

ABSTRACT

Title of Document: EMG BIOFEEDBACK VIDEOGAME
SYSTEM FOR THE GAIT REHABILITATION
OF HEMIPARETIC INDIVIDUALS

Avi Bardack, Pratik Bhandari, James Doggett,
Maxwell Epstein, Nicolas Gagliolo, Steven
Graff, Erik Li, Elaine Petro, Mark Sailey, Natalie
Salaets, Benjamin Tousley, John Turner

Directed By: Dr. Pamela Abshire, Department of Electrical
and Computer Engineering

We report a novel approach to electromyographic (EMG) biofeedback for post-stroke hemiparetic gait rehabilitation, using a videogame. An integrated hardware/software system facilitates gameplay of *Tiger Woods PGA Tour 2004* in driving range mode by performing rehabilitation exercises. Real-time visual EMG biofeedback is provided as the patient performs exercises. Custom-built bioamplifiers and software collect, amplify, and filter the surface EMG signals from six lower-limb muscles, and score them by feature extraction. The ball is driven a distance proportional to each score. Exercises are scored by comparing the patient's EMG activation with target profiles. The user-friendly system is controlled by prompts on a personal computer. We envision two major benefits from this system. First, the biofeedback is offered in real-time, in a clear, intuitive form, and coupled with task-specific motions. Second, we hypothesize that adopting rehabilitation exercises to control a fun videogame will lead to greater adherence to the exercise regime, with accompanying improvements in gait.

EMG BIOFEEDBACK VIDEOGAME SYSTEM FOR THE GAIT
REHABILITATION OF HEMIPARETIC INDIVIDUALS

By

Team CHIP

Avi Bardack, Pratik Bhandari, James Doggett, Maxwell Epstein, Nicolas Gagliolo,
Steven Graff, Erik Li, Elaine Petro, Mark Sailey, Natalie Salaets, Benjamin Tousley,
John Turner

Thesis submitted in partial fulfillment of the requirements of the
Gemstone Program, University of Maryland, 2010

Advisory Committee:

Dr. Pamela Abshire, University of Maryland, Chair

Dr. José Contreras-Vidal, University of Maryland

Dr. Wes Lawson, University of Maryland

Dr. Scott Mathews, Catholic University

Dr. Sandra McCombe Waller, University of Maryland School of Medicine

Dr. Jill Whitall, University of Maryland School of Medicine

© Copyright by
Team CHIP

Avi Bardack
Pratik Bhandari
James Doggett
Maxwell Epstein
Nicolas Gagliolo
Steven Graff
Erik Li
Elaine Petro
Mark Sailey
Natalie Salaets
Benjamin Tousley
John Turner

2010

Acknowledgements

We would like to thank our mentor, Dr. Pamela Abshire, for her guidance and support. We'd also like to thank our partners from the UM School of Medicine Dr. Sandra McCombe Waller and Dr. Jill Whittall. We greatly appreciate the expert feedback from our panel of experts: Dr. José Contreras-Vidal, Dr. Wesley Lawson, and Dr. Scott Mathews. We are very grateful for the assistance from our librarian Mr. Bob Kackley, our graduate assistant Mr. Timir Datta, and others in the IBIS Laboratory of ISR. We'd also like to thank Dr. Alfred Haas, the Gemstone Program, including Dr. Wallace and Dr. Thomas, and most of all, our family and friends for their love and support over the past four years.

Table of Contents

Acknowledgements.....	ii
Table of Contents.....	iii
List of Tables.....	v
List of Figures.....	vi
Chapter 1: Introduction.....	1
Background.....	1
Current Methods and Trends.....	5
Research Objectives.....	7
Chapter 2: Literature Review.....	9
Electromyography and Signal Acquisition.....	9
Current State of the Art in EMG Sensing Systems.....	12
Current State of the Art in EMG Classification Algorithms.....	15
Current State of the Art for Hemiparetic Treatment.....	21
Current State of the Art in Videogame Rehabilitation.....	23
Chapter 3: Methodology.....	33
Stage Zero.....	33
Sensor System Design.....	34
Signal Acquisition System Validation.....	42
Stage One.....	42
Rehabilitation Exercises.....	43
Muscle Selection Criteria.....	44
Final Muscle Selection and Electrode Placement.....	45
The Improved, More User-friendly Integrated Sensor System and Hardware ...	47
Hemiparetic Test Subjects and Testing Procedure.....	49
Signal Conditioning.....	51
Stage Two.....	55
Videogame Interface.....	56
EMG Visual Feedback.....	57
Classification.....	61
Chapter 4: Results.....	69
Stage One.....	69
Stage Two.....	73
Determination of Features.....	73
Dynamic Time Warping.....	74
Qualitative Results.....	82
Chapter 5: Conclusions.....	86
Challenges & Limitations.....	86
Contributions to the Field.....	88
Chapter 6: Future Directions.....	90
System Efficacy Testing: Pre/Post-Training Functional Assessments.....	90
Randomized Controlled Trial in the Home Setting.....	93
iEMG Training.....	94
Control Group.....	94

Analysis.....	95
Hardware Improvement	95
Videogame Development.....	98
Assessment of Videogame	101
Appendices.....	102
Appendix I: Procedure for Testing	102
Appendix II: IRB Application and Addendum.....	111
Appendix III: MATLAB Data Acquisition Code (Final Version).....	143
Appendix IV: MATLAB Classification Code	162
Glossary	202
Bibliography	203

List of Tables

Table 1: Crosbie, et al sources: Empirical Research (Levels I and II)	25
Table 2: Crosbie, et al sources: Empirical Research (Levels III and IV)	25
Table 3: Crosbie, et al. sources: Non-empirical Studies (Level V)	26

List of Figures

Figure 1: An overview of the overall EMG interactive videogame system from patient input to videogame operation and feedback.	4
Figure 2: A view of the driving range of the EA Sports <i>Tiger Woods PGA Tour 2004</i> game for the PC.	5
Figure 3: Diagram of the motor unit.	10
Figure 4: Final schematic representing the behavior of the working amplification and filtering circuits.	35
Figure 5: Phase and magnitude response for the final circuitry. The frequency axis is on a logarithmic scale, and the magnitude response axis is in terms of total gain, which is the ratio between the output voltage and the input differential voltage.	37
Figure 6: Transient output responses at different frequencies. Green: 10 Hz; Red: 500 Hz; Blue: 1 kHz; Yellow: 2 kHz; Pink: 3 kHz. Note that the relative amplitudes agree with the gains shown in Figure 5.	38
Figure 7: The top copper, solder mask and silkscreen is shown in 7a, and the bottom copper, solder mask and silkscreen is shown in 7b. Note the unevenly-spaced holes designed for the USB receptacle at left, the solder pads for the quad op-amp chip U1, the single op-amp follower U2, and the 40-pin chip U3. POT1 is a placeholder for the final op-amp's feedback resistance, which eventually was implemented using a 5 M Ω potentiometer.	40
Figure 8: EMG sensor hardware. Left: power board with connectors for six USB cables and two 9V batteries. Middle: sensor board. Right: sensor board in place on the lower leg (tibialis anterior).	42
Figure 9: The methods for determining the electrode placement for each of the six targeted muscles; from left to right, top to bottom: tibialis anterior, gastrocnemius medialis, vastus medialis, biceps femoris, tensor fasciae latae, and gluteus medius (Hermens et al., 2006).	46
Figure 10: A muscular view of the leg muscles to be monitored, and the precise placement for each electrode pair for each muscle. The muscles are labeled in Figure 9.	47
Figure 11: Picture of Velcro elastic strap, which holds the sensor board.	48
Figure 12: Test subjects participating in the EMG data collection for Stage One. From left to right, the patients are performing the exercises: weight shift, knee lift, knee swing, step support. These pictures represent 3 of the 4 participants thus far. The extra person in the foreground in the two right panels is a team member who was coaching the study participants during the trial.	50
Figure 13: EMG data recorded from control subject EMG06, 4/29/2009. GUI program screenshots with (a) raw, (b) detrended and (c) filtered versions of data recorded from Exercise 3.	54
Figure 14: EMG data recorded from tibialis anterior during a step support exercise by patient 3. a) Raw data from paretic leg. b) Detrended, rectified, filtered data from paretic leg. c) Raw data from nonparetic leg. d) Detrended, rectified, filtered data from nonparetic leg. Data in (a) and (c) were amplified by a factor of 2500 and bandpass filtered from 10 Hz to 2 kHz by the EMG sensor boards. Data in (b) and (d)	

was additionally detrended, rectified, and filtered using 5 th -order lowpass Butterworth filter.....	55
Figure 15: Data collection and biofeedback displayed with <i>Tiger Woods PGA Tour 2004</i>	59
Figure 16: Slider bar when color change activated.....	60
Figure 17: Close up of LabVIEW GUI with signal graph. The videogame is hidden.....	61
Figure 18: <i>Different templates were used to detect features in the data.</i> Based on qualitative study of the data, it was found that most signals corresponded to having one, two, three or no peaks. Therefore, these templates were chosen.	66
Figure 19: Signal demonstrating 60 Hz Noise	67
Figure 20: Signal demonstrating minimal change in the EMG envelope profile before and after filtering out 60 Hz noise. Specifically, Fourier Transform peaks corresponding to 59-61 Hz were set to 0.	68
Figure 21: EMG02 data for hemiparetic patient, Exercise 2, May 20 th , 2009. Green line is for TFL, blue line is for biceps femoris.	70
Figure 22: EMG08 data for control patient, Exercise 2, May 20 th , 2009. Green line is for TFL, blue line is for biceps femoris.	70
Figure 23: EMG03 data for hemiparetic patient, Exercise 2, April 3 rd , 2009. Green line is for TFL, blue line is for biceps femoris.	71
Figure 24: <i>Alignment shows good matching of template and data.</i> Additionally, the best template was automatically chosen based on minimizing the root mean square of the signal and template difference.	75
Figure 25: <i>Exercise 3 patterns.</i> Here is an example of the pattern which was seen regarding Exercise 3. The solid line is the vastus medialis and the dotted is the biceps femoris. Although this example is from data collected from non-hemiparetic patients, most data did not look as clear. Nevertheless, most showed similarities to this pattern.	76
Figure 26: Typical spectrogram of control patient EMG signal for biceps femoris (top) and vastus medialis (bottom). Spectrogram is shown with corresponding raw EMG signal. Green line represents current enveloping algorithm result. Red line represents envelope generated using first principal component of the frequency distribution.	79
Figure 27: Typical spectrogram of a paretic patient EMG signal for biceps femoris (top) and vastus medialis (bottom). Spectrogram is shown with corresponding raw EMG signal. Green line represents current enveloping algorithm result. Red line represents envelope generated using first principal component of the frequency distribution.	80

Chapter 1: Introduction

Our research aimed to create a system that improves the physical rehabilitation of post-stroke patients by encouraging them to play a videogame that provides real-time feedback about their progress. This game was controlled by monitoring electromyographic (EMG) signals that reflect the patients' muscle activations in their weak limb. While the use of EMG in gait rehabilitation has been common in clinical settings, an interface with a videogame that is adaptable for home-use has not been previously implemented. The main components of the system include the EMG sensor hardware, the signal conditioning and classification software, and the videogame feedback.

Background

A stroke occurs when blood clots in the brain, resulting in damaged brain tissue and impaired brain function. About 795,000 individuals in the U.S. suffer a stroke every year. 87 percent of these cases are ischemic strokes, caused by lack of glucose and oxygen supply, and 13 percent are hemorrhagic strokes, caused from blood escaping the circulatory system (American Heart Association, 2009). Of these people, only 37 percent of hemorrhagic cases and ten percent of ischemic cases die within 30 days. 50 percent of those who survive suffer from some form of hemiparesis, relative weakness or neurodegeneration of one side of the body in comparison to the other. In order for the survivors to overcome these obstacles and regain effective use of their muscles, they require physical rehabilitation. In our

study, we focused on hemiparetic patients, and our system was intended for the rehabilitation of lower-limb functionality.

Over a lifetime, stroke recovery costs an average of \$104,048 per person, with the majority of costs coming from a combination of inpatient, rehabilitation, and secondary care costs (American Heart Association, 2009). The cost of stroke rehabilitation for hemiparetic patients is about 2.5 times higher than it is for nonparetic patients (Zorowitz, Chen, & Tong, 2008). However, these outpatient rehabilitation costs are only ten percent of those for inpatient treatment (Mauritz, 2004).

Perhaps the biggest challenge for post-stroke patients to overcome is community ambulation, the ability and willingness of stroke patients to walk with normal gait outside of a medical setting. In a recent study, community ambulation was viewed as extremely important in 91 percent of the stroke population (Lord, McPherson, McNaughton, Rochester, & Weatherall, 2004). While achievement of community ambulation could be clearly defined in clinical studies as based on gait velocity (Lord & Rochester, 2005), it is difficult for stroke patients, especially those with hemiparesis, to achieve community ambulation (Mayo et al., 2005; Michael, Allen, & Macko, 2005; Wade, Wood, Heller, Maggs, & Langton Hower, 1987; Goldie, Matyas, & Evans, 1996).

Since the focus of physical rehabilitation is on retraining neural networks and strengthening muscles rather than simply moving body parts, our rehabilitation system used electromyography to monitor specific muscle activations. Every time a muscle extends or contracts, it generates an electric potential that can be measured

with surface sensors on the subject's skin. These EMG signals contain information about which muscles are activating, when they are activating, and to what extent they are being activated. This information was used by a classification algorithm to determine how correctly the subject performed a particular exercise.

Displaying these signals to the patient is a form of biofeedback. By using electromyographic information as biofeedback, the patients could see which muscles they were activating and relative activation levels between muscles. Biofeedback has been proven as an effective technique in rehabilitation for improving muscle recruitment and even reforming neural connections to compensate for those lost due to stroke (Glanz et al., 1995).

Our system aimed to take this biofeedback method one step further by instead of just displaying the signals, incorporating them into an interactive videogame. Based on the accuracy of the user's performance on the exercise, our system reacted with a corresponding performance by the character in the videogame. The user was also able to see sliding bars in real-time that indicated the user's level of muscle contraction for the exercise's two most relevant muscles, relative to the desired level of contraction. The user immediately saw how much higher s/he had to make the bars move, which indicated how close s/he is to a healthy performance.

The following chart in Figure 1 shows the sequence of events that took place in our system's feedback loop. The patient performs an exercise, and the resulting voltages from the patient's muscles are read by the system electrodes. These signals are immediately amplified and filtered by the sensor boards before passing through cables to the data acquisition card in the system's computer. On the computer, the

signals are processed in several stages including further filtering, detrending, feature extraction, and possible classification. A computer program determines the patient's score for that exercise, and controls the videogame feedback appropriately. Once the patient sees the result in the game, she knows how well she did and how to improve. With this knowledge she repeats the exercise, and the feedback loop continues. An example of the videogame screen is shown in Figure 2. The realtime feedback bars showing target muscle activation levels is to the left, while the videogame driving range with Tiger Woods avatar is to the upper-right. In this version there are various other boxes and controls for adjusting the difficulty, as well as for debugging purposes.

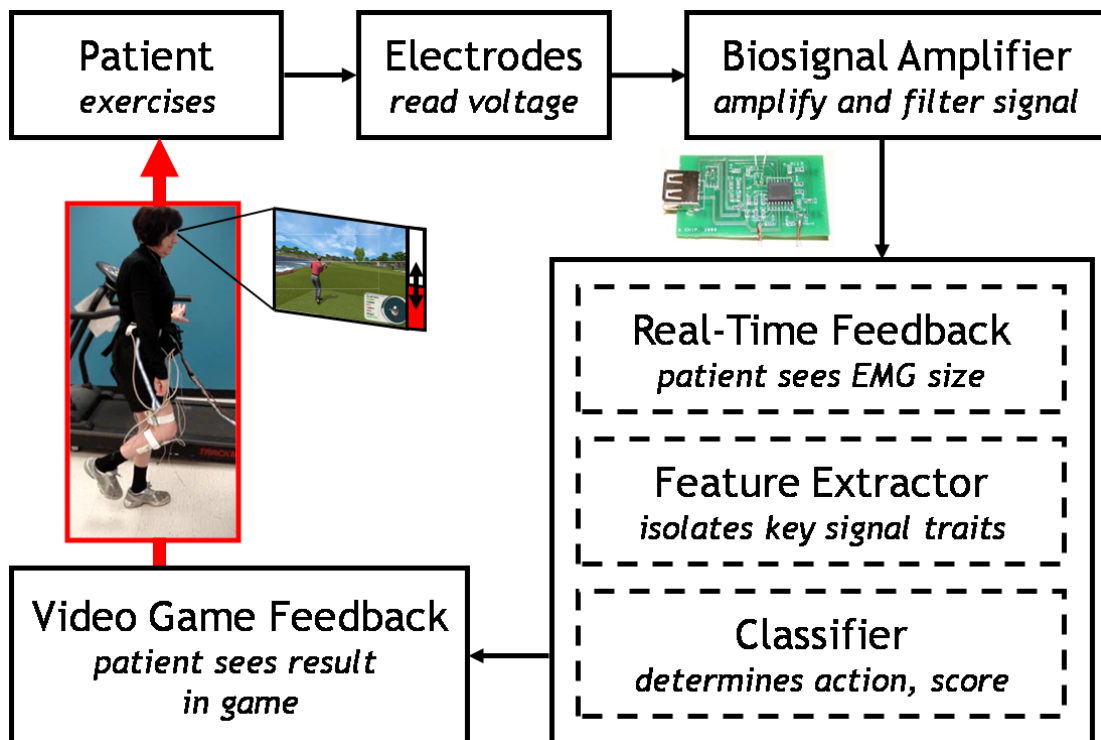


Figure 1: An overview of the overall EMG interactive videogame system from patient input to videogame operation and feedback.

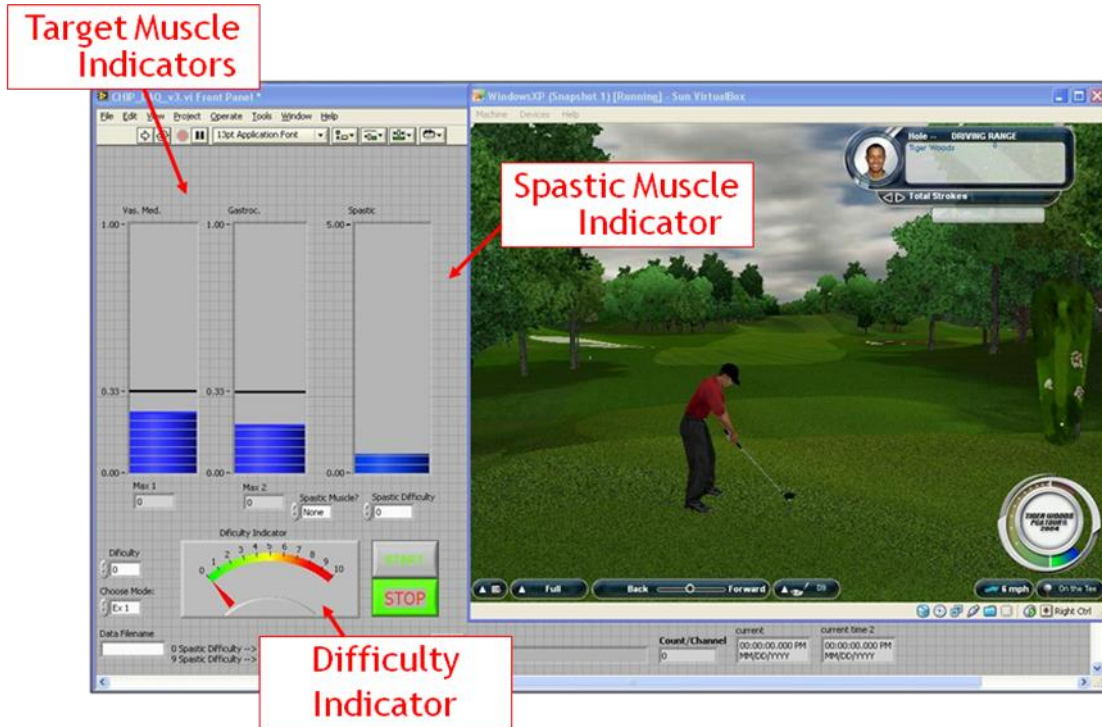


Figure 2: A view of the driving range of the EA Sports *Tiger Woods PGA Tour 2004* game for the PC.

Current Methods and Trends

EMG signal acquisition and classification is a rather developed field, with investigations about the signals beginning in the late eighteenth century (Basmajian, 1979). Needle electrodes, electrodes placed internally via surgery, and surface electrodes are all used today. EMG signals have been used to control many devices such as prostheses and robot arms, and also have other applications in medicine because of their ability to measure muscle activation (Artemiadis & Kyriakopoulos, 2008; Fukuda, Tsuji, Kaneko, & Otsuka, 2003; Shenoy, Miller, Crawford, & Rao, 2008; Song, Tong, Hu, & Li, 2008; Veneman et al., 2007). Consequently, many physical therapy centers possess EMG sensors, even though the sensors are often bulky and expensive (Hughes, 2007).

EMG-based rehabilitation is commonly used as a method to improve gait in post-stroke patients. Currently, there are several methods to achieve community ambulation – namely gait symmetry, prosthetic or assistive devices, and robotics. Gait symmetry methods have traditionally revolved around utilizing EMG as a diagnostic tool to measure muscle strength. These methods, which have focused on eliminating irregular contractions, did not prove to make hemiparetic stroke patients any more ambulatory after clinical treatment (Mauritz, 2004). Robotic therapy has also emerged as a promising stroke rehabilitation tool. While robotic prosthetics, which compensate for the lack of neural activity in affected regions during gait, have been proven relatively effective for rehabilitation, they do not present enough of a clinical and occupational advantage to be used outside of a controlled setting (Johnson, 2006; Hornby et al., 2008). Furthermore, traditional assistive devices, such as walkers, canes, and ankle-foot orthoses (AFOs) have remained as the primary means of rehabilitation, even with the advent of EMG sensing capabilities (Mauritz, 2004). For an EMG-based rehabilitation to be successful, it must revolve around retraining and re-strengthening target muscles and their associated neurons, rather than focus on gait symmetry.

The feasibility and effectiveness of videogames have recently become more prominent in the physical therapy field, likely inspired by the popularity of the Nintendo Wii gaming console (Johnson, Ramachandran, Paranjape, & Kosasih, 2006). EMG-controlled videogames have been attempted (Dipietro et al., 2005), but the area remains largely unexplored, especially for lower-limb rehabilitation. For this

reason, we believe our project contributes significantly to this developing field. Our project was guided by one overarching research question:

How can a custom EMG controlled videogame be designed and built to improve the efficacy of gait rehabilitation for hemiparetic stroke patients?

Research Objectives

Although EMG biofeedback has been used successfully to retrain specific isolated muscles to fire appropriately (Aiello et al., 2005), training of isolated movements often demonstrates little transfer into improvements in the activities of daily living (Moreland, Thomson, & Fuoco, 1998). Important factors that have been noted in reports of successful EMG biofeedback in the literature include: real-time feedback which occurs as the task is conducted, training which performs functional task-related actions rather than isolated movements, and biofeedback that motivates patients and engages their attention in the rehabilitation task (Huang, Wolf, & He, 2006). We aimed to create a system that would allow patients to play an interactive videogame, controlling an on-screen avatar by performing rehabilitation exercises. To accomplish this task, we developed a portable EMG sensor system to gather data from patients undergoing rehabilitation. The data was interpreted to aid the development of our signal processing and classification algorithms by providing a standard for comparison against patients' EMG signals while performing particular tasks. The sensors initially used to gather data were later integrated with this classification software and a videogame to create a complete biofeedback rehabilitation system. Finally, we investigated the performance of the system in conducting physical therapy of stroke victims.

The essential, desired characteristics necessary for our classification subsystem were fast runtime, and robust and accurate signal classification. Accuracy in signal classification is the ability to consistently map a particular voluntary motion to a particular signal. Robustness means that the classification system will produce equally valid results across a diverse user population.

Chapter 2: Literature Review

Electromyography and Signal Acquisition

The discovery that the contraction of muscles could be initiated electrically and that an electric voltage could be measured across contracting muscles first came in the late eighteenth century after a series of experiments by Italian biologist and physician Luigi Galvani (Basmajian, 1979). However, it was not until the twentieth century that improved methods for the detection of small electrical signals permitted investigation into the electrical properties of muscle contractions (Basmajian, 1979). Since then, there have been an increasing number of papers written regarding the topic of electromyography, the study of the relationship between muscles and electricity. An understanding of this literature contributes to the design of an EMG signal classifier because it provides valuable insight into EMG signal acquisition, EMG signal waveform characteristics, current methods of correlating EMG signals to muscle tension states, and existing hardware implementations of EMG signal classification and their applications.

A clear understanding of the EMG signal and its different characteristics is essential for signal acquisition and classification. It is first important to understand how the biological process of a muscle contraction invokes an electrical potential and how specific muscle actions can be identified as the cause. EMG signals represent the electrical potential across a motor unit as specific muscles contract. The motor unit consists of: the nerve cell body; the long axon running down the motor nerve; its terminal branches; and muscle fibers (Basmajian, 1979). Muscle fibers are

responsible for contraction, wherein small bundles of muscle fibers contract simultaneously, corresponding to the activation of a particular muscle. A chemically induced localized depolarization of the nerve cell membrane, or action potential, propagates along a nerve fiber and eventually activates a muscle fiber by an amplified cascade of net ion movements that results in an amplified voltage in the muscle.

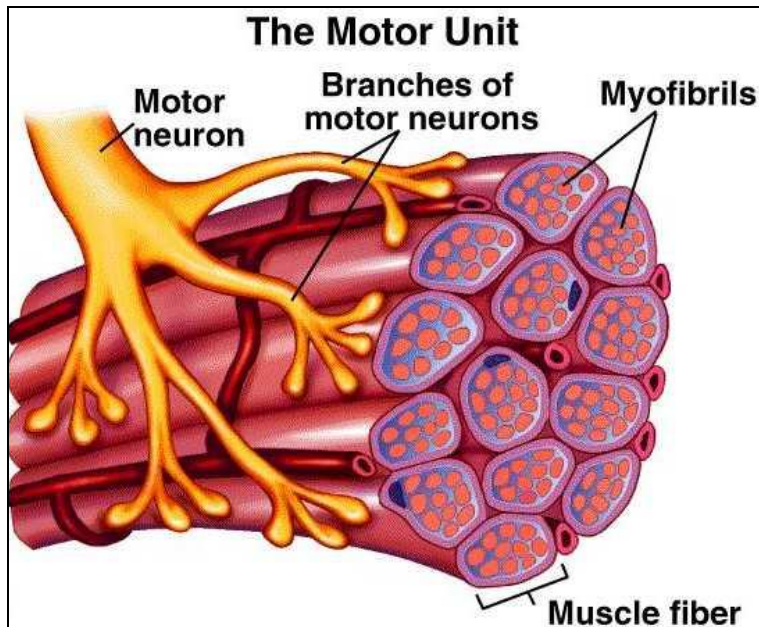


Figure 3: Diagram of the motor unit.

A surface electrode can measure a signal, which is the combination of all the muscle fiber action potentials occurring underneath the skin where it is placed. This combination of electrical potentials from a particular motor unit is known as the motor unit action potential (MUAP). The duration of the electrical potential generated by just one twitch is between 5 and 12 milliseconds and usually varies between 0 and 10 millivolts (mV) prior to amplification (Basmajian, 1979). Larger motor units generally correspond to higher signal amplitudes and therefore to larger recorded potentials. It has been proven that the amount of force that a muscle is

capable of producing is directly proportional to its cross-sectional area, though not directly related to its length (Lamb, 1992). This information will be important when considering which muscle groups to focus on and which groups will emit the strongest and most easily resolvable signals. Stronger muscle contractions also correspond with higher rates of motor unit firing, with the upper limit of activation lying around 50 Hz (Basmajian, 1979). In addition to these physiological considerations, we must also consider the documented effects of age, fatigue, handedness, and gender on these signals. The recorded MUAP is a function of muscle fiber characteristics, orientation of muscles with regard to electrodes, and the specific electronic measurement equipment used (Lamb, 1992).

Of equal importance to the signal acquisition process is electrode preparation and placement, which can cause significant variations in the recorded EMG signal (Basmajian, 1979). One general assumption in terms of muscle selection is that smaller muscles usually require smaller electrodes to measure the signal, with electrode diameter ranging anywhere between 1 and 5 millimeters (Soderberg, 1992). There have been several studies to determine the optimal arrangement for electrodes in order to obtain the best recorded signals. In general, electrode contacts should lie parallel to muscle fibers, and should have a center-to-center distance ranging between 2 and 10 mm (Soderberg, 1992). Electrodes are also most commonly arranged in a bipolar configuration, which includes two electrodes dedicated to detection and a third electrode that serves as a ground at a different electrically neutral location, such as a bony prominence that is unlikely to detect any EMG signals (Soderberg, 1992).

One area still in question within electromyography literature concerns the test-retest reliability of electrode recording. Although one should expect slight variance within different tests, surface electrodes demonstrate higher test-retest reliability than fine wire electrodes (Soderberg, 1992). This variability was therefore essential to consider in regards to the algorithm and signal recognition, because a less consistent signal requires a more flexible signal classifier to discern intended motions. Regarding skin and electrode preparation, conduction gel is generally applied to the skin after thorough cleansing with alcohol, and electrodes are then attached by means of a double-sided adhesive washer (Soderberg, 1992). A further investigation of the effects and methods for electrode placement will be conducted in order to achieve the highest possible signal quality. For example, to simplify the process of searching for the strongest signal associated with certain hand motions, Konishi et al. (2007) developed an elastic band to be worn around the forearm, embedded with multiple channels of electrodes.

Current State of the Art in EMG Sensing Systems

EMG systems are usually bulky affairs consisting of multiple sensors being placed on the skin with wires connecting the sensors to a central hub, usually worn in a backpack or other storage device placed on the body. These systems are usually rather awkward and often very expensive, with estimated costs at around 2000 dollars per channel (Hughes, 2007). Typical electromyographic sensor systems consist of some form of electrode and signal amplifier. An analog-to-digital (A/D) converter is needed to feed the signal into a computer if signal processing is necessary. The computer does all of the signal processing and feature extraction. To reduce the need

for bulky desktop or laptop interfaces, some more advanced systems implement signal processing algorithms on microprocessors. In 2006, a group from Japan introduced a prototype for a “Wearable EMG Sensor” system that measures the two forearm muscles to detect a grasping movement (Nishida, Kawakami, & Mizoguchi, 2006). This system records the signal with an electrode and then passes it through a 5 Hz high pass filter, an amplifier, and finally a 3 Hz high pass filter, to cut out low frequency noise. The signal then passes through a full wave rectifier before entering the microprocessor, which does the A/D conversion.

Although medical wireless sensor systems have not yet made it into practical use, EMG sensing systems, especially for rehabilitation applications, are quickly moving in this direction. In 2005, a group from the University of Alabama developed a prototype wireless sensor system, coined “Wireless Body Area Network,” to be used with computer-assisted rehabilitation (Jovanov, Milenkovic, Otto, & de Groen, 2005). The wireless sensors communicate with a computer that performs real-time data processing and provides feedback to the user. The computer can further upload this data to medical servers to store it in research databases or a patient’s medical file. This system was designed to combat two of the major problems with sensor-based rehabilitation today. The first problem is the issue of bulky wires, which inhibit the patient’s movement and comfort. The second problem is the need for extensive physical therapist supervision that is simply not available. These wireless sensor rehabilitation systems, which utilize off-the-shelf sensors, may be able to reduce the amount of direct supervision by a physical therapist.

Another wireless surface EMG system has recently been developed using the ZigBee wireless standard (Hughes, 2007). Each node has a gain of 300, a low pass filter with a 500 Hz cut-off, and a microcontroller with an internal 10-bit ADC. The authors claim that their system could reduce the costs of each channel by 50 percent or more. They also state that the system can accommodate up to 256 channels, although only two have been tested so far.

Recently, Noraxon Corporation's ZeroWire™ wireless EMG system has arrived on the market. The system enables more remote operation, allowing data collection from up to 80 meters away from the subject. The system shows a sampling range of EMG signals from 10 to 1000 Hz for each of its 32 channels, and has a battery life of up to eight hours (Noraxon, 2010). The ZeroWire™ system has additional gait analysis features for rehabilitation-specific applications (Noraxon, 2010). However, as a complete EMG system, it was well beyond our budget range, and was released after we had built our own wired EMG system.

Another innovative design to reduce the need for inconvenient wires involves implementing EMG sensors into conductive fabrics. A group in Japan has created the "TextileNet" cable-free network system using two layers of conductive fabric with a shielding layer in between. The prototype they created can transmit data at 9.6 kbps and supply 3 watts of power to each device. Another group has also begun working on surface electrodes capable of reading EMG signals through a thin insulator, specifically cloth. They were able to obtain a synchronized signal of comparable quality to those obtained via individual electrodes, albeit with a lower signal-to-noise ratio (Ueno et al., 2007).

Current State of the Art in EMG Classification Algorithms

There are several different methods currently in use for decomposition and classification of surface EMG (sEMG) signals. One of the simplest techniques for evaluating the activity of multiple muscles during a complicated activity is to use a simple “on/off” classification for each muscle involved, a technique that has been in use for over a decade. Pelland and McKinley (2004) accomplished this by subtracting the mean from each collected signal and then performing a full-wave rectification. On-off activity was determined by comparing 50-ms windows of rectified data to a baseline noise threshold, computed using data collected in a motionless initial trial. The noise threshold was computed individually for each subject using one-second bins across a six-second trial. Pelland and McKinley found that this provided a relatively stable baseline for their noise threshold, and moreover was largely consistent across patients. Thus transformed into “on” or “off”, the state of all muscles involved in a given action can be implemented in a vector to see which muscles work together for a given activity.

EMG signal acquisition yields an electrical signal represented as an amplitude or voltage in the time domain, which must then be processed to yield an output corresponding to a movement or muscle contraction. Signal processing is a vast field, with applications at every level of science and engineering. The scope of this literature review precludes mention of many signal processing techniques which might potentially have application in the study of EMG. However, many of the signal processing techniques that have been used effectively for the purpose of EMG classification are mentioned here.

In general, signal processing of EMG can be split into several phases: conditioning, clustering or criteria selection, feature extraction, and classification. The raw signal is an aggregate of the electrical potential changes caused by the contraction of the targeted muscle cells, and of the noise components from various electrical devices and other, untargeted muscle groups. Therefore, it is necessary to condition the signal to minimize the effect of the noise components before the signal can be classified, and to better isolate the data of interest. Then, depending on the characteristics of the phenomena studied, a variety of techniques may be used to identify distinct groups of signals that map to an action, and to classify individual signals as belonging to a specific group.

Ideally, an acquired signal contains only information relevant to identifying it to a particular class. However, in practice, it is necessary to use filters to remove unwanted noise from the data, such as electrical interference, motion artifacts, and crosstalk between muscle groups. At the most basic level, a filter acts by attenuating certain frequencies within the signal (Kuruganti, 1995). Often, this filtering is either implemented within the signal acquisition hardware or through the use of the Fourier transform, which converts the time domain of a signal to the frequency domain of the signal, so that the signal is re-expressed as the sum of sinusoidal waves. Particular frequencies can then be easily filtered out by zeroing the ranges corresponding to noise.

In developing an algorithm for classifying the signal, it may be advantageous to remove information from the signal that is not noise, but that does not help to distinguish between the different target classes. Reducing the space of the dataset in

this way can make data-intensive algorithms more efficient or help to identify patterns in the data. One algorithm that accomplishes this task is principal components analysis (PCA). In PCA, the eigenvalues of a covariance matrix are used in order to determine the most distinguishing features of the signal. The less distinguishing characteristics can then be discarded in order to reduce the dimensionality of the data with minimal loss in resolution. Güler and Koçer (1995) used PCA to determine the most distinguishing components of the EMG frequency domain.

After the data has been conditioned, signal classification can begin. One of the easiest and most widely used methods of classifying EMG signals is based on amplitude thresholding. A threshold value is either manually set after looking at the data or automatically set from previously collected data. Abbink et al. (1998) took the latter approach and used characteristics of the amplitude distribution of the entire signal to detect repetitive muscle contractions associated with the chewing of food. Because the EMG signal is bipolar, thresholding of the raw signal resulted in very short bursts of activity as peaks in high-amplitude sections of the signal crossed the threshold. In most cases, this type of data would result in incorrect behavior. A better approach would classify the muscle as tensing over the entire high-amplitude section. Inman et al. (1952) applied signal rectification and low-pass filtering to better estimate the sections of the EMG signal characterized by high-amplitude peaks. Because the resultant signal appears to outline the raw signal, this technique is termed linear envelope generation.

Simple amplitude thresholding is a fast, efficient way to classify the contractile state of a muscle, but has two major drawbacks. The first drawback is that thresholding allows only binary classification of the muscle groups as relaxed or tensed at a specific force. Clearly, tensed muscles can produce varying levels of force and do not act in an all-or-nothing fashion. The second drawback is that a simple threshold may be susceptible to rapid switching between relaxed and tensed muscle classifications when the signal amplitude is close to the threshold level, or during the transition from relaxed to tensed muscle states. This switching is caused by the high-frequency components of the signal.

In addition to noise filtering applications, the Fourier transform may also be used to extract features from the signal that are important for classification. For example, signals generated from muscle contractions may have a specific frequency signature that can be detected using the transform. For EMG signals, even if the waveform of a particular muscle contraction does not have the exact same voltage-versus-time graph as another, the frequency components of waveforms originating from the same muscle groups performing the same motion should be similar. Fourier analysis can thus facilitate the comparison of two signals that are fundamentally similar in terms of their frequency components and underlying physiology. This is true even if the signals occur at different times or vary in amplitude, due to otherwise confounding factors.

However, the Fourier transform does have limitations. Conversion of the time domain to the frequency domain results in loss of time resolution, so that particular points in the transformed data no longer correspond to exact times, but rather to spans

of time. A simple method of partially compensating for this limitation is to use a windowed Fourier transform, in which a Fourier transform is performed not on the entire signal, but rather on discrete sections. The size of each of these sections, or windows, is determined by both the sampling rate at which the data was collected and by the duration of the features targeted for detection. Fourier limitations are especially evident in the study of signals that were not originally sinusoidal in nature. Signals with time-varying frequency components, such as EMG signals, often result in less than optimal Fourier decompositions (De Michele et al., 2003).

Wavelet analysis was introduced specifically to address this problem, and permits better resolution of time and frequency components of signals (De Michele et al., 2003). By expressing the signal as a sum of non-sinusoidal component signals, a more precise decomposition may be achieved. Specifically, if the correct basis of component signals is chosen, such that they more closely correlate to the MUAP signals within EMG signals than would simple sinusoids, then the original EMG signal can be expressed more compactly as the sum of a smaller number of these basis waveforms. This aspect is useful because features in the signal that are shaped similarly to those of the chosen wavelet basis can be more readily resolved from background noise.

Another method used to classify the signals is an artificial neural network (ANN). A neural network is comprised of a set of simple, interconnected signal processing nodes, the model for which is approximately based on a physiological understanding of biological neurons (Hassoun et al., 1994). Such a network can be employed with relatively simple general learning algorithms to the problem of pattern

recognition, in our case, EMG signal classification. They are particularly well-suited to applications where little is known about the input signal, and are also highly effective in recognizing noisy variations of previously learned patterns (Hassoun et al., 1994). One drawback of such systems is that they can be computationally expensive; however, it is possible to implement simple neural networks very efficiently. The outputs of the neural network may generate very useful signal processing models that can be coded or realized explicitly on a chip to increase efficiency. Many current EMG-analysis algorithms make use of ANNs in order to interpret and categorize the EMG signals (Hassoun et al., 1994).

A different approach to the classification of EMG signals is the analysis of statistical properties such as the Euclidean distance between MUAP waveforms, and the mean and standard deviation of inter-pulse intervals of individual MUAP trains (Raez, Hussain, & Mohd-Yasin, 2006). In essence, these algorithms attempt to detect signal events corresponding with MUAPs based on the presence of features that differ from a control or base signal in a statistically significant manner.

Each of these signal processing techniques has been useful in the classification of EMG waveforms for a particular application (Pelland and McKinley, 2004; Kuruganti, 1995; Güler and Koçer, 1995; Abbink et al., 1998; Inman et al., 1952; De Michele et al., 2003; Hassound et al. 1994; Raez, Hussain, & Mohd-Yasin, 2006). In our study, we sought to determine a signal processing algorithm for a classification algorithm that was accurate, efficient, and robust. In addition, in order to produce a marketable product, the selected signal processing algorithm aimed to

limit complexity in order to reduce the costs of implementation and to increase efficiency, while minimizing any resultant decrease in classification accuracy.

Current State of the Art for Hemiparetic Treatment

Methods to improve community ambulation have historically included the use of orthoses, assistive walking devices, and restoration of walking symmetry (Mauritz, 2004). In particular, metal ankle-foot orthoses (AFOs) placed on the paretic leg have been shown to improve the gait within the paretic leg (Mauritz, 2004). Unfortunately, they only achieve this by holding the paretic foot in gait position, to prevent spastic foot movement. Assistive walking devices, such as walkers and canes used opposite of the paretic leg, can also improve community ambulation, or at least mask a lack of community ambulation (Mauritz, 2004). These devices require the patients to have relatively normal upper body control and coordination (Mauritz, 2004). Restoration of gait symmetry has traditionally included practicing walking patterns and weight shifting on the paretic leg, and using electromyography to track neuromuscular recruitment in the clinic. Sadly, when patients left the clinic, researchers found that concentrated walking did not necessarily translate to community ambulation (Mauritz, 2004).

Thus, methods of neural restoration and improvements in gait endurance have proved more promising with the advent of treadmill rehabilitation and gait machines. For this reason, treadmill rehabilitation showed high potential in animal studies. However, patients were significantly more comfortable walking on a stationary floor rather than on a treadmill (Mauritz, 2004). Similarly-designed gait machines may provide further improved gait analysis. Still, there have not been any large-scale

studies to prove their effectiveness or improvement in comfort over treadmill treatment (Tong, Ng, Li, & So, 2006).

Robot-assisted locomotion is another promising method of treatment, in which a robot orthosis corrects or compensates for the lack of proper neural activity in the paretic leg during gait. These robot-assisted locomotors may also use virtual reality (VR) or game-based systems to provide this correction (M. J. Johnson, 2006). Hornby et al. (2008) showed that such a device would provide the needed adjustments. However, this improvement was too small relative to traditional therapist-assisted locomotion to currently be more widely used (Hornby et al., 2008). Even with its high potential, the costs associated with such a complex virtual learning or similar learning environment may be too cost-ineffective due to the high cost of processing such elements. There are few such devices on the market outside of the Lokomat® Gait Trainer systems (Brincks, Nielsen & Kock-Jensen, 2009). These limited choices suggest that alternative stroke rehabilitation treatments are necessary to provide successful rehabilitation to patients who may neither have access to robot assisted locomotion nor want to use a treadmill-based device.

Brain-computer interfaces (BCIs) using non-invasive electroencephalography (EEG) have also been investigated with the potential to improve motor control in individuals with severe neurological disorders or impairment brought on by amyotrophic lateral sclerosis (ALS) or stroke. For post-stroke patients, BCI technology could be used as a form of neurofeedback to indicate brain activity and to help them isolate and decrease abnormal activity. The safest method of neural signal acquisition is through the use of surface electrodes. Non-invasive signal acquisition

has a limited frequency range and resolution but is safer and easier to implement than invasive methods (Daly and Wolfpaw, 2008).

The signal processing occurs in two steps. The first is feature extraction where signal features such as amplitudes, sensorimotor rhythms, or firing rates are used to characterize the signal. The second step is to translate the decoded signal into an output command. These commands could be anything from moving a cursor on a screen to controlling the movement of a prosthetic device. One aim of BCIs is to restore normal function in the central nervous system (CNS) through plasticity. Neuroplasticity is the changing of neuron functions or rearrangement of networks to perform new functions based on experience. Therefore, for stroke patients, activity-based CNS plasticity can create a framework for motor re-learning. However, rehabilitation efforts involving BCIs are still in a developmental phase and future implementations will depend on further research into the type of signals and signal features necessary to effectively retrain motor functionality (Daly and Wolfpaw, 2008).

Current State of the Art in Videogame Rehabilitation

There are several forms of videogame control systems that can be used in rehabilitation, including some that use EMG or EEG techniques. While very few gaming studies focus on rehabilitation applications, these existing games can still serve this purpose. The current state of the art in gaming proves that EMG signals are a viable approach to entertainment systems. Therefore, incorporating gaming into rehabilitation should encourage our patients to want to use our system and aid in their rehabilitative processes through increased exercising.

One study presented an analysis of the strengths, weaknesses, opportunities, and threats (SWOT) of virtual reality in rehabilitation (Rizzo, 2006). The authors concluded that the field is still in its developmental stages because several “proof of concepts” have shown successful results; however, few systems have made it into mainstream or clinical use. One weakness they noted was the bulky interfaces that required many wires and different types of hardware. These physical issues imposed limitations on patient movement, comfort, and even the game and system functions. However, the authors suggested that the incorporation of wireless technologies would reduce these limitations. They insisted that strong multidisciplinary collaboration and user-centered input and evaluation methods were essential in developing successful applications. They also suggested that the front-end of the systems needed to be more user-friendly and aimed towards the physical therapist, so that computer science technical support would not be necessary. This suggestion was one of the big problems they found with university systems. For clinical utility, programs must also deliver basic summary scores, comparison statistics with accumulated normative data, and graphical representations of the data for the physical therapists to evaluate the results. They suggested these aspects as the main back-end limitation.

Another group of authors reviewed studies on virtual reality-based stroke rehabilitation from 1980 to 2005 and chose 11 studies for their evaluation (Crosbie et al., 2007). Five studies addressed upper limb rehabilitation, three studies addressed gait and balance, two studies addressed cognitive interventions, and one study addressed both upper and lower limb rehabilitation. They assessed these studies as a review of the state of virtual reality in rehabilitation. Tables 1, 2, and 3 below

describe the parameters of each system/study. They found that eight of the 11 studies presented positive results, while the remaining three studies had statistically significant results. Two of the studies examined are the Rutgers glove and ankle, which are cited in greater detail below.

Table 1: Crosbie et al. sources: Empirical Research (Levels I and II)

Study	Level & Quality	Design	Reported type	Age (years)	VR Intervention	Display Device	Interface	Control
1	I/W	RCT	Vascular brain injury (n=24) and stroke (n=24)	25-85	Active & passive desktop VE to explore virtual bungalow for memory and layout training	Desktop PC	Mouse	Non-impaired group (n=48)
2	I/W	RCT	Hemiplegia single incident (n=20)	41-79	Immersive VR system in conjunction with treadmill training	Head-mounted display	Video camera	Real training (n=10)
3	I/W	RCT	Hemiplegia (n=14)	Not reported	VR augmented treadmill training, interactive balance training	LCD monitor	Speed/position sensors	Treadmill training (n=7)

* All results statistically significant with intervention.

Table 2: Crosbie et al. sources: Empirical Research (Levels III and IV)

Study	Level & Quality	Design	Reported type & time since stroke	Age (years)	VR Intervention	Display Device	Interface	Control
4	III/W	ABA	Right internal capsule 12 weeks	Late 50s	Non-immersive arm training	See through glasses	Haptics	Healthy males as reference group
5	IV/W	Before & after case series	Infarct, embolus, other 6-46 months	26-72	Augmented VR system with virtual teacher for arm motor tasks	Desktop PC	Electromagnetic sensors	None

6	IV/W	Before & after case series	Internal capsule infarct, parietal hemorrhage 3-6 years	54-83	Desktop system for hand function	Desktop PC	Electromagnetic sensors	None
7	III/W	Before & after case series with control	Right hemisphere stroke ≥ 6 weeks	55-75	PC based system to train people w/ spatial neglect to cross a street safely	Desktop PC	Keyboard	Unequal number of age-matched controls (n=6)

* All results were positive effects with intervention.

Table 3: Crosbie et al. sources: Non-empirical Studies (Level V)

Study	Reported type & time since stroke	Age (years)	VR Intervention	Display Device	Interface
8	Right CVA (n=1), right middle cerebral artery ischemia (n=1) 18 months & 2 years	73 & 70	Haptic feedback with glove & ankle VR devices	Desktop PC	Haptics
9	Left & right internal carotid occlusion (n=2) 1.5 & 3.5 years	76	Virtual teacher & VE training for upper extremity tasks	Desktop PC	Electromagnetic Sensors
10	Right hemisphere stroke (n=1) 6 months	72	Gesture Xtreme project system for range of sitting & standing balance tasks	Video projection	Large screen

One form of rehabilitative gaming includes the use of computer games as a training aid. Specifically, there was a case study on the use of games to resolve sensorimotor deficits in a 34-year-old male recovering from subarachnoid hemorrhage. Researchers found that with increased training via the game, the patient improved control significantly (Taylor & Berry, 1998). The patient used the keyboard to exercise control over his fingers in a shooting range game, and over time,

gradually improved his response time and accuracy. Even though this study did not use EMG, this article proves that gaming can be used to aid in rehabilitation.

Another noteworthy system is the Rutgers Ankle, a platform-type haptic interface that supplies resistive forces in six degrees of freedom in response to virtual reality-based exercise (Deutsch, 2001). Patients exercise their ankle in different positions in order to virtually pilot an airplane in a game. This system can be adjusted to each individual patient, and can be adjusted throughout long-term use by changing the force applied by the interface, the difficulty of the simulation, and the degrees of freedom involved in the game. This variability makes the system applicable to patients with a broader range of disabilities, and allows for targeted adjustments as the patient progresses. This product focuses only, however, on ankle movement as the patient sits in a chair. The system does not monitor all muscle activation associated with walking, nor does it allow the patient to practice the complete motions involved. In a nine-month case study after six sessions of training, results showed improvement in force generation, endurance, and coordination of the affected ankle (Deutsch, 2001). Improvements in the simulation tasks correlated with improvements in walking and stair climbing. The findings of this study helped guide us towards our end goal of an effective rehabilitation system with high variability.

A second system under development at Rutgers University targets upper-extremity post-stroke rehabilitation, specifically for the hand. It uses a CyberGlove and a Rutgers Master II-ND haptic glove to reduce impairments in patients' finger range of motion, speed, fractionation independence, and strength. Feedback is displayed through virtual reality through a PC in the forms of playing a piano, scaring

away a butterfly, and displacing pistons. In one study, four subjects, aged between 58 to 72 years old, used the system over a three-week period (Boian, 2002). The system was implemented over the internet, so that the physical therapist could remotely follow patient progress. The CyberGlove was used in exercises for the first three parameters (range of motion, speed, independence), while the haptic glove was used for the last parameter (strength). Along with task-oriented feedback, the program displayed a numerical evaluation of each exercise. The patients requested this feature as a better indication of their progress. Results from the study showed improvements in fractionation for all four subjects, in thumb range and finger speed for three patients, and only modest improvement in strength, which they attributed to a hardware malfunction. Both the subjects and the therapist gave positive evaluations of the system. Our system also incorporated real-time quantitative feedback because this study proved it to be a desirable feature for the patients and therapists.

Perhaps most similar to our project is a system designed for arm rehabilitation for stroke survivors (Dipietro et al., 2005). They used the patient's EMG signals to manipulate a robot that assists him or her to perform point-to-point movements in a horizontal plane. The tasks are presented as a simplistic game, where the user moves in one of eight directions. Clinical testing has not yet been completed, so no firm conclusions can be made about its efficacy.

Another EMG videogame system was developed at Catholic University using an Xbox and Xbox360 (Mathews, Judge, & Ortega, 2007). The system involves four electrodes that are used to control four buttons on a video game controller. The

system also includes software to assess a user's ability to contract specific muscles without also activating unwanted muscles.

A U.S. Defense Advanced Research Projects Agency (DARPA)-funded study utilized surface EMG as an interface to the commercial videogame *Guitar Hero* (Armiger & Vogelstein, 2008). The researchers created a novel rehabilitation device for upper-extremity amputees. The system trains the amputees using a virtual integration environment (VIE). A VIE is a virtual reality training tool in which an onscreen animated arm mimics the user's intended movements in real-time, based on inputs from multiple electrodes attached to the user's residual arm. The user controls the game by flexing his or her index, middle, or ring finger muscles of his or her residual hand. The muscle flexion's resulting myoelectric activity is recorded by six or more EMG electrodes placed on the forearm. Real-time data processing via pattern recognition algorithms determines the user's intended motion and resulting control of the game. A built-in scoring system evaluated the classifier's performance. Three healthy control subjects found the EMG-controlled gaming system to be "effective, fun, and engaging" (Armiger & Vogelstein, 2008).

One example of an EMG gaming system that has yet to be applied to rehabilitation is a new EMG entertainment interface for the Othello videogame on the Sony PlayStation 2 gaming console (Shima, Bu, Okamoto, & Tsuji, 2005). Researchers found that by using a probabilistic neural network (PNN) based on measured EMG signals, they could estimate the user's intended motion with a high degree of precision to determine game control commands. Furthermore, the system could be easily changed so that the final product could run on a multitude of various

gaming or other entertainment systems. The researchers grouped various types of games based on two sets of criteria: fast or slow response, and low or high number of commands needed to be executed in the game. The games that were slow in response with a low number of commands were clearly easier to control via this EMG system. This result indicates that ideally, we should select one of the suggested games in this category.

Another product called the TheraDrive system consists of an off-the-shelf steering wheel that provides force feedback and a software platform that records wheel movement, allowing patients to complete tasks displayed on a screen. It incorporates a combination of custom and commercial games, such as Nintendo's *MarioKart* and EA Games' *Need for Speed*. Subjects moved the wheel to track objects or simply steered to follow a course using only their impaired arm. Johnson et al. presented two major studies: one to determine the usability of the TheraDrive system for stroke therapy, and another to measure the ability of the system to motivate subjects and keep them engaged during therapy (2006). For the first study, the researchers found a correlation between performance in the game and motor impairment levels, but no correlation between improved performance and improved clinical measurements of functionality. For the second study, results showed that the subject enjoyed and was motivated by the game. She felt driven to play by the desire to obtain better scores. Aside from accurate and effective results, this type of attitude is the ultimate goal of videogame systems in rehabilitation. Therefore, this outcome also suggests that a gaming system, such as the one we developed, can be an effective tool for motivating patients in rehabilitation.

Weiss and colleagues gave a good general overview of virtual reality in rehabilitation of cognitive and motor impairments (2004). In particular, video capture virtual reality involves a system of camera-based motion sensors that records the user's movement and translates it to a screen in a simulated environment. One specific example was the use of Sony's EyeToy application used for the PlayStation 2. The digital camera device allows the user to interact with virtual objects by capturing and displaying real-time images of the user on a TV monitor. There are other similar systems, such as the VividGroup GX system that was used to allow a stroke patient to play the role of a soccer goalie in a simulated environment. At the time of Weiss' review, the EyeToy was cheaper and more easily implemented, while the GX was more advanced, yet also more expensive and difficult to use. The advantage of video capture VR over other gaming rehabilitation systems is the first person experience, where the user can see him or herself instead of an avatar. Additionally, video capture VR does not require the patient to wear or be wired to extraneous devices. However, video capture VR does not provide any information about the actual muscle contractions, and therefore is of limited use for our project.

There are several EEG gaming systems in place or slated for launch in the near future. These include the Neural Impulse Actuator, the Mindball Game, Emotiv Systems' EPOC, and NeuroSky products. The Neural Impulse Actuator costs roughly \$150, and simply reads EEG signals to map them to keyboard strokes on a PC (OCZ Technology, 2009). The Mindball Game involves a \$20,000 table that contains a physical ball in which two players compete to move the ball with their minds (Interactive Productline IP AB, 2008). The ball is controlled via an EEG

interface that both players attach themselves to. The EPOC is a \$300 EEG system released in 2009 that uses two-axis gyros to measure head rotation in addition to EEG signals (Emotiv Systems, 2008). Finally, NeuroSky products use a single electrode as opposed to the other systems that require more electrodes, which makes their products cheap and easy to set up (NeuroSky, 2009). These EEG systems prove that biological signals can be used to control games and are capable of gaining acceptance in the general gaming world.

Chapter 3: Methodology

Our approach to designing and building an EMG biofeedback videogame consisted of three stages of development:

Stage Zero: To design and build the prototype EMG sensor hardware, create an initial signal processing algorithm, and gather preliminary data from team members.

Stage One: To refine and test the sensor system, determine rehabilitation exercises and muscles of interest, develop an experimental procedure for patient testing, and finalize the signal processing algorithms.

Stage Two: To develop a classification algorithm, interface the system with the chosen videogame, and test the completed system on hemiparetic subjects in a laboratory setting.

Stage Zero

A full data acquisition system was designed that includes multiple hardware components that connect together with some degree of modularity. Pairs of gel electrodes capable of detecting EMG signals are placed on the skin over selected muscle groups. Each of these pairs of electrodes is attached to a sensor circuit board containing a simple differential amplifier with filtering.

Although commercial EMG electrodes with built-in signal conditioning are available, such as from B&L Engineering, each sensor carries a price tag of at least \$200 (Whyte Griffith Associates, 2008). We did not have a budget that would allow for this kind of purchase, but we did luckily have the time to design our own sensor boards. This had the added advantage that we were familiar with the underlying

functionality of the boards, which allowed the team to debug problems and tailor the signal conditioning to specific preferences. Six of these boards are powered concurrently by a central battery board held in a fanny pack around the waist, which also contains a connection to reference ground and a connector for sending the outputs from each board to a computer with data acquisition software. The design of each of these components will be discussed in further detail below.

Sensor System Design

The sensor boards were designed as amplifying and filtering circuits, which have been shown to be effective for electromyographic recording in previous research using operational-amplifiers (op-amps) (Pierce, 2004). The same design has also been used to read input for an EMG classification algorithm (Eisenhower & McMichael, 2005), a goal similar to that of our research. Low-noise surface-mount components were chosen to obtain a clean, clear signal. Also, the differential inputs to the boards were obtained through lead wire connections adapted from EKG recording methods. The lead wires have button snaps to connect onto the gel electrodes, and the other ends were cut, stripped, and soldered directly onto the input pads of the boards. Several other improvements intended to address safety and ease of implementation were considered in the design stage, not all of which were incorporated into the final circuit.

First, a four-op-amp circuit implementation was simulated using PSpice ORCAD, giving the frequency response, phase response, and transient output for various input frequencies. This initial circuit was reviewed and broken down into separate stages using hand calculations, to further understand the purpose of each op-

amp, resistor, and capacitor within the circuit. In the initial parts ordering, a very-low-noise quad op-amp OP470 chip was chosen to allow for the clearest potential signal outputs, a decision that turned out to be instrumental in the system's ultimate success. Each component of the circuit was then chosen to take the specific op-amp characteristics into account. For example, the pass band range of the filtering component needed to include the range of the majority of EMG frequencies. Initially, the resistor-capacitor pairs in the circuit that constituted the low-pass filters seemed to include frequencies that were too high for what we wanted, so the values of the capacitors were halved by adding another capacitor of the same value in series. The resulting frequency response range proved to be adequate. The final schematic of the printed circuit board is shown in Figure 4. A detailed explanation of the functionality of each part of the circuit follows.

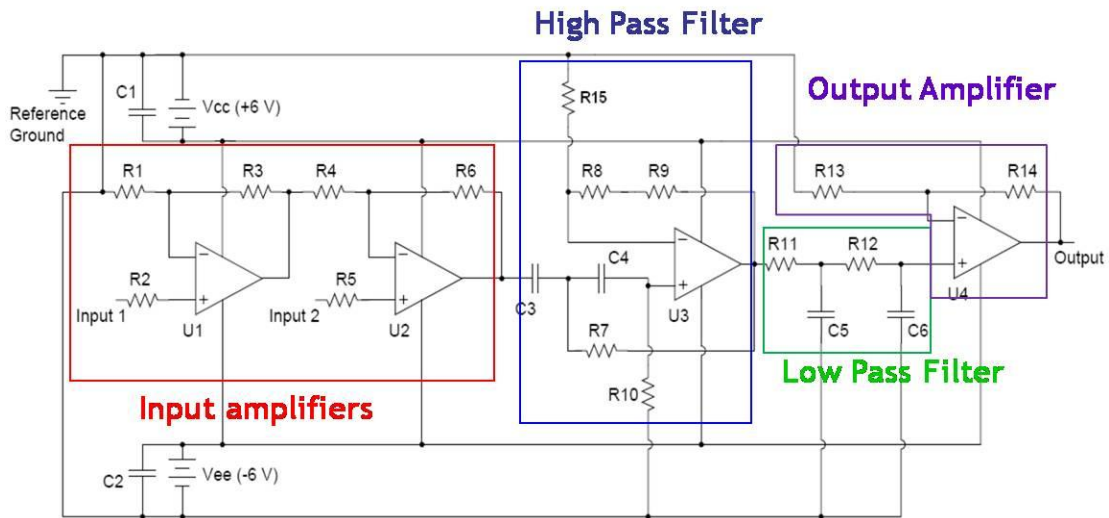


Figure 4: Final schematic representing the behavior of the working amplification and filtering circuits.

U1 and U2 are the two op-amps that make up the first stage of the circuit, a differential amplifier. R1 and R6 are both 100 k Ω , and R3 and R4 are both 10 k Ω .

R2 and R5 are simply 1 kΩ and have no bearing on these op-amps' function.

According to hand calculations, the voltage level before capacitor C3 is equal to $11V_2 - 11V_1$, where V1 and V2 are electrode inputs 1 and 2, respectively. The third op-amp, U3, and the capacitors and resistors connected to it create the second stage, a second-order high-pass filter, with a transfer function:

$$H(s) = \frac{1}{\frac{R_{15}(sC_3R_7 + 1)(sC_4R_{10} + 1)}{(R_{15} + R_8 + R_9)R_7R_{10}C_3C_4s^2} + 1}$$

where R7 = 150 kΩ, R10 = 150 kΩ, R15 = 10 kΩ, R8 = 5.6 kΩ, R9 = 330 Ω, C3 = C4 = 0.1 μF, and s is the complex frequency of the voltage before it reaches C3. The gain of this stage is dependent on both the specifications of the op-amp and the values of resistors R15, R8, and R9, but is about 1.6 V/V. The cut-in frequency is about 10 Hz.

The two resistor-capacitor segments in the third rectangle are the third stage, a second-order low-pass filter. The resistors and capacitors have values of 10 kΩ and 0.005 μF, respectively, which cause the frequency response using these particular op-amps to cut off significantly around 2 kHz. The final op-amp and accompanying resistors is the fourth and final stage of the sensor board circuit, and is simply a post-filtering amplifier, to bring the signal up to a voltage level that allows for easy feature extraction. R13 is 10 kΩ and R14 in the final working boards is a potentiometer, with its resistance tuned to a value anywhere between 1 MΩ and 5 MΩ. The exact value of the potentiometer's resistance varies from board to board in an effort to counteract the imprecise nature of all of the tiny components and their connections to each other, which result in differing gains at the end of the third stage. The potentiometer adjusts

for this and greatly amplifies the signal so that every board's final gain is 2500 V/V at a frequency of 100 Hz.

The magnitude and phase response vs. frequency for this circuit is seen in Figure 5. The transient output response for several select input frequencies is shown in Figure 6.

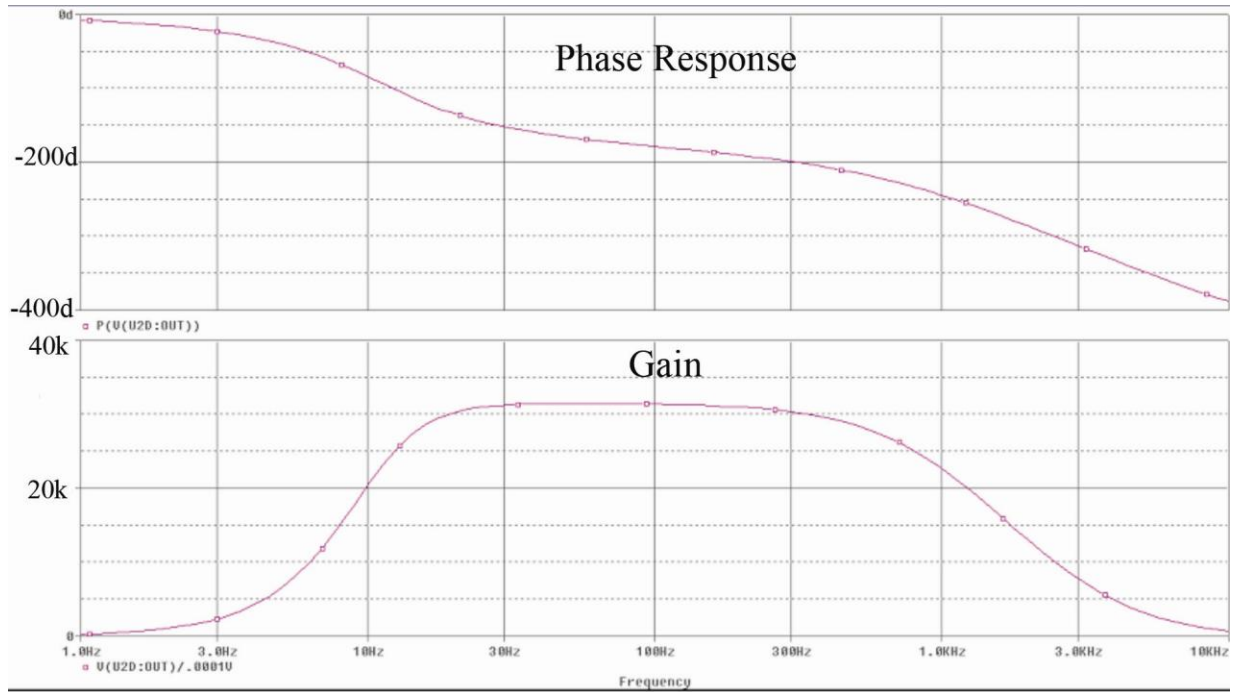


Figure 5: Phase and magnitude response for the final circuitry. The frequency axis is on a logarithmic scale, and the magnitude response axis is in terms of total gain, which is the ratio between the output voltage and the input differential voltage.

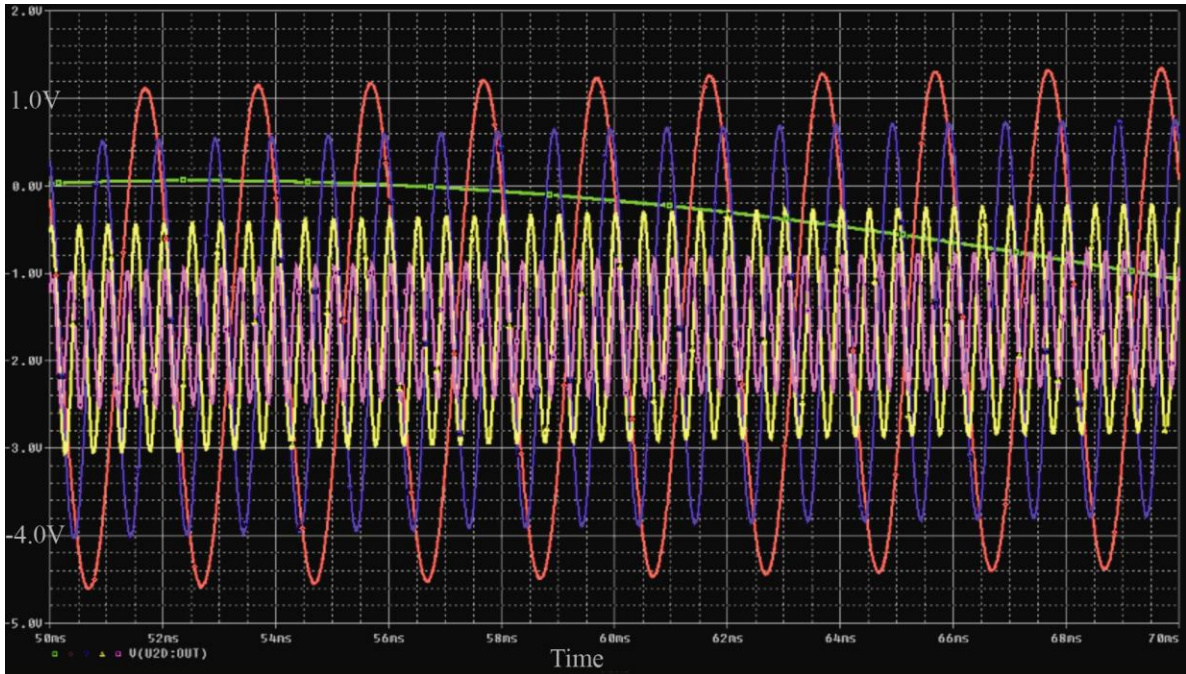


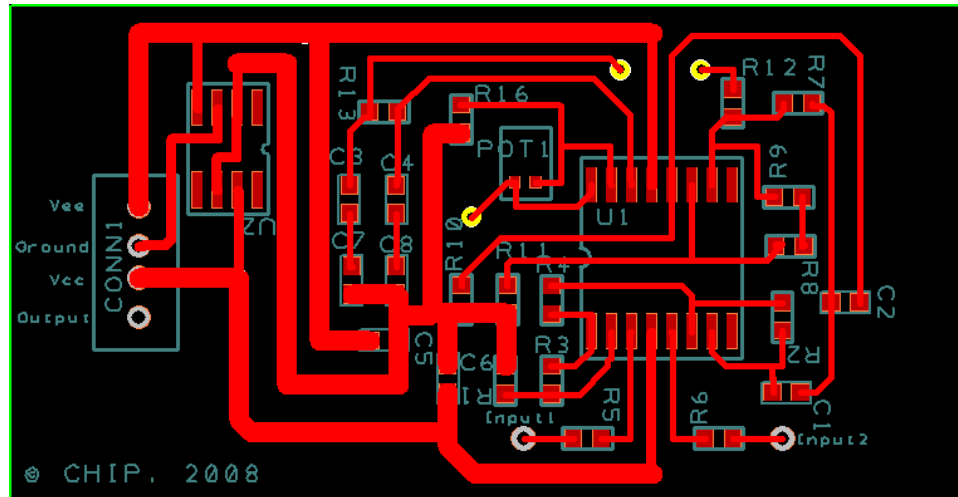
Figure 6: Transient output responses at different frequencies. Green: 10 Hz; Red: 500 Hz; Blue: 1 kHz; Yellow: 2 kHz; Pink: 3 kHz. Note that the relative amplitudes agree with the gains shown in Figure 5.

Additionally, in preparation for further development of the EMG system, conductor pads for a 40-pin chip were placed on the backside of the printed circuit board, including connections to what is intended to be a positive voltage rail, negative voltage rail, and ground. However, the intended use for these pads was soon deemed to be unnecessary, so no actual component is attached to them. There is no perceptible difference in the output or voltages at any point in the circuit due to the presence of these pads. Another component that appears in the board layout but was not used in the final sensor circuit is a single-op-amp chip that was intended as a simple voltage follower between reference ground and the sensor circuit. It was meant as an added precaution for limiting the current that a patient wearing the sensor boards could be subject to. It was taken out early on because it prevented the board

from working as intended and is unnecessary due to the high impedance of the electrodes and other parts of the circuit. Instead, a solder bridge was used to make a direct connection between the USB ground pin and the circuit's reference ground.

The layout for these sensor boards was done using PCB Artist, a free printed circuit board layout program available at 4pcb.com, from Advanced Circuits. The entire board was designed from scratch, with overall dimensions of 6 cm x 3.12 cm. The conductor paths were largely inspired by the paths done in David Pierce's EMG sensors (2004), since his circuit also included a general instrumentation amplifier with a quad op-amp chip. The printed circuit board design, front and back, as seen in PCB Artist, is shown in Figure 7. The USB connector for each of the boards is explained next.

7a)



7b)

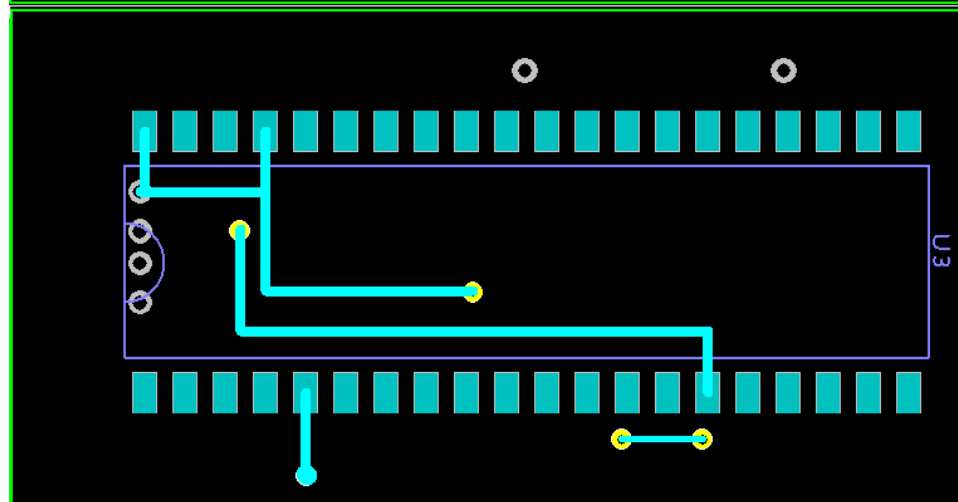


Figure 7: The top copper, solder mask and silkscreen is shown in 7a, and the bottom copper, solder mask and silkscreen is shown in 7b. Note the unevenly-spaced holes designed for the USB receptacle at left, the solder pads for the quad op-amp chip U1, the single op-amp follower U2, and the 40-pin chip U3. POT1 is a placeholder for the final op-amp's feedback resistance, which eventually was implemented using a 5 M Ω potentiometer.

These sensor boards required power, a reference ground connection, and a method of sending its output to the data acquisition system. To this end, every sensor board was designed with a USB connector, and a central board was created with two battery holders, a 50-pin connector for a data acquisition card cable, and six USB connectors. Each sensor board was connected via a USB cable to one of the connectors on the central board. Every USB connection had 4 pins that allowed

positive voltage rail VCC, negative voltage rail VEE, reference ground, and the sensor board output to be shared between the sensor board and the central board as needed.

- a) Two 9-volt batteries, which were placed in the central board's battery holders, were wired up to every USB connector to provide the positive and negative 9-volt rails, which were essential for the functionality of the op-amps on the sensor boards.
- b) The reference ground for every board is connected to a common pin on the central board, which has a four-foot long lead wire connected to it. This lead wire was designed to be long enough to reach away from the central board to snap onto a reference gel electrode, which was usually placed on the back of a testing subject's hand.
- c) The output from each sensor board was wired to its corresponding pin on a 50-pin connector which was attached to the central board. A cable from a data acquisition card hooked onto this 50-pin connector, allowing the outputs to be recorded on a computer using a preprogrammed data acquisition program in MATLAB.

This powering board can be held in a waist pack worn by the patient and allows for minimal wire connections from the body of the patient to the feedback system.

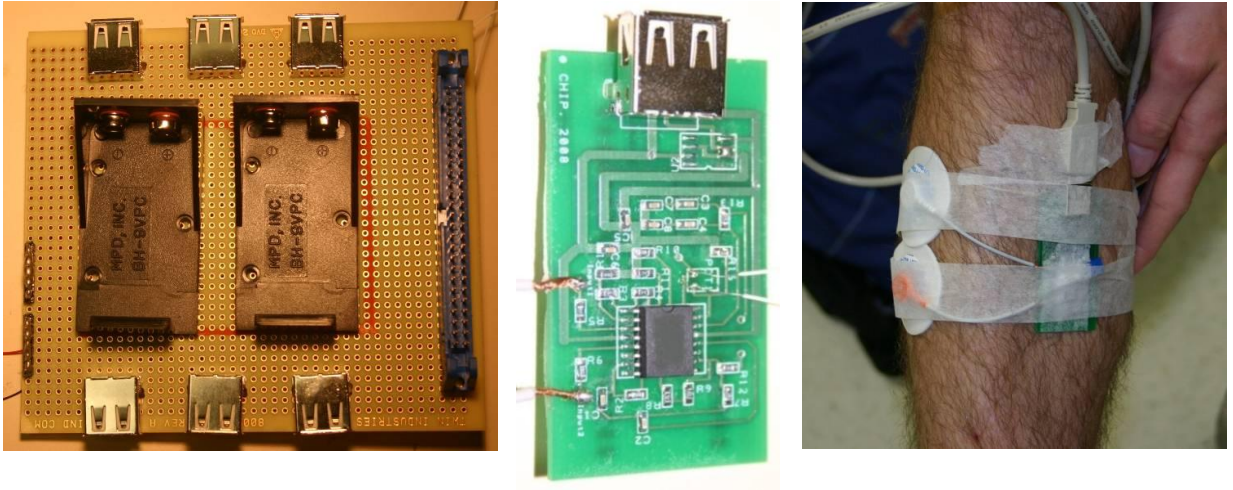


Figure 8: EMG sensor hardware. Left: power board with connectors for six USB cables and two 9V batteries. Middle: sensor board. Right: sensor board in place on the lower leg (tibialis anterior).

Signal Acquisition System Validation

After completion of the sensor boards, the first round of testing was conducted on members of the team itself. There were two main objectives for this round: first, to evaluate the functionality of the team-built hardware and acquire preliminary EMG signals for the signal classifier subgroup to work with, and second to determine which muscles to monitor to best classify the rehabilitation exercises. Team members were fitted with electrodes on a single muscle, such as the calf, and performed basic movements to test whether the system operated correctly.

Stage One

Once the signal acquisition system was functioning, we needed to refine and further test the sensor system, determine muscles of interest, determine the rehabilitation exercises, develop an experimental procedure for patient testing, and finalize the signal processing algorithms. To gain access to post-stroke patients, we secured a partnership with Dr. Jill Whitall and Dr. Sandra McCombe Waller,

professors in the Department of Physical Therapy and Rehabilitation Science (PTRS) in the School of Medicine at the University of Maryland, Baltimore (UMB). Dr. McCombe Waller is also a physical therapist.

Rehabilitation Exercises

Drs. McCombe Waller and Whitall encouraged focusing the design of the EMG videogame system around four gait training exercises they developed. For each exercise, specific muscles to monitor with EMG were chosen from groups of muscles activated by that exercise. The first exercise entails hip abduction and knee extension; the second, hip and knee flexion; the third, knee flexion and ankle dorsiflexion, and the fourth, knee extension and ankle plantarflexion. The exercises are described in more detail below.

Exercise #1

In a standing position, the patient shifts their weight to their paretic leg, and then attempts to straighten their knee. This exercise activates the gluteus medius and the quadriceps.

Exercise #2

In a standing position, the patient shifts their weight to their nonparetic leg and swings their paretic leg forward, bringing their knee up and their heel towards the gluteus. It flexes the hip and knee. This exercise activates the tensor fasciae latae (TFL) and the hamstrings.

Exercise #3

In a standing position, the patient shifts their weight to their nonparetic leg and focuses on kicking their paretic leg out by straightening the knee and

pulling the toes upward (i.e. like kicking a ball). This exercise activates the anterior tibialis and the quadriceps.

Exercise #4

In a standing position, the patient shifts their weight to their nonparetic leg, then steps forward with their paretic leg and plants their foot on the ground, heel first, “controlling” the amount of knee contraction. They are putting weight on their paretic leg without knee buckling. This exercise activates the quadriceps and the gastrocnemius.

Muscle Selection Criteria

We received instructions from Drs. McCombe Waller and Whitall on which physiological functions each exercise involves, but that left us with several potential muscles to choose from for some exercises. Exercise 1 uses hip abductors and knee extensors, Exercise 2 uses hip flexors and knee flexors, Exercise 3 uses knee extensors and ankle dorsiflexion, and Exercise 4 uses hip flexors and knee extensors. The gluteus medius is the major hip abductor, and the tibialis anterior is the sole muscle responsible for ankle dorsiflexion, so these muscles were automatically chosen for their respective functions. Still left to be determined were which muscles to use to measure knee extension (Exercises 1, 3 and 4), hip flexion (Exercises 2 and 4), and knee flexion (Exercise 2). The criteria for selection for these muscles were the strength and consistency of the signal during their designated exercises and the relative lack of activity during other exercises. Together these two features determined the usefulness of a particular muscle in discriminating certain exercises from others.

The quadriceps muscles are responsible for knee extension, so the muscles in this group were our candidates for measuring knee extension. These muscles are: the vastus medialis, vastus lateralis, and rectus femoris. The rectus femoris was eliminated from contention because it is also a hip flexor. Thus, selecting the rectus femoris to measure knee flexion would not help us discriminate Exercises 1, 3 and 4 from Exercise 2, because we would still expect to see significant activity from the rectus femoris during Exercise 2 as a hip flexor. Choosing a knee extensor muscle that is not involved in hip contraction allowed us to better distinguish Exercise 4 from the others. This method, used across all muscles, hopefully provided enough information to consistently distinguish each exercise from every other.

Final Muscle Selection and Electrode Placement

The gluteus medius was selected to measure hip abduction. The quadriceps control knee extension, and of these, the vastus medialis was selected for our EMG monitoring. As explained previously, the rectus femoris was passed over because of its dual role in hip flexion. Of the vastus medialis and vastus lateralis, the medialis was chosen due to observed superior signal quality relative to the lateralis in testing on a (20-year-old male) team member. The hamstring muscle group controls knee contraction, and of these, the biceps femoris was selected by virtue of greatest signal strength and quality during testing on the team member. Specifically, superior signal quality was observed relative to the semitendinosus. For ankle dorsiflexion, the major muscle involved is the tibialis anterior. For ankle plantarflexion, the gastrocnemius (lateral and medial head) and soleus muscles are at play. Gastrocnemius was initially chosen over the soleus for ease of electrode

placement/location. Between the medial and lateral head, the medial head was chosen because the lateral head showed spikes in activity whenever the (group member) test subject shifted weight from his monitored leg to his other leg. For hip flexor, tensor fascia latae (TFL) was chosen over rectus femoris due to significantly stronger signal quality observed during group member testing.

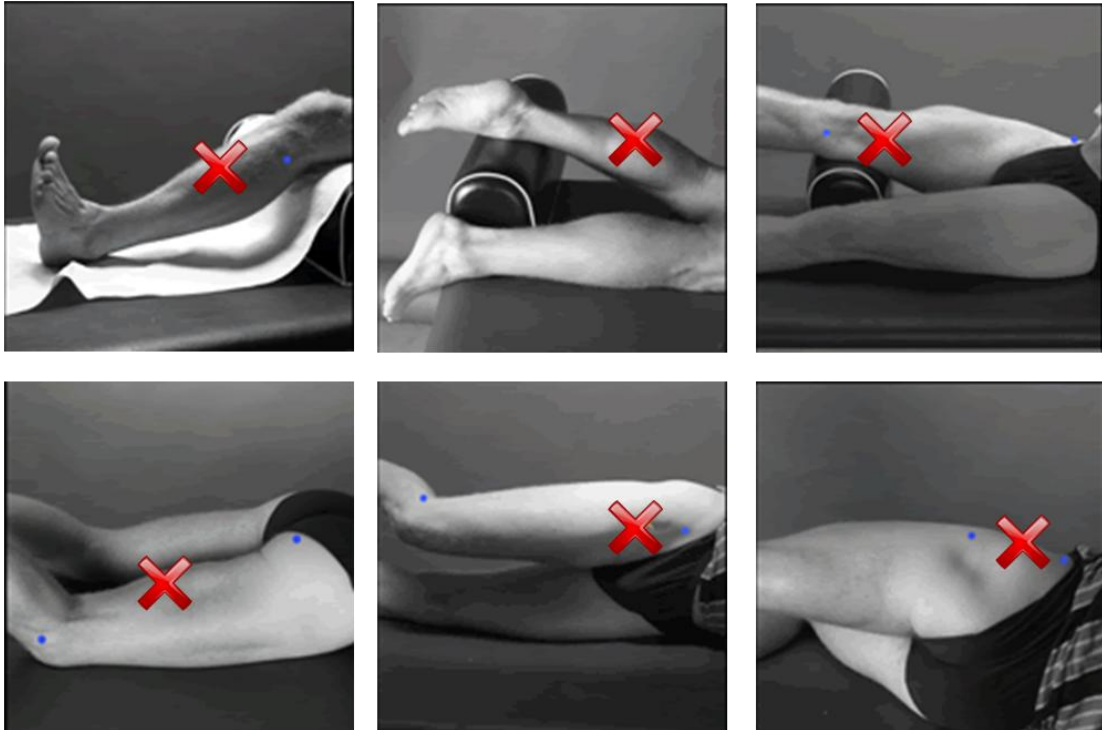


Figure 9: The methods for determining the electrode placement for each of the six targeted muscles; from left to right, top to bottom: tibialis anterior, gastrocnemius medialis, vastus medialis, biceps femoris, tensor fasciae latae, and gluteus medius (Hermens et al., 2006).

For all muscles, electrodes were placed in the middle of the muscles, both length- and width-wise, shown in Figure 9 and Figure 10. For all muscles except the TFL, electrode placement guidelines were taken from the SENIAM project group (Hermens et al., 2006), although testing on group members was done to ensure that suggested locations did indeed yield good signals.

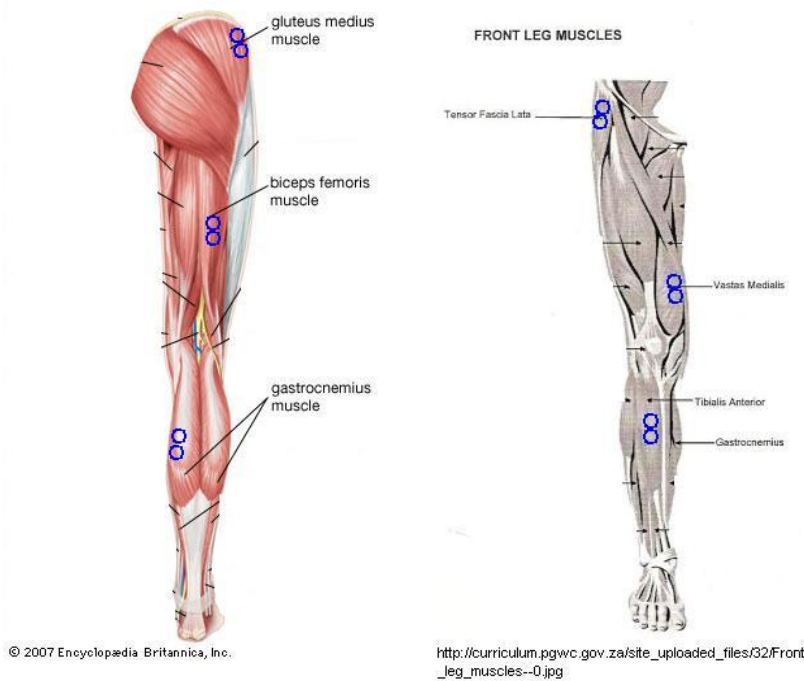


Figure 10: A muscular view of the leg muscles to be monitored, and the precise placement for each electrode pair for each muscle. The muscles are labeled in Figure 9.

The Improved, More User-friendly Integrated Sensor System and Hardware

For Stage One, the sensor system stayed largely the same as in Stage Zero in terms of functionality. However, there were many improvements in terms of aesthetics and convenience that greatly improved the testing process and the testers' experience.

First of all, the initial waist pack was replaced with a new one that could better hold the central board, and had a zipper allowing the USB cables to come out of the sides of the pack. It also had a fully-adjustable waist strap that did not stretch, preventing the pack from sagging to a level that would get in the subject's way as they performed their leg exercises. Most importantly, we developed a safer, more comfortable method of attaching the sensor boards to the testing subject. Instead of using tape to hold the sensor boards onto the body, we fabricated elastic straps with

plastic buckles that could wrap around the leg and tighten in place. The sensor boards were then attached on the outside of these elastic straps using strips of Velcro placed on the straps and the back of each of the boards. This arrangement meant that the boards were no longer in direct contact with the patient's skin. Also, the straps were not sticky, unlike the tape used before, and allowed for a secure connection without being too tight. These changes provided a much more comfortable experience for the testing subjects. In addition, as long as the Velcro stayed secure, this arrangement was much safer for the sensor boards, which were no longer tightly wrapped up in tape that caused stress on the boards as the subjects contracted their muscles. A feature of the new waist pack was that a few Velcro elastic straps could be looped around the waist strap of the pack and hang down, so that the sensor boards connected to the hard-to-reach muscles on the hip and waist could attach to these straps. This practice avoided the need to wrap an elastic strap all the way around the waist or the upper thigh, which was hard to reach and would have undoubtedly been uncomfortable.

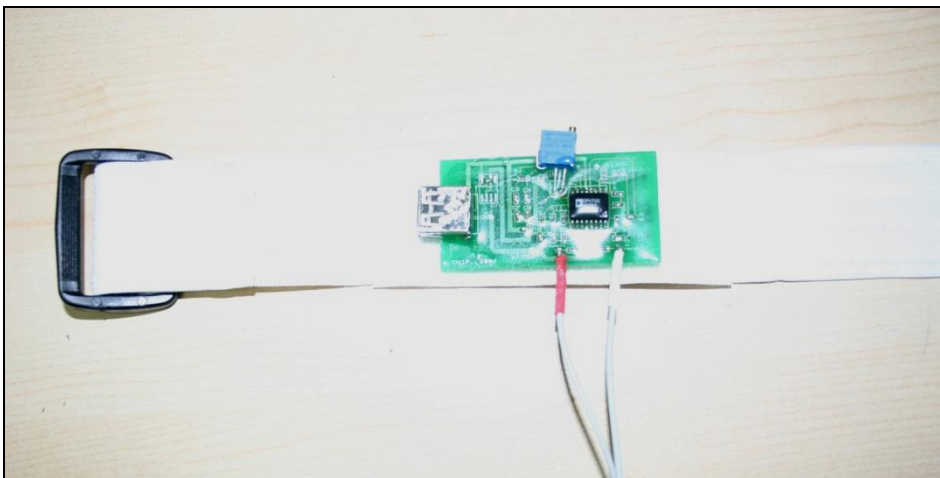


Figure 11: Picture of Velcro elastic strap, which holds the sensor board.

Late in the sensor system development, a pushbutton was added that connected directly to the central board. This pushbutton was simply a single-pole, double-throw (SPDT) switch that had an AAA battery connected to one pole, and reference ground connected to the other pole. The output of the switch was connected to a pin on the 50-pin connector, so the data acquisition software could read it the same way it read the sensor board outputs. Every time this switch was pressed, its output switched between 1.2 V (the output of an AAA battery) and reference ground. This button was used to indicate precisely when a testing subject performed particular landmark actions in an exercise. For example, consecutive button presses indicated when a subject stepped forward and planted their foot on the ground, when they pushed off, and when they returned to standing position again. Using this method, the team could study the EMG data with its corresponding pushbutton output to determine the times when a patient performed a certain task. Matching the EMG data to its corresponding physical movement was supposed to be easier. This was essential for the development and fine-tuning of the classification algorithm. It also helped to quantify any latency problems that may have been present, although realistically human error in the timing of the button pushes may have obscured the lag time between muscle activation and the data acquisition system registering an EMG spike.

Hemiparetic Test Subjects and Testing Procedure

Drs. Whitall and McCombe Waller developed the patients' training regimen, and supervised the preliminary testing to ensure safe collection of electromyographic data from multiple subjects. The testing was completed at a physical therapy

laboratory in the Department of PTRS in Baltimore at UMB. For step-by-step details of the data collection procedures, see the protocol in Appendix I.

The subject pool was comprised of post-stroke patients that had suffered from hemiparesis or spasticity in leg muscles, in order for the data to be most relevant to the intended application. There were four middle-aged patients, two men and two women, labeled as EMG01, EMG02, EMG03 and EMG04. Participants came from a preexisting pool of post-stroke therapy patients who had voluntarily consented in writing to be contacted when research studies associated with the Department of PTRS were in need of testing participants. These patients, who have participated in similar research before, received an e-mail, letter or telephone call inviting them to participate in the research from Dr. Whitall or Dr. McCombe Waller. The contents of the e-mail and the letter appear as part of the Institutional Review Board application included in Appendix II. We also recruited four healthy age- and gender-matched control subjects, labeled as EMG05, EMG06, EMG07, and EMG08.



Figure 12: Test subjects participating in the EMG data collection for Stage One. From left to right, the patients are performing the exercises: weight shift, knee lift, knee swing, step support. These pictures represent 3 of the 4 participants. The extra person in the foreground in the two right panels is a team member who was coaching the study participants during the trial.

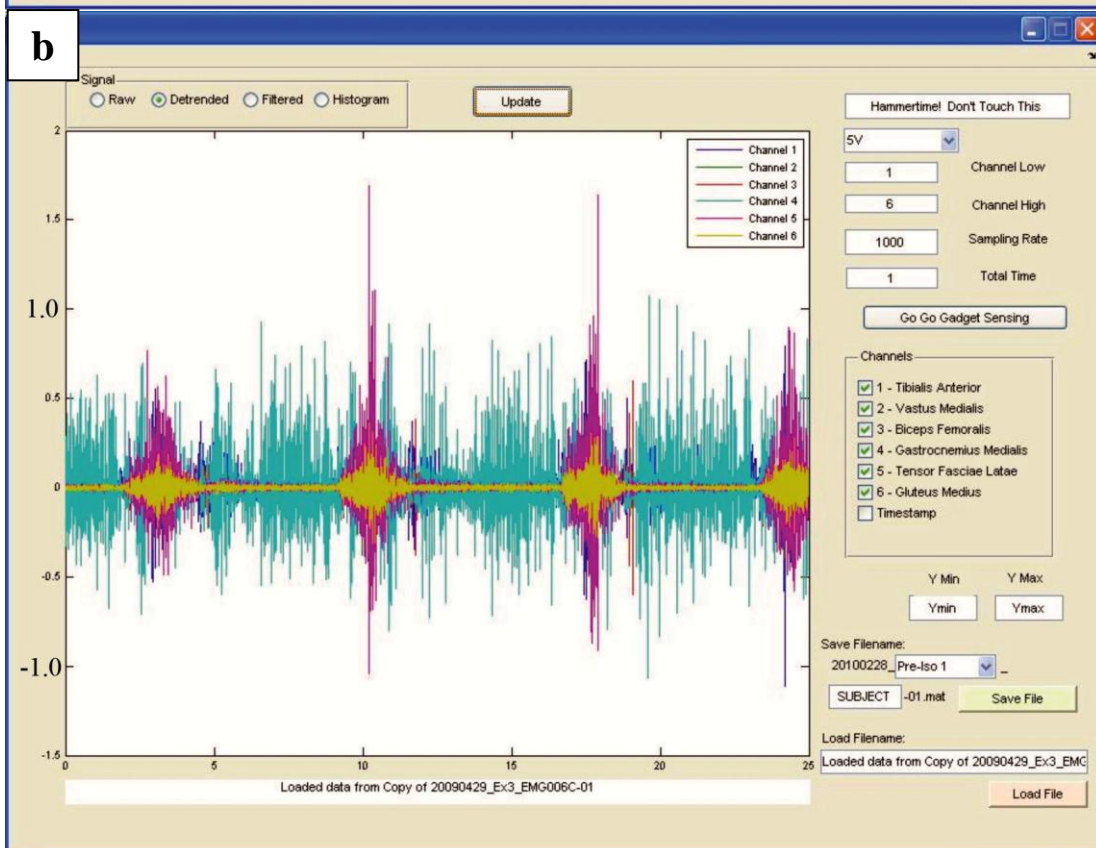
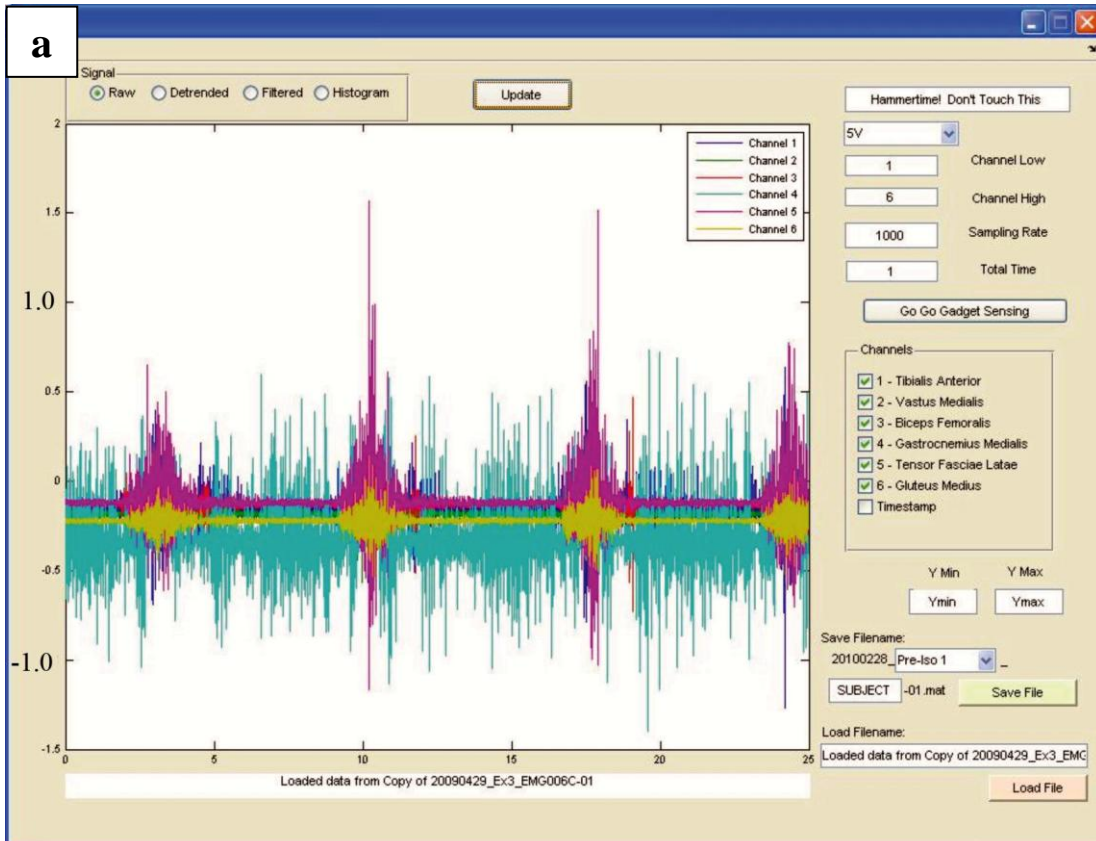
During testing sessions, Dr. McCombe Waller was responsible for prepping and placing the electrodes on each patient before data collection. For each of the four

patients, we collected EMG data of an isolation exercise for each target muscle, as well as several repetitions of all four exercises, shown in Figure 12. The purpose of the five isolation exercises was to develop baseline readings of the strength of each target muscle. We repeated this data collection three times for the paretic leg and once for the healthy leg, over the span of several data collection sessions. The data taken from the non-paretic leg was intended to be used as a template for what the EMG signal from the paretic leg should look like, but the data was too dissimilar to be compared in this way. We also completed one testing session for each of the control subjects using the same testing procedure. This data was also not useful as a relevant template for the paretic data. Data collection generally involved one team member demonstrating and guiding the patient through each exercise, one team member controlling the data collection software on the system PC (to be described in further detail in the next section), and one or more members for taking video and recording notes. To provide comparison to the hemiparetic patient data, we also collected data from four age-matched healthy control subjects using the same testing procedure described above.

Signal Conditioning

Our MATLAB interface for data collection is shown in Figure 13. Our initial signal conditioning process converted the raw EMG data into a more convenient form for our classification program. The data acquisition card was set to record samples at 1000 Hz. The raw signal was detrended, rectified, and then passed through a fifth-order low pass Butterworth filter with a 5 Hz cutoff frequency in MATLAB. For our purposes, 5 Hz was found to be under the lower limit of frequency content for

conscious muscle contraction. The higher limit of frequency contents of the signal were irrelevant since EMG signals have an upper limit of 2 kHz, so the Butterworth filter was used to remove these components as well (Basmajian, 1979). As the testing went on, newer versions of the program were created to add new features including a graphical user interface (GUI) and the ability to save and load data files. The final version of the MATLAB program is included in Appendix III.



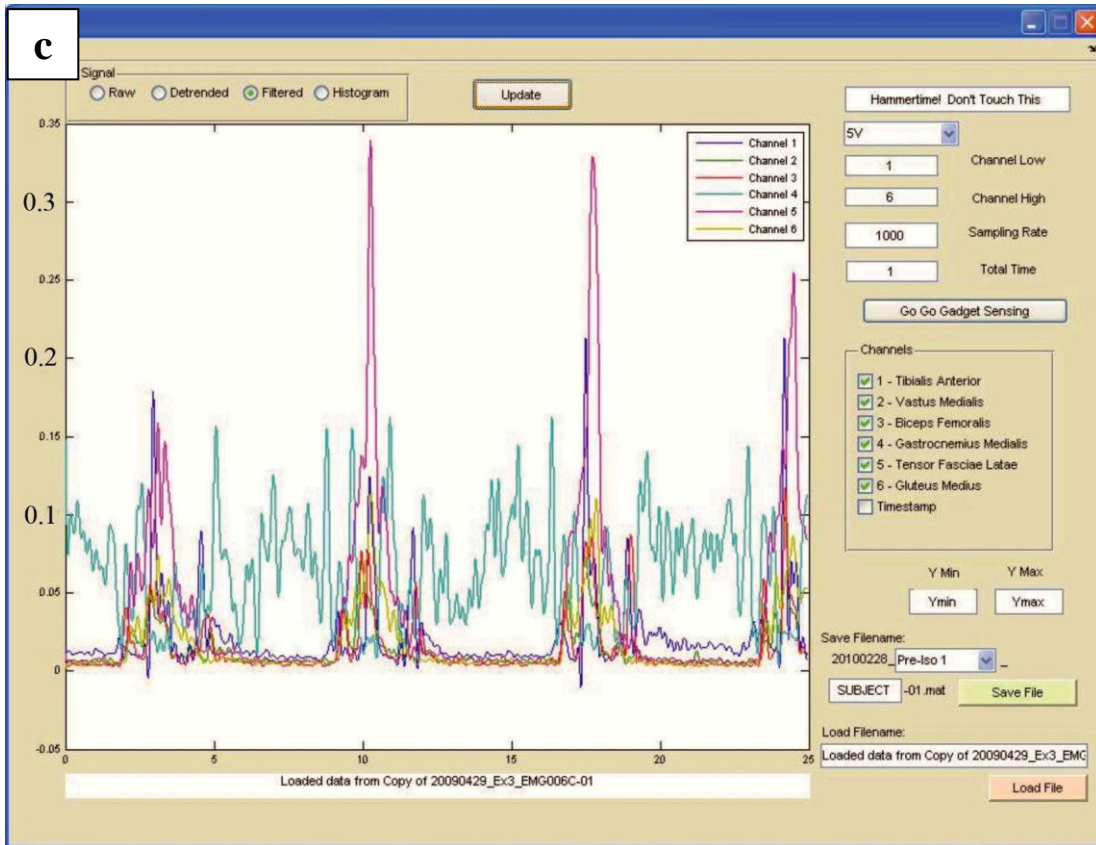


Figure 13: EMG data recorded from control subject EMG06, 4/29/2009. GUI program screenshots with (a) raw, (b) detrended and (c) filtered versions of data recorded from Exercise 3.

Figure 14 shows an example of the data comparison between a patient's paretic and non-paretic tibialis anterior. Note in particular the scales for each graph, which show the difference in EMG signal strength, representing the discrepancy in muscle strength. Post-conditioned results of these signals are also shown, after detrending, rectification, and further filtering. The conditioned signals were then used to determine the amplitudes and thresholds used by the Stage Two grading programs.

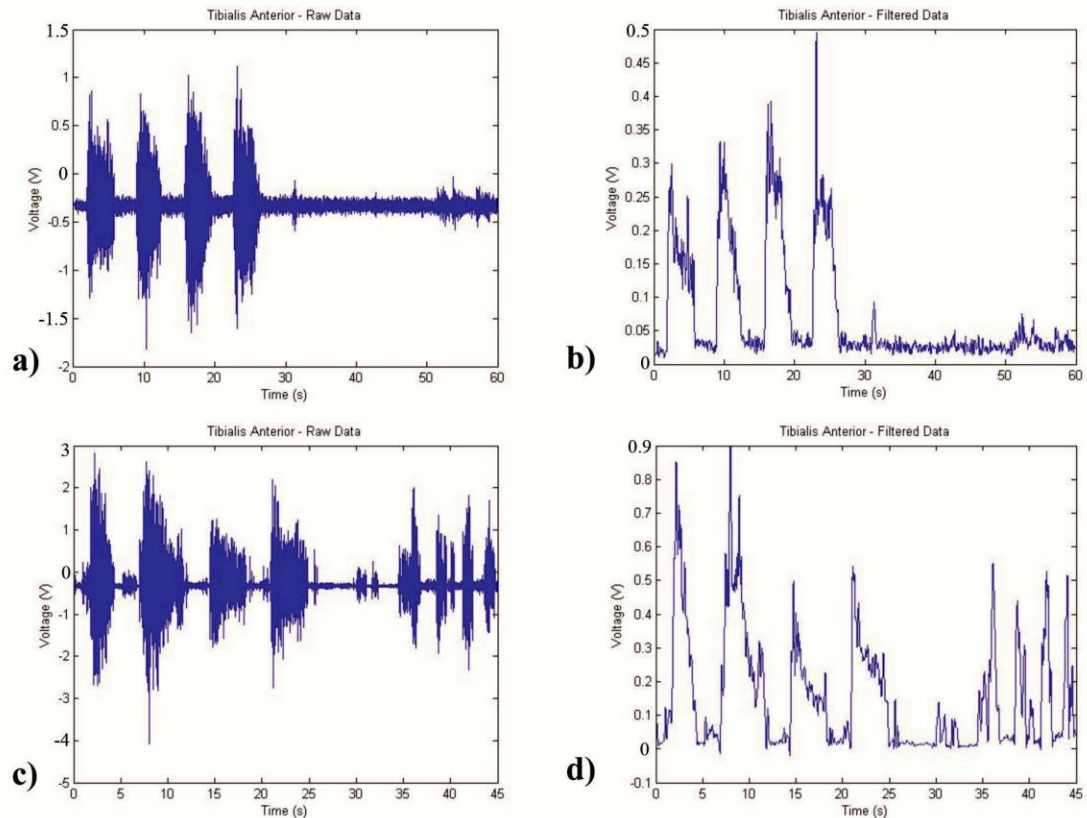


Figure 14: EMG data recorded from tibialis anterior during a step support exercise by patient 3. a) Raw data from paretic leg. b) Detrended, rectified, filtered data from paretic leg. c) Raw data from nonparetic leg. d) Detrended, rectified, filtered data from nonparetic leg. Data in (a) and (c) were amplified by a factor of 2500 and bandpass filtered from 10 Hz to 2 kHz by the EMG sensor boards. Data in (b) and (d) was additionally detrended, rectified, and filtered using 5th-order lowpass Butterworth filter.

Stage Two

As opposed to Stage One, which focused on EMG data collection from patients, Stage Two focused on data interpretation in order to provide rehabilitative feedback. Stage Two therefore integrated both real-time visual biofeedback, to allow patients to monitor muscle activation, and game play that rewarded patients for improving exercise performance. LabVIEW was used as a front end real-time data collection and display module. User-selected options determined which muscle signals were displayed in real-time as the patient was performing an exercise. Each

patient's performance was related to an individually-set benchmark, obtained from three calibration trials of each exercise, performed before playing the game. It was necessary to calibrate the system to each individual patient because activation levels vary for different muscles and between different patients. As a result, the system is custom-tailored to each patient's individual rehabilitation needs. The patient exercise data was scored and classified using the classification program, and the results were used to generate a swing for the on-screen videogame avatar. The quality of the shot was determined by the patient's performance of the exercise. Computer macros were written to interface the videogame with the classification program.

Videogame Interface

Our desired visual interface was to have a windowed version of *Tiger Woods PGA Tour 2004* in the center of the screen with various feedback meters on the borders of the screen. Because *Tiger Woods PGA Tour 2004* does not have a built-in window mode, we had to improvise by running a second instance of our operating system in a window. To accomplish this feat, we used Sun Microsystems' VirtualBox program and ran the Tiger Woods videogame in this windowed instance of Windows.

To allow for communication between the signal processing program and the videogame, we used an open-source macro program called AutoHotkey. The processing algorithm evaluated the user's exercise and returned a numerical value corresponding to the activation level of the desired muscles for the particular exercise. This value was written to a .txt file named "goodness.txt". We wrote an

AutoHotkey script that continually monitored this .txt file. When a value was written, the script would then generate a golf swing corresponding to the value.

EMG Visual Feedback

For Stage Two testing, a LabVIEW program was used to provide real-time visual indicators of muscle activity during an exercise. Depending on the exercise, the program automatically displayed the real-time EMG data of the muscles of interest for that particular action. The sliders displaying the data also change color when certain benchmarks have been exceeded (from blue to red), so the patient can easily see when they have performed well. Furthermore, the program not only collects data but also scores the patient's performance based on amplitude in relation to a set of calibration amplitudes. The difficulty and sensitivity of the scoring system is adjustable and can be modified during a session with the patient easily. The score is passed to a programmed macro that controls an avatar in *Tiger Woods PGA Tour 2004*. The avatar hits the ball in the driving range mode with the distance and accuracy of the hit determined by the performance of the patient, with a higher score resulting in a longer, straighter shot.

As can be seen in Figure 15 below, there are three vertical slide indicators on the left side of the picture that represent the two important muscles for the particular exercise and a third indicator to monitor a spastic muscle, if desired. There is also a difficulty dial located just beneath the vertical slide indicators that shows the chosen difficulty, which can vary between 1 and 10, with 10 being the most difficult. The benchmark activation is marked by the solid black line across the slider and varies based on the difficulty level.

The slide indicators represent *concurrent feedback* that the patient receives while performing the exercises. This form of biofeedback was highlighted in our literature review as crucial for successful EMG biofeedback therapy. The Tiger Woods game itself also serves as *terminal feedback*, giving the patient an easily-understandable final evaluation of their degree of success in performing the previous exercise. Our primary motivation in including this terminal feedback was to engage the patient in a goal-oriented and motivating task to maintain patient interest and compliance with the exercise regimen.

The system does not use *faded feedback*, or feedback that is faded out over time as the patient learns the exercise on his/her own. However, the adjustable difficulty dial does allow for adjustment of the game so that it adapts to improvements in the patient's ability. Without adjustment, patients may quickly improve and find the game too easy, but with adjustment the patient will not find it too easy to score well in the game even after a period of improvement. Furthermore, if the patient masters the exercises to the point where the sliders are redundant or annoying and only desires to play the Tiger Woods game, a simple button could be added to make the sliders invisible.

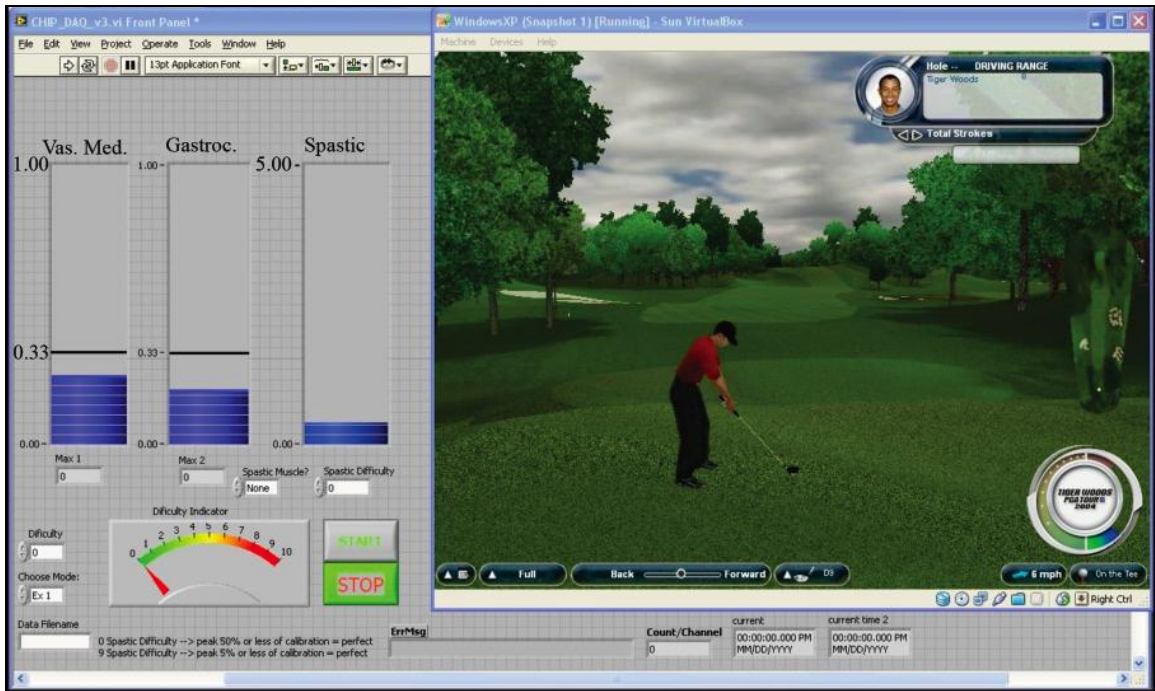


Figure 15: Data collection and biofeedback displayed with *Tiger Woods PGA Tour 2004*.

Figure 16 below shows how the vertical slide indicators change color if the patient has exceeded the benchmark level in accordance with the chosen difficulty level.

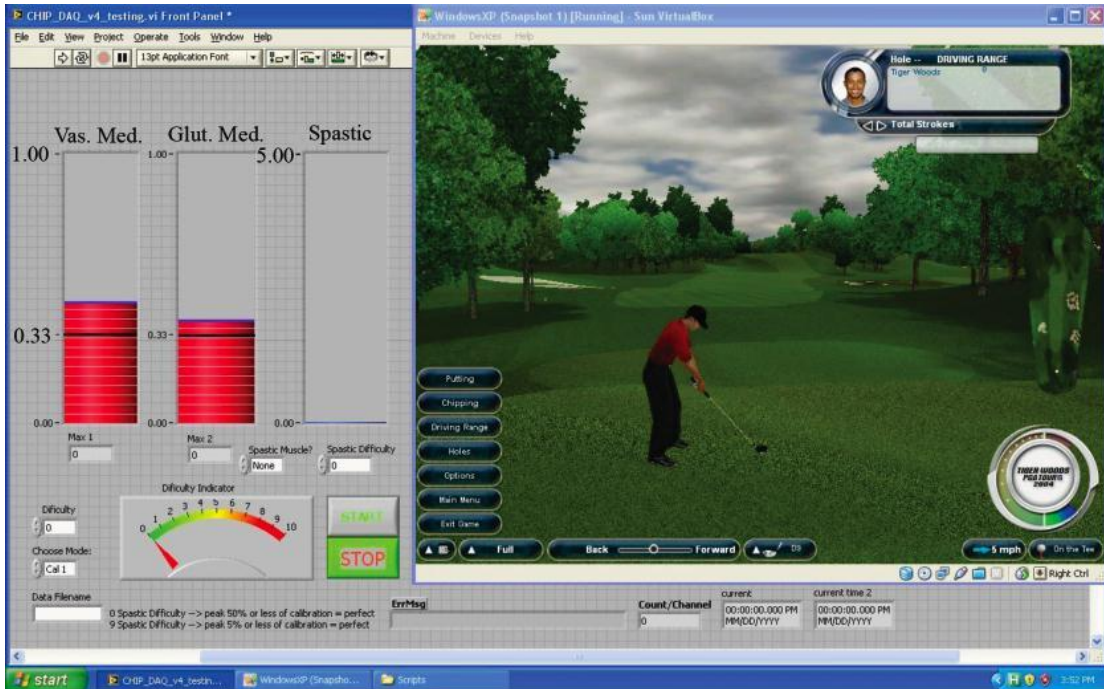


Figure 16: Slider bar when color change activated.

The LabVIEW program has another feature that remains hidden while the game is in play. It displays a plot of the activations of the two targeted muscles over time for each exercise after the patient has completed the exercise. The patient or therapist can refer back to this chart to see a graphical recording of the muscle activation over the entire time that the patient performed the exercise as a secondary reference, similar to the output of the previous MATLAB program from Stage One.

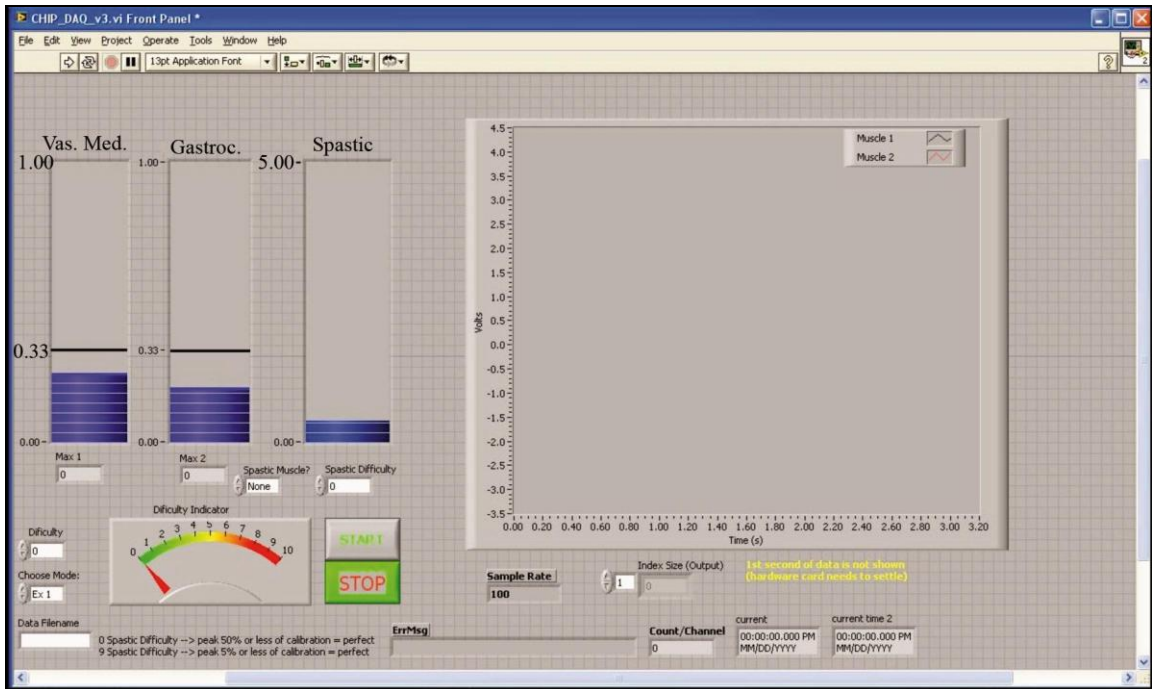


Figure 17: Close up of LabVIEW GUI with signal graph. The videogame is hidden.

Classification

Measuring improvement in muscle recruitment levels offers valuable feedback that can be used to improve patient rehabilitation. However, it is also important to look at the sequence of muscle activations when evaluating exercise performance. At the current time, this is an ongoing effort and both the attempted methods and ongoing avenues of research will be discussed in this section. In order to study the muscle activation sequences, we attempted to develop a signal classification method that would allow us to categorize incoming EMG signals into the four different exercises. This serves two primary purposes. The first is to quantify exercise performance based on the identifying characteristics of the four exercises. The classification process was confounded by the difficulty in determining normative data

to classify against, however. The exercises performed by the patients were unique and therefore, no standard template of EMG activation sequence was available.

The second purpose is to use the EMG classifications to enhance gameplay. For example, in a golfing game, the four exercises could be linked to the use of differing types of golf clubs. Implementations of the following classification methods have been done in MATLAB.

Linear Discriminant Analysis

In order to classify and score the EMG signals, our team attempted to use a technique called Linear Discriminant Analysis (LDA). In order to use LDA, the signals must first be represented as a vector of numbers, each of which quantifies a given signal characteristic, such as frequency content, amplitude or slope of the signal over a specific time frame. Therefore, the determination of which characteristics to use in this feature vector is extremely important for classification, and will be discussed at a later point.

LDA is a mathematical technique which can be used to determine the optimal n -dimensional plane, where n is the length of the feature vectors, which best separates two given sets of training data corresponding to two different classes of interest. In other words, LDA splits the n -dimensional space into two domains, each of which corresponds to vectors which have a higher probability of indicating the corresponding class. LDA thus attempts to determine into which class, **A** or **B**, any given vector \mathbf{x} belongs. Mathematically, if $P(\mathbf{A}|\mathbf{x}) > P(\mathbf{B}|\mathbf{x})$, then \mathbf{x} will be characterized as belonging to class **A**, otherwise it will be classified as belonging to

class **B** (Friedman, 1989). The solution to this problem is not trivial, but is made easier by the Bayes' Theorem, which states:

$$P(\mathbf{A}|\mathbf{x}) = P(\mathbf{x}|\mathbf{A})P(\mathbf{A})/P(\mathbf{x})$$

Therefore, a vector \mathbf{x} will be classified as belonging to class A if

$$P(\mathbf{x}|\mathbf{A})P(\mathbf{A})/P(\mathbf{x}) > P(\mathbf{x}|\mathbf{B})P(\mathbf{B})/P(\mathbf{x})$$

Multiplying by $P(\mathbf{x})$ on both sides:

$$P(\mathbf{x}|\mathbf{A})P(\mathbf{A}) > P(\mathbf{x}|\mathbf{B})P(\mathbf{B})$$

Assuming that $P(\mathbf{A}) = P(\mathbf{B})$ for the classification problem, this is reduced further to:

$$P(\mathbf{x}|\mathbf{A}) > P(\mathbf{x}|\mathbf{B})$$

This can be restated using matrix notation as:

$$\mathbf{u}_A \mathbf{C}^{-1} \mathbf{x}^T - 0.5 \mathbf{u}_A \mathbf{C}^{-1} \mathbf{u}_A^T > \mathbf{u}_B \mathbf{C}^{-1} \mathbf{x}^T - 0.5 \mathbf{u}_B \mathbf{C}^{-1} \mathbf{u}_B^T$$

where \mathbf{u}_A is the average and \mathbf{C} is the covariance matrix.

LDA solves this equation to find the plane that splits the space properly. On one side of this plane, data corresponds to class **A**, and the other side corresponds to class **B**.

Determination of Features

The original choice of feature selection was to use average signal amplitude of each of the six EMG channels over six equally spaced time windows in the data. This showed good classification performance when other signals from the same subject were included, but poorly otherwise. In order to rectify this error, genetic algorithms were used based on average amplitude over differing frames. Genetic algorithms are stochastic methods of optimizing solutions by treating potential solutions to a given problem as individuals within a population. A random population of solutions is then

generated and evaluated using a problem-specific fitness function. Based on the individual's fitness relative to the fitness of other population members', a number of members “die” probabilistically. The population is restored to its original size by the mating of other individuals in the population. This mating is also problem specific.

In our study, individuals were represented by randomly generated time windows over which the signals were averaged to create smaller vector representations. The fitness measure was based on the ability of these vectors to be used as input to an LDA classification and produce accurate classifications. Specifically, for data from n patients, LDA was used to classify patient i based on training data from the other $n-1$ patients. The fitness score was the root-mean-square of the true positive and true negative fractions across all individuals. Mating between individuals produced offspring with half of their windows from one parent and half from the other. Time windows were passed randomly, with time windows which contributed to higher fitness scores being selected preferentially.

Other methods of representing solutions were attempted, but showed generally poorer results. These included counting the number of times muscles toggled between contracted and relaxed states within specific time windows.

Dynamic Time Warping

Classification of EMG signals to determine sequential correctness is complicated by the fact that different people may perform the same action differently. Local accelerations and decelerations cause features to be misaligned between different people, even though the overall sequence is preserved. For example, during exercise two, patients could choose to keep their legs lifted for longer periods of time

relative to the time spent lifting and lowering the leg. Although previous time-window based attempts to classify the data attempted to compensate for this by averaging broad time windows, time resolution was lost and the results were not able to account for these timing differences satisfactorily.

Dynamic Time Warping (DTW) aims to solve this problem by finding the best alignment between two signals while permitting local temporal distortions, or time slowing and speeding up. The overall sequence of the signal is preserved, but differences in relative timing can be reduced. DTW is accomplished by finding the least-cost path through a cost-matrix between the two signals which are being aligned (Ellis, 2003). For a signal f of length m being aligned to a signal g of length n , the cost matrix is an $m \times n$ matrix. In each element (i,j) of the matrix, the cost between $f(i)$ and $g(j)$ is calculated. Currently, we are experimenting with two different cost functions. The first cost function is simply the absolute value of the difference $|f(i) - g(j)|$. Additionally, the cost function corresponding to the absolute value of the difference of gradients $|f'(i) - g'(j)|$ is being investigated. Our project currently uses the open source implementation of DTW provided by D. Ellis (2003).

In order to generate appropriate templates, the data for each exercise was visualized in sequence in order to determine typical data patterns. From this qualitative investigation, it was determined that the four typical patterns to see in the data were signals with one through three peaks, or else no significant peaks.

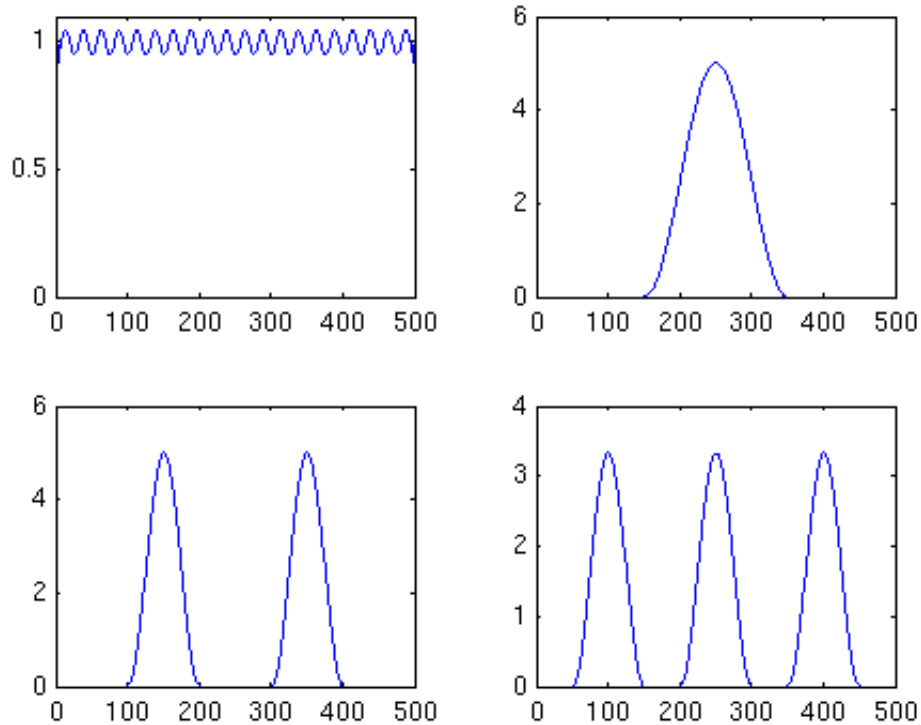


Figure 18: *Different templates were used to detect features in the data.* Based on qualitative study of the data, it was found that most signals corresponded to having one, two, three or no peaks. Therefore, these templates were chosen.

Determination of which template was best matched to the signal was accomplished by minimizing the root mean square of the difference between the input signal and the aligned template.

60 Hz Electrical Interference

Electrical power in the United States is delivered through wall outlets at 120V and 60 Hz AC. This high-voltage electricity source can become capacitively coupled between the sensor boards and either devices connected to the power outlet or the 120V supply wires themselves. This effect can be observed by taking the Fourier transform of the data, where interference can be seen as a higher-amplitude spike at a frequency of 60 Hz (Figure 19).

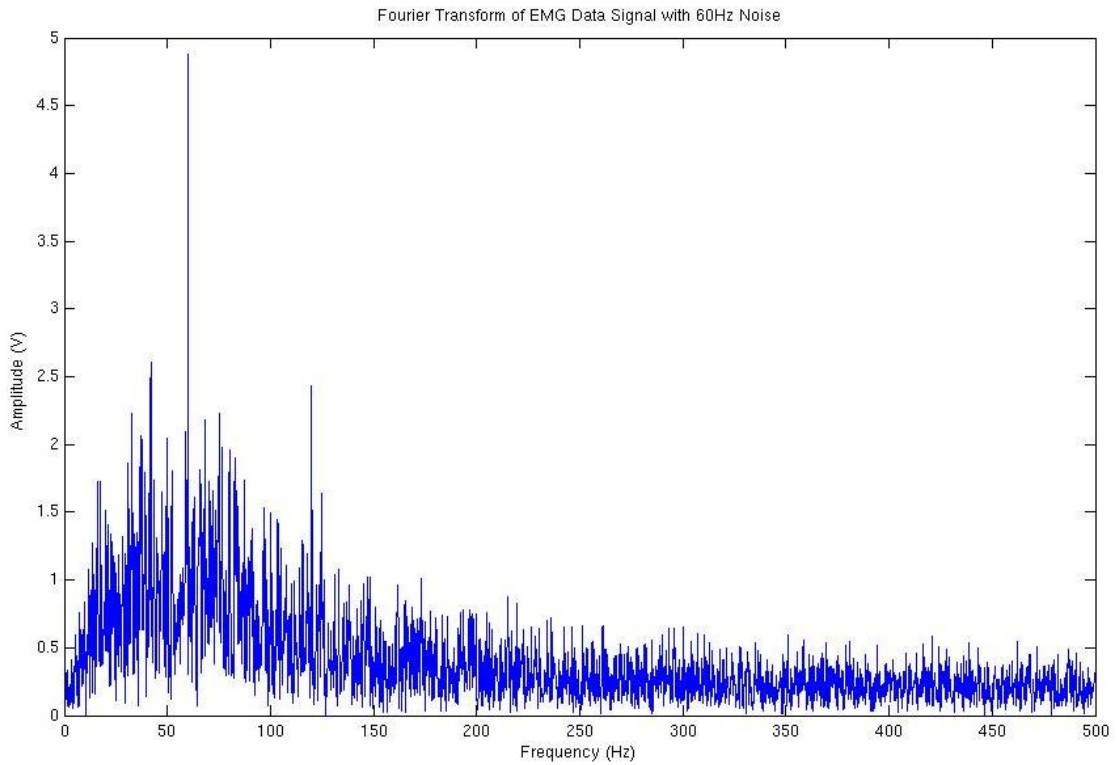


Figure 19: Signal demonstrating 60 Hz Noise

Although higher-amplitude spikes were observed at 60 Hz for many of the collected EMG signals, it was also absent in many of the collected EMG signals. Furthermore, filtering this peak out completely resulted in minimal changes to the observed signal envelope (Figure 20).

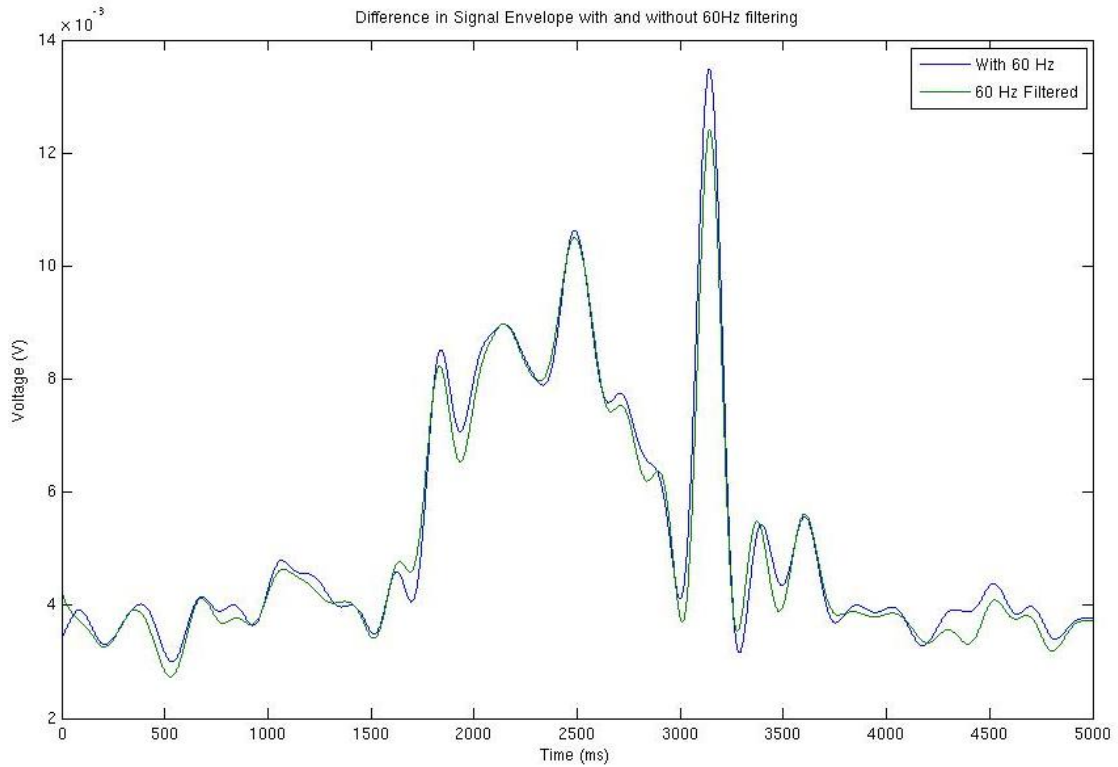


Figure 20: Signal demonstrating minimal change in the EMG envelope profile before and after filtering out 60 Hz noise. Specifically, Fourier transform peaks corresponding to 59-61 Hz were set to 0.

However, in the interest of reducing the risk that 60 Hz noise substantially affects signal quality, we have decided to filter out the 60 Hz noise during the digital processing of the collected signals.

Chapter 4: Results

Stage One

During each data collection session, we recorded EMG data for four different exercises, each performed three or four times. We monitored six muscles at all times, although each exercise required the use of only two muscles in particular that we represented using visual biofeedback to the patient. The main difference we observed between EMG data from hemiparetic patients and control patients EMG data was the amplitude of the signal, which was much lower for hemiparetic patients than for controls. For example, displayed in Figure 21, for both subjects tested May 20th, 2009, the EMG peaks for both the hip flexor and knee flexor were about one tenth in magnitude for the hemiparetic patient as for the control subject. Data for a control subject is displayed in Figure 22. Observe the scale on the left-hand side of the graphs for perspective.

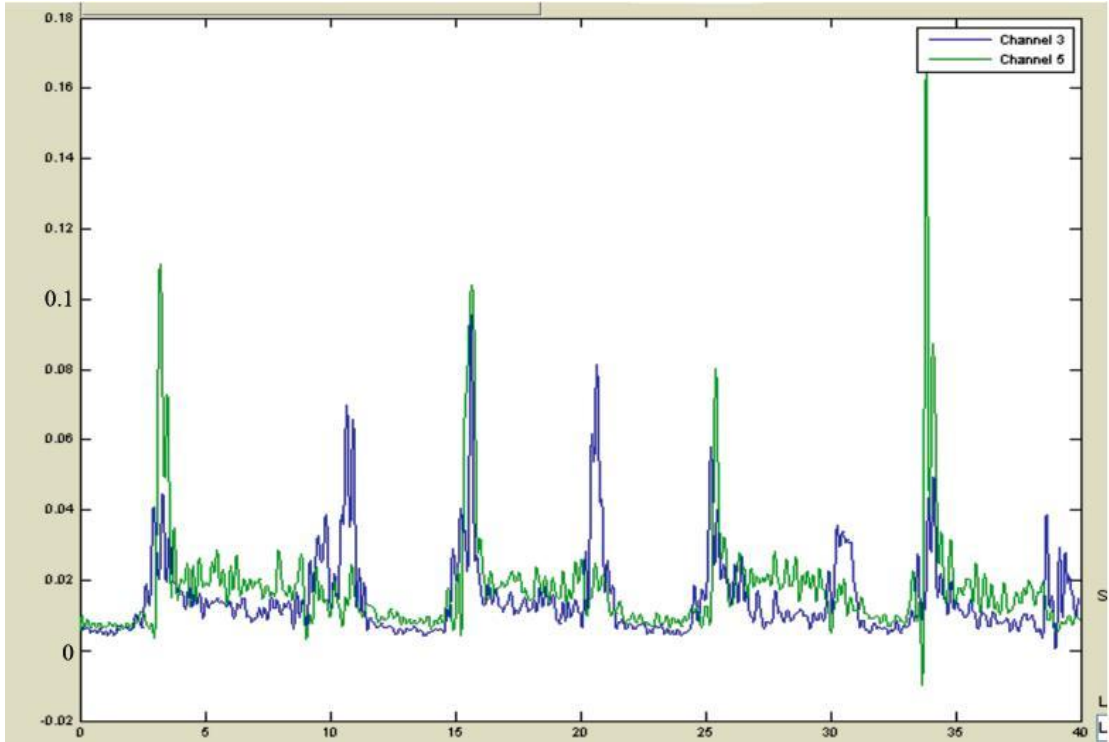


Figure 21: EMG02 data for hemiparetic patient, Exercise 2, May 20th, 2009. Green line is for TFL, blue line is for biceps femoris.

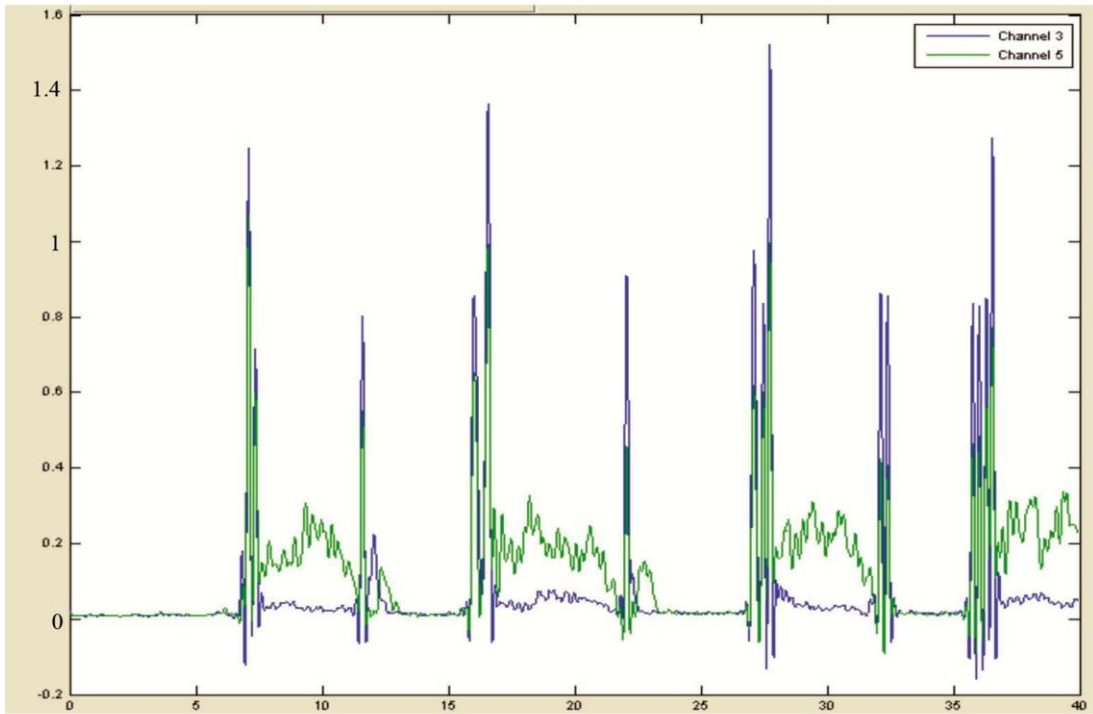


Figure 22: EMG08 data for control patient, Exercise 2, May 20th, 2009. Green line is for TFL, blue line is for biceps femoris.

The low amplitudes complicated the task of designing a signal classifier to determine which exercise was performed. Specifically, the spikes in the main muscles of interest in the hemiparetic population were less distinguishable from background noise than the spikes in other muscles when exercises were performed. In fact, the hemiparetic data was often much worse. Figure 23 shows a lower-functioning patient who performed the same exercise as the subjects above, but in this instance it was difficult to ascertain when the exercises were initiated. Even in ideal situations, classifying the signal was still challenging. Potential complications arose from background noise, signals from other muscles, hardware inconsistencies, and the need for a classifier to distinguish between all four exercises in deciding which exercise was attempted and when it occurred.

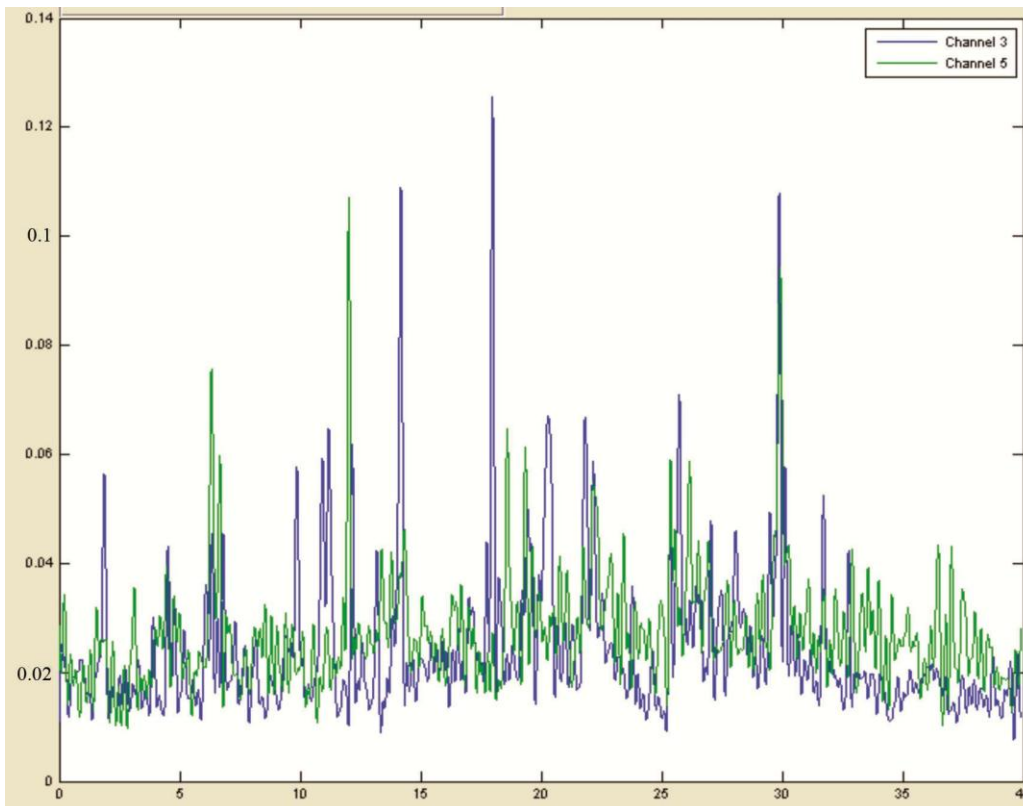


Figure 23: EMG03 data for hemiparetic patient, Exercise 2, April 3rd, 2009. Green line is for TFL, blue line is for biceps femoris.

The last complication we encountered from the hemiparetic EMG signals was an inconsistent ability across paretic patients to perform certain “isolation” exercises. We initially hoped to use these to develop a baseline reading of the strength of each muscle. Since we monitored and evaluated EMG from six different muscles, we anticipated that the EMG readings from the muscles of interest for each patient's performance of the actual exercises would be calibrated against the scores of those same muscles during the initial isolation exercises. This was intended to compensate for the fact that, especially among hemiparetic patients, different subjects have different strength levels and body types. For example, some patients may register EMG readings of around 0.2 V for a muscle of interest during a specific exercise, when that exercise would be characterized by an EMG amplitude that was at least three times than that for the same muscle during the isolation exercise.

Unfortunately, however, we found that different patients exhibited significantly varying levels of difficulty in performing the isolation exercises themselves. Specifically, they not only had different strength levels, but different *relative* strength levels between muscles. Thus, one patient's exercise might be characterized by three times the isolation exercise, while another's might be five times. To address this issue, we scrapped the whole process of calibrating exercise scores by separate isolation exercises. We then proposed to calibrate muscle-specific EMG scores during exercises by initial repetitions of those very same exercises, taken at the beginning of a testing session. This process was put into place during Stage Two, and was intended to aid in our ability to use the EMG data from the exercises to not only score the exercises, but determine which exercise was attempted.

After Stage One testing, we created a new version of the data collection program to incorporate EMG biofeedback during the exercises. EMG biofeedback has been used since the late 1960s as a part of stroke rehabilitation. Feedback is given to indicate that a specific neuromuscular pathway has been activated. Theoretically, undamaged pathways can be recruited to assume the functionality of those that have been damaged (Glanz et al., 1995). Furthermore, EMG biofeedback has been shown to be more effective at improving functional abilities if the training is done with task-specific actions (Huang et al., 2005).

Stage Two

Determination of Features

Initially, vectors were reduced by averaging the signals over six equally spaced time windows. Because the data tracked six muscles, this produced 36-element vector representations of each exercise. When first estimating the ability of these vectors to be used for signal classification, each trial was classified using all other data as training data for LDA. This method showed reasonable rates of classification, approaching 50 to 60 percent correct classification. In order to improve this rate, a genetic algorithm was used to find the optimal set of time windows in order to classify the data. After several thousand generations, it was found that classification could be as high as 90 percent. However, when the data was separated by person and each exercise was classified using the data trials collected from other people, the classification rate immediately dropped to roughly 20 percent. Attempting to use genetic programming to optimize the time windows resulted in classification rate of almost 80 percent; however, when looking at the resulting planes

of separation, it was found that there was no consistency between the planes generated with different sets of data. The lack of consistency convinced us that there was no underlying reason for the high rate of classification and that rather the small sample size of our data set permitted numerical manipulation to determine windows that would happen to produce good separation in our data set.

Despite this inconclusive result, this phase of testing confirmed some very important conclusions regarding the data. The first conclusion was that within a single person on a single day, the EMG signals collected for each of the four exercises were highly consistent. This matches physiological predictions, as it was expected that one person performing the same exercise twice would be more similar than two different people performing the same exercise. Additionally, it was indicative of the fact that the EMG signal is capable of differentiating between the different exercises since there were consistencies in the data. Second, it demonstrated that the temporal alignment between different people was too disparate in order to produce a good classification using any method which did not attempt to compensate for the differences.

Dynamic Time Warping

At the moment, dynamic time warping (DTW) alignment appears to perform very well in determining the best-aligned template and in aligning the template to the signal. Representative figures are shown below in Figure 24. Red lines are the data, and white lines are the aligned template.

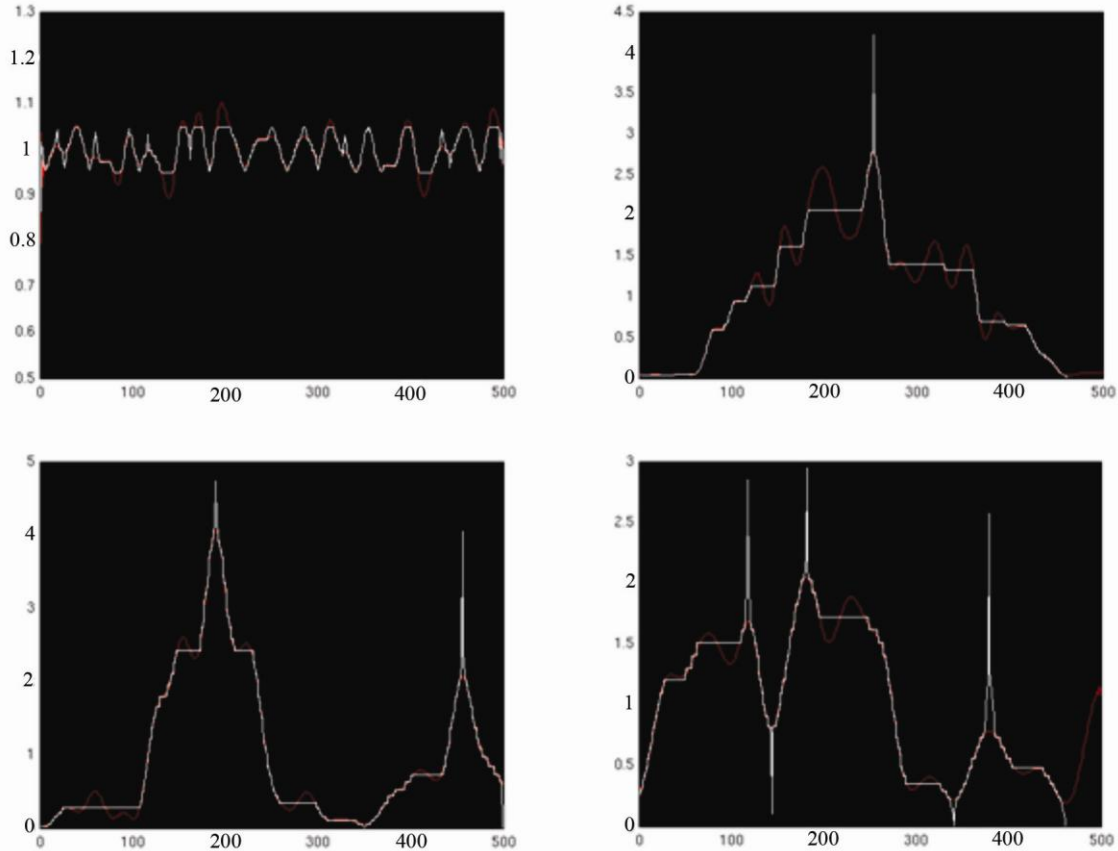


Figure 24: Alignment shows good matching of template and data. Additionally, the best template was automatically chosen based on minimizing the root mean square of the signal and template difference.

This alignment and automatic detection of optimal template is very encouraging, and we believe that it will serve as the first step towards better classification. Additionally, during our qualitative analysis of the data, we have begun to recognize patterns in the data. As a small example shown in Figure 25, for Exercise 3, we have seen that the biceps femoris tends to fire before the vastus medialis, and potentially again afterwards. Additionally, there may be co-contraction between the two muscles. Encouragingly, these patterns also make sense physiologically as the leg lifts, kicks out, is brought back in, and lowered.

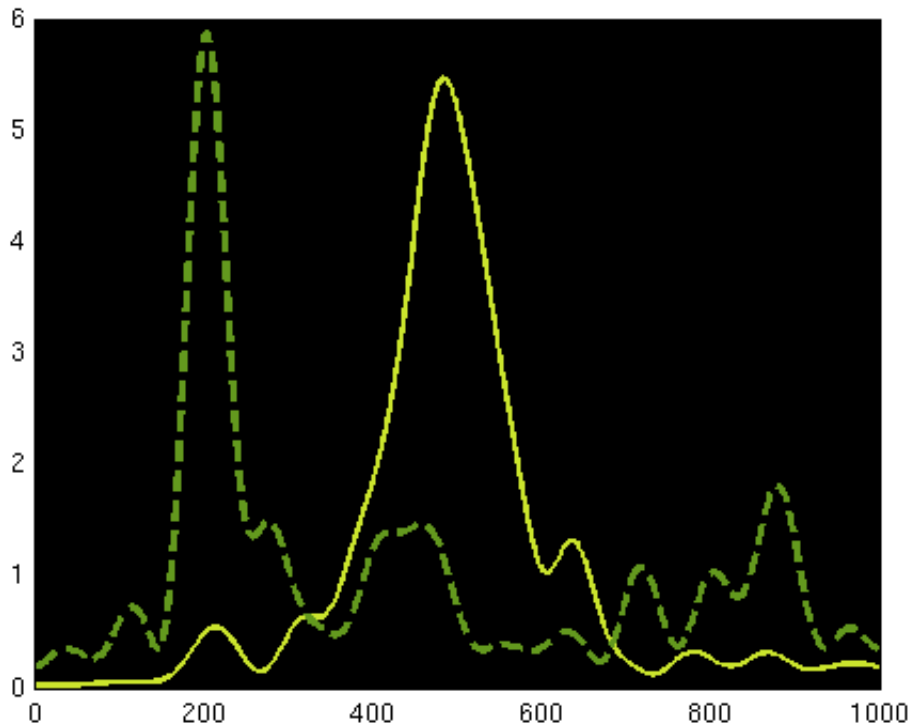


Figure 25: *Exercise 3 patterns.* Here is an example of the pattern which was seen regarding Exercise 3. The solid line is the vastus medialis and the dotted is the biceps femoris. Although this example is from data collected from non-hemiparetic patients, most data did not look as clear. Nevertheless, most showed similarities to this pattern.

Patients can be scored based on their adherence to the expected muscle patterns. Currently, Exercise 3 is awarded 15% for strong TFL or gluteus medius activation and 20-55% for the presence of a strong vastus medialis contraction occurring during the TFL/gluteus medius contractions. An additional 25% is awarded for the presence of flanking biceps femoris contractions, with penalties occurring for temporal overlap between the biceps femoris and vastus medialis contractions. The final 5% is rewarded for ankle flexion occurring between the vastus medialis and subsequent biceps femoris contractions.

This score can be integrated into the game by affecting the ability of the Tiger Woods avatar to aim the ball during shots. Additionally, the detection of incorrect muscle activation sequences using this method can be used to provide feedback to the patients about how to better perform the exercise.

Although this method of scoring exercises offers a promising approach to both detecting movement quality and providing rehabilitative feedback, there remain several concerns. Currently, the percentage distribution for scoring Exercise 3 was developed arbitrarily based on the perceived importance of particular signal features to proper exercise performance. These percentages could potentially be better determined based on the feature distribution present in the overall patient population. However, given the small sample size, implementing such a solution would be a poor choice for this study. Additionally, the arbitrary number of points associated with each feature is not necessarily important so long as the provided feedback allows patients to determine which exercise feature they need to improve upon.

A more critical issue arises when considering the scoring of Exercises 1, 2 and 4. From an EMG envelope perspective, exercises three and four appear very similar, as the muscles involved in stabilizing a step landing are similar to those involved in kicking a leg forward. The inability of the above scoring mechanism to consistently score Exercise 3 signals higher than Exercise 4 signals is therefore troubling, as it suggests that some of the provided feedback may not be appropriate. For instance, a user may be able to fool the system into believing a leg kick was performed when instead a modified lunge was performed. Exercise 2 demonstrates a different activation pattern, but has not yet been well-defined. Exercise 1 is believed to be

inconsistent to the point that it is considered “patternless.” This arises from both the fact that there are a large number of methods in which a weight shift might be accomplished, and also from the fact that a weight shift primarily involves contraction of a hip abductor on the *non-paretic leg*, which was not monitored at all in our study.

Some of these issues might be addressable through the consideration of frequency-domain signal information. At the current level of investigation, contraction patterns were determined solely based on time-series amplitude information. It is thought that the incorporation of time-series frequency information into our classification scheme could provide additional data that would allow better discrimination between the exercises. To this end, an initial investigation into the frequency-domain characteristics of the collected EMG signals has been conducted.

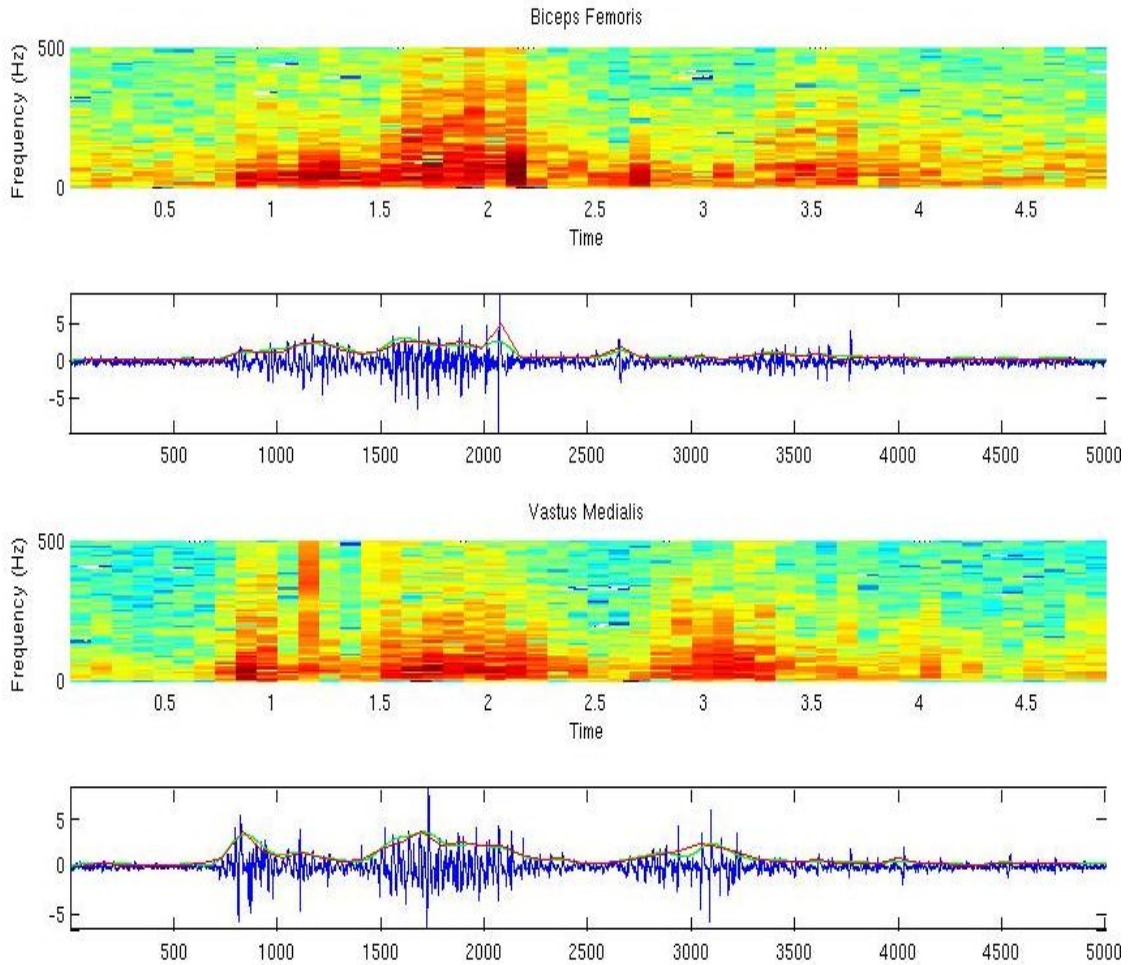


Figure 26: Typical spectrogram of control patient EMG signal for biceps femoris (top) and vastus medialis (bottom). Spectrogram is shown with corresponding raw EMG signal. Green line represents current enveloping algorithm result. Red line represents envelope generated using first principal component of the frequency distribution.

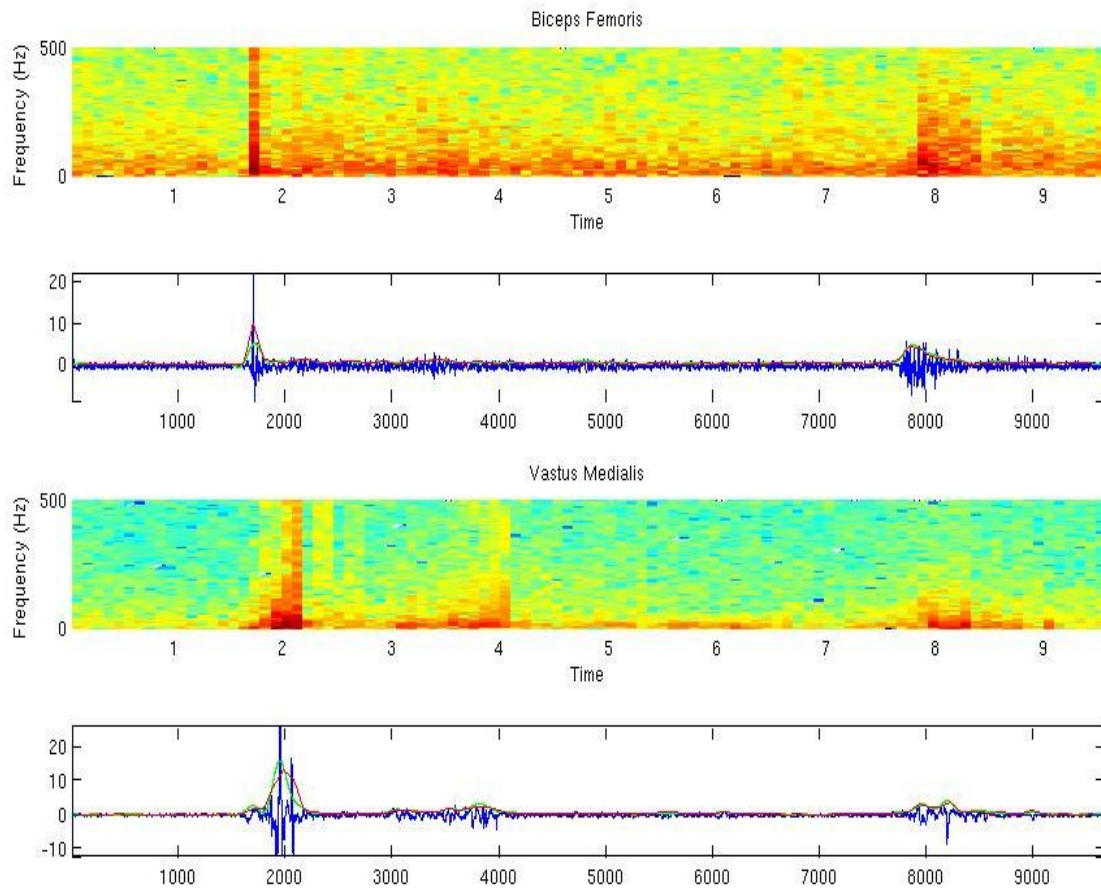


Figure 27: Typical spectrogram of a paretic patient EMG signal for biceps femoris (top) and vastus medialis (bottom). Spectrogram is shown with corresponding raw EMG signal. Green line represents current enveloping algorithm result. Red line represents envelope generated using first principal component of the frequency distribution.

Shown in the above figures are the typical spectrogram for control and paretic patients (Figure 26 and Figure 27, respectively). It can be seen that for both control and paretic patients, muscle contractions are characterized by increased amplitude of signal frequency components, especially high-frequency components. There is evidence of a build-up of higher-frequency components in the earlier contraction

stage, followed by peak, followed by a reversal process where the higher-frequency components dissipate first.

It can be seen, however, that the paretic patient may have more difficulty in achieving the high amplitude high-frequency components, evidenced by the central peak of the vastus medialis plot for the paretic patients. This suggests that patients might be awarded for increased high-frequency activation in the collected EMG signal. The presence of high-frequency components is also valuable because it provides a certain level of noise immunity from 60Hz noise, and also confirms that a 60 Hz notch filter will not significantly alter the frequency distribution during contraction.

When studying the spectral distribution of the EMG signal using the short-term Fourier transform, it appears that the spectral distribution of the signal during muscle contraction can be characterized by a random distribution with the highest power density below 300 Hz. From this, we believe that without more sophisticated techniques it will be difficult to extract more information from the frequency content than is already extracted via envelope generation techniques. These results are consistent with a pink noise model of the EMG signal during contractions.

At this stage of investigation, it is unclear whether or not frequency-domain content could help to discriminate between Exercises 3 and 4. If differences are seen, we expect that they will appear either in the sustained frequency distribution difference during the leg kick/leg land and stabilization regimes, or else in the frequency content build-up and roll-off. In order to detect these features, a wavelet

transform might allow better visualization of the signal frequency content, as it would better preserve time and frequency resolution.

Qualitative Results

During Stage Two testing at the laboratory in UMB, we were able to test our complete EMG videogame system on our original four hemiparetic patients. We used a very similar testing protocol as Stage One, in that we had one teammate operating the computer with the LabVIEW display, one teammate coaching the patient through exercises, and one or two teammates taking notes and videos. Each patient completed a single training session consisting of all four exercises. Before gathering data for each exercise, at least three calibration values were collected. The values were purely based on the amplitudes of muscle contraction for that particular exercise. These values set the baseline average that the patient had to strive for and improve upon. Then, the patient performed multiple trials of all four exercises, receiving real-time feedback from our LabVIEW interface with the Tiger Woods game. After doing the required set of tests of each exercise, the patient could opt to repeat specific exercises to perfect their performance.

We noticed a variety of user styles amongst the patients when using our system for gait rehabilitation. One patient tended to just stare at his legs while completing the exercises, rather than look at the real-time slider bars on the computer monitor. However, he did watch Tiger Woods' performance in the driving range after he completed an iteration of an exercise. He suggested we add an audio feature to complement the slider bars to accommodate those who do not look at the monitor directly while doing the exercises. Other patients used the real-time slider bars to

gauge their muscle contraction and accuracy while executing the exercises. All the patients were intrigued by their ability to track their performance throughout the session based on their driving range distance scores.

There were a few problems with our system that we noticed during Stage Two testing with the patients. The pop-up feedback messages that showed up while the patients were performing exercises did not always provide accurate guidance. During a few trials we may have experienced some technical difficulties with the macro communication. This problem caused the patient to miss the ball when they took their swing after performing an exercise. The macro was easily reloaded and we did not experience this problem again. There were also a few minor issues with changing the names of the files. Ultimately, files should be named automatically, eliminating the need for an extra person.

The patients provided valuable feedback about the preliminary prototype of our system. One patient said he would play our game at home if there was a way for him to hook himself up to the system without any assistance. He stated that it would be nice to be able to connect our system to a big screen television at his house to enhance the overall experience. Other feedback was focused on our LabVIEW display. The patients and physical therapists both agreed that the muscle labels were too small to read. The muscles could also be written in basic terms such as “calf” because many patients are not familiar with the scientific names of muscles. Patients were captivated by the slider bars because they directly correlated to their efforts associated with the exercise. It would be more helpful if the bars remained red longer to give the user more time to see, instead of just briefly flashing red for one second.

The physical therapists commented that although the bars were going up, we still are not sure if the exercise is actually being performed correctly.

In regards to the *Tiger Woods PGA Tour 2004* game, the patients had mixed feelings. One patient thought it would be more enjoyable to choose his own avatar, such as a character that is more cartoon-like and larger than Tiger Woods appears. They also wished the driving range scores were much larger since we were merely reading out the scores to them. Another patient said he would enjoy a wider variety of games to play. Patients also suggested we add a video of someone demonstrating the exercises along with the patient to make home-use more effective. Another feature that patients would like to see is a scorecard showing all their past scores in the driving range, so they can track their progress across sessions. The physical therapists wished that the golf swing could occur more concurrently with the patients exercise performance.

Despite these suggestions for improvements to the prototype, our system did show promise in making rehabilitation more fun. In terms of motivation, many of the patients desired to continue the session after the initial regimen of four exercises in order to top their previous scores. In particular, one patient would not move on to the next exercise until she surpassed her personal best driving range score. We recalibrated Exercise 2 for her, since she had improved significantly since the start of the session. It was evident that she was actually performing the exercise more accurately. For another patient, we decided to test out the difficulty variable by adjusting the difficulty indicator from the default setting of 1 up to 2 then to 4. Since he started to get numerous high scores in the 300 yard range with muscle amplitudes

above baseline, we increased the difficulty level. This adjustment caused him to have to try harder to get those high scores, allowing him to become considerably better at the exercise. The patient said the difficulty adjustment made him want to push harder.

We are pleased with these qualitative results, however informal they are. We hope that future work can support these observations with a more quantitative process.

Chapter 5: Conclusions

Challenges & Limitations

One of the main challenges of our study was gathering a large test population to collect data from. Establishing the partnership with the Department of Physical Therapy and Rehabilitation Science at the UM School of Medicine was a major step forward in validating our project by testing our system on actual post-stroke hemiparetic individuals. However, among the patients receiving therapy from this institution, only four patients were able to participate in our study, along with four healthy, age-matched controls. The small test population of hemiparetic patients may have limited our study in terms of applicability for all post-stroke individuals, especially given the wide range of ability/disability among this target group. However, our results still demonstrate the possibility of such a system being realized for stroke rehabilitation.

We encountered many challenges during the testing phases of Stage One and Stage Two. Our system had numerous technical problems that caused delays in testing on subjects in Baltimore. It was difficult to establish a consistent training schedule with the patients due to the need for repairs to our system.

Our first hardware problem with our sensor boards was that they were too fragile, with the electrode wires sometimes falling off and potentiometers becoming miscalibrated over time from getting bumped. These problems would sometimes manifest during testing sessions themselves, leading to delays in testing or even loss of data collected when the system was not functioning properly. One such problem in

particular was the button snaps becoming disconnected from the sensor electrodes. Other times we detected no problems or obvious failings of the hardware during testing but still returned to campus to find a channel or two that had collected nonsensical data. We attempted to remedy these problems by encasing our sensor boards in a potting compound to make them sturdier, but this provided only a partial fix.

We also experienced delays when the power supply of our computer failed during our first testing session in which we attempted to test the full integrated iEMG video game system. The power supply failed abruptly so we had to abandon testing for that day and wait weeks to return after a new power supply had been procured. Aside from hardware problems, we also experienced several delays resulting from coordinating schedules between ourselves, the doctors at UMB and their patients. The initial rounds of testing were also delayed due to poor coordination of IRB submissions between ourselves and UMB.

Another challenge during testing was gel electrode placement. This procedure was a time consuming, intricate process, requiring up to 30 minutes to place electrodes on all six muscles of each subject. Electrode placement requires precision in order to ensure the EMG signals will be the strongest. To avoid further delays in testing, we started staggering our patient training so that the therapists would begin placing electrodes on the next patient, while the current patient completed their training. However, there were concerns over the first few sessions that EMG electrode placement was not precisely standardized, which could lead to inconsistencies in data collection. After the first few sessions, skin markers were

used to ensure electrodes were placed in the exact same location across different testing sessions.

The greatest challenge in our study was the classification algorithm. Many classification methods were attempted, but drawing concrete conclusions was very difficult. For the LDA algorithm, there was low success in classification when the training group only included data from the other patients not currently being graded. We believe the problem is due in large part to two factors – first, that hemiparetic EMG data across different subjects was very heterogeneous even within certain exercises, and second, that different individuals sometimes did the same exercise slightly differently. Even if neither way was 'wrong' they would yield different EMG signatures. To address this problem, we are looking to use a patient's own EMG data from the initial exercises they do to classify later exercises, since EMG data was more uniform within patients over time than across patients. Scoring exercises, which would be handled separately than simply identifying whether an action was Exercise 1 or Exercise 2 for example, would be done by measuring the amplitude of EMG data for certain target muscles for each exercise.

Contributions to the Field

Our team has many accomplishments and has provided contributions to the field of rehabilitation therapy. We have developed an EMG signal collection hardware and software system from the ground up. We tested this system on team members, a small group of hemiparetic patients, and a small group of healthy controls. We have a physical, working prototype that is ready to be tested in clinical trials to see if it can improve gait rehabilitation relative to traditional methods. While

we were unable to conduct such clinical trials ourselves due to limited time, we are optimistic about how our system would perform under such testing. This is because it was designed in accordance with important elements from the EMG rehabilitation literature, namely that exercises be task-specific (in this case gait) and that visual feedback be delivered in real-time.

Our further contribution is creating a system with not only these functional characteristics of other successful EMG training programs, but which is also fun and engaging because it transforms an integral part of the rehabilitation process itself into a way to play a golf videogame. While we found in our literature review instances of virtual reality used in interactive rehabilitation programs, as well as one instance of a simple computer game controlled by EMG data from one simple motion, we believe our contribution is the first truly integrated system for allowing seamless control of an established, popular videogame via functional rehabilitation exercises. Finally, we presented our research findings in the poster session at the 3rd International Congress on Gait & Mental Function in February 2010.

Chapter 6: Future Directions

There are many ways that our system can be used or improved to further evaluate its potential impact on physical rehabilitation. Some of these future directions were originally goals of this research, but could not be completed because of time and resource constraints. First of all, the results of the research reported here would be more significant if the system were thoroughly tested on a larger population of post-stroke hemiparetic individuals. Multiple test groups and a several-week training program are required to reliably evaluate the clinical benefits of the system. One test group would use the interactive EMG biofeedback videogame for gait rehabilitation training, while a control group would perform standard rehabilitation exercises without our system. The way in which data from these testing groups could be used to formulate further results is described below in “System Efficacy Testing: Pre/Post-Training Functional Assessments,” and is similar to the process described in “Randomized Controlled Trial in the Home Setting,” with the exception that it would occur in the clinic under therapist supervision.

System Efficacy Testing: Pre/Post-Training Functional Assessments

In order to determine how effective our system is for improving patients’ rehabilitation experience, we are interested in the quantitative improvement in patient gait performance as well as level of comfort and motivation while using the system. To determine these factors, patients will be evaluated according to the suite of functional assessments described below, once prior to and once following six weeks of lower extremity training with the interactive EMG system. The time commitment

for each pre- and post-training test session will be approximately 90 minutes.

1. Kinematic analysis of joint/segmental coordination during gait: A three dimensional motion analysis system (OPTOTRAK, Northern Digital Inc., Waterloo Ontario) will be used to measure joint kinematics at each joint (peak and amplitude velocity) and across joints (cross-correlations). We will also collect the limb movement during the gaitmat trials in procedure 2. The variables that will be measured in this test are hip, knee, and ankle joint angles, as well as hip-knee, hip-ankle and knee-ankle cross correlations within and between legs (Barela et al., 2000; Whitall and Getchell, 1995).
2. Spatial/temporal parameters of functional gait: Subjects will walk along a GAITRite instrumented gait walkway (CIR Systems, Inc., Clifton, NJ) to measure spatial/temporal parameters particularly of velocity and symmetry. This 8 m rubberized walkway is instrumented with pressure sensors to record location and time of foot contacts. Five trials will be at a preferred speed and five at a fast but safe speed. Two speeds are used because one represents activities of daily living and the other represents the ability to adapt the gait according to changing environmental situations.
3. Active and Passive Lower Extremity (LE) Range of Motion (A/PROM): Standard goniometry will be used to determine subject impairment levels of bilateral hip, knee and ankle.

4. Isometric LE strength: A Chatillon Force Dynamometer will be used to measure the strength of key muscles involved in gait. This dynamometer has been validated for use on stroke patients.
5. EMG Analysis: Subjects will perform five trials each of the exercises used to collect the EMG library with the EMG sensors placed over the same 6 muscles. We will analyze the EMG waveforms measured by the sensors and will record such features as frequency, amplitude, and whatever other characteristics of the signal are important for the control of the videogame.
6. Dynamic Gait Index: This quick, useful, reliable and valid instrument evaluates dynamic balance in a number of ambulatory situations in order to assess the likelihood of falling.
7. Pre-Survey: A Lickert scale questionnaire will be administered to determine comfort level with videogames and exercise.
8. Post-Survey: The same Lickert scale questionnaire will also include questions related to acceptability, self-efficacy and tolerability of iEMG.
9. Stroke Impact Scale: This tool measures quality of life including mobility in both the home and community. This will be used to assess behavioral change of the patient in response to training.
10. Analysis: Pre to post data will be analyzed using a one-way ANCOVA with covariates including age, gender, time since stroke and stroke severity.

Randomized Controlled Trial in the Home Setting

Another future aim is to test the iEMG system for training of hemiparetic subjects at home. For a six week training period, participants will be asked to participate in training sessions in the home setting at least three times per week for about one hour each session. They will be encouraged to participate more frequently if they elect to, but no more frequently than one hour of training per day since we do not know the limits for fatigue or repetitive stress injuries. Subjects will be recruited and screened using the same criteria used before, with one addition. Since the interactive EMG videogame involves the execution of exercises in a standing position, the therapist screening the subjects must determine whether the interactive EMG-controlled videogame is safe for the subject to perform without assistance. If it is deemed unsafe, then in order to continue in the study the subject must have an assistant available at home at least three days per week to help with the setup and safe execution of the game.

The home therapy will be preceded by two sessions in the laboratory. Session 1 will be screening, consent and a repeat of the existing procedure in order to personalize the game parameters. Session 2 will be pre-testing as described above in “System Efficacy Testing: Pre/Post-Training Functional Assessments.” Session 3 will comprise a visit by a therapist and a technician to the subject’s home and will consist of installation of the EMG system, training of the subject (and the subject’s assistant as appropriate) in the safe execution of the exercises and use of the system, and the first training session with the interactive EMG-controlled videogame. For session 4, the therapist will again visit the subject’s home and will coach the subject

during the second training session with the game. All remaining sessions will occur at home in a self-directed fashion. The therapist and technician will both be available during regular business hours to answer any questions that arise during the training period. In order to ensure treatment fidelity, the trainer will visit each subject at least once during the six week period (about week 3/4). At the end of the six week period, a final visit will be scheduled to the subject's home to retrieve the equipment. The final session will be in the laboratory and will consist of the post-testing as described above. Participants will also be asked to complete midpoint and final surveys to assess the subject's perceptions of the interactive EMG training. In addition, participants will be invited to attend a focus group in order to provide feedback about their experience with the videogame.

iEMG Training

Subjects randomized to the iEMG group will complete the same training as in the lab setting, except that it will be adapted to the home setting as described above. A description of the training regimen completed by subjects randomized to the control group follows:

Control Group

The control group for the interactive EMG training will consist of subjects that participate in conventional home-based self-training based on the isolated exercise of each of the six target muscles in sitting. The self-training for the "conventional care" group would be three sets of ten repetitions of the following exercises:

1. Active hip flexion of paretic leg in sitting
2. Active knee extension of paretic leg in sitting
3. Active knee flexion (from extended position) of paretic leg in sitting
4. Active ankle dorsiflexion of paretic leg in sitting
5. Theraband exercises for hip abduction in sitting (theraband around thighs, subject pushes knees apart)

Control subjects would be evaluated using the same pre/post tests as for the interactive EMG group. Data from these tests will be stored by the data collection system and returned to the researchers along with the hardware at the end of the training period. Each patient will keep a log of their exercising and the program will also save a diary of exercise execution and performance (how often used, how long, performance records for each session).

Analysis

Pre to post data will be analyzed using paired t-tests for the whole group as well as within individual for the gait mat/joint kinematics/muscle patterns. However, given the small sample size and the potential heterogeneity across trials or subjects, it is probable that the most meaningful results will be reported on an individual basis as absolute and relative changes. A group by time ANCOVA will also compare the training effects of both groups with covariates including age, gender, time since stroke and stroke severity.

Hardware Improvement

There are also several ways in which the system can, and will have to, change in order to better suit the end-users. Ideally, the system should be transformed into a

more portable, take-home system. One of the greatest hardware improvements possible is to remove the need for a physical cable connection between the user and the computer containing the system's software. This can be done by introducing a wireless transmitting device on the central board held in the user's waist pack. This transmitter can send each of the output signals and the system's reference ground voltage through the air to a receiver plugged into the system's data acquisition card in the computer. As the data is received, it can be read and processed by the signal conditioning and feedback programs in the same way as before, since the signals are still being sampled and formatted by the acquisition card.

With the absence of the data acquisition cable, users will no longer be limited to within 10 feet of the system's computer. The cable will no longer be a concern for tripping or pulling, which may potentially injure the user or damage the system. This is particularly prudent considering that the user will be moving their legs extensively while performing the training exercises. The range of the wireless transmitter depends on the specific model used, but can generally be at least 15 feet, a comfortable improvement over the cable's 10 foot length.

Additionally, the system could be constructed with more user-friendly and durable materials to allow the patients to easily put on and remove the system with minimal discomfort. An alternative way to accomplish this is to minimize the size and number of parts by attaching a miniature wireless transmitter to each of the six sensor boards, similar to the aforementioned "Wireless Body Area Network" developed by University of Alabama researchers (Jovanov et al., 2005). This would remove the need for the waist pack and any cable connections at all, theoretically

maximizing user mobility and eliminating the risk of wires catching or pulling out while the user performs exercises. However, doing so would require completely redesigning the sensor network, since each sensor board would have to handle its own power, reference ground, and output transmitting, functions that are presently handled by the waist pack's central board. This option is not feasible if the addition of these capabilities to each sensor board increases the board size to the point where it cannot be easily worn in all six muscle locations.

A significant barrier to the use of our system in the home is the current hassle of placing the electrodes on the patient and wiring all the boards together. While this hassle would be largely reduced by switching to wireless transmission of EMG data, retaining the current six individual sensor boards for placement would still require a patient to remember the correct locations for placement in the home.

To facilitate power for wireless sensors and allow patients to place their own electrodes, a neoprene "sleeve" could be fashioned with embedded sensors that would be placed around the leg to align each sensor in the desired position. The first time the system is used, the sleeve could be custom-fit to each user with the aid of physical therapists. The patient could later ensure the sleeve was oriented properly by using simple body landmarks. For example, a small hole that fits around the kneecap could indicate that the sleeve is in place, as well as the embedded sensors. Power could potentially be integrated as well. This feature would make sensor placement much quicker and more accurate.

The main issue with this method is that a leg-sleeve would only encompass four of our current six muscles/electrode placements – it would likely be cumbersome

or impractical to extend it all the way up to the hip muscles. However, given the strong correlation we have seen in our data between our two hip muscles (flexor and abductor), a future system might be able to work by monitoring only one (likely the TFL). In this case, a leg sleeve would likely encompass the four muscles of the lower and upper leg, and a single wireless sensor might be applied to the TFL.

Portability could be further improved by running the videogame on a handheld computer with LabVIEW. This computer could use an IEEE 802.11 protocol (wireless local area network) to communicate to a central computer displaying and playing the videogame. Networking also allows the system to add new methods of patient-doctor interaction. If connected to the internet, the system could send the physical therapist updates on the patient's frequency of practice and performance scores, allowing the therapist to better shape and direct the patient's physical therapy.

Any changes to the design of the sensor network and overall system should improve its ease of use in the home setting. These changes should also be suitable for mass production in order to facilitate successful commercialization. The system should be relatively low-cost, so that patients can afford it themselves. The current computer system could easily be plugged into a home TV or PC monitor, so users will not have to spend extra money for another one.

Videogame Development

In terms of the videogame, future systems will have multiple games for the user to choose from, possibly corresponding to different exercises required for gait. An increase in game complexity could also be beneficial to reduce the chances of the users becoming bored with the game once they begin to do well. A partnership with a

gaming development company would be optimal, so that videogames could be conceptualized and designed from the beginning with gait rehabilitation patients in mind. For example, the user should be able to train and switch between different exercises without the need for a mouse or keyboard, which may be annoying to use since the patient is already wearing the signal collection system and standing a good distance away from the computer. Other important features that should be incorporated into the game include an improved graphical interface that provides real-time feedback and records performance online via the Internet.

The improved interface must include a graphical representation depicting a target level of muscle activation for the patient for a given exercise, with his or her own EMG activation level superimposed, as well as visual guidance for the correct sequence of activation of separate muscles (Aiello et al., 2005). Providing a more visually appealing virtual environment is also desirable, as it would help enrich the user's experience.

To keep the patient engaged and motivated to continue using the system, certain design changes can be made to the game. The addition of characters or features that can only be used once the participant has attained a certain level of performance or investment in the game (i.e. time played) will reward the player for effort and improvement, providing extra motivation to continue playing. Other additions of this sort include benchmark goals or challenges that change depending on the skill level and displayed records of performance history. By adapting to the user's ability, the game will remain challenging, so that the user does not become complacent and can continue to develop his or her skills. It would also be greatly

beneficial to incorporate a wide array of game modes and activities to prevent the game from becoming monotonous. Although the exercises may remain the same, simply a change in setting or visual feedback, such as a jump or hurdle instead of a golf swing, should keep the game feeling fresh. We currently use only the driving range mode from *Tiger Woods PGA Tour 2004*, which the patient will likely become bored of after extensive use. By expanding the game to include completing actual golf courses or activities other than golf, we will better be able to keep the user engaged even after prolonged use of the system.

The incorporation of online records and improved real-time feedback can also further improve the efficacy of our system. The use of online performance information will give the therapist the option of reviewing a patient's data from a separate location. This feature is particularly convenient if the patient's records are automatically uploaded to the therapist's computer. The current real-time feedback can also be improved by using dynamic scoring of the exercise as it is performed. This improvement provides an opportunity to use simple auditory feedback, such as clapping of varying volume and intensity. In this way, both visual and auditory real-time feedback would instantly and intuitively notify the patient of bad habits or particularly good performances. The game could also be designed to give the user specific hints and suggestions of how to improve his or her performance using the real-time data acquisition. For example, the game could tell the user to try relaxing a certain muscle that is incorrectly activating. The inclusion of this feedback would help improve the utility of the game, as the user will theoretically only have to follow the suggested instructions to improve their gait.

A partnership with a game software developer would also provide a chance to diversify the games offered to patients. This is important for realizing the objective of developing a rehabilitation system that is fun and engaging, and motivates the patients to engage in long term use of the system in order to prevent regression after cessation of regular rehabilitation. After development of software to properly classify EMG signals from various interacting muscles, identifying physiological bad habits by EMG signatures, and scoring particular exercises, the extension of the program to additional gaming platforms should not pose a significant difficulty.

Assessment of Videogame

It is critical that the videogame be interesting enough to engage the patient's attention and motivate them to continue their participation in the game, both in the lab setting and at home. In order to gauge the degree to which the game is enjoyable and compelling, the participants' experience of the game during training will be assessed through surveys and focus groups. The information received can be used to modify the game and/or create additional modules for the game. Guidelines for these surveys can be seen in the IRB Application in Appendix II-F and II-G. Since we were unable to complete training with our complete integrated system as extensively as we originally intended, we were unable to administer these surveys in a relevant way.

Appendices

Appendix I: Procedure for Testing

Topics to go over with the Drs before working with the patients.

-To make sure we are completely on the same page, go over every exercise.

-What position should patients be in for electrode placement? Tib ant, gastroc medialis and vastus medialis should all be able to be done simply with patient sitting. For the biceps femoris, it would probably be helpful for the patient to lie on their stomach (SENIAM recommendation). The tensor fasciae latae and gluteus medius might be a bit trickier, SENIAM recommend lying sidewise on a table.

-For reference (patient-specific) iso contractions, we probably need a backup if subjects (likely) can't lift themselves up onto their toes with their calves. Perhaps subject could extend their leg while sitting and push against a flat surface with their toes?

-Any obvious reference iso for hip abductor gluteus medius?

Potential Issues

-Do we have the little loops of Velcro to hang down from the fanny pack for the gluteus medius and the tensor fasciae latae? If not it might be quite hard to comfortably reach.

Also, can we attach longer electrode wires, or extensions to a board or two for these muscles? This might be more comfortable for the patient so that they don't have to have boards under their shorts.

1) Greet subject. Brief on experiment procedure and movements.

- (a) We're students at the University of Maryland working on modifying a popular golf computer game so that it can be played by performing certain exercises
- (b) The green boards we have here record electrical activity, or EMG data, that indicates when the muscles under them have fired, and how strongly they fired. With a bit of software which we are working on, these signals will tell our golf game when to swing, how hard, and how accurately.
- (c) We have tested these boards and recorded data on group members several times. We really appreciate your participating in this project.

2) Have patient read and sign consent form.

3) One group member will set up laptop.

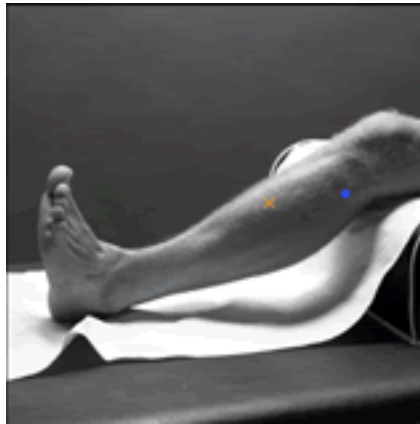
- a) Open MATLAB
- b) Navigate to the directory with the files temp.fig and temp.m (These should also have the necessary c code files in there as well but we will check this before the testing begins)
- c) Type "temp" into MATLAB (w/o quotes) and the program should start
- d) After program starts, change voltage range to 5V
- e) Change total time to a desired number in seconds → If you are unsure with

how long the trial will take, overestimate the amount.

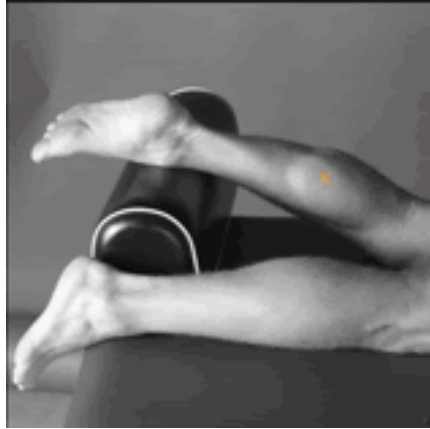
- f) After you have set the total time and changed the voltage range to 5V, wait for the electrodes to be attached, the exercise to be explained and for the patient to be ready.

4) Place each pair of electrodes (6) on the targeted muscles (so that the location of the orange X's falls between the electrodes). Electrode pairs should be placed right next to each other.

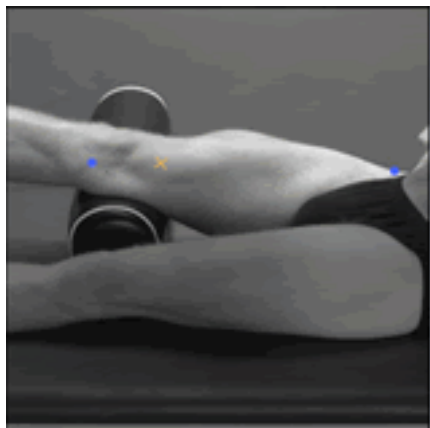
- (a) *Tibialis Anterior*: Place one set of electrodes at 1/3 on the line between the (top) of the fibula and the tip of the medial malleolus (ankle on the inside of the foot).



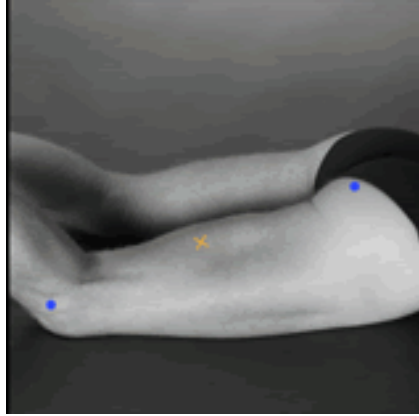
- (b) *Gastrocnemius Medialis*: Place one set of electrodes on the most prominent bulge of the gastrocnemius medialis.



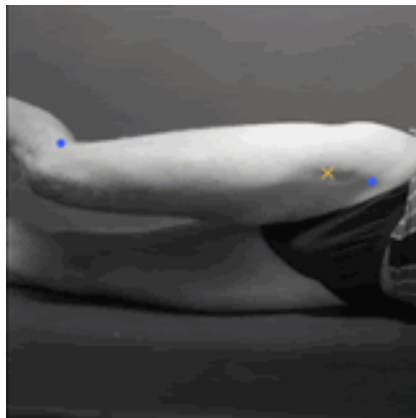
(c) *Vastus Medialis*: Place one set of electrodes at 80 percent on the line between the anterior spina iliaca superior and the joint space in front of the anterior border of the medial ligament.



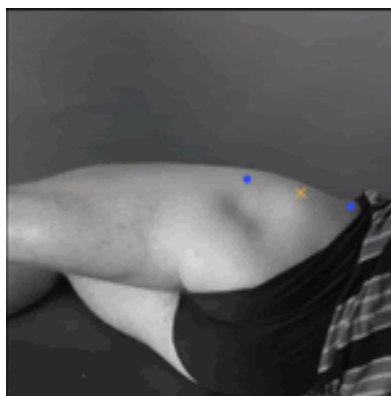
(d) *Biceps Femoris*: Place one set of electrodes at 50 percent on the line between the ischial tuberosity and the lateral epicondyle of the tibia.



(e) *Tensor Fasciae Latae*: Place one set of electrodes on the line from the anterior spina iliaca superior to the lateral femoral condyle in the proximal 1/6.



(f) *Gluteus Medius*: Place one set of electrodes at 50 percent on the line from the crista iliaca to the trochanter.



(g) Place Velcro straps around leg and thigh near electrode pairs.

Depending on the subject, the tibialis anterior and gastroc may be able to share one Velcro strap. There is one strap with Velcro on both sides for this case.

(h) Attach one sensor board to each Velcro strap, attaching the electrodes on the board to the pair of gel electrodes on the muscles.

(i) Have Patient attach fanny pack around waist.

(i) Drop down Velcro loops from fanny pack for the hip muscles?

(j) Attach USB cables from Fanny Pack to USB inputs on boards

5) Connect fanny pack to DAQ

(a) (INSERT DIAGRAM HERE)

ii) Ensure that no wires create any potential hazard to subject.

6) **Begin Recording Data**

a) *For all Data Recordings:*

Hit the “Go Go Gadget Sensing” button

i) The program feedback bar will display “Recording...”

ii) After the total time has been reached, the program will display: “Finished Recording.”

b) **IMPORTANT:** Save the file after you have recorded the data by typing in DATE_SUBJECT_TRIAL_Exercise-Repetitions.mat, substituting the date

for DATE, subject for SUBJECT, etc. and hitting the save button. (Do not use slashes as that will indicate you are trying to save in a different directory)

The program feedback should indicate the file was saved.

- i) If you make a mistake, simply retype what you want the filename to be and hit save once again.
- c) Once you have done this, you can start with a new exercise. You can simply change the time and rehit the sensing button or you can quit out of temp and re run the program. Either option will work fine.
- d) Set up DAQ
- e) **Collect reference maximum comfortable contraction data on iso's for relevant muscles. Do two reps for each exercise and save data.**
 - i) *Tib Ant*: As patient sits with feet flat on the ground, ask patient to raise his or her toes while leaving the heel on the ground.
 - ii) *Vastus Lateralis*: Also while patient is sitting, ask patient to swing knee out as far as possible or comfortable.
 - iii) *Biceps Femoris*: Ask patient to stand up, and offer a surface to hold onto for support and balance. Ask patient to lift their heel towards their gluteus.
 - iv) *Gastrocnemius Medialis*: While patient is still standing and has support available if needed, ask patient to lift themselves onto their toes with their calves.
 - v) *Tensor Fasciae Latae*: While patient is still standing, ask patient to lift knee upwards, flexing the hip. (similar to Exercise 2 but would like patient

to focus exclusively on hip flexion without knee flexion).

f) Explain and Demonstrate exercise #1

g) (1) *In a standing position, patient shifts weight to paretic leg and then attempts to straighten their knee (activation of Gluteus medius and quadriceps).*

h) Instruct Patient to perform exercise #1 4 times. Set GUI to 60 seconds. If patient completes exercises easily within the allotted time, may adjust time down accordingly for future sets.

7) Explain and Demonstrate *exercise #2*

a) (2) *In a standing position, patient shifts weight to nonparetic leg and swings the paretic leg forward, bringing knee up and heel towards gluteus. Flexes hip and knee (activation of tensor fasciae latae and hamstrings).*

b) Instruct Patient to perform exercise #2 4 times. Set GUI to 60 seconds.

8) Explain and Demonstrate *exercise #3*

a) (3) *In a standing position, patient shifts weight to nonparetic leg and focuses on kicking paretic leg out by straightening the knee and pulling the toes upward (i.e. like kicking a ball) (activation of anterior tibialis and quadriceps).*

b) Instruct Patient to perform exercise #3 4 times. Set GUI to 60 seconds.

9) Explain and Demonstrate *exercise #4*

- a) (4) *In a standing position, the patient shifts weight to nonparetic leg steps forward with paretic leg and plants foot on the ground, **heel first,**(?) “controlling” the amount of knee flexion (putting weight on paretic leg without knee buckling) (activation of quadriceps and gastrocnemius).*
- b) Instruct Patient to perform exercise #4 4 times. Set GUI to 75 seconds.

10) Go over reference iso contractions for all muscles again.

11) If there is time and patient is willing, go through each set of exercises once more.

12) Disconnect patient from system.

13) Disconnect fanny pack from DAC.

14) Disconnect USB Cables.

15) Remove Fanny Pack from patient’s waist.

16) Remove elastic straps.

17) Remove Sensor Boards and Electrodes.

18) Thank patient for participation in testing.

Appendix II: IRB Application and Addendum

UNIVERSITY OF MARYLAND, COLLEGE PARK Institutional Review Board Initial Application for Research Involving Human Subjects

Name of Principal Investigator (PI) or Project Faculty Advisor Dr. Pamela Abshire **Tel. No.** 301-405-6629
(*NOT a student or fellow*)

Name of Co-Investigator (Co-PI) Alfred Haas **Tel. No.** 301-405-8974

E-Mail Address of PI pabshire@umd.edu **E-Mail Address of Co-PI** ahaas@umd.edu

Name and address of contact to receive approval documents
Dr. Pamela Abshire
Institute for Systems Research
University of Maryland, College Park
College Park, MD, 20742

Name of Student Investigator Benjamin Tousley **Tel. No.** 240-478-7240

E-Mail Address of Student Investigator btousley@umd.edu

Check here if this is a student master's thesis **or a dissertation research project**
Department or Unit Administering the Project UMCP Gemstone Program

Project Title Interactive EMG Training System for the Rehabilitation of Hemiparetic Individuals

Funding Agency: UMCP Gemstone Program
ORAA Proposal ID Number: N/A
Names of any additional Federal agencies providing funds or other support for this research project: None

Target Population: The study population will include (Check all that apply):

- | | | |
|--|------------------------------------|--|
| <input type="checkbox"/> pregnant women | <input type="checkbox"/> neonates | <input type="checkbox"/> individuals with mental disabilities |
| <input type="checkbox"/> minors/children | <input type="checkbox"/> prisoners | <input checked="" type="checkbox"/> individuals with physical disabilities |
| <input type="checkbox"/> human fetuses | <input type="checkbox"/> students | |

Exempt or Nonexempt (Optional): You may recommend your research for exemption or nonexemption by checking the appropriate box below. For exempt recommendation, list the numbers for the exempt category(s) that apply. Refer to pages 6-7 of this document.

Exempt----List Exemption Category(s) **Or** **Non-Exempt**
If exempt, briefly describe the reason(s) for exemption.

1. Abstract

Our research goal is to create a training regimen that is enjoyable enough to induce improved patient cooperation over conventional rehabilitation regimens. We intend to meet this goal using a simple videogame, where game control is achieved by contracting certain muscles, as opposed to pushing buttons. The muscle contractions will be read by sensors on the skin and classified to produce an appropriate output to control the videogame.

The videogame system will include a gaming console, television screen, and wires that are connected to one or more sensor pairs placed on the skin over different lower leg muscles. Up to six sensors will be held in place on the leg using comfortable surgical tape, to ensure placement during the testing sessions. We have consulted with electrical engineers and clinicians to ensure that direct skin contact with the sensors during testing is safe and that there is no significant risk of shock to the patient. Furthermore, to keep the participants safe, an electrical engineering expert will always be present to ensure proper use of the electrodes and game interface. The testing will take place at the University of Maryland, Baltimore, Department of Physical Therapy and Rehabilitation Science Research Lab, and a physical therapist will be on hand at all times for supervision. Our partners at UMB have helped us to design the training regimen to ensure the maximum safety of the patients.

2. Subject Selection

a. We have secured a partnership with Dr. Jill Whitall and Dr. Sandra McCombe Waller, professors and physical therapist, respectively, in the Physical Therapy and Rehabilitation Science Department in the School of Medicine at the University of Maryland, Baltimore (UMB). They have worked with us already in developing the patients' training regimen, and will supervise the preliminary testing to ensure safe collection of electromyographic data from multiple subjects. Electromyographical (EMG) data are the electrical potentials generated when muscles contract, which can be detected at the skin surface by EMG sensor electrodes. The EMG data, along with videotaped recordings of the muscle movements ("EMG data library"), will be used to train a signal classifier that can identify a particular muscle movement from the correlated EMG signals measured by one or more sensors. After we have trained a working classifier, we would like to test our final game on these same hemiparetic patients.

The subject pool will comprise post-stroke patients that have suffered from hemiparesis or spasticity in leg muscles, in order for the data to be most relevant to

the intended application. Hemiparesis is muscle weakness on one side of the body often induced by a stroke. Muscle spasticity, which can also be induced by stroke, causes involuntary contractions and stiffness that may interfere with functions such as gait.

Participants will come from a preexisting pool of post-stroke therapy patients who have voluntarily consented in writing to be contacted when research studies associated with the Physical Therapy Department are in need of testing participants. These patients, who have participated in similar research before, will receive an e-mail, letter or telephone call inviting them to participate in the research from Dr. Whitall or Dr. Waller. The contents of the e-mail and the letter appear in Appendix A. The text in Appendix A will also serve as the script for telephone recruitment.

b. Yes, subjects will be selected for specific characteristics. The general *inclusion* criteria, as determined by our partners from UMB, are those who (1) have had a unilateral ischemic or hemorrhagic stroke (cortical or subcortical), (2) are aged 40 – 80 years, (3) are at least six months post stroke having completed all conventional inpatient and outpatient therapy, (4) have residual hemiparetic motor dysfunction of the leg, and (5) are able to stand for 5 minutes with arm support.

The general *exclusion* criteria include those subjects who have:

(1) An alcohol consumption > 3 oz liquor, or 3 x 4oz of wine, or 3 x 12 oz of beer per day; or

(2) Cardiac history of (a) unstable angina (a partial blockage of a coronary artery leading to decreased blood flow to heart); (b) recent (< 3 months) myocardial infarction (heart attack, generally full blockage of a coronary artery); (c) congestive heart failure (NYHA category II); or (d) hemodynamically significant valvular dysfunction; or

(3) Medical history of (a) recent hospitalization for severe disease or surgery, (b) poorly controlled hypertension (>160/100) or uncontrolled diabetes (c) significant orthopedic or chronic pain conditions limiting exercise, (d) active tumor(s) on one or both legs, (e) debilitating respiratory or kidney disease, or (f) presence of any serious medical condition with a prognosis for death or dependency in the next 2 months; or

- (4) Neurological history of (a) untreated major post-stroke depression, (b) dementia based on Folstein Mini-Mental Status Score or diagnostic confirmation by neurologist or geropsychiatrist, or (c) severe receptive or global aphasia (loss of ability to communicate with others following brain damage) that confounds testing and training (unable to follow 2 point commands); or
- (5) A non-stroke neuromuscular disorder restricting exercise (e.g., Parkinson's Syndrome); or
- (6) A notable history of noncompliance; or
- (7) Dense flaccid hemiparesis of the lower extremity.

These characteristics will be evaluated by our clinical partners, Dr. Waller and Dr. Whitall, who will examine the subjects' medical records and conduct a brief pre-study screening interview to ensure that all of the inclusion criteria are met and none of the exclusion criteria apply to a subject prior to allowing that individual to participate in the study. In order to do this, Drs. Waller and Whitall must obtain permission from subjects to review their medical records if they previously did not have such permission. In addition to the eligibility criteria listed above, our clinical partners will also exclude a potential subject if they show any signs of recent illness or infection. Hemiparesis will be determined through professional evaluation by Dr. Waller. Dr. Waller and Dr. Whitall will perform the same screening evaluations on all subjects prior to their participation in Stage Two, to confirm that each subject is still eligible.

c. The bases of post-stroke hemiparesis and age are essential for selecting subjects because the EMG data we record to recognize a weak agonist or incorrect antagonist muscle movement will only be applicable if it is obtained from subjects who match our end-user demographic. The nonparetic leg of each participant will provide the most accurate data reflecting the EMG signals that s/he should strive towards using the paretic limb. All of the other inclusion and exclusion criteria are necessary to ensure that no other unforeseen, unrelated, and uncontrollable risks complicate or compromise the safety and comfort of the patients during testing.

d. We hope to obtain initial EMG data from about six test subjects. While we understand this is an ambitious target, we believe that more heterogeneity in the

initial data will enable us to construct a more robust classifier that will be more successful during final testing, and eventually applicable to a wider population of hemiparetic individuals. The first few subjects tested will ideally include subjects from both genders with a fair range of walking ability/stroke severity.

3. Procedures

We intend to conduct Stage One testing in fall 2008, collecting initial EMG data that will be used to determine the thresholds and specific characteristics of our classification algorithm and guide subsequent hardware development. Stage Two testing will commence in the summer of 2009, combining the classifier with the game and evaluating how effective the classifier is at matching movements to corresponding actions in the game. The rate of muscular improvement and gait efficiency in the patients will also be tracked. All testing will occur in the clinic in the Allied Health Building, 100 Penn Street, Baltimore, MD, which is part of the University of Maryland School of Medicine.

Stage One

When the prospective subjects initially come in to the lab, before any testing is actually done, they will have enough time to read and fill out the consent form located in Appendix B. Any subject can discontinue his/her participation in the study at any time, and need only contact the research team, Dr. Whitall, or Dr. Waller to do so. In Stage One testing, we will have the patients perform physical exercises intrinsic to traditional physical therapy. For each subject we test, we will conduct four testing sessions on four different days, for about 30 minutes each. The first of these sessions will involve testing of the patient's healthy leg, to serve as a basis for comparison against the performance of the paretic leg. The following three sessions, which will be done on the patient's paretic (weaker) leg, will help us to determine the level of variation within individuals, and to make sure we encompass the full range. We will request that subjects do not conduct strenuous physical activity or therapy prior to our tests, as fatigue may play a confounding role in the data collected.

“Prior” will be relative to the recovery period for the test subject, but the essence will be that we request they not begin our testing sessions tired.

EMG data from 6 key muscles will be collected from subjects during the execution of 4 discrete exercises, as detailed in Appendix E. The procedure that researchers will adhere to when interacting with patients is found in Appendix C. The Stage One dataset will consist of the amplitude of the electromyographic signal detected by each sensor, versus time. Signal frequency will also be a characteristic of interest. This will take the form of multiple EMG waveforms per patient, per testing session. In addition, in order to accurately map movements to EMG signals, we will videotape testing sessions, and this data will be stored digitally. These data will be used to determine the general characteristics and changes in EMG signals recorded from particular muscles when patients attempt a particular muscle movement.

Contractions generally will be slow and deliberate, at a pace comfortable for the subject. However, every fourth contraction we will ask the subject to move faster, at close to their maximum speed without feeling uncomfortable, risking overexertion, or pulling a muscle. In the event that our eventual patients truly do get involved in the games, we need to ensure that the signal classifier can correctly identify quicker, more forceful contractions. Having these more forceful contractions at regular intervals will also allow us to observe the development of the signal over the course of the training period, as the subject gets more tired. The study of this development is essential to an effective, robust classifier. We will also ask for several (three to five depending on the age and physical condition of the patient) quick contractions in a row at the end, to determine to what extent these contractions accelerate the fatiguing process or the potential change in the signal. Leaving successive forceful contractions until the end of the testing session will avoid fatiguing the participant because they will be able to rest afterwards, since they will be finished with their testing session for that day. The total commitment time for each patient for Stage One testing should not exceed 2 hours with the patient’s cooperation.

Stage Two

Stage Two testing will likely not occur until around summer 2009, and will quantify any differences in effectiveness between our interactive training program with videogame versus conventional rehabilitation. The videogame will be controlled by the same pairs of agonist/antagonist muscles that were tested in Stage One, i.e., gluteus medius, quadriceps, iliopsoas, hamstrings, anterior tibialis, and the gastrocnemius. The videogame will contain unstartling, entertaining material appropriate for the advanced age group of our participants. Music and visual stimulation will be included as part of the videogame, but volume levels will be adjustable to the participant's choosing. A single USB cable from each sensor will connect to the classification and power system, attached near the patient's waist. These cables will be sized to the individual subjects – they will be just long enough to permit freedom of movement, but will not extend below the calf sensors, so that they do not pose a tripping or falling hazard. Similarly, the cable that sends output from this system to control the game will be of sufficient length to allow a healthy distance from the monitor as well as freedom of movement. However, this cable should not constitute a falling or tripping hazard because it will be kept away from the active leg, at waist level, and all other individuals will be kept away from the area between the videogame system and the patient. As an added precaution, all wires attached to the sensors on the patients and the classification system can be easily detached when any pulling force is applied. This will eliminate the possibility of injury to the patient or damage to the videogame interface due to any sudden movements.

We will be able to see if the signals from a patient's agonist muscles on the paretic (post-stroke weakened) limb are strong enough, compared to the signals taken from the healthy limb, or if the signals from the antagonist muscles are too strong (due to spasticity) to achieve the desired movement. The strength of the contraction from each muscle will be represented graphically and/or with tone cues in the videogame, which will be used as feedback for the patient to analyze their own performance. Gains from our program would be measured by evaluating walking performance of the patients before undergoing the training program and at the end of the program. Subjects will receive the testing and analysis procedures listed in Appendix D twice, once prior to and once following 6 weeks of lower extremity

training with the EMG-activated Videogame. The time commitment for each pre- and post-training test session is approximately 90 minutes.

During the 6 week training program, participants will come into the clinic at the Allied Health Building three times a week for approximately 1 hour each to perform the same exercises found in Appendix E, but with the classification system and videogame attached as well. We will test the non-paretic (healthy) leg in the first and final sessions of the training program as a baseline for comparison with the performance of the paretic leg, both before and after training. In each of these sessions, we will test the paretic leg for the first 60 minutes, and will test the non-paretic leg for the final 30 minutes.

Activation of a particular combination of muscles will produce a corresponding action in the videogame. Subjects will be asked to perform each exercise as a different part of the game, for approximately 5 minutes each, with a 5 - 10 minute rest between each part of the game. The procedure is roughly the same as what is shown in Appendix C, with the exception that the patient will be reacting in response to a videogame rather than simply performing actions for data collection. The one hour training sessions that occur three times a week for six weeks will total eighteen hours. With the addition of the preliminary and post-training testing sessions which each last about 90 minutes each, the total time commitment for Stage Two testing and training is 21 hours.

Although it is each study member's personal choice to come and participate on each testing day, we will also give them an option between the videogame and conventional therapy without the EMG system. We will also keep logs of how often each study member played/trained. Each study member's participation will allow us to judge whether the game was entertaining enough to spur impressive patient cooperation. Furthermore, in the event that not much in the way of improvement is made over the course of the program, the logs will allow us to examine the issue of whether the exercises were fundamentally flawed (e.g. not strenuous enough, too far removed from actual walking, etc.) or if in fact the training regimen was never earnestly adhered to. If the latter were the case, we would then examine whether fundamentally our hypothesis that a game could improve patient cooperation was

flawed, or if perhaps our game just wasn't fun enough. In order to help determine this, each participant will also be asked to complete a very brief survey at the post-training testing session, shown in Appendix F. They will also be asked to complete a similar survey halfway through Stage Two, shown in Appendix G.

4. Risks and Benefits

For our study participants, we will choose otherwise healthy subjects free of illness or infection, as the same sensors will be used for every participant. This will be determined based on the subjects' medical histories as evaluated by our clinical partners, and the pre-study screening interview conducted by Dr. Waller and/or Dr. Whitall that they must complete before participating. As an additional precaution the sensors will be sanitized with alcohol swabs between each testing session.

There is some risk of muscle overexertion or overextension associated with any repetitive muscle movement, especially for those trials that are asked to be at maximal contraction speed. The desired hemiparetic state of our subjects may also affect this risk, as their muscles may be unfamiliar with the movements that we will ask of them, after long periods of disuse or misuse. The patients' age and condition may therefore make them more susceptible to fatigue as well, which further increases the risk of accidental muscle overexertion. Also, following chronic stroke, this same muscle disuse and spasticity is shown to lead to a decrease in bone mineral density and cortical thickness, which can increase the risk of bone fracture.¹ However, ultimately the subject's goal is to correctly perform the movements that we will ask of them, since those movements are fundamental to healthy, efficient walking. The subjects run a very similar risk every day that they continue to walk incorrectly. The energy-inefficient, potentially dangerous walking and the clinically-supervised training regimen carry the same risks, however in the training there is an additional end-goal of strengthening those muscles and bones to allow for the development of a healthy and risk-free walking technique.

¹ "Muscle weakness, spasticity and disuse contribute to demineralization and geometric changes in the radius following chronic stroke," Pang, M. Y. C., et al. (2007) *Osteoporosis International*, Vol. 18, No. 9, pp. 1243-1252.

Due to this increased fragility, the patients will be at increased risk of injury not only during game play, but also in every action they partake in. For this reason, Dr. Waller and Dr. Whitall will be the sole persons authorized to handle the patients during testing, as the researchers do not have the appropriate training. They will also supervise the placement of the sensors on the patient. In the event that the sensor is physically compromised and an unanticipated packaging failure occurs and the patient's skin is damp, there may be slight risk of shock. We have taken every reasonable precaution to ensure that the sensor packaging may not be compromised.

The Stage One testing will lead to the development of a classifier that can be used with a videogame to make the physical therapy that subjects must already take part in more entertaining. For Stage Two testing, to the extent that some improvement may be gained from repetitive exercise of certain muscles, we hypothesize several benefits of our proposed program over a traditional exercise regimen prescribed for home practice. First, there is the potential for improved motor function recovery, or retention of previous gains, due to improved patient cooperation. The best physical therapy training program in the world will achieve limited to no success if the patient does not follow it diligently. By combining exercise with videogame play, and making it fun, we hope to circumvent the problem of lack of patient cooperation. We also hypothesize that the improvement with our program relative to others would increase with time, as a patient's enthusiasm for repetitive daily exercise may erode over time.

Also, our training regimen might be able to reduce the necessary duration of later physical therapy sessions. Furthermore, it could potentially reduce the need for, or frequency of, return visits to the therapist after the initial therapy sessions. Return visits would likely be necessary generally to avoid patient regression from gains made in therapy. To the extent that our game might achieve even incremental improvements in either of these aims, it could offer significant savings of both the patient's money, and the time of health care professionals.

There is also an inherent benefit to our program, even if the patient would otherwise be willing to engage in steady exercise such as walking. Post stroke patients with gait problems may develop bad habits, which arise as undesired

compensatory measures to achieve the task (such as stride) without the full aid of certain important muscles. These bad habits can introduce risk of biomechanical injury.² Our game, however, will focus on training and reinforcing those discrete muscular movements that as a whole can be strung together to be immediately applied to healthy ambulation. Thus, our program would give post-stroke individuals a means of strengthening weakened muscles crucial to walking and reminding their bodies how to walk correctly, without developing the detrimental habits that result from a struggling walker's need to balance or support their own weight.

5. Confidentiality

Identifiable information of subjects such as name and contact information must be kept while the testing phases are in progress, so that we can organize data collected by each individual subject. We will need to compare the results gathered within each subject's trials as well as between all of the subjects, which can only be done if we know who each subject is every time they come in for another testing session. We will keep our information and data from the testing in electronic form, most likely in charts, graphs, tables, documents, and videos using various organizational computer applications. All of our data will be associated with an alphanumeric code for each patient. We will separately maintain a file that matches these codes to their corresponding patient information. These labels should prevent anyone uninvolved with the testing procedure from inferring the identity of the testing subjects. This identification key as well as all of our collected data will be kept on password-protected memory drives accessible only by Dr. Whittall, Dr. Waller, members of Gemstone Team CHIP, our mentors, namely, Dr. Pamela Abshire and Alfred Haas, and system administrators from the Office of Information Technology of the Department of Electrical and Computer Engineering at the University of Maryland, College Park. The Office of Information Technology is well accustomed to maintaining sensitive information in a confidential and professional manner. After all final testing sessions are complete, however, there will be no need to refer to our

² "An algorithm to assess stiff-legged gait in traumatic brain injury," Kerrigan, D.C., et al. *J Head Trauma Rehabil*, 1999. **14**(2): p. 136-45.

subjects by any personally identifiable information, so simple numbers or letters will be adequate to organize our data. Any identifiable information will be destroyed when we no longer have need to associate person with data, which will most likely be in the spring of 2010. No personally identifiable information will be disclosed in our final thesis or in any of the data that will be made publicly available as a result of our project.

To determine the effect our program has had on the subjects' walking technique using the measurements described in the procedures section and Appendix D, we will videotape the testing sessions. These tapes will be digital files that are kept long enough to analyze information that we wish to use, and possibly kept until after final testing for reference purposes. Videos may be used as a part of presentations and University of Maryland websites, but only with the written consent of subjects, as provided for in the consent form in Appendix B. Once again, these files will be saved on password-protected lab computers at UMB and the ECE IBIS lab, and be available only to our UMB partners, Team CHIP, and our mentors. Without the patient's expressed written consent to keep them, the videos will be destroyed or erased at the time that the rest of the identifiable information is erased, in or before the spring of 2010. In keeping with University of Maryland records retention policy, a copy of this final IRB application and all of the consent forms signed by study participants will *not* be destroyed, and instead will be kept until spring 2020, 10 years after the official completion of this research, in possession of the UMCP Gemstone program.

6. Information and Consent Forms

We will provide detailed information about the investigation, free of any deception, to the subjects using a consent form that must be signed before the testing sessions begin. An example of the exact wording and structure of the consent form is located in Appendix B. It will be provided in English only, as we do not foresee that any of the subjects will be non-English speakers.

7. Conflict of Interest

We anticipate no conflicts of interest.

8. HIPAA Compliance

Yes, we will require access to HIPAA protected health information, and will comply with all applicable laws and regulations. Specifically, we will be interested in name, age, and length of time since most recent stroke for all of the patients in our study, as well as duration of any previous lower limb physical therapy and all of the other inclusion and exclusion criteria that are detailed in the subject selection section (2.b.) of this application.

9. Research Outside of the United States

Not Applicable.

10. Research Involving Prisoners

Not Applicable.

Appendix A

This is the text of the e-mail and/or letter to be sent to pre-existing pool of post-stroke patients who have previously volunteered to help with research, asking for study participants. Any telephone invitations will also use this text as a script.

Department of
Physical Therapy and
Rehabilitation Science



**University of Maryland
School of Medicine**

100 Penn Street • Baltimore, Maryland 21201-1082 • 410 706 7720, 7721 • 410 706 6387 fax

Dear _____

My name is Sandy McCombe Waller and I am a researcher at the University of Maryland, Baltimore. I am writing to ask you to participate in a research study.

The purpose of this research is to find out if exercising your leg muscles with the use of interactive videogames will help improve leg function and walking ability in people such as you who have had a stroke.

Your participation in this study would include two distinct stages. Stage One includes a total of 4 testing sessions. Testing would include looking at how well you can move your legs and how well you can walk as well as how the muscles in your legs function. These testing sessions will take place on four separate days, over no more than two weeks. Stage Two is planned to occur about six months later, and we would be grateful for your participation in this part as well. Stage Two is comprised of a six week training program that will include completing leg exercises in a standing position either with or without the use of an interactive videogame, three times a week. A 90 minute testing session will occur before and after this six week program. The testing and all training will take place at the Department of Physical Therapy and Rehabilitation Science, 100 Penn Street, Baltimore, MD, 21201.

Overview:

We will determine your eligibility for this study by examining the degree of weakness you have in your weak leg since stroke and by evaluating your medical history. Once we have determined you are eligible for the study and you have signed the consent form we will complete testing to see how well your leg functions before we begin training.

The Stage One procedures involve a simple workout of the leg muscles on your weaker side, which include weight shifting, hip swinging, knee bending, and leg

stretching in such a way that will prepare you for the actions necessary for a healthy gait. You will be asked to perform both slow, simple movements and quick, stronger movements at a speed you feel is safe. You will have several sensors placed over the muscles you are exercising during this procedure. Each testing session will take no longer than 30 minutes with your cooperation.

During Stage Two, which will be approximately six months later, you will be performing the same exercises, but your movements will be used to control a videogame. You may use this videogame as part of a physical therapy program for no more than six weeks. Prior to and after this six week period, you will be asked to participate in a data collection session composed of tests to see how well you can perform certain leg exercises while standing in place, and also some functional tests of your walking ability (walking speed, etc), while we monitor your muscle contractions with our sensors. This testing is necessary so that we may quantify the improvements you have made over the six week training program. The total time commitment for Stage One is approximately 2 hours. The total time commitment for Stage Two is approximately 21 hours over six weeks including all training and testing sessions.

There is little to no risk introduced as a result of the leg function tests, other than temporary muscle fatigue. A physical therapist will be on hand to assist you as needed to ensure safety. Breach of confidentiality is a potential study risk; however, all records will be housed in a locked file or password-protected memory drive to minimize the possibility of confidentiality problems. Only the researchers will have access to personally identifiable information. In the case that you are a student and/or employee of UMB, UMMS or any other UMB affiliate, your academic standing or employment will not be adversely affected by your decision to participate or not participate in this research study.

Your participation is entirely voluntary and you may withdraw from the study or stop the interview at any time. The benefits of participating in this study are that you may have improved movement of and use of your weak leg after participating in this study. However, you may receive no benefit from participating. The tests performed will be free of charge and we will provide the results to you if you so choose. Your participation may help investigators in individualizing stroke rehabilitation. Your decision to participate or not participate in this research study will not affect any current or future care you receive at UMB. UMMS arrangements to and from the test are also available if needed.

If I do not hear from you in two weeks, I will contact you by telephone to answer any questions you may have about the study. If you have any questions at any time, please feel free to contact me at 410-706-0787.

Sincerely

Sandy McCombe Waller PhD, PT, NCS

Appendix B - CONSENT FORM

Project Title	Interactive EMG Training System for the Rehabilitation of Hemiparetic Individuals
<p>Why is this research being done?</p> <p>This is a research project being conducted by the undergraduate Gemstone team CHIP at the University of Maryland, College Park. We are inviting you to participate in this research project because you have shown evidence of muscular hemiparesis (muscle weakness on one side of the body) and/or spasticity (stiffness and/or involuntary muscle contractions, loss of muscle control) in one of your legs. We would like to study the electrical signals generated by your muscles as you perform certain leg exercises. The data we collect when you perform these exercises will help us to develop a computer program that can determine what leg movement a person is making (for example, kicking) just by measuring their muscle contractions. If we are successful, this program will allow you to control a videogame just by moving certain leg muscles, instead of pushing buttons. We plan to use the program to enable a person to play a fun videogame, like golf, just by performing normal rehabilitation exercises. We are trying to determine if playing such a "rehab videogame" helps with the effectiveness of therapy and with the retention of proper walking technique versus traditional rehabilitation.</p>	
<p>What will I be asked to do?</p> <p>The Stage One procedures involve a simple workout of the leg muscles on your paretic (weaker) side, which include weight shifting, hip swinging, knee bending, and leg stretching in such a way that will prepare you for the actions necessary for a healthy gait. You will be asked to perform both slow, simple movements and quick, stronger movements at a speed you feel is safe. You will have several sensors placed over the muscles you are exercising during this procedure. In preparation for sensor placement, parts of your leg may be shaved and the area cleaned with an alcohol swab. A pack will be placed around your waist that weighs about 1 pound and contains the necessary connections for the sensors. There will be four testing sessions, each on different days, spreading over no more than two weeks, which will all occur at the UMMC rehabilitation clinic in the Allied Health Building. The first session will focus on your healthier leg, and the last three sessions will focus on your weaker leg. Each testing session will take no longer than 30 minutes with your cooperation. The total time commitment for Stage One will be 2 hours.</p> <p>During Stage Two, which will occur approximately six months after Stage One, you will be performing the same exercises with your weaker leg, but your movements will be used to control a videogame. You may use this videogame as part of a physical therapy program for no more than 3 days a week for six weeks. You may also choose to forgo the videogame physical therapy program for a conventional physical therapy session on any given therapy day. These training sessions will be approximately an hour long. Prior to and after this six week period, you will be asked to participate in a 90 minute data collection session that will evaluate your muscle strength and walking ability. This is necessary so that we may quantify the improvements you have made over the six week training program. On these 90 minute days, you will be asked to play the game using your healthy leg as well. Also, during these 90 minute days, the testing will include the attachment of IR emitting diodes (small infrared lights, like the ones in a television remote control) along various locations on your leg and torso. You will be asked to walk on a treadmill that will analyze your gait both with and without any assistive walking device you may normally use. In addition, halfway through Stage Two and again at the end of your participation in the study, we will ask you to respond to a very brief survey so that we may determine the effectiveness of our training program. The total time commitment for Stage Two will be 21 hours including all training and testing.</p> <p>As far as attire is concerned, shorts should be worn or brought to all testing and training sessions. During all training and testing sessions, Dr. Waller and/or Dr. Whitall will be on hand and will be the</p>	

only individuals to place sensors on your body. They will also supervise any exercises you are asked to perform.

Do I have to be in this research? May I stop participating at any time?

Your participation in this research is completely voluntary. You may choose not to take part at all. If you decide to participate in this research, you may stop participating at any time. If you decide not to participate in this study or if you stop participating at any time, you will not be penalized or lose any benefits to which you otherwise qualify. All data collected as a result of your participation may still be used for research purposes, and all identifiable information will be destroyed at the conclusion of this research project.

Project Title	Interactive EMG Training System for the Rehabilitation of Hemiparetic Individuals
<p>What about confidentiality?</p> <p>We will do our best to keep your personal information confidential. To help protect your confidentiality, all data collected, including any videos, sensor data, analysis, or observations will be kept digitally in password-protected computer files. Also, (1) your name will not be included on any collected data; (2) a code will be placed on all collected data; (3) through the use of an identification key, the researchers will be able to link your data files to your identity; and (4) only the researchers will have access to the identification key. If we write a report or article about this research project, your identity will be protected to the maximum extent possible. Your information may be shared with representatives of the University of Maryland, College Park or governmental authorities if you or someone else is in danger or if we are required to do so by law.</p> <p>This research project involves making digital videos of you, so that we may analyze your walking gait and technique and determine if our training program is an improvement over conventional physical therapy. These videos may be used in presentations or placed on University of Maryland websites, but only with your approval. The researchers alone will have access to these videos and any copies that may be made, which will be stored digitally on password-protected computer files. Unless you ask in writing otherwise, all files and any copies that pertain to your participation in this study will be erased at the end of all analysis, by the end of spring 2010.</p> <p><input type="checkbox"/> I agree to be videotaped during my participation in this study and allow these videos to be used in presentations or placed on University of Maryland websites.</p> <p><input type="checkbox"/> I agree to be videotaped during my participation in this study but DO NOT allow these videos to be used in presentations or placed on University of Maryland websites.</p> <p><input type="checkbox"/> I do not agree to be videotaped during my participation in this study.</p>	
<p>What are the risks of this research?</p> <p>Temporary muscle fatigue in the lower leg is expected, due to the repetitive movements you will be asked to perform. There may also be some risk of muscle overexertion or overextension associated with any repetitive muscle movement, especially for those trials that are asked to be at maximal contraction speed, but the level of force that you demonstrate is up to your discretion. As a hemiparetic individual, you are also at increased risk of bone fracture. In the event that the sensor is compromised and an unanticipated packaging failure occurs, there may be a risk of shock. Irritation may occur from the surgical tape used to secure the sensors to your body. A clinician will be on hand to assist you as needed to ensure safety. For Stage Two, the videogame system that you will be using is experimental and its effectiveness has not been established. Progress made in previous physical therapy sessions may be lost or expected progress may not be made.</p>	
<p>What are the benefits of this research?</p> <p>The benefits to you include participation in an alternate training program similar to traditional physical therapy for hemiparetic patients, but with the addition of a videogame as feedback, which aims to be simple enough to be conducted at home and enjoyable enough to make physical therapy more appealing. We hope that, in the future, other hemiparetic patients might benefit from this study in the same way, through utilization of a videogame designed to make physical therapy more appealing and effective, and easier to implement.</p>	
<p>Is any medical treatment available if I am injured?</p> <p>The University of Maryland does not provide any medical, hospitalization or other insurance for participants in this research study, nor will the University of Maryland provide any medical treatment or compensation for any injury sustained as a result of participation in this research study, except as required by law.</p>	

Page 3 of 3 Initials _____ Date _____

Project Title	Interactive EMG Training System for the Rehabilitation of Hemiparetic Individuals
What if I have questions?	
<p>This research is being conducted under the supervision of Dr. Pamela Abshire of the University of Maryland Electrical and Computer Engineering department at the University of Maryland, College Park. If you have any questions about the research study itself, please contact Pamela Abshire at: 2211 A.V. Williams, University of Maryland, College Park, MD, 20742; tele: (301)405-6629.</p> <p>If you have questions about your rights as a research subject or wish to report a research-related injury, please contact: Institutional Review Board Office, University of Maryland, College Park, Maryland, 20742; (e-mail) irb@deans.umd.edu; (telephone) 301-405-0678. This research has been reviewed according to the University of Maryland, College Park IRB procedures for research involving human subjects.</p>	
Statement of Age of Subject and Consent	
Your signature indicates that: <ul style="list-style-type: none"> -you are at least 18 years of age; -the research has been explained to you; -your questions have been fully answered; and -you freely and voluntarily choose to participate in this research project. 	
NAME OF SUBJECT	
SIGNATURE OF SUBJECT	
DATE	

Appendix C

Potential Script for Stage One Testing

At this time we only anticipate having the equipment necessary to test on one subject at a time.

1. Greet the patient: “Hi, nice to meet you, thank you for expressing interest in our research.”
2. “We are going to ask you to do some simple leg movements while wearing six electromyographic sensors over various muscles in your leg. These movements will use both your paretic (weaker) and non-paretic (healthy) legs, and are used in normal walking. Before we begin, we’d like you to read over this consent form to get more details about what we will be asking you to do. We will answer any questions you may have before you sign the form. If you read it and decide you don’t want to participate, you don’t have to sign the form or participate.”
3. When the consent form is signed, attach each of the six EMG sensors over each muscle that we wish to measure data from. All channels will be checked for adequate EMG recording, which will require shaving any excessive leg hair in the sensor area, and scrubbing the skin surface with an alcohol cleansing pad to reduce skin resistance and EMG noise. Since exact placement is very important to reading clear and consistent signals, the physical therapist will supervise. Then attach the central system around the patient’s waist (it will be in a fanny-pack-type holder with the batteries and output connectors – together it weighs about 1 lb.). Connect the cables between the sensor boards and the central system, and then connect the output from the central system to the data acquisition device plugged into a laptop, using a long cable.
4. Patients will be asked to complete the exercises found in Appendix E, each for 5 minutes duration. A 10 minute rest will be given in between each exercise.
5. At the end of testing for that patient, save all data, disconnect all connectors, remove sensors as carefully as possible from patient leg, thank patient for their time and help, and see them away safely.

Appendix D

Pre-training and Post-training Testing Procedures / Script

During Stage Two testing, patient performance will be evaluated using the procedures described below, once prior to and once following 6 weeks of lower extremity training with the EMG-activated Videogame. The time commitment for each pre- and post-training test session is approximately 90 minutes.

1. Kinematic analysis of joint/segmental coordination during gait: Before undertaking the following trials, subjects will first be seated while infra-red emitting diodes are attached to the sacrum and the iliac crest, greater trochanter, lateral condyle of the tibia, lateral malleolus and head of the 5th metatarsal on the shoe of each leg. Subjects will be requested to wear shorts. These sensors will be used to detect the coordination of the limb segments both within and between legs using a three dimensional motion analysis system collecting at 100 Hz (OPTOTRAK, Northern Digital Inc., Waterloo Ontario). Custom-designed software (Co-I Morton) will be used to analyze limb kinematics. Therefore, during the trials on the gaitmat in procedure 2, we will also collect the movement of the limbs in order to detect whether changes in speed and symmetry are also reflected in intra- and interlimb coordination changes at the joint/segmental levels. The variables that will be measured in this test are hip, knee, and ankle joint angles, as well as hip-knee, hip-ankle and knee-ankle cross correlations within and between legs.^{3,4}

2. Functional Gait: Temporal-spatial parameters

Subjects will be asked to walk along a GAITRite instrumented gait walkway (CIR Systems, Inc., Clifton, NJ). This is a portable 8-meter rubberized walkway that has pressure sensors embedded throughout the device, centered at 0.5” increments to give full coverage of the active area. The effective sampling rate is 33 Hz. Analysis

³ Barela JA, Whittall J, Black PO, Clark JE. Can Intralimb Coordination in Hemiplegic Gait be Explained by Speed and Mechanical Constraints. *Human Movement Science*. 2000;19:251-273.

⁴ Whittall J, Getchell N. From walking to running: using dynamical systems approach on the development of locomotor skills. *Child Development*. 1995;67:1541-1553.

software identifies both location and time of sensor activations due to foot contacts, providing spatial-temporal data for each gait cycle that is generated immediately following each trial. Subjects who habitually use an assistive device will be requested to undergo half the trials without their assistive device (two out of four trials for each condition). This will enable us to control for the effect of the assistive device in pre-post comparisons but at the same time determine exactly how each subject ends up since we expect that some of our subjects may not need their assistive device after the 6 week training period. All subjects will wear a gait belt and be closely attended by a research assistant to ensure safety on all trials. The subject will be asked to rest as needed after every second trial. The trials will be as follows:

1. Four trials-comfortable speed. “Walk at a comfortable speed until we say stop.”
2. Four trials-fast, safe speed. “Walk as fast as you can but safely until we say stop.”

We assess two speeds because one represents everyday activity of daily living and the other represents the ability to adapt the gait according to changing environmental situations (e.g., crossing a road). It demonstrates the “best” performance for this movement skill and the existing stride length/time ratio changes across speed. The specific variables that will be measured include gait speed, which quantifies time taken to walk across a 24-foot walkway, and step-time and step-length symmetry ratios, which quantify the temporal and spatial relationships between each leg stride (paretic/non-paretic). For symmetry the ratio is 1.

3. Active and Passive Lower Extremity (LE) Range of Motion (A/PROM): Use of standard goniometry to determine subject impairment levels of bilateral hip, knee and ankle; demonstrated to be reliable and sensitive (within 5°).⁵

⁵ Norkin C, White D. Validity and Reliability. *Measurement of Joint Motion A Guide to Goniometry Edition 2.* 1995:36-37.

4. Isometric LE strength: Use of Chatillon Force Dynamometer to measure the strength of key muscles involved in gait. This dynamometer has been validated for use on stroke patients.⁶

5. EMG Analysis: Subjects will also perform five trials each of the exercises laid out in the Stage One testing procedures with the EMG sensors placed over the same 6 muscles. Similar to Stage One testing, we will analyze the EMG waveforms measured by the sensors, and will record such features as frequency, amplitude, and whatever other characteristics of the signal are important for the control of the videogame.

⁶ Bohannon R, Smith M. Assessment of Strength Deficits in Eight Paretic Upper Extremity Muscle Groups of Stroke Patients with Hemiplegia. *Physical Therapy*. 1987a;67:522-525.

Appendix E

These are the exercises that each patient will be asked to perform for Stage One testing and the Stage Two training program, as well as for the EMG analysis that is part of the Stage Two post-and pre-training testing procedures. There are four different exercises, which are designed to encompass the full range of muscle movements and muscles used in a healthy gait. Subjects will have the EMG sensors placed on the gluteus medius, quadriceps, iliopsoas, hamstrings, anterior tibialis, and the gastrocnemius. All exercises are to be completed in a standing position with upper extremity support for balance. The patient will perform each exercise one at a time, in the following order of complexity:

- (1) In a standing position, patient shifts weight to paretic leg and attempts to then straighten their knee (activation of Gluteus medius and quadriceps).
- (2) In a standing position, patient shifts weight to nonparetic leg and swings the paretic leg forward, bending at the hip and knee (activation of iliopsoas and hamstrings).
- (3) In a standing position, patient shifts weight to nonparetic leg and focuses on kicking paretic leg out by straightening the knee and pulling the toes upward (i.e. like kicking a ball) (activation of anterior tibialis and quadriceps).
- (4) In a standing position, the patient shifts weight to nonparetic leg steps forward with paretic leg and plants foot on the ground “controlling” the amount of knee flexion (putting weight on paretic leg without knee buckling) (activation of quadriceps and gastrocnemius).

Appendix F

Survey to be completed by each participant at the end of the Stage Two six week training program

Post-Training Survey

IDENTIFICATION NUMBER: _____ (for use by researchers only)

Respond to each of the following statements in terms of your own personal opinion or experience while participating in our study. All questions are in reference to the Stage Two six-week training program that involved training three times a week. Put a mark on the line that appears in front of the option you choose.

1. How would you describe the overall effect that the training program has had on your walking ability?

- _____ It helped me improve a lot.
- _____ It helped me improve a little.
- _____ I improved a little, but I don't think the training program had an effect.
- _____ I did not improve at all.
- _____ The training program was detrimental to my walking ability. I would have been better off with other/no rehabilitation procedures.

2. How would you describe the degree to which your interaction with the videogame is responsible for the effect of the training program (as determined in question 1)?

- _____ The inclusion of the videogame made all of the difference.
- _____ The inclusion of the videogame made just enough of an impact to make it unlike any other rehabilitation that does not include a videogame.
- _____ In terms of walking rehabilitation, this program would have had the same effect with or without the videogame. The exercises and other factors included in the training program were the only things that mattered in terms of the amount of recovery I experienced.

3. How often did you dedicate time solely to physical therapy exercises for your leg muscles during the six-week training period (aside from your participation in this research study)? Choose the option closest to your answer.

- _____ I never did any other lower-limb physical therapy during the six-week period.
- _____ Less than once a week.
- _____ About once a week.
- _____ About twice a week.
- _____ About three times a week.
- _____ More than three times a week.

4. How do the exercises that you were asked to perform for the training program compare to other *lower-limb* physical therapy you have done before?

- _____ The exercises were more strenuous than anything else I have done before.

_____ The exercises were just as strenuous as some of the physical therapy I have done.

_____ The exercises were less strenuous than some of the physical therapy I have done.

_____ The exercises were less strenuous than most or all of the physical therapy I have done.

5. Regardless of the effect that the training program has had on your walking ability, how would you describe your motivation as a result of the videogame in your training?

_____ The videogame made physical therapy fun and engaging.

_____ The videogame just barely kept my interest throughout the training program.

_____ The game did not affect my motivation throughout the training program.

_____ I was more motivated to participate in physical therapy without the game. I did not like some part of the game.

_____ I was more motivated to participate in physical therapy without the game. Wearing the sensors or the control system was too much of a hassle.

6.a. Do you plan to continue with some form of physical therapy in the future?

_____ Yes.

_____ No.

_____ I am unsure.

b. If so, would you be interested in further physical therapy involving a videogame, much as you have experienced in this study?

_____ Yes.

_____ No.

_____ Neutral.

Thank you very much for your generous participation in Team CHIP's research study! We could not have done it without you!

Please return this survey to a researcher.

Appendix G

Survey to be completed by each participant at the midway point of the Stage Two six week training program

Mid-Training Survey

IDENTIFICATION NUMBER: _____ (for use by researchers only)

Respond to each of the following statements in terms of your own personal opinion or experience while participating in our study. All questions are in reference to the Stage Two training you are currently receiving. Put a mark on the line that appears in front of the option you choose.

1. How would you describe the overall effect that the training program has had on your walking ability?

- _____ It helped me improve a lot.
- _____ It helped me improve a little.
- _____ I improved a little, but I don't think the training program had an effect.
- _____ I did not improve at all.
- _____ The training program was detrimental to my walking ability. I would have been better off with other/no rehabilitation procedures.

2. How often did you dedicate time solely to physical therapy exercises for your leg muscles during the three-week training period (aside from your participation in this research study)? Choose the option closest to your answer.

- _____ I never did any other lower-limb physical therapy during the three-week period.
- _____ Less than once a week.
- _____ About once a week.
- _____ About twice a week.
- _____ About three times a week.
- _____ More than three times a week.

3. How do the exercises that you were asked to perform for the training program compare to other *lower-limb* physical therapy you have done before?

- _____ The exercises were more strenuous than anything else I have done before.
- _____ The exercises were just as strenuous as some of the physical therapy I have done.
- _____ The exercises were less strenuous than some of the physical therapy I have done.
- _____ The exercises were less strenuous than most or all of the physical therapy I have done.

UNIVERSITY OF MARYLAND COLLEGE PARK
 Institutional Review Board
Addendum Application

Protocol Number	08-0421
Protocol Title	Interactive EMG Training System for the Rehabilitation of Hemiparetic Individuals
Risk Classification (<i>check one</i>)	<input type="checkbox"/> Greater than Minimal Risk <input checked="" type="checkbox"/> Minimal Risk

Principal Investigator	Dr. Pamela Abshire	Email Address	pabshire@umd.edu
Address for Approval Letter	Dr. Pamela Abshire Institute for Systems Research Univ. of Maryland, College Park College Park, MD, 20742	Telephone Number	301-405-6629

Student/Co-Investigators	Benjamin Tousley	Email Address	btousley@umd.edu
Telephone Number	240-478-7240		

To ensure an accurate and streamlined review of your Addendum Application, please provide the following information:

- 1) State what is being proposed and where in the protocol and/or consent what was changed.

Our protocol for Phase II testing was described in the original IRB application; very little has changed from that description, but we can now be more specific on some details that were left out of the original. The protocol will follow closely the procedures established in Phase I testing, and will analyze the behaviors of the same muscles.

The patient will wear the same data collection board and sensors as in Phase I, which will be attached to a computer as before. For Phase II the patient will be able to see what is on the computer monitor, which will display the game window for the driving range minigame for the computer game *Tiger Woods PGA Tour 2004* as well as a LabVIEW readout. As the patient performs one of the four exercises the LabVIEW window will display a real-time bar graph of the EMG signal amplitude from each of the six muscles being tested. This display will allow the patients to see how their different muscles are working.

After each exercise is completed, the classification algorithm will analyze the EMG data collected and score the quality of the exercise based on how closely the muscle activations matched the goal for that particular exercise. Based on this qualitative assessment the classifier will output a file that will act as an input to the game. Tiger Woods will swing his club and the quality of the hit will correspond to the quality of the exercise; a properly performed exercise will result in a good hit, and an improper one in a shank. There will be some gradation on the quality of hit based on the final resolution we are able to achieve with the classifier; either 5 different levels, or 10, etc.

The patient will perform four repetitions of each of the four exercises, with the ability to adjust between repetitions based on both the real-time biofeedback and the qualitative score of the golf game. The EMG data will be collected just as in Stage One, and the exercises will again be video recorded and stored digitally. This data will allow us to iterate the design of our hardware and classifier and to analyze its success.

2) Provide the rationale/justification for the change.

When the original IRB application was submitted, we had developed hardware and detailed experimental protocols for Phase I testing but were still developing the hardware and protocol for Phase II testing. It has always been our intention to submit an addendum to describe any changes in protocol for the Phase II testing, in order to cover any details that arose as the system for Phase II testing was developed. Since the initial application and its renewal we have decided which game to use (*Tiger Woods PGA Tour 2004*, and specifically its driving range minigame) and how we will be using it. We have also decided to provide real-time biofeedback to the patients as they perform the exercises, in the form of the LabVIEW readout.

3) State what impact the change has on risks to participants. Please state the number of currently enrolled participants and if the change in risk will require re-consent. If the changes will not require re-consent, please state why.

The changes described above will have minimal impact on patients. Phase I testing has been carried out on 8 subjects, 4 hemiparetic patients and 4 age-matched controls. Phase II testing will likely try to recruit the same pool of subjects. There is no change in risk, and the minor changes described above simply add more detail and will not require re-consent. The consent form will remain the same.

4) State whether the change has an impact on the scientific integrity of the study, (i.e. decreases, increases, no impact).

These changes have no impact on the scientific integrity of the study, and merely reflect the anticipated changes in the study as the tools have been developed and more detail has been available on how those tools will be used.

5) List the documents included with the application that have been modified (consent forms, flyers, data collection forms, surveys). State what has been changed in each modified document.

The modified testing procedure is reflected in the attached Appendix H, the potential script for Stage Two testing.

Appendix H

Potential Script for Stage Two Testing

At this time we only anticipate having the equipment necessary to test on one subject at a time.

1. Greet the patient: “Hi, nice to meet you, thank you for expressing interest in our research.”
2. “We are going to ask you to do some simple leg movements while wearing six electromyographic sensors over various muscles in your leg. These movements will use both your paretic (weaker) and non-paretic (healthy) legs, and are used in normal walking. Before we begin, we’d like you to read over this consent form to get more details about what we will be asking you to do. We will answer any questions you may have before you sign the form. If you read it and decide you don’t want to participate, you don’t have to sign the form or participate.”
3. When the consent form is signed, attach each of the six EMG sensors over each

muscle that we wish to measure data from. All channels will be checked for adequate EMG recording, which will require shaving any excessive leg hair in the sensor area, and scrubbing the skin surface with an alcohol cleansing pad to reduce skin resistance and EMG noise. Since exact placement is very important to reading clear and consistent signals, the physical therapist will supervise. Then attach the central system around the patient's waist (it will be in a fanny-pack-type holder with the batteries and output connectors – together it weighs about 1 lb.). Connect the cables between the sensor boards and the central system, and then connect the output from the central system to the data acquisition device plugged into a laptop, using a long cable.

4. Patients will be asked to complete the exercises found in Appendix E, each for 5 minutes duration. As the patients perform each repetition, they will see real-time feedback in the form of LabVIEW-generated sliders representing muscle activation. After each repetition, they will see the in-videogame Tiger Woods make a swing on the driving range, with the quality of the exercise corresponding to the quality of the swing. A 10 minute rest will be given in between each exercise.

5. At the end of testing for that patient, save all data, disconnect all connectors, remove sensors as carefully as possible from patient leg, thank patient for their time and help, and see them away safely.

Appendix III: MATLAB Data Acquisition Code (Final Version)

```
function varargout = temp(varargin)
% TEMP M-file for temp.fig
%     TEMP, by itself, creates a new TEMP or raises the existing
%     singleton*.
%
%     H = TEMP returns the handle to a new TEMP or the handle to
%     the existing singleton*.
%
%     TEMP('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in TEMP.M with the given input
arguments.
%
%     TEMP('Property','Value',...) creates a new TEMP or raises the
existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before temp_OpeningFunction gets called. An
%     unrecognized property name or invalid value makes property
%     application
%     stop. All inputs are passed to temp_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help temp

% Last Modified by GUIDE v2.5 21-Apr-2009 16:29:56

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @temp_OpeningFcn, ...
                  'gui_OutputFcn',  @temp_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before temp is made visible.
function temp_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```



```

% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to temp (see VARARGIN)

% Choose default command line output for temp
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% This sets up the initial plot - only do when we are invisible
% so window can get raised using temp.
if strcmp(get(hObject, 'Visible'), 'off')
    plot(rand(5));
end

% UIWAIT makes temp wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = temp_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data;
axes(handles.axes1);
cla;

data_to_plot = [];
legend_string = '';
for i=1:6;
    if( eval(['get(handles.checkbox_channel' num2str(i) ', 'Value')
== 1']) )
        data_to_plot = [data_to_plot; data.voltage(i,:)];
        legend_string = [legend_string; 'Channel ' num2str(i)];
    end
end

if(get(handles.radio_raw, 'Value') == 0 && length(data_to_plot) > 0 )
    y = data_to_plot';
    data_to_plot = detrend(y);
    if( get(handles.radio_detrended, 'Value') == 1)
        data to plot = data to plot';
    end
end

```

```

else
    [b,a]=butter(5,10/data.SS,'low');
    data_to_plot = filtfilt(b,a,abs(data_to_plot));
    if( get(handles.radio_filtered,'Value') == 0 )
        %define 101 steps between min and max
        numsteps = 51;
        v_min = min(data_to_plot);
        v_max = max(data_to_plot);
        histogram_bins = v_min:(v_max-v_min)/(numsteps-1):v_max;
        histogram_values = zeros(numsteps,1);
        for i=1:length(data_to_plot)
            histogram_values_index = floor((data_to_plot(i)-
v_min)/(v_max-v_min)*(numsteps-1))+1;
            histogram_values( histogram_values_index ) =
histogram_values( histogram_values_index ) + 1;
        end

        for i=51:-1:2
            histogram_values( i ) = histogram_values(i)-
histogram_values(i-1);
        end
        histogram_values(1) = histogram_values(2);
    end
end
end

if( get(handles.radio_histogram, 'Value') == 1 )
    time_axis = histogram_bins;
    data_to_plot = histogram_values;
else
    if( get(handles.checkbox_timestamp, 'Value') == 1 )
        data_to_plot = [data_to_plot; data.timestamp];
        legend_string = [legend_string; 'Timestamp'];
    end
    time_axis = 0:length(data_to_plot)-1;%may want to use size()
function for conditions where number of channels exceeds number of
samples.... when would you want to do this?
    time_axis = time_axis/data.SS;
end

plot( time_axis, data_to_plot' );
legend( handles.axes1 , legend_string );

% popup_sel_index = get(handles.popupmenu1, 'Value');
%
% if(get(handles.radio_raw,'Value') == 1)
%     switch popup_sel_index
%         case 1
%             plot(data.voltage(1,:))
%         case 2
%             plot(data.voltage(2,:))
%         case 3
%             plot(data.voltage(3,:))
%         case 4
%             plot(data.voltage(4,:))
%         case 5

```

```

%         plot(data.voltage(5,:))
%     case 6
%         plot(data.voltage(6,:))
%     case 7
%         plot(data.timestamp)
%     end
% else
%     switch popup_sel_index
%     case 1
%         y = data.voltage(1,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 2
%         y = data.voltage(2,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 3
%         y = data.voltage(3,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 4
%         y = data.voltage(4,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 5
%         y = data.voltage(5,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 6
%         y = data.voltage(6,:);
%         y = abs(detrend(y));
%         [b,a]=butter(5,10/data.SS,'low');
%         filter_y = filtfilt(b,a,y);
%         plot(filter_y)
%     case 7
%         plot(data.timestamp)
%     end
% end

% -----
function FileMenu_Callback(hObject, eventdata, handles)
% hObject    handle to FileMenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)

```

```

% -----
function OpenMenuItem_Callback(hObject, eventdata, handles)
% hObject    handle to OpenMenuItem (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)
file = uigetfile('*.fig');
if ~isequal(file, 0)
    open(file);
end

% -----
function PrintMenuItem_Callback(hObject, eventdata, handles)
% hObject    handle to PrintMenuItem (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)
prindlg(handles.figure1)

% -----
function CloseMenuItem_Callback(hObject, eventdata, handles)
% hObject    handle to CloseMenuItem (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)
selection = questdlg(['Close ' get(handles.figure1, 'Name') '?'], ...
                    ['Close ' get(handles.figure1, 'Name') '...'], ...
                    'Yes', 'No', 'Yes');
if strcmp(selection, 'No')
    return;
end

delete(handles.figure1)

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor')
);
end

set(hObject, 'String', {'Channel 1', 'Channel 2', 'Channel 3',
'Channel 4', 'Channel 5', 'Channel 6', 'Time Stamp'});

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit1 as text
%          str2double(get(hObject,'String')) returns contents of edit1
as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit2 as text
%          str2double(get(hObject,'String')) returns contents of edit2
as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit3 as text
%          str2double(get(hObject,'String')) returns contents of edit3
as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit4 as text
%          str2double(get(hObject,'String')) returns contents of edit4
as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5
as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function radio_detrended_Callback(hObject, eventdata, handles)
% hObject    handle to radio_detrended (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)

% Hint: get(hObject,'Value') returns toggle state of radio_detrended

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user eventdata (see
GUIeventdata)

global data;

data.CHlo = str2double(get(handles.edit1,'String'))-1;
data.CHhi = str2double(get(handles.edit2,'String'))-1;
data.SS = str2double(get(handles.edit4,'String'));
data.TT = str2double(get(handles.edit5,'String'));

```

```

if ((data.CHlo<0)|| (data.CHlo>5)|| ((floor(data.CHlo)-data.CHlo)~=0))
    set(handles.text_feedback, 'String', 'Bad channel low')
elseif ((data.CHhi<data.CHlo)|| (data.CHhi>5)|| ((floor(data.CHhi)-
data.CHhi)~=0))
    set(handles.text_feedback, 'String', 'Bad channel high')
elseif ((data.SS<1)|| (data.SS>200000)|| ((floor(data.SS)-data.SS)~=0))
    set(handles.text_feedback, 'String', 'Bad sampling rate')
elseif ((data.TT<=0)|| ((floor(data.TT*data.SS)-(data.TT*data.SS))~=0))
    set(handles.text_feedback, 'String', 'Bad total time')
else
    set(handles.text_feedback, 'String', 'Recording...')
end

drawnow;
data.TS = data.TT*data.SS;

popup_sel_index = get(handles.popupmenu2, 'Value');
switch popup_sel_index
    case 1
        data.VR = 0;
    case 2
        data.VR = 1;
    case 3
        data.VR = 2;
    case 4
        data.VR = 3;
end

% data.voltage=pclamp(1,data.CHlo,data.CHhi,data.VR,data.SS,data.TS);
% data.voltage = reshape(data.voltage, (data.CHhi-
data.CHlo+1),data.TS);
% [m n] = size(data.voltage)
% data.timestamp = data.voltage(m,:);
% data.voltage=data.voltage(data.CHlo+1:data.CHhi,:);
data.voltage = pclamp(0,0,6,data.VR,data.SS,data.TS);
data.voltage = reshape(data.voltage,7,data.TS);
data.timestamp = data.voltage(7,:);
data.voltage = data.voltage(data.CHlo+1:data.CHhi+1,:);

set(handles.text_feedback, 'String', 'Finished Recording.')

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of edit6 as text
%          str2double(get(hObject,'String')) returns contents of edit6
as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user eventdata (see
GUIeventdata)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents
as cell array
% contents{get(hObject,'Value')} returns selected item from
popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

set(hObject, 'String', {'10V', '5V', '2.5V', '1.25V'});

function text_feedback_Callback(hObject, eventdata, handles)
% hObject handle to text_feedback (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user eventdata (see
GUIeventdata)

% Hints: get(hObject,'String') returns contents of text_feedback as
text
% str2double(get(hObject,'String')) returns contents of

```

```

text_feedback as a double

% --- Executes during object creation, after setting all properties.
function text_feedback_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text_feedback (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_filename_Callback(hObject, eventdata, handles)
% hObject    handle to edit_filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_filename as
text
%         str2double(get(hObject,'String')) returns contents of
edit_filename as a double

% --- Executes during object creation, after setting all properties.
function edit_filename_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global data;

```

```

global fn_num;

DATE = datestr(now, 'yyyymmdd');
ex_val = get(handles.exercise_popupmenu, 'Value');
EX='';
switch(ex_val)
    case 1
        EX = 'PreIso1';
    case 2
        EX = 'PreIso2';
    case 3
        EX = 'PreIso3';
    case 4
        EX = 'PreIso4';
    case 5
        EX = 'PreIso5';
    case 6
        EX = 'Ex1';
    case 7
        EX='Ex2';
    case 8
        EX='Ex3';
    case 9
        EX='Ex4';
    case 10
        EX = 'PostIso1';
    case 11
        EX = 'PostIso2';
    case 12
        EX = 'PostIso3';
    case 13
        EX = 'PostIso4';
    case 14
        EX = 'PostIso5';
    case 15
        EX='Other';
end
SUBJECT = get( handles.subject_text, 'String');
if(fn_num < 10)
    FN_END = ['-0' num2str(fn_num) '.mat'];
else
    FN_END = ['- ' num2str(fn_num) '.mat'];
end

file_handle = [DATE '_' EX '_' SUBJECT FN_END];

fn_num = fn_num + 1;
if(fn_num < 10)
    FN_END = ['-0' num2str(fn_num) '.mat'];
else
    FN_END = ['- ' num2str(fn_num) '.mat'];
end
set(handles.filename_end_text, 'String', FN_END);

%file_handle = get(handles.edit_filename, 'String')
% save(file_handle{1,1}, 'data');

```

```

dir_listing = struct2cell(dir);
while( length(intersect(dir_listing(1,:),file_handle)) > 0 )
file_handle = [DATE '_' EX '_' SUBJECT FN_END];

fn_num = fn_num + 1;
if(fn_num < 10)
    FN_END = ['-0' num2str(fn_num) '.mat'];
else
    FN_END = ['- ' num2str(fn_num) '.mat'];
end
set(handles.filename_end_text,'String',FN_END);
end

save(file_handle, 'data');
set(handles.text_feedback,'String',['Saved data to ' file_handle]);

% --- Executes on button press in checkbox_channel1.
function checkbox_channel1_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_channel1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_channel1

% --- Executes on button press in checkbox4.
function checkbox4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox4

% --- Executes on button press in checkbox5.
function checkbox5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox5

% --- Executes on button press in checkbox_channel4.
function checkbox_channel4_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_channel4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hint: get(hObject,'Value') returns toggle state of checkbox_channel4

% --- Executes on button press in checkbox_channel5.
function checkbox_channel5_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_channel5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox_channel5

% --- Executes on button press in checkbox8.
function checkbox8_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox8

% --- Executes on button press in checkbox_timestamp.
function checkbox_timestamp_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox_timestamp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
checkbox_timestamp

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global data;

file_handle = get(handles.edit_filename, 'String')
load(file_handle);
set(handles.text_feedback, 'String', ['Loaded data from ' file_handle]);
set(handles.edit_filename, 'String', ['Loaded data from ' file_handle]);

function peak_array = peak_detection( channel_number , threshold_low ,
theshold_high )
global data;
[m n] = size(data.voltage(channel_number,:));
peak_array = zeroes(m,n);
peak bool = 0;

```

```

for i=1:n
    if(peak_bool == 0)
        if(data.voltage(channel_number,i) > threshold_high )
            peak_bool = 1;
        end
    else
        if(data.voltage(channel_number,i) < threshold_low )
            peak_bool = 0;
        end
    end
    peak_array(i) = peak_bool;
end

%general integration function
function integration_values =
triggered_integration(trigger_channel_number,
integration_channel_number, threshold_low, threshold_high,
integration_window )

global data;
channel_length = length(data.voltage(integration_channel_number));
%ASSERT: all channels are of same length;

trigger_peak_channel = peak_detection( trigger_channel_number,
threshold_low , threshold_high );
%three methods of determining the start time for integration:
% 1. edge flip
% 2. max value within peak
% 3. amplitude-weighted average (moment of rotation!)
%also may potentially want to use some relationship between these
different
%peak values in order to create an adaptive time window for
integration

%EDGE FLIP ALGORITHM!
edgeflip_indices = [];
for i=1:channel_length-1
    if( trigger_peak_channel(i) == 0 && trigger_peak_channel(i+1) == 1
)
        edgeflip_indices = [edgeflip_indices i];
    end
end
integration_values = [edgeflip_indices;edgeflip_indices];
%setup ==> final form will be integration_values(1,:) as the indices
and
integration_values(2,:) as the integration values
for i=1:length(edgeflip_indices)
    if(edgeflip_indices(i)+integration_window > channel_length)
        integration_sum = 0;
    else
        integration_sum = sum(
data.voltage(integration_channel_number,edgeflip_indices(i):edgeflip_i
ndices(i)+integration_window) );
    end
    integration values(2,i) = integration sum;

```

```

end

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu3

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor')
);
end

function subject_text_Callback(hObject, eventdata, handles)
% hObject    handle to subject_text (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of subject_text as
text
%         str2double(get(hObject,'String')) returns contents of
subject_text as a double

global fn_num;

fn_num = 1;
set( handles.filename_end_text , 'String' , '-01.mat' );

% --- Executes during object creation, after setting all properties.
function subject_text_CreateFcn(hObject, eventdata, handles)
% hObject    handle to subject text (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor')
);
end

% --- Executes during object creation, after setting all properties.
function filename_end_text_CreateFcn(hObject, eventdata, handles)
% hObject handle to filename_end_text (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

global fn_num;

fn_num = 1;

% --- Executes during object creation, after setting all properties.
function date_text_CreateFcn(hObject, eventdata, handles)
% hObject handle to date_text (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

set(hObject,'String',[datestr(now,'yyyymmdd') '_']);

% --- Executes during object creation, after setting all properties.
function pushbutton5_CreateFcn(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% --- Executes on key press over pushbutton5 with no controls
selected.
function pushbutton5_KeyPressFcn(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```



```

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
pushbutton5.
function pushbutton5_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
filename_end_text.
function filename_end_text_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to filename_end_text (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% global fn_num;
%
% fn_num = 1;
% set( hObject , 'String' , '-01.mat' );

% --- Executes on selection change in exercise_popupmenu.
function exercise_popupmenu_Callback(hObject, eventdata, handles)
% hObject    handle to exercise_popupmenu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns exercise_popupmenu
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
exercise_popupmenu

global fn_num;

fn_num = 1;
set( handles.filename_end_text , 'String' , '-01.mat' );

function edit_ymax_Callback(hObject, eventdata, handles)
% hObject    handle to edit_ymax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_ymax as text
%         str2double(get(hObject,'String')) returns contents of

```

```

edit_ymax as
%         a double

currentaxes = axis;
currentaxes(4) = str2double(get(hObject,'String'));
axis(handles.axes1,currentaxes);

% --- Executes during object creation, after setting all properties.
function edit_ymax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_ymax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit_ymin_Callback(hObject, eventdata, handles)
% hObject    handle to edit_ymin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_ymin as text
%         str2double(get(hObject,'String')) returns contents of
edit_ymin as a double

currentaxes = axis;
currentaxes(3) = str2double(get(hObject,'String'));
axis(handles.axes1,currentaxes);

% --- Executes during object creation, after setting all properties.
function edit_ymin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_ymin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

Appendix IV: MATLAB Classification Code

```
@abstract_vector_generator/abstract_vector_generator.m
classdef abstract_vector_generator < handle
    properties (SetAccess=protected,GetAccess=public)
        exercises = cell(0);
        vectordesc = [];
    end

    methods
        function this = abstract_vector_generator(varargin)
            if nargin > 0 && length(varargin) > 0
                emg_exercises = varargin{1};
                if isa(emg_exercises{1,1}, 'char')
                    this.load_exercises(emg_exercises);
                elseif isa(emg_exercises{1,1}, 'cell') &&
isa(emg_exercises{1,1}{1}, 'double')
                    this.emg_exercises = emg_exercises;
                else
                    disp('abstract_vector_generator: argument given,
but no exercises loaded. Wrong format.');
```

@abstract_vector_generator/load_exercises.m

```
function load_exercises(this,emg_cell_list)

size_emg_cell_list = size(emg_cell_list);
emg_cell_emg = cell(size_emg_cell_list);
for( i = 1:size_emg_cell_list(1) )
    for( j = 1:size_emg_cell_list(2) )
        patient_ex_list = emg_cell_list{i,j};
        emg_cell_emg_ij = cell(0);
        for( k = 1:size( patient_ex_list , 1 ) )
            load(strtrim(patient_ex_list(k,:)),'-MAT');
            if(isfield(data,'valid_regions'))
                emg_array = +slp.emg_filter(data.voltage);
                %emg_array = +slp.emg_detrend(data.voltage);
                region_i = 1;
                emg_size = size(emg_array);
                while( region_i <= emg_size(2) )
                    if( data.valid_regions(1,region_i) == 0 )
                        region_i = region_i + 1;
                    else
                        start_index = region_i;
                        while( region_i <= emg_size(2) &&
data.valid_regions(1,region_i) == 1 )
                            region_i = region_i+1;
                        end
                        end_index = region_i - 1;
                        emg_vector_region =
emg_array(:,start_index:end_index);
                        emg_cell_emg_ij{end+1,1} =
emg_vector_region;
                    end
                end
            else
                disp( 'Empty valid_regions array' );
            end
            emg_cell_emg{ij} = emg_cell_emg_ij;
        end
    end
end

this.exercises = emg_cell_emg;
```

@avgregions_vg/avgregions_vg.m

```
classdef avgregions_vg < abstract_vector_generator
    methods
        function this = avgregions_vg(varargin)
            this = this@abstract_vector_generator(varargin{:});
        end
    end
end
```

@avgregions_vg/genvector.m

```

function vector = genvector( this , exercise , varargin )
    %varargin is only for a vectordesc

if nargin > 2
    vectordesc = varargin{1};
else
    vectordesc = this.vectordesc;
end

emg_size = size(exercise);
vectordesc_size = size(vectordesc);
vector = zeros(emg_size(1),vectordesc_size(1));
emg_mean = mean(exercise,2);
for( i = 1:vectordesc_size(1) )
    start_index =
emg_in_bounds(round(vectordesc(i,1)*emg_size(2)),1,emg_size(2));
    stop_index =
emg_in_bounds(round(vectordesc(i,2)*emg_size(2)),1,emg_size(2));
    vector(:,i) =
mean(exercise(:,start_index:stop_index),2)./emg_mean;
end

vector = reshape(vector',1,emg_size(1)*vectordesc_size(1));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%SUBFUNCTION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function emg_bounds_value = emg_in_bounds( emg_value , emg_lower ,
emg_higher )

if(emg_value < emg_lower )
    emg_bounds_value = emg_lower;
elseif( emg_value > emg_higher )
    emg_bounds_value = emg_higher;
else
    emg_bounds_value = emg_value;
end

@avgregions_vg/mutate.m
function mutated = mutate(this,vectordesc)

mutated = this.randomdesc();

@avgregions_vg/pop2lda.m
function mapping = pop2lda(this)

mapping = [ 1 2 3 4 5 6 ...

```

```

1 2 3 4 5 6 ...
1 2 3 4 5 6 ...
1 2 3 4 5 6 ...
1 2 3 4 5 6 ...
1 2 3 4 5 6 ];

end

@avgregions_vg/randomdesc.m
function vectordesc = randomdesc(this)
vectordesc = rand(6,2);
for(i=1:6)
    vectordesc(i,:) = sort(vectordesc(i,:));
end

@toggle_gene/toggle_gene.m
classdef toggle_gene

    properties
        bp
        vg
    end

    methods
        function this = toggle_gene(vg,varargin)
            this.vg = vg;
            if nargin > 0 && length(varargin) > 0
                this.bp = varargin{1};
            else
                channel = floor(6*rand()+1);
                if(channel == 7)
                    channel = 6;
                end
                this.bp = [channel round(rand()+1) rand()];
            end
        end
    end
end

end

@toggle_gene/eq.m
function iseq = eq( gene1 , gene2 )

if size(gene1) ~= size(gene2)
    iseq = 0;
    disp( 'Error: gene1 == gene2: gene1 and gene2 must be same size'
)
    return;
end

```

```

if( length(gene1) ~= 1 )
    iseq = zeros(size(gene1));
    for i = 1:size(gene1,1)
        for j = 1:size(gene1,2)
            iseq(i,j) = gene1(i,j) == gene2(i,j);
        end
    end
    return;
end

if( gene1.bp == gene2.bp )
    iseq = 1;
    return;
end

g1_vectors = gene1.vg.extract_vectors(gene1);
g2_vectors = gene2.vg.extract_vectors(gene2);

num_exercises = size(g1_vectors,2);

iseq = 0;

for( i = 1:num_exercises )
    tf = cell2mat(g1_vectors(:,i)) == cell2mat(g2_vectors(:,i));
    tf = max(min(tf));
    %if either of the two vector columns that result from the gene
    are not linearly independent
    if( tf )
        iseq = 1;
    end
end

@toggle_gene/gt.m


---


function gtval = gt( gene1 , gene2 )

    gtval = gene1 >= gene2 && gene1 ~= gene2;

@toggle_gene/gte.m


---


function gteval = gte( gene1 , gene2 )

if( gene1 == gene2 )
    gteval = 1;
end

[y,i] = sortrows( [gene1.bp;gene2.bp] );
if( i(1) = 1 )
    gteval = 0;
else

```

```
gteval = 1;
end
```

@toggle_gene/lt.m

```
function ltval = lt( gene1 , gene2 )

    ltval = gene1 <= gene2 && gene1 ~= gene2;
```

@toggle_gene/lte.m

```
function lteval = lte( gene1 , gene2 )

if( gene1 == gene2 )
    lteval = 1;
end

[y,i] = sortrows( [gene1.bp;gene2.bp] );
if( i(1) = 1 )
    lteval = 1;
else
    lteval = 0;
end
```

@toggle_gene/ne.m

```
function n = ne( gene1 , gene2 )

n = ~(gene1 == gene2);
```

@toggle_gene/sort.m

```
function sorted = sort( gene_array );

for n = 1:size( gene_array, 2 )
    bp_array = zeros( size( gene_array, 1 ), 2 );

    for i = 1:length( gene_array )
        bp_array( i, : ) = gene_array( i, n ).bp;
    end

    [y, gene_i] = sortrows( bp_array );

    sorted( :, n ) = gene_array( gene_i, n );
end
```

@toggle_gene/sortrows.m

```
function [sorted, gene_i, J] = sortrows(gene_array)
```

```
bp_array = zeros(size(gene_array,1),3);
```

```
for i = 1:length(gene_array)
    bp_array(i,:) = gene_array(i,1).bp;
end
```

```
[y, gene_i] = sortrows(bp_array);
```

```
sorted = gene_array(gene_i,:);
```

```
J = {};
```

@toggle_gene/unique.m

```
function [niq,i,j] = unique( gene_array , rr )
```

```
if( nargin > 1 )
    if( rr ~= 'rows' )
        disp('unique on toggle_gene objects should only use
''rows''');
    end
end
```

```
j = {}; %j is for LOSERS... j/k
```

```
rep_i = [];
niq = sortrows(gene_array);
for( i = 2:size(niq,1) )
    comp_i = i-1;
    while( ~isempty(find(rep_i == comp_i)) )
        comp_i = comp_i - 1;
    end
    if( niq(comp_i) == niq(i) )
        if(round(rand()) == 0 )
            rep_i(end+1) = i;
        else
            rep_i(end+1) = comp_i;
        end
    end
end
```

```
niq(rep_i,:) = [];
```

```
i = 1:size(gene_array,1);
i(rep_i) = [];
```

@toggle_vg/toggle_vg.m

```
classdef toggle_vg < abstract_vector_generator
    methods
```

```

        function this = toggle_vg(varargin)
            this = this@abstract_vector_generator(varargin{:});
        end
    end
end

```

@toggle_vg/genvector.m

```

function vector = genvector( this , exercise , varargin )
    %varargin is only for a vectordesc
    %vectordesc is a 18x1 toggle_gene array

if nargin > 2
    vectordesc = varargin{1};
else
    vectordesc = this.vectordesc;
end

vector = zeros(1,length(vectordesc));

for i=1:length(vectordesc)
    channel = vectordesc(i).bp(1);
    tog_type = vectordesc(i).bp(2);%inactive to active/active to
inactive
    step_point = vectordesc(i).bp(3);
    vector(i) = nearest_step(
exercise{vectordesc(i).bp(1)}{tog_type} , step_point );
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function nn = nearest_step( step_array , step_point )
if( isempty(step_array) )
    nn = -3;
    return
end
step_diff = abs(step_array - step_point);
[y,step_i] = min(step_diff);

nn = step_array(step_i);

```

@toggle_vg/load_exercises.m

```

function load_exercises(this,emg_cell_list)

size_emg_cell_list = size(emg_cell_list);
emg_cell_emg = cell(size_emg_cell_list);
for( i = 1:size_emg_cell_list(1) )
    for( j = 1:size_emg_cell_list(2) )
        patient_ex_list = emg_cell_list{i,j};
        emg_cell_emg_ij = cell(0);
        for( k = 1:size( patient_ex_list , 1 ) )

```

```

load(strtrim(patient_ex_list(k,:)));
if(isfield(data,'valid_regions'))
    emg_array = data.voltage;
    region_i = 1;
    emg_size = size(emg_array);
    while( region_i <= emg_size(2) )
        if( data.valid_regions(1,region_i) == 0 )
            region_i = region_i + 1;
        else
            start_index = region_i;
            while( region_i <= emg_size(2) &&
data.valid_regions(1,region_i) == 1 )
                region_i = region_i+1;
            end
            end_index = region_i - 1;
            emg_vector_region =
emg_array(:,start_index:end_index);
            emg_cell_emg_ij{end+1,1} =
+slp.emg_filter(emg_vector_region);
            emg_thresh =
+slp.emg_find_threshold(emg_cell_emg_ij{end,1});
            emg_cell_emg_ij{end,1} =
+slp.emg_apply_threshold(emg_cell_emg_ij{end,1},emg_thresh);
            emg_cell_emg_ij{end,1} =
+slp.emg_threshold_stutter_join(emg_cell_emg_ij{end,1});
            emg_cell_emg_ij{end,1} =
+slp.emg_threshold_stutter_eliminate(emg_cell_emg_ij{end,1});
            emg_cell_emg_ij{end,1} =
toggle_loc(emg_cell_emg_ij{end,1});
        end
    end
    else
        disp( 'Empty valid_regions array' );
    end
    emg_cell_emg{i,j} = emg_cell_emg_ij;
end
end
end
end

this.exercises = emg_cell_emg;

```

@toggle_vg/ randomdesc.m

```

function vectordesc = randomdesc(this)
vectordesc = [toggle_gene(this)];
for( i=2:this.num_genes() )
    vectordesc(i,1) = toggle_gene(this);
end
vectordesc = unique(vectordesc);

while(size(vectordesc,1) < this.num_genes() )
    num_to_add = this.num_genes() - size(vectordesc,1);
    size(vectordesc,1)
    for( i = 1:num_to_add )
        vectordesc(end+1,1) = toggle_gene(this);
    end
end

```

```

    end
    vectordesc = unique(vectordesc);
end

```

@toggle_vg/mutate.m

```

function mutated = mutate(this,vectordesc)

```

```

%i = floor(6*rand()+1);
%if i == 7
%    i = 6;
%end
%mutated = vectordesc;

%mutated(i,:) = sort(rand(1,2));

mutated = this.randomdesc();

```

@toggle_vg/pop2lda.m

```

function mapping = pop2lda(this)

```

```

mapping = 1:this.num_genes();

end

```

@toggle_vg/private/toggle_loc.m

```

function locations = toggle_loc( emg_threshed )

```

```

num_channels = size(emg_threshed,1);
num_steps = size(emg_threshed,2);

locations = cell(num_channels,1);

for i=1:num_channels
    locations{i} = cell(2,1);
    locations{i}{1} = [];
    locations{i}{2} = [];
    state = emg_threshed(i,1);
    for step_i = 2:num_steps
        if(state ~= emg_threshed(i,step_i))
            locations{i}{state+1}(end+1) = step_i/num_steps;
        end
        state = emg_threshed(i,step_i);
    end
end
end

```

@toggle_vg/private/num_genes.m

```
function ng = num_genes( this );
```

```
    ng = 20;
```

```
end
```

@emg_ga_engine/emg_ga_engine.m

```
classdef emg_ga_engine < handle
```

```
    properties (Access=public)
```

```
        emg_ga_vg = emg_vector_generator();
```

```
        emg_ga_population = cell(0);
```

```
    end
```

```
    properties (SetAccess=private,GetAccess=public)
```

```
        emg_ga_lda_coeff = cell(0);
```

```
        emg_ga_roc = cell(0);
```

```
        emg_ga_score = [];
```

```
    end
```

```
    methods
```

```
        function this = emg_ga_engine(emg_vector_generator ,  
emg_ga_population)
```

```
            this.emg_ga_vg = emg_vector_generator;
```

```
            if( nargin > 1 )
```

```
                this.emg_ga_population = emg_ga_population;
```

```
            end
```

```
        end
```

```
    end
```

```
    methods (Access=private,Static)
```

```
        roc = calc_roc(found,expected);
```

```
        score = roc_score(ga_roc_cell);
```

```
        score = deviance_score(ga_lda_coeff_cell);
```

```
        [emg_cell_training , emg_cell_group , emg_to_classify ,  
emg_expected_groups ] = ...
```

```
        emg_create_training_group_from_cell(emg_cell_vectors,  
emg_subject , emg_exercise )
```

```
    end
```

```
end
```

@emg_ga_engine/init_pop.m

```
function init_pop(this,size)
```

```
    this.emg_ga_population = cell(size,1);
```

```
    for(i=1:size)
```

```
        disp( ['init vectordesc ' num2str(i) ' of ' num2str(size)]
```

```
);
```

```
        this.emg_ga_population{i} = this.emg_ga_vg.randomdesc();
```

```
    end
```

```
end
```

@emg_ga_engine/score_pop.m

```

function score_pop(this)

emg_ga_lda_coeff = cell(size(this.emg_ga_population,1),1);
emg_ga_roc = cell(size(this.emg_ga_population,1),1);
emg_ga_score = zeros(size(this.emg_ga_population,1),1);

%For each individual in the population
for n=1:size(this.emg_ga_population)

    emg_vectors =
this.emg_ga_vg.extract_vectors(this.emg_ga_population{n});

    emg_ga_lda_coeff{n} = cell(size(emg_vectors));
    emg_ga_roc{n} = cell(size(emg_vectors));

    %for each person/exercise pair
    for i=1:size(emg_vectors,1)
        for j=1:size(emg_vectors,2)

[emg_cell_training,emg_cell_group,emg_to_classify,emg_expected_group
s] = create_training_group(emg_vectors,i,j);
        [class,err,POSTERIOR,logp,coeff] =
classify(emg_to_classify,emg_cell_training,emg_cell_group);
        emg_ga_lda_coeff{n}{i,j} = coeff;
        emg_ga_roc{n}{i,j} =
calc_roc(class,emg_expected_groups);
        end
    end

    emg_ga_score(n) =
roc_score(emg_ga_roc{n})/deviance_score(emg_ga_lda_coeff{n});

end

this.emg_ga_lda_coeff = emg_ga_lda_coeff;
this.emg_ga_roc = emg_ga_roc;
this.emg_ga_score = emg_ga_score;

```

@emg_ga_engine/recombine_pop.m

```

function recombine_pop(this)
new_pop = cell(size(this.emg_ga_population));
for i=1:size(new_pop,1)
    %mommy and daddy are structs with field: pop,score,lda_coeff
    [mommy,m_i] = this.select_pop_to_mate();
    [daddy,d_i] = this.select_pop_to_mate();
    while( m_i == d_i )
        [daddy,d_i] = this.select_pop_to_mate();
    end
    new_pop{i} = this.mate_pops( mommy , daddy );
    if( rand() < .05 )
        new_pop{i} = this.emg_ga_vg.mutate(new_pop{i});
    end
end
end

```

```

        this.emg_ga_population = new_pop;
    end

```

@emg_ga_engine/private/calc_roc.m

```

function roc = calc_roc(found,expected);

roc = zeros(1,4);
for i=1:size(expected)
    roc_i = 4 - (found(i)*2+expected(i)); % 1 is positive, 0 is
negative, so [TP FP FN TP]
    roc(roc_i) = roc(roc_i) + 1;
end

```

@emg_ga_engine/private/create_training_group.m

```

function [emg_cell_training , emg_cell_group , emg_to_classify ,
emg_expected_groups ] = emg_create_training_group(emg_cell_vectors,
emg_subject , emg_exercise )
% Retrieve information about the size of the cell array and of the
arrays
% filling the individual cells of the array
size_emg_cell_vectors = size(emg_cell_vectors);
emg_cell_vector_count = zeros(size_emg_cell_vectors);
for( i=1:size_emg_cell_vectors(1) )
    for(j=1:size_emg_cell_vectors(2))
        emg_tmp_size = size(emg_cell_vectors{i,j});
        emg_cell_vector_count(i,j) = emg_tmp_size(1);
        if( emg_cell_vector_count(i,j) ~= 0 )
            emg_vector_size = emg_tmp_size(2);
        end
    end
end

%setup the training groups
emg_cell_training =
zeros(sum(sum([emg_cell_vector_count(1:emg_subject-
1,:);emg_cell_vector_count(emg_subject+1:end,:) ])),emg_vector_size);
emg_cell_group = zeros(size(emg_cell_training,1),1);
emg_training_rows = 1:size_emg_cell_vectors;
emg_training_rows(emg_subject) = [];
emg_training_row_index = 1;
for( i=emg_training_rows )
    for( j=1:size_emg_cell_vectors(2) )
        if(size(emg_cell_vectors{i,j},1) > 0 )
            emg_start_i = emg_training_row_index;
            emg_training_row_index = emg_start_i +
emg_cell_vector_count(i,j);
            emg_stop_i = emg_training_row_index - 1;
            emg_cell_training(emg_start_i:emg_stop_i,:) =
emg_cell_vectors{i,j};
            if( j == emg_exercise )
                emg_cell_group(emg_start_i:emg_stop_i) = 1;
            else
                emg_cell_group(emg_start_i:emg_stop_i) = 0;
            end
        end
    end
end

```

```

        end
    end
end

if( size(emg_cell_training,1) ~= size(emg_cell_group,1) )
    disp('Error: training groups dim unmatched');
end
i = NaN;

%setup the groups to classify
emg_to_classify =
zeros(sum(emg_cell_vector_count(emg_subject,:),),emg_vector_size);
emg_expected_groups = zeros(size(emg_to_classify,1),1);
emg_to_classify_row_index = 1;
for( j=1:size_emg_cell_vectors(2) )
    if(size(emg_cell_vectors{emg_subject,j},1) > 0 )
        emg_start_i = emg_to_classify_row_index;
        emg_to_classify_row_index = emg_start_i +
emg_cell_vector_count(emg_subject,j);
        emg_stop_i = emg_to_classify_row_index - 1;
        emg_to_classify(emg_start_i:emg_stop_i,:) =
emg_cell_vectors{emg_subject,j};
        if( j == emg_exercise )
            emg_expected_groups(emg_start_i:emg_stop_i) = 1;
        else
            emg_expected_groups(emg_start_i:emg_stop_i) = 0;
        end
    end
end

if( size(emg_to_classify,1) ~= size(emg_expected_groups,1) )
    disp('Error: to classify groups dim unmatched');
end

```

@emg_ga_engine/private/deviance_score.m

```

function score = deviance_score(coeff_cell)

num_exercises = size(coeff_cell,2);
num_people = size(coeff_cell,1);
exercise_deviance = zeros(1,num_exercises);

for i=1:num_exercises
    coeff_vectors =
ones(num_people,size(coeff_cell{1,1}(1,2).linear,1));
    for j=1:num_people
        coeff_struct = coeff_cell{j,i}(1,2);
        coeff_vectors(j,:) = coeff_struct.linear/coeff_struct.const;
    end
    exercise_deviance(i) = sqrt( mean(std(coeff_vectors,0,1).^2) );
end

score = sqrt( mean(exercise_deviance.^2,2) );

```


@emg_ga_engine/private/lda_contribution.m

```
function contrib = lda_contribution(coeff_cell)

num_exercises = size(coeff_cell,2);
num_people = size(coeff_cell,1);
exercise_lda = cell(num_exercises,1);

for i=1:num_exercises
    coeff_vectors =
ones(num_people,size(coeff_cell{1,1}(1,2).linear,1));
    for j=1:num_people
        coeff_struct = coeff_cell{j,i}(1,2);
        coeff_vectors(j,:) = coeff_struct.linear/coeff_struct.const;
    end
    exercise_lda{i} = sqrt( mean(coeff_vectors.^2,1) );
end

contrib = sqrt( mean(cell2mat(exercise_lda).^2,1) );
```

@emg_ga_engine/private/mate_pops.m

```
function child = mate_pops( this , mommy , daddy )
    %mommy and daddy are structs with fields pop,score,lda_coeff
    %child is just a pop

    mommy_lda_score = zeros(size(mommy.pop,1),1);
    daddy_lda_score = zeros(size(daddy.pop,1),1);

    pop_to_lda_mapping = this.emg_ga_vg.pop2lda();

    % mommy_cumulative_lda =
    reshape(mommy.lda_coeff,size(mommy.lda_coeff,1)*size(mommy.lda_coeff
    ,2),1);
    % mommy_cumulative_lda = mean(cell2mat(mommy_cumulative_lda),2);
    % daddy_cumulative_lda =
    reshape(daddy.lda_coeff,size(daddy.lda_coeff,1)*size(daddy.lda_coeff
    ,2),1);
    % daddy_cumulative_lda = mean(cell2mat(daddy_cumulative_lda),2);
    mommy_cumulative_lda = mommy.lda_coeff;
    daddy_cumulative_lda = daddy.lda_coeff;

    vector_indices = unique(pop_to_lda_mapping);
    for i=vector_indices
        lda_indices = find(pop_to_lda_mapping==i);
        mommy_lda_score(i) = sqrt(
    mean(mommy_cumulative_lda(lda_indices).^2) );
        daddy_lda_score(i) = sqrt(
    mean(daddy_cumulative_lda(lda_indices).^2) );
    end
```

```

gene_pool = [mommy.pop;daddy.pop];
gene_pool_score = [mommy_lda_score;daddy_lda_score];
[gene_pool,I] = sortrows(gene_pool);
gene_pool_score = gene_pool_score(I);
[gene_pool,I,J] = unique(gene_pool, 'rows');
gene_pool_score = gene_pool_score(I);

child = mommy.pop;
for child_i = 1:size(child,1);
    i = 0;
    countdown = sum(gene_pool_score)*rand();
    if countdown == 0
        i = 1;
    end
    while( countdown > 0 )
        i = i + 1;
        countdown = countdown - gene_pool_score(i);
    end
    child(child_i,:) = gene_pool(i,:);
    gene_pool_score(i) = 0;
end
end

```

@emg_ga_engine/private/roc_score.m

```

function score = roc_score(roc_cell)

num_exercises = size(roc_cell,2);
num_people = size(roc_cell,1);
roc_score_by_exercise = zeros( 1 , num_exercises );
roc_score_by_person = zeros( 1 , num_people );

for i = 1:num_exercises
    exercise_roc = sum(cell2mat(roc_cell(:,i)),1);
    tp_frac = exercise_roc(1)/(exercise_roc(1)+exercise_roc(2));
    tn_frac = exercise_roc(4)/(exercise_roc(3)+exercise_roc(4));
    roc_score_by_exercise(i) = sqrt( mean([tp_frac^2,tn_frac^2]) );
end
for i = 1:num_people
    person_roc =
sum(reshape(cell2mat(roc_cell(i,:)),num_exercises,4)',1);
    tp_frac = person_roc(1)/(person_roc(1)+person_roc(2));
    tn_frac = person_roc(4)/(person_roc(3)+person_roc(4));
    roc_score_by_person(i) = sqrt( mean([tp_frac^2,tn_frac^2]) );
end

score = sqrt( mean( [roc_score_by_exercise roc_score_by_person].^2 )
);

```

@emg_ga_engine/private/select_pop_to_mate.m

```

function [parent,i] = select_pop_to_mate(this)
    i = 0;

```

```

countdown = sum(this.emg_ga_score)*rand();
if countdown == 0
    i = 1;
end
while( countdown > 0 )
    i = i + 1;
    countdown = countdown - this.emg_ga_score(i);
end
parent.pop = this.emg_ga_population{i};
parent.score = this.emg_ga_score(i);
parent.lda_coeff = lda_contribution(this.emg_ga_lda_coeff{i});
end

```

+slp/emg_threshold_stutter_join.m

```

function [emg_joined,emg_stutter_areas] =
emg_threshold_stutter_join( emg_thresh , emg_min_time ,
emg_sampspsec)

import slp.*;

if(nargin < 2 || isempty(emg_min_time) )
    emg_min_time = 0.2;%seconds
end
if(nargin < 3 || isempty(emg_sampspsec) )
    emg_sampspsec = 1000;
end

emg_min_window = round(emg_min_time*emg_sampspsec);

emg_size = size(emg_thresh);

emg_joined = emg_thresh;
emg_stutter_areas = zeros(emg_size);

emg_last_value = emg_thresh(:,1);
emg_last_flip = ones(emg_size(1),1);

%start marking stutter areas
for( j = 1:emg_size(2) - 1 )
    for(i = 1:emg_size(1))
        if(emg_thresh(i,j) ~= emg_last_value(i) )
            emg_last_value(i) = emg_thresh(i,j);
            if(j - emg_last_flip(i) < emg_min_window)
                emg_stutter_areas(i,emg_last_flip(i):j) = ones(1,j-
emg_last_flip(i)+1);
            end
            emg_last_flip(i) = j;
        end
    end
end
end

j = emg_size(2);

```

```

for(i = 1:emg_size(1))
    if(j - emg_last_flip(i) < emg_min_window)
        emg_stutter_areas(i,emg_last_flip(i):j) = ones(1,j-
emg_last_flip(i)+1);
    end
end
%end marking stutter areas

%start consolidate stutter areas
for( i = 1:emg_size(1) )
    j = 1;
    while( j <= emg_size(2) )
        if(emg_stutter_areas(i,j) == 1)
            emg_window_start = j;
            while( j <= emg_size(2) && emg_stutter_areas(i,j) ==1 )
                j = j + 1;
            end
            emg_joined(i,emg_window_start:j-1) =
round(mean(emg_thresh(i , emg_window_start:j-1) ));
            if( j-emg_window_start >= emg_min_window )
                emg_stutter_areas(i,emg_window_start:j-1) =
zeros(1,j-emg_window_start);
            end
        else
            j = j + 1;
        end
    end
end
%end consolidate stutter areas

```

+slp/emg_threshold_stutter_eliminate.m

```

function emg_elim = emg_threshold_stutter_eliminate( emg_thresh ,
emg_min_time , emg_sampspsec)

import slp.*;

if(nargin < 2 || isempty(emg_mintime) )
    emg_min_time = 0.2;%seconds
end
if(nargin < 3 || isempty(emg_sampspsec) )
    emg_sampspsec = 1000;
end

[emg_elim, emg_stutter_areas] = emg_threshold_stutter_join(
emg_thresh , emg_min_time , emg_sampspsec );

emg_min_window = round(emg_min_time*emg_sampspsec);

emg_size = size(emg_stutter_areas);

%start consolidate stutter areas
for( i = 1:emg_size(1) )
    j = 1;

```

```

while( j <= emg_size(2) )
    if(emg_stutter_areas(i,j) == 1)
        emg_window_start = j;
        while( j <= emg_size(2) && emg_stutter_areas(i,j) ==1 )
            j = j + 1;
        end
        %stutter_area_found in range (emg_window_start:j-1)
        if( j-emg_window_start < emg_min_window ) %window is too
small
            if( j <= emg_size(2) )
                emg_elim(i,emg_window_start:j-1) =
emg_elim(i,j)*ones(1,j-emg_window_start);
            elseif( emg_window_start > 0 )
                emg_elim(i,emg_window_start:j-1) =
emg_elim(i,emg_window_start)*ones(1,j-emg_window_start);
            end
        end
    else
        j = j + 1;
    end
end
end
%end consolidate stutter areas

```

+slp/emg_threshold_filter_test.m

```

function emg_threshold_filter_test( emg_array )

import slp.*;

    emg_det = abs(detrend(emg_array));
    emg_filt = emg_filter( emg_array );
    emg_thresh = emg_find_threshold( emg_array ) .* ones(
size(emg_array) );

```

+slp/emg_histogram.m

```

function [emg_hist , emg_bins] = emg_histogram( emg_array ,
emg_num_bins )

import slp.*;

if( nargin < 2 || isempty( emg_num_bins ) )
    emg_num_bins = 50;
end

emg_size = size(emg_array);
emg_bins = zeros( emg_size(1) , emg_num_bins );
emg_hist = zeros( emg_size(1) , emg_num_bins );

emg_min = min( emg_array , [] , 2 );
emg_max = max( emg_array , [] , 2 );
emg_range = emg_max - emg_min;

```

```

for( i = 1:emg_num_bins )
    emg_bins(:,i) = emg_min + emg_range * (i-0.5)/emg_num_bins;
end

for( i = 1:emg_size(1) )
    for( j = 1:emg_size(2) )
        if( emg_array(i,j) == emg_max(i) )
            emg_bin_num = emg_num_bins;
        else
            emg_bin_num = floor((emg_array(i,j) -
emg_min(i))*emg_num_bins/emg_range(i)) + 1;
        end
        emg_hist(i,emg_bin_num) = emg_hist(i,emg_bin_num) + 1;
    end
end
end

```

+slp/emg_find_threshold.m

```

function emg_thresh = emg_find_threshold(emg_array)

import slp.*;

[emg_hist , emg_bins] = emg_histogram( emg_array );
emg_drv = emg_derivative(emg_hist);
[emg_min,emg_mindex] = min(emg_drv, [], 2);

emg_thresh = zeros( size(emg_array, 1) , 1 );

for( i = 1:size(emg_drv,1) )
    emg_flag_zero = false; %zero was found after the min
    for( j = emg_mindex(i):size(emg_drv,2) )
        if(emg_drv(i,j) >= 0 )
            emg_thresh(i) = emg_bins(i,j);
            emg_flag_zero = true;
            break;
        end
    end
    if( ~emg_flag_zero )
        if( emg_mindex*2 <= size(emg_bins(1)) )
            emg_thresh(i) = emg_bins(i,emg_mindex*2);
        else
            emg_thresh(i) = emg_bins(i,size(emg_drv,2));
        end
    end
end
end

```

+slp/emg_filter.m

```

function emg_filtered = emg_filter( emg_array , emg_samppersec )

import slp.*;

```

```

if( nargin < 2 || isempty( emg_samppersec ) )
    emg_samppersec = 1000;
end

emg_filtered = emg_detrend(emg_array);
[b,a]=butter(5,10/emg_samppersec,'low');
emg_filtered = filtfilt(b,a,emg_filtered)';

```

+slp/emg_detrend.m

```

function emg_det = emg_detrend( emg_array );

import slp.*;

emg_size = size(emg_array);
emg_det = zeros(emg_size);

for( i=1:emg_size(1) )
    emg_det(i,:) = abs(detrend(emg_array(i,:)));
end

```

+slp/emg_derivative.m

```

function emg_drv = emg_derivative( emg_array )

import slp.*;

emg_array_size = size(emg_array);
emg_drv = ones( emg_array_size );
for( i=2:emg_array_size(2)-1 )
    emg_drv(:,i) = (emg_array(:,i+1)-emg_array(:,i-1))/2;
end

emg_drv(:,1) = emg_array(:,2) - emg_array(:,1);
emg_drv(:,emg_array_size(2)) = emg_array(:,emg_array_size(2)) -
emg_array( :, emg_array_size(2) - 1 );

```

+slp/emg_apply_threshold.m

```

function emg_threshed = emg_apply_threshold( emg_array , emg_thresh
)

import slp.*;

emg_size = size(emg_array);
emg_threshed = zeros(emg_size);

for( i = 1:emg_size(1) )
    for( j = 1:emg_size(2) )
        if( emg_array(i,j) > emg_thresh(i) )

```

```

        emg_threshed(i,j) = 1;
    end
end
end

+dtw/dpfast.m
function [p,q,D,sc] = dpfast(M,C,T,G)
% [p,q,D,sc] = dpfast(M,C,T,G)
%   Use dynamic programming to find a min-cost path through matrix
M.
%   Return state sequence in p,q; full min cost matrix as D and
%   local costs along best path in sc.
%   This version gives the same results as dp.m, but uses
dpcore.mex
%   to run ~200x faster.
%   C is a step matrix, with rows (i step, j step, cost factor)
%   Default is [1 1 1.0;0 1 1.0;1 0 1.0];
%   Another good one is [1 1 1;1 0 1;0 1 1;1 2 2;2 1 2]
%   T selects traceback origin: 0 is to any edge; 1 is top right
(default);
%   T > 1 finds path to min of anti-diagonal T points away from
top-right.
%   Optional G defines length of 'gulleys' for T=0 mode; default
0.5
%   (i.e. accept path to only 50% of edge nearest top-right)
% 2003-04-04,2005-04-04 dpwe@ee.columbia.edu $Header:
/Users/dpwe/projects/dtw/RCS/dpfast.m,v 1.6 2008/03/14 14:40:50 dpwe
Exp $

% Copyright (c) 2003 Dan Ellis <dpwe@ee.columbia.edu>
% released under GPL - see file COPYRIGHT

import dtw.*;

if nargin < 2
    % Default step / cost matrix
    C = [1 1 1.0;0 1 1.0;1 0 1.0];
end

if nargin < 3
    % Default: path to top-right
    T = 1;
end

if nargin < 4
    % how big are gulleys?
    G = 0.5; % half the extent
end

if sum(isnan(M(:)))>0
    error('dpwe:dpfast:NAN','Error: Cost matrix includes NaNs');
end

```



```

if min(M(:)) < 0
    disp('Warning: cost matrix includes negative values; results may
not be what you expect');
end

[r,c] = size(M);

% Core cumulative cost calculation coded as mex
[D,phi] = dpcore(M,C);

p = [];
q = [];

%% Traceback from top left?
%i = r;
%j = c;

if T == 0
    % Traceback from lowest cost "to edge" (gulleys)
    TE = D(r,:);
    RE = D(:,c);
    % eliminate points not in gulleys
    TE(1:round((1-G)*c)) = max(max(D));
    RE(1:round((1-G)*r)) = max(max(D));
    if (min(TE) < min(RE))
        i = r;
        j = max(find(TE==min(TE)));
    else
        i = max(find(RE==min(RE)));
        j = c;
    end
else
    % Traceback from min of antidiagonal
    %stepback = floor(0.1*c);
    stepback = T;
    slice = diag(fliplr(D), -(r-stepback));
    [mm,ii] = min(slice);
    i = r - stepback + ii;
    j = c + 1 - ii;
end

p=i;
q=j;

sc = M(p,q);

while i > 1 & j > 1
    % disp(['i=',num2str(i),' j=',num2str(j)]);
    tb = phi(i,j);
    i = i - C(tb,1);
    j = j - C(tb,2);
    p = [i,p];
    q = [j,q];
    sc = [M(i,j),sc];
end

```

+dtw/dpcore.c

```
/*
 * dpcore.c
 * Core of dynamic programming/DTW calculation
 * 2003-04-02 dpwe@ee.columbia.edu
 * $Header: /Users/dpwe/projects/dtw/RCS/dpcore.c,v 1.4 2009/07/27
22:54:53 dpwe Exp $
% Copyright (c) 2003-05 Dan Ellis <dpwe@ee.columbia.edu>
% released under GPL - see file COPYRIGHT
 */

#include <stdio.h>
#include <math.h>
#include <ctype.h>
#include "mex.h"

/* #define DEBUG */

/* #define INF HUGE_VAL */
#define INF DBL_MAX

void
mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray
*prhs[])
{
    int i,j;
    long pvl, pvb[16];

#ifdef DEBUG
    mexPrintf("dpcore: Got %d lhs args and %d rhs args.\n",
        nlhs, nrhs);
    for (i=0;i<nrhs;i++) {
        mexPrintf("RHArg #%d is size %d x %d\n",
            (long)i, mxGetM(prhs[i]), mxGetN(prhs[i]));
    }
    for (i=0;i<nlhs;i++)
        if (plhs[i]) {
            mexPrintf("LHArg #%d is size %d x %d\n",
                (long)i, mxGetM(plhs[i]), mxGetN(plhs[i]));
        }
#endif /* DEBUG */

    if (nrhs < 1){
        mexPrintf("dpcore [D,P] = dpcore(S[,C]) dynamic programming
core\n");
        mexPrintf(" Calculate the best cost to every point in
score\n");
        mexPrintf(" cost matrix S; return it in D along with
traceback\n");
        mexPrintf(" indices in P. Optional C defines allowable
steps\n");
        mexPrintf(" and costs; default [1 1 1.0;1 0 1.0;0 1
1.0]\n");
    }
}
```

```

return;
}

if (nlhs > 0){
mxArray *DMatrix, *PMatrix;
int rows, cols, i, j, k, tb;
double *pM, *pD, *pP, *pC;
double d1, d2, d3, v;
double *costs;
int *steps;
int ncosts;

rows = mxGetM(prhs[0]);
cols = mxGetN(prhs[0]);
pM = mxGetPr(prhs[0]);

DMatrix = mxCreateDoubleMatrix(rows, cols, mxREAL);
pD = mxGetPr(DMatrix);
PMatrix = mxCreateDoubleMatrix(rows, cols, mxREAL);
pP = mxGetPr(PMatrix);
plhs[0] = DMatrix;
if (nlhs > 1) {
    plhs[1] = PMatrix;
}

/* setup costs */
if (nrhs == 1) {
    /* default C matrix */
    int ii;

    ncosts = 3;
    costs = (double *)malloc(ncosts*sizeof(double));
    for (ii = 0; ii<ncosts; ++ii) costs[ii] = 1.0;
    steps = (int *)malloc(ncosts*2*sizeof(int));
    steps[0] = 1;    steps[1] = 1;
    steps[2] = 1;    steps[3] = 0;
    steps[4] = 0;    steps[5] = 1;
} else {
    int ii, crows, ccols;
    crows = mxGetM(prhs[1]);
    ccols = mxGetN(prhs[1]);
    pC = mxGetPr(prhs[1]);
    /* mexPrintf("C has %d rows and %d cols\n", crows, ccols);
*/

    if (ccols != 3) {
        mexPrintf("Cost matrix must have 3 cols (i step, j step,
cost factor)\n");
        return;
    }
    ncosts = crows;
    costs = (double *)malloc(ncosts*sizeof(double));
    steps = (int *)malloc(ncosts*2*sizeof(int));
    for (ii = 0; ii < ncosts; ++ii) {
        steps[2*ii] = (int) (pC[ii]);
        steps[2*ii+1] = (int) (pC[ii+crows]);
        costs[ii] = pC[ii+2*crows];

```

```

        /* mexPrintf("step=%d,%d cost=%f\n",
steps[2*ii],steps[2*ii+1],costs[ii]); */
    }
}

/* do dp */
v = 0;
tb = 1; /* value to use for 0,0 */
for (j = 0; j < cols; ++j) {
    for (i = 0; i < rows; ++i) {
        d1 = pM[i + j*rows];
        for (k = 0; k < ncosts; ++k) {
            if ( i >= steps[2*k] && j >= steps[2*k+1] ) {
                d2 = costs[k]*d1 + pD[(i-steps[2*k]) + (j-
steps[2*k+1])*rows];
                if (d2 < v) {
                    v = d2;
                    tb = k+1;
                }
            }
        }

        pD[i + j*rows] = v;
        pP[i + j*rows] = (double)tb;
        v = INF;
    }
}
free((void *)costs);
free((void *)steps);
}

#ifdef DEBUG
    mexPrintf("dpcore: returning...\n");
#endif /* DEBUG */
}

```

temporary_viewer.m

```

function temporary_viewer( ex , exnum , ch , ch2 , moviefile )

figure;
pause(0.5);
hsv_val = 0;

arghasmovie = 0;
arghasch2 = 0;

if( nargin > 3 && ~isempty(ch2) )
    arghasch2 = 1;
end
if( nargin > 4 && ~isempty(moviefile) )
    arghasmovie = 1;
end

```

```

pause_length = 1.0;
if( arghasch2 )
    pause_length = 2.0;
end
if( arghasmovie )
    pause_length = 0;
    pos = get(gcf, 'Position');
    pos(1:2) = [0,0];
    numframes = 0;
    for(i=1:size(ex,1))
        persex=ex{i,exnum};
        numframes = numframes+size(persex,1);
    end
    moviearray = moviein(numframes,gcf,pos);
    framenum = 0;
end
widthadd = 0;

for(i=1:size(ex,1))
    persex = ex{i,exnum};
    hsv_val = mod(hsv_val + 0.19,1);
    col = hsv2rgb( [hsv_val , 0.8 , 0.9] );
    for(j = 1:size(persex,1));
        pause(pause_length);
        a=persex{j};
        if( arghasmovie )
            %add width as another distinguishing factor between people,
            because the color isn't as vibrant
            if(mod(i,2)==1)
                widthadd = 2;
            else
                widthadd = 0;
            end
        end
        plot(a(ch,:), 'Color', col , 'LineWidth', 2+widthadd);
        if( arghasch2 )
            hold on;
            scale_factor = mean(a(ch,:))/mean(a(ch2,:));
            plot(
a(ch2,:)*scale_factor, 'Color',hsv2rgb([mod(hsv_val+0.05,1) , 0.8 ,
0.6]), 'LineWidth', 3+widthadd , 'LineStyle', '--');
            hold off;
        end
        set(gca, 'Color', 'black');
        if( arghasmovie )
            framenum = framenum+1;
            moviearray(:, framenum)=getframe(gcf,pos);
        end
    end
end
end

pause(pause_length);
for(i=1:size(ex,1))
    persex = ex{i,exnum};

```

```

    hsv_val = i/size(ex,1);
    col = hsv2rgb( [hsv_val , 0.8 , 0.8] );
    for(j = 1:size(persex,1));
        a=persex{j}(ch,:);
        plot((1:size(a,2))*1000/size(a,2),a/mean(a), 'Color',col );
        set(gca, 'Color', 'black');
        hold on;
    end
end

if( arghasmovie )
    movie2avi(moviearray,moviefile, 'fps',1, 'colormap',hsv);
end

```

resample_arr.m

```

function r = resample_arr( arr , p , q )

for( i = 1:size(arr,1) )
    r(i,:) = resample(arr(i,:)/mean(arr(i,:)), p , q );
end

```

emg_marking_facilitator.m

```

function varargout = emg_marking_facilitator(varargin)
% EMG_MARKING_FACILITATOR M-file for emg_marking_facilitator.fig
%     EMG_MARKING_FACILITATOR, by itself, creates a new
EMG_MARKING_FACILITATOR or raises the existing
%     singleton*.
%
%     H = EMG_MARKING_FACILITATOR returns the handle to a new
EMG_MARKING_FACILITATOR or the handle to
%     the existing singleton*.
%
%
EMG_MARKING_FACILITATOR('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in EMG_MARKING_FACILITATOR.M with the
given input arguments.
%
%     EMG_MARKING_FACILITATOR('Property','Value',...) creates a new
EMG_MARKING_FACILITATOR or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before emg_marking_facilitator_OpeningFcn
gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
emg_marking_facilitator_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```

% Edit the above text to modify the response to help
emg_marking_facilitator

% Last Modified by GUIDE v2.5 08-Sep-2009 02:39:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @emg_marking_facilitator_OpeningFcn, ...
                  'gui_OutputFcn',  @emg_marking_facilitator_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before emg_marking_facilitator is made visible.
function emg_marking_facilitator_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to emg_marking_facilitator (see
VARARGIN)

% Choose default command line output for emg_marking_facilitator
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes emg_marking_facilitator wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = emg_marking_facilitator_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);

```

```

% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton_mark.
function pushbutton_mark_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_mark (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
global data;
start_sample =
round(str2num(get(handles.edit_start, 'String'))*data.SS);
stop_sample =
round(str2num(get(handles.edit_stop, 'String'))*data.SS);
switch( get(handles.bg_mark, 'SelectedObject' ) )
    case handles.radio_valid
        data.valid_regions(start_sample:stop_sample) =
ones(1, stop_sample-start_sample+1);
    case handles.radio_invalid
        data.valid_regions(start_sample:stop_sample) =
zeros(1, stop_sample-start_sample+1);
end

bg_channel_SelectionChangeFcn(get(handles.bg_channel, 'SelectedObject
'), [], handles)

function edit_start_Callback(hObject, eventdata, handles)
% hObject    handle to edit_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit_start as
text
%         str2double(get(hObject, 'String')) returns contents of
edit_start as a double

% --- Executes during object creation, after setting all properties.
function edit_start_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),

```



```

get(0,'defaultUicontrolBackgroundColor')
    set(hObject,'BackgroundColor','white');
end

function edit_stop_Callback(hObject, eventdata, handles)
% hObject    handle to edit_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit_stop as text
%        str2double(get(hObject,'String')) returns contents of
edit_stop as a double

% --- Executes during object creation, after setting all properties.
function edit_stop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when selected object is changed in bg_channel.
function bg_channel_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in bg_channel
% eventdata  structure with the following fields (see UIBUTTONGROUP)
%   EventName: string 'SelectionChanged' (read only)
%   OldValue: handle of the previously selected object or empty if
none was selected
%   NewValue: handle of the currently selected object
% handles    structure with handles and user data (see GUIDATA)
import slp.*;
global data;
global threshold_regions;
axes_data = handles.axes_data;
axes_valid = handles.axes_valid;
data_to_plot = [];
switch( hObject )
    case handles.radio_all
        data_to_plot = data.voltage;
    case handles.radio_combined
        data_to_plot = data.voltage;
    case handles.radio_tibant
        data_to_plot = data.voltage(1,:);
    case handles.radio_vastmed
        data_to_plot = data.voltage(2,:);

```

```

    case handles.radio_bicfem
        data_to_plot = data.voltage(3,:);
    case handles.radio_gastmed
        data_to_plot = data.voltage(4,:);
    case handles.radio_tfl
        data_to_plot = data.voltage(5,:);
    case handles.radio_glutmed
        data_to_plot = data.voltage(6,:);
end
data_to_plot = emg_filter(data_to_plot);
if( hObject == handles.radio_combined )
    voltage_mean = mean(data_to_plot,2)
    voltage_scale = 1./voltage_mean;
    data_to_plot = diag(voltage_scale)*data_to_plot;
    data_to_plot = max(data_to_plot);
end
threshold_regions =
emg_threshold_stutter_eliminate(emg_apply_threshold(data_to_plot,emg
_find_threshold(data_to_plot)));
if( size(threshold_regions,1) > 1 )
    for( i = 2:size(threshold_regions,1) )
        threshold_regions(1,:) = threshold_regions(1,:) |
threshold_regions(i,:);
    end
end
threshold_regions = threshold_regions(1,:);
data_to_plot = [data_to_plot;
1.5*max(max(data_to_plot))*threshold_regions ];
bar( axes_valid , (1:size(data.valid_regions,2))/data.SS ,
data.valid_regions , 'hist' );
plot( axes_data , (1:size(data.voltage,2))/data.SS , data_to_plot );
%reset clicky clicky callback for the axes_data
b_down_callback =
@(hObj,eData)emg_marking_facilitator('axes_data_ButtonDownFcn',hObj,
eData,guidata(hObj));
set( axes_data , 'ButtonDownFcn' , b_down_callback );
%make valid regions plot line up with the other plot
axis_data = axis(axes_data);
axis_data(3:4) = [0 1];
axis(axes_valid,axis_data);

% --- Executes on selection change in popupmenu_filelist.
function popupmenu_filelist_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu_filelist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu_filelist
contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
popupmenu_filelist
global prefix;
contents = get(hObject,'String');
selected = strtrim(contents(get(hObject,'Value'),:));

```

```

if(isdir([prefix selected]))
    prefix = [prefix selected];
    set(hObject, 'Value', 1);
    set(hObject, 'String', format_directory_string(ls(prefix)))
end

% --- Process Directory Strings to add trailing slash
function slashed_list = format_directory_string(list)
global prefix;
slashed_list = [];
listsize = size(list);
for i = 1:listsize(1)
    if( isdir([prefix strtrim(list(i,:))]) )
        slashed_str = [strtrim(list(i,:)) '/'];
        slashed_list = strvcats(slashed_list,slashed_str);
    else
        slashed_list = strvcats(slashed_list,list(i,:));
    end
end

% --- Executes during object creation, after setting all properties.
function popupmenu_filelist_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu_filelist (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
global prefix;
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
set(hObject, 'String', format_directory_string(ls))
prefix = './';

% --- Executes on button press in pushbutton_save.
function pushbutton_save_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data;
global prefix;

contents = get(handles.popupmenu_filelist, 'String');
selected =
strtrim(contents(get(handles.popupmenu_filelist, 'Value'), :));

```

```

save( [prefix selected] , 'data' );
set_status( 'Saved File' , handles );

% --- Executes on button press in pushbutton_load.
function pushbutton_load_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton_load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data;
global prefix;
filelist = handles.popupmenu_filelist;

filelist_contents = get(filelist, 'String');
filelist_selected =
strtrim(filelist_contents(get(filelist, 'Value'),:));

load( [prefix filelist_selected] );
if( ~isfield(data, 'valid_regions') )
    %data.valid_regions does not exist
    data.valid_regions = zeros( 1 , size(data.voltage,2) );
end

bg_channel_SelectionChangeFcn(get(handles.bg_channel, 'SelectedObject
'), [], handles)
set_status( 'Loaded File' , handles );

% --- Executes on mouse press over axes background.
function axes_data_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes_data (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global data;
global threshold_regions;

current_point = get(hObject, 'CurrentPoint');
current_sample = round( current_point(1,1)*data.SS );
if(current_sample <= size(threshold_regions,2) && current_sample >=
0 )
    selected_region = zeros(size(threshold_regions));
    if( threshold_regions(1,current_sample) == 1 )
        i = current_sample;
        while( i > 0 && threshold_regions(1,i) == 1 )
            selected_region(1,i) = 1;
            i = i - 1;
        end
        i = current_sample;
        while( i <= size(threshold_regions,2) &&
threshold_regions(1,i) == 1 )
            selected_region(1,i) = 1;
            i = i + 1;
        end
    end
end
end

```

```

s_object = get(handles.bg_mark, 'SelectedObject');

switch( s_object )
    case handles.radio_valid
        data.valid_regions = data.valid_regions |
selected_region;
    case handles.radio_invalid
        data.valid_regions = data.valid_regions &
~selected_region;
    end

bg_channel_SelectionChangeFcn(get(handles.bg_channel, 'SelectedObject
'), [], handles)
end

function set_status( status , handles )
set(handles.text_status, 'String', status);
% return_func = @(source,event)
set(handles.text_status, 'String', 'Status');
% t = timer( 'TimerFcn' , return_func , 'StartDelay' , 5 );
% start(t);

% % --- Executes on key press with focus on figure1 or any of its
controls.
% function figure1_WindowKeyPressFcn(hObject, eventdata, handles)
% % hObject      handle to figure1 (see GCBO)
% % eventdata    structure with the following fields (see FIGURE)
% % Key: name of the key that was pressed, in lower case
% % Character: character interpretation of the key(s) that was
pressed
% % Modifier: name(s) of the modifier key(s) (i.e., control, shift)
pressed
% % handles      structure with handles and user data (see GUIDATA)
% switch(eventdata.Character)
%     case '0'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_all);
%     case '1'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_tibant);
%     case '2'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_vastmed);
%     case '3'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_bicfem);
%     case '4'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_gastmed);
%     case '5'
%
set(handles.bg_channel, 'SelectedObject', handles.radio_tfl);
%     case '6'
%

```

```

set(handles.bg_channel,'SelectedObject',handles.radio_glutmed);
%     case 'v'
%         set(handles.bg_mark,'SelectedObject',handles.radio_valid);
%     case 'i'
%
set(handles.bg_mark,'SelectedObject',handles.radio_invalid);
% end
%
bg_channel_SelectionChangeFcn(get(handles.bg_channel,'SelectedObject
'),[], handles)

```

```

% --- Executes during object creation, after setting all properties.
function figure1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
global data;
data = [];

```

dtw_viewer.m

```

function temporary_viewer( target , ex , exnum , ch , moviefile )

resample_length = 500;

disp( 'Setting fixed 2 bump target for now' );
target = zeros(4,1000);
peak_width=50;
target( 1,: ) = 20-cos( (0:999) * 2 * pi/peak_width );
peak_width = 400;
target( 2,300:300+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );
peak_width = 200;
target( 3,200:200+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );
target( 3,600:600+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );
target( 4,100:100+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );
target( 4,400:400+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );
target( 4,700:700+peak_width ) = 1-cos( (0:peak_width) * 2 *
pi/peak_width );

target = resample_arr(target,resample_length,length(target));

hsv_val = 0;
col = hsv2rgb( [hsv_val , 0.8 , 0.9] );

arghasmovie = 0;

if( nargin > 5 && ~isempty(moviefile) )

```

```

    arghasmovie = 1;
end

pause_length = 1.0;
if( arghasmovie )
    pause_length = 0;
    pos = get(gcf, 'Position');
    pos(1:2) = [0,0];
    numframes = 0;
    for(i=1:size(ex,1))
        persex=ex{i,exnum};
        numframes = numframes+size(persex,1);
    end
    moviearray = moviein(numframes,gcf,pos);
    framenum = 0;
end

disp( 'Squashing exercises' );
ex = dtw_squashex( ex , exnum );

for( i = 1:size(ex,1) )
    disp(['Resampling sample ' num2str(i) ' of ' num2str(size(ex,1))
])
    ex{i,1} = resample_arr(ex{i},resample_length,length(ex{i}));
end

fitted_target = {};
scores = [];
for(i=1:size(ex,1))
    %score and fit!
    disp(['Scoring sample ' num2str(i) ' of ' num2str(size(ex,1)) ])

    curr_ex = ex{i}(ch,:);
    [fitted_target{end+1},p,q,D,sc] = dtw_transform( target(1,:) ,
curr_ex );
    ft = fitted_target{end};

    sc = (ft-curr_ex);
    sc = sqrt(mean(sc.*sc));
    scores(end+1) = sc;

    for(j = 2:size(target,1))
        [ft,p,q,D,sc] = dtw_transform( target(j,:) , curr_ex );
        sc = (ft-curr_ex);
        sc = sqrt(mean(sc.*sc));
        if( sc < scores(end) )
            fitted_target{end} = ft;
            scores(end) = sc;
        end
    end
end

[scores_sorted,scores_i] = sort( scores );

fitted_target = fitted_target(scores_i);

```

```

ex = ex(scores_i);

figure;
for(i=1:size(ex,1))
    pause(pause_length);
    a=ex{i};
    plot(a(ch,:), 'Color', col );
    hold on;
    plot(fitted_target{i}, 'Color', [1,1,1]);
    hold off;
    set(gca, 'Color', 'black');
    if( argasmovie )
        framenum = framenum+1;
        moviearray(:, framenum)=getframe(gcf, pos);
    end
end

%pause(pause_length);
%for(i=1:size(ex,1))
%    persex = ex{i, exnum};
%    hsv_val = i/size(ex,1);
%    col = hsv2rgb( [hsv_val , 0.8 , 0.8] );
%    for(j = 1:size(persex,1));
%        a=persex{j}(ch, :);
%        plot((1:size(a,2))*1000/size(a,2), a/mean(a), 'Color', col );
%    set(gca, 'Color', 'black');
%    hold on;
%    end
%end

if( argasmovie )
    movie2avi(moviearray, moviefile, 'fps', 1, 'colormap', hsv);
end

```

dtw_transform.m

```

function [T,p,q,D,sc] = dtw_transform(A , B)
%transforms A to look like B

M = dtw_cost(A,B);
C = [1 1 1.0;0 1 1.0;1 0 1.0];%default dpfast C

[p,q,D,sc] = dtw.dpfast(M,C);

T = zeros(size(B));

last_value = 0;
for i = 1:size(T,2)
    ind = p(find(q == i));
    if( isempty(ind) )
        T(:,i) = last_value;
    else
        last_value = mean( A(:,ind) , 2 );
        T(:,i) = last_value;
    end
end

```



```
end  
end
```

dtw_squashex.m

```
function ex = dtw_squashex( exercises , exnum )
```

```
if( nargin < 2 || isempty(exnum) )  
    exnum = 1;  
end
```

```
ex = exercises(:,exnum);
```

```
while(size(ex,1) ~= 1)  
    ex{1} = [ex{1};ex{2}];  
    ex(2) = [];  
end
```

```
ex = ex{1};
```

dtw_fitex.m

```
function ex_s = dtw_fitex(exercises,exnum,fitnum)  
%resamples to 1000 in addition to everything else
```

```
ex = dtw_squashex(exercises,exnum);
```

```
for( i = 1:size(ex,1) )  
    ex_s{i,1} = resample_arr(ex{i},1000,length(ex{i}));  
end
```

```
idx = 1:size(ex,1);  
idx(fitnum) = [];
```

```
for(i = idx)  
    ex_s{i} = dtw_transform(ex_s{i},ex_s{fitnum});  
end
```

ditw_cost.m

```
function D = ditw_cost(A,B)
```

```
n = size(A,1);  
if( n ~= size(B,1) )  
    disp('Error: A and B must have same number of rows');  
    return;  
end
```

```
rt_n = sqrt(n);
```

```
D = zeros( size(A,2) , size(B,2) );
```

```
%dA = dtw_derivative( A );
```

```

%dB = dtw_derivative( B );

dA = A;
dB = B;

for( i = 1:size(D,1) )
    for( j = 1:size(D,2) )
        diff = (dA(:,i) - dB(:,j));
        D(i,j) = norm(diff)/rt_n;
    end
end

end

function der = dtw_derivative( array2d )
% der = dtw_derivative( array2d ): returns the derivative of the
rows of
% data in array2d

der = zeros(size(array2d));
for i = 2:size(array2d,2)-1
    der(:,i) = (array2d(:,i)-array2d(:,i-1) + (array2d(:,i+1)-
array2d(:,i-1))/2)/2; % formula taken from Keogg and Pazzani
end
der(:,1) = der(:,2);
der(:,end) = der(:,end-1);

end

```

Glossary

Amplifier: A device that proportionately increases the voltage level of an entire signal, so that it is easier to process. Raw EMG signals are normally in the range of millivolts at most.

Biofeedback: Real-time visual or auditory cue indicating one's own muscle behavior.

Classification: Based on our expectation of the signal features for each muscle for each exercise, an algorithm will read the user's six muscle signals and determine which exercise the user was attempting to perform.

Electrode: An adhesive gel pad with a metal contact, placed on the body to conduct EMG signals.

Electromyography (EMG): The study of the electric potential (or EMG signal) generated when a muscle contracts.

Filtering: A process that causes certain input frequency ranges to result in very low gain, while other frequency ranges result in higher gain. This is used because we wish to only record EMG signals, which exist within a particular frequency range (~50 Hz – ~2 kHz). Signals outside this range will not be EMG, and thus will only cause noise if not filtered out.

Gain: A number that indicates amplification in terms of output voltage level divided by input voltage level for a circuit. This number differs based on signal frequency.

Gait: The muscular actions involved in a normal walking stride, including leg and hip muscles.

Hemiparesis: Impaired muscle control on only one side of the body.

Rehabilitation: Carefully directed, well-focused, repetitive exercises that train healthy neuromuscular pathways to replace damaged pathways.

Signal feature: Characteristics of the EMG voltage waveform such as amplitude, frequency, and timing.

Stroke: Blood clotting in the brain, leading to damaged brain tissue, and therefore impaired brain function. After a stroke, survivors must retrain the brain to redirect neural pathways to assume the functionality of previously destroyed pathways.

Bibliography

- AB, I. P. I. (2008). Products: Mindball Game Retrieved May 5, 2009, from <http://www.mindball.se/product.html>
- Abbink, V. D. B., & Glas, V. D. (1998). Detection of onset and termination of muscle activity in surface electromyograms. *Journal of Oral Rehabilitation*, 25(5), 365-369.
- Aiello, E., Gates, D. H., Patrilli, B. L., Cairns, K. D., Meister, M., Clancy, E. A., et al. (2005). Visual EMG biofeedback to improve ankle function in hemiparetic gait (pp. 7703-7706). Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the.
- Armiger, R. S., & Vogelstein, R. J. (2008). *Air-Guitar Hero: A real-time video game interface for training and evaluation of dexterous upper-extremity neuroprosthetic control algorithms*. Paper presented at the Biomedical Circuits and Systems Conference, 2008. IEEE., Baltimore, MD.
- Artemiadis, P. K., & Kyriakopoulos, K. J. (2008). Assessment of muscle fatigue using a probabilistic framework for an EMG-based robot control scenario.
- Association, A. H. (2009). Heart Disease and Stroke Statistics - 2009 Update. Dallas, Texas: American Heart Association.
- Basmajian, J. V. (1979). *Muscles alive, their functions revealed by electromyography* (4th ed.). Baltimore: Williams & Wilkins.
- Boian, R., Sharma, A., Han, C., Merians, A., Burdea, G., Adamovich, S., et al. (2002). *Virtual reality-based post-stroke hand rehabilitation*. Paper presented at the Medicine Meets Virtual Reality Conference, Newport Beach, CA.
- Brincks, J., Nielsen, J., & Kock-Jensen, C. (2009). Gait training for persons with stroke (GTS): Hammel Neurorehabilitation and Research Center, University of Aarhus, Denmark.
- Crosbie, J. H., Lennon, S., Basford, J. R., & McDonough, S. M. (2007). Virtual reality in stroke rehabilitation: Still more virtual than real. *Disability and Rehabilitation*, 29(14), 1139 – 1146.
- Daly, J. J., & R., W. J. (2008). Brain-computer interfaces in neurological rehabilitation. *Lancet Neurology*, 7, 1032-1043.
- De Michele, G., Sello, S., Carboncini, M. C., Rossi, B., & Strambi, S. (2003). Cross-correlation time-frequency analysis for multiple EMG signals in Parkinson's disease: A wavelet approach. *Medical Engineering and Physics*, 25(5), 361-369.
- Deutsch, J. E. (2001). Post-Stroke Rehabilitation with the Rutgers Ankle System: A Case Study. *Presence: Teleoperations and Virtual Environments*, 10(4), 416-430.
- Dipietro, L., Ferraro, M., Palazzolo, J. J., H.I., K., Volpe, B. T., & Hogan, N. (2005). Customized interactive robotic treatment for stroke: EMG-triggered therapy. *IEEE Trans Neural Syst Rehabil Eng*, 13(3), 325-334.
- Eisenhower, W., & McMichael, S. (2005). *Analog control with surface electromyography*. Electrical and Computer Engineering. University of Maryland, College Park.

- Ellis, D. (2003). Dynamic Time Warp (DTW) in Matlab, 2010, from <http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/>
- Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405), 165-175.
- Fukuda, O., Tsuji, T., Kaneko, M., & Otsuka, A. (2003). A human-assisting manipulator teleoperated by EMG signals and arm motions (Vol. 19, pp. 210 - 222). *Robotics and Automation, IEEE Transactions on*.
- Glanz, M., Klawansky, S., Stason, W., Berkey, C., Shah, N., Phan, H., et al. (1995). Biofeedback therapy in poststroke rehabilitation: A meta-analysis of the randomized controlled trials. *Archives of Physical Medicine and Rehabilitation*, 76(6), 508-515.
- Goldie, P. A., Matyas, T. A., & Evans, O. M. (1996). Deficit and change in gait velocity during rehabilitation after stroke. *Archives of Physical Medicine and Rehabilitation*, 77(10), 1074-1082.
- Güler, N. F., & Koçer, S. (2008). Classification of EMG signals using PCA and FFT. *Journal of Medical Systems*, 29(3), 241-250.
- Hassoun, M. H., Wang, C., & Spitzer, A. R. (1994). NNERVE: Neural network extraction of repetitive vectors for electromyography II, performance analysis. *IEEE Transactions on Biomedical Engineering*, 41(11), 1053-1061.
- Hermens, H. J., Freriks, B., Merletti, R., Rau, G., Disselhorst-Klug, C., Stegeman, D. F., et al. (2006). SENIAM (Surface ElectroMyography for the Non-Invasive Assessment of Muscles) Project, 2008, from <http://seniam.org/images/SEMGlocations/>
- Hornby, T. G., Campbell, D. D., Kahn, J. H., Demott, T., Moore, J. L., & Roth, H. R. (2008). Enhanced gait-related improvements after therapist versus robotic-assisted locomotor training in subjects with chronic stroke: A randomized controlled study. *Stroke*, 39(6), 1786-1792.
- Huang, H., Ingalls, T., Olson, L., Ganley, K., Rikakis, T., & He, J. (2005). *Interactive Multimodal Biofeedback for Task-Oriented Neural Rehabilitation*. Paper presented at the IEEE Engineering in Medicine and Biology Society, 2005. 27th Annual International Conference of the.
- Huang, H., Wolf, S. L., & He, J. (2006). Recent developments of biofeedback for neuromotor rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 3(1), 11.
- Hughes, E., & Bell, A. (2007). A Wireless Surface Electromyography System. *IEEE Southeast Conference Proceedings*, 253.
- Inman, V. T., Ralston, H. J., Saunders, J. B., Feinstein, B., & Wright, E. W. (1952). Relation of human electromyogram to muscular tension. *Electroencephalography and Clinical Neurophysiology*, 2, 187-194.
- Johnson, M., Ramachandran, B., Paranjape, R., & Kosasih, J. (2006). Feasibility study of TheraDrive: A low-cost game-based environment for the delivery of upper arm stroke therapy. *Proceedings of the 28th IEEE EMBS Annual International Conference*.
- Johnson, M. J. (2006). Recent trends in robot-assisted therapy environments to improve real-life functional performance after stroke. *Journal of NeuroEngineering and Rehabilitation*, 3(29).

- Jovanov, E., Milenkovic, A., Otto, C., & de Groen, P. C. (2005). A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *NeuroEngineering and Rehabilitation*, 2(6).
- Kuruganti, U., Hudgins, B., & Scott, R. N. (1995). Two-channel enhancement of a multifunction control system. *IEEE Transactions on Biomedical Engineering*, 42(1), 109-111.
- Lamb, R. (1992). *Anatomic and physiologic basis for surface electromyography* (Vol. 6): U.S. Department of Health and Human Services, National Institute for Occupational Safety and Health.
- Lord, S. E., McPherson, K., McNaughton, H. K., Rochester, L., & Weatherall, M. (2005). Community ambulation after stroke: How important and obtainable is it and what measures appear predictive? *Archives of Physical Medicine and Rehabilitation*, 85(2), 234-239.
- Lord, S. E., & Rochester, L. (2005). Measurement of community ambulation after stroke: Current status and future developments. *Stroke*, 36(7), 1457-1461.
- Mathews, S. A., Judge, J., & Ortega, C. (2007). *A myoelectric interface for video games*, *Biomedical Engineering: Recent Developments*. Paper presented at the 23rd Southern Biomedical Engineering Conference, Washington, D.C.
- Mauritz, K. H. (2004). *Gait Training in Hemiparetic Stroke Patients*. Paper presented at the Europa Medicophysica.
- Mayo, N. E., Wood-Dauphinee, S., S., A., & al., e. (1999). Disablement following stroke. *Disability and Rehabilitation*, 21(5-6), 258-268.
- Michael, K. M., Allen, J. K., & Macko, R. F. (2005). Reduced ambulatory activity after stroke: The role of balance, gait, and cardiovascular fitness. *Archives of Physical Medicine and Rehabilitation*, 86(8), 1552-1556.
- Moreland, J., Thomson, M., & Fuoco, A. (1998). Electromyographic biofeedback to improve lower extremity function after stroke: A meta-analysis. *Archives of Physical Medicine and Rehabilitation*, 79(2), 134-140.
- NeuroSky. (2009). NeuroSky Products Retrieved May 5, 2009, from <http://company.neurosky.com/products/>
- Nishida, Y., Kawakami, G., & Mizoguchi, H. (2006). *Everyday grasping behavior measurement with wearable electromyography*. Paper presented at the Sensors, 2006. 5th IEEE Conference on, Daegu.
- Noraxon, I. (2010). Zero-Wire - True Wireless EMG Retrieved 4 April 2010
- Pelland, L., & McKinley, P. (2004). A pattern recognition technique to characterize the differential modulation of co-activating muscles at the performer/environment interface. *Journal of Electromyography and Kinesiology*, 14, 539-554.
- Pierce, D. K. (2004). *Independent Study Project Report ENEE499*. Electrical and Computer Engineering. University of Maryland, College Park.
- Raez, M. B. I., Hussain, S., & Mohd-Yasin, F. (2006). Techniques of EMG signal analysis: Detection, processing, classification and applications. *Biological Procedures Online*, 8, 11-35.
- Rizzo, A., & Kim, G. (2006). A SWOT analysis of the field of virtual reality rehabilitation and therapy. *Presence: Teleoperators and Virtual Environments*, 14(2), 119-146.

- Shenoy, P., Miller, K. J., Crawford, B., & Rao, R. N. (2008). Online electromyographic control of a robotic prosthesis (Vol. 55, pp. 1128-1135). *IEEE Trans Biomed Eng.*
- Shima, K., Bu, N., Okamoto, M., & Tsuji, T. (2005). *A universal interface for video game machines using biological signals*. Paper presented at the International Conference on Entertainment Computing 2005.
- Soderberg, G. (1992). *Recording techniques*: U.S. Department of Health and Human Services, National Institute for Occupational Safety and Health.
- Song, R., Tong, K. Y., Hu, X., & Li, L. (2008). Assistive control system using continuous myoelectric signal in robot-aided arm training for patients after stroke. (Vol. 16, pp. 371-379). *IEEE Trans Neural Syst Rehabil Eng.*
- Subasi, A., Alkan, A., Koklukaya, E., & Kiyimik, M. K. (2005). Wavelet neural network classification of EEG signals by using AR model with MLE preprocessing. *Neural Networks, 18*, 985-997.
- Systems, E. (2008). Emotiv EPOC Retrieved May 5, 2009, from <http://emotiv.com/epoc.html>
- Taylor, R. J., & Berry, E. (1998). The use of a computer game to rehabilitate sensorimotor functional deficits following a subarachnoid hemorrhage. *Neuropsychological Rehabilitation, 8*(2), 113-122.
- Technology, O. (2009). OCZ Peripherals: nia - Neural Impulse Acuator Retrieved May 5, 2009, from http://www.ocztechnology.com/products/ocz_peripherals/nia-neural_impulse_actuator
- Tong, R. K. Y., Ng, M. F. W., Li, L. S. W., & So, E. F. M. (2006). Gait training of patients after stroke using an electromechanical gait trainer combined with simultaneous functional electrical stimulation: Physical Therapy.
- Ueno, A., Uchikawa, Y., & Noshiro, M. (2007). *A capacitive sensor system for measuring laplacian electromyogram through cloth: A pilot study*. Paper presented at the 29th Annual International Conference of the IEEE EMBS, Cite Internationale, Lyon, France.
- Veneman, J. F., Kruidhof, R., Hekman, E. E. G., Ekkelenkamp, R., Van Asseldonk, E. H. F., & van der Kooij, H. (2007). Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation (Vol. 15, pp. 379 – 386). *Neural Systems and Rehabilitation Engineering, IEEE Transactions on.*
- Wade, D. T., Wood, V. A., Heller, A., Maggs, J., & Langton Hewer, R. (1987). Walking after stroke: Measurement and recovery over the first 3 months. *Scandinavian Journal of Rehabilitation Medicine, 19*(1), 25-30.
- Whyte Griffith Associates, L. (2008). B&L Engineering Retrieved April 4, 2010, from <http://www.bleng.com/electrod.htm>