Camera to Robot Arm End-Effector Calibration

T. van Wershoven (s2464721)

September 1, 2024

1 Abstract

Robots are fulfilling more and more jobs in current society, and for a robot arm to be able to know what it is going to do it needs to know how the end-effector needs to move in relation to what it sees on the cameras.
For a robot to know where their end-effector needs to move it needs to be calibrated to a camera feed. Therefore this study calibrated a Franka Research 3 robot arm to a Stereolabs ZED 2 stereo camera.
This calibration had errors of up to three meters when the camera was attached in the world frame, while attaching the camera to the end-effector reduced this error to a maximum of half a meter. A likely cause for this is that the used program for calibration does not account for the possibility of the camera being attached in the world frame. This is supported by that in the documentation only the camera being attached to the end-effector is shown.[1]
This error is further magnified by the addition of noise to the rotational vector of the camera pose transformation added after the measurement, or by removing the end of datasets.

Contents

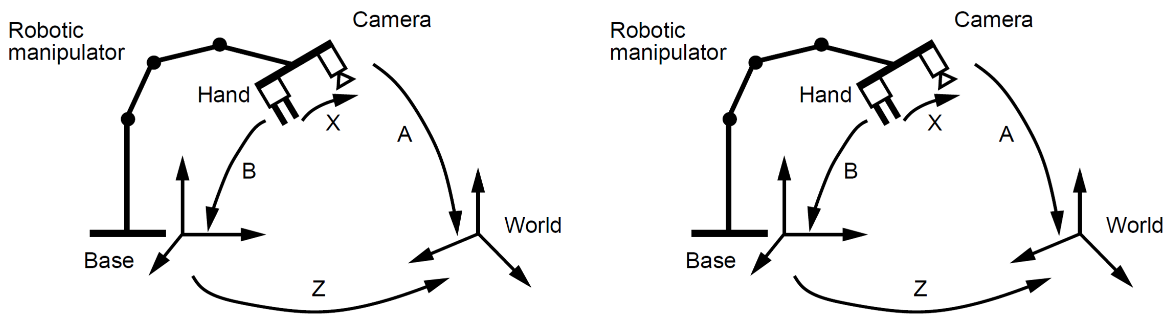
2 Introduction

Robotics are taking in a ever more present role in our society, being used more and more to phase out menial jobs.[2] For a robot to be able to do menial jobs, like stocking shelves or getting the right tools from a cabinet it must be able to know where the objects for its task are located, and how to find them in a potentially foreign environment.
A common method for teaching robots how to complete tasks that can't easily be pre-scripted, or defined as an optimisation problem is by having them learn from demonstration.[3] This study does not aim to replace any of these methods, instead it aims to enhance them by firmly coupling the input of a camera to the subsequent optimal movement of the robot arm. If the robot arm end-effector is calibrated to the feed of the attached camera it can properly do as instructed without readjustments during the task.

2.1 Pose transitions

To quantify the change in position and rotation of one object to another this paper uses pose transition, consisting of rotational and translational vectors. These vectors symbolize the change a vector must undertake to match with the new object, in this case what rotations and translations it must undertake.

2.2 Hand-Eye coordination



((a)) Robot-World Hand-Eye.[4]

((b)) Robot-Eye Hand-World.[4]

Figure 1: Two calibration methods as described by Dornaika, F and Horaud, R.[4]

Two scenario's are common for hand-eye coordination[4]: Robot-World Hand-Eye and Robot-Eye Hand-World, both of which can be seen in Figure 1. Both these scenarios can be described with the system of matrix equations AX = ZB. With B and A known in multiple measurements the values for X and Z can be found.
The goal of the research is to test the accuracy of these calibrations, calculating potential errors and finding their cause.

3 Methods

3.1 Setup

In this research there was chosen to use the following camera and robot arm:
A Stereolabs ZED 2 stereo camera, from which both lenses were used (as seen in Figure 3)
And a 7 degrees freedom of movement Franka Research 3 robot arm from Frank Emika, which was put in free move Guiding Mode for the duration of all measurements.

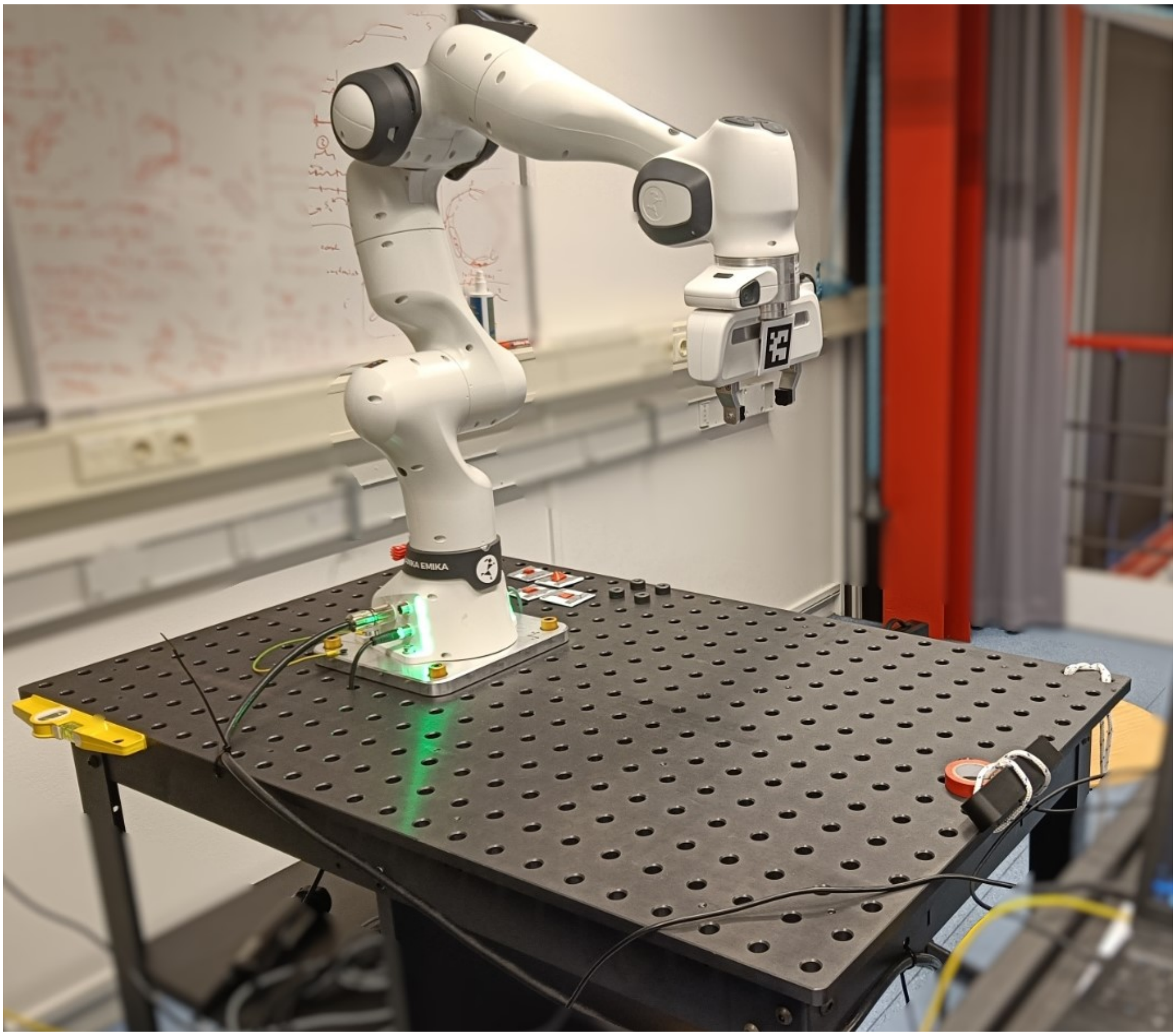


Figure 2: Here the overall setup can be seen, with the Franka Research 3 robot arm bolted to the table and the Stereolabs ZED 2 stereo camera tied to the edge at an angle.

As seen in Figure 2 the camera is not attached to the end-effector, instead it is tied to the table. This was done as it was easier in this and other setups to have the camera operate from a separate location from the end-effector.
An ArUco marker was attached to the end-effector of the Franka Emika robot arm, to allow for positional post processing. The ZED camera was positioned in such a way to ensure an angle was reached where the ArUco marker is within view of the camera in most end-effector positions of the robot arm.

3.2 ArUco marker

ArUco markers are patterns of blocks which can be used by computer programs to determine the rotational and translational vectors of the camera lens to whatever the ArUco marker is attached to. The program calculated the A pose transformation, which consists of the rotational and translational vectors of the camera to the marker or vice-versa, by detecting the discrepancies in the expected form of the ArUco marker to that perceived. .[5]

Figure 3: Here an ArUco marker attached to the end-effector is shown that has been processed, the computer recognized the 3 Cartesian's directions stemming from the marker.

3.3 Data collection

During each measurement the Franka Emika robot arm was put in guiding mode and manually moved in patterns decided on the spot by the operator that maximized the different angles and distances between the Stereolabs ZED 2 stereo camera and the ArUco marker.
These measurements lasted around the ten seconds to allow ample time to take on different poses. The data from these experiments was then collected in binary and sent to a computer system for processing.
To collect the robot state data from the Franka Emika robot arm their inbuilt robot state export was used. For the camera sent the data to the attached computer system in binary arrays, which were then sent to the central computer over the local network.
As the frequency of the robot data and camera data wasn't synchronized a method was used that saved the most recent output until both systems had sent data, saving them both. New data overwrites old data to ensure data points are as close to one another as possible.

3.4 Data processing

From the camera data the pictures were extracted and using ArUco marker detection the rotational and translational vectors of the camera pose transformation were determined per frame. Both the image extraction and the ArUco marker detection were done using OpenCV version 4.8.0.76, with their ArUco module.
The Franka Emika robot arm rotational and translational vectors were extracted from the robot state data to determine the robot pose transition.[6]
Having gotten both pose transitions together the calibration can be done. This calibration gave the estimate of the pose transformation A and B.[1]

3.4.1 Noise study

To determine the impact each rotational and translational vector has on the calibration randomized noise proportional to the inputted data was added before being analyzed.

3.4.2 Excision of the data

The measurements were taken with care to only include smooth uninterrupted movements of the end-effector. This was done as the expectation was that the method with which the calibration operates needed connected movements to fulfill its task. By removing parts of the dataset and comparing it to the averages where nothing was removed the research tried to see if smooth movement is needed to take proper measurements

3.4.3 Flipped perspective

To determine if attaching the camera to the end-effector induces a significant difference in comparison to attaching the camera to the world frame extra measurements were done in the same manner with the camera attached to the end-effector.

4 Results

4.1 Averages

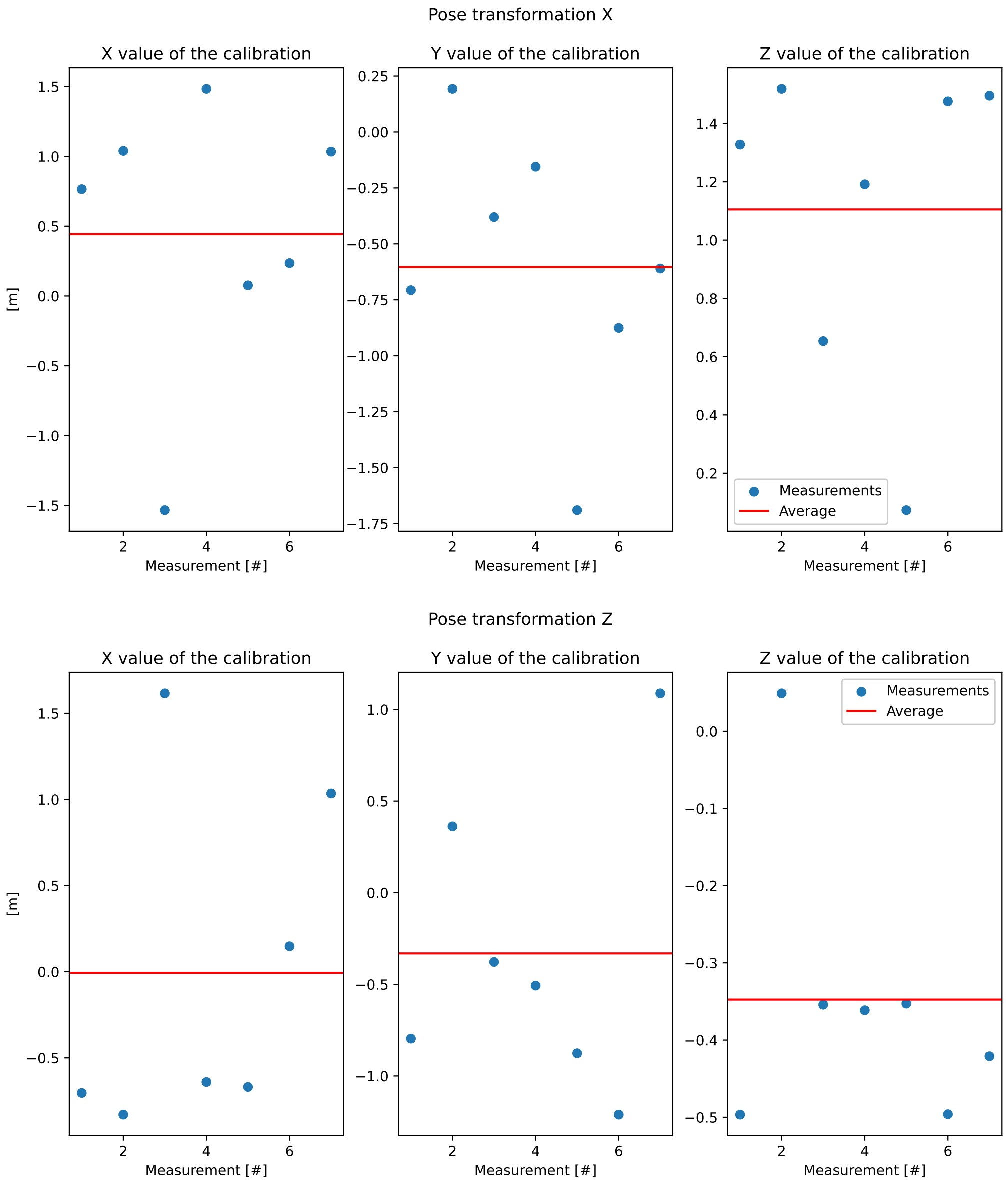


Figure 4: Calibration results of the measurements with the average of that coordinate included. The top 3 graphs are the x-y-z coordinates of pose transformation X, the bottom three graphs are the same coordinates for pose transformation Z. The blue dots are the values of separate measurement, with the red line representing their average.

The x-y-z values represent where the pose transformation is in relation to the base. What base and end point are used for pose transformation X and Y is further explained in Chapter 2.2.
The deviations as seen in Figure 4 result in some measurements having a difference of up to three meters in the same coordinate. In comparison to the error of three meters, the robot arm has a reach of 0.855 meter.[7]
Although some coherence can be seen in the values, the outliers of these measurements are not

the same each time. Due to this the outlying of data cannot be blamed on a faulty measurement, instead it has to be caused by some underlying issue.

4.2 Noise study

As described in Chapter 3.4.1, to discern what parts of the inputted data have the most influence proportional randomized noise was added to different parts: The rotational and translational vectors of both pose transformation A and B.

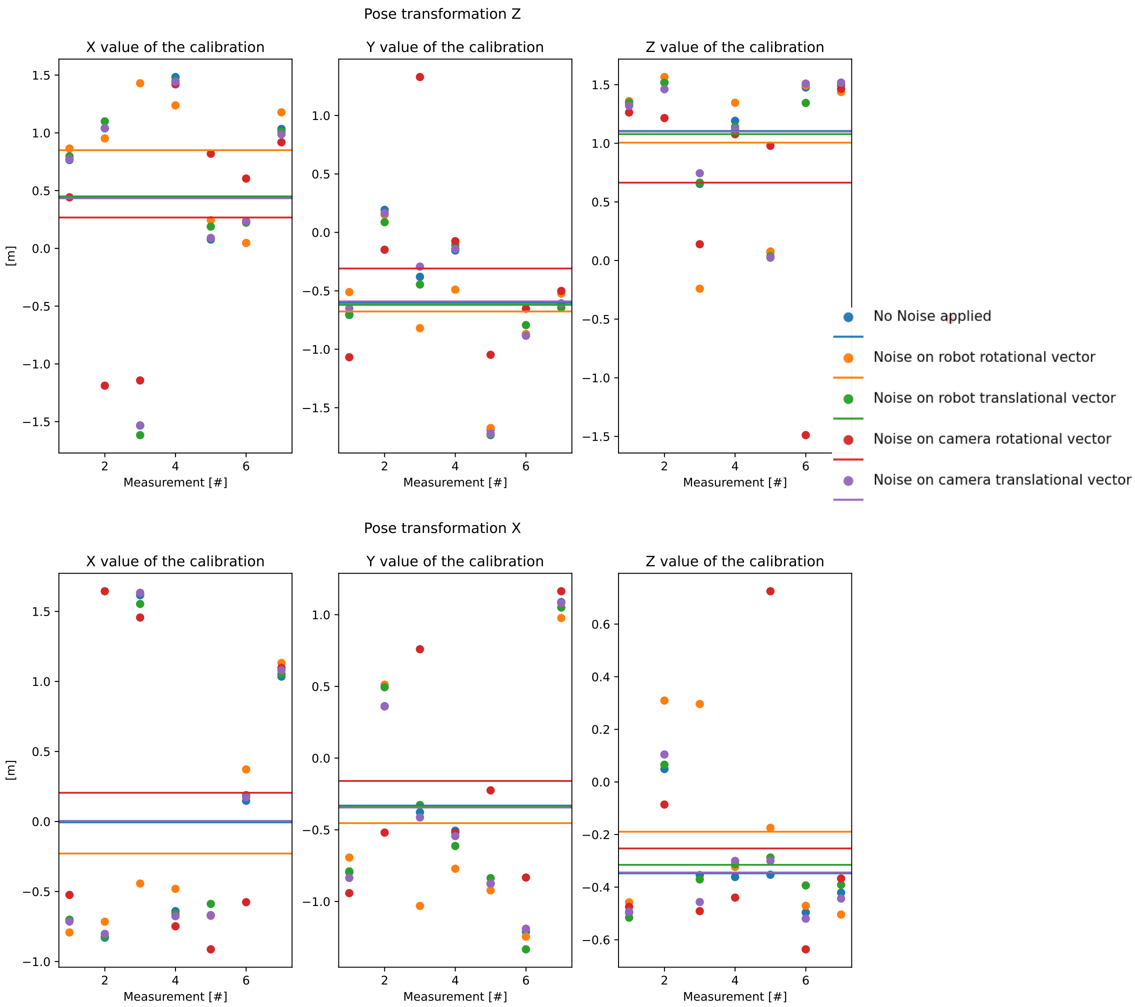


Figure 5: Calibration results of 7 measurements with noise applied on different factors. The lines represent the averages over all measurements for that noise application.
The different colours indicate when noise is applied on different parts of the input vectors. Blue no noise is applied; yellow and green noise is added to the rotational and translational vector of pose transformation B; and red and purple noise is applied to the rotational and translational vectors of pose transformation A

As can be seen in Figure 5 when noise is applied on the rotational vectors (yellow and red) there is a major shift in the average calibration results, ranging up to half a meter of error.
When noise is applied on the translational vectors (green and purple) however there is often only small shifts in the average calibration results, staying below the ten centimeters in the

most extreme case.
The application of noise to the translational vectors has no impact on the rotational vectors of the calibration results.

4.3 Excision of data

The importance of the length of the measurement and consistent movement during the measurement are analysed by excising randomized lengths of data-points from the datasets.

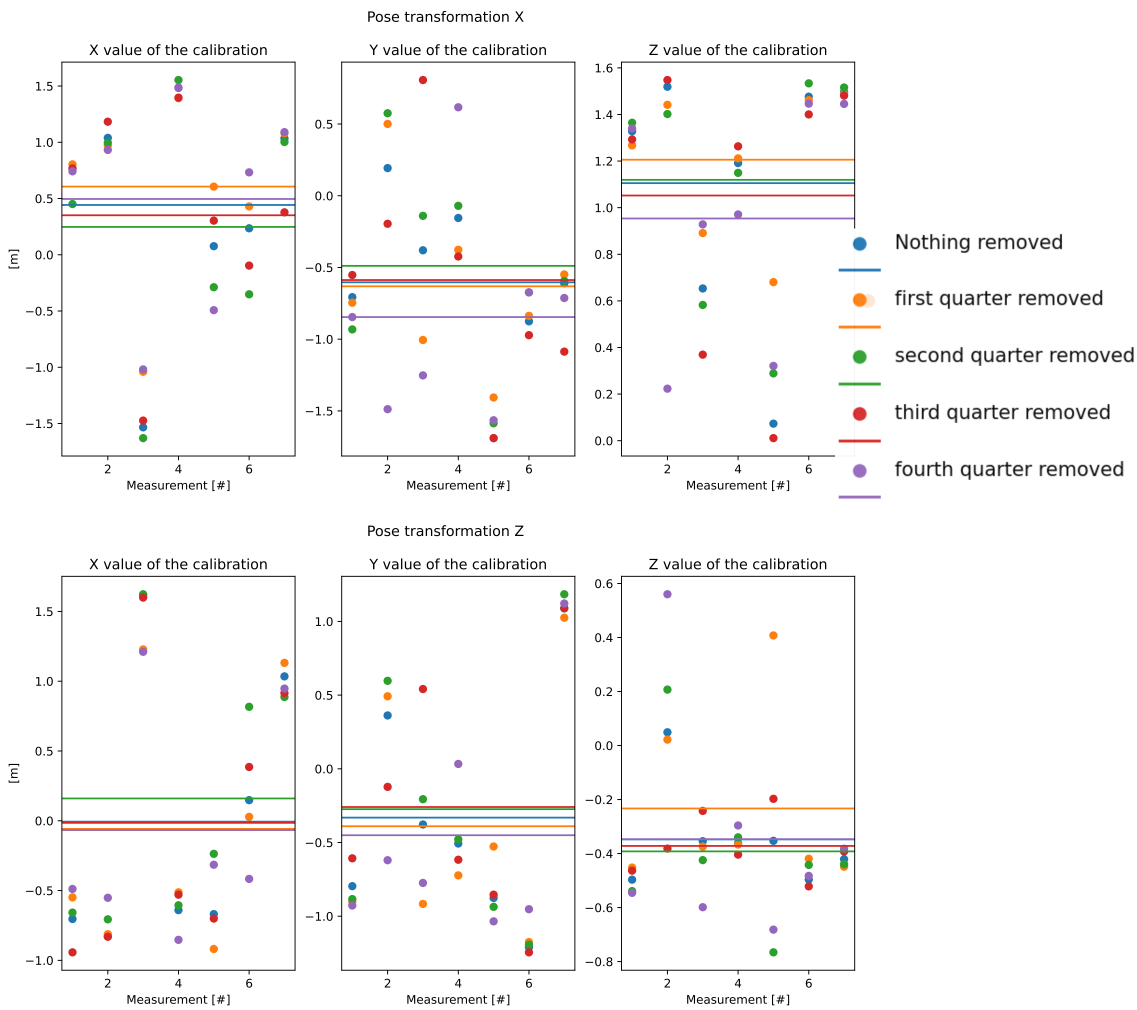


Figure 6: Calibration results of 7 measurements with parts of the data removed. The lines represent the averages over all measurements for the calibration with a certain part removed. The different colours indicate what part is removed from the dataset. With blue nothing is removed; yellow, green, red, and purple indicate the first, second, third, and fourth quarter being removed, in that order.

When data is removed the averages of the calibration results shift. The largest shifts in calibration averages happens when the data at the end of the dataset is excised, while removing data in the middle has less of an effect.
In contrast to adding noise, removing data always shifts the averages.

4.4 Flipped perspectives

To ensure that the switching of the camera doesn't have an influence on the working of the calibration, the measurements were done in the same style as noted in the Methods Section, with as only change that the Stereolabs ZED 2 stereo camera and the ArUco marker switched positions.

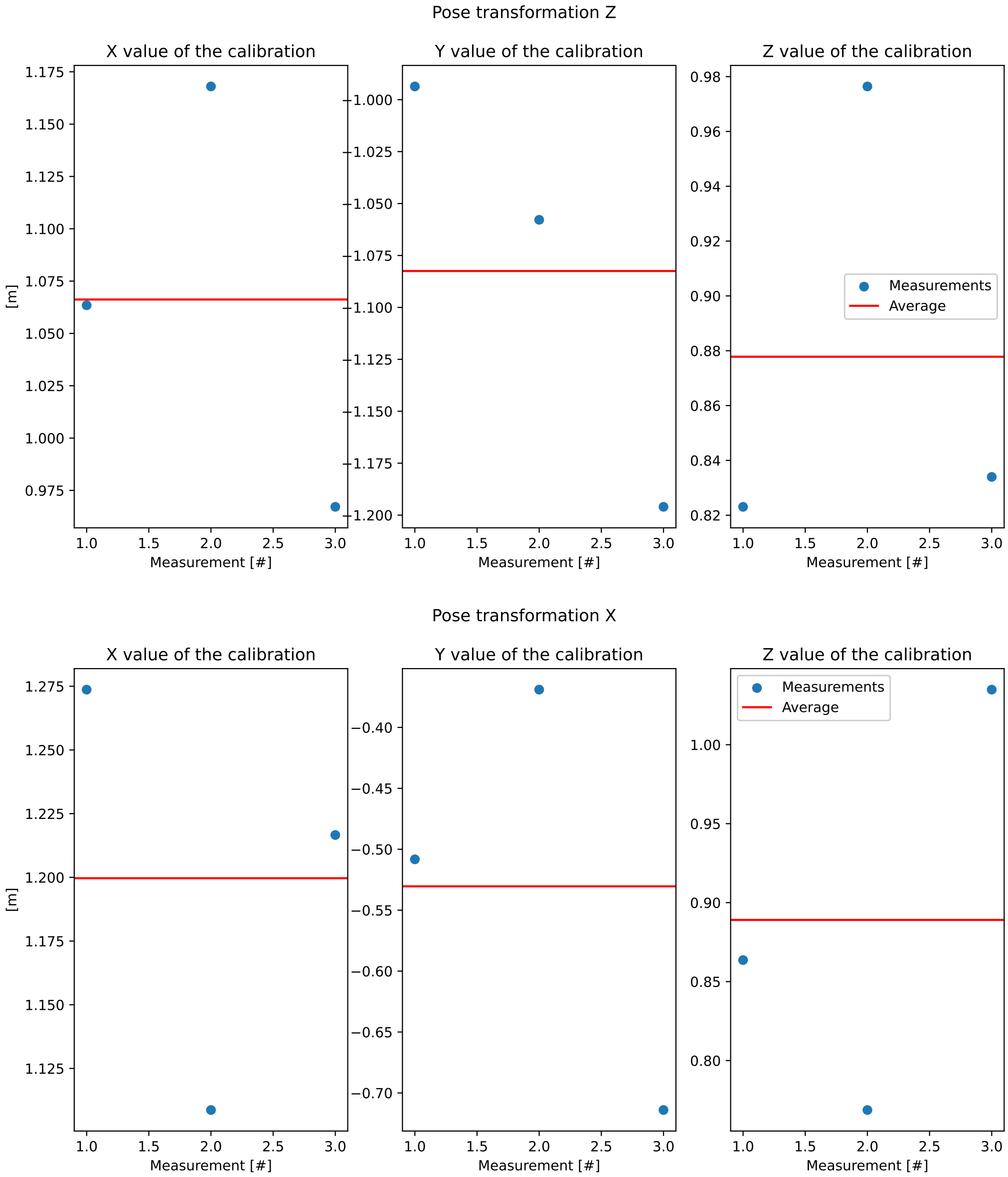


Figure 7: Calibration results of the measurements with the average of that coordinate included. The top 3 graphs are the x-y-z coordinates of pose transformation X, the bottom three graphs are the same coordinates for pose transformation Z. In these three measurements the camera was attached to the end-effector instead of to the table.
The blue dots are the values of separate measurement, with the red line representing their average.

Figure 7 paints a very different picture then Figure 4. Where in Figure 4 the measurements differ in some places up to three meters in Figure 7 the differences are at max half a meter, often staying within a decimeter of one another. The method data samples are taken with does not differ in the flipped perspective except for the switching of the camera to the end-effector. This indicates that the only influence on this difference is the switching of the camera to the end-effector.

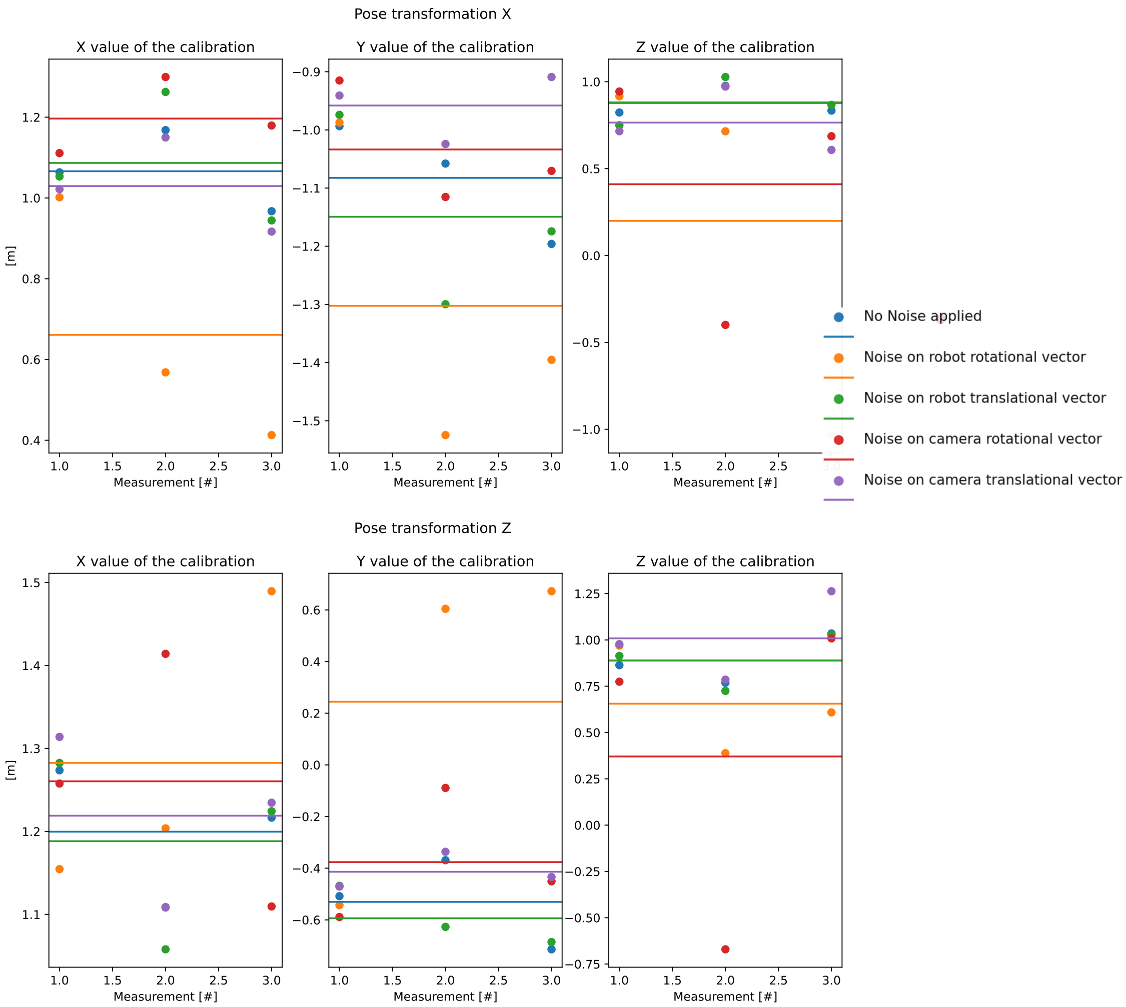


Figure 8: Calibration results of 3 measurements with noise applied on different factors. The lines represent the averages over all measurements for that noise application. In these three measurements the camera was attached to the end-effector instead of to the table
The different colours indicate when noise is applied on different parts of the input vectors. Blue no noise is applied; yellow and green noise is added to the rotational and translational vector of pose transformation B; and red and purple noise is applied to the rotational and translational vectors of pose transformation A

In contrast to Figure 5 in Figure 8 it can be seen that adding noise to the rotational vector of the robot has majorly decreased impact, while the impact of adding noise to the translational vector of the robot has increased.
The impact of adding noise to the rotational vector of the camera is both in Figure 5 and in Figure 8 a cause for major shifts in the calibration values.

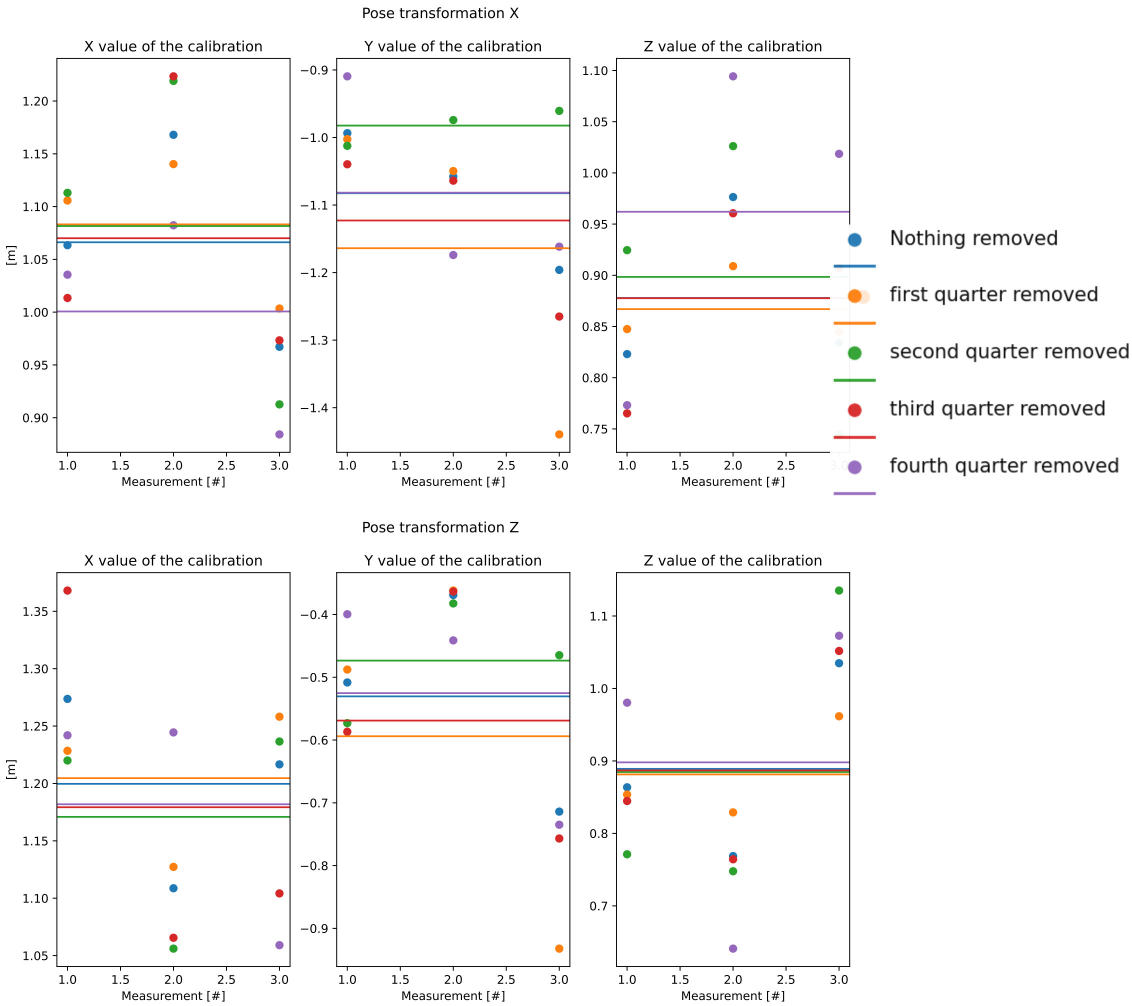


Figure 9: Calibration results of 3 measurements with parts of the data removed. The lines represent the averages over all measurements for the calibration with a certain part removed. In these three measurements the camera was attached to the end-effector instead of to the table.
The different colours indicate what part is removed from the dataset. With blue nothing is removed; yellow, green, red, and purple indicate the first, second, third, and fourth quarter being removed, in that order.

While removing the fourth quarter has a sizable impact in both Figure 6 and Figure 9, removing the first quarter has less of an impact in Figure 9 then it had in Figure 6. Instead the impact of removing the second quarter seems to have increased.

5 Conclusion

The data when measured with a camera on the table instead of on a robot arm gives inconsistent results compared to attaching a camera to the end-effector. An error of three meters between measurements is too large for a robot to be able to stock shelves.
Regardless of the position of a camera, on the table or on the end-effector, adding noise to the rotational vector of the pose transformation A seems to induce major errors, while doing the same to the translational vector of the same pose transformation seems to barely induce errors. This implies that while the translational vector of the pose transformation A does not hold much influence on the calibration, the rotational vector does.
The application of noise to the translational vectors has no impact on the rotational vectors of the calibration results.
In the same vein excising the last quarter of the dataset induces major errors in both positions of a camera. While the expectation was (as stated in Chapter 3.4.2) be that removing central parts of the dataset would cause the largest errors due to positions not making full smooth transitions this only sparingly seems to be the case when the used camera is attached to the end-effector, and even then it makes less of an impact then removing the last data.
While the error in all cases decreases when the Stereolabs ZED 2 stereo camera is attached to the end-effector of the Franka Research 3 robot arm, the error in the calibration error generally still stays within the ten centimeters. Without additional algorithms to aid the use of the robot arm in manual jobs this is not sufficient.

6 Discussion

The research is limited in the sample sizes, the data varies greatly between measurement sets. Three measurements in one position means potential outliers are difficult to filter out. The values of the calibration results often don't show clear correlation, best shown in Figure 6. Further studies should take in account that the same amount of data is taken when comparing camera positions.
Furthermore based on that distorting smooth movement does not cause major errors, as seen in Figure 6 and Figure 9, further research does not need to take such caution when devising movement pattern for data measurement.

References

[1] OpenCV. Camera Calibration and 3D Reconstruction;. Available from: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html.

[2] Montobbio F, Staccioli J, Virgillito ME, Vivarelli M. Labour-saving automation: A direct measure of occupational exposure. The World Economy. 2024 1;47(1):332-61.

[3] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, Aude Billard. Recent Advances in Robot Learning from Demonstration. In: Annual Review of Control, Robotics, and Autonomous Systems 2020.. vol. Volume 3. Annual Reviews; 2019. p. 1-33.

[4] Dornaika F, Horaud R. Simultaneous robot-world and hand-eye calibration. IEEE Transactions on Robotics and Automation. 1998;14(4):617-22.

[5] OpenCV. Detection of ArUco Markers;. Available from: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.

[6] Franka Robotics. franka::RobotState Struct Reference;. Available from: https://frankaemika.github.io/libfranka/structfranka_1_1RobotState.html.

[7] Franka Emika. Franka Research 3;. Available from: https://franka.de/research.