UNIVERSAL ROBOTS

Calibration manual

Version:
January 8, 2013

Contents

1 Calibration by relative known positions

This user manual goes through the calibration method for the kinematics, implemented on the Universal Robots controller from software version 1.4. The manual is organised as a step by step tutorial.

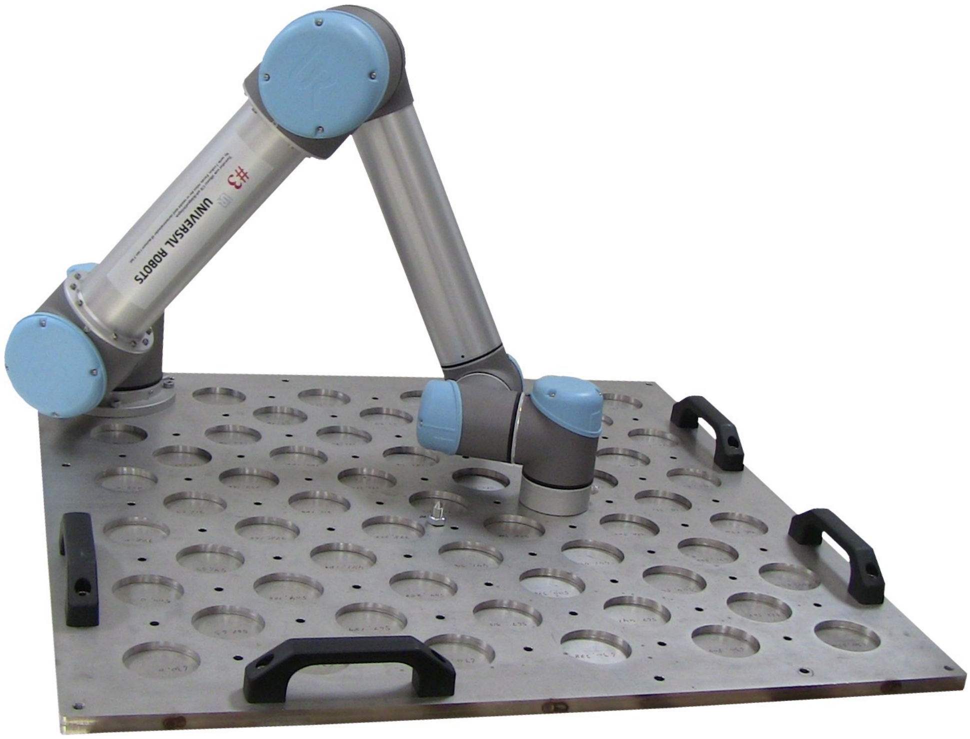

Figure 1.1: Robot mounted on a plate with holes to align the robot tool flange.

1.1 WARNING

It is important to point out that not all calibrations are good calibrations. Please pay attention to the generated statistics before saving the result of the calibration. If a calibration is not performed with due care, it is could make the robot very inaccurate.

1.2 Required equipment

The method is based on the assumption that it is possible to fixate the tool in multiple positions with the tool flange parallel to the plane. In this tutorial a stiff plate produced with holes on known positions suited to fit the the tool flanges as displayed in figure 1.1 is used, described in appendix A on page 15.

1.3 Access the functionality

The starting point for the steps of the tutorial is a screen that looks like figure 1.2.
The steps are as follows:

(1) The functionality can be found in the "Expert Mode" on the controller screen by pressing the "Kinematics Calibration" button as illustrated in figure 1.2.

(2) This enables a new tab called "Calibration" in the normal program mode. In this tab three cases can be chosen, see figure 1.3.

   (a) Program correction
   A method for adjusting programs after joint or robot replacement, see section 2 on page 7

   (b) Linearising the robot
   Calibrate the robot by known positions on plates. Press this button to begin the calibration procedure.

(c) Load an existing calibration
Makes it possible to load a previously saved calibration.

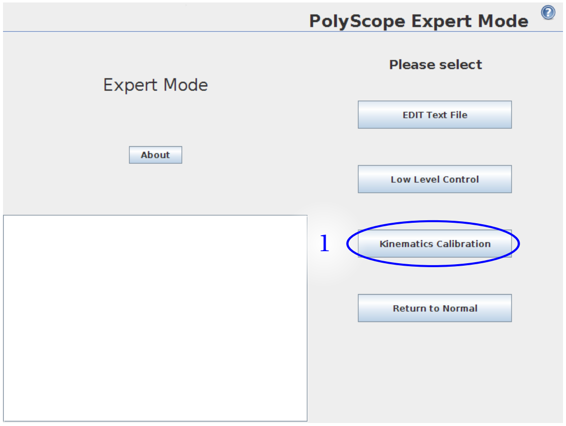

Figure 1.2: Select "Kinematics Calibration" in "Expert Mode" to get access to the calibration functionality.

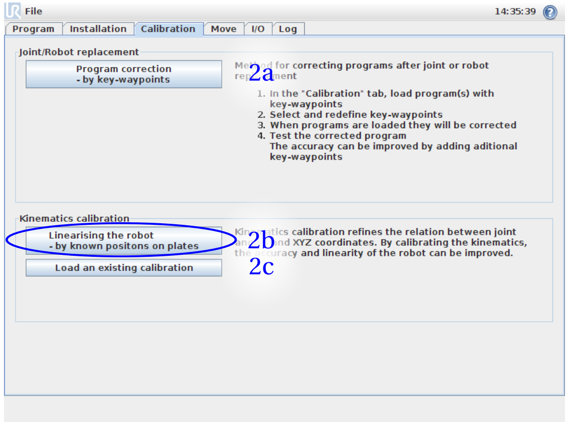

Figure 1.3: Select "Linearising the robot" to select this method.

1.4 Measure the holes

Multiple metal plates with a number of holes can be used. The holes are used to fixated the tool to a known position on the selected plate. Therefore add a plate to start adding positions.

(3) Add the first plate for the positions.

(4) Add the first position to the selected plate by pressing "Add new position".

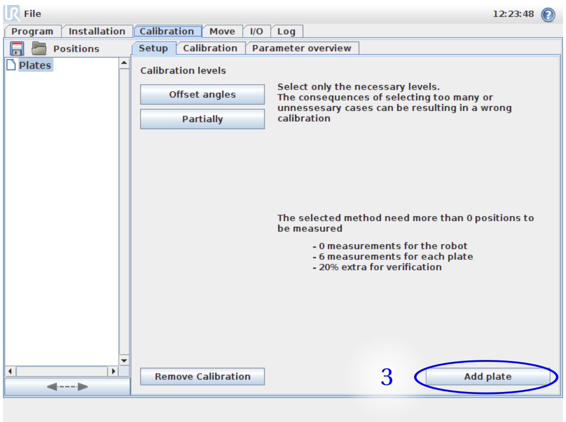

Figure 1.4: Add a plate for the hole positions.

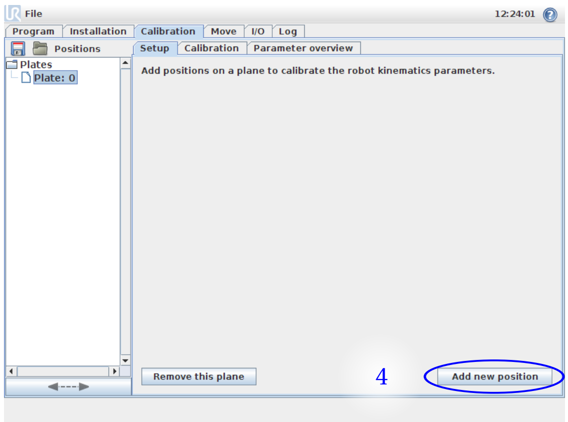

Figure 1.5: Select "Add new positions" to add a position to the selected plate.

(5) Insert the (x, y, z) coordinate relative to the plate for the position, see figure 1.6.

(6) Move the related robot configuration by pressing "Teach new position", see figure 1.6.

(7) Teach the robot to the correct position and pressing "OK" to complete one position, see figure 1.7.

(8) To continue adding multiple positions pressing "Add new position", see figure 1.6.

(9) Step no.: 5 to 8 is repeated until enough positions are added.

(10) Before starting on the calibration part it is recommenced to save the collected data. This is done by pressing on the disk in the top left corner. Please note that the save function in the file menu save robot programs and not calibrations and therefore can not be used, see figure 1.6.

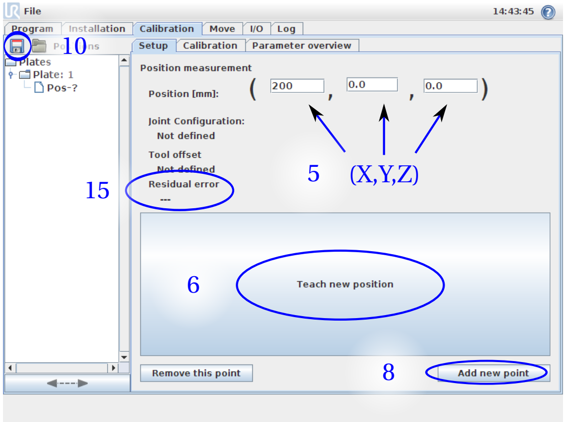

Figure 1.6: The property screen for the selected position. Insert the (x, y, z) coordinate, teach the related robot configuration. The screen has also a button to add additional positions.

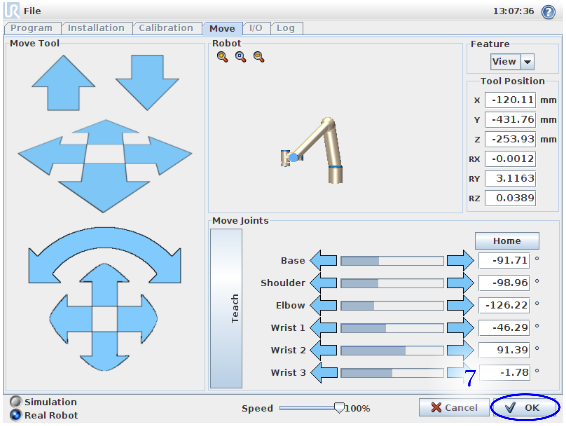

Figure 1.7: Move the robot to the correct position and press "OK".

The configuration used in the calibration needs to be as distributed as possible on the plate but also in the configuration space of the robot. Ensure that all the combinations of both elbow and wrist up and down solutions are used, see figure 1.8.
It is recommended that a full calibration at least includes 30 positions on each plate. To add additional plate select the top of the tree item "Positions" on the left and repeat the steps from 3.

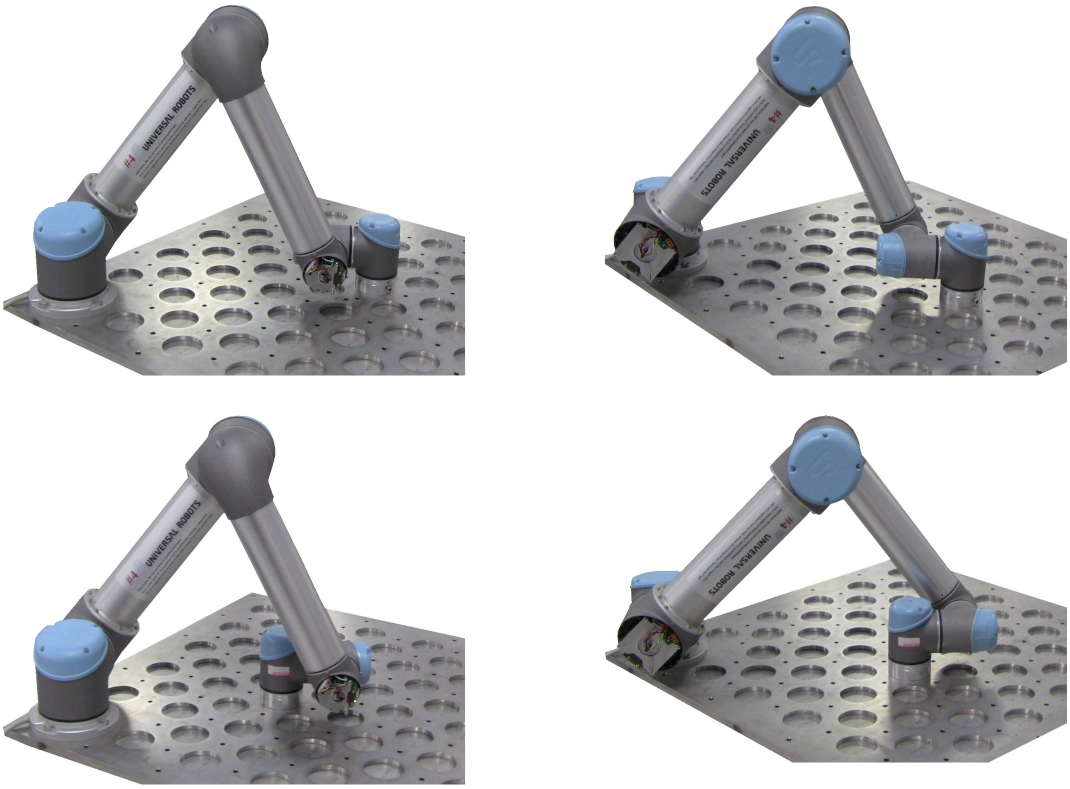

Figure 1.8: Configuration need to be used.

1.5 Calibration

Now all the needed positions for the calibration are added to the tree. In this example 26 Positions on one plate are used.

(11) Select the top of the tree item "Positions" on the left to be able to select the level of calibration, see figure 1.9.

(a) Offset angles
This level calibrates the offsets angles for the shoulder, elbow, wrist 1 and wrist 2. The method does not provide enough information to calibrate the base and wrist 3.

(b) Partially
This level calibrates more parameters than 11a to cater for other deviations than offset angles. It is not a full calibration, but as full as the nature of the method allow it to be.

(12) The calibration is started from the "Calibration" tab, see figure 1.10.

(13) Start the calibration by press "Make Calibration", see figure 1.10.

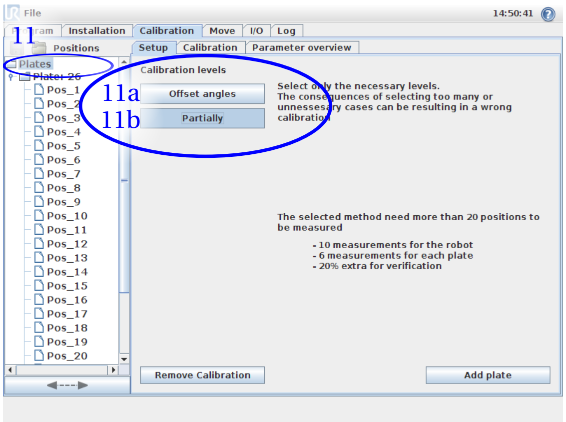

Figure 1.9: Select the top of the tree item on the left to be able to select the level of calibration.

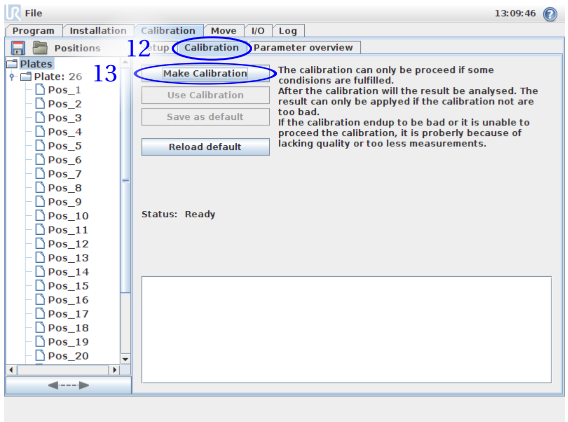

Figure 1.10: Select the calibration tab and press "Make Calibration" to start the calibration.

1.6 Evaluation

(14) After the calibration has been performed the results needs to be evaluated to ensure that the calibration ended successfully.

The results from the calibration are the parameters displayed in the "Parameter overview" tab and the calculated statistics of the archived precision for the given positions. The last part is displayed lower part of figure 1.11 and is used to evaluate the calibration.

The calculated statistics is divided into to groups. Statistics for positions used in the calibration, called "Measurement data", and Statistics for positions reserved for only the evaluation called "Evaluation data" and shall be used to control the calibration results.

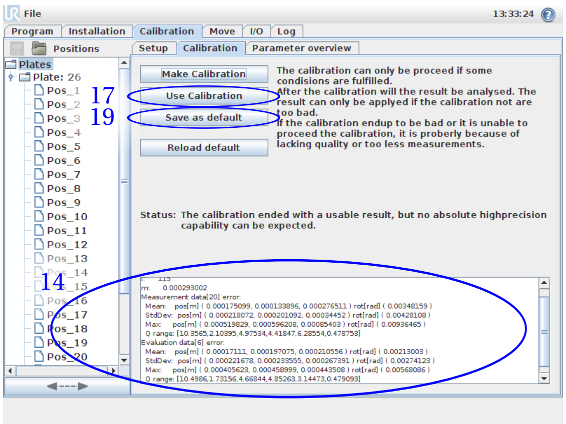

Figure 1.11: Then the calibration is done, the result needs to be evaluated, it is afterwards possible to try the new parameters and apply them to the robot.

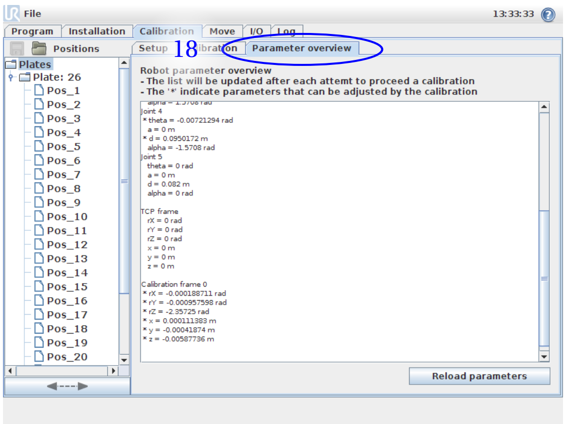

Figure 1.12: By the "Parameter overview" the parameter for the calibration can be inspected.

1.6.1 Explanation of the printed statistics

An example of the printed statistics:

```
1  ##############
2  I:      118
3  m:      0.000292995
4  Measurement data[20] error:
5    Mean:    pos[m] ( 0.000175164, 0.000134324, 0.00027655 ) rot[rad] ( 0.00348157 )
6    StdDev:  pos[m] ( 0.000218127, 0.000201774, 0.000344546 ) rot[rad] ( 0.00428109 )
7    Max:     pos[m] ( 0.000518104, 0.000596791, 0.000853992 ) rot[rad] ( 0.00936474 )
8    Q range: [ 10.3565, 2.10395, 4.97534, 4.41847, 6.28554, 0.478753 ]
9  Evaluation data[6] error:
10   Mean:    pos[m] ( 0.000171146, 0.00019818, 0.000210623 ) rot[rad] ( 0.00212999 )
11   StdDev:  pos[m] ( 0.000221324, 0.00023433, 0.000267463 ) rot[rad] ( 0.00274118 )
12   Max:     pos[m] ( 0.000405676, 0.000459457, 0.000443527 ) rot[rad] ( 0.00568077 )
13   Q range: [ 10.4986, 1.73156, 4.66844, 4.85263, 3.14473, 0.479093 ]
```

Listing 1.1: Example of printed statistics for a partially calibration

Explanation of line number:

1 This line is used to indicate that the calibration is running. *Not important.*

2 The used iterations finding the solution. *Not important.*

3 The sum of the calibrations residual error. *Not important for the normal user.*

4 to **8** is the statics for the positions used in the calibrate. In this case is the calibration based on 20 positions.

9 to **13** is the statics for the positions reserved for evaluation only. In this case 6 positions. If there is a big differentce between the results of the evaluation and the measurement data, the calibration did not succeed and the positions needs to be verified.

5 and **10** shows the mean error. First with position error as a vector(x,y,z) follow by the rotation error.

6 and **11** shows the standard deviation of the error. formatted like the mean error.

7 and **12** shows the maximum error in the data. formatted like the mean error.

8 and **13** are the range of the used robot configuration. The range can give an indication of how well the configuration are distributed.

The printed statistics is also saved in *calibration.log* file.

1.6.2 Debug

To pinpoint the source for the maximum error each positions residual error contribution is displayed in each positions property screen, see figure 1.6.

(15) If the evaluation did not succeed. The defined positions needs to be examine that the correct coordinate is given for the learned robot configuration.

(16) If the configuration used is too monotonous can also result in a unsuccessful calibration. To fix this additional positions needs to be added.

1.6.3 Apply the calibration

When a satisfied calibration has been obtained it is recommended to try the new parameters before they are stored permanently.

(17) Try the new calibration without saving them as the default by press "Use Calibration", see figure 1.11.

The current default kinematics can be reloaded with the "Reload default". This button is not a "load factory default" button. So please be aware of the quality of the calibration before applying them as permanently.

(18) The found parameters can be viewed in the "Parameter overview" tab, see figure 1.12.

(19) Save the calibration for future use by pressing "Save as default", see figure 1.11.

(20) The Robot is know calibrated.

1.7 Reset calibration

The calibration can manually be adjusted or reset by editing the */root/.urcontrol/calibration.conf* file placed together with the other configurations. To reset the calibration all decimal and hex numbers is reset to zero like the file listing 1.2.

```
[mounting]
delta_a = [0, 0, 0, 0, 0, 0]
delta_d = [0, 0, 0, 0, 0, 0]
delta_alpha = [0, 0, 0, 0, 0, 0]
delta_theta = [0, 0, 0, 0, 0, 0]
encoder_sin_excentricity = [0, 0, 0, 0, 0, 0]
encoder_cos_excentricity = [0, 0, 0, 0, 0, 0]
part_d_calib_voltage_low_speed_front = [0, 0, 0, 0, 0, 0]
part_d_calib_voltage_low_speed_back = [0, 0, 0, 0, 0, 0]
part_d_calib_voltage_high_speed_front = [0, 0, 0, 0, 0, 0]
part_d_calib_voltage_high_speed_back = [0, 0, 0, 0, 0, 0]
part_d_calib_current_low_speed_front = [0, 0, 0, 0, 0, 0]
part_d_calib_current_low_speed_back = [0, 0, 0, 0, 0, 0]
part_d_calib_current_high_speed_front = [0, 0, 0, 0, 0, 0]
part_d_calib_current_high_speed_back = [0, 0, 0, 0, 0, 0]
joint_checksum = [0, 0, 0, 0, 0, 0]
calibration_status = 0 # 0 == notInitialized / 1 == notLinearised / 2 == Linearised
```

Listing 1.2: calibration.conf file filled with zeros

2 Program correction by key-waypoints

This step-by-step tutorial user manual describes how to correct a program from one uncalibrated robot to another by redefine key-waypoints from saved programs. Either if a single joint or a whole robot has been replaced.

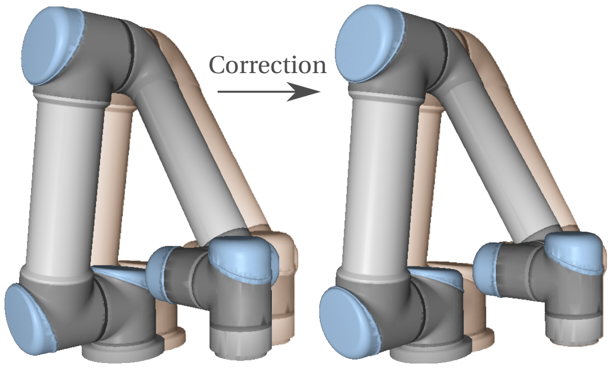

Figure 2.1: Illsutration of the correction

The function exists on the Universal Robot controllers from software version 1.7.

2.1 Introduction

With properly selected and redefined key-waypoints, it is possible to make a model which describes the difference between the old and the new robot, and use this model to correct programs. The model is specific for each installation file on the robot. Key-waypoints are not shared between models. After the model has been built, the programs are corrected when loaded. The model can be extended/improved at any time by defining more key-waypoints.

In general is the quality of the model is given by the number of key-waypoints and the accuracy with which they are defined. If further correction is desired, the model can be improved by adding more key-waypoints at a later time.

The accuracy of a corrected waypoint correlates with the quality of the model, and the distance to the nearest key-waypoint.

2.2 Access the functionality

The starting point for the tutorial steps is the expert screen as shown in figure 2.2.
The steps are as follows:

(1) The functionality can be found in the "Expert Mode" on the controller screen by pressing the "Kinematics Calibration" button as illustrated in figure 2.2.

(2) This enables a new tab called "Calibration" in the normal program mode. In this tab is three options available. Chose "Program correction - by key-waypoints", see figure 2.3.

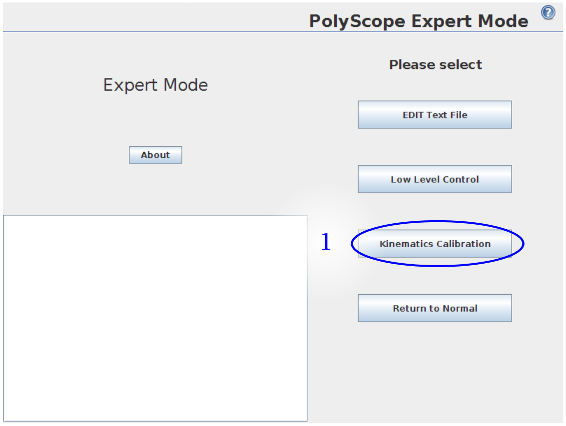

Figure 2.2: Select "Kinematics Calibration" in "Expert Mode" to get access to the functionality.

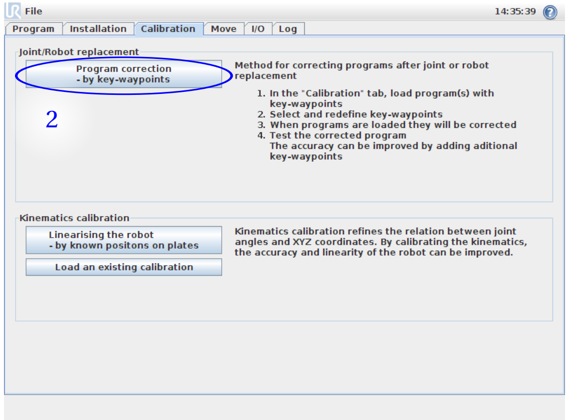

Figure 2.3: Select "Program correction - by key-waypoints".

2.3 Redefine key-waypoints

The program used in this tutorial is a simple pick and place program with two key-waypoints, the waypoint for the pick and place positions.

(3) The chosen program can now be loaded by press "Add program", see figure 2.4.

(4) Select one of the decided key-waypoints in the program. The waypoint called "pick" is selected in figure 2.5.

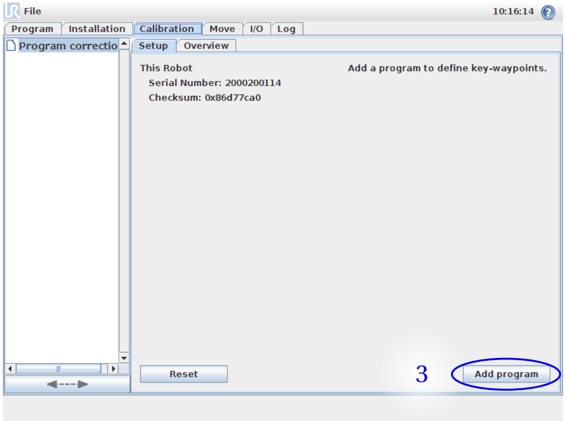

Figure 2.4: Select "Add program" to import a program.

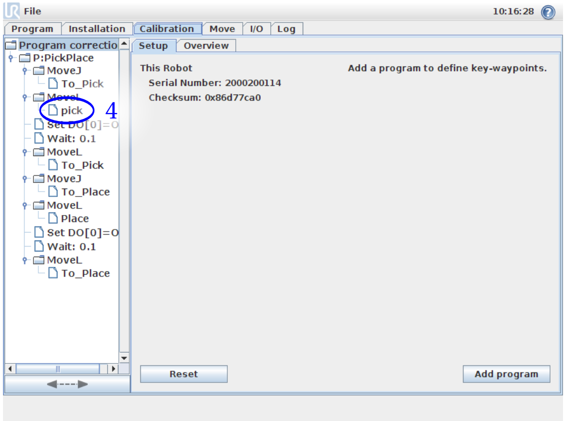

Figure 2.5: Select one of the key-waypoints. In this case waypoint named "pick"

(5) Select "New configuration" to redefine the configuration for the selected waypoint as seen in figure 2.6.

(6) Move the robot to the new position and press "OK", see figure 2.7.

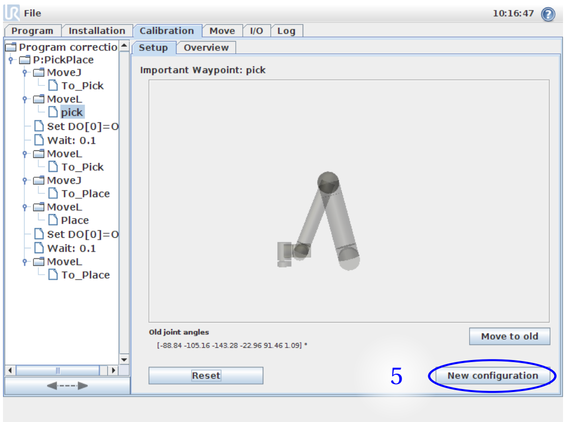

Figure 2.6: Select "New configuration" to redefine the configuration for this waypoint.

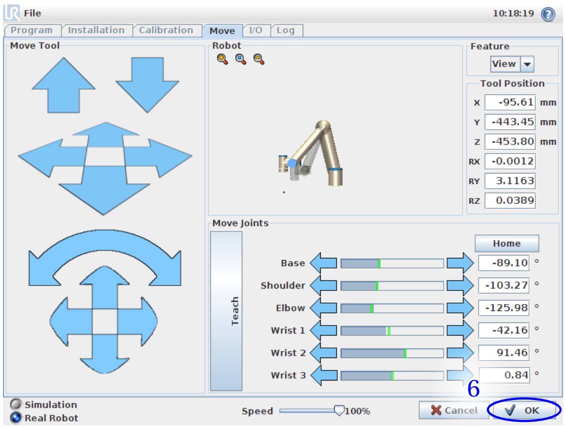

Figure 2.7: Move the robot to the position and press "OK".

2.3.1 Corresponding Tool Position

To help the method, it is important to adjust the Corresponding Tool Position, CTP, to the offset that is visually used for the comparison during the redefining of the waypoint. The CTP can be defined individual for each key-waypoint and improve the method's accuracy. As default is the tool center point from the program's installation used.

Typically CTP locations:

- At the tools center point when the robot are going to grasp an object.
- End of the object when the object is going to be placed.

(7) Specify the Corresponding Tool Position by press "Change CTP", see figure 2.8.

(8) Change the CTP coordinates and press OK, see figure 2.9.

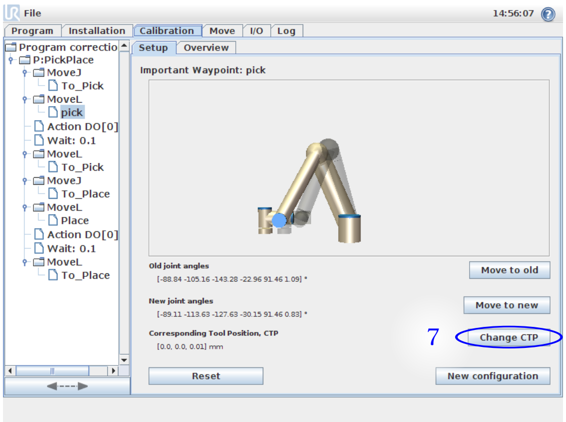

Figure 2.8: Change the Corresponding Tool Position by press "Change CTP"

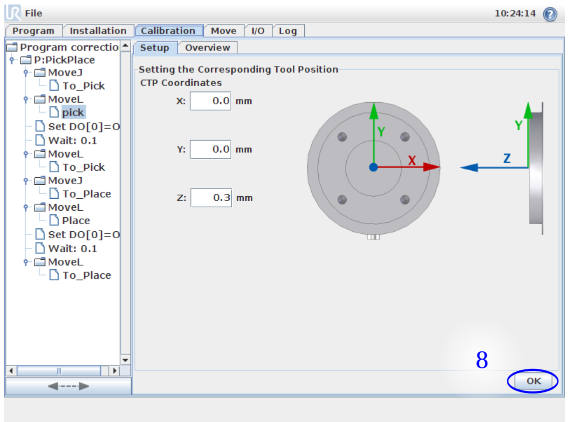

Figure 2.9: Change the CTP coordinates and press OK

(9) This completes redefining the "pick" key-waypoints. Continue by repeating step 4 to 8 until all key-waypoints are redefined.

2.3.2 Waypoints from multiple programs

It is possible to add key-waypoints from multiple programs. This is done by adding additional programs and afterwards select and redefine as previously described in step 2.5 to 9.

(10) If the key-waypoints are distributed over multiple programs, select the root node of the program tree and repeat from step 3, see figure 2.10.

(11) Add an additional program by press "Add program", as in step 3 and repeat from step 2.5, see figure 2.11.

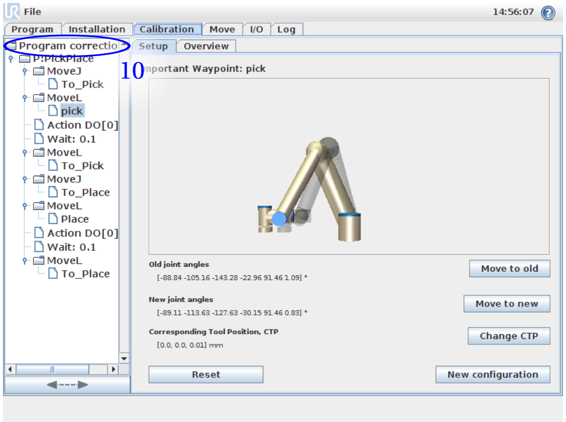

Figure 2.10: Multiple programs can be added by selecting the root of the program tree and repeat from step 2.5.

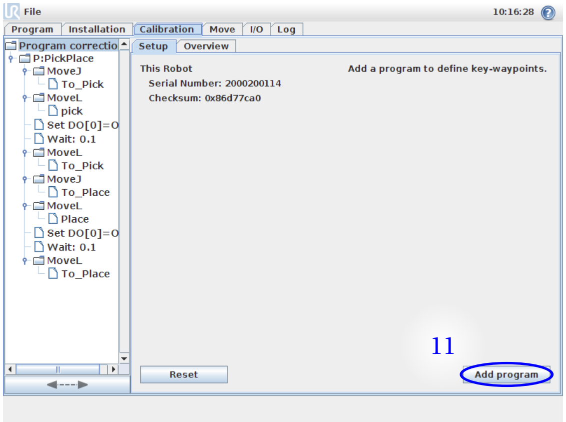

Figure 2.11: Add an additional program by press "Add program" and repeat from step 3.

2.4 Handle key-waypoints

An overview of the redefined key-waypoints is showed in the "Overview" tab, see figure 2.12. The key-waypoints are grouped by the source robot's relationship to this robot and installation. Each key-waypoint is marked with its name and the program it is coming from as shown in figure 2.12 and 2.13. It is possible to delete key-waypoints from the database by selecting a waypoint or a group of waypoints and press "Delete".

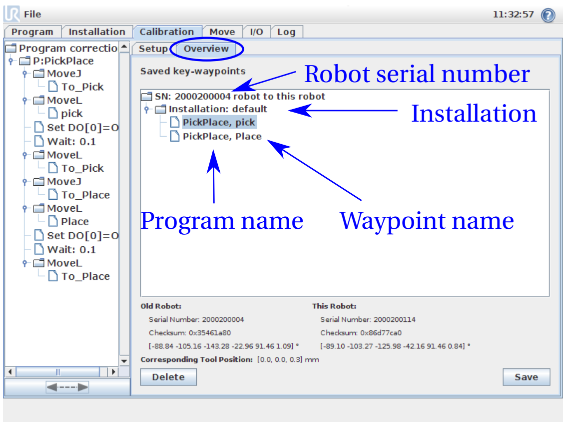

Figure 2.12: The overview tab

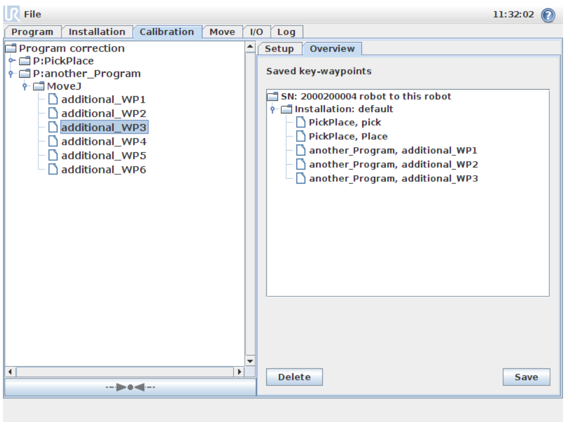

Figure 2.13: Waypoints from multiple programs can be added and displayed in the "Overview" tab

2.5 Correct a program

After the key-waypoints are redefined, the programs can be corrected during a normal program loading.

(12) Load the program to be corrected, see figure 2.14.

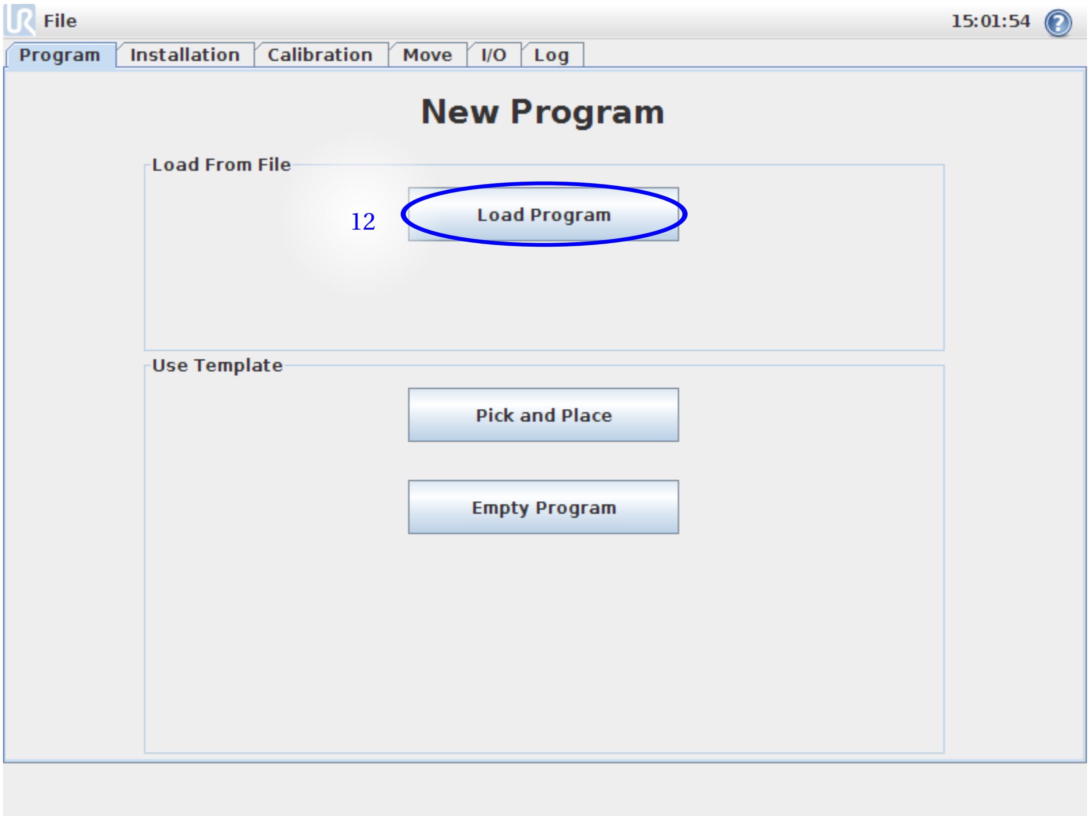

Figure 2.14: Load the program which are going to be corrected

(13) The robot controller realizes that the program comes from a different robot and asks whether you want to correct the program. To correct the program press "yes" as seen in figure 2.15.

(14) Another popup tells when the corrected is done. As the correction can take some time, please be patient, see figure 2.16. If the correction failed, please verify your key-waypoints and improve their accuracy.

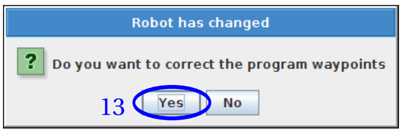

Figure 2.15: To correct the program press "yes"

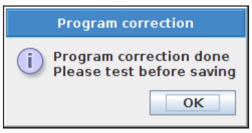

Figure 2.16: Correction done

(15) After the program is corrected, it is recommended that you test before you overwrite the old program as is indicated in figure 2.17. If a better accuracy is needed, add additional key-waypoints and repeat from step 12.

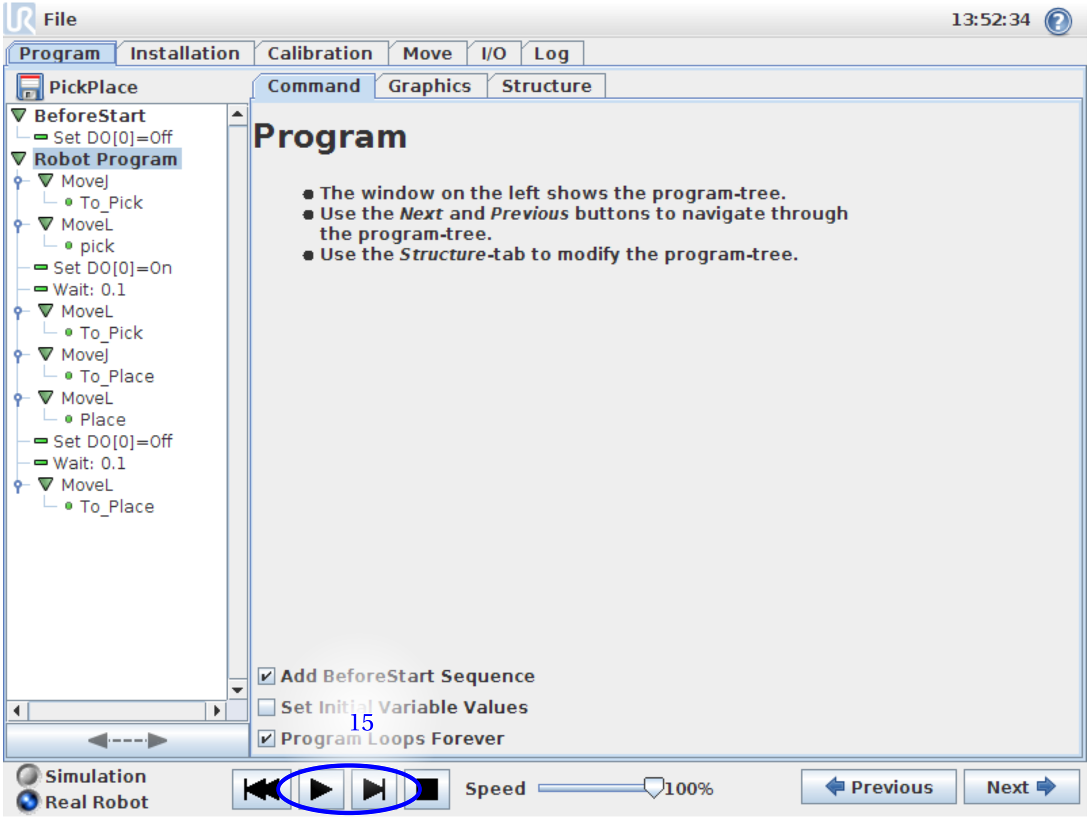

Figure 2.17: Test the program before overwrite the old program

(16) Save the program when the program is tested and works as intended.

(17) Correct other programs by repeating from step 12. Note that the key-waypoints are stored across programs, thus helping other programs to be corrected.

A Calibration board

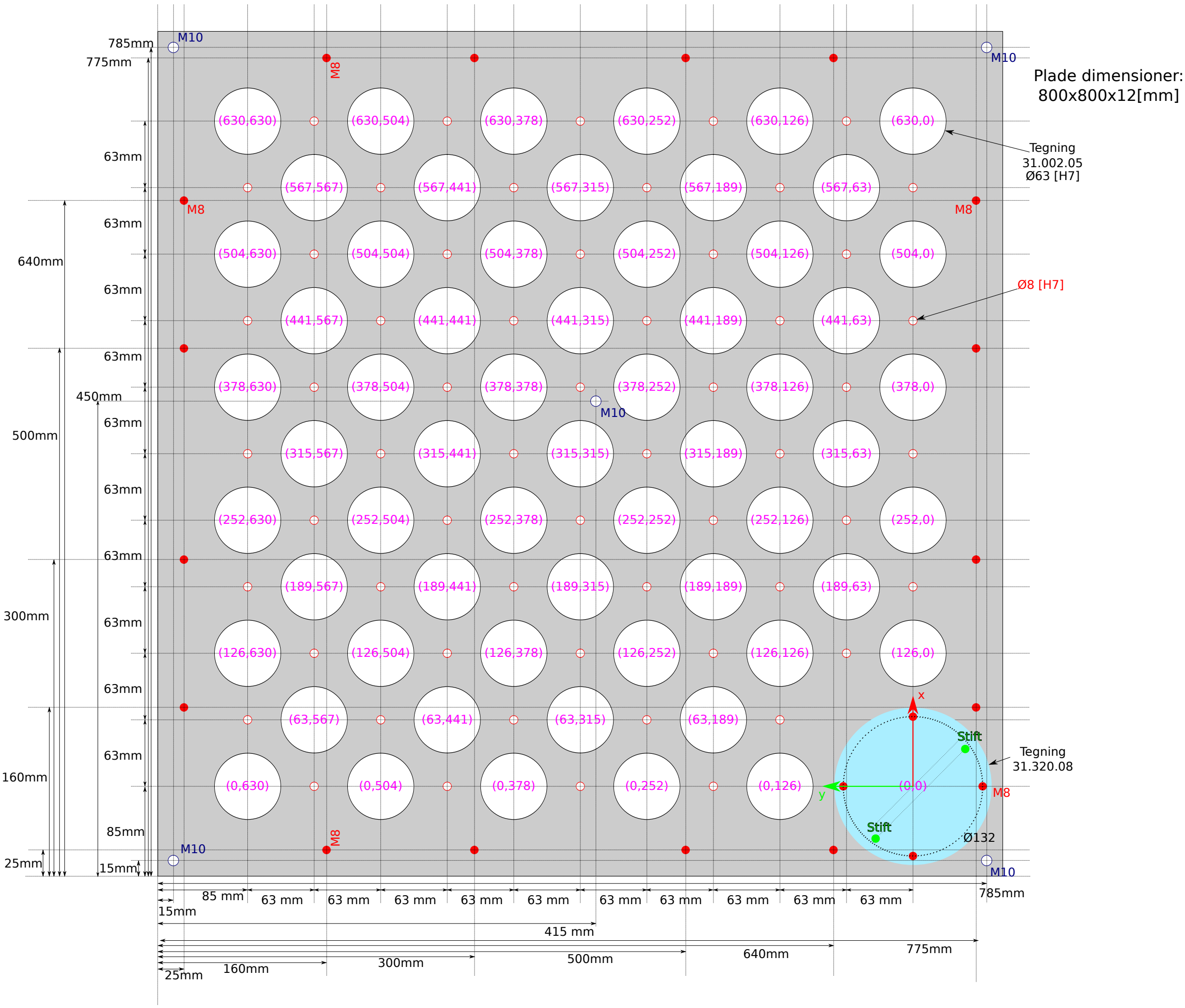

Figure A.1: Drawing of the calibration board used for calibrating the robot by the known point on surface method.

B Robot Parameter

B.1 UR5

The robots kinematic transformations for each link are given by Denavite-Hartenberg parameters. The parameters for UR5 is displayed in tabel B.1 to B.3.

	θ $[rad]$	a $[m]$	d $[m]$	α $[rad]$
Joint 1:	0	0	0.08920	$\frac{\pi}{2}$
Joint 2:	0	-0.42500	0	0
Joint 3:	0	-0.39243	0	0
Joint 4:	0	0	0.10900	$\frac{\pi}{2}$
Joint 5:	0	0	0.09300	$\frac{-\pi}{2}$
Joint 6:	0	0	0.08200	0

Table B.1: Denavit-Hartenberg parameters for the UR5 **serie 1**.

	θ $[rad]$	a $[m]$	d $[m]$	α $[rad]$
Joint 1:	0	0	0.08920	$\frac{\pi}{2}$
Joint 2:	0	-0.42500	0	0
Joint 3:	0	-0.39225	0	0
Joint 4:	0	0	0.11000	$\frac{\pi}{2}$
Joint 5:	0	0	0.09475	$\frac{-\pi}{2}$
Joint 6:	0	0	0.08250	0

Table B.2: Denavit-Hartenberg parameters for the UR5 **serie 2**.

	θ $[rad]$	a $[m]$	d $[m]$	α $[rad]$
Joint 1:	0	0	0.089159	$\frac{\pi}{2}$
Joint 2:	0	-0.42500	0	0
Joint 3:	0	-0.39225	0	0
Joint 4:	0	0	0.10915	$\frac{\pi}{2}$
Joint 5:	0	0	0.09465	$\frac{-\pi}{2}$
Joint 6:	0	0	0.08230	0

Table B.3: Denavit-Hartenberg parameters for the UR5 **serie 3**.

B.2 UR10

	θ $[rad]$	a $[m]$	d $[m]$	α $[rad]$
Joint 1:	0	0	0.118	$\frac{\pi}{2}$
Joint 2:	0	-0.6127	0	0
Joint 3:	0	-0.5716	0	0
Joint 4:	0	0	0.1639	$\frac{\pi}{2}$
Joint 5:	0	0	0.1157	$\frac{-\pi}{2}$
Joint 6:	0	0	0.0922	0

Table B.4: Denavit-Hartenberg parameters for the UR10 robot **serie 1**.