BME/EECS 458 F-2022 - Chang Li
Module 6: Design Project

PDF Version generated by

Chang Li

on

Dec 07, 2022 @04:06 PM EST

Table of Contents

Objective

Chang Li - Dec 05, 2022, 7:07 PM EST

The main objective of this project was to create an EMG-based interface for playing video games that can be used by people who may lack the fine motor control needed for traditional video games (ie, those with Parkinson's Disease). In terms of hardware, the main objective was to both amplify and filter the raw EMG signal to obtain a signal that clearly showed when the user was contracting their muscle. The main objective of the software portion of this project was to detect when muscle flexion was occurring and determine which limb was being contracted in order to move the snake in the video game in the correct direction.

Introduction and Materials

Chang Li - Dec 05, 2022, 7:10 PM EST

Introduction

In this lab, we are creating a system where users can control keys on the keyboard to play simple games using their limbs. For people struggling with fine motor skills due to diseases such as Parkinson's, it can be tough to use the keyboard in daily life, so even simple pleasures like computer games can become extremely difficult for them to use. We designed a system with 2 electrodes on each limb and 1 electrode on the knee so that by flexing their muscles on their limbs, people can control the WASD keys and play games. Although this system is intended for games and WASD, the controls can be easily altered to apply to anything and even control the arrow keys. To accomplish our goal, we set up hardware with filters and differential amplifiers for each of the limbs. The output of the filter would then go into an Arduino which leads to the computer. In the software portion, the Arduino readings are acquired and quickly analyzed to press the key corresponding to the limb that was activated in the sample time.

Indicate your role in the project

My role in the project is Hardware engineer.

Lab Partners:

Daniel Najarian - Software

Michael Stiffler - Software

Ritika Pansare - Hardware

Chang Li - Hardware

Equipment List

- Electronic Components
  - AD620 op-amp (4)
  - LM 741 op-amp (4)
  - 2.4kΩ resistor (4), actual: 2.388kΩ, 2.3923kΩ, 2.3920kΩ and 2.3907kΩ
  - 110kΩ resistor (4), actual: 109.60kΩ, 109.93kΩ, 109.81kΩ and 109.60kΩ
  - 3kΩ resistor (4), actual: 3.0004kΩ, 3.0067kΩ, 2.9982kΩ and 3.004kΩ
  - 0.6μF capacitor (4), actual: 0.602μF, 0.601μF, 0.599 μF and 0.602 μF
  - 10nF capacitor (4), actual:10.14nF, 10.10nF, 10.04nF and 10.14nF
  - Arduino Leonardo (1)
- System Setup Equipment
  - Black, red, and green wires (1 spool each)
  - Breadboard (1)
  - Wire strippers (1)
  - EMG stick-on electrodes (7)

- Alcohol wipes (1)
- Arduino cable (1)

- Electronic Test Equipment
  - Agilent 33220A Function Generator (1)
  - Tektronix TDS 2012 Two-Channel Digital Storage Oscilloscope (1)
  - Multimeter (1)
  - Agilent E3630AE7 Power Supply (1)
- Software
  - Google Sheets
  - MATLAB
- Computer (1)

Preliminary Analysis

Chang Li - Dec 07, 2022, 3:14 PM EST

Design and Construct Signal-Conditioning Circuit

1. Design each of the three circuits to have a combined gain of 500 (20x in AD620 stage and 25x in LM741 stage) and a band-pass cut-off frequency of 90Hz and 150Hz.
2. After calculation, we decide to choose R_G = 2.4kΩ, R_1 = 3.1kΩ, R_2 = 106kΩ, C_1 = 0.6μF, C_2= 10nF. The detailed calculation is shown below:

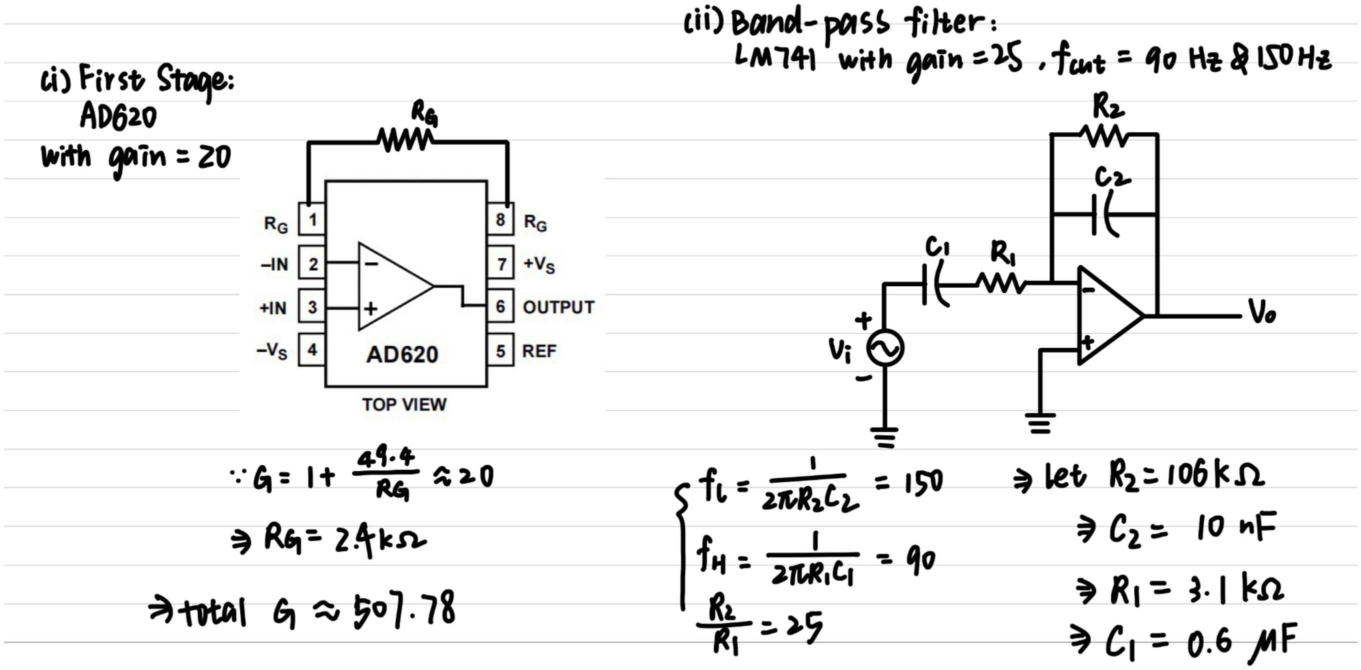


Fig.1 Calculation Process of Preliminary Circuit Analysis

Experimental Setup

Chang Li - Dec 07, 2022, 3:34 PM EST

Hardware

For the hardware set-up of this lab, we used two electrodes on each bicep and calf muscle. An additional electrode was used on the knee as a physiological ground. The signal from each of these electrodes was fed into our conditioning circuit.

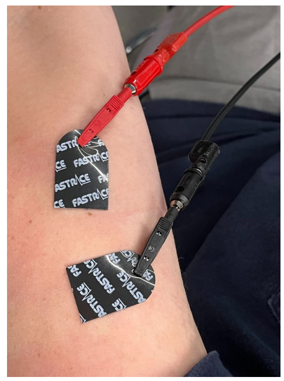


Fig.2 Connection of Electrodes on Each Muscle

The conditioning circuit consisted of a two-stage amplifying and filtering circuit. The overall gain of the circuit was 500x and the passband was 90-150Hz. We found that even though EMG signals exist in the 50-150 Hz range, having our high-pass frequency cutoff set to 90Hz meant that we would eliminate any 60Hz noise, giving us a cleaner signal. The first stage was made using a differential amplifier (AD620) and had a gain of 10. The signals from both electrodes on one muscle were fed into this stage of the circuit, so the amplifier could take the difference between the two electrodes. The second stage of the circuit was a bandpass filter with a gain of 50x and a passband of 90-150Hz. This two-stage amplifying circuit was repeated four times, once for each limb.

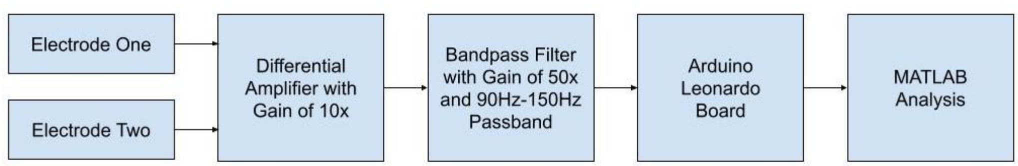


Fig.3 Hardware Functional Block Diagram

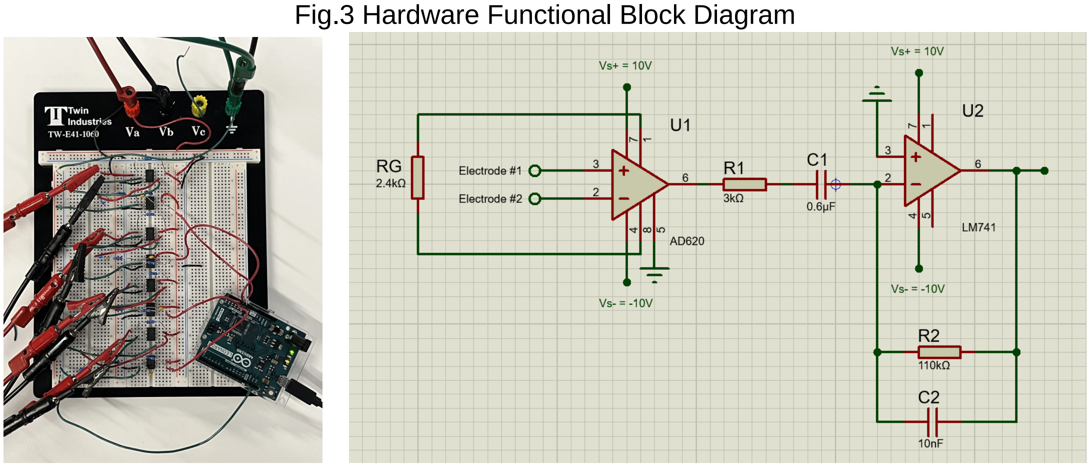


Fig.4 Circuit Schematic

We then use function generator and oscilloscope to test the signal-conditioning circuit for the four filters separately, and build corresponding Bode Plots. We choose sine wave as the input waveform, and the acquired raw data for the four identical filters are shown in Table 1, Table 2, Table 3 and Table 4 respectively.

Table 1 - Test Data for Filter 1 (Left Arm)

Frequency (Hz)	Vin (Vpp)	Vout (Vpp)	Gain	Gain1(dB)
30	0.02	4.4	220	46.84845362
50	0.02	6.4	320	50.10299957
70	0.02	6.8	340	50.62957834
80	0.02	8	400	52.04119983
85	0.02	8	400	52.04119983
90	0.02	8.6	430	52.66936911
100	0.02	8.6	430	52.66936911
110	0.02	9	450	53.06425028
120	0.02	9	450	53.06425028
140	0.02	9	450	53.06425028
150	0.02	9	450	53.06425028
155	0.02	8.8	440	52.86905353
160	0.02	8.8	440	52.86905353
170	0.02	8.6	430	52.66936911

Table 2 - Test Data for Filter 2 (Right Arm)

Frequency (Hz)	Vin (Vpp)	Vout (Vpp)	Gain	Gain2(dB)
30	0.02	4.4	220	46.84845362
50	0.02	6.4	320	50.10299957
70	0.02	7.6	380	51.59567193
80	0.02	8.2	410	52.25567713
85	0.02	8.4	420	52.46498581
90	0.02	8.4	420	52.46498581
100	0.02	8.8	440	52.86905353
110	0.02	8.8	440	52.86905353
120	0.02	9	450	53.06425028
140	0.02	9	450	53.06425028
150	0.02	8.8	440	52.86905353
155	0.02	8.8	440	52.86905353
160	0.02	8.4	420	52.46498581
170	0.02	8.2	410	52.25567713

Table 3 - Test Data for Filter 3 (Left Leg)

Frequency (Hz)	Vin (Vpp)	Vout (Vpp)	Gain	Gain3(dB)
30	0.02	4.4	220	46.84845362
50	0.02	6.2	310	49.82723388
70	0.02	7.6	380	51.59567193
80	0.02	8.2	410	52.25567713
85	0.02	8.4	420	52.46498581
90	0.02	8.6	430	52.66936911
100	0.02	8.6	430	52.66936911
110	0.02	8.8	440	52.86905353
120	0.02	9	450	53.06425028
140	0.02	9	450	53.06425028
150	0.02	9	450	53.06425028
155	0.02	8.8	440	52.86905353

160	0.02	8.6	430	52.66936911
170	0.02	8.4	420	52.46498581

Table 4 - Test Data for Filter 4 (Right Leg)

Frequency (Hz)	Vin (Vpp)	Vout (Vpp)	Gain	Gain4(dB)
30	0.02	3.6	180	45.1054501
50	0.02	5.4	270	48.62727528
70	0.02	6.8	340	50.62957834
80	0.02	7	350	50.88136089
85	0.02	7.4	370	51.36403448
90	0.02	7.8	390	51.82129214
100	0.02	7.8	390	51.82129214
110	0.02	8	400	52.04119983
120	0.02	8	400	52.04119983
140	0.02	8	400	52.04119983
150	0.02	8	400	52.04119983
155	0.02	7.8	390	51.82129214
160	0.02	7.6	380	51.59567193
170	0.02	7.4	370	51.36403448

And the corresponding Bode Plot we obtain is shown below in Fig.5.

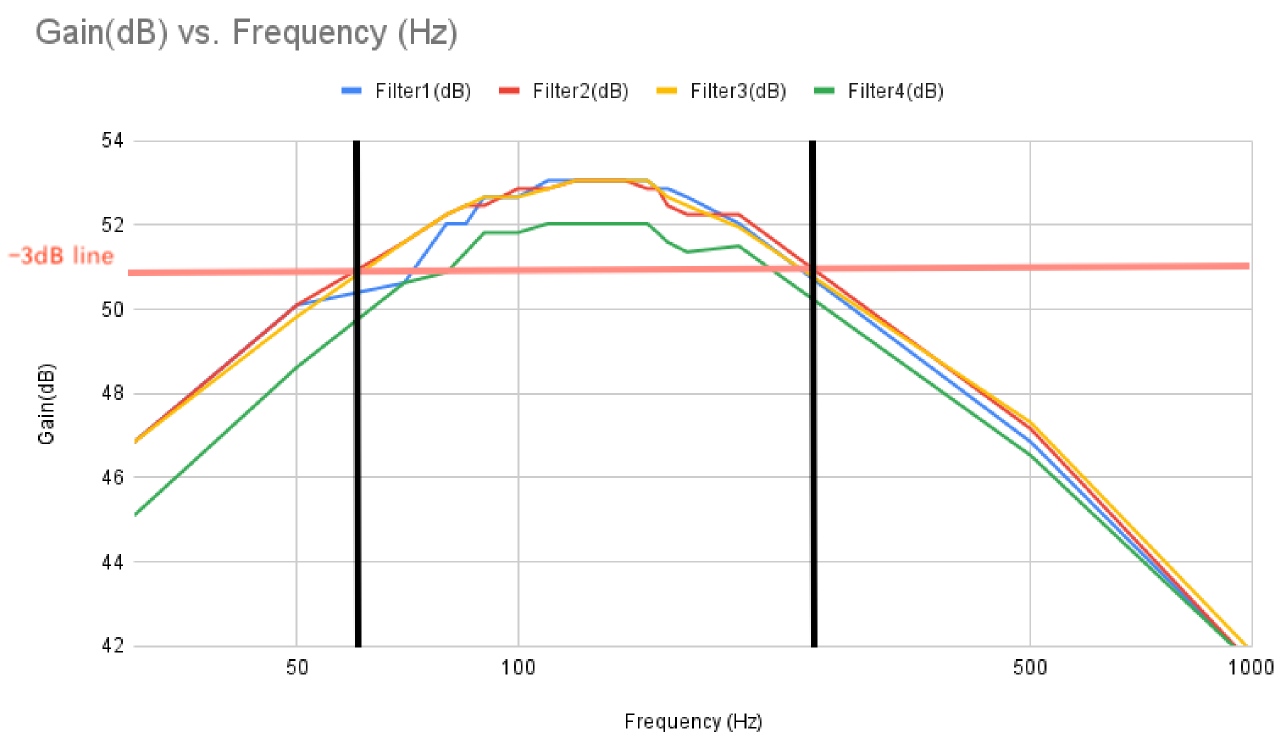


Fig.5 Bode Plots of Four Identical Band-pass Filters with Passband Set as 90Hz and 150Hz

According to the Bode Plots and the raw data, we can see that the circuit behaves as expected: The cut-off frequencies for each identical band-pass filters are approximately 90Hz and 150Hz, and gain is approximately 53, which is close to 20*log (500) = 53.98, where 500 is the gain we design.

Therefore, the whole circuit can filter the undesired noises beyond 90Hz-150Hz as well as amplify the input signal by 500, which serves our purpose.

After passing the signal conditioning circuits, the processed EMG signals are then fed into the Arduino Leonardo board and prepared for MATLAB analysis.

<u>Software</u>

As seen in the figure below, the first step was to read the data from the Arduino Leonardo into a MATLAB matrix. We took 10 samples per limb and used this as the dataset for each processing interval. We found the thresholds experimentally to be 0.05 so that noise would not trigger keys to be pressed but also so that it is not high to the point that data goes missing. We found the most frequently activated limb in that sample time and used that to press the corresponding keys on the keyboard. For the left arm, right arm, left leg, and right leg, the keys were W, S, A, and D respectively. At the end of this cycle, the number of activations per limb is reported in the command window of matlab and plots are made for the users to see the data in case they need to make adjustments to the thresholds in the future. The process is in an infinite loop so the users can use it for as long as they need.

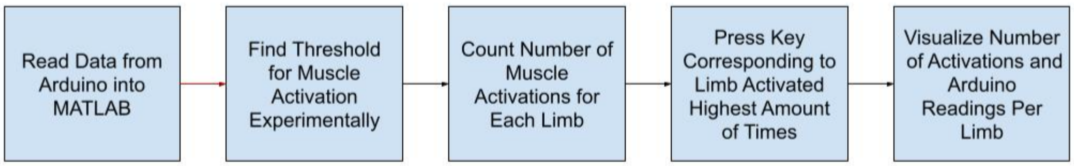


Fig.6 Flow Chart for LabVIEW

For the command to trigger the key in MATLAB, we had to import the external java.awt.Robot library. From this library we were able to access commands such as `robot.keyPress(java.awt.event.KeyEvent.VK_A)`and `robot.keyRelease(java.awt.event.KeyEvent.VK_A)` which press and release the ‘A’ key respectively using subroutines.

Results

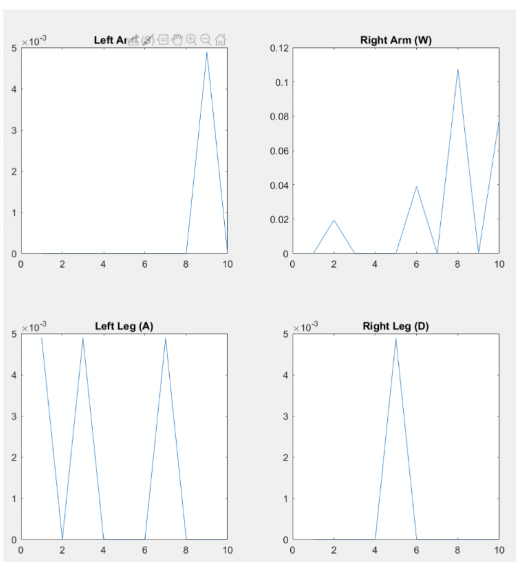
Chang Li - Dec 07, 2022, 4:05 PM EST

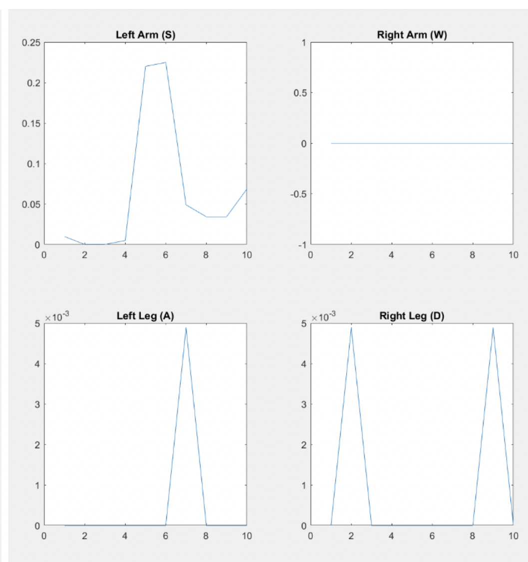
Data Collection

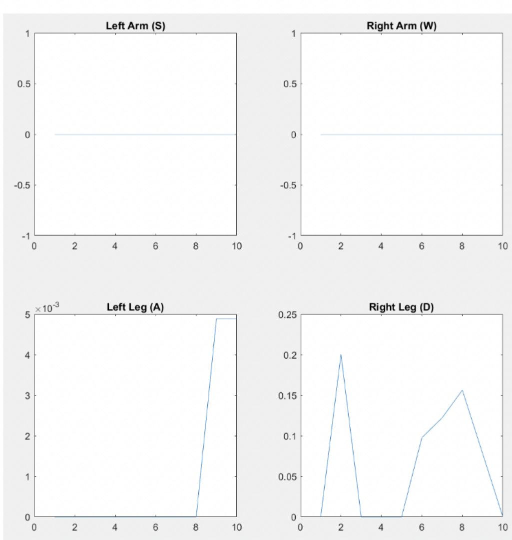
The data collection for our project was fairly straightforward. Much of our measurement of success was based on qualitative observations rather than quantitative. With this being said, we did need to collect some quantitative data. The first data collected was data for the outputs of each of our four circuits that we made. We tested each individual circuit we made using the function generator at different frequencies and measured the output voltage at each frequency to ensure that our circuits were performing as they should. Other data we collected was the output voltage for our entire system. This data was collected through Matlab by using graphs and measuring how many times each circuit's output passed a certain threshold. We continued to do these measurements and adjust the threshold used until we were satisfied with our value.

Raw Data

Below are the output voltage graphs for our entire system obtained through MATLAB code.







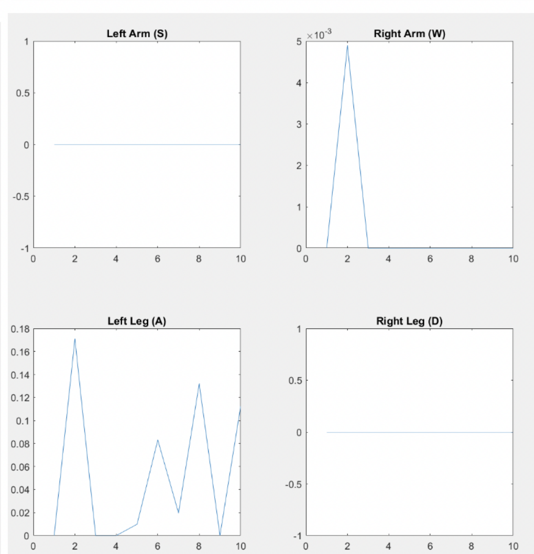


Fig.7 Output Graphs from MATLAB

(Right Arm, Left Arm, Right Leg, and Left Leg Activity Respectively)

Qualitative Observations

The main qualitative observations that we made were with regards to the code working. Since we were using EMG as a form of keyboard manipulation, we were able to tell if our system was working by seeing if we were able to type certain keys on a blank text file. Depending on what occurred in the text file, we were then able to edit the threshold voltages to make the system either more or less sensitive to movement. We consistently got higher peaks from the arm electrodes rather than the leg electrodes, so we eventually made the leg thresholds lower than the arms. After this change, the system worked as intended.

Discussion and Conclusion

The overall goal of this project was to develop an EMG-based gaming experience for users who have limited fine motor skills. The project utilized EMG signals from four limbs to activate keys on a computer keyboard. The user placed two electrodes on each of their bicep muscles and calf muscles, as well as another electrode on the knee to serve as a ground. The signals from each muscle were fed through a conditioning circuit to both amplify and filter them, then they were ultimately fed into MATLAB for further processing and analysis. In MATLAB, activation thresholds were determined experimentally for each muscle. Each limb corresponded to a particular key on the keyboard so that as the user contracted a particular limb, the avatar in the video game would move in the corresponding direction.

Summary

Chang Li - Dec 07, 2022, 3:59 PM EST

In this module, we create an EMG system where users can control keys on the keyboard to play simple games using their limbs. To achieve our objectives, we design a hardware part to acquire and process the EMG signals from the user, and use Arduino Leonardo board together with a MATLAB matrix to analyze the EMG signals. We learned how to acquire EMG signals by attaching two electrodes to each of the user's muscle, and we also strengthened our ability to design the signal conditioning circuits according to our own needs. We got familiar with Arduino Leonardo board and this is the first time we use MATLAB to acquire and analyze signals. Most of the experimental results meet our expectations, the user can play real-time video game by using their limbs, and the system also operates with minimal lag. There were very few instances when the system would not detect a muscle contraction or would incorrectly detect a contraction in the wrong limb. The only goal we didn't achieve was to create a video game on our own through LabVIEW. Instead of that, we decided to generalize the system so that it can be used to play any existing keyboard video games which require no more than 4 control keys.

The most difficult part we found was that how to eliminate the noises and make the EMG signals cleaner. We found that the frequency range of EMG signals lies in 50Hz - 150Hz, so we designed the passband of our band-pass filter as 50Hz and 150Hz initially. However ,the signal we got is quite noisy. So we built our new band-pass filter with the high-pass cutoff frequency set as 90Hz, so that we can get rid of the noises around 60Hz. After changing the design of our band-pass filter, the signals became much cleaner and had distinct thresholds.