3.3 Data Analysis

When working with alpha waves, one of the biggest challenges is to determine the 'frequency bins' over which to perform the Fast Fourier Transform. As explained in Section 2.3, alpha waves occur in the 9 to 14 cycle frequency range, but the range of frequency 'bins' cannot be too wide (e.g. 9 to 14) because it obfuscates the calculations for alpha wave detection. Based on the graph outlined in Section 2.3, we decided to only consider frequency ranges from 9 up to but not including 10. Another challenge with alpha waves is determining the time length over which the data should be collected before analyzing it. A small time frame (for example, half a second) allows for quicker robot motion control, but since alpha waves do not occur at a constant rate, it induces instability in the motion as well. A larger time frame allows for a more stable motion control using alpha waves, but only up to a point - too long of a time frame makes it harder to find a clear limit between alpha waves and no alpha waves.

For the right vs left classifiers, there are various challenges that did not permit us to perform right or left robot motion using only brain signals. The data, even after processing it with a Fast Fourier Transform quickly overfit (even at a low number of iterations). This was due to our main source of OpenBCI data coming from only one team member, and only being taken for two continuous minutes each. A larger sample of data from multiple sources and taken at multiple occasions would have helped mitigate the overfitting. Furthermore, many unintended facial and body motions cause spikes in the data, among other unexpected behaviors that obfuscate a clear left vs right divide. Electrode placements are also hard to ensure that they stay constant, making the raw data very noisy.

3.4 Serial Communication

The main challenge in the serial communication was ensuring that the endian order was correct while parsing the incoming bits in the data packet. For a while, since the endian order was wrong during data collection, our results were noisy and impossible to convert into a viable task, but once the byte order was reversed, data collection and analysis was relatively fluid.

Another challenge was the activation of the notch filter implementation in the board. For a while, we were obtaining a strong 60Hz signal because of outlet interference during serial communication, but we were able to stop this 60Hz signal by enabling filters internal to the board (namely, the 60Hz signal).

3.5 Robot Control

One major challenge in implementing our control law was in implementing a path trajectory. To define a path, we define a list of positions so that the robot can cycle through these positions. Next, we define a threshold for the robot's tracking error so that the robot knows it has gotten close enough to the next position and can move to the next position in the path. Unfortunately, even after manipulating the gains, the robot would not reach our desired positions within reasonable thresholds because the internal gravity compensation of the robot was different than what we had in simulation. This meant that we had to interpolate positions in our path in order to accomplish the task and remove the threshold requirement.

Another implementation challenge was to control multiple directions with the robotic arm for Task 2. Because alpha waves are spuriously detected when the head or muscles on the face move, we had to specify in our implementation that `ALPHA_WAVE_VALUE` be set to zero when the magnitude of `ACCEL_VALUE` was above some threshold (we specified about $0.2g$). We had a bit of latency in our control due to the WAM's suboptimal gravity compensation (or, in the case of the demo, the alpha waves took a while to show up), but the robot eventually succumbs to our commands, which is a great sign.