Evaluation of a Robot Calibration Method

Gustaf Bergström and Björn Green

May 2017

Introduction

It is essential that the robots in industry maintain a high precision in terms of position. Therefore they have to be calibrated before usage, and recalibrated after some times use. In this thesis, a calibration method [1] was extended and evaluated for a dual arm robot, see Figure 1.

Figure 1: YuMi from ABB.

The calibration

Why calibrate?

To solder a circuit, or assemble small-part electronics, require precision. It is out of the question to get different results every time the same task is performed. Unfortunately, the precision will decrease over time, because of, for example, wear and tear and the robot will need to be calibrated again to maintain a high precision. Calibrations can be performed in different ways. One way is by creating a model describing the relationship between errors in the pose (position and orientation) of the robot's tool and the errors in the kinematic parameters of the robot. The kinematic parameters are based on the physical specifications of the robot, such as link lengths and orientations, and describe the relative position and orientation between joints and links in the robot. The error model is then used by iteratively updating the values of the kinematic parameters to compensate for the errors in the tool's pose. It is also of interest to do so called joint-offset calibration. This strategy is used to compensate for offsets in the angles that represent orientation of the different links.

Today, external sensors such as advanced cameras are often used to do the calibration, and it is common to send the robot to the retailer's service center to do this calibration. It sounds expensive, and it is. Especially because of production stoppage. Luckily, the robot is filled with sensors that can be used to do this calibration.

In this thesis, a method for calibration is evaluated on a new robot from ABB, namely the IRB 14000, also called

YuMi [2]. It is a two-armed robot (most industrial robots have one arm) and each arm has an extra joint compared to the common industrial robot from ABB. This extra joint is excellent for small-part assembly and for avoiding obstacles because the robot can complete a specific task using multiple different configurations (arm positioning) precisely like a human can. This setup has its benefits, but it does make the calibration more complicated.

Method

The idea behind the calibration method is to use information about the position and orientation of the robot's hands, without using external sensors, and compare this knowledge with the logged data from the robot joints. If we fix one arm at a point, like the one in Figure 2 (compare to pressing your own palm against a table in front of you), and move the joints (try it yourself), an uncalibrated robot will look at the joint angles and think that the palm is moving, but it is not! If we can tell the robot to slightly change what it thinks is its link lengths and its link orientations, its kinematic parameters, and add or remove some number from the logged joint values using clever algorithms, we can make the robot think that the palm is fixed (which it is). The robot is now calibrated!

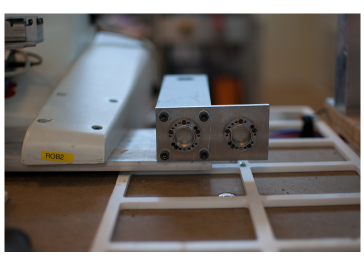

Figure 2: Fixed docking points.

Such calibration is what has been investigated in this thesis (for one fixed-point and for several fixed-points with known relative distances and orientation in between points) along with a method that uses both arms, by simply connecting those two arms as seen in Figure 3. When the arms were connected, one hand's position and orientation affected the other hand, and by knowing how the hands were fixed to each other, a similar calibration could be performed.

Figure 3: The two hands connected.

Execution

Those two described methods were used, and the joint values were logged. By using the nominal kinematic parameters, given from the producer, and the joint values, the hand's position and its orientation could be computed by using the logged joint values. This computation was performed for the logged data, and via a numerical optimization algorithm, the lengths, orientations, and joint offsets were identified.

Results

The evaluated methods turned out to work pretty well. In all cases, the robot became better to reach the same point several times. In some methods, the exact joint offsets and/or link orientation and length were found. A summary can be seen in Table 1 and here comes an explanation to the table.

If the hand is fixed to one single point and moved around, it is required that the fixed point is known exactly to find the joint offsets. Using several points with known distance and orientation in between helped a bit, but the shoulder joint, the one closest to the robot's body, was still unidentifiable. This problem is because finding out where the single point of the robot was fixed at was part of the optimization and was computed to be at the wrong position (and not the actual one) and therefore made the joint offset identification difficult.

When the hands were connected and moved around, the exact kinematic parameters were not found. This result is because of that one arm is used as measurement for the other. The only thing that is computed is the error between the hands, and then the error is divided equally. This leads to a kind of mean calibration and the exact parameters were therefore not found. However, the exact joint offsets were found using a combination of the several fixed-point implementation and the dual-arm implementation, if all kinematic parameters are known. If one arm was considered calibrated in the dual-arm identification, all kinematic parameters were found. This method could also compensate for both offset and kinematic parameter errors in the uncalibrated arm to minimize the end-effector position and orientation error. The correct values were not identified with this strategy, however.

In no case, it was possible to identify the correct kinematic parameters and the correct joint offsets simultaneously.

Table 1: Methods where the correct parameters were identified in simulations.

Method	Offsets	Kin. par	Kin. par + offsets
One unknown fixed point	no	-	-
One known fixed point	yes	-	-
Several fixed-point	no	-	-
Dual arm (one arm calibrated)	yes	yes	no[1]
Dual arm (no arm calibrated)	yes[2]	no	no

References

[1] Gustaf Bergström and Björn Green. "Evaluation of Calibration Method for Redundant Dual Arm Industrial Robots Using Internal Sensors". MSc Thesis No. 6012. Department of Automatic Control, Lund University, Sweden, May 2017. URL: `http://control.lth.se/publications/`.

[2] *YuMi*. 2016. URL: `http://new.abb.com/products/robotics/yumi` (visited on 06/01/2016).

[1]The end-effector's pose error did, however, converge to zero in this implementation, but the parameters did not converge to the correct values.

[2]In combination with the several fixed-point implementation.