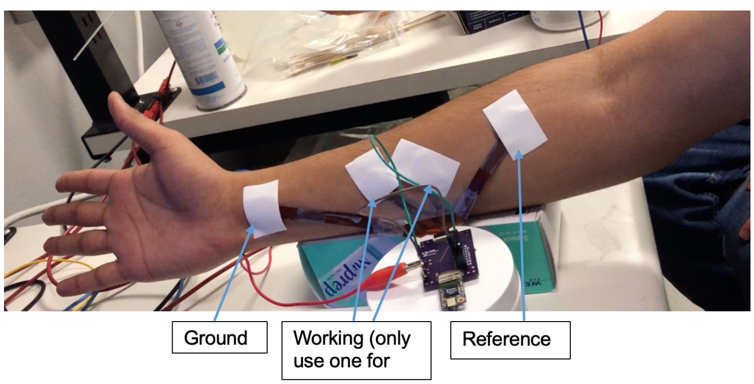
EMG Pong
ECE 202 Final Project
Mohsin Naqvi, Hongyu Lu, Wendy Yu, Josh Rosenberg

Introduction:
We implemented the classic 2-player Pong game in MATLAB using EMG (electromyography) signals as controls for each player's paddle. The EMG signals were recorded and streamed with an Intan RHD2000 electrophysiology amplifier and signal acquisition board. Our electrodes were custom graphene electrodes. EMG signals are amplified and then passed through a low pass filter before being normalized and mapped to each paddle's command.

System Overview:
Electrode setup: Our electrode setup is shown above. We record EMG activity from the brachioradialis and flexor carpi forearm muscles. Our electrode was originally designed for recording EEG but fit our application nicely. The second electrode array is laid out similarly with slightly differing geometry. Both electrodes interface with Intan preamplifiers before being read onto the Intan RHD2000 board.



MATLAB Process:
The main process of our game is controlled by a timer object set to 30Hz and with custom startup and loop functions. Upon starting the timer, it executes the startup function, which we set to calibrate the controls. Once the startup function completes, the timer executes the looping function at the predefined frequency to run the main game. To stop the timer, the loop function checks if the game is over and deletes the timer.

Startup: During startup, our code calibrates the paddle speed to the players' EMG. This is done by recording the player's EMG signal at two states: resting state (muscle relaxed) and active state (fist clenched). This signal is then filtered and the square root of the power of these two signals is calculated and stored as the minimum and maximum controls respectively. These controls define the paddle speed: the recorded minimum corresponds to maximum downward speed, and the recorded maximum corresponds to the maximum upward speed. This process is then repeated for the second player.

Loop: In the looping section of our code, 3 separate processes occur: EMG data is recorded from the Intan board, object states are recalculated, and all game objects are redrawn. In the first process, the EMG signal from both players over the current loop period is read from the Intan buffer. The square root of the power of the signal is calculated and the resulting value is scaled and mapped between our calibrated limits. Any values that exceed the determined limits are clipped, preventing the player from moving the paddles above the maximum speed in the upwards or downwards direction.

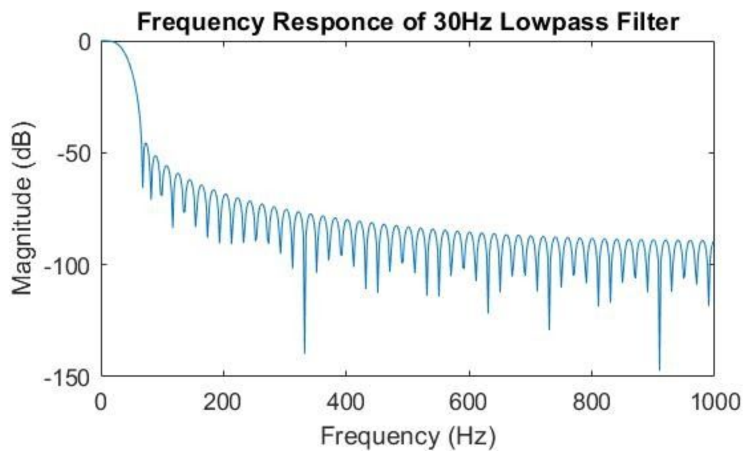
In the second phase stage, all objects states are recalculated. This is accomplished by storing the current and previous position of each object. The speed and direction of each object can be calculated from the stored positions, which can then be used to calculate the new position. During this stage, we must also check each object for collisions. Collisions occur when the boundaries of one object overlap the boundaries of another object, and are treated slightly differently for the paddles versus the ball. For the paddles, a collision with the ball causes no change in position. However, a collision with a wall stops the paddle from continuing to move in its current direction. For the ball, collisions can occur off of the walls or off of the paddles. A collision with any of these objects will result in a recalculation of the ball's new position: the new position is calculated by mirroring the old position with respect to the line that is orthogonal to the collision surface and summed with a fraction of relative speed difference multiplied by the time interval. The endgame criteria is evaluated by checking if the ball collides with any of the walls behind the paddles.. If this condition is realized, the game is ended, the timer is cleared, and the game over screen is shown.

In the final stage of the loop, the updated positions from the previous stage are used to replot each object on the same set of axes. This prevents issues with the MATLAB plot rescaling in the middle of the game.

Signal Processing:

Filter: Before calculating the speed of each paddle, the raw EMG data from each player is filtered through a low pass finite impulse response filter using a Kaiser window. We chose this particular filter because a Kaiser window filter maximizes the ratio of the pass lobe power to the side lobe power. For a Kaiser window, the pass lobe is the lobe that encompasses the pass band in the frequency response, whereas the side lobes occur in the attenuated portion of the frequency response, corresponding to the stop band.

However, our filter had transient effects that caused an error at the beginning of our filtered signal due to the impulse response of the filter. To get rid of the effect of filter transient, which is caused by the initialization of FIR filter, the first 100 samples of the output signal is discarded. To obtain a raw data sequence longer than 100 samples, we chose to periodically extend each data block 5 times before passing it through our filter. High-frequency components at the edge of 2 consecutive repetitions are limited by the lowpass filter.



EMG Signal Power: For the paddle controls, we decided to calculate the square root of the signal power due to its linear relationship to muscle strength. In our research, we found that the exerted muscle strength is proportional to the root mean square of the EMG signal[1]. The root mean square is calculated as:

$$\sqrt{\underbrace{\frac{1}{t_2-t_1}}_{\text{Loop Period}} \underbrace{\int_{t_1}^{t_2}|f(t)|^2}_{\text{EMG signal Power}}}$$

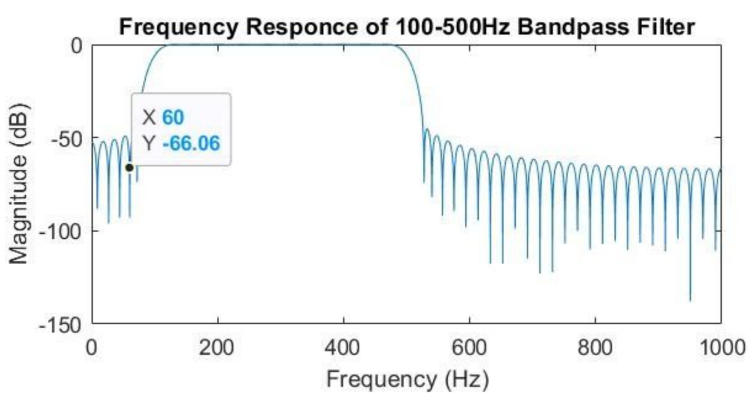
Where the loop period is constant (defined by our 30Hz loopin frequency). This allows us to linearly map exerted strength to the speed of the pong paddles by scaling by the calculated signal power by the calibrated maximum and minimum values.

Results:

We would have a fairly good control over the paddles if both reference and signal channels have high signal-to-noise ratio. We found out that calibration had a significant effect on the gameplay experience. If the user calibrated the maximum upward speed while clenching fist with maximum effort, the paddle might not reach the upper bound of the game window during the actual game. We recommend the user to use less effort during the calibration of the maximum upward speed in order to have more control over the paddle in game. It was also shown that motion artifacts had a high impact on the calibration: motion artifacts during calibration could result in improper mapping of the controls, sometimes resulting in the inability of the players to control their paddle. High background noise from the Intan board also contributed signal power, and as a result, could interfere with calibration, especially in the relaxed state. This would result in the paddles moving outside of the players control, or getting stuck. The latency is calculated from the delay between muscle exertion and paddle movement by counting the number of frames of 2 video clips synchronized by their audio track. The latency of our game using the bandpass filter was 6 frames recording at 60fps, equivalent to 100ms. Using the low pass filter our latency was 7 frames at 60fps, which equals 117ms.

Conclusion:

The main issue with our design was with motion artifacts: our low pass filter removes the high frequency component of the EMG signal, leaving components that occur in the same frequency range as motion artifacts. As a result, motion artifacts had a strong influence over control calibration and over paddle control. To mitigate these effects, a high pass filter to remove the low frequency motion artifact component is desirable. We redesigned our filter such that it removes all motion artifacts and provides significant attenuation at 60Hz by selecting a Kaiser window 100-500Hz bandpass filter; beta is carefully chosen such that 60Hz noise is placed at the null between stopband lobes. With the redesigned filter, our game was no longer susceptible to motion artifacts - the paddles would exhibit no response to physical motion of the board or the player's arm in the absence of muscle exertion. This improved calibration of the controls, and as a result, playability of the game: paddles were easier to control and would no longer become stuck. Our system was more robust to calibration error so that poor calibration did not result in the inability of a player to move their paddle. However, poor signal or high noise from the board could still cause control error. <u>This filter was only implemented after the fact and was not present during our demo.</u>



References:

1. H. Li, G. Zhao, Y. Zhou, X. Chen, Z. Ji, and L. Wang, "Relationship of EMG/SMG features and muscle strength level: an exploratory study on tibialis anterior muscles during plantar-flexion among hemiplegia patients," *BioMedical Engineering OnLine*, 2014.