

Spring 2019

Assessment of Hand Gestures Using Wearable Sensors and Fuzzy Logic

Angel Cardenas

Ryan Messersmith

Will Newcomb

Follow this and additional works at: https://scholarcommons.scu.edu/bioe_senior

 Part of the [Biomedical Engineering and Bioengineering Commons](#)

SANTA CLARA UNIVERSITY

Department of Bioengineering

**I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY**

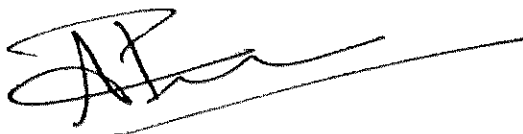
Angel Cardenas, Ryan Messersmith, Will Newcomb

ENTITLED

**ASSESSMENT OF HAND GESTURES USING WEARABLE
SENSORS AND FUZZY LOGIC**

**BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF**

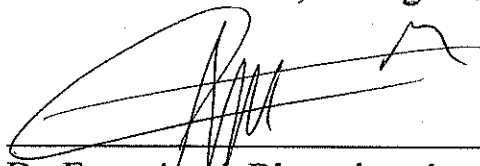
**BACHELOR OF SCIENCE
IN
BIOENGINEERING**



Dr. Prashanth Asuri, Bioengineering Advisor

6-13-19

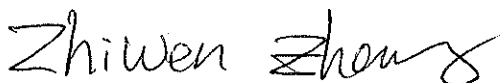
Date



Dr. Emre Araci, Bioengineering Advisor

6/13/19

Date



Dr. Zhiwen Zhang, Bioengineering Department Chair

6/13/19

Date

**ASSESSMENT OF HAND GESTURES USING WEARABLE SENSORS
AND FUZZY LOGIC**

By

Angel Cardenas, Ryan Messersmith, Will Newcomb
Equal participants

SENIOR DESIGN PROJECT REPORT

Submitted to
The Department of Bioengineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Bioengineering

Santa Clara, California

Spring 2019

Assessment of Hand Gestures using Wearable Sensors and Fuzzy Logic

Angel Cardenas, Ryan Messersmith, Will Newcomb

Department of Bioengineering
Santa Clara University
2019

Abstract

Hand dexterity and motor control are critical in our everyday lives because a significant portion of the daily motions we perform are with our hands and require some degree of repetition and skill. Therefore, development of technologies for hand and extremity rehabilitation is a significant area of research that will directly help patients recovering from hand debilities sustained from causes ranging from stroke and Parkinson's disease to trauma and common injuries. Cyclic activity recognition and assessment is appropriate for hand and extremity rehabilitation because a majority of our essential motions are cyclic in their nature. For a patient on the road to regaining functional independence with daily skills, the improvement in cyclic motions constitutes an important and quantifiable rehabilitation goal. However, challenges exist with hand rehabilitation sensor technologies preventing acquisition of long-term, continuous, accurate and actionable motion data. These challenges include complicated and uncomfortable system assemblies, and a lack of integration with consumer electronics for easy readout. In our research, we have developed a glove based system where the inertial measurement unit (IMU) sensors are used synergistically with the flexible sensors to minimize the number of IMU sensors. The classification capability of our system is improved by utilizing a fuzzy logic data analysis algorithm. We tested a total of 25 different subjects using a glove-based apparatus to gather data on two-dimensional motions with one accelerometer and three-dimensional motions with one accelerometer and two flexible sensors. Our research provides an approach that has the potential to utilize both activity recognition and activity assessment using simple sensor systems to help patients recover and improve their overall quality of life.

Acknowledgements

We would like to thank our advisors, Dr. Emre Araci and Dr. Prashanth Asuri for their continued guidance and support throughout the past year. With your assistance, insight, and expertise, we were able to present a project we were proud of at our Senior Design Conference. We have all grown so much as students and researchers, and want to thank you for your patience throughout the process. We greatly appreciate the time and effort you put into helping us with our project and making sure we were successful. Thank you for all you have done for us; we will be forever grateful.

We would also like to thank all of our friends and family for their continued support in our academic endeavors, and for always being available to assist with anything we asked of them.

Finally, we would like to thank the Santa Clara University School of Engineering for providing us with the funds necessary to complete our project. Without your generosity and financial support, none of this would have been possible.

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vii
List of Tables	viii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Literature Review	2
1.2.1 Existing sensor technologies within the field of extremity rehabilitation	2
1.2.2 Motor rehabilitation technologies are becoming case specific	3
1.2.3 Fuzzy Logic systems	4
1.2.4 Summary of challenges with hand rehabilitation sensor technologies	5
1.3 Project Goals and Objectives	6
1.4 Project Budget	8
1.5 Project Timeline	9
Chapter 2: Design Process	10
2.1 Design Goals	10
2.2 Consumer Needs and System Requirements	10
2.3 Preliminary Design	11
2.4 Final Design	12
Chapter 3: Materials and Methods	13
3.1 Hardware Components	13
3.1.1 Accelerometers	13
3.1.1.1 Accelerometer Calibration	14
3.1.2 Flex Sensors	14
3.1.2.1 Flex Sensor Characterization	14
3.1.3 Wearable Glove Assembly	16
3.1.4 Data Acquisition Board	16
3.2 Software Components	17
3.2.1 Processing and Arduino	17
3.2.2 MATLAB	17
3.2.2.1 Data Optimization	18
3.2.2.2 Fourier Transforms	18
3.2.3 Fuzzy Logic Designer	18
3.2.3.1 Determining Membership Functions	20
3.2.3.2 Establishing Rules	21
3.3 Testing Procedures	22

Chapter 4: Results	24
4.1 Hypothesis	24
4.2 2-D Motion Testing	24
4.2.1 Acceleration Data	25
4.2.2 Data Analysis using Fourier Transforms	25
4.2.2.1 Significance of Axial Ratios	28
4.3 3-D Motion Testing	29
4.3.1 Acceleration Data	29
4.3.1.1 Data Analysis using Fourier Transforms – Acceleration Data	31
4.3.2 Flex Sensor Data	33
4.3.2.1 Data Analysis using Fourier Transforms – Flex Sensor Data	35
4.3.2.2 Significance of Fourier Spectrum Magnitude	37
4.4 Motion Classification using Fuzzy Logic	38
4.5 Summary of Results	39
Chapter 5: Discussion	40
5.1 Discussion of Results	40
5.2 Future Work	40
5.2.1 Continued Testing on a Wider Body of Subjects	41
5.2.2 Incorporation of Machine Learning	41
5.2.3 Testing using Embedded Sensors	41
5.2.4 Inclusion of a Wrist-bound Accelerometer	43
5.2.5 Integration of Consumer Electronics and Smart Devices	43
5.2.6 Potential to Test for Tremor Detection	44
5.2.7 An At-home Rehabilitation Program	44
Chapter 6: Engineering Standards and Constraints	46
6.1 Ethical	46
6.2 Science, Technology, and Society	46
6.3 Civic Engagement	46
6.4 Economic	47
6.5 Health and Safety	47
6.6 Manufacturability	48
6.7 Usability	48
6.8 Sustainability	49
6.9 Environmental Impact	49
Appendices	50
Appendix 1: Arduino Code	50
Calibration Code	50
Data Acquisition Code	51
Appendix 2: Processing Code	52
Processing Code for Data Acquisition	52

Appendix 3: MATLAB Code	53
Data Optimization Code	53
Motion Classification Code	53
Appendix 4: Fuzzy Logic Designer Code	57
System Setup	57
Defining Membership Functions for Inputs and Outputs	57
Establishing Rules for the Fuzzy Logic System	59
Appendix 5: Circuit Diagrams	60
Accelerometer Circuitry Schematic	60
Flex Sensor Circuitry Schematic	60
Appendix 6: Component Data Sheets	61
Accelerometer Data Sheets	61
Flex Sensor Data Sheets	64
Appendix 7: Protocol for Ecoflex Glove	66
Materials	66
3D Hand	66
Flex Sensor	66
Ecoflex Glove	67
Bibliography	68

List of Figures

Figure 1: Two and Three-Dimensional Motions	7
Figure 2: Design Schematic of the Preliminary Testing Apparatus	11
Figure 3: Design Schematic of the Final Testing Apparatus	12
Figure 4: Flex Sensor Characterization	15
Figure 5: Test Setup for Flex Sensor Characterization	15
Figure 6: Hardware Components in Wearable Glove Assembly	16
Figure 7: Membership Function Plots	20
Figure 8: Creating Rules for the Fuzzy Logic Designer	21
Figure 9: Two-Dimensional Motions	24
Figure 10: Raw Acceleration Data of a Circular Motion in MATLAB	25
Figure 11: Fourier Spectrum of a Circular Motion in MATLAB	26
Figure 12: Fourier Spectrum of a Waving Motion in MATLAB	27
Figure 13: Fourier Spectrum of a Figure 8 Motion in MATLAB	27
Figure 14: Raw Acceleration Data of a Clenching Motion in MATLAB	30
Figure 15: Raw Acceleration Data of a Eating Motion in MATLAB	30
Figure 16: Raw Acceleration Data of a Pouring Motion in MATLAB	31
Figure 17: Fourier Spectrum of Acceleration for a Clenching Motion in MATLAB	31
Figure 18: Fourier Spectrum of Acceleration for a Eating Motion in MATLAB	32
Figure 19: Fourier Spectrum of Acceleration for a Pouring Motion in MATLAB	32
Figure 20: Raw Flex Sensor Data of a Clenching Motion in MATLAB	34
Figure 21: Raw Flex Sensor Data of a Eating Motion in MATLAB	34
Figure 22: Raw Flex Sensor Data of a Pouring Motion in MATLAB	35
Figure 23: Fourier Spectrum of Flex Sensor Data of a Clenching Motion in MATLAB	36
Figure 24: Fourier Spectrum of Flex Sensor Data of a Eating Motion in MATLAB	36
Figure 25: Fourier Spectrum of Flex Sensor Data of a Pouring Motion in MATLAB	37
Figure 26: Example of Motion Outputs Displayed to the User	39
Figure 27: Ecoflex Embedded Sensors Glove	42
Figure 28: Example of a Potential Application Interface for Future Research	44

List of Tables

Table 1: Project Budget	8
Table 2: Project Timeline	9
Table 3: Comparing Frequency and Magnitude Ratios of Two-Dimensional Motions	28
Table 4: Comparing the Magnitude Ranges of Fourier Spectrums of Flex Sensor Data	38

Chapter 1: Introduction

1.1 Background:

Hand dexterity is a vital part of our everyday lives. From eating food with a utensil, to swinging a baseball bat, to drinking water from a reusable bottle, we humans rely on manually controlling our hands for learning essential, occupational, and leisure skills that all help improve the quality of our lives. However, hand dexterity is not something to take for granted, as hand injuries and debilities are significantly common and require professional rehabilitation.

In the United States alone, more than 700,000 people suffer a stroke each year, and approximately two in three of these individuals actually survive the stroke and require some form of motor rehabilitation, which often focuses on hand movement and control [1]. On a larger scale, the World Health Organization estimates a worldwide average of 15 million stroke victims every year [2]. According to the National Institute of Neurological Disorders and Stroke, “the main objectives of rehabilitation are to help survivors become as independent as possible and to attain the best possible quality of life” [1]. Hand rehabilitation is not purely limited to stroke patients, but also patients seeking motor improvements from Parkinson’s Disease and various forms of trauma and common injuries.

For patients with severe hand motor debilitations, reacquiring the ability to perform simple daily motions, including eating and writing, is a monumental step towards becoming independent again [2]. Learning and performing these motions typically correlates with developing a cyclic, natural frequency as a repetitive motor output. Most people develop and practice these daily motions in early childhood and become routinely acclimated to them overtime, yet relearning such fundamental, basic skills requires similar conscious practice and effort over long periods of time [4]. The NIH further highlights the importance of sustained, consistent exercising for long term improvements and retention of movements: “There is a strong consensus among rehabilitation experts that the most important element in any rehabilitation program is carefully directed, well-focused, repetitive practice—the same kind of practice used by all people when they learn a new skill” [1].

Not only is the recurrence of a specific hand motion or activity important to improve muscle memory and dexterity, it is also crucial to maintaining an acquired skilled motion overtime. Therefore, even though clinical rehabilitation within a physical facility under the direct care of a rehabilitation specialist may help patients improve in motor control and skill, long term improvements and maintenance of practical daily motions necessitates conscientious repetition and performance feedback [5].

1.2 Literature Review:

1.2.1 Existing sensor technologies within the field of extremity rehabilitation

In the modern field of extremity and hand rehabilitation, many sensor technologies currently exist both in terms of various types of motion sensors and larger scale motor rehabilitation devices. In both cases, sensor technologies incorporate one or multiple types of motion sensors, and the most widely used sensor technologies include accelerometers, force-based sensors , flex sensors , gyroscopes, and magnetometers [3].

Inertial Measurement Units (IMUs) are multi-sensor technologies that incorporate both an accelerometer, gyroscope, and magnetometer to track general hand motion and sensor orientation [6, 7, 8, 9], while flex sensors are resistive sensors that provide angular data for multi-joint movements from the fingers [4, 10, 11, 12]. Additionally, a majority of motion sensors have become widely accessible and affordable for research applications, paving the way for new technological developments that address more unique, patient-specific consumer needs [2].

Accelerometry is a widely used as a benchmark tool for motion activity recognition and assessment within the modern field of hand and extremity rehabilitation. Researchers at the University of California at Santa Barbara developed two different testing apparatuses, with 15 and 30 accelerometers in each respective apparatus, to measure the spatial patterns of skin tissue vibration during hand haptic interactions [13]. The three axis, miniature accelerometers in this experiment became of particular interest to our project design because of their versatility and high sensitivity in tracking broader hand movement and smaller vibrations and tremors, which are both key facets of extremity rehabilitation involving daily cyclic motions [3].

1.2.2 Motor Rehabilitation Technologies are Becoming Case Specific

In the modern field of motor rehabilitation, technologies are becoming increasingly case specific, which opens up a realm of opportunities for research and product design that address the individual rehabilitation needs and the unique physiology of the patient [3, 15, 5, 16, 17]. Here, we will provide a background of existing sensor technologies and suggest relative benefits and challenges associated with each technology when considering implementation within benchmark cyclic activity recognition and assessment systems for hand rehabilitation.

Researchers from the Institute for Biomedical Technology and Technical Medicine developed a hand motion assessment system design that uses inertial and magnetic sensors to create a full 3D reconstruction of all finger and thumb joints as well as the absolute orientation of the hand [7]. The system is highly sensitive and allows for an extensive dynamic range because the researchers attach the sensors individually to each finger and thumb joint on the test subject's hand as well as multiple positions on the backside of the hand. However, the complexity of the system design indicates that it can only be replicated and utilized in clinical rehabilitation settings through trained professionals. In addition, the widespread points of attachment for sensors would likely inhibit natural dexterous movement and be somewhat uncomfortable for sustained patient usage. The reusability of the system design is also very low since the sensors are not incorporated within a wearable device and necessitate individual, manual attachment.

A recent May 2019 research publication from Carnegie Mellon University used ultrasonic transducers mounted to a wristband to capture ultrasonic, acoustic beam-forms reflecting off of the hand [18]. Ultrasonic beamforming provides a highly sensitive and accurate solution to hand gesture recognition, and the transducer assembly on the wrist is practically non invasive because it allows for a full range of hand dexterity. The researchers compiled data from ten different test subjects using two gesture sets for both common hand poses and more complex gestures in three dimensions. The two sets achieved accuracies between “86.0% and 89.4...in sessions after the band is removed and reworn” [18]. One of the key benefits contributing to the average high accuracy is that the localization of sensors onto a rewearable wristband allows for more predictable and consistent sensor alignment and calibration. The only foreseeable drawback for the ultrasonic wristband research is that replicating this experiment requires additional

manufacturing for the physical hardware that the ultrasonic transducers mount to around the wrist.

Data gloves have become one of the most popular sensor integration materials that meet dynamic, case-specific needs for wearable gesture sensing technologies, and allow for greater user comfort and natural dexterity. Existing data gloves vary in the type of sensors attached to or embedded within the glove material, although most clinical or marketed data gloves most commonly use fiber optic sensors, resistive flex sensors, or IMUs [2, 10, 12, 19]. The reusability of data gloves is another important element that leads to both better retention of individual model accuracy across multiple usages without extensive calibration and a much faster, more comfortable system setup [12, 6].

1.2.3 Fuzzy Logic Systems

Data analysis for activity recognition and assessment over the last decade has increasingly shifted towards integrating fuzzy logic systems that simplify the software decision-making hierarchies and program operations for specialists. Within the field of hand and extremity rehabilitation, motor rehabilitation systems require significant processing power in order to sift through and apply fuzzy logic decision parameters to the large pools of data being collected from various sensors [20, 21, 15, 19]. Moreover, active fuzzy logic systems automatically generate new decision parameters that account for modeled unknown, or uncertain datasets that do not fall into pre-existing parametric classification [21].

Researchers from the Cheng Institute of Technology in Taiwan constructed a wearable heat stroke detection device in 2017 that integrates a fuzzy logic control parameter system to evaluate the physiological data from the sensor and determine whether the data is parametrically safe or crosses risk threshold values [22]. In the latter case, the user is notified that their monitored physiology is at risk of heat stroke and should take preventative action in order to lower the risk down to safe physiological levels. The fuzzy logic inference system the researchers implemented uses four risk level classification outputs defined by triangular membership functions, which are common for classification purposes with non-overlapping measurements within each input dataset.

A research paper published in 2017 developed an iterative, machine learning framework based on fuzzy logic in order to simultaneously account for several experimental constraints including “sensor data alignment, data losses, and noise [which] deteriorate[s] data quality and model accuracy” [20]. The researchers extracted motion data from an experiment dataset that contains over 800,000 data samples from four different test subjects. In the experiment linked to the dataset, eleven triaxial accelerometers and seven inertial sensors were mounted independently in thirteen unique placements ranging from lower to upper extremities on the surface of each test subject’s body. The accelerometers collected data at a rapid rate of 30 Hz, which translates to the massive number of data samples collected for the four test subjects.

The fuzzy logic inference system can save valuable processing time that would normally be squandered running data analysis through every individual data sample because it operates “using only a fraction of the data, improving significantly the computation time” [20]. However, the downsides to using less data are ensuing decreases in performance accuracy and sensitivity. Specifically, the study’s “iterative learning framework produced an average accuracy of 74.08% while using only 6.94% of the samples in the input domain for training. This result compares to the average accuracy of 81.07% obtained by the supervised method when using 80% of samples for training” [20]. Even though the former study accuracy using 6.94% of the data included 73% less sample data than the latter study, there was only a small 7% total difference between sample accuracies. This comparison demonstrates that in clinical applications that warrant faster processing and analysis speeds, fuzzy logic systems using optimized sample fractions can deliver fairly accurate and slightly less sensitive data analysis.

1.2.4 Summary of Challenges with Hand Rehabilitation Sensor Technologies

Despite the trends in digitization of health data and case specific technologies for motor assessment, challenges still exist with hand rehabilitation sensor technologies.

First, there are issues with interfacing actionable and accurate data because the sensors are actively reading and transmitting so much data, that the outputs are often difficult to interpret in real time and only understood by healthcare professionals [20, 23]. Simplification of data

processing and representation often leads to a loss in accuracy and sensitivity in data analysis [20].

Second, many clinical and research-based rehabilitation technologies require a complicated assembly that takes extensive amounts of time to set up with the patient and require 10-30 sensors for higher accuracy clinical assessment [13, 7, 8].

Third, many of the larger hand rehabilitation devices integrating smaller sensor technology components have sensors distributed at many points of contact or are uncomfortable to attach, which affects the patient's natural hand movement and dexterity and thus, affects the quality of data received [7, 24, 3, 8].

Lastly, there is a lack of integration of hand rehabilitation sensor technologies with consumer electronics to streamline the data analysis and visualized results for the benefit of both physician and patient [14, 5, 8].

1.3 Project goals and objectives:

Our preliminary project goal was to create a wearable system design for intention sensing and quality assessment to assist individuals with extremity motion rehabilitation. Intention sensing is often referred to as activity recognition within clinical research publications, and is the first step in constructing a more comprehensive rehabilitation system that utilizes both activity recognition and activity assessment to help patients [25]. Since intention sensing commonly requires a hardware sensor setup that either attaches directly to the patient's skin tissue or is incorporated within a wearable device, it is important to keep the patient's user experience and comfort in mind in order to allow for the most natural motor responses from the patient. Therefore, one paramount consideration for our system design was to minimize the number of sensors needed for intention sensing because the greater the number of sensors integrated within the system, the more likely the system will inhibit the user's natural dexterous motion. Additionally, reducing the number of sensors lowers the total amount of required materials and contributes to a more sustainable project design.

Activity recognition through intention sensing also requires the ability to classify a range of patient motions within a comprehensive rehabilitation system. For extremity motion rehabilitation, our benchmark activity recognition objective was to accurately identify six different motions that included both simple two-dimensional and more complex three-dimensional motions to account for a diverse, yet empirically accessible range of extremity movements and dexterous hand manipulations (Figure 1).

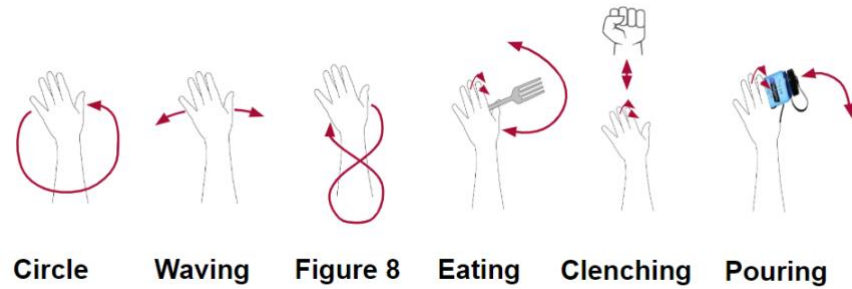


Figure 1: Two and Three-Dimensional Motions. The arrows stemming from the hands above diagram the general movement patterns and hand manipulations for our six objective motions.

1.4 Project Budget:

Item	Quantity	Price
Accelerometer w/ Evaluation Board	1	\$39.38
Arduino Uno	1	\$20.00
Armband	1	\$5.00
Bread Board	1	\$5.99
Dexterous Glove	1	\$12.99
Ecoflex	1	\$32.21
Electrical tape	1	\$1.95
Flex Sensor	2	\$15.90
MATLAB Student license	1	\$99.99
Fuzzy Logic Designer	1	\$29.99
50 cm Jumper Wires (120)	1	\$13.99
Sewing Kit	1	\$11.97
Total		\$291.31

Table 1: Project Budget.

1.5 Project Timeline:

Design Project Activities	Fall 2018			Winter 2018			Spring 2019		
	Week 1-4	Week 5-8	Week 9-11	Week 1-4	Week 5-8	Week 9-11	Week 1-4	Week 5-8	Week 9-11
Research	█	█	█	█					
Brainstorming		█							
Concept formulation		█							
Preliminary design review			█						
Conceptual design review			█						
Lab training					█				
Experiment Design				█	█				
Presentation Preparation				█	█	█			
Practice Presentation				█	█	█	█	█	
Final Presentation								█	
Thesis								█	█

Table 2: Project Timeline. This table represents a Gantt chart detailing the timeline and tasks of our project.

Chapter 2: Design Process

2.1 Design Goals:

The design goals for our project are to prove that we can obtain actionable, accurate data using a minimal amount of sensors and concurrently maintain an array design that is user-friendly and simplistic enough for interfacing with everyday rehabilitation usage.

2.2 Consumer Needs and System Requirements:

From our literature searches, we understood that using a large amount of sensors would provide us with an abundance of quality data, but we felt that many existing sensor arrays in research and clinical settings were uncomfortable, difficult to assemble, and not very user friendly [3, 7, 8, 13, 20, 23, 24]. Since our system design has the potential to eventually be incorporated within an at-home rehabilitation activity recognition and assessment device, the consumer needs must revolve around patient usage and accessibility. Therefore, we intend for our system design to serve as a benchmark model that maximizes wearable comfort, minimizes the complexity of assembly, and simplifies data analysis for user friendly interfacing strategies.

2.3 Preliminary Design:

Our preliminary design was used in all of our initial two-dimensional motion testing, but was not successfully implemented during our three-dimensional motion testing stage. This design incorporated a singular accelerometer attached with electrical tape on the back of a nitrile glove.



Figure 2: Design Schematic of the Preliminary Testing Apparatus. This image displays the testing apparatus used in the testing of our initial two-dimensional motions, detailing the components used and how they are connected.

2.4 Final Design:

Our final design was used for all of our three-dimensional motion testing as well as the bulk of our two-dimensional motion testing beyond establishing a baseline for classification. This design incorporated a single accelerometer sewn to the back of a dexterous glove. We also incorporated two flex sensors sewn to the base of the knuckle, each finger joint, and the tip of the finger to ensure the flex sensor followed the form of the fingers as they bent.

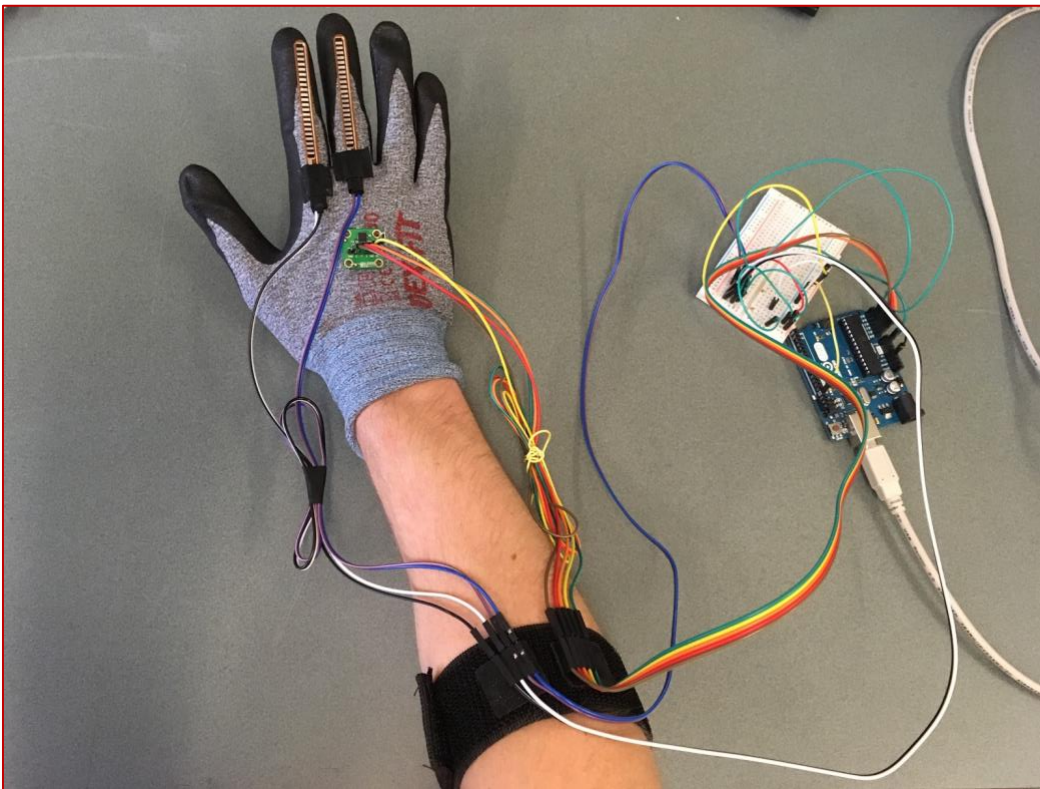


Figure 3: Design Schematic of the Final Testing Apparatus. This image displays the testing apparatus used in the testing of our three-dimensional motions after incorporating flex sensors into a reusable, dexterous glove.

Chapter 3: Methods and Materials

3.1 Hardware Components:

In designing our various testing apparatuses, we needed to incorporate several different hardware components to achieve the end results we desired. These hardware components would be responsible for transmitting motion data to be further processed and analyzed using software.

3.1.1 Accelerometers:

Accelerometers are electromechanical devices that measure the forces of acceleration, which is classified as the change in velocity over time ($a = \frac{\Delta V}{t}$ m/s²). By sensing the amount of dynamic acceleration, we are able to analyze how the specific device is moving in three dimensions, specifically, the X, Y, and Z axes. Through this information, we are able to determine how a test subject using our apparatus is moving their hand. We chose to use an accelerometer known as ADXL335 coupled with the model evaluation board (See Appendix 6 for data sheet) based on a previously conducted experiment at the University of California, Santa Barbara [13]. This experiment used the same accelerometers to measure the propagation of vibrational patterns within an individual's hand.

Knowing that vibrations generate small-scale accelerations, we determined that if this particular accelerometer was effective in detecting the most minute movements, it would be effective in capturing the larger movements we would be conducting in our experiments while maintaining a high level of sensitivity. The ADXL335 immediately reads acceleration with gravitational acceleration ($G = 9.8$ m/s²) as the initial unit of measurement. However, the accelerometer yields outputs values in mV using the calibrated function relating both voltage and gravitational acceleration, where the ratiometric correlation is defined by 300mV/G. The preset acceleration sampling rate of our ADXL335 sensors was 15 Hz, which implies 15 acceleration readings were recorded every second of operation.

3.1.1.1 Accelerometer Calibration:

In order to obtain accurate and consistent results across all of our tests, we had to make sure that our device was correctly calibrated. To do this, we used an Arduino Uno microcontroller board to process the acceleration signals, and wrote code in the Arduino software program that would first isolate the Z axis, and would then isolate the X axis (See accelerometer calibration code in Appendix 1). By aligning the Z axis of the accelerometer vertically upward, we are ensuring that the acceleration in both the X and Y axes are 0 m/s^2 since gravity, an acceleration force, is only acting in the Z direction. Once the X and Y axes are calibrated, we perform the same steps, but instead place the X axis straight up, allowing us to calibrate the Z axis. Once these calibration values are calculated, we can use them in our software analysis to make sure that the acceleration data we are gathering is correctly baselined and does not produce false results.

3.1.2 Flex Sensors:

Flex sensors are based upon resistive carbon elements and function as a variable resistor within a circuit. As the device is bent, the sensor produces a resistance output corresponding to the angle at which it is bent. By characterizing this correlation, we can use flex sensors to determine how an individual's fingers are moving during different hand motions.

3.1.2.1 Flex Sensor Characterization:

In order to integrate flex sensors into our system design, we must fully characterize the resistance behavior of each individual flex sensor (See Appendix 6 for data sheet. Characterizing the sensors allows us to create a relationship between the sensor's resistance output and the angle of bending. Each of our flex sensors were tested by securing the sensor to the table, and then bending it at a pivot roughly 3 cm from the tip of the sensor, simulating the bending of the sensor when attached to the finger (See Figure 4). The sensors were bent from $0-90^\circ$ at 10° increments in both the forward ($0-90^\circ$) and backward ($90-0^\circ$) directions. We made sure to include both directions to ensure that we would be able to record accurate bend angle values during both the contraction and extension of the fingers. Once characterized, our flex sensors were characterized (See figure 5) using a multimeter, we were able to integrate them into our sensor arrays.

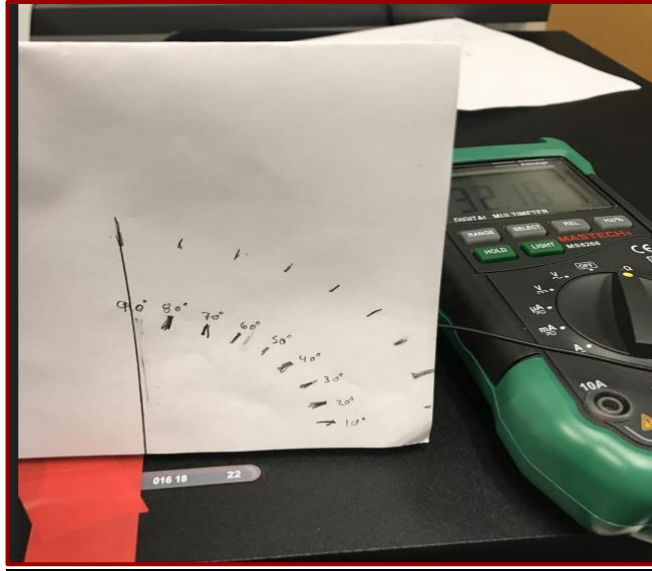


Figure 4: Test Setup for Flex Sensor Characterization. This image indicates how we characterized the flex sensors for our experiments.

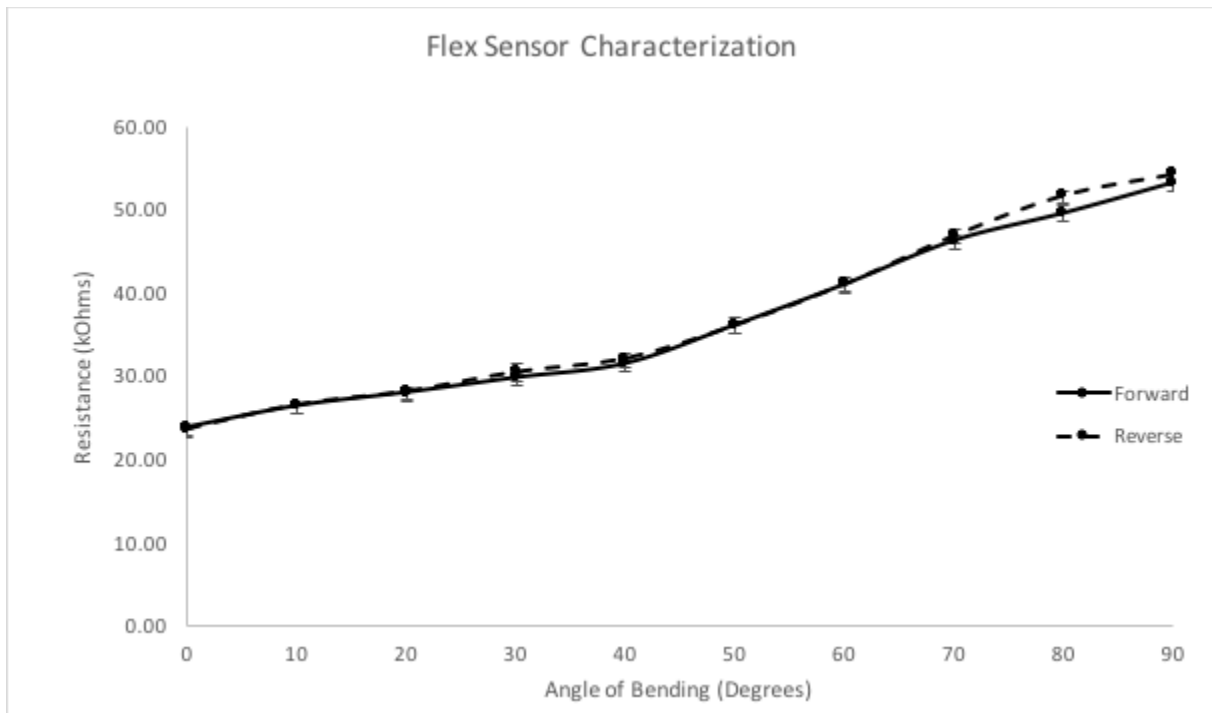


Figure 5: Flex Sensor Characterization. Average resistance values used were taken from five tests at angle increments of 10 degrees from 0-90 degrees. Error bars were calculated using the following equation for standard deviation ($\sigma = \sqrt{\frac{\sum(x-\bar{x})^2}{n}}$) where x = the specific value, \bar{x} = the average value, and n = the number of samples.

3.1.3 Wearable Glove Assembly:

We sewed the accelerometer directly to the top of the right-handed dexterous glove through the four holes on the corners of the ADXL335 evaluation board. We attached two wires to the I/O pins at the base of each flex sensor and folded electric tape multiple times around the pins to ensure the sensor pins wouldn't dislodge from the wires during usage. We then sewed the folded electric tape tightly to the glove below the metacarpal joints of the pointer and middle fingers and also sewed multiple loops further down each finger to maintain close angular contact between the flex sensors and bending fingers. The Velcro wristband allowed us to alleviate tension from the jumper wires directly onto the sensors and move a majority of wires out of range of the hand to mitigate any potential hardware interference with the user's natural hand dexterity and motion. Appendix 5 contains specific diagrams detailing the pin connections and circuitry for both the accelerometer and the flex sensor.

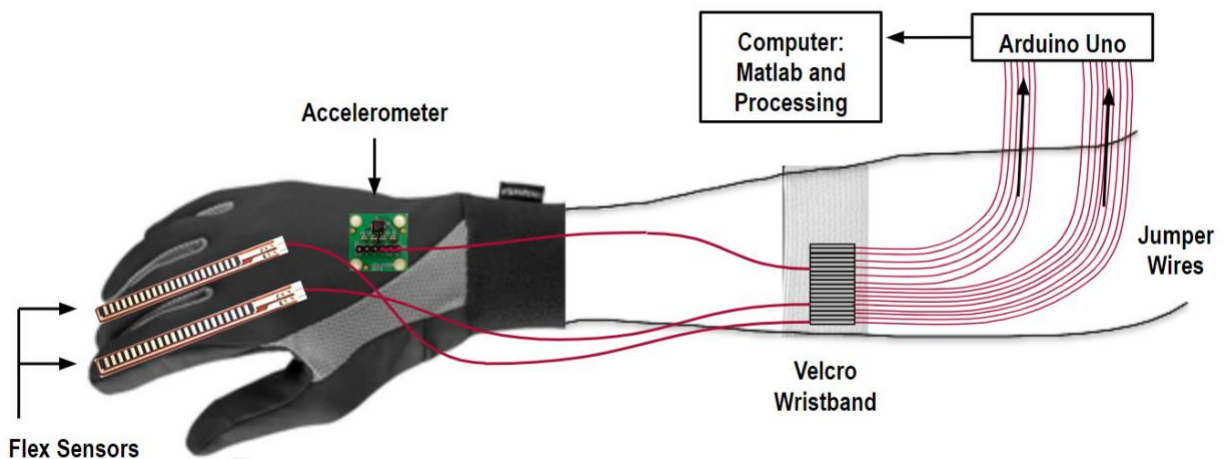


Figure 6: Hardware Components in Wearable Glove Assembly.

3.1.4 Data Acquisition Board:

For our data acquisition and processing transmission, we decided to use the popular Arduino Uno microcontroller board. We selected this particular model because it is relatively inexpensive, easily accessible to university researchers in laboratory settings, has a dynamic range of input and output capabilities, and our team members were already familiar with it from

previous experiences in bioengineering and electrical engineering courses. The board was suitable to both our preliminary design specifications and objectives as well as our final design specifications. For future applications involving motion sensor data processing and transmission, we recommend considering the Arduino Mega as a more reasonable, versatile option since the greater number of input pins would allow for users to include more sensors in their system design.

3.2 Software Components:

In order to collect, process, and analyze the data from our various hardware components, we incorporated several different software components that would be used to translate the quantitative motion data into qualitative classifications of movements.

3.2.1 Processing and Arduino:

Processing and Arduino are two software programs that allowed us to acquire the data being transmitted by the accelerometers and flex sensors into a form we could use later in our analysis. Arduino is a C/C++ based language that we used to communicate directly with the hardware components by taking analog output signals and converting them to digital signals (See data acquisition code in Appendix 1). Digital signals are discrete compared to the continuous analog data from the sensors, and therefore are easier to process and interpret when through data analysis. Processing is a Java based language that we used to communicate between the Arduino board and our computer (See Processing code in Appendix 2). Using Processing allowed us to compile our accelerometer data into an Excel file for each individual motion test that we could then analyze later using MATLAB.

3.2.2 MATLAB:

MATLAB is a programming language designed specifically with engineers and researchers in mind because of its fluidity and ease of use in the realm of computation. For our project, we used MATLAB for all of the data analysis including data transformation into the frequency domain and optimization of both data collection and fuzzy logic systems (See Appendix 3 for MATLAB code).

3.2.2.1 Data Optimization:

As we began running experiments and trials with our preliminary two dimensional sensor array, we found that there were often periods of miscommunication between the Arduino board and the Processing software. This period of delay led to a collection of “blank” values scattered throughout our Excel sheets of accelerometer data. In addition, based on our calibration data and our preliminary testing, we decided that accelerometer readings exceeding 500 mV and falling below 200 mV were considered to be outliers. Furthermore, when initially trying on our glove sensor apparatus with sewn flex sensors, we physically measured the maximum potential bend of flex sensors from the carpometacarpal joint to the proximal interphalangeal joint to be 110°. Despite this, transduced flex sensor angular values sometimes exceeded 110°, so we set an angular outlier threshold value at 110° for the flex sensor measurements.

In order to accommodate for these discrepancies in our collected data, at any point in the Excel file where a blank value or outlier occurred, we programmed MATLAB to remove the entire row of data containing the outlier (See data optimization code in Appendix 3). Through this outlier removal process, we ensured that we retained each individual instance of uncompromised data that remained.

3.2.2.2 Fourier Transforms:

A Fourier transform decomposes a complex signal in the time domain into the frequencies of individual values that make up the signal and is a widely used linear transformation method within the fields of bioanalytics and bioengineering. Executing a Fourier transform on a particular dataset provides us with magnitude values corresponding to a series of frequencies. The greater the magnitude at a given frequency within the Fourier spectrum, the greater the presence or frequency of a specific measured value in the original signal over a given time period.

3.2.3 Fuzzy Logic Designer:

Fuzzy Logic is a toolbox feature within the MATLAB software that allows us to perform data analysis through establishing hierarchical decision-making parameters that enable complex pattern recognition. For each input to the system, we created a group of membership functions to

further define the inputs. Membership functions are curves that define how each point in the input space is mapped to a membership value between 0-1. If a particular input falls within a particular membership function, its membership value will be between 0-1, while all of membership values for all other membership functions within that input will have a value of 0. The use of membership functions is how the fuzzy logic designer is able to determine an output from a set of inputs (See figure 7 below). For our project's fuzzy logic system in both two and three-dimensional motion classification, the accelerometer inputs are the axial ratios of frequency and magnitude values from each of the Fourier transforms corresponding with a specific trial dataset. The flex sensor inputs for motion classification are the frequency magnitude values from each of the Fourier transforms corresponding with a specific trial dataset. Each of these inputs, and their significance, are explained more in depth in Chapter 4. The outputs of our fuzzy logic designer are the six motions performed during our testing (circle, waving, figure 8, clenching, eating, and pouring). For each data set, we compile the eight inputs into a single matrix, and then run it through our fuzzy logic designer.

3.2.3.1 Determining Membership Functions:

Before writing and organizing a specific hierarchy of rules within the fuzzy logic designer to isolate similar motion patterns from multiple datasets, we first need to manually define membership functions that segment the various inputs into subsections, as seen in figure 7 below. Fuzzy logic designers have the ability to utilize a wide variety of membership functions, ranging from triangular to trapezoidal to Gaussian, with each type having inherent strengths and weaknesses depending on the type of data you are analyzing. In the research paper discussing the wearable heat stroke detection device, the authors utilized triangular membership functions for their outputs and several inputs because of its specificity and simplicity in determining whether or not an input fell within a particular membership function [22]. Thus, we specifically selected triangular membership functions for our system design because there is no need for overlapping of membership functions for our inputs.

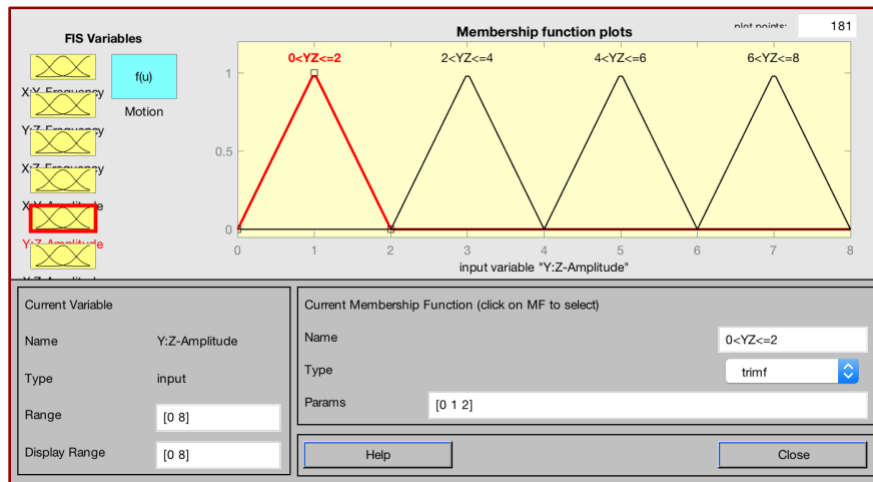


Figure 7: Membership Function Plots. This image displays the triangular membership functions mentioned above for one of our chosen inputs.

3.2.3.2 Establishing Rules:

The rules system within fuzzy logic functions as a grouping of “if...then” statements incorporating as many inputs as needed to specify an output. With the absence of machine learning, each individual dataset had to be analyzed by hand, and new rules were written to classify each additional dataset. Each rule requires the user to specify which membership function each of the inputs falls into, as well as the intended output (See figure 8 below). Once a rule is specified, if any further datasets result in the same sequence of membership functions, the motion will be classified as such. The more rules established within the system, the more accurate your motion classification will be (See Appendix 4 for all of our written rules).

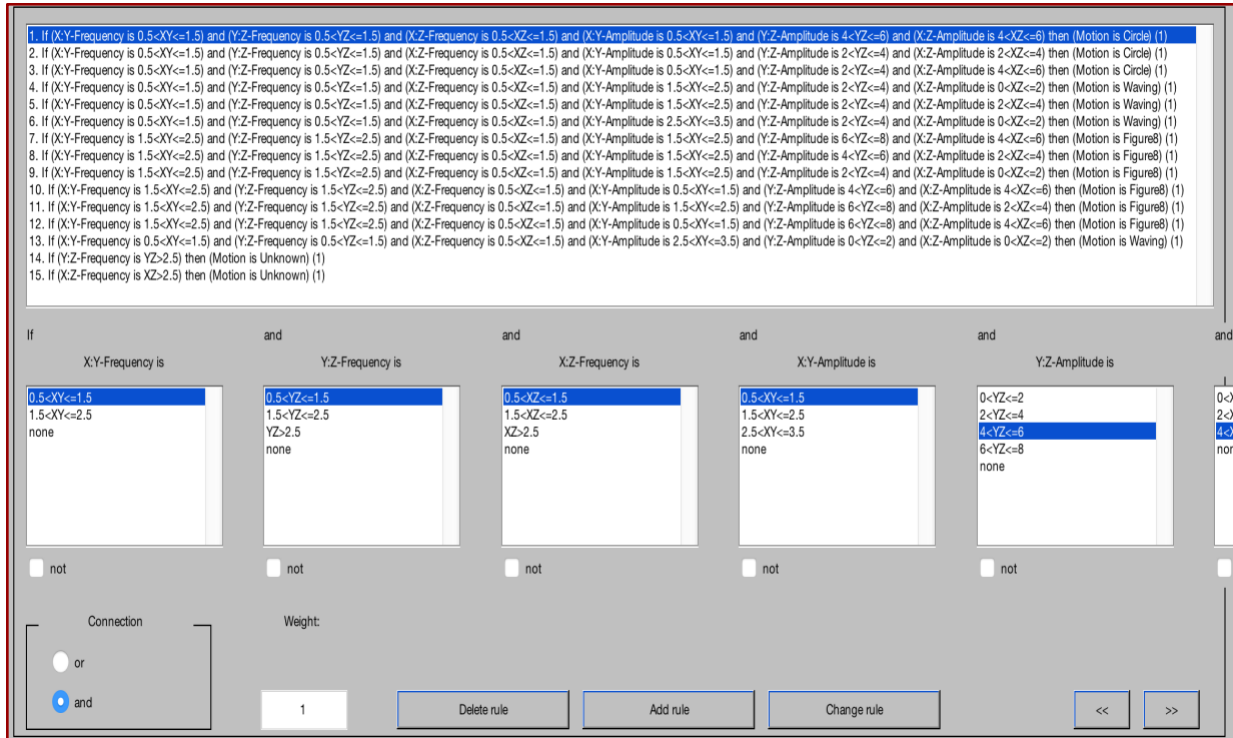


Figure 8: Creating Rules for the Fuzzy Logic Designer. This image shows the system used to implement new rules within our fuzzy logic designer.

3.3 Testing Procedures:

For both two and three-dimensional motion testing, we designed our experimental protocol around the user experience through maximizing the efficiency, comfort, and simplicity of our operational system design. Beyond that, we took into account health and safety considerations to make sure that the rewearable glove didn't collect residual germs and could be sustainably reused for an indefinite number of tests.

Testing procedures begin after hardware setup is complete and connected with the Processing and Arduino software preprogrammed with our system design code on the professional or researcher's computer. For a majority of the two and three-dimensional motion tests, each patient performed two or three tests of the same motion before moving onto the next, new motion.

For the researcher or professional conducting the tests, it is important to navigate the subject through each step of the process and quickly demonstrate the individual motions before beginning the experiment. The researcher must also make sure to rename each specific test file before each 15 second motion tests is conducted, while abiding by the nomenclature described in the following testing procedures.

1. Subject washes their hands with soap and water
2. Subject applies hand sanitizer to their hands to kill off residual germs
3. Subject slides their right arm through the armband and slides the armband to approximately $\frac{1}{2}$ or $\frac{2}{3}$ down the forearm towards the elbow before tightening the Velcro strap to secure the band
4. Subject inserts their right hand into the glove and maneuvers their hand and fingers to check and adjust for maximum dexterity
5. For each motion test:
 - a. Researcher names the specific test file in Processing using the following nomenclature:
 - i. Data(Motion)(Date)(Patient's initials)(Test # of the same motion).csv
 - ii. An example would be "DataFig8May6ML3.csv"
 - b. Researcher presses "Run" in Processing on the computer

- c. 2 second delay between pressing “Run” and when sensor data acquisition begins
 - d. Subject begins performing the intended motion after the 2 second delay while the researcher keeps track the 15 seconds using an external timer
 - e. Researcher presses “Stop” in Processing on the computer once the 15 seconds have elapsed
 - f. Subject stops performing the intended motion
6. Subject takes off glove and armband
 7. Subject applies hand sanitizer once again

Chapter 4: Results

4.1 Hypothesis:

We will be able to integrate a specific combination and minimal amount of sensors into a wearable device that will allow us to use Fourier transform based data analysis and fuzzy logic to enable the automation of activity recognition and assessment.

4.2 2-D Motion Testing:

The first step in our testing process required us to establish a baseline for classifying motions. While the end goal of our project was to be able to accurately classify essential, everyday motions, we needed to start with a simple preliminary benchmark to ensure that our method of classification was both viable and successful. In order to accomplish this, we chose three distinct, cyclic, two-dimensional motions (Figure 9). We selected cyclic motions because they exhibit natural frequencies and are represented clearly through Fourier transforms, our main method of data analysis. By restricting the motions to two dimensions, we kept the motion complexity low to prove our strategy could work on a simpler scale before incorporating more complex three-dimensional motions.

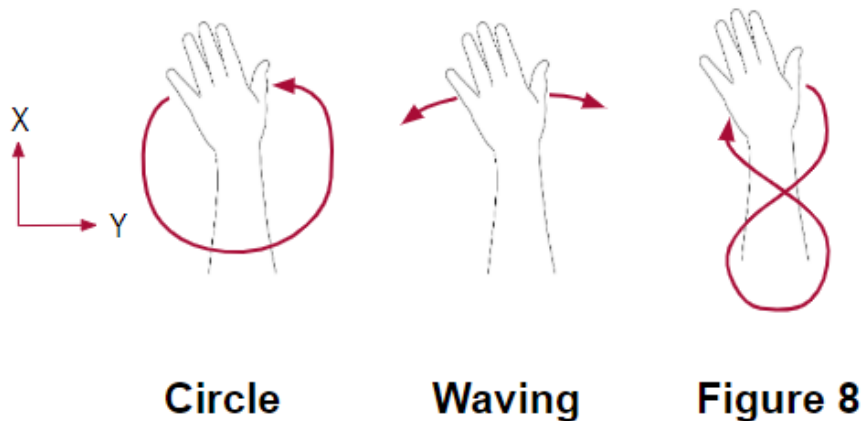


Figure 9: Two-Dimensional Motions. The arrows stemming from the hands above diagram the two-dimensional motions we used in our preliminary testing (circle, waving, figure 8).

4.2.1 Acceleration Data:

Our first data tests were conducted with our preliminary testing design consisting of only a single accelerometer. Figure 10 shows the raw acceleration data displayed within MATLAB from one of our preliminary tests. When broken down into sections and surveyed closely to compare axial relationships of acceleration at specific times, we were able to determine that our circular motion was indeed represented by the figure below. Despite this, the raw data was not consistent enough from test to test to visually distinguish between motions with confidence. However, this was a crucial step in our research process because it proved we could represent motions graphically using an accelerometer as a data source.

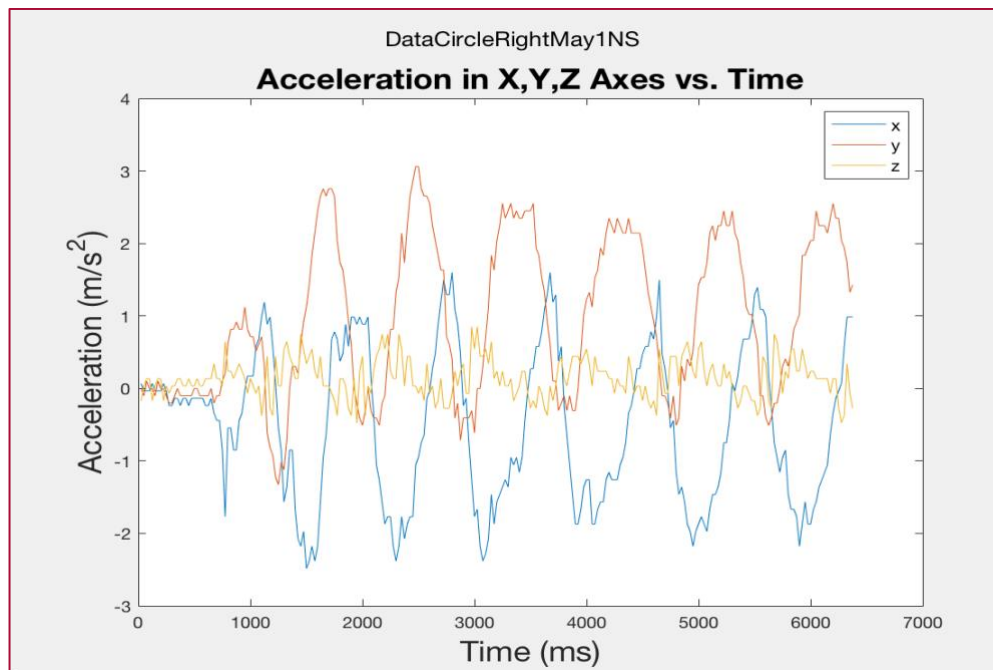


Figure 10: Raw Acceleration Data of a Circular Motion in MATLAB. This image depicts the acceleration in the X, Y, and Z axes of a repeated circular motion.

4.2.2 Data Analysis using Fourier Transforms:

After obtaining data from the accelerometers and graphically representing them within MATLAB, we performed Fourier transforms on our datasets. As previously mentioned, Fourier transforms break down our complex acceleration signals and simplify them into a collection of frequency spectrums. We found that implementing these Fourier transforms on our data provided us with a much more efficient and effective way to distinguish between motions.

As you can see in Figures 11-13 below, the Fourier spectrums of the different two-dimensional motions showcase distinct variances. For the initial circle motions, the overwhelming majority of our data showed magnitude peaks at the same frequencies as well as a similar magnitude values for both the X and Y axes. In an ideal test, the magnitudes for both the X and Y axes would be identical due to the symmetrical nature of a circle. In contrast, the waving motion also shows the peaks at the same frequency, but we see a much higher magnitude in the X axis compared to the Y axis.

These results make sense because a waving motion is predominantly planar, and a majority of motion occurs along one axis. In the figure 8 motion, we see the magnitude peaks for the X and Y axes occur at different frequencies, indicating a key difference from the other two-dimensional motions. This is primarily because when motioning through one cycle of a figure 8, the hand accelerates and changes direction along the X axis four separate times, while only accelerating and changing in direction twice along the Y axis. In all of these graphs, the magnitude for the Z axis is negligible since there is no predominant motion along this axis.

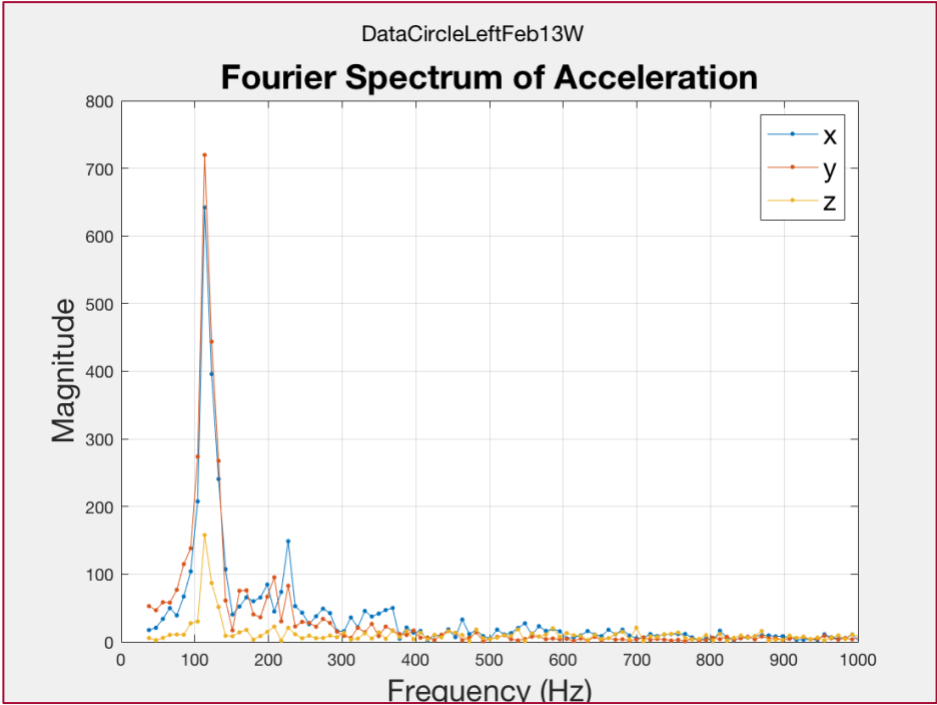


Figure 11: Fourier Spectrum of a Circular Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) for acceleration in each of the three axes (X, Y, and Z).

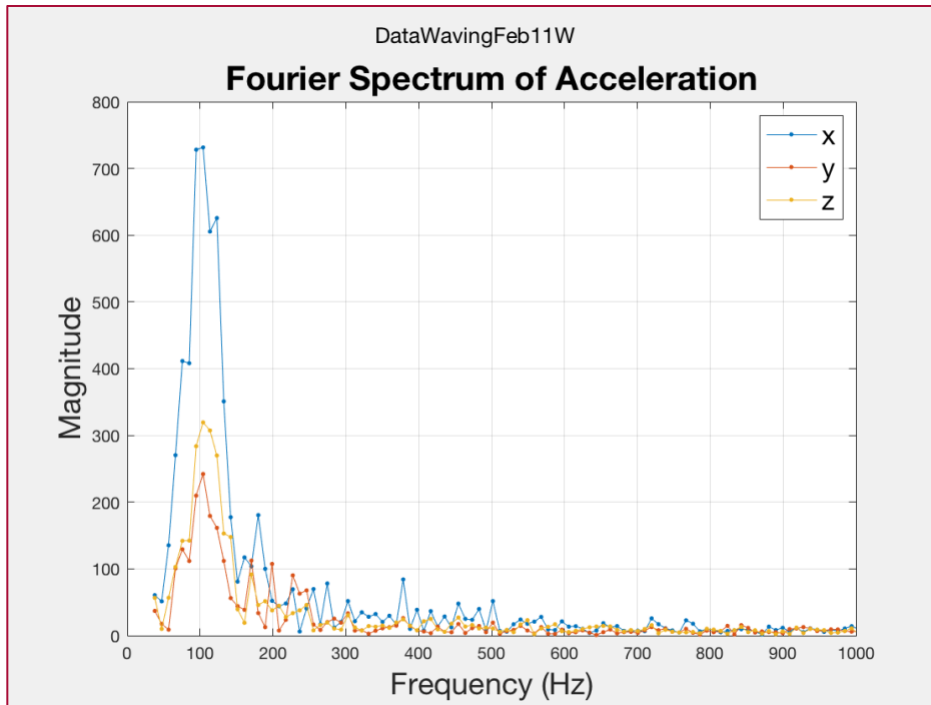


Figure 12: Fourier Spectrum of a Waving Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) for acceleration in each of the three axes (X, Y, and Z).

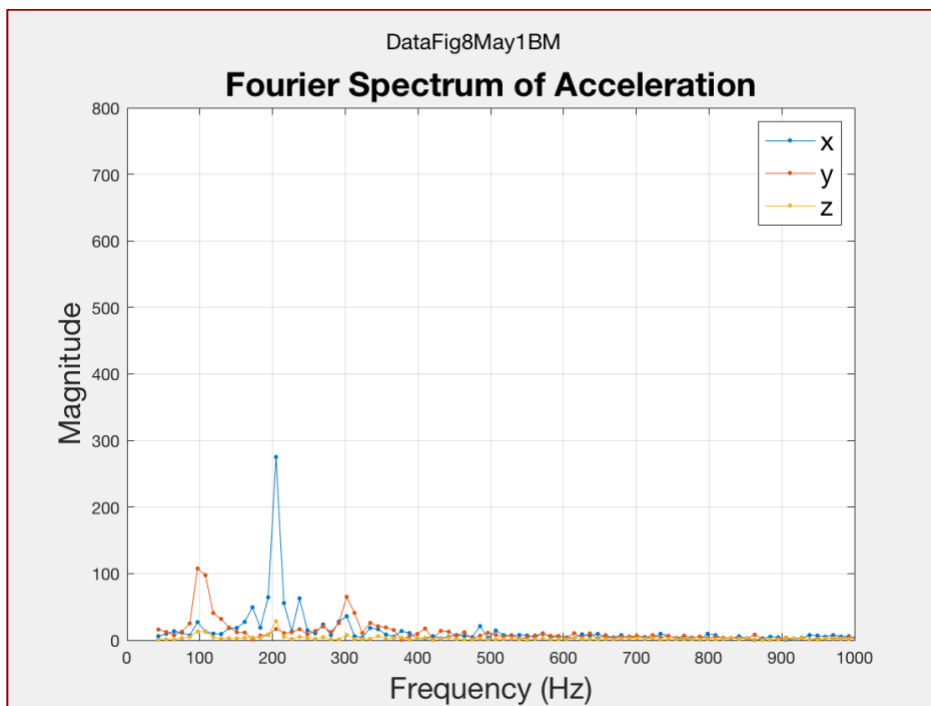


Figure 13: Fourier Spectrum of a Figure 8 Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) for acceleration in each of the three axes (X, Y, and Z).

4.2.2.1 Significance of Axial Ratios:

Upon performing Fourier spectrums across a multitude of trials for each of the three two-dimensional motions, we noticed an inconsistency in the magnitude peak values, as well as the frequency at which these peaks occurred across tests of the same motion. While this was initially concerning, we were able to find a distinct, distinguishing factor from these Fourier transforms. While the raw values of magnitude peak and frequency were not consistent for the same motion, the ratios of the peak magnitude between axes and the ratio of the frequencies between axes were consistent from test to test across datasets with the same motion. In addition, as depicted in Table 3 below, the ratios provide us with a quantifiable metric with which to compare motions.

Frequency Ratios			
	Circle	Waving	Figure 8
X:Y	1.0	1.0	2.1
X:Z	1.0	1.0	1.0
Y:Z	1.0	1.0	0.5

Magnitude Ratios			
	Circle	Waving	Figure 8
X:Y	0.9	3.0	2.6
X:Z	4.1	2.3	9.7
Y:Z	4.6	0.8	3.8

Table 3: Comparing Frequency and Magnitude Ratios of Two-Dimensional Motions. This table shows the significance of the ratios in distinguishing between the various two-dimensional motions.

As the table shows, the figure 8 motion is distinguishable from the circle and waving motions by reviewing only the values of the frequency ratios. While the magnitude ratios aren't essential in distinguishing the figure 8 motion, they provide additional parameters to reference in order to be certain that the correct motion is being classified.

In the case of distinguishing between the circle and waving motions, the frequency ratios alone are not enough since they are identical. Including the magnitude ratios into our comparison is essential to classifying these two motions. As we can see from Table 3 and Figures 11 and 12, the waving motion has significantly higher magnitude peaks on the X axis than either the Y or the Z axis, whereas the circular motion we see peaks of similar values corresponding to a ratio

between the X and Y axes of 1. The axial ratios were thus significant to our ability to distinguish between various two-dimensional motions, and also served to prove that our minimal sensor design consisting of a single accelerometer was sufficient to obtain actionable data.

4.3 3-D Motion Testing:

Upon successful completion of our preliminary testing using two-dimensional motions, we aimed to use a similar methodology to distinguish between three-dimensional motions. We again chose motions that were cyclic in nature, but also made sure that the motions held significance in a majority of people's' everyday lives. We decided to focus on the motions of eating, clenching, and pouring in our three-dimensional testing (See Figure 1). In conducting these experiments, we used our final testing apparatus diagrammed in Figure 3.

4.3.1 Acceleration Data:

Similar to our two-dimensional motion testing, our three-dimensional motion testing began using the same single accelerometer system design depicted in Figure 2. After several tests across the various three-dimensional motions, we discovered that once again the raw acceleration data would not be sufficient in classifying the various three-dimensional motions (see Figures 14-16). Rather, the raw acceleration data would suffice as an intermediate reference to display the axial relationships of motion.

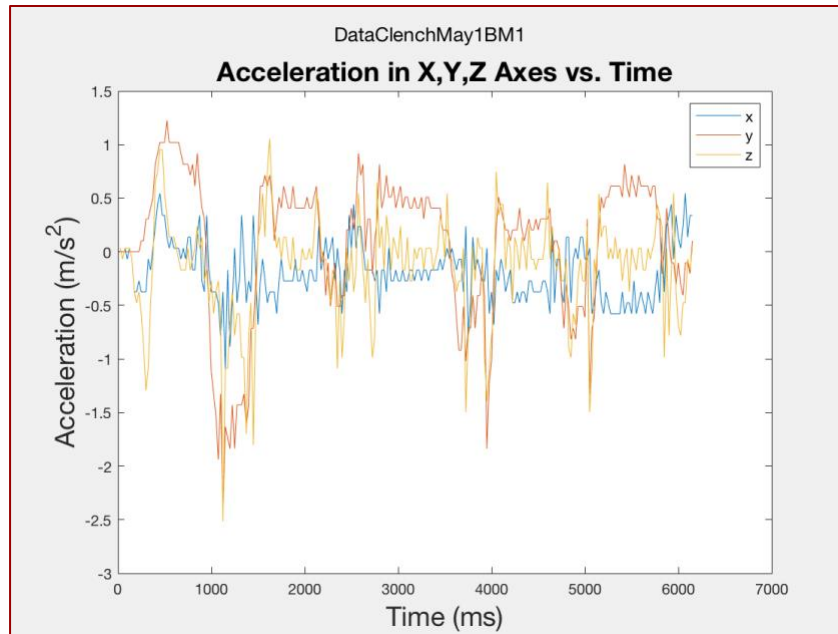


Figure 14: Raw Acceleration Data of a Clenching Motion in MATLAB. This image depicts the acceleration in the X, Y, and Z axes of a repeated clenching motion.

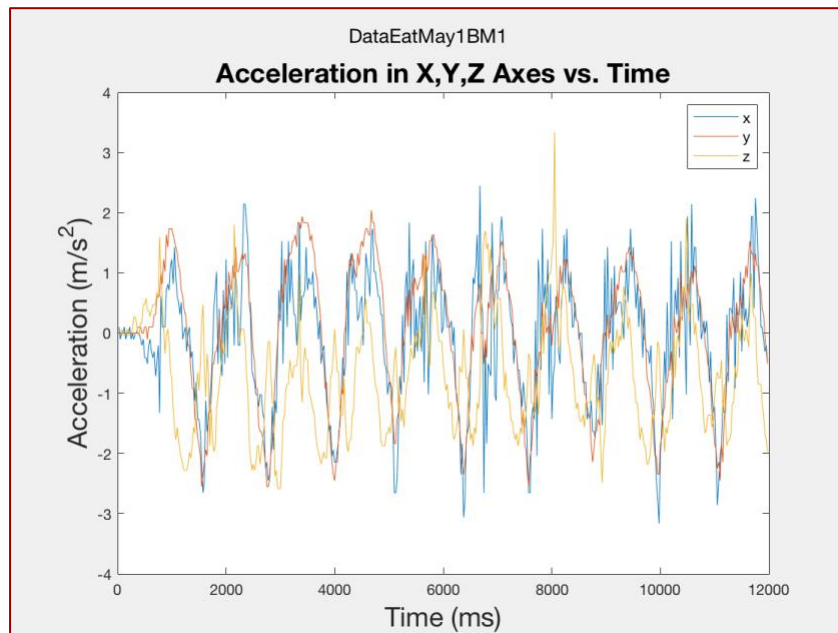


Figure 15: Raw Acceleration Data of an Eating Motion in MATLAB. This image depicts the acceleration in the X, Y, and Z axes of a repeated eating motion.

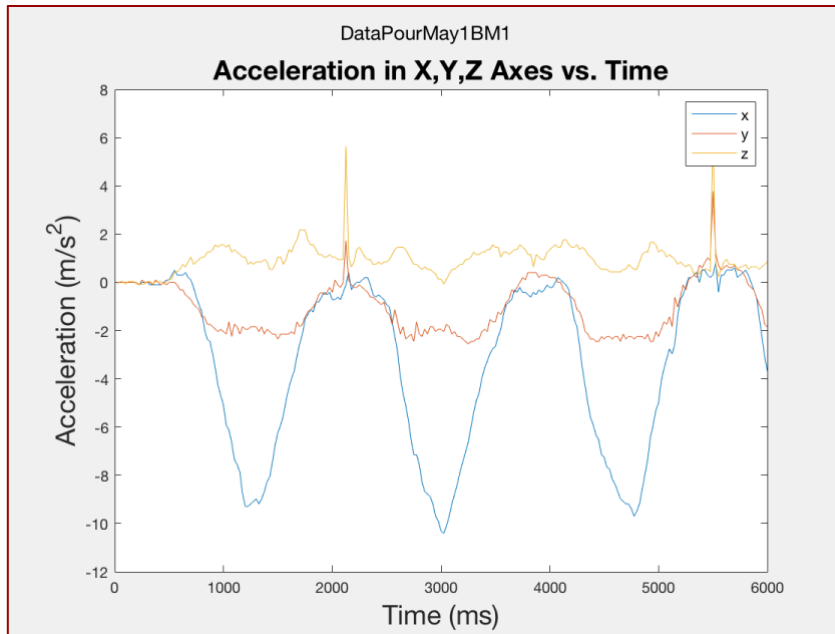


Figure 16: Raw Acceleration Data of a Pouring Motion in MATLAB. This image depicts the acceleration in the X, Y, and Z axes of a repeated pouring motion.

4.3.2.1) Data Analysis using Fourier Transforms -- Acceleration Data:

Similar to our two-dimensional motion testing, we conducted Fourier transforms on the raw acceleration data from all of our three-dimensional motions (see Figures 17-19 below).

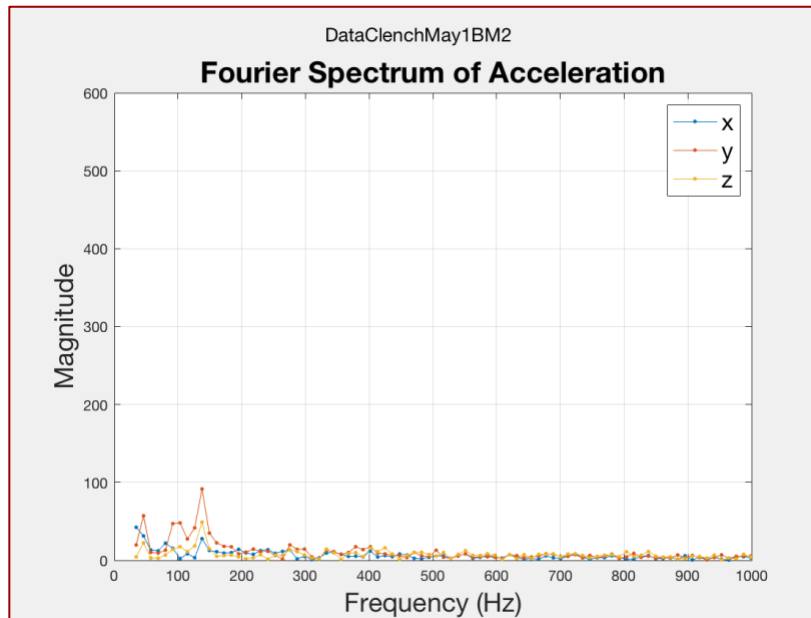


Figure 17: Fourier Spectrum of Acceleration for a Clenching Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in each of the three axes (X, Y, and Z).

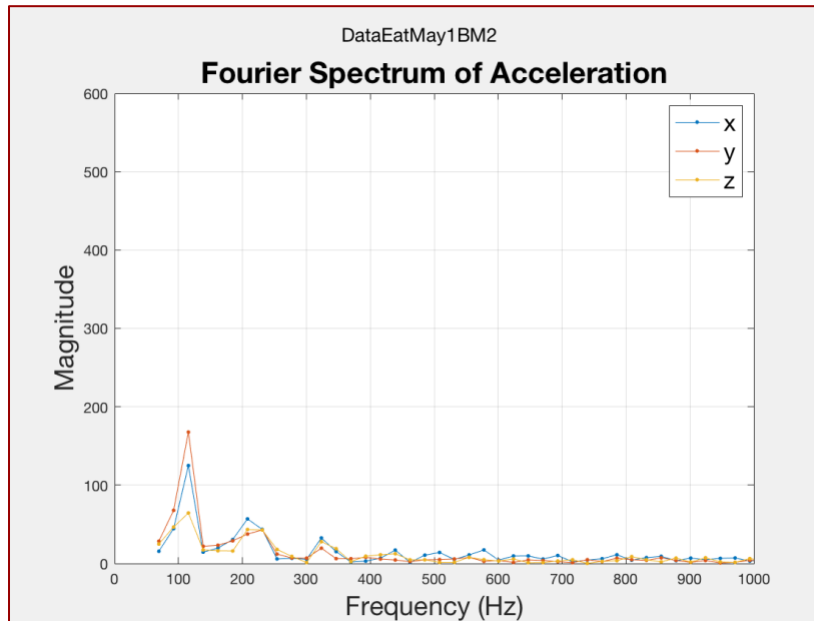


Figure 18: Fourier Spectrum of Acceleration for an Eating Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in each of the three axes (X, Y, and Z).

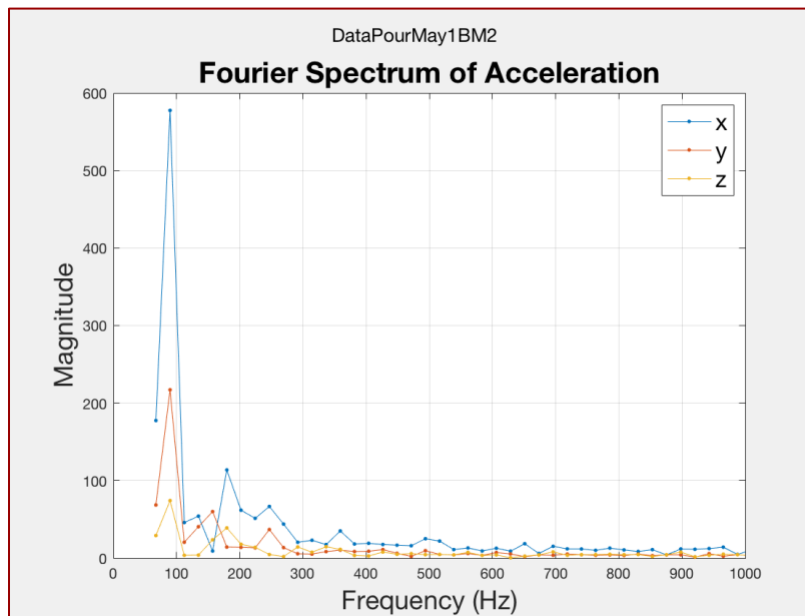


Figure 19: Fourier Spectrum of Acceleration for a Pouring Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in each of the three axes (X, Y, and Z).

We used the same ratio comparisons as in the two-dimensional motion analysis in order to attempt distinguishing between motions. However, classification proved to be ever more complicated with our more complex, three-dimensional motions. We found that many of our trials contained magnitude and frequency ratios similar to those of our two-dimensional motions. Thus, distinguishing between all six of the motions became increasingly difficult. We quickly realized that a single, central accelerometer on the back of the hand was not sufficient to distinguish between three-dimensional motions and necessitated including an additional parameter to aid us in classification.

4.3.2 Flex Sensor Data:

A critical difference we noticed between the two-dimensional motions and the three-dimensional motions were the usage of the fingers in the three-dimensional motions. In each of the two-dimensional motions, the fingers of the test subject were either straight, or bent so slightly that it could be deemed negligible. In contrast, all of our three-dimensional motions incorporated a substantial amount of finger bending and manipulation, with the eating and pouring motions requiring the test subject to grip an object, and the clenching motion requiring the repeated bending of the fingers into a fist. The bending described above is depicted in Figures 20-22 below.

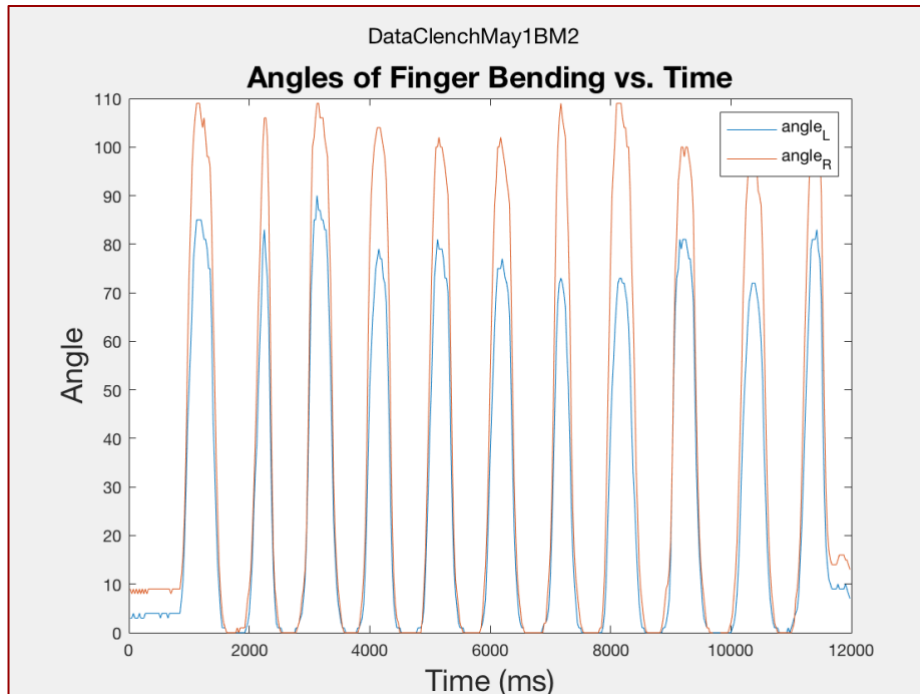


Figure 20: Raw Flex Sensor Data of a Clenching Motion in MATLAB. This image depicts the angle of finger bending in the index (L) and middle (R) fingers of a repeated clenching motion.

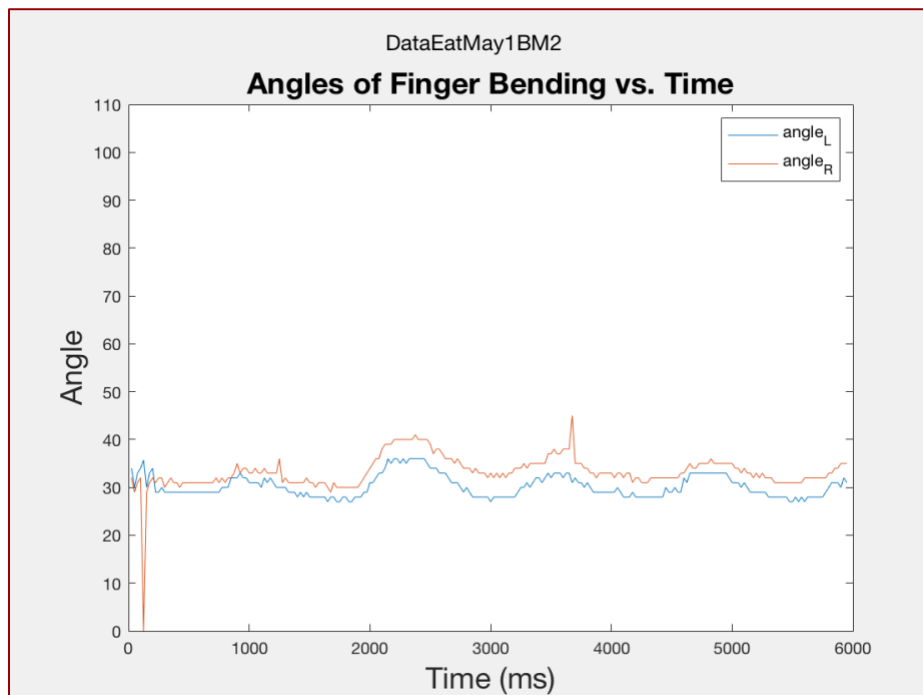


Figure 21: Raw Flex Sensor Data of an Eating Motion in MATLAB. This image depicts the angle of finger bending in the index (L) and middle (R) fingers of a repeated eating motion.

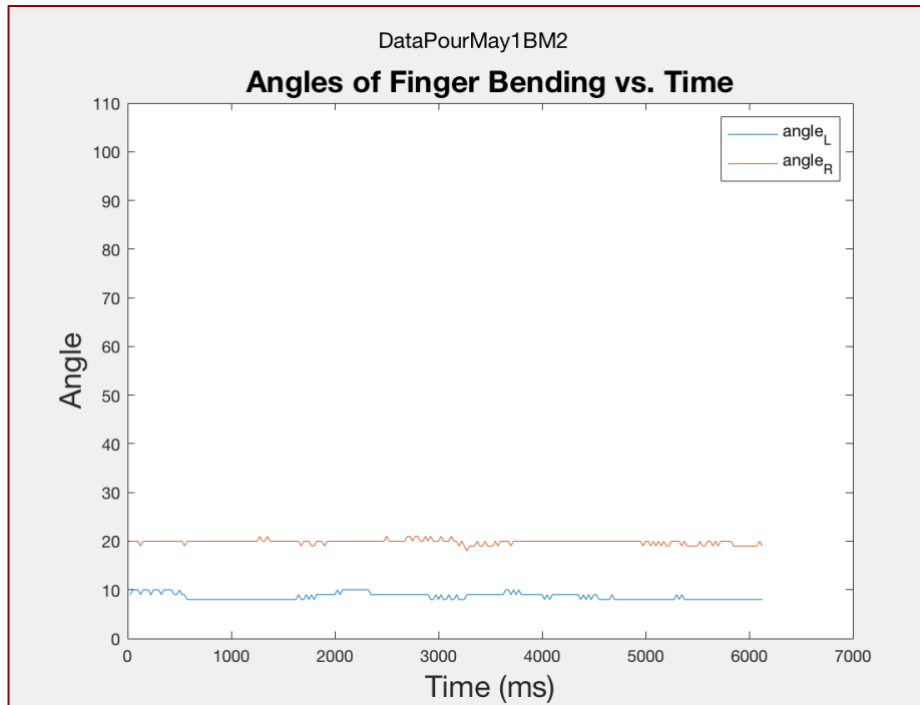


Figure 22: Raw Flex Sensor Data of a Pouring Motion in MATLAB. This image depicts the angle of finger bending in the index (L) and middle (R) fingers of a repeated pouring motion.

We can see from the preceding figures that each of the three-dimensional motions requires the use of the fingers in some capacity due to fluctuations in the angle values on the Y axes overtime. Similar to the raw acceleration data, simply looking at the raw flex sensor data was not conclusive in distinguishing between different motions.

4.3.2.1 Data Analysis using Fourier Transforms - Flex Sensor Data:

Given the cyclic nature of our motions, we implemented Fourier transforms once again as analysis tools for our data. As we can see in Figures 23-25 below, the Fourier spectrum magnitudes for the three-dimensional motions have a wide range.

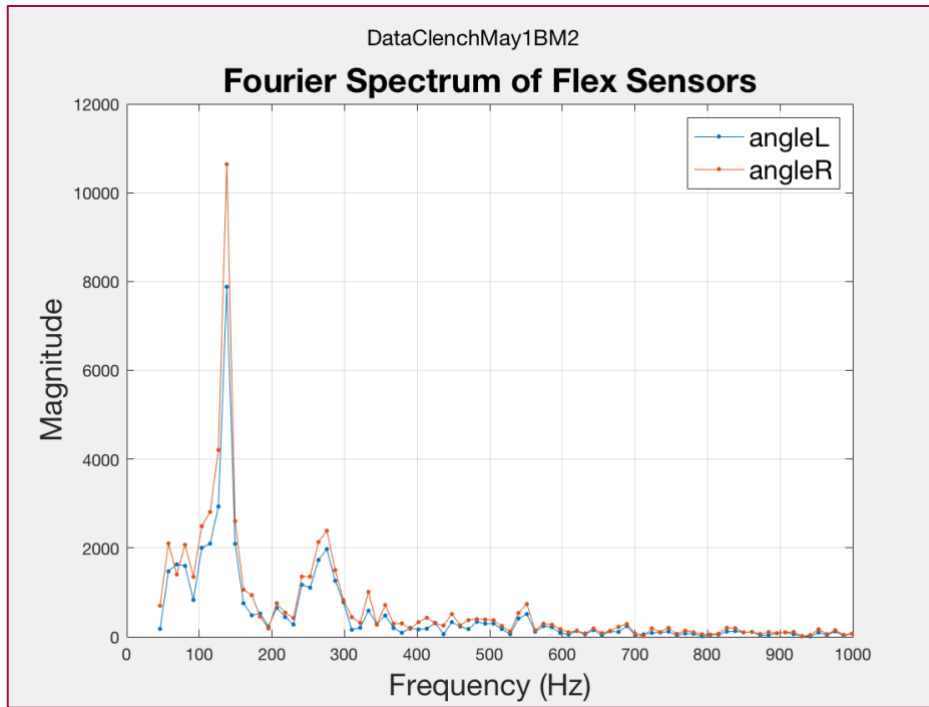


Figure 23: Fourier Spectrum of Flex Sensor Data for a Clenching Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in the index (L) and middle (R) fingers.

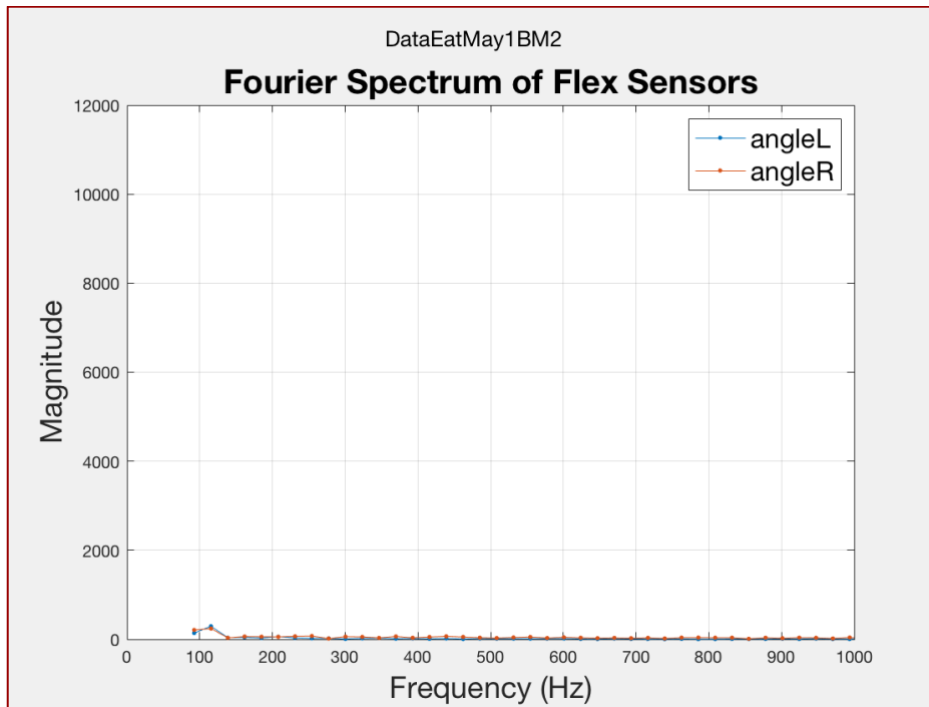


Figure 24: Fourier Spectrum of Flex Sensor Data for an Eating Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in the index (L) and middle (R) fingers.

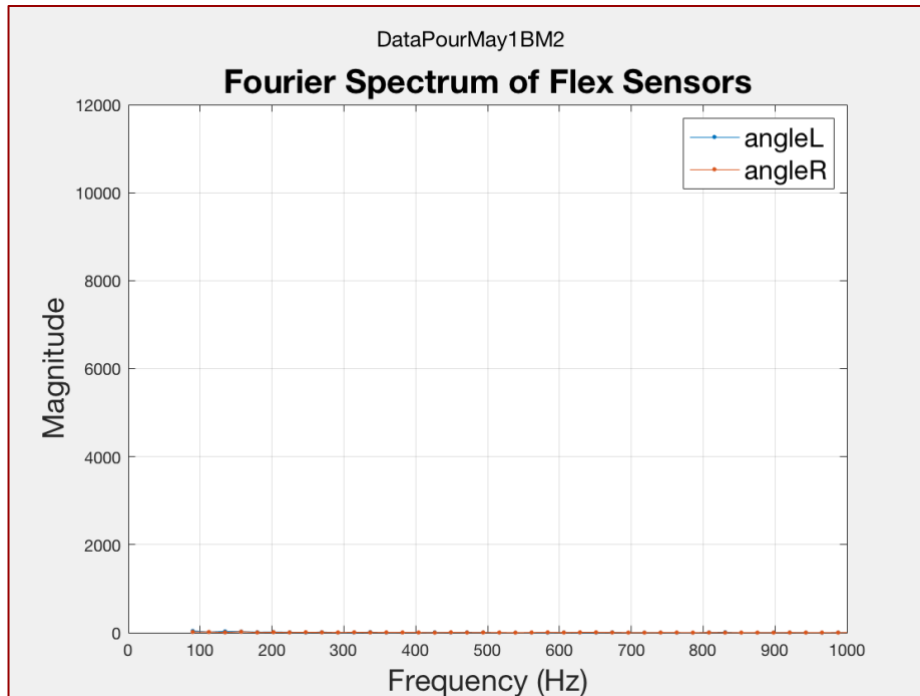


Figure 25: Fourier Spectrum of Flex Sensor Data for a Pouring Motion in MATLAB. This image shows the magnitude vs. frequency (Hz) in the index (L) and middle (R) fingers.

We noticed that the magnitude of the clenching motion was significantly higher than the magnitude of the other two motions. This is primarily because in the pouring and eating motions, the fingers mostly remained bent at the same angle throughout the course of the repeated motions, while the clenching motion required repeated bending of the fingers to alternate between flat and gripped orientations.

4.3.2.2 Significance of Fourier Spectrum Magnitude:

From Figures 23-25, we were able to determine that the frequency magnitude from the Fourier spectrum for the flex sensors was essential in distinguishing between the two-dimensional and three-dimensional motions.

	Two-Dimensional Motions			Three-Dimensional Motions		
	Circle	Waving	Figure 8	Clenching	Eating	Pouring
Magnitude Range	0-50	0-150	0-50	>1000	150-1000	100-1000

Table 4: Comparing the Magnitude Ranges of Fourier Spectrums of Flex Sensor Data.

This table shows the significance of the magnitude of finger bending in distinguishing between two-dimensional and three-dimensional motions.

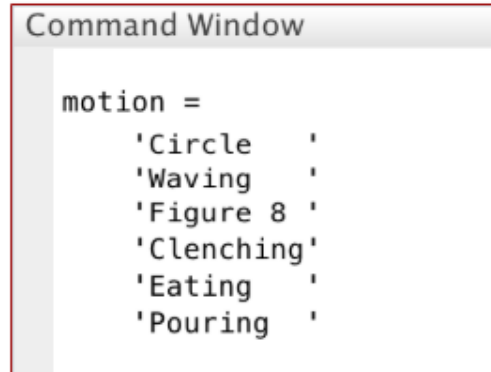
From the table above, we are able to deduce that incorporating the flex sensors into our sensor array has allowed us to successfully distinguish between two-dimensional and three-dimensional motions. Movements that saw overlap from the accelerometer data, like the circle and eating motions, are now easily distinguishable by incorporating the additional parameter provided by the flex sensor data.

4.4 Motion Classification using Fuzzy Logic:

After establishing connections between our data (frequency ratios, magnitude peak ratios, and magnitude ranges of bending) and our six motions, we used the Fuzzy Logic Designer system within MATLAB to classify our data in a more automated fashion. While we were able to visually determine an unknown motion by looking at the graphed data ourselves, this method did not give us the precision or speed needed for further applications of our project. The fuzzy logic system provided us with a way to achieve quick results for a large number of data sets.

After establishing the membership functions and rules for the system (details in sections 3.2.3.1 and 3.2.3.2 respectively) we were able to use each individual ratio (three frequency ratios and three magnitude ratios) as well as the magnitude of bending (index and middle finger) as inputs to our fuzzy logic inference system adding to a total of 8 inputs for data analysis. Our system contained six total motion outputs, one for each of the unique motions performed in our test experiments. As we run our data sets through our fuzzy logic design system, each data set is assigned a numerical output corresponding to a particular motion. Using this numerical value, we can determine the motion performed in each data set (See motion classification code in

Appendix 3). We observe the motion outputs as character strings in the MATLAB command window similar to the figure below.

A screenshot of a MATLAB Command Window. The window title is "Command Window". The text inside shows the variable "motion" assigned a cell array of six strings: 'Circle', 'Waving', 'Figure 8', 'Clenching', 'Eating', and 'Pouring'. Each string is followed by a single quote character on the same line.

```
motion =  
    'Circle  '  
    'Waving  '  
    'Figure 8 '  
    'Clenching'  
    'Eating  '  
    'Pouring  '
```

Figure 26: Example of Motion Outputs Displayed to the User. This output represents a series of six data sets, where each data set corresponds to one of the six unique motions performed during our testing. In this case, all of the motions were correctly classified.

4.5 Summary of Results

Upon the conclusion of our testing, we had run a total of 144 two-dimensional tests, and 105 three-dimensional tests on a total of 25 test subjects. After constructing membership functions and logic rules for each of these data sets, we were able to correctly classify the individual motion performed in a specific data set from the larger pool of 249 total tests with 80% accuracy.

These results indicate that an individual who we have already tested would have an 80% chance of performing one of our six predefined motions that will be correctly classified by our logic system. While there is still potential to improve upon this number for individuals who have already been tested, the biggest potential remaining lies in the ability of our system and logic designs to accurately classify the unknown motions of individuals who have never been tested before and account for motion uncertainty. These topics will be discussed more thoroughly in our future works section (5.2).

Chapter 5: Discussion

5.1 Discussion of Results:

We have achieved a fairly high level of accuracy in activity recognition through intention sensing with our system design, but the ongoing project challenge will be quality activity assessment.

Comparing our average classification accuracy of 80% with the higher accuracy classification devices from our literature review, we are convinced that more research and experimentation must be conducted with our system design to improve the accuracy beyond 95% in order to validate further implementation with activity assessment and integration within a more comprehensive, clinical-level rehabilitation system [26]. For instance, the iterative learning framework for extremity and full body motion classification produced an “average accuracy of 81.07%...when using 80% of samples for training” [20]. This means that using a smaller fraction of the entire data collection, the researchers were still able to obtain a higher classification accuracy.

Moreover, the model accuracy for the ultrasonic transducer wristband device for motion recognition is comparatively high (86.0% - 89.4%) for experiments in which the wristband was removed and reused by test subjects [5]. The reasoning as to why our average classification accuracy falls slightly below these other experimental accuracies remains unclear and necessitates further optimization and manipulation of testing variables such as type of wearable material and other considerations examined in the “Future Works” section below.

5.2 Future Work:

Although the results of our research and testing were positive in regards to activity recognition, we acknowledge that there is huge potential for integration of our system design within activity motion assessment. Here, we propose various testing procedures and system improvements for future progression of our system design framework.

5.2.1 Continued Testing on a Wider Body of Subjects:

In order to further optimize our system design to reach clinical-level accuracies for activity recognition, further tests need to be conducted in assessing hand gestures using fuzzy logic. Even though we were able to gather hand motion data from 25 different people, our datasets were largely homogeneous because the individuals we tested all demonstrated reasonably healthy, natural motions and have never required any preceding clinical hand or extremity rehabilitation. With more test subjects exhibiting a more diverse range of extremity physiological characteristics, we can obtain more quantitative data and expand our system capabilities for case-specific applications of activity recognition and future assessment.

5.2.2 Incorporation of Machine Learning:

The fuzzy logic inference system implemented in our current project design is a passive system, meaning any new input data that does not fall within existing parametric classification requires additional manual changes within the fuzzy logic rules and membership functions. In an active fuzzy logic system, the decision making process transforms into an iterative learning framework that utilizes machine learning to automatically assess patient rehabilitation progress and generate new classification rules whenever incoming data does not meet existing analysis criteria [27]. Machine learning can be incorporated into our system design to allow the system to automatically process larger amounts of data and streamline the efficiency of data analysis [20].

5.2.3 Testing using Embedded Sensors:

In our benchmark activity recognition, we have shown how a sensor glove is an ideal rewearable system component for minimized sensor attachment that enables us to completely mitigate direct sensor-skin contact, while maintaining a fairly high classification accuracy for multiple test subjects. However, our current dexterous glove requires physically sewing the sensors on top of the glove material for close attachment, and over repeated usages there is a strong chance that the threading will become detached and that the sensors will lose their optimal contact with the glove.

A polymer-based glove with embedded sensors would provide quality transmission of data and reduce the risks of sensor detachment and compromised data acquisition. Researchers from

University of California, Berkeley and Lawrence Berkeley National Laboratory suggest both a PDMS tactile sensing glove as well as a PDMS wristband, which allow for a greater probability of uncompromised contact and data sensitivity because of the elastic and adhesive properties of PDMS [13]. One potential challenge to using PDMS for a manufacturable rehabilitation glove device material is that repeated usages can cause the PDMS to fracture.

As a secondary project experiment to evaluate the feasibility of a polymer-based glove within our system design, we embedded one flex sensor in between two thin layers of Ecoflex polymer material. To embed the sensor, we used a three-dimensional anatomical hand model as the framework to shape and mold the multiple layers of Ecoflex around the flex sensor and output wiring. One drawback to embedded sensors is that once the mold forms around the sensors and wiring, it becomes very difficult to adjust or access the hardware components without compromising or fracturing the mold itself. Ecoflex was chosen specifically because it is less prone to fracturing as well as its versatile elastic characteristics and ease of manufacturing.

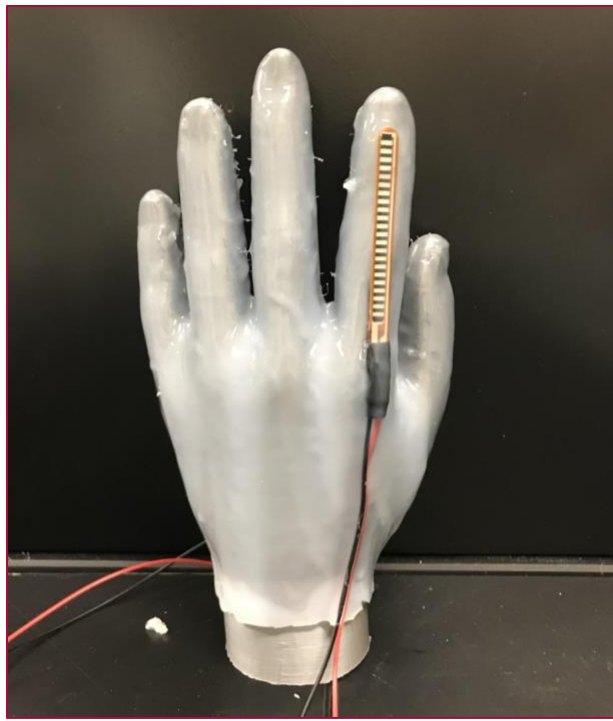


Figure 27: Ecoflex Embedded Sensors Glove. This image shows an Ecoflex glove containing a flex sensor embedded within the layers of the material poured over a hand mold.

5.2.4 Inclusion of a Wrist-bound Accelerometer:

Including a wrist-bound accelerometer, we can increase the possible degrees of freedom within our system design and allow for data acquisition of a broader range of extremity motions. Researchers in [24] organized simple hand motions into five encompassing categories for hand motion classification: static gestures, touching, stable grasps without external forces, simple shifts, and rotating an object in-hand. Furthermore, they categorized complex hand motions that extend beyond our senior design project's scope of 2-3 dimensional, task-specific motions because they involve dynamic movements of multiple fingers and wrist rotations as well. Incorporating a wrist-bound accelerometer would increase our system design capabilities to recognize and assess more extensive extremity motions, while preserving system accuracy.

5.2.5 Integration of Consumer Electronics and Smart Devices:

Our system design has widespread potential for future implementation that aligns with the increasing shift in the digitization of personal healthcare technologies, with the most interesting facet being the integration of sensors and data representation within smart devices and consumer electronics. The latest edition of the Apple Watch not only features an accelerometer to track motion, but also an ECG sensor that monitors heart rate [28]. Glucose sensor patches are also becoming prevalent, with active wireless communication between the patch and a phone or smart device that allows the user to easily read and understand the data [29].

Within modern rehabilitation, smart devices have great potential to not only serve as hubs for data transmission and processing, but to interface directly with the patient or user with diagnostic evaluation and rehabilitative instruction [5]. This is primarily because “providing feedback like visual information on smartphones is common and effective, especially for the systems intended for remote monitoring” [14].



Figure 28: Example of a Potential Application Interface for Future Research. This image depicts what we imagine the user interface might look like if our research was applied to use as a rehabilitation application for at home use.

5.2.6 Potential to Test for Tremor Detection:

Tremor output recognition and assessment would be applicable in both cyclic and static activities for patients recovering from stroke and Parkinson’s disease. Resting tremors can be recognized and assessed in a static environment, while moving tremor intensity for an individual patient may vary between different cyclic motions and would entail a much more dynamic feedback system for legitimate rehabilitation feedback [25]. In either case, tremor classification could be integrated easily within the fuzzy logic inference system to provide additional rehabilitation detection capabilities.

5.2.7 An At-home Rehabilitation Program:

An at-home device with patient interfacing and instruction would be extremely beneficial to patients undergoing rehabilitation. Researchers from the Adelante Rehabilitation Centre evaluated long term patient rehabilitation progress using modern accelerometer-based sensor technologies [23]. They also asked oral questions directly to the patients to gain a better understanding of how the patient subjectively perceived the status of their motor debilitating

conditions and subsequent progress in rehabilitation. This would be a crucial element for interfacing within an at home, patient specific rehabilitation device, yet it would not serve as a diagnostic replacement.

A future objective for our system design for at home usage would be to compare individual patient data on extremity motor function and improvements over time with data pooled from other patients that all follow an established or authenticated extremity rehabilitation program [5, 30]. The program parameters would need to be subcategorized based on the type of rehabilitation and motor debilitation specific to the patient in order to generate accurate, validated diagnostic feedback and updated rehabilitation protocol.

Chapter 6: Engineering Standards and Constraints

6.1 Ethical:

While our current research and experimental design to this point does not pose any major ethical concerns, the possibility of future work and applications of our research may bring about further ethical considerations. The potential for our experimental design to be streamlined into an at-home diagnostic tool could pose issues of misuse resulting in misdiagnosis. Since the patients using the device would not be professionally trained in the field of hand rehabilitation, it would be crucial to emphasize the use of our device as a diagnostic tool intended to aid professionals in evaluating ongoing rehabilitation progress.

6.2 Science, Technology, and Society:

In addition to serving as an inexpensive professional diagnostic tool, our system design has the potential to become a portable device that allows hand rehabilitation users to periodically monitor the functionality and health of their hand motions without requiring excessive routine checkups with their rehabilitation professional. This helps save users time and money that would typically be spent covering the cost of attending routine physical checkups and consultations. However, the implementation of a portable device to monitor rehabilitation progress and function does not substitute the need to continually engage with rehabilitation professionals for expert clinical diagnostic feedback and guidance. The same considerations apply for nonprofessional usage of initial diagnostics without previous professional consultation.

6.3 Civic Engagement:

If used for professional rehabilitation applications, our project system design may need approval remarks from the Council on Rehabilitation Education in the United States. Aside from this, rehabilitation doctors and professionals in the United States are required to gain accreditation from Healthcare Facilities Accreditation Program. For nonprofessional usage in an at home setting, our system requires would require approval from the U.S. Food and Drug Administration. Since future system design incorporation would lean towards a wearable rehabilitation device, it is noninvasive and poses no health concerns aside from the inevitable

sanitary issues that come with using rewearable glove, which is addressed in the section “Health and Safety”. Additionally, if our system design has any chance of becoming profitable within a wearable glove technology or smart device, we would need to file a patent for innovation and intellectual property in the United States and any other country requiring patent registration for similar biologically noninvasive rehabilitation products.

6.4 Economic:

Since our project offers a system design rather than a physical manufacturable product, it allows future researchers significant flexibility in selecting and purchasing the system hardware components to meet specific hand rehabilitation needs requiring motion classification. The system hardware components, including the wearable sensors and Arduino board, are inexpensive and easily accessible on online commercial markets. The system software components, Matlab and Processing, are also relatively affordable and accessible within medical rehabilitation fields. In this regard, our system design is incredibly frugal and economically versatile for researchers who either wish to replicate or expand on our system design for rehabilitation and extremity sensing applications.

With the potential for creating a manufactured wearable device product for at home rehabilitation, the economics of manufacturing are similarly inexpensive in regards to purchasing existing inexpensive market accelerometers and flex sensors. The more notable expenses would be the hardware assembly and sensor integration within the reusable device. For example, if the future design were to embed sensors within an Ecoflex or PDMS glove, the cost of constructing the machinery for production as well as the time required to successfully embed the sensors would be two economic considerations in order to enable large scale manufacturing.

6.5 Health and Safety:

Reusability of the woven glove with our system design is one potential health concern because repeated use by multiple users may leave residual germs if users do not undergo proper sanitation procedures both before and after wearing the glove. To account for this health concern, users also have the option of wearing a disposable latex or nitrile glove to prevent direct contact between skin tissue and the reusable woven glove containing our array of sensors. In our

project, we ran approximately 250 different motion tests with a total of 25 individual test subjects, so one way we considered sustainable and healthy reusage was to have users apply hand sanitizer to their hands both before and after wearing the glove.

6.6 Manufacturability:

As mentioned in the “Science, Technology, and Society” section, our system design has the potential to become a portable device that allows hand rehabilitation users to periodically monitor the functionality and health of their hand motions without requiring excessive, routine checkups with their rehabilitation professional. The hardware components of a portable device integrated into our system can be manufactured including the woven glove with imbedded sensors, soldered wires, and connection with an Arduino board, or other suitable microcontroller. The manufacturing costs would be moderately inexpensive, with the most time consuming portions of manufacturing being sewing or embedding the sensors to the dexterous glove and soldering the connector cables. The only potential issues that may arise from manufacturing is the bulkiness of the Arduino board. If our system design reached the manufacturing stage, we would likely integrate the sensors with a Bluetooth technology to wirelessly communicate with the chosen microcontroller. This would not only eliminate the need for a majority of wiring currently used with the glove, but would also make the device more simplistic and user-friendly.

6.7 Usability:

At its current stage, our system design is intended for professional and research usage. The physical hardware components of our testing apparatus are clearly defined for professional users to purchase and assemble. Our protocol contains procedures on how to use the software in Matlab and Processing as well as instructions to modify current fuzzy logic parameters and rules for user specific motion classification. With all of the given information, informed users with a background understanding in the field should be able to successfully use our designs and replicate our results.

6.8 Sustainability:

Our system is especially sustainable in the acquisition of hardware materials because many professional medical industries and rehabilitation centers have access to both physical sensors, connector cables and Arduino boards. These hardware components can also be purchased and reused for a variety of other medical industry and technologically related applications. The main relevant issue of sustainability with our project is in the reusability of the woven glove with embedded sensors because of the residual germ health concerns with repeated usage by multiple patients, so wearing disposable latex or nitrile gloves or apply hand sanitizer are useful considerations to sustain the longevity of a wearable device with embedded sensors.

6.9 Environmental Impact:

As mentioned in the previous section on sustainability, the most relevant aspect of sustainability and likewise the environmental impact of our system design is in the reusability of the woven glove with embedded sensors. Disposable latex or nitrile gloves were used in the beginning stages of our testing, but we eventually moved to a reusable dexterous glove. In doing so, we have limited the environmental impact of our system by ensuring that all of the components can be used for long durations, over a large number of tests without needing replacement.

Appendices

Appendix 1: Arduino Code

Calibration Code:

```
Calibration_Code_2 § ADXL335.cpp ADXL335.h
#include "ADXL335.h"
int zero_x;
int zero_y;
int zero_z;
int max_x,max_y,max_z;//when 1g
float sensitivity;
ADXL335 accelerometer;
void setup()
{
  Serial.begin(9600);
  accelerometer.begin();
  int x,y,z;
  for(int i = 0; i < 20; i ++){accelerometer.getXYZ(&x,&y,&z);
  Serial.println("The calibration starts: ");
  Serial.println("First, make sure that Z-axis direction is straight up"); //isolation of the x and y axes to determine zeros
  Serial.println("please type any character if you are ready");
  while(Serial.available() == 0);
  delay(100);
  while(Serial.available() > 0)Serial.read();
  calibrate(&x,&y,&z);
  zero_x = x;
  zero_y = y;
  max_z = z;
  Serial.print("max_z =");
  Serial.println(max_z);
  Serial.println("zero_x =");
  Serial.println(zero_x);
  Serial.println("zero_y =");
  Serial.println(zero_y);
  Serial.println("Second, make sure that X-axis direction is straight up"); //isolation of the y and z axes to determine z zero
  Serial.println("please type any character again if you are ready");
  while(Serial.available() == 0);

  delay(100);
  while(Serial.available() > 0)Serial.read();
  calibrate(&x,&y,&z);
  zero_z = z;
  Serial.print("zero_z =");
  Serial.println(zero_z);
  float zero_xv,zero_yv,zero_zv;
  zero_xv = (float)zero_x*ADC_REF/ADC_AMPLITUDE;
  zero_yv = (float)zero_y*ADC_REF/ADC_AMPLITUDE;
  zero_zv = (float)zero_z*ADC_REF/ADC_AMPLITUDE;
  sensitivity =abs((float)(max_x-zero_x)*ADC_REF/ADC_AMPLITUDE); //determines sensitivity of the accelerometer and prints the final zeros
  Serial.print("ZERO_X = ");
  Serial.println(zero_xv);
  Serial.print("ZERO_Y = ");
  Serial.println(zero_yv);
  Serial.print("ZERO_Z = ");
  Serial.println(zero_zv);
  Serial.println("SENSITIVITY = ");
  Serial.println(sensitivity,2);
  Serial.println("please modified the macro definitions with these results in ADXL335.h");
  //take the values and modify the constants in the header file to obtain a more accurate calibration; may need to be run several
  //times until you get consistent results
}
void loop()
{
}
void calibrate(int* _x,int* _y,int* _z)
{
  int x,y,z;
  int sum_x,sum_y,sum_z;

  accelerometer.getXYZ(&x,&y,&z);
  float ax,ay,az;
  accelerometer.getAcceleration(&ax,&ay,&az); //references function in the C++ file in the ADXL335 library
  ax = abs(ax);
  ay = abs(ay);
  az = abs(az);
  Serial.print("ax = ");
  Serial.println(ax);
  Serial.print("ay = ");
  Serial.println(ay);
  Serial.print("az = ");
  Serial.println(az);
  Serial.println(ax);

  *_x = x;
  *_y = y;
  *_z = z;
}
}
```

Data Acquisition Code:

Glove_Testing

```
//establish global variables for input pins and constant values
const int FLEX_PIN_L = A3;
const int FLEX_PIN_R = A4;
const float VCC = 3.3;
const float R_DIV = 10120.0;

const float STRAIGHT_RESISTANCE_L = 61992.0;
const float BEND_RESISTANCE_L = 159270.0;
const float STRAIGHT_RESISTANCE_R = 55900.0;
const float BEND_RESISTANCE_R = 104573.0;

void setup() {
  //establish the location of the input pins
  pinMode(FLEX_PIN_L, INPUT);
  pinMode(FLEX_PIN_R, INPUT);
  pinMode(14, INPUT);
  pinMode(15, INPUT);
  pinMode(16, INPUT);

  Serial.begin(9600); //serial communication begin
  delay(10);
}

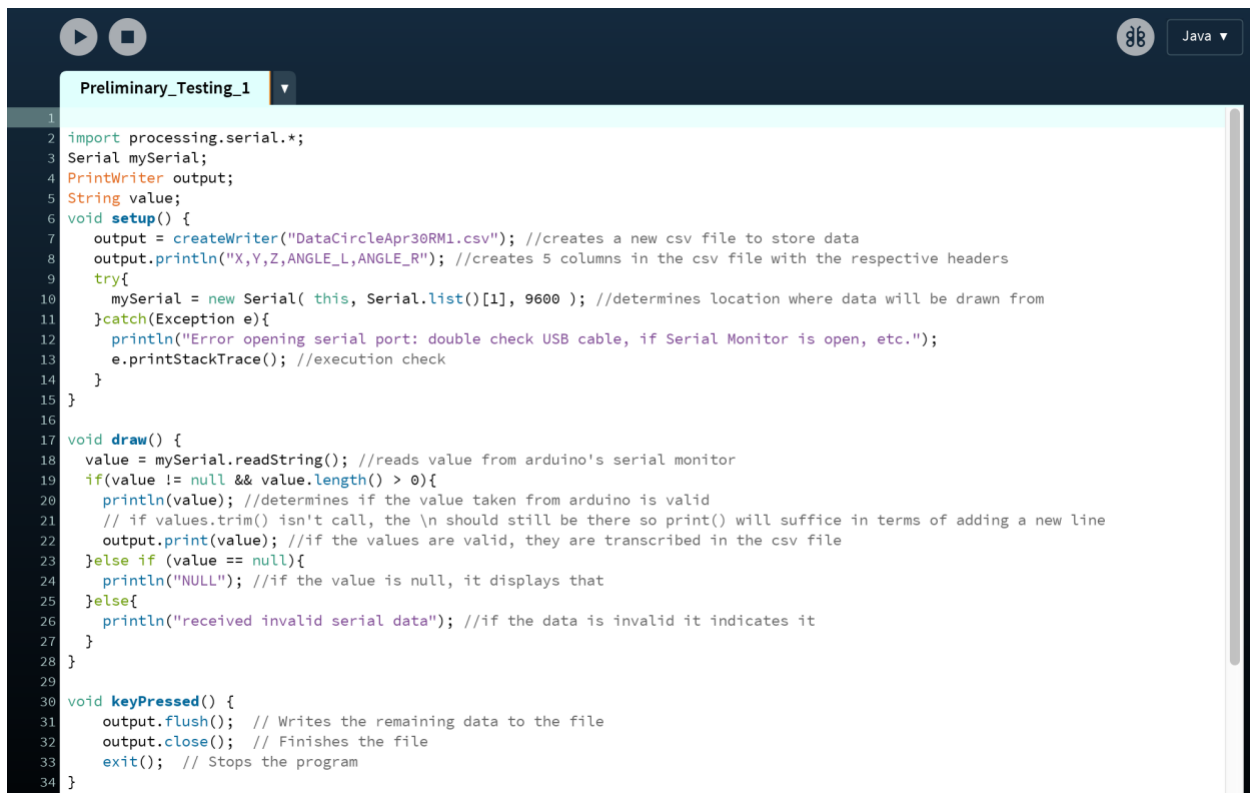
void loop()
{
  int flexADC_L = analogRead(FLEX_PIN_L);
  float flexV_L = flexADC_L * VCC / 1023.0;
  float flexR_L = R_DIV * (VCC / flexV_L);
  float angle_L = map(flexR_L, STRAIGHT_RESISTANCE_L, BEND_RESISTANCE_L, 0, 90.0); //converts resistance value to an angle
  int flexADC_R = analogRead(FLEX_PIN_R);
  float flexV_R = flexADC_R * VCC / 1023.0;

  int flexADC_R = analogRead(FLEX_PIN_R);
  float flexV_R = flexADC_R * VCC / 1023.0;
  float flexR_R = R_DIV * (VCC / flexV_R);
  float angle_R = map(flexR_R, STRAIGHT_RESISTANCE_R, BEND_RESISTANCE_R, 0, 90.0); //converts resistance value to an angle

  //print out the values of acceleration and bend angle to the serial monitor
  Serial.print(analogRead(14));
  Serial.print(",");
  Serial.print(analogRead(15));
  Serial.print(",");
  Serial.print(analogRead(16));
  Serial.print(",");
  Serial.print(angle_L);
  Serial.print(",");
  Serial.print(angle_R);
  Serial.println(",");
}
```

Appendix 2: Processing Code

Processing Code for Data Acquisition



```
1
2 import processing.serial.*;
3 Serial mySerial;
4 PrintWriter output;
5 String value;
6 void setup() {
7   output = createWriter("DataCircleApr30RM1.csv"); //creates a new csv file to store data
8   output.println("X,Y,Z,ANGLE_L,ANGLE_R"); //creates 5 columns in the csv file with the respective headers
9   try{
10    mySerial = new Serial( this, Serial.list()[1], 9600 ); //determines location where data will be drawn from
11  }catch(Exception e){
12    println("Error opening serial port: double check USB cable, if Serial Monitor is open, etc.");
13    e.printStackTrace(); //execution check
14  }
15 }
16
17 void draw() {
18   value = mySerial.readString(); //reads value from arduino's serial monitor
19   if(value != null && value.length() > 0){
20     println(value); //determines if the value taken from arduino is valid
21     // if values.trim() isn't call, the \n should still be there so print() will suffice in terms of adding a new line
22     output.print(value); //if the values are valid, they are transcribed in the csv file
23   }else if (value == null){
24     println("NULL"); //if the value is null, it displays that
25   }else{
26     println("received invalid serial data"); //if the data is invalid it indicates it
27   }
28 }
29
30 void keyPressed() {
31   output.flush(); // Writes the remaining data to the file
32   output.close(); // Finishes the file
33   exit(); // Stops the program
34 }
```

Appendix 3: MATLAB Code

Data Optimization -- Lines 411-430

Motion Classification -- Lines 334-390

```
329 - inputs_total = [];
330 - motion = [];
331
332
333 - for i = 0:(length(filename)-1) %this loop begins the reading of the file names
334 -     if i == (length(filename)-1) %if all of the files are finished being read, we feed the data inputs into the fuzzy logic designer
335 -         fis = readfis('LUT');
336 -         output = [];
337 -         %defining the string outputs for motion classification
338 -         circ = 'Circle ';
339 -         wav = 'Waving ';
340 -         unk = 'Unknown ';
341 -         fig8 = 'Figure 8 ';
342 -         clen = 'Clenching';
343 -         eat = 'Eating ';
344 -         pour = 'Pouring ';
345 -         no_rule = 'No Rule '; %if none of the motions are classified, it tells us there is no rule written for that dataset
346 -         motion = [];
347
348 -         %these if statements ensure that we run through the fuzzy logic
349 -         %system the correct number of times
350 -         if min(size(inputs_total)) < 8
351 -             for i = 0:(min(size(inputs_total))-1)
352 -                 if i == (min(size(inputs_total)))
353 -                     break
354 -                 end
355 -                 data_file = inputs_total(i+1,:);
356 -                 new_output = evalfis(fis, data_file);
357 -                 output = [output, new_output]; %concatenates a matrix of outputs from the fuzzy logic system
358 -             end
359 -         else
360 -             for i = 0:(max(size(inputs_total))-1)
361 -                 if i == (max(size(inputs_total)))
362 -                     break
363 -                 end
364 -                 data_file = inputs_total(i+1,:);
365 -                 new_output = evalfis(fis, data_file);
366 -                 output = [output, new_output]; %concatenates a matrix of outputs from the fuzzy logic system
367 -             end
368 -         end
369
370 -         for i = 0:(length(output)-1) %determines which motions were performed based on the interger output from fuzzy logic
371 -             if output(i+1) > 0.9 && output(i+1) < 1.1
372 -                 motion = [motion; circ];
373 -             elseif output(i+1) > 1.9 && output(i+1) < 2.1
374 -                 motion = [motion; wav];
375 -             elseif output(i+1) > 2.9 && output(i+1) < 3.1
376 -                 motion = [motion; fig8];
377 -             elseif output(i+1) > 3.9 && output(i+1) < 4.1
378 -                 motion = [motion; eat];
379 -             elseif output(i+1) > 7.9 && output(i+1) < 8.1
380 -                 motion = [motion; clen];
381 -             elseif output(i+1) > 8.9 && output(i+1) < 9.1
382 -                 motion = [motion; pour];
383 -             elseif output(i+1) > 9.9 && output(i+1) < 10.1
384 -                 motion = [motion; no_rule];
385 -             else
386 -                 motion = [motion; unk]; %an unknown motion means a single dataset was classified as two different motions
387 -             end
388 -         end
389 -         motion %displays the motions in the command window
390 -         break
391 -         %this if statement checks for the sequence 'csv' in our data files,
392 -         %signifying the end of a data set
393 -     elseif filenames(i+2) == 'v' && filenames(i+1) == 's' && filenames(i) == 'c'
394 -         file = [file, filenames(i+1), filenames(i+2)];
395 -         data = csvread(file,1,0);
396 -         size_array = size(data);
```



```

398 - x = data(:,1); %data is separated into individual axes
399 - y = data(:,2);
400 - z = data(:,3);
401
402 - if size_array(2) > 5 || size_array(2) == 5 %if there is flex sensor data, it also separated
403 -     angle_L = data(:,4);
404 -     angle_R = data(:,5);
405 - end
406
407 - angle_R = round(angle_R);
408 - angle_L = round(angle_L);
409
410 - lengths = [length(x),length(y),length(z),length(angle_L),length(angle_R)];
411 - %this for loop executes our data optimization
412 - %if any of the conditions are met, that row is marked as an outlier
413 - for i = 0:(min(lengths)-1)
414 -     if i == (length(x))
415 -         break
416 -     elseif x(i+1) > 500 || x(i+1) < 200 || y(i+1) > 500 || y(i+1) < 200 || z(i+1) > 500 || z(i+1) < 200 || angle_L(i+1) > 110
417 -         outlier = i+1;
418 -         error = [error, outlier];
419 -     end
420 - end
421 - error_rev = fliplr(error);
422 - %this loop removes the rows with errors in all of the inputs from the bottom to the top
423 - for i = 0:(length(error_rev)-1)
424 -     row = error_rev(i+1);
425 -     x((row),:) = [];
426 -     y((row),:) = [];
427 -     z((row),:) = [];
428 -     angle_L((row),:) = [];
429 -     angle_R((row),:) = [];
430 - end
431
432 - time = 25*(1:length(y)).

434 - x_baseline = mean(x(1:3)); %establish a '0' for the dataset
435 - y_baseline = mean(y(1:3));
436 - z_baseline = mean(z(1:3));
437
438 - x_real = (x-x_baseline)/9.8; %determine acceleration by dividing by gravity
439 - y_real = (y-y_baseline)/9.8;
440 - z_real = (z-z_baseline)/9.8;
441
442 - data_real = [x_real,y_real,z_real];
443 - angles = [angle_L,angle_R];
444
445 - %this figure graphs our raw acceleration and flex sensor data
446 - figure
447 - sgtitle(file(1:(length(file)-4)))
448
449 -     subplot(2,1,1)
450 -     plot(time,data_real), title({'Acceleration in X,Y,Z Axes vs. Time'},'FontSize',18);...
451 -     xlabel({'Time (ms)'},'FontSize',18), ylabel({'Acceleration (m/s^2)'},'FontSize',18);...
452 -     legend({'x','y','z'},'FontSize',10), ylim auto, xlim auto;
453 -     subplot(2,1,2)
454 -     plot(time,angles), title({'Angles of Finger Bending vs. Time'}, 'FontSize', 18);...
455 -     xlabel({'Time (ms)'}, 'FontSize',18), ylabel({'Angle'},'FontSize',18);...
456 -     legend({'angle_L','angle_R'},'FontSize',10), ylim ([0,110]), xlim auto;
457
458 - %this statement runs fourier transforms on our data if all inputs
459 - %are the same length
460 - if length(x_real) == length(y_real)
461 -     N = length(x_real);
462 -     fs = 5500; %(ms)
463 -     dt = 1/fs;
464 -     sl = N*dt;
465 -     fr = 1/sl;
466 -     n = 1:N;
467 -     freq = 0:fr:fs;

```

```

469 -         xf_x = fft(x_real,N);
470 -         mag_x = abs(xf_x); %magnitude of xf_x
471 -
472 -         xf_y = fft(y_real,N);
473 -         mag_y = abs(xf_y); %magnitude of xf_y
474 -
475 -         xf_z = fft(z_real,N);
476 -         mag_z = abs(xf_z); %magnitude of xf_z
477 -
478 -         xf_angle_L = fft(angle_L,N);
479 -         mag_angle_L = abs(xf_angle_L); %magnitude of xf_angle_L
480 -
481 -         xf_angle_R = fft(angle_R,N);
482 -         mag_angle_R = abs(xf_angle_R); %magnitude of xf_angle_R
483 -
484 -         %determines the value of the fundamental frequency and the sample that produced it
485 -         [ff_value_x, ff_location_x] = max(mag_x(3:round((length(mag_x)/2))));
486 -         ff_x = freq(ff_location_x+2); %in Hz
487 -
488 -         [ff_value_y, ff_location_y] = max(mag_y(3:round((length(mag_y)/2))));
489 -         ff_y = freq(ff_location_y+2); %in Hz
490 -
491 -         [ff_value_z, ff_location_z] = max(mag_z(3:round((length(mag_z)/2))));
492 -         ff_z = freq(ff_location_z+2); %in Hz
493 -
494 -         [ff_value_angle_L, ff_location_angle_L] = max(mag_angle_L(3:round((length(mag_angle_L)/2))));
495 -         ff_angle_L = freq(ff_location_angle_L+2);
496 -
497 -         [ff_value_angle_R, ff_location_angle_R] = max(mag_angle_R(3:round((length(mag_angle_R)/2))));
498 -         ff_angle_R = freq(ff_location_angle_R+2);
499 -
500 -         ff_values = [ff_value_x, ff_value_y, ff_value_z, ff_value_angle_L, ff_value_angle_R];
501 -
502 -
503 -         %graphs the fourier transforms of the dataset
504 -         figure
505 -         sgtitle(file(1:(length(file)-4)))
506 -
507 -         subplot(2,1,1)
508 -         plot(freq(4:round(N/2+1)),mag_x(4:round(N/2+1)),'.-')
509 -         hold on
510 -         plot(freq(4:round(N/2+1)),mag_y(4:round(N/2+1)),'.-')
511 -         hold on
512 -         plot(freq(4:round(N/2+1)),mag_z(4:round(N/2+1)),'.-')
513 -         grid
514 -         xlabel({'Frequency (Hz)'},'FontSize',18)
515 -         ylabel({'Magnitude'},'FontSize',18)
516 -         title({'Fourier Spectrum of Acceleration'},'FontSize',20)
517 -         legend({'x','y','z'],'FontSize',16)
518 -         xlim ([0,1000])
519 -         ylim auto
520 -
521 -         subplot(2,1,2)
522 -         plot(freq(5:round(N/2+1)),mag_angle_L(5:round(N/2+1)),'.-')
523 -         hold on
524 -         plot(freq(5:round(N/2+1)),mag_angle_R(5:round(N/2+1)),'.-')
525 -         grid
526 -         xlabel({'Frequency (Hz)'},'FontSize',18)
527 -         ylabel({'Magnitude'},'FontSize',18)
528 -         title({'Fourier Spectrum of Flex Sensors'},'FontSize',20)
529 -         legend({'angleL','angleR'],'FontSize',16)
530 -         xlim ([0,1000])
531 -         ylim auto
532 -
533 -         ratio_frequency = [ff_x/ff_y, ff_x/ff_z, ff_y/ff_z];
534 -         ratio_amplitude = [ff_value_x/ff_value_y, ff_value_x/ff_value_z, ff_value_y/ff_value_z];
535 -         angle_magnitude = [ff_value_angle_L, ff_value_angle_R];

```

```

%concatenates all of the inputs for the data sets to be
%analyzed so that it can be run through the fuzzy logic system
inputs = [ratio_freq, ratio_amplitude, angle_magnitude];
inputs_total = [inputs_total; inputs];

%displays the dataset, as well as all of the inputs in written
%form in the command window
fprintf('For the dataset %s \n\n', file(1:(length(file)-4)));
fprintf('The maximum amplitude for acceleration on the x-axis is %3.1f, and occurs at a frequency of %3.1f Hz. \n', ff_value_x, ff_x);
fprintf('The maximum amplitude for acceleration on the y-axis is %3.1f, and occurs at a frequency of %3.1f Hz. \n', ff_value_y, ff_y);
fprintf('The maximum amplitude for acceleration on the z-axis is %3.1f, and occurs at a frequency of %3.1f Hz. \n', ff_value_z, ff_z);
fprintf('The maximum amplitude for angle_L is %3.1f, and occurs at a frequency of %3.1f Hz. \n', ff_value_angle_L, ff_angle_L);
fprintf('The maximum amplitude for angle_R is %3.1f, and occurs at a frequency of %3.1f Hz. \n\n', ff_value_angle_R, ff_angle_R);
fprintf('The ratio of Fourier Spectrum max amplitude frequencies: \n x:y = %3.1f \n x:z = %3.1f \n y:z = %3.1f \n\n', ratio_freq);
fprintf('The ratio of Fourier Spectrum max amplitudes: \n x:y = %3.1f \n x:z = %3.1f \n y:z = %3.1f \n\n', ratio_amplitude);
end
%once the file is finished being analyzed, the file and error
%matrices are reset to begin on a new file
file = [];
error = [];
elseif file == 'v'
file = (filenames(i+1));
else
file = [file, filenames(i+1)]; %continues adding characters to the file array until an entire filename is present
end
end
end

```

Appendix 4: Fuzzy Logic Designer Code

System Setup

```
Glove_Prototype_Data_Analysis_2.m x LUT.fis x +
1 [System]
2 Name='LUT'
3 Type='sugeno'
4 Version=2.0
5 NumInputs=8
6 NumOutputs=1
7 NumRules=152
8 AndMethod='min'
9 OrMethod='max'
10 ImpMethod='prod'
11 AggMethod='sum'
12 DefuzzMethod='wtaver'
```

Defining Membership Functions for Inputs and Outputs

```
14 [Input1]
15 Name='X:Y-Frequency'
16 Range=[0 3]
17 NumMFs=3
18 MF1='0<XY<=0.5': 'trimf', [0 0.25 0.49]
19 MF2='0.5<XY<=1.5': 'trimf', [0.49 1 1.49]
20 MF3='1.5<XY<=2.5': 'trimf', [1.49 2 2.49]
21
22 [Input2]
23 Name='X:Z-Frequency'
24 Range=[0 15]
25 NumMFs=4
26 MF1='0<XZ<=0.5': 'trimf', [0 0.25 0.49]
27 MF2='0.5<XZ<=1.5': 'trimf', [0.49 1 1.49]
28 MF3='1.5<XZ<=2.5': 'trimf', [1.49 2 2.49]
29 MF4='XZ>2.5': 'trimf', [2.49 8 14.49]
30
31 [Input3]
32 Name='Y:Z-Frequency'
33 Range=[0 15]
34 NumMFs=4
35 MF1='0<YZ<=0.5': 'trimf', [0 0.25 0.49]
36 MF2='0.5<YZ<=1.5': 'trimf', [0.49 1 1.49]
37 MF3='1.5<YZ<=2.5': 'trimf', [1.49 2 2.49]
38 MF4='YZ>2.5': 'trimf', [2.5 8 14.49]
39
40 [Input4]
41 Name='X:Y-Amplitude'
42 Range=[0 26]
43 NumMFs=13
44 MF1='0<XY<=0.5': 'trimf', [0 0.25 0.5]
45 MF2='0.5<XY<=1.5': 'trimf', [0.5 1 1.5]
46 MF3='1.5<XY<=2.5': 'trimf', [1.5 2 2.5]
47 MF4='2.5<XY<=3.5': 'trimf', [2.5 3 3.5]
48 MF5='3.5<XY<=4.5': 'trimf', [3.5 4 4.5]
49 MF6='4.5<XY<=5.5': 'trimf', [4.5 5 5.5]
50 MF7='5.5<XY<=6.5': 'trimf', [5.5 6 6.5]
51 MF8='6.5<XY<=7.5': 'trimf', [6.5 7 7.5]
52 MF9='7.5<XY<=8.5': 'trimf', [7.5 8 8.5]
53 MF10='8.5<XY<=9.5': 'trimf', [8.5 9 9.5]
54 MF11='9.5<XY<=10.5': 'trimf', [9.5 10 10.5]
55 MF12='10.5<XY<=13.5': 'trimf', [10.5 12 13.5]
56 MF13='13.5<XY<=13.5': 'trimf', [13.5 19.5 25.5]
```

```

58 [Input5]
59 Name='X:Z-Amplitude'
60 Range=[0 24]
61 NumMFs=10
62 MF1='0<XZ<=2': 'trimf', [0 1 2]
63 MF2='2<XZ<=4': 'trimf', [2 3 4]
64 MF3='4<XZ<=6': 'trimf', [4 5 6]
65 MF4='6<XZ<=8': 'trimf', [6 7 8]
66 MF5='8<XZ<=10': 'trimf', [8 9 10]
67 MF6='10<XZ<=12': 'trimf', [10 11 12]
68 MF7='12<XZ<=14': 'trimf', [12 13 14]
69 MF8='14<XZ<=16': 'trimf', [14 15 16]
70 MF9='16<XZ<=18': 'trimf', [16 17 18]
71 MF10='18<XZ<=24': 'trimf', [18 21 24]
72
73 [Input6]
74 Name='Y:Z-Amplitude'
75 Range=[0 12]
76 NumMFs=6
77 MF1='0<YZ<=2': 'trimf', [0 1 2]
78 MF2='2<YZ<=4': 'trimf', [2 3 4]
79 MF3='4<YZ<=6': 'trimf', [4 5 6]
80 MF4='6<YZ<=8': 'trimf', [6 7 8]
81 MF5='8<YZ<=10': 'trimf', [8 9 10]
82 MF6='10<YZ<=12': 'trimf', [10 11 12]
83
84 [Input7]
85 Name='Angle_L-Amplitude'
86 Range=[0 11000]
87 NumMFs=5
88 MF1='0<Angle_L<50': 'trimf', [0 25 50]
89 MF2='50<Angle_L<150': 'trimf', [50 100 150]
90 MF3='150<Angle_L<400': 'trimf', [150 275 400]
91 MF4='400<Angle_L<1000': 'trimf', [400 700 1000]
92 MF5='1000<Angle_L<11000': 'trimf', [1000 6000 11000]
93
94 [Input8]
95 Name='Angle_R-Amplitude'
96 Range=[0 11000]
97 NumMFs=5
98 MF1='0<Angle_R<50': 'trimf', [0 25 50]
99 MF2='50<Angle_R<150': 'trimf', [50 100 150]
100 MF3='150<Angle_R<400': 'trimf', [150 275 400]
101 MF4='400<Angle_R<1000': 'trimf', [400 700 1000]
102 MF5='1000<Angle_R<11000': 'trimf', [1000 6000 11000]
103
104 [Output1]
105 Name='Motion'
106 Range=[0 20]
107 NumMFs=9
108 MF1='Circle': 'constant', [1]
109 MF2='Waving': 'constant', [2]
110 MF3='Figure8': 'constant', [3]
111 MF4='Eating': 'constant', [4]
112 MF5='5': 'constant', [5]
113 MF6='6': 'constant', [6]
114 MF7='7': 'constant', [7]
115 MF8='Clenching': 'constant', [8]
116 MF9='Pouring': 'constant', [9]

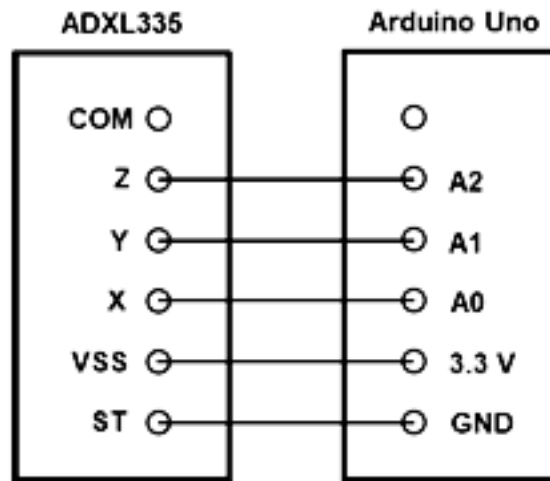
```

Establishing Rules for the Fuzzy Logic System

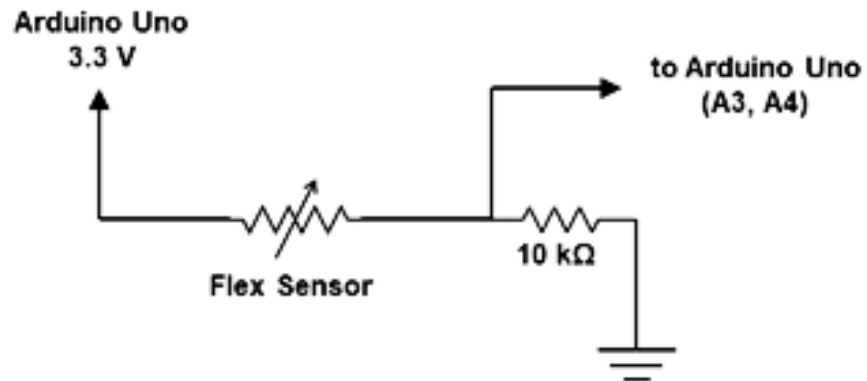
118	[Rules]	190	2 2 1 3 4 2 1 1, 3 (1) : 1
119	2 2 2 2 3 3 0 0, 1 (1) : 1	191	2 2 2 1 1 1 0 0, 4 (1) : 1
120	2 2 2 2 2 2 0 0, 1 (1) : 1	192	2 2 2 2 1 1 0 0, 4 (1) : 1
121	2 2 2 2 2 3 0 0, 1 (1) : 1	193	2 2 2 2 1 1 4 4, 4 (1) : 1
122	2 2 2 2 4 3 0 0, 1 (1) : 1	194	3 3 2 1 1 1 1 3, 4 (1) : 1
123	2 2 2 2 2 2 1 1, 1 (1) : 1	195	2 2 2 2 1 1 2 3, 4 (1) : 1
124	2 2 2 2 2 3 1 1, 1 (1) : 1	196	2 2 2 2 1 1 2 2, 4 (1) : 1
125	2 2 2 2 4 3 1 1, 1 (1) : 1	197	2 1 1 2 1 2 3 4, 4 (1) : 1
126	2 2 2 2 3 3 1 1, 1 (1) : 1	198	2 1 1 2 2 2 4 4, 4 (1) : 1
127	2 2 2 2 2 1 1 1, 1 (1) : 1	199	3 3 2 3 1 1 4 3, 4 (1) : 1
128	2 2 2 2 1 2 1 1, 1 (1) : 1	200	2 3 3 3 1 1 4 4, 4 (1) : 1
129	2 2 2 2 4 4 1 1, 1 (1) : 1	201	2 2 2 2 1 1 3 3, 4 (1) : 1
130	2 2 2 2 2 2 1 1, 1 (1) : 1	202	2 2 2 2 1 1 3 2, 4 (1) : 1
131	2 2 2 2 4 4 1 2, 1 (1) : 1	203	2 2 2 2 1 2 3 3, 4 (1) : 1
132	2 2 2 2 3 3 1 1, 1 (1) : 1	204	3 3 2 1 1 1 4 4, 4 (1) : 1
133	2 2 2 2 3 2 1 1, 1 (1) : 1	205	2 2 2 1 1 1 3 4, 4 (1) : 1
134	2 2 2 3 2 1 0 0, 2 (1) : 1	206	2 2 2 1 1 2 3 4, 4 (1) : 1
135	2 2 2 3 2 2 0 0, 2 (1) : 1	207	2 2 2 1 2 2 4 4, 4 (1) : 1
136	2 2 2 4 2 1 0 0, 2 (1) : 1	208	0 0 0 0 0 0 5 5, 8 (1) : 1
137	2 2 2 4 1 1 0 0, 2 (1) : 1	209	2 1 1 2 1 2 0 0, 8 (1) : 1
138	2 2 2 7 6 1 0 0, 2 (1) : 1	210	2 1 1 1 1 2 0 0, 8 (1) : 1
139	2 2 2 6 8 2 0 0, 2 (1) : 1	211	2 1 1 2 1 1 0 0, 8 (1) : 1
140	2 2 2 6 3 1 0 0, 2 (1) : 1	212	2 1 1 2 3 3 0 0, 8 (1) : 1
141	2 2 1 7 6 1 0 0, 2 (1) : 1	213	2 1 1 2 2 2 0 0, 8 (1) : 1
142	2 1 2 7 6 1 0 0, 2 (1) : 1	214	1 1 2 1 1 2 5 5, 8 (1) : 1
143	2 1 1 7 6 1 0 0, 2 (1) : 1	215	1 1 2 1 1 1 5 5, 8 (1) : 1
144	2 2 2 6 5 1 0 0, 2 (1) : 1	216	3 2 1 2 1 1 5 5, 8 (1) : 1
145	2 2 2 3 4 2 0 0, 2 (1) : 1	217	2 1 2 2 1 1 5 5, 8 (1) : 1
146	2 2 2 13 2 1 0 0, 2 (1) : 1	218	2 1 1 1 1 1 5 5, 8 (1) : 1
147	2 2 2 12 2 1 0 0, 2 (1) : 1	219	2 2 2 1 1 1 5 5, 8 (1) : 1
148	2 2 2 5 2 1 1 1, 2 (1) : 1	220	2 2 2 1 2 4 5 5, 8 (1) : 1
149	2 2 2 8 2 1 2 1, 2 (1) : 1	221	2 2 2 1 1 2 5 5, 8 (1) : 1
150	2 2 2 5 2 1 2 2, 2 (1) : 1	222	1 1 2 2 1 1 5 5, 8 (1) : 1
151	2 2 2 6 2 1 2 2, 2 (1) : 1	223	2 2 2 2 2 2 5 5, 8 (1) : 1
152	2 2 2 5 2 1 1 3, 2 (1) : 1	224	2 2 2 1 1 3 5 5, 8 (1) : 1
153	2 2 2 5 1 1 1 2, 2 (1) : 1	225	1 1 2 1 1 3 5 5, 8 (1) : 1
154	2 2 2 5 3 1 1 2, 2 (1) : 1	226	2 2 2 4 4 1 0 0, 9 (1) : 1
155	2 2 2 4 2 1 2 2, 2 (1) : 1	227	2 2 2 8 3 1 0 0, 9 (1) : 1
156	1 1 1 5 2 1 2 3, 2 (1) : 1	228	2 1 1 9 3 1 0 0, 9 (1) : 1
157	3 3 2 3 4 3 0 0, 3 (1) : 1	229	2 2 3 9 7 1 0 0, 9 (1) : 1
158	3 3 2 3 3 2 0 0, 3 (1) : 1	230	2 2 3 11 7 1 0 0, 9 (1) : 1
159	3 3 2 3 2 1 0 0, 3 (1) : 1	231	3 2 2 11 6 1 0 0, 9 (1) : 1
160	3 3 2 2 3 3 0 0, 3 (1) : 1	232	2 3 3 4 6 2 1 2, 9 (1) : 1
161	3 3 2 3 4 2 0 0, 3 (1) : 1	233	2 2 2 4 4 2 2 1, 9 (1) : 1
162	3 3 2 2 4 3 0 0, 3 (1) : 1	234	2 2 3 6 2 1 2 4, 9 (1) : 1
163	2 2 2 3 6 3 0 0, 3 (1) : 1	235	2 2 2 4 2 1 2 3, 9 (1) : 1
164	2 2 2 4 7 3 0 0, 3 (1) : 1	236	2 2 2 3 3 2 1 2, 9 (1) : 1
165	2 2 2 3 5 3 0 0, 3 (1) : 1	237	2 2 2 3 2 1 1 1, 9 (1) : 1
166	3 3 2 3 4 3 0 0, 3 (1) : 1	238	2 2 2 4 5 2 2 2, 9 (1) : 1
167	3 3 2 3 5 3 0 0, 3 (1) : 1	239	2 2 2 3 5 2 2 2, 9 (1) : 1
168	3 3 2 3 4 2 0 0, 3 (1) : 1	240	2 2 2 5 6 2 2 2, 9 (1) : 1
169	2 2 2 4 3 1 0 0, 3 (1) : 1	241	2 2 2 4 9 3 2 2, 9 (1) : 1
170	3 2 2 4 2 1 0 0, 3 (1) : 1	242	2 2 2 4 10 4 2 2, 9 (1) : 1
171	2 2 3 4 2 1 0 0, 3 (1) : 1	243	2 2 2 3 2 1 2 3, 9 (1) : 1
172	2 2 3 3 2 1 0 0, 3 (1) : 1	244	2 2 2 3 4 2 2 4, 9 (1) : 1
173	2 2 3 3 3 1 0 0, 3 (1) : 1	245	2 2 2 4 3 2 2 1, 9 (1) : 1
174	3 2 2 4 3 1 0 0, 3 (1) : 1	246	2 2 2 3 4 2 3 2, 9 (1) : 1
175	3 2 2 4 3 2 0 0, 3 (1) : 1	247	2 2 2 3 3 2 3 4, 9 (1) : 1
176	3 3 2 4 3 2 0 0, 3 (1) : 1	248	2 2 2 3 9 5 1 3, 9 (1) : 1
177	2 2 3 4 3 1 0 0, 3 (1) : 1	249	2 2 2 3 7 4 3 3, 9 (1) : 1
178	2 2 2 3 3 1 0 0, 3 (1) : 1	250	2 2 2 3 10 6 2 3, 9 (1) : 1
179	3 2 1 4 5 2 1 1, 3 (1) : 1	251	2 2 2 3 6 3 1 3, 9 (1) : 1
180	2 2 2 3 3 2 1 2, 3 (1) : 1	252	2 2 2 5 3 1 2 2, 9 (1) : 1
181	3 2 2 3 4 2 2 2, 3 (1) : 1	253	2 2 2 9 2 1 2 2, 9 (1) : 1
182	3 2 1 3 3 2 2 3, 3 (1) : 1	254	2 2 2 4 6 2 2 2, 9 (1) : 1
183	3 2 2 3 4 2 1 1, 3 (1) : 1	255	2 2 2 4 9 3 2 2, 9 (1) : 1
184	3 2 1 3 5 3 1 1, 3 (1) : 1		
185	3 2 2 3 3 2 1 1, 3 (1) : 1		
186	3 2 1 3 2 1 1 1, 3 (1) : 1		
187	2 2 2 3 5 2 1 1, 3 (1) : 1		
188	3 2 1 3 3 1 1 1, 3 (1) : 1		
189	2 2 2 2 3 3 1 1, 3 (1) : 1		

Appendix 5: Circuit Diagrams

Accelerometer Circuitry Schematic



Flex Sensor Circuitry Schematic



Appendix 6: Component Data Sheets

Accelerometer Data Sheets

ADXL335

SPECIFICATIONS

T_A = 25°C, V_S = 3 V, C_X = C_Y = C_Z = 0.1 μF, acceleration = 0 g, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis	±3	±3.6		g
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity ¹			±1		%
SENSITIVITY (RATIOMETRIC)²					
Sensitivity at X _{OUT} , Y _{OUT} , Z _{OUT}	V _S = 3 V	270	300	330	mV/g
Sensitivity Change Due to Temperature ³	V _S = 3 V		±0.01		%/°C
ZERO g BIAS LEVEL (RATIOMETRIC)					
0 g Voltage at X _{OUT} , Y _{OUT}	V _S = 3 V	1.35	1.5	1.65	V
0 g Voltage at Z _{OUT}	V _S = 3 V	1.2	1.5	1.8	V
0 g Offset vs. Temperature			±1		mg/°C
NOISE PERFORMANCE					
Noise Density X _{OUT} , Y _{OUT}			150		μg/√Hz rms
Noise Density Z _{OUT}			300		μg/√Hz rms
FREQUENCY RESPONSE⁴					
Bandwidth X _{OUT} , Y _{OUT} ⁵	No external filter		1600		Hz
Bandwidth Z _{OUT} ⁵	No external filter		550		Hz
R _{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST⁶					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		μA
Output Change at X _{OUT}	Self-Test 0 to Self-Test 1	-150	-325	-600	mV
Output Change at Y _{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z _{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	V _S = 3 V		350		μA
Turn-On Time ⁷	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		-40		+85	°C

¹ Defined as coupling between any two axes.

² Sensitivity is essentially ratiometric to V_S.

³ Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.

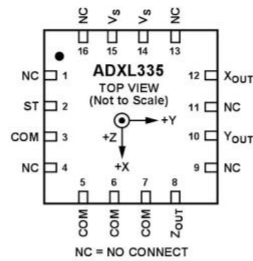
⁴ Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).

⁵ Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 μF, bandwidth = 1.6 kHz. For C_Z = 0.01 μF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 μF, bandwidth = 0.5 Hz.

⁶ Self-test response changes cubically with V_S.

⁷ Turn-on time is dependent on C_X, C_Y, C_Z and is approximately 160 × C_X or C_Y or C_Z + 1 ms, where C_X, C_Y, C_Z are in microfarads (μF).

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NC = NO CONNECT

NOTES
 1. EXPOSED PAD IS NOT INTERNALLY CONNECTED BUT SHOULD BE SOLDERED FOR MECHANICAL INTEGRITY.

Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect ¹ .
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect ¹ .
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z _{OUT}	Z Channel Output.
9	NC	No Connect ¹ .
10	Y _{OUT}	Y Channel Output.
11	NC	No Connect ¹ .
12	X _{OUT}	X Channel Output.
13	NC	No Connect ¹ .
14	V _S	Supply Voltage (1.8 V to 3.6 V).
15	V _S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect ¹ .
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

¹NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

EVAL-ADXL335Z

DESCRIPTION

The EVAL-ADXL335Z is a simple evaluation board that allows quick evaluation of the performance of the ADXL335 three-axis accelerometer. The EVAL-ADXL335Z has a 6-pin, 0.1 inch spaced header for access to all power and signal lines that the user can attach to a prototyping board (breadboard) or wire using a standard plug. Four holes are provided for mechanical attachment of the EVAL-ADXL335Z to the application.

The dimensions of the EVAL-ADXL335Z are 20 mm × 20 mm with mounting holes set 15 mm × 15 mm at the corners of the printed circuit board (PCB).

CIRCUIT DESCRIPTION

The schematic of the EVAL-ADXL335Z is shown in Figure 1. Analog bandwidth can be set by changing Capacitors C2, C3, and C4. See the ADXL335 data sheet for a complete description of the operation of the accelerometer.

The part layout of the EVAL-ADXL335Z is shown in Figure 2. The EVAL-ADXL335Z has four factory installed 100 nF capacitors. C1 at V_S is a bypass capacitor to reduce supply noise. C2, C3, and C4 at X_{OUT}, Y_{OUT}, and Z_{OUT} are filter capacitors to set the bandwidth to 50 Hz (see Figure 1). Many applications require a different bandwidth, in which case the user can change C2, C3, and C4 as appropriate.

SPECIAL NOTES ON HANDLING

The EVAL-ADXL335Z is not reverse polarity protected. Reversing the +V supply and ground pins can cause damage to the ADXL335.

Dropping the EVAL-ADXL335Z on a hard surface can generate several thousand gees (g) of acceleration, which may exceed the data sheet absolute maximum limits. See the ADXL335 data sheet for more information.

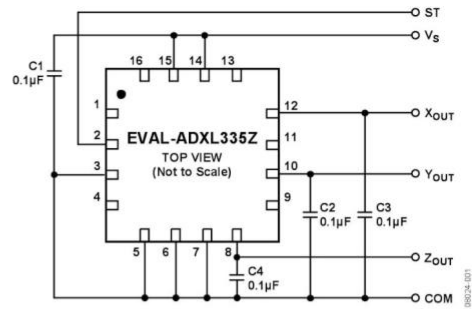


Figure 1. EVAL-ADXL335Z Schematic

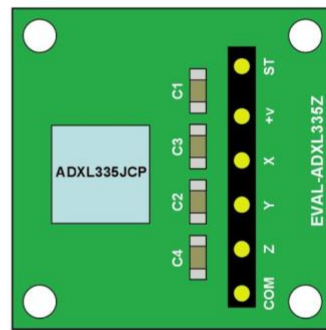


Figure 2. EVAL-ADXL335Z Physical Layout

ORDERING GUIDE

Model	Package Description
EVAL-ADXL335Z ¹	Evaluation Board

¹ Z = RoHS Compliant Part.

Rev. 0

Evaluation boards are only intended for device evaluation and not for production purposes. Evaluation boards are supplied "as is" and without warranties of any kind, express, implied, or statutory including, but not limited to, any implied warranty of merchantability or fitness for a particular purpose. No license is granted by implication or otherwise under any patents or other intellectual property by application or use of evaluation boards. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Analog Devices reserves the right to change devices or specifications at any time without notice. Trademarks and registered trademarks are the property of their respective owners. Evaluation boards are not authorized to be used in life support devices or systems.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
 Tel: 781.329.4700 www.analog.com
 Fax: 781.461.3113 ©2009 Analog Devices, Inc. All rights reserved.

Flex Sensor Data Sheets



FLEX SENSOR FS

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
 - Robotics
 - Gaming (Virtual Motion)
 - Medical Devices
 - Computer Peripherals
 - Musical Instruments
 - Physical Therapy
- Simple Construction
- Low Profile

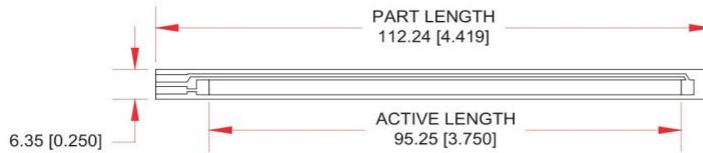
Mechanical Specifications

- Life Cycle: >1 million
- Height: $\leq 0.43\text{mm}$ (0.017")
- Temperature Range: -35°C to $+80^{\circ}\text{C}$

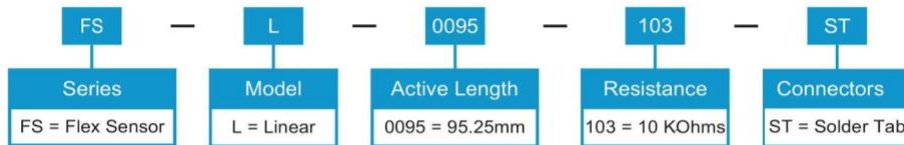
Electrical Specifications

- Flat Resistance: 10K Ohms $\pm 30\%$
- Bend Resistance: minimum 2 times greater than the flat resistance at 180° pinch bend (see "How it Works" below)
- Power Rating : 0.5 Watts continuous; 1 Watt Peak

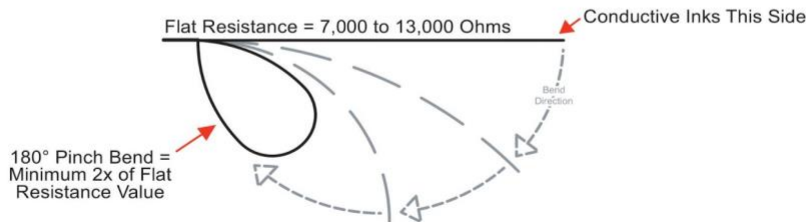
Dimensional Diagram - Stock Flex Sensor

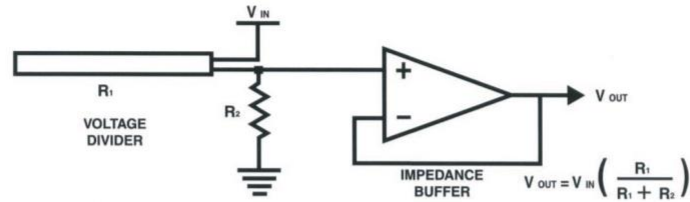


How to Order - Stock Flex Sensor



How It Works



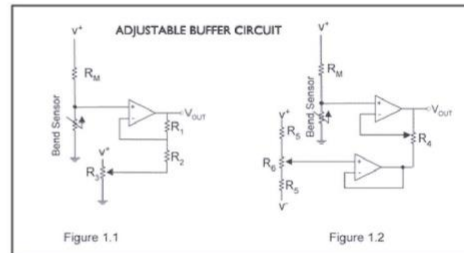
BASIC FLEX SENSOR CIRCUIT:

Following are notes from the ITP Flex Sensor Workshop

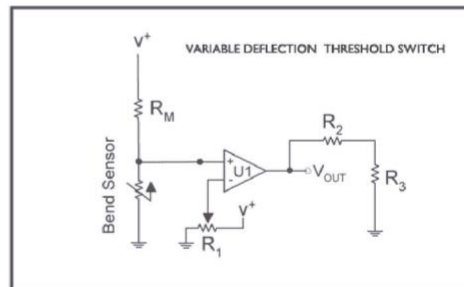
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circuit, and skip the op amp."

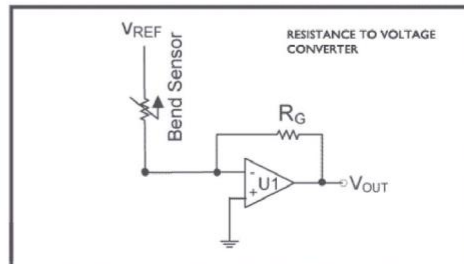
"**Adjustable Buffer** - a potentiometer can be added to the circuit to adjust the sensitivity range."



"**Variable Deflection Threshold Switch** - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"**Resistance to Voltage Converter** - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



Appendix 7: Procedure for Ecoflex Glove

Materials: CAD file for 3D hand, SD card, ABS, Prusa i3 MK3 3D printer, sandpaper, acetone, Ecoflex compound 1A(yellow) and 1B(blue), short flex sensor, two male to female jumper wires(black and red), electrical tape, Elmer's glue stick, small x inch plastic cup, scale, x(brand of mixer) mixer, scalpel.

3D hand:

1. CAD files for a human left hand downloaded from website:
<https://grabcad.com/library/human-hand-1>
2. Download CAD files onto a USB drive
3. Load hand.stl to Ultimaker Cura software
4. Adjust height to x cm and density to 20%
5. Export file to an SD card
6. Insert SD card into the Prusa 3D printer
7. Verify that there is enough ABS in spool
8. Power on Prusa
9. Start print using the selection wheel
10. Once print is complete, remove 3D hand from platform
11. Remove printer debris and smoothen using sandpaper and acetone

Flex Sensor:

1. Attach one black jumper wire and one red jumper wire to flex sensor
2. Use electrical tape to secure jumper wires to flex sensor

Ecoflex Glove:

1. Place a plastic x in. cup on scale and press the tare button
2. Using a 1A:1B mix ratio, add 25g of mixture A(yellow container) into plastic x in. cup
3. Add 25g of B(blue container) into the same plastic cup
4. Adjust x mixer setting to 90g
 - a. Need to adjust to 90g because weight of cup holder is 40g
5. Start mixer
6. Place aluminum on surface of work area
7. Place 3D hand in the center of the aluminum foil
8. Once Ecoflex has finished mixing, evenly pour the viscous liquid onto 3D hand
9. Allow Ecoflex to dry for 20 minutes
10. Repeats steps 2-5
11. Pour second layer
12. Allow Ecoflex to dry for 20 minutes
13. Repeat steps 2-5
14. Coat bottom of flex sensor and wires with Elmer's glue stick and place sensors on the index finger and wires along the top of the Ecoflex covered 3D hand
15. Pour third layer
16. Allow Ecoflex to dry for 20 minutes
17. Repeats steps 2-5
18. Pour fourth layer
19. Allow Ecoflex to dry for 20 minutes
20. Using scalpel, cut off extra Ecoflex that is at the at the base of the 3D hand

Bibliography

- [1] Office of Communications and Public Liaison. "Post-Stroke Rehabilitation Fact Sheet." *National Institute of Neurological Disorders and Stroke*, U.S. Department of Health and Human Services, Sept. 2014, <www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Post-Stroke-Rehabilitation-Fact-Sheet>
- [2] Yue, Zan et al. "Hand Rehabilitation Robotics on Poststroke Motor Recovery." *Behavioural neurology* vol. 2017 (2017): 3908135. doi:10.1155/2017/3908135, <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5688261/>>
- [3] Porciuncula, Franchino, et al. "Wearable Movement Sensors for Rehabilitation: A Focused Review of Technological and Clinical Advances." *PM & R : the Journal of Injury, Function, and Rehabilitation*, U.S. National Library of Medicine, 10 Sept. 2018, <www.ncbi.nlm.nih.gov/pubmed/30269807>
- [4] Dong Hyun Kim, S. W. Lee and Hyung-Soon Park, "Sensor evaluation for soft robotic hand rehabilitation devices," *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, Singapore, 2016, pp. 1220-1223. doi: 10.1109/BIOROB.2016.7523797, <<https://ieeexplore.ieee.org/document/7523797>>
- [5] Placidi, Giuseppe et al. "Measurements by A LEAP-Based Virtual Glove for the Hand Rehabilitation," *Sensors*, Volume 18, Issue 3, MPDI. March 2018, <<https://www.mdpi.com/1424-8220/18/3/834>>
- [6] Lin, BS., Lee, IJ., Chiang, PY. et al. "A Modular Data Glove System for Finger and Hand Motion Capture Based on Inertial Sensors," *Journal of Medical and Biological Engineering*, June 2018, <<https://doi.org/10.1007/s40846-018-0434-6>>
- [7] Kortier, Henk G. et al. "Assessment of hand kinematics using inertial and magnetic sensors." *Journal of NeuroEngineering and Rehabilitation* 2014, 11:70, <<https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-11-70>>
- [8] P. Hsiao, S. Yang, B. Lin, I. Lee and W. Chou, "Data glove embedded with 9-axis IMU and force sensing sensors for evaluation of hand function," *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, 2015, pp. 4631-4634. doi: 10.1109/EMBC.2015.7319426, <<https://ieeexplore.ieee.org/abstract/document/7319426>>
- [9] Rivera, Patricio et al. "Recognition of Human Hand Activities Based on a Single Wrist IMU Using Recurrent Neural Networks," *International Journal of Pharma Medicine and Biological Sciences*, Vol. 6, No. 4, October 2017. <<http://www.ijpmbs.com/uploadfile/2017/1227/20171227050020234.pdf>>
- [10] Borghetti, Michela et al. "Sensorized Glove for Measuring Hand Finger Flexion for Rehabilitation Purposes," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, pp.

3308-3314, December 2013,
<<https://ieeexplore.ieee.org/document/6566034>>

[11] Ninja P. Oess, Johann Wanek, and Hubertus J. A. van Hedel, "Enhancement of bend sensor properties as applied in a glove for use in neurorehabilitation settings," *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Buenos Aires, Argentina, 31 August-4 September 2010, pp. 5903-5906,
<<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5627534>>

[12] Ganeson, Suhassni et al. "Design of a Low-Cost Instrumented Glove for Hand Rehabilitation Monitoring System," *IEEE International Conference on Control System, Computing and Engineering*, Penang, Malaysia, November 2016,
<<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7893569&tag=1>>

[13] Shao, Yitian et al. "Spatial patterns of cutaneous vibration during whole-hand haptic interactions." *Proceedings of the National Academy of Sciences of the United States of America* vol. 113,15 (2016): 4188-93. doi:10.1073/pnas.1520866113,
<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4839404/>>

[14] Wang et al. "Interactive wearable systems for upper body rehabilitation: a systematic review." *Journal of NeuroEngineering and Rehabilitation* (2017) 14:20 doi: 10.1186/s12984-017-0229-y,
<<https://www.ncbi.nlm.nih.gov/pubmed/28284228>>

[15] Lu, Sitong et al. "A 3-D finger motion measurement system via soft strain sensors for hand rehabilitation," *Sensors and Actuators A: Physical*, Vol. 285, Science Direct, Jan. 2019,
<<https://www.sciencedirect.com/science/article/pii/S0924424717317417>>

[16] Lien, Jaime et al. "Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar," *ACM Transactions on Graphics (TOG)*, Vol. 35, Issue 4, July 2016, Article No. 142,
doi: 10.1145/2897824.2925953,
<<https://dl.acm.org/citation.cfm?id=2925953>>

[17] Bu, Tianzhao et al. "Stretchable Triboelectric-Photonic Smart Skin for Tactile and Gesture Sensing," *Advanced Materials*, Vol. 30, Issue 16, March 2018.
<<https://onlinelibrary.wiley.com/doi/full/10.1002/adma.201800066>>

[18] Iravantchi, Yasha et al. "BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming." *CHI '19 Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* Paper No. 15, Glasgow, Scotland, UK, 4 - 9 May 2019, doi:10.1145/3290605.3300245,
<http://delivery.acm.org/10.1145/3310000/3300245/paper015.pdf?ip=71.198.195.93&id=3300245&acc=OPEN&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E6D218144511F3437&__acm__=1559512423_07708f401cf6cf530a38a8046defd806>

[19] H. Chang and J. Chang, "Sensor Glove based on Novel Inertial Sensor Fusion Control Algorithm for 3D Real-time Hand Gestures Measurements," in *IEEE Transactions on Industrial*

Electronics. doi: 10.1109/TIE.2019.2912765,
<<https://ieeexplore.ieee.org/abstract/document/8701582>>

[20] Davila, Juan Carlos et al. "Wearable Sensor Data Classification for Human Activity Recognition Based on an Iterative Learning Framework." *Sensors (Basel, Switzerland)* vol. 17,6 1287. 7 Jun. 2017, doi:10.3390/s17061287,
<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5492798/>>

[21] Bruno, Barbara et al. "Using Fuzzy Logic to Enhance Classification of Human Motion Primitives." *Communications in Computer and Information Science*, July 2014. doi: 10.1007/978-3-319-08855-6_60,
<https://www.researchgate.net/publication/264829303_Using_Fuzzy_Logic_to_Enhance_Classification_of_Human_Motion_Primitives>

[22] Chen, Sheng-Tao et al. "Design and Development of a Wearable Device for Heat Stroke Detection." *Sensors (Basel, Switzerland)* vol. 18,1 17. 22 Dec. 2017, doi:10.3390/s18010017,
<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5796472/>>

[23] Franck JA, Smeets RJEM, Seelen HAM. "Changes in actual arm-hand use in stroke patients during and after clinical rehabilitation involving a well-defined arm-hand rehabilitation program: A prospective cohort study." *PLoS ONE* 14(4): e0214651. Adelante Rehabilitation Centre, dept. of Brain Injury Rehabilitation, Hoensbroek, the Netherlands, 1 April 2019,
<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6443150/>>

[24] Y. Xue, Z. Ju, K. Xiang, J. Chen and H. Liu. "Multimodal Human Hand Motion Sensing and Analysis - A Review." *IEEE Transactions on Cognitive and Developmental Systems*, 31 Jan 2018. DOI: 10.1109/TCDS.2018.2800167, <<https://ieeexplore.ieee.org/document/8277185>>

[25] M. Seiffert, F. Holstein, R. Schlosser and J. Schiller, "Next Generation Cooperative Wearables: Generalized Activity Assessment Computed Fully Distributed Within a Wireless Body Area Network," in *IEEE Access*, vol. 5, pp. 16793-16807, 2017. doi: 10.1109/ACCESS.2017.2749005, <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8025582&isnumber=7859429>>

[26] Kim, Dong Hyun et al. "Improving Kinematic Accuracy of Soft Wearable Data Gloves by Optimizing Sensor Locations." *Sensors (Basel, Switzerland)* vol. 16,6 766. 26 May. 2016, doi:10.3390/s16060766,
<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4934192/>>

[27] Yang, Che-Chang, and Yeh-Liang Hsu. "A review of accelerometry-based wearable motion detectors for physical activity monitoring." *Sensors (Basel, Switzerland)* vol. 10,8 (): 7772-88. doi:10.3390/s100807772, <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231187/>>

[28] Wiggers, Kyle. "Apple Watch Series 4 Can Detect Falls, Take ECGs, and Lead You through Breathing Exercises." *VentureBeat*, VentureBeat, 13 Sept. 2018,

<venturebeat.com/2018/09/12/apple-watch-series-4-can-detect-falls-take-ecgs-and-lead-you-through-breathing-exercises/>

[29] Black, Erin. "Dexcom's Newest Diabetes Device Can Read Your Blood Sugar without Any Blood, and It's a Life-Changer." *CNBC*, CNBC, 12 May 2018, <www.cnbc.com/2018/05/12/dexcom-g6-review-a-cgm-that-doesnt-require-finger-pricks.html>

[30] I. Bisio, A. Delfino, F. Lavagetto and A. Sciarrone, "Enabling IoT for In-Home Rehabilitation: Accelerometer Signals Classification Methods for Activity and Movement Recognition," in *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 135-146, Feb. 2017. doi: 10.1109/JIOT.2016.2628938, <https://ieeexplore.ieee.org/abstract/document/7744470>

ADX1335 datasheet

<https://www.sparkfun.com/datasheets/Components/SMD/adx1335.pdf>

ADXL335 EB datasheet

<https://www.analog.com/media/en/technical-documentation/evaluation-documentation/EVAL-ADXL335Z.pdf>

Flex Sensor Datasheet

<https://www.spectrasymbol.com/wp-content/uploads/2016/12/FLEX-SENSOR-DATA-SHEET-v2014-Rev-A.pdf>