16-681 - MRSD PROJECT 1 | ILR #3

Individual Lab Report #1 | October 28, 2016

SAMBUDDHA SARKAR

TEAM G

EXCALIBR

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

1. INDIVIDUAL PROGRESS

1.1 Overview

As highlighted in the previous ILR, it is of paramount importance that one has the ability to plan and execute the desired trajectory of the calibration target in order to achieve a successful geometric calibration of the camera. The trajectory is embedded with set points at which the camera is triggered by the PSO to capture images of the calibration target. This dictates some amount of motion planning. The AEROTECH programming environment is equipped with command sets which come in handy for trajectory generation. In this ILR I would describe how I executed the motion of the robot in 3D space.

Topics covered have been listed below for a quick overview.

1.2 Continuous pulse output – Testing PSO

1.3 Multiple axis PSO triggering

1.4 Trajectory generation

1.2 Continuous pulse output – Testing PSO

This describes the use of the PSO pulse generator that is not tied to the position of the axis in any way. The pulse generator is configured and forced to fire continuously until it is manually disabled. This helps us test the PSO trigger operation beforehand.

The algorithm used is stated 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.

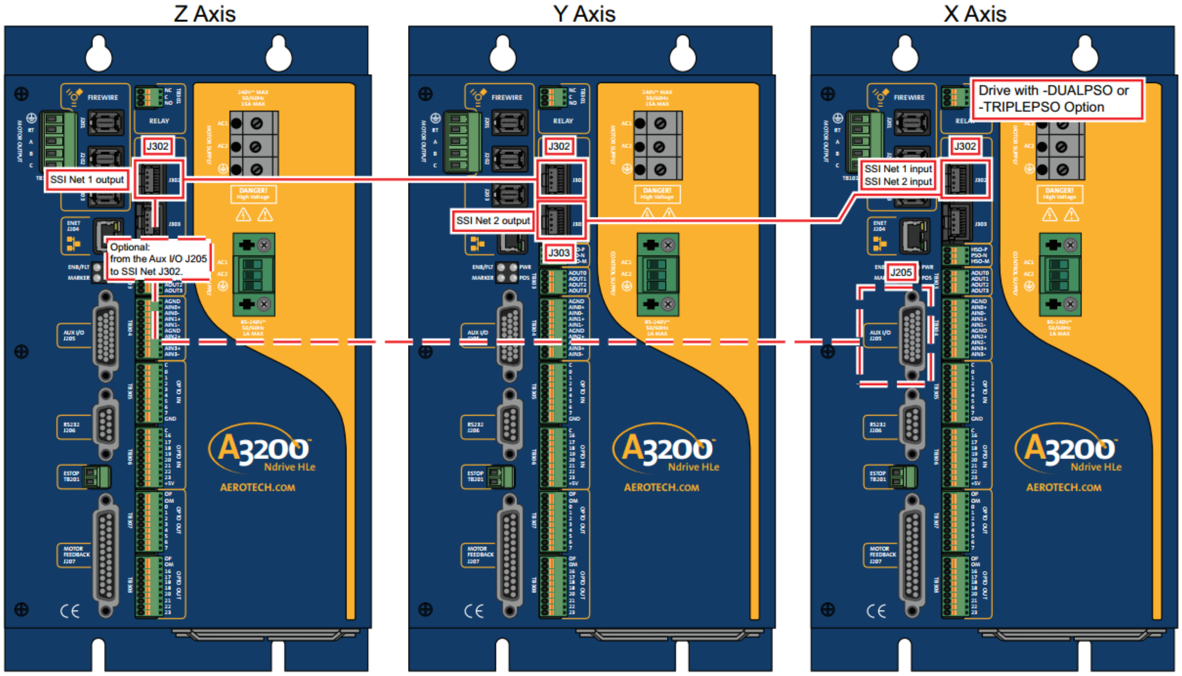
1.3 Multiple axis PSO triggering

Before diving into the algorithm of multiple axis triggering, a brief summary about the connection scheme is important. 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.

The PSO can be programmed to generate an output synchronized to the encoder position, typically used to fire a laser or sequence an external device (here, camera). Trigger signals may be derived from the standard encoder channel, auxiliary encoder channel, or a software trigger. The synchronized output pulse is generated using high-speed hardware, allowing minimal latency (200 nanoseconds) between the trigger condition and the output.

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 Z axis must have its SSINet port 1 configured as an output for its encoder feedback signal, and the Y axis must have its SSINet port 2 configured as an output for its encoder feedback signal. Since the Z axis does not use its SSINet port 2, it should be configured as an Input. Similarly, the Y axis does not use its SSINet port 1 so it should also be configured as an Input. The X axis must have both SSINet ports 1 and 2 configured as Inputs. In general, if an axis has an SSINet port configured as an output, then that SSINet port must be configured as an input on all other axes which it is connected to. Otherwise, encoder or marker data may be faulty. The connection scheme of the SSINet ports is shown in Fig. 1.3.

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.



NOTES:
J302 and J303 contain both SSI Net ports 1 and 2.
Use Nconnect-SSINet-XX cable between J302 and J303.
Use C19891-X or C19892-X cable between J302/J303 and J205.

Fig. 1.3, SSINet ports location and connection scheme.

SSINet1Setup and SSINet2Setup parameters must be set appropriately on all three axes and the SSI Net ports must be connected correctly in order for encoder information from the Y and Z axes to reach the X axis. The drive used for the X axis must have the -TRIPLEPSO option enabled.

The algorithm used for triple axis tracking is stated below:

1. Enable and home the axes.
2. Reset the PSO. This resets the internal state of the PSO hardware, including the distance tracking counters and the window counters.
3. 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.
4. Configure the PSO for three axis tracking using the standard encoder input, SSI Net Port1 and SSI Net Port 2. This configuration is only valid when using the Ndrive HPe or HLe.
5. Configure the PSO to generate a firing event at a fixed distance.
6. Configure the pulse generator by setting the total and off times and the number of cycles.
7. Specify the output of the PSO to come from the pulse generator.
8. Arm the PSO to start the distance tracking.
9. Move the axis so as to cause the PSO to fire.
10. Delay slightly to allow the PSO firing to finish.
11. Disable the PSO pulse generator.

1.4 Trajectory generation

Though it seems to be a tough problem at hand, the motion planning for the AEROTECH arm is pretty straight forward as the arm has to move in a fixed pattern in 3D space, in this case a sphere. I haven't gone into the details of the trajectory generation as most parts of it are beyond the scope of this report.

The algorithm used for trajectory generation is stated below:

Motion Composer:

1. Enter Sphere dimensions and set-points for trigger in Aerotech Motion Composer.
2. Execute motion in motion composer.
3. Check for errors and conflicts.

PSO Triggering & Motion:

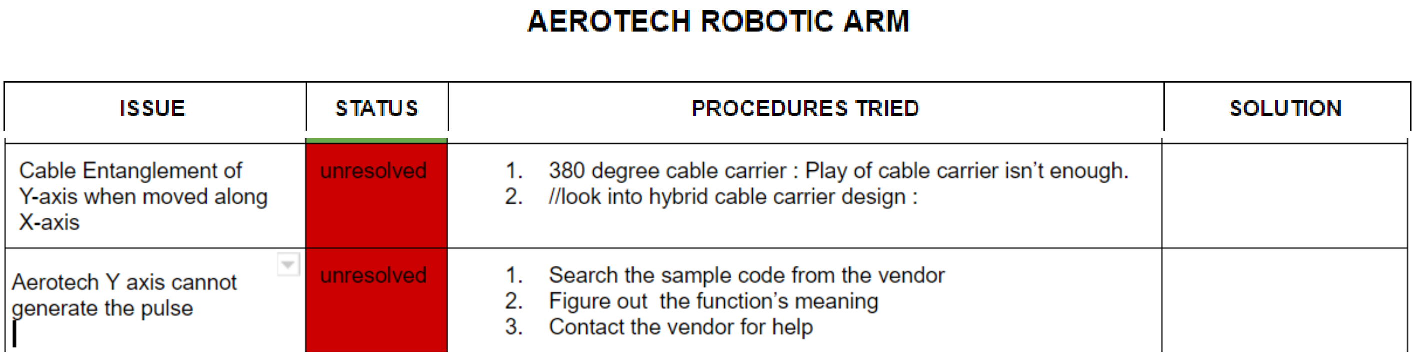
Setup PSO for multiple axis firing based on variable distances. An array containing the distance values is downloaded to the hardware and used in conjunction with individual-axis distance tracking. At each firing event, a user-configured output pulse stream is generated and a new firing distance is loaded from the array. After all of the array values are used, the hardware changes back to fixed-distance mode using the last value that was read from the array.

1. Define the number of distances we want to put in our array.
2. Fill the local array with the incremental firing distances.
3. Write the distance values from the local variables into the drive array.

4. Enable and home the axis.
5. Reset the PSO. This resets the internal state of the PSO hardware, including the distance tracking counters and the window counters.
6. 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.
7. Configure the PSO for multi-axis tracking from the primary encoder input.
8. Configure the PSO to generate a firing event based on distances in the array.
9. In this case the distances will be used starting at location 0 in the drive array.
10. Configure the pulse generator by setting the total and off times and the number of cycles.
11. Specify the output of the PSO to come from the pulse generator.
12. Arm the PSO to start the distance tracking.
13. Move the axis so as to cause the PSO to fire.
14. Disable the PSO pulse generator.

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.



AEROTECH ROBOTIC ARM

ISSUE	STATUS	PROCEDURES TRIED	SOLUTION
Cable Entanglement of Y-axis when moved along X-axis	unresolved	1. 380 degree cable carrier : Play of cable carrier isn't enough. 2. //look into hybrid cable carrier design :	
Aerotech Y axis cannot generate the pulse	unresolved	1. Search the sample code from the vendor 2. Figure out the function's meaning 3. Contact the vendor for help	

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	FPN algorithm, Gain filter and Median filter design and implementation.
Huan-Yang Chang	FPN & Geometric Calibration.
Sambuddha Sarkar	AEROTECH robot trajectory generation.

Table 3, Task Division.

4. FUTURE PLANS

My future plans until the next progress report is to get the AEROTECH robotic arm to get all the axes' PSO triggering. Another target of mine is to answer the problem of delay between the PSO trigger fire and the actual shutter control of the camera. Also I have to tackle cable management issues once the extra cable carriers arrive. I plan to get a head start on the ABB arm operation manuals and literatures because they seem to be extensive.