# AN ACCELEROMETER-BASED GESTURE RECOGNITION SYSTEM FOR A TACTICAL COMMUNICATIONS APPLICATION

Robert S. Tidwell Jr. B.S

Thesis Prepared for the Degree of

MASTER OF SCIENCE

UNIVERSITY OF NORTH TEXAS

December 2015

APPROVED:

Dr. Krishna Kavi, Major Professor
Dr. Robert Akl, Co-Major Professor
Dr. Armin Mikler, Committee Member

Tidwell Jr., Robert S. An Accelerometer-Based Gesture Recognition System for a Tactical Communications Application. Master of Science (Computer Engineering), December 2015, 46 pp., 5 tables, 22 illustrations, 21 numbered references.

In modern society, computers are primarily interacted with via keyboards, touch screens, voice recognition, video analysis, and many others. For certain applications, these methods may be the most efficient interface. However, there are applications that we can conceive where a more natural interface could be convenient and connect humans and computers in a more intuitive and natural way. These applications are gesture recognition systems and range from the interpretation of sign language by a computer to virtual reality control. This Thesis proposes a gesture recognition system that primarily uses accelerometers to capture gestures from a tactical communications application. A segmentation algorithm is developed based on the accelerometer energy to segment these gestures from an input sequence. Using signal processing and machine learning techniques, the segments are reduced to mathematical features and classified with support vector machines. Experimental results show that the system achieves an overall gesture recognition accuracy of 98.9%. Additional methods, such as non-gesture recognition/suppression, are also proposed and tested.

Copyright 2015

by

Robert S. Tidwell Jr.

## ACKNOWLEDGMENTS

I would like to thank Dr. Krishna Kavi for his continual support, mentorship, and guidance in the development of this thesis. I would also like to thank Dr. Robert Akl for his support and guidance throughout this process. Additionally, thank you to my committee member Dr. Armin Mikler for his advice, mentorship, and assistance with the writing of this document and also Dave Strubble, formerly of Raytheon Company, for his support and sponsorship of this project. Lastly, thanks are due to all of the previous Biocomm project members for their contributions before this thesis.

# TABLE OF CONTENTS

# LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

Gesture recognition is the undertaking of interpreting body motion and state using a computer or computer program. Gesture recognition is a component of Human-Computer Interaction (HCI). In modern society, interfacing to computers is primarily accomplished via keyboards, touch screens, voice recognition, video analysis, and other hardware. For certain applications, these methods may be the most efficient interface. However, one can envision applications where a more natural interface would be convenient and could connect humans with computers in a more intuitive and natural way. These applications are gesture recognition systems and range from the interpretation of sign language by a computer to virtual reality control. This work specifically focuses on an application designed to interpret human arm and hand gestures for the purpose of a communications application.

## 1.1. The Human-Computer Interaction

HCI is the study of how, when, where, and why humans interact with computers. All of the known interactions between humans and computers are encompassed in this topic and since the advent of the computer, people have been interested in and studied this field. In some ways, it is extremely important and the commercial success of computing in general can be traced back to advances in HCI. The article [14] gives an interesting description and development timelines for some of the more popular interface inventions including the mouse, graphical user interfaces, text editing, hyper text for use in internet browsers, and even gesture recognition. Furthering our ability to more naturally interact with computers and systems is the motivating factor behind this work.

## 1.2. Gesture Recognition: As a Process

Gesture Recognition in general is best described as a process. The process begins with user input and ends with classification. Most systems documented in the literature follow some form of this process, either completely or partially, and so it is important to understand this process as it is the framework upon which gesture recognition systems are built.

### 1.2.1. User Input

The process of gesture recognition begins with user input. Gesture recognition systems are designed to sense user input and perform some action. User input is usually in the form of discreet gestures. The first step in the process of designing a gesture recognition system is to analyze and understand exactly what type of gestures the system must recognize. This determines the actions and movements that the user will make when the gestures are performed. Understanding of these movements is required to select the appropriate sensors to collect the data. Gesture recognition research projects include many types of user inputs including: facial expression, eye movement, hand and arm movement, and body movement. These inputs represent the scope of the gesture recognition system and are in turn measured with a variety of sensors.

### 1.2.2. Sensors

In a gesture recognition system, input is described to the system by data collected from one or more sensors. These sensors can be varied or in some cases can even be a single sensor. Facial expression and eye motion applications typically involve a camera or photo sensitive device. Body, hand, and arm movement have been measured with accelerometers, surface EMG sensors, and bend resistive sensors. Sensors used or previously considered for this work are described below.

#### 1.2.2.1. Accelerometers

Accelerometers are devices that measure acceleration due to force. Two-dimensional accelerometers measure acceleration in two dimensions, and three-dimensional accelerometers in three dimensions. When working with accelerometers it is important to note that since these devices measure acceleration acting on its sensing element, gravity is included in the overall output and must be considered. The acceleration due to Earth's gravity is a unit of measure typically shown as 1g. The accelerometers used in this work have a sensitivity of 2.5g or two and a half times the force of Earth's gravity. This sensitivity is suitable for capturing human arm movement in a normal setting (although moving extremely quickly can saturate the sensor). Figure 1.1 is an example of the output data from a three dimensional accelerometer used in this work. Note here that "Force Value" in the plot is simply an abstract calculation of the amount of force applied to the sensors. This sample is 4.2 seconds of data where the sensor, which was attached to the back of the hand, was briefly moved then stopped. This process was repeated for the duration of the sample.

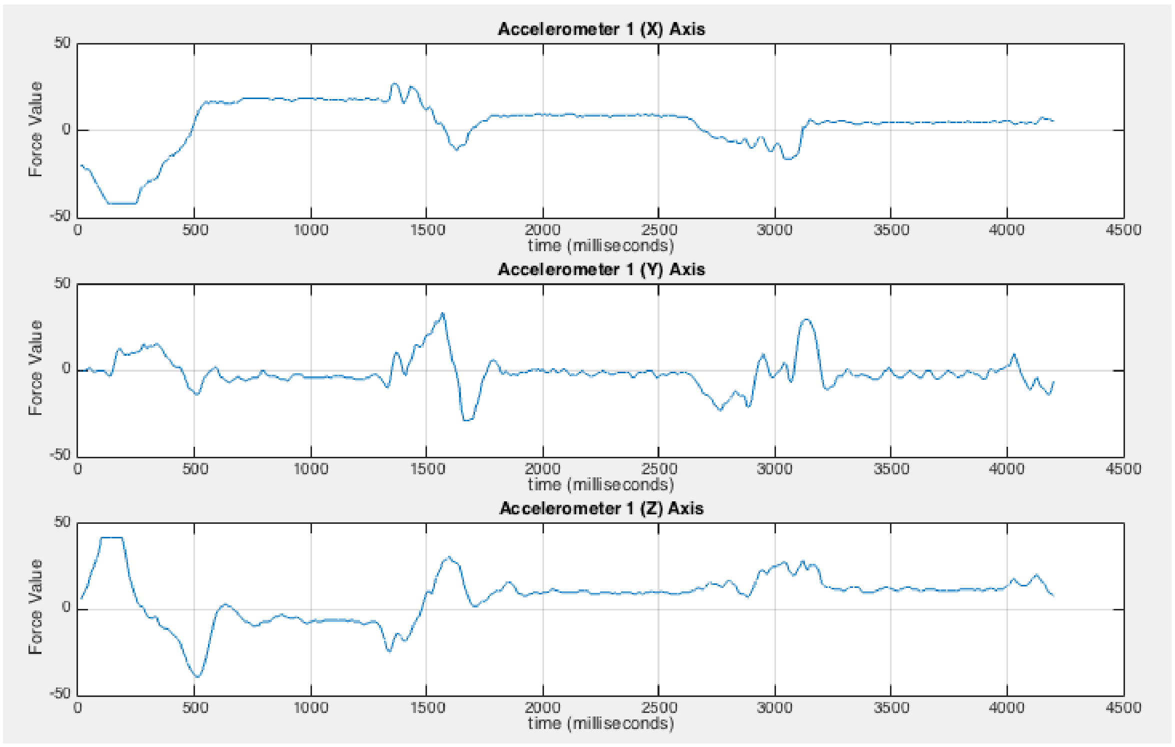


FIGURE 1.1. Example Accelerometer Data

### 1.2.2.2. Surface EMG

Surface EMG (Electromyography) sensors are devices that measure the electrical activity produced by human muscles. When the muscles are activated they emit an electrical potential. These sensors, which are attached to the surface of the skin, measure the potential difference (in the mV range) of the area which when plotted in the time domain appears as a very noisy signal. Figure 1.2 is an example of the output of an SEMG sensor. In this example a single SEMG sensor is placed on the forearm while a single gesture is formed by the hand and briefly held. The time domain signal of SEMG is typically processed further due to noise and other factors. Sometimes it will be passed through a bandpass filter to remove mechanical and contact noise below 5Hz and target only the desired range of frequency. Also, it is typical to use some features from the frequency domain, such as the mean frequency, in the gesture recognition problem.

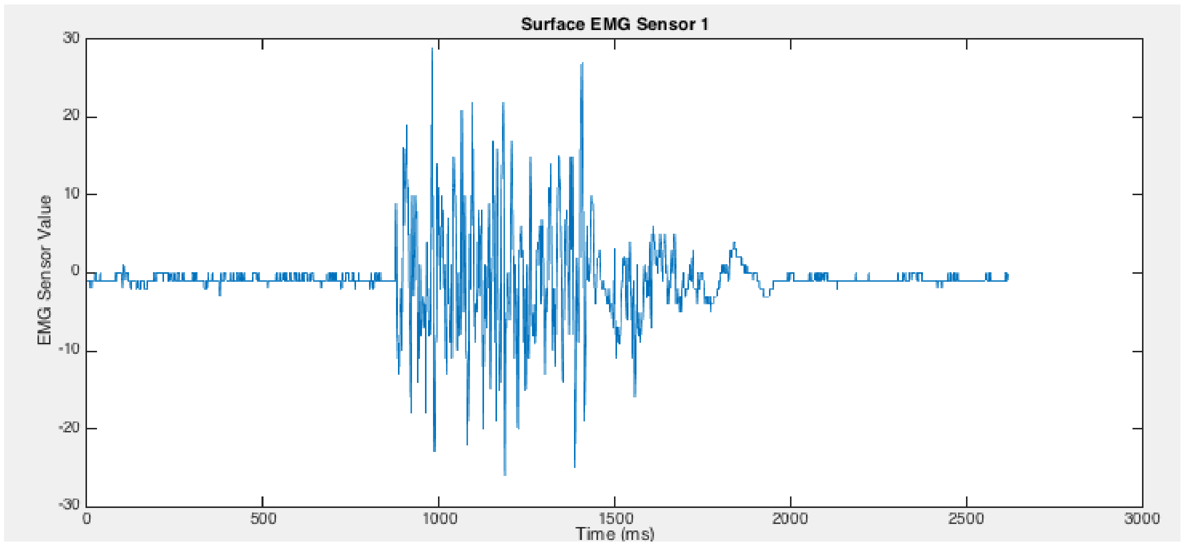


Figure 1.2. Example SEMG Data

### 1.2.2.3. Bend Resistive Sensors

Bend resistive (or Piezoresistive) sensors are devices that change electrical resistivity when bent. The sensors in this work are 3.8 inches long, .24 inches wide and flat. They are attached to a glove that is worn on the hand and they bend with the movement of the fingers to accurately determine hand orientation. The output values are determined by the angle and radius of the bend. Figure 1.3 shows a short sample of one bend sensor being moderately bent towards a 90 degree angle.

## 1.2.3. Data Collection

Data collection is a very important part of the gesture recognition system. Decisions on how the data will be collected and segmented for further processing are necessary to solve

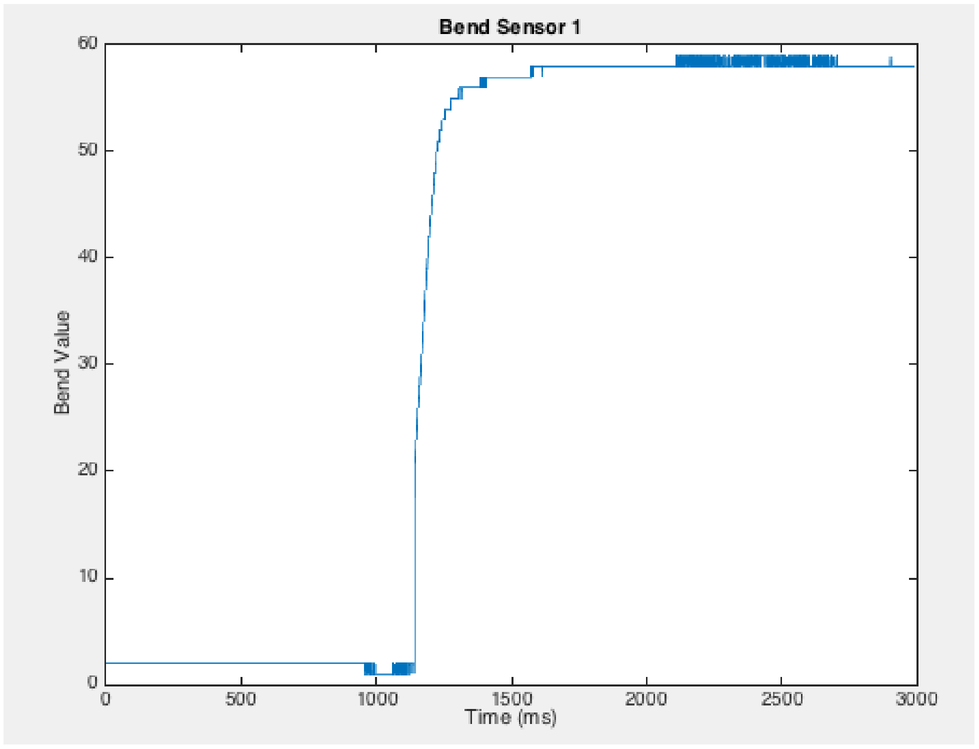


Figure 1.3. Example Bend Sensor Data

gesture recognition problem. For example, a system may be built to be constantly collecting data from the sensors, but a decision needs to be made on how this data is segmented and forwarded for further analysis. Consider the system in [20], which uses the average energy of multiple SEMG sensors to determine the start and endpoints of the active gesture segments. This allows the system to create equal length vectors of data to be further processed. This thesis follows a similar process that will be described later and labels the active gesture segments as gesture windows. The gesture windows are representative of each individual gesture from the user. Another aspect of the system that can be determined here is sampling rates of the sensors. For some sensors, like SEMG, the frequency of the output signal is well known and a general sampling rate can be given to all EMG sensors (0-500Hz range requires a 1ms sampling rate). But for other sensors, like accelerometers, the sampling rate may only need to be as fast as necessary to accurately capture the input being measured. So the context of the data can define this choice of sampling rate for each sensor. The sampling rate of sensors is the primary factor in determining the amount of data; A shorter sampling rate will produce more data collected over the same time period. This has a significant impact on the system design.

### 1.2.4. Feature Extraction

Feature extraction is the act of taking raw input signal vectors and extracting some meaningful information from the data. When the vectors are input to a classification system they are referred to as features. These features are attempts to represent the salient aspects of the data, while lowering its dimensional order for classification. The necessity of using features is apparent in the classification algorithms. Each value is considered a feature, and each feature represents a new dimension to the solution space. If the raw data vectors are input to the classification algorithm, each data point becomes a new dimension. Considering that this project uses data vectors with lengths of greater than 400 data points, one can quickly see that a solution with 400 dimensions is very complex. Therefore the data vectors must be represented with meaningful attributes. Researchers have used a wide variety of calculations and methods to create these values for each type of sensor used. These values include mean average value, standard deviation, root mean square, variance, zero crossing rate, Fourier transform coefficients, mean frequency, median frequency, and others. Not all features are useful for classification in every system. The inclusion of a feature should be scrutinized carefully when using classification algorithms because each feature increases the dimensionality of the classification system and can reduce accuracy or make the classification problem too hard to solve. Choosing the correct and most valuable features for classifying data is a significant part of solving the overall gesture recognition problem.

### 1.2.5. Classification

Classification is the process of distinguishing the appropriate class or gesture from the input features of the data. In gesture recognition systems, this is typically the most complex part. The algorithms used for classification are part of the computer science field for machine learning. These computational models are designed to recognize patterns in data, and by doing so achieve classification. There have been several classification algorithms used in gesture recognition systems such as neural networks, hidden Markov models, and support vector machines. The type of classification algorithm used depends on the choices in feature selection and the data itself.

Hidden Markov Models have been extensively used in speech recognition and computer vision applications [7]. These models work by calculating a distribution of probabilities over a sequence of observations. Each observation is assumed to be generated by a unobserved process and the process which is hidden from the user and satisfies the Markov property. The Markov property states that the current state includes all of the information we need to know and that it is independent of all states prior to it. These algorithms have also been used to solve classification problems in gesture recognition systems [20] [13] [12].

Artificial Neural Networks have been used in forecasting models and decision making processes [8]. The computational devices are inspired by how biology processes information. ANNs consist of a large number of processing elements that simulate neurons. Each ANN is configured for a specific task, such as pattern recognition, and learns by training it with data. The ANN learns and creates firing rules for it's neurons based on the training examples. This method has been used in some gesture recognition projects [1].

Support Vector Machines (SVM) are a form of supervised machine learning. These learning systems use linear functions in a multidimensional feature space to learn trained data based on statistical learning methods. SVMs are very powerful and can outperform most other systems in a wide variety of applications [5]. Included in the SVM framework is the ability to use various forms of kernels. The kernel of an SVM describes the basic principles on which the target function is created. For instance, this work uses both a polynomial kernel, which builds the target function based on a polynomial function, and a linear kernel, which does the same with a linear function. The appropriateness of the choice of kernel depends on how separable the data is and if it is linearly separable or not.

In this project multiple Support Vector Machines are used throughout the system to classify all of the instances the system produces. By using SVM, excellent accuracy has been achieved and is shown in the results chapter.

# CHAPTER 2

# RELATED WORK

## 2.1. Previous work on the Biocomm Project

The Biocomm project in the UNT Computer Science and Engineering department began in 2011. Since that time multiple students (including the author) have worked on the project. Of these students, two have produced masters theses based on the systems developed in the project. The earliest work was reported in [3]. The focus of that work was to evaluate surface EMG sensors and bend sensors for classifying hand gestures. The system differs considerably from the system described in this work. That system was a manually windowed/segmented system that included the transition from the ”rest” gesture to the target gesture in each gesture window. The transition data was also included in the feature extraction and classification system and was a primary component in the accuracy of the system. In the second master thesis, [11], accelerometers were used in the same manner of manual windowing and inclusion of transition data. This work is different in both the method of creating the gesture window and the data that is presented for classification which significantly alters both the problem and result. Furthermore, this work is substantially different in the approach by not excluding movement data and focusing only on the gestures while they are performed. This work also goes beyond previous efforts by including some system functions and non-gesture tolerance, which is described later.

## 2.2. Other Gesture Recognition Works

Gesture recognition projects differ in many ways. Following with the process outlined previously, the first category is the gesture set attempting to be classified. This describes not only the input to the system, but also the end goal of the system in terms of classification. In this section, focus is given to gesture recognition works that deal mostly with hand and arm movements which are directly comparable to the work described in this thesis.

Some gesture recognition projects use a set of gestures that are defined by the system. That is to say, a human wouldn't normally do this. The user is only making the gesture because it is meant to control the system they are interfacing with. These are derived gestures and they can be stationary or active motion gestures. In [1] the authors use four hand gestures with motion at the wrist: up, down, left, right. This is a good example of a system that might be used to control motion in a virtual object, or possibly a vehicle of some type. Another example is the second test in [20], in which the authors define 12 circular gestures to rotate the 6 faces of a virtual Rubik's Cube. In [16] the authors define a set of 12 active motion gestures that are meant to be basic shapes. Some shapes are simple, such as making a circle or a square, and some are more complicated such as using active motion to make a letter in the alphabet. This could be used as an interface to a drawing program or even a very simple communications application, but because this is not something a human would naturally do, it is still a derived set of gestures. Another example is [13], in which the authors have a derived set of directional and shape motions for use in a 3D virtual environment application. There are other works that take a similar approach in the gesture set including [18], [17], and [2]. All of these works use a basic derived shape or movement to define an input class or gesture.

In this thesis and other works, the goal is to capture and classify gestures that are natural human gestures. Or more appropriately, gestures that humans are already performing for various reasons. The goal of this type of project is to create an unobtrusive wearable system that can capture and accurately classify these gestures. The usefulness of these gestures can have a broad range of applications from communications to the medical industry. In [4], the authors use a set of gestures that includes derived movements as well as actual ASL/CSL (American/Chinese Sign Language) hand gestures. Also, the first test in [20] uses a set of 72 CSL gestures. Another interesting project is the use of gesture recognition in medicine. In [12] the authors capture and analyze hand movement and position while performing a laproscopic surgical procedure. In [10], the authors employ the same devices used in [9], but in this application they attempt to analyze the gait of patients diagnosed with Parkinson's Disease.

Some works have been done that do not define a set of gestures but attempt to monitor positioning of the body in real-time and in a more general way. For example in [9], the authors measure arm and hand position in real-time. This has a definite application to virtual control systems and possibly many others. This is still gesture recognition but not as defined as to actually implement a gesture set. They are attempting to capture a broader scope of movement using their sensors. The same is also true of [12], which captures all movements and compares them to either novice or experienced surgeons.

Accelerometer based gesture recognition systems have a growing base of research. Some systems, such as [16], [17], and [2], use only a single 3-axis accelerometer, while other systems such as, [4], [20], [9], [10], [18], and [13], use multiple sensors or a fusion of sensors. Sensor fusion is the use of different types of sensors creating a heterogenous sensor network in the gesture recognition system. This is comparable to the work done for this thesis where information about the hand gestures and arm movements is gathered separately with different sensor types. Fusion of the data leads to classification and recognition of the overall (composite) gesture.

SEMG sensors have also been used for hand gesture recognition in other works including [1], [4], and [20]. The main difference here is that the EMG signals are much more complex and difficult to process. Large noise variances and a degree of randomness must be accounted for in these systems which require much more complex feature extraction techniques and a greater number of features.

Image based gesture recognition systems are also fairly popular as described in [6] [19], and [21]. These works focus mostly on separating the user image from the background and using improved methods of classification. A good literature review of 37 such works can be found in [15]. This work describes the prevalent data processing and classification methods used in image based gesture recognition. Other sensors include gyros, such as in [9] and [10], and optical bend sensors in [12].

Feature extraction methods vary widely with no two exactly alike. However, classification algorithms used in these systems remain largely the same. The majority of gesture recognition systems use HMMs, SVMs, or Neural Networks.

# CHAPTER 3

# PROJECT BACKGROUND

The Biocomm project is a communication system for soldiers in the field. The purpose of the system is to allow hand and arm gesture communication between soldiers in the absence of a line of sight and during other impairments to communication, such as cover of darkness and radio silence conditions. The intent is to use a set of sensors in a wearable device, a transmission unit, and a heads up display to facilitate tactical communication of commands or information from a tactical leader to a squad or other group of soldiers.

This thesis focuses on a critical part of the Biocomm project, i.e., the subsystem that responds to sensory input and classifies the input into specific commands. The transmission vehicle, display of information, and security enhancements are possible works for future advancement of the project.

## 3.1. Gesture Selection

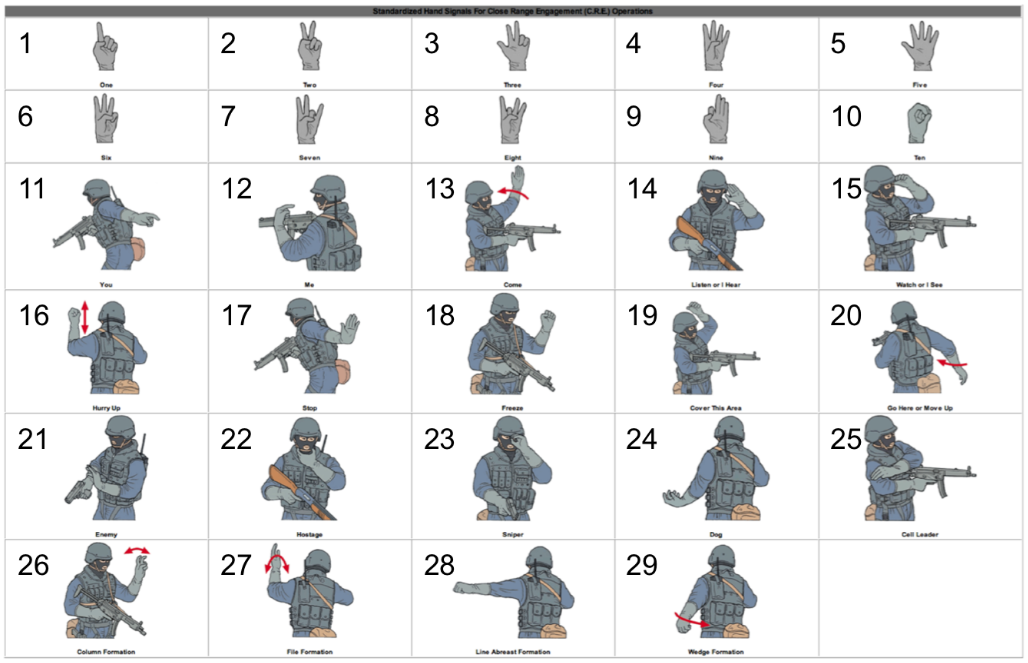
Gesture selection is a key component of realizing any gesture recognition system. The set of gestures that the system can recognize must be carefully planned and is the primary justification for sensor selection and placement.

### 3.1.1. The Gesture (CRE) Chart

The Biocomm project uses the chart in figure 3.1 chart as the base set of gestures to be recognized. This chart is named Standardized Hand Signals for Close Range Engagement Operations and depicts hand and arm gestures that are useful for communicating tactical information. This chart is referenced for the remainder of this document as the CRE Chart.

This chart includes 29 total gestures. For the purpose of gesture recognition, not all of the gestures present the same sensor output. Because of this, the chart is broken down into categories that are useful for determining how the system will interpret them. The categories are: hand, stationary, and active motion gestures. It is important to note that these categories are not based on any contextual information of the gesture being performed

Figure 3.1. The Gesture (CRE) Chart



but on which sensors play the most significant role in recognition. This is explained further in the system description.

### 3.1.2. Hand Gestures

The top two rows of the CRE chart depict hand gestures and are shown in figure 3.2. The system responds to these gestures separately than the other gestures and how these are dealt with is described later in this document. The context for these gestures is presentation of numerical information. The gestures are labelled 1 through 10 and only in this case is the gesture number the same as the context of the gesture.

One other gesture included in the set of hand gestures is ”Freeze”. Figure 3.3 shows this gesture which is also identified as number 18 on the CRE chart.

This gesture is important in the system because the orientation of the arm and hand is precisely the same for all of the other hand gestures. None of the other gestures in the

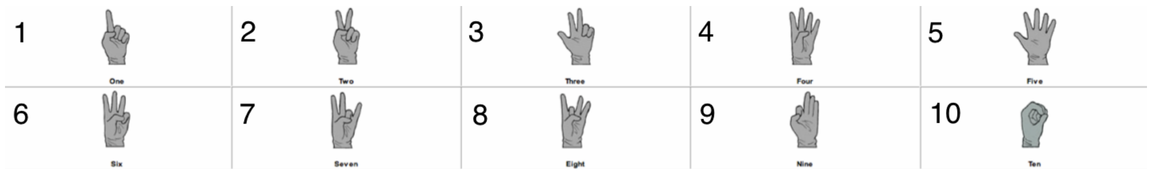


FIGURE 3.2. Hand Gestures 1-10

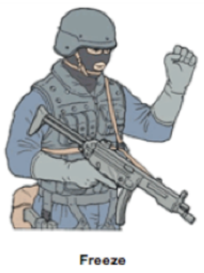


FIGURE 3.3. Gesture 18: Freeze

CRE chart have this specific orientation so we use this gesture as the stimulus required for the system to further classify all of the other hand gestures.

### 3.1.3. Stationary Gestures

Non-hand (or arm) gestures are further categorized into two parts. The first is stationary gestures. These gestures from the CRE chart are viewed as stationary because when they are performed, the user stops moving for sometime. As Described later, the system uses this lack of motion to initiate classification. The stationary gestures make up the majority of all non-hand gestures in the chart and along with the non-hand gestures are the primary focus of this work.

Table 3.1 shows the number and context of the stationary gestures. The number corresponds to its position on the CRE chart continuing in the same fashion that the hand gestures are numbered. Another gesture that is artificially added to the system is the ”Rest” gesture, which is not shown. This gesture corresponds to the hand and arm being motionless at the user’s side which is considered the natural resting state.

| Number | Gesture Context | Number | Gesture Context |
|---|---|---|---|
| 11 | You | 21 | Enemy |
| 12 | Me | 22 | Hostage |
| 14 | Listen | 23 | Sniper |
| 15 | Watch | 24 | Dog |
| 17 | Stop | 25 | Cell Leader |
| 18 | Freeze* | 28 | Line Abreast |
| 19 | Cover | 30 | Resting |

TABLE 3.1. Stationary Gestures including Freeze, and Resting

### 3.1.4. Active Motion Gestures

Active motion gestures are identified as gestures in the CRE chart that are continually moving while being performed. These gestures are the remaining 6 gestures from the CRE chart that have not been previously listed. The information each gesture communicates is in the position of the arm, the movement the arm and the hand make, and the shape formed by the hand. A similar gesture not included in this work might be to wave at someone. It is the movement that creates the context of this gesture. It is also this movement that makes active motion gestures unsuitable for the current version of this system. This work does not currently include classification and gesture recognition of these active gestures, therefore they are not listed here.

## 3.2. Sensors and Placement

The primary sensors selected for this project are two 3-dimensional accelerometers. These sensors are placed on the back of the hand and the forearm just above the wrist as shown in figure 3.4. The reason for placing the sensors in this location is described in Chapter 4 as part of the system overview.

The other set of sensors used on this project are the bend sensors (which are later described as supplementary system sensors) and are attached to the five fingers of a glove

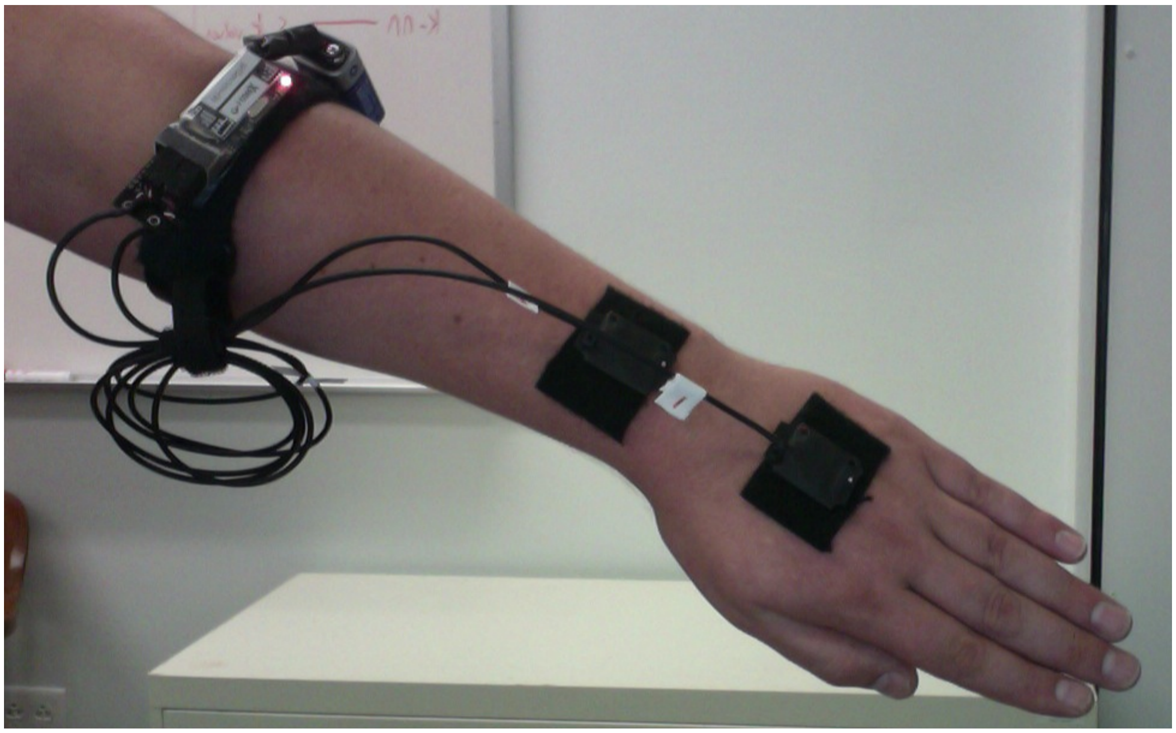


Figure 3.4. Accelerometer Placement

worn on the hand. Figure 3.5 shows the glove and sensors. When the glove and accelerometers are combined the accelerometer on the back of the hand is attached to the glove in the same position shown in figure 3.4.

The total number of sensors used on this project is seven (five bendsensors and two accelerometers) which results in eleven separate data streams (includes 3x2 axis of accelerometer data). The selection and placement of these sensors was intentional to deal specifically with the gesture set chosen for this project and also the conditions that might be present for a soldier in a tactical environment. It is assumed that the soldier is already wearing a glove and that in a final product the bend sensors and accelerometers can be incorporated into the fabric of this glove in an unobtrusive way as not to affect finger or hand dexterity which is of extreme importance in a tactical situation. The second accelerometer might be incorporated into a wrist band of some kind. The most important aspect of using accelerometers in the manner chosen for this project is that they remain in a constant position and orientation. This orientation which will become apparent in the system overview is the prime element

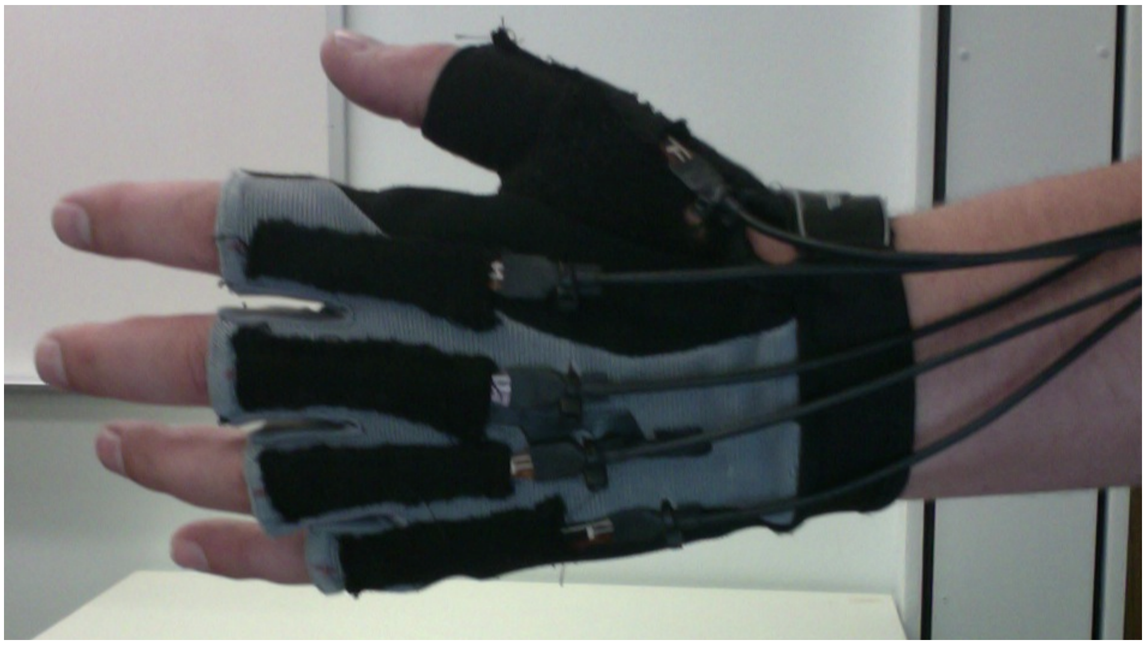


Figure 3.5. Bend Sensor Glove

for gesture recognition. As a matter of necessity, it is assumed that a functional, deployable product can be created which, when worn by a user, is easy and practical to place the sensors in a static position.

# CHAPTER 4

# SYSTEM OVERVIEW

## 4.1. Experimental Equipment

This system uses a computer, a sensor collection device, sensors, and computer software to achieve the gesture recognition system. The computer is an iMac with OSX 10.9.5. Installed on the mac is MATLAB R2014A which is used to analyze the data collected from the system. MATLAB is also used to program our classifiers, algorithms, features, etc. The interface to collection device is achieved via Bluetooth serial port, accessed through MATLAB. The use of a computer and MATLAB is only suitable for the lab environment. In a final standalone product, this would be replaced by using an embedded processor and implementations of the algorithms and classifiers would be stored there.

The collection device used in this project is an I-Cube-X Wi-microDig v6.1, shown in Figure 4.1. This small device is the collection point for all of the sensors used in this project. The device has eight ports for sensors to be connected to it (accelerometers require three ports, bend sensors require one port) and is responsible for collecting the sensor values, digitizing the values, and transmitting them via Bluetooth to the computer.

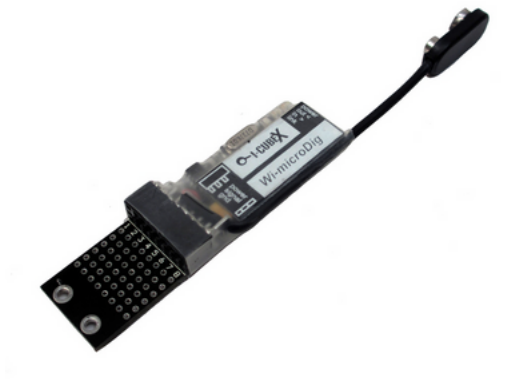


FIGURE 4.1. I-Cube-X Wi-Micro Digitizer

The sensors used in this project are I-Cube-X accelerometers and bend sensors. The accelerometers are the I-Cube-X GForce-3D v6. Figure 4.2 shows the device. Two accelerometers are used as the primary sensors for the system. One is placed on the back of the hand and the other is placed on the fore arm a few inches away from the wrist. A sampling rate of 10ms is used for the accelerometers which allows acquisition of a very high resolution signal when compared to the speed of the gestures performed.

Figure 4.2. I-Cube-X GForce-3D

The bend sensors are I-Cube-X Bendshort v2 as shown in figure 4.3. Five of these sensors, one on each finger, are attached to a glove that is worn on the hand. The sensors bend with the fingers and cover each knuckle including the base of the thumb. Again, for the bend sensors a sampling rate of 10ms is used.

Figure 4.3. I-Cube-X Bendshort

## 4.2. Overview of the System

The system is comprised of multiple stages from user input to final classification. Figure 4.4 shows a block diagram of the flow of data through the system.

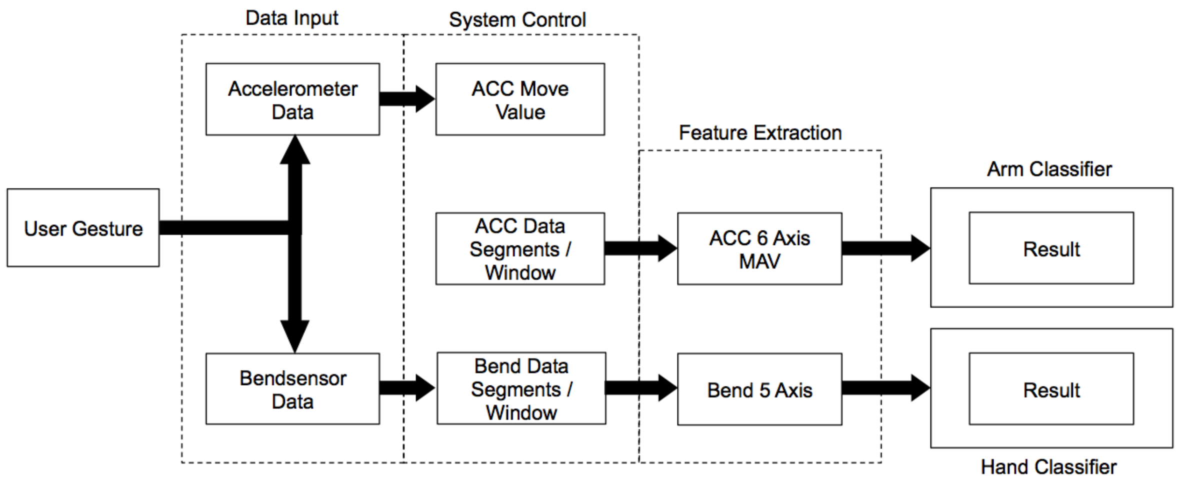


Figure 4.4. Block Diagram of the System

As shown in the figure, accelerometer and bendsensor data are treated separately and move through several stages from system control to feature extraction and finally, classification. Many decisions must be made for the system to function properly and correctly react and classify user input. These decisions are made by the system control section.

## 4.2.1. The Gesture Window

Data segmentation is key to the functionality of the system and correct classification. For the system to operate, one must define how and when the system control identifies the correct data to be sent for feature extraction. These are called gesture windows. To create this, the system is defined based on a simple principle. Since the focus is on the stationary gestures as mentioned in section 3.1.3, the important data will be in the movement of the accelerometers, or more accurately, the lack of movement. An important point to make here is that the data we use throughout the classification process is actually the gravity profile acting on the accelerometers which is explained in more detail later. My system is designed to continuously monitor the change in values of the accelerometers and create the gesture window (or active data segments) based on the movement of the accelerometers. The capturing of movement is defined as the Move value.

The Move value can be thought of as a computation of the energy in the movement a user is currently performing. It is an average of the difference of each accelerometer axis from one point to the next. At time t, where c is the index of the axis, N is the number of axes, and X is a matrix of all six axes of the accelerometer data:

$$MOVE(t) = \frac{1}{N_c}\sum_{c=1}^{N_c} |(X(c,t) - X(c,t-1))| \tag{1}$$

Calculating the Move value for all t gives a vector of Move values, $\mathrm{MOVE_v}$. For added stability I also use a moving average on the Move vector. This ensures that very brief anomalies, where the accelerometers show no movement, are not incorrectly identified as a gesture window. This can happen, for example, when changing movement to the opposite direction and not currently performing a gesture. Calculating the average window with W = 10 samples corresponding to 100ms:

$$MOVE_{avg}(t) = \frac{1}{W}\sum_{i=t-W+1}^{t} MOVE_v(i) \tag{2}$$

The system control tests the moving average defined above against a threshold level which was experimentally derived in this work. When the moving average falls below this threshold, the system identifies this point as the beginning of a gesture window. It then captures the accelerometer data and bend sensor data for the next 100ms or ten samples. This is the gesture window. The system assumes that because the accelerometer is now stationary and not moving, the user is making a gesture. The manner in which this data is handled and forwarded for feature extraction is described later.

Figure 4.5 shows a sample of accelerometer data, along with the MOVE vector and the gesture window creation function. In this sample, three gestures are performed and held briefly. The high periods in the Gesture Window function represent the windows.

### 4.2.2. Gesture Timeout Period

The system also requires a rule to control and define how the user interacts with the system and also how the system reacts to expected input. This rule is a wait period or timeout period after gesture window creation. This is referred to as the Gesture Timeout Period. The reason for introducing this period is that the sampling, window length, and

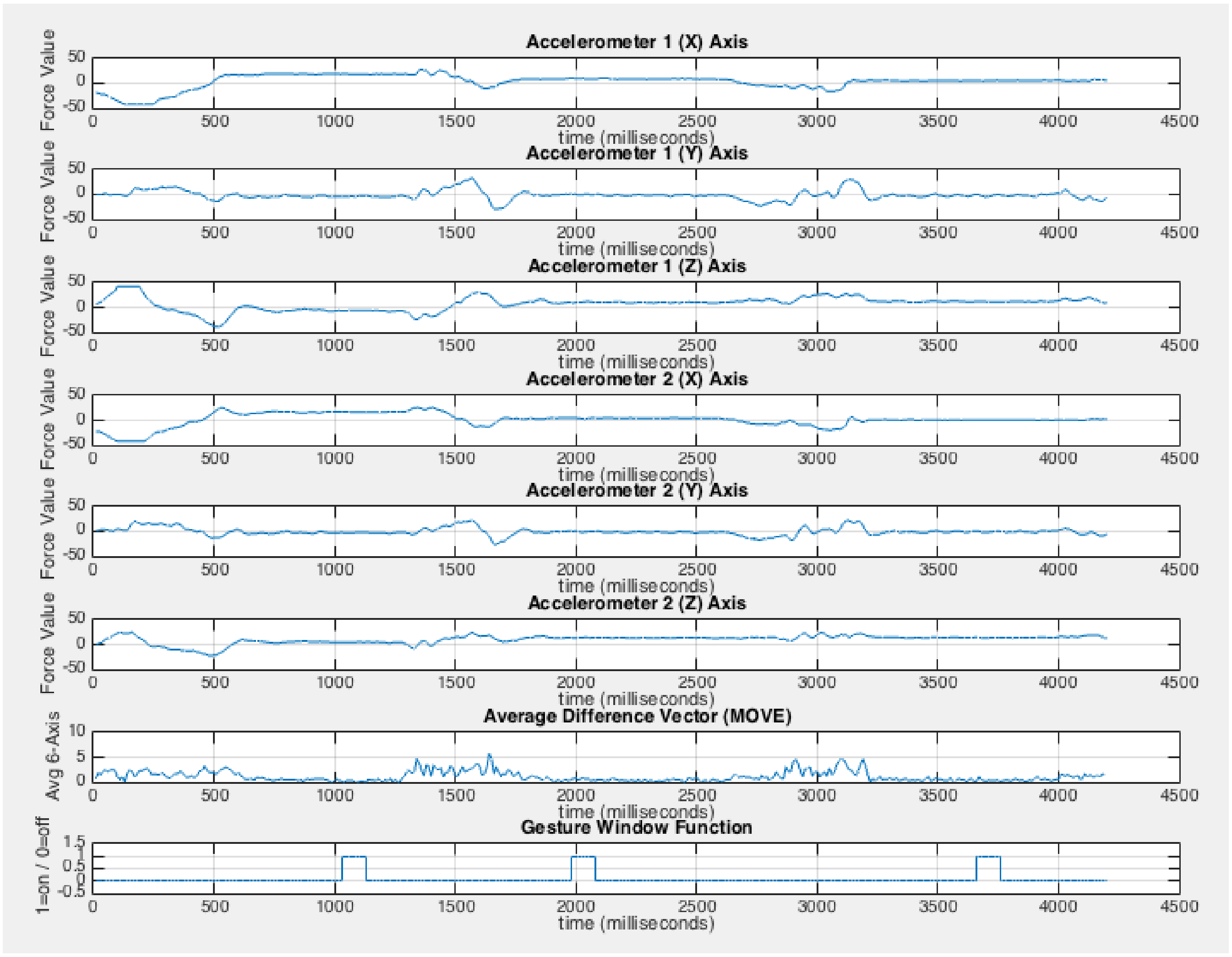


FIGURE 4.5. Example Data with Gesture Window Creation

continuous testing of the MOVE average is happening at a rate that is far faster than what a human would normally do to create gestures. Without this period in place, the system would classify a single gesture made by the user multiple times over. Not only is this an undesired outcome in the output, it also results in wasted computing cycles and power, especially when this data is forwarded for feature extraction and classification.

This timeout period is also a key design element of the system. Since this period directly affects when a new gesture can be classified, it also impacts the timing of the users actions. Because of this, it follows that this period could also be customized or even dynamically adapted to fit a particular user. For this thesis however, a static timeout period of 800ms is used, which is suitable for the lab testing environment.

The control system described thus far can be summarized by Figure 4.6. This state chart shows four states of the gesture windowing system. After an initial period, the system enters the ready state, where the it is now actively testing the MOVE average against the threshold. When the test returns positive (below the threshold), the system enters the capture state for ten samples. The system then moves to the wait state where the gesture timeout period is counting down. When the timeout period reaches zero, the system returns to the ready state.

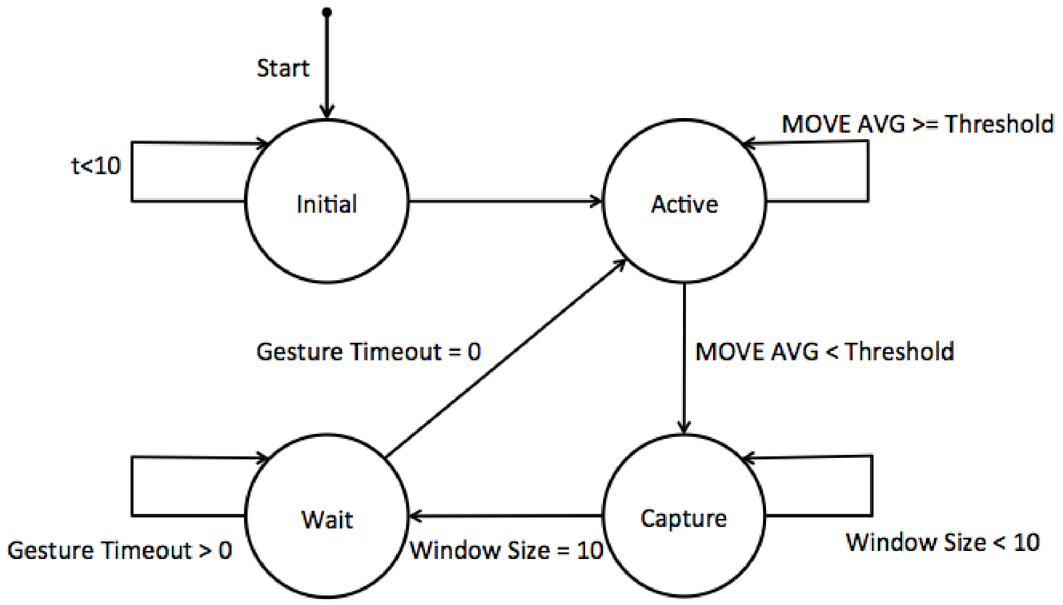


FIGURE 4.6. State Chart of the Gesture Window Creator

## 4.2.3. Classification Feedback

Classification feedback is the returning of the classification output to the system control for the purposes of making further decisions before final output. There are two primary reasons necessitating this feedback. The first is to prevent the system from erroneously outputting the same gesture that was previously output. In some situations this is necessary, for example if the user unintentionally holds the gesture beyond the timeout period. The second is to allow the system an opportunity to use supplementary sensor data toward the overall goal of classification as explained later.

To prevent duplicated output, which happens when user holds the same gesture beyond the gesture timeout period, the classification output is returned to the system control. This output is then compared against the previous output and if certain conditions are met, the output is suppressed. To understand these conditions, consider the context of the gestures in the CRE chart. The arm gestures, which are 11 through 29 of the gesture chart, are gestures that do not need to be sent twice. It is unlikely that a situation that would warrant sending the same communication (”hostage” for example) to the team twice in a row. However, the hand gestures 1 through 10, which are representative of numbers, do present a situation in which the user might want to send a single gesture back to back. One can conceive of a natural way a user might do this. For instance, consider using the hand to communicate the number 112. This requires raising the index finger, then lowering and raising the index finger again, and then lowering and raising the index and middle fingers to present the number 2. This is how a human would naturally communicate the number 112 and the system must be able to respond to this. Likewise, because the only movement happening with the hand and arm is in the hand itself, the accelerometers must be bypassed. This can be accomplished by letting the gesture timeout period expire and, if the system classifies the gesture as a hand gesture, the output is not suppressed. All other gestures however will be suppressed if they are the same as the previous gesture.

Output suppression from classification feedback is summarized in Table 4.1. It is important to note here that the suppression actually happens after final classification. For example in the event of the gesture ”freeze”, the primary accelerometer classification happens and the system enters the supplementary classification system. The feedback is not considered final until it has completed the supplementary classifier and if the double output of ”freeze” is encountered, it is suppressed. This is described later in more detail later.

| Gesture | Suppression Rule | Gesture | Suppression Rule | Gesture | Suppression Rule |
|---|---|---|---|---|---|
| 1:One | None | 9:Nine | None | 19:Cover | Suppress |
| 2:Two | None | 10:One | None | 21:Enemy | Suppress |
| 3:Three | None | 11:You | Suppress | 22:Hostage | Suppress |
| 4:Four | None | 12:Me | Suppress | 23:Sniper | Suppress |
| 5:Five | None | 14:Listen | Suppress | 24:Sniper | Suppress |
| 6:Six | None | 15:Watch | Suppress | 25:Cell Leader | Suppress |
| 7:Seven | None | 17:Stop | Suppress | 28:Line Abreast | Suppress |
| 8:Eight | None | 18:Freeze | Suppress | 30:Resting | Suppress |

Table 4.1. Suppression Rules for Classification Feedback

## 4.3. Supplementary Sensors and Classifiers

The supplementary sensors in this system are the bend sensors. In this system they are used to determine the orientation of the fingers of the hand. These sensors are only included in the gesture recognition system when it makes sense to use them. For instance, to classify the gestures ”stop” or ”enemy”, numbers 17 and 21 on the CRE chart respectively, the classification algorithm can easily differentiate the data of the accelerometers. Therefore there is no need to rely on supplementary sensor data. In fact, using supplementary data in our primary classification system would make the classification problem harder and reduce accuracy. There is however a small subset of gestures from the CRE chart that have similar gravity profiles. In these instances an additional step is taken to classify the gestures with a second classifier using supplementary sensor data. Also, all of the hand gestures (numbers 1 through 10) are classified with the supplementary sensor classifier. This results in a total of three classifiers.

The main supplementary hand classifier includes the hand gesture 1 through 10 and the gesture ”freeze”. The bend sensor data is the only data present in the classifier. Entrance to this classifier is attained by the primary classifier returning the gesture ”freeze”. The reason for this is that the gesture "freeze" and all other hand gestures are in the exact same place when viewed from the accelerometer perspective. They all have the same accelerometer gravity profile. Once the classification feedback returns "freeze", the system control sends the bendsensor data for feature extraction and classification with this separate classifier.

The secondary supplementary hand classifier includes the gestures "watch", "line abreast", and "you". To justify the necessity of using supplementary sensors for these specific gestures, one need only look at the data to be classified. The following plot views are feature extracted data from the accelerometers. Figure 4.7 and 4.8 shows a 3-dimensional plot of the accelerometer data for five samples of two gestures, "freeze" and "rest". If one examines the plots you can see that at a minimum, for each accelerometer there is a large difference in at least one axis or dimension. The classifier can easily separate this data with a hyper plane. In initial testing these were classified at a rate of 100%.

On the other hand, consider a set of data that is more difficult to differentiate. Figure 4.9 and 4.10 is again a 3-dimensional plot of accelerometer data for five samples of two gestures on the same scale as the previous, "watch" and "line abreast". As you can see, the data are much closer together and in some cases overlapping in the plots. This makes the classification problem difficult and prone to error. Initial testing classified these at a rate of less than 100% in a very small sample size.

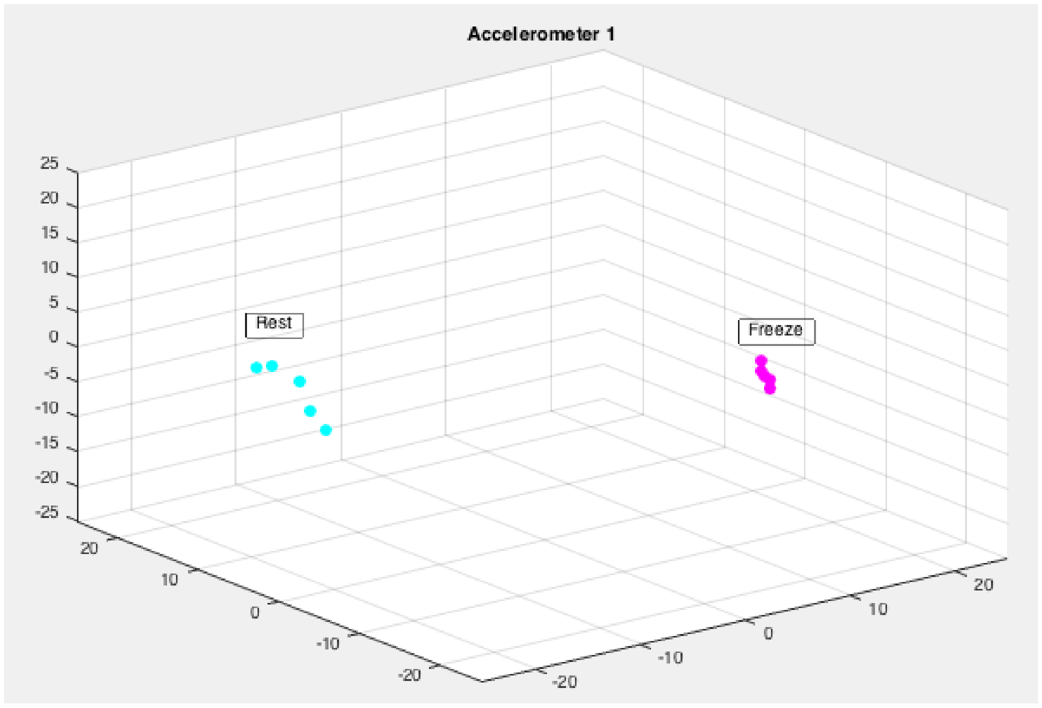


Figure 4.7. 3D Scatter of Accelerometer 1 for: Rest and Freeze

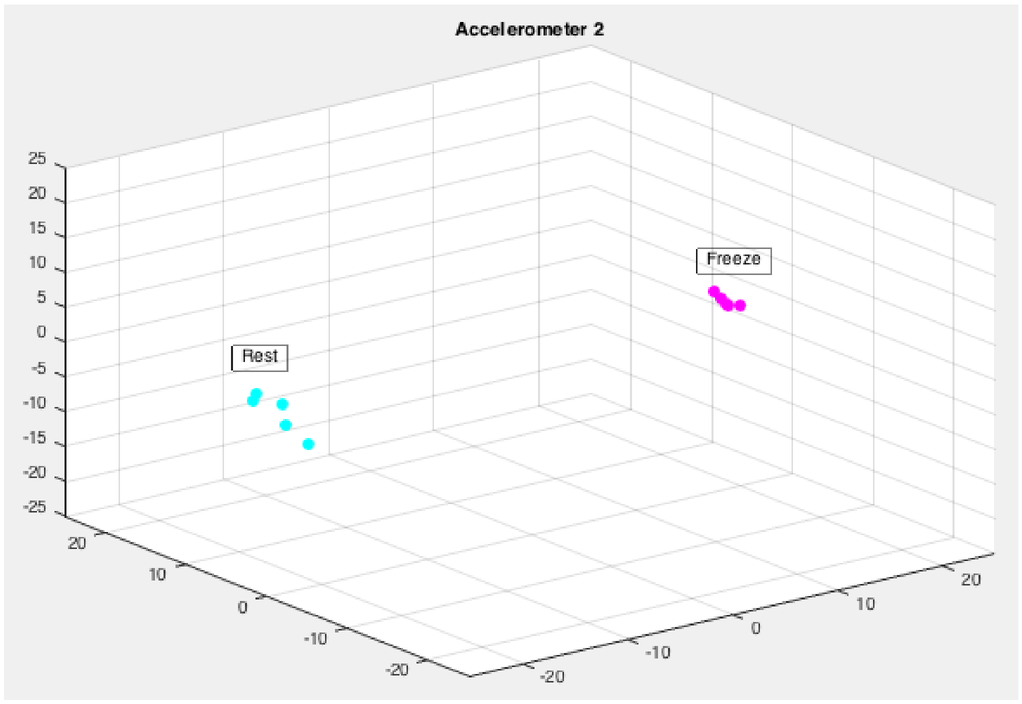


Figure 4.8. 3D Scatter of Accelerometer 2 for: Rest and Freeze

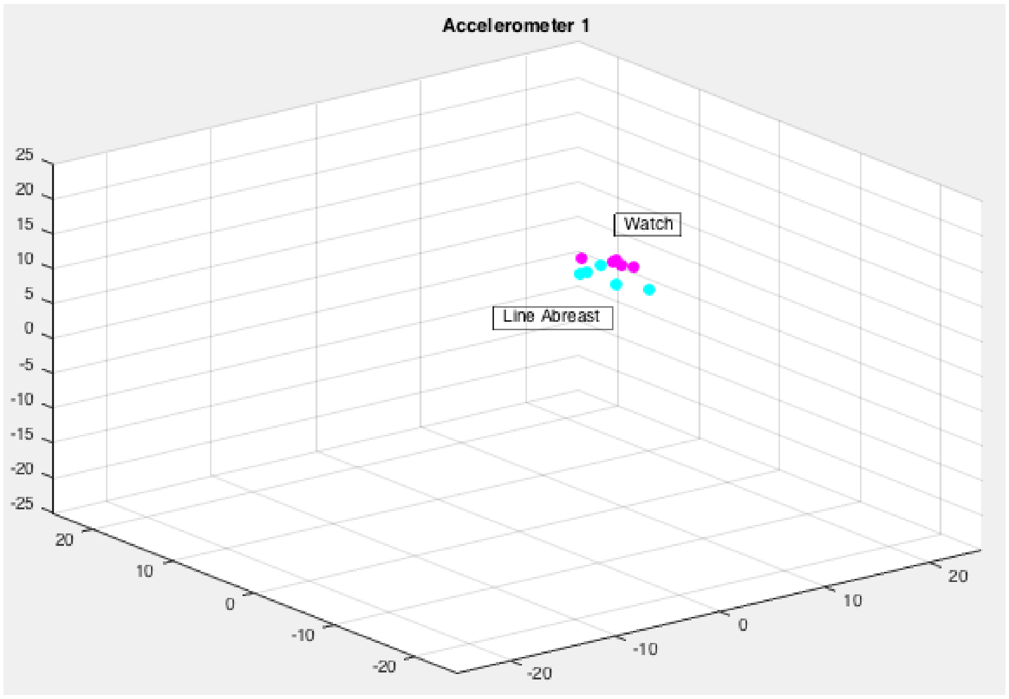


FIGURE 4.9. 3D Scatter of Accelerometer 1 for: Watch and Line Abreast

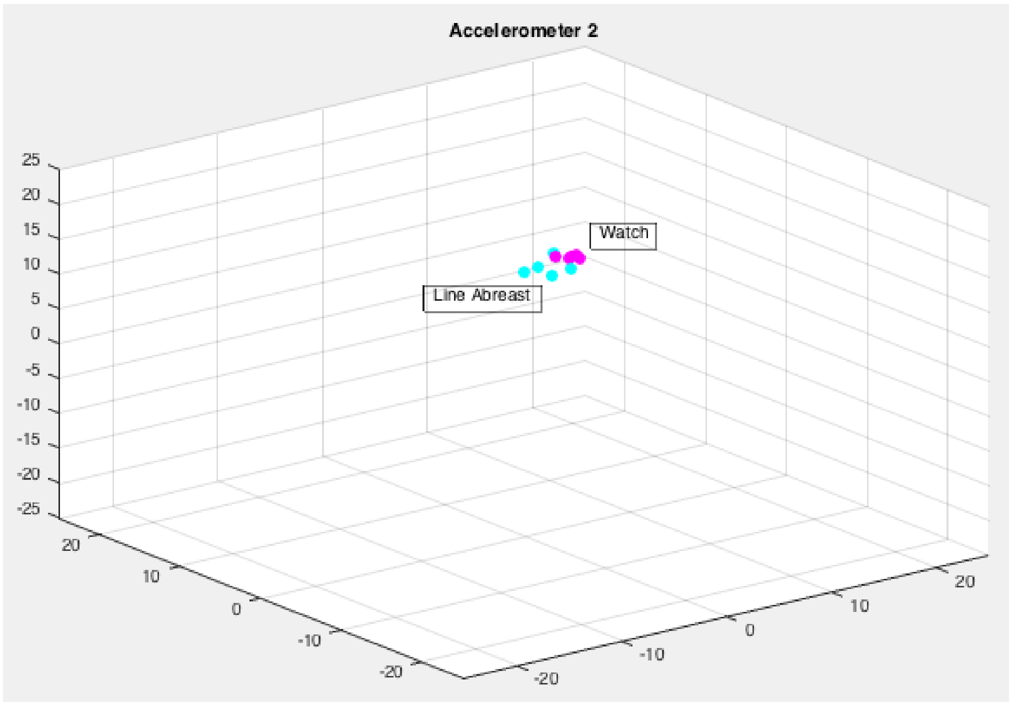


FIGURE 4.10. 3D Scatter of Accelerometer 2 for: Watch and Line Abreast

These figures highlight why it is necessary to accommodate a small subset of gestures into a third supplementary classifier. While not a complete picture of how the classification algorithm operates, it does show some of the problems with classifying these gestures. For the rest of this document the classification of these gestures and the supplementary classifier itself is referred to as the zone 1 supplementary classifier. The hand gestures classifier is referred to as the hand supplementary classifier.

## 4.4. Feature Extraction

This system uses a set of features that are both few in number and have a low complexity. It is efficient for use with most any classification algorithm. Because the system is designed to take only a small window of stationary data, it is only necessary to take the Mean Average Value of each sensor over the gesture window to get an accurate and efficient feature for use in the feature vector.

The mean average value (or MAV) is simply the mathematical mean of all the data points in the gesture window. The gesture window is a row matrix X, where n is the number of rows and represents the number of sensors (11 in the system), and m is the number of columns and represents each 10ms data point in the sample window. To get the single sensor feature $\mathrm{F_n}$ where n is the index into the row:

$$F_n = \frac{1}{m}\sum_{i=1}^{m} X(n,i) \tag{3}$$

The feature vector is constructed from all of the single sensor values, where n is the number of sensors in the gesture window:

$$F_v =< F_1 >< F_2 > ... < F_n > \tag{4}$$

For accelerometers, the single sensor value represents the force applied to a single axis. When combined into the 6 feature vector it represents the gravity profile of the sensor. For bend sensors the single sensor value represents the degree to which a single finger is bent. When combined in to the five feature vector it represents the orientation of the hand. The feature vectors are then sent on for classification.

## 4.5. Classification

The classification method used in this work is a multi-class Support Vector Machine (SVM). This machine learning algorithm is tested on our data with a linear and polynomial kernel for each of the primary and supplementary classifiers. The SVM must first be trained on existing labeled data. For this work, training was done with up to 20 samples of each gesture for a total of 560 samples across all three primary and supplementary classifiers. Once the SVM is trained, it is treated as a target function. New data can then be acted on by the target function and the SVM will classify the data.

## 4.6. Non-Gesture Tolerance

Non-gesture tolerance is defined as the ability of the system to recognize when a user has performed a gesture that is not a part of the defined gesture set and suppress this output. In a final product of this system, it is assumed by this work that some form of user feedback is present before transmission. This could potentially be in the form of visual feedback through the user HUD. This feature could also be designed to allow the user to correct mistakes or cancel a message and thereby directly achieve a level of non-gesture tolerance in the system. However, as an alternative to that, one could attempt to achieve non-gesture tolerance with smart system design and the inclusion of some simple post classification algorithms. These algorithms define how and when gesture would be permitted, or alternatively, when a gesture has violated some set of rules and should be suppressed.

Consistent with the operation of the windowing system, if a user performs a movement but does not stop moving long enough for the MOVE average to determine an active gesture window, then the movement is automatically suppressed as it will not create an active gesture window. This could potentially resist a wide range of possible unintended results based on a users actions while the system is active. An example would be if the user unintentionally moves to adjust some tactical gear or clothing, or has some other unforeseen need to use the hand and arm the communicator is attached to. As long as the user does not stop moving for very long, the result is automatically suppressed. However, if the user does stop movement long enough for an active window segment to be created, the system will send the data for feature extraction and classification. The resulting classified gesture will be the closest matching trained gesture in the support vector machine. It is here that some verification function of the input must be added to achieve another level of non-gesture tolerance.

The first step toward achieving this non-gesture tolerance is to define the upper and lower bounds of each dimension for each gesture present in the system. This is essentially the min and max of each dimension, for each training sample in the class. For example, if the classification result of the primary accelerometer classifier is the gesture "stop", we can compare the input feature vector with a stored set of values from the training database for that gesture. We get the min and max of all training samples of "stop" for each feature in the vector and we compare the input features to those min and max values. The training bounds T, where i is the class number of the gesture and F is a vector of all features in the column of the training matrix for that class is:

$$T_i =< F1_{min}, F1_{max} >< F2_{min}, F2_{max} > ... < Fn_{min}, Fn_{max} > \tag{5}$$

If the input features are outside the range of the training bounds T, then some decisions can be made about whether the gesture is valid or not. However the buffer zones must be added to the bounds. It is important to understand what the expected input of these feature values are. Since one can reasonably expect that a correctly performed gesture may actually fall just outside the min-max range for a subset of its features, a tolerance value must be added to the min-max range. Determining this value requires decoupling of the value from the stored training set. Since these accelerometers are producing real world values, one can determine what the range of expected values are by how they are used. Recall from earlier in this section that for active gesture windows the gravity profile of the accelerometers is being measured. This means that at any given time where an active gesture window occurs, the total acceleration being measured by the accelerometer will be 1g. Knowing this allows decoupling of the tolerance value from the stored values which can vary greatly in range and give a more stable tolerance. This leads to the definition of non-gesture tolerance rules.

For the first rule, the tolerance value is set to 40% of 1g or .4g. This value is added to the max and subtracted from the min to get a new min-max range for each dimensional feature. In the primary accelerometer classifier, there are six dimensions. Along with the tolerance value, if a single dimensional feature falls outside this tolerance range, then the gesture is rejected or suppressed. For the second rule, a value of 30% of 1g or .3g is used, and we reject a gesture if data for two different dimensions fall outside of this tolerance. These rules are applied simultaneously and if a single or both rules are violated then the gesture is rejected by the system. For stability, after the system rejects a gesture in this way, the system then enters the normal gesture timeout period to prevent further anomalous readings. Lastly, these rules are only applied to the primary accelerometer data and classifier.

## 4.7. System Summary

In this section I provide summary tables of the system starting with Table 4.2 which details the system parameters described in this chapter.

Table 4.2. System Parameters

| System Parameter | Value |
|---|---|
| Gesture Window | 100ms (10 samples @ 10ms) |
| Gesture Timeout Period | 800ms |
| Move Avg. Threshold | .03g |
| Move Avg. Window Size | 10 samples |
| Arm Gesture | Suppress Double Output |
| Hand Gesture | No Suppression |
| Tolerance Rule 1 | .4g, 1 feature |
| Tolerance Rule 2 | .3g, 2 feature |

Table 4.3 is a listing of all the gestures classified by this work and in which classifier they terminate. As shown in the table, some will terminate in the primary classifier while

others will eventually terminate in a supplementary classifier.

Table 4.3. The Primary Accelerometer Classifier

| Gesture | Final Classifier |
|---|---|
| 1: One | Supplementary Hand Classifier |
| 2: Two | Supplementary Hand Classifier |
| 3: Three | Supplementary Hand Classifier |
| 4: Four | Supplementary Hand Classifier |
| 5: Five | Supplementary Hand Classifier |
| 6: Six | Supplementary Hand Classifier |
| 7: Seven | Supplementary Hand Classifier |
| 8: Eight | Supplementary Hand Classifier |
| 9: Nine | Supplementary Hand Classifier |
| 10: Ten | Supplementary Hand Classifier |
| 11: You | Supplementary Zone 1 Classifier |
| 12: Me | Primary Accelerometer Classifier |
| 14: Listen | Primary Accelerometer Classifier |
| 15: Watch | Supplementary Zone 1 Classifier |
| 17: Stop | Primary Accelerometer Classifier |
| 18: Freeze | Supplementary Hand Classifier |
| 19: Cover | Primary Accelerometer Classifier |
| 21: Enemy | Primary Accelerometer Classifier |
| 22: Hostage | Primary Accelerometer Classifier |
| 23: Sniper | Primary Accelerometer Classifier |
| 24: Dog | Primary Accelerometer Classifier |
| 25: Cell Leader | Primary Accelerometer Classifier |
| 28: Line Abreast | Supplementary Zone 1 Classifier |
| 30: Control / Rest | Primary Accelerometer Classifier |

# CHAPTER 5

# RESULTS

## 5.1. Experimental Setup

To evaluate the gesture recognition system, all of the system functions were implemented in MATLAB and data was collected for each of the classifiers. During the process of collecting the data for the classifiers, the other system functions such as the effectiveness of the gesture windowing system and non-gesture tolerance were evaluated. These results are collected on a single male user and are considered user-dependent.

The primary classifier data was collected over a period of several days, during which the system was donned and removed several times. Manual observation was used to ensure that the accelerometers were oriented as close as possible to the same position for all collections. The gestures were then collected in a series of gesture strings. The gesture strings were performed for three gestures in series and a single sample of this process was approximately 420 individual data points which corresponds to 4.2 seconds of data. After each string was performed, the effectiveness of the gesture window system was calculated manually. If the system failed to correctly identify an active window segment, the entire sample was discarded and the failure was recorded. If the gesture window succeeded, the system was allowed to perform feature extraction and the sample was placed in the database.

The secondary classifiers data were collected over the same period of time. However, limitations in the digital collection device required that the primary and secondary data be collected separately. Specifically, the digital collection device only had enough ports to support either the data glove or the two accelerometers. Meaning that when wearing the bend sensor glove, one could not also wear the accelerometers and vice versa. The gestures where performed again for the secondary classifiers. This does not affect the outcome of classification results in any substantial way and the only limiting affect is the inability to test the accuracy of the secondary entrance ”freeze” gesture to all of the other hand gestures. Also because the gesture windowing system is attached to the data of the accelerometers, a manual window was used to capture the bend sensor data. All other attributes, such as window length, are the same.

## 5.2. Classification Results

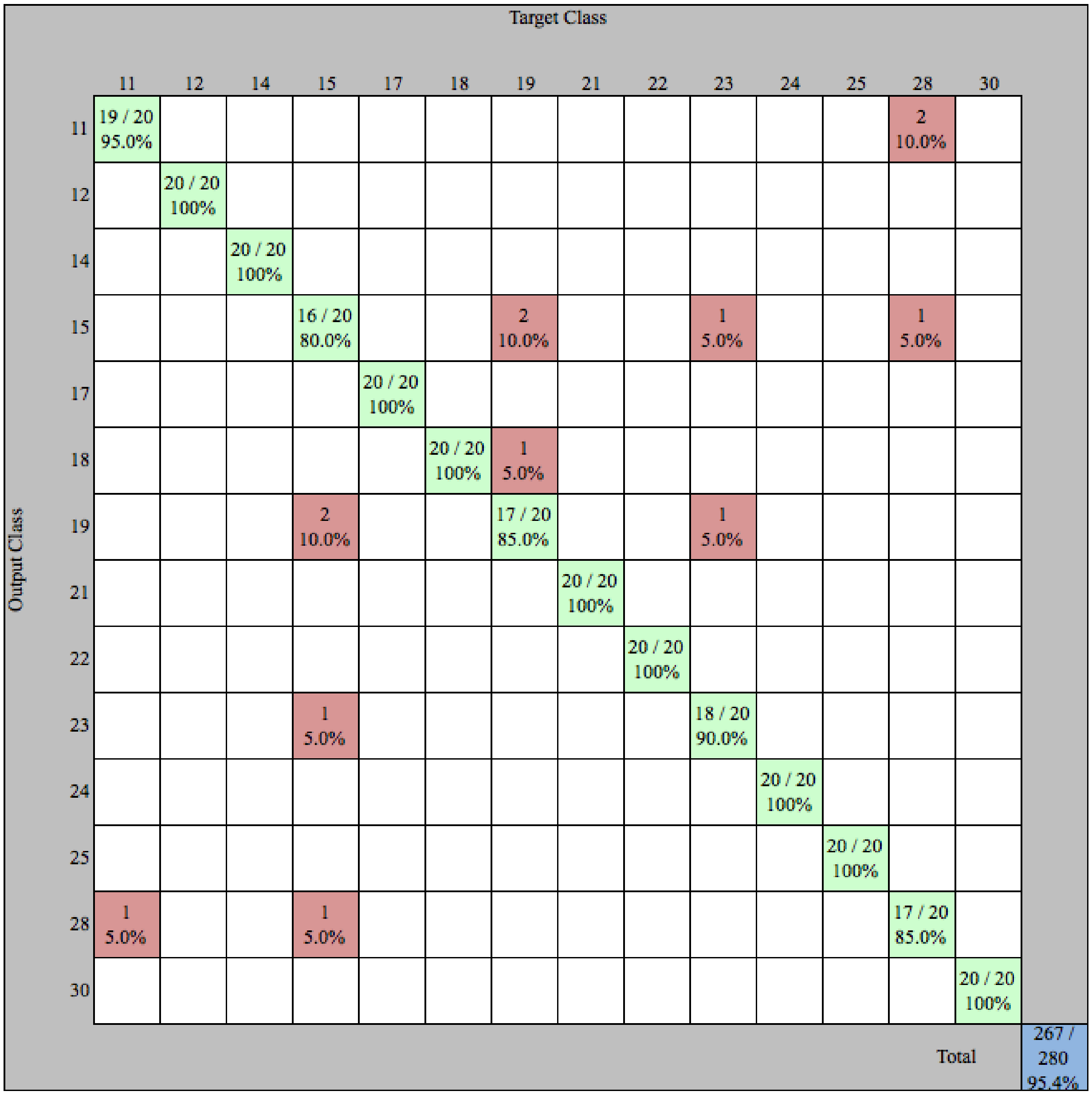
### 5.2.1. Primary Classifier

In the experiment, twenty samples of each gesture present in the primary classifier are captured for a total of 280 individual gestures. For the SVM, 75% of the data is used to train and 25% of the data is used to test. A 4-fold cross validation technique is also employed. This involves splitting the data into four parts. All four parts are trained and tested separately against the remaining three parts. This not only allows testing of all samples collected, but also gives a more complete result with higher confidence.

Figure 5.1 is a matrix that shows the classification results of the SVM with a linear kernel. The diagonal of the matrix represents correct classification. The entries not on the diagonal represent misclassifications. Finally the entry at the bottom right hand corner is the total classification result. A result of greater than 95% shows that most of the data is linearly separable. However, improvement of accuracy using a polynomial kernel is also shown. The figure 5.2 shows this result, which is greater than 96%. What these figures do not show is the use of the secondary classifier for zone 1. If the three gestures (eleven, fifteen, and twenty-eight) are excluded, the number of misclassifications reduces from eleven to five. However, this is only appropriate because of the accuracy of the secondary zone 1 classifier which is shown later.

### 5.2.2. Secondary Hand Classifier

For the secondary hand classifier, which includes gestures one through ten and twenty-eight, eighteen samples of the gestures were collected for a total of 196 samples. The characteristics of the bend sensor data proved to be completely linearly separable, therefore only the linear result is included. Again for this test, a cross validation technique (3-fold this time) is employed to complete the testing of the SVMs. Figure 5.3 shows the result of classification. The results show 100% accuracy in the recognition of gestures.



Target Class

Output Class

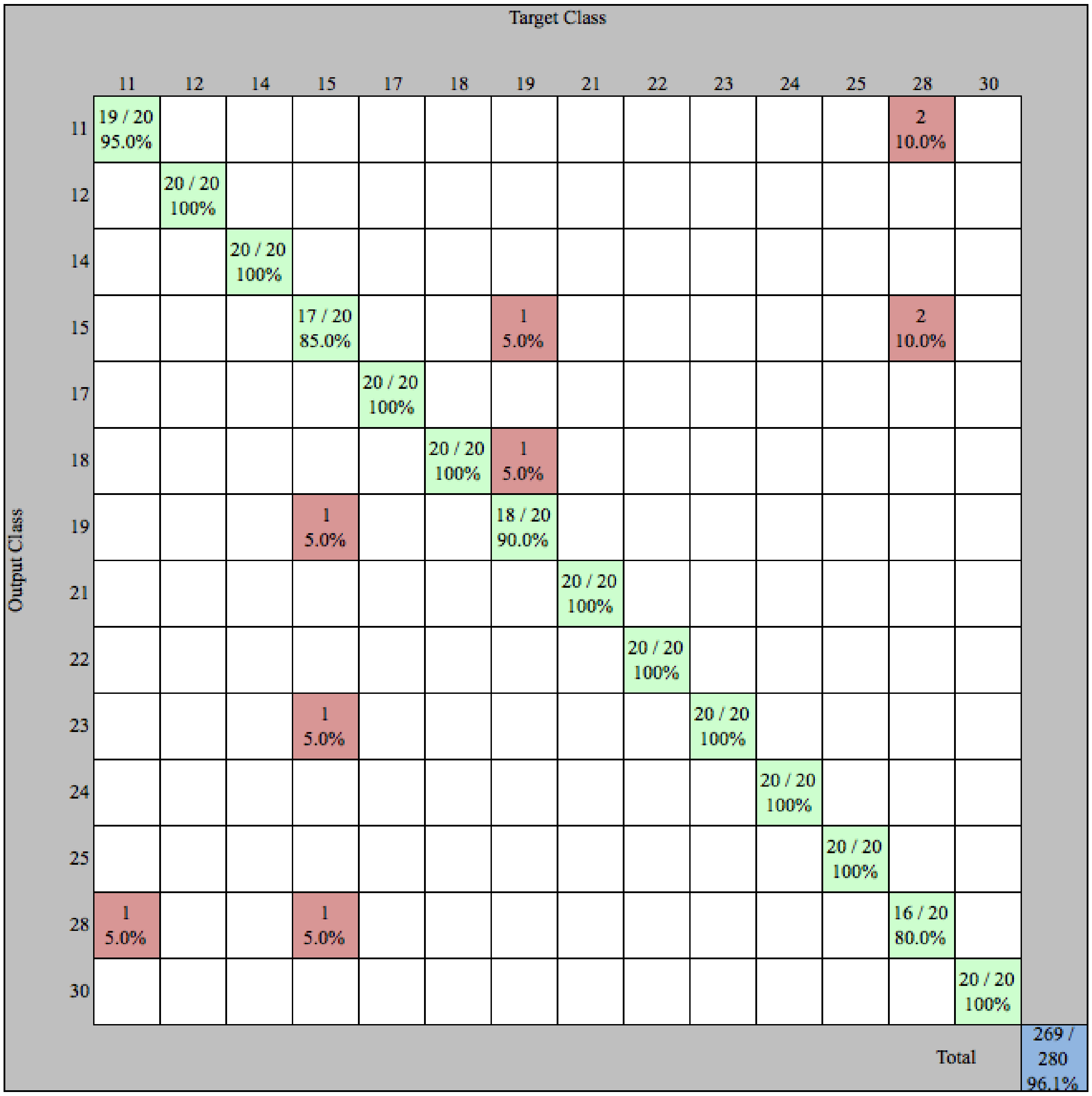
| | 11 | 12 | 14 | 15 | 17 | 18 | 19 | 21 | 22 | 23 | 24 | 25 | 28 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 19 / 20 95.0% | | | | | | | | | | | | 2 10.0% | |
| 12 | | 20 / 20 100% | | | | | | | | | | | | |
| 14 | | | 20 / 20 100% | | | | | | | | | | | |
| 15 | | | | 16 / 20 80.0% | | | 2 10.0% | | | 1 5.0% | | | 1 5.0% | |
| 17 | | | | | 20 / 20 100% | | | | | | | | | |
| 18 | | | | | | 20 / 20 100% | 1 5.0% | | | | | | | |
| 19 | | | | 2 10.0% | | | 17 / 20 85.0% | | | 1 5.0% | | | | |
| 21 | | | | | | | | 20 / 20 100% | | | | | | |
| 22 | | | | | | | | | 20 / 20 100% | | | | | |
| 23 | | | | 1 5.0% | | | | | | 18 / 20 90.0% | | | | |
| 24 | | | | | | | | | | | 20 / 20 100% | | | |
| 25 | | | | | | | | | | | | 20 / 20 100% | | |
| 28 | 1 5.0% | | | 1 5.0% | | | | | | | | | 17 / 20 85.0% | |
| 30 | | | | | | | | | | | | | | 20 / 20 100% |

Total 267 / 280 95.4%

Figure 5.1. Confusion Matrix of Primary Classifier with a Linear SVM Kernal

### 5.2.3. Secondary Zone 1 Classifier

Testing of the secondary zone 1 classifier was achieved in the same manner as with the secondary hand classifier. Eighteen samples of the three gestures (eleven, fifteen, and twenty-eight) were collected for a total of 54 samples. Again 3-fold cross validation is used. Figure 5.4 is the resulting confusion matrix. There is 100% classification accuracy which is the primary reason the secondary system is used to classify this subset of gestures. This in



Target Class

Output Class

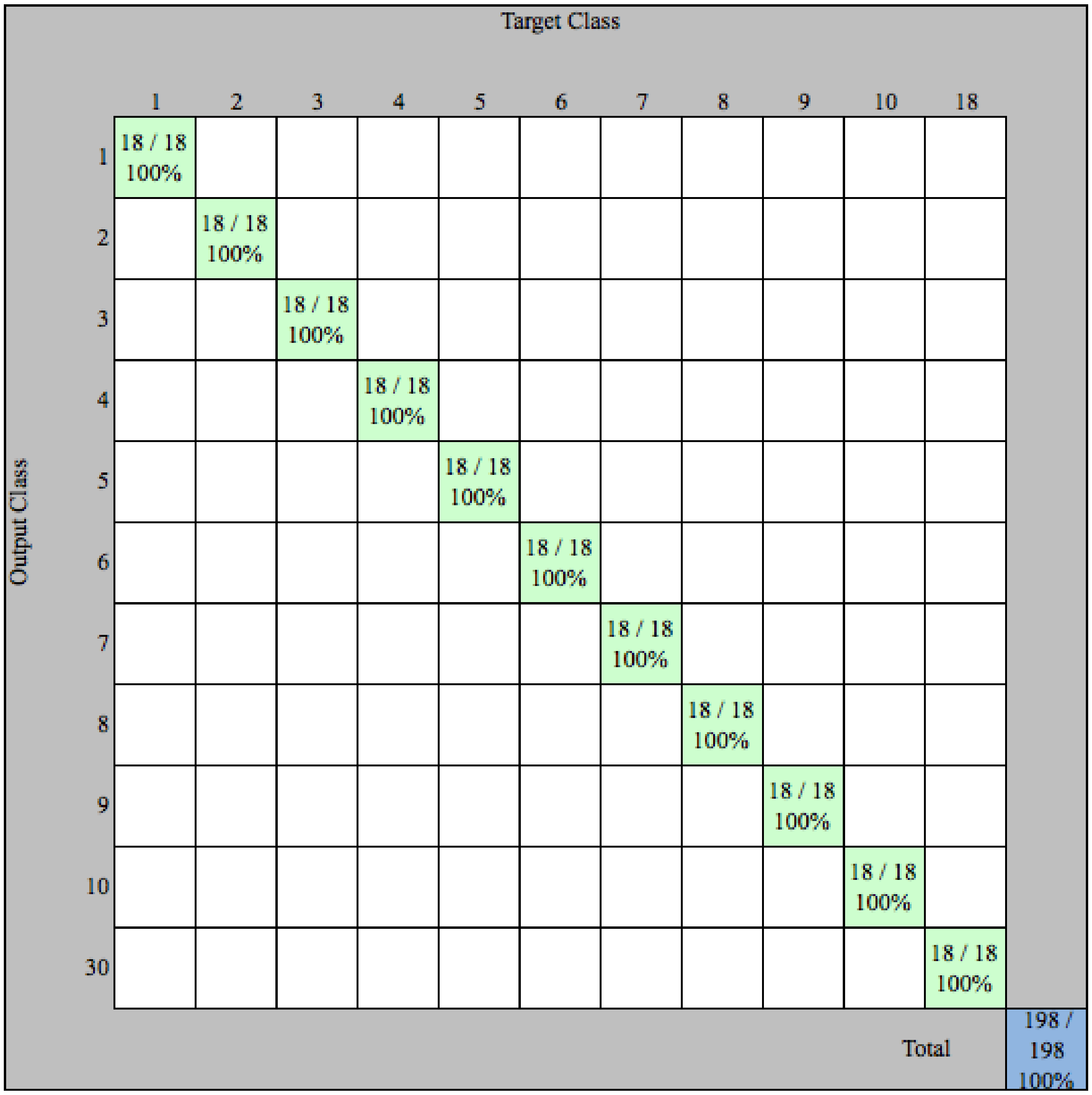
| | 11 | 12 | 14 | 15 | 17 | 18 | 19 | 21 | 22 | 23 | 24 | 25 | 28 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 19 / 20 95.0% | | | | | | | | | | | | 2 10.0% | |
| 12 | | 20 / 20 100% | | | | | | | | | | | | |
| 14 | | | 20 / 20 100% | | | | | | | | | | | |
| 15 | | | | 17 / 20 85.0% | | | 1 5.0% | | | | | | 2 10.0% | |
| 17 | | | | | 20 / 20 100% | | | | | | | | | |
| 18 | | | | | | 20 / 20 100% | 1 5.0% | | | | | | | |
| 19 | | | | 1 5.0% | | | 18 / 20 90.0% | | | | | | | |
| 21 | | | | | | | | 20 / 20 100% | | | | | | |
| 22 | | | | | | | | | 20 / 20 100% | | | | | |
| 23 | | | | 1 5.0% | | | | | | 20 / 20 100% | | | | |
| 24 | | | | | | | | | | | 20 / 20 100% | | | |
| 25 | | | | | | | | | | | | 20 / 20 100% | | |
| 28 | 1 5.0% | | | 1 5.0% | | | | | | | | | 16 / 20 80.0% | |
| 30 | | | | | | | | | | | | | | 20 / 20 100% |

Total 269 / 280 96.1%

Figure 5.2. Confusion Matrix of Primary Classifier with a Polynomial SVM Kernal

effect raises the accuracy of the entire system by augmenting the primary classifier.

The results of the secondary classifiers show that if there is even a small difference in the shape of the hand, such as in the case of gesture ”ten” and gesture ”fist”, the data will be differentiable by a linear SVM. This is also a result of proper placement of the bend sensors and an enclosing device (such as the glove) that keeps their positions fixed. If their positions in relation to the users hand were not fixed, the data would be less representative



Target Class

Output Class

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 18 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 18 / 18 100% | | | | | | | | | | | |
| 2 | | 18 / 18 100% | | | | | | | | | | |
| 3 | | | 18 / 18 100% | | | | | | | | | |
| 4 | | | | 18 / 18 100% | | | | | | | | |
| 5 | | | | | 18 / 18 100% | | | | | | | |
| 6 | | | | | | 18 / 18 100% | | | | | | |
| 7 | | | | | | | 18 / 18 100% | | | | | |
| 8 | | | | | | | | 18 / 18 100% | | | | |
| 9 | | | | | | | | | 18 / 18 100% | | | |
| 10 | | | | | | | | | | 18 / 18 100% | | |
| 30 | | | | | | | | | | | 18 / 18 100% | |
| | | | | | | | | | | | Total | 198 / 198 100% |

FIGURE 5.3. Confusion Matrix of Secondary Hand Classifier (Linear SVM)

of the trained gestures in the SVM and reduce the accuracy of the system.

### 5.2.4. Summary of Classification Results

Table 5.1 summarizes the results from all classification experiments. In this table the results of the classifiers in their native state are shown, i.e., before the supplementary classifiers are taken into account. The final classification accuracy is captured when the supplementary results are included. To calculate this result, we take the entire system into

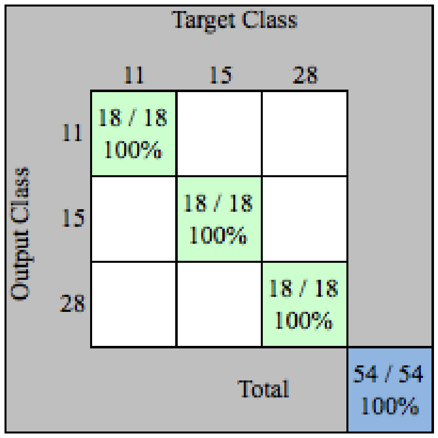


FIGURE 5.4. Confusion Matrix of Secondary Zone1 Classifier (Linear SVM)

account. When the user performs a hand or "freeze" gesture it is not counted in the primary classifier. In fact it is simply omitted from the primary classifier accuracy counts and added to the supplementary accuracy counts. The same is true for the zone 1 gestures. This means that the total gestures used for the primary classifier in the final value calculation drops from 280 to 200. Of these, only five were misclassified. The values 198 and 54 are then added to the total number of gestures correctly classified for the supplementary hand and zone 1 classifiers respectively. The total accuracy of the system is calculated at 98.9%.

TABLE 5.1. Summary of Classification Results

| Name | Attempts | Correct | % Accuracy |
|---|---|---|---|
| Primary Classifier (Native) | 280 | 269 | 96.1 |
| Primary Classifier (System Result) | 200 | 195 | 97.5 |
| Secondary Hand Classifier | 198 | 198 | 100 |
| Secondary Zone1 Classifier | 54 | 54 | 100 |
| Final System Classification Result | 452 | 447 | 98.9 |

## 5.3. Other Results

Along with the classification results, results of the gesture windowing system were tested. In the experiment, for every gesture string that was performed, a manual inspection of the gesture windowing system occurred. Manual inspection was performed for each gesture performed on the project to confirm that the gesture window occurred appropriately at the correct non-movement time (i.e., when the user stopped during performance of the gesture) and whether the system appropriately identified this as an active window for gesture recognition. This test is related to the movement threshold in the gesture windowing system. The cause of a failure in this situation is that the MOVE threshold was never reached. Slight movement while holding a gesture caused the moving window average to remain above the threshold and not activate the window. Raising the threshold in the system is a trade-off of performance versus usability. Lowering the value prevents false windows from occurring during a movement period. Raising the value provides additional tolerance and obviates the need for the user to hold a gesture ”completely still”. The threshold value used in all of the tests, .03g (which was experimentally derived from initial testing), is low enough that during all of the testing, not a single false window was recorded. However, there were a number of failures with the system not recognizing a correct gesture window because of slight movement while holding a gesture. This is a more desirable outcome than creating false windows because a false window is harder for the system to reconcile. The potential for error is much higher in a false window which has unexpected accelerometer values leading to features that may or may not be caught by the non-gesture tolerance rules. For this reason, a value was chosen that made it harder to meet the threshold, but gave no false windows. This failure occurred only 3.6% of the time. In these instances the user must repeat the gesture.

Manual inspection was also used to verify that gesture timeout periods were correctly performed by the user. A gesture timeout period is correctly performed if it produces one active window per movement of the accelerometers during a gesture. In simpler terms, one gesture equals one window. In these cases, a failure occurs because the user held the gesture for too long and the system incorrectly identified a second active window in the hold period. This failure occurred 2.1% of the time. These failures can potentially represent an overall failure of the system in certain cases. Refer back to the system overview where duplicate output suppression was discussed. The only cases which are not covered are hand gestures. In this case it is assumed that in a final version of the product, user feedback will allow an incorrect message to be cancelled. Taking the results from both windowing failures, the overall gesture windowing system accuracy was 94.3% in the testing environment.

Finally, the non-gesture tolerance function was tested. The classifier consisted of a full 20 samples for each training gesture and the rules for suppression were applied to the function?s output. The test consisted of two parts. The first part was a test against normal gestures and all gestures in the primary classifier. This test was intended to determine if the suppression system would erroneously suppress a correct gesture. The system improperly suppressed 10.5% of all attempts in this case with all of the suppression instances happening with rule 1. The system was also tested against gestures not in the base set shown in the CRE Chart. Two such gestures were attempted and the system suppressed them 83.3% of the time.

# CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

## 6.1. Conclusions

This thesis describes the implementation and testing of an accelerometer based gesture recognition system applied to a tactical communications application. The principal outcome of this work is the use of accelerometers to define gesture windows and the classification of accelerometer data to achieve gesture recognition of arm movements. The accelerometers are subsequently used as an entry point into a secondary classifier to recognize hand orientation. Other accelerometer based functions were also described and tested including non-gesture tolerance and, secondarily, supplementary sensors for the hand orientation portion of the problem.

This application requires sensing and classifying a wide variety of arm and hand movements. Based on the results achieved by the system, one can conclude that accelerometers are useful and appropriate for this class of applications. The gesture windowing system responded appropriately in over 94% of all test cases and achieved an overall classification success rate of 98.9%. The use of a support vector machine with a linear or polynomial kernel has also been shown to be advantageous for solving the problem with a minimum number of training samples.

While there is still work to be done (see the next section), the base system described here provides foundational capabilities for design and implementation of more robust gesture recognition systems. The project has satisfied several substantive requirements from the project?s sponsor and met challenges in the areas of signal analysis, machine learning, systems engineering, and embedded systems. Over the course of this work the author gained an understanding of feature extraction and statistical analysis in support of gesture classification, contributing to the advancement of HCI and gesture recognition systems. Future implementations can leverage these capabilities to meet evolving requirements for natural and useful human-computer interaction.

## 6.2. Future Work

The system described in this work solves a portion of the overall gesture recognition problem, but also presents opportunities for follow-on research:

The use of bend sensors for hand gesture recognition requires that the system incorporate a data glove that could potentially impact the dexterity of the wearer (for example, a soldier may have trouble firing a weapon while wearing the glove). However, it might be feasible to fabricate a glove using modern techniques and fabrics to minimize or eliminate a glove?s intrusive elements. Likewise, alternate sensors such as SEMG may be more ergonomically suitable. For these reasons the author chose to designate the hand gesture portion of the system solution as the supplementary classifier and sensors. The system has been described in such a way that any other sensors used in the supplementary portion will work with the primary accelerometer system. Future implementations could incorporate a new hand gesture sensor quite easily as long as the system is able to differentiate the bending of all five fingers on the hand. The new sensors would need their own portion of feature extraction and classification, which would likely be significantly different. However, the system could be easily modified to include it without affecting the primary accelerometer system. Additionally, wrist angle is actually captured by the accelerometers, so there is no need for the supplementary system to include it.

This system and the test results are meant to show the feasibility of the proposed solution and provide a basic description of the functions an end product would have. The product would be self-contained and likely part of an embedded system. A general purpose computer (like the iMac used in testing) and large programs with high overhead (like MATLAB) would not be running on the system. With this in mind, some portions of the code and data were analyzed to get some idea of computational and memory requirements. Running on the iMac with an Intel 2.7Ghz Core i7 processor, the SVM model took .0083 seconds to train in MATLAB. The training of the model is the most computationally complex part of the system, but is less important in terms of time to compute than other functions (like testing a result against the SVM model) provided it remains reasonable in terms of end user experience. Because the system is running MATLAB on top of a full operating system, much of the 8.3 milliseconds is overhead. The trained SVM model for 20 samples per gesture occupied 40k bytes of memory, and the training database itself with 20 samples per gesture and using double-precision floating point values occupied 4.4k bytes of memory. Directly coding the SVM and other functions of the system would be significantly more efficient in terms of memory, speed, and power. Consequently, the system could potentially be suitable for a small embedded processor with some SSD memory.

Another problem to be solved is that of capturing and classifying the gestures in the CRE chart that are identified as active motion gestures. While this is not in the current solution, it is possible to design the system to handle this function. Some changes to the timing and tracking of movement in each axis would be necessary to determine the beginning and end points of the gesture windows produced by active motions, as well as a separate feature extraction process and classifier. However, it is unlikely that these would constitute significant changes to the stationary gesture system.

# BIBLIOGRAPHY

[1] Md. R. Ahsan, Muhammad I. Ibrahimy, and Othman O. Khalifa, *Hand motion detection from emg signals by using ann based classifier for human computer interaction*, 4th International Conference on Modeling Simulation and Applied Optimization (2011), 1–6.

[2] Ahmed Akl, Chen Feng, and Shahrokh Valaee, *A novel accelerometer-based gesture recognition system*, IEEE Transactions on Signal Processing 59 (2011), no. 12, 6197–6205.

[3] Sarath Chandra Akumalla, *Evaluating approprietness of emg and flex sensors for classifying hand gestures*, Master's thesis, University of North Texas, 2012.

[4] Xiang Chen and et al., *Hand gesture recognition research based on surface emg sensors and 2d-accelerometers*, 11th IEEE International Symposium on Wearable Computers (2007), 11–14.

[5] Nello Christianini and John Shawe-Taylor, *An introduction support vector machines and other kernal-based learning methods*, Cambridge University Press, 2000.

[6] Tamilnadu Dindigul, *Hand gesture recognition based on shape parameters*, International Conference on Computing, Communications and Applications, Feb. 2012, pp. 1–6.

[7] Zoubin Ghahramani, *An introduction to hidden markov models and bayesian networks*, International Journal of Pattern Recognition and Artificial Intelligence 15 (2001), no. 1, 9–42.

[8] Tim Hill and et al., *Artificial neural network models for forecasting and decision making*, International Journal of Forecasting 10 (1994), no. 1, 5–15.

[9] E. Jovanov and et al., *Avatar – a multi-sensory system for real time body position monitoring*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2009), 2462–2465.

[10] ______, *defog – a real time system for detection and unfreezing of gait of parkinson's*

*patients*, Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2009), 5151–5154.

[11] Sarada Karlaputi, *Evaluating the approprietness of accelerometers in hand gestures*, Master's thesis, University of North Texas, 2015.

[12] Rachel King and et al., *Hand gesture recognition with body sensor networks*, Imperial College, 2005.

[13] Jianfeng Liu, Zhigeng Pan, and Xiancheng Li, *An accelerometer-based gesture recognition algorithm and its application for 3d interaction*, Computer Science and Information Systems 7 (2010), no. 1, 177–188.

[14] Brad A. Myers, *A brief history of human computer interaction technology*, ACM interactions 5 (1998), no. 2, 44–54.

[15] Jesus Suarez and Robin Murphy, *Hand gesture recognition with depth images: A review*, The 21st IEEE International Symposium on Robot and Human Interactive Communication, Sep. 2012, pp. 411–417.

[16] Jiahui Wu and et al., *Gesture recognition with a 3-d accelerometer*, 6th International Conference on Ubiquitous Intelligence and Computing (2009), 25–38.

[17] Renqiang Xie and et al., *Similarity matching-based extensible hand gesture recognition*, IEEE Sensors Journal 15 (2015), no. 6, 3475–3483.

[18] Ruize Xu and et al., *Mems accelerometer based nonspecific-user hand gesture recognition*, IEEE Sensors Journal 12 (2012), no. 5, 1166–1173.

[19] Tingfang Zhang and Zhiquan Feng, *Dynamic gesture recognition based on fusing frame images*, Fourth International Conference on Intelligent Systems Design and Engineering Applications, Nov. 2013, pp. 280–283.

[20] Xu Zhang and et al., *A framework for hand gesture recognition based on accelerometer and emg sensors*, IEEE Transactions on Systems, Man and Cybernetics 41 (2011), no. 6, 1064–1076.

[21] Xinshuang Zhao, Ahmed Naguib, and Sukhan Lee, *Kinect based calling gesture recog-*

*nition for taking order service of elderly care robot*, The 23rd IEEE International Symposium on Robot and Human Interactive Communication, Aug. 2014, pp. 525–530.