Grasp Detection with Force Myography for Upper-Extremity Stroke Rehabilitation Applications

by

Gautam Pradip Sadarangani

B.A.Sc. (Hons), Simon Fraser University, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Engineering Science
Faculty of Applied Sciences

© Gautam Pradip Sadarangani 2017

SIMON FRASER UNIVERSITY

Spring 2017

Approval

Name:	**Gautam Pradip Sadarangani**
Degree:	**Master of Applied Science**
Title:	***Grasp Detection with Force Myography for Upper-extremity Stroke Rehabilitation Applications***
Examining Committee:	**Chair:** Dr. Craig Scratchley Senior Lecturer

Dr. Carlo Menon
Senior Supervisor
Professor

_____

Dr. Bozena Kaminska
Supervisor
Professor

_____

Dr. Ryan D'Arcy
Internal Examiner
Professor

_____

Date Defended/Approved: November 25, 2016

Ethics Statement

SFU

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

    a.   human research ethics approval from the Simon Fraser University Office of Research Ethics

or

    b.   advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

    c.   as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Grasp training is a key aspect of stroke rehabilitation. This thesis explores the suitability of Force Myography (FMG) classification for the two-class problem of grasping, regardless of grasp-type, versus a lack of grasping, for rehabilitation applications.

FMG-based grasp detection in individuals with stroke was assessed with a protocol comprising of three grasp-and-move tasks, requiring a single grasp-type. Accuracy was lower, and required more training data for individuals with stroke when compared to healthy volunteers. Despite this, accuracy was above 90% in individuals with stroke.

FMG-based grasp detection was further evaluated using a second protocol comprising of multiple grasp-types and upper-extremity movements, with healthy volunteers. The utility of classifying temporal features of the FMG signal was also assessed using Area under the Receiver Operator Curve (AUC). Accuracy with the raw FMG signal was 88.8%. At certain window configurations, model-based temporal features yielded up to a 6.1% relative increase in AUC over the raw FMG signal.

Keywords: Activity Monitoring; Force Myography; Functional Activity Tracking; Stroke Rehabilitation; Grasp Detection; Wearable Sensors;

To my family and friends, for your tireless support in this, and all, endeavours.

The best is yet to be.

Acknowledgements

Table of Contents

List of Tables

List of Figures

List of Acronyms

ADL	Activities of Daily Living
AMG	Acoustic Myography
AUC	Area under the Receiver Operating Curve
FMG	Force Myography
FSR	Force Sensitive Resistor
IMU	Inertial Measurement Unit
IR	Infra-red
LDA	Linear Discriminant Analysis
LF	Linear (two degree of freedom) Fit
MAV	Mean Absolute Value
PC	Personal Computer
PF	Parabolic (three degree of freedom) Fit
RBF	Radial Basis Function
RMS	Root Mean Squared
ROC	Receiver Operating Curve
SEMG	Surface Electromyography
SVM	Support Vector Machine

Glossary

Activities of Daily Living	Activities required for daily living, such as dressing and feeding.
Area under the Receiver Operating Curve	A measure of classification performance from 0 to 1.
Classification Accuracy	Ratio of correctly predicted samples to total number of samples.
False-negative	Datum for which a grasp detection classifier incorrectly predicts no grasping when the participant was grasping an object.
False-positive	Datum for which a grasp detection classifier incorrectly predicts grasping when the participant was not grasping an object.
Grasp-and-move	Activity involving a participant grasping an object in his/her hand, and moving the object to another location.
Grasping	The participant holding an object in his/her hand, regardless of the grasp-type or object.
Lack of grasping	The participant not holding any object in his/her hand. Interchangeable with no grasping.
Learned non-use	Phenomenon in which a paretic limb further deteriorates due to a lack of use.
Movements-without-grasping	The participant engaging in upper-extremity movements without an object grasped in his/her hand.
Musculo-tendinous	Muscle and tendon tissue.
No grasping	The participant not holding any object in his/her hand. Interchangeable with lack of grasping.
Paretic limb	A limb with partial paralysis.
Raw FMG Signal	Unprocessed FMG data, in the form of instantaneous FMG samples, from multiple sensors.
Receiver Operating Curve	A plot that depicts the classification performance of a classifier. Plots the true-positive rate against the false-positive rate.
Signal-to-noise-ratio	Ratio of the portion of the signal containing information to the portion of the signal containing noise, or interference.
Testing phase	The phase during which the performance of a classifier model is tested using unseen data.

Testing-set	Set of unseen data, used to test classifier performance during the testing phase.
Training phase	The phase during which a classifier model is generated using data in the training-set.
Training-set	Set of input data which has been labelled with the appropriate class label. The training-set is used to train a classifier during the training phase.
True-negative	Datum for which a grasp detection classifier correctly predicts a lack of grasping when the participant was not grasping.
True-positive	Datum for which a grasp detection classifier correctly predicts grasping when the participant was grasping an object.
Two-class grasp detection problem	The classification problem of detecting grasping, regardless of the grasp type used, versus a lack of grasping.

Chapter 1.   Introduction

1.1. Chapter Overview

This chapter begins by outlining the motivation for this thesis. Subsequently, the specific objectives of this thesis are defined. Finally, a layout of the remaining chapters of this thesis is provided.

1.2. Motivation

Stroke is one of the most prevalent causes of long-term adult disability [1, 2]. Individuals with stroke often experience impairments in the upper-extremity, including a reduction in fine motor control [1] and a reduced ability to grasp [3]. These impairments contribute to difficulties in performing Activities of Daily Living (ADL), such as dressing, feeding, and home management [1], which in turn leads to lack of use and further deterioration of the paretic limb [4, 5]. It has been shown that repetitive task-specific training helps restore upper-extremity motor control in stoke survivors [6]. However, there is increasing evidence that it is necessary to practice hundreds, if not thousands, of grasp and release repetitions to optimize hand motor recovery and rehabilitate fine motor control after stroke [7, 8]. While a variety of rehabilitation techniques exist, in general, stroke rehabilitation is a labor and cost intensive activity, carried out in a one-to-one fashion, with a therapist individually guiding the patient through the large number of repetitions of functional task practice required for motor recovery [1].

In Canada, the standard of care involves assessing all hospitalized, acute stroke patients, prior to discharge, to identify the appropriate type of rehabilitation intervention for the patient. It is estimated that 10% of patients are discharged to long-term care facilities [9]. 19% of patients are discharged to rehabilitation facilities within the health-care system [9]. Two thirds of patients admitted to rehabilitation facilities continue to require in-person,

out-patient, rehabilitation services, or at-home care, subsequent to discharge from the rehabilitation facility [10]. 58% of patients are discharged directly to the home environment, and rely on out-patient, in-person rehabilitation services, and other self-guided rehabilitation programs [9].

The demand for stroke rehabilitation services is expected to rise. A rapidly ageing population is expected to lead to an increase in the incidence of stroke and other age-related ailments [11]. Concurrently, a reduction in the mortality associated with stroke implies that more individuals will need post-stroke rehabilitation [12]. This increasing demand for stroke rehabilitation services, and the labor and cost intensive nature of rehabilitation, has resulted in an increase in research on methods to improve the efficiency of the rehabilitation process [13]. One such area of research is the creation of wearable sensing systems which can assist therapists and patients in monitoring the large amount of functional task practice necessary for motor recovery [7, 8]. A device that is capable of monitoring and encouraging functional use of the affected limb, in stroke survivors, could potentially enhance and optimize the rehabilitation process at multiple stages.

In the clinic, such a device could keep track of the number of times a patient uses their paretic limb functionally in a given period of time, and provide feedback to the therapist and patient, without requiring the therapist to constantly monitor the progress of the patient. This in turn could increase the efficiency of the in-person rehabilitation process. The proposed device could also act as a monitoring tool for the clinician and a motivational tool for the patient as part of a home-based rehabilitation program, allowing for therapy to continue beyond traditional face-to-face therapy sessions. Enabling continued rehabilitation in the home environment presents a number of advantages. Firstly, it has been shown that improvement in motor control and functional ability are more likely to occur if in-person, clinician-supervised, therapy is supplemented with task practice at home [6, 14]. Furthermore, the ability to achieve clinician-supervised rehabilitation in the home environment, would optimize the therapists' utilization of in-person clinical time, and ensure that the increasing demands for rehabilitation services [12, 15] continue to be met. Finally, the proposed device could also encourage individuals with stroke to continue to use their affected limbs as part of ADL. It has been shown that a portion of residual impairment after a stroke can be attributed to lack of use, or "learned non-use" [5], of the

paretic limb. This lack of use can lead to further deterioration in functional ability of the paretic limb [4, 5]. Furthermore, it has been noted that the gains made with patients during rehabilitation are often lost post-rehabilitation due to factors such as lack of stimulation, encouragement, and an unsuitable physical environment [1]. Hence, continued encouragement of functional use of the paretic limb, as part of daily living, could potentially preserve functional gains reaped during rehabilitation and avoid the learned non-use phenomenon [5].

A number of sensing technologies have shown promising suitability for the creation of devices that detect, track, and encourage functional activity for stroke rehabilitation applications. The potential ability for sensing technologies to monitor grasping is of especial relevance in rehabilitation applications, given the central role of grasping in enabling functional ability [3], and the emphasis placed on grasp training in upper-extremity stroke rehabilitation [16, 17]. Two main classes of technologies exist: environment-based sensors and body-worn sensors. Environment-based sensors such as machine vision and Infra-Red (IR) sensing technologies have been shown to be able to effectively track human motion [18]. Suggested applications of this technology range from gaming to localized tracking of stroke rehabilitation activity [19]. Despite the affirmative results, such devices are excluded from the scope of this thesis. These devices require retrofitting of the operating environment, and hence, are not easily deployable in the multiple settings in which activity tracking is required in rehabilitation applications. Several body-worn sensing technologies have shown similarly promising results in activity tracking, and have the benefit of being comparatively portable, and easily deployable in a variety of environments. Body-worn technologies include: (i) inertial measurement devices [20, 21, 22], such as accelerometers, and Inertial Measurement Units (IMUs), (ii) proximity sensing devices using magnetic sensing principles [23], (iii) glove-based devices embedded with bend sensors [24], and (iv) Myography methodologies, such as Surface Electromyography (SEMG) [25], Acoustic Myography (AMG) [26], and Force Myography (FMG) [27]. The benefits and detriments of each of the aforementioned sensing technologies are discussed in detail in Chapter 2.

Amongst these technologies, the use of FMG for activity tracking has been the subject of increasing research in recent years. FMG involves monitoring the force, or

3

pressure, at the surface of the limb, as a means to characterize the state of the underlying musculo-tendinous complex [27]. The benefits of FMG include: (i) simple signal acquisition and processing requirements [28], (ii) ease of set-up and lack of need to place sensors at specific anatomical locations [29], (iii) low power consumption requirements [29], and (iv) low cost [30]. With the use of appropriate signal processing and machine learning techniques, FMG has been shown to be able to: (i) predict grip strength [27], (ii) predict single finger forces [30], (iii) detect a variety of grasp-types [28, 29], (iv) detect grasp-and-move actions [31], and (v) detect upper-extremity postures of the elbow, forearm, and wrist [32, 33]. Amongst these advancements, the potential use of FMG for detecting grasping [28, 29, 31] is especially relevant for rehabilitation applications. Given that FMG has been shown to be capable of detecting and distinguishing between a variety of grasp-types in other applications, it is hypothesized that FMG technology could be applied to monitoring and encouraging grasping, in individuals with stroke, for rehabilitation applications.

Despite promising advances in FMG research, several research questions remain in order to establish the suitability of using FMG for tracking and encouraging grasping in upper-extremity stroke rehabilitation applications. Firstly, the feasibility of acquiring and classifying the FMG signal from individuals with stroke with upper-extremity impairments, who might ultimately benefit from FMG technology, has yet to be established. Secondly, a majority of contemporary FMG grasp detection research has focused on distinguishing between various types of grasps (i.e. multi-class problem) in the absence of significant upper-extremity movements [28, 29, 34]. The ability to distinguish between a discrete set of grasp-types is a necessary feature in human machine interface applications [28, 29, 34]. However, the presence of grasping, regardless of the grasp type involved, can be indicative of functional use of a limb, which is a key goal in stroke rehabilitation [15], in order to avoid learned non-use [5]. In stroke rehabilitation applications, FMG-based devices will have to detect and encourage grasping instead of a lack of grasping, regardless of which of the wide variety of grasp-types necessary to complete ADL [35], was used. The accuracy of FMG classification in the "two-class" grasp detection problem of grasping, regardless of grasp-type, versus no grasping (i.e. no object in hand), has yet to be established. Additionally, in order to be deployed in uncontrolled environments outside of the rehabilitation clinic, FMG-based grasp detection will have to be robust in the

4

presence of the upper-extremity movements that are required to complete ADL. Hence, further research is required to establish the accuracy achievable with FMG for the above defined two-class grasp detection problem, in the presence of upper-extremity movements that would be expected as part of ADL. Thirdly, contemporary FMG research has focused on the classification of the raw FMG signal, in the form of instantaneous FMG samples from multiple channels (i.e. sensors), in order to detect the functional state of the user's limb [29, 31, 32, 36]. The potential utility of feature extraction techniques has yet to be investigated. Given the temporal nature of grasping [37], it is hypothesized that the identification of a suitable set of temporal features of the FMG signal may improve grasp detection accuracy.

1.3. Thesis Objectives

Based on these considerations, which are discussed in greater detail in Chapter 2, this thesis seeks to explore the suitability of using FMG for grasp detection in upper-extremity stroke rehabilitation applications. This thesis consists of three main objectives.

Objective 1 is to perform a preliminary investigation on the accuracy of FMG-based grasp detection in individuals with stroke, who have upper-extremity impairments, in a controlled environment.

Objective 2 is to perform a preliminary investigation on the accuracy of FMG classification for the two-class grasp detection problem, using a variety of grasp-types and upper-extremity movements, with healthy volunteers, in a controlled environment.

Objective 3 is to perform a preliminary investigation on the utility of classifying temporal features of the FMG signal for the two-class grasp detection problem, using a variety of grasp-types and upper-extremity movements, with healthy volunteers, in a controlled environment.

It is noteworthy that these three objectives were not fulfilled concurrently. Instead, the completion of **Objective 1** guided refinement of the work that was conducted to

achieve **Objective 2** and **Objective 3.** The extent of the experimental protocol associated with **Objective 1** was limited to that which was achievable without triggering muscular and mental fatigue in participants with stroke. The investigations associated with **Objective 2** and **Objective 3** were conducted with healthy volunteers, to allow for a more extensive experimental protocol than would have been possible with participants with stroke.

1.4. Thesis Layout

The subsequent chapters of this thesis are organized as follows. In Chapter 2, a summary of the relevant literature surrounding upper-extremity activity tracking solutions, and FMG is provided. Chapter 3 and Chapter 4 describe the two studies that were run as part of this thesis. In Chapter 3, the design, execution, and results of the first study, which was conducted to meet **Objective 1** of this thesis is described. In this study, the feasibility of acquiring and classifying the FMG signal from individuals with stroke, with upper-extremity impairments, for the purpose of grasp detection, was explored. In section 3.5, the key observations and findings of the study, which contributed to the definition of the second study, described in Chapter 4, are discussed.

In Chapter 4, the design, execution, and results of the second study, which was conducted to meet **Objective 2** and **Objective 3** of this thesis is reported. In this study, the accuracy achievable with FMG classification for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, as would be expected in daily living, was investigated. Finally, in Chapter 5 a summary of the findings of this thesis is provided.

6

Chapter 2.   Literature Review

2.1. Chapter Overview

In this chapter, a brief summary of the literature surrounding body-worn sensing technology for activity tracking is provided, with emphasis on the key themes of FMG research. In section 2.2, an overview of various upper-extremity sensing technologies that are alternatives to FMG is provided. Additionally, their benefits and detriments with regards to functional activity tracking in stroke rehabilitation applications is discussed. In section 2.3, a brief overview of FMG, and its principle of operation is provided. Subsequently, in section 2.4, the methodologies and results of contemporary FMG research are summarized and potential areas of additional research are identified, a subset of which correspond to the objectives identified for this thesis.

2.2. Upper-extremity Activity Tracking Technology

2.2.1.   Inertial Sensing

The use of inertial sensing devices, such as accelerometers, for activity tracking is well-established. Several commercially available activity tracking systems use inertial sensing technology, including medical grade devices such as the ActiGraph WGT3X-BT, which is depicted in Figure 2-1 [38]. Inertial sensing devices have been shown to capture inertial data associated with arm and hand movement. Advantages of such devices include their ease of use, low cost, and relative insensitivity to placement [19]. On the other hand, relating the metrics of these devices to functional activity is an active area of research [20, 21, 22, 39]. With appropriate signal processing, accelerometer-based devices have been shown to provide ratio-metric counts that relate to the amount of upper-extremity movement achieved by individuals with stroke [20, 21, 22], and the amount of hand-use in older adults [39]. However, such devices are unable to assess the functional state of the user's hand and determine when the user has successfully grasped an object. Hence, inertial sensing devices are unable to distinguish functional use of the

upper-extremity from non-functional movements, such as shaking or swinging of the arm [22].

Figure 2-1: Commercially Available Inertial Sensing Based Activity Tracker. Reproduced from [38]

Given the emphasis placed on functional activity [40] and grasp training [16, 17] in rehabilitation, it is desirable for sensing devices to be able to distinguish between functional and non-functional movements, by detecting grasping of the hand. Accordingly, the remaining technologies discussed in this thesis provide means to detect the functional state of the hand. All of the subsequent sensing technologies discussed can be used in conjunction with inertial sensing, in order to provide data representative of both hand use, and arm and hand movement.

2.2.2.   Magnetic Sensing

The use of magnetic sensing systems for detecting wrist and hand movements has also been proposed [23, 41]. Such devices use a wrist-worn magnetometer to detect relative movement of a magnetic object donned on a different body part, such as a finger. Friedman et al. propose one such device comprising of a wrist-worn unit, containing a magnetometer and signal processing unit, that senses the relative movement of a magnetic beacon, which is embedded in a ring worn on the finger [23]. The device, which is depicted in Figure 2-2, has been shown to be capable of detecting fine upper-extremity movements including wrist flexion/extension, wrist deviation, and finger flexion/extension [23, 41]. The inclusion of an accelerometer within the device provides additional metrics relating to arm and hand movement [23]. Advantages of such devices include their relatively low cost, portability, and sensitivity to fine upper-extremity movements [19]. However, magnetic sensing systems are susceptible to magnetic interference from other ferromagnetic materials, such as household electronics, and hence, may have difficulty accurately monitoring functional activity in the home environment [19, 23].

Figure 2-2: Magnetic Sensing System. Reproduced from [41]

2.2.3. Sensing Gloves

Several gloves embedded with sensors for tracking hand postures have been developed and successfully evaluated [19, 24]. These devices apply an encoder-based approach, by adding bend sensors that measure bending of localized areas of the glove as a measure of the degree of flexion or extension of each finger [19].Figure 2-3 depicts a commercially available sensing glove [42]. Glove-based devices have been shown to be able to detect fine finger movements and gestures of the hand [24]. However, for individuals with a limited range of motion and muscle spasticity, such devices can be challenging to don [19]. Additionally, the wearing of gloves results in a reduction in palmar sensation and may hamper the completion of fine-motor tasks [19].

Figure 2-3: Commercially Available Sensing Glove. Reproduced from [42]

2.2.4.   SEMG

SEMG is a widely used technology for collecting information related to the contraction or relaxation of muscles by measuring the electrical activity of the muscles at the surface of the skin proximal to the muscle [25]. The use of SEMG in conjunction with machine learning techniques for classifying and detecting upper-extremity postures is well established [25]. Furthermore, SEMG has been successfully used to detect hand activity in individuals with stroke [43, 44, 45]. Lee et al. used SEMG data from individuals with stroke to identify six different hand postures [44]. In recent years, commercially available SEMG armbands have been introduced, including the Myo^{TM} arm band from Thalmic Labs, which is depicted in   Despite these promising results, challenges remain in the use of SEMG for unobtrusive monitoring of hand motion. SEMG requires complex signal acquisition and amplification hardware, and a high signal sampling rate, with correspondingly large computational and power consumption requirements [46]. Additionally, given that SEMG is measuring the electrical activity of muscles, its signal-to-noise ratio is sensitive to the positioning of electrodes on the surface of the skin, as well as skin impedance [47, 48], which may be affected by hair, sweat, and skin creams.

11

Figure 2-4: Commercially Available SEMG Arm Band. Reproduced from [49]

2.2.5.   AMG

AMG is a complimentary sensing technology to SEMG, that involves transducing the acoustic signatures of the vibrations of muscles as they contract [26, 50]. AMG senses the acoustic analog of the electrical activity measured by SEMG as muscles contract [26, 50]. The AMG signal can be transduced using several sensing elements, including microphones and accelerometers [50]. AMG has been shown to be sensitive to the force of contraction [26, 50], and capable of detecting the functional state of the hand using pattern recognition and classification techniques [50]. However, the performance of AMG is sensitive to sensor placement, and sensor pressure (i.e. adherence) on the surface of the skin [26, 50]. AMG has also been shown to require similarly complex signal processing methodologies, associated with large computational requirements, as is used in SEMG classification [50].

2.3. FMG: Overview and Principle of Operation

FMG is an alternative sensing technology that involves monitoring the force, or pressure, at the surface of the limb, as a means to characterize the state of the underlying musculo-tendinous complex [27]. FMG is a relatively new sensing technology, with the majority of FMG research taking place from the 1990s to date [23, 27, 29, 30, 32, 33, 34, 36, 43, 51]. The underlying principle behind FMG is that of monitoring the expansion or contraction of muscles by sensing the force exerted by the same muscles onto force sensors that are placed normal to the surface of the limb. Figure 2-5 provides an illustrative example of an FMG sensing system. A force sensing band is constructed using multiple force sensors, embedded into a flexible and conformable back layer. The band is wrapped around a user's working limb, such that the sensors (S1 to S6 in Figure 2-5) are normal to, and in close contact with, the surface of the working limb. The muscles within the limb expand and contract as the limb is used functionally. This expansion and contraction results in the manifestation of forces on the surface of sensors, which are quantified, and are termed the FMG signal [27]. The FMG signal has been shown to relate to grip strength [27] and single-finger forces [30]. Furthermore, with the use of machine learning techniques it has been shown that the patterns of forces in the FMG signal, can be used to predict the functional state of the limb, including: (i) detecting hand gestures [29], (ii) upper-extremity postures [32], and (iii) repetitions of grasp-and-move tasks [31] in healthy participants.
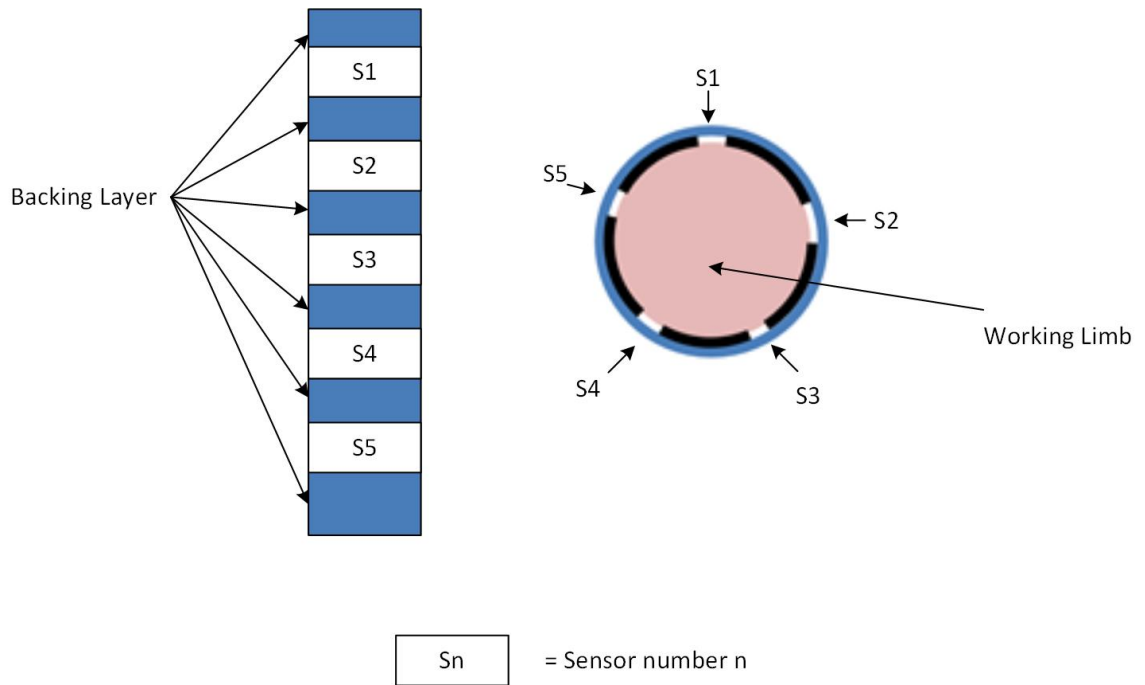
Figure 2-5: FMG Principle of Operation

FMG presents several advantages when compared to alternative upper-extremity sensing technologies discussing in section 2.2. In contrast to SEMG, the FMG signal can be extracted using off-the-shelf force sensing elements and does not require complex signal processing circuitry or high sampling rates, which would increase power consumption [29]. Additionally, FMG does not require sensors to be placed at specific anatomical points on the body [32], as is required in SEMG and AMG [47]. Given these promising results, the use of FMG with machine learning techniques has been investigated for several applications, including: (i) human machine interfaces for controlling robots and industrial machinery to enhance productivity and safety [52, 53], (ii) human machine interfaces for controlling prosthetic devices [51, 54], (iii) human machine interfaces to act as an input device to computers and other digital electronics [29], and (iv) tracking specific rehabilitation exercises for tele-rehabilitation applications [32, 36]. However, the use of FMG to detect and encourage grasping, regardless of grasp-type, in order to encourage functional use of the paretic limb, in individuals with stroke, has not been investigated. Given the abilities of FMG for detecting a variety of grasp-types, it is hypothesized that FMG technology could be capable of monitoring grasping versus a lack of grasping,

14

regardless of grasp-type, in individuals with stroke, in order to achieve functional activity monitoring for rehabilitation applications.

Despite the promising advancements in FMG research, additional research questions remain in order to establish the feasibility of using FMG for grasp detection in rehabilitation applications. In section 2.4 key research issues relating to FMG-based grasp detection are discussed. Additionally, research issues that motivate the objectives of this thesis are identified.

2.4. FMG: Key Research Elements and Progress to Date

2.4.1. Classifier Training Paradigms

The goal of classification is to predict, with accuracy, the class (i.e. category) that a set of data belongs to [55]. A classifier model is a mathematical function that translates a set of input signal data into output data, which represents a predicted class amongst the range of available classes [55]. A linear classification scheme employs a linear function to separate the data into classes [55]. A non-linear classification scheme seeks a non-linear boundary with which to separate data into classes [55]. In a supervised learning paradigm, the classifier model is generated with a set of signal data that have been labelled with the appropriate class labels; this set of data is termed training data, or the training-set [55, 56]. During the learning, or training phase, the parameters of the mathematical function(s) that constitute the classifier model are iteratively adapted in order to meet the goals of a minimization routine, such that, the error between the predicted output of the current version of the model, and the target output, specified in the class label, is minimized for each datum [55, 56]. In doing so, the model is progressively adapted to reflect the relationship between the signal data and the class labels provided [55, 56]. Subsequently, unseen data, known as testing data, or the testing-set, are classified with this previously generated model [55, 56]. A key goal in classification is the ability to generalize the relationship that exists between the signal data and the classes, in order to be able to predict the class associated with unseen signal data, which the model was not trained with.

To date, most FMG research that has focused on upper-extremity tracking has focused on supervised learning [29, 31, 32, 36, 57]. While supervised training has yielded high accuracy [29, 31, 32, 36, 57], the necessity for an operator, or a secondary sensor, to label the FMG data before training, makes the training process a critical step in establishing the feasibility of FMG-based functional activity tracking systems. The size of the training data required to obtain acceptable accuracy when classifying unseen data, and the time associated with collecting the necessary training data will impact the feasibility of deploying FMG systems in stroke rehabilitation applications. For FMG to be feasible in busy clinical settings, the initial set-up of the device would need to be minimal. Hence, an investigation into the effect of training-set size in FMG classification, and methods to reduce training-set size requirements, are key research questions.

Semi-supervised and unsupervised training are two alternative training paradigms. In unsupervised training schemes, the classifier model is generated with no labels provided, with the goal of building a model that identifies clusters within the data [55]. In semi-supervised learning, a classifier model is constructed using a data-set in which some samples have been partially labelled with the appropriate class label, while some samples remain unlabelled, as is the case in unsupervised learning [58]. The use of unlabelled data, in conjunction with labeled data has been shown to increase the accuracy of classification over supervised learning, in cases where the labeled training-set is small [58]. The use of unsupervised and semi-supervised training paradigms to reduce the amount of training data and effort required for device set-up is an untapped area of FMG research, and may be a key step in realizing the potential of this technology.

2.4.2. Acquisition Methodologies

Several methods of acquiring FMG have been identified, including pneumatic sensors, force sensors, and strain sensors. Abboudi et al. fabricated pneumatic pressure-sensing systems by creating vacuums in polyethylene bags [51]. The pressure exerted on the surface of the bags was measured via pneumatically connected pressure sensors. The system demonstrated the ability to adequately acquire the FMG signal [51]. However, the sensitivity of pneumatic transducers to temperature and atmospheric pressure conditions, and the need to calibrate each sensor [51] makes them less than

ideal for wide-spread deployment in activity tracking systems. Furthermore, the thickness of these sensors would be prohibitive to the creation of low-profile, discreet, devices that can be unobtrusively worn as part of daily use.

Force Sensitive Resistors (FSRs) are an alternative method of acquiring the FMG signal. FSRs are fabricated using a polymer thick film that displays a resistance that is inversely proportional to the force that is applied to its surface [59]. The reliability, sensitivity, and low thickness of these sensors have led to them being prolifically used in FMG research [28, 29, 31, 32, 33, 57]. FSRs have been embedded in bands to be donned around a user's limb, in order to capture the FMG signal by sensing surface forces on the working limb normal to the surface of the FSRs [28, 29, 31, 32, 33, 57]. However, the tightness or slackness of the band around the user's limb needs to be controlled in order to maintain sufficient adherence of the FSRs to the surface of the limb. A lack of sufficient tightness will reduce the effectiveness of the FSRs at detecting forces as the muscle expands or contracts, and will adversely impact classification performance [57]. For this reason, FSR bands are fabricated using flexible materials, such as foam, in order to enhance user comfort while achieving sufficient band tightness and adherence to the curvature of the working limb [31, 32].

Strain sensors are an alternative technology to FSRs that can be embedded into stretchable bands to be worn around the user's working limb. These sensors quantify the deformation of the band as a means to detect the volumetric expansion and contraction of the working limb [60, 61]. Strain sensors can be fabricated to be stretchable [60], and hence, may be able to conform to the expansion and contraction of a user's limb while maintaining adherence with the surface of the limb, which may lead to enhanced user comfort. A comprehensive comparison of the performance and reliability of these sensors in relation to FSR bands has yet to be conducted. However, the ability for strain sensors to be fabricated in a manner that allows them to stretch and conform to the expansion of a user's limb, without losing sensitivity to the FMG signal, makes them a promising alternative for FMG sensing, that merits further study.

2.4.3.   Sensor Placement

Two main sensing locations have been proposed for FMG-based tracking of the upper-extremity: (i) below the elbow, at the forearm [28, 30, 31, 32, 33, 57], and (ii) the wrist [29, 62]. Using a band around the forearm to capture the FMG signal is well established [28, 30, 31, 32, 33, 57]. The FMG signal has also been shown to be adequately extracted at the wrist [29, 62]. Dementyev et al. demonstrated that a wrist donned FMG sensing system can achieve accuracies in excess of 80% when classifying six different grasp-types [29]. Given the comparative intuitiveness of donning a band on the wrist, as opposed to the forearm, characterization of benefits and detriments of acquiring the FMG signal at the wrist may prove pivotal in moving the field of FMG-based activity tracking forward.

The use of high-density arrays of FSRs to form an FMG band that covers the entire forearm, from below the elbow to the wrist has also been proposed [28]. Li et al. demonstrated classification accuracy of 99% at classifying seventeen different grasp-types using a high-density FMG array, with 32 sensors, donned on the forearm [28]. However, the use of a band that covers the entire forearm may lead to a more obtrusive user experience.

2.4.4.   FMG in Individuals with Stroke

Despite promising results obtained with healthy individuals [28, 29, 32, 34, 36, 57], little work has been done to establish the feasibility of FMG for activity monitoring in populations with upper-extremity impairments, who might ultimately benefit from this technology. Several characteristics associated with individuals with stroke may impact the feasibility of FMG in this population. Individuals with stroke have reduced muscular strength [63] and greater muscle spasticity [64], which may affect the magnitude and quality of the FMG signal. Additionally, movements completed with the stroke affected limb have a reduced range of motion, are less smooth, and involve variations in speed and acceleration when compared to the less affected limb [65]. These movement features of the paretic limb could introduce challenges when detecting grasps in the presence of movement in the three-dimensional workspace, and may also increase the amount of training data required to accurately classify grasps. Thus, there is a pressing need to

evaluate the feasibility of acquiring and classifying the FMG signal from individuals with stroke, with upper-extremity impairments, for the purpose of grasp detection. **Objective 1** of this thesis seeks to offer insights into this question.

2.4.5.　Number of Classes and Presence of Upper-extremity Movements

Several studies have established the accuracy of FMG classification at detecting and distinguishing between various types of grasps (i.e multi-class problem), in the absence of significant upper-extremity movements [28, 29, 34]. Li et al. demonstrated classification accuracy of 99% at classifying seventeen different grasp-types using a high-density FMG array, with 32 sensors, donned on the forearm [28]. Dementyev at al. demonstrated classification accuracy in excess of 80% when classifying six different grasp-types using a FMG band, with 15 sensors, donned on the wrist [29]. The results of these studies give credence to the concept of using FMG sensing for detecting grasping. However, to be useful for activity tracking in take-home rehabilitation settings, an activity monitor would need to be able to detect, and distinguish between grasping versus a lack of grasping, regardless of which of the wide variety of grasp-types necessary for ADL [35], was used.

Additionally, the grasp detection accuracy achievable with FMG in the presence of upper-extremity movements, as would be expected when completing ADL, has yet to be established. The FMG signal has been shown to be sensitive to postures of the hand, wrist, forearm, and elbow [32, 33]. While the detection of joint positions may be useful in some applications, the sensitivity of FMG to joint positions could adversely impact grasp detection, as joint positions will vary when the user is grasping and moving an object. Variations in joint position and movement trajectories, which would be expected as part of daily living, could confound the grasp classification scheme and reduce accuracy. Further research is required to ensure the robustness of FMG-based grasp detection, such that, it may be eventually used for grasp detection in uncontrolled environments.

Hence, the accuracy achievable with FMG classification for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, as

would be expected in daily living, is a crucial unanswered research question. **Objective 2** of this thesis seeks to offer insights into this question.

2.4.6.   Feature Extraction Techniques

Contemporary FMG research has focused on the classification of the raw FMG signal, in the form of instantaneous FMG samples from multiple channels (i.e. sensors), in order to detect the grasping of an object [28, 29, 31, 32, 36]. However, the grasping of an object is not a discrete, instantaneous action. Instead, it is a multi-stage process [37]. The force generated during grasping will increase as the participant first makes contact with the object, and then generates additional force in order to be able to grip the object securely, and lift the object against gravity [37]. Subsequently, the releasing process involves a reduction in force, and the opening of the hand into a relaxed posture [37]. Given that grasping involves a force profile that changes over the duration of the grasp, it is hypothesized that the temporal sequence of FMG values adjacent in time may provide descriptive information on the current grasping state of the hand (i.e. that grasp detection with FMG can be represented as a time series problem).

Traditional machine learning methods, such as the Support Vector Machine (SVM) and Neural Network classifiers, use an instance-based method for classification [66]. In such methods, the instantaneous classifier output is dependent on the instantaneous inputs to the classifier; historical or future inputs do not affect the instantaneous classifier output. The use of temporal feature extraction has been shown to increase accuracy in applications that involve classifying data that have temporal or sequential dependence, such as the classification of SEMG signals [67]. The temporal feature extraction step involves calculating parameters (i.e. features) that represent the time series within a certain window of data. The calculated features are then provided to the classifier as instantaneous inputs [66].

Given that grasping involves a force profile that changes over the duration of the grasp [37], it is hypothesized that temporal features of the FMG signal may provide increased grasp detection performance, as is the case in other time series problems [66]. The utility of using temporal feature extraction in FMG classification problems is, hence,

another potential research question that may substantially increase the applicability of FMG-based sensing to rehabilitation activity monitoring. **Objective 3** of this thesis seeks to provide preliminary insight into the utility of classifying features of the raw FMG signal for the two-class grasp detection problem, in the presence of a variety of grasp-types and upper-extremity movements.

2.5. Summary of Literature

A variety of upper-extremity sensing technologies have demonstrated suitability for upper-extremity activity tracking for rehabilitation applications. Amongst these potential technologies, FMG has shown promising applicability to upper-extremity activity tracking for rehabilitation applications due to the following attributes: (i) simple signal acquisition and processing requirements [28], (ii) ease of set-up and lack of need to place sensors over specific anatomical locations [32], (iii) low power consumption requirements [29], and (iv) low cost [30].

A significant amount of progress has been made in the field of FMG-based grasp detection. Several different signal acquisition methods have been developed, evaluated, and continue to be refined [51, 60, 61]. The ability to use FMG to detect and distinguish between various grasp-types has been established [28, 29, 34]. A partial list of unanswered questions includes: (i) the possibility of using semi-supervised, and unsupervised training for FMG-based grasp detection, (ii) the benefits and detriments of using strain sensors for acquiring the FMG signal, (iii) the benefits and detriments of acquiring FMG at the wrist versus the forearm, (iv) the ability to use FMG for grasp detection in individuals with stroke with upper-extremity impairments, (v) the accuracy of FMG-based grasp detection for the two-class grasp detection problem, in the presence of upper-extremity movements, and (vi) the utility of classifying temporal features of the FMG signal for the two-class grasp detection problem, in the presence of upper-extremity movements.

Addressing, all these questions is beyond the scope of thesis. However, the objectives of this thesis will offer preliminary insights into: (i) the ability to use FMG for grasp detection in individuals with stroke with mild-to-moderate upper-extremity

impairments, in a controlled environment (ii) the accuracy of FMG-based grasp detection for the two-class grasp detection problem, in the presence of upper-extremity movements, in a controlled environment, and (iii) the utility of classifying temporal features of the FMG signal for the two-class grasp detection problem, in the presence of upper-extremity movements, in a controlled environment.

Chapter 3.   Feasibility of Using FMG for Grasp Detection in Individuals with Stroke

3.1. Chapter Overview

This chapter describes the design, execution, and results of a study intended to explore the feasibility of acquiring and classifying the FMG signal from individuals with stroke, with upper-extremity impairments, for the purpose of grasp detection. The work described in this chapter was intended to meet **Objective 1** of this thesis. In section 3.2, an overview of the study is provided. Section 3.3 and section 3.4 discuss the experimental methods and experimental results obtained. In section 3.5, the key observations and findings of the study, which contributed to the definition of the second study of this thesis, described in Chapter 4, are discussed.

3.2. Study Overview

The overall purpose of this study was to investigate the accuracy of FMG-based grasp detection in individuals with stroke with upper-extremity impairments, in a controlled environment. The experimental protocol comprised of multiple repetitions of grasp-and-move tasks. In this preliminary study, the experimental protocol was limited to the use of a single object that required a single grasp-type, and a limited task workspace in order to avoid mental and muscular fatigue in individuals with stroke. Accuracy was investigated by: (i) establishing and comparing the off-line classification accuracy of FMG-based grasp detection for participants with stroke, and healthy participants, using linear and non-linear classifiers, and (ii) determining and comparing the amount of training data necessary to achieve commensurate grasp classification accuracy, in participants with stroke, and healthy participants.

3.3. Experimental Methods

3.3.1. Participants

Experimental data were collected from eight participants with stroke, and eight healthy participants, and stored to file for off-line analysis. Inclusion criteria for the participants with stroke were: (i) cerebrovascular accident confirmed by MRI or CT scan, (ii) chronic stroke (> 12 months post-stroke), (iii) mild to moderate impairment of the paretic hand (Chedoke Hand Score > 5) [25], and (iv) poorer performance on the Box & Blocks test [26] for the paretic hand compared to the non-paretic hand, indicating residual impairment. Healthy participants were a sample of convenience of right-dominant adults, with no history of injuries or impairments to their upper limbs.

3.3.2. Data Collection Devices

Six commercial off-the-shelf FSRs manufactured by Interlink Electronics [59] were embedded into a 40 cm flexible foam band, 4 cm apart from each other, to form a force-sensing band to collect FMG data from participants (Figure 3-1). The band was positioned around the participant's working forearm, approximately 8 cm from the olecranon, in order to detect FMG signals associated with the functional state of the participant's working hand (Figure 3-2). The band was fastened with Velcro® so that the band was tight, but comfortable for each participant. None of the participants reported discomfort due to band tightness. To emulate the intended use condition, only the distance from the olecranon was specified, and the band could be fastened on any part of the forearm musculature (e.g., anterior or posterior). The device was donned on the paretic limb for participants with stroke, and donned on the dominant limb for healthy participants.
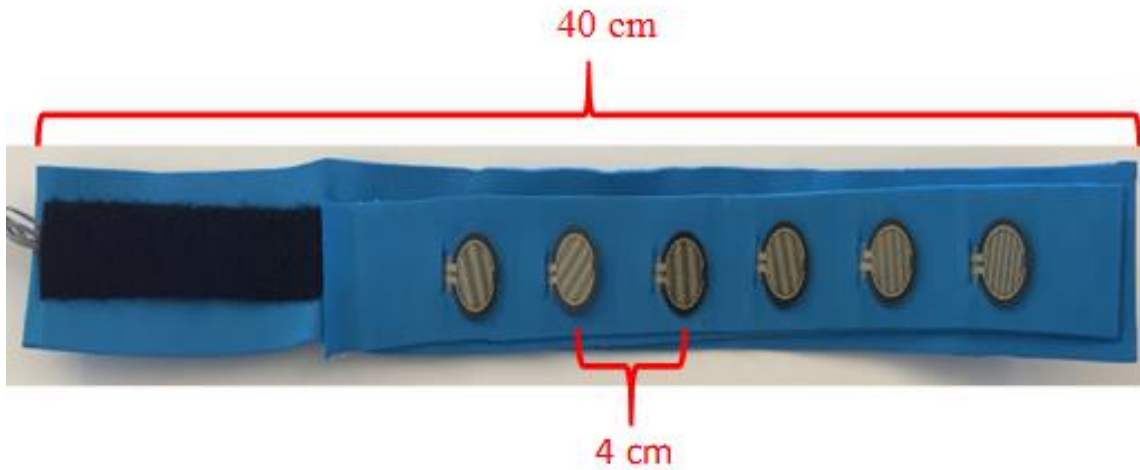
Figure 3-1: Force Sensing Band

Figure 3-2: Force Sensing Band on Forearm

The FMG signal was quantified from the FSRs using a voltage divider circuit. The voltage across the sense resistors was sampled between the 0-5 V range at 20 Hz, using a National Instruments DAQ Device [68] interfaced with custom LabVIEW™ software, running on a Personal Computer (PC). The resulting data acquisition system is schematically depicted in Figure 3-3.
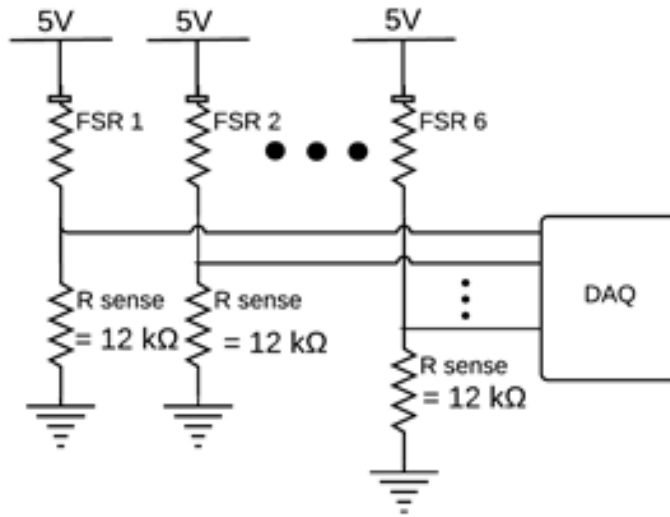
Figure 3-3: Data Acquisition System

A coffee cup, weighing 530 g, was selected as the object for all grasp-and-move tasks in the experimental protocol. The cup was instrumented with a validation sensor to detect when the cup had been grasped. The validation sensor comprised of a pressure-sensitive conductive sheet attached to conductive copper tape. The pressure-sensitive conductive sheet, which demonstrates a reduction in resistance as the applied force increases, was wrapped around the inside of the cup handle. Copper tape was attached to the two ends of the sheet and interfaced to the data acquisition system. The pressure-sensitive conductive sheet and copper tape on the cup handle were wrapped in insulation tape to ensure no other source of conductance or resistance was electrically connected to the copper tape. Data from the validation sensor was used to label FMG data as corresponding to a grasp or no grasp, which was used to train the grasp detection classifier and calculate grasp detection accuracy. The experimenter observed the study to ensure that no objects, other than the instrumented mug, were grasped by the participant throughout the duration of the experimental protocol. The validation sensor, with the insulation tape peeled away to reveal the copper tape and pressure-sensitive conductive sheet is depicted in Figure 3-4.
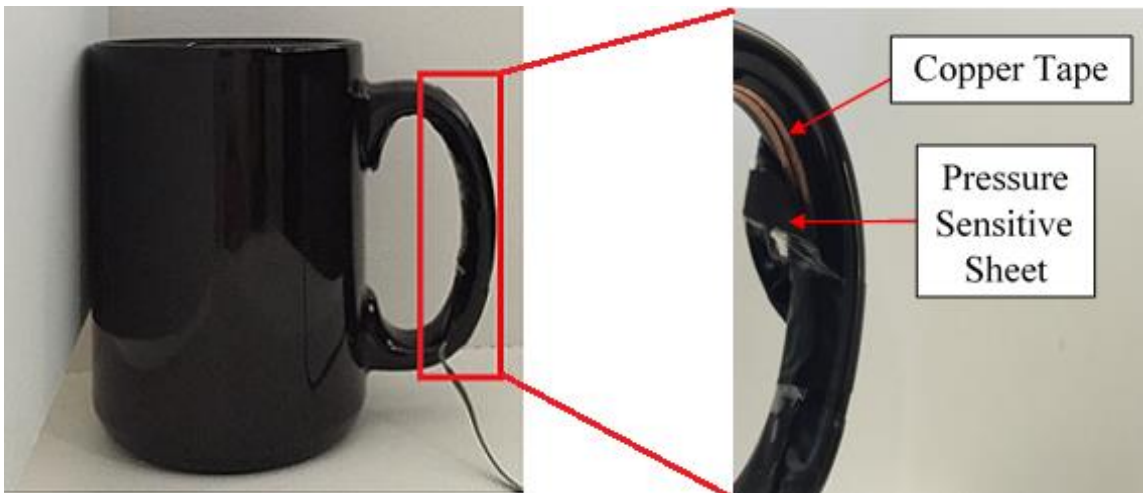
Figure 3-4: Internal View of Validation Sensor on Cup Handle

3.3.3.   Experimental Protocol

The experimental protocol was defined via iterative refinement with the objective of collecting the largest possible amount of FMG data corresponding to grasping in the presence of upper-extremity movements, without triggering muscular or mental fatigue in individuals with stroke. The duration of the protocol was limited to 60 minutes, including the execution of functional assessments.

Participants were seated comfortably in an armless chair with their working hand resting on the table in front of them. They were asked to reach for, grasp (i.e. pick up) and move a cup in each of the three different planes-of-movement (i.e. superior-inferior, medial-lateral, and ventral-dorsal), resulting in three different tasks (Figure 3-5). For each task, participants were asked to reach for the cup at the start position, grasp the cup, lift it off the table, move it to the end position, place the cup down on the table at the end position, and completely release the cup. Subsequently, they were asked to reach for the cup at the end position, grasp the cup, lift it off the table, move it to the start position, place the cup down on the table at the start position, and completely release the cup, before returning their hands to rest on the table in front of them. In Task 1 the end position was to the left of the start position, such that, participants moved the cup laterally (i.e. to the side). In Task 2 the end position was the top of a custom shelf, above the start position, such that, participants lifted the cup superiorly (i.e. overhead). In Task 3 the end position

27

was ahead of the start position, such that, participants moved the cup ventrally (i.e. forward). The target distance for each task was 90% of the maximum active range of motion of each participant. The participants were asked to repeat each task, by first moving the cup from the start position to the end position, and subsequently moving the cup from the end position back to the start position, ten times at a comfortable pace. This resulted in twenty grasp-and-move repetitions. In moving the cup from the end position back to the start positions, participants also moved the cup in the medial (Task 1), inferior (Task 2), and dorsal (Task 3) directions. The described protocol allowed for the collection of FMG data corresponding to grasping with the presence of upper-extremity movements, in the three planes-of-movement that encompass the majority of space within which we perform our daily activities.
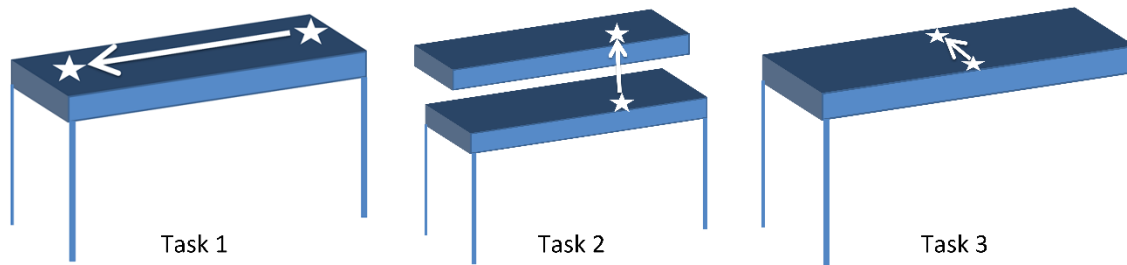
Figure 3-5: Task Start and End Positions. White arrow indicates direction of motion from start to end position.

3.3.4.   Data Analysis

Feasibility of FMG grasp detection among individuals with stroke was assessed by examining and comparing grasp classification accuracy for participants with stroke and healthy participants. As a secondary measure of feasibility, the amount of training data necessary for commensurate grasp classification accuracy, in participants with stroke and healthy participants, was investigated. Additionally, classification accuracy with linear and non-linear classification schemes was also established.

Data were temporally divided into repetitions to allow for evaluation of the effect of training-set size on classification accuracy. Data from the start of a grasp-and-move action, to the start of the next grasp-and-move action, were assigned to one repetition, based on data from the validation sensor. Each repetition comprised of FMG data (from

all six FSRs) corresponding to the grasp-and-move action required to move the cup, and the subsequent hand and arm activity corresponding to reaching and other preparatory movement (i.e. no grasping). Data corresponding to the reaching and other preparatory movement, in-between grasp-and-move actions, were included in order to establish the classifiers' robustness to false-positives (i.e. incorrectly predicting a grasp when the participant had no object in hand). The division scheme is depicted in Figure 3-6. Points of time labeled as grasp (grasp-and-move action) by the validation sensor are shaded in grey, points of time labeled as absent of grasping (reaching and other preparatory movement) are not shaded.
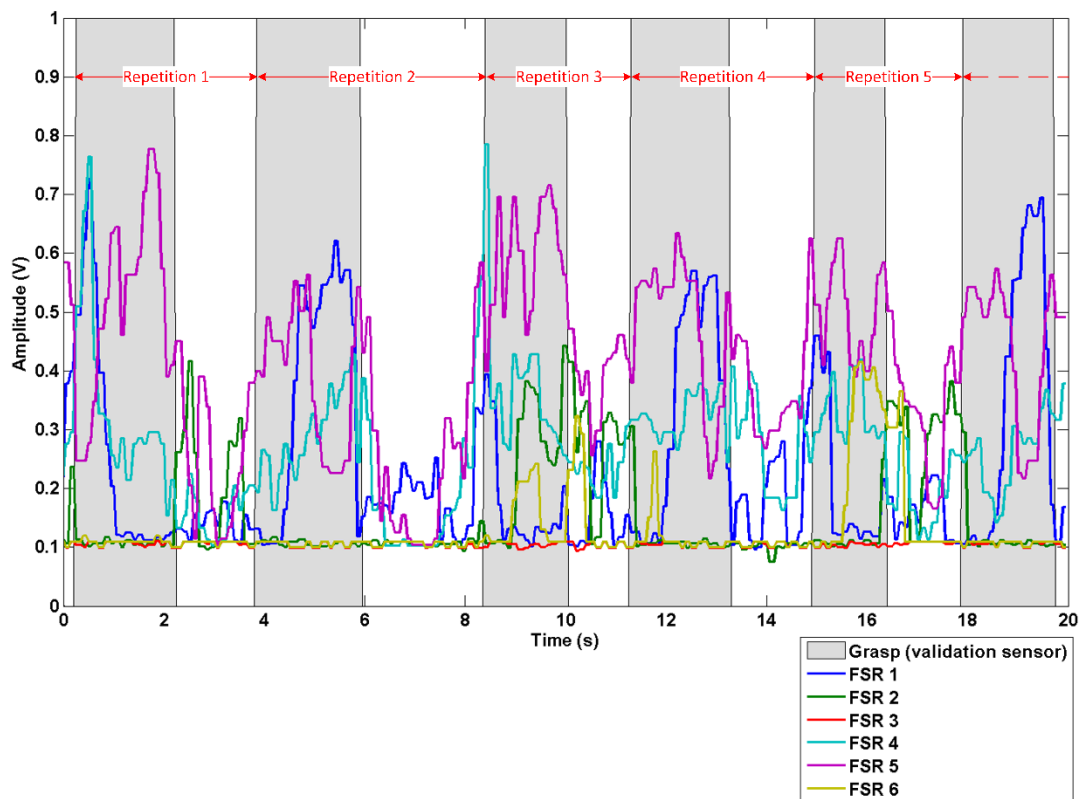
Figure 3-6: Repetition Division Scheme. Red arrows demark the division of data into repetitions based on the signal from the validation sensor (gray shading)

For the purpose of accuracy analysis, classification output was categorized into true-positive, true-negative, false-positive, and false-negative. A true-positive corresponds to the classifier correctly predicting grasping, when the participant was grasping the object,

as labeled by the validation sensor. A true-negative corresponds to the classifier correctly predicting a lack of grasping, when the participant was not grasping (during reaching and preparatory motion), as labeled by the validation sensor. A false-positive corresponds to the classifier incorrectly predicting grasping, when the participant was not grasping, as labeled by the validation sensor. A false-negative corresponds to the classifier incorrectly predicting no grasping, when the participant was grasping the object, as labeled by the validation sensor. Grasp detection accuracy was then computed as per (eq 3.1); where *TP* is the number of true-positives, *TN* is the number of true-negatives, *FP* is the number of false-positives and *FN* is the number of false-negatives.

$$accuracy\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{(eq 3.1)}$$

For example, a typical participant took approximately 4 seconds to complete a repetition (i.e. a grasp-and-move action, and the subsequent reaching and preparatory motion leading up to the next grasp-and-move action), resulting in 80 instantaneous 6-channel samples (sampled at 20 Hz). Of these 80 samples, approximately 60% (48) of the samples would correspond to points of time when the participant was grasping and moving the mug, as labeled by the validation sensor. The remaining 40% (32) of the samples would correspond to points of time when the participant was not grasping. During these points of time the participant could have been reaching for the mug, engaging in other preparatory motion, or returning to the neutral position. A 100% accuracy would indicate that the classifier was able to correctly identify the aforementioned 48 grasp samples as grasp, and the remaining 32 samples as absent of grasping.

Classification was carried out using a SVM classifier with a non-linear Radial Basis Function (RBF) [55] kernel given by expression (eq 3.2).

$$K(x_1, x_2)\ =\ e^{(-\gamma \|x_1 - x_2\|^2)} \qquad \text{(eq 3.2)}$$

Where γ is the symbol for gamma, a parameter used to control the fitting behavior of the SVM. The LIBSVM library was used off-line in the MATLAB® environment to evaluate the accuracy of the RBF-SVM, with default cost and gamma parameters [69]. In

order to evaluate the ease of separation of FMG data, the use of a Linear Discriminant Analysis (LDA) classifier was also evaluated using the MATLAB® Statistics and Machine Learning Toolbox [70]. Unlike the RBF-SVM that seeks a linear separator in the nonlinear feature space, the LDA seeks a linear decision boundary in the data space [55]. The ability to use simpler linear signal processing and classification methods would indicate that a FMG-based grasp detection system could potentially be more easily embedded into a compact and portable, low-power device, that is capable of running independently.

In order to establish grasp detection accuracy, each data set related to a task for a participant (20 repetitions) was divided into training and testing sets, with the 1st to 10th repetitions forming the training set and the remaining ten repetitions forming the testing set. The training set was used for generating a model for a task for each subject, and the testing set was used for model evaluation. Despite the fact that data was divided into repetitions to create testing and training sets, data were serially provided to the classifier on a unit-time basis (6 instantaneous FSR samples), regardless of the repetition they were part of, in order to emulate a real-time classification problem.

Differences in accuracy between healthy participant and participants with stroke, and the LDA and SVM classifiers were compared using t-tests. All analyses utilized an alpha of 0.05.

In order to examine the amount of training data necessary for accurate classification, the size of the training-set was varied from one repetition to ten repetitions. In all cases, accuracy was determined by classifying data within the fixed-sized testing-set, comprised of data associated with ten repetitions. The training-set size required to achieve 90% accuracy was established for participants with stroke and healthy participants, for both types of classifiers. The correlation coefficient between training-set size and classification accuracy was also derived for participants with stroke and healthy participants.

3.4. Experimental Results

3.4.1. Participants

The participants with stroke were a mean of 69 years (σ = 6 years) of age, and were a mean 9.5 years (σ = 7.2 years) post-stroke (i.e., chronic stroke). The Chedoke Arm Score was a mean of 6.75 (σ = 0.71), and the Chedoke Hand Score was a mean of 6.38 (σ = 0.75), both with a range of 5-7 for all participants with stroke. Paretic side performance was 62.1% (σ = 22.3%) of the non-paretic side performance for the Box and Blocks Test, indicating residual impairment. Table 3-1 lists the functional measures and characteristics associated with the participants with stroke. Healthy participants were a mean of 27 years (σ = 7 years) of age. All participants provided informed consent for participation in the study.

Table 3-1: Functional Measures and Characteristics Associated with Participants with Stroke

ID	Chedoke Arm	Chedoke Hand	Box & Blocks Paretic Side	Box & Blocks Non-paretic Side	Paretic Side	Age	Years post stroke
S1	7	7	39	50	R	64	5.5
S2	7	7	46	53	L	69	5.5
S3	7	7	48	64	R	62	4.5
S4	7	5	12	50	L	67	6.4
S5	7	6	11	28	R	78	14.3
S6	7	6	33	48	L	76	8.2
S7	7	7	48	62	R	74	6.1
S8	5	6	21	44	R	63	25.5

3.4.2. FMG Grasp Detection (Classification) Accuracy

Figure 3-7 shows the accuracies associated with grasp detection for both participants with stroke and healthy participants using the RBF-SVM and LDA for each task. Classification accuracy was lower in individuals with stroke when compared to healthy volunteers. Average grasp detection accuracy was 92.6% (σ = 3.20%) and 91.5% (σ = 3.20%) across all tasks with participants with stroke using the RBF-SVM and LDA

respectively. Average grasp detection accuracy was 96.1% (σ = 1.25%) and 94.1% (σ = 3.04%) across all tasks with healthy participants using the RBF-SVM and LDA respectively.
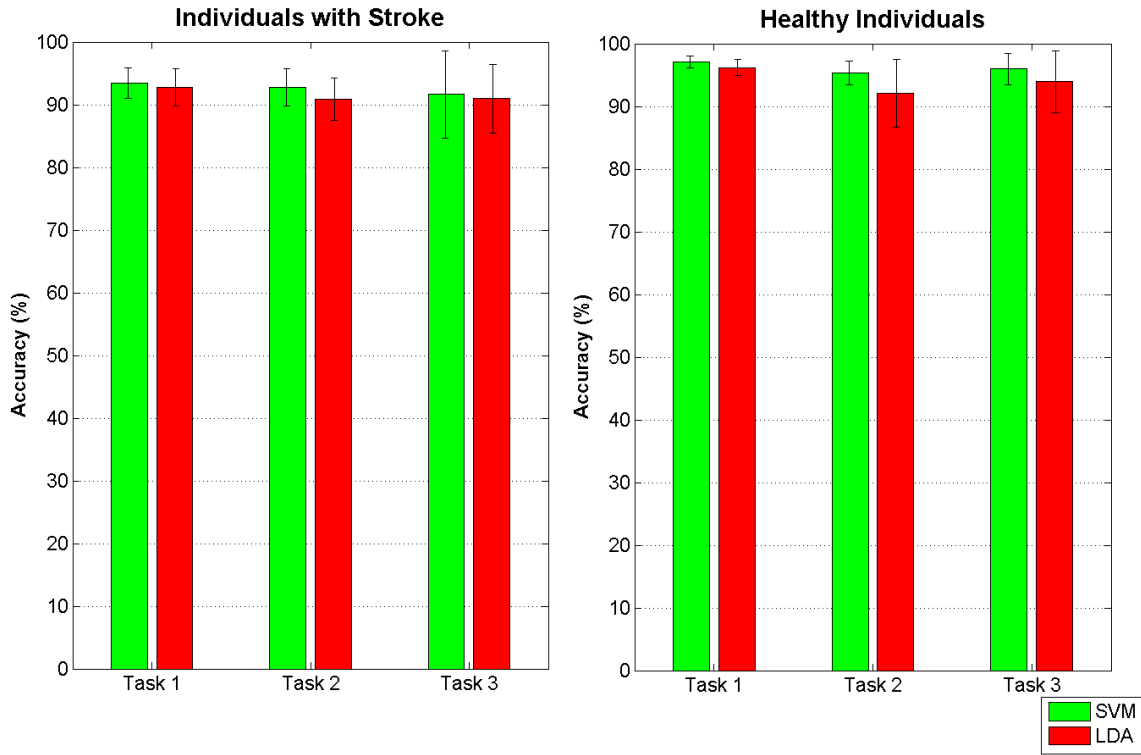
Figure 3-7: Classification Accuracy for Each Task

The lower average accuracies for participants with stroke when compared to healthy participants are statistically significant for the RBF-SVM (P = 0.010), and are not significant for the LDA (P = 0.071). Differences in accuracy between participants with stroke and healthy participants were larger for Task 2 and Task 3 when compared to Task 1, using both the RBF-SVM (Task 1 = 1.8%, Task 2 = 3.2%, Task 3 = 4.5%) and the LDA (Task 1 = 0.6%, Task 2 = 3.0%, Task 3 = 3.1%). The lower accuracies obtained when using the LDA are significant for both participants with stroke (P = 0.006) and healthy participants (P = 0.013).

Figure 3-8 depicts the average accuracies taken across all tasks for training-sets of differing sizes for participants with stroke and healthy participants. The accuracies associated with each task are depicted in Figure 3-9. The accuracy was dependent on the

size of the training-set provided. The correlation coefficients between the number of training samples and the accuracy for participants with stroke are 0.9012 (p< 0.001) using the RBF-SVM, and 0.8824 (p<0.001) using the LDA. The correlation coefficients between the number of training samples and the accuracy obtained for healthy participants are 0.9616 (p<0.001) using the RBF-SVM, and 0.6532 (P = 0.04) using the LDA. The use of four and six repetitions was necessary for achieving greater than 90% accuracy for participants with stroke using the RBF-SVM and LDA respectively. The use of one and two repetitions was sufficient for achieving greater than 90% accuracy for healthy participants using the RBF-SVM and LDA respectively.
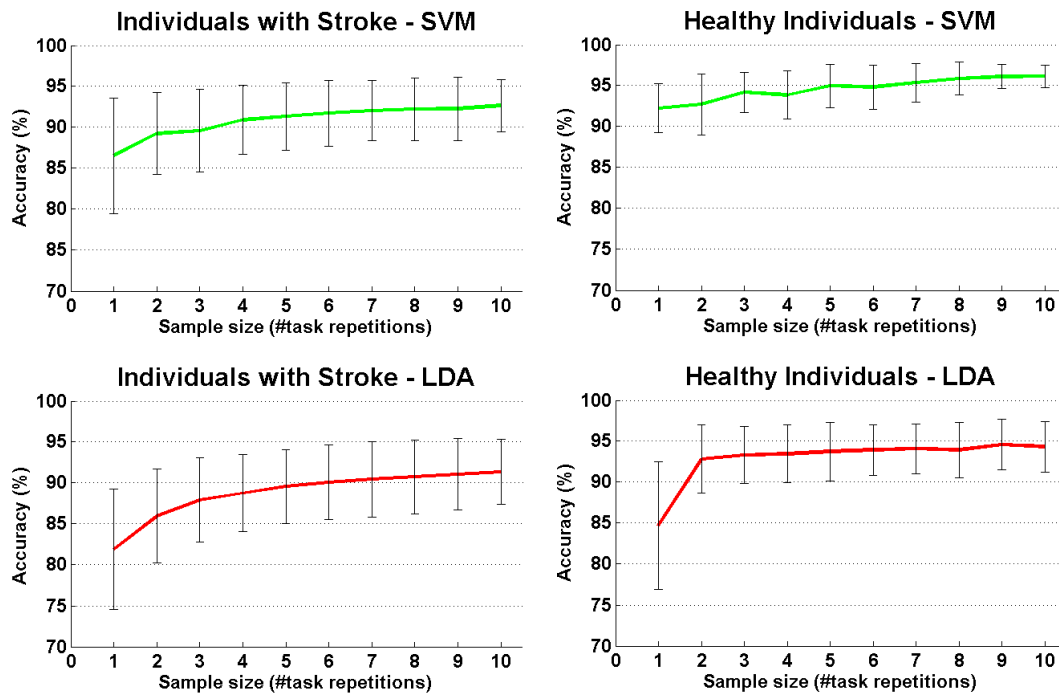
Figure 3-8: Average Classification Accuracy Versus Training-set Size
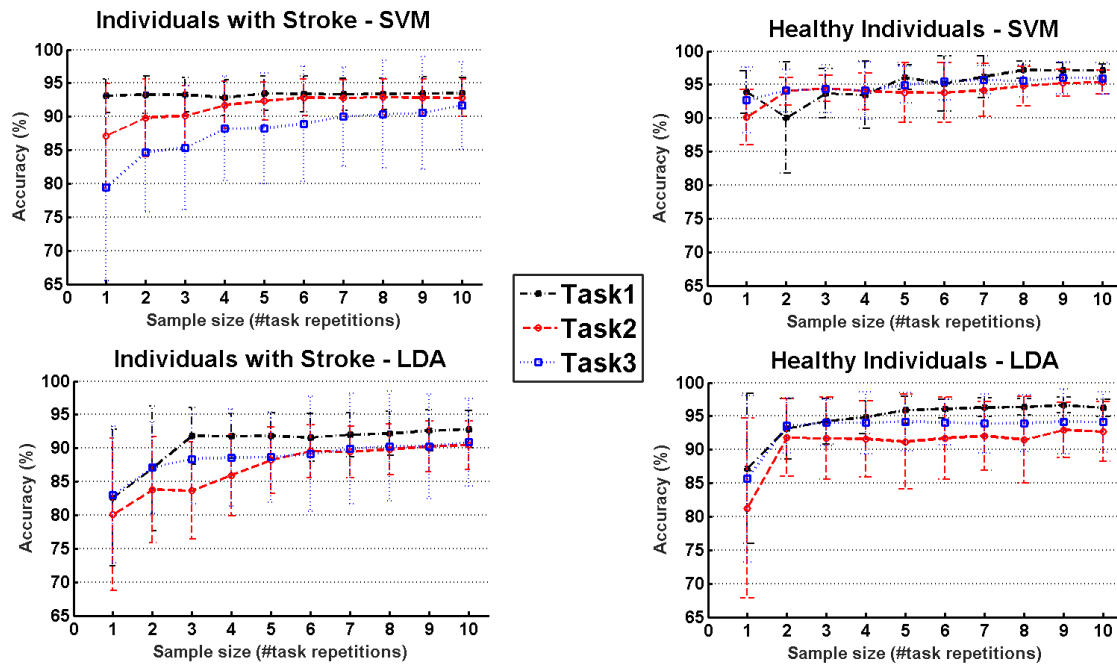
Figure 3-9: Classification Accuracy Versus Training-set Size for Each Task

3.4.3. Discussion of Results

The overall objective of this study was to investigate the accuracy of FMG-based grasp detection in individuals with stroke with upper-extremity impairments, in a controlled environment. FMG-based grasp detection accuracy was evaluated using an experimental protocol comprising of three grasp-and-move tasks. Grasp classification accuracy was established and compared for participants with stroke and healthy participants, using the RBF-SVM and LDA classifiers. The susceptibility of classification accuracy to the amount of training-data provided was also investigated.

Overall, FMG-based grasp detection demonstrated high accuracy of approximately 92% with participants with stroke. Grasp detection accuracy was lower for participants with stroke when compared to healthy participants, especially when using the RBF-SVM. A potential explanation for the lower grasp detection accuracy is the empirically observed variability in grasp-types, movement trajectories, and additional compensatory mechanisms demonstrated by participants with stroke. This variability may have resulted in a more challenging classification problem. Additionally, participants with stroke often

had difficulty in grasping and completely releasing the cup at the start and end positions, which potentially made classification of those data more challenging.

As a retrospective analysis step, the correlation between the classification accuracy obtained and the severity of impairment, in individuals with stroke, was established. The results of this analysis are summarized in Appendix A. No statistically significant correlation was established between the severity of impairment, as quantified by the Box and Blocks test, and grasp classification accuracy. This result indicates that variations in classification accuracy between participants with stroke are not likely due to differences in severity of impairment, and could possibly be attributed to the aforementioned empirically observed variations in task repetitions, which was more pronounced in individuals with stroke, when compared to healthy individuals.

It is noteworthy that classification accuracies were lower for moving upwards (Task 2), or forwards (Task 3), when compared to moving laterally (Task 1) for both participants with stroke and healthy participants. This suggests that FMG classification may be sensitive to movement direction, as would be expected given its sensitivity to upper-extremity postures. Additionally, the lower accuracies observed for participants with stroke when compared to healthy participants were larger for moving superiorly (Task 2), or ventrally (Task 3), when compared to moving laterally (Task 1). A potential explanation for this larger difference is that moving superiorly (Task 2), and ventrally (Task 3), have been shown to be especially challenging for individuals with stroke [65], which may have led to increased variability, and a more challenging classification problem. The use of a linear classification scheme did significantly reduce classifier performance for both participants with stroke and healthy participants. However, average grasp detection accuracy remained above 90% for both participants with stroke and healthy participants, suggesting that linear classification may be adequate for FMG-based grasp detection.

Grasp detection accuracy was significantly dependent on training-set size in all cases. However, a larger training-set size was required to achieve 90% accuracy with participants with stroke, when compared to healthy participants. It is possible that a larger training-set size was necessary for the classifier to generalize the variability in movement trajectories and grip patterns observed in participants with stroke. Despite the reduction in

36

accuracies, the use of a training-set that was 50% the size of the testing-set resulted in 91.4% (σ = 4.14%) accuracy (Figure 3-8) with participants with stroke, using the RBF-SVM. These promising results are preliminary indications of the feasibility of deploying FMG-based sensing systems for grasp detection in individuals with stroke in clinical settings.

3.5. Summary and Implications of Results

In this chapter, the preliminary feasibility of using FMG for detecting grasping in individuals with stroke was established by investigating the accuracy of FMG-based grasp detection with individuals with stroke, in a controlled environment. The FMG grasp detection classifier was able to achieve greater than 90% grasp detection accuracy, with a training-set that was 50% of the size of the testing-set. Despite these promising results, additional questions with regards to FMG-based grasp-detection remain. Some of the observations and results from this study, summarized in this section, were used to refine the approach taken to meet **Objective 2** and **Objective 3** of this thesis.

The accuracy obtainable when using FMG for grasp detection in the presence of upper-extremity movements remains an open question. The protocol executed in this study did mandate the collection of data corresponding to grasping while moving in a controlled workspace. However, the set of movements mandated in this protocol were limited to those which could be accommodated by participants with stroke without the onset of fatigue. Results obtained indicate that FMG-based grasp detection may be sensitive to movement trajectories. However, the accuracy obtainable with FMG-based grasp detection while grasping-and-moving, and moving-without-grasping, in the wider three-dimensional workspace has yet to be established. In order to investigate further, the experimental protocol in the second study of this thesis, mandated participants simultaneously move their hands across all three planes-of-movement for each grasp-and-move task completed. This protocol allowed for further investigation into the accuracy obtainable with FMG-based grasp detection in the presence of the type of upper-extremity movements that may be encountered in daily use.

Due to the constraints of onset of fatigue when testing with individuals with stroke, the aforementioned protocol was limited to the grasping of a single object, requiring a single grasp-type. Despite this, it was empirically noted that individuals with stroke varied grasp-types, movement trajectories, and employed a variety of compensatory mechanisms for repetitions of the same grasp-and-move action. In order to be practically deployable in take-home rehabilitation settings, FMG-based grasp detection systems will have to be able to detect and encourage the large variety of grasp-types that may be used in daily living. In the second study of this thesis, a protocol that mandated the use of a variety of grasp-types and objects was executed, in order to evaluate the capabilities of FMG-based grasp detection when detecting the wide variety of grasp types used in daily living [35], for the two-class grasp detection problem.

It was empirically noted that the donning of the FMG band at the forearm was problematic due to the upper-extremity movements involved in the protocol. On occasion, the band was observed slipping away from the elbow (i.e. distally) as participants completed the tasks in the experimental protocol. Hence, the acquisition of the FMG signal at the wrist instead of the forearm may prove advantageous from the perspectives of band and FMG signal stability, and intuitiveness of donning. Additionally, the presence of wires between the FMG band on the forearm and the data acquisition PC was observed to be impeding the completion of the tasks specified in the protocol. In the second study of this thesis, a wireless FMG band that was donned on the wrist was utilized. It is noteworthy that the development of this wireless FMG band was independent work that was not part of this thesis.

Chapter 4.   Two-class FMG Grasp Detection with Multiple Grasp-types and Upper-extremity Movements

4.1. Chapter Overview

This chapter describes the design, execution, and results of a study intended to investigate the accuracy achievable with FMG classification for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, as would be expected in daily living. The work presented in this chapter was intended to meet **Objective 2** and **Objective 3** of this thesis. In section 4.2, an overview of the study is provided. Section 4.3 and section 4.4 discuss the experimental methods and experimental results obtained. In section 4.5, the key observations and findings of the study are discussed.

4.2. Study Overview

In this study the accuracy of FMG classification for the two-class grasp detection problem, in the presence of upper-extremity movements, in a controlled environment was investigated. Additionally, the utility of classifying several temporal features of the FMG signal for the aforementioned two-class grasp detection problem was explored.

Based on the observations from the study executed in order to meet **Objective 1**, described in section 3.5, this study sought to evaluate FMG grasp detection with a larger variety of grasp-types, and a more substantial amount of upper-extremity movements in a controlled environment. In order to allow for a more extensive evaluation of grasp-types and upper-extremity movements, via a lengthier experimental protocol, healthy volunteers were recruited for this study. The duration of the experimental protocol in this study was limited to 150 minutes. Additionally, the FMG signal was acquired at the wrist in an effort to yield a more stable donning position and, consequently, a more stable signal. Lastly,

the FMG signal was acquired via a wireless device in order to reduce the obtrusiveness of the experimental device and minimize impediments to participants' movements.

The experimental protocol comprised of grasp-and-move tasks, requiring the use of six different grasp types frequently used in daily living, in conjunction with arm and hand movements. Data corresponding to movements-without-grasping were also included to evaluate robustness to false-positives. Several preliminary candidate temporal features, and window configurations, were identified for evaluation. Off-line classification performance of raw FMG signal was determined. Subsequently, off-line classification performance of candidate temporal features, at various window configurations, were determined and compared to that of the raw FMG signal.

4.3. Experimental Methods

4.3.1. Participants

Healthy volunteers, with full upper-extremity functional ability, were recruited for the study. Exclusion criteria for participants were: (1) less than 100% upper-extremity functional ability, and (2) height greater than allowed for by the experimental protocol.

4.3.2. Data Collection Device

The experimental device consisted of a force sensing band embedded with sixteen FSRs, and an additional external FSR that was connected by flexible wire to the device's housing. The force sensing band was 28 cm long and 2 cm wide; the center-to-center distance between successive sensors on the band was 1.7 cm. The band was donned on the participant's wrist, on the distal side of the styloid process of the Ulna bone (Figure 4-1). The band was fastened with Velcro® so that the band was tight, but comfortable for each participant. None of the participants reported discomfort due to band tightness. In the event that the band was longer than the participant's wrist circumference, the remaining length of the band was taped down to the device housing. The additional FSR was taped to the participant's thumb and served as a validation sensor. The validation sensor, connected to the device housing, is depicted in Figure 4-2. The signal from the

validation sensor was used to label each datum that corresponded to a grasp (i.e. whenever force was observed on the thumb due to the grasping of an object). Data from the validation sensor served as the class label, for training classifiers and evaluating FMG grasp classification performance. Data from the sixteen channel (i.e. sensors) force sensing band and validation sensor were sampled concurrently, at 10 Hz, with an AT Mega 328 Microcontroller, located within the device housing, and wirelessly transmitted to a data acquisition PC via Bluetooth. The 10 Hz sampling rate was selected based on data-throughput constraints of the wireless communication protocol used for the experimental device, and has been shown to be sufficient for FMG-based grasp detection [31], as a majority of volitional upper-extremity motion occurs in the 0-3 Hz range [71]. Custom LabVIEW™ [72] software was written to collect data from the data acquisition device. The software consisted of a communications module that received and logged data from the FMG data acquisition device and a graphical user interface that allowed the operator to insert a label associated with each round of data collection in the experimental protocol.

Figure 4-1: Device Donning Position
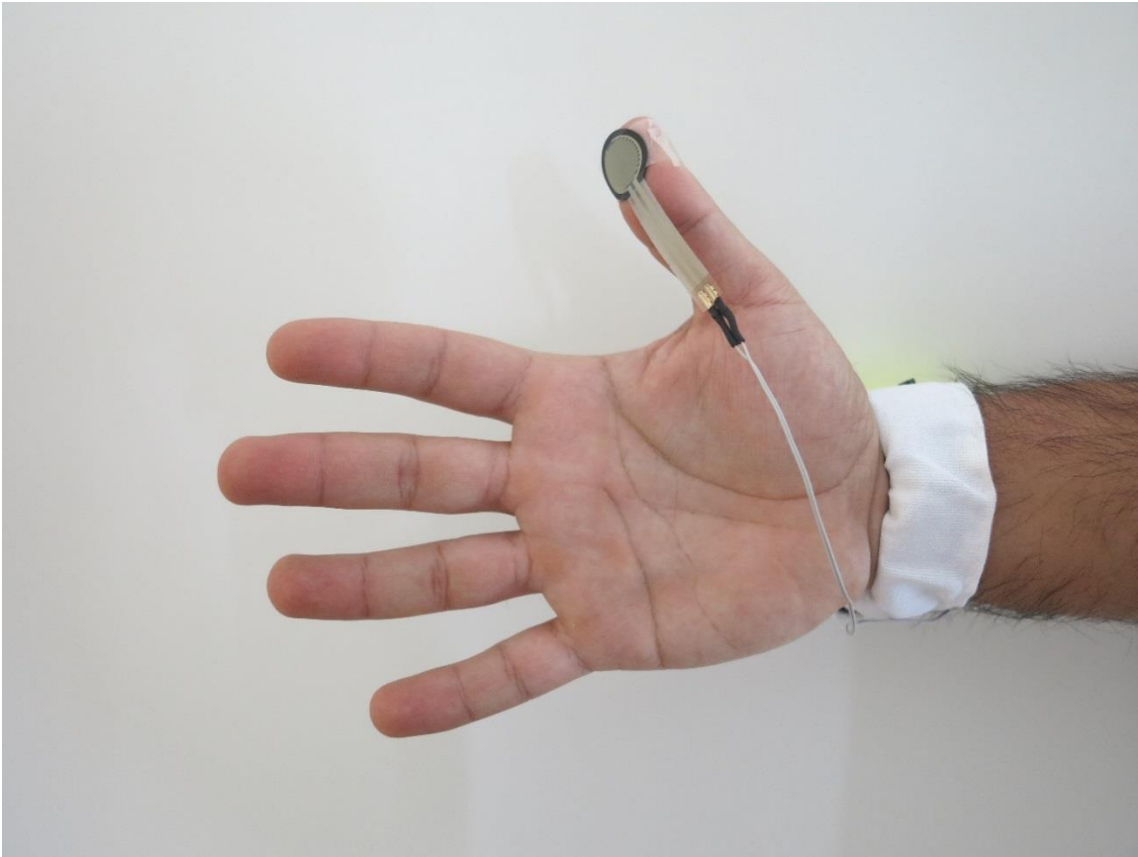
Figure 4-2: Validation Sensor on Thumb for Labelling Data

4.3.3.   Experimental Protocol

The experimental protocol comprised of multiple repetitions of several grasp-and-move tasks. As noted previously, research has shown that the FMG signal is sensitive to joint positions of the wrist, forearm and elbow [32, 33], which are expected to vary with the varying movement trajectories that are likely to be employed when grasping and moving objects in daily use. This phenomenon was also observed in the study executed in Chapter 3. In order to explore the limits of FMG grasp detection, the protocol was designed to involve simultaneous movement across all three planes-of-movement, and the use of joint movement (such as wrist flexion and extension, elbow flexion and extension, and shoulder flexion and extension) for task completion. Additionally, to evaluate the classifiers' robustness to false-positives, portions of the protocol required movements-without-grasping.

Three-dimensional Workspace

Figure 4-3 shows a model of the three-dimensional workspace created for the protocol. The workspace consisted of five shelving units placed on top of a U-shaped table. The U-shaped table was 80 cm tall. Four shelving units, 64 cm tall, were positioned at the four corners of the workspace. The space below the shelving units (i.e. the surface of the table) was accessible. Each shelving unit was used to create two target positions: (i) one at the top of the shelf (height above table = 64 cm), and (ii) one at the bottom of the shelf (height above table = 0 cm). This created a total of eight target positions (positions 1-8 in Figure 4-3). The fifth shelving unit was 32 cm tall (half the height of the other shelving units) and was placed in the center of the workspace to create the origin position (position 0 in Figure 4-3). The participant was asked to stand in the center of the U-shaped table, directly in front of the origin position, and grasp-and-move the objects to-and-from the various shelving units on the table. The protocol required the top of the target shelves (positions 1, 3, 5 and 7 in Figure 4-3) to be 5 cm above the height of the participant's shoulder, such that, the participant would be required to forward flex his/her shoulder above the horizontal to reach the target positions on top of the shelving unit. In the event that the participant was too short for this constraint, an adjustable platform was provided for the participant to stand on (Figure 4-4). Potential participants who were too tall to meet this constraint were excluded from this study. The number of shelves and their positions relative to the participant were selected to ensure that participants had to simultaneously move their arms across all three planes-of-movement (i.e. superior-inferior, medial-lateral, and ventral-dorsal) in order to transport objects to-and-from each of the shelves. Figure 4-5 shows the actual workspace created.
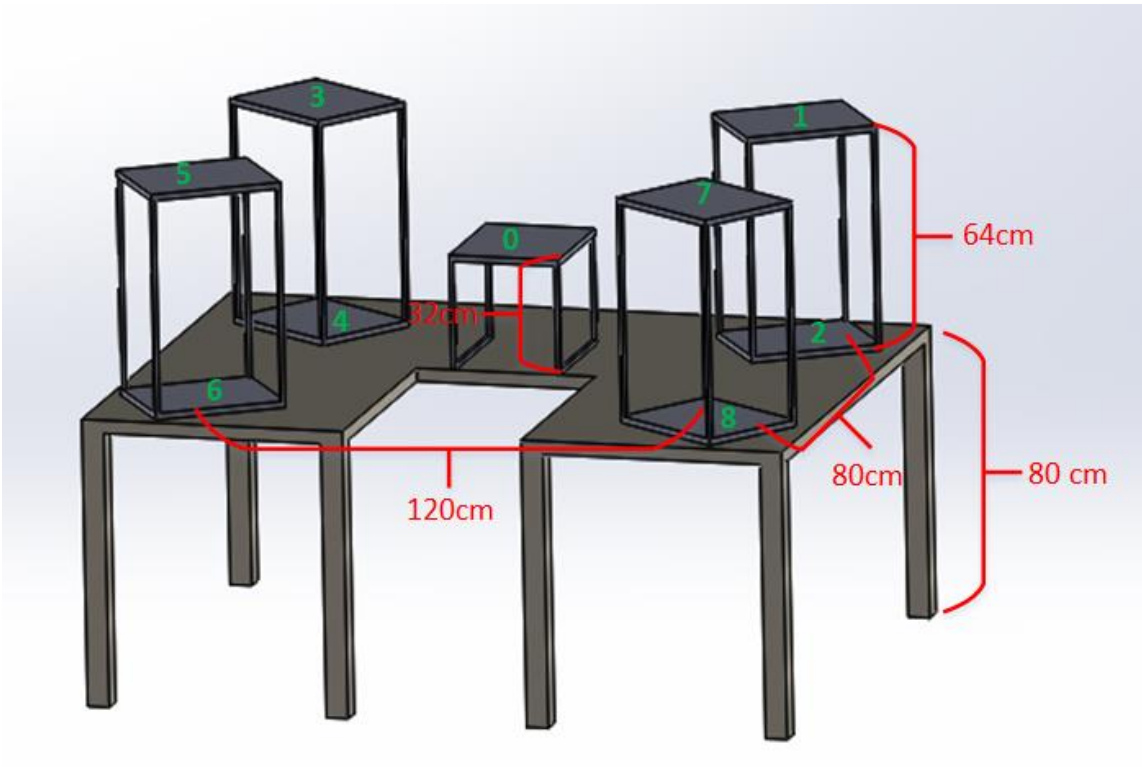
Figure 4-3: Model of Task Workspace

Figure 4-4: Adjustable Platform
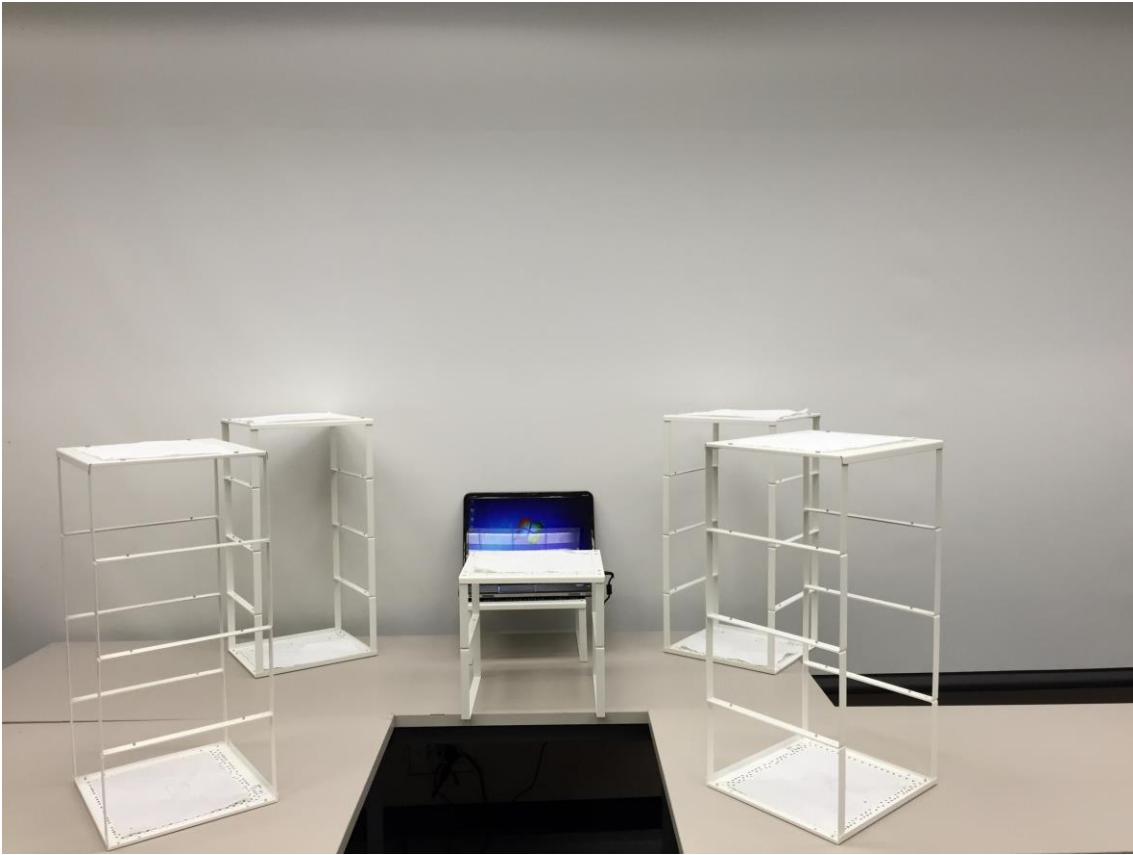
Figure 4-5: Actual Task Workspace used for Protocol

Grasp Types

Grasp types that are frequently used in ADL were selected for the protocol. Bullock et al. monitored the relative frequency of the different grasp-types used by two machinists and two housekeepers as part of their daily activities [35]. Seven of the most frequently used grasp-types identified by Bullock et al. were selected for evaluation: (i) medium wrap, (ii) precision disk, (iii) lateral pinch, (iv) tripod, (v) lateral tripod, (vi) power sphere, and (vii) thumb-2 finger. Combined, these grasp-types account for more than 50% of the grasps that occurred for the two machinists and two housekeepers in Bullock et al.'s study [35]. A specific object was selected for each of the grasp types. The tripod and lateral tripod grasp-types were considered identical from the FMG perspective, as they involve identical finger positions and are in fact variations of hand orientation in the three-dimensional workspace. It is noteworthy that other variants of grasp taxonomy do not distinguish between the tripod and lateral tripod grasp-types [73]. Table 4-1 lists the objects selected

for each grasp-type. Figure 4-6 to Figure 4-11 depict the objects selected for each grasp-type.

Table 4-1: Objects Used for Each Grasp-type Evaluated

Index	Grasp-type	Object
1	Medium Wrap	Drinking Glass
2	Precision Disk	Quarter Bowl
3	Lateral Pinch	Quarter Plate
4	Tripod / Lateral Tripod	Block from Box & Blocks Test [74]
5	Power Sphere	Tennis Ball
6	Thumb-2 Finger	Pencil

Figure 4-6: Drinking Glass for Medium Wrap Grasp-type

Figure 4-7: Quarter Bowl for Precision Disk Grasp-type

Figure 4-8: Quarter Plate for Lateral Pinch Grasp-type
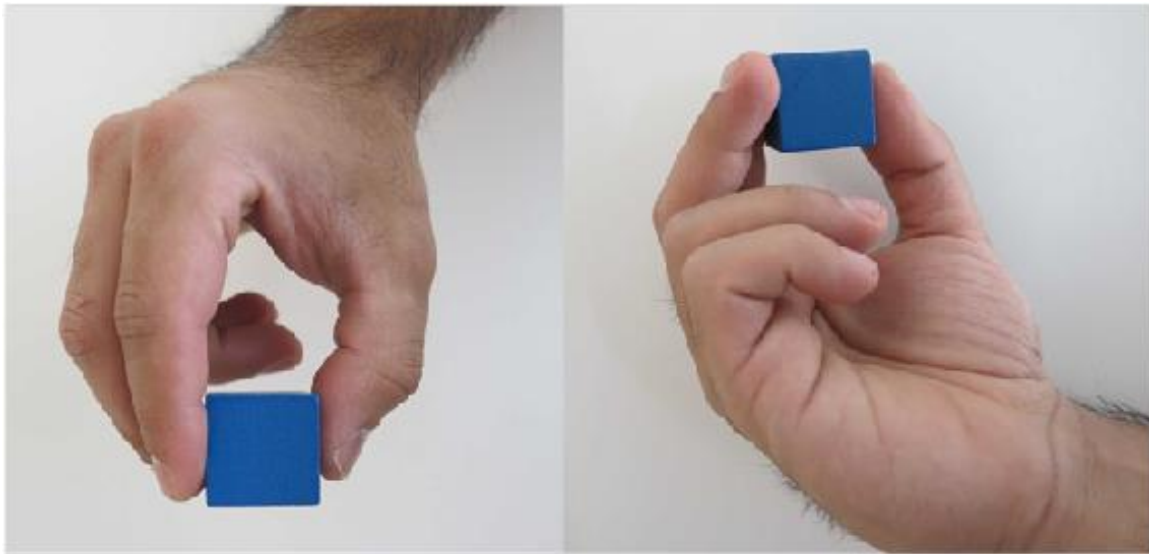
Figure 4-9: Block for Tripod/Lateral Tripod Grasp-type
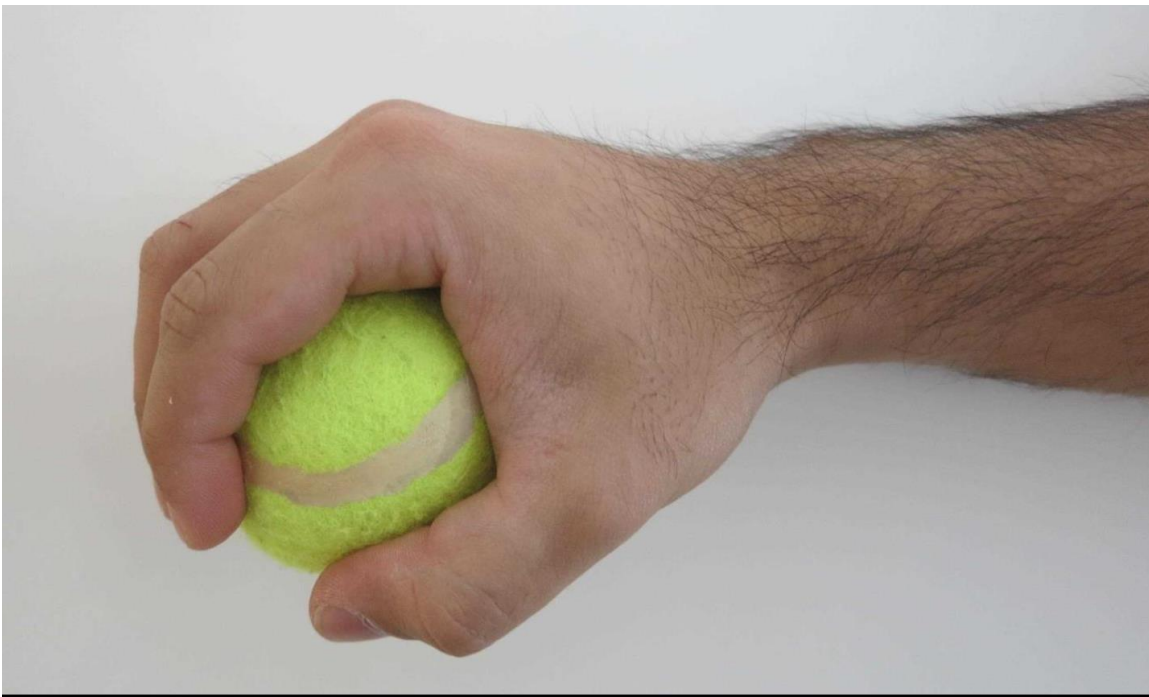
Figure 4-10: Tennis Ball for Power Sphere Grasp-type

Figure 4-11: Pencil for Thumb-2 Finger Grasp-type

Instructions to Participants

The protocol consisted of three identical rounds of data collection, each of which consisted of grasp-and-move activity, and movements-without-grasping. The number of rounds was selected as a compromise between the goal of collecting the maximum amount of data possible, while avoiding muscular and mental fatigue in participants. Participants were asked to use their dominant hands for all activity. Participants were asked to start the protocol with their hand in the neutral position (i.e. hand by their side). For grasp-and-move activity, the participant was asked to move his/her hand to the origin position (position 0 in Figure 4-3), grasp and lift the object from the origin position, move and release the object at one of the target positions (positions 1-8 in Figure 4-3), and to return his/her hand to the neutral position (by his/her side). Subsequently, the participant was asked to move their hand to the previously used target position (positions 1-8 in Figure 4-3), grasp and lift the object from the target position (positions 1-8 in Figure 4-3), and to move and release the object at the origin position, before retuning his/her hand to the neutral position (by his/her side). The participant was asked to repeat this process for each

of the target positions starting with position 1 in Figure 4-3, and ending with position 8 in Figure 4-3. This resulted in sixteen grasp-and-move actions for each object. It is noteworthy that the movement and placement of the objects at the target locations required the participant to move his/her wrist, elbow, and shoulder joints, and involved arm movement across all three planes-of-movement. This allowed for the evaluation of the false-negative rate of FMG-based grasp detection while grasping in the presence of upper-extremity movements in the controlled three-dimensional workspace.

Data corresponding to movements-without-grasping in the three-dimensional workspace were collected in-between the grasp-and-move activity for each object. Specifically, participants were asked to move their hands to the positions that would correspond to the movement involved in the grasp-and-move case, but without grasping any object. The participant was asked to move his/her hand from the neutral position, to hover over the origin position (position 0 in Figure 4-3), and to then move his/her hand to hover over a target position (positions 1-8 in Figure 4-3), before returning his/her hand to the neutral position. The participant was subsequently asked to move his/her hand from the neutral position to the previously used target position (positions 1-8 in Figure 4-3), and then to move his/her hand to hover over the origin position (position 0 in Figure 4-3), before returning to the neutral position. The sequence was repeated for each target position, resulting in sixteen rounds of movement. Data corresponding to movements-without-grasping were included in order to evaluate the false-positive rate of FMG-based grasp detection in the presence of upper-extremity movements, which may be encountered as part of ADL. The above described movements-without-grasping, and grasp-and-move actions, were repeated for each grasp type and object. The entire sequence was repeated three times to form the three rounds of data collection. The instructions used for each round of data collection are detailed in Appendix B.

4.3.4.   Data Analysis

Features Evaluated

Four basic temporal feature extraction techniques were identified for evaluation: (i) Mean Absolute Value (MAV), (ii) Root Mean Squared (RMS), (iii) Coefficients of Linear (two degree of freedom) Fit (LF), and (iv) Coefficients of Parabolic (three degree of freedom) Fit (PF). These feature extraction techniques derive from two of the several ways in which time series can be represented. The MAV and RMS provide a representation of the overall magnitude of the data within the feature window. The LF, and PF are methods of modeling trends within the data. While more advanced methods of representing time series and extracting temporal features exist, the use of these features allowed for clear interpretation of the merits of temporal feature extraction for FMG classification, in this preliminary study. The specific rationale for evaluating each of these features is as follows.

MAV is a representation of the overall FMG activity, for the given channel, within the window. The use of an overall representation may lead to a reduction in misclassification due to changes in the FMG signal caused by upper-extremity movements and other outliers in the data. MAV has been used as a feature in the classification of SEMG signals [75], and AMG signals [76]. The MAV for each channel can be calculated using (eq 4.1) ; where i is the sample number from 1 to n, n is the window size, and FMG_i is the ith FMG signal sample for the given channel.

$$MAV = \frac{1}{n}\sum_{i=1}^{n}|FMG_i| \qquad \text{(eq 4.1)}$$

RMS is a representation of the power of the FMG signal for the given channel, within the window. RMS may relate to grip strength, which could potentially increase the ease of separation of data corresponding to the participant grasping, or not grasping, an object. RMS has been shown to be an effective predictor of force of contraction in AMG signals [77], and has been used in EMG classification [78]. The RMS for each channel can be derived using (eq 4.2); where n is the window size, and $FMG_{k,i}$ is the ith sample of the FMG signal for channel k.

$$RMS_k = \sqrt{\frac{1}{n}\left(FMG_{k,1}^2 + FMG_{k,2}^2 \cdots + FMG_{k,n}^2\right)}$$

(eq 4.2)

LF models the trend and amplitude of the FMG for the given channel, within the window. The model may reflect a change in the force exerted by the participant's musculo-tendinous complex and may capture the temporal change in force that occurs as the user grasps, and releases an object. The LF is part of a wider class of models that are linear in the data space [55]. The use of LF for the preliminary evaluation of temporal features for FMG classification allows for easier interpretation of the utility of linear models. The LF for each channel can be calculated by computationally applying an error minimization routine to the least-squares residual expression in (eq 4.3) to solve for fit parameters *a* and *b*; where R^2 is the residual error term of the fit, *i* is the sample number from 1 to *n*, and *n* is the window size, and FMG_i is the ith FMG signal sample for the given channel. In this study, the fitting routine was carried out using the polyfit function in MATLAB® [79].

$$R^2 = \sum_{i=1}^{n} [FMG_i - (a + bi)]^2$$

(eq 4.3)

PF builds upon the LF and allows for modeling with three degrees of freedom. The additional fit term provided by the PF allows for the modeling of non-linear trends in the data within the window. The model may reflect a change in the force exerted by the participant's musculo-tendinous complex and may capture the temporal change in force that occurs as the user grasps, and releases an object. The PF has been shown to be an effective feature in handwritten character recognition [80]. The PF is part of a wider class of models that are non-linear in the data space [55]. The use of PF for the preliminary evaluation of temporal features for FMG classification allows for the interpretation of the utility of models that are non-liner in the data space. The PF for each channel can be calculated by computationally applying an error minimization routine to the least-squares residual expression in (eq 4.4) to solve for fit parameters *a*, *b*, and *c*; where R^2 is the residual error term of the fit, *i* is the sample number from 1 to *n*, and *n* is the window size,

and FMG_i is the ith FMG signal sample for the given channel. In this study, the fitting routine was carried out using the polyfit function in MATLAB® [79].

$$R^2 = \sum_{i=1}^{n} [FMG_i - (a + bi + ci^2)]^2$$

(eq 4.4)

In order to explore the effect of window configuration on the effectiveness of temporal feature extraction, each feature extraction technique was run on the FMG data using overlapping windows of two sizes: (1) a three sample window (0.3 seconds), and (2) five sample window (0.5 seconds). The window sizes were selected based on an empirical analysis of the time-scale in which a transition between a grasp and release occurs. In all cases, the feature value associated with a single instantaneous FMG sample for a given channel was found by transforming data within a window around the instantaneous FMG sample. The symmetry of each window was also varied. Each feature was calculated with three different symmetries: (1) left justified (i.e. feature value corresponding to the current sample depends on the current sample and samples in the window before it), (2) right justified (i.e. feature value corresponding to the current sample depends on the current sample and samples in the window after it), and (3) centre justified (i.e. feature value corresponding to the current sample depends on the current sample and a symmetrically equal number of samples before and after it). The feature values were constructed sample-by-sample and channel-by-channel. An exemplary feature extraction scheme for a center-justified, 3 sample window is depicted in Figure 4-12. These variations in window configuration resulted in a total of twenty-four feature sets for evaluation (4 features x 2 window sizes x 3 window symmetries). Table 4-2 lists the abbreviated names for each feature evaluated. Features were named using an {X,Y} suffix, where X denotes the number of samples prior to the current sample that are used in the feature transformation, and Y denotes the number of samples after the current sample that are used in the feature transformation.
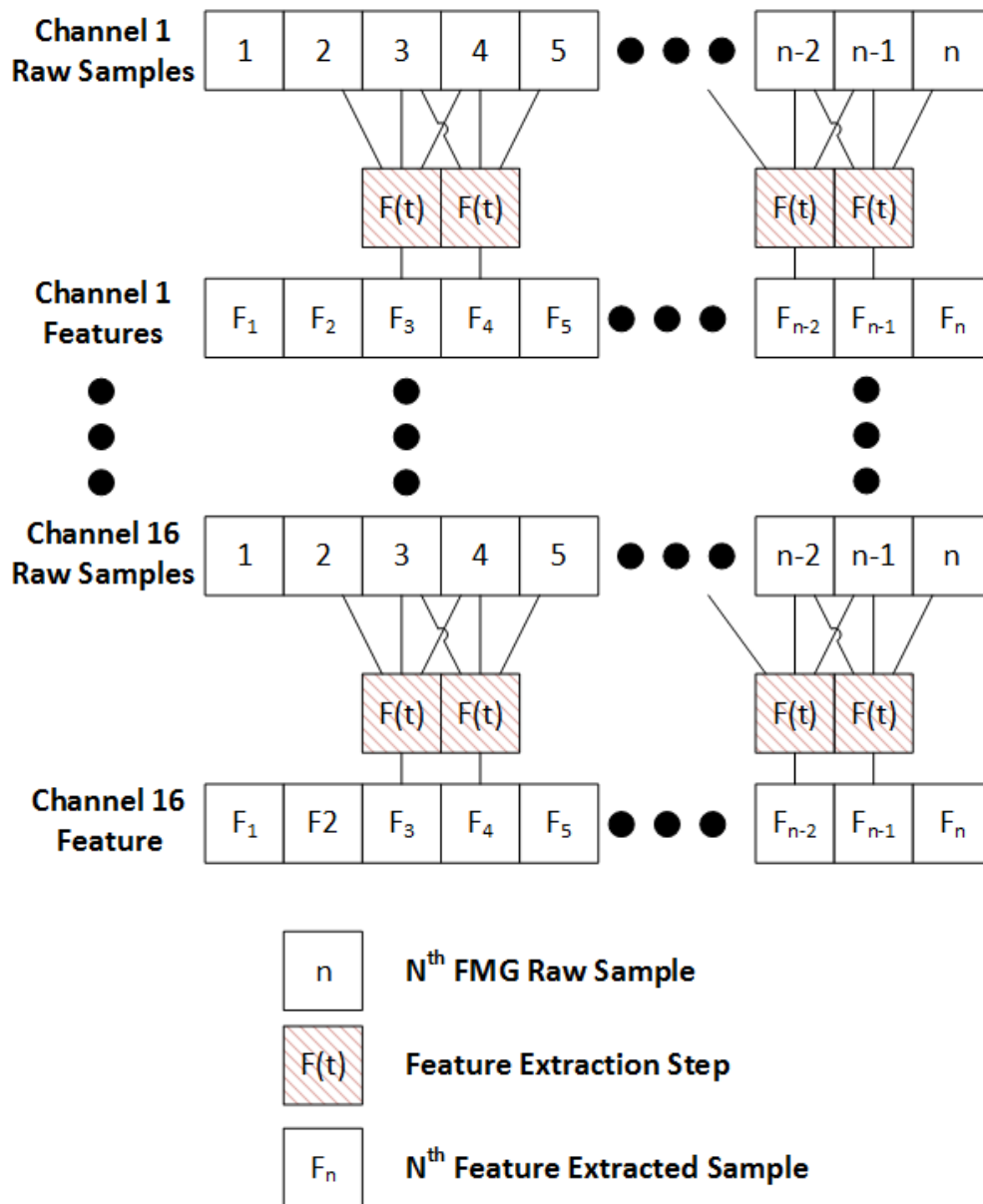
Figure 4-12: Exemplary Feature Extraction Scheme for Centre-justified, 3 Sample Window

Table 4-2: Abbreviated Names for Each Feature Type, Window Size, and Window Symmetry Evaluated

Type	3 Sample Window			5 Sample Window		
	Left justified	Centre justified	Right justified	Left justified	Centre justified	Right justified
MAV	MAV {2,0}	MAV {1,1}	MAV {0,2}	MAV {4,0}	MAV {2,2}	MAV {0,4}
RMS	RMS {2,0}	RMS {1,1}	RMS {0,2}	RMS {4,0}	RMS {2,2}	RMS {0,4}
LF	LF {2,0}	LF {1,1}	LF {0,2}	LF {4,0}	LF {2,2}	LF {0,4}
PF	PF {2,0}	PF {1,1}	PF {0,2}	PF {4,0}	PF {2,2}	PF {0,4}

Feature Evaluation Program

A feature evaluation program was developed in MATLAB® [81] in order to be able to evaluate the off-line classification performance obtained with different features of the FMG signal, in comparison to the classification performance of the raw FMG signal. The program was based on the wrapper feature evaluation method [82, 83, 84]. In the wrapper method, a classifier is trained and tested using each candidate feature in order to understand their individual effectiveness for classification purposes. In this study, the wrapper method was used to evaluate off-line classification performance of the raw FMG signal and each of the twenty-four candidate feature configurations identified. While other more computationally efficient methods for feature evaluation and selection exist, the wrapper method was chosen as it allows for unambiguous understanding of the effectiveness of each feature proposed [82]. The effectiveness of each feature in comparison to the raw FMG signal was evaluated using the Area under the Receiver Operating Curve (AUC). Receiver Operating Curves (ROCs) plot the true-positive rate (sensitivity) against the false-positive rate (1 - specificity) for a classifier [85]. The ideal classifier will have a true-positive rate of 1 (i.e. sensitive) and a false-positive rate of 0 (i.e. specific), yielding an AUC of 1. The classification accuracy was also calculated for each feature, but was not used as a primary measure of the effectiveness of candidate features. AUC was chosen as the measure of effectiveness as it is insensitive to the relative distribution of the classes within the data, and hence, provides a robust method of assessing classification performance when compared to classification accuracy [85].

Detailed design and implementation details of the feature evaluation program are summarized in Appendix C. The program consisted of two run-time stages, as depicted in Figure 4-13. In the first stage, features were extracted for all three rounds of data collection

for each participant and stored to file for future classification. In the second stage, classification performance with each of the extracted features was evaluated for each participant. An RBF-SVM, implemented via the LIBSVM library [69] for MATLAB®, was used for classification. AUC and accuracy were evaluated in a three-fold cross-validation scheme. Classification accuracy associated with each feature was evaluated for each round of data collection, for each participant. Specifically, data from a given round was classified using a classifier that was constructed with data from the remaining two rounds for the participant. For each round, a ten-fold cross-validation scheme within the training data was used to determine the optimal cost [69] and gamma [69] for the RBF-SVM via a grid search. Combinations of six different cost (cost = 0.01, 0.1, 1, 10, 100, 1000) and six different gamma values (γ = 0.01, 0.1, 1, 10, 100, 1000) were evaluated. The optimal cost and gamma, found using cross-validation within the training data was then used to for classifying the unseen, testing data (i.e. data for the round under evaluation). The same process was repeated for the next round of data collection, until all rounds of data collection for the participant were evaluated. Subsequently, the average AUC and accuracy obtained across three-folds for a given feature and participant was calculated.

The above process was repeated for each participant. The average AUCs and accuracies obtained across all participants for each feature were then calculated. Features that resulted in increased AUC when compared to the raw FMG signal were identified. A single-tailed, paired samples Student's t-test with alpha of 0.05 was used to determine if the increase in AUC seen due to the use of a particular feature was statistically significant.
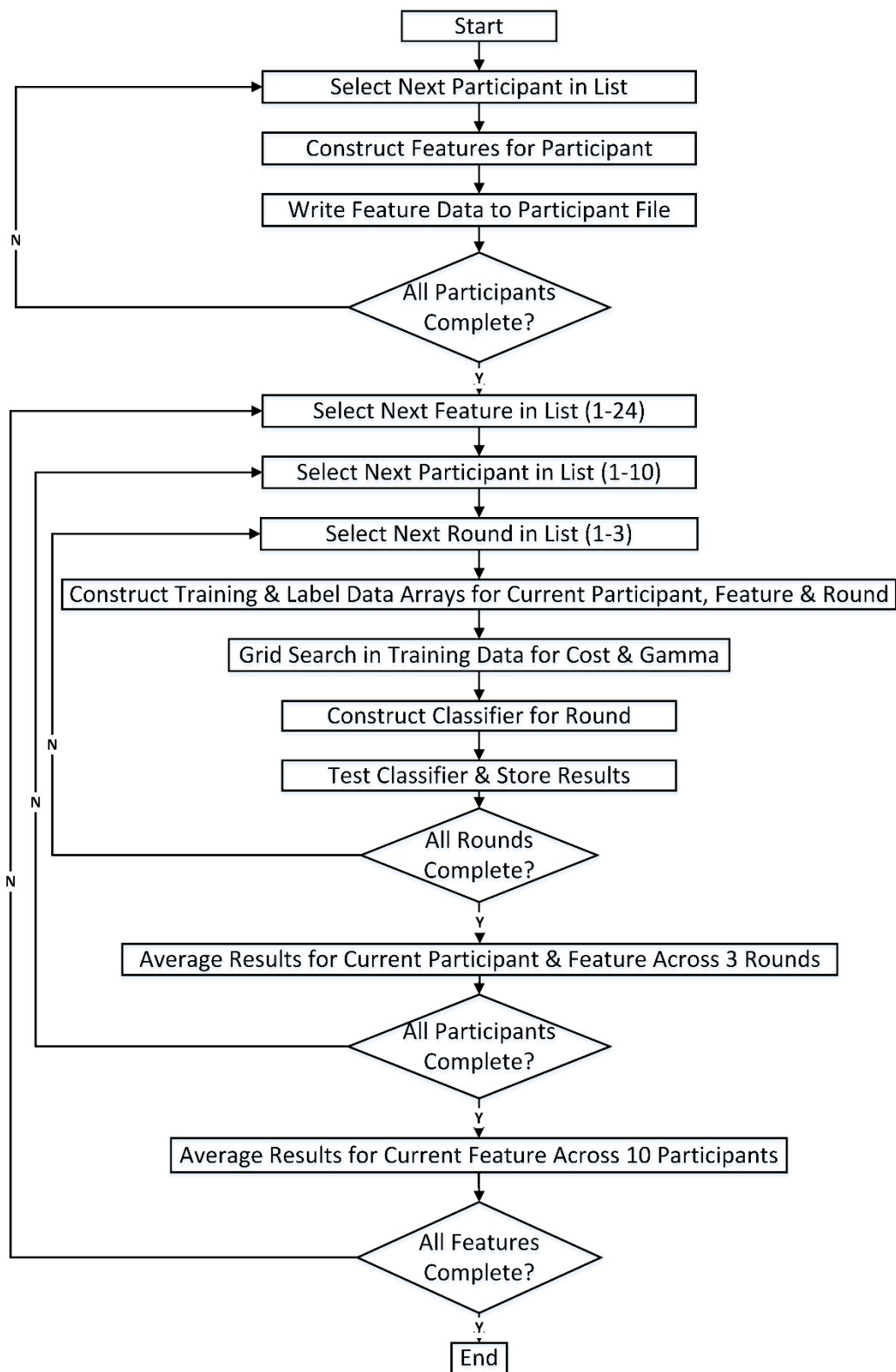
Figure 4-13: Feature Evaluation Flowchart

4.4. Experimental Results

4.4.1. Participants

Ten healthy participants were recruited for this study. Participant age, gender, and the need for the height adjusting step to meet the protocol's height restriction, are listed in Table 4-3. All participants provided informed consent for participation in the study.

Table 4-3: Participant Information

ID	Age	Gender	Height
1	23	Female	Yes
2	22	Female	No
3	27	Female	Yes
4	21	Male	No
5	23	Male	No
6	23	Male	No
7	24	Male	No
8	25	Female	Yes
9	21	Male	No
10	27	Male	No

4.4.2. Feature Evaluation Results

Figure 4-14 depicts the AUC obtained for the raw FMG signal and for each feature evaluated. Figure 4-15 depicts the corresponding classification accuracy obtained for the raw FMG signal, and for each feature evaluated. Eighteen of the twenty-four features evaluated resulted in higher AUC when compared to the raw FMG signal; eleven features produced statistically significant increases in AUC. The AUC, accuracies, associated standard deviations, and P-values for the eighteen features that demonstrated increased performance are listed in Table 4-4. The raw FMG signal yielded an AUC of 0.819 ($\sigma = 0.098$) and an accuracy of 88.8% ($\sigma = 5.18\%$). The largest increase in AUC over the raw FMG signal was obtained using PF {2,2}, yielding an AUC of 0.869 ($\sigma = 0.061$) and an accuracy of 90.6% ($\sigma = 4.30\%$) The increases in AUC (P = 0.011) and accuracy (P = 0.031) observed over the raw FMG signal are statistically significant for PF {2,2}.
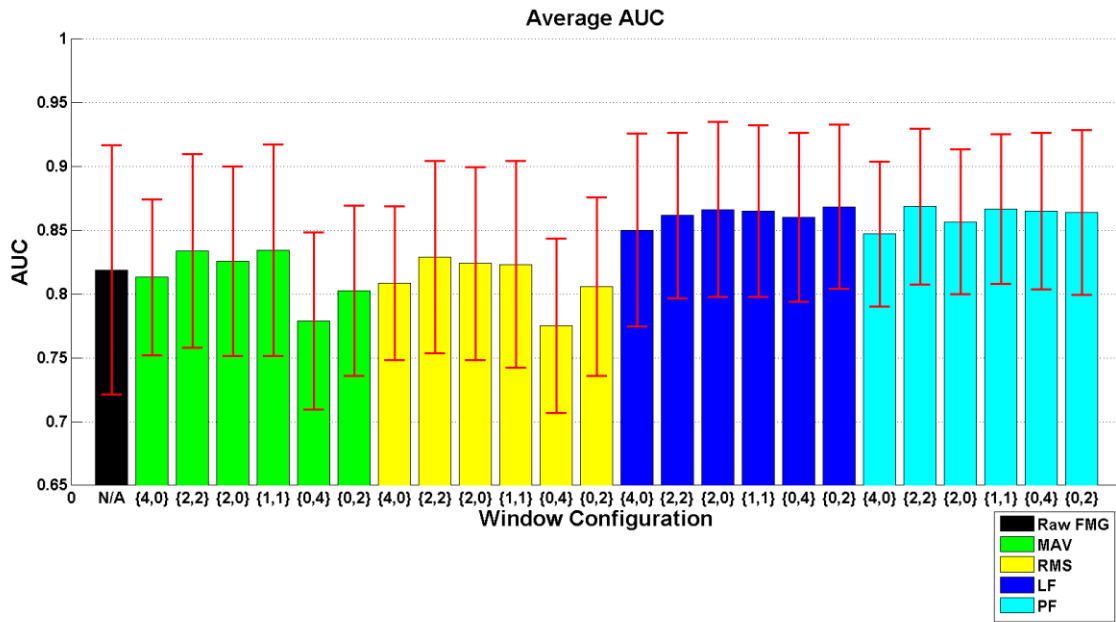
Figure 4-14: AUC for all Features Evaluated
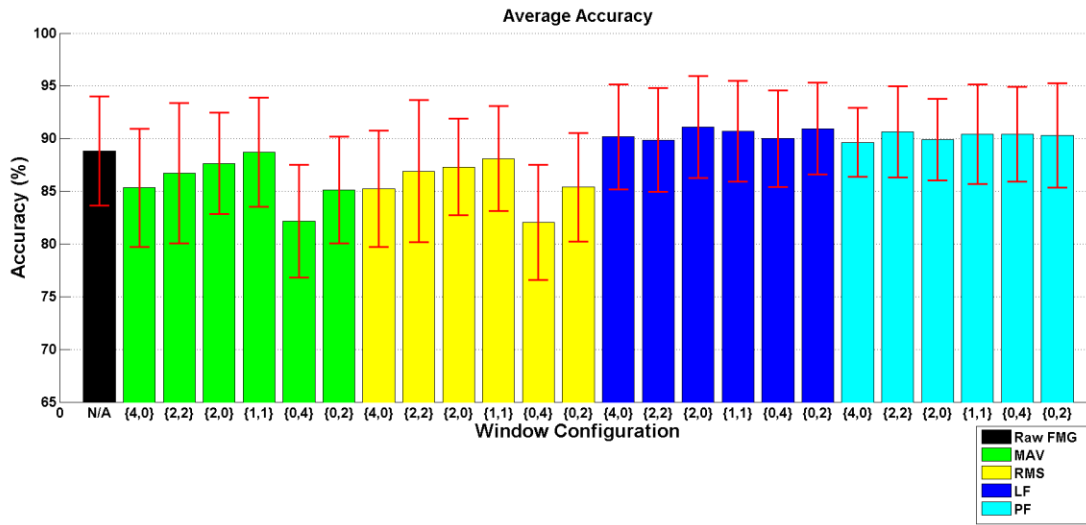
Figure 4-15: Classification Accuracy for all Features Evaluated

Table 4-4: Features with Increased AUC over the Raw FMG Signal

Feature	AUC		Accuracy		P-Value
	Value	σ	Value (%)	σ (%)	
PF {2,2}	0.869	0.061	90.6	4.30	0.011
LF {0,2}	0.869	0.064	90.9	4.34	0.004
PF {1,1}	0.867	0.059	90.4	4.71	0.017
LF {2,0}	0.866	0.069	91.1	4.83	0.007
PF {0,4}	0.865	0.061	90.4	4.47	0.012
LF {1,1}	0.865	0.067	90.7	4.78	0.013
PF {0,2}	0.864	0.064	90.3	4.95	0.023
LF {2,2}	0.862	0.065	89.9	4.94	0.017
LF {0,4}	0.860	0.066	90.0	4.56	0.007
PF {2,0}	0.857	0.057	89.9	3.86	0.050
LF {4,0}	0.850	0.076	90.2	4.99	0.038
PF {4,0}	0.847	0.057	89.6	3.29	0.103
MAV {1,1}	0.835	0.083	88.7	5.18	0.078
MAV {2,2}	0.834	0.076	86.7	6.67	0.152
RMS {2,2}	0.829	0.075	86.9	6.73	0.253
MAV {2,0}	0.826	0.074	87.7	4.79	0.312
RMS {2,0}	0.824	0.076	87.3	4.59	0.359
RMS {1,1}	0.823	0.081	88.1	4.97	0.363
Raw FMG	0.819	0.098	88.8	5.18	N/A

4.4.3.   Discussion of Results

The first objective of this study was to investigate the accuracy of FMG classification for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, in a controlled environment. Classification of the raw FMG signal yielded an accuracy of 88.8% (σ = 5.2 %) for the aforementioned two-class grasp detection problem.

The second objective of this study was explore the utility of classifying temporal features of the FMG signal for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, in a controlled environment. It was hypothesized that the identification of a suitable set of temporal features of the FMG signal may improve grasp classification accuracy. The MAV, RMS, LF, and PF features all yielded increases in AUC when compared to the raw FMG signal at certain window sizes.

However, increases seen for the MAV and RMS were not statistically significant and were susceptible to window configuration, as increases in performance were only seen at certain window configuration. In contrast, LF and PF features yielded increases in performance for all window sizes, and yielded larger increases in performance when compared to MAV and RMS. Increases in classification performance were statistically significant for LF, regardless of the window configuration used. For the PF feature, increases in classification performance were significant for all but one window configuration. A potential explanation for the superior performance and insensitivity to window size of the LF and PF features is the fact that they attempt to model the trends within the data window, while the MAV and RMS both calculate an estimate of the overall magnitude of the data for the given window. It is possible that the capturing of trend information allows for greater ease of separation of FMG data when compared to capturing only magnitude information.

These results suggest that the use of feature extraction techniques that attempt to model the FMG data as a linear, or non-linear, time series may yield increased grasp detection performance, when compared to feature extraction techniques that attempt to capture an overview of the FMG data within the window, and when compared to the classification of the raw FMG signal. In addition, results obtained suggest that features that model trends within the window are less susceptible to the size, or symmetry, of the data window from which the model is generated. It is noteworthy, however, that the use of model-based features result in additional feature value(s) for each sample of data on each FMG channel, which in turn increases the computation cost of classification. The benefits obtained by using this class of features will have to be traded off with additional computation time and complexity for the different applications of FMG-based grasp detection. The use of an additional degree of freedom in the PF did not significantly improve classification performance over the LF despite the additional feature value for each sample of data on each FMG channel. This indicates that the use of non-linear models may not provide benefit to offset the additional computation cost, when compared to linear models.

4.5. Summary and Implications of Results

The scope of this study was limited to establishing the accuracy of classifying the raw FMG signal for the two-class grasp detection problem, and investigating the utility of classifying a preliminary set of temporal features of the FMG signal for the two-class grasp detection problem. Classification of the raw FMG signal yielded an accuracy of 88.8% (σ = 5.18 %) for a data-set corresponding to grasp-and-move actions, that required a variety of grasp-types and upper-extremity movements, and also contained data corresponding to movements-without-grasping. The utility of classifying temporal features of the FMG signal was explored by comparing the AUC obtained with raw FMG to that obtained with the various types of temporal features evaluated at various window configurations. Eighteen of the twenty-four feature configurations evaluated resulted in higher AUC when compared to the raw FMG signal; eleven feature configurations produced statistically significant increases in AUC. The largest increase obtained was with PF, yielding AUC of 0.869 (σ = 0.061), corresponding to a 6.1% relative increase over the AUC of 0.819 (σ = 0.098) obtained with the raw FMG signal. The results obtained indicate that features that model temporal trends within the data may increase FMG-based grasp detection performance. In future studies, the use of a larger set of model-based temporal features should be evaluated with FMG data from individuals with stroke, who might ultimately benefit from this technology. Additionally, the accuracy of FMG-based grasp detection with upper-extremity movements in an uncontrolled environment, such as the home environment, should be evaluated.

Chapter 5. Conclusions

5.1. Chapter Overview

This chapter provides a summary of the findings of this thesis, and an outline of potential areas of future research. In section 5.2 the thesis objectives are recalled, and their related findings are presented. In section 5.3 future work is suggested.

5.2. Summary of Objectives and Findings

This thesis sought to explore the suitability of using FMG for grasp detection in stroke rehabilitation applications. Given the emphasis placed on grasp training in upper-extremity stroke rehabilitation [16, 17], a FMG-based device that is capable of monitoring and encouraging grasping in stroke survivors could potentially enhance and optimize the rehabilitation process. Despite the promising results on FMG classification for a variety of applications, additional research questions remained in order to establish the suitability of the use of FMG for grasp detection in stroke rehabilitation applications. Based on these key research questions, which are summarized in Chapter 2, three objectives were identified for this thesis.

Objective 1 was to perform a preliminary investigation on the accuracy of FMG-based grasp detection in individuals with stroke, who have upper-extremity impairments, in a controlled environment. In order to meet this objective, a study was designed and executed to explore the feasibility of acquiring and classifying the FMG signal from individuals with stroke, with upper-extremity impairments, for the purpose of grasp detection. The experimental protocol consisted of twenty repetitions of three grasp-and-move tasks, using a single object and grasp-type. Experimental data were collected from eight individuals with stroke and eight healthy volunteers. FMG classification accuracy was found to be lower for participants with stroke when compared to healthy participants. Despite this, FMG-based sensing achieved a high accuracy of 92.6% (σ = 3.20%) for grasp detection with participants with stroke, using a RBF-SVM classifier. In order to evaluate the ease of separation of FMG data, the use of a LDA

classifier was also evaluated. Classification accuracy was lower with the LDA, when compared to the RBF-SVM, for both individuals with stroke and healthy individuals. However, average grasp detection accuracy remained above 90%, for both participants with stroke and healthy participants, with the LDA, suggesting that linear classification may be adequate for FMG-based grasp detection. The effect of training-set size on classification performance was also investigated by varying the training-set size for a fixed-size testing-set. Experimental results indicate that FMG-based grasp detection required more training data to achieve commensurate classification accuracy for individuals with stroke, when compared to healthy participants. However, a training-set size that was 50% of the testing-set size was sufficient to achieve greater than 90% accuracy for individuals with stroke. These promising results indicate that FMG sensing may be capable of monitoring grasping in individuals with stroke, with mild to moderate upper-extremity impairments.

Objective 2 was to perform a preliminary investigation on the accuracy of FMG classification for the two-class grasp detection problem, using a variety of grasp-types and upper-extremity movements, with healthy volunteers, in a controlled environment. **Objective 3** was to perform a preliminary investigation on the utility of classifying temporal features of the FMG signal for the two-class grasp detection problem, using a variety of grasp-types and upper-extremity movements, with healthy volunteers, in a controlled environment. In order to meet these objectives, a study was designed and executed to investigate the accuracy achievable with FMG classification for the two-class grasp detection problem, with a variety of grasp-types and upper-extremity movements, as would be expected in daily living. The experimental protocol comprised of grasp-and-move tasks, requiring the use of the six most frequently used grasp-types, in conjunction with arm and hand movements, in a controlled environment. Data corresponding to movements-without-grasping were also included to evaluate robustness to false-positives. The raw FMG signal yielded an accuracy of 88.8% (σ = 5.18 %) for the two-class grasp detection problem. The results obtained provide evidence that the use of FMG classification for the two-class grasp detection problem, with a wide variety of grasp-types and upper-extremity movements, as would be expected in daily living, may be feasible.

The utility of classifying temporal features of the FMG signal was explored by comparing the AUC obtained with classification of the raw FMG signal, to the AUCs obtained with classification of four candidate temporal features, evaluated at six window configurations. Eighteen of the twenty-four feature configurations evaluated resulted in higher AUC when compared to the raw FMG signal; eleven feature configurations produced statistically significant increases in AUC. The largest increase obtained was with PF, yielding AUC of 0.869 ($\sigma = 0.061$), corresponding to a 6.1% relative increase over the AUC of 0.819 ($\sigma = 0.098$) obtained with the raw FMG signal. Furthermore, all model-based features evaluated yielded increases in classification performance for all window sizes, and yielded larger increases in performance, when compared to features that attempt to capture an overview of the FMG data within the window. These results suggest that the use of feature extraction techniques that attempt to model the FMG data as a linear, or non-linear, time series may yield increased grasp detection performance, for the two-class grasp detection problem.

The results obtained in this thesis provide preliminary confirmation of the suitability of using FMG classification for grasp detection in upper-extremity stroke rehabilitation applications, and pave the way for further research. FMG classification was shown to be effective at detecting grasps in individuals with stroke; achieving accuracy in excess of 90%, with a training-set size that was 50% of the testing-set size. Furthermore, in healthy volunteers, FMG grasp detection was shown to be capable of discriminating between occurrences of any of the six most frequently used grasp-types in daily living, versus a lack of grasping, in the presence of upper-extremity movements, in the three-dimensional workspace. Finally, the use of model-based temporal features of FMG improved FMG-based grasp detection accuracy in the two-class grasp detection problem, paving the way for investigation into more advanced model-based feature extraction techniques. These promising results suggest that FMG based grasp-detection may indeed be suitable for monitoring grasping, in individuals with stroke, for rehabilitation applications.

5.3. Future Work

The results obtained from the studies executed in this thesis provide preliminary affirmation that the use of FMG-based grasp detection for detecting and encouraging

grasping in individuals with stroke is feasible. Despite these promising results, additional research questions and development challenges remain. A partial list of questions, which should be considered for future work, are summarized below.

5.3.1.   Experimental Protocol in an Uncontrolled Environment

The experimental protocols developed for both studies in this thesis were executed in a controlled environment. The protocols mandated a variety of grasp-and-move actions, and movements-without-grasping, with the intent of evaluating the robustness of the classifier to false-positives and false-negatives. However, an evaluation of FMG sensing in an uncontrolled environment, with participants engaging in ADL of their choosing, would be a key test for this technology. One method of achieving such a protocol would involve video recording participants as they go about completing ADL in the home setting, while wearing an FMG band. The output of the FMG classifier can then be compared to the occurrence of grasping in the video, by the experimenter.

5.3.2.   Further Evaluation with Individuals with Stroke

In this thesis, a preliminary investigation suggested that the acquisition and classification of the FMG signal in individuals with stroke, with mild-to-moderate upper-extremity impairments is feasible. In future studies, a more extensive evaluation of the feasibility of FMG classification in individuals with stroke is recommended. The accuracy of FMG when classifying FMG signals from individuals with stroke with moderate-to-severe upper-extremity impairments, who may be more susceptible to learned non-use, has yet to be conducted.

5.3.3.   Further Evaluation of Features of FMG

In this thesis, the utility of classifying temporal features of the FMG signal for grasp detection was investigated. Results obtained suggest that temporal features, that model trends of the data within a window, may improve FMG classification performance. In future studies, the use of more sophisticated model-based feature extraction methods should be evaluated. Additionally, the use of other classes of features, including measures of

entropy, complexity, frequency domain characterization, and spatial trend modeling should also be evaluated.

In this thesis, the classification performance with individual features was considered. It is possible that the combination of two, or more, features would have resulted in increased accuracy. In future studies, the accuracy achievable with a combination of feature values resulting from various feature extraction techniques should be evaluated. Additionally, the use of feature selection methods to select a partial set of the feature values produced by a feature extraction step, in order to increase accuracy, should also be evaluated.

5.3.4. Investigation of Methods to Reduce Training and Set-up Time

In real-world scenarios, FMG activity tracking devices would need to be donned and removed as part of daily use. The use of semi-supervised and unsupervised training paradigms may reduce the set-up time and effort required upon each donning of FMG-based devices. An evaluation of the ability to deploy activity tracking devices with minimal set-up is likely to be a pivotal step in their adoption in the field. The ability to achieve FMG grasp detection with semi-supervised and unsupervised training paradigms, is hence, a recommended future objective.

References

[1] G. E. Gresham, P. W. Duncan and W. B. Stason, Post-Stroke Rehabilitation: Clinical Practice Guideline, Rockville: Agency for Health Care Policy and Research, 2004.

[2] WHO, "Neurological Disorders public health challenges," World Health Organization, Switzerland, 2006.

[3] D. A. Nowak, C. Grefkes, M. Dafotakis, J. Küst, H. Karbe and G. R. Fink, "Dexterity is impaired at both hands following unilateral subcortical middle cerebral artery stroke," *European Journal of Neuroscience,* vol. 25, no. 10, pp. 3173-84, 2007.

[4] G. Kwakkel and B. Kollen, "Predicting improvement in the upper paretic limb after stroke: A longitudinal prospective study," *Restorative Neurology and Neuroscience,* vol. 25, pp. 453-60, 2007.

[5] E. Taub, G. Uswatte and D. Morris, "The learned nonuse phenomenon: implications for rehabilitation," *Europa medicophysica,* vol. 42, pp. 241-55, 2006.

[6] B. H. Dobkin, "Statergies for stroke rehabilitation," *The Lancet Neurology,* vol. 3, pp. 528-36, 2004.

[7] Y. Murata, N. Higo, T. Oishi, A. Yamashita, K. Matsuda, M. Hayashi and S. Yamane, "Effects of motor training on the recovery of manual dexterity after primary motor cortex lesion in macaque monkeys," *Journal of Nuerophysiology,* vol. 99, no. 2, pp. 773-86, 2008.

[8] R. Nudo, B. Wise, F. SiFuentes and G. Milliken, "Neural substrates for the effects of rehabilitative training on motor recovery after ischemic infarct," *Science,* vol. 272, no. 5269, pp. 1791-4, 1996.

[9] J. K. Yao, "New Developments in Stroke Care in BC," Stroke Recovery BC, 2011.

[10] Canadian Stroke Network, "The Quality of Stroke Care in Canada," Canadian
       Stroke Network, 2011.

[11] Heart & Stroke Foundation, "Statistics," Heart & Stroke Foundation, 2012.
       [Online]. Available:
       http://www.heartandstroke.com/site/c.ikIQLcMWJtE/b.3483991/k.34A8/Sta
       tistics.htm#strokeprevalence.

[12] A. E. Kunst, M. Amiri and F. Janssen, "The Decline in Stroke Mortality Exploration
       of Future Trends in 7 Western European Countries," *Stroke,* vol. 42, no. 8,
       pp. 2126-30, 2011.

[13] H. Krebs, B. Volpe, M. Aisen and N. Hogan, "Increasing Productivity and Quality
       of Care: Robot-Aided Neuro-Rehabilitation," *Journal of Rehabilitation
       Research and Development,* vol. 37, no. 6, pp. 639-52, 2000.

[14] J. E. Harris, J. J. Eng, W. C. Miller and A. S. Dawson, "A Self-Administered
       Graded Repetitive Arm Supplementary Program (GRASP) Improves Arm
       Function During Inpatient Stroke Rehabilitation," *Stroke,* vol. 40, pp. 2123-
       8, 2009.

[15] G. E. Gresham, D. Alexander, D. S. Bishop, C. Giuliani, G. Goldberg, A. Holland,
       M. Kelly-Hayes, R. T. Linn, E. J. Roth, W. B. Stason and C. A. Trombly,
       "Rehabilitation," *Stroke,* vol. 28, no. 7, pp. 1522-6, 1997.

[16] D. Yungher and W. Craelius, "Improving fine motor function after brain injury
       using gesture recognition biofeedback," *Disability and Rehabilitation:
       Assitive Technology,* vol. 7, no. 6, pp. 464-8, 2012.

[17] A. Pollock, S. E. Farmer, M. C. Brady, P. Langhorne, G. E. Mead, J. Mehrholz and
       F. v. Wijck, "Interventions for improving upper limb function after stroke,"
       Cochrane Database of Systematic Reviews, 2014.

[18] W. Ting and T. S. Huang, "Vision-based gesture recognition: A review," in
       International Gesture Workshop, Berline Heidelberg, 1999.

[19] H. Zhou and H. Hu, "Human motion tracking for rehabiliation - A survey,"
       Biomedical Signal Processing and Control, vol. 3, no. 1, pp. 1-18, 2008.

[20]   G. Uswatte, W. L. Foo, H. Olmstead, K. Lopez, A. Holand and L. B. Simms, "Ambulatory Monitoring of Arm Movement Using Accelerometry: An Objective Measure of Upper-Extremity Rehabilitation in Persons With Chronic Stroke," *Archives of physical medicine and rehabilitation,* vol. 86, pp. 1498-501, 2005.

[21]   G. Uswatte, C. Giuliani, C. Winstein, A. Zeringue, L. Hobbs and S. L. Wolf, "Validity of Accelerometry for Monitoring Real-Wold Arm Acitivity in Patients with Subactute Stroke: Evidence From the Exremity Constraint-Induced Therapy Evaluation Trial," *Archives of Physical Medicine and Rehabilitation,* vol. 87, no. 10, pp. 1340-5, 2006.

[22]   G. Uswatte, W. H. Miltner, B. Foo, M. Varma, S. Moran and E. Taub, "Objetive Measurement of Functional Upper-Extremity Movement Using Accelerometer Recordings Transformed with a Threshold Filter," *Stroke,* vol. 31, no. 3, pp. 662-7, 2000.

[23]   N. Friedman, J. B. Rowe and D. J. Reinkensmeyer, "The Manumeter: A Wearable Device for Monitoring Daily Use of the Wrist and Fingers," *IEEE Journal of Biomedical and Healthy Informatics,* vol. 18, no. 6, pp. 1804-12, 2014.

[24]   L. Dipetro, A. M. Sabatini and P. Dario, "A Survey of Glove-Based Systems and their Applications," *IEEE Transactions of Systems, Man, and Cybernetics,* vol. 38, no. 4, pp. 461-82, 2008.

[25]   R. H. Chowdhury, M. B. Reaz, M. A. Ali, A. A. Bakar, K. Chellappan and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors (Basel),* vol. 13, no. 9, pp. 12431-66, 2013.

[26]   T. G. Smith and M. J. Stokes, "Technical aspects of acoustic myography (AMG) of human skeletal muscle: contact pressure and force/AMG relationships," *Journal of Neuroscience Methods,* vol. 47, pp. 85-92, 1993.

[27]   M. Wininger, N.-H. Kim and William Craelius, "Pressure Signature of Forearm as Predictor of Grip Force," *Journal of Rehabilitation Research & Development,* vol. 45, no. 6, pp. 883-92, 2008.

[28]   N. Li, D. Yang, L. Jiang, H. Liu and H. Cai, "Combined Use of FSR Sensor Array and SVM Classifier for Finger Motion Recognition Based on Pressure Distribution Map," *Journal of Bionic Engineering,* vol. 9, pp. 39-47, 2012.

[29] A. Dementyev and J. A. Paradiso, "WristFlex: Low-Power Gesture Input with Wrist-Worn Presusre Sensors," *Proceedings of the 27th annual ACM symoposium on User Interface Software and Technology,* pp. 161-6, 2014.

[30] C. Castellini and V. Ravindra, "A Wearable Low-Cost Device Based Upon Force-Sensing Resistors to Detect Single-Finger Forces," in *5th IEEE RAS & EMBC International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, 2014.

[31] G. Sadarangani and C. Menon, "A Wearable Sensor System for Rehabilitation Applications," in *IEEE International Conference on Rehabilitation Robotics*, Singapore, 2015.

[32] Z. G. Xiao and C. Menon, "Towards the Development of a Wearable Feedback System for Monitoring the Activities of the Upper-Extremities," *Journal of Neuroengineering and Rehabilitation,* vol. 11, no. 2, 2014.

[33] R. B. Ambar, H. B. M. Poad, A. M. B. M. Ali, M. S. B. Ahmad and M. M. B. A. Jamil, "Multi-sensor Arm Rehabilitation Monitoring Device," in *International Conference on Biomedical Engineering*, Penang, 2012.

[34] D. Yungher and W. Craelius, "Discriminating 6 grasps using force myography of the forearm," in *BMES Annual Fall Meeting*, 2006.

[35] I. M. Bullock, J. Z. Zheng, S. D. L. Rosa, C. Guertler and A. M. Dollar, "Grasp Frequency and Usage in Daily Household and Machine Shope Tasks," *IEEE Transactions on Haptics,* vol. 6, no. 3, pp. 296-308, 2013.

[36] H. K. Yap, A. Mao, J. C. H. Goh and C.-H. Yeow, "Design of A Wearable FMG Sensing System for User Intent Detection during Hand Rehabilitation with a Soft Roboti Glove," in *IEEE International Conference on Biomedical Reobotics and Biomechatronics*, Singapore, 2016.

[37] C. L. MacKenzie and T. Iberall, The Grasping Hand, Elsevier, 1994.

[38] ActiGraph, "ActiGraph wGT3X-BT," ActiGraph, [Online]. Available: http://actigraphcorp.com/products-showcase/activity-monitors/actigraph-wgt3x-bt/. [Accessed 23 12 2016].

[39] D. Rand and J. J. Eng, "Arm-Hand Usage in Healthy Older Adults," *The Amerian Jouranl of Occupational Therapy,* vol. 64, no. 6, pp. 877-85, 2010.

[40] C. V. Granger, "The Emerging Science of Functional Assessment: our Tool for Outcomes Analysis," *Archives of Physical Medicine and Rehabiliation,* vol. 79, no. 3, pp. 235-240, 1998.

[41] J. B. Rowe, N. Friedman, M. Bachman and D. J. Reinkensmeyer, "The Manumeter: A non-obtrusive wearable device for monitoring spontaneous use of the wrist and fingers," in *IEEE International Conference on Rehabilitation Robotics*, Seattle, 2013.

[42] CyberGlove Systems, "CyberGlove II," CyberGlove, [Online]. Available: http://www.cyberglovesystems.com/cyberglove-ii/#photos. [Accessed 23 12 2016].

[43] X. Zhang and P. Zhou, "High-Density Myoelectric Pattern Recognition Toward Improved Stroke Rehabiliation," *IEEE Transactions on Biomedical Engineering,* vol. 59, no. 6, pp. 1649-57, 2012.

[44] S. W. Lee, K. M. Wilson, B. A. Lock and D. G. Kamper, "Subject-Specific Myoelectric Pattern Classification of Functional Hand Movements for Stroke Survivors," *IEEE Transations on Neural Systems and Rehabilitation Engineering,* vol. 19, no. 5, pp. 558-66, 2011.

[45] Y. Li, X. Chen, X. Zhang and P. Zhou, "Several Practical Issues Toward Implementing Myoelectric Pattern Reognition for Stroke Rehabilitation," *Medical Engineering & Physics,* vol. 36, pp. 754-60, 2014.

[46] J. Wang, L. Tang and J. E. Brounlund, "Surface EMG Signal Ampliciation and Filtering," *International Journal of Computer Applications,* vol. 82, no. 1, pp. 15-22, 2013.

[47] R. Merletti, A. Rainoldi and D. Farina, "Surface Electromyography for Noninvasive Characterization of Muscle," *Exercise and Sport Sciences Reviews,* vol. 29, pp. 20-25, 2001.

[48] H. J. Hermens, B. Freriks, C. Disselhorst-Klug and G. Rau, "Development of recommendations for SEMG sensors and sensor placement procedures," *Journal of Electromyography and Kinesiology,* vol. 10, no. 5, pp. 361-74, 2000.

[49] Thalmic Labs, "Tech Spec | Myo Battery Life, Dimensions, Compatibility and More," Thalmic Labs, [Online]. Available: https://www.myo.com/techspecs. [Accessed 23 12 2016].

[50] M. A. Islam, K. Subdaraj, R. B. Ahmad, N. U. Ahamed and M. A. Ali, "Mechanomyography Sensor Development, Related Singal Processing, and Applications: A Systematic Review," *IEEE Sensors Journal,* vol. 13, no. 7, pp. 2499-516, 2013.

[51] R. L. Abboudi, C. A. Glass, N. A. Newby, J. A. Flint and W. Craelius, "A Biomimetic Controller for a Multifinger Prosthesis," *IEEE Transcations on Rehabilitation Engineering,* vol. 7, no. 2, pp. 121-9, 1999.

[52] M. Sakr and C. Menon, "Regressing force-myographic signals collected by an armband to estimate torque exerted by the wrist: A preliminary investigation," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Vancouver, 2016.

[53] M. Sakr and C. Menon, "On the estimation of isometric wrist/forearm torque about three axes using Force Myography," in *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechantronics*, Singapore, 2016.

[54] E. Cho, R. Chen, L.-K. Merhi, Z. Xiao, B. Pousett and C. Menon, "Force Myography to Control Robotic Upper Extremity Prostheses: A Feasability Study," *frontiers in Bioengineering and Biotechnology,* vol. 4, no. 18, 2016.

[55] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[56] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Frontiers in Artificial Intelligence and Applications,* pp. 3-24, 2007.

[57] A. Kadkhodayan, X. Jiang and C. Menon, "Continuous Prediction of Finger Movements Using Force Myography," *Journal of Medical and Biological Engineering,* pp. 594-604, 2016.

[58] X. Zhu, "Semi-Supervised Learning," in *Encyclopedia of Machine Learning*, Springer, 2010, pp. 892-7.

[59] Interlink Electronics, "FSR 400 Series Data Sheet," Interlink Electronics, [Online]. Available: http://www.interlinkelectronics.com/FSR402.php. [Accessed 01 03 2015].

[60] A. Ferrone, F. Maita, L. Maiolo, M. Arquilla, A. Castiello, A. Pecora, X. Jiang, C. Menon and L. Colace, "Wearable Band for Hand Gesture Recognition based on Strain Sensors," in *6th IEEE RA/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2016.

[61] C. Pang, G.-Y. Lee, T.-i. Kim, S. M. Kim, H. N. Kim, S.-H. Ahn and K.-Y. Suh, "A flexible and highly sensitive strain-gauge sensor using reversible interlocking of nanofibres," *Nature Materials,* vol. 11, pp. 795-801, 2012.

[62] E. Morganti, L. Angelini, A. Adami, D. Lalanne, L. Lorenzelli and E. Mugellini, "A smart watch wtich embedded sensors to recognize objects, grasps and forearm gestures," *International Symposium on Robotics and Intelligent Sensors,* vol. 41, pp. 1169-75, 2012.

[63] R. W. Bohannon, "Relationship between static strength and various other measures in hemiparetic stroke patients," *International Rehabilitation Medicine,* vol. 8, no. 3, pp. 125-8, 1987.

[64] C. Watkins, M. Leathley, A. Moore, T. Smith and A. Sharma, "Prevalance of spatiscity post stroke," *Clinical Rehabilitation,* vol. 16, pp. 515-22, 2002.

[65] D. G. Kamper, A. N. Mckenna-Cole, L. E. Kahn and D. J. Reinkensmeyer, "Alterations in Reaching After Stroke and Their Relation to Movement Direction and Impairment Severity," *Archives in Physical Medicine and Rehabilitation,* vol. 83, pp. 702-7, 2002.

[66] B. D. Fulcher and N. S. Jones, "Highly Comparative Feature-Based Time-Series Classification," *IEEE Transactions in Knowledge and Data Engineering,* vol. 26, no. 12, pp. 3026-37, 2014.

[67] M. Zardoshti-Kermani, B. Wheeler, K. Badier and R. Hashemi, "Classification of the myoelectric signal using time-frequency based representation," *Medical Engineering ,* vol. 3, no. 4, pp. 324-33, 1995.

[68] National Instruments, "USB-6000 - National Instruments," National Instruments, [Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/211872. [Accessed 20 03 2016].

[69] C.-C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, p. 27, 2011.

[70] MathWorks, "fitcdiscr," MathWorks, [Online]. Available: http://www.mathworks.com/help/stats/fitcdiscr.html. [Accessed 28 09 2015].

[71] D. A. Heldman, J. Jankovic, D. E. Vaillancourt, J. Prodoehl, R. J. Elble and J. P. Giuffrida, "Essential tremor quantification during activities of daily living," *Parkinsonism and Related Disorders,* vol. 17, pp. 537-42, 2011.

[72] National Instruments, "LabVIEW System Design Software," National Instruments, [Online]. Available: http://www.ni.com/labview/. [Accessed 19 01 2016].

[73] M. R. Cutkosky, "On Grasp Choice, Grasp Models, and the Design of Hands for Manufacturing Tasks," *IEEE Transactions on Robotics and Automation,* vol. 5, no. 3, pp. 269-79, 1989.

[74] V. Mathiowetz, G. Volland, N. Kashman and K. Weber, "Adult Norms for the Box and Blocks Test of Manual Dexterity," *American Journal of Occupational Therapy,* vol. 39, pp. 396-91, 1985.

[75] B. Hudgins, P. Parker and R. N. Scott, "A New Statergy for Multigunction Myoelectric Control," *IEEE Transactions of Biomedical Engineering,* vol. 40, no. 1, 1993.

[76]   N. Alves and T. Chau, "Recognition of forearm muscle activity by continuous classification of multi-site mechanomyogram signals," in *International Coference of the IEEE EMBS*, Buenos Aires, 2010.

[77]   P. Madeleine, P. Bajaj, K. Søgaard and L. Ardengt-Nielsen, "Mechanomyography and electromyohraphy force relationships during concentric isometric eccentric contractions," *Journal of Electromyogprahy and Kiensiology,* vol. 11, no. 2, pp. 113-21, 2001.

[78]   Z. O. Khokar, Z. G. Xiao and C. Menon, "Surface EMG pattern recognition for real-time control of a wrist exoskeleton," *BioMedical Engineering OnLine,* vol. 9, no. 1, 2010.

[79]   MathWorks, "Polyfit," MathWorks, [Online]. Available: http://www.mathworks.com/help/matlab/ref/polyfit.html. [Accessed 05 06 2016].

[80]   M. Kumar, M. K. Jundal and R. K. Sharma, "Weka-Based Classification Techniques for Offline Handwritten Gurmukhi Character Recongition," in *International Conference on Soft Computing for Problem Solving*, 2014.

[81]   MathWorks, "www.mathworks.com," MathWorks, [Online]. Available: http://www.mathworks.com/products/matlab/. [Accessed 16 01 2016].

[82]   I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-82, 2003.

[83]   R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence ,* vol. 97, pp. 273-324, 1997.

[84]   R. Caruana and D. Freitag, "Greedy Attribute Selection," *International Conference on Machine Learning,* pp. 28-36, 1994.

[85]   A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition,* vol. 30, no. 7, pp. 1145-59, 1997.

Appendix A.

Correlation Coefficient between Classification Accuracy and Severity of Impairment in Individuals with Stroke

In this section the results of an analysis conducted to investigate the correlation between the classification accuracy obtained and the severity of impairment in individuals with stroke is described. A high degree of correlation could indicate that FMG grasp-detection performance is dependent on the level of impairment in individuals with stroke, and potentially limit the application of FMG sensing to a subset of the proposed population.

The correlation coefficient between the classification accuracy and severity of impairment, for individuals with stroke, was calculated with an alpha of 0.05. The ratio of the Box and Blocks score for the paretic limb to the non-paretic limb was used as a measure of impairment, as it provides a more granular scale when compared to the Chedoke scale. The scatter plots of the classification accuracy obtained for each task, with the SVM and LDA classifiers, versus the severity of impairment are depicted in Figure A-1 to Figure A-6. The correlation coefficient between the average classification accuracy, across all three tasks, and severity of impairment is 0.4468 (p = 0.2671) for the RBF-SVM, and 0.2782 (P = 0.5047) for the LDA, indicating a low likelihood of correlation between classification accuracy and severity of impairment.
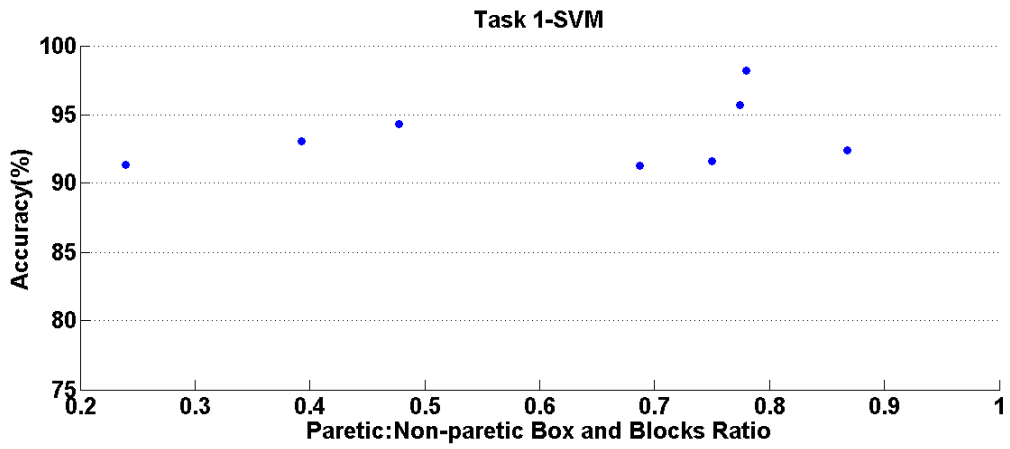
Figure A-1: Scatter plot of SVM classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 1
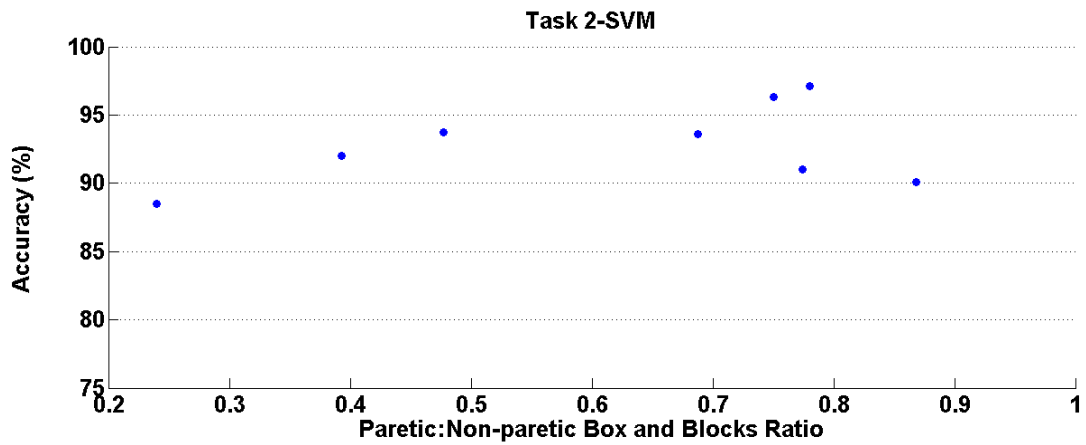
Figure A-2: Scatter plot of SVM classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 2
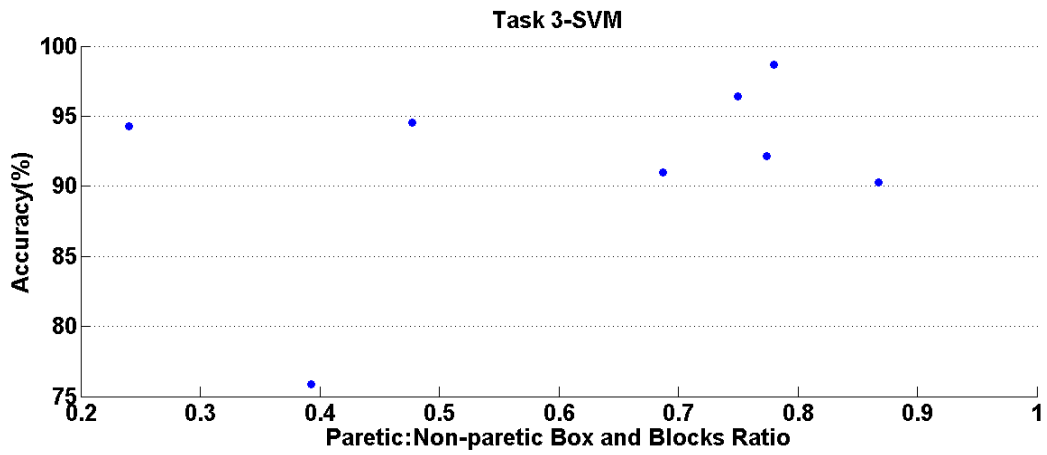
Figure A-3: Scatter plot of SVM classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 3
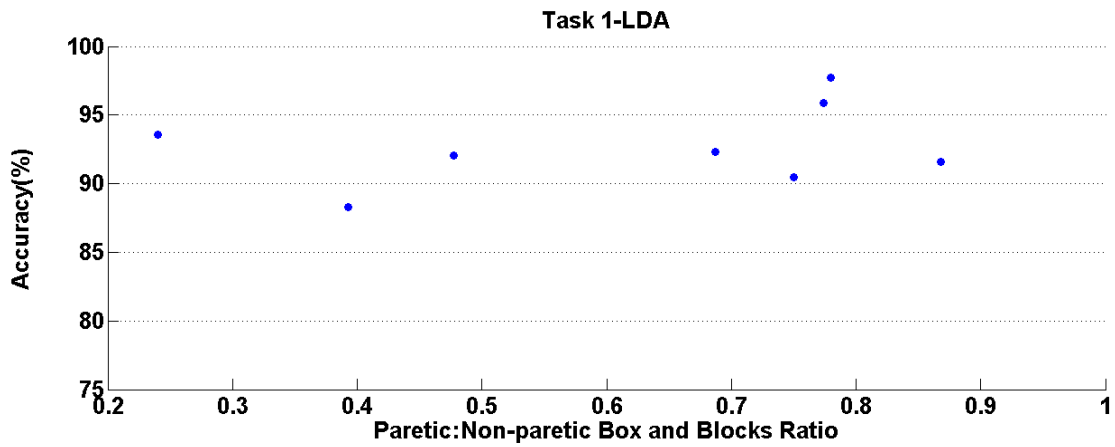

Figure A-4: Scatter plot of LDA classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 1
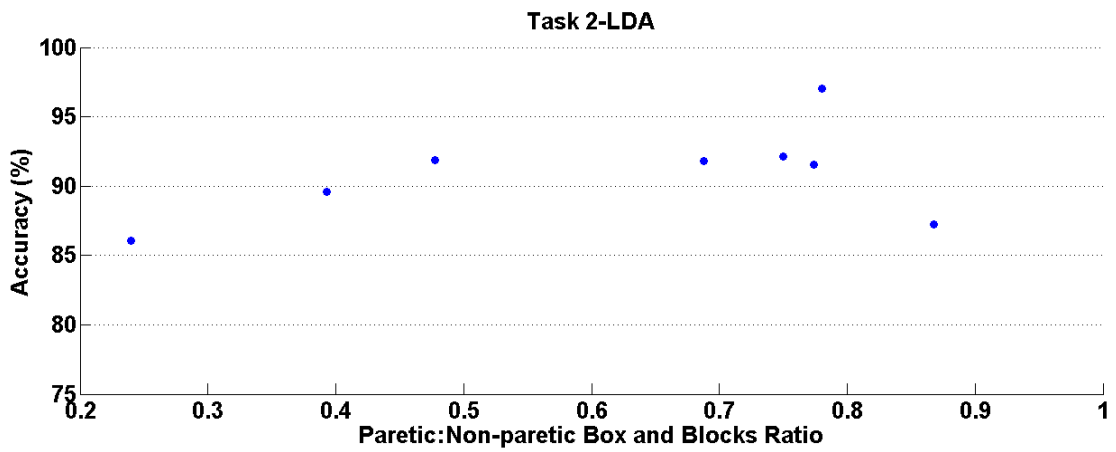
Figure A-5: Scatter plot of LDA classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 2
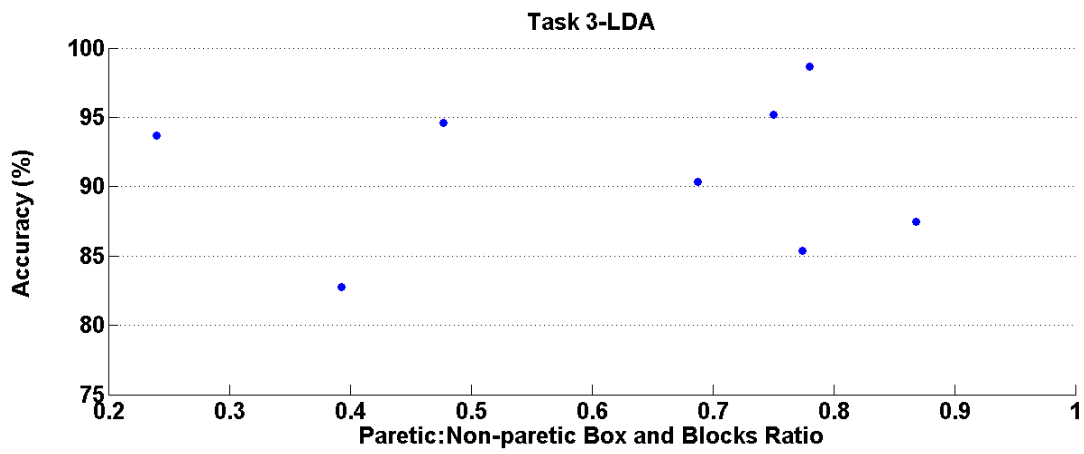
Figure A-6: Scatter plot of LDA classification accuracy versus Paretic: Non-paretic Box and Blocks ratio for Task 3

Appendix B.

Instructions to Participants for Protocol in Section 4.3.3

Instructions are summarized in Table B-1, below.

Table B-1: Instructions to participants

Type	Grasp Type & Object	Instructions to Participant
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.
Grasp and Move	Medium Wrap: Glass	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.

Type	Grasp Type & Object	Instructions to Participant
Grasp and Move	Precision Disk: Bowl	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.
Grasp and Move	Lateral Pinch: Quarter Plate	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.

Type	Grasp Type & Object	Instructions to Participant
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.
Grasp and Move	Tripod: Block	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position back, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.

Type	Grasp Type & Object	Instructions to Participant
Grasp and Move	Power Sphere: Ball	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.
Movement without Grasping	N/A	Move from neutral to origin position, and then to target position before returning to neutral. Subsequently, move from neutral to target position, and then to origin position, before returning to neutral. Repeat across 8 target locations, resulting in 16 movement actions.
Grasp and Move	Thumb, 2 Finger: Pen	Move from neutral to origin position, grasp and pick-up object, move object, place and release object at target position, before returning hand to neutral. Subsequently, move from neutral to previous target position, grasp and pick-up object, move object, place and release object at origin position, before returning hand to neutral. Repeat across 8 target locations, resulting in 16 grasp and move actions.

Appendix C.

Design and Implementation of Feature Evaluation Program

In this section, the design and implementation details of the feature evaluation program developed for the study in Chapter 4 are described. The high-level algorithmic flow of the program is described in section 4.3.4.

Design Goals and Constraints

The primary design goal was to create a feature evaluation program that is capable of calculating the classification performance associated with the features proposed in section 4.3.4. An additional design goal was to have the software capable of running on off-the-shelf computers with 64-bit operating systems, and a nominal memory size of 8 GB, such that, the program can be easily deployed for future FMG research.

Platform Selection

MATLAB® was selected as the computation platform, due to its native ability to execute matrix operations, and the large number of classification and statistical processing libraries available. The target platform was a PC running Windows 7 Professional (64-bit) on an Intel i7-4700MQ CPU with 2.40Ghz clock speed, and 8 GB of system memory.

Class Hierarchy

Figure C-1 depicts the class hierarchy for the feature evaluation software. In this section, the data structure, purpose, and associated methods for each class is described.

experiment_class

+ participants:participant_class
+cum_results:accumulated_results_class

participant_class

+ data_rounds:data_round_class
+results:participant_results_class

accumulated_results_class

+auc:<1x25>float
+auc_sd:<1x25>float
+acc:<1x25>float
+acc_sd:<1x25>float

participant_results_class

+auc<1x25>:float
+auc_sd<1x25>:float
+acc<1x25>:float
+acc_sd<1x25>:float

data_round_class

+raw_data:raw_data_class
+features:feature_data_class
+feature_results:feature_results_class

raw_data_class

+FMG:<16xn>int
+LABEL:<1xn>int

feature_data_class

+RAW_FMG:<16xn>int
+TEMPORAL_MAV_2_2:<16xn>int
+TEMPORAL_RMS_2_2:<16xn>int
+TEMPORAL_LINFIT_2_2:<32xn>int
+TEMPORAL_POLYFIT_2_2:<48xn>int
+TEMPORAL_MAV_0_4:<16xn>int
+TEMPORAL_RMS_0_4:<16xn>int
+TEMPORAL_LINFIT_0_4:<32xn>int
+TEMPORAL_POLYFIT_0_4:<48xn>int
+TEMPORAL_MAV_4_0:<16xn>int
+TEMPORAL_RMS_4_0:<16xn>int
+TEMPORAL_LINFIT_4_0:<32xn>int
+TEMPORAL_POLYFIT_4_0:<48xn>int
+TEMPORAL_MAV_1_1:<16xn>int
+TEMPORAL_RMS_1_1:<16xn>int
+TEMPORAL_LINFIT_1_1:<32xn>int
+TEMPORAL_POLYFIT_1_1:<48xn>int
+TEMPORAL_MAV_2_0:<16xn>int
+TEMPORAL_RMS_2_0:<16xn>int
+TEMPORAL_LINFIT_2_0:<32xn>int
+TEMPORAL_POLYFIT_2_0:<48xn>int
+TEMPORAL_MAV_0_2:<16xn>int
+TEMPORAL_RMS_0_2:<16xn>int
+TEMPORAL_LINFIT_0_2:<32xn>int
+TEMPORAL_POLYFIT_0_2:<48xn>int

feature_results_class

+grid_search_AUCs:<6x6>float
+cost:int
+gamma:int
+test_auc:float
+test_acc:float

Figure C-1: UML Class Hierarchy for Feature Evaluation Program

Experiment Class

The *experiment_class* is the overarching master class for the entire program, and contains all data associated with the study. It consists of ten objects of type *participant_class* (i.e. one for each participant), and one object of type *accumulated_results_class.*

87

Participant Class

The *participant_class* consists of all experimental data associated with an individual participant. Each instance of the class comprises three objects of type *data_round_class* (i.e. one for each round of data collection, for the given participant) and one object of type *participant_results_class*.

Data Round Class

The *data_round_class* contains all experimental data associated with a single round of data collection, for a single participant. Each instance of the *data_round_class* consists of one object of type *raw_data_class*, one object of type *feature_data_class*, and twenty-five objects of type *feature_results_class* (i.e. one for each feature evaluated and the raw FMG signal).

Raw Data Class

The *raw_data_class* consists of all FMG and label data collected for a single round of data collection, for a single participant. The raw FMG signal data is stored as a single object, named *FMG*, consisting of a two-dimensional array of *<16xn>* elements, where *n* is the number of samples taken for each of the sixteen FMG channels, dictated by the time the participant took to complete the round of instructions. The true label is stored in an object named *LABEL*, which is a *<1xn>* element array, where *n* is the number of samples taken from the validation sensor during the same period of data collection.

Feature Data Class

The *feature_data_class* stores data associated with all features extracted from the raw FMG signal, and the FMG signal as well. Data from the *feature_data_class* is used to evaluate the performance of each of the proposed features, and the raw FMG signal. The class consists of a single object of an *<mxn>* dimensional array, where m is the number of dimensions associated with the feature evaluated, and *n* is the number of samples collected during this particular round. The number of dimensions, *m*, ranges from sixteen to forty-eight, for the four features selected for evaluation in this study.

Feature Results Class

The *feature_results_class* is responsible for storing performance data associated with a single feature, for a particular round of data collection, for a particular participant. The class consists of multiple objects including: (i) a <6x6> array of floating point numbers that stores the AUCs associated with the grid-search for cost and gamma (named *grid_search_AUCs*), (ii) an integer that stores the optimal cost (named *cost*), found during the grid search, for this participant and round, (iii) an integer that stores the optimal gamma (named *gamma*), found during the grid search, for this participant and round, (iv) the AUC associated with the feature for this round (named *test_auc*), and (v) the accuracy associated with the feature for this round (named *test_acc*).

Participants Results Class

The *participants_results_class* is meant to store the classification performance associated with all evaluated features, for a given participant. An instance of the class contains four *<1x25>* floating point arrays: the average AUC (named *auc*), the standard deviation in AUC (named *auc_sd*), the average accuracy (named *acc*), and the standard deviation in accuracy (named *acc_sd*) associated with each feature, across all three rounds, for a given participant.

Accumulated Results Class

The *accumulated_results_class* is meant to contain all data associated with the classification performance for all evaluated features, across all participants. The *accumulated_results_class* is populated after the performance of a given feature has been evaluated for all participants (see Figure 4-13). An instance of the class contains four <1x25> arrays of floating point numbers for: (i) average AUC for each feature (named *auc*), (ii) standard deviation related to average AUC for each feature (named *auc_sd*), (iii) average Accuracy for each feature (named *acc*), (iv) standard deviation related to average accuracy for each feature (named *acc_sd*).

Source Code

This section includes the source code for all classes, functions and scripts that constitute the feature extraction program. Source code is presented in the form of code snippets. When necessary, files have been split into multiple snippets. Table C-1 provides a list of all files in the feature extraction program, and identifies the associated code snippets.

Table C-1: Summary of files (organized alphabetically)

File Name	File Type	Purpose	Code Snippets
accumulate_results	Script	Script that populates the *accumulated_results_class* object, using data from the *participant_results_class,* once all classification has been completed.	1
accumulated_results_class	Class Definition	Class definition for the *accumulated_results_class.*	2
classify	Script	Script that loops through the three rounds of data to determine the three-fold cross-validation accuracy associated will all features evaluated.	3-6
config_class	Class Definition	Class definition for the *config_class.*	7
configuration	Script	Script with configuration parameters for software (e.g. number of participants)	8
data_round_class	Class Definition	Class definition for the *data_round_class.*	9
evaluate_features	Script	Contains nested loops that loop through all participants, for each feature being evaluated.	10
experiment_class	Class Definition	Class definition for the *experiment_class.*	11

File Name	File Type	Purpose	Code Snippets
extract_features	Script	Contains loop that successively calls feature extraction functions for each round of data collection.	12
feature_data_class	Class Definition	Class definition for the *feature_data_class.*	13-15
feature_results_class	Class Definition	Class definition for the *feature_results_class.*	16
format_data	Script	Script that is responsible for importing experimental data, splitting data into rounds, and normalizing the signal from the validation sensor.	17
go_to_data_folder	Script	Script that switches MATLAB directory to a pre-specified data folder, such that the program can save extracted feature data to the file system (see Figure 4-13).	18
grid_search	Script	Contains nested loops that evaluate the performance of various cost and gamma values to determine the optimal cost and optimal gamma for the given round and participant.	19
main	Script	Execution start point. Loop that extracts features is within this script	20
participant_class	Class Definition	Class definition for the *participant_class.*	21
participant_results_class	Class Definition	Class definition for the *participant_results_class.*	22

File Name	File Type	Purpose	Code Snippets
raw_data_class	Class Definition	Class definition for the *raw_data_class.*	23
return_to_main_folder	Script	Script that returns MATLAB directory to the folder in which all functions/script code is stored. See *(go_to_data_folder).*	24
temporal_features_0_2	Function Definition	Definition for function to extract all feature types with widow {0,2}.	25
temporal_features_0_4	Function Definition	Definition for function to extract all feature types with widow {0,4}.	26
temporal_features_1_1	Function Definition	Definition for function to extract all feature types with widow {1,1}.	27
temporal_features_2_0	Function Definition	Definition for function to extract all feature types with widow {2,0}.	28
temporal_features_2_2	Function Definition	Definition for function to extract all feature types with widow {2,2}.	29
temporal_features_4_0	Function Definition	Definition for function to extract all feature types with widow {4,0}.	30

```
1     %Filename:accumulate_results.m
2     %
3     % Populates the accumulated_results_class object, using data from the
4     % participant_results_class, once all classification has been completed
5
6 -   for participant_idx =1:1:config.num_participants
7 -       accuracies(participant_idx)=experiment.participants...
8           (participant_idx).results.auc(idx_features);
9 -   end
10
11 -      experiment.cum_results.auc(idx_features)=mean(accuracies);
12 -      experiment.cum_results.auc_sd(idx_features)=std(accuracies);
13 -      accuracies=[];
14
15 -   for participant_idx =1:1:config.num_participants
16 -       accuracies(participant_idx)=experiment.participants...
17           (participant_idx).results.acc(idx_features);
18 -   end
19
20 -      experiment.cum_results.acc(idx_features)=mean(accuracies);
21 -      experiment.cum_results.acc_sd(idx_features)=std(accuracies);
22 -      accuracies=[];
23
```

Code Snippet 1 – *accumulate_results*

```
1     %Filename: accumulated_results_class.m
2     |
3     classdef accumulated_results_class
4         %stores performance data associated with all features, across all
5         %participants.
6
7         properties
8             auc
9             auc_sd
10            acc
11            acc_sd
12        end
13
14        methods
15            function cum_results=accumulated_results_class
16            auc=[];
17            auc_sd=[];
18            acc=[];
19            acc_sd=[];
20            end
21        end
22
23    end
24
25
```

Code Snippet 2 – *accumulated_results_class*

```
1       % Filename: classify.m
2       % Loops through the three rounds of data to determine the three-fold
3       % cross-validation accuracy associated will all features evaluated.
4
5  -    disp('classificiation starting');
6
7       %SVM Configuration
8  -    svm_type=0; %C-SVC SVM
9  -    kernel_type=2; %Linear Kernel
10 -    val_rounds=10;
11 -    cost=[0.01,0.1,1,10,100,1000];
12 -    gamma=[0.01,0.1,1,10,100,1000];
13
14      %3-fold cross-validation%
15 -    for idx_round = 1:3
16 -        TRAINING_DATA=[];
17 -        TRAINING_LABEL=[];
18 -        TESTING_DATA=[];
19 -        TESTING_LABEL=[];
20
21          %allocate testing and training data sets for this round
22 -        TESTING_DATA = experiment.participants(participant_idx).data_rounds...
23              (idx_round).features.get_feature(idx_features);
24 -        TESTING_LABEL = experiment.participants(participant_idx).data_rounds...
25              (idx_round).raw_data.LABEL;
26
```

Code Snippet 3 – *classify (snippet 1 of 4)*

```
27 -        switch idx_round
28 -            case 1
29 -                TRAINING_DATA = [experiment.participants(participant_idx...
30                      ).data_rounds(2).features.get_feature(idx_features...
31                      );experiment.participants(participant_idx...
32                      ).data_rounds(3).features.get_feature(idx_features)];
33 -                TRAINING_LABEL = [experiment.participants(participant_idx...
34                      ).data_rounds(2).raw_data.LABEL; ...
35                      experiment.participants(participant_idx...
36                      ).data_rounds(3).raw_data.LABEL];
37 -            case 2
38 -                TRAINING_DATA = [experiment.participants(participant_idx...
39                      ).data_rounds(1).features.get_feature(idx_features...
40                      );experiment.participants(participant_idx...
41                      ).data_rounds(3).features.get_feature(idx_features)];
42 -                TRAINING_LABEL = [experiment.participants(participant_idx...
43                      ).data_rounds(1).raw_data.LABEL;...
44                      experiment.participants(participant_idx...
45                      ).data_rounds(3).raw_data.LABEL];
46 -            case 3
47 -                TRAINING_DATA = [experiment.participants(participant_idx...
48                      ).data_rounds(1).features.get_feature(idx_features...
49                      );experiment.participants(participant_idx...
50                      ).data_rounds(2).features.get_feature(idx_features)];
51 -                TRAINING_LABEL = [experiment.participants(participant_idx...
52                      ).data_rounds(1).raw_data.LABEL; ...
53                      experiment.participants(participant_idx...
54                      ).data_rounds(2).raw_data.LABEL];
55 -        end
```

Code Snippet 4 – *classify (snippet 2 of 4)*

```
56
57 -       grid_search;
58
59 -       SVM = svmtrain(TRAINING_LABEL,TRAINING_DATA, ...
60           sprintf('-s %f -t %f -c %f -g %f -w1 %f -w2 %f -q ',svm_type, ...
61           kernel_type, cost(cost_idx), gamma(gamma_idx), 1, 1));
62
63 -       [PREDICTION,ACCURACY,PROBABILITIES]=...
64           svmpredict(TESTING_LABEL,TESTING_DATA,SVM,'-q');
65
66 -       [true_positive,false_positive,~,ROC_area]=...
67           perfcurve(TESTING_LABEL,PREDICTION,1);
68 -       experiment.participants(participant_idx).data_rounds(idx_round...
69           ).feature_results(idx_features).test_auc=ROC_area;
70 -       experiment.participants(participant_idx).data_rounds(idx_round...
71           ).feature_results(idx_features).test_acc=ACCURACY(1);
72 -   end
73 -   experiment.participants(participant_idx).results.auc(idx_features)=...
74       mean([experiment.participants(participant_idx...
75       ).data_rounds(1).feature_results(idx_features...
76       ).test_auc,experiment.participants(participant_idx...
77       ).data_rounds(2).feature_results(idx_features...
78       ).test_auc,experiment.participants(participant_idx...
79       ).data_rounds(3).feature_results(idx_features).test_auc]);
80
```

Code Snippet 5 – *classify (snippet 3 of 4)*

```
81 -   experiment.participants(participant_idx).results.auc_sd(idx_features...
82       )=std([experiment.participants(participant_idx...
83       ).data_rounds(1).feature_results(idx_features...
84       ).test_auc,experiment.participants(participant_idx...
85       ).data_rounds(2).feature_results(idx_features...
86       ).test_auc,experiment.participants(participant_idx...
87       ).data_rounds(3).feature_results(idx_features).test_auc]);
88
89 -   experiment.participants(participant_idx).results.acc(idx_features)=...
90       mean([experiment.participants(participant_idx...
91       ).data_rounds(1).feature_results(idx_features...
92       ).test_acc,experiment.participants(participant_idx...
93       ).data_rounds(2).feature_results(idx_features...
94       ).test_acc,experiment.participants(participant_idx...
95       ).data_rounds(3).feature_results(idx_features).test_acc]);
96
97 -   experiment.participants(participant_idx).results.acc_sd(idx_features...
98       )=std([experiment.participants(participant_idx...
99       ).data_rounds(1).feature_results(idx_features...
100      ).test_acc,experiment.participants(participant_idx...
101      ).data_rounds(2).feature_results(idx_features...
102      ).test_acc,experiment.participants(participant_idx...
103      ).data_rounds(3).feature_results(idx_features).test_acc]);
104 - disp('classificiation complete!');
105
```

Code Snippet 6 – *classify (snippet 4 of 4)*

```matlab
1        %Filename: config_class.m
2    classdef config_class
3        %stores configuration data.
4
5        properties
6
7            num_participants
8            num_features
9            label_channel
10        end
11
12        methods
13            function config = config_class
14                config.num_participants=[];
15                config.num_features=[];
16                config.label_channel=[];
17            end
18        end
19
20    end
21
```

Code Snippet 7 – *config_class*

```matlab
1        %Filename: configuration.m
2        %configuration settings
3
4        %Label Settings
5        config.label_channel=2; %channel on the label source
6        %Feature Evaluation Settings%
7        config.num_participants=10;
8        config.num_features=25;
9
```

Code Snippet 8 – *configuration*

```matlab
1       %Filename: data_round_class.m
2       classdef data_round_class
3           %stores the data associated with each round of data collection for the
4           %user
5
6           properties
7               raw_data
8               features
9               feature_results
10          end
11
12          methods
13              function data_round = data_round_class
14                  data_round.raw_data=raw_data_class;
15                  data_round.features=feature_data_class;
16                  data_round.feature_results=feature_results_class;
17              end
18
19          end
20
21      end
22
23
```

Code Snippet 9 – *data_round_class*

```matlab
1       %Filename:evaluate_features.m
2       %Nested loops that loop through all participants, for each feature being
3       %evaluated.
4
5       disp('Feature evaluation starting');
6
7       for idx_features = 1:config.num_features
8           for participant_idx=1:1:config.num_participants
9               clc
10              disp(idx_features);
11              disp(participant_idx);
12              classify;
13          end
14          go_to_data_folder;
15          save('experiment.mat','experiment');
16          return_to_main_folder;
17      end
18
19      accumulate_results;
20      go_to_data_folder;
21      save('experiment.mat','experiment');
22      return_to_main_folder;
23
```

Code Snippet 10 – *evaluate_features*

```matlab
1      %Filename:experiment_class
2
3      classdef experiment_class
4          %overarching class - stores all data associated with the study%
5          properties
6              participants
7              cum_results
8          end
9
10         methods
11             function experiment = experiment_class
12                 experiment.participants=participant_class;
13                 experiment.cum_results=accumulated_results_class;
14             end
15         end
16
17     end
18
19
```

Code Snippet 11 – *experiment_class*

```matlab
 1      %Filename:extract_features.m
 2      %Loop that successively calls feature extraction functions for each round
 3  -   %of data collection. disp('feature extraction starting');
 4
 5  -  for round_iter = 1:3
 6  -       experiment.participants(participant_idx).data_rounds(round_iter...
 7                ).features.RAW_FMG = experiment.participants(participant_idx...
 8                ).data_rounds(round_iter).raw_data.FMG;
 9
10  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
11               temporal_features_2_2(experiment.participants(participant_idx...
12               ).data_rounds(round_iter));
13
14  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
15               temporal_features_0_4(experiment.participants(participant_idx...
16               ).data_rounds(round_iter));
17
18  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
19               temporal_features_4_0(experiment.participants(participant_idx...
20               ).data_rounds(round_iter));
21
22  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
23               temporal_features_1_1(experiment.participants(participant_idx...
24               ).data_rounds(round_iter));
25
26  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
27               temporal_features_0_2(experiment.participants(participant_idx...
28               ).data_rounds(round_iter));
29
30  -       experiment.participants(participant_idx).data_rounds(round_iter) = ...
31               temporal_features_2_0(experiment.participants(participant_idx...
32               ).data_rounds(round_iter));
33  -  end
34
35
36  -  disp('feature extraction complete');
```

Code Snippet 12 – *extract_features*

```
1        %Filename:feature_data_class.m
2      classdef feature_data_class
3        %stores all feature data for a given round, for a given participant. %
4
5      properties
6          RAW_FMG
7
8          TEMPORAL_MAV_2_2
9          TEMPORAL_RMS_2_2
10         TEMPORAL_LINFIT_2_2
11         TEMPORAL_POLYFIT_2_2
12
13
14         TEMPORAL_MAV_0_4
15         TEMPORAL_RMS_0_4
16         TEMPORAL_LINFIT_0_4
17         TEMPORAL_POLYFIT_0_4
18
19
20         TEMPORAL_MAV_4_0
21         TEMPORAL_RMS_4_0
22         TEMPORAL_LINFIT_4_0
23         TEMPORAL_POLYFIT_4_0
24
25
26         TEMPORAL_MAV_1_1
27         TEMPORAL_RMS_1_1
28         TEMPORAL_LINFIT_1_1
29         TEMPORAL_POLYFIT_1_1
30
31
32         TEMPORAL_MAV_2_0
33         TEMPORAL_RMS_2_0
34         TEMPORAL_LINFIT_2_0
35         TEMPORAL_POLYFIT_2_0
36
37
38         TEMPORAL_MAV_0_2
39         TEMPORAL_RMS_0_2
40         TEMPORAL_LINFIT_0_2
41         TEMPORAL_POLYFIT_0_2
42
43         end
44
```

Code Snippet 13 – *feature_data_class*

```matlab
45          methods
46              function [feature,feature_idxs]= get_feature(obj,idx)
47                  switch idx
48                      case 1
49                          feature = obj.RAW_FMG;
50                          feature_idxs = ['Raw FMG'];
51                          % {2,2} window features
52                      case 2
53                          feature = obj.TEMPORAL_MAV_2_2;
54                          feature_idxs = ['TEMPORAL_MAV_2_2'];
55                      case 3
56                          feature = obj.TEMPORAL_RMS_2_2;
57                          feature_idxs = ['TEMPORAL_RMS_2_2'];
58                      case 4
59                          feature = obj.TEMPORAL_LINFIT_2_2;
60                          feature_idxs = ['TEMPORAL_LINFIT_2_2'];
61                      case 5
62                          feature = obj.TEMPORAL_POLYFIT_2_2;
63                          feature_idxs = ['TEMPORAL_POLYFIT_2_2'];
64                          % {0,4} window features
65                      case 6
66                          feature = obj.TEMPORAL_MAV_0_4;
67                          feature_idxs = ['TEMPORAL_MAV_0_4'];
68                      case 7
69                          feature = obj.TEMPORAL_RMS_0_4;
70                          feature_idxs = ['TEMPORAL_RMS_0_4'];
71                      case 8
72                          feature = obj.TEMPORAL_LINFIT_0_4;
73                          feature_idxs = ['TEMPORAL_LINFIT_0_4'];
74                      case 9
75                          feature = obj.TEMPORAL_POLYFIT_0_4;
76                          feature_idxs = ['TEMPORAL_POLYFIT_0_4'];
77                          %  {4,0} window features
78                      case 10
79                          feature = obj.TEMPORAL_MAV_4_0;
80                          feature_idxs = ['TEMPORAL_MAV_4_0'];
81                      case 11
82                          feature = obj.TEMPORAL_RMS_4_0;
83                          feature_idxs = ['TEMPORAL_RMS_4_0'];
84                      case 12
85                          feature = obj.TEMPORAL_LINFIT_4_0;
86                          feature_idxs = ['TEMPORAL_LINFIT_4_0'];
87                      case 13
88                          feature = obj.TEMPORAL_POLYFIT_4_0;
89                          feature_idxs = ['TEMPORAL_POLYFIT_4_0'];
```

Code Snippet 14 – *feature_data_class (snippet 1 of 2)*

```matlab
 90                              % {1,1} window features
 91                          case 14
 92                              feature = obj.TEMPORAL_MAV_1_1;
 93                              feature_idxs = ['TEMPORAL_MAV_1_1'];
 94                          case 15
 95                              feature = obj.TEMPORAL_RMS_1_1;
 96                              feature_idxs = ['TEMPORAL_RMS_1_1'];
 97                          case 16
 98                              feature = obj.TEMPORAL_LINFIT_1_1;
 99                              feature_idxs = ['TEMPORAL_LINFIT_1_1'];
100                              % {2,0} window features
101                              feature = obj.TEMPORAL_POLYFIT_1_1;
102                              feature_idxs = ['TEMPORAL_POLYFIT_1_1'];
103                              % {2,0} window features
104                          case 18
105                              feature = obj.TEMPORAL_MAV_2_0;
106                              feature_idxs = ['TEMPORAL_MAV_2_0'];
107                          case 19
108                              feature = obj.TEMPORAL_RMS_2_0;
109                              feature_idxs = ['TEMPORAL_RMS_2_0'];
110                          case 20
111                              feature = obj.TEMPORAL_LINFIT_2_0;
112                              feature_idxs = ['TEMPORAL_LINFIT_2_0'];
113                          case 21
114                              feature = obj.TEMPORAL_POLYFIT_2_0;
115                              feature_idxs = ['TEMPORAL_POLYFIT_2_0'];
116                              % {0,2} window features
117                          case 22
118                              feature = obj.TEMPORAL_MAV_0_2;
119                              feature_idxs = ['TEMPORAL_MAV_0_2'];
120                          case 23
121                              feature = obj.TEMPORAL_RMS_0_2;
122                              feature_idxs = ['TEMPORAL_RMS_0_2'];
123                          case 24
124                              feature = obj.TEMPORAL_LINFIT_0_2;
125                              feature_idxs = ['TEMPORAL_LINFIT_0_2'];
126                          case 25
127                              feature = obj.TEMPORAL_POLYFIT_0_2;
128                              feature_idxs = ['TEMPORAL_POLYFIT_0_2'];
129                              %
130                          otherwise
131                              disp('error')
132                      end
133
134          end
135
136      end
137  end
```

Code Snippet 15 – *feature_data_class (snippet 2 of 2)*

```matlab
1     %Filename:feature_results_class
2     classdef feature_results_class
3         %stores results associated with a given feature, for a given round, for
4         % a given participant.
5         properties
6             grid_search_AUCs
7             cost
8             gamma
9             test_auc
10            test_acc
11
12        end
13
14        methods
15            function feature_results=feature_results_class
16                grid_search_AUCs=[];
17                cost=[];
18                gamma=[];
19                test_auc=[];
20                test_acc=[];
21            end
22        end
23
24    end
25
26
```

Code Snippet 16 – *feature_results_class*

```
1       %Filename:format_data.m
2       %Imports experimental data, splits data into rounds, and normalizes
3       %the signal from the validation sensor.
4   -   disp('format data starting')
5
6       %Scale Data with respect to max register value%
7   -   SCALED_DAQ_LABEL=double((DAQ_LABEL/5)>0.03);
8   -   SCALED_BAND_LABEL=double((BAND_LABEL/1024)>0.1);
9   -   SCALED_FMG=FMG/1024;
10
11  -   SELECTED_LABEL=SCALED_BAND_LABEL(:,config.label_channel);
12
13      %split data into rounds of data collection
14  -   for round_iter = 1:3
15  -       RELEVANT_INDICES=[];
16  -       RELEVANT_INDICES = find(ROUND(:,1)==round_iter);
17  -       experiment.participants(participant_idx).data_rounds(round_iter...
18              ).raw_data.FMG = SCALED_FMG(RELEVANT_INDICES,:);
19  -       experiment.participants(participant_idx).data_rounds(round_iter...
20              ).raw_data.LABEL = SELECTED_LABEL(RELEVANT_INDICES);
21  -   end
22
23
24  -   disp('format data done!')
25
```

Code Snippet 17 – *format_data*

```
1       %Filename:go_to_data_folder.m
2       %navigates to data folder
3   -   cd ('C:\Users\Gautam Sadarangani\Desktop\data');
```

Code Snippet 18 – *go_to_data_folder*

```matlab
1      %Filename:grid_search.m
2      %6x6 grid search of gamma, cost
3 -    disp('grid search starting');
4
5
6 -    for cost_idx = 1: numel(cost)
7 -        for gamma_idx = 1:numel(gamma)
8 -            grid_search_ROC_areas(cost_idx,gamma_idx)=...
9                svmtrain(TRAINING_LABEL,TRAINING_DATA, ...
10               sprintf('-s %f -t %f  -c %f -g %f -w1 %f -w2 %f -v % f-q',...
11               svm_type, kernel_type, cost(cost_idx), gamma(gamma_idx),...
12               1, 1,val_rounds));
13 -        end
14 -    end
15 -    [~,rows]=max(grid_search_ROC_areas);
16 -    [~,col]=max(max(grid_search_ROC_areas));
17 -    row=rows(1,col);
18
19 -    gamma_idx=col;
20 -    cost_idx=row;
21 -    experiment.participants(participant_idx).data_rounds(idx_round...
22        ).feature_results(idx_features).cost=cost(cost_idx);
23 -    experiment.participants(participant_idx).data_rounds(idx_round...
24        ).feature_results(idx_features).gamma=gamma(gamma_idx);
25 -    experiment.participants(participant_idx).data_rounds(idx_round...
26        ).feature_results(idx_features).grid_search_AUCs=grid_search_ROC_areas;
27
28 -    disp('grid search complete!');
```

Code Snippet 19 – *grid_search*

```matlab
1      % Filename:main.m
2 -    clc;
3 -    clear;
4
5      %object initiation
6 -    config=config_class;
7 -    configuration;
8 -    experiment=experiment_class;
9
10     %first extract all features
11 -   for participant_idx=1:1:config.num_participants
12 -           go_to_data_folder;
13 -           load(sprintf('%d.mat',participant_idx));
14 -           return_to_main_folder;
15 -           format_data;
16 -           extract_features;
17 -           clearvars -except experiment participant_idx config
18 -   end
19     %store features %
20 -   go_to_data_folder;
21 -   save('experiment.mat','experiment');
22 -   return_to_main_folder;
23
24     %all features extracted, now evaluate features
25 -   evaluate_features;
26
27
```

Code Snippet 20 – *main*

```matlab
1      %Filename:participant_class.m
2      classdef participant_class
3      %stores data associated with any single participant in the study%
4
5          properties
6              data_rounds
7              results
8          end
9
10         methods
11             function participant = participant_class
12                     participant.data_rounds = data_round_class;
13                     participant.results=participant_results_class;
14             end
15         end
16
17     end
18
```

Code Snippet 21 – *participant_class*

107

```
1       %Filename:participant_results_class.m
2    ⊟ classdef participant_results_class
3          %stores data associated with all features for a given participant
4    ⊟      properties
5              auc
6              auc_sd
7              acc
8              acc_sd
9          end
10
11   ⊟      methods
12   ⊟          function results=participant_results_class
13 -               auc=[];
14 -               auc_sd=[];
15 -               acc=[];
16 -               acc_sd=[];
17 -           end
18          end
19
20   └ end
21
```

Code Snippet 22 – *participant_results_class*

```
1       %Filename:raw_data_class.m
2    ⊟ classdef raw_data_class
3          %stores raw data for each participant.
4
5    ⊟      properties
6              FMG
7              LABEL
8
9          end
10
11   ⊟      methods
12   ⊟          function raw_data=raw_data_class
13 -                 FMG=[];
14 -                 LABEL=[];
15 -             end
16          end
17
18   └ end
19
20
```

Code Snippet 23 – *raw_data_class*

```
1       %Filename:return_to_main_folder.m
2 -     cd ('C:\Users\Gautam Sadarangani\Desktop\code');
```

Code Snippet 24 – *return_to_main_folder*

```matlab
%Filename:temporal_features_0_2.m
function [data_round] = temporal_features_0_2(data_round)
%extracts all feature types for the {0,2} window.

epoch_left_swing=0;
epoch_right_swing=2;

ARRAY=data_round.features.RAW_FMG;

%create blank array to pad 0s on the edges of the features array
blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
EPOCH_MAV=blank_array;
EPOCH_RMS=blank_array;
EPOCH_TREND_SLOPE=blank_array;
EPOCH_TREND_OFFSET=blank_array;
EPOCH_POLY_TWO=blank_array;
EPOCH_POLY_ONE=blank_array;
EPOCH_POLY_ZERO=blank_array;


for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
    EPOCH = ...
        ARRAY((idx_samples-epoch_left_swing...
        ):(idx_samples+epoch_right_swing),:);
    for idx_cols = 1: size(ARRAY,2)
        EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
        EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
        temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
        EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
        EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
        temp=[];
        temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
        EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
        EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
        EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
        temp=[];
    end
end


data_round.features.TEMPORAL_MAV_0_2=EPOCH_MAV;
data_round.features.TEMPORAL_RMS_0_2=EPOCH_RMS;
data_round.features.TEMPORAL_LINFIT_0_2=...
    [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
data_round.features.TEMPORAL_POLYFIT_0_2=...
    [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
end
```

Code Snippet 25 – *temporal_features_0_2*

```
1      %Filename:temporal_features_0_4.m
2      function [data_round] = temporal_features_0_4(data_round)
3      %extracts all feature types for the {0,4} window.
4
5   -  epoch_left_swing=0;
6   -  epoch_right_swing=4;
7
8   -  ARRAY=data_round.features.RAW_FMG;
9
10     %create blank array to pad 0s on the edges of the features array
11  -  blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
12  -  EPOCH_MAV=blank_array;
13  -  EPOCH_RMS=blank_array;
14  -  EPOCH_TREND_SLOPE=blank_array;
15  -  EPOCH_TREND_OFFSET=blank_array;
16  -  EPOCH_POLY_TWO=blank_array;
17  -  EPOCH_POLY_ONE=blank_array;
18  -  EPOCH_POLY_ZERO=blank_array;
19
20
21  -  for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
22  -      EPOCH = ...
23             ARRAY((idx_samples-epoch_left_swing...
24             ):(idx_samples+epoch_right_swing),:);
25  -      for idx_cols = 1: size(ARRAY,2)
26  -          EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
27  -          EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
28  -          temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
29  -          EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
30  -          EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
31  -          temp=[];
32  -          temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
33  -          EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
34  -          EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
35  -          EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
36  -          temp=[];
37  -      end
38  -  end
39
40
41  -  data_round.features.TEMPORAL_MAV_0_4=EPOCH_MAV;
42  -  data_round.features.TEMPORAL_RMS_0_4=EPOCH_RMS;
43  -  data_round.features.TEMPORAL_LINFIT_0_4=...
44         [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
45  -  data_round.features.TEMPORAL_POLYFIT_0_4=...
46         [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
47  -  end
```

Code Snippet 26 – *temporal_features_0_4*

```matlab
 1     %Filename:temporal_features_1_1.m
 2     function [data_round] = temporal_features_1_1(data_round)
 3     %extracts all feature types for the {1,1} window.
 4
 5     epoch_left_swing=1;
 6     epoch_right_swing=1;
 7
 8     ARRAY=data_round.features.RAW_FMG;
 9
10     %create blank array to pad 0s on the edges of the features array
11     blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
12     EPOCH_MAV=blank_array;
13     EPOCH_RMS=blank_array;
14     EPOCH_TREND_SLOPE=blank_array;
15     EPOCH_TREND_OFFSET=blank_array;
16     EPOCH_POLY_TWO=blank_array;
17     EPOCH_POLY_ONE=blank_array;
18     EPOCH_POLY_ZERO=blank_array;
19
20
21     for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
22         EPOCH = ...
23             ARRAY((idx_samples-epoch_left_swing...
24             ):(idx_samples+epoch_right_swing),:);
25         for idx_cols = 1: size(ARRAY,2)
26             EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
27             EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
28             temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
29             EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
30             EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
31             temp=[];
32             temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
33             EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
34             EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
35             EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
36             temp=[];
37         end
38     end
39
40
41     data_round.features.TEMPORAL_MAV_1_1=EPOCH_MAV;
42     data_round.features.TEMPORAL_RMS_1_1=EPOCH_RMS;
43     data_round.features.TEMPORAL_LINFIT_1_1=...
44         [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
45     data_round.features.TEMPORAL_POLYFIT_1_1=...
46         [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
47     end
```

Code Snippet 27 – *temporal_features_1_1*

```matlab
1    %Filename:temporal_features_2_0.m
2    function [data_round] = temporal_features_2_0(data_round)
3    %extracts all feature types for the {2,0} window.
4
5    epoch_left_swing=2;
6    epoch_right_swing=0;
7
8    ARRAY=data_round.features.RAW_FMG;
9
10   %create blank array to pad 0s on the edges of the features array
11   blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
12   EPOCH_MAV=blank_array;
13   EPOCH_RMS=blank_array;
14   EPOCH_TREND_SLOPE=blank_array;
15   EPOCH_TREND_OFFSET=blank_array;
16   EPOCH_POLY_TWO=blank_array;
17   EPOCH_POLY_ONE=blank_array;
18   EPOCH_POLY_ZERO=blank_array;
19
20
21   for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
22       EPOCH = ...
23           ARRAY((idx_samples-epoch_left_swing...
24           ):(idx_samples+epoch_right_swing),:);
25       for idx_cols = 1: size(ARRAY,2)
26           EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
27           EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
28           temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
29           EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
30           EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
31           temp=[];
32           temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
33           EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
34           EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
35           EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
36           temp=[];
37       end
38   end
39
40
41   data_round.features.TEMPORAL_MAV_2_0=EPOCH_MAV;
42   data_round.features.TEMPORAL_RMS_2_0=EPOCH_RMS;
43   data_round.features.TEMPORAL_LINFIT_2_0=...
44       [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
45   data_round.features.TEMPORAL_POLYFIT_2_0=...
46       [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
47   end
```

Code Snippet 28 – *temporal_features_2_0*

```matlab
1       %Filename:temporal_features_2_2.m
2       function [data_round] = temporal_features_2_2(data_round)
3         %extracts all feature types for the {2,2} window.
4
5   -     epoch_left_swing=2;
6   -     epoch_right_swing=2;
7
8   -     ARRAY=data_round.features.RAW_FMG;
9
10        %create blank array to pad 0s on the edges of the features array
11  -     blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
12  -     EPOCH_MAV=blank_array;
13  -     EPOCH_RMS=blank_array;
14  -     EPOCH_TREND_SLOPE=blank_array;
15  -     EPOCH_TREND_OFFSET=blank_array;
16  -     EPOCH_POLY_TWO=blank_array;
17  -     EPOCH_POLY_ONE=blank_array;
18  -     EPOCH_POLY_ZERO=blank_array;
19
20
21  -     for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
22  -         EPOCH = ...
23               ARRAY((idx_samples-epoch_left_swing...
24               ):(idx_samples+epoch_right_swing),:);
25  -         for idx_cols = 1: size(ARRAY,2)
26  -             EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
27  -             EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
28  -             temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
29  -             EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
30  -             EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
31  -             temp=[];
32  -             temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
33  -             EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
34  -             EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
35  -             EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
36  -             temp=[];
37  -         end
38  -     end
39
40
41  -     data_round.features.TEMPORAL_MAV_2_2=EPOCH_MAV;
42  -     data_round.features.TEMPORAL_RMS_2_2=EPOCH_RMS;
43  -     data_round.features.TEMPORAL_LINFIT_2_2=...
44           [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
45  -     data_round.features.TEMPORAL_POLYFIT_2_2=...
46           [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
47  -     end
```

Code Snippet 29 – *temporal_features_2_2*

113

```matlab
1      %Filename:temporal_features_4_0.m
2      function [data_round] = temporal_features_4_0(data_round)
3      %extracts all feature types for the {4,0} window.
4
5  -   epoch_left_swing=4;
6  -   epoch_right_swing=0;
7
8  -   ARRAY=data_round.features.RAW_FMG;
9
10     %create blank array to pad 0s on the edges of the features array
11 -   blank_array(1:size(ARRAY,1),1:size(ARRAY,2))=0;
12 -   EPOCH_MAV=blank_array;
13 -   EPOCH_RMS=blank_array;
14 -   EPOCH_TREND_SLOPE=blank_array;
15 -   EPOCH_TREND_OFFSET=blank_array;
16 -   EPOCH_POLY_TWO=blank_array;
17 -   EPOCH_POLY_ONE=blank_array;
18 -   EPOCH_POLY_ZERO=blank_array;
19
20
21 -   for idx_samples=(1+epoch_left_swing):(size(ARRAY,1)-epoch_right_swing)
22 -       EPOCH = ...
23             ARRAY((idx_samples-epoch_left_swing...
24             ):(idx_samples+epoch_right_swing),:);
25 -       for idx_cols = 1: size(ARRAY,2)
26 -           EPOCH_MAV(idx_samples,idx_cols)= mean(abs(EPOCH(:,idx_cols)));
27 -           EPOCH_RMS(idx_samples,idx_cols)= sqrt(mean(EPOCH(:,idx_cols).^2));
28 -           temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),1);
29 -           EPOCH_TREND_SLOPE(idx_samples,idx_cols)=temp(1);
30 -           EPOCH_TREND_OFFSET(idx_samples,idx_cols)=temp(2);
31 -           temp=[];
32 -           temp= polyfit([1:size(EPOCH,1)]',EPOCH(:,idx_cols),2);
33 -           EPOCH_POLY_TWO(idx_samples,idx_cols)=temp(1);
34 -           EPOCH_POLY_ONE(idx_samples,idx_cols)=temp(2);
35 -           EPOCH_POLY_ZERO(idx_samples,idx_cols)=temp(3);
36 -           temp=[];
37 -       end
38 -   end
39
40
41 -   data_round.features.TEMPORAL_MAV_4_0=EPOCH_MAV;
42 -   data_round.features.TEMPORAL_RMS_4_0=EPOCH_RMS;
43 -   data_round.features.TEMPORAL_LINFIT_4_0=...
44         [EPOCH_TREND_SLOPE,EPOCH_TREND_OFFSET];
45 -   data_round.features.TEMPORAL_POLYFIT_4_0=...
46         [EPOCH_POLY_TWO,EPOCH_POLY_ONE,EPOCH_POLY_ZERO];
47 -   end
```

Code Snippet 30 – *temporal_features_4_0*

Appendix D.

Contributions

A list of publications that have been accepted, or submitted, as a result of the work associated with this thesis is provided below.

Submitted Refereed Journal Papers

1. G.P. Sadarangani, X. Jiang, L.A. Simpson, J.J. Eng, C. Menon, (submitted) "Force Myography for Monitoring Grasping in Individuals with Stroke with Mild to Moderate Upper-extremity Impairments: a Preliminary Investigation in a Controlled Environment," *Frontiers in bioengineering and biotechnology.*

2. G.P. Sadarangani, C. Menon, (submitted) "A preliminary investigation on the utility of temporal features of Force Myography in the two-class problem of grasp vs. no-grasp in the presence of upper-extremity movements," *BioMedical Engineering OnLine.*

Published/Accepted Refereed Conference Extended Abstracts

1. L. Simpson, J. Eng, G. Sadarangani, A. Hodgson, C. Menon, "Wearable devices to capture upper limb activity post stroke: clinician perceptions of their utility," *Canadian Stroke Congress*, Toronto, Canada, 2015.

2. G. Sadarangani, X. Jiang, Z. Xiao, L. Merhi, C. Menon, "A Wearable Band for Detecting Functional Use of the Upper-extremities in Rehabilitation Applications, Symposium on Rehabilitation Robotics and Human-Robot Interaction," *IEEE International Conference on Robotics and Automation,* Seattle, USA, 2015.

3. G.P. Sadarangani, X. Jiang, Z. Xiao, L. Merhi, C. Menon, "Wearable Sensor for Detecting Functional Use of the Upper-extremities in Take-Home or Tele-Rehabilitation Settings," 24th John K. Friesen *Conference*, Vancouver, Canada, 2015.

4. G. Sadarangani, X. Jiang, Z. Xiao, L. Merhi, C. Menon, "A Universal Band for Detecting Functional Use in Rehabilitation Applications," *GF Strong Day*, Vancouver, Canada, 2015.