University of Rhode Island

DigitalCommons@URI

Open Access Master's Theses

2019

STATE MACHINE APPROACH FOR GESTURE CLASSIFICATION

David O. Cipoletta

University of Rhode Island, davidred15@gmail.com

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Recommended Citation

Cipoletta, David O., "STATE MACHINE APPROACH FOR GESTURE CLASSIFICATION" (2019). *Open Access Master's Theses*. Paper 1741.

<https://digitalcommons.uri.edu/theses/1741>

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

STATE MACHINE APPROACH FOR GESTURE
CLASSIFICATION

BY

DAVID O CIPOLETTA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2019

MASTER OF SCIENCE THESIS

OF

DAVID O CIPOLETTA

APPROVED:

Thesis Committee:

Major Professor Frederick J. Vetter

Qing Yang

Manbir Sing Sodhi

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2019

ABSTRACT

There are many new developments in technology in the last decades. Even though the world is moving towards more autonomy, the human in the loop control of technology is still crucial. To improve efficiency and advance the technology further, there are new ways of human to machine interfaces (HMI) being developed: for example, eye control, voice control, brain control interface (BCI), gesture control, etc. All of these new paradigms of control face some common issues such as adoption, learning curve, and reliability. The goal for this project is to demonstrate a new state machine approach for gesture classification and demonstrate a use case for gesture classification.

ACKNOWLEDGMENTS

There are no words that can describe how grateful I am to the entire department of Electrical Computer and Biomedical Engineering and the graduate school of the University of Rhode Island. For the past three years, I was able to pursue my dream to create and run a technology company. It went from 2 people to a team of 20+ people, all due to the support I have gotten from the ECBE department and the Graduate School.

My journey first started with Professor Vetter, who is the graduate director for ECBE who supported me with my decision to pursue a master's while working full time at Charles River Analytics. Then when I took Professor Mankodiya's wearable IoT course I met Bob Foster, who was just diagnosed with ALS, and soon we became friends. With the support of Professor Mankodiya, my team and I were able to create an assistive robot controlled using eye motion. During this class, Professor Mankodiya and Mr. Tom Kowalczyk encouraged me to venture off and start my business. After meeting Dexter Ang shortly after the class ended, I quit my job and began Pison; during this time I did not want to give up my education at URI, and Professor Vetter once again was supportive and understanding of my situation and was able to meet my needs with a leave of absence and strategies for my graduation requirements. During my part-time master's at URI, I was able to take some very relevant courses for my company where I can hone the skills I did not have. Professor Shahriari's BCI course is one of them. Her vision of helping people with disabilities in her lab have tied in with my company's vision very closely. Her course has taught me some of the machine learning techniques that I was able to experiment with our sensors. Professor

Yang is always extremely supportive of both my time at URI and also my entrepreneurship since he has gone through the journey of running a company and has given me information with common investors. Lastly, Professor Sodhi, has also been very accommodating with my hectic schedule as one of my committee members.

I would also like to thank the students of ECBE, Professor Fischer, Professor Sunak, Tanya Wang, my family, and friends for the continued support and encouragement, which helped me to overcome many hurdles and achieve where I am today.

TABLE OF CONTENTS

ABSTRACT.....	ii
ACKNOWLEDGMENTS.....	iii
TABLE OF CONTENTS.....	v
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
CHAPTER 1.....	1
INTRODUCTION.....	1
CHAPTER 2.....	4
REVIEW OF LITERATURE.....	4
CHAPTER 3.....	11
METHODOLOGY.....	11
CHAPTER 4.....	35
CONCLUSION.....	35
CHAPTER 5.....	40
FUTURE WORKS.....	40
APPENDICES.....	41
BIBLIOGRAPHY.....	43

LIST OF TABLES

TABLE	PAGE
Table 1. Raw classification vs. Filtered Classification Data	22
Table 2. List of Actions to Classify.....	24
Table 3. TPR and FPR Calculation Formula.....	25
Table 4. Configuration for Action.....	32
Table 5. Configuration for IMU.....	33
Table 6. Types of Feedback.....	33
Table 7. Feedback Configuration.....	34

LIST OF FIGURES

FIGURE	PAGE
Figure 1. Myo Gestures.....	6
Figure 2. Classification of Myo gesture from “Hand Gesture Recognition Using Machine Learning and the Myo Armband”	7
Figure 3. Surface electromyographic recording showing the effects of a cell phone located 1 cm (0.4 inches) away from the electrodes.....	9
Figure 4. Surface electromyographic recording showing the effects of a cell phone located 30 cm (1.2 inches) away from the electrodes.....	9
Figure 5. Surface electromyographic recording showing the effects of a cell phone ringing at different distances.....	10
Figure 6. Euler Representation.....	13
Figure 7. Projected Pitch.....	16
Figure 8. Projected Yaw.....	17
Figure 9. Projected Roll.....	18
Figure 10. Comparison of ideal vs. practical anti-aliasing filter.....	20
Figure 11. Effect of choice of cut-off frequency on the gain at 300 Hz for a first order.	22
Figure 12. EMI vs. no EMI of Index Finger Clicking.....	23
Figure 13. Frequency Spectrum of EMI vs. no Emi of Index Finger Clicking.....	23
Figure 14. Bandstop Filtering Effect on EM I noise.....	24
Figure 15. ROC Curve (the left graph is raw data, the right graph is filtered data)....	26

Figure 16. Example Signal collected with Pison Device.....	27
Figure 17. Example of feature values for testing data. The green box denotes the action number (refer to the Action set table). Note that different level of values is generated for different actions.....	29
Figure 18. Example of sensitivity vs. TPR and FPR for Pison device. Sensitivity value of 0.4 would be chosen, which would yield 100% TPR and 34% FPR.....	30
Figure 19. Average Energy Difference Between 2 Channels.....	31
Figure 20. State Machine Design of the Box Classifier.....	33
Figure 21. Box Classifier UI	34
Figure 22. Gesture State Machine.....	35

CHAPTER 1

INTRODUCTION

1.1 Overview and Motivation

Currently, wearable gesture sensor classification is done using pattern recognition techniques. An existing product such as Myo armband[1] requires the classifier to first be trained by having the user perform the gesture multiple times and labeling the sensor data for the gesture the user is performing. The classifier then produces a verdict for the classification. This classification method is very similar to pattern recognition for voice-based control. This technique has many drawbacks. First, the accuracy of the classification depends on the amount of labeled data set and the consistency of the pattern. Second, the user does not get confirmation of the complete gesture unless the motion sequence is finished. Third, the classifier is typically overfit if labeled data are done consistently. Lastly for the general population no one wants to train the gesture over and over again.

The significance of this study is to propose a new way to classify user gestures that is not a pattern recognition technique. Pattern recognition limits the number of gestures the user can perform since there is a requirement for a training set of data for every gesture the classifier tries to accomplish. From the paper "A Survey on Hand Gesture Recognition in the Context of Soft Computing" by Cheng[2], a camera-based recognition approach is needed to recognize the gestures. The developers have to create classes for each of the hand signs they are trying to

classify. Users need to perform the gestures for the trained model correctly every time. The pattern recognition technique is not ideal for movement-based gesture classification, especially when the user's movement can be different due to muscle fatigue or other environmental factors. Lastly, in the pattern recognition approach for movement gestures, the user continuously performs the gesture, and the result of the classification is feedback to the user. The user will have to complete the gesture fluently in the first attempt, and if the user performs any motion incorrectly during the gesture, there is no feedback mechanism to correct the user, and this demoralizes the user in performing the gesture. Due to these drawbacks of the pattern recognition technique, there has not been a gesture platform that can work for everyone, or that is customizable for the user.

1.2 Objective and Scope

The goal is to develop a human to machine interface (HMI) that is easy to train and provides immediate user feedback. The objective of this project demonstrates a heuristic classification technique to classify gestures which allow user adjustability, and the combination of the binary classifier with the motion to create a mouse-like HMI to allow multiple gestures using a combination of the inertial measurement unit (IMU) and bio-potential sensor. The goal of this project is to show the ease of use with this state machine approach. Each state is providing immediate feedback, and the user is not required to repeat the same action during the training to allow for flexibility of user intention. Since everyone moves slightly different, there is no intention to control user behavior but rather teach the user how to gesture. Finally for this project is to do all the classification at the top of

the wrist. Since this is more acceptable form factor. The project will utilize a binary click action via a bio-potential sensor and IMU for more gradient control.

The reason the bio-potential and IMU sensor is selected for this project is to demonstrate the simplicity of using gestures. Though many novel new interfaces have been developed with the new gesture, there is a problem with the adoption rate of these gestures. A couple of reasons for this is due to the number of gestures that the user has to learn and that the acceptance of the movement may not be natural to the user. Therefore simplicity should be the key to creating gestures that are adoptable by the user population. For example, the computer mouse was originally created with one button and a tracking sensor to suffice control on a 2D interface. In this scenario, a 2D interface was developed to complement the mouse and gained user adoption. The instant feedback when a user moves the mouse and the cursor moves as well as when the user clicks the button, and audio feedback was created or when a digital object becomes highlighted due to the click. The action followed by the feedback created a closed-loop system to allow a user to learn how to use this technology instantly. Therefore, this project will demonstrate that by creating a combination of actions and feedback with another action and feedback, a user should be able to create multiple simple gestures.

It will be difficult to test the performance of this approach, since there are not many commercial off the shelf devices that allow the user the customizability to adjust for personal user gestures. Many gesture systems currently available attempt to gain user adoption by teaching them how to gesture, and these have not succeeded due to the tedious training and calibration process as well as the

reliability of the gesture classifier. Therefore, true accuracy should not be the primary goal of this project. Rather this project should be focused on the novel state machine approach to allow customization as well as immediate feedback to create acceptable user adoption.

List of References

- [1] Thalmic Labs. “Myo armband gesture control - wearable technology by Thalmic Labs.” 2015. [Online]. Available: <https://www.myo.com/>
- [2] Cheng, H., Yang, L., & Liu, Z. (2016). Survey on 3D Hand Gesture Recognition. IEEE Trans. Circuits Syst. Video Techn., 26(9), 1659-1673.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Literature Review

Much research has been focused on gesture control applications as well as the usage of a combination of bio-potential and inertial measurement units (IMU). Typically the IMU is a combination of accelerometers, gyroscope, and magnetometer. The fusion of the three sensors can provide a viable 3 degree of freedom output, such as yaw, pitch, and roll. An example would be “Mouse HCI through combined EMG and IMU” [3]. However, the approach is to leverage the existing control interface and adapt the IMU and bio-potential as an alternative sensor to the optical tracking and physical button as a replacement. Forbes concluded [3], the combination fell short in comparison to the usability of the device versus a mouse. There are many similar approaches where the novelty of utilizing a binary output and an analog output to create a user interface we know today as the mouse. The “Magic Mouse” [4] is another example where the author created a pair of gloves for both hands; each glove contains IMU and multiple contact sensors and is fully tethered to the processing computer. Once again, it is not sufficient as a mouse replacement. The problem seems to be translating the 3 degrees of freedom to the two-dimensional screen of the computer, and this translation can be rather difficult.

An alternative to gesture tracking can be translated more in a one-to-one fashion since a camera provides a 2D output. Camera-based example is described “Hand Region Extraction and Gesture Recognition using entropy analysis” [5];

however, this approach traditionally uses computer vision techniques, which also leverages pattern recognition techniques, which requires training data to be collected but this leads to limited feature extraction on the controlled data.

One of the most related gesture devices is discontinued but was made a similar use case as the Pison device is the Myo armband by Thalmic labs. This product uses an IMU and multiple electromyography sensors to classify a set of provided gestures.

Figure 1. Myo Gestures

These gestures require the user to perform a repeatable iteration of them. According to “Evaluation of the Myo armband for the classification of hand motions” [6], the author evaluated 8 subjects with eight gestures for classification. The average error for the six-channel Myo armband was only 10%; however, the standard deviation was 8.05%. The author mentioned that a pattern recognition technique was used. Subjects 4 and 7 contributed a high error rate, approximately 20%, due to inconsistency between their movements.

2.2 Analysis of Literature

One of the starting points of this project is to evaluate a similar gesture device in the commercial space. The Myo armband from Thalmic Labs is one great example. However, the approaches are much different. Many papers suggest high accuracy using the device. This high accuracy is evaluating between the gesture difference but does not demonstrate real-world classification error where certain gestures will introduce errors since these movements are not part of the classification matrix.

	Actual Gesture						%ACCURACY %ERROR
		FIST	OPEN	WAVE IN	WAVE OUT	PINCH	
Prediction	FIST	260 17.33%	17 1.13%	34 2.27%	1 0.07%	3 0.2%	82.5% 17.5%
	OPEN	13 0.87%	261 17.4%	12 0.08%	6 0.4%	1 0.07%	89.1% 10.9%
	WAVE IN	1 0.07%	0 0%	211 14.07%	2 0.13%	0 0%	98.6% 1.4%
	WAVE OUT	0 0%	0 0%	0 0%	255 17%	0 0%	100% 0%
	PINCH	4 0.27%	1 0.07%	1 0.07%	25 1.67%	259 17.27%	89.3% 10.7%
	NO GESTURE	22 1.47%	21 1.4%	42 2.8%	11 0.73%	37 2.47%	0% 100%
	%ACCURACY %ERROR	86.67% 13.33%	87% 13%	70.33% 29.67%	85% 15%	86.3% 13.7%	83.07% 16.93%

Figure 2. Classification of Myo gesture from “Hand Gesture Recognition Using Machine Learning and the Myo armband”.

The chart in Figure 2 is from “Hand Gesture Recognition Using Machine Learning and the Myo armband” [7] which demonstrated the data collected with the Myo armband. Differently from Figure 2, there is a “no gesture” classes. The “no gesture” classes in this paper is the subject simply putting his/her arm down at a rest position. This “no gesture” class still does not demonstrate the day to day activity in this classification matrix.

Bio-signal acquisition and filtering are some of the biggest contributors to the accuracy of the gesture classification. For muscle activation, especially since the EMG frequency is between 0-500 Hz, the most dominant frequency lies between 0-150 Hz. [8] Many of the papers stated above mentioned that the Myo armband limited the output to <100 Hz over a wireless link. Hence, a significant amount of information is lost due to wireless transmission. The Pison device is different: the wireless output is between 300Hz – 600Hz, but the wired output can be up to 2kHz since the sensor is designed for action potentials. For this project, we will only focus on the EMG spectrum of the signal, and the Pison device contains the information required to do classification since the targeted frequency for this paper would be 0-150Hz, and following the Nyquist sampling requirement, 300Hz is sufficient. Also the gains in the Pison device is modified for EMG purpose.

There are many artifacts in the physiological data. Since the sensor is acquiring small voltages at the surface of the skin, it is prone to electrical and magnetic interference (EMI) and motion artifacts. All these artifacts can be environmental, and if the training data contains these artifacts, this will confound the classification. As shown in Figures 3-5 below, these EMI can penetrate the electrodes and cause a different effect based on the distance from the electrode.

Figure 3. Surface electromyographic recording showing the effects of a cell phone located 1 cm (0.4 inches) away from the electrodes.[9]

Figure 4. Surface electromyographic recording showing the effects of a cell phone located 30 cm (11.8 inches) away from the electrodes.[9]

Figure 3 and 4 have the y-axis as voltage in millivolt and x-axis as time in seconds.

Figure 5. Surface electromyographic recording showing the effects of a cell phone ringing at different distances.[9]

Figure 5 is from “Keep Cell Phones and PDAs Away From EMG Sensors and the Human Body to Prevent Electromagnetic Interference Artifacts and Cancer” [9], where the author demonstrated that there is a linear correlation between EMI and the signal acquired from EMG sensors. These artifacts have saturated the activation of the sensor with noise. Therefore these artifacts need to be properly preprocessed to provide better classification.

List of References

- [3] Forbes, Timothy, "Mouse HCI Through Combined EMG and IMU" (2013). *Open Access Master's Theses*. Paper 43.
<https://digitalcommons.uri.edu/theses/43>
- [4] E. L. Sean Chen. Cornell University. "Mister gloves - a wireless usb gesture input system." 2010. [Online]. Available: <https://courses.cit.cornell.edu/>
- [5] Shin, Jae-Ho, et al. "Hand region extraction and gesture recognition using entropy analysis." *IJCSNS International Journal of Computer Science and Network Security* 6.2A (2006): 216-222.
- [6] Mendez, Irene, et al. "Evaluation of the Myo armband for the classification of hand motions." 2017 International Conference on Rehabilitation Robotics (ICORR). IEEE, 2017.
- [7] Benalcázar, Marco E., et al. "Hand gesture recognition using machine learning and the Myo armband." 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017.
- [8] Shaw, Laxmi, and Sangeeta Bagha. "Online EMG signal analysis for diagnosis of neuromuscular diseases by using PCA and PNN." *International Journal of Engineering Science and Technology* 4.10 (2012): 4453-4459.
- [9] Lin, I-Mei, and Erik Peper. "Keep cell phones and PDAs away from EMG sensors and the human body to prevent electromagnetic interference artifacts and cancer." *Biofeedback* 37.3 (2009): 114-116.

CHAPTER 3

METHODOLOGY

3.1 Bio-Potential and IMU Sensors

For any system, first we need to identify the input. For this project, the Pison Device is used to acquire the bio-potential data as well as the IMU data. The data can be streamed both wirelessly or via a USB virtual serial port. The bio-potential sensor has a set of six electrodes: two for each of the two data channels and one channel for reference and one for case grounding. The IMU used is a nine-axis (gyro, accelerometer, and magnetometer) device. Data from both of these sensors will be controlled by a microcontroller with a Bluetooth module transmitting the data to an Android phone to do the action classification. The IMU sensor that is embedded in the Pison device is a system in a package with a microcontroller in the IC to provide filtered nine-axis data to quaternion representation of the device; quaternion is a unit system that provides a convenient mathematical notation for representing orientations and rotations of objects in three dimensions.[10]

3.2 IMU Processing

The Pison device IMU coordinate system is a right-handed coordinate system with X pointing East, Y pointing North, and Z pointing up (ENU). When the device is on the right wrist, this corresponds to +X pointing towards the 5th finger (ulnar side), +Y pointing along the forearm out through the fingers, and +Z up. Pitch is defined as positive in a counter-clockwise rotation around the X-axis. Positive yaw is counter-

clockwise about the Z-axis, and Roll is positive in a clockwise direction about the Y-axis (the forearm).

Figure 6. Coordinate axes defined relative to subject right hand

The Pison device sends the data to the Android phone via byte code listed in “Appendix A. Transmission specification of the device data”. The Kotlin version of the code uses the Libgdx quaternion library. Due to the differences between the standard Libgdx coordinate system, and the IMU generated quaternion coordinate system, the signs of the angles sometimes need to be changed to match our model.

3.2.1 Calculating Pitch

The pitch angle of the device should always be positive when the user is moving the arm up, and should be negative when the user is moving the arm down. Since we want the pitch angle to be the elevation angle in the reference frame of the user, we cannot use the strict Euler angle.

Figure 7. Projected Pitch

When the user is pointing north and raises the arm, the IMU registers a counterclockwise rotation, and the Euler pitch angle is positive. When the user is pointing south, that same movement is a clockwise rotation in the inertial frame and will return a negative angle for the pitch.

Additionally, as the user moves east or west, the pitch and roll angles will be swapped. If a user is facing east, and raises the hand, inertially, that is a rolling movement (since it is now about the north-south axis, not the east-west axis).

In order to get the pitch angle in the reference frame of the user regardless of which direction the user is facing (or how the arm has moved), what we really want is to find out the elevation angle of the transformed y-axis of the device (which corresponds to the forearm). To do this, we take a unit vector along the y-axis and transform it with the quaternion from the IMU. This gives us a 3D vector pointing along the current

direction of the forearm of the user, and we can calculate the elevation of this vector using trigonometric functions. The pitch angle range is -90 to + 90 degrees.

3.2.2 Calculating Yaw

In yaw, the Euler angle works fine when the pitch angle is 0, but as the user's arm moves up, the yaw angles and the pitch angle combine the same way they do it in the Pitch case. The solution is the same: transform the unit vector along the forearm axis and calculate the angle the transformed vector makes with the YZ plane. Yaw angle range is -180 to +180 degrees. North is 0, west (CCW) is positive.

Figure 8. Projected Yaw

3.2.3 Calculating Roll

Roll is defined as the clockwise rotation of the device around the y-axis (the roll axis). This can be calculated by taking the dot product of the unit vector V_y with the device quaternion, Q , using the formula:

Figure 9. Projected Roll

$$\theta_{\text{roll}} = \text{acos}[2(V_y \cdot Q) - 1]$$

3.3 Bio-Potential Processing

Before any classification can occur, the bio-potential must be processed to reduce possible bad data feeding into the classifier. In order to keep the data consistent, the wireless data from the bluetooth low energy (BLE) module is only capable of transmitting at 600 Hz. This limitation will require us to linearly interpolate samples to keep the sample rate consistent. There are three noticeable artifacts that can interfere with bio-potential sensors such as EMI, motion, and aliasing. Therefore, filters must be implemented to prevent these artifacts.

3.3.1 Anti-Aliasing Filter

According to the Nyquist sampling theorem, the sampling rate should be greater than twice the maximum frequency component of the signal of interest. In other words, the maximum frequency of the input signal should be less than half of the sampling

rate. To ensure this is the case, the anti-aliasing filter, typically a low-pass filter, is used to attenuate frequencies above the Nyquist frequency and prevent aliasing components from being sampled.

Figure 10. Comparison of ideal vs practical anti-aliasing filter

The Pison device has a sampling rate of 600 Hz and is the key limitation of the allowable input frequency of interest. Therefore the calculation has to be done backwards. Let b in Eq. (1) be the width of the transition band of the anti-aliasing filter, the maximum signal bandwidth that can be achieved is

$$f_1 \leq 600/2 - b \quad (1)$$

If we are using a simple first order RC filter, the roll off is -20 dB/decade. We want the gain to be sufficiently small at $f_2 = 300\text{Hz}$. We sufficiently small as $< 10\%$ of the pass band gain, or -10 dB, and then using Eq. (1) $f_2 = 5f_1$ and $b = f_2 - f_1 = 4f_1$.

The above analysis shows that to ensure good anti-aliasing given the current sampling rate and choice of anti-aliasing filter architecture, the cut-off frequency of the RC filter has to be $\leq 60\text{ Hz}$. Figure 11 further illustrates the effect of the cut-off frequency of a first order RC filter cut-off on the gain at 300 Hz. It is evident that the

original design of anti-aliasing filter with a cut-off of 250 Hz is far from ideal, i.e., it provides little attenuation of higher frequencies. To achieve good attenuation, we have to lower the cut-off frequency significantly, using:

$$f_1 \leq 300 - 4f_1 \Rightarrow f_1 \leq 60 \text{ Hz} \quad (2)$$

However, lowering the cut-off frequency, say to 60 Hz, will attenuate the dominant frequency of the EMG signal, which is in the 50 - 150 Hz range. Therefore to retain greater EMG signal power while preserving anti-aliasing properties at the given sampling rate, we need an anti-aliasing filter with a cut-off frequency above 150 Hz and rapid roll-off to achieve high attenuation by 300 Hz. At a cut-off of 150 Hz, the attenuation achieved with a first-order RC filter at 300 Hz is only -3.9 dB or around 40%.

Figure 11. Effect of choice of cut-off frequency on gain at 300 Hz for a first order RC anti-aliasing filter. The amount of attenuation provided by the original anti-aliasing filter with a 250 Hz cut off is shown with a blue triangle. The cut off which gives 0.1 attenuation is shown with a red circle. The maximum attenuation that can be achieved with a cutoff of 150 or above is shown with a yellow square.

3.3.1 EMI Filter

In order to create an effective EMI filter we first need to collect an activation signal with EMI and without EMI to compare the effect of the artifact on a user performing an activation, so the following data is collected and plotted:

Figure 12. EMI vs no EMI of Index Finger Clicking EMG Data

As shown in Figure 12, the EMI has a significant effect on the bio-potential data.

There is clearly no visible signal in the right panel.

A spectral analysis of this data will reveal the EMI frequencies that interfere with the EMI Signal.

Figure 13. Frequency Spectrum of EMI vs no EMI of Index Finger Clicking

As shown in Figure 13, the EMI contains harmonics at 59 Hz, 118 Hz, 178 Hz, 237 Hz, and 297 Hz. Therefore, a multi-stage notch filter is designed to remove these

frequencies. Below is a simulated 10th order Chebyshev Type II Bandstop filter which is also shown on a filter analyzer (MATLAB).

Figure 14. Bandstop Filtering Effect on EMI noise

The top plot in both left and right panels has the x-axis for the number of samples of EMG data and y-axis for digitized voltage data in bits. The bottom of both the left and right panel has the x-axis as a normalized frequency measured in radians per sample, and the y-axis is the power spectrum measured in dB.

The results proven effective in Figure 14. The right panel shows the buried 11 index activation data after filtering; the left panel shows the buried EMG data.

3.3.1 Motion Artifact Filter

In order to reduce the motion artifact, which can be generated by the user's intended moving or movement vibration created by a user in a moving vehicle, a 3rd order 20 Hz high pass filter is used in conjunction with the EMI notch filter. In order to evaluate the effect of the combination filter, the signal is tested under motion artifact conditions. The true positive rate, false-positive rate, and signal-to-noise ratio are calculated to evaluate with a box classifier which is a heuristic classifier with a state machine approach (See section 3.4 for classifier design).

Signal-Noise-Ratios (SNR) calculated by:

$$\text{SNR} = \log \sqrt{\frac{\text{RMS}_{\text{activation}}}{\text{RMS}_{\text{Resting Data}}}}$$

	True Positive Rate	False Positive Rate	Ch0- SNR	Ch1- SNR
Raw Data	0.6568	0.3961	0.1912	1.5053
Filtered Data	0.7682	0.2164	7.0119	5.8488

Table 1. Raw classification vs Filtered Classification Data

As shown in Table 1, the signal-to-noise ratio is increased significantly when the myoelectric signals are filtered by a high pass filter and notch filter. At the same time, the filtration improves the result of signal classification to achieve a higher true positive rate, a lower false postive rate and a significant higher signal to noise ratio. The receiver operating characteristic curve, i.e., ROC curve, is created by plotting the true positive rate against the false-positive rate at various threshold settings.

Figure 15. ROC Curve (the left graph is raw data, the right graph is filtered data).

The ROC curve indicates that the classification can achieve a higher true positive rate with lower false positive rates when signals are filtered.

3.4 Classifier Design

The targeted action to classify is the index finger extension action. However, due to the placement of the Pison device, the index finger extension and other actions can produce a similar signal. Therefore, designing a classifier to focus on the index extension is slightly difficult; therefore, different classifiers are evaluated and designed to overcome this problem. After evaluation, the actions are shown in Table 2.

1	Rest
2	Thumb Extension
3	Index Extension
4	Pinky Extension
5	Hand Open
6	Wrist Extension
Table 2. List of Actions to Classify

Figure 16. Example signal collected with Pison device

The labeled gesture matches the actions shown in Table 2.

The performance metric is calculated below:

Metrics	Formula
True Positive Rate (TPR)	True Positive / (True Postive + False Negative) * 100
False Positive Rate (FPR)	False Positive / (False Postive + True Negative) * 100

Table 3. TPR and FPR calculation formula

Next, the LDA Classifier in this research and Box Classifier are compared.

3.4.1 LDA Classifier

Linear Discriminant Analysis (LDA) is a machine learning technique, “which easily handles the case where the within-class frequencies are unequal and their performances has been examined on randomly generated test data” [11] which is ideal for gesture classification problems. The data were segmented into 100 ms windows (with 25 ms window increment) and six features were extracted from each

window. The following features were chosen from common EMG features selection process.

1. Mean Square Root (MSR) from channel 0 and channel 1.
2. Maximum Fractal Length (MFL) from channel 0 and channel 1.
3. Time-Domain Power Spectral Moments (TDPSD) from channel 0.
4. Mean Absolute Values of the Second Difference (MAVSD) from channel 0.

Figure 17. Example of features from test data. The boxed numbers denote the action number (refer to the Action Set in Table 2). Note that different level of values is generated for different actions.

Each decision output that was classified as “active” was verified in a post-processing stage, where each feature value of the decision window was checked to see if they were within the sensitivity boundary or not. The activation was triggered only if all features were inside the boundary.

Sensitivity value determines the size of the boundary; thus, it directly impacts the performance. Figure 18 shows the example of the relationship between the sensitivity values and the performance metrics (TPR & FPR). In this analysis, sensitivity values were manually chosen, aiming to achieve as high TPR as possible while maintaining relatively lower FPR to maximize TPR to FPR ratio. For the use case scenario, the user could have a sliding bar for setting the sensitivity.

Figure 18. Example of sensitivity vs TPR and FPR for Pison device. Sensitivity value of 0.4 would be chosen, which would yield 100% TPR and 34% FPR.

3.4.2 Box Classifier

The box classifier is a custom heuristic algorithm that allows users to select an area of interest in a real-time FFT plot to allow for real-time sensitivity adjustment. The idea of the box classifier is to threshold both amplitude and frequency by creating a box in an FFT plot. However, there are two channels of bio-potential being used for

box classifier; therefore, potential difference between the two channels for the action of focus must be identified.

3.4.2.1 Channel 0 vs Channel 1 Analysis

The actions that are almost identical in the frequency energy gains are action 3 (index extension) and action 2 (thumb extension). Therefore 50 activations per action are collected to compare and calculate the difference. We applied the 0-100 Hz threshold and tried to explore the relation between the signal of Ch0 and Ch1. Average energy for both channels and the ratio of Ch0 and Ch1 are plotted in Figure 19.

Figure 19. Average Energy difference between two channels

As shown in Figure 19, there is a difference in the average FFT energy between Channel 0 and Channel 1. Channel 1 has a much higher average energy than Channel 0 with thumb extension, and Channel 0 has a higher average energy than Channel 1 with index extension.

3.4.2.2 Box Classifier System Diagram

The box classifier is a state machine approach to classify signals in real-time. Shown in Figure 20.

Figure 20. State machine design of the Box Classifier

3.4.2.3 Box Classifier UI Design

To create a threshold in both amplitude and frequency, a user interface is needed to provide for customization. Figure 21 is the example of the classification configuration window.

Figure 21. Box Classifier UI

Figure 21 is a plot with the x-axis showing frequency in Hz and the y-axis is amplitude in Bits. The two vertical sliders control the frequency threshold, and the horizontal line controls the energy threshold. This allows for adjustment of the sensitivity; the user can identify his or her movement and move the box according to tune to their liking.

3.5 Gesture User Experience

After the sensor data has been processed, it is useful to be interpreted as a gesture; therefore, combining the IMU and bio-potential classification would create a gesture experience. The idea of the gesture user experience would be demonstrated via this finite state machine approach. There are three main components to the gesture

user experience: action, feedback, and transition. The gesture experience will be fluent when chaining these blocks together to create a combination gesture as seen in Figure 22

Figure 22. Gesture State Machine

In order to create a gesture, it should contain action, feedback, and transition, and multiples of these gestures can create a combination gesture.

3.5.1 Action

In order to demonstrate the finite state machine approach, only the robust index extension action will be used as a binary action. The granular action provided from the yaw, pitch and roll would be an analog action. In this scenario, the user is provided with a binary click and three degrees of freedom with analog actions.

3.5.1.1 Binary Action

The binary action is rather straight forward. It produces a 1 or 0. The only customization of this would be the sensitivity, which can be adjusted via the energy slider. Example configuration shown in Table 4.

Action	Sensitivity
Index Extension	0-100%

Table 4. Configuration for Action

3.5.1.2 Analog Action

For the state machine to work, the analog actions must exceed the threshold to provide a binary output for the state to traverse forward. However, for each analog axis (yaw, pitch, and roll), there are two ways to the threshold. One would be the absolute angle, and another would be the change of angle. The change of angle is calculated with the angle change during a period of time. The absolute angle is threshold for whether the Pison device is above a certain angle. The change of angle can be classified as speed, and there are 2 parameters to adjust, such as duration in milliseconds, or the angle change in degrees. See Table 5 for more information.

Action	Mode	Parameter	
Yaw	Absolute	Check between range 0-360	
	Differential	>5 degrees	1-1000 milliseconds
Pitch	Absolute	Check between range 0-360	
	Differential	>5 degrees	1-1000 milliseconds
Roll	Absolute	Check between range 0-360	
	Differential	>5 degrees	1-1000 milliseconds

Table 5. Configuration for IMU

3.5.2 Feedback

There are four options of feedback that can be selected by the user as in Table 6.

Feedback	
None	-
Haptic	Sends a vibration from the Pison device
Visual	Software user interface changes color
Audio	Sound being played from the phone

Table 6. Types of feedback

The feedback also should have customization to allow users to identify which gesture the user is currently performing as seen in Table 7.

Feedback		
Haptic	Duration	0 - 200ms
Visual	Color	#000000 - #FFFFFF
Audio	Tone	1 - 5

Table 7. Feedback Configuration

3.5.3 Transition

Transition is rather tricky. There are two possible ways to configure the state machine two gestures can be combined in a cascade or parallel approach. In the cascade approach, one gesture must be completed before the second gesture can start. In the parallel approach, two gestures can start at the same time; however, it must remain true for the finite state machine to advance to the next stage. This project used only the cascade transition; however, future work can expand on this.

3.6 Output

Finally, when the gesture state advanced to the end, and a user has finished the sequence of the action, the device must be able to output something meaningful. In the software the finished sequence of gesture can be mapped to a web request or key press . This mapping will allow the user to control many internet of things devices since the gesture now is translated to an understandable request.

CHAPTER 4

CONCLUSION

To create the state machine approach gesture classifier, a back end software component is provided with processed data. As shown in section 3.4.1 the true positive rate is close to 90%, and the false positive rate is around 20%. This is quite acceptable for tested user to control internet of things. For this initial approach since there can be a usability trade-off as we reduce the false positive rate. The positive rate would be reduced as well. To make this useful for the user, A 90% true positive rate and 20% false positive rate is acceptable for the user. For this project, a visual adjustable classifier has been made using the box classifier.

Figure 24. Box Classifier Android Interface

Figure 24 shows a screenshot of the box classifier configuration page on an Android phone, which has four possible sliders. The two vertical sliders control the frequency cut off, and the two horizontal sliders control the energy threshold in a Schmidt trigger fashion. Essentially this allows the user to have two domains of control.

To demonstrate the usability of the state machine, a use case of gesture needs to be created. One of the demos which integrated the state machine approach of gesture classification is drone control. The gesture demonstrated certain gesture short cuts such as wipe up/down/left/right. On top of that we can map the yaw/pitch/roll of the Pison device to the drones yaw/pitch/roll. For this scenerio, instead of mapping the gesture to a key, it is mapped to an action of the drone as listed in Table 9.

Gesture	Drone Action
Index Lift	Recenter Zero
Swipe Right	Unlock
Swipe Left	Lock
Swipe Up	Lift Off
Swipe Down	Land

Table 9. Mapping Combination Gesture to Drone Action

In order to develop the rest of Android application, the drone ground control station is implemented using the Parrot Drone SDK[12].

Main Screen	Status in Main Screen

Drone Connection Config	Drone Management

Table 10. Setting Screens

Main Screen:

- Showing status and response of the Pison device. The middle circle will respond to when the pointing gesture has been detected
- Status in the main menu shows all the computed IMU data

Drone Connection Configuration

- This screen shows the past and active paired drones that are on the wireless network

Drone Management

- Configuration of the drone settings such as:
- Maximum Speed for yaw/pitch/roll
- Return to Home Location

Table 11. Drone HUD

The Drone Heads Up Display (HUD) interface shows the camera feed from the drone along with some touch screen input such as joysticks in the case of emergency control. However, the joystick control is mapped to the yaw/pitch/roll of the Pison device to allow control of the drone.

By incorporating the state machine approach of gesture classification in a usable and relatable manner, it is rather successful.

CHAPTER 5

FUTURE WORK

There are some pain points regarding maintaining the scalability of the state machine approach. To create more gestures, the developer requires to develop another custom classifier to work.

To demonstrate the usability of the state machine approach, a customizable interface must be developed to avoid manual coding of the gesture classifiers. A visual interface is required that allows a user to select the action, feedback, and transition, eventually choosing the output. This will enable the user to quickly prototype use cases instead of copying and pasting the same block of gesture classification code to a different type of platform.

Additionally, the parallel transition also needs to be developed. This type of transition will allow the user to combine two actions at the same time to create more gesture possibilities. For example, instead of the user extending the index finger and letting go and then rolling the wrist, the user will be able to hold the index finger extension and roll the wrist at the same time.

APPENDICES

Appendix A. Transmission specification of device data

Byte index	Data
0	First byte of microsecond timestamp
1	Second byte of microsecond timestamp
2	Third byte of microsecond timestamp
3	Fourth byte of microsecond timestamp
4	Fifth byte of microsecond timestamp
5	Last byte of microsecond timestamp
6	Upper byte of quat_w
7	Lower byte of quat_w
8	Upper byte of quat_x
9	Lower byte of quat_x
10	Upper byte of quat_y
11	Lower byte of quat_y
12	Upper byte of quat_z
13	Lower byte of quat_z
14	Upper byte of adc raw data (channel 0)
15	Middle byte of adc raw data (channel 0)
16	Lower byte of adc raw data (channel 0)
17	Upper byte of adc raw data (channel 1)
18	Middle byte of adc raw data (channel 1)

19	Lower byte of adc raw data (channel 1)
20	Upper byte of filtered data (channel 0)
21	Upper middle byte of filtered data (channel 0)
22	Lower middle byte of filtered data (channel 0)
23	Lower byte of filtered data (channel 0)
24	Upper byte of filtered data (channel 1)
25	Upper middle byte of filtered data (channel 1)
26	Lower middle byte of filtered data (channel 1)
27	Lower byte of filtered data (channel 1)
28	Upper byte of accelerometer x
29	Lower byte of accelerometer x
30	Upper byte of accelerometer y
31	Lower byte of accelerometer y
32	Upper byte of accelerometer z
33	Lower byte of accelerometer z
34	Upper byte of gyro x
35	Lower byte of gyro x
36	Upper byte of gyro y
37	Lower byte of gyro y
38	Upper byte of gyro z
39	Lower byte of gyro z

BIBLIOGRAPHY

- [1] Thalmic Labs. "Myo gesture control armband - wearable technology by thalmic labs." 2015. [Online]. Retrieved May 20, 2019 from <https://www.myo.com/>
- [2] Cheng, H., Yang, L., & Liu, Z. (2016). Survey on 3D Hand Gesture Recognition. *IEEE Trans. Circuits Syst. Video Techn.*, 26(9), 1659-1673.
- [3] Forbes, Timothy, "Mouse HCI Through Combined EMG and IMU" (2013). *Open Access Master's Theses*. Paper 43. Retrieved August 22, 2019 from <https://digitalcommons.uri.edu/theses/43>
- [4] E. L. Sean Chen. Cornell University. "Mister gloves - a wireless usb gesture input system." 2010. [Online]. Retrieved September 2, 2019 from <https://courses.cit.cornell.edu/>
- [5] Shin, Jae-Ho, et al. "Hand region extraction and gesture recognition using entropy analysis." *IJCSNS International Journal of Computer Science and Network Security* 6.2A (2006): 216-222.
- [6] Mendez, Irene, et al. "Evaluation of the Myo armband for the classification of hand motions." 2017 International Conference on Rehabilitation Robotics (ICORR). IEEE, 2017.
- [7] Benalcázar, Marco E., et al. "Hand gesture recognition using machine learning and the Myo armband." 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017.

[8] Shaw, Laxmi, and Sangeeta Bagha. "Online EMG signal analysis for diagnosis of neuromuscular diseases by using PCA and PNN." *International Journal of Engineering Science and Technology* 4.10 (2012): 4453-4459.

[9] Lin, I-Mei, and Erik Peper. "Keep cell phones and PDAs away from EMG sensors and the human body to prevent electromagnetic interference artifacts and cancer." *Biofeedback* 37.3 (2009): 114-116.

[10] Quaternion

<https://en.wikipedia.org/wiki/Quaternion>

[11] Balakrishnama, Suresh, and Aravind Ganapathiraju. "Linear discriminant analysis-a brief tutorial." *Institute for Signal and information Processing* 18 (1998): 1-8.

[12] Parrot Ground Sdk

<https://developer.parrot.com/>