

16-681 - MRSD PROJECT 1 | ILR #4
Individual Lab Report #1 | November 11, 2016

SAMBUDDHA SARKAR

TEAM G
EXCALIBUR

Huan-Yang Chang
Man-ning Chen
Sambuddha Sarkar
Siddharth Raina
Yiqing Cai

1. INDIVIDUAL PROGRESS

1.1 Overview

In this ILR I would describe how I executed the motion of the robot in 3D space. We decided on going with a Helical path as it gives us a good test bed to perform camera calibration in a wide variety of positions. Also I have successfully been able to fire the PSO for all 3 axes with multiple window tracking, so that helps us trigger the camera at desired set points even when traversing a 3D path. Some work has been done on velocity profiling, to prevent jerky motions when the robot changes acceleration or velocity or both.

Topics covered have been listed below for a quick overview.

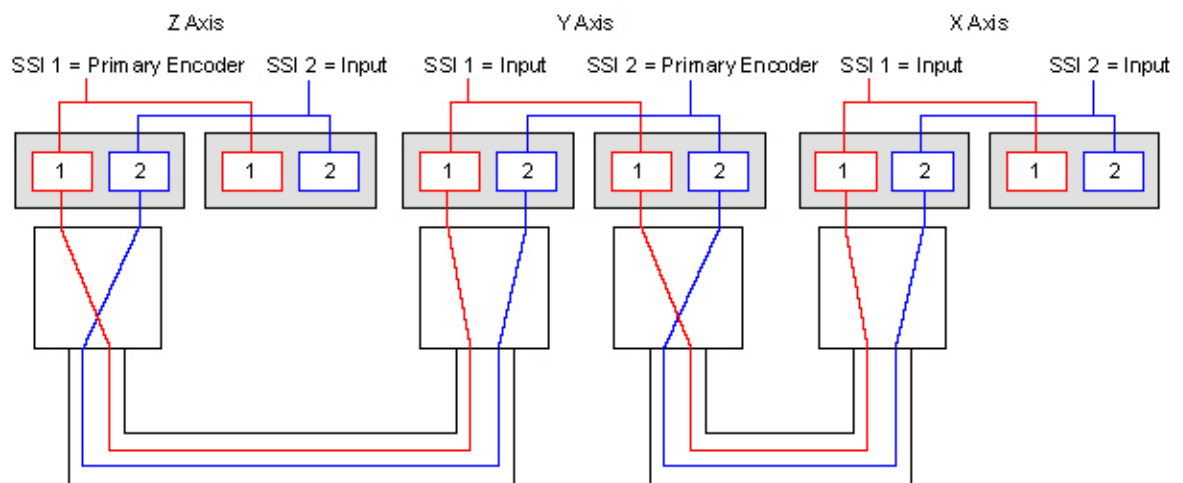
1.2 Multiple axis PSO triggering, PULSE mode: Final Completion

1.3 Trajectory generation: Helical Path

1.4 Velocity Profiling

1.2 Multiple axis PSO triggering, PULSE mode: Final Completion

A brief summary about the connection scheme is important, although it was discussed in the previous ILR, it is important to recapitulate. The connection scheme has been shown in Fig. 1.2



NOTES: Z axis is routing its primary encoder data to X axis (as well as Y axis) through SSI port 1.
Y axis is routing its primary encoder data to X axis (as well as Z axis) through SSI port 2.

Fig. 1.2, Connection Scheme of controllers for triple axis triggering.

This function is made available by the SSI Net ports that are present on the A3200 HPe & HLe, these help to perform three axis PSO tracking using multiple drives. At Oculus, the X axis has the -TRIPLEPSO option and the Y and Z axes represent the other two axes that will be tracked. The Y and Z axes output their encoder feedback through SSINet ports 1 and 2, respectively, to the X axis.

The algorithm for triple axis fire has been given below:

1. Reset the PSO.
2. Configure the PSO hardware to send the PSO signal out the auxiliary marker. If using the dedicated PSO output, this command should not be used.
3. Configure the pulse generator by setting the pulse on/off times.
4. Specify the output of the PSO to come from the pulse generator.
5. Enable the PSO and force it to fire continuously.
6. Allow the PSO output to pulse for 5 seconds by waiting.
7. Disable the PSO pulse generator.

It is important to note that triple axis PSO trigger works by tracking multiple axis windows and triggers the PSO when any one of the axis crosses the fire distance or set point. It can be imagined to be a small threshold 3D sphere, and whenever any of the axes crosses this 3D threshold sphere, the PSO is triggered to fire. The location at which the PSO is fired can be logged and then used as a ground truth when calibrating the camera.

1.3 Trajectory generation: Helical Path

The Helical trajectory is a compact yet diverse path for the calibration target to follow, as it not only gives using circular motion in one plane (in this case the X-Z plane), but also a depth component along which the Helix is pitched (in this case the Y axis). The Helical path's co-ordinates were generated in MATLAB and then ported to the Motion Composer Suite of AEROTECH.

The algorithm used for generating a helix has been given below

1. Get the number of steps or turns required.
2. Split the time into small time steps.
3. Plot a sine wave on the X-axis using the time steps above.
4. Plot a cosine wave on the Z-axis using the time steps above.
5. Use the time steps as the incremental location update for the Y-axis.
6. Scale the output of all three above axes to the physical dimensions of the Aerotech arm.
7. Store data points in a matrix.
8. Plot the result for clarity.

The diagrams shown below give the trajectory shape (Fig 1.3.1) and the trajectory shape in robot's workspace (Fig. 1.3.2). The number of turns here are 2.

Trajectory Shape

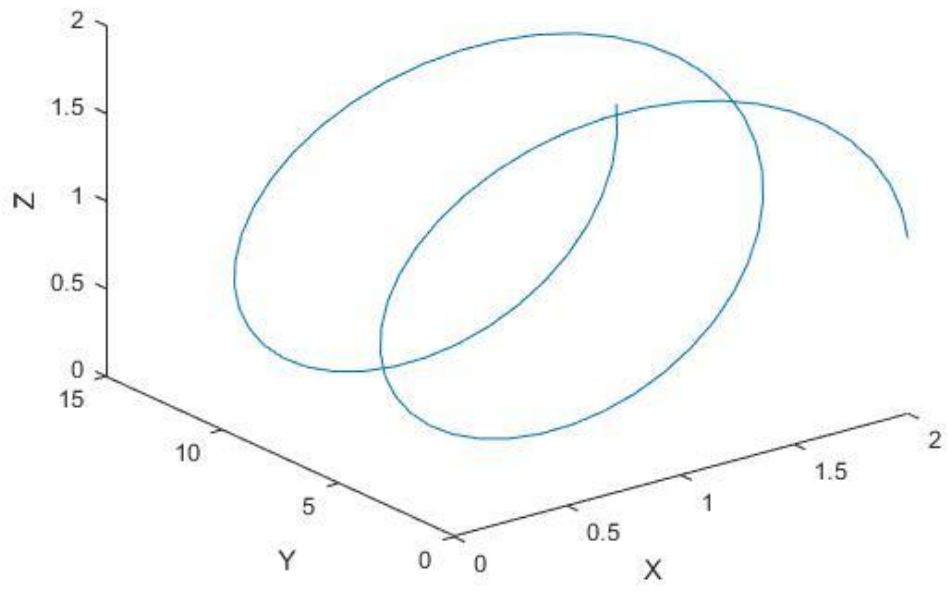


Fig. 1.3.1, Trajectory Shape.

Trajectory in AEROTECH Workspace

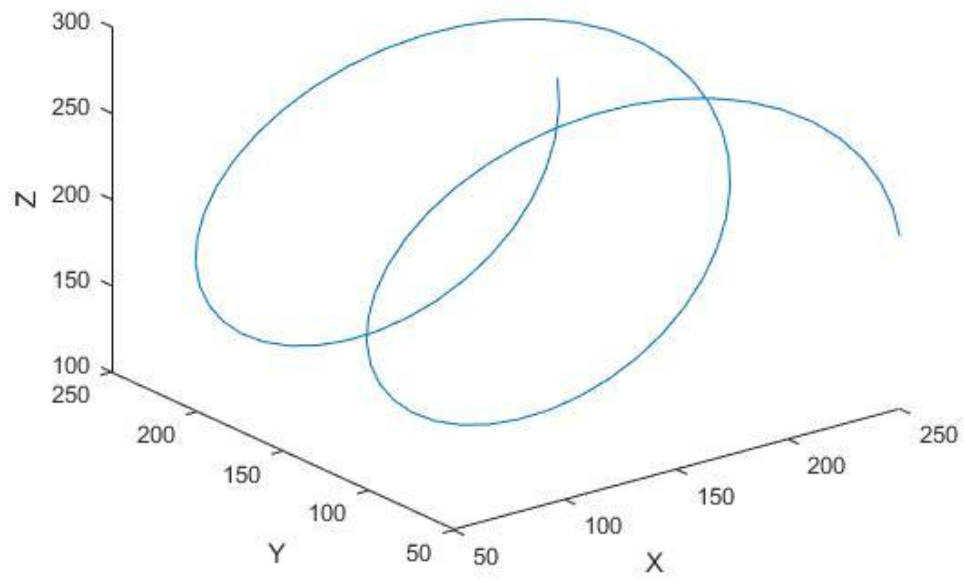


Fig. 1.3.2, Scaled trajectory in AEROTECH's workspace

1.4 Velocity Profiling

Velocity profiling is nothing but ensuring that the velocity transitions between successive positions of the linear actuator is smooth and without any jerks or sudden motions.

The algorithm for velocity profiling is pretty straight forward:

1. Set rate-based acceleration/deceleration.
2. Execute profiled velocity on X axis only, then on Y axis and then on Z axis.
3. Perform velocity blending based on current and future positions of the linear actuator along each axis individually.

The concept of velocity profiling has been shown in Fig. 1.4.1 and Fig. 1.4.2 which shows the difference between sudden velocity drops when there is a transition. This is overcome by velocity profiling.

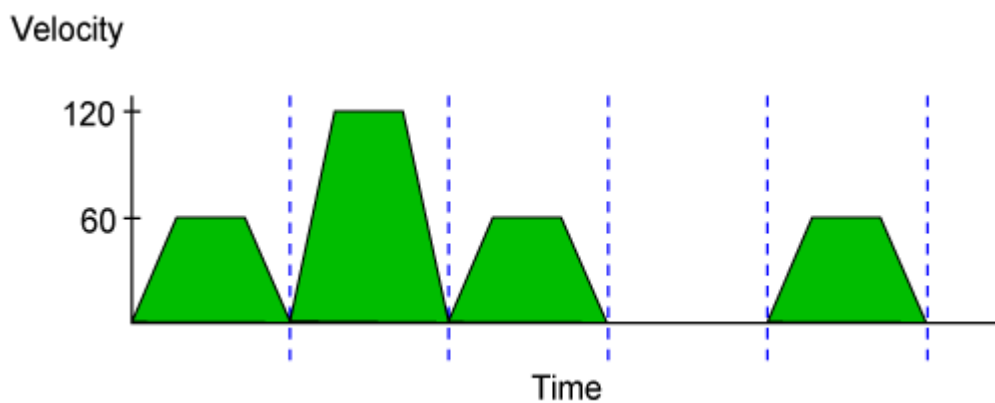


Fig. 1.4.1, Without velocity profiling.

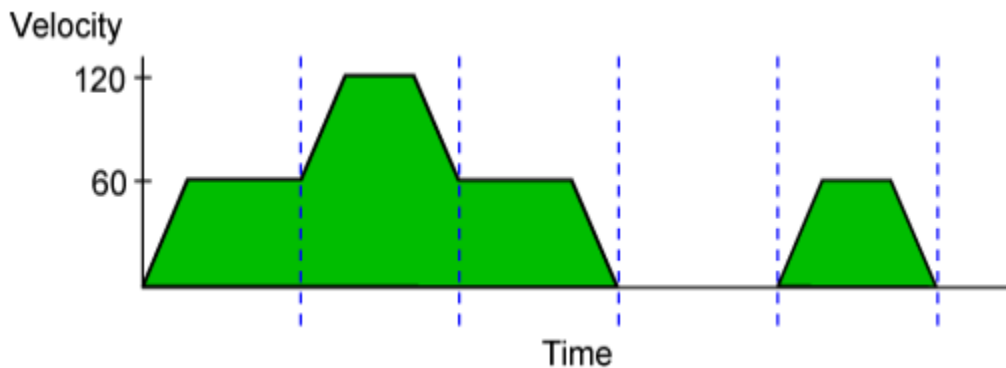


Fig. 1.4.2, With velocity profiling.

2. CHALLENGES

There were a few challenges which I encountered during my work, they are yet to be dealt with as we are waiting for the parts to arrive. The challenges have been highlighted in Fig. 2. These were not resolved when the PR#3 was presented, but have been resolved now.

AEROTECH ROBOTIC ARM

ISSUE	STATUS	PROCEDURES TRIED	SOLUTION
Cable Entanglement of Y-axis when moved along X-axis	resolved	<ol style="list-style-type: none"> 1. 380 degree cable carrier : Play of cable carrier isn't enough. 2. Look into hybrid cable carrier design: Not worth the effort 3. Place acrylic sheet below the cable carrier, allows for slipping and dragging without much friction from the carpet.: RESOLVED 	Use Acrylic sheet to reduce surface friction.
Y axis doesn't generate PSO trigger pulse.	resolved	<ol style="list-style-type: none"> 1. Search the sample code from the vendor 2. Figure out the function's meaning 3. Contact the vendor for help 4. Peter was firing the Z-axis instead of the Y, Don't let Peter near the Aerotech codes: RESOLVED 	Limit Peter's access and editing privileges to Aerotech Arm's core programs.
Triple Axes not firing in a single PSO Window.		<ol style="list-style-type: none"> 1. Check controller configuration 2. Check the wiring and daisy chained network. 3. Enable the auxiliary encoders after enabling the primary encoders for PSO, sequence matters.: RESOLVED 	Enable the auxiliary encoders after enabling the primary encoders for PSO, sequence matters.
License registration failure on iNtime	resolved	<ol style="list-style-type: none"> 1. Restart the Aerotech Motion Composer and Reset the controller: RESOLVED 	Reset the A3200 controller.

Fig. 2, Challenges faced and steps taken to resolve them.

3. TEAM WORK

The project work was divided among the team members and the task was assigned according to the strengths of the team members. The task division has been listed below in Table 3. The divided tasks can be completed in parallel; hence others can pitch in when some team members fall behind in their work.

Team Member	Task
Yiqing Cai, Man-ning Chen, Siddharth Raina & Huan- Yang Chang	Implement the geometry calibration code and evaluate the geometry calibration accuracy methods.
Sambuddha Sarkar	AEROTECH robot trajectory generation, velocity profiling & troubleshooting.

Table 3, Task Division.

4. FUTURE PLANS

My future plans until the next progress report is to write a trajectory code generator for Aerotech programming environment implemented in MATLAB which enables me to just enter the trajectory type and required parameters and it will generate a code which can be directly run by Motion Composer Suite of Aerotech. As I am principally done with my work on Aerotech's robot, I will help with the integration of AEROTECH robot and Camera setup (one button operation for capturing images). Also, I would keep studying the documentation on ABB's arm and their programming environment called 'Robot Studio'.