ME 4773/5493 Fundamental of Robotics
Fall 2016
San Antonio, TX, USA

DETERMINING ACCURACY OF AN ABB IRB1600 MANIPULATOR AND FORMING COMMON REFERENCE FRAME WITH A FARO ARM

Geoffrey Chiou
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
geoffrey.chiou@utsa.edu

ABSTRACT

The accuracy of an ABB IRB 1600 serial manipulator was measured using a FaroArm coordinate measuring system and a method was developed to create a common frame of reference between the end effector of the ABB robot and the base of the Faro. This method involved calculating the homogeneous transform matrix between a cloud of points measured at the end effector of the robot using the Faro and the position the robot thought it was moving to. On average, the analysis showed that the robot was accurate to about 2.5 mm with a maximum error of around 5.0 mm. This shows that an alternative method of calibration will need to be developed for applications requiring accuracy greater than 2.5 mm.

1. INTRODUCTION

An unnamed aviation firm manufactures parts for their planes by clamping sheet metal between two fixtures called Hand Routing Fixtures or HRFs. These fixtures are used as a guide for a technician to remove excess material from the part. This process has been automated using a robot with a routing tool as the end effector.

Since there are many different HRFs and CAD data is not available, a technician has to manually teach the routing trajectories for each HRF to the robot using a teach pendant. This process can take many hours since the geometry of each HRF can vary and there are many complex shapes.

A new method to acquire the routing trajectories was devised. This new method used a FaroArm probe to trace the parts and acquire the trajectory. A FaroArm is a portable coordinate measuring machine [1]. The figure below shows a part between two HRFs as it is being measured by a Faro. To accomplish this task, a common frame of reference was developed to communicate the information between the FaroArm and the robot. The FaroArm would then be used to acquire points by tracing the outline of the part. These points were exported in the frame of the robot and fed into software that automatically generated a trajectory.

Figure 1. Sheet Metal Part Traced by Faro

The method used to develop a common frame of reference involved touching off points on a small machined cube with the FaroArm and the tip of the robot and forming those points into a common coordinate system. Since the cube was small, any errors in the geometry of the cube were amplified at greater distances. This led to inaccurate trajectories being generated where the center point of the actual tool was several centimeters away from the predicted location. Unfortunately, the project had expended all funding and further development was discontinued.

My task for this project involved updating the software to run on the latest version of ROS, making the program work on an ABB IRB1600 robot, determining the accuracy of the robot, and developing a different method to determine a common frame of reference between the FaroArm and the robot. This report will cover the method used to determine the accuracy of the robot and forming a common frame of reference.

2. NOMENCLATURE

Acronyms

ABB	Swedish-Swiss multinational corporation operating mainly in robotics, power, and automation.
FlexPendant	Also known as a Teach Pendant, it is a hand held operator unit used to perform tasks involved with operating a robot. Connected directly to the controller.
HRF	Hand Routing Fixtures
RAPID	Programming language used to control ABB industrial robots.
ROS	Robot Operating System
URDF	Unified Robot Description Format

Variables

R	$3x3\ Rotation\ Matrix$
$RMSE_x$	Root Mean Squared Error in x.
$RMSE_y$	Root Mean Squared Error in y.
$RMSE_z$	Root Mean Squared Error in z.
t	$3x1\ Translation\ Matrix$
T	$4x4\ Homogeneous\ Transform\ Matrix$
x_{std}	Standard deviation in the x direction.
y_{std}	Standard deviation in the y direction.
z_{std}	Standard deviation in the z direction.

3. METHODS

The FaroArm was used to determine the accuracy of the ABB IRB1600 manipulator. The general idea in determining the accuracy of the robot was to jog the end effector of the ABB to essentially draw shapes in the air on three different planes and then measuring the position of the end effector at several points along each shape with the Faro. The data from the FaroArm would be exported and fitted to the data used to generate the shapes. This would result in a homogeneous transformation matrix which was applied back to the data from the FaroArm and analyzed for errors.

Drawing the shapes on all three planes with the robot involved writing a RAPID program to jog the robot for the ABB controller. In this program, a work object coordinate system was defined by setting all joints to zero and offsetting the end effector in the negative z direction by a couple hundred millimeters until the robot end effector was in its supposed area of operation. The accuracy of the robot was not measured for all possible areas of operation since the FaroArm was fixed to a table.

The RAPID program consisted of predefined points in reference to the work object coordinate system and used simple move commands to jog the robot to each of those points. For simplicity, the robot was set to stop and give the operator time to measure along each point of the unit circle. The points were generated using equations for a circle with a radius of 200 millimeters. A circle was drawn in the air for the YZ, XZ, and XY planes. These circles are shown below in Figure 1.

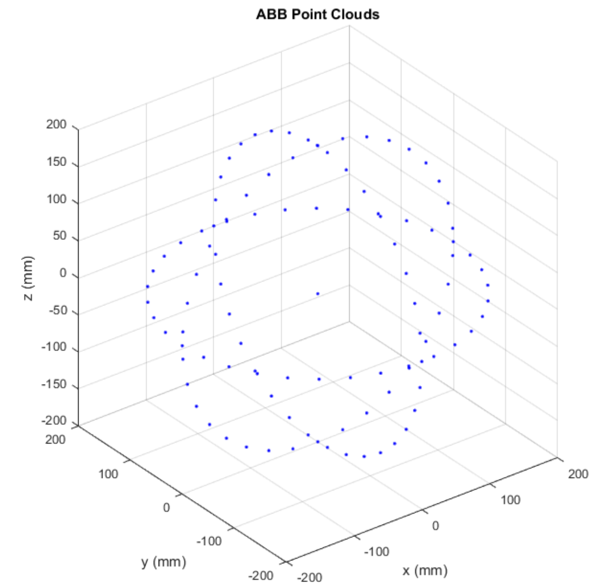

Figure 2. Points where ABB Stops

The FaroArm probe was stuck in the center of the end effector where there was a small groove and captured a single point every time the robot stopped. This resulted in two separate point clouds, one in the coordinate system of the ABB work object and one in the coordinate system of the FaroArm base. The figure below shows a plot of the comparison of the two point clouds in their respective coordinate systems.

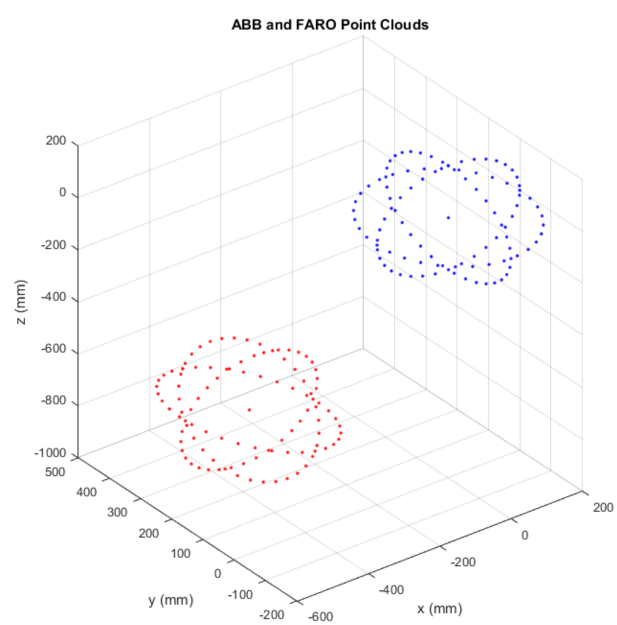

Figure 3. Comparison of Points Measured by Faro vs. Set Points of the Robot

Calculating the transform between the two point clouds was done with a Python library written by Clay Flannigan [2]. The specific function used from the library calculated the least squares best fit transform between two 3D point clouds and returned the homogeneous transform matrix. The function from this library has since been ported to C++ to better integrate with the ROS ecosystem.

Determining the accuracy of the robot also solved the problem of forming a common frame of reference between the FaroArm and the ABB since the best fit transform will transform points in the Faro coordinate system into the ABB coordinate system.

To streamline the process of transforming points exported from the Faro software to the coordinate system of the ABB, a ROS node was written which accepts 4 arguments. The first two arguments contain the circles generated by the ABB and the respective measurements of the circles measured by the Faro. These two files would serve as the calibration files. The next argument is the file consisting of points exported by the Faro software that need to be transformed into the ABB coordinate system. The final argument was the name of file to save the output of the transform.

4. RESULTS

Running the point clouds through the library results in the following rotation, translation, and homogeneous transform matrices.

$$R = \begin{bmatrix} 0.945777096 & 0.0424008 & -0.32205506 \\ -0.32247296 & 0.00322814 & -0.94657317 \\ -0.03909582 & 0.99909547 & 0.01672619 \end{bmatrix}$$

$$t = \begin{bmatrix} 41.49561374 \\ -844.48890326 \\ -269.67414628 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.945777 & 0.042400 & -0.322055 & 41.495613 \\ -0.32247 & 0.003228 & -0.946573 & -844.488903 \\ -0.03909 & 0.999095 & 0.016726 & -269.674146 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The resulting homogeneous transform was applied to the points measured by the Faro using the equation: $(Transformed = R * Original + t)$ with the results of the transform shown in the figure below.

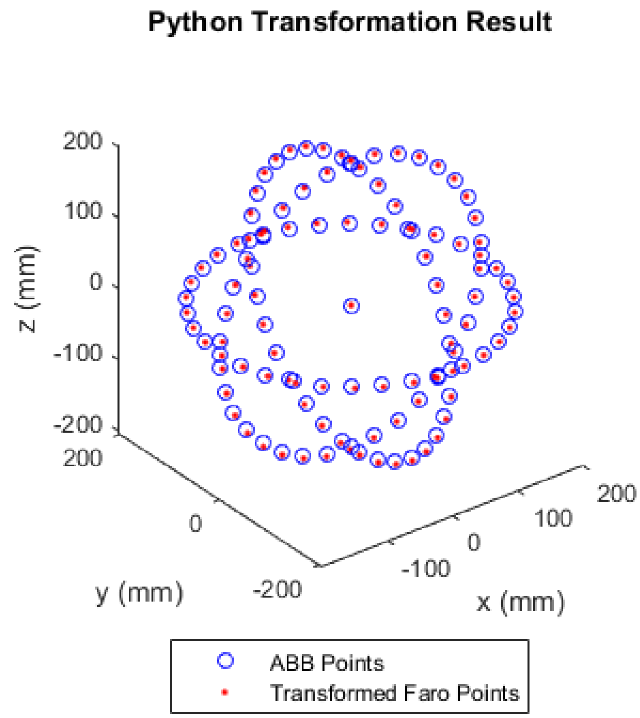

Figure 4. Comparison of Transformed Faro Points with ABB Points

From the plot it is clear that the transform was not perfect, and there are still errors that remain as a result of the transform.

5. DISCUSSION

The residual error for each axis was calculated by subtracting the transformed Faro point from the set ABB point. The standard deviation of the residual error for each axis were calculated in MATLAB and are as follows:

$$x_{std} = 1.3803\ mm$$
$$y_{std} = 2.4609\ mm$$
$$z_{std} = 2.0477\ mm$$

Using the residual error values, a residual plot was generated for every axis as shown in the figure below. From the plot, it is obvious when each plane changed based off of the shapes of the curve, especially in the z-axis. Note, the one outlier on the x-axis may have been due to a bad measurement caused by operator error.

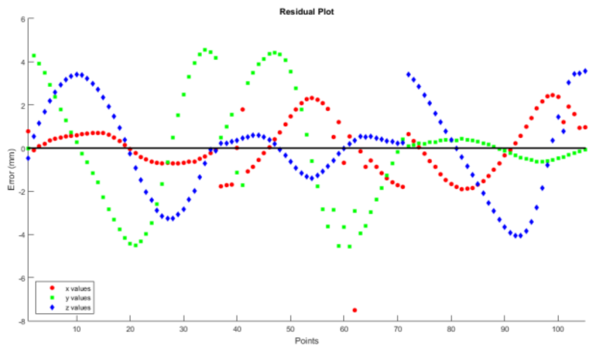

Figure 5. Residual Plot for Each Axis

The Root Mean Square Error (RMSE) was calculated for each axis by taking the square root of the sum of squares error divided by the total number of points. The RMSE would give a gauge of the average accuracy of the ABB manipulator assuming the assumptions made above were true. The values for the RMSE are shown below. The maximum possible errors are shown in the residual plot in Figure 4.

$$RMSE_x = 1.3738\ mm$$
$$RMSE_y = 2.4494\ mm$$
$$RMSE_z = 2.0381\ mm$$

The error between the norm of each point of the ABB and transformed Faro points was also calculated and plotted in the figure below. This gives a representation of the total residual error at each point.

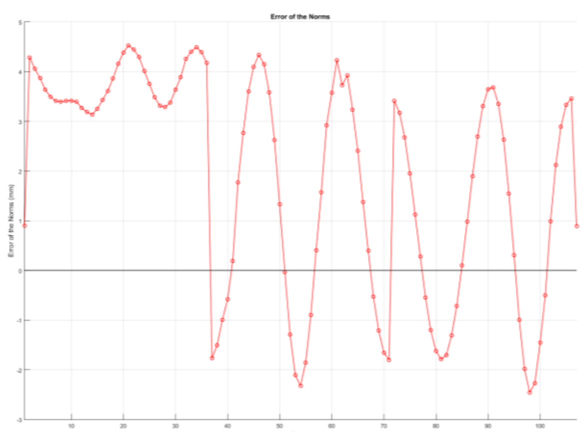

Figure 6. Error of the Norms between ABB and Transformed Faro Points

The standard deviation of the error of the norms is shown below. This shows that at each point the robot could have a possible error of 2.2 mm in all directions.

$$ErrorNorms_{std} = 2.2039\ mm$$

Since the points were collected simultaneously as the robot moved in circles across each plane, dividing the plot into three sections can give an idea of the accuracy of each plane. From Figure 5. It is clear that the first plane has high errors while the second and third plane and cyclic errors.

6. CONCLUSION AND FUTURE WORK

As shown in this report, the accuracy of the ABB IRB1600 is not perfect even after applying the calibration method. The overall calibration process could use refinements and multiple calibrations should have been performed to remove any of the outliers from the data. This calibration resulted in a root mean squared error of 1.37 mm in the x, 2.45 mm in the y, and 2.04 mm in the z, which gives a gauge of the average accuracy of the manipulator.

Future work will include refining the data collection process to ensure that the data does not have outliers. For actual implementation, the calibration process will involve measuring the tip of the end effector tool instead of the tip of the ABB default end effector to avoid tool changes.

7. OTHER WORK PERFORMED DURING SEMSETER

The accuracy experiment and development of a common frame of reference between the Faro and the ABB were only a small part of the project I worked on this semester. The main goal for the semester, as explained in the introduction, was to set up a working demo of the routing project on an ABB IRB1600 robot at the labs at Southwest Research Institute. A working demo consists of tracing a part, simulating the routing in ROS, and then simulating the routing with the actual robot. Instead of actually cutting into the sheet metal, the robot would move the tool center point along the calculated trajectory which would essentially trace the outline of the part.

This project was broken down into six separate milestones. The first milestone consisted of calibrating the ABB and Faro coordinate systems and estimating the accuracy of the ABB. The result of that milestone was documented in the first section of this report. The second milestone was to upgrade the software package to run on the latest version of ROS, which is ROS Kinetic. The software was originally written to run on ROS Hydro which is three versions older than Kinetic. Keep in mind that most libraries used in this project have not yet been upgraded to Kinetic. A lot of the libraries and dependencies were either deprecated or totally unsupported. A tremendous amount of time was spent porting the software or fixing any bugs introduced during the port. The third milestone was to upgrade the package named Descartes to run with the latest improvements. Descartes is a path planner for under-defined Cartesian trajectories. The fourth milestone was to use the transformed points from the Faro software to generate tool paths. Second, third, and fourth milestone were all tied to each other since they all depended on the software working correctly.

Another aspect of the fourth milestone was to create a new MoveIt package for the new work cell for the demo. This consisted of creating a URDF for the robot and generating IKFast solutions for the manipulator. IKFast is a library that generates closed form kinematic solutions for a manipulator. A comparison of what the work cell looks like in real life and what it looks like when visualized in RViz is shown in the figure below.

Figure 7. Comparison of Simulated Work Cell vs. Real Life

The sixth milestone for this project was to output the joint trajectories to the ABB robot. This is accomplished using a custom driver written by Jonathan Meyer from SwRI [3]. This custom driver automatically generates RAPID code from given inputs and uploads the code via FTP to the robot controller.

The final milestone is to have a working demo set up to demonstrate simulated robotic routing with an ABB IRB1600 manipulator. This milestone has not been completed as of this time. There are still a few bugs in the software and the measuring process.

In conclusion, there is still some work to be completed in this project, but it will ultimately result in a working demo. This has been a great introduction to using ROS in a real world application. The knowledge I have gained from this project has been extremely beneficial in my development as a robotics engineer and software developer.

ACKNOWLEDGMENTS

- Dr. Pranav Bhounsule for providing advice on the calibration.
- Mr. Clay Flannigan for his ICP library.
- Mr. Paul Hvass for suggesting this project.
- Dr. Chris Lewis for taking an interest in what I was doing one day and suggesting this method of calibration.
- Mr. Jonathan Meyer for helping me with bugs and answering my questions.
- Mr. Jorge Nicho for helping me with bugs and answering my questions.
- Southwest Research Institute for providing the facilities and expertise to help with this project.

REFERENCES

[1] Faro, "FaroArm Measuring Arm Overview," Faro, [Online]. Available: http://www.faro.com/products/metrology/faroarm-measuring-arm/overview. [Accessed 6 December 2016].

[2] C. Flannigan, "Iterative Closest Point," [Online]. Available: https://github.com/ClayFlannigan/icp. [Accessed October 2016].

[3] J. Meyer, "abb_file_suite," [Online]. Available: https://github.com/Jmeyer1292/abb_file_suite/. [Accessed December 2016].