Introduction

This is a user guide to perform the calibration of the robot. It consists of two steps that have to be performed in the provided order:

1. Intrinsic camera calibration
2. Check the initial calibration status
3. Eye-hand calibration procedure

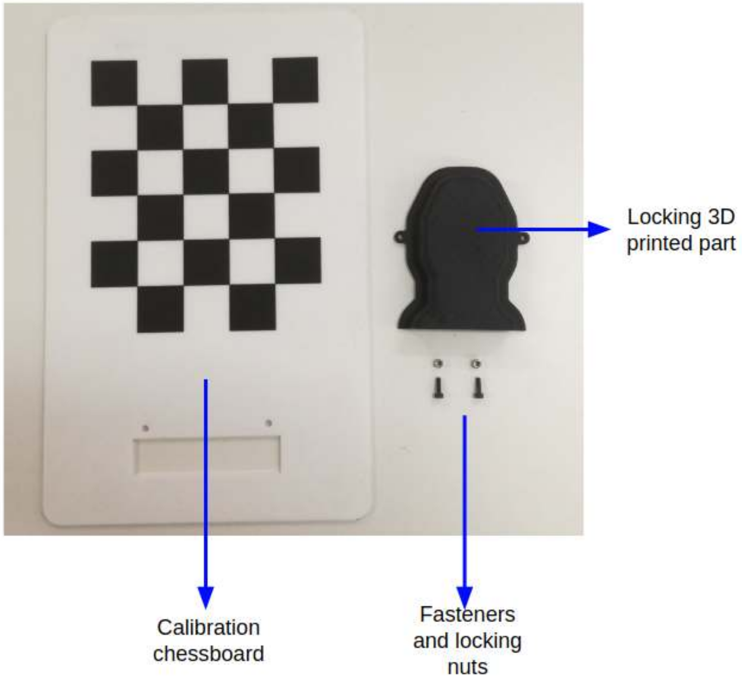
Prerequisites

Here the hardware and software need to successfully perform the outlined procedure as described.

Hardware requisites

The following components are required in order to do the eye-hand calibration. You should have received a kit with three items:

1. A calibration chessboard
2. Locking 3D printed part for the PAL Gripper
3. Fasteners for the previous.



Software requisites

Here one has to check whether all required packages are installed in the robot.

For TIAGo

1. tiago_description_calibration
2. pal_eye_hand_calibration_tiago

For TIAGo++

1. tiago_dual_description_calibration
2. pal_eye_hand_calibration_tiago_dual

To do so first connect to the robot via ssh:
ssh pal@tiago-Xc

Recall that here *X* stands for the serial number of your TIAGo robot, without leading 0s.

Once in the robot, you can check whether these packages are installed by doing:
dpkg -l | grep calibration

You should see, listed:

For TIAGo

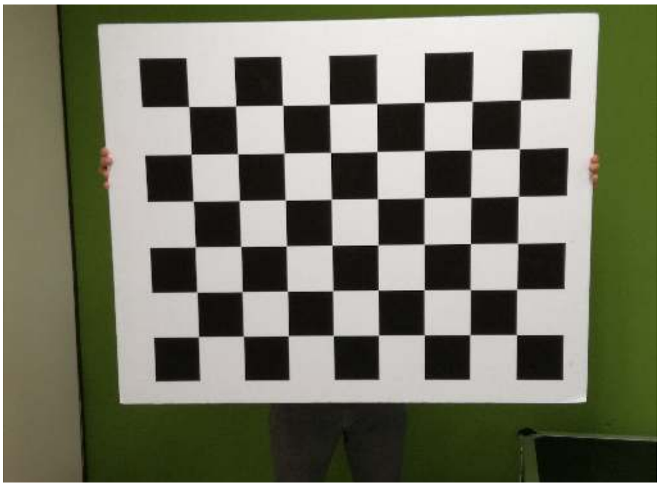
pal-<distro>-tiago-description-calibration
pal-<distro>-pal-eye-hand-calibration-tiago

For TIAGo++

pal-<distro>-tiago-dual-description-calibration
pal-<distro>-pal-eye-hand-calibration-tiago-dual

Intrinsic calibration

First of all, it is necessary to calibrate the intrinsic parameters of the camera. For this purpose it is recommended to have a big chessboard like the one shown below:



In the above example, the size of the chessboard cells is 10.8 cm and there are 8x6 intersections.

In order to calibrate the intrinsics, connect to the robot with X server support, and run the following commands:

ssh -X pal@tiago-Xc
pal-stop head_manager
rosrun camera_calibration cameracalibrator.py --size 8x6 --square 0.108 image:=/xtion/rgb/image_raw camera:=/xtion/rgb --approximate=0.1 _image_transport:=compressed

WARNING: make sure to adapt the last command to account for the right dimensions of your chessboard.

Follow the instructions in the link:
http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration

WARNING: Remember to push the button **Calibrate** and when the calibration is computed (it will takes few minutes) push **Commit** to save the calibration data in the default ros directory and file, i.e. */home/pal/.ros/camera_info/rgb_xtion.yaml*, before closing the node.

Check the calibration status

Unfold the arm of the robot by running first the offer motion. You can either do it via Web Commander in the Movements Tab, or by connecting to the robot and running the corresponding motion:
ssh pal@tiago-Xc
rosrun play_motion run_motion offer (TIAGo)
rosrun play_motion run_motion offer_both (TIAGo++)

Wait until the execution is finished and then stop the head_manager. Do it from the Web Commander in the Startup tab or by running the following command in the robot:
pal-stop head_manager

Use the following calibration motion to put the robot arm(s) in the so-called *check_calibration* position from inside the robot:

For TIAGo

rosparam load `rospack find pal_eye_hand_calibration_tiago`/config/tiago_calibration_motions.yaml
rosrun play_motion run_motion check_calibration

For TIAGo++

rosparam load `rospack find pal_eye_hand_calibration_tiago_dual`/config/tiago_dual_calibration_motions.yaml
rosrun play_motion run_motion check_left_calibration
rosrun play_motion run_motion check_right_calibration

Then from your development computer use RViZ to verify the calibration:
export ROS_MASTER_URI=http://tiago-Xc:11311
rosrun rviz rviz -d `rospack find tiago_bringup`/config/tiago.rviz

Take a snapshot of the “RGB-D camera” viewer where the model of the end-effector will appear overlaid on the actual end-effector. You may disable the “Point cloud” viewer to better see the match between the real end-effector and the overlaid one.

Move the end-effector slightly to have another point of view. You can do it connected to the robot via ssh:

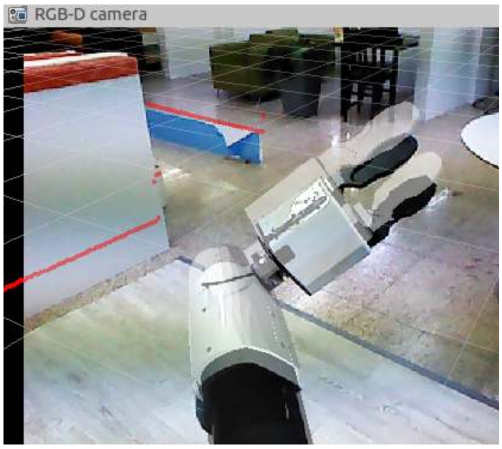
For TIAGo

rosrun play_motion move_joint arm_7_joint 0.02 0.5

For TIAGo++

rosrun play_motion move_joint arm_left_7_joint 0.02 0.5
rosrun play_motion move_joint arm_right_7_joint 0.02 0.5

Take another snapshot with this configuration. The snapshots should look like as below:



A misalignment of the end-effector between the model and the actual one is clear in the figure above.

Eye Hand Calibration

Mounting the chessboard

The Eye-hand calibration procedure should be done with the PAL Gripper. Please change the end-effector to the PAL Gripper following the instructions of the user manual.

First move the robot to a suitable pose to mount the chessboard. Put the robot in *offer* position as done above for the calibration check: either from the Web Commander or by connecting first to the robot and running the offer motion:

ssh pal@tiago-Xc
rosrun play_motion run_motion offer (TIAGo)
rosrun play_motion run_motion offer_both (TIAGo++)

After this, close the gripper with the following motion:

For TIAGo
rosrun play_motion run_motion close_gripper

For TIAGo++
rosrun play_motion run_motion close_left *rosrun play_motion run_motion close_right*

Or via the corresponding button in the Web commander. Feel free to move the 7th joint of the arm to put it in a position that eases the mounting of the rest of the parts.

Then mount the chessboard as described below.

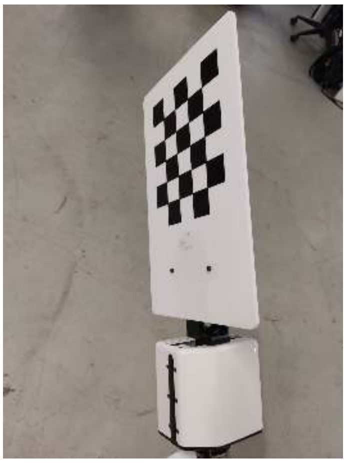
1. For the gripper of interest, reduce the current from a development PC:
   export ROS_MASTER_URI=http://tiago-Xc:11311
   rosrun rqt_current_limit_controller rqt_current_limit_controller
   And set the limit current of the corresponding controller to its minimum value, as shown below

Note that for a single armed TIAGo, only a single slider will appear.

2. This will allow you to open the gripper freely.
   Attach the printed part in between the gripper fingers:

3. After this is done, close the gripper bamanually and set back the current limit to its maximum value in *rqt_current_limit_controller*

4. Attach the chessboard using a wrench and an allen key



For TIAGo

The chessboard should face the left hand side of the robot

For TIAGo++

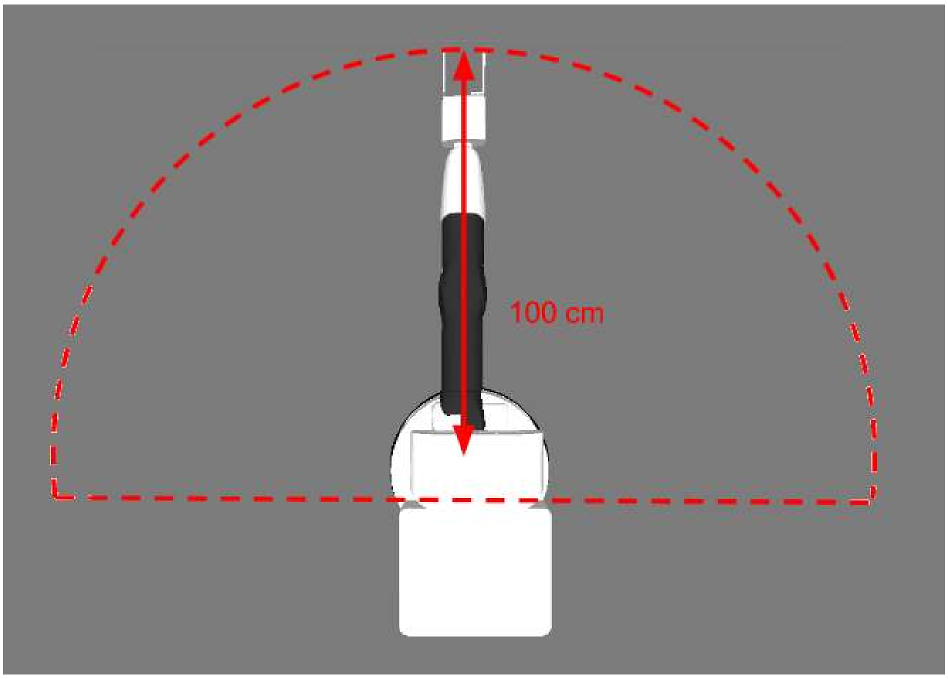
The chessboard should be aimed as shown in the two pictures below

WARNING: The chessboard should not move from inside of the gripper, otherwise the calibration will have no sense. Make sure that the fasteners are tightly fixed.

WARNING: For TIAGo++, with the WebCommander move the arm_1 joint of the arm you are not calibrating in a way that it cannot collide with the other arm. Do as shown in the picture above.

Calibration recorder

To record the calibration data make sure to move the robot to a place with a large empty area around to prevent collisions. Use the following sketch as a guide:



First of all, disable the **head_manager** node, which commands the robot's head, as follows:
ssh pal@tiago-Xc
pal-stop head_manager

Then run the calibration recorder as follows. Remember to specify the name of the recorded data and the correct end effector, which in this case is always *gripper*.

For TIAGo

roslaunch pal_eye_hand_calibration_tiago tiago_calibration_recorder.launch recorder_name:=test end_effector:=gripper

For TIAGo++

For the left arm:

For TIAGo

roslaunch pal_eye_hand_calibration_tiago_dual tiago_dual_calibration_recorder.launch side:=left end_effector:=gripper recorder_name:=test_left_1

For the right arm:
Wait until the procedure has finished for the left arm and then mount the chessboard in the end and then follow again the steps outlined above. After this, run:
roslaunch pal_eye_hand_calibration_tiago_dual tiago_dual_calibration_recorder.launch side:=right end_effector:=gripper recorder_name:=test_right_1

WARNING: The default ros folder is **~/.ros** . The *recorder_name* will be saved in this folder.

The calibration recorder will move the robot to 500 different poses. This process takes about 1h. When the recording is completed the following messages will appear in the console:

```
[ INFO] [1510220122.672553836]: Finished recording calibration data
[pal_robot_calibration_recorder-2] process has finished cleanly
log file:
/tmp/ros_logs/341b9aa6-c520-11e7-b275-80193476cfcd/pal_robot_calibration_recorder
-2*.log
```

Press CTRL+C in order to terminate the node.

Launching calibration estimator

Once the iterations have finished, you must launch the calibration estimator to keep the values recorded:

For TIAGo

roslaunch pal_eye_hand_calibration_core calibration_estimator.launch recorder_name:=test robot:=tiago

For TIAGo++

roslaunch pal_eye_hand_calibration_core calibration_estimator.launch recorder_name:=test_left_1 recorder_name_2:=test_right_1 robot:=tiago_dual

Example output:

```
Estimated camera position: 0.0469328 0.0681747 0.0148909
Original camera position: 0.0908 0.08  0
Estimated camera orientation RPY:     -1.5578 0.00626016  0.0182208
Original camera orientation: -1.57 0 0
*********
Estimated tip Position: 2.49622e-05 0.0167143  0.185674
Original tip position:    0   -0 0.19
Estimated tip orientation RPY:  1.54626     0.025915 -0.00531544
Original tip orientation: 1.57 0 0
joint offsets:
  arm_1_joint: -0.0350499
  arm_2_joint: 0.0260576
  arm_3_joint: 0.00495167
  arm_4_joint: 0.0442725
  arm_5_joint: -0.0418086
  arm_6_joint: 0.0366973
  arm_7_joint: 0.00238509
```

```
head_1_joint: 0
head_2_joint: 0.0121784
```

After the results appear, the following message will keep being printed:

```
[INFO] [1510660349.464763243]: Waiting for subscriber
```

Press CTRL+C to terminate the calibration estimation node in the robot. The node will generate also the file with the calibration parameters in:
~/.ros/[name of the recorder]/calibration_constants.urdf.xacro

IMPORTANT: Unmount the calibration chessboard before proceeding with the following steps.

For the changes to have an effect, you must do the following command:
pal_restart_deployer

And after that, reboot the robot. Then, put the robot back in home position and proceed to check the calibration results.

Verifying the calibration

Proceed as in the Check the calibration status section.