program, do not call any best fit macros. The 'Run datum builder' macro best fits the features as part of the alignment calculation.

Check datum step tolerance

When the Server calculates a frame update, it saves the transformation from the original frame to the new frame. This transformation is called a ‘step’. There are two ‘step’ macros, which can either retrieve the step values from the last frame calculation, or assess whether a datum update is within a required tolerance.

The step is a translation and a rotation in space relative to the original frame. To check if a step is within a required tolerance, you provide maximum values that the datum can translate and rotate by. The Server then assesses whether both the length and angle of the step vector is less than these values.

The example in **Figure 42** uses exaggerated values to show how the Server calculates the step where the original frame A_1 updates to A_2.

The datums are defined relative to the World frame (W) with a translational value along each axis, and a rotational value about each axis (Tx, Ty, Tz, Rx, Ry, Rz).

Datum A_1 is (1 mm, 0 mm, 1 mm, 90°, 0°, 0°), and A_2 is (2 mm, 2 mm, 0 mm, 90°, 90°, 0°).

In this case, the step is (1 mm, –1 mm, –2 mm, 0°, 90°, 0°), because the datum transforms from A_1 to A_2 in the following ways:

- Moves by 1 mm along the X axis of A_1
- Moves by –1 mm along the Y axis of A_1
- Moves by –2 mm along the Z axis of A_1
- Rotates 90° about the Y axis of A_1
- Does not rotate about the X or Z axes of A_1

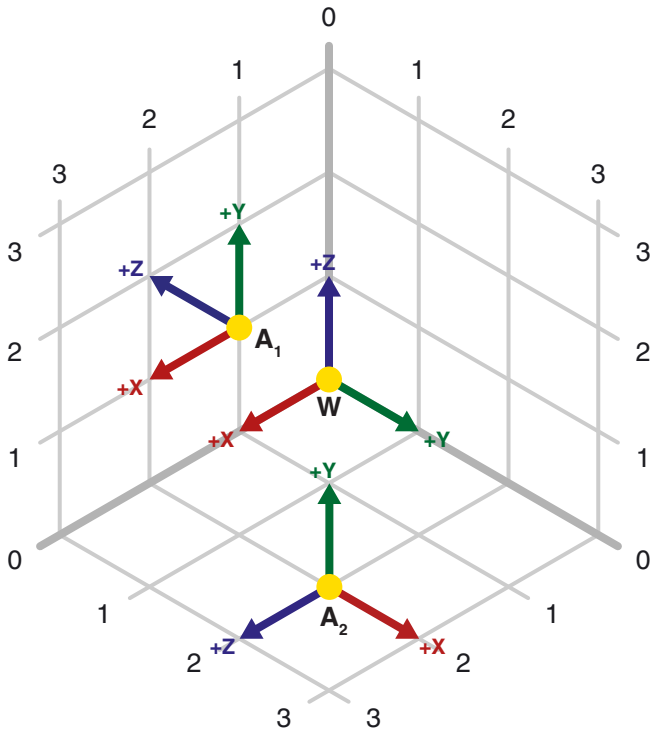


Figure 42 The transformation from datum A_1 to datum A_2.

Analyse feature relationships

Running a best fit calculation allows the Server to understand certain characteristics of a feature, and analyse its relationship to another feature.

See “Evaluate features”, on page **33** for further information.

NOTE: These functions require the RCS advanced probing licence.

Angle between features

The Server can analyse angles between planes, cylinders, and cones.

This calculation uses the features’ unit vectors. For a plane, this is the normal direction. For cylinders and cones, this is the axis direction. The angle result is between 0° and 180°.

The angle (θ) between two planes, or between two lines (cylinders and cones), is the angle between their directions. **Figure 43** and **Figure 44** show how the Server calculates the angle. For planes, the normal direction always points out of the feature. For lines, the axis direction can either point into or out of the feature, depending on the order that the touch points are collected.

The angle (θ) between a plane and a line subtracts the angle between their directions from 90° (see **Figure 45**).

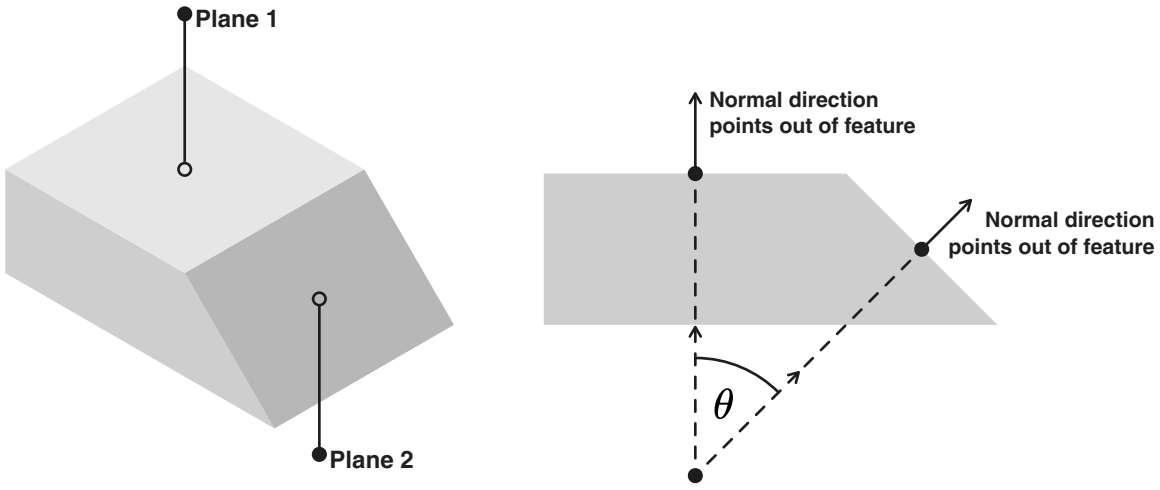


Figure 43 The angle (θ) between two planes is the angle between their normal directions, which is less than or equal to 180°.

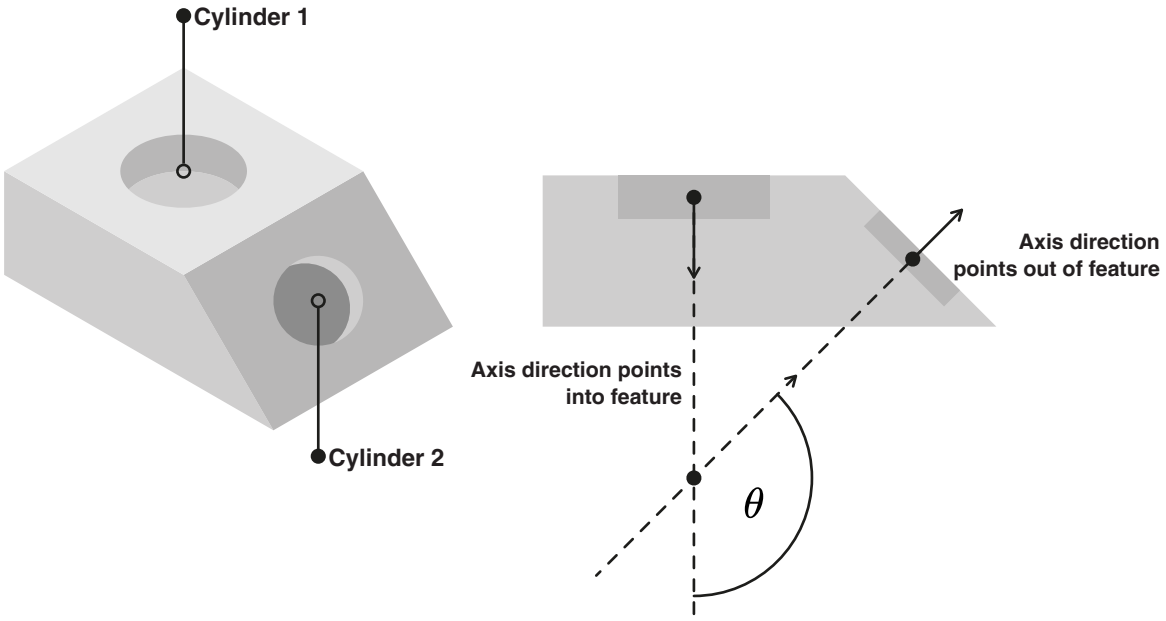


Figure 44 The angle (θ) between two lines (cones or cylinders) is the angle between their axis directions, which is less than or equal to 180°.

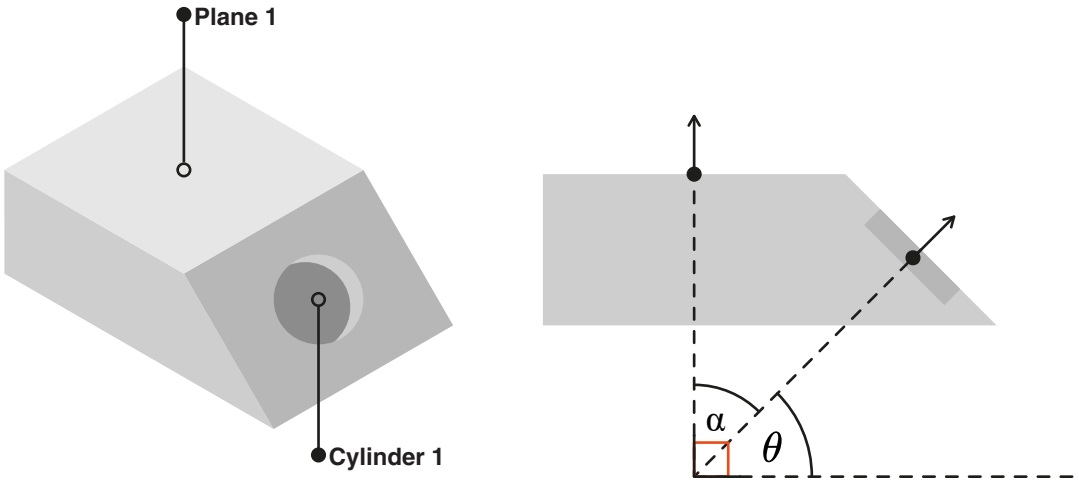


Figure 45 The angle (θ) between a plane and a line is the angle between their directions (α) subtracted from 90°.

Distance between features

The Server can analyse the distance (d) between any two features (plane, line, or point) in millimetres.

The distance between two points is the length of the hypotenuse of the right-angled triangle that can be drawn between the points.

The distances between all other features is calculated where a perpendicular line can be drawn between them. For a cylinder or cone, this line is always perpendicular to its axis. For a plane, this line is always parallel to its normal.

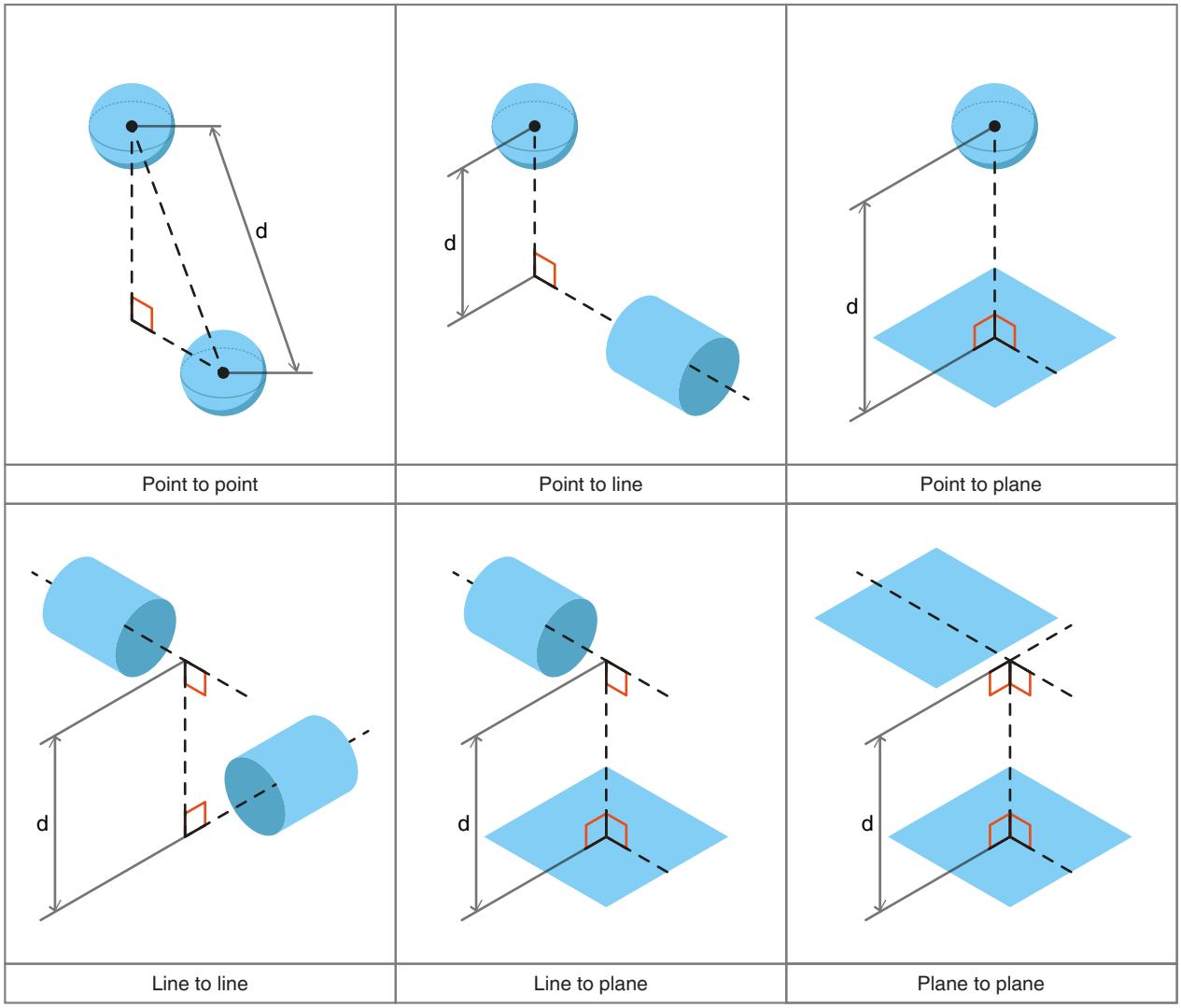


Figure 46 Distances are calculated at the point where two features are perpendicular, apart from when measuring between two points.

Transform between features

The Server can calculate the six-dimensional (6D) transformation to align any two feature types. The result gives the translation and rotation required to align the features. Refer to “Angle between features”, on page **57** and “Distance between features”, on page **59** to understand how the Server calculates these values.

For features which have a direction (planes and lines), the Server can calculate the transformations required to make them either parellel or orthogonal to each other.

You can use the transformation result to manually align the real features to each other.

Signed distance between a touch point and a feature

The Server can calculate the signed (positive or negative) distance between any feature type and a given touch point. In the macro for this calculation, you need to define which surface ID the touch point comes from, and its index within that surface ID (see “Point index”, on page **71**).

A positive distance means the touch point is positive to the feature normal. A negative distance means the touch point is negative to the feature normal.

The example in **Figure 47** shows two different touch points. The touch points are stored in Surface ID 2, and have an index of 0 (P2.0) and 1 (P2.1). The best fitted feature Plane 1 has a normal N1.

P2.0 is positive to N1, so its signed distance from Plane 1 is positive. P2.1 is negative to N1, so its signed distance from Plane 1 is negative.

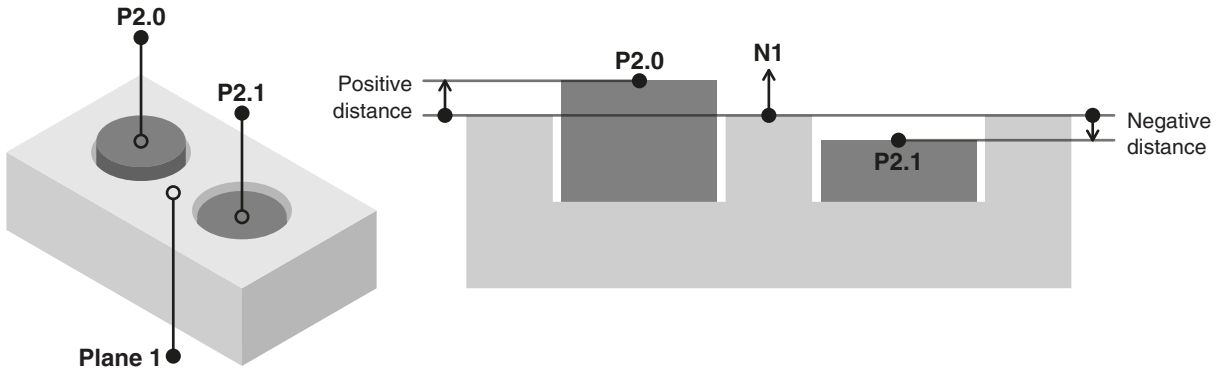


Figure 47 Measuring the signed distance between Plane 1 and two different touch points stored in Surface ID 2.

Spindle tool calibration

For robotic applications which use a cutting tool, you need to set accurate tool vectors and lengths. This ensures the tool follows cutting paths and machines features correctly.

To calibrate a spindle cutting tool, you need the following items:

- RCS tool setter (Renishaw part no. A-6967-6130 or A-6967-6031 or A-6967-6200).
- RCS spindle calibration licence (Renishaw part no. CS-SOF-SW-19-7221).
- RCS spindle calibration kit (Renishaw part no. A-6852-6500 or A-6852-6501 or A-6967-6195), which includes the following components:
  - Disc stylus to use with the tool setter (small or large)
  - Ball-ended spindle calibration stem (small or large)

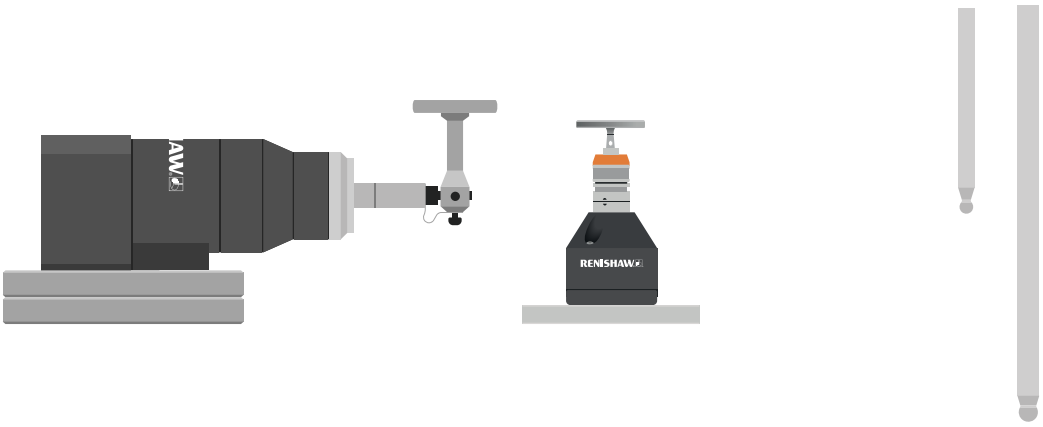
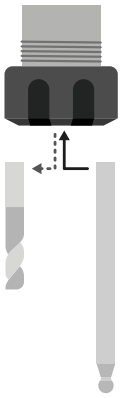
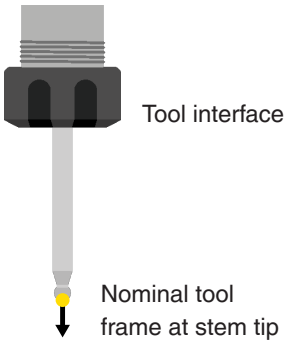


Figure 48 Calibrating a spindle requires a tool setter, disc stylus, and a spindle calibration stem. You need an RCS spindle calibration licence to run the required calculations.

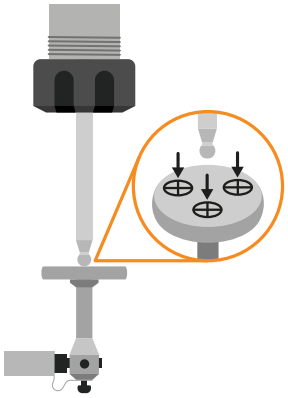
During this process, you will calibrate the spindle calibration stem to locate two tool frames and a part frame. The Server uses this information to calculate the spindle cutting tool TCP and diameter.



1. Swap your robot's spindle cutting tool with the calibration stem.

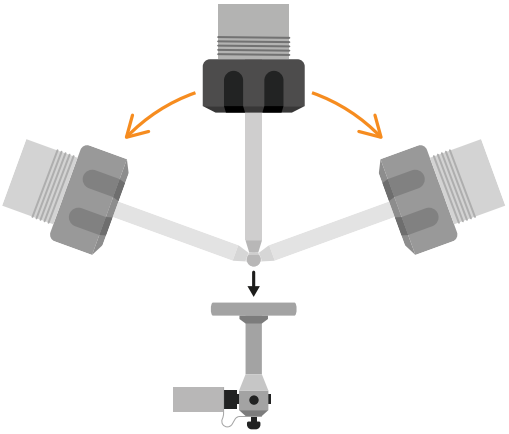


2. The spindle direction points from the tool interface to its tip relative to the active tool frame. Set a nominal tool frame that matches your actual spindle direction.



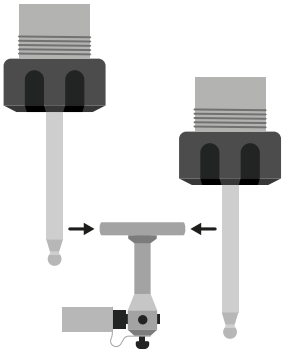
3. Touch the tip of the calibration stem at least three times around the top plane of the disc.

   The Server uses this data to calculate the normal direction of the disc plane.



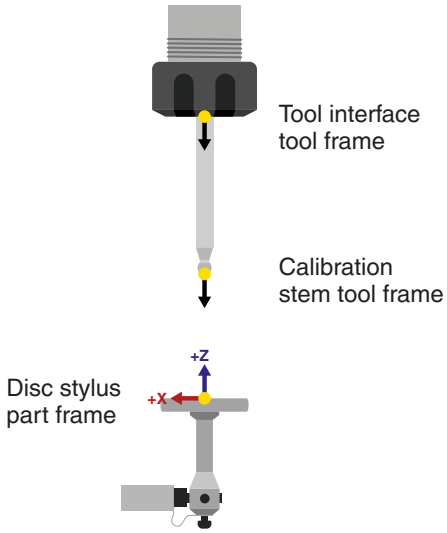
4. Touch the tip of the calibration stem on the top plane of the disc at five different orientations. For each touch, rotate between 45°–90° about the stem's nominal TCP.

   The Server uses this data to calculate the positions of the calibration stem tip and the disc plane.



5. Touch the side of the calibration stem around the side of the disc stylus. Use an external circle touch macro at two different heights.

   The Server uses this data to calculate the direction of the calibration stem and cutting tool.



6. Run the spindle stem calibration.

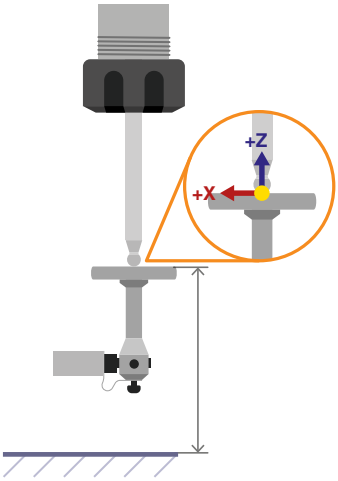
   This uses data collected in steps 3–5 to set:

   a. A tool frame at the spindle tool interface.

   b. A tool frame at the calibration stem tip.

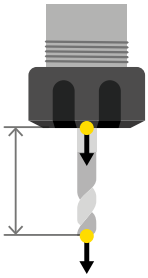
   c. A part frame on the disc stylus plane.

NOTE: Before the calibration, check that the spindle direction on the RPU browser page matches your nominal frame set in step 2.



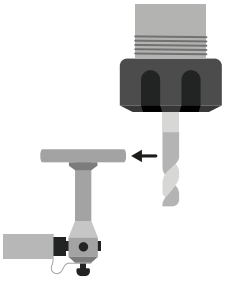
7. Align the calibration stem normal to the disc plane. Take one touch point. Run the disc height calculation.

   The Server uses the calculated height of the plane to update the disc stylus part frame.



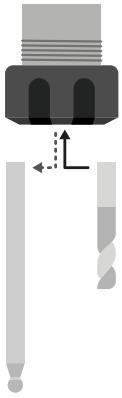
9. Use a nominal value for the cutting tool's length to run the nominal cutting tool TCP calculation.

   This moves the tool frame calculated in step 6 from the cutting tool's base to its nominal tip.

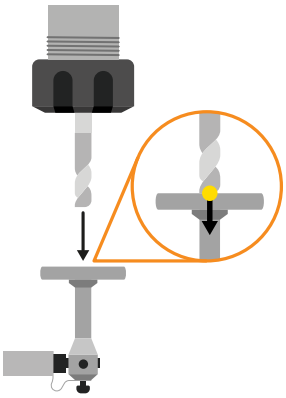


11. Use an external circle touch macro to touch around the sides of the disc stylus. Run the cutting tool size calculation.

    This calculates the tool's actual diameter.



8. Swap your calibration stem with the spindle cutting tool.



10. Touch the cutting tool tip once on top of the disc stylus plane. Run the cutting tool calibration.

    This calculates the cutting tool's actual length, and updates the tool frame.

NOTE: For steps 10 and 11, run the spindle cutting tool in reverse when probing for its exact length and diameter

Statistical analysis of residual errors in calculations

The Server can provide statistical data for the residual errors from the most recent calculation macro that the system has run (see “Residual error”, on page **68**). The macro provides the following data:

- Average value of the residual errors
- Minimum value of the residual errors
- Maximum value of the residual errors
- Absolute maximum value of the residual errors
- Standard deviation of the residual errors
- Root mean square of the residual errors
- The total number of touch points included in the calculation
- The index of the touch point that had the minimum residual error
- The index of the touch point that had the maximum residual error

The macro arguments require you to provide locations in your robot controller for the Server to write each of these values to. These must be locations which can store numbers, which may differ across manufacturers. For further information, refer to the macros user guide for your robot manufacturer at **www.renishaw.com/rcs-support**.

Clearing and exporting data

Clear or export the Server’s data environment using macros, or from the RPU browser page.

Clearing the data environment clears all Server data apart from datum configuration files.

The entire data environment, including datum configuration files, can be exported as an XML file and saved to the working folder on the RPU:

- For RCS RPU-1: /home/renishaw/RCSProbingServer
- For RCS RPU-2: C:\Renishaw\RCS Probing Server

You can share the folder with your network, so that you can access the data from your PC’s file explorer.

The XML file will have the following naming convention: DataEnv_YYYYMMDD_HHMMSS.xml

Glossary of terms

Active

An adjective to describe the part frames, tool frames, and probes that the RCS P-series system uses in a given program. Activate frames and probes in your robot program before collecting touch point data.

The robot checks the active probe's digital input for trigger signals. You can use macros to change the active probe in your robot program.

The robot moves the active tool frame within the active part frame. Most initialisation macros require you to include active frames as arguments. You can also use frame management macros to independently set active tool and part frames.

Alignment

The process of relating two co-ordinate systems to each other. For example, locating a part frame relative to a robot's base co-ordinate system. See "Simple alignments", on page **42** and "Complex alignments", on page **53** for information about different types of alignments.

Apparent diameter

The virtual diameter of a probe stylus tip, which the Server calculates during a probe calibration. Due to delays in the probe interfaces and robot I/O management, probe triggers are detected later than a mechanical touch actually occurs. To account for the trigger delay, the apparent diameter is always smaller than the size of the real stylus tip. The Server stores the apparent diameter to use in all subsequent calculations.

Approach position

A point from which a robot starts moving to collect data by touching a surface with a probe.

Argument

A value or a variable that you pass into a macro. It provides the necessary information for the robot or Server to perform the macro instruction.

Back off distance

The distance that the robot moves away from a surface after touching it. Back off distance is between the touched surface and the surface of the probe stylus tip.

Best fit

A calculated model to represent a set of data.

The Server calculates best fit features from position data. This uses data collected from a real feature to create a model for the best approximation of its position and orientation in space. This mathematical calculation, which uses the least squares method, constructs a specified feature type to a set of data points. See "Evaluate features", on page **33**.

Least-squares method

A technique to calculate the best fit model for a set of data. It minimises the sum of squared residual errors between the model and the data points.

RCS probing macros use the least-squares method for all best fit calculations.

Co-ordinate system, frame

Three orthogonal vectors (X, Y, Z) which originate at a single point. From this origin, any point in space can be positioned using three numerical co-ordinates.

Base frame

The fixed reference co-ordinate system of a robot. Located at the centre of the robot's base, defining the origin (0, 0, 0) along the X, Y, and Z axes. All other co-ordinate systems are related to this frame.

Part frame

A co-ordinate system that defines a part position and orientation within a robot's work envelope. It is usually located relative to the world co-ordinate system.

Once set, the active tool co-ordinates are referenced relative to the part frame. Some robot manufacturers use different names to refer to part frames, such as user frames.

Tool frame, tool centre point (TCP)

A co-ordinate system that defines a tool position and orientation attached to a robot arm. It is located relative to the robot flange frame.

Once set, a robot can drive the active tool point in position and orientation within the active part frame to perform its work.

Datum

Datums can be used to locate part and tool frames. With RCS P-series, you can set a datum using any combination of geometric features (see "Complex alignments", on page **53**).

Datum configuration file

A file required to conduct complex alignments of datums. It contains the following information about the feature(s) used in the alignment:

- The feature's datum priority in the alignment
- The feature type
- The position and orientation of the feature relative to the datum
- The surface ID which contains position data for the real feature

The Server uses the feature information to define a co-ordinate system.

Datum priority

The order in which a set of datum features' constrained degrees of freedom are respected. Datum features can have first, second, or third priority.

Degrees of freedom (DOF)

The number of independent ways a physical system can move. An object in three-dimensional space has six degrees of freedom, because it can move along the X, Y, or Z axis, as well as rotate about each axis.

Feature

A geometric object such as a plane, circle, cylinder, or sphere.

Move speed

The speed that the robot moves at when running any macro-controlled moves, apart from collecting touch point data.

Nominal

A stated quantity or dimension but not necessarily the exact value.

Normal

Intersecting a line or surface at a right angle (90°). A vector perpendicular to a surface.

Orthogonal

Two elements are orthogonal when they are at right angles (90°) to each other.

Overtravel

The movement of a mechanical part or tool beyond its intended limit.

Parameter

An editable piece of information a robot uses to control its movement or behaviour.

Perpendicular

A line or surface which meets another line or surface at a 90° angle.

Part setter

A touch-trigger probe mounted on a robot arm. A part-setting probe takes touch points on parts within the robot's work envelope to set part frames.

Residual error

The error which remains after an optimisation takes place. An optimisation could be a correction, calibration, alignment, or best fit.

When the Server calculates a feature, it will best fit the feature touch points to create a model. A residual error is the deviation of a touch point from the best fit model that the Server creates. In a set of data, there will be values for the maximum, minimum, average and standard deviation of the residual errors.

Absolute maximum residual error

The largest residual error value, regardless of its sign.

For example, in a set of data where the maximum residual error is 50 and the minimum residual error is –100, the absolute maximum residual error is 100.

RCS probing server, Server

An application which runs on the RPU controller for robot probing. It stores and analyses all data collected using probes. The Server exchanges data with the robot controller to perform the processes associated with an RCS P-series system.

Right-hand rules

Methods to determine the direction of vectors in three-dimensional space.

Cross product rule

This method works out the cross product of two orthogonal vectors. Use this to find the X, Y, or Z axis directions of a Cartesian co-ordinate system.

Point your right-hand thumb, index finger, and middle finger so that they are all 90° to each other. These are the three axes of the co-ordinate system.

- Thumb is the Z axis
- Index finger is the X axis
- Middle finger is the Y axis

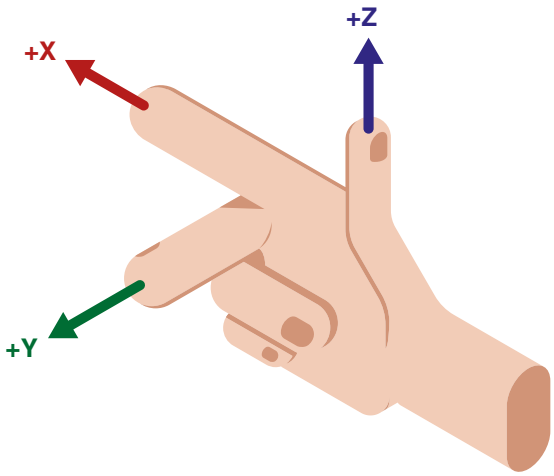


Figure 49 The right-hand rule for cross products when used to find the three positive directions of a Cartesian co-ordinate system.

When you know the direction of two axes of a co-ordinate system, you can work out the direction of the third axis. For example, if you know the +Z and +X directions of a co-ordinate system, rotate your right hand so that your thumb points in the direction of +Z, and your index finger points in the direction of +X. Your middle finger will point in the direction of +Y.

Axial vector rule

Use this method to find the direction of positive rotation about an axial vector, and vice versa.

Point your right-hand thumb in the direction of the axial vector and curl your fingers. Your fingers will point in the direction of positive rotation. Or, if you curl your right-hand fingers in the direction of a known rotation about the axis, your thumb will point in the direction of the axial vector.

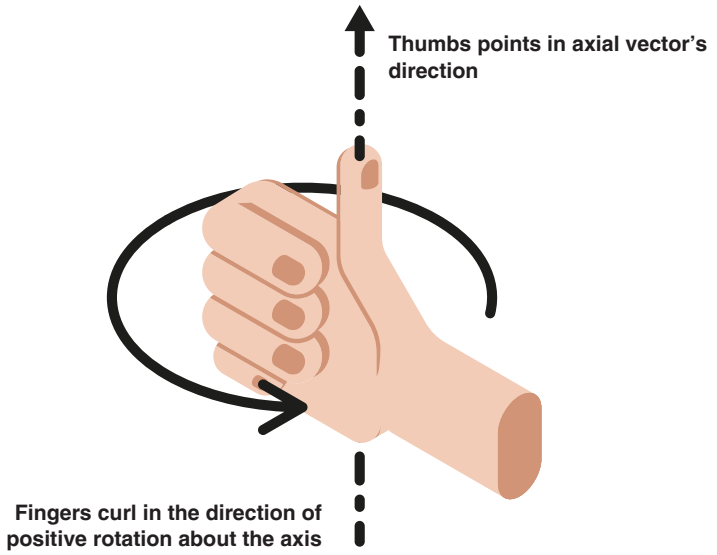


Figure 50 For axial vectors, making a 'thumbs up' gesture shows the vector direction, and the direction of positive rotation about the axis.

Search distance

The maximum distance that the robot travels when searching for a surface. If the robot does not detect a probe trigger after travelling the search distance, it will stop and display an error in the teach pendant.

Signed distance

The orthogonal distance of a given point from a feature. A positive distance is positive to the feature normal, a negative distance is negative to the feature normal.

Stem direction

A stem is the supporting structure of a sphere. Its direction is the line pointing from the stem base towards the sphere centre.

Surface

The outside part or uppermost layer of an object.

Tool setter

A touch-trigger probe mounted in a robot's work envelope. A tool-setting probe takes touch points on tools mounted on the robot's arm to set tool frames.

Touch point

Also called a 'touch'. This is a position on a surface that a probe touches, causing the probe to trigger. A touch point provides position data of a surface. The Server saves touch point data to surface IDs. It uses the touch point data to calculate positions and orientations within a given frame. Taking touch points on a surface is called 'probing'.

Point index

A number assigned to each touch point saved in a surface ID, starting in order from 0. For three touch points saved in a surface ID, the first measured touch point will have index 0, the second index 1, and the third index 2.

Touch speed

The robot's speed when approaching a surface to collect position data (touch points).

Vector

A quantity with both direction and magnitude. In RCS probing macros, vectors are defined with three numerical components (X, Y, Z).

Unit vector

A vector that has a magnitude of 1. It is used to define only the direction of a vector.

RCS probing macros use unit vectors to define directions in 3D space. The magnitudes of touch directions are defined by the 'search' and 'back off' distances as set within the touch parameters file for your robot manufacturer.

RCS P-series components



	Part name	Description
1	RCS RPU-1	Controller which enables robot probing with one robot connection.
2	RCS RPU-2	Controller which enables robot probing with up to five robot connections.
3	USB licence key	Dongle that holds RCS probing server licences.
4	RCS datum ball kit	Datum ball on a stem, with extensions to adjust height.
5	RCS PK1	Hard-wired part setter.
6	RCS TS1	Hard-wired tool setter.
7	RCS R-PK1 (Flat)	Radio part setter with ISO 9409 bracket mount.
8	RCS R-PK1 (Shank)	Radio part setter with shank mount.
9	RCS RTS	Radio tool setter.
10	RCS RMI-QE	Interface for radio probes.
11	RCS O-PK1 (Flat)	Optical part setter with ISO 9409 bracket mount.
12	RCS O-PK1 (Shank)	Optical part setter with shank mount.
13	RCS OMI-2	Interface for optical probes.
14	M4 straight stylus L100	100 mm carbon fibre stylus with M4 thread.
15	M4 straight stylus L50	50 mm stylus with M4 thread.
16	RCS spindle calibration stem L152	152 mm tool for calibrating spindle cutting tools.
17	RCS spindle calibration stem L75	75 mm tool for calibrating spindle cutting tools.
18	Disc stylus Ø25 mm	Tool-setting probe stylus for calibrating spindle cutting tools.
19	M4 break stem	Crash protection device for M4 straight styli.
20	M4 extension L10	10 mm extension for Ø25 mm tool-setting probe disc stylus.
21	PK1-S probe body	Mounting interface for TP20 probe modules for small robot tool and part-setting.
22	90° angle bracket	PK1-S probe body mounting option.
23	ISO 9409 mount bracket Ø58 mm	PK1-S probe body mounting option.
24	Standard cable	M8 3-pin female 5 m cable. Available separately.
25	TP20 standard force probe module	For PK1-S tool-setting with a straight stylus.
26	TP20 extended force probe module	For PK1-S part-setting and spindle calibration.
27	PKi	Interface for PK1-S probe.
28	M2 straight stylus L50	50 mm ceramic stylus with M2 thread.
29	M2 straight stylus L30	30 mm ceramic stylus with M2 thread.
30	Disc stylus Ø19.5 mm	Small tool-setting probe stylus for calibrating spindle cutting tools with PK1-S.
31	M2 extension L10	10 mm extension for Ø19.5 mm tool-setting probe disc stylus.

Further information

Refer to **www.renishaw.com/rcs-support** for the following resources:

- Installation guides for RCS P-series hardware.
- User guides specific to your robot manufacturer, including macro syntax and installation.
- Additional support material, such as video guides.

For further support, contact us at **www.renishaw.com/contact**.

RENISHAW
apply innovation™

www.renishaw.com/contact

#renishaw

+44 (0) 1453 524524 uk@renishaw.com

© 2026 Renishaw plc. All rights reserved. This document may not be copied or reproduced in whole or in part, or transferred to any other media or language by any means, without the prior written permission of Renishaw.
RENISHAW® and the probe symbol are registered trade marks of Renishaw plc. Renishaw product names, designations and the mark 'apply innovation' are trade marks of Renishaw plc or its subsidiaries. Other brand, product or company names are trade marks of their respective owners.
WHILE CONSIDERABLE EFFORT WAS MADE TO VERIFY THE ACCURACY OF THIS DOCUMENT AT PUBLICATION, ALL WARRANTIES, CONDITIONS, REPRESENTATIONS AND LIABILITY, HOWSOEVER ARISING, ARE EXCLUDED TO THE EXTENT PERMITTED BY LAW.
RENISHAW RESERVES THE RIGHT TO MAKE CHANGES TO THIS DOCUMENT AND TO THE EQUIPMENT, AND/OR SOFTWARE AND THE SPECIFICATION DESCRIBED HEREIN WITHOUT OBLIGATION TO PROVIDE NOTICE OF SUCH CHANGES.
Renishaw plc. Registered in England and Wales. Company no: 1106260. Registered office: New Mills, Wotton-under-Edge, Glos, GL12 8JR, UK.

Part no.: H-6852-8001-03-A
Issued: 11.2026