CONTENTS

Page

Calibration

1 Coordinate systems

All robot positions in a robot program, are stored in rectangular coordinates (e.g. xyz values for position), related to a defined coordinate system (or frame). This coordinate system may in turn be related to another coordinate system etc. in a chain. Some of these coordinate systems are embedded in the configuration of the robot system, and are not visible to the user, while others may be programmed by the user. The table below provides an overview of the various coordinate systems (or frames) used in the robot system:

Coordinate system	Defined where	Related to
Base Frame of robot	Service/View:BaseFrame. Base frame definition of robot gives relation between world and base frame.	World Frame
World Frame	No definition needed	Nothing
User Frame, fixed in room. (Tool mounted on robot)	Program/View: Data Types - wobjdata In any work object data	World Frame
User Frame, fixed on robot mounting plate. (Tool fixed in room)	Program/View: Data Types - wobjdata In any work object data	Wrist Frame
User Frame, coordinated to an external axis	Service/View:/BaseFrame. In the base frame definition of an external mechanical unit	World Frame
Object Frame	Program/View: Data Types - wobjdata In any work object data	User Frame
Program Displacement Frame	In the system variable C_PROGDISP, set up by instructions PDispSet or PDispOn etc.	Object Frame
Robtarget frame (Programmed position)	When a position is programmed.	Program Displacement Frame
Base Frame of a mechanical unit (only for internal system use)	Service/View:BaseFrame. In the base frame definition of an external mechanical unit or as configuration parameter.	World Frame
Wrist Frame	Implicit in the kinematic model of robot	Base Frame of the robot.
Tool Frame (Tool mounted on robot)	Program/View: Data Types - tooldata In any tool data	Wrist Frame
Tool Frame (Tool fixed in room)	Program/View: Data Types - tooldata In any tool data	World Frame

Now any programmed position, e.g. p1, will be related to the World Coordinate system through the chain:

world frame ’ user frame ’ object frame ’ program displacement frame ’ p1

The current position of the robot, i.e. the location of the tool, is related to the World Coordinate system through the chain:

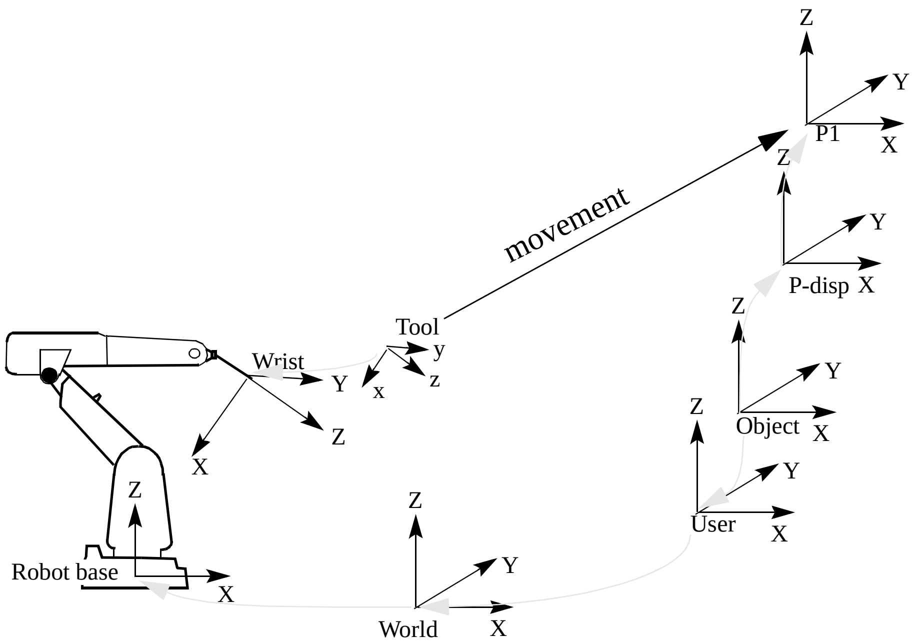
world frame ’ base frame ’ kinematic model ’ wrist centre frame ’ tool frame

When the robot is moved in automatic mode to a programmed position, the aim is to bring the tool (tool frame) to coincide with the programmed position, i.e. to close the chain:

user frame ’ object frame ’ program displacement frame ’ p1

world frame

base frame ’ kinematic model ’ wrist centre frame ’ tool frame



The accuracy of the robot, i.e. how well the tool frame will coincide with the programmed position, is normally independent of the accuracy of the various coordinate systems. This is true, however, only if the same coordinate systems are used as when programming the robot, pointing out all positions with the robot (repetition accuracy). If the coordinate systems are changed, making it possible to displace the program, then the accuracy is dependent on every single link in the chain. This means that the accuracy is directly dependent on the calibration accuracy of the various frames. This is even more important for off-line programming.

In the following chapters, an overview will be given of the steps to be taken to calibrate and define the robot and the different coordinate systems mentioned above.

2 Coordinated axes

2.1 External axes, general

All external axes are handled in mechanical units. This means that before an external axis may be moved, the mechanical unit to which it belongs, must be activated. Within a mechanical unit, the different axes will be given a logical name, from *a* to *f*. In the system parameters, these logical axes will be connected to the external axes joints. For each joint a motor and a drive unit is defined. Different joints may share the same motor and drive unit.

Two or more mechanical units may be activated at the same time, as long as they do not have the same logical axes defined in their set of external axes. However, two or more mechanical units may have the same logical axes, if they are not activated simultaneously.

Two or more mechanical units may not be activated at the same time, if they share one or more drive units, even if they use separate logical axes. This means that two logical axes, each belonging to different mechanical units, may control the same drive unit, but not at the same time.

2.2 Coordination

A mechanical unit may be coordinated or not coordinated with the robot movements.

If it is not coordinated, each axis will be moved independent of the robot movements, e.g. when jogging, only the separate axis will move. However during program execution, the external axes will be synchronized to the robot movement, in such a way that both movements will be completed in the same time.

If the mechanical unit is coordinated, the TCP velocity in the object coordinate system, will be the programmed velocity irrespective of the movements of the external axes.

Two types of coordination categories exist. The first category of coordination is when the robot base is moved, e.g. the coordination to a gantry or track movement. This means that the robot is mounted on a gantry or a track, and may be moved along these axes. The world and user/object coordinate systems, however, will be fixed in the room, and the robot movements in these coordinate systems will be independent of simultaneous gantry or track movements. This coordination is automatically active, if the mechanical unit with the track motion is active.

The second coordination category, is when the robot movements are coordinated to the movements of a user frame connected to a mechanical unit. E.g. a user frame may be placed on a turntable and connected to its movements. An ordinary work object may be used for this purpose, if it is marked with the name of the mechanical unit to be connected to, and that it should be moveable. The coordination will be active if the mechanical unit is active, <u>and</u> the “coordinated” work object is active. When such a “coordinated” work object is used, in jogging or in a move instruction, the data in the “uframe” component will be ignored and the location of the user coordinate system will only depend on the movements of the mechanical unit. However the “oframe” component will still work giving an object frame related to the user frame and also the displacement frame may be used.

3 Calibration

3.1 What is calibration?

Calibration involves setting the calibration positions (zero positions) of the axes and is used as the basis for their positioning. If the robot or external axes are not correctly calibrated, this will result in incorrect positioning and will have a negative effect on the agility of the robot. The robot is calibrated on delivery.

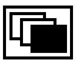
The position of the robot axes is determined using a resolver and a counter that counts the number of resolver revolutions. If the robot is correctly calibrated, it is automatically able to calculate the current position on start-up.

Calibration is carried out in two stages:

- Calibration of resolvers (fine calibration): the axes are placed in their specific calibration positions and the current resolver values are stored. For information on how to do this, see the chapter on *Repairs* in the Product Manual.
- Update of revolution counters: the correct motor revolution for the calibration is defined; the axes are placed close to their calibration positions and the revolution counters are updated.

The position of an external axis is determined using sync. switches. The same method used for the robot can be used.

3.2 Viewing the calibration status

• Press the Miscellaneous key  and select the Service window.

• Choose **View: Calibration.**

This window displays an overview of the status of all the mechanical units in the robot system (see Figure 1).

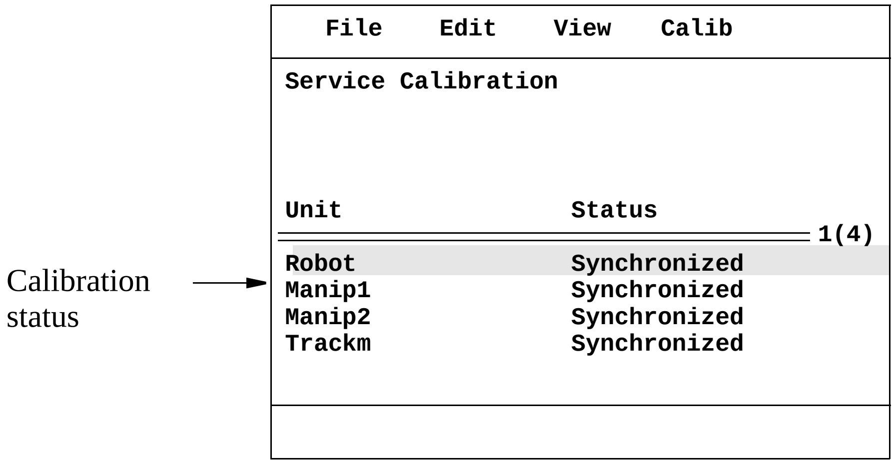


Figure 1 The Service Calibrate window shows whether or not the robot system is calibrated.

The calibration status can be any of the following:

- **Synchronized**
  All axes are calibrated and their positions are known. The unit is ready for use.
- **Not updated Rev. Counter**
  All axes are fine-calibrated but one (or more) of the axes has a revolution counter that is NOT updated. This or these must thus be updated.
- **Not calibrated**
  One (or more) of the axes is NOT fine-calibrated. This or these must thus be fine-calibrated.
- **Unsynchronized**
  At least one of the axes has a position that is NOT known. An external axis with a sync. switch must thus be synchronized. See Section 5, *Starting up*, in this manual.

3.3 Checking the calibration

If a revolution counter is incorrectly updated, it will cause incorrect positioning. Thus, check the calibration very carefully after each update.
An incorrect update can damage the robot system or injure someone.

- Run the calibration program under the /SERVICE/CALIBRAT/ directory on the system diskette, *Set up*. An alternative method is to jog the robot axis-by-axis until the axis angles in the Jogging window equal zero.
- Check each axis to see if the marks are positioned exactly opposite one another. If they are not, the calibration must be redone.

The marks may be scribed lines, vernier scales or the like. Their location is described in the chapter on *Installation and Commissioning* in the Product Manual.

3.4 Updating revolution counters

- Open the Service window.
- Choose **View: Calibration**.
- Select the desired unit.
- Move the robot or the chosen unit close to (half a motor revolution at the furthest) the calibration pose. The latter is usually indicated by a scribed line or a vernier scale. The calibration pose of the robot is described in the chapter on *Installation and Commissioning* in the Product Manual.
- Choose **Calib: Rev.Counter Update.**

A dialog box will appear, in which you can choose the axis you want to update (see Figure 2).

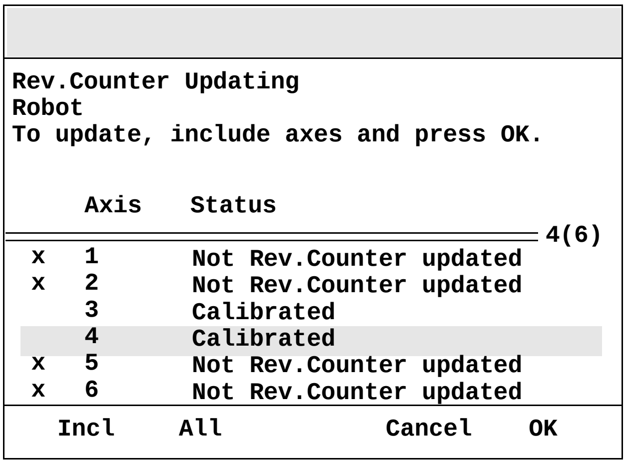


Figure 2 The dialog box used to select axes when updating the revolution counter.

- Select the axis to be updated and press the ***Incl*** function key.
  An **x** to the left indicates that the axis is to be updated.
- Use the same procedure on the remaining axes or press the function key ***All*** which selects all axes. A selected axis can be deselected by pressing the ***Excl*** function key.
- Confirm the choice of axes by pressing ***OK***.
- Start updating by pressing ***OK*** in the confirmation dialog box.

4 Base Frame for the Robot

4.1 Defining the Base Frame for the Robot

The following methods are used to define the location of the robot’s base frame in relation to the world coordinate system.

In order to define a robot base frame you need a world fixed tip within the robot’s working range, and optionally an elongator attached to the tip. If the robot is mounted on a track or similar, the track should be in its calibration position. The calibration procedure consists of a number of positionings for the robot’s TCP to a reference point. The reference point´s coordinates in the world coordinate system, must be known. The coordinates must be stated before the calibration can be done.

The following positions on the world fixed tip device are involved in the calibration:

- the tip itself (with known coordinates in world), used when defining the base frame translation
- one point on the elongator defining the positive z direction for the world coordinate system
- one point on the elongator defining the positive x direction for the world coordinate system.

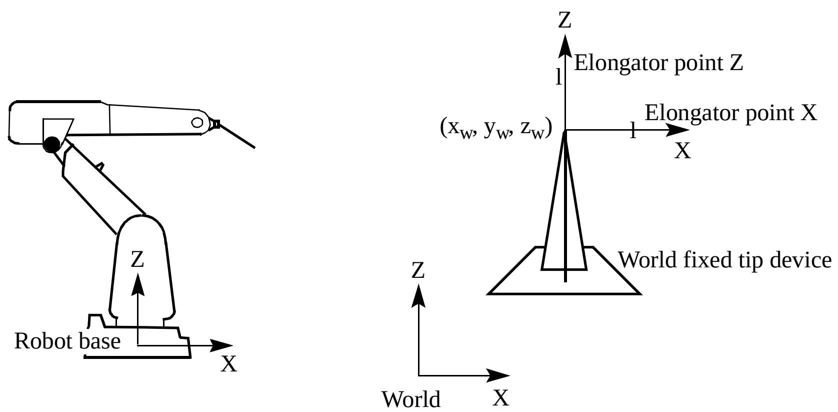


Figure 3 Robot base frame definition points.

When the necessary conditions are fulfilled the definition of the robot base frame can be performed. Please observe, that in the case of a track mounted robot, the track must be in the calibration position before the base frame of the robot may be defined.

• Press the Miscellaneous key and select the Service window.

• Choose **View:BaseFrame**.

A dialog containing all synchronized mechanical units is shown.

• Select the robot and press Enter or ***Def***.

A dialog like the one in Figure 4 will appear.

```
Robot Base Frame Definition
   Unit          : MASTER_ROBOT
   Method        : 4 points...
Point              Status
                                      1(4)
 Point 1           Modified
 Point 2           -
 Point 3           -
 Point 4           -
   Set…   ModPos          Cancel   (OK)
```

Figure 4 Robot base frame definition dialog.

To choose a definition method

Before you start modifying any positions, make sure the desired method is displayed.

• Select the field ***Method*** and press Enter ↵.

• Choose method for definition and press ***OK***.

The method requires an elongator attached to the world fixed tip.

Input of world coordinates of the reference point

• Press ***Set***.

• Input the x, y and z values.

• Verify that the input is correct and press ***OK***.

To record world fixed reference points

• Select the first point ***Point 1***.

• Jog the robot as close as possible to the world fixed tip.

• Modify the position by pressing the function key ***ModPos***.

• Repeat the above for the points ***Point 2*** to ***Point n***.

To record the elongator X point

• Select the elongator point ***Point X***.

• Jog the robot as close as possible to the elongator point on the positive X axis.

• Modify the position by pressing the function key ***ModPos***.

To record the elongator Z point

• Select the elongator point ***Point Z***.

• Jog the robot as close as possible to the elongator point on the positive Z axis.

• Modify the position by pressing the function key ***ModPos***.

To calculate the robot base frame

• Press ***OK*** to calculate the robot base frame for the selected mechanical unit.

When the calculation is finished, a dialog like the one in Figure 5 will appear.

```
Robot Base Frame Calculation Result

  Unit          : MASTER_ROBOT

Calculation Log
                                              1(10)
Method              n points (n=4)
Mean error          1.12
Max error           2.31
Cartesian X         10.34
Cartesian Y         234.56
Cartesian Z         -78.56

 File…                      Cancel    OK
```

Figure 5 The result of a robot base frame calculation.

Field	Description
Unit	The name of the mechanical unit for which the definition of robot base frame is to be done.

List contents	Description
Method	Displays the selected calibration method.
Mean error	The accuracy of the robot positioning against the tip.
Max error	The maximum error for one positioning.
Cartesian X	The x coordinate for the base frame.
Cartesian Y	The y coordinate for the base frame.
Cartesian Z	The z coordinate for the base frame.
Quaternion 1-4	Orientation components for the base frame.

The result of the calculation is expressed in the world coordinate system.

The calculation result can be saved in a separate file for later use in a PC:

• Press the function key ***File***.

• Specify a name and a location where to save the result.

• Choose ***OK*** to confirm the save.

If the estimated error is

- acceptable, press ***OK*** to confirm the new robot base frame.
- not acceptable, redefine by pressing ***Cancel***.

• Choose **File: Restart** in the Service window to activate the base frame.

The definition is now complete, but before proceeding with other tasks, verify it by jogging the robot in the world coordinate system.

5 Coordinated track motion

5.1 How to get started with a coordinated track motion

In the checklist below, the steps required to coordinate track motion are described. In each step, there may be a reference to another chapter in this manual, where more details of the specific actions to be taken will be found.

• Check that the track is not defined as moving the robot base. Find out the name of this single.

• Calibrate the robot and the track motion, i.e. the zero position of the measuring system for both robot and track must be carefully determined. See *Calibration* on page 6.

• Define the base frame of the robot, see *Defining the Base Frame for the Robot* on page 9. Please observe that the track must be in its calibration position when the robot base frame is defined. (Defines the robot base frame relative to the world frame. The procedure is necessary only if the world frame is separate from the robot base frame.)

• Define the base frame of the track, see *Defining the Base Frame for a track motion* on page 13. (Defines the rotation of the robot base relative to the track.)

• Activate the base frame coordination, see chapter 12 in this manual System Parameters/*Defining a track motion with coordinated motion*.

Example

(Name of the single for track = TRACK)

--Chose Types: Robot

--Select the robot and press Enter ↵

--Set Base frame moved by to the name of the single = TRACK

• Store all these definitions on a diskette by giving the command **File: Save All as** in the System parameter window. See chapter 12 in this manual.

• Activate the track unit in the jogging window and check that the coordination is working satisfactorily. This may be done by choosing ***World*** or ***Wobj*** in the field ***Coord*** and then jogging the track axis. The robot TCP should not move, but be fixed relative to the object coordinate system.

5.2 Defining the Base Frame for a track motion

To make coordinated track motion possible it is necessary to define the base frame of the track. This frame is located in the calibration position of the track, see Figure 6.

For the definition of a track base frame you need a world fixed tip within the robot's working range. The calibration procedure consists of a number of positionings of the TCP to the reference point. Please note that before the base frame of the track may be defined, the base frame of the robot must be defined with the track in the calibration position.

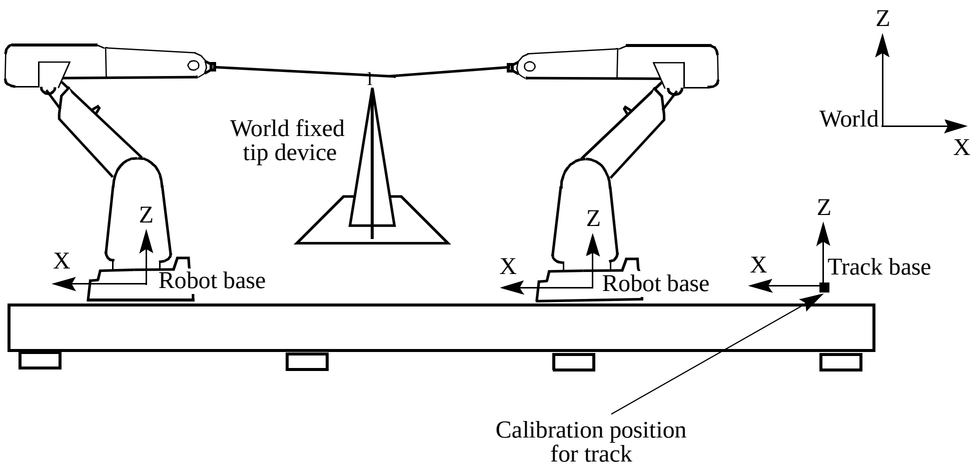


Figure 6 Track base frame definition procedure.

The track's base coordinate system has its origin in the robot's base when the track is in its calibration position. The x direction is pointing along the linear track path and the z axis of the track's coordinate system is parallel with the z axis of the robot's base coordinate system.

Figure 7 shows an example of how the base systems are oriented for a specific robot mounting. In this case the robot is mounted on the track at an angle of 45 degrees.

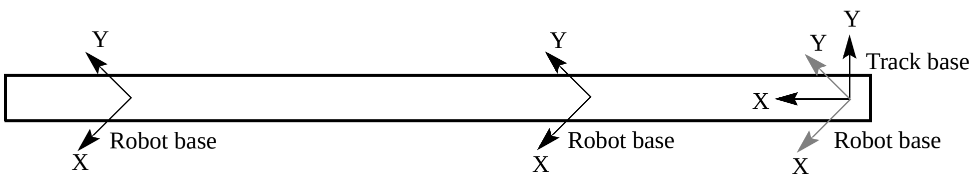
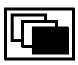


Figure 7 Track and robot base coordinate systems seen from above.

- Press the Miscellaneous key  and select the Service window.
- Choose **View:BaseFrame**.

A dialog containing all synchronized mechanical units is shown.

• Select the track unit and press Enter ↵ or ***Def***.

A dialog like the one in Figure 8 will appear.

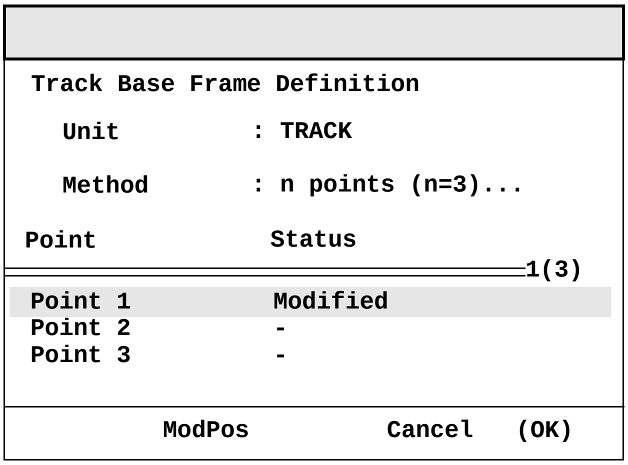


Figure 8 Track base frame definition dialog.

To choose definition methods

Before you start modifying any positions, make sure the desired method is displayed. The method defines the number of track positions from where the robot TCP will be moved to the reference point.

• Select the field ***Method*** and press Enter ↵.

• Choose the number of points to be used for definition and press ***OK***. (Currently only the three point method is implemented.)

To record world fixed reference points

Activate the track unit and run it to the calibration position, i.e. zero position should be displayed on the teach pendant.

• Select the first point ***Point 1***.

• Jog the robot as close as possible to the world fixed tip.

• Modify the position by pressing the function key ***ModPos***.

• Move the robot along the track and repeat the steps above for the points ***Point 2*** and ***Point 3***.

To calculate the track base frame

• Press ***OK*** to calculate the track base frame for the selected mechanical unit.

When the calculation is finished, a dialog like the one in Figure 9 will appear.

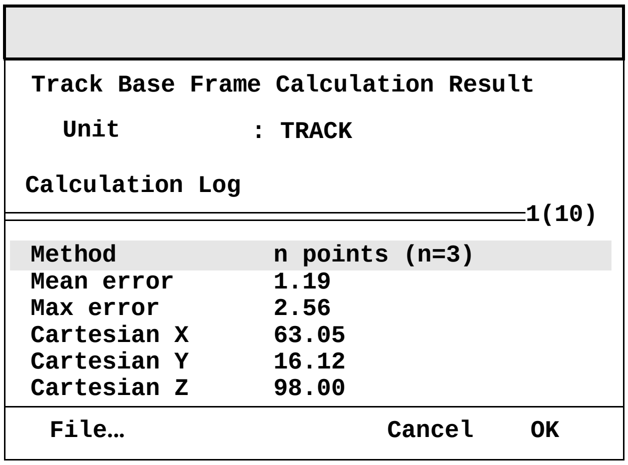


Figure 9 The result of a track base frame calculation.

The result of the calculation is expressed in the world coordinate system.

Field	Description
Unit	The name of the mechanical unit for which the definition of base frame is to be done.

List contents	Description
Method	Displays the selected track definition method.
Mean error	The accuracy of the robot positioning against the tip.
Max error	The maximum error for one positioning.
Cartesian X	The x coordinate for the base frame. (x, y, z is the same as for the robot base frame).
Cartesian Y	The y coordinate for the base frame.
Cartesian Z	The z coordinate for the base frame.
Quaternion 1-4	Orientation components for the base frame.

The calculation result can be saved in a separate file for later use in a PC:

- Press the function key ***File***.
- Specify a name and a location where to save the result.
- Choose ***OK*** to confirm the save.

If the estimated error is

- acceptable, press ***OK*** to confirm the new track base frame.
- not acceptable, redefine by pressing ***Cancel***.

- Choose **File: Restart** in the Service window to activate the track base frame.

The definition is now complete but before proceeding with other tasks, verify it by doing the following:

- Point out with the robot, in coordinated mode, the world fixed reference point with the track in different positions, and print out the position in world coordinates. Jog the track in coordinated mode.

6 Coordinated external axes

6.1 How to get started with a coordinated (moveable) user coordinate system

In the checklist below, the steps required to coordinate a user coordinate system are described. In each step, there may be a reference to another chapter in this manual, where more details of the specific actions to be taken will be found.

- Define the system parameters for the external mechanical unit, see chapter 12 in this manual, System Parameters/*Defining an external mechanical unit coordinated with the robot*. Find out the name of this mechanical unit. (E.g. *my_unit*.)
- Calibrate the robot and the mechanical unit, i.e. the zero position of the measuring system for both robot and mechanical unit must be carefully determined. See *Calibration* on page 6.
- Define the base frame of the robot, see *Defining the Base Frame for the Robot* on page 9.
- Define the user frame of the mechanical unit, see *Defining the User Frame for a rotational axis (single)* on page 17 or *Defining the User Frame for a two-axes mechanical unit, Method 1* on page 20 or *Defining the User Frame for a two-axes mechanical unit, Method 2* on page 23.
- Store all these definitions on a diskette, by giving the command **File: Save All as** in the System parameter window. See chapter 12 in this manual.
- Create a new work object data and give it a name, e.g. *turntable*. In this work object, change the component *ufprog* to FALSE, indicating that the user object should be connected to a moveable mechanical unit. Also change the component *ufmec* to the name of the mechanical unit. (E.g. *my_unit*)
- If you want the object frame to be displaced relative to the user frame, you may write the displacement in the x, y, z values of the “oframe” component of the work object. For other methods see *Defining a moveable object frame* on page 40.
- Activate the mechanical unit in the jogging window and check that the coordination is working satisfactorily. This may be done by choosing ***Wobj*** in the field ***Coord***, and the work object, e.g. *turntable*, in the field ***Wobj***, and then jogging one of the mechanical unit axes. The robot TCP should also move, following the moveable object coordinate system.
- When programming, it is important to have the coordinated work object, in this case *turntable*, programmed as an argument in each move instruction. This will be automatically added to the move instruction, if the work object is activated in the jogging window before starting the programming.

6.2 Defining the User Frame for a rotational axis (single)

This method will define the location of the user coordinate system of a rotational single axis type mechanical unit, relative to the world coordinate system. This user coordinate system should be used when a coordinated work object is used.

The definition of a user frame for a rotational external axis requires that the turntable on the external axis has a marked reference point. The calibration procedure consists of a number of positionings for the robot's TCP on the reference point when the turntable is rotated to different angles. See Figure 10.

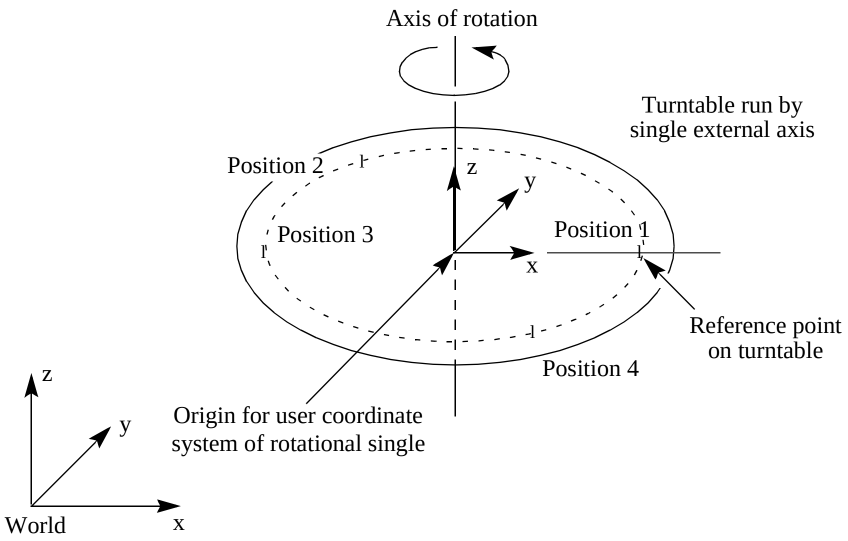


Figure 10 Definition points for a rotational axis.

The user coordinate system for the rotational axis has its origin in the centre of the turntable. The z direction coincides with the axis of rotation and the x axis goes through the reference point. Figure 11 shows the user coordinate system for two different positionings of the turntable (turntable seen from above).

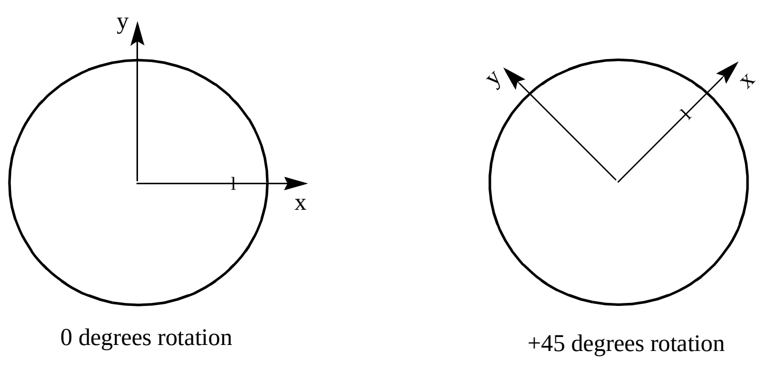


Figure 11 The user coordinate system at various angles of rotation.

• Press the Miscellaneous key and select the Service window.

• Choose **View:BaseFrame**.

A dialog containing all synchronized mechanical units is shown.

• Select the mechanical unit and press Enter or ***Def***.

A dialog like the one in Figure 12 will appear.

```
Rot Single User Frame Definition
  Unit        : ROT_SINGLE

  Method      : n points (n=4)...

Point              Status
                                    1(3)
Point 1            Modified
Point 2            -
Point 3            -

         ModPos          Cancel   (OK)
```

Figure 12 Dialog for definition of user frame for a rotational axis.

To choose a definition method

Before you start modifying any positions, make sure the desired method is displayed.

• Select the field ***Method*** and press Enter .

• Choose number of points to use for definition and press ***OK***. (Currently only the four point method is implemented.)

To record turntable reference points

Activate the mechanical unit and run it to its calibration position, i.e. zero position should be displayed on the teach pendant.

• Select the first point ***Point 1***.

• Point out the reference point on the turntable with the robot's TCP.

• Modify the position by pressing the function key ***ModPos***.

• Rotate the turntable in the positive direction and repeat the above for the points ***Point 2*** and ***Point 3***.

To calculate the user frame

• Press ***OK*** to calculate the user frame for the selected mechanical unit.

When the calculation is finished a dialog like the one in Figure 13 will appear.

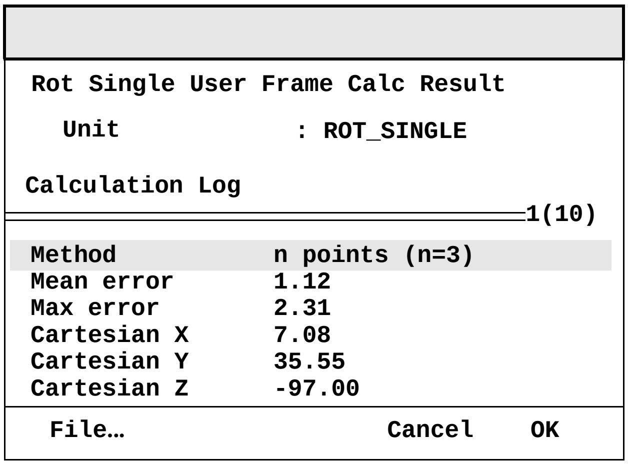


Figure 13 The result of a user frame calculation for a rotating single.

The calculation log shows the user frame expressed in the world coordinate system when the mechanical unit is in its calibration position.

Field	Description
Unit	The name of the mechanical unit for which the definition of user frame is to be done.

List contents	Description
Method	Displays the selected calibration method.
Mean error	The accuracy of the robot positioning against the reference point.
Max error	The maximum error for one positioning.
Cartesian X	The x coordinate for the user frame.
Cartesian Y	The y coordinate for the user frame.
Cartesian Z	The z coordinate for the user frame.
Quaternion 1-4	Orientation components for the user frame.

The calculation result can be saved in a separate file for later use in a PC:

- Press the function key ***File***.
- Specify a name and a location where to save the result.
- Choose ***OK*** to confirm the save.

If the estimated error is

- acceptable, press ***OK*** to confirm the new user frame.
- not acceptable, redefine by pressing ***Cancel***.

• Choose **File: Restart** in the Service window to activate the user frame.

The definition is now complete, but before proceeding with other tasks, verify it by jogging the mechanical unit in coordinated mode.

Note When defining a work object for a coordinated motion, the user frame part of the work object is left empty (unit frame). Instead the user part is computed when needed using the kinematic model and the joint position for the mechanical unit.

6.3 Defining the User Frame for a two-axes mechanical unit, Method 1

This method will define the location of the user coordinate system of an “Orbit” type mechanical unit, relative to the world coordinate system. This user coordinate system should be used when a coordinated work object is used.

It should be noted that this method requires that the kinematics (relationship between two axes) of the mechanical unit are defined in the robot system configuration. Therefore, this method can only be used for workpiece manipulators supplied by ABB, where a ready-made configuration was included in the delivery. For other types of workpiece manipulator see *Defining the User Frame for a two-axes mechanical unit, Method 2* on page 23.

The definition of this user coordinate system requires that the orbit turntable is marked with a coordinate system as shown in Figure 14. The coordinate system must have the x axis in the plane of the two turning axes of the Orbit station, when the turn table is in its calibration position.

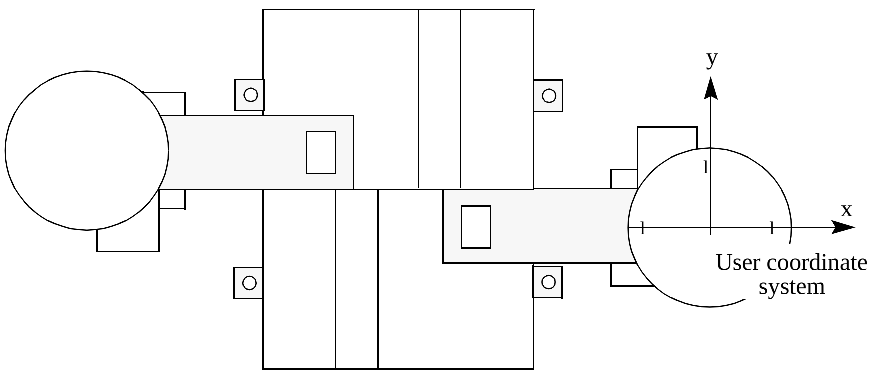
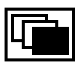


Figure 14 Orbit user coordinate system.

The coordinate system of the orbit station has its xy plane in the surface of the turntable, and the origin is located in the centre of the turntable, i.e. the z axis will coincide with the second axis.

• Press the Miscellaneous key  and select the Service window.

• Choose **View:BaseFrame**.

A dialog containing all synchronized mechanical units is shown.

• Select the mechanical unit and press Enter ↵ or ***Def***.

A dialog like the one in Figure 15 will appear.

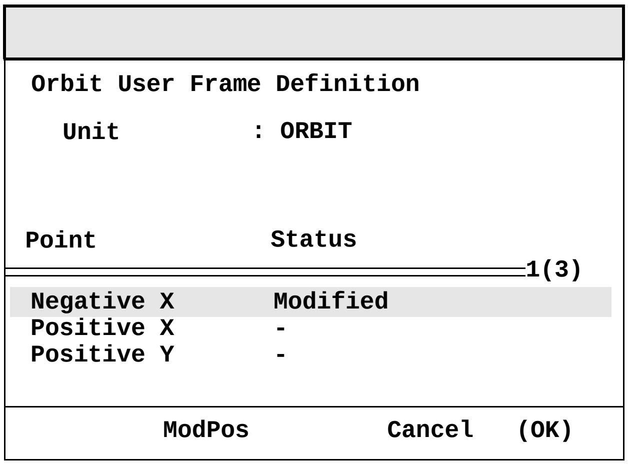


Figure 15 Dialog for definition of user frame for orbit station.

To record reference points

• Activate the mechanical unit and run it to its calibration position, i.e. zero position should be displayed on the teach pendant.

• Select the first point ***Negative X***.

• Point out the reference point on the negative x axis with the robot’s TCP (it is not necessary that the position is on the negative side of the origin, but it must be on the negative side relative to the next point “Positive X”).

• Modify the position by pressing the function key ***ModPos***.

• Select the point ***Positive X***.

• Point out the reference point on the positive x axis with the robot’s TCP.

• Modify the position by pressing the function key ***ModPos***.

• Select the point ***Positive Y***.

• Point out the reference point on the positive y axis with the robot’s TCP.

• Modify the position by pressing the function key ***ModPos***.

To calculate the user frame

• Press ***OK*** to calculate the user frame for the selected mechanical unit.

When the calculation is finished, a dialog like the one in Figure 16 will appear.

The calculation log shows the user frame expressed in the world coordinate system when the mechanical unit is in its calibration position.

```
Orbit User Frame Calculation Result

  Unit          : ORBIT

Calculation Log
                                    1(9)
 Cartesian X       123.45
 Cartesian Y       45.67
 Cartesian Z       398.56
 Quaternion 1      0.382683
 Quaternion 2      0.000000
 Quaternion 3      0.923880

  File…                   Cancel    OK
```

Figure 16 The result of a linear moving base frame calculation.

Field	Description
Unit	The name of the mechanical unit for which the definition of user frame is to be done.

List contents	Description
Cartesian X	The x coordinate for the user frame.
Cartesian Y	The y coordinate for the user frame.
Cartesian Z	The z coordinate for the user frame.
Quaternion 1-4	Orientation components for the user frame.

The calculation result can be saved in a separate file for later use in a PC:

- Press the function key ***File***.
- Specify a name and a location where to save the result.
- Choose ***OK*** to confirm the save.

If the estimated error is

- acceptable, press ***OK*** to confirm the new user frame.
- not acceptable, redefine by pressing ***Cancel***.

• Choose **File: Restart** in the *Service* window to activate the user frame.

The definition is now complete, but before proceeding with other tasks, verify it by jogging the mechanical unit in coordinated mode.

Note When defining a work object for a coordinated motion, the user frame part of the work object is left empty (unit frame). Instead the user part is computed when needed using the kinematic model and the joint position for the mechanical unit.

6.4 Defining the User Frame for a two-axes mechanical unit, Method 2

This method will define the location of the user coordinate system of an “Orbit” type mechanical unit, relative to the world coordinate system. This user coordinate system should be used when a coordinated work object is used.

The definition requires that the parameter error_type under ROBOT for the mechanical unit is set to ERROR (-error_type ERROR)

It should be noted that this method does not require that the kinematics (relationship between two axes) of the mechanical unit are defined in the robot system configuration. If this is a known factor, another method can be used. See *Defining the User Frame for a two-axes mechanical unit, Method 1* on page 20.

Figure 17 shows an orbit station with two rotational axes and a turntable mounted on the second axis.

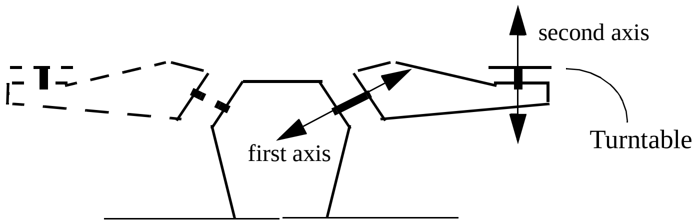


Figure 17 Geometric structure of an orbit station.

The definition of the user frame requires that the turntable has a marked reference point. The origin of the user frame is located in the centre of the turntable with the z axis coinciding with the second axis of rotation. The x axis goes through the reference point (see Figure 18).

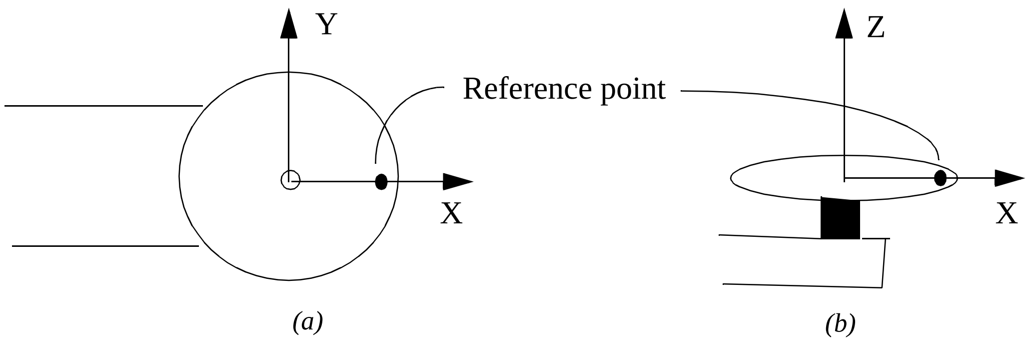
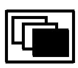


Figure 18 The turntable seen from above (a) and side (b).

The user frame is determined by two definition procedures. One procedure for the first axis and another similar procedure for the second axis. These two procedures are performed separately but both are necessary to complete the user frame definition.

• Press the Miscellaneous key  and select the Service window.

• Choose **View: Two Axes Definition**

A dialog containing all synchronized mechanical units is shown.

• Select the mechanical unit and press Enter [↵] or ***Def***.

A dialog like the one in Figure 19 will appear.

```
Mechanical Unit Axes Definition

   Unit          : MHA160B1
   Method        : n points (n=4)...
   Axis          : 1

Point                  Status
                                          1(4)
 Point 1               Modified
 Point 2               -
 Point 3               -
 Point 4               -

            ModPos          Cancel    (OK)
```

Figure 19 Dialog for definition of axes.

Defining the first axis

Before defining the first axis, both axes must be run to their calibration positions. The procedure to define the first axis consists of a number of positionings for the robot's TCP on the reference point when the first axis is rotated to different angles. Position 1 is the position of the reference point when both axes are fixed to their calibration positions. The following positions, position 2, 3, 4 etc., are the positions of the reference point when the first axis is rotated to greater angles in successive steps. See Figure 20.

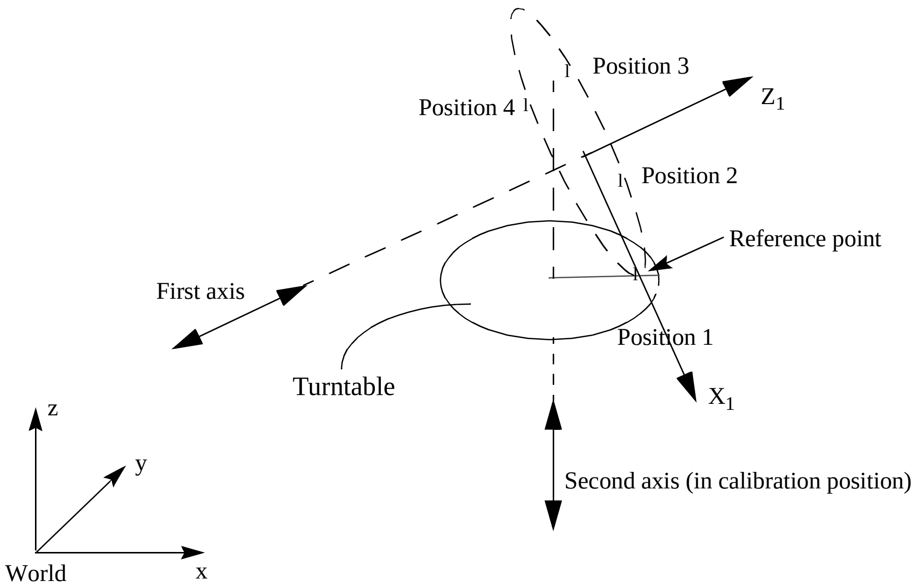


Figure 20 Definition of the first axis. Four positionings of the robot's TCP on the reference point are performed with the first axis rotated to different angles.

Defining the second axis

Before defining the second axis, both axes must be run to their calibration positions. The procedure to define the second axis consists of a number of positionings for the robot's TCP on the reference point when the second axis is rotated to different angles. Position 1 is the position of the reference point when both axes are fixed to their calibration positions. The following positions, position 2, 3, 4 etc., are the positions of the reference point when the second axis is rotated to greater angles in successive steps. See Figure 21.

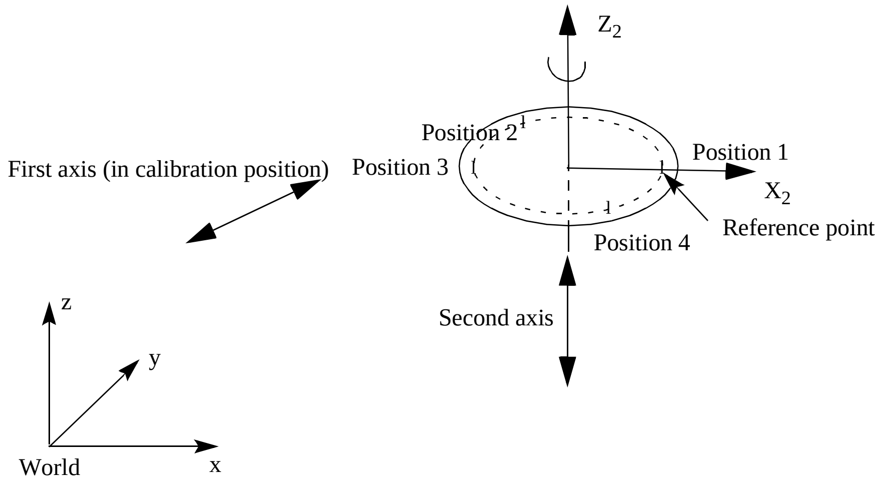


Figure 21 Definition of the second axis. Four positionings of the robot's TCP on the reference point are performed with the second axis rotated to different angles.

This frame coincides with the user frame when both axes are fixed to their calibration positions.

To choose a definition method

Before you start modifying any positions, make sure the desired method is displayed and the mechanical unit is activated.

• Select the field ***Method*** and press Enter ↵ .

• Choose the number of points to use for the axis definition and press **OK**.

To choose axis

You can choose which one of the axes you want to define. Remember that both axes must be defined to complete the user frame definition. It is possible to redefine both axes or just one of them.

• Select the field ***Axis*** and press Enter ↵ to switch axis.

To record reference points for the first axis definition

Make sure Axis 1 is chosen. Run the mechanical unit to its calibration position.

- Select the first point, ***Point 1***.
- Point out the reference point on the turntable with the robot’s TCP.
- Modify the position by pressing the function key ***ModPos***.
- Rotate the first axis to a greater angle and repeat the above for the points ***Point 2*** to ***Point n***.
- Press ***OK*** to calculate the frame of the first axis.

To record reference points for the second axis definition

Make sure Axis 2 is chosen. Run the mechanical unit to its calibration position.

- Select the first point ***Point 1***.
- Point out the reference point on the turntable with the robot’s TCP.
- Modify the position by pressing the function key ***ModPos***.
- Rotate the second axis to a greater angle and repeat the above for the points ***Point 2*** to ***Point n***.
- Press ***OK*** to calculate the frame of the second axis.

To confirm/cancel the new axis definition

When ***OK*** is pressed after the points have been modified for an axis, a dialog like the one in Figure 22 will appear.

```
Mechanical Unit Axes Calc Result
Unit        : MHA160B1
Axis        : 1
Calculation Log
                                   1(10)
 Method            n points (n=4)
 Mean error        0.57
 Max error         0.98
 Cartesian X       7.08
 Cartesian Y       35.55
 Cartesian Z       -97.00

  File...                 Cancel    OK
```

Figure 22 The result of the first axis definition.

The calculation log shows the calculated frame expressed in the world coordinate system.

Field	Description
Unit	The name of the mechanical unit for which the definition of the axis is to be done.
Axis	The chosen axis.

List contents	Description
Method	Displays the selected method.
Mean error	The accuracy of the robot positioning relative to the reference point.
Max error	The maximum error for one positioning.
Cartesian X	The x coordinate for the frame.
Cartesian Y	The y coordinate for the frame.
Cartesian Z	The z coordinate for the frame.
Quaternion 1-4	Orientation components for the frame.

The calculation result can be saved in a separate file for later use in a PC:

- Press the function key ***File***.
- Specify a name and a location where to save the result.
- Choose ***OK*** to confirm the save.

If the estimated error is

- acceptable, press ***OK*** to confirm the new axis definition. Now the next axis can be defined if necessary.
- not acceptable, redefine by pressing ***Cancel***.

• Choose **File: Restart** in the *Service* window to activate the user frame.

The user frame definition is now completed, but before proceeding with other tasks, verify it by jogging the mechanical unit in coordinated mode.

Note When defining a work object for a coordinated motion, the user frame part of the work object is left empty (unit frame). Instead the user part is computed when needed using the kinematic model and the joint position for the mechanical unit.

7 Defining Tools

The position of the robot and its movements are always related to its tool coordinate system, i.e. the TCP and tool orientation (see Figure 23). To get the best performance, it is important to define the tool coordinate system as correctly as possible. For more information, see the RAPID Reference Manual/ Motion and I/O Principles.

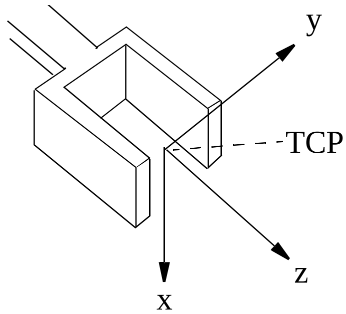


Figure 23 The tool coordinate system for a gripper.

A tool coordinate system can either be defined manually or the robot can be used as the measuring tool. Manual definitions can be used if accurate data for the dimensions of the tool is available or if minor corrections are to be done.

7.1 Creating a new tool

A tool should normally be placed in the system module, *User*. In that way, it will be common to all programs, which means that if a TCP is modified, all programs will automatically be affected. The tool can then also be used for jogging when there is no program in the program memory.

• Open the *Program Data Types* window by choosing **View: Data Types**.

• Select the type *tooldata* and press Enter ↵ .

• Create the new tool using one of the following alternatives:

  - **alt 1**. Press the function key ***New***.
    The tool's TCP and orientation will then be the same as the robot's mounting flange.

  - **alt 2**. Select an existing tool and press the function key ***Dupl***.
    The tool's TCP and orientation will then be the same as the one duplicated.

A window appears, displaying the name of the data.

• If you want to change the name, press Enter ↵ and specify a new name.

• Press the function key ***Decl***.

A dialog box appears, displaying the basic tooldata declaration.

• If you want to save the data in another module, select the field ***In Module*** and press Enter ↵ . Specify the name of the module in which the data is to be saved.

• Press ***OK*** to confirm.

Note: Do not change the type of the tool. This must always be of the persistent type.

7.2 Manually updating the TCP and weight of a tool

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *tooldata* and press Enter ↵ .
- Select the tool to be changed and press Enter ↵ .
- Select the TCP component (x, y, z) that you wish to change.
- Change the value using the numeric keyboard. To enter a decimal point (.) or minus sign (-), use the function keys.
- Select the *mass* component.
- Change the weight using the numeric keyboard.
- If the tool is stationary, i.e. not mounted on the robot, change the component *robhold* to FALSE. For more information about stationary tools see *Stationary tool* on page 33.
- Choose ***OK*** to confirm the change.

Note: Only the mass of the tool should be specified. A payload handled by a gripper is specified by the instruction GripLoad.

7.3 Methods of defining the tool coordinate system

To define the TCP of a tool, you need a world fixed tip within the robot's working space. You then jog to (at least) four robot positions with different orientations, as close as possible to the world fixed tip (see Figure 24). These positions are called *approach points*.

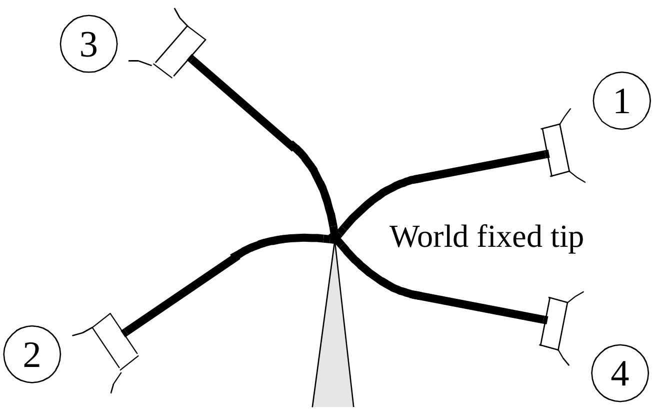


Figure 24 Approach points for a tool's TCP.

To define a complete orientation of a tool, you move any position on the desired z axis and any position on the desired x axis to the world fixed tip. These positions are called *elongator points* (see Figure 25). This can be done by fitting an elongator to the tool to define the z and x directions or by aligning the tool according to the world coordinate system and then jogging the robot in these directions.

Note The elongator points must be defined with the same orientation as the last approach point used.

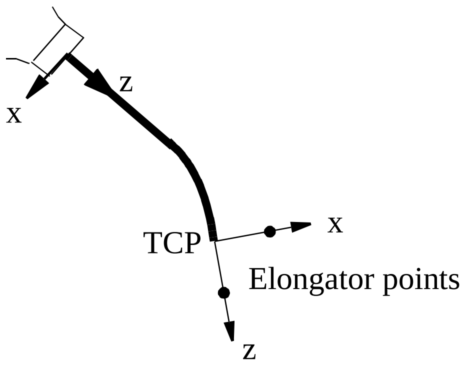


Figure 25 Elongator points for a tool's orientation.

If you only want to define the TCP, only the world fixed tip is needed. If you only need a definition of the orientation in the z direction, the elongator will only point to z.

The following methods are supported:

- ***4-point TCP***
  Four approach points are used to define the TCP. The orientation will be set according to the wrist coordinate system (see Figure 26).

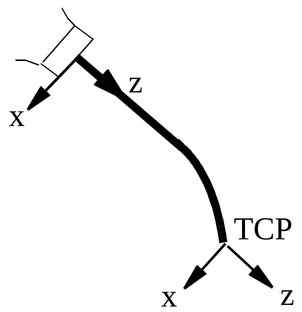


Figure 26 Using the 4-point method, only the TCP is defined. The tool direction will correspond to the wrist coordinate system.

- ***4-p TCP ORIENT NOT SET***
  The same as 4-point TCP but the orientation will not be changed.

- ***5-point TCP&Z***
  Four approach points are used to define the TCP and one elongator point is used to define the z direction of the tool. The x and y directions will be as close as possible to the corresponding axes in the wrist coordinate system (see Figure 27).

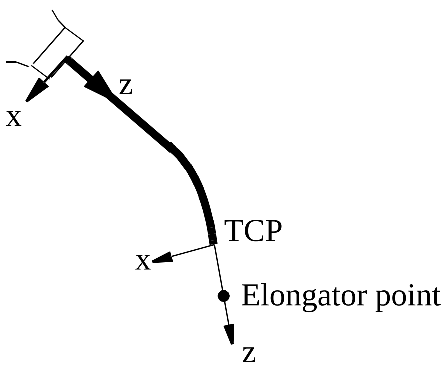


Figure 27 Using the 5-point method, the TCP and the tool's z direction are defined. The x and y directions are set automatically by the robot.

- ***6-point TCP&ZX***
  Four approach points are used to define the TCP, one elongator point is used to define the z direction and one elongator point is used to define the x direction of the tool (see Figure 28).

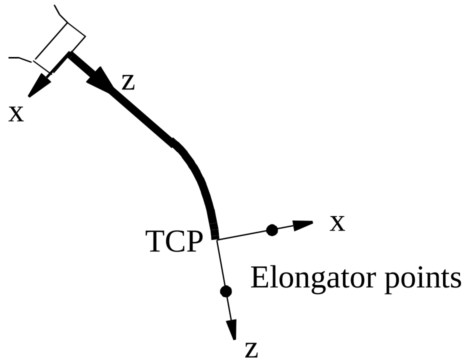


Figure 28 Using the 6-point method, the TCP and all the tool's directions are defined.

7.4 Using the robot to change the TCP and orientation of a tool

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *tooldata* and press Enter ↵ .
- Select a tool (or create a new tool, see ***Creating a new tool*** on page 28).
- Choose **Special: Define Coord**.

A dialog box appears, displaying the points defined by whichever method was used (see Figure 29).

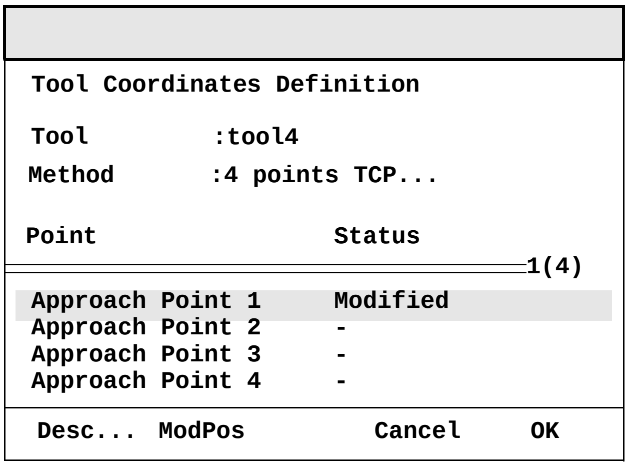


Figure 29 The robot can be used to define the tool coordinate system.

The status can be defined as follows:

<u>Status</u>	<u>Meaning</u>
-	No position defined
Modified	Position modified

To choose a definition method

Before you start modifying any positions, make sure the desired method is displayed. See ***Methods of defining the tool coordinate system*** on page 29.

• Select the field ***Method*** and press Enter ↵ .

• Choose a method and press ***OK***.

To record Approach Points

• Select the first point ***Approach Point 1***.

• Jog the robot as close as possible to the world fixed tip.

• Modify the position by pressing the function key ***ModPos***.

• Repeat the above for the points ***Approach Point 2-4***.

To record Elongator Point Z (if the 4-point TCP method is not used)

• Select ***Elongator z Point***.

• Jog - without changing the orientation from the last approach point - any point on the desired positive z axis to the world fixed tip. An extension should be fitted to obtain better accuracy.

• Modify the position by pressing the function key ***ModPos***.

To record Elongator Point X (only if the 6-Point TCP&XZ method is used)

• Select ***Elongator x Point***.

• Jog - without changing the orientation from the last approach point - any point on the desired positive x axis to the world fixed tip.

• Modify the position by pressing the function key ***ModPos***.

To calculate the tool coordinate system

• Press ***OK*** to calculate the tool coordinate system.

When the calculation is finished, a dialog like the one in Figure 30 will appear.

```
Tool Calculation Result
  Tool      : tool4
  TCP       : (50.57, 0.00, 231.82)
Calculation Log
                                      1(4)
Method              4 points TCP
Mean Error          1.12
Max Error           2.31
Quaternion 1        0.978453

File...                  Cancel     OK
```

Figure 30 The result of a tool calculation.

Field	Description
TCP	The values of the calculated TCP.
Mean Error	The average distance that the approach points are from the calculated TCP, i.e. how accurately the robot was positioned relative to the tip.
Max Error	The maximum error for one approach point.

The calculation result can be saved in a separate file for later use in a PC. However, this file cannot be read by the robot:

- Press the function key ***File***.
- Specify a name and a place to save the result.
- Choose ***OK*** to confirm the save.

• If the estimated error is
  - acceptable, press ***OK*** to confirm the new tool coordinate system;
  - not acceptable, redefine by pressing ***Cancel***.

The definition is now complete, but before proceeding with other tasks, verify it by linearly jogging in the tool coordinate system and by reorienting the TCP.

If the tool has been stored in a system module, save this module.

7.5 Stationary tool

When using a stationary tool, the robot is holding the work piece and the tool is stationary in the room. In this case the TCP coordinates are related to the world coordinate system, and the work object (i.e. the user coordinate system) is related to the wrist coordinate system.

Creating a new tool.

• The tool is created as described in previous chapters.

- The component *robhold* is changed to FALSE.

Creating a corresponding work object

When using a stationary tool, it is also necessary to use a work object held by the robot.

- The work object is created as described in *Creating a new work object* on page 36.
- The component *robhold* is changed to TRUE.

Methods for defining the stationary tool coordinate system

The methods are the same as for a TCP mounted on the robot. However in this case, the reference tip is mounted on the robot and the robot is moved, so as to bring the tip to the stationary tool TCP. The tip must be defined and activated as a tool before the definition of the stationary tool may be done.

- Define and activate the tool, which should be used as a pointing tip, and which is mounted on the robot.
- Now the same methods for defining the stationary tool may be used, as described in *Manually updating the TCP and weight of a tool* on page 29 and *Using the robot to change the TCP and orientation of a tool* on page 31. Use the robot mounted tip to point out the stationary TCP with four approach points, and if needed, the z and x directions of the axes. It is possible to use the same positioning for all four TCP approach points to perform a faster frame definition. However, it is recommended to point out the stationary TCP with different orientations to obtain a reliable statistical result. The point that is used to approach the stationary TCP must be the active TCP (hold by the robot).

Note: If the stationary tool is to be used with coordinated track motion, the coordination must be active during the calibration of the stationary tool.

8 Work Objects and Program Displacements

8.1 General

All programmed positions are related to a program displacement frame, which in turn is related to the object frame, related to the user frame, related to the world frame. Both object and user frames are included in a work object, which may be added to each move instruction. See Figure 31.

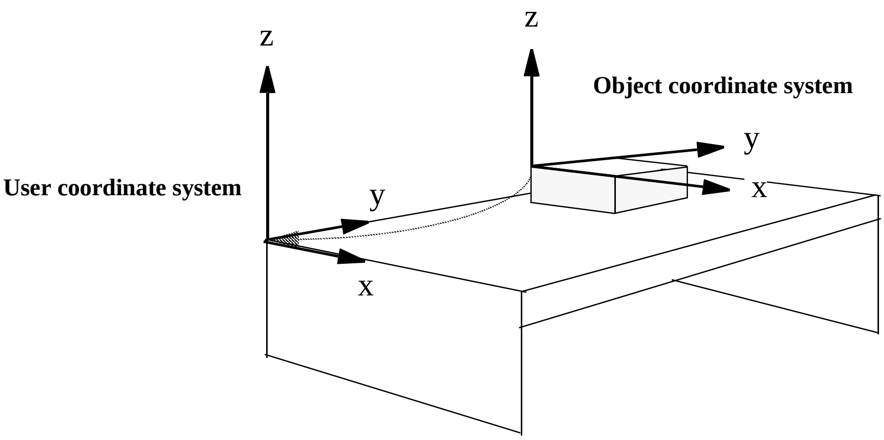


Figure 31 A user and an object coordinate system describe the position of a work object.

The intention is to use the work object to define both the position of a table (user frame) and the position of the object to work on (object frame). When the table or the object is moved, the program may still work if the corresponding work object is updated. These coordinate systems are very well suited to off-line programming since the positions specified can usually be taken directly from a drawing of the work object.

The program displacement coordinate system is used for small temporary displacements, e.g. as the result of a search operation. This displacement is modal, i.e. it is activated in a separate instruction and then it remains active until it is deactivated in another separate instruction. See Figure 32.

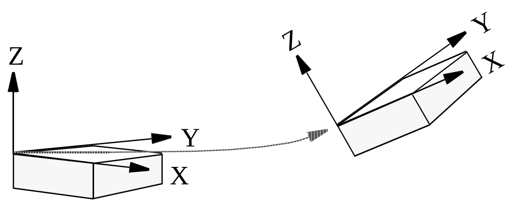


Figure 32 Using a displacement frame, all positions in the program can be displaced.

All such program displacements include both robot displacements and external axes displacements.

Please note the difference between work object and program displacement. The work object used must be added to each move instruction and it must be active when programming the move instruction. It should be included from the beginning because it is a little tricky to add it afterwards. A program displacement, however, which is activated in a separate instruction, is very easy to add afterwards.

8.2 Using work objects

In the checklist below, the steps required to define and use a work object are described. In each step, there may be a reference to another chapter in this manual, where more details of the specific actions to be taken will be found.

- Before starting to program, the work objects to be used must be defined. First create a new work object and give it a name, e.g. "wobj1", see *Creating a new work object* on page 36.
- Define the work object by using the robot to point out three points on the user frame and the object frame respectively. See *Using the robot to change the work object* on page 38. Please note that if the same positions are used both for the user frame and for the object frame, then all the locations will go into the user frame and the object frame will still be zero. It should also be noted that it is possible to update the values of the work object manually. See *Manually updating the user and object coordinate system of the work object* on page 37.
- Now check that the definition of the work object is correct by jogging the robot in the object coordinate system. This may be done by choosing the ***Wobj*** in the field ***Coord*** in the jogging window, and the work object, e.g *wobj1*, in the field ***Wobj***, and then jogging the robot.
- When programming it is important to have the work object, in this case *wobj1*, programmed as an argument in each move instruction. This will be automatically added to the move instruction, if the work object is activated in the jogging window before starting the programming.

8.3 Creating a new work object

A work object should normally be placed in the system module, *User*. In this way it will be common to all programs, which means that if a work object is modified, all programs will also automatically be modified. The work object can also be used for jogging when there is no program in the program memory.

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *wobjdata* and press Enter ↵ .
- Create the new work object using one of the following alternatives:
  - **alt 1**. Press the function key ***New***.
    The user and object coordinate systems will then coincide with the world coordinate system.
  - **alt 2**. Select an existing work object and press the function key ***Dupl***.
    The coordinate systems will then be the same as those duplicated.

A window appears, displaying the name of the data.

- If you want to change the name, press Enter ↵ and specify a new name.
- Press the function key ***Decl***.

A dialog box appears, displaying the basic wobjdata declaration.

- If you want to save the data in another module, select the field ***In Module*** and press Enter [↵] . State the name of the module where the data is to be sent.
- Press ***OK*** to confirm.

Note: Do not change the work object type. This must always be of the persistent type.

8.4 Manually updating the user and object coordinate system of the work object

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *wobjdata* and press Enter [↵] .
- Select the work object to be changed and press Enter [↵] .
- Select the component (x, y, z, q1-q4) that you wish to change.
- Change the value using the numeric keyboard. To enter a decimal point (.) and minus sign (-), use the function keys.
- Choose ***OK*** to confirm.

Note If the work object is defined using a movable user coordinate system, only the object coordinate system need be defined. The user coordinate system is defined in the Service window. See *Coordinated external axes* on page 16.

8.5 Methods of defining a work object

The methods used to define the user and object coordinate system are called:

- ***No change***
  No changes to the definition of the user or object coordinate system will be made, i.e. the definition of the user or object frame will be left as it is.
- ***3-point***
  Three points are used: two points on the x axis and one point on the y axis (see Figure 33). A tool with a known TCP is required.

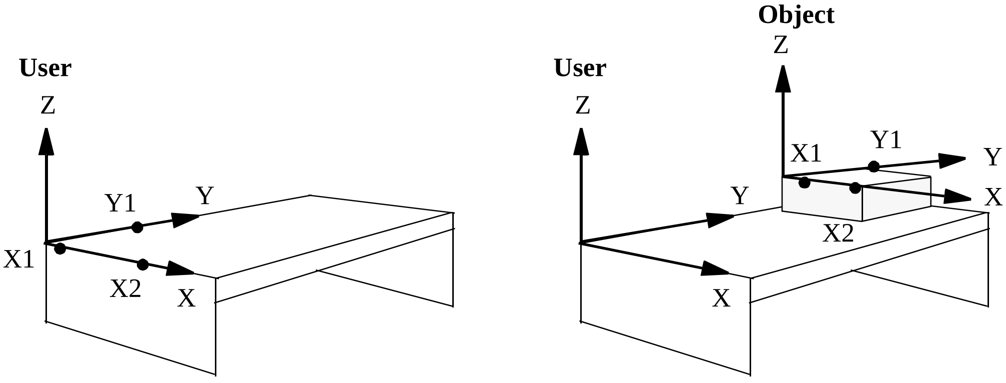


Figure 33 Measuring points for defining a work object.

8.6 Using the robot to change the work object

- Choose **View: Data Types**.
- Select the type *wobjdata* and press Enter ↵ .
- Select the work object to be defined (or create a new one, see *Creating a new work object* on page 36).
- Choose **Special: Define Coord**.

A dialog box appears, displaying the points defined by the method that was used (see Figure 34).

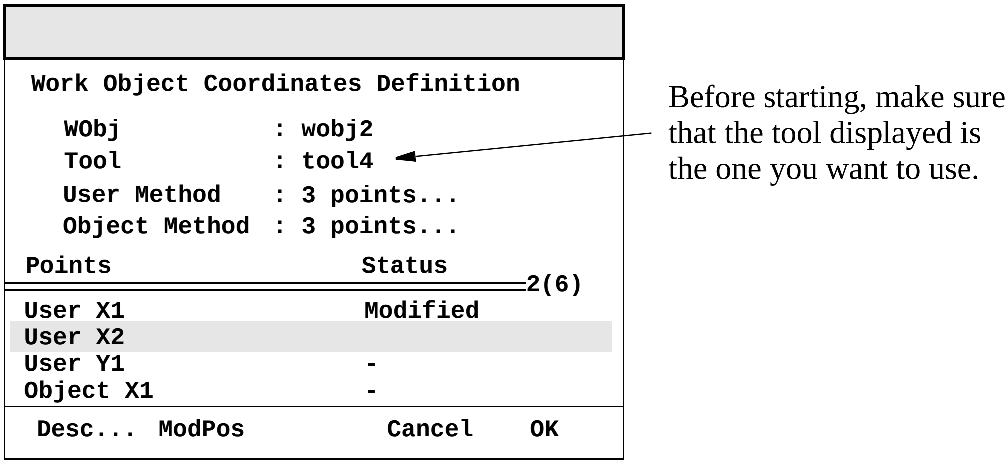


Figure 34 The robot can be used to define the position of the work object.

The status can be defined as follows:

Status	Meaning
-	No position defined
Modified	Position modified

To record Measuring Points for the user coordinate system

Note If the work object is defined using a movable user coordinate system, the user coordinate system is defined in the Service window. See *Coordinated external axes* on page 16.

- Select the first measuring point ***User X1***.
- Jog the robot as close as possible to a point on the x axis.
- Modify the position by pressing the function key ***ModPos***.
- Select the measuring point ***User X2***.
- Jog the robot as close as possible to a point on the x axis defining the positive x direction.
- Modify the position by pressing the function key ***ModPos***.

- Select the measuring point ***User Y1***.
- Jog the robot as close as possible to a point on the positive y axis.
- Modify the position by pressing the function key ***ModPos***.

To record measuring Points for the object coordinate system

- Select the first measuring point ***Object X1***.
- Jog the robot as close as possible to a point on the x axis.
- Modify the position by pressing the function key ***ModPos***.
- Select the measuring point ***Object X2***.
- Jog the robot as close as possible to a point on the x axis defining the positive x direction.
- Modify the position by pressing the function key ***ModPos***.
- Select the measuring point ***Object Y1***.
- Jog the robot as close as possible to a point on the positive y axis.
- Modify the position by pressing the function key ***ModPos***.

To calculate the user and object coordinate system

- Press ***OK*** to calculate the coordinate systems.

When the calculation is finished, a dialog like the one shown in Figure 35 will appear.

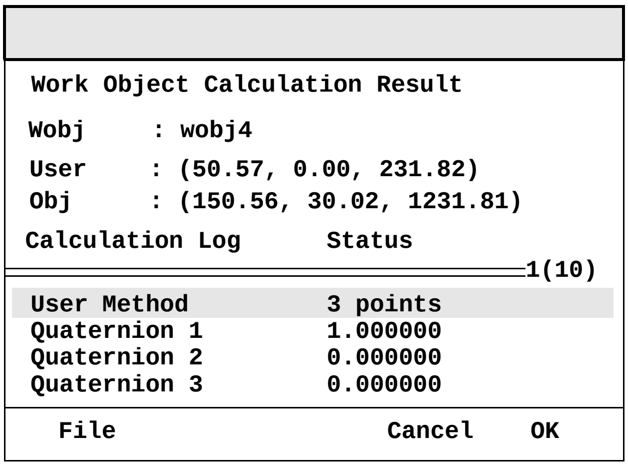


Figure 35 The result of a work object calculation.

Field	Description
User	The origin of the user coordinate system.
Obj	The origin of the object coordinate system.

The calculation result can be saved in a separate file for later use in a PC.
Note, however, that this file cannot be read by the robot:

- Press the function key ***File***.

- Specify a name and a place to save the result.
- Choose ***OK*** to confirm the save.

The definition is now complete, press ***OK*** to confirm the new work object, but before proceeding with other tasks, verify it by jogging linearly in the work object's coordinate system.

If the work object was stored in a system module, save this module.

8.7 Defining a moveable object frame

Method 1

- Use the method for defining a work object. See *Using the robot to change the work object* on page 38. When using this method, please observe that the coordination flag, i.e. the component *ufprog* in the work object data must be temporarily set to TRUE. You must point out three positions for the user system (which must be placed as the coordinated one) and three positions for the object system.

If the user system is not possible to reach, use method 2 or 3 below.

Method 2

- Activate the coordinated work object and jog the robot to the point where you want to place the origin of the object frame.
- Read the coordinates, x, y, z for this position in the jogging window.
- Write these values in the *o_frame* component of the work object data.

This will shift the object frame to the new position, with the same orientation as the user frame. If you want another orientation, use method 3.

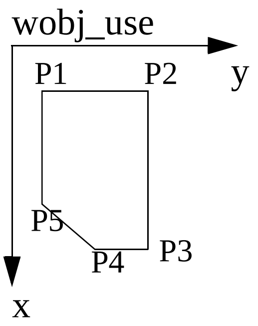
Method 3

- Activate the coordinated work object (suppose it is named *co_wobj*), create three positions, e.g. *p1*, *p2* and *p3*. *p1* should be located at the origin of the shifted object frame, *p2* on the x axis and *p3* in the x-y plane.
- Program and execute the instruction
  co_wobj.oframe: = DefFrame(p1, p2, p3);

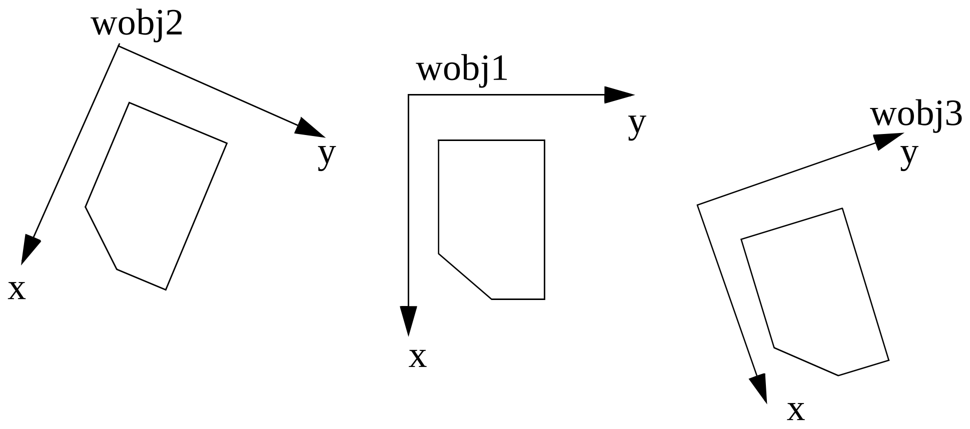
8.8 How to use different work objects to get different displacements

Suppose you have used the work object *wobj_use* when creating a procedure, *draw_fig*, as below.

```
MoveL p1, v200, z1, tool1\WObj:=wobj_use;
MoveL p2, v200, z1, tool1\WObj:=wobj_use;
MoveL p3, v200, z1, tool1\WObj:=wobj_use;
MoveL p4, v200, z1, tool1\WObj:=wobj_use;
MoveL p5, v200, z1, tool1\WObj:=wobj_use;
```



Now you want it to be performed displaced, corresponding to *wobj1*, *wobj2* or *wobj3*, see below.



Suppose that the value of *reg1* is used to control which work object should be used.

If *reg1 = 1*, *wobj1* should be used; if *reg1 = 2*, *wobj2* should be used; and if *reg1 = 3*, *wobj3* should be used.

The program below will set wobj_use = wobj1 if reg1 = 1, then call the *draw_fig* procedure, etc.

```
IF reg1=1 THEN
                    wobj_use:=wobj1;
                    draw_fig;
ENDIF
IF reg1=2 THEN
                    wobj_use:=wobj2;
                    draw_fig;
ENDIF
IF reg1=31 THEN
                    wobj_use:=wobj3;
                    draw_fig;
ENDIF
```

8.9 How to adjust the program vertically using the object frame

When running your program in the location defined by *wobj2*, suppose you find it is positioned a little too high. The vertical position can be adjusted by moving the object coordinate system a small amount vertically, relative to the user coordinate system, i.e. the z coordinate for object is changed. E.g. if the robot is to work a little lower, then the z value should be decreased.

8.10 Using program displacement

A program displacement is set with a *pose* data, using a *PDispSet* instruction. This will store the program displacement in a system variable, *C_PROGDISP*, holding also displacement values for external axes. The current value in *C_PROGDISP* is used in all movement instructions and added to the programmed positions. The program displacement is cleared, when a *PDispOff* instruction is executed, resulting in no further displacement.

A *PDispOn* instruction will both calculate a new program displacement, from the difference between two positions, and store this displacement in the *C_PROGDISP* variable. When this instruction has been executed a new program displacement will become active.

The following example will illustrate how to use a *PDispOn* instruction in combination with a *SearchL* instruction, to make a movement on different locations, depending on the search point.

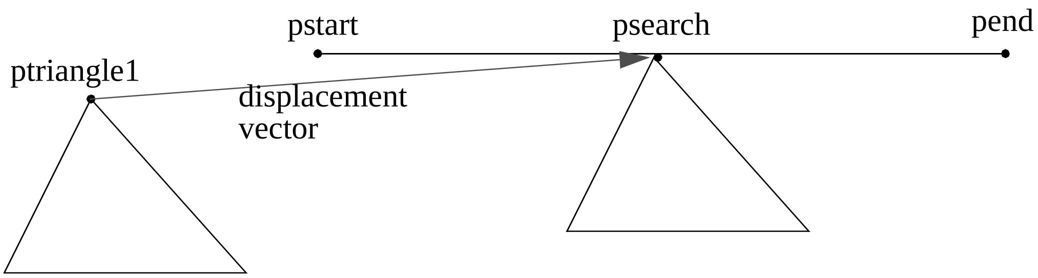
The program should do the following:

- Go to a start point, *pstart*, for searching.
- Make a linear search from the start position to an end position, *pend*. When a digital input *di1* is set, the robot should stop the movement and draw a figure, *triangle*, the position of which will depend on the search point, *psearch*.

The figure, *triangle*, is programmed with no displacement active and with the first position in *ptriangle1*.

The program may look like:

```
MoveL pstart, v200, fine, tool1;
SearchL \Stop, di1, psearch, pend, v100, tool1;
PDispOn \ExeP: = psearch, ptriangle1, tool1;
triangle;
PDispOff
etc.
```



8.11 Creating a new displacement frame

• Open the *Program Data Types* window by choosing **View: Data Types**.

• Select the type *pose* and press Enter [↵].

• Create the new displacement frame using one of the following alternatives:

- **alt 1**. Press the function key ***New***.
  The displacement frame will then have no translation or rotation.
- **alt 2**. Select an existing displacement frame and press the function key ***Dupl***.
  The displacement frame will then be the same as the one duplicated.

A window appears, displaying the name of the data.

• If you want to change the name, press Enter [↵] and specify a new name.

• Press ***OK*** to confirm.

8.12 Manually updating a displacement frame

• Open the *Program Data Types* window by choosing **View: Data Types**.

• Select the type *pose* and press Enter [↵].

• Select the displacement to be changed and press Enter [↵].

• Select the frame component (x, y, z, q1-q4) that you wish to change.

• Change the value using the numeric keyboard. To enter a decimal point (.) and minus (-), use the function keys.

• Choose ***OK*** to confirm the change.

8.13 Methods for defining a displacement frame

The following method is supported:

- ***n-point***
  At least three well-defined points on an object at its initial position and the same points when the object is in its new position (see Figure 36) are used to define the displacement frame.

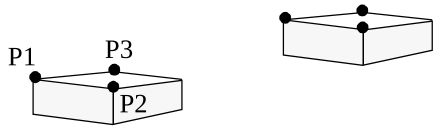


Figure 36 A displacement frame can be defined by moving the robot to a number of points.

8.14 Using the robot to change a displacement frame

- Open the *Program Data Types* window by choosing **View: Data Types**.
- Select the type *pose* and press Enter ↵ .
- Select the displacement frame to be defined (or create a new one, see ***Creating a new displacement frame*** on page 43).
- Choose **Special: Define Coord**.

A dialog box appears, displaying the points defined by the method that was used (see Figure 37).

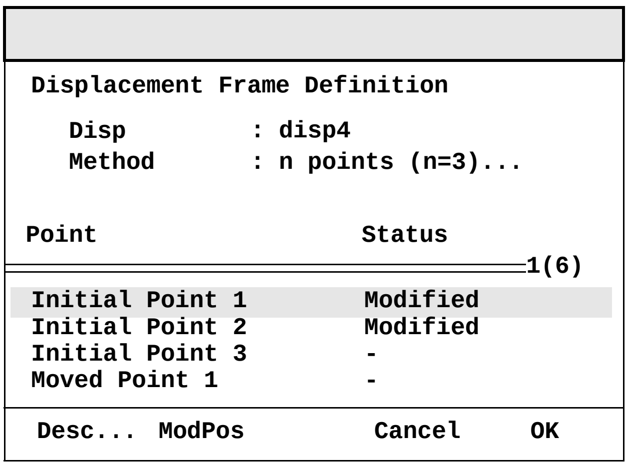


Figure 37 Displacement frame definition dialog

The status can be defined as follows:

Status	Meaning
-	No position defined
Modified	Position modified

To choose the definition method

Before you start modifying any positions, make sure the ***n-point*** method, together with the number of points that you want to use, is displayed:

- Select the field ***Method*** and press Enter ↵ .
- Enter the desired number of points and press ***OK***.

To record the Initial Points

- Select the first definition point ***Initial Point 1***.
- Jog the robot as close as possible to a well-defined position on the object.
- Modify the position by pressing the function key ***ModPos***.
- Repeat the above for the points ***Initial Point 2***, ***Initial Point 3***, etc.

To record Moved Points

- Move the object to its new position.
- Select the first definition point ***Moved Point 1***.
- Jog the robot as close as possible to the same position on the object as for ***Initial Point 1***.
- Modify the position by pressing the function key ***ModPos***.
- Repeat the above for the points ***Moved Point 2***, ***Moved Point 3,*** etc.

To calculate the displacement frame

- Press ***OK*** to calculate the displacement frame.

When the calculation is finished, a dialog like the one shown in Figure 38 will appear.

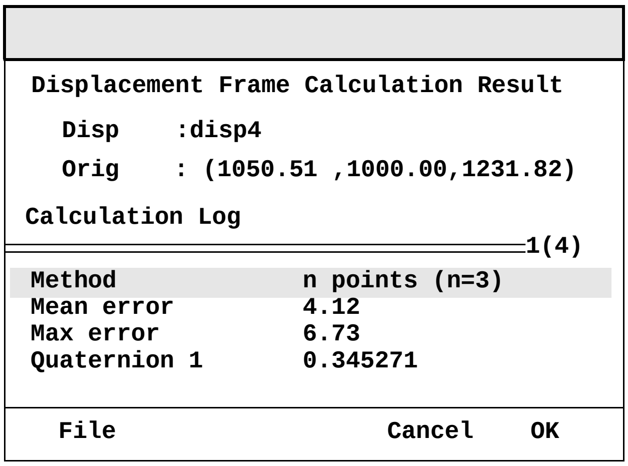


Figure 38 The result after a displacement frame calculation.

Field	Description
Orig	The origin of the displacement frame.
Mean Error	The average distance that the points are from the original points, i.e. how accurately the robot was positioned.
Max Error	The maximum error for one point.

The calculation result can be saved in a separate file for later use in a PC.
Note, however, that this file cannot be read by the robot:

- Press the function key ***File***.
- Specify a name and a place to save the result.
- Choose ***OK*** to confirm the save.

• If the estimated error is
  - acceptable, press ***OK*** to confirm the new displacement frame;
  - not acceptable, redefine by pressing ***Cancel***.