MYOELECTRIC PATTERN RECOGNITION STATE-BASED CLASSIFICATION METHOD FOR IMPROVED DYNAMIC PERFORMANCE AND REAL-TIME USABILITY

by

Katerina Biron

B.Sc.E (ME) - Old Dominion University, 2007
M.Sc.E (EE) - University of New Brunswick, 2010

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of

Doctor of Philosophy

in the Graduate Academic Unit of Electrical and Computer Engineering

Supervisors: Kevin Englehart, Ph.D., Electrical and Computer Engineering
Erik Scheme, Ph.D., Electrical and Computer Engineering

Examining Board: Maryhelen Stevenson, Ph.D., Electrical and Computer Engineering
Brent Petersen, Ph.D., Electrical and Computer Engineering
Scott Batemen, Ph.D., Computer Science

External Examiner: Adrian Chan, Ph.D., Systems and Computer Engineering, Carleton University

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

February, 2017

ABSTRACT

Pattern-recognition-based myoelectric control systems have been shown to be accurate in controlled laboratory experiments where users are often restricted to perform discrete, segmented and constant force contractions. Such types of contractions are insufficient for real functional use that requires a controller to interpret both constant force contractions and dynamic transitions between motion classes. This explains why pattern recognition based myoelectric control systems have often been unreliable in real-world settings and the dynamic conditions required for functional use.

This work develops an original pattern recognition based myoelectric control system that improves system performance during dynamic transitions between motion classes. The proposed architecture uses feature estimate methods inspired by Kalman filters and enables different control strategies depending on whether users perform constant force contractions or dynamic transitions. The system was developed offline and was then evaluated in a real-time (user-in-the-loop) task to verify if it improved usability. For both offline and real-time analyses, the system was compared to state-of-the-art pattern recognition systems based on linear discriminant analysis.

The real-time results showed that the proposed system allowed users to perform significantly ($p < .05$) more tasks in a significantly ($p < .05$) less amount of time, and the proposed system also obtained a better speed-accuracy tradeoff, suggesting improved usability. Furthermore, the system improved the quality of descriptive features and allowed new users to better learn how to control pattern recognition based myoelectric control system.

Table of Contents

List of Tables

List of Figures

List of Acronyms

A_{CC}	*Steady-state classification accuracy*
ACE	*Acquisition and control environment*
ACV	*Adaptive cepstrum vector*
AER	*Active error rate*
ANN	*Artificial neural networks*
AR	*Auto-regressive*
ARMA	*Auto-regressive moving average*
D	*Target distance*
ED	*Euclidian Distance*
EMG	*Electromyogram*
FC rate	*Failed completion rate*
FI	*Frame increment*
FL	*Frame length*
FN	*False negative*
FP	*False positive*
FPR	*False positive rate*
ID	*Index of difficulty*
HMM	*Hidden Markov model*
KF	*Kalman filter*
LDA	*Linear discriminant analysis*
MAV	*Mean absolute value*
MBC	*Multiple binary classifier*
MMAE	*Midpoint moving average estimators*

MT	*Motion time*
MV	*Majority voting*
NN_C	*1-nearest neighboring class center*
PCA	*Principal component analysis*
ROC	*Receiving operating characteristics*
RMS	*Root mean square*
SE	*Standard Error*
SNR	*Signal-to-noise ratio*
SVM	*Support vector machine*
TAC	*Target Achievement Control*
TD	*Time domain features*
TER	*Total error rate*
TN	*True negative*
TP	*True positive*
TPR	*True positive rate*
TURNS	*Number of slope sign changes*
W	*Target width*
WL	*Wave length*
ZC	*Zero crossings*

List of Symbols

a	*Fitts' law linear y-intercept*
b	*Fitts' law linear slope*
$E[\cdot]$	*Expected value*
e	*Motion model Gaussian noise*
ED_{NN}	*Euclidian distance from the nearest neighboring class center*
d	*Measurement model Gaussian noise*
D_{ED}	*State-evaluator's output*
Δ	*Kalman filter innovation*
$\overline{ID}$	*Average index of difficulty*
k	*Update gain*
$\boldsymbol{K}$	*Update gain matrix*
KF_{euc}	*Kalman filter based feature conditioning method with Euclidian distance features*
KF_{td}	*Kalman filter based feature conditioning method with time domain features*
m	*Measured feature*
$\overrightarrow{\boldsymbol{m}}$	*Measured feature vector*
No-cond	*No feature conditioning*
$\overline{MT}$	*Average motion time*
Q	*Feature conditioning measurement model variance*
R	*Feature conditioning motion model variance*
R^2	*Linear regression coefficient of determination*
t	*Time*
S_{aggr}	*Simple feature conditioning method with aggressive updates*

S_{cons}	*Simple feature conditioning method with conservative updates*
T_d	*Window interval increments*
τ_N	*Time it takes users to reach N correct decisions*
T_τ	*Processing delay*
x	*Compliant feature*
$\bar{x}$	*Feature estimate*
x'	*Predicted feature*
$\vec{\bar{x}}$	*Feature estimate vector*
$\vec{x}$	*Compliant feature vector*
$\vec{x}'$	*Predicted feature vector*
$\vec{y}$	*Measurement vector*
x^{ED}	*Euclidian distance feature set*
$[x,y,r]$	*Cartesian axis coordinate of Fitts' law test virtual environment*
$[d_x\, d_y\, d_r]$	*Motion direction of the cursor (in Fitts' law test) along each axis*
V	*Motion speed of Fitts' law cursor*
Δt	*Time between two consecutive sample of the Fitts' law test*

Chapter 1 - Introduction

1.1 Motivation

Upper limb deficiency, either congenital or due to amputation, can restrict mobility. Some amputees may be able to function with high levels of independence, but developmental complications may occur in young children. Additionally, traumatic or surgical amputation may result in withdrawal, frustration, or physical and psychological stress due to the radical lifestyle changes needed to accommodate to the amputation.

To reduce the burden of amputation, various prosthetic limbs are available; these can be passive or powered. While passive devices do not provide much dynamic function, they can be aesthetically appealing and help body symmetry. Powered prostheses are either body powered using a cable-hook system driven by shoulder motion, or electrically powered using the muscle signals (myoelectric signals) as the control source. While body powered prostheses can restore some function, they can only actuate a limited number of degrees of freedom and they require shoulder mobility. Myoelectric prostheses are more versatile and can be both functional and aesthetically appealing.

Myoelectric prostheses are not without shortcomings. They have been available for decades, but commercially available systems have scarcely evolved [1] [2] [3]. In attempts to improve myoelectrically controlled prostheses, researchers have studied various pattern recognition techniques [4]. Although pattern recognition based myoelectric prostheses are accurate in controlled laboratory experiments [5], they are often unreliable when used in real-world settings for a number of reasons. One commonly reported reason is the unreliable performance during dynamic contractions required during functional use [5] [6] [7] [8].

In fact, several pattern recognition based systems capable of detecting users' intended motions have been proposed, but these have primarily been tested while users performed discrete, segmented and constant force contractions. Such types of contractions are insufficient for real functional use, suggesting that a controller should be able to interpret both sustained contractions (within a desired class) and dynamic transitions between motion classes.

1.2 Objectives

The general objective of this work was to improve the performance of myoelectric control systems during dynamic use. The specific objective was to develop a pattern recognition based myoelectric control system that improved the users' ability to transition between intended motion classes. Such a system:

- Must be able to detect the users intended motion while performing sustained (within class) contractions;
- Must correctly interpret within-class variation in contraction intensity, permitting proportional control;
- Must quickly respond to class transitions, and;
- Must be capable of making reliable decisions during class transitions.

Pattern recognition control systems that provide additional stability during class transitions have been developed, but these impose undesired delay or can result in false motion activation making it difficult for users to perform real-time tasks. This work addresses this issue by introducing a new hybrid pattern recognition based myoelectric control scheme that allows distinctive handling of within class contractions and between

class transitions. To develop such a system, this work evaluates methods to determine *when* users are transitioning and *how* the system should respond to class transitions.

1.3 Terminology

A number of the more commonly used terminologies in this thesis have been summarized in Table 1-1.

Table 1-1: List of terminology used throughout the work.

Class transition	*Transitory portion between two different contraction types.*
Compliant features	*Feature estimates that allow the system to be operationally compliant. In other words, the most favorable, yet unknown features.*
Contraction	*The action of activating one or more muscles to generate* a *desired motion or force.*
Contraction type	*The muscle contraction pattern resulting from eliciting a specific class of motion.*
Dynamic contraction	*A contraction that changes the state of the prosthesis. This includes, but is not limited to, a force varying contraction, or a class transition.*
Dynamic level (of a contraction)	*The degree of change in a contraction. For example, steady-state contractions have a low dynamic level, where as rapid class transitions have a high dynamic level.*
Electromyogram (EMG)	*A manifestation of the electrical activity of skeletal muscles.*
Extracted features	*Features obtained from data segments of processed EMG. In this work, they can be time domain features or Euclidian distance features* (*described later*).
Feature conditioning	*Processing applied before classification to reduce the signal-to-noise ratio (SNR) of extracted features.*
Feature estimates ($\vec{\vec{x}}$)	*Output of the feature conditioning methods.*
Frame increment (FI)	*The time increment between the start of two consecutive data windows.*

Frame length (FL)	*Length of data window used to extract features.*
Indeterminate state	*A muscle contraction (typically a class transition) that produces features that have a low probability of belonging to a trained class.*
Offline testing	*Analysis of data that has been pre-recorded. During data collection, users do not actively control the output of the system, and have no feedback.*
Measured features ($\vec{y}$)	*Output of the measurement model.*
Measurement model	*State of a system obtained by a "measurement device." For feature conditioning, the measurement model associates the extracted features to the current class decision.*
Motion model	*Temporal trajectory of the features. For feature conditioning, the motion model relates the current feature position to the previous feature position.*
Predicted features	*Output of the motion model.*
Proportional control	*Modulation of prostheses speed by the intensity of muscle contraction.*
Raw EMG	*Unprocessed EMG signal recorded from electrodes.*
Real-time testing	*Real-time testing typically incorporates some form of visual feedback to enable users to adjust the control output to meet a desired target.*
Reference configuration	*A widely implemented classification method against which novel methods will be compared.*
Response time	*Delay between the generation of EMG and the actuation of a control signal to the prosthesis.*
Signal-to-noise ratio (SNR)	*The ratio of the mean of the computed control signal to its standard deviation over a given timeframe.*
Steady state contractions	*Sustained, constant intensity muscle contractions within a class.*
Usability	*The speed, dexterity and ease of performing functional tasks.*

1.4 Thesis Organization

Chapter 1 defined the objectives of the work and provided an overview of terminology used throughout this document. Chapter 2 provides a literature review of myoelectric controls and research efforts to control dynamic contractions, as well as a brief review of the current assessment methods used to evaluate the performance of state-of-the art systems. Chapter 3 and 4 present the proposed hybrid control method as well as offline studies evaluating its performance. Chapter 5 describes a real-time experiment to test the hybrid classifier and compare the usability of the system to existing systems. Finally, Chapter 6 concludes the document.

Chapter 2 - Background

2.1 Conventional Myoelectric Control

The *electromyogram* (EMG) is a manifestation of the electrical activity of skeletal muscles. It has been shown that EMG activity increases in proportion to contraction force making it possible to monitor muscle activation [9].

The concept of using EMG to control myoelectric prostheses was introduced in the 1940s by a physics student at Munich University [10], but it was not until the late 1950s, when transistors emerged, that researchers around the world started developing myoelectric prostheses that could meet clinical requirements in size and weight. The first myoelectric prosthesis to see clinical use was developed in the USSR in the early 1960s [10]. There are now various commercially available upper limb myoelectric prostheses with various control strategies. Two conventional control strategies most often used are discussed below along with their available speed actuation strategies.

2.1.1 Dual Site Control

Dual site myoelectric control measures the EMG signal from two surface electrodes placed over antagonist muscle sites. The EMG signal recorded from one muscle site is used to activate one function, and the signal recorded from the other muscle site is used to activate another function. The control strategy uses basic signal processing techniques to extract an amplitude feature from the signal (either the mean absolute value (MAV) or the root mean square (RMS)) [11] and the prosthesis is activated when the amplitude of one channel, or difference between two channels, reaches a preset threshold value.

This strategy is similar to the one used in the first clinically viable system from the 1960s and is the most widely used [1]. Although conceptually simple, amputees must

learn how to isolate the contractions of two independent muscles, which are not always available as amputee muscles can be damaged or atrophied. Furthermore, control of any additional devices requires further independent muscle sites [1] or a mode switching strategy (activated by the residual muscles) such as a hardware switch or a "switch" activated by an EMG pulse.

2.1.2 Single Site Control

Single site control strategies [4] [12], use the EMG signal recorded from a single muscle site to actuate different functions. As with the method listed above, an amplitude feature is extracted from the signal and used to determine the desired state. Because amplitude features are proportional to the intensity of a muscular contraction [9], users can select different functions by changing their contraction levels. In a method developed and popularized by the Institute of Biomedical Engineering at the University of New Brunswick, the dynamic range of intensities is segmented into three sections (or states). In this 3-State Control system, little or no contraction results in no activation, while low level activations drive one function, and increased activation drives the other.

This strategy is useful when few residual muscles are available, but, as with dual site control, it assumes that users are capable of accurately and repeatedly generating the level of amplitude necessary to select the desired function [4] [12]. This amplitude coding is unnatural and requires additional cognitive burden; consequently such control is limited to three functions [3].

2.1.3 Actuation Speed and Proportional Control

Once a function is selected, the speed of the prostheses can be either constant (ON/OFF control) or varied (proportional control) [4]. While ON/OFF control is simple,

to improve the user's ability to control the prosthesis dynamically, the EMG amplitude feature (MAV or RMS) is often used to control the speed of the prosthesis [12]. This is intuitive, since the amplitude is roughly proportional to the intensity of muscular contraction [13]. Reports of mapping contraction strength to device force or speed date back to 1955 [14] and has been heavily investigated [15] [16] [17] [18] [19]. The most widely adopted method consists of estimating the EMG amplitude with a simple low-pass rectification approach [18] [19] (described in the next section (2.1.4)). Proportional control is offered by all manufacturers of commercial myoelectric prostheses [20]. Proportional control can be used with both dual site and single site control types; however, the intuitiveness of proportional control is most limited with single site control.

2.1.4 EMG Amplitude Estimation

The amplitude estimate must represent, as accurately as possible, the contraction intensity, especially in the case of control methods with proportional control as these must continuously represent the desired speed. This is also important for systems that attempt to control dynamic contractions as these require rapid response to users' contraction changes.

Various methods have been suggested to estimate the signal amplitude [4] [18] [21], but it is common to full-wave rectify the EMG [4] [21] and then filter the rectified signal with a low pass filter as illustrated in Figure 2-1.

It has been suggested that the filter that provides the best amplitude estimate while performing constant force contractions is a rectangular-window averaging filter which computes the arithmetic mean of a fixed time history of data [22] [23]. Such smoothing filter operates on the assumption that it is possible to smooth the signal's randomness

given a sufficiently long time history of data [24]. St-Amant *et al*. [25] found that, for constant-force contractions, the amplitude estimate improved (i.e. it was less variable) proportionally to the square root of the length of the averaging window. Since large averaging windows increase delay, the observation by St-Amant *et al*. [25] suggests that large averaging times results in slow response during contraction changes [22], whereas very short averaging times can drastically degrade the amplitude estimation. It has been reported that the optimal amplitude estimation (i.e. less varying amplitude with minimal delay perceived by the user) is achieved using averaging window that ranges between 250 ms [23] and 300 ms [26].

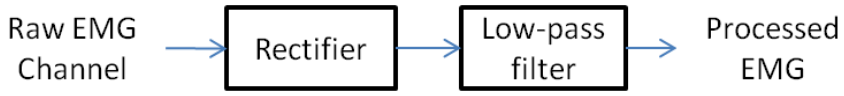

Figure 2-1: To estimate the EMG amplitude the raw signal is rectified than low-pass filtered. The amplitude estimate is often referred to as the processed EMG.

This is satisfactory provided that users perform constant force contractions where the EMG can be assumed stationary over the averaging window [27]. While performing contraction changes the EMG signal may become non-stationary [27]; therefore, averaging with relatively long smoothing times becomes problematic as it is difficult to capture the changes in the signal resulting in delay noticeable by the user [26]. Some controller delay may be acceptable since there is some natural delay between the onset of EMG activity to the change in acceleration of the forearm (electromechanical delay); in fact, physiological delays have been reported to be in the order of 25-45 ms [28]. Farrel *et al.* [29] found that the most favorable controller delay ranges between 100-125 ms depending on the prehensor speed [29] (for fast prehensors the delay should be smaller). Hence smoothing times of 250-300 ms for optimal amplitude estimation may result in

unfavorable controller delay [23] [26]. This tradeoff is often observed and is referred to as the fundamental filtering paradox [24] [30]; with a fixed time-constant smoothing filter it is possible to have a fast response or an amplitude estimate with low variance, but not both.

To improve this tradeoff, some have suggested an averaging filter with smoothing time that adapts to the contractions rate of changes [22] [31]. Miyano *et al*. [32] reported that the optimal averaging window length is a function of the contraction's rate of change; for fast varying contractions the window length should be short, while for slow varying contractions the window length can be longer. However, these methods have only been found to be beneficial while users performed slowly varying contractions [22] [31].

Xiong *et al.* [21] proposed that estimation errors caused by the non-stationary EMG could be compensated when using a window of past and new inputs, as past data, during amplitude rise (for example), underestimated the amplitude, while current data overestimates the amplitude. The midpoint moving average estimators (MMAE) [21] and the Kalman filter (KF) [18] [27] are two methods that operate on both past data and new incoming data. While the MMAE is an arithmetic mean over a window of past and new data, the KF is a weighted arithmetic mean, where, instead of each of the data points contributing equally to the final average, some data points contribute more than others. In fact, the Kalman filter assumes that the amplitude estimate can be obtained from a parametric (linear or non-linear) equation of past outputs and recent inputs. Additionally, these parameters adapt to changes in the signal.

It has been suggested that Kalman filtering was not of great use in myoelectric control because of its computational complexity [33] and because the mathematical modeling of the EMG does not adhere to Kalman filtering theory [19] [34]. The EMG has generally been modeled as amplitude-modulated noise [32] meaning that the "noise" and the desired signal are multipliers whereas Kalman filters are typically applied to signals with additive noise. The EMG signal can be transformed into a signal with additive noise variance by replacing the full-wave rectifier (Figure 2-1) with a log nonlinearizer [18]. Regardless, the Kalman filter is simply a recursive data processing algorithm [34]; it is optimal and easier to define when applied to signals with additive noise, but it can be applied to most processes [33] provided that sufficient information about their time evolution and noise variance is known.

2.1.5 Limitations

The clinical success of conventional control systems is largely due to their simplicity [2] [3]. Unfortunately, they cannot easily accommodate multiple functions [3] or control several devices since this would require multiple signals from independent muscles (generally unavailable [1] [2]). Accommodating for multiple functions necessitates mode switching increasing the control complexity and decreasing intuitiveness for the user [35].

More recent techniques for developing multifunctional devices include postural control schemes [36] and pattern recognition systems [37] [38] [39]. Postural control schemes, as presented by Segil *et al*. [36], use EMG signals modulated by the user. The system transforms these signals into a continuous two-dimensional domain, which is then

translated into a hand posture. However, it is only recently that the studies in postural control focus on clinical implementations [36].

Pattern recognition control systems have been studied clinically [40]. These systems use the available muscle sites as effectively as possible [1] [4] and are discussed in the next section.

2.2 Pattern Recognition Control

Pattern-recognition-based myoelectric control systems date back to the late 1960s and early 1970s [37] [38] [39]. Such systems assume that the EMG signal obtained while performing a given *contraction type* (i.e. muscle contraction required to induce a desired upper limb function) is repetitive and distinct from other contraction types. These signal exemplars, or *patterns*, can be classified. Such systems involve three parts: 1) signal acquisition and rectification, 2) data segmentation and feature extraction, and 3) classification.

Feature extraction and classification have been widely studied for pattern recognition control systems; the first obtains representation of the multi-channel EMG (i.e. the patterns), while the second makes sense of these patterns.

Early studies found that EMG patterns could be represented by auto-regressive moving average (ARMA) features to discriminate between contraction types [33] by assuming the EMG is sufficiently stationary. Unfortunately, the computing technology at the time was not sufficient for real-time control in clinical settings. Consequently, this method did not evolve beyond laboratory implementation [2] [41].

Later, Hudgins *et al.* [2] used a different approach that obtained patterns by extracting time-domain (TD) features from EMG transient bursts and used an artificial neural

network (ANN) to classify the patterns in order to discriminate between functions. The main drawback of using the transient EMG as a control input is that it requires initiating a single contraction from rest [42], which does not facilitate a continuous form of control. Englehart *et al.* [43]were the first to introduce the concept of continuous control in pattern recognition based prostheses. They suggested extracting patterns of TD features on data segments obtained from a sliding analysis window in time. A continuous decision stream was obtained by classifying the pattern of each segment using a linear discriminant analysis (LDA) classifier [42]. Although these systems have shown promising results, erratic misclassification often occurs at movement onset or during *class transitions* (i.e. when going from one contraction type to another) [42] [44]. Post-processing techniques are sometimes applied in order to override erratic misclassifications resulting in smoother transitions between classes. This continuous control scheme with proportional control, shown Figure 2-2, has become widely adopted.

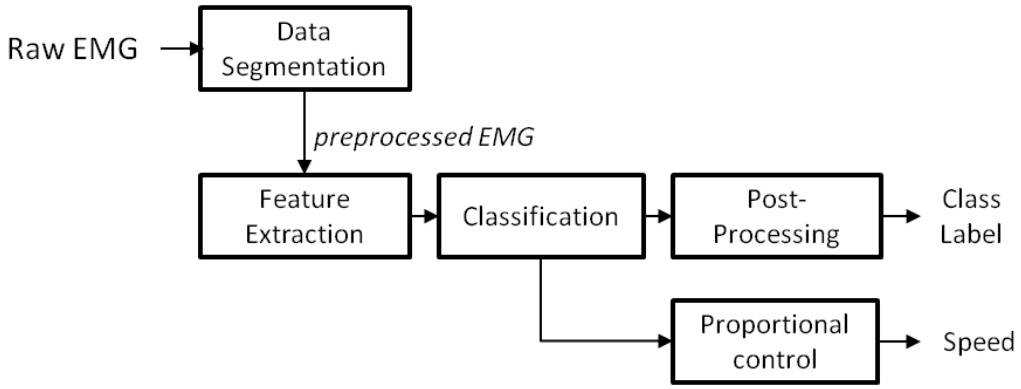

Figure 2-2: A block diagram of a generalized pattern recognition based myoelectric control scheme that includes proportional control.

It is difficult to measure the performance of systems similar to the one described above because they are dynamic; their output is constantly changing as the users actively

adjust their contractions to reach the desired output. In the context of this work, contractions performed to actively change the state of the prostheses are referred to as *dynamic contractions* (force varying contractions or class transitions for example). In the past, it was customary to evaluate the performance of these systems *offline* where users did not actively control the device; in fact, they strictly performed constant force contractions held in a static position (referred to in this work as *steady-state contractions*). Classification accuracy (defined as the ratio of correct decisions) has typically been measured to evaluate the performance of these systems offline.

The system described above has shown classification accuracies above 90% with a response time below 300 ms [43]. Recent studies have found that classification accuracy may not be representative of the efficacy with which users can perform functional tasks (i.e. *usability*) [5]. It has been speculated that this may be due to the fact that classification is measured in constrained experimental settings that do not represent humans' dynamic motion and environment. Recent studies have evaluated novel assessment approaches [6] [7] [8] that may be more representative of real-world applications. These assessment methods as well as the elements of the diagram shown in Figure 2-2 are discussed below.

2.2.1 Data Acquisition and Rectification

The *raw EMG signal* is the EMG signal recorded from electrodes that are usually placed on the surface of the skin. Data acquisition includes EMG amplification (typically using differential amplifiers to eliminate the common mode between two electrodes), and filtering typically with high-pass cutoff frequency between 5-10 Hz and low-pass cutoff frequency commonly around 500 Hz. High-pass filtering removes baseline drift, any DC

offset, and movement artifacts, and low-pass filtering prevents aliasing while sampling the signal.

2.2.2 Data Segmentation and Feature Extraction

Due to the EMG's random nature, features are extracted from raw EMG segments obtained from an analysis window which progresses across the data. The only preprocessing performed on the raw data is to remove the mean, computed from a large EMG sample. The EMG data segments can be disjoint or overlapping depending on their *frame increment* (FI), and the amount of data in a segment depends on their *frame length* (FL). This is illustrated in

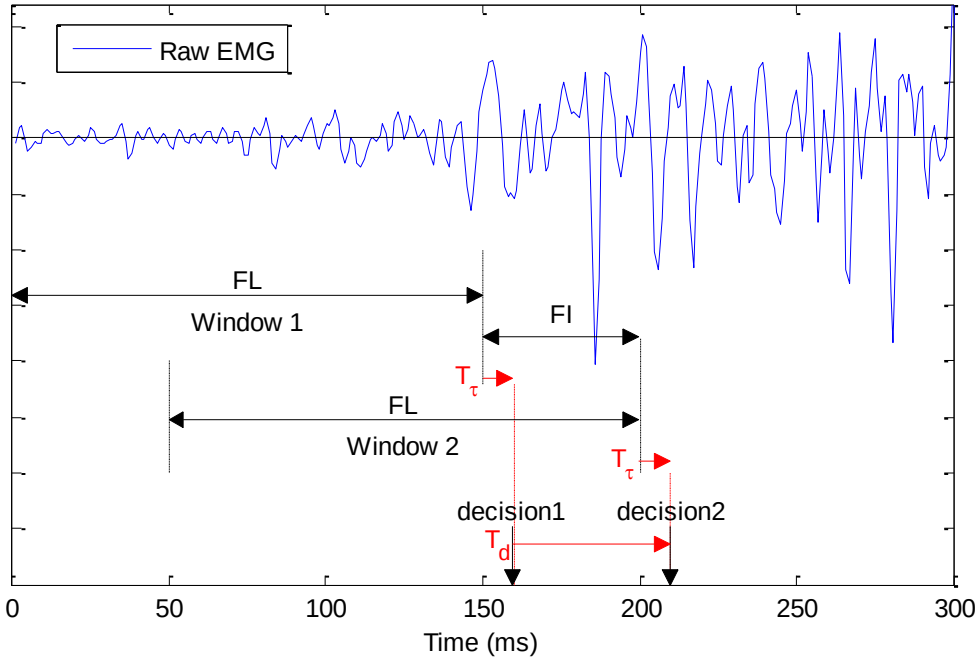

Figure 2-3 for two consecutive window frames:

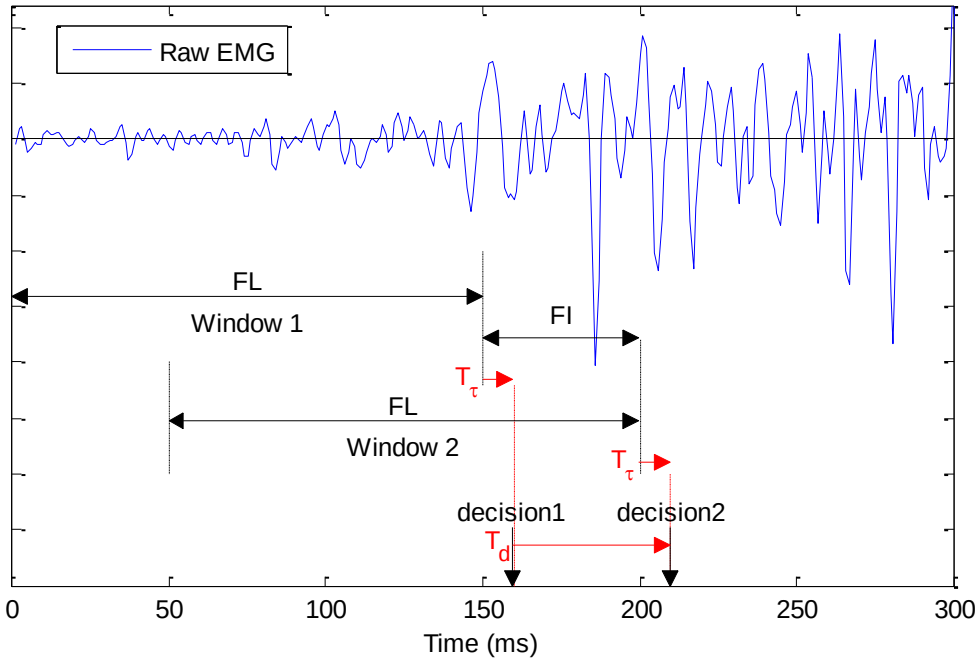

Figure 2-3: Overlapping window frames with frame length size FL and frame increment size FI for pattern recognition control. Feature extraction is performed on the preprocessed data segment of windows 1 and 2 with T_τ processing delay. Decision1 and decision2 are obtained at T_d interval increments.

The example illustrated in

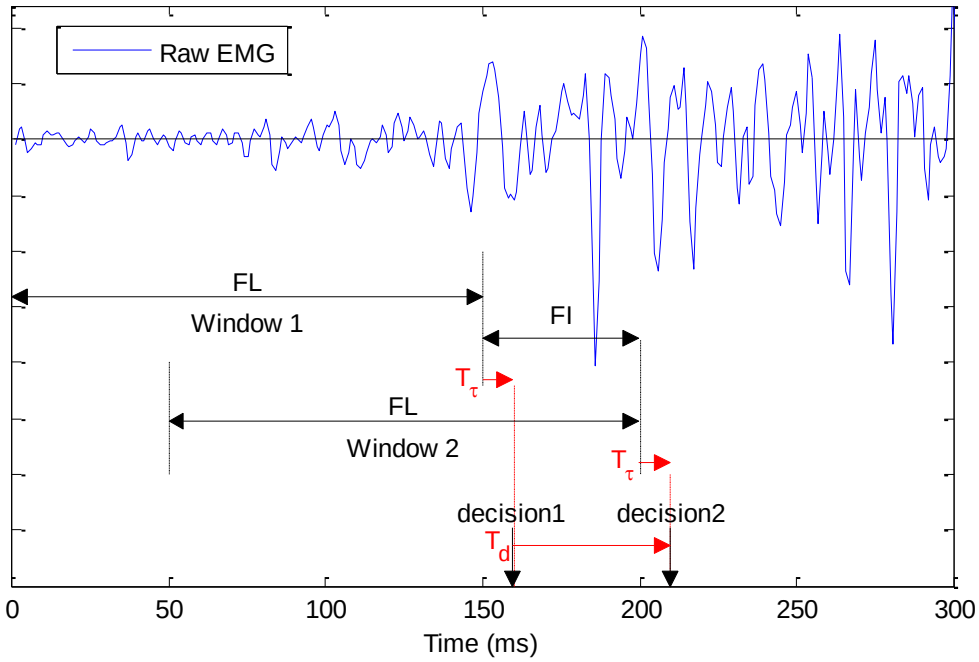

Figure 2-3 shows overlapping data segments with a FI = 50 ms and a FL = 150 ms.

The variable T_τ represents the processing time to extract features from the data segment,

and to classify the features. For disjoint segments, the FI and FL are equal. However, a much denser decision stream is obtained when the FI is shorter than the FL, but the FI cannot be shorter than the processing time.

The *response time* is the delay between the generation of EMG and the actuation of a control signal to the prosthesis [45]. This is the sum of the length of the analysis window (FL) and the processing time (T_τ). Consequently, the FL must be short enough so that an excessive delay is not introduced to the controller [46], but it must also be long enough to minimize random fluctuations in the features (due to the raw EMG's random nature) as this results in more reliable decisions [11]. It has been shown that, while users perform steady-state contractions, fewer fluctuations in the features yield higher classification accuracies [42] [47].

It is generally regarded that an acceptable response time for prosthetic devices is shorter than 300 ms [42] [45]. Yet estimates for the maximum allowable response time reported vary from 50 ms to up to 400 ms [29] [46] [48]. Some authors studied the effects of controller delay on conventional control prosthesis usability and showed that the usability of the prosthesis is reduced for controller delays longer than 100 ms [29], speculating that it may be desirable to use shorter FL to produce a more usable, yet slightly less accurate control system [11]. Later, Smith *et al.* evaluated the effects of FL on users ability to control a virtual prosthesis using pattern recognition based myoelectric control, and found that window lengths in the range of 150 to 250 ms [46] were optimal in terms of the tradeoff between accuracy and responsiveness.

Rather than increasing the FL to minimize feature fluctuations, various authors have suggested incorporating decorrelation filters, based on principal component analysis

(PCA) [49] or on whitening filters [31] [47], in the preprocessing step before rectifying the signal. PCA decorrelates data while whitening filters decorrelate the data and equalize the spectrum of the signal making it more similar to the white noise spectrum.

It has been reported that, for steady-state contractions, both the PCA and whitening filters improve classification accuracy by ~5% on average [47] [49]. It has also been reported that the whitening filter improves the features *signal-to-noise ratio* (SNR) [31] which has been used to characterize signal variability. Although the conventional definition of SNR is the ratio between the power of the desired signal and the noise, in the context of using the EMG to extract a control signal, a modified definition of the SNR has been used to characterize signal variability. It has been defined as the ratio of the mean of the computed control signal to its standard deviation over a given timeframe. An improvement in SNR indicates less random fluctuation in the control signal.

2.2.3 Features

Different features have been used in pattern recognition involving both time-domain and time-frequency domain features. Some include the mean absolute value [2] [42] [50] [51] [52], zero-crossing [2] [42] [50] [51] [52], slope sign changes [2] [42] [50] [51], autoregressive (AR) model coefficients [2] [50] [53] [54] [55], cepstrum coefficients [41] [54] [56], waveform length [2] [42] [50] [52], wavelet transform [57], wavelet packet transform [51] [58], and short time Fourier transform [57].

Features are computed based on the assumption that the EMG signal is stationary over the analysis window and it has been generally accepted that the most separable features are expected to make the best estimation of user intent [59] as this minimizes

misclassification. Cluster separability, however, is not the only condition for high-quality features. These must also [59]:

1) have low computational complexity. This is desirable for computation using the embedded microprocessors necessary for clinical deployment;
2) be repeatable. The selected features should provide consistent patterns between repetitions of a given contraction type, and preserve their cluster separability during non-ideal conditions encountered during practical use of the prosthesis, and;
3) be extracted from a small window size to allow the control system to be responsive.

Most contemporary systems employ some combination of TD and AR features as they are straightforward to compute, do not require dimensionality reduction, and can be computed in real-time on simple embedded processors.

Although it is generally assumed that the EMG signal is stationary over the analysis window, this is not the case during dynamic contractions when the signal is rapidly changing [59]. Some researchers have suggested that it is difficult to precisely and accurately compute an unbiased feature estimate from segments of data when the EMG is non-stationary [56]. They suggest that data segmenting should be combined with adaptive models to improve the feature estimate when the EMG is non-stationary [56] [60]. In fact, the adaptive cepstrum vector (ACV) [56] [60] features, which assume that the features from a current window can be predicted by a linear combination of the features from past windows, have shown promising results [60]. However, it is unclear from the

ACV studies (i.e. [56] [60]) if ACV methods improve features while users perform dynamic contractions (the authors evaluated steady-state contractions).

2.2.4 Classification

Artificial Neural Networks (ANN) [2] [55] [58] [61] [62] [63] [64] and Linear Discriminant Analysis (LDA) classifiers [42] [57] [64] have been investigated with considerable depth in pattern recognition based myoelectric control. The use of heuristic fuzzy logic [51] [52] [53] [54] [65], Gaussian mixture models (GMM) [50] [66] [67], hidden Markov models (HMM) [67], and multi-class support vector machines (SVMs) [47] [62] [68] [69] [70] have also been reported.

The LDA has been frequently employed because of its relative simplicity and ease of implementation. It is commonly trained using patterns extracted from the EMG of steady-state contractions. Such classification methods have shown promising results [43], and although trained under steady-state conditions, they have been shown to provide relatively high offline performance when subjects performed contractions starting from rest [71]. The clinical impact of these systems remains limited due to their lack of reliability [72] in real-time practical use. Since the classifier assumes that all patterns belong to one of the trained motions, any pattern presented that represents some other movement, or that is unknown (i.e. different from the trained patterns), will often result in involuntary active movements because the pattern is likely erroneously classified. Patterns that differ from the training set (resulting in misclassifications) may arise due to (but not limited to):

1) Physical or physiological changes, such as electrodes shifting or muscle fatigue [73] [74] [75] [76];
2) A change in contraction intensity or limb position [77] [78], or;
3) A class transition [45] [79] [80] [81] [82].

Efforts have been made to minimize misclassifications under these three conditions, each of which are discussed below.

Designing the LDA classifier to adapt to changes in the patterns due to electrode shift or muscle fatigue has been suggested, these adapt to gradual changes but can only adapt to drastic changes while supervised [73] [74] [83] which is not feasible in real-time applications where the intended motion is unknown. Prosthesis guided training [84], which performs on-demand training using the prosthesis as the training prompt while it is worn, has been introduced as a clinically viable method of retraining when disruptive changes occur.

It has been shown that variations in contraction intensity can be accommodated by training using varying contraction intensities, such as gradually increasing the force through the collection process [71] [77] [85]. Similar approaches have been proposed to alleviate error introduced by limb movement. Although, it is not practical to include all possible contraction intensities and positions during training, it has been shown that even a subset of positions can improve system robustness [77].

Class transitions can be classified using a classifier trained with transitory feature patterns [2], but it is not practical to train each possible class transitions. Hidden Markov models (HMM), which assume that discriminatory information exists in the temporal evolution of features, have been proposed for myoelectric control [67] [79] to improve

class transition control. However, the HMM suggested by Chan *et al.* [67] showed performance degradation during class transitions, while others [79] found that the HMM performed poorly in classification of steady-state contractions. Others have suggested minimizing unexpected misclassifications due to class transitions by using state-based approaches [72] or hybrid models [79] [86]. These methods detect class transitions by conditioning the raw EMG signal using an operator (such as the Teager-Kaiser energy operator [87]) that improves the signal's SNR. This processed EMG signal is then used to detect the onset of motion [45] [88] or any sudden changes in EMG variability [72] which are representative of class transitions. These methods have shown promising results as they minimize sporadic class changes during class transitions [72] [86]. However the authors state that the methods used to detect class transition may increase latency [72], and may depend on thresholds [72] [88] that are highly variable between subjects.

Some have proposed a more general approach to the three problems listed above; rejecting any unfamiliar or confounding patterns [89] [90] [91]. In the event of unfamiliar patterns, defaulting to a no-motion state [89] [90] has been proposed to prevent involuntary and erroneous active movements. Such methods assume that inadvertent activations of the limb are more "costly" than those that cause a pause in the motion. This has been attributed to the fact that extraneous motions require users to elicit additional corrective motions, increasing motion times, and user frustration [90]. It is difficult to determine which patterns are unknown and should be rejected. Rejection has been accomplished by using majority voting of binary classifiers such as the multiple binary classifiers (MBC) [89], a selective multiclass one-versus-one classifier [90], or a

confidence based rejection threshold classifier [92] [93]. These were configured to reject patterns if the decisions of the classifiers were not unanimous. Unfortunately, this could result in difficulties activating a poorly trained class [6] as its patterns may be unfamiliar or confounding resulting in rejection.

2.2.5 Post-Processing

Post-processing can be performed to remove erroneous active movements and smooth transitions between classes. Two main post-processing methods have been suggested to improve transitions between class decisions: majority vote [42] and decision-based velocity ramp [94].

Majority vote (MV) uses a window of the most recent classification results to make a decision based on the class that appears most often [42]. Consequently, it reduces variability in decision outputs, but results in system response delays. The number of decisions used to form the majority voting window is therefore subject to a tradeoff between smoothing class decisions and controller delays.

In a decision-based velocity ramp, the decision remains unchanged but the speed of the prostheses is attenuated if a change in the decision occurs. If the current decision is different from the previous decision, the output speed is attenuated to a fraction of the signal amplitude and increases as the class decisions becomes more stable [94]. This attenuates the effects of misclassifications without delay in the decision stream. Simon *et al.* [44] compared both post-processing methods and found that the decision-based velocity ramp offered better real-time control. It is interesting to note, however, that this approach does not eliminate classification errors. Instead, it reduces their effect by

attenuating the proportional control output that is commonly paired with pattern recognition.

2.2.6 Proportional Control

As stated in Section 2.1.3, the idea of mapping the speed of the device to the contraction intensity is not new. Proportional control, when used in pattern recognition, is not as straightforward as when used in conventional control strategies due to the fact that pattern recognition utilizes multiple EMG channels simultaneously. In pattern recognition-based control practice, the corresponding proportional control signal is often derived from the weighted average of the mean absolute value (MAV) of the EMG channels. The weighted average may be normalized for each class movement [95]. Although it is unclear if this is a correct estimation of contraction speed, Simon *et al.* [96] showed that it improves controllability when compared to a system with a binary on/off control where the device operates at constant speed while moving.

Later, Scheme *et al.* [97] evaluated two normalizing approaches for proportional control which, unlike the traditional weighted average method, do not require a direct linear mapping between contraction intensity and device speed. In fact, these methods allow users to reach both 100% and 0% speed more easily. This has for effect of faster speed at higher contraction intensity, and lower speeds at lower contraction intensities. Results obtained, while performing a real-time test, suggest that these approaches allow a more natural form of control.

2.2.7 Assessment

It is common to assess pattern recognition based myoelectric controls by evaluating offline classification accuracy (the percentage of correct decisions overall decisions).

Lock *et al.* [5] found that a system with high classification accuracy does not necessarily produce a system with good usability. This may be due to the fact that accuracy is determined offline, where users have no feedback resulting in experimental settings that do not reflect dynamic use. Furthermore, users are typically asked to perform discrete, steady-state contractions. This is unnatural; to accomplish a functional task with a prosthesis one must contract dynamically to transition between motions or to modulate the proportional control.

Various efforts have been made to develop real-time experiments that capture the dynamic nature of hand and wrist movements to assess pattern recognition systems. Clothespin relocation tests [98] and box and block tests have been proposed, and are commonly used in functional clinical assessment [29] [99]. Although informative, these tests are not practical while developing pattern recognition control schemes. It is costly to acquire the hardware necessary to perform the tests, and they offer only crude measures of performance. Consequently, many researchers have employed virtual environments to study functional tasks with pattern recognition EMG control systems [5] [6] [7] [8] [82].

Lock [6] simulated the clothespin relocation test in a virtual environment. The test demonstrated that classification accuracy alone was not a good measure of usability which was determined by the number of pins dropped and the time it took users to complete each task. They suggested that additional measures should be evaluated to quantify usability. Selection time, completion time, and completion rate were later introduced by Kuiken *et al.* [7]. During a motion test, users were asked to complete steady-state contractions starting from rest. Selection and completion time were used to measure the system and user responses respectively, while completion rate measured the

percentage of motions that the classifier was able to sustain for more than one second. Although this test provides information about the system's ability to respond to dynamic contractions as users were provided feedback, participants were restricted to motions that started from rest and reached steady-state sustained contractions. The *Target Achievement Control* (TAC) test later proposed by Simon *et al.* [100] did not restrict motion. The test required participants to control a virtual limb to reach a target posture. A system with low completion times, high completion rates, and low overall test times was considered to have good usability.

The results obtained from a virtual test can be affected by the acceptance and perception of the visual feedback provided by the virtual environment [6]. Both the virtual clothespin and TAC tests are three dimensional tasks presented on a two dimensional screen. Users' ability to interpret the three dimensional image may greatly affect the results. Scheme *et al.* [8] attempted to alleviate visual misinterpretation by proposing a Fitts' law test which required participants to acquire targets in a two-dimensional Cartesian space. In this test, users were prompted to move a controlled cursor to targets of varying distance and size. The prosthetic degrees of freedom were mapped to control the horizontal and vertical velocity of the cursor (for example, wrist flexion and extension were used to move the cursor left and right respectively). Following Fitts' law, it was assumed that the task difficulty increased as the target distance increased or the target size decreased. Throughput was defined as a measure of performance and was computed as a function of the time it takes users to reach the target and the task's index of difficulty (ID) [101] which increases for smaller and more distal targets.

2.3 Discussion and Motivation

Although pattern recognition methods for myoelectric control have shown promising results, the literature reports a clear discrepancy between offline performance and real-time usability. It has been speculated that this is directly linked to the fact that offline performance typically evaluates system performance while users perform steady-state contractions. These static contractions are not representative of real-time applications which require dynamic contractions.

The control of dynamic contractions, however, is a complex problem with several tradeoffs. In fact, for current systems (such as in [82]), the LDA classifier is trained with features obtained from a data window of limited FL, wherein it is implicitly assumed that the EMG signal is stationary (i.e. constant). When this assumption is true (as for the case of steady-state contraction), it is often possible to obtain highly discriminative features for many contraction types. The literature reports that the longer the FL, the higher the classification accuracy [46] [102]; however, longer FL results in controller delays while users perform class transitions where the EMG is non-stationary [86] [103]. In fact, Al-Assaf *et al.* [86] suggested that feature estimation methods that facilitate shorter FL would be beneficial, allowing the system to respond more quickly to changes. As discussed in Section 2.1.4 and Section 2.2.3, there is a large body of research that has focused on improving the control signals estimation for shorter FL. The methods proposed either applied additional pre-processing to the raw signal or supplemented feature extraction with adaptive capabilities to account for non-stationary EMG.

Although these methods have shown some promising results, they do not consider and/or resolve the fact that, in pattern recognition control, the class or state of the prosthesis can only be determined when patterns fall within the trained class boundaries.

Patterns obtained during class transitions do not fall within the trained class boundaries [67] [104] (typically trained using steady-state contractions); consequently, there are many challenges with correctly classifying transitory signals using classical methods [86] [94]. As discussed in Section 2.2.4, the literature reports systems that are capable of classifying class transitions, but these require unreasonable training scenarios, cannot be used in conjunction with steady-state contractions, or have shown poor usability in real-time settings. In other words, there is evidence that both static and dynamic contractions can be classified, but these must be controlled using different approaches. The tendency in recent literature is to develop a system that is capable of treating steady-state and class transitions separately [86] [88]. Such hybrid models require proper characterization of class transitions which is challenging because of the EMG intrinsic nature.

This work focuses on the development of a hybrid model to improve system reliability during class transitions in real-time settings. It is anticipated that this will improve usability while users perform dynamic contractions. It is motivated by the assumption that both controller delays and inadvertent class activations that can occur while users perform class transitions are costly and frustrating. Therefore methods such as using an analysis window with longer FL [42], post-processing of the decision stream [42], or rejecting inadvertent classes [89] [90] [91] may not suffice to improve class transition controls; although they result in smoother class transitions, they increase controller delays.

It is worth clarifying that the state of the prosthesis is indeterminate when patterns are not associated to the trained classes. For ease of readability, features or patterns are

referred to as being in an *indeterminate state* when they cannot be associated to the trained classes.

From these observations, it is noticeable that a system capable of providing users with reliable control while in class transitions must resolve the following problems:

1) Features obtained from longer FL obtain better classification accuracy but increase controller delay;
2) Systems with good performance during steady-state contractions do not always provide good performance while users perform class transitions, and;
3) Patterns of class transitions are often misclassified because they are in an indeterminate state. Incorporating all possible class transitions in the training set could mediate this problem, but this results in unreasonable training scenarios which are not clinically viable.

This motivates the need to develop a system which is capable of:

1) Improving the SNR of the extracted features for short FL;
2) Prohibiting features from staying in an indeterminate state for long periods of time;
3) Separating class transitions and steady-state contractions to allow appropriate control methods for each, and;
4) Improving the decision making strategy during class transitions.

To reach these objectives, this work performed offline and real-time analysis. The offline analysis was performed to help develop a system that improves class transition control, whereas the real-time analysis verifies if the novel system improved usability.

Chapter 3 - Feature Conditioning

3.1 Overview

In the previous Chapter, it was proposed that a system capable of providing users with reliable control while in class transitions should:

1) Improve (i.e. reduce) the features' variability for short FL in order to minimize the systems response time with no cost in classification accuracy;
2) Prohibit features from staying in an indeterminate state for long periods of time;
3) Separate class transitions and steady-state contractions to allow appropriate decision logic for each, and;
4) Improve the decision making strategy during class transitions.

The first and second conditions listed above motivated the development of *feature conditioning* methods designed to improve the features variability and to minimize the time features may spend in an indeterminate state. Feature conditioning is applied after feature extraction as shown in Figure 3-1:

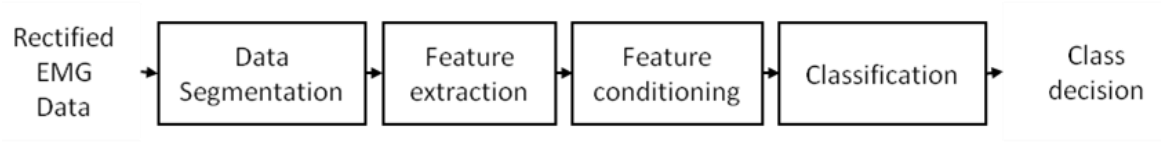

Figure 3-1: Block diagram illustrating the data flow of pattern recognition-based myoelectric control scheme supplemented with feature conditioning.

This chapter presents feature conditioning methods and is divided into two parts. In the first part, the different feature conditioning methods are presented, whereas in the second part, the system illustrated in Figure 3-1 for different feature conditioning

methods is evaluated and compared against current pattern recognition control methods found in the literature to determine if, as anticipated, feature conditioning satisfies the conditions 1) and 2) listed above.

3.2 Feature conditioning methods

Much research has been published in the study of features to determine the most robust [2] [41] [42] [50] [51] [52] [53] [54] [55] [56] [57] [58]. Features are computed based on the assumption that the EMG signal is stationary over the analysis window. It is important to consider the cases where the EMG is non-stationary as this occurs while users perform dynamic contractions. Since Kalman filters have been applied effectively to both stationary and non-stationary signals [105], this work presents feature conditioning methods inspired by Kalman filter theory.

In pattern recognition control systems (as in Figure 3-1), the EMG is segmented using a sliding window of a given frame length (FL) and frame increment (FI). Features are extracted for each preprocessed data segment resulting in a series of patterns that continuously evolve over time. The feature conditioning methods attempt to represent this time evolution of the features to obtain less variable feature estimates. They operate on the assumption that the current features are somewhere between the previous features and the class associated to the most current extracted features (i.e. the class decision). The feature conditioning methods, just as the Kalman filter, can be broken down into three parts:

1) The *motion model* which relates the current features position (i.e. value) relative to the previous features;

2) The *measurement model* which relates the current features position relative to the class decision;
3) The *update* which estimates where (in between the previous features and the output class) the current features are most likely to be found.

In this work, the outputs of the feature conditioning methods are referred to as *feature estimates*. Unlike Kalman filters that attempt to obtain an estimate of a system's true state (see Appendix D for Kalman filter theory), the feature conditioning methods want to obtain a feature that would allow the system to be operationally compliant. That is, features with less variability while in steady-state and better behavior while in class transitions. Therefore the feature estimates obtained using feature conditioning represent, as accurately as possible, the most favorable, yet unknown, features referred to as *compliant features*. *Extracted features* refer to the features obtained from data segments of raw EMG data.

The motion and measurement models, as well as the update method are discussed in this section. Afterwards, different feature conditioning designs are presented.

3.2.1 Motion Model

The motion model is based on the assumption that, for very short time increments, the compliant features at time *t* are essentially the same as those at time *t-1*. This could be modeled as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}_t = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}_{t-1} \quad (3\text{-}1)$$

$$\vec{\boldsymbol{x}}_t = \vec{\boldsymbol{x}}_{t-1} \quad (3\text{-}2)$$

where *D* is the total number of features in a set, $\vec{\boldsymbol{x}}_t$ is the compliant features vector at time *t*, and $\vec{\boldsymbol{x}}_{t-1}$ the previous compliant features. Since the compliant features at time *t-1* are unknown, the motion model must rely on the previous feature estimates to make prediction about the compliant features position at time *t*. Therefore, the motion model outputs, referred to as the *predicted features*, are assumed to be at the same location as those of the previous features estimate. This is modeled as:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_D \end{bmatrix}_t = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_D \end{bmatrix}_{t-1} \quad (3\text{-}3)$$

$$\vec{\boldsymbol{x}}'_t = \vec{\bar{\boldsymbol{x}}}_{t-1} \quad (3\text{-}4)$$

where $\vec{\boldsymbol{x}}'_t$ is the predicted feature vector, and $\vec{\bar{\boldsymbol{x}}}_{t-1}$ the previous feature estimates. This model's reliability may vary depending on the *dynamic level* of the contractions performed by the users. Figure 3-2 illustrates the dynamic level in the contractions (with some example cases) on the scale of the model's reliability (increasing reliability from left to right).

When users perform steady-state contractions, it is assumed that this model is reliable; however, it becomes less and less reliable as the dynamic level in the contractions increases. The reliability in the motion model is defined by the *motion model variance(σ^2) or covariance*[1] *(R)* (for multidimensional problems). From random theory (Appendix C), the motion model is modeled as a linear system with added Gaussian noise $N(0, R_t)$:

[1] In this work, variance and covariance are used interchangeably to designate the variance in the multidimensional model.

$$\vec{\boldsymbol{x}}_t' = \vec{\boldsymbol{x}}_{t-1} + \boldsymbol{e_t}, \qquad \boldsymbol{e_t} \sim N(0, R_t) \tag{3-5}$$

where e_t is Gaussian noise with covariance R_t. In this work, the model variance is a measure of uncertainty in the predicted features (high variability indicates predicted features with low confidence (i.e. low reliability)).

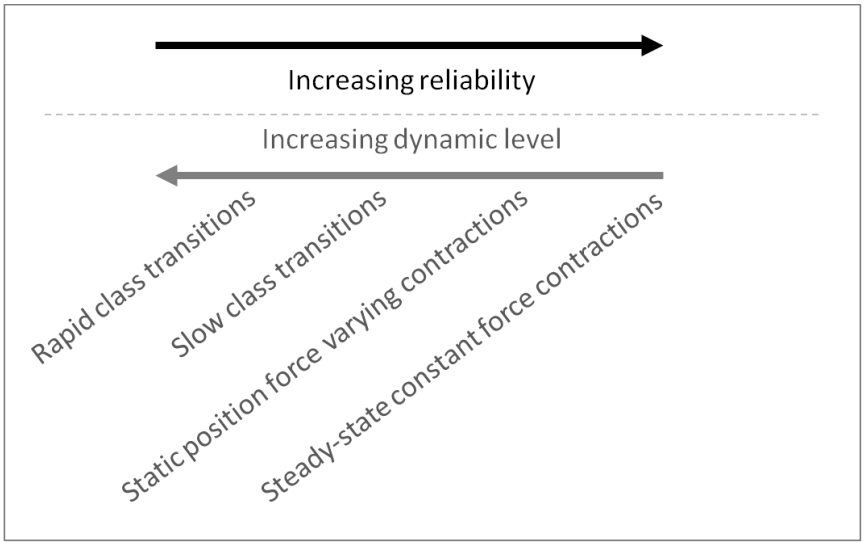

Figure 3-2: Motion model reliability. As the contractions' dynamic level increases, the motion model is less reliable.

3.2.2 Measurement Model

The measurement model is designed using the LDA class decisions obtained by classifying the extracted features (see Appendix B for theory on the LDA). The *measured features* are assigned to the class decisions mean (see Appendix B) because, from random theory (Appendix C), the compliant features are most likely to fall at the class center:

$$\boldsymbol{E}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}_t = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_D \end{bmatrix}_t \tag{3-6}$$

$$\boldsymbol{E}[\vec{\boldsymbol{x}}_t] = \vec{\boldsymbol{m}}_t \tag{3-7}$$

where $E[\cdot]$ represents the expectation, and $\vec{\boldsymbol{m}}$ is the measured features vector located at the class decision means.

Since the class decisions are obtained from the extracted features, which are noisy, and that the compliant features may not be at the class center, there is some uncertainty to the measurement. This uncertainty is defined by the *measurement variance (δ^2) or covariance (Q_t)*. From random theory (Appendix C), the measurement model is modeled as a linear system with added Gaussian noise $N(0, Q_t)$:

$$\vec{\boldsymbol{y}}_{\boldsymbol{t}} = \vec{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{d}_t, \quad \boldsymbol{d}_{\boldsymbol{t}} \sim N(0, Q_t) \tag{3-8}$$

where $\vec{\boldsymbol{y}}$ is the measurement, and $\boldsymbol{d}_t$ is Gaussian noise with variance $\boldsymbol{Q}_{\boldsymbol{t}}$. In this work, the measurement variance is a measure of uncertainty in the measured features (high variability indicates measured features with low confidence). From random theory (Appendix C) and combining Equations 3-7 and 3-8:

$$\boldsymbol{E}[\vec{\boldsymbol{y}}_{\boldsymbol{t}}] = \boldsymbol{E}[\vec{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{d}_t] = \boldsymbol{E}[\vec{\boldsymbol{x}}_{\boldsymbol{t}}] + \mathbf{0} = \vec{\boldsymbol{m}}_{\boldsymbol{t}}$$

3.2.3 Update

The update refers to the estimate update performed by the feature conditioning methods from time t-1 to t. Given the motion and measurement models, the previous feature estimates are updated as follows:

$$\vec{\bar{\boldsymbol{x}}}_t = \vec{\bar{\boldsymbol{x}}}_{t-1} + \boldsymbol{K}_{\boldsymbol{t}}(\vec{\boldsymbol{\Delta}}_{\mathbf{t}}) \tag{3-9}$$

where $\vec{\bar{\boldsymbol{x}}}$ are the feature estimates vector, $\boldsymbol{K}$ is a diagonal matrix with update gains $[k_1, k_2, \ldots k_D]$, and $\vec{\boldsymbol{\Delta}}_{\mathbf{t}}$ is the distance between the measured and predicted features defined as:

$$\boldsymbol{\Delta}_{\boldsymbol{t}} = \boldsymbol{E}[\vec{\boldsymbol{y}}_{\boldsymbol{t}} - \vec{\boldsymbol{x}}'_{\boldsymbol{t}}] = \boldsymbol{E}[\vec{\boldsymbol{y}}_{\boldsymbol{t}}] - \boldsymbol{E}\left[\vec{\bar{\boldsymbol{x}}}_{\boldsymbol{t-1}} + \vec{\boldsymbol{e}}_{\boldsymbol{t}}\right] = \vec{\boldsymbol{m}}_{\boldsymbol{t}} - \vec{\bar{\boldsymbol{x}}}_{\boldsymbol{t-1}} \tag{3-10}$$

The gains are chosen to allow the estimates to fall closer to the most confident amongst the predicted or the measured features. A gain closer to zero forces the estimates closer to the predicted features (i.e. previous estimate), and a gain closer to one forces the estimates closer to the measured features. The gains are determined based on a weighting of the motion and measurement model variances defining the confidence in the predicted and measured features respectively.

The feature conditioning methods proposed here are based on this general equation. This equation is analogous to the linear Kalman filter (KF) (see Appendix D or [106] for a complete derivation of the KF). The challenge in KF development is to model the measurement model and motion model variances. In this work, these are modeled to allow the feature estimates to converge towards class centers while users transition and to stay in the current class during steady-state contractions. It is anticipated that this will improve the SNR of the features for short FL, prohibit features from staying in an indeterminate state for long periods of time, and potentially facilitate the characterization of steady-state classes from class transitions.

The different feature conditioning methods proposed are grouped into two categories: 1) simple methods, and 2) KF based methods.

3.2.4 Simple Feature Conditioning Methods

The simplest method to design the measurement and motion models variances is to assume that they are equal. In such a case, the estimate at time *t* would fall at the mid-point between the measured features and the predicted features, equating to a gain of 0.5. This is represented as:

$$\vec{\bar{x}}_t = \vec{\bar{x}}_{t-1} + \boldsymbol{K_t}(\vec{\boldsymbol{\Delta}}_\mathbf{t}) \,, \qquad \boldsymbol{K} = \begin{bmatrix} .5 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & .5 \end{bmatrix} \tag{3-11}$$

This model assumes that the current class decision is correct, which is not always the case. Unintentional or incorrect class activations could have the effect of forcing the estimate to converge towards the wrong class mean. It is proposed to reduce the effects of unintentional class activations by updating the feature estimate more conservatively. Conservative updates are performed by only allowing those where the class decision at time *t* is the same as the class decisions at time *t-1*. It is expected that this may minimize the effects of any sudden or drastic changes in the measurement. Herein, this method is referred to as the *simple conservative feature conditioning method*, whereas the previous method is referred to as the *simple aggressive feature conditioning method* (see Appendix E.2 for an illustrative example of the aggressive feature conditioning method).

Since these methods employ fixed gains of 0.5 they are not KF. It is possible to design a KF approach with dynamic gains obtained by modeling the motion and measurement variances.

3.2.5 Kalman Filter Based Feature Conditioning Methods

Typically, it is very difficult to obtain the measurement and motion model variances as there is no information about the true state of a system. For example, while measuring the position of a falling object by radar (Appendix D), there is no information about the object's true position, but there are radar measurements and the equation of falling objects is well known. Given such information, it is possible to predict the path of the object and "double check it" with the radar's measurement. To determine if the object's

position is most likely on the equation's path or most likely given by the measurement, the KF must evaluate which is most confident.

Similarly, in feature conditioning for myoelectric control systems, there is a measurement (the class decision) and a known (or assumed) motion evolution of the features in time. The motion model variance was assumed to be proportional to the extracted feature variance as these are a noisy record of the true features (see Appendix E for the derivation of the motion model variance). The measurement model's variance was designed by evaluating the *probability* of the class decisions. Class "probability", in this context, can be obtained *a priori* by computing a confusion matrix using offline training data (see Appendix E for the derivation of the measurement model variance and Appendix B for a note on the confusion matrix computation) which makes it easy to determine if the classifier is commonly mislabeling one class for another. For example, if the class decision changes from flexion at time *t-1* to pronation at time *t*, and training samples of pronation were commonly labeled as flexion, the measurement would have low confidence (i.e. high variance).

Because the extracted features have relatively low variance while users maintain steady-state contractions (whereas high variance is expected while users transition between classes), such a modeling has the effect of forcing features to stay constant during steady-state contractions while allowing them to converge towards the new classes when performing a class transition.

From the motion and measurement model variances (R_t and Q_t respectively), it is possible to compute the Kalman gain as a function of the models' variances [106] (see [106] for the gain's complete definition):

$$K_t \propto \frac{R_t}{R_t + Q_t} \qquad (3\text{-}12)$$

If the motion model variance is larger than the measurement model variance, the gains are closer to one which shifts the estimate towards the measurement, whereas if the measurement model variance is larger the gains are closer to zero shifting the estimate towards the predicted features. A complete definition of the Kalman gain and the Kalman filter algorithm for multi-dimensional problems can be found in Appendix D. A feature estimate update given the measurement and motion models with approximated variances is illustrated in Figure 3-3 (for a 2-dimensional problem):

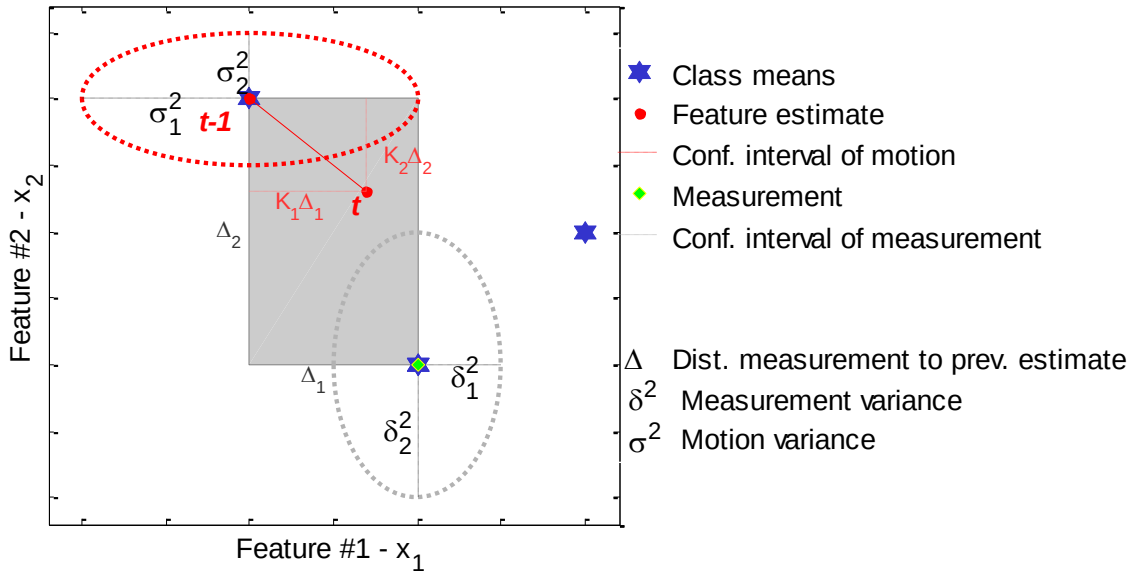

Figure 3-3: Feature estimate update given the measurement and motion models. The measurement model had lower variance in the dimension of Feature#1 whereas the motion model had lower variance in the dimension of Feature#2. Therefore, the current estimate at time *t* fell closer to the measurement in the dimension of Feature#1 and closer to the previous estimate (i.e. predicted features) in the dimension of Feature#2

In this figure, the measurement variance is smaller in the dimension of Feature#1, whereas the motion model variance is smaller in the dimension of Feature#2. Hence the

estimate in the x_1 dimension is closer to the measured features whereas the estimate in the x_2 is closer to the previous estimate at time *t-1* (i.e. the predicted features). Herein, this method is referred to as the *KF with approximated probability*.

The KF method presented tracks the TD feature positions; therefore, the KF state vector has increasing number of features for increasing number of channels (i.e. 4 features per channel). It has been proposed that, during class transitions, the features are in an indeterminate state between contraction types [107]. It may be advantageous to track the Euclidian distances (ED) (Appendix A) between the features and the class means, as this may allow rapid detection of class convergence. In this case, the feature vector has N_C dimensions where N_C is the total number of classes and is independent of the number of channels.

In KF theory (Appendix D), the initial conditions are critical. If these are poorly chosen, the motion model is not initialized properly resulting in an unstable KF that may never converge towards a desired state. In this work, the initial conditions can be determined by assuming that the user begins in a known class, such as no movement.

3.2.6 Feature Conditioning Methods Summary

The feature conditioning methods proposed here estimate the current feature positions given the motion model of the features and a measurement. Careful design of the feature conditioning methods may allow the features to remain constant (within a class) while users maintain steady-state contractions and to converge more quickly towards new classes when users transition to a new class. They have for effect of forcing the features to converge towards the class means while users perform steady-state contractions.

The feature conditioning methods presented can be grouped into two approaches: simple methods and KF based methods. All methods update the features from window to window by assuming that the current feature estimates are between the previous estimates (motion model) and the position of the current class center (measurement model). The simple methods assume that the features lie at the mid-point between the position predicted by the motion and measurement models, whereas the KF methods update the features using dynamically varying weights.

Two simple methods were presented: one that updates the features more aggressively, while the other updates the features more conservatively. While the former performs feature updates at each window, the latter only performs updates when two consecutive similar class decisions are obtained. Two KF methods were presented: one that evaluates the TD features, while the other evaluates the Euclidian distance (ED) features (Appendix A). For comparison purposes, conditioned features will be compared to unconditioned features. The feature conditioning methods are summarized in Table 3-1 along with how they are denoted for the remainder of this work. Unconditioned features are referred to as features without conditioning (i.e. No-cond).

Table 3-1: Feature conditioning methods evaluated

	Unconditioned	**Simple conditioning**		**Kalman filter conditioning**	
Denoted:	**No-cond**	**S_{aggr}**	**S_{cons}**	**KF_{td}**	**KF_{euc}**
Features:	TD	TD	TD	TD	ED
Update rate:		Aggressive	Conservative	Aggressive	Aggressive
Motion variance:				Variance of extracted features	
Measurement variance:				Probability of the class	

It is worth mentioning that all the different setting permutations (i.e. KF or simple base feature conditioning method, conservative or aggressive updates, and TD features or Euclidian distance features) were explored. The methods listed in Table 3-1 and denoted as *No-cond*, S_{aggr}, S_{cons}, KF_{td}, and KF_{euc} were the only ones reported in this work as they showed the most prominent results. It is also worth mentioning that different KF measurement models were evaluated. Those worth mentioning are briefly discussed in Appendix E.

3.3 Feature Conditioning Analysis

To investigate the anticipated benefits of feature conditioning, a study comparing the steady-state and class transition performances for the conditioned and unconditioned features was performed offline. This section describes the offline data collection performed, the data processing, and both the analysis of steady-state and class transition performance of the feature conditioning methods.

3.3.1 Offline Data Collection

Fifteen healthy subjects (aged 20-48) with no known motor or sensory impairment and no amputation participated in the experiment. To minimize learning effects, these subjects had some experience with EMG use. Data acquisition was done using a custom Matlab-based acquisition and control environment (ACE) [108]. Subjects were fitted with a cuff (illustrated in Figure 3-4) containing eight equally spaced bus-based smart myoelectric electrodes [109]. The cuff was placed on the dominant forearm around the area of largest muscle bulk. Data were sampled with a sampling frequency of 1000Hz using a 16-bit analog-to-digital converter.

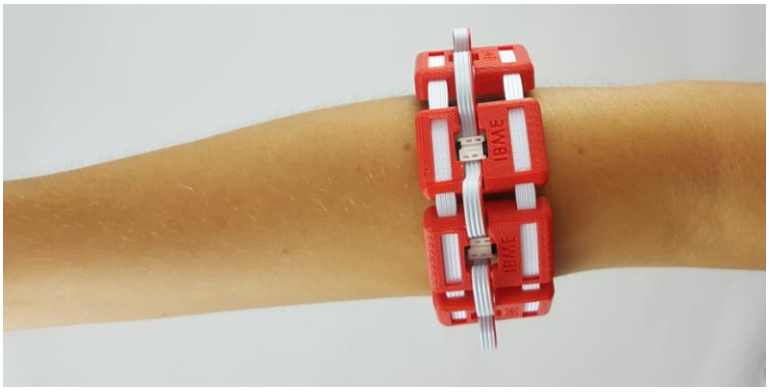

Figure 3-4: Cuff fitted on subjects during offline data collection.

Training and testing data were acquired while performing contractions consistent with wrist flexion, wrist extension, wrist supination, wrist pronation, chuck grip, hand open, and no motion. For training, subjects were visually prompted the contraction in the ACE virtual environment. Subjects were not given any feedback. Each contraction was performed four times and in random order. The subjects held the contractions for 3 seconds and had 2 seconds break between contractions.

After training, testing data were collected. Subjects were prompted to elicit both sustained constant force contractions and transitory contractions by performing transitions between different contraction types. Users transitioned between the different active contractions randomly and were given 3 seconds to perform the transition. A trial was completed after every permutation of transitions between contraction types was completed (7 different contraction types x 6 possible contraction types to transition to for a total of 42 transitions per trial). Subjects started a trial from rest and were asked to complete transitions comfortably with a subjectively moderate intensity. No feedback was provided to the users, allowing the data to be analyzed *a posteriori* using various control configurations. Although such a series of contractions may not be representative

of activation patterns seen during functional use of pattern recognition based myoelectric control scheme, these data were constrained to simple sequential transitions between different contraction types to allow for offline analyses. Subsequent real-time analyses are discussed in Chapter 5.

Empirical observations found that under such conditions, it would take subjects between 0.2 and 1 seconds to react to the prompted class, and 1 to 1.5 seconds to transition to the prompted class, making it likely that at least 1 second of steady state data were collected. A trial was repeated 3 times. Subjects were allowed a break between each trial. As a result, each subject performed 126 transitions from active-to-active class. These data were used throughout this work.

3.3.2 Data Processing

The training data and testing data from each channel were segmented into frames of 5 ms, 10 ms, 20 ms, 30 ms, 40 ms, 50 ms, 100 ms, 150 ms, 200 ms, 250 ms, and 300 ms in length with 5 ms frame increments. From these frame lengths, a set of time domain (TD) features was computed as in [110]. Included in the TD feature set are: the mean absolute value (MAV), the waveform length (WL), the number of zero crossings (ZC), and the number of slope sign changes (TURNS) for a given data window (Appendix A). The Euclidian distance features (Appendix A) were also computed from the preprocessed data segments of different frame lengths.

An LDA classifier (Appendix B) was trained using the TD features extracted from the training data. It was used to build the feature conditioning methods which were then applied to the extracted features from the testing data. The inputs to the feature conditioning methods are the extracted features and the initial conditions (see Appendix

E for an iterative example). The initial conditions are set to be the class center of the no-motion class as users began data collection while in rest.

The conditioned and unconditioned features were then classified using LDA classifiers. Four different LDA classifiers were trained; one with 2 channels, 4 channels, 6 channels, and 8 channels of features extracted from the training data. These were tested with 2, 4, 6, and 8 channels of data, respectively, of conditioned and unconditioned features. This allows evaluation of channel effect on the results.

3.3.3 Feature Conditioning Effect during Steady-State Contractions

This section evaluates the effects of feature conditioning on the steady-state portion of the contractions performed by users. It was anticipated that users reached steady-state sustained force contractions 2 seconds after a class was prompted. Therefore, the last second of each prompted contraction was evaluated.

3.3.3.1 Assessment Metrics

The steady-state *signal-to-noise ratio* (SNR)[2] was computed for each feature estimate where deviations about the mean value of the estimate were considered as noise [111]. Here, the SNR is defined as:

$$SNR(dB) = 10\log_{10}\left(\frac{\mu}{\sigma}\right) \qquad (3\text{-}13)$$

where μ is the mean and σ the standard deviation of the features computed over the last 1 second of each contraction.

[2] The SNR cannot properly be computed during class transitions as the signal is not stationary.

To measure the system's ability to maintain correct decisions, the *steady-state classification accuracy* (A_{CC}) [82] (defined as the percentage of correct class decisions over the final 1 seconds of the prompted contractions) was computed.

Since these metrics were computed on the last second of each contraction, this resulted in a total of 1890 values (7 contractions types x 6 different contractions to reach x 3 trials x 15 subjects) for each metrics.

Statistical analyses of the metrics (SNR and A_{CC}) were performed. The tests performed can be grouped into three types:

1) The first group of tests evaluate the FL, number of channel, and subject effects;
2) The second group of tests determined if the proposed feature conditioning methods significantly change the results when compared to the system without feature conditioning, and;
3) The third group of tests determine if the proposed methods obtained significantly different results from those obtained using a system commonly found in the literature [46] [89] (i.e. LDA classification, no-conditioning with FL = 150 ms). This configuration is referred to as the *reference configuration* for the remainder of this work.

All statistical analyses were conducted using a multi-factor ANOVA test with repeated measurements (random subjects). The other factors were fixed and included the following: trial, number of channels, and/or FL. An alpha level of 0.05 was considered significant.

3.3.3.2 Results

(i) Signal-to-Noise Ratio (SNR)

The average feature SNR obtained with and without feature conditioning, along with their standard error, are illustrated in Figure 3-5 for varying FL.

The SNR was first averaged across the 4 TD features of a single channel or the 7 ED features prior to being averaged across each contraction. The results show that the SNR degrades for shorter FL and that the feature conditioning methods noticeably improved the average feature SNR. Improvement in SNR for longer FL can be attributed to the increased amount of temporal information which results in a decrease in feature variability [42], whereas improvement in SNR while using the feature conditioning methods is expected because they force the features to converge towards class means when users perform steady-state contractions.

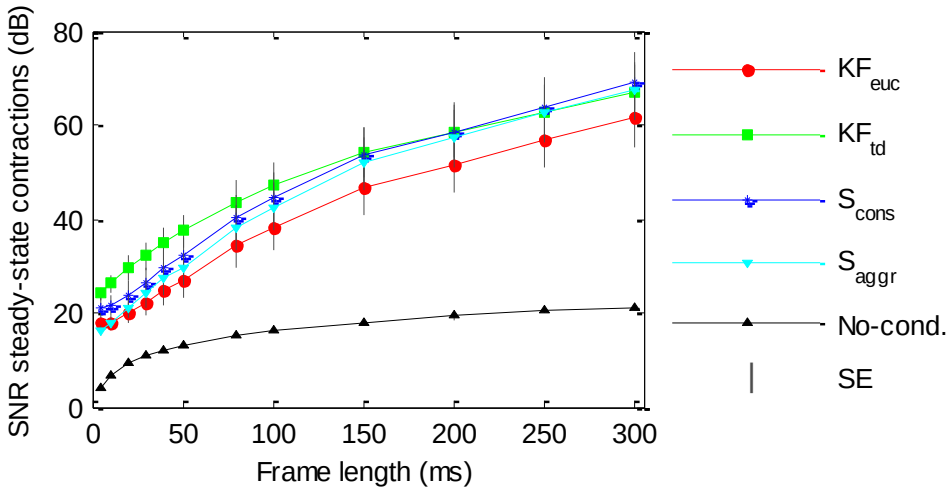

Figure 3-5: The SNR of the features computed on the steady-state contractions (last 1 second of the contractions). The SNR is averaged across the TD features of a single channel or all the ED features. The vertical bars represent the standard error.

When performing statistical analysis on the results, it was noticed that the degradation in SNR for short FL, and that the improvement in SNR when using feature conditioning were significant ($p < .05$). It was also noticed and worth mentioning that the subjects only

had a significant effect on the results ($p < .05$) when using the method without feature conditioning. This suggests that the feature conditioning methods reduced the effect of inter-subject variability on the SNR of the features. This may imply that the feature conditioning methods' ability to force the features towards the class means has a reasonably consistent effect amongst subjects.

The statistical analyses also revealed that most methods obtained significantly ($p < .05$) different results when compared to the reference configuration. In fact, only the features of the S_{aggr} with FL < 20 ms and the KF_{euc} with FL < 20 ms did not obtain a significantly better SNR than the reference configuration. These results along with those in Figure 3-5 imply that, as the FL becomes smaller, the SNR of features (without feature conditioning) significantly degrade, but can be significantly improved when using any feature conditioning methods with FL > 20 ms.

(ii) Steady-State Classification Accuracy

Figure 3-6 summarizes the steady-state A_{CC} results obtained while using the conditioning methods (and without feature conditioning) for different FL and number of channels.

The steady-state A_{CC} was computed using only the last second of each contraction to ensure that users had reached steady-state. Because of the high variability between the different targets (i.e. flexion, extention, etc) and between subjects, the values displayed first computed the median across contractions for each trial (42 contractions per trial), than computed the median value across trials (3 trials per 15 subjects for a total of 45 trials). The standard error was computed over the median values obtained for each trial (45 values total). Results displayed are for FL $\leq$ 200 ms because of a noticeable

plateauing of steady-state A_{CC} after FL $\geq$ 150 ms. This further supports the common occurrence of FL = 150 ms that is often found in the literature [29] [46].

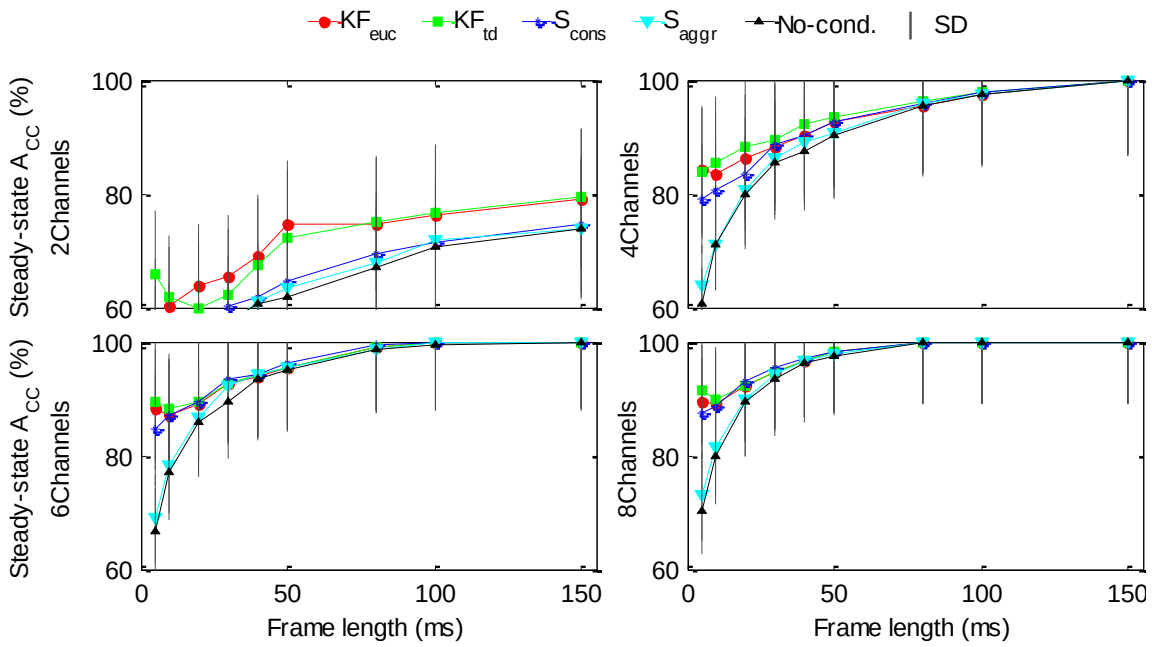

Figure 3-6: Steady-state classification accuracy computed on the last 1 second of each contraction. The vertical bars represent the standard error.

The results confirm that the steady-state A_{CC} improves with increasing FL. Once again, this can be attributed to the increased amount of temporal information which results in a decrease in feature variability [42] (i.e. SNR). The feature conditioning methods moderately improved the steady-state A_{CC} for short FL. This improvement (which was significant ($p < .05$) for FL $\leq$ 20 ms when using the KF_{euc}, KF_{td} and S_{cons} and for FL = 5 ms when using the S_{aggr}) is likely due to the fact that the feature conditioning methods improve the SNR of the features (Figure 3-5). The literature has suggested that an improvement in features SNR should result in better classification accuracy [102]; however, the improvement in SNR for longer FL provided by the feature conditioning methods (Figure 3-5) did not result in an improved steady-state A_{CC} (Figure 3-6). This suggests that the steady-state misclassifications are not only due to feature

variability (they could be due to users selecting the incorrect class for example) and that the effects of improved SNR eventually settles at higher levels of SNR.

It can be seen that increasing the number of channels had a greater effect on the steady-state A_{CC} than the feature conditioning methods. In fact, without feature conditioning, there is a significant ($p < .05$) 5% increase when using 8 channels over 4 channels (but no significant increase while using 8 channels over 6 channels, nor 6 channels over 4 channels). The literature has shown that increasing the number of electrode sites (or channels) improves classification accuracy [40] and attributes it to a more complete coverage of the muscles involved in the contraction [46]. A large number of channels is sometimes impractical due to limitations in the available residual limb musculature, space in a socket, and lack of physical recording surface [40]. When comparing the steady-state A_{CC} obtained using configurations with 8 channels of unconditioned features to those obtained using 4 channels of conditioned features (notably that KF_{euc}, KF_{td} or S_{cons}) results were comparable (i.e. no significant differences were noted). These results suggest that these feature conditioning methods may prevent the significant degradation in steady-state A_{CC} caused by a reduced number of channels, and that the use of 4 channels may be a practical tradeoff when features are conditioned.

Figure 3-7 compares the steady-state accuracy of the different configurations to the reference configuration (i.e. no feature conditioning and FL = 150 ms). In this plot, median values are computed as in Figure 3-6 are shown and a negative value indicates degradation in steady-state A_{CC} when compared to the reference configuration with 150 ms FL. These results show that, with or without feature conditioning, as the FL becomes shorter, the steady-state A_{CC} degrades (when compared to the control configuration).

However, feature conditioning methods and configurations with more channels can sustain shorter values of FL before incurring significant ($p < .05$) degradation in steady-state A_{CC}. For example, the configurations with KF_{euc} obtained significantly worse results from the reference configuration for FL $\leq$ 40 ms when using 4 channels and for FL $\leq$ 30 ms when using 8 channels, whereas the configurations without feature conditioning obtained significantly worse results for FL $\leq$ 50 ms (regardless the number of channels). These results, contrary to the literature [46], suggest that commonly referenced configurations could employ FL as short as 80 ms without significant degradation in the steady-state A_{CC}. This discrepancy with the results found in the literature could be attributed to the fact that users may not always reach steady-state contractions during the last second of a contraction which may generate results with higher variability.

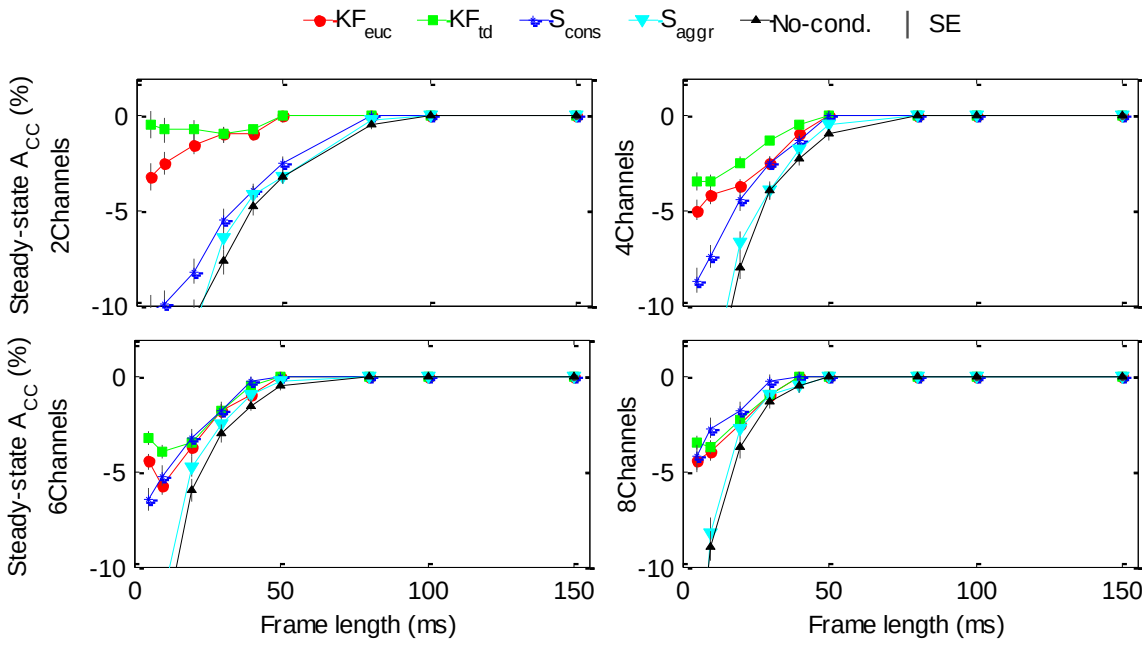

Figure 3-7: Difference in the steady-state classification accuracy between the different configurations and the reference configuration without feature conditioning and FL = 150 ms for varying frame length and number of channels. The vertical bars represent the standard error. A negative value indicates degradation in steady-state A_{CC} when compared to the reference configuration with 150 ms FL.

The results also suggest that, to prevent steady-state A_{CC} degradation for shorter FL, it may be best to use the KF_{euc} or KF_{td} over the simple configuration methods (S_{aggr} or S_{cons}). In fact, configurations with 4 channels and simple feature configuration methods obtained significantly worse steady-state A_{CC} when compared to the reference configuration for FL $\leq$ 50 ms whereas this occurred for FL $\leq$ 40 ms while using the KF based feature conditioning methods (i.e. KF_{euc} and KF_{td}). This could be due to the fact that the KF methods perform statistical updates of the features based on the confidence of the previous features and the current class output, whereas the simple methods assume that the features are at midpoint between the current class output and the previous features.

3.3.4 Feature Conditioning Effect on Transitions

Pattern recognition systems have been characterized primarily by use of classification accuracy [5] [7] [82] [89], however it has been reported that accuracy may not fully describe, or predict, the systems' usability [82]. This may be due to the fact that it has mostly been used to measure offline performance during sustained force contractions which differs from the dynamic nature of functional control. In this section, unlike in the previous section, the transitory portions of the data were evaluated and metrics were computed over the entire class prompts (unlike the steady-state A_{CC} and SNR in the previous section which were computed on the last second of the contractions). Since using 4 channels with feature conditioning showed a practical tradeoff, only configurations with 4 channels were evaluated further.

3.3.4.1 Assessment Metrics

It is challenging to evaluate the performance of systems during class transitions because the features are in an undetermined state. It was assumed that systems allowing users to complete the most contractions and to rapidly transition between contractions would provide users with the best dynamic control.

Therefore, the failed completion rate (FC rate), evaluating the users' ability to complete the prompted transitions, was computed and defined as the percentage of transitions that were not completed over an entire trial. A transition was considered failed if subjects were unable to sustain the prompted class for a specified number N of consecutive windows. Since the frame increment was of 5 ms, N windows require at least $(5 * N + FL)$ ms of time for users to correctly select a contraction for the transition to be completed. The choice of N has a direct impact on the FC rate and this was evaluated in Appendix F. In this Appendix, it can be seen that increasing N will result in higher values of FC. The value of N was chosen to be 30 windows because 1) for N>30, subjects obtained an average FC rate greater than 20%, and 2) the average value of FC rate was comparable across most FL for systems with conditioned features (notably the KF_{euc}, KF_{td} and S_{cons}). The FC rate was computed on the class decision obtained in Section 3.3.3 for each trial resulting in 45 values (3 trials x 15 subjects).

The time it took users to reach steady-state was also computed by evaluating the *time to reach N=30 correct consecutive decisions (τ_N)*. The τ_N is the time elapsed between the moment a class is prompted to the time the user was capable of sustaining *N=30* correct consecutive decisions. In the case where users failed the transitions, τ_N was given a value of 3 seconds (the maximum allotted time to complete transitions). Since user delay and

reaction times are the same for each system, it is anticipated that systems with shortest τ_N allow users to perform transitions in a shorter period of time. The τ_N was computed on the decision stream obtained in Section 3.3.3 for each transition. This resulted in a total of 1890 values (42 transitions x 3 trials x 15 subjects).

Statistical analyses of the metrics (FC rate and τ_N) were performed in a similar fashion as those performed on the steady-state A_{CC} and SNR (Section 3.3.3.1).

3.3.4.2 Results

(i) Failed Completion (FC) Rate

Figure 3-8 shows the average failed completion rate (left) and the difference in failed completion rate when compared to the reference configuration (right) for the different methods evaluated (a negative result indicates better FC rate):

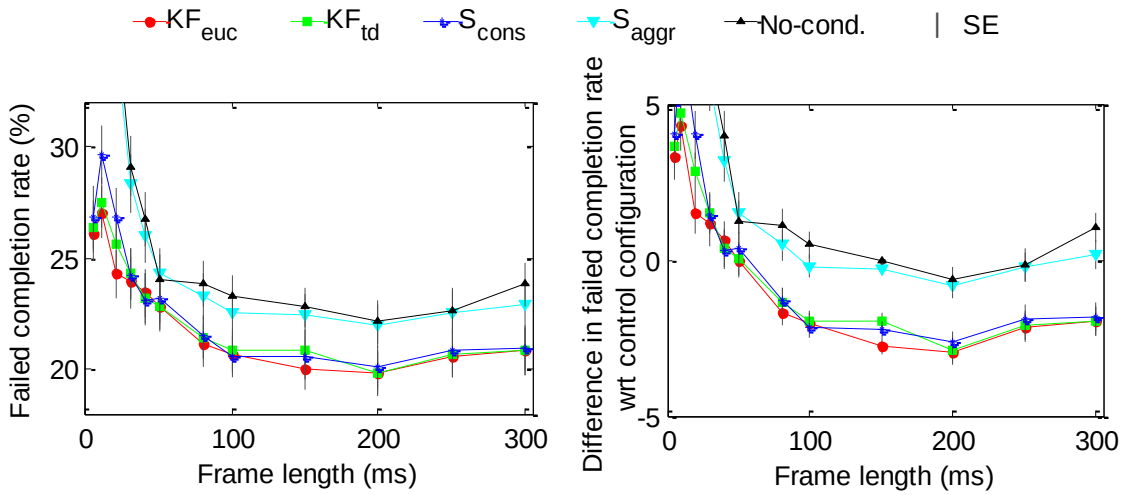

Figure 3-8: The left hand plot shows the average percentage of failed completion rate for varying frame lengths. The right hand plot shows the difference in failed completion rate when compared to the reference configuration. A transition was failed if subjects were unable to consecutively select 30 correct decisions. The vertical bars represent the standard error.

A transition was completed once subjects were able to select the prompted class for 30 consecutive windows. The results on the left hand plot show that the failed completion

rate was high suggesting that the tasks were challenging: the subjects had only 3 seconds to both react to the prompted class and complete the transition. Increasing the FL as well as using feature conditioning methods (with the exception of the S_{aggr}) resulted in increased successful transitions. Statistical results show that the FL had a significant effect on the results (i.e. longer FL significantly improved results), and that for FL $\leq$ 30 ms the KF_{euc}, KF_{td} and S_{cons} obtained significantly ($p < .05$) better results than the configurations without feature conditioning, whereas the S_{aggr} obtained significantly ($p < .05$) better results than the configurations without feature conditioning for FL $\leq$ 10 ms. These results are expected as longer FL and the feature conditioning methods improved the SNR (Figure 3-5). These results suggest that minimizing the feature variability should therefore allow a user to select a class more reliably and potentially faster. The results also suggest that the KF and the S_{cons} methods resulted in more successfully completed transitions than the S_{aggr} method. This could be due to the different way with which the feature conditioning methods update the features. As stated previously, the KF_{euc} and KF_{td} determine feature updates statistically while evaluating both the feature and the class decision confidence, whereas the S_{cons} determines updates by evaluating only the confidence of the class decisions. The S_{aggr}, on the other hand, averages the previous estimate with the class output mean; this suggests that using a simple averager may be insufficient to improve transition performance, and that some logical decision must supplement the feature estimation method in order to facilitate a user's ability to reach a sustained contraction.

The results on the right hand plot of Figure 3-8 show the difference in failed completion rate when compared to the reference configuration. A negative result

indicates that the configuration obtained better FC rate than the reference configuration. For FL $\geq$ 50 ms the configuration without feature conditioning and with S_{aggr} obtained comparable results (with no significant difference) to the reference configuration. For shorter FL, the aforementioned configurations obtained significantly worse FC rate when compared to the reference configuration. The other feature conditioning methods (KF_{euc}, KF_{td} and S_{cons}) obtained better but yet comparable results (with no significant difference) to the reference configuration for FL $\geq$ 20 ms when using the KF methods and for FL $\geq$ 30 ms when using the S_{cons}. For shorter FL than those just mentioned, the aforementioned configurations obtained significantly ($p < .05$) worse FC rate when compared to the reference configuration.

These results suggest that a configuration with KF based feature conditioning can sustain shorter FL (20 ms) than the simple methods (FL = 30 ms) before obtaining a significantly higher FC rate than the reference configuration. They also suggest that increasing the FL > 50 ms did not allow a significant improvement in FC rate.

(ii) Time to reach N Correct Consecutive Decisions (τ_N)

The average time it takes subjects to select 30 consecutive correct decisions ($\boldsymbol{\tau}_{N=30}$) and the average difference with respect to the reference configuration (a negative result indicates that the configuration obtained better results than the reference configuration) are illustrated in Figure 3-9.

The results in Figure 3-9 illustrate the impact of failing transitions; for short FL, subjects had more difficulty, on average, of selecting 30 correct consecutive windows, and this is especially true when using unconditioned features (No-cond) or features conditioned using the S_{aggr}.

Statistical analysis comparing configurations with shorter FL (FL < 150 ms) to the reference configuration (FL = 150 ms) showed that configurations with KF_{euc}, KF_{td} or S_{cons} and between 40 ms ≤ FL ≤ 100 ms obtained significantly better results than the reference configuration. These results and those illustrated in Figure 3-9 suggest that users were most likely to complete transitions ~100 ms faster while using configurations with KF_{euc}, KF_{td} or S_{cons} with FL between 40 ms and 100 ms than while using the reference configuration.

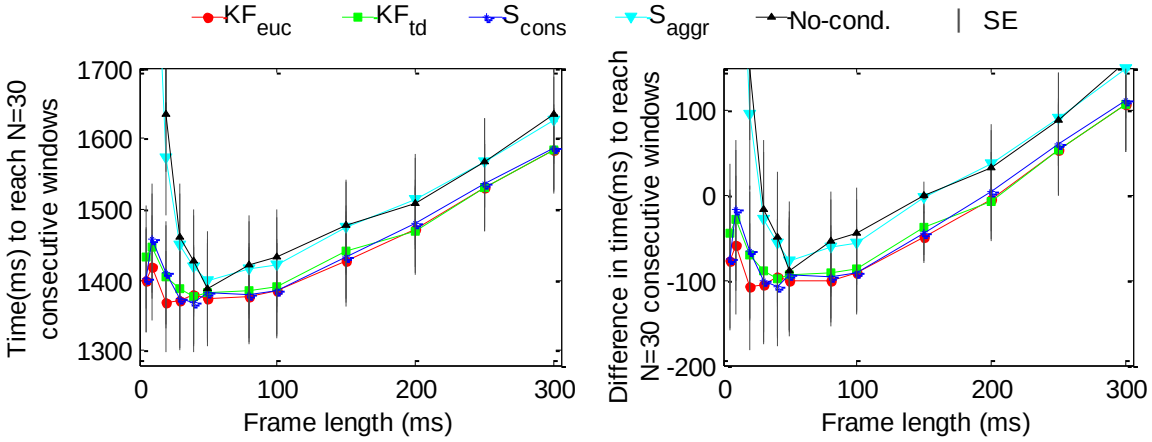

Figure 3-9: The left hand plot illustrates the time elapse between the moment a class is prompted and 30 consecutive correct decisions for varying frame length and configurations. The right hand plot illustrates the difference in the time to reach 30 correct decisions when compared to the reference configuration. Transitions where users were unable to reach 30 correct decisions were given times of 3 seconds (equal to the total amount of time users were given to complete transitions).

The results in Figure 3-9 also illustrate the impact of delay often caused by long FL; for longer FL, it takes users more time to select 30 consecutive correct decisions. Statistical analysis comparing configurations with longer FL (FL > 150 ms) to the reference configurations (FL = 150 ms) showed that:

- The configurations with unconditioned features (No-cond) and features conditioned with S_{cons} or S_{aggr} obtained significantly higher times than the reference configuration for FL $\geq$ 250 ms, and;
- The configurations with KF_{euc} or KF_{td} conditioning obtained significantly ($p < .05$) the worst results than the reference configuration for FL = 300 ms.

These results suggest that, when using longer FL (especially FL = 250 ms), it is best to use the KF feature conditioning methods to prevent a significant increase in delay.

3.4 Chapter Summary

This chapter presented a novel feature conditioning method to supplement current myoelectric pattern recognition systems. The design of the feature conditioning methods was motivated to minimize system response time without cost in steady-state A_{CC} and to minimize the amount of time that features were in an undetermined state relative to the trained classes.

The literature reports that increasing the FL reduces features variability [22] [23] [26] [27] [29] which has a direct impact on classification accuracy [42] [47]. Long FL, however, increases system response time [11] [46]. Short system response time is important when using myoelectric prostheses dynamically so that an excessive delay is not introduced to the controller [46]. It was anticipated that the feature conditioning methods would allow the use of shorter FL without cost in steady-state classification accuracy. The results showed that the features SNR decreased for shorter FL causing a drop in steady-state A_{CC}; however, feature conditioning prevented significant SNR degradation in the features while sustaining steady-state A_{CC} for shorter FL when compared to the reference configuration. The choice of the feature conditioning method

may also impact results; it was found that for shorter FL it may be best to use KF based conditioning methods.

These results suggest that the feature conditioning method have the potential of improving system response time while in dynamic conditions as they allowed the use of shorter FL with no cost in steady-state A_{CC}. While evaluating the time it took users to reach 30 correct consecutive decisions, the results showed that conditioned features (except those conditioned using S_{aggr}) with FL between 40 ms and 100 ms allowed users to complete transitions ~100 ms faster than while using the reference configuration; suggesting a faster response time to contraction changes.

The literature reports that features are in an undetermined state while in class transitions [67] [104] making it difficult to properly classify the patterns. It was anticipated that since the feature conditioning methods force the patterns towards the class means that this would reduce the amount of time that patterns were in an undetermined state. Although it is not always clear when the system is in an undetermined state nor is it obvious how to evaluate performance in such condition, it was anticipated that a system allowing users to complete tasks faster as well as complete more tasks was in an undetermined state for a shorter period of time. Conditioning the features reduced the time it took users to complete transitions (when using optimal configurations), and allowed users to complete more transitions. In fact, reducing the FL of conditioned features had no significant effect on the FC rate (for FL $\geq$ 20 ms) when compared to the reference configuration.

Another benefit of supplementing myoelectric prostheses with feature conditioning is that the system can have 4 channels while obtaining comparable steady-state performance to the reference configuration with 8 channels.

Although the results obtained from the SNR, steady-state A_{CC}, failed completion (FC) rate and the time to reach N=30 correct consecutive decisions ($\tau_{N=30}$) are promising and allowed some insight as to which feature conditioning methods to use (for shorter FL it is best to use KF based methods), this Chapter evaluated the effects of feature conditioning on a portion of the proposed system: the feature estimations. The next Chapter evaluates the systems decision logic and how feature conditioning can potentially improve the system's ability to make class decisions.

Since the S_{aggr} did not provide significant improvement when compared to unconditioned features, this feature conditioning method was not evaluated further.

Chapter 4 – Decision logic

4.1 Introduction

In Chapter 2, it was proposed that a system capable of providing users with reliable control while in class transitions should:

1) Improve the SNR of EMG features for short FL in order to minimize controller delay without compromising classification accuracy;
2) Prohibit features from staying in an indeterminate state for long periods of time;
3) Separate class transitions and steady-state contractions to allow appropriate decision logic for each, and;
4) Improve the decision making strategy during class transitions.

Figure 4-1 illustrates the proposed system, consisting of *feature estimation* and *decision* logic stages.

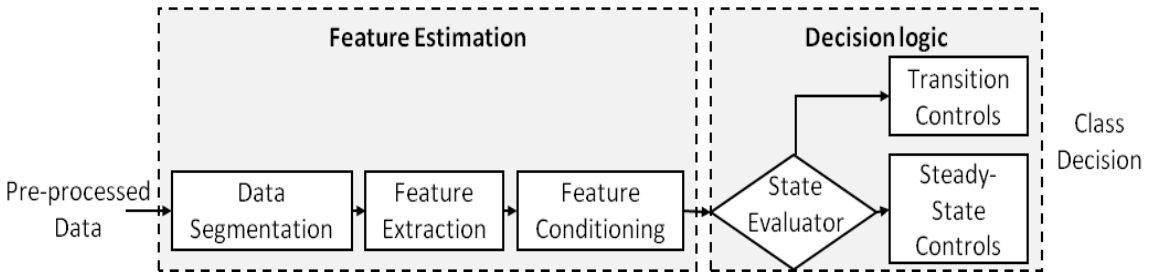

Figure 4-1: Block diagram of proposed system consisting of two parts: 1) feature estimation which includes data segmentation, feature extraction, and feature conditioning, and 2) decision logic which includes state classification using a state-evaluator, transition controls, and steady-state controls.

The previous chapter addressed the feature conditioning block illustrated in Figure 4-1 and evaluated topics 1) and 2) listed above. It was found that the proposed feature conditioning methods improved the SNR and the time it took users to complete 30 consecutive correct decisions without significantly affecting the failed completion rate or

the steady-state classification accuracy. This chapter addresses the *decision logic* block illustrated in Figure 4-1 and topics 3) and 4) listed above. The proposed method is referred to as *state-based classification*.

The *state-evaluator*, separating class transitions from steady-state contractions, is presented in Part I of this chapter, and different decision making strategies are presented in Part II and compared to current control methods found in the literature.

4.2 Part I: State-Evaluator

The goal of this section was to design a state-evaluator to detect class transitions. Other class transition detection methods [45] [72] [88] have been described in previous work. These methods condition the signal using an operator (such as the Teager-Kaiser energy operator [87]) to improve the SNR of the EMG. This facilitates the detection of the onset of motion [45] [88] or any sudden changes in EMG variability [72] which are often representative of an upcoming class transition. Such methods increase latency [72], depend on empirically determined thresholds [72] [88], and/or are limited to contractions performed starting from rest [45]. This work proposes that conditioning the features, rather than the signal, simplifies class transition detection.

During steady-state contractions, the feature conditioning methods described in Chapter 3 forced the features to converge towards the class means, improving their SNR. While in class transitions, the features travel along some path between the current and next class means. As a result, the proposed state-evaluator partitions the features into two distinct states:

1) The *steady state class* that occurs when the features are held up at the class means, and;

2) The *indeterminate state* that occurs when the features travel between two classes.

For simplicity, and since the data collection required users to perform simple transitions between different contraction types, it was assumed that the system was in an indeterminate state only while in class transitions. Although this may not always be the case (for example, users performing a contraction that is unknown to the classifier results in a system that is in an indeterminate state), it was proposed that detecting these states would allow the system to isolate potential class transitions from steady state classes to prevent unintentional activation of the prosthesis.

4.2.1 State-Evaluator Definition

The state-evaluator measures the Euclidian distance of the features from the class means (Appendix A) to determine if the features are either in steady-state or in an indeterminate state. Since the feature conditioning methods force the features towards the class means, it is anticipated that, if the distance from the nearest class is non-zero (or above a specified threshold), then the state-evaluator is positive indicating that the features are in an indeterminate state, otherwise the output of the state-evaluator is zero indicating a negative event and the user is assumed to be in a steady state class. It is anticipated that the threshold should, not only be zero (or close to zero), but also be the same across the different subjects and contractions types.

Mathematically, the state-evaluator can be represented as follows:

$$D_{ED} = \begin{cases} 1 & if\ ED_{NN} > THRS \\ 0 & otherwise \end{cases} \quad (4\text{-}1)$$

where a value of 1 represents positive events and a value of 0 represents negative events, *THRS* is the threshold coefficient, D_{ED} is the state-evaluator's output, and ED_{NN} is the Euclidian distance of the features to the nearest neighboring class center defined as:

$$ED_{NN} = \underset{N_C}{\text{argmin}}(\boldsymbol{x}^{\boldsymbol{ED}}) \quad (4\text{-}2)$$

where $\boldsymbol{x}^{\boldsymbol{ED}}$ is the Euclidian distance feature set. For unconditioned features or features conditioned using the S_{cons} and KF_{td} , the $\boldsymbol{x}^{\boldsymbol{ED}}$ must be computed from the TD features as explained in Appendix A.

The performance of the state-evaluator can be measured by computing the *accuracy*, *precision*, *true positive rate (TPR)*, and *false positive rate (FPR)* [112] [113] [114] which are standard metrics of binary classifiers as defined in Appendix G. Accuracy, defined as:

$$accuracy = \frac{\#\ of\ correct\ outputs}{\#\ of\ outputs} = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN} \quad (4\text{-}3)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative, can measure a binary classifier's ability to correctly identify the true state of a system [112]. For the state-evaluator presented in this work, accuracy implies correct classification between the steady and indeterminate states. Precision, defined as:

$$precision = \frac{\#\ of\ correctly\ classified\ indeterminate\ feature\ vectors}{\#\ of\ indeterminate\ feature\ vectors} = \frac{\sum TP}{\sum TP + \sum FP} \quad (4\text{-}4)$$

evaluates a binary classifier's ability to reproduce the same results when a system's conditions are unchanged [112]. The state-evaluator would have high precision if identical (or quasi identical) features would be classified in the same state.

The TPR and FPR evaluate the correctness of labels [114]; a system that correctly identifies positive (1) labels has high TPR, and a system that incorrectly identify positive (1) labels has high FPR [114]. Ideally, the TPR should be 100% and the FPR should be 0%. This occurs when all positive (1) labels and negative (0) labels of a binary classifier are correctly identified. Unless the latter occurs, there is a tradeoff between the TPR and the FPR [113] [114]; as the error rate increases, either the number of positive (1) output error increases (i.e. FPR) or the number of negative (0) output error increases (i.e. FNR). In the case of the state-evaluator presented in this work, the threshold can be varied to an appropriate compromise using Receiver Operating Characteristics (ROC) (Appendix G) to evaluate the TPR and the FPR obtained using different threshold coefficients.

4.2.2 Data Processing

The state-evaluator was applied to the features conditioned using the KF_{euc}, KF_{td} and S_{cons}, and the unconditioned features (as obtained in Section 3.3.2). The state-evaluator threshold coefficients ranged from 10^{-45} (i.e. close to zero) to 1 (see Equation (4-1)).

The performance of the state-evaluator (for varying thresholds) was assessed by comparing its binary outputs with state labels obtained by visual inspection of the rectified EMG where steady-state contractions and transition segments were labeled as in Appendix H. Automated segmentation could not be performed due to user response time, electro-mechanical delays, and the random nature of the EMG as demonstrated in Appendix H. The state-evaluator should classify the steady-state contraction segments as steady state class, and the transition segments as indeterminate state.

The TPR and FPR were computed from the state-evaluator's outputs. The state-evaluator's threshold was selected using the following equation [114]:

$$i = \arg \max_{n=1:N_{th}} \left(TPR(n) - FPR(n)\right) \quad (4\text{-}5)$$

where i is the index of the selected threshold amongst the $n = 1, 2, 3, \ldots N_{th}$ thresholds ranging from close to zero (10^{-45}) to 1. This equation guarantees that the TPR is higher than the FPR and that the TPR is maximized for minimal FPR.

The accuracy and precision of the state-evaluator (with the selected threshold coefficient) were computed for each trial. This resulted in a total of the 45 values (3 trials x 15 subjects).

Statistical analyses were performed on the selected threshold coefficient, on the accuracy, and on the precision to evaluate the effects of the different feature conditioning methods, subjects, trial, and FL on the results. All statistical analyses were conducted using a multi-factor ANOVA test with repeated measurements (random subjects). The other factors were fixed and included all or some of the following: *trial*, *feature conditioning method*, and FL. An alpha level of 0.05 was considered significant.

4.2.3 Results and Discussion

To improve readability, only the ROC curves of configurations with FL = 150 ms are illustrated in Figure 4-2. The ROC plots using other FL (5 ms, 10 ms, 20 ms, 30 ms, 40 ms, 50 ms, 80 ms, 100 ms, 200 ms, 250 ms, and 300 ms) are illustrated in Appendix G.

From these results, it is noticeable that none of the configurations obtained results near optimum ROC (i.e. 100% TPR for 0% FPR see Appendix G for ROC theory). This is due to the fact that, although features in class transition are likely in an indeterminate state, not all features in an indeterminate state are in class transition. This is illustrated in Appendix G that shows that the state-evaluator is capable of detecting features that result in class changes in the decision stream and that are assumed in an indeterminate. The

state-evaluator separated the steady-state class from the indeterminate state. This work assumed that the system was in an indeterminate state only while in class transitions, but this was not always the case resulting in a lower than expected TPR. Nevertheless, configurations with feature conditioning obtained better ROC than the configuration without feature conditioning. In fact, for the case where feature conditioning methods were not used, the state-evaluator's TPR and FPR increase at the same rate along the diagonal line. This indicates that the state-evaluator was unable to determine the state of the unconditioned features (see Appendix G for ROC theory). When using the feature conditioning methods, the TPR was always higher than the FPR indicating that there are more correct decisions than incorrect decisions. These results suggest that the feature conditioning methods facilitate the state-evaluator's ability to distinguish features in class transitions from those in steady-state contractions (this is true for all FL as illustrated in Figure G-3 of Appendix G).

The results also show that as the threshold becomes smaller (towards the right), both the TPR and the FPR increase. For very small thresholds, the TPR and FPR stop increasing when using the feature conditioning methods. This is because the feature conditioning methods force the features to converge towards the class means; therefore, results eventually stop changing as the threshold becomes close to zero. This indicates that the state-evaluator could be represented by:

$$D_{ED} = \begin{cases} 1 \ \ if \ ED_{NN} \neq {}^{+}0 \\ 0 \ \ otherwise \end{cases} \qquad (4\text{-}6)$$

where ED_{NN} is defined as in Equation (4-2), and ${}^{+}0$ represents a positive value close to zero.

The $\log_{10}$ average and $\log_{10}$ range (i.e. $\log_{10}$(average - standard deviation) - $\log_{10}$(average + standard deviation)) of the selected threshold coefficient (as defined in Equation (4-5)) are illustrated in Figure 4-3.

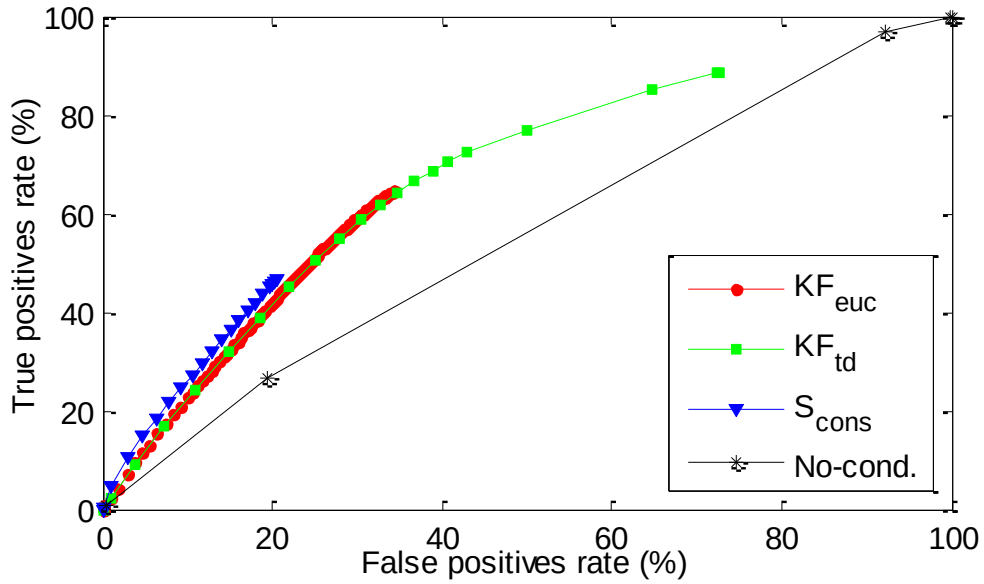

Figure 4-2: ROC plot of the state-evaluator. The configurations had FL = 150 ms and different feature conditioning methods. Thresholds vary from left to right between 1 and 10^{-45}. The true positive and false positive are computed for each trial and threshold, then averaged across trials.

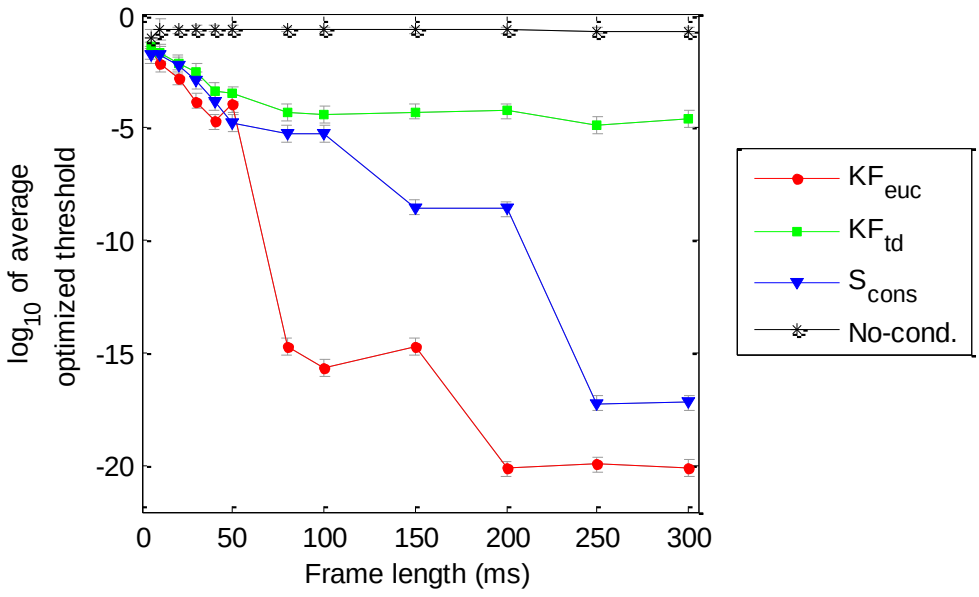

Figure 4-3: Average $\log_{10}$ of optimized threshold value. Values are averaged across each trial. The markers represent the average whereas the vertical bars show the standard error.

For FL > 20 ms, the results show that the threshold coefficient of the state-evaluator while using conditioned features is smaller than while using unconditioned features (No-cond). Statistical analyses showed that the state-evaluator without feature conditioning obtained significantly ($p < .05$) higher threshold coefficient than the state-evaluator with KF_{euc} (for FL > 20 ms), KF_{td} (for FL > 50 ms), and S_{cons} (for FL > 20 ms). Statistical results also showed that the subject had a significant ($p < .05$) effect on the selected threshold coefficient except when using the KF_{euc} conditioning method. This is expected because the KF_{euc} tracks the features Euclidian distance from the class mean which is used by the state-evaluator to make decisions (Equation (4-1)). Since the KF_{euc} obtained such a small threshold and is not significantly affected by the subject, the results suggest that while using this feature conditioning method, the state-evaluator could be defined as in Equation (4-6) for all subjects, whereas while using the other methods (or without feature conditioning), the threshold coefficient may have to be determined for each subject. These results also suggest that the threshold coefficient is dependent on the features FL. This is likely due to the fact that the features SNR changes with varying FL. Hence, features with better SNR can have a state-evaluator threshold closer to zero.

The state-evaluator's accuracy and precision obtained while using the selected threshold are illustrated in Figure 4-4 for all FLs. The results show that the state-evaluator has significantly ($p < .05$) higher accuracy and precision when using conditioned features (KF_{euc}, KF_{td} or S_{cons}) compared to unconditioned features (No-cond). The results in Figure 4-4 also show that the state-evaluator had higher accuracy, but lower precision, while using the KF based conditioning methods (KF_{euc} or KF_{td}) than while using the S_{cons} (for FL > 50 ms).

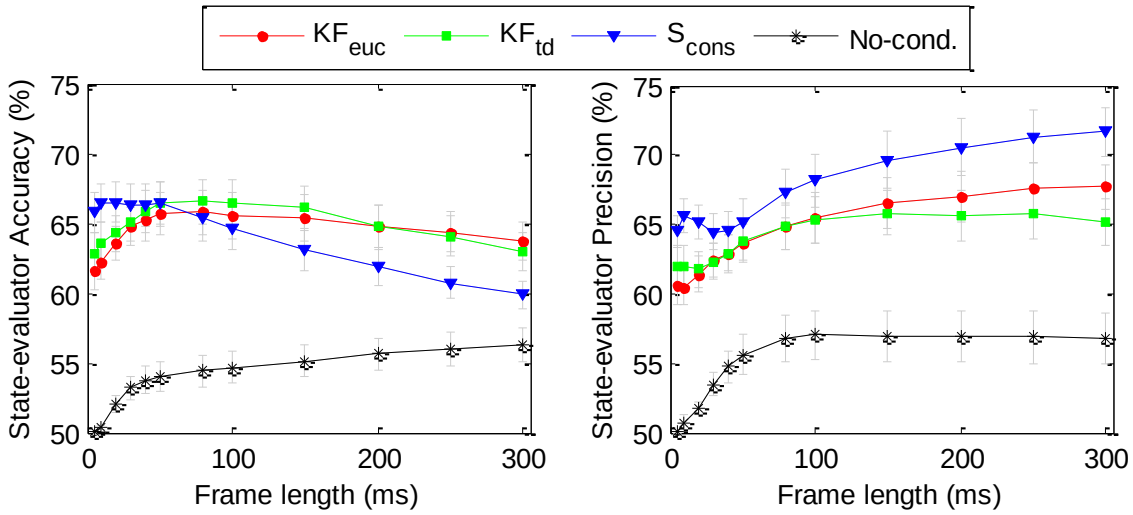

Figure 4-4: State-evaluator's accuracy (left) and precision (right) with and without feature conditioning for FL varying between 5 ms and 300 ms. Results are averaged across all 45 values (15 subjects x 3 trials). The markers represent the average whereas the vertical bars represent the standard error.

The KF methods performed feature updates using statistical methods, whereas the S_{cons} performed feature updates by averaging the current features with the current class mean. These results suggest that the statistical updates may allow the features to reach the class means more often allowing the state-evaluator to make more accurate decisions. Conversely, the state-evaluator's decisions may quickly change between states due to these statistical updates which are less conservative than the feature updates performed by the S_{cons}. This suggests that there are fewer changes in the state-evaluator's decisions when using the S_{cons}, but the state-evaluator may stay in the wrong state more often (and for a longer period of time). Regardless of the method used, results were below 70% for both the accuracy and the precision. As reported previously, this is due to the fact that, although features in class transition are likely in an indeterminate state, not all features labeled as being in an indeterminate state are in class transition. This could result in over-rejection while attempting to control the device (this will be evaluated in Section 4.3). However, as illustrated in Figure G-3 of Appendix G, the state-evaluator with feature

conditioning is capable of detecting features in an indeterminate state with a high level of accuracy; therefore, the state-evaluator may not result in over-rejection, but may rather be detecting not only class transitions, but, more generally, features in an indeterminate state.

4.2.4 Summary

This section presented and evaluated a state-evaluator that labeled the features as being in a steady state class or in an indeterminate state by evaluating the Euclidian distance of the features from the nearest class center. The results showed that the feature conditioning methods significantly improved the state-evaluator's ability to correctly label class transition features as being in the indeterminate state, although not all features in the indeterminate state were in class transition. Additionally, results suggest that the state-evaluator's threshold coefficient can be zero for all subjects when using the KF_{euc}, but it may have to be determined for each subject while using any other feature conditioning method or while using unconditioned features. Finally, it was noticed that the state-evaluator had higher accuracy when using the KF based conditioning methods, but had higher precision when using the S_{cons} feature conditioning method; this could be explained by the different methods at which the feature conditioning methods perform feature updates.

Since the KF_{euc} and KF_{td} obtained comparable accuracy and precision but that the threshold coefficient was easier to determine when using the KF_{euc}, the KF_{td} was not evaluated further.

4.3 Part II: Decision Methods

Current systems based on conventional classification methods are reported to be highly accurate in constrained situations where users perform steady-state sustained contractions. The most commonly reported of these systems use TD features in combination with an LDA classifier [82]. It has also been found that the best FL for such systems ranges between 150 ms and 250 ms [46]. However, misclassifications may occur during class transitions (which are inevitable during real-time use) [45] [94], resulting in unintended activation of the limb, and directly impacting the system's usability [94].

It has been suggested that the effects of unintended class activation can be minimized by rejection of indeterminate state class decisions [90]. This method assumes that inadvertent activations of the limb are more "costly" than those that cause a momentary pause in the motion. This has been attributed to the fact that extraneous motions require the user to elicit additional corrective motions, requiring additional effort and increasing frustration [90]. However, rejection can lead to an overly-inactive system [44] [90] .

Post-processing can also minimize the effects of misclassifications due to class transitions by smoothing the decision stream [44], but at the expense of increased controller delay [44]. Two popular post-processing methods that have shown promising results include majority voting [42], which selects the most frequent class amongst the last decisions, and velocity ramp [94], which attenuates the device velocity when class changes occur in the decision stream. Majority voting has shown promising offline results, but research showed that it does not provide improved performance in real-time because of controller delays [44]. It has been reported that, in real-time, velocity ramp

reduces the impact of misclassification with less delay than MV [44] because it does not change the systems output.

In this work, preliminary analyses evaluated different class decisions strategies for the proposed state-based classification method (Figure 4-1). These included Hidden Markov Models as in [67], support vector machines (SVM), Parzen classifiers, nearest-neighbor classifiers, rejection of indeterminate state, and post-processing. Those worth mentioning are rejection and post-processing as others had comparable results to the commonly used LDA classifier.

Herein, the proposed state-based classification method with rejection of features labeled in an indeterminate state was compared to conventional classification methods and the reference configuration while considering the impact of feature conditioning and of post-processing. This investigation of decision methods is separated into three sections:

1) The first section evaluates conventional classification methods. The goal of this section is to determine the best classification method for each feature conditioning method. These methods are used as a baseline for comparison to evaluate the effects of feature conditioning, rejection, and post-processing on the proposed configuration;
2) The second section evaluates the state-based classification method with rejection by comparing it to conventional classification methods, to the reference configuration, and to other rejection methods found in the literature; and,

3) The third section evaluates the effects of post-processing on conventional classification and rejection methods.

4.3.1 Conventional Classification

The goal of this section was to determine the best classification method for the different feature conditioning methods, as depicted in Figure 4-5.

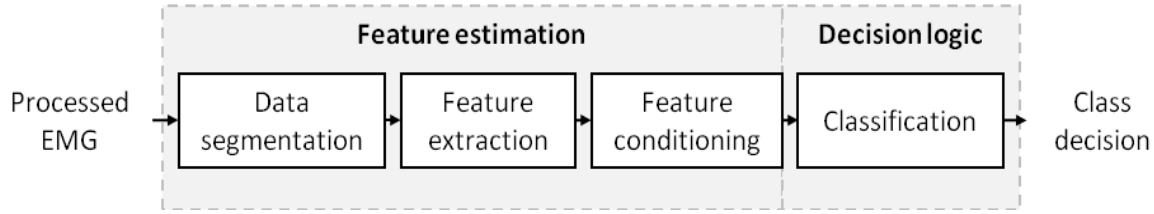

Figure 4-5: Flow diagram of conventional classification methods with feature conditioning**.**

The processed EMG data were segmented and features were extracted from each segment of data. The features were either unconditioned or conditioned (using the KF_{euc} or the S_{cons}), prior to classification.

Several classification methods such as support vector machines (SVM), hidden Markov Models, k-nearest neighbors, and Parzen classifier were evaluated and compared against the common LDA. The feature conditioning methods minimized the effects of varying the classification method and, for a given feature conditioning method, no single classification method significantly outperformed the LDA. One simple classification approach, however, evaluated the Euclidian distance of the patterns from each class mean and associated the pattern to the class with nearest center. This classification method was of particular interest because:

1) When using the conditioned ED features obtained using the KF_{euc}, the ED distance from the class centers does not need to be re-computed and patterns are assigned to the class associated with the smallest ED; and,

2) It only requires the trained class centers to make decisions.

This method is referred to as the 1-nearest neighboring class center (NN_C) and was compared against the commonly used LDA classifier.

4.3.1.1 Data Processing

Several LDA classifiers were trained with the unconditioned ED features or the unconditioned TD features extracted from 4 channels of training data as in Section 3.3.2.

The LDA classifiers trained using ED features were tested against the conditioned ED features (obtained from the KF_{euc}), whereas the LDA classifiers trained using TD features were tested against the conditioned (obtained from the S_{cons}) and unconditioned TD features.

The class decisions of the NN_C classifier were obtained by evaluating the ED distance from the class centers. When using the conditioned or unconditioned features TD features, the ED distance from the class centers were computed as in Appendix A. When using the conditioned ED features obtained using the KF_{euc}, patterns were assigned to the class associated with the smallest feature.

The steady-state classification accuracy (A_{CC}) (as defined in Section 3.3.3.1) was computed on the last second of each contraction. This resulted in a total of 1890 values (7 contractions types x 6 reps x 3 trials x 15 subjects).

The failed completion rate (FC) (as defined in Section 3.3.4.1) was computed for each trial. This resulted in a total of 45 values (3 trials x 15 subjects).

The time to reach N=30 correct consecutive decisions ($\tau_{N=30}$) (as defined in Section 3.3.4.1) was computed for each transition. Failed transitions were given a value of 3000

ms; the maximum time to complete a transition. This resulted in a total of 1890 values (7 contractions types x 6 reps x 3 trials x 15 subjects).

Statistical analyses were performed on these metrics to determine if the NN_C outperformed the LDA. All statistical analyses were conducted using a multi-factor ANOVA test with repeated measurements (random subjects). The other factors were fixed and included all or some of the following: *trial, contraction type, classification method,* and FL. An alpha level of 0.05 was considered significant.

4.3.1.2 Results and Discussion

The mean and standard error of the resulting steady-state A_{CC}, FC rate, and $\tau_{N=30}$ are illustrated in Figure 4-6, Figure 4-7, and Figure 4-8 respectively. A single plot in the figures represents the results obtained using the KF_{euc} (left hand plot), using the S_{cons} (center plot), and using unconditioned features (right hand plot). Because of the high variability in the results between the different contraction types, the average and standard error displayed in these figures were taken across median values computed for every trial. There was 42 contractions performed in a given trial, and a total of 45 trials (15 subjects each performed 3 trials). Therefore, the average and SE illustrated in these figures were taken over 45 values.

It is worth mentioning that varying the classification method only had a significant effect on the configurations without feature conditioning, this suggests that feature conditioning minimizes the effects of varying the classification method. This is due to the fact that the feature conditioning methods force the features to converge towards the class means, grouping patterns closer together and reducing the importance of the classifier boundaries.

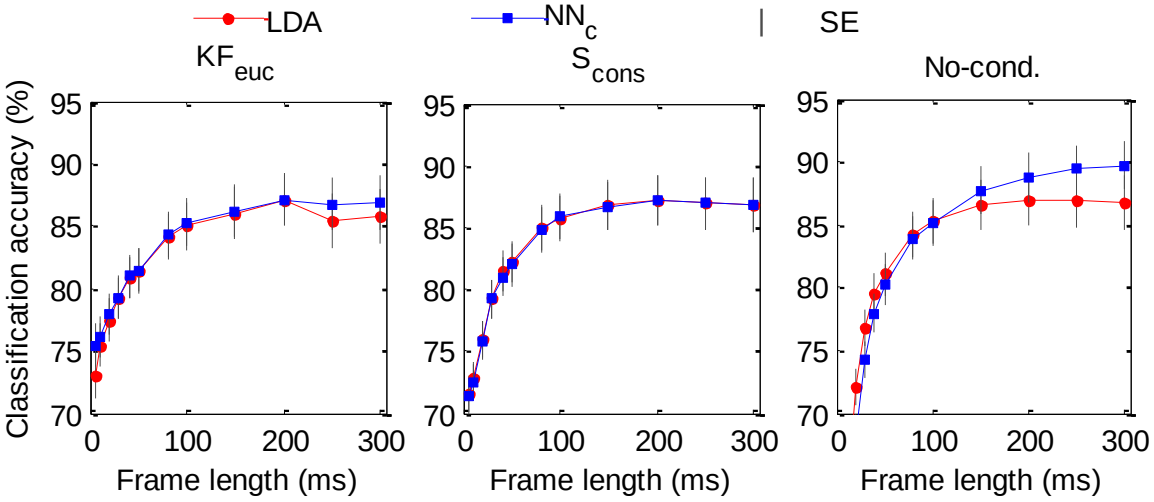

Figure 4-6: Classification accuracy obtained while performing conventional classification with different classifiers and features conditioned using the KF_{euc} (left hand plot) and the S_{cons} (center plot), and the unconditioned features (right hand plot).

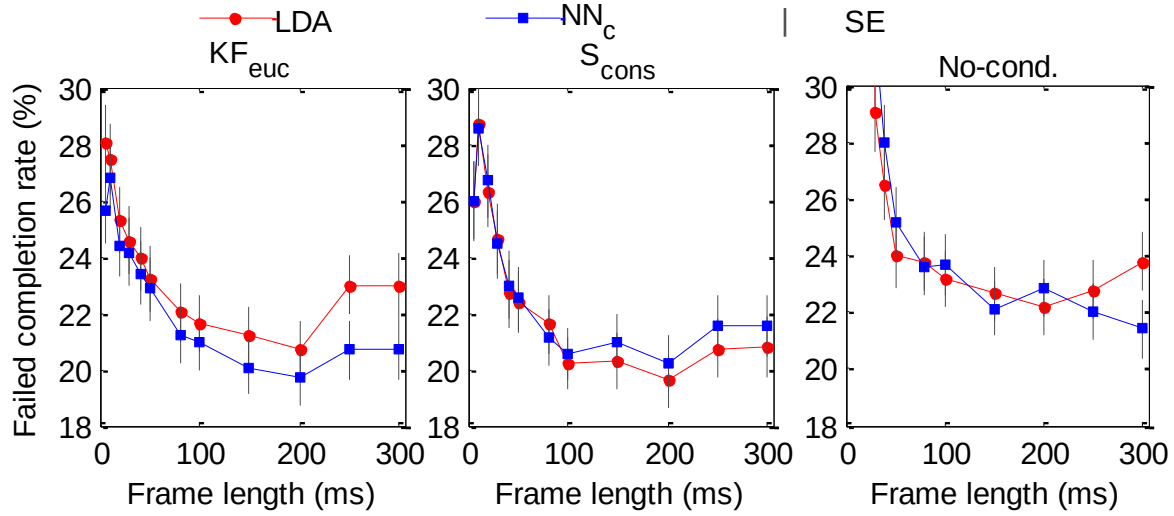

Figure 4-7: Failed completion rate obtained while performing conventional classification with different classifiers and features conditioned using the KF_{euc} (left hand plot) and the S_{cons} (center plot), and the unconditioned features (right hand plot).

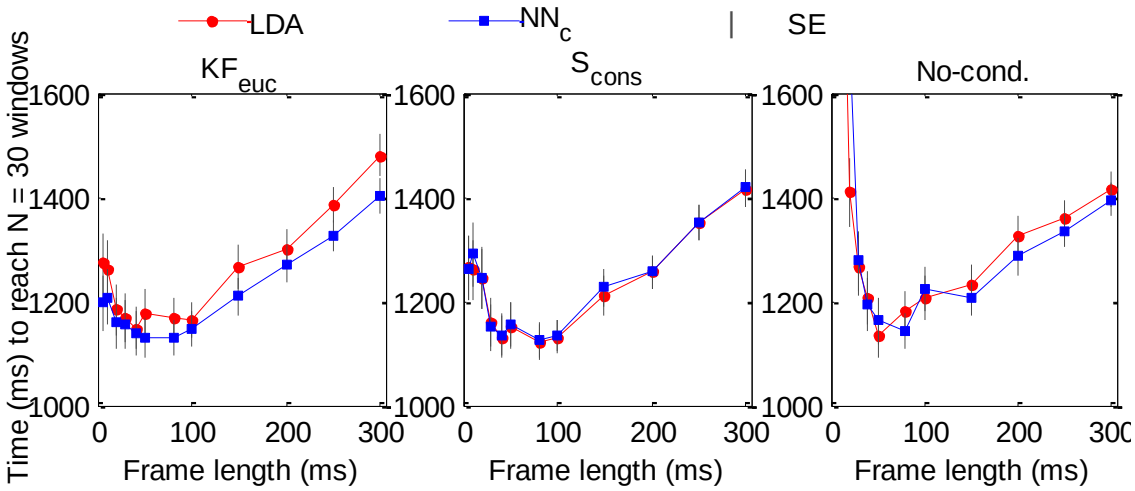

Figure 4-8: Time to reach N = 30 consecutive windows while performing conventional classification with different classifiers and features conditioned using the KF_{euc} (left hand plot) and the S_{cons} (center plot), and the unconditioned features (right hand plot).

4.3.1.3 Summary

The goal of this section was to determine the best classification method for the feature conditioning methods evaluated. Several classification methods were evaluated but only one was worth reporting as it had benefits over the LDA: the NN_C. This method allows rapid classification of the ED features obtained using the KF_{euc}.

The results demonstrated that while using feature conditioning the NN_C classification method could be used without cost in performance. When using the KF_{euc}, the NN_C slightly outperformed the LDA.

For the remainder of this work, configurations with S_{cons} or without feature conditioning will use LDA classification. Configurations with KF_{euc} conditioning will use the NN_C classification method. These configurations are summarized in Table 4-1 and referred to as *conventional classification* methods. They are used as a baseline for comparison against other configurations evaluated in the remainder of this work.

Table 4-1: Summary of the conventional classification methods.

Feature conditioning method	Feature type	Best classifier
KF_{euc}	ED	NNc
S_{cons}	TD	LDA
No-cond. (td)	TD	LDA

For clarity, unless specified otherwise, when referring to configurations using the KF_{euc} it is understood that these perform NN_C classification, and when referring to configurations using the S_{cons} or without feature conditioning it is understood that these perform LDA classification.

4.3.2 State-Based Classification with Rejection

The configuration shown in Figure 4-9 was designed to reject decisions in an indeterminate state:

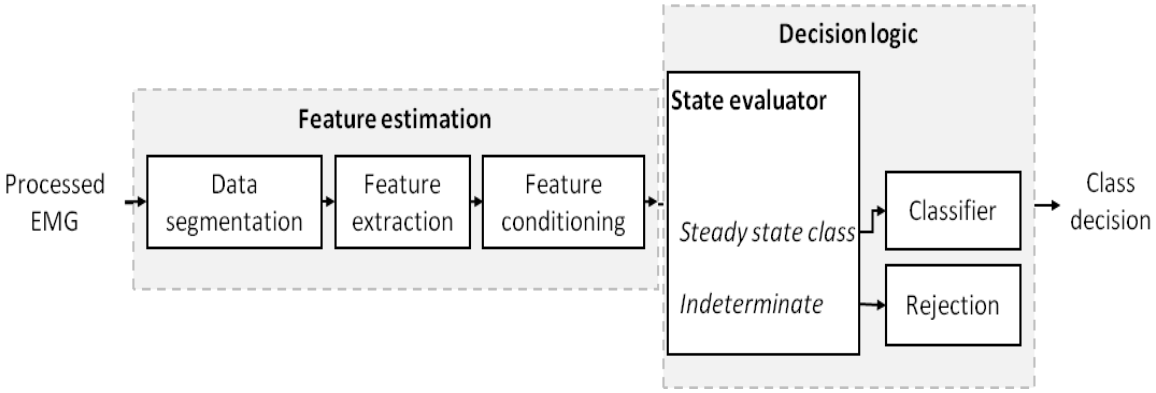

Figure 4-9: Flow diagram state-based classification with rejection.

The processed EMG data were segmented, features were extracted from each segment of data and feature conditioning was performed on the extracted features. The state-evaluator then partitioned the conditioned (or unconditioned) features into steady state class or indeterminate state. While in steady state class, class decisions were obtained

using classification; all indeterminate state class decisions were rejected as suggested by Scheme *et al.* [90].

The effects of rejection are difficult to capture offline. In real-time usage, when the system defaults to rest, the device's velocity is at rest, but the controller stays in the same position; in other words, in the same class. During offline studies, the effects of velocity control cannot be evaluated; however, the effects of staying in the same position can be measured by evaluating the effects of staying in the previously selected class. Therefore, two rejection methods were evaluated:

1) Defaulting to a rest class, and;
2) Repeating the previous class decision.

The configurations evaluated used the feature conditioning and classifier settings listed in Table 4-1. These were compared to a confidence-based rejection method as in [92] that used an LDA classifier and rejected decisions if the class posterior probability was below a threshold (see Appendix B). This rejection method requires class-specific thresholds that can be trained. These thresholds are selected to balance between the active error rate (AER) and total error rate (TER).

4.3.2.1 Data Processing

The state-evaluator was applied to the features conditioned using the KF_{euc} and S_{cons}, and the unconditioned features (as obtained in Section 3.3.2). The state-evaluator threshold coefficient was selected as in Section 4.2.2.

If the state-evaluator determined that the features were in an indeterminate state, the system either defaulted to no-motion or stayed in the previous class. If it determined that the features were in steady state class, the class decisions were obtained by classification

of the conditioned (or unconditioned) features. The different classification methods were trained as described in Section 4.3.1.1.

The confidence-based rejection method was also applied to the decisions obtained using the KF_{euc} or the S_{cons} as well as the unconditioned features. In all cases, features were classified using the LDA classifier. Since the training features were unconditioned, it was not possible to train the confidence threshold when using with the KF_{euc} or the S_{cons} as in [92]. Therefore, the confidence threshold was determined empirically and was set to .90 for simplicity and consistency.

Performance measures were computed on the class decisions as in Section 4.3.1.1. Results were compared to those obtained using the conventional classification method (Figure 4-5) with settings listed in Table 4-1. Results were also compared to the reference configuration (i.e. unconditioned TD features with LDA classification and FL = 150 ms).

Statistical analyses were performed on the metrics to 1) evaluate the effects of the state-based and confidence rejection approaches, and 2) to compare results to those obtained using the reference configuration. Multi-factor ANOVA tests were performed similarly as in Section 4.3.1.1.

4.3.2.2 Results and Discussion

The mean and standard error of the resulting steady-state A_{CC}, FC rate, and $\tau_{N=30}$ are shown in Figure 4-10, Figure 4-11, and Figure 4-12 respectively. In these figures, the results are illustrated for conventional classification methods and state-based rejection methods that, for unknown patterns, stay in the previous class (i.e. state-based$_{prev}$) or default to no-motion (i.e. state-based$_{rest}$). For comparison purposes, results obtained using the confidence rejection methods that default to rest (Conf$_{rest}$) or stay in the previous

($Conf_{prev}$) class are also included. A single plot in the figures represents the results obtained using the KF_{euc} (left hand plot), the S_{cons} (center plot), and the No-cond. (right hand plot) (see Table 4-1). The grey shaded area represents the results obtained while using the reference configuration (i.e. unconditioned TD features with LDA classification and FL = 150 ms); the dotted line is the mean results and the width of the shaded area the standard error. This grey area spans all figures for comparison purposes. The averages reported in these figures were computed similarly to those reported in Figure 4-6 to Figure 4-8.

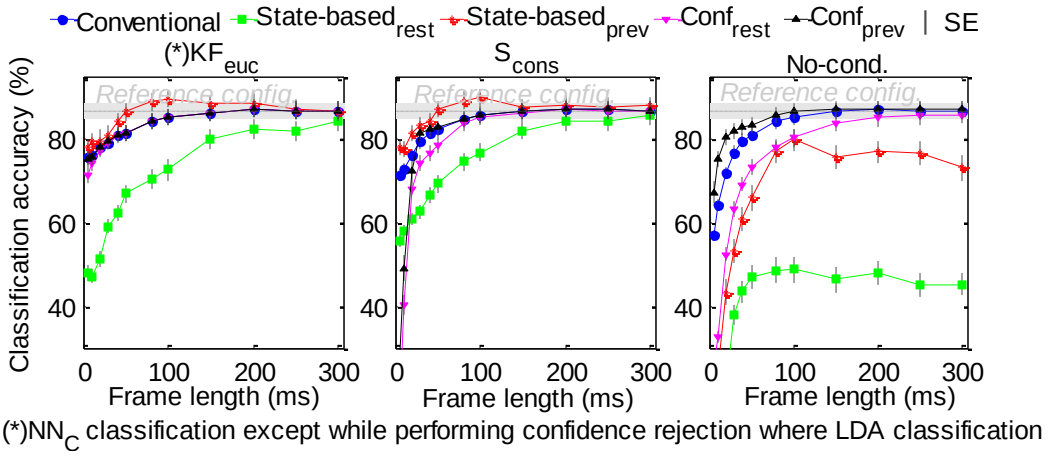

Figure 4-10: Classification accuracy obtained using different state-based methods compared to the various conventional classification methods listed in Table 4-1 and the confidence rejection method classifier found in the literature [92]. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. The confidence rejection method either rejects a decision ($Conf_{rest}$) or stays in the previous ($Conf_{prev}$) class when its confidence is below a threshold. Each plot represents different feature conditioning and classification settings listed in Table 4-1.

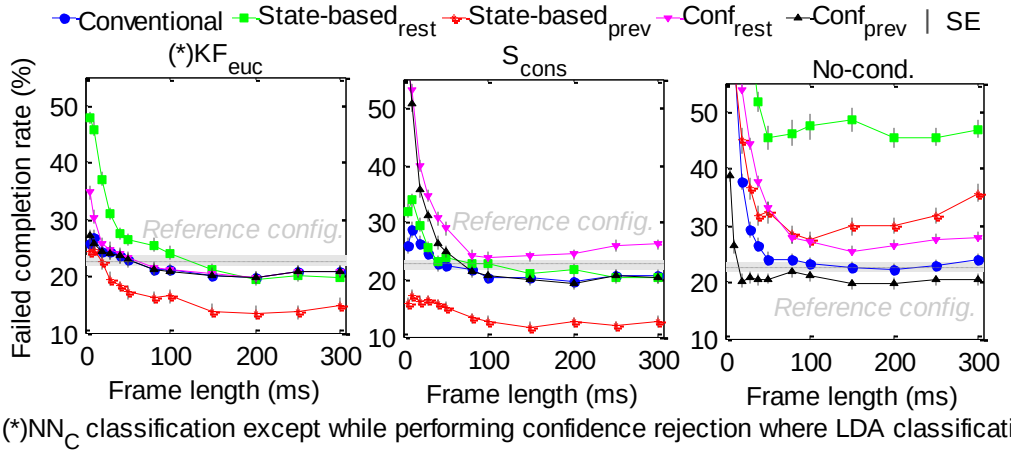

Figure 4-11: Failed completion obtained using different state-based methods compared to the various conventional classification methods listed in Table 4-1 and the confidence rejection method classifier found in the literature [92]. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. The confidence rejection method either rejects a decision (Conf$_{rest}$) or stays in the previous (Conf$_{prev}$) class when its confidence is below a threshold. Each plot represents different feature conditioning and classification settings listed in Table 4-1.

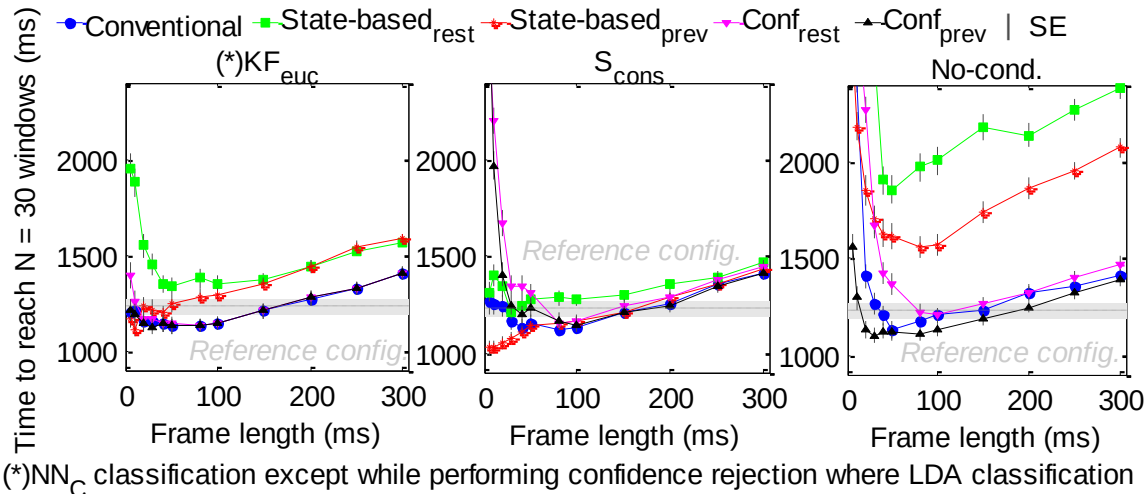

Figure 4-12: Time to reach N = 30 consecutive obtained using different state-based methods compared to the various conventional classification methods listed in Table 4-1 and the confidence rejection method classifier found in the literature [92]. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. The confidence rejection method either rejects a decision (Conf$_{rest}$) or stays in the previous (Conf$_{prev}$) class when its confidence is bellow a threshold. Each plot represents different feature conditioning and classification settings listed in Table 4-1.

(i) Observations of the State-Based Rejection Methods (State-based$_{rest}$, State-based$_{prev}$)

The state-based methods using unconditioned features (right hand plots) obtained significantly ($p < .05$) worse results when compared to the conventional classification method using unconditioned features. This is likely due to an overly aggressive rejection (caused by the state-evaluators inability to detect class transitions) prohibiting correct completion of the different contractions having a direct impact on all metrics evaluated. In fact, as shown in Appendix I, when using the manual labels (obtained in Section 4.2.2) to determine the state of the features, rejection caused degradation in accuracy and $\tau_{N=30}$, but the FC rate was not significantly affected. This suggests that a system capable of detecting and rejecting transitions with 100% accuracy would result in large response delays affecting classification accuracy, but this rejection would not cause inactivity as users would be able to complete the same amount of contractions.

When using feature conditioning, the state-based methods obtained better results. This is expected because the feature conditioning methods are needed to improve the performance of the state-evaluator (see Section 4.2.3).

However, some aggressive rejection is noticeable while using the state-based$_{rest}$ with feature conditioning. In fact, the results show significant ($p < .05$) degradation in classification accuracy for shorter FL. Surprisingly, this did not always results in significant FC rate degradation. In fact, configurations with state-based$_{rest}$ and KF_{euc} obtained significant degradation in A_{CC} for FL $\leq$ 250 ms, whereas significant degradation in FC rate was only noticeable for FL $\leq$ 80 ms. Similarly for the configurations with state-based$_{rest}$ and S_{cons}; the A_{CC} significantly degraded for FL$\leq$ 150 ms but the FC rate significantly degraded only for FL$\leq$ 20 ms. Therefore, the state-based$_{rest}$ for 80 ms < FL

< 250 ms when using the KF_{euc} and for 20 ms < FL < 150 ms when using the S_{cons}, although less accurate than their conventional classification method, was accurate enough to allow users to select 30 consecutive correct decisions for contractions to be marked as completed. This suggests that the state-based$_{rest}$ had a smooth (i.e. consistent), but not always accurate, class decision stream. This is illustrated in Figure 4-13 where output decisions obtained using the conventional classification and state-based$_{rest}$ with KF_{euc} are plotted (for a single contraction):

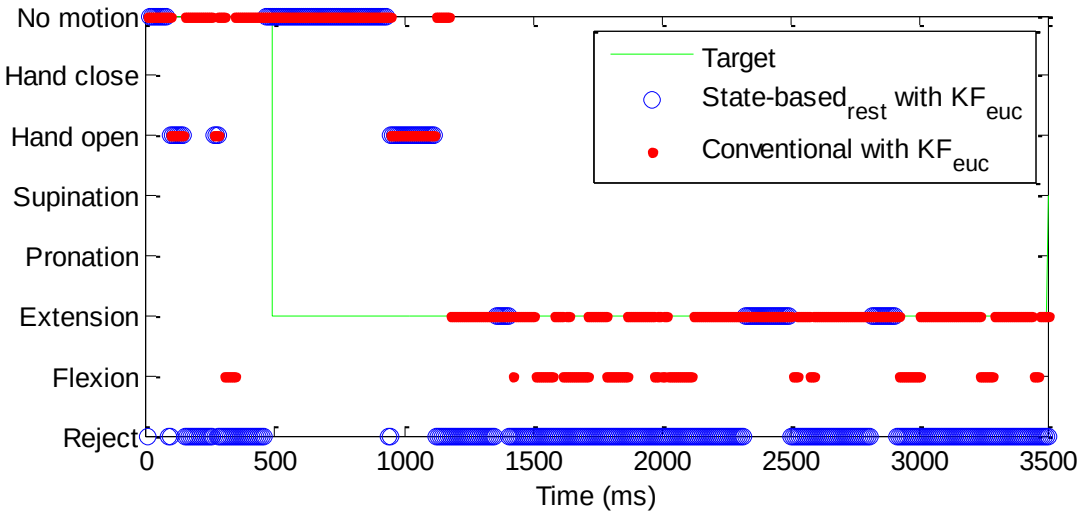

Figure 4-13: Class decisions obtained using the conventional and state-based$_{rest}$ with KF_{euc} conditioning for a single class transition.

It would also be expected that an overly inactive system would inadvertently increase the τ_{30}; this was the case for the configuration with KF_{euc} where significant increase in τ_{30} was observed for all FLs, but it was only the case for 80 ms $\geq$ FL $\geq$ 150 ms when using the configuration with S_{cons}. This can be explained by the fact that, with S_{cons}, the contractions that obtained longer τ_{30} when using conventional classification were often failed while using the state-based$_{rest}$. Since the failed contractions were given a τ_{30} of

3000 ms, this naturally increased the average τ_{30}. This increase is less obvious when contractions that had higher τ_{30} have failed.

The results also show that staying in the previous class when observing an unknown pattern (i.e. state-based$_{prev}$), unlike the state-based$_{rest}$, may facilitate completion of the different contractions when using feature conditioning. In fact, when compared to the conventional classification method, the state-based$_{prev}$ significantly ($p < .05$) improved the FC rate for all FL (with the exception of FL $\leq$ 20 ms when using the KF$_{euc}$). It is anticipated that a significant improvement in FC rate should result in a significant improvement in A$_{CC}$, but this was only the case for FL between 50 ms and 100 ms when using the KF$_{euc}$ and for FL$\leq$80 ms when using the S$_{cons}$. This suggests that while using the state-based$_{prev}$ the class decisions were smoother but not necessarily more accurate than while using the conventional classification method. This is illustrated in Figure 4-14.

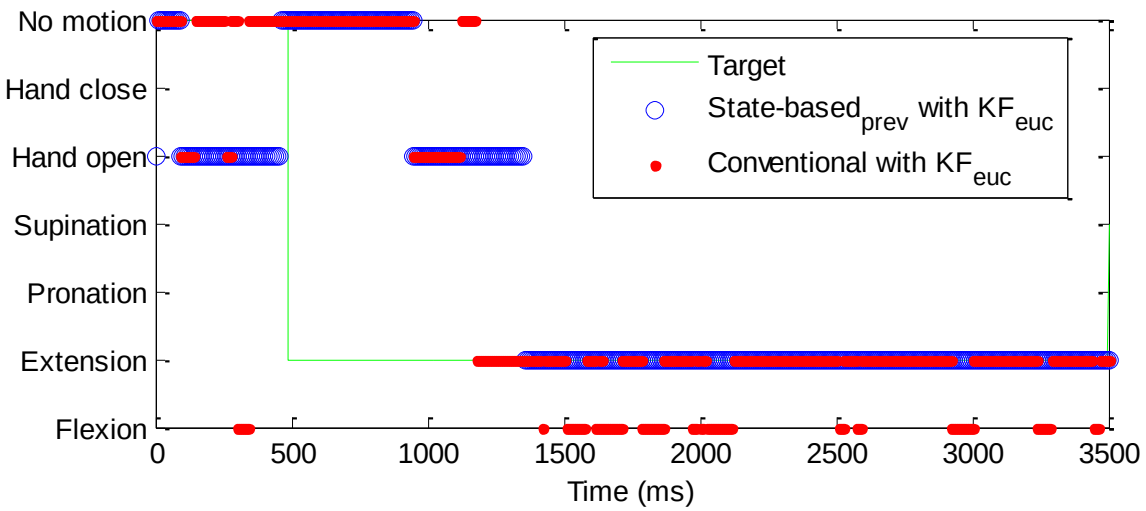

Figure 4-14: Class decisions obtained using the conventional and state-based$_{prev}$ with KF$_{euc}$ conditioning for a single class transition.

Given the improved FC rate (and in some cases the improved A$_{CC}$), it would be expected that the state-based$_{prev}$ obtain better τ_{30} than the conventional classification. But

this is only the case for the state-based$_{prev}$ with S_{cons} and short FL. This suggests that, in some cases (especially while using the KF_{euc}), it took the system some time to reach a smooth and accurate decision. In other words, the state-based$_{prev}$ allowed users to complete more tasks, but these tasks were not completed faster than while using conventional classification methods.

It is worth mentioning that the effects of defaulting to a rest state are difficult to measure offline because real-time speed control cannot be evaluated. The results show that the conventional classification methods allowed users to complete contractions faster than the state-based$_{prev}$, but that the latter (with feature conditioning) allowed users to complete more tasks with (in some cases) higher accuracy. The conventional classification and state-based rejection methods are closely related; they have the same feature conditioning and classification method. Therefore, in real-time, it would be possible to combine the advantages of both by slowing down the device's velocity rather than defaulting to rest while in the indeterminate state. This may, as with the conventional classification method, allow faster completion time, while allowing users to complete more tasks by minimizing the effects of inadvertent class activations as with the state-based$_{prev}$. Approaches that slow down the velocity of the device while decisions are uncertain, such as the velocity ramp [94], have been proposed and have shown promising results [44].

When compared to the reference configuration (grey shaded area in the figures shown above), the state-based$_{prev}$ with feature conditioning obtained significantly ($p<.05$) better FC rate for FL > 40 ms. For such FL, these configurations A_{CC} was comparable to the reference configuration, whereas the τ_{30} was comparable for FL < 150 ms. These results

suggests that a state-based$_{prev}$ method with feature conditioning and FL settings between 40 ms and 150 ms could significantly improve the FC rate for no cost in classification accuracy nor τ_{30} when compared to current state-of-the art systems. Again, it would be expected that an improvement in FC rate would result in an improved A_{CC} suggesting that the state-based$_{prev}$ methods with FL between 40 ms and 150 ms allowed smoother but not more accurate controls than the reference configuration.

(ii) Observations of the Confidence Rejection Methods (Conf$_{rest}$ and Conf$_{prev}$)

While comparing the results obtained using the confidence rejection methods to the conventional classification methods, it is evident that the effects of rejection are most noticeable when applied to unconditioned features and that improvement is obtained only while applying the Conf$_{prev}$ to unconditioned features.

In fact, when compared to the conventional classification method, the Conf$_{rest}$ significantly degraded performance for most FL when applied to unconditioned features, whereas significant degradation only occurred for shorter FL when applied to the conditioned features. The Conf$_{prev}$ on the other hand, significantly improved results when applied to configurations with unconditioned features (for shorter FL), but obtained comparable results when applied to configurations with feature conditioning.

The fact that the confidence rejection method has less of an effect on the configurations with conditioned features suggests that the LDA decisions obtained are more confident than those obtained from the configurations with unconditioned features. This is illustrated in Figure 4-15 where the average (left hand plot) and standard error (right hand plot) of all class decision confidence are illustrated for the KF_{euc}, S_{cons} and unconditioned features with varying FL.

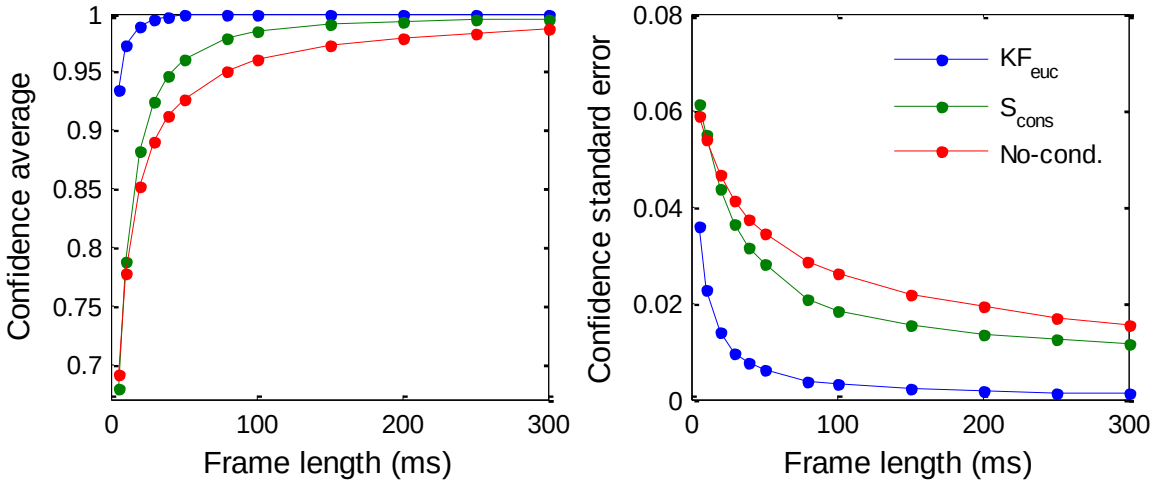

Figure 4-15: Confidence and confidence standard error of the class decision for varying FL and feature conditioning method.

These results can be explained by the fact that the feature conditioning methods force the features to converge towards the class means minimizing the number of decisions for which the features are near the boundaries where the LDA's confidence may degrade. Applying the confidence rejection method on the conditioned features was therefore not further evaluated.

It is worth mentioning that the $Conf_{prev}$ applied to the configuration with unconditioned features (right hand plots of Figure 4-10-Figure 4-12) obtained comparable results to the conventional classification methods with feature conditioning (blue marker in left hand and center plots of Figure 4-10-Figure 4-12). However, the latter do not require thresholds.

4.3.2.3 Summary

As shown in previous work [90] [92], rejection methods that default to a rest state when patterns are unknown may lead to over-rejection unless properly tuned. Feature conditioning did not improve the performance of the confidence rejection method found

in the literature because conditioned features improve the class decisions' confidence. However, feature conditioning applied in conjunction with conventional classification methods obtained comparable results to the confidence rejection method evaluated. These methods, unlike the confidence rejection method, do not require thresholds.

Feature conditioning was required for state-based rejection methods to improve the state estimation and avoid over-rejection. When compared to conventional classification methods, these offline results suggested that state-based rejection could improve the users' ability to complete tasks by allowing smoother class transitions, but at some cost in task completion time. In real-time, however, it is anticipated that this could be remedied by adjusting the velocity of the device, rather than defaulting to rest, while patterns are in an indeterminate state. Regardless, the state-based rejection methods showed a significant improvement in FC rate over current state-of-the art system (the reference configuration) for no cost in A_{CC} or τ_{30} provided that the FL was between 40-150 ms.

The different advantages between using the S_{cons} or the KF_{euc} are not clearly shown in the offline study. Although results suggest that the state-based configuration with KF_{euc} was less responsive to contraction changes than those that use the S_{cons}, the state-evaluator's threshold was easier to compute while using the KF_{euc} (see Section 4.2.3), and the decisions were more confident (Figure 4-15). Both feature conditioning methods were evaluated further.

Table 4-2 summarizes the rejection methods to use with the KF_{euc}, S_{cons}, and No-cond. For clarity, when referring to configurations with rejection, it is understood that rejected decisions result in a repetition of the previous class. Also, when referring to state-based rejection these refer to the first two configurations listed in the table above,

whereas when referring to confidence rejection this refers to the last configuration listed in the table above.

Table 4-2: Summary of the settings for the rejection base methods. The last column indicates the anticipated optimal FL.

Feature conditioning	Feature type	Classifier	Rejection method	Rejection	Anticipated optimal FL
KF_{euc}	ED	NN_C	State-based	PREV	40-150 ms
S_{cons}	TD	LDA	State-based	PREV	40-150 ms
No-cond.	TD	LDA	Confidence	PREV	150-250 ms

4.3.3 Post-Processing

Post-processing of the class decisions obtained using conventional classification (settings listed in Table 4-1) and rejection based methods (settings listed in Table 4-2) was performed using majority voting (MV) (as in [42]). It is worth mentioning that post-processing using the velocity ramp (as in [94]) cannot be evaluated offline since the effects of varying the device's velocity cannot be measured.

In majority vote, the class that occurs most frequently across the last *n* decisions is selected as the class decision [44]. The number of votes (*n*) used in the majority vote is determined by the frame length (FL), frame increment (FI), and the acceptable delay (T_d) [42] [115]:

$$T_d = FL + (n-1)FI \quad (4\text{-}7)$$

It has been generally accepted that the total decision time should not exceed 300 ms to avoid introducing a delay that is perceivable by the user [116].

4.3.3.1 Data Processing

Post-processing was performed on the class decisions obtained in Section 4.3.1.1 while using the conventional classification methods listed in Table 4-1, and class

decisions obtained in Section 4.3.2.1 while using the rejection methods listed in Table 4-2 for the various FL.

The number of votes, *n*, used to perform MV were selected to generate a T_d = 300 ms (Equation (4-7)). For each FL evaluated, the number of votes, *n*, used is listed in Table 4-3:

Table 4-3: Number of votes used for majority vote

Frame length (ms)	**5**	**10**	**20**	**30**	**40**	**50**	**80**	**100**	**150**	**200**	**250**	**300**
Number of votes	60	59	57	55	53	51	45	41	31	21	11	1

Performance measures (steady-state A_{CC}, FC, and $\tau_{N=30}$) were computed on the class decisions as in Section 4.3.1.1. Results were compared to those obtained using the conventional classification methods with settings listed in Table 4-1. Results were also compared to the reference configuration.

Statistical analyses were performed on the metrics to 1) evaluate the effects of post-processing, and 2) compare results to those obtained using the reference configuration. Multi-factor ANOVA tests were performed as in Section 4.3.1.1.

4.3.3.2 Results and Discussion

The mean and standard error of the resulting steady-state A_{CC}, FC rate, and $\tau_{N=30}$ are illustrated in Figure 4-16, Figure 4-17, and Figure 4-18 respectively. Results are presented as in Figure 4-10 to Figure 4-12 but for the conventional classification methods and conventional classification methods with MV ($Conventional_{MV}$), and rejection base methods (Rejection) and rejection base methods with MV ($Rejection_{MV}$). For the KF_{euc} and the S_{cons} the rejection is performed using state-based methods, whereas rejection for the unconditioned features is performed using the confidence rejection method.

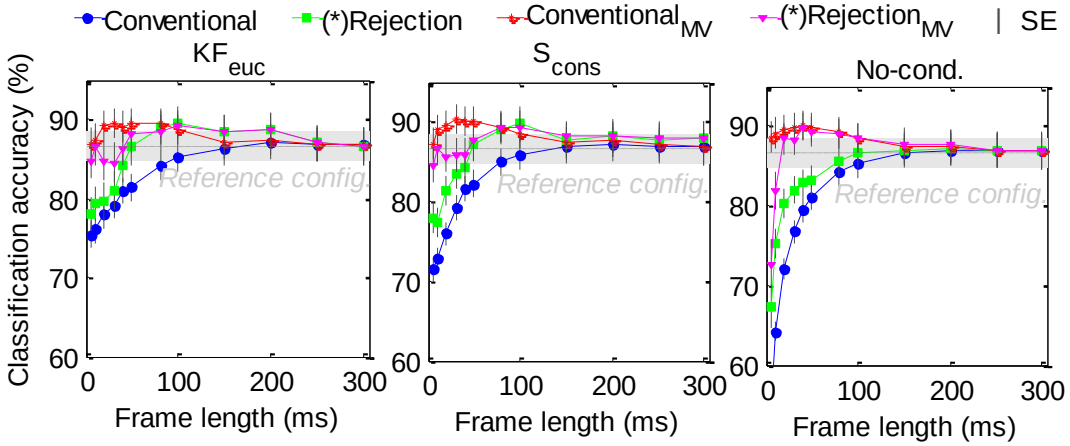

Figure 4-16: Classification accuracy obtained using post-processing applied to the class decisions of the conventional classification methods listed in Table 4-1 and the rejection based methods listed in Table 4-2. Post-processing was applied using majority voting (MV) with 300 ms of voting windows. Each plot represents results obtained using different feature conditioning method.

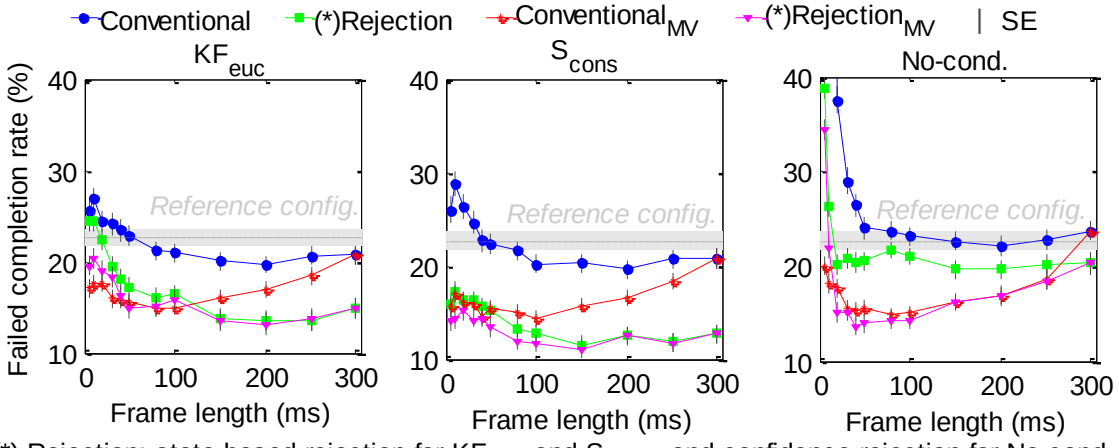

Figure 4-17: Failed completion rate obtained using post-processing applied to the class decisions of the conventional classification methods listed in Table 4-1 and the rejection based methods listed in Table 4-2. Post-processing was applied using majority voting (MV) with 300 ms of voting windows. Each plot represents results obtained using different feature conditioning method.

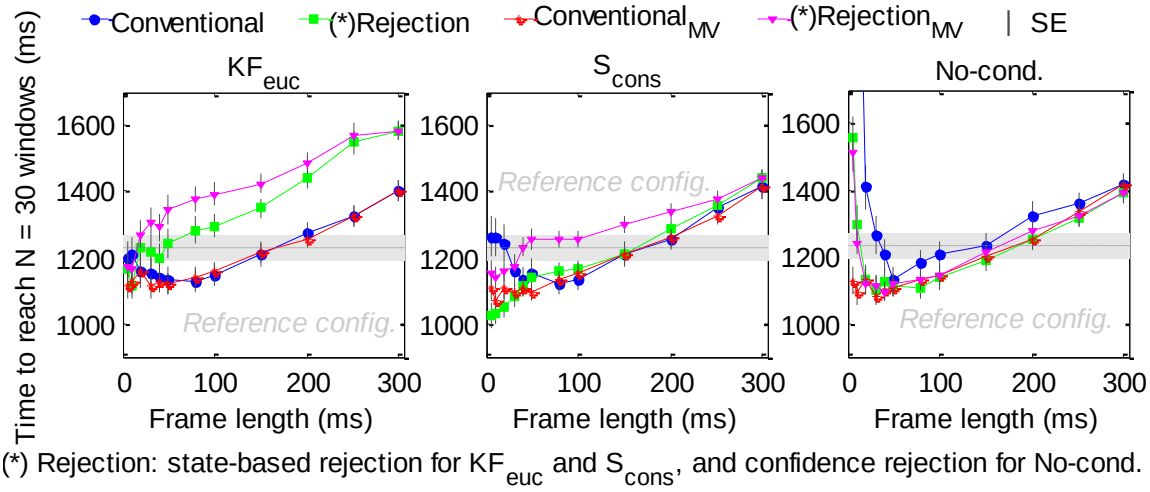

Figure 4-18: Time to reach N = 30 consecutive windows obtained using post-processing applied to the class decisions of the conventional classification methods listed in Table 4-1 and the rejection based methods listed in Table 4-2. Post-processing was applied using majority voting (MV) with 300 ms of voting windows. Each plot represents results obtained using different feature conditioning method.

The above figures show that applying MV to the state-based rejection method did not provide any significant improvement (in some cases it degraded performance although not significantly). Applying MV to the state-based rejection method was not evaluated further.

The results also show that applying MV on the decisions obtained using conventional classification with or without feature conditioning and on the decisions obtained using confidence rejection significantly improve ($p < .05$) the A_{CC} ($FL \leq 80$ ms) and the FC rate (for all $FL \neq 300$ ms), and that these methods all obtained comparable results. The fact that these methods obtained comparable results suggest that feature conditioning and confidence rejection had no effects on MV.

This suggests that the improvement provided by the feature conditioning and confidence rejection is suppressed by the MV methods. This is likely due to the fact that a delay of 300 ms required to make the votes is longer than any smoothing effect caused

by the feature conditioning methods or the confidence rejection method. Applying MV to these configurations was therefore not evaluated further.

When compared to the reference configuration (grey shaded area), it is noticeable that conventional$_{MV}$ with unconditioned features (right hand plot) obtained comparable A_{CC} for all FL, significantly ($p < .05$) improved the FC rate for most FL (10 ms $\geq$ FL $\geq$ 250 ms), and significantly improved the $\tau_{N=30}$ for FL between 30 ms and 40 ms.

These results suggest that adding MV to configurations with unconditioned features significantly ($p < .05$) improves the users' ability to complete contractions without cost in classification accuracy. It was also possible to significantly improve the $\tau_{N=30}$ provided the FL was between 30-40 ms. The $\tau_{N=30}$ improvement obtained for shorter FL can be explained by the fact that shorter FL facilitates the system's ability to capture contraction changes. As seen in Chapter 3, shorter FL improves $\tau_{N=30}$ but degrades classification accuracy. The results obtained in this Section suggest that, although classification accuracy degrades for shorter FLs (see Chapter 3), the most frequently selected class over the last 300 ms of data is likely correct explaining why MV improves the $\tau_{N=30}$. In other words, MV has a smoothing effect on the decision stream, allowing the use of shorter FL.

When comparing conventional$_{MV}$ (without feature conditioning) to the state-based rejection methods (with feature conditioning), it is noticeable that the conventional$_{MV}$ significantly outperformed the state-based rejection methods $\tau_{N=30}$ for FL $\leq$ 150 ms, but the state-based rejection method significantly outperformed the conventional$_{MV}$ FC rate for FL $\geq$ 150 ms. This is due to the fact that as the FL approaches the MV decisions window length (300 ms), the number of votes used for majority voting decreases minimizing the MV effect on the results.

4.3.3.3 Summary

The results obtained in this section showed that feature conditioning and confidence rejection had no effects on post-processing and that applying post-processing to the state-based rejection methods did not provide additional improvement.

Results also showed that for short FL, applying majority voting to common configuration methods found in the literature (i.e. conventional classification method with unconditioned features) outperform the state-based rejection method proposed as majority voting provided improved completion time. However, the later allowed users to complete more tasks when selecting configurations with longer FL.

As a result, the only configuration with post-processing retained for further analysis was the conventional classification method with unconditioned features. It is anticipated that the best FL for such a configuration would range between 30 ms and 40 ms as such short FL allow the system to capture contraction changes. This is summarized in Table 4-4:

Table 4-4: Summary of the settings for the post-processing method. The last column indicates the anticipated optimal FL.

Feature conditioning	Feature type	Classifier	Anticipated optimal FL
No-cond.	TD	LDA	30-40 ms

4.4 Chapter Summary

This chapter evaluated decision logic strategies for the proposed state-based classification method presented in this work (Figure 4-1). The first part of this chapter presented and evaluated a threshold based state-evaluator that partitioned feature patterns as belonging to steady state or an indeterminate state by evaluating the patterns distance from the nearest class center. It was anticipated that this would allow characterization of

the class transitions from the steady-state class. Results obtained suggest that patterns in class transitions were not the only patterns in an indeterminate state; this made it difficult to evaluate the performance of the state-evaluator. Regardless, feature conditioning significantly improved the state-evaluator's performance suggesting that they improved class transitions characterization over unconditioned features. Additionally, feature conditioning with the KF_{euc} allowed the use of a threshold that was not only subject invariant but also close to zero.

In the second part of this chapter, different decision making strategies were evaluated. Conventional classification methods were compared to rejection approaches while considering the impact of feature conditioning and of post-processing. The results showed that:

- State-based rejection methods require feature conditioning. These significantly improved the failed completion rate when compared to conventional classification as they provided smoother class decisions, but in some cases obtained higher $\tau_{N=30}$;
- State-based rejection methods with feature conditioning and FL between 40-150 ms allowed users to perform more tasks (i.e. lower failed completion rate) than while using the current state-of-the art system (the reference configuration) for no cost in A_{CC} or $\tau_{N=30}$;
- Conventional classification methods with feature conditioning obtained comparable results to a confidence rejection method found in the literature (as in [92]) without the need to select thresholds;
- Feature conditioning had no effects on post-processing;

- Post-processing did not improve the performance of state-based rejection methods; and,
- Post-processing applied to conventional classification methods with unconditioned features extracted from FL between 30 ms and 40 ms obtained better offline performance than the reference configuration, and obtained best offline performance amongst all configurations evaluated.

The analyses performed in this and previous chapters were done offline. The literature reports, however, that offline studies are not always indicative of real-time performance [5] [44]. In fact, these offline results show that post-processing applied to the commonly used classification method (TD features with LDA classification) with short FL settings should provide better performance than all other methods evaluated in this work. However, it has been reported that although MV post-processing improves offline performance, it has no significant effect on real-time performance [44]. Additionally, in offline studies, the effects of rejection are difficult to capture because the effects of varying the device's velocity cannot be evaluated. Rejection in real-time disables the device's velocity; it does not change its position (or class).

The results in this work showed that the state-based rejection methods with feature conditioning allowed users to complete more tasks, but such a configuration increased the task completion time when compared to conventional classification methods. In real-time, however, the device's velocity could slow down rather than defaulting to a rest state when features are labeled as in an indeterminate state. It is anticipated that this would, as with the conventional classification method, allow faster completion time, while allowing users to complete more tasks as with the state-based classification method with rejection.

Slowing down the device's velocity while decisions are uncertain have been proposed [44] [94] and showed promising results.

In the next Chapter, real-time analysis will be performed. The results found in this Chapter suggest that the configurations listed in Table 4-5 are most likely to provide better class transition control then current system found in the literature.

Table 4-5: Summary of optimal settings during offline analysis.

Type	**Feat.**	**Feature cond.**	**Classifier**
Conv. classification with MV post-processing	**TD**	n/a	LDA
State-based classification with rejection*	**TD**	S_{cons}	LDA
State-based classification with rejection*	**ED**	KF_{euc}	NN_C

***The system waits in the previous class when patterns are in an indeterminate state**

Chapter 5 – Real-Time Analysis

5.1 Overview

It has been reported that pattern recognition based myoelectric control systems provide good steady-state control but suffer from poor robustness when challenged with more dynamic use [104] [116]. The fundamental hypothesis of this work was that improving behavior during dynamic contractions, such as those occurring during class transitions, results in enhanced usability overall. It was proposed that improved class transition controls could be achieved by 1) using novel feature conditioning techniques in order to improve the responsiveness to class changes and facilitate characterization of indeterminate states, and by 2) rejecting indeterminate state class decisions to prevent inadvertent activations of the limb.

The offline studies presented in this work showed promising results, but did not characterize the dynamic effects of real-time control which more aptly represent the usability of a prosthesis [40]. In this chapter, the real-time control performance of a selection of the proposed configurations was compared to that of the reference configuration (LDA classification of TD features extracted from windows of FL = 150 ms).

This chapter is organized as follows: an overview of the real-time test is provided, followed by a definition of the configurations evaluated. The methodology of the test follows, followed by presentation of results and a discussion.

5.2 Fitts' Law Virtual Target Test for Real-Time Analysis

Fitts' law is a model that describes the human psychomotor behavior [117] [118] [119] for equipment, devices, and processes that interact with the human body. It models

target pointing motions as a function of acquisition time and target acquisition difficulty, as determined by the ratio between the distance traveled to the target and the size of the target. It predicts that as the target acquisition difficulty increases, the time to point the target will also increase [118]. A fundamental property of human motor behavior is the tradeoff between speed and accuracy in target directed tasks [120]. This speed-accuracy tradeoff is classically described by Fitts' law; systems with the best speed-accuracy tradeoff show less motion time increase as the difficulty increases [101].

The real-time test performed in this work is a virtual target acquisition task inspired by Fitts' law as presented by Scheme *et al.* [8]. The authors of [8] demonstrated that virtual target-reaching using velocity based pattern recognition myoelectric control satisfied Fitts' Law. The main graphical interface designed is illustrated in Figure 5-1:

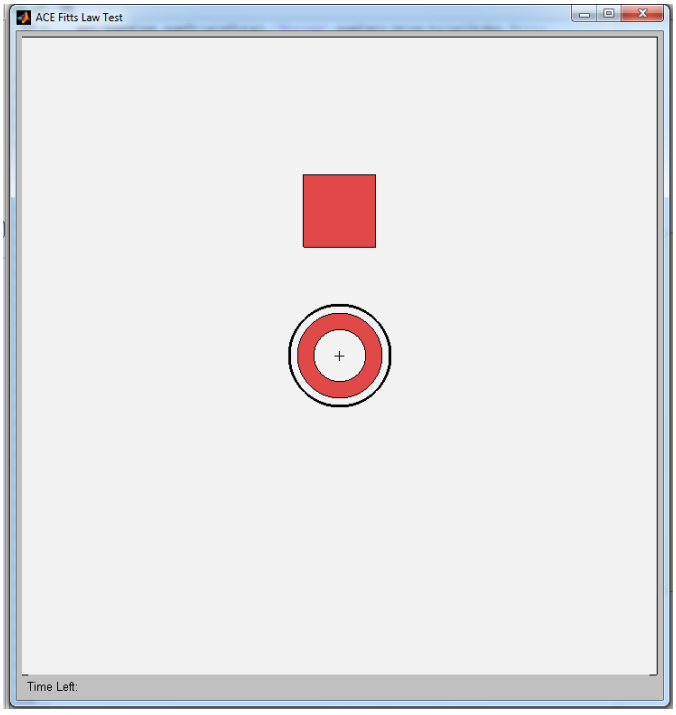

Figure 5-1: Fitts' Law test game interface [8]. Users must superimpose the black circle over the red, and place the crosshairs at the center of the circle within the red box target.

In this figure, the position of the crosshairs at the center of the circles is controlled in 2 Cartesian dimensions by 2 DOF, and the radius of the black circle is controlled by a third (radial) DOF. The target is described by the red box (Cartesian DOF) and red circle (radial DOF). Movement of the cursor is controlled using muscle contractions that are interpreted using the previously described pattern recognition methods. Each class is mapped to an action of the cursor; wrist flexion and extension are mapped to horizontal movements, pronation and supination are mapped to vertical movement, and hand open and close control the diameter of the circular cursor as illustrated in Figure 5-2. The cursor moves along the *x*-horizontal, y-vertical, and r-span diameter planes as follows:

$$\begin{bmatrix} x \\ y \\ r \end{bmatrix}_t = \begin{bmatrix} x \\ y \\ r \end{bmatrix}_{t-1} + \begin{bmatrix} d_x \\ d_y \\ d_r \end{bmatrix}_t * V * \Delta t \qquad (5\text{-}1)$$

where $[x,y,r]_{t-1}$ are the coordinates at time *t-1*, $[d_x\, d_y\, d_r]$ is the direction at which the cursor moves along each axis (directed by the users contraction), V is the motion speed of the cursor which can be constant or controlled proportionally, and Δt is the time between two consecutive samples.

The main objective of the virtual task is for users to reach the 3DOF red target on the screen as quickly as possible, and within a maximum amount of time. This is achieved when the diameter of the circular cursor is within the area of the target circle, and their center (designated by a black cross) is placed within the target box. Figure 5-3 and Figure 5-4 illustrate example tasks where users had to bring the circular cursor within the target circle and the crosshairs within the box target respectively. In both cases (and for all tasks) the initial position of the cursors is at the center of the game interface. To complete the task in Figure 5-3 the user must perform hand close, whereas to complete the task in

Figure 5-4 the user must perform wrist supination. In both cases the targets become green once the user has correctly placed the cursors within the targets. No other motions are required to complete the tasks unless involuntary motions are activated.

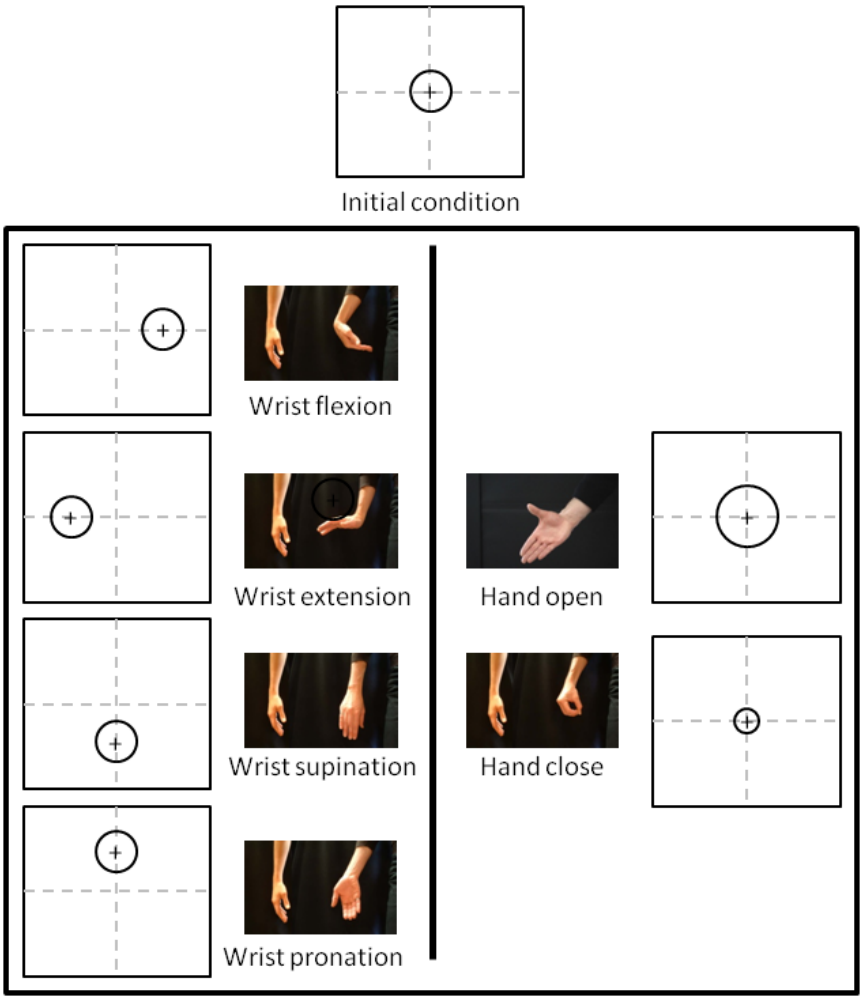

Figure 5-2: Muscle contractions used to control the game interface. Each class is mapped to an action of the cursor; wrist flexion and extension are mapped to horizontal movements, pronation and supination are mapped to vertical movement, and hand open and close control the diameter of the circular cursor

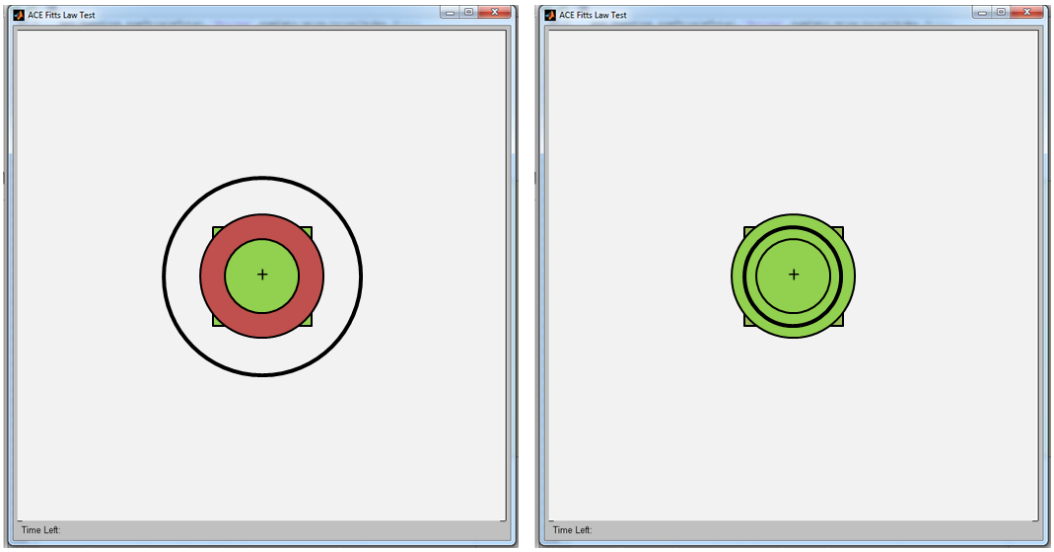

Figure 5-3: Example of a task where the user had to perform a hand close contraction to change the circular cursor radius to complete the task. The initial position of the targets and cursors are at the center of the screen. The circular target is red and its diameter is smaller than that of the circular cursor. Once the user stabilizes the black circular cursor on the circular target, the cursor turns green and the task is complete.

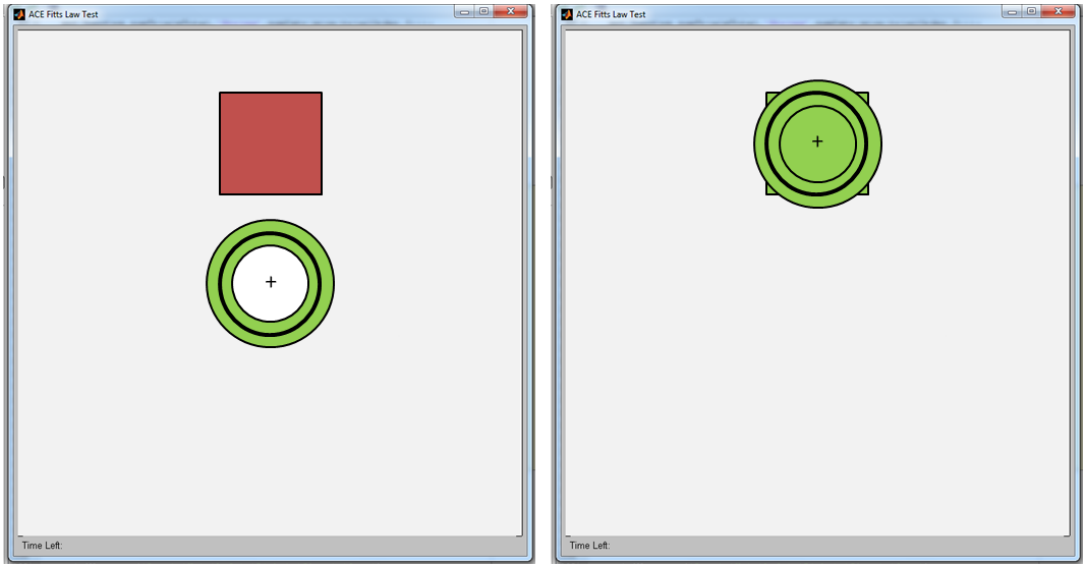

Figure 5-4: Example of a task where the user had to perform wrist supination (vertical deviation) to complete the task. The initial position of the cursor is at the center of the screen, with the circular target already acquired. The red box, representing the Cartesian target, is located on the upper y-axis of the interface. Once the user stabilizes the black crosshairs on the box target, the target changes to green and the task is complete.

Fitts' law states that there exists a linear relationship between motion time and task difficulty [118]; as task difficulty increases so will the time required to complete tasks. The task difficulty can be defined as a function of the target width (W) and distance (D), commonly referred to as the *index of difficulty (ID)* given by the following equation [8]:

$$ID = \log_2\left(\frac{D}{W} + 1\right) \qquad (5\text{-}2)$$

Therefore, increasing the target distance (D) or reducing the target width (W) increases the task difficulty. For the Fitts' law test performed here, the target width and distance are defined as in Figure 5-5 where the target circle's width is given by the difference in radius of its inner and outer edges, and its distance is given by its radius (measured at the midpoint between its inner and outer edges). The width of the Cartesian target is measured as the length of the box's edge, and its distance is measured from the origin to the center of the box. This is illustrated in Figure 5-5:

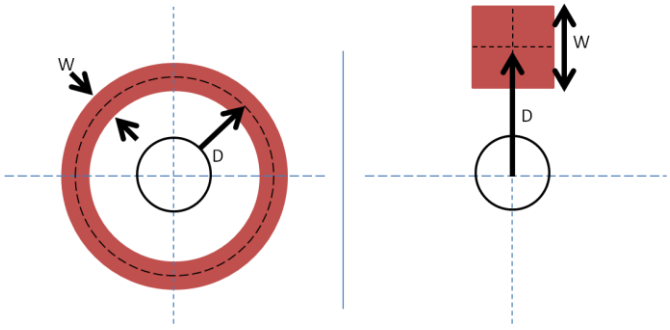

Figure 5-5: Target width (W) and distance (D) measurements for the target circle and target box. The target circle's width is measured as the width of the circles contour, and its distance is measured from the edge of the black circle to the center of the target circles contour. The width of the Cartesian target is measured as the length of the box's edges, and its distance is measured from the origin to the center of the box.

In Fitts' theory, it is anticipated that the motion time should increase with increasing ID. The relationship between ID and the motion time (MT) is described by the following linear equation [118]:

$$MT = b \cdot ID \tag{5-3}$$

where b is the linear slope between the MT and the ID, and is a measure of the speed-accuracy tradeoff. A system with the best speed-accuracy tradeoff would show less motion time increase as the difficulty increases (i.e. smaller slope) [101]. The slope inverse (1/b) is commonly referred to as the *throughput* and is used to compare the performance of different control devices [121].

Throughput is commonly computed as [8] [121]:

$$Throughput = \frac{\overline{ID}}{\overline{MT}} \tag{5-4}$$

where $\overline{\text{ID}}$ and $\overline{\text{MT}}$ are the average ID and MT respectively across all tasks completed.

When a linear regression model is built using experimental data, Fitts' law often appears with an a-intercept value described by the equation [101] [118]:

$$MT = a + b \cdot ID_n \tag{5-5}$$

It has been reported that the linear relationship between MT and ID is extremely robust in human motion, often with a correlation above 0.95 [119], but the inclusion of a non-zero a-intercept cannot be explained theoretically; a task requiring no movement should conceptually take no time. Several interpretations of the non-zero intercepts have been reported [119], suggesting that it is due to modeling errors, user reaction time, or delay and/or feedback processing time in the psychomotor system. Regardless of the cause of the non-zero intercept, it indicates the presence of an additive factor unrelated to the index of difficulty [118].

Zhai *et al.* [121] argued that throughput defined in Equation (5-4) may be unstable when the a-intercept value is non-zero [121] claiming it is no longer intrinsic to the input

system because it incorporates the *a*-intercept (which depends on other added factors that may be unrelated to the ID). They suggest that the b-slope defined in Equation (5-5) is more representative of a system's speed accuracy trade-off.

It is worth mentioning that Fitts' Law was originally proposed for ballistic motions, and not the velocity controlled human-computer interaction described in this work [118]. Consequently, it is important to consider that motion time is not a perfect or comprehensive measure of control. There are many factors that contribute to the quality of control, including but not limited to speed, task completion, intuitiveness, and the path travelled. To account for these considerations, it is important to also measure *path efficiency* [8] and *failed completion rate* (FC rate) [8]. The path efficiency is defined as the ratio of the shortest possible path to the actual path travelled, whereas the FC rate is the percentage of tasks that were not completed within the allowed time. These metrics, along with the throughput (Equation (5-4)) and b-slope (Equation (5-5)), better describe the systems' overall quality of controls.

The performance metrics evaluated in the real-time test are summarized in Table 5-1:

Table 5-1: Performance metrics evaluated.

Metric	Description
Throughput	Describes the systems information capacity; a ratio of average ID to the average MT of all tasks.
b-slope	Describes the speed and accuracy tradeoff of the systems; the linear slope of the MT vs ID slope obtained through linear regression.
Efficiency	Describes the systems quality of control; a ratio of shortest path to the target to the actual path travelled.
Failed completion rate	Describes overall target acquisition success; the percentage of tasks that were not completed within the allowed time. Low values of FC rate indicate better overall success

5.3 Configurations Evaluated

In the previous Chapter, offline analyses were performed to determine which configurations were most likely to provide users with better real-time usability. Three different control settings were retained for further real-time testing. These are repeated in Table 5-2 for clarity:

Table 5-2: Summary of optimal settings during offline analysis.

Type	Feat.	Feature cond.	Classifier	FL
Conv. classification with MV post-process	**TD**	n/a	LDA	30-40 ms
State-based classification with rejection	**TD**	S_{cons}	LDA	40-150 ms
State-based classification with rejection	**ED**	KF_{euc}	NN_C	40-150 ms

The goal of this section was to evaluate real-time performance to determine if, as predicted, the configurations listed in Table 5-2 would improve usability when compared to the reference configuration.

In real-time, feedback is provided to the users; therefore, the user becomes an integral part of the control loop. In effect, the control settings and configurations cannot be altered after the data have been collected. Therefore, prior to conducting a thorough data collection and analysis, a pilot study and some empirical evaluations were performed to validate the configurations listed in Table 5-2 and determine their FL for real-time use. This was required to limit the number of control configurations tested to maintain a reasonable experiment length to prevent participant exhaustion.

It was found, in the pilot study (Appendix J), that MV did not provide significant improvement in real-time performance and actually underperformed compared to the reference system, obtaining reduced throughput (Figure J-8) and increased failed completion rates (Figure J-2) ($p < 0.05$). These results agree with others found in the

literature [96]; MV can improve offline performance but its benefits are negated by the user-in-the-loop delay during real-time use. Since majority voting underperformed the reference system in the pilot study, it was not selected as one of the configurations evaluated in the real-time study presented in this section.

The pilot study also evaluated the effects of slowing down the state-based classification system's velocity while in an indeterminate state (as suggested in Chapter 4 and initially described by Hargrove *et al.* [89]). Such a method is referred to as *state-dependent proportional control* and was compared to the state-based classification system that defaults to rest while in an indeterminate state. The results showed that both the state-based classification with rejection and with state-dependent proportional control significantly improved the FC rate when compared to the reference system. However, the system with state-dependent proportional control significantly improved the time it took users to complete tasks and the overall performance. These results suggest that, adjusting the controller velocity when decisions are not confident may be an alternative to rejecting to rest which can often be overly aggressive [6] [90]. Therefore, in the study presented in this chapter, both the state-based classification method with state-dependent proportional control and rejection listed in Table J-2 were evaluated and compared.

During the pilot study, empirical analyses were performed to determine the best FL to use with the state-based classification methods. Users reported that systems with shorter FL were more responsive but harder to control (Appendix J). These observations agreed with the offline results (Chapter 4); the τ_{30} increased for increasing FL. It is worth mentioning that, contrary to the participants in Appendix J that had some level of experience with EMG pattern recognition controls, most subjects that participated in the

study presented in this chapter had no previous knowledge of EMG pattern recognition controls. Therefore, the FL used for the state-based classification methods was augmented from 40 ms (in the pilot study, see Table J-2) to 100 ms to facilitate controls (100 ms is within the higher range of optimal FL found during offline analyses (see Table 5-2)).

Similarly, empirical analyses were performed to determine the effects of varying the feature conditioning method. While the S_{cons} was reported to be less responsive than the KF_{euc} when performing rejection, some subjects reported it as being more responsive when using state-dependent proportional control. These observations agreed with the offline results (Chapter 4); the S_{cons} allowed better τ_{30} than the KF_{euc}. Therefore, a state-based classification method with state-dependent control, S_{cons}, and short FL was also evaluated in an attempt to improve the systems responsiveness and hence the time it takes users to complete tasks.

From these observations, three control configurations were evaluated and compared to the reference configuration. These are listed in Table 5-3 and are referred to as *Fast*, *Medium*, and *Slow* system in accordance to their expected response speeds.

All of the systems used frame increments of 16 ms (to account for processing time) and employed a simple base proportional velocity control where movement speed was calculated by averaging the mean absolute values of all channels of EMG (as in [96]). The state-based classification methods with state-dependent proportional speed control then reduced this proportional amount by 50% when in an indeterminate state. As shown in Appendix K, the average threshold for the state-evaluator with KF_{euc} and FL = 100 ms (Systems#2 and System#3) ranges between $\sim 10^{-4}$ and $\sim 10^{-5}$, whereas the average

threshold for the state-evaluator with S_{cons} and FL = 50 ms ranges between ~10^{-2} and 10^{-4}. For consistency, the threshold of the state-based systems was set to the orders of 10^{-4} (more specifically: 0.0004).

Table 5-3: Systems evaluated in real-time. The real-time performance of systems with fast, medium, and slow response to contraction changes were evaluated and compared to the reference system's real-time performance.

Referred to as:	System#1 **Fast**	System#2 **Medium**	System#3 **Slow**	System#4 **Reference**
Type	State-based*	State-based*	State-based*	Conventional
Feature cond.	S_{cons}	KF_{euc}	KF_{euc}	n/a
FL	50 ms	100 ms	100 ms	150 ms
FI	16 ms	16 ms	16 ms	16 ms
Classifier	LDA	NN_C	NN_C	LDA
Proportional speed	MAV**	MAV**	MAV**	MAV**
Indeterminate state	State-dependent prop. 50% MAV**	State-dependent prop. 50% MAV**	Reject to rest	

*The state-evaluator threshold was the same for each subject and system (.0004)
** Mean-absolute value (MAV) on all EMG channels.

As described in Appendix K, the discrepancies in the results obtained in Appendix K and in Chapter 4 for the state-evaluator's threshold are likely due to the fact that the offline data used in the analysis of Chapters 3 and 4 were acquired using a different data acquisition system than the system used while acquiring real-time data. The data acquired for the threshold determination in Appendix K was performed using the same system used in the real-time analyses. It was anticipated that the first control configuration would allow the fastest response because it had the shortest FL, employed state-dependent speed control, and performed feature conditioning using the S_{cons}. The third system tested was expected to be the slowest of the evaluated controllers because it defaults to rest while in an indeterminate state. The second system (the *Medium* system) was evaluated to measure the benefits of using state-dependent proportional control.

5.4 Real-Time Assessment

5.4.1 Experimental Methods

Twelve subjects ranging in age from 14 to 53 with no known motor or sensory impairment participated in the experiment. All experiments were approved by the UNB Ethics Review Board under the Surface EMG Data Collection REB #2008-083. The experience level of the subjects with pattern recognition based myoelectric control ranged from novice (11 subjects had no previous knowledge of pattern recognition controls) to expert. Four silver-silver chloride Duotrodes^{TM} (3M, Inc.) were placed at equal distance around the area of largest muscle bulk of the dominant forearm. A silver-silver chloride electrode was placed on the bone of the elbow as a reference electrode. Data acquisition was performed using custom MATLAB®-based software (developed at the Institute of Biomedical Engineering at UNB [108]). The four bipolar pairs of analog EMG were differentially amplified and low pass filtered. Data were sampled with a sampling frequency of 1000Hz using a 16-bit analog-to-digital converter.

Steady-state training data were collected while subjects performed wrist flexion, wrist extension, wrist supination, wrist pronation, chuck grip, and hand open. Each contraction and the no-motion class were repeated twice, for 3 seconds each, in random order. These data were used to train all of the different control configurations (listed in Table 5-3).

For each control configuration, data were recorded while subjects performed 24 tasks with targets of varying width, distance, and location. Two different target widths (W=25 and W=10) and distances (D=50 and D=100) were tested. For a given target width and distance, there were 6 different positions tested, as shown in Figure 5-6:

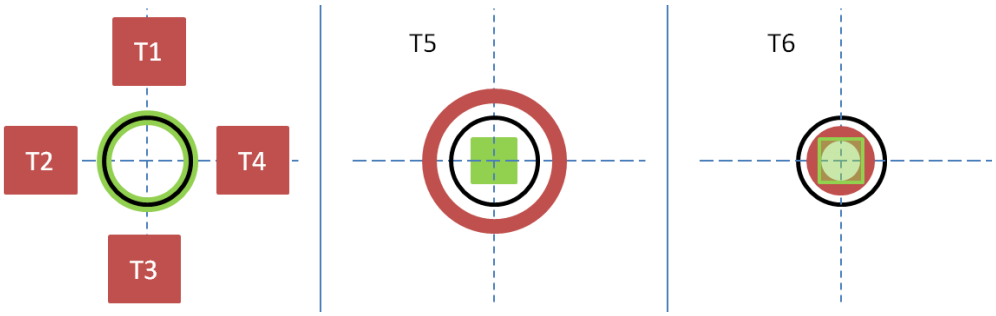

Figure 5-6: The 6 possible strategic tasks. For tasks T1 to T4 the target box was on either axis and for tasks T5 and T6 the target circle diameter was larger or smaller than the cursor's diameter. If no errant motions were activated, users could complete tasks T1 to T5 with wrist motions (flexion, extension, supination, and pronation), and T5 and T6 with hand motions (hand open and close).

Each task began with the cursor at the center of the interface (the origin of the axes illustrated in the figure above). Depending on the task, the target box was located somewhere on the screen along one of the axes or at the origin, and the target circle diameter was smaller, larger or equal to that of the cursor's diameter. At the beginning of each task, a single target required acquisition; therefore, users could complete the task with a single contraction unless an errant motion was activated requiring compensation. Once acquired, a target changed from red to green.

Users were given 15 seconds to complete a task before it would automatically time out and move on to the next one. A task was considered completed after both targets were acquired for one continuous second giving the users up to 14 seconds to reach the targets. A trial was completed once subjects had attempted all 24 tasks using each control configuration (Table 5-3). Training data were collected only once at the beginning of a trial and used to test all control schemes. To minimize anticipation and learning effects, the order of the different configurations and target locations were randomized. Each user was given the chance to become familiarized with the interface and the controls before completing three trials. This familiarization process switched at random between the

different controls methods evaluated. Participants were given the chance to rest between each configuration, and at the end of each trial.

The state-evaluator threshold (see Chapter 4 and Appendix K) for the proposed systems was set to 0.0004 and was validated while the first participant became familiarized with the interface and the different control configurations. The threshold was kept constant throughout the remainder of the data collection and was the same for all subjects as it was observed (in Chapter 4) that the state-evaluator`s threshold while using the KF_{euc} was subject invariant.

5.4.2 Data Processing

The indices of difficulty for the target widths (W = 25 or W = 10) and distances (D = 50 or D = 100) tested in this work are listed in Table 5-4. Target W and D, in this context, are defined as a percentage of the game interface. For example, a width of 25 indicates a target that is ¼ the size of the total length of both the x-axis and the y-axis.

Table 5-4: Combinations of distances (D) and widths (W) indices of difficulty (ID). The ID increases as W decreases or D increases.

W	D	ID
25	50	1.585
25	100	2.322
10	50	2.585
10	100	3.459

The motion time (MT), defined as the time elapsed between the moment the cursor started to move and the moment the users reached the target (i.e. it excludes the 1 second dwell time required to successfully complete a task), was computed for each task. Other metrics computed were the FC rate, efficiency, throughput, and b-slope as listed in Table 5-1.

The FC rate was computed across trials and across the various IDs. The trial FC rate was computed over the 24 tasks of each trial. Since 12 subjects participated in the study, there were a total of 144 values (4 systems x 3 trials x 12 subjects) per metric. The ID FC rate was computed over the 18 tasks that subjects performed for each ID. This resulted in a total of 192 values (4 systems x 4 ID x 12 subjects).

Efficiency was computed for all 3245 tasks (4 systems x 3 trials x 24 tasks x 12 subjects). The motion time (MT) was computed for all tasks that were completed within the 15 seconds cut-off time.

For each trial, the throughput was computed as in Equation (5-4) using the average MT and ID across the 24 tasks, and the b-slope was obtained by performing linear regression using least-square methods on the MT values averaged across the 6 tasks performed per IDs. To ensure that failed tasks were recognized in the throughput and b-slope computation, failed tasks were assigned a value of 15 seconds (the time at which the task timed out). Since 12 subjects participated in this study, there were a total of 144 values (4 systems x 3 trials x 12 subjects) per metric.

Multi-factor ANOVA tests were performed on the FC rate, efficiency, motion time, throughput and b-slope to evaluate the trial or ID effect, subject effect, and the system effect (along with their interactions). For each, subject effect was considered random whereas the other factors were fixed.

For each system, linear regression was performed on the MT averaged across the 216 tasks for each ID (3 trials x 6 positions x 12 subjects). The *a*-intercept and *b*-slope parameters that best fit the data were estimated using least-square methods. The coefficient of determination (R^2) was computed for each system.

5.5 Results and Discussion

The box plots of average FC rate, efficiency, and motion time are illustrated in the left hand plots of Figure 5-8, Figure 5-9, and Figure 5-10 respectively. On each box, the central red line mark is the median, the edges of the box are the 25th and 75th percentiles, the outliers (more than 3 standard deviations away from the mean) are plotted individually, and the whiskers extend to the most extreme values (excluding outliers). The average values are represented by yellow markers. The average motion time was computed for the fast, medium, slow and reference systems (with 666, 675, 683, and 566 completed tasks, respectively). The right hand plot of these figures illustrates the average difference in results when compared to the reference configuration. The average difference in motion time was computed over the 566 tasks completed while using the reference configuration.

The failed completion rate was found to be high for all systems. Figure 5-7 shows the histogram of the time it took users to complete tasks. The distribution illustrated is the average distribution across subjects. This suggests that, on average, the subjects completed most tasks within 10 seconds, whereas fewer tasks were completed within 10 to 14 seconds. The peak at 15 seconds demonstrates the failed tasks. These results suggest that increasing the amount of time given to the users to complete tasks beyond 15 seconds would not have had a noticeable effect on the failed completion rate. These users likely failed the task because they were unable to control the cursors, not because they ran out of time.

These results show that the proposed systems obtained significantly better FC rate ($p < 0.05$) than the reference, with average improvements of 11.57% (fast system), 12.62% (medium system) and 13.54% (slow system); significantly better efficiency by 20.54%

(fast system), 17.31% (medium system), and 28.59% (slow system); and significantly better motion time by 0.42s (fast system), 0.49s (medium system), and 1.15s (slow system). The results also show that, although the proposed systems all obtained comparable FC rate, the slow system obtained significantly ($p < .05$) better efficiency and motion time than all other systems evaluated.

These results suggest that defaulting to a rest state, or slowing down the cursor's speed while in an indeterminate state allowed participants to perform more tasks, however only the system that default to rest allowed users to perform tasks in less time. This could be explained by fewer inadvertent class activations (i.e. better path efficiency) suggesting that users performed fewer corrective motions to over-compensate for the false activations allowing them to complete tasks faster than while using the reference system.

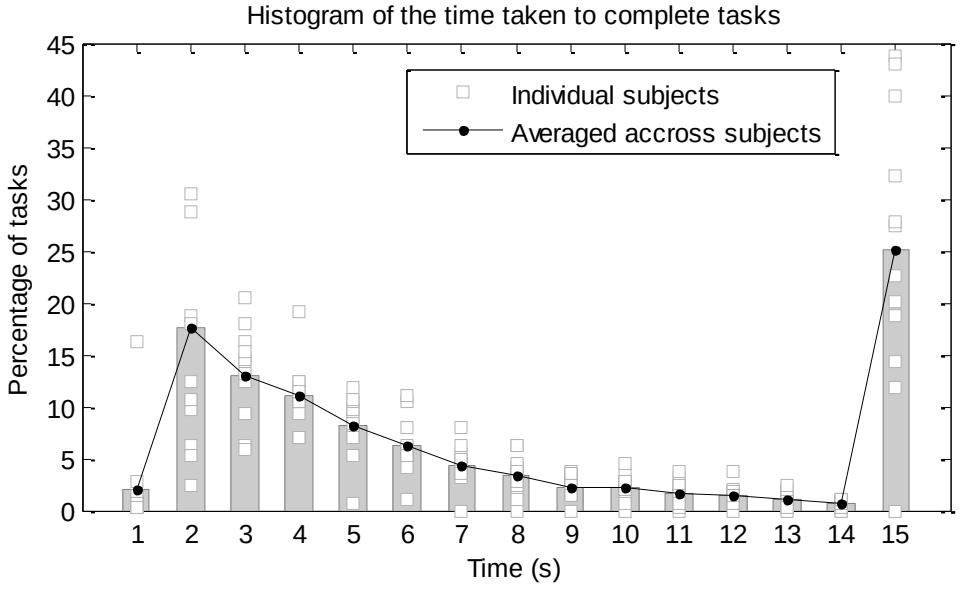

Figure 5-7: Histogram of the time taken to complete tasks. The circular markers represent the average percentage of tasks completed within a time bin across subjects.

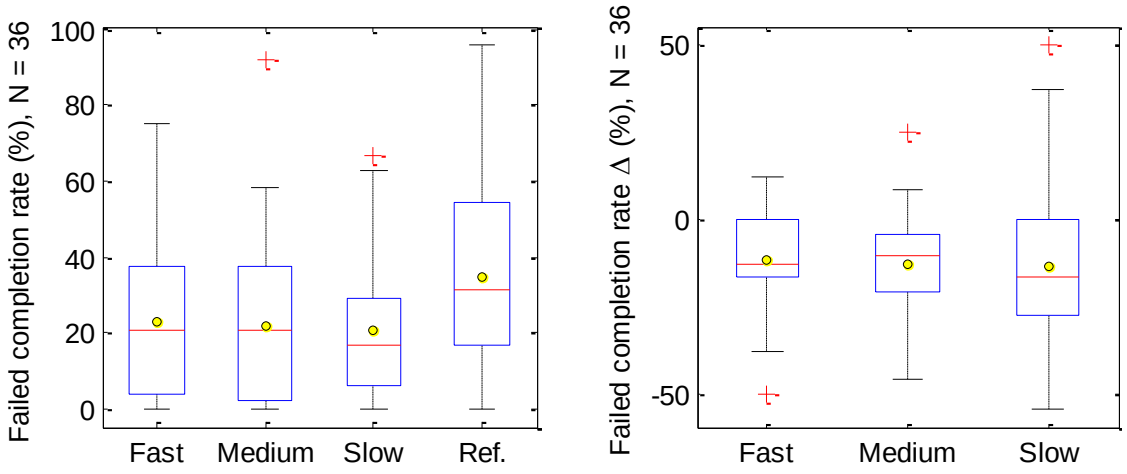

Figure 5-8: Box plot of the failed completion rate overall trial (left hand plot) and difference in failed completion rate when compared to the reference configuration (right hand plot). Averages represented by the yellow markers were computed over N = 36 trials.

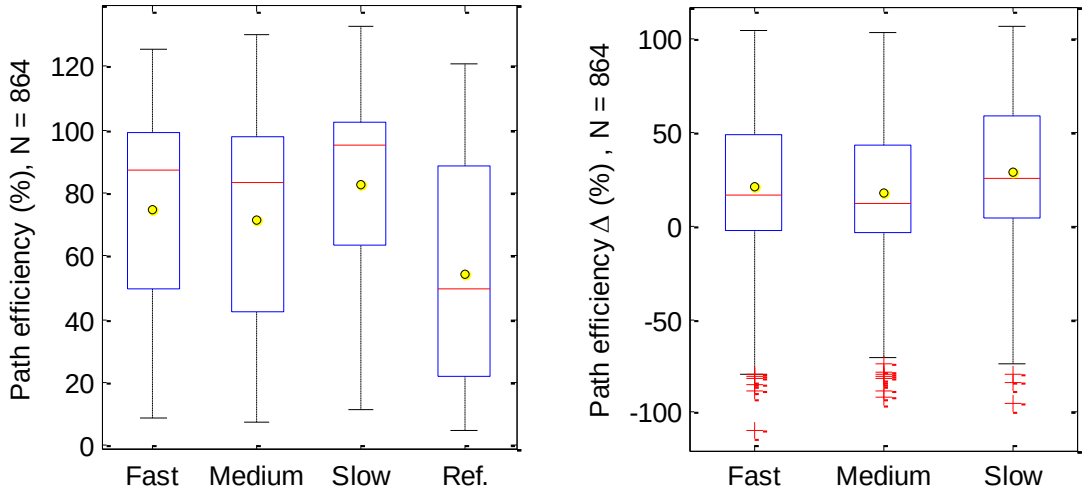

Figure 5-9: Average path efficiency overall tasks (left hand plot) and difference in path efficiency when compared to the reference configuration (right hand plot). Averages represented by the yellow markers were computed over N = 864 trials.

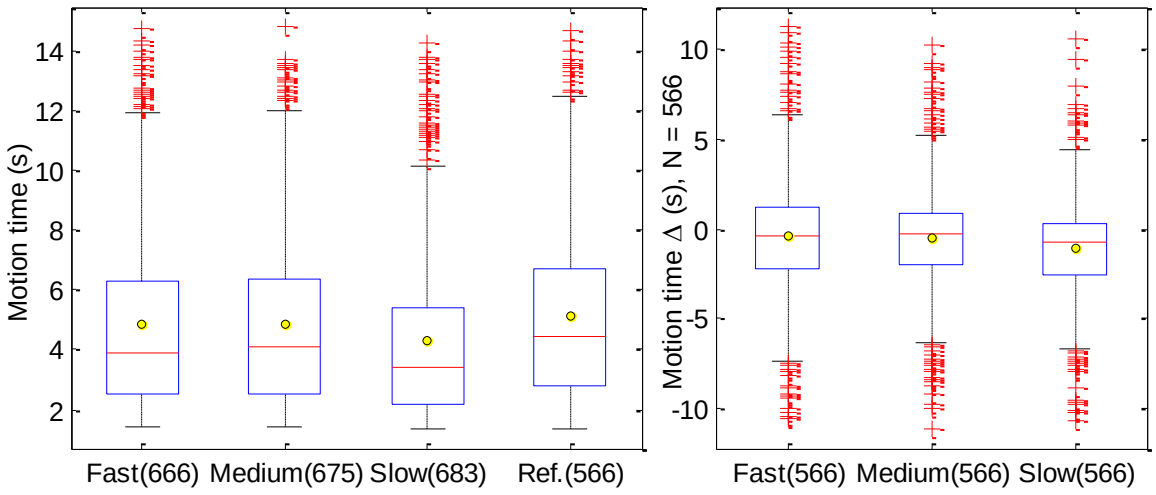

Figure 5-10: Average motion time overall tasks (left hand plot) and difference in motion when compared to the reference configuration (right hand plot). The average motion time was computed over N = 666, N = 675, and N = 566 completed tasks for the fast, medium, and slow system respectively. The average in the motion time difference with respect to the reference configuration was computed over the N=566 completed tasks while using the reference system.

They also suggest that defaulting to rest while in an indeterminate state allowed users to perform tasks faster than reducing the cursor speed, but did not allow users to perform a significantly larger amount of tasks when compared to the other proposed configurations. This is, again, likely due to fewer corrective motions (i.e. significantly better path efficiency).

However, statistical analysis of the FC rate, efficiency, and motion time revealed that both the trial and task ID had a significant effect on results.

The average FC rate, efficiency and motion time per trial are shown in Figure 5-11. From these plots it can be seen that, for all systems, the FC rate significantly ($p < .05$) reduced from trial#1 to trial#2 (there was no significant difference from trial#2 to trial#3) and the efficiency significantly improved from trial-to-trial (from trial#1, to trial#2, to trial#3), indicating a learning effect.

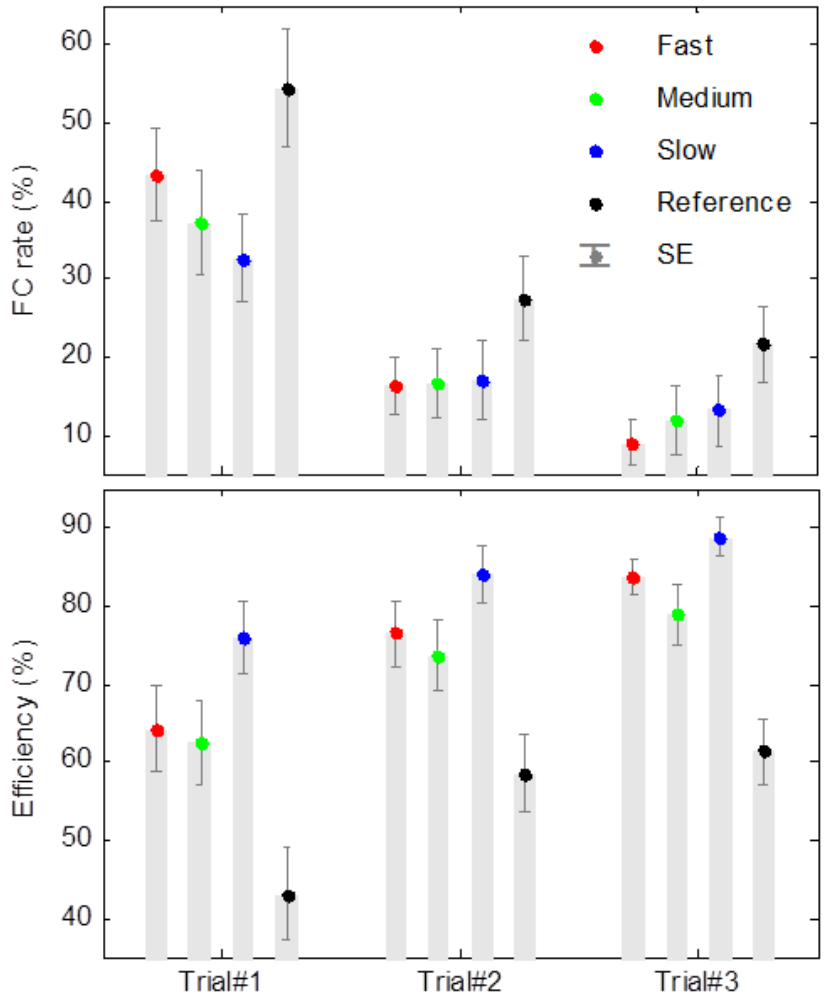

Figure 5-11: Averaged failed completion rate (FC rate, top plot), and efficiency (bottom plot) for the first, second and third trials obtained using each system. Values were averaged across subjects for each system and trial. The vertical bars represent the standard error (SE).

The significant improvement in efficiency from trial-to-trial indicates that, as users became more experienced, they became more capable of controlling the cursor with fewer inadvertent class activations and fewer failed tasks.

The results also show that the observations made in Figure 5-8 to Figure 5-10 are consistent across all trials. The proposed systems yielded significantly lower FC rates and higher efficiency than the reference system, and that the slow system obtained

significantly better efficiency than all systems. The average FC rate and efficiency obtained for each ID and system and the standard error (SE) are shown in Figure 5-12.

These plots show that, for all systems, the FC rate was significantly ($p < .05$) higher for smaller targets (ID3and ID4) than larger targets (ID1 and ID2). When compared to the reference system, the proposed systems significantly reduced the FC rate for ID3 and ID4 ($p<0.05$) but they obtained comparable results for ID1 and ID2 (with the exception of the medium system, which significantly outperformed the reference system).

These results suggest that tasks with smaller targets (i.e. ID3 and ID4) were significantly more difficult to perform. The difficulty completing tasks with smaller targets may have been due to less lower path efficiency as tasks with smaller targets (ID3 and ID4) obtained significantly worst worse efficiency than tasks with larger targets (ID1 and ID2). The proposed systems, however, enabled users to reach and stay within smaller targets more frequently. This is due to the fact that the proposed systems attenuate the output speed while in indeterminate states (i.e. either default to rest or operate at half the speed) allowing the users to slow down the cursor and stabilize it on the small targets with fewer inadvertent class activations. This agrees with Fitts' law which assumes that users must slow down to select smaller targets which require greater accuracy [120].

It is worth mentioning that the slow system was the only system to produce significantly different results between tasks ID1 and ID2. Tasks ID1 and ID2 differed only in distance, as they shared the same large target width ($W = 25$). This suggests that the slow system prevented users from selecting larger targets when they were further. Since the slow system stopped the cursor's motion while in the indeterminate state, the results suggest that over-rejection may have prevented users from completing tasks that

were at further distances (Figure 5-12). This is likely caused by the fact that users were trying to move faster towards the further targets elicitation inadvertent contractions and therefore rejection. Rejection has a direct impact on path efficiency; for example, over-rejection only allows very confident decisions to pass resulting in very few inadvertent class activations and hence a high efficiency. However, if over-rejection occurs, this can also lead to higher movement times and FC rate.

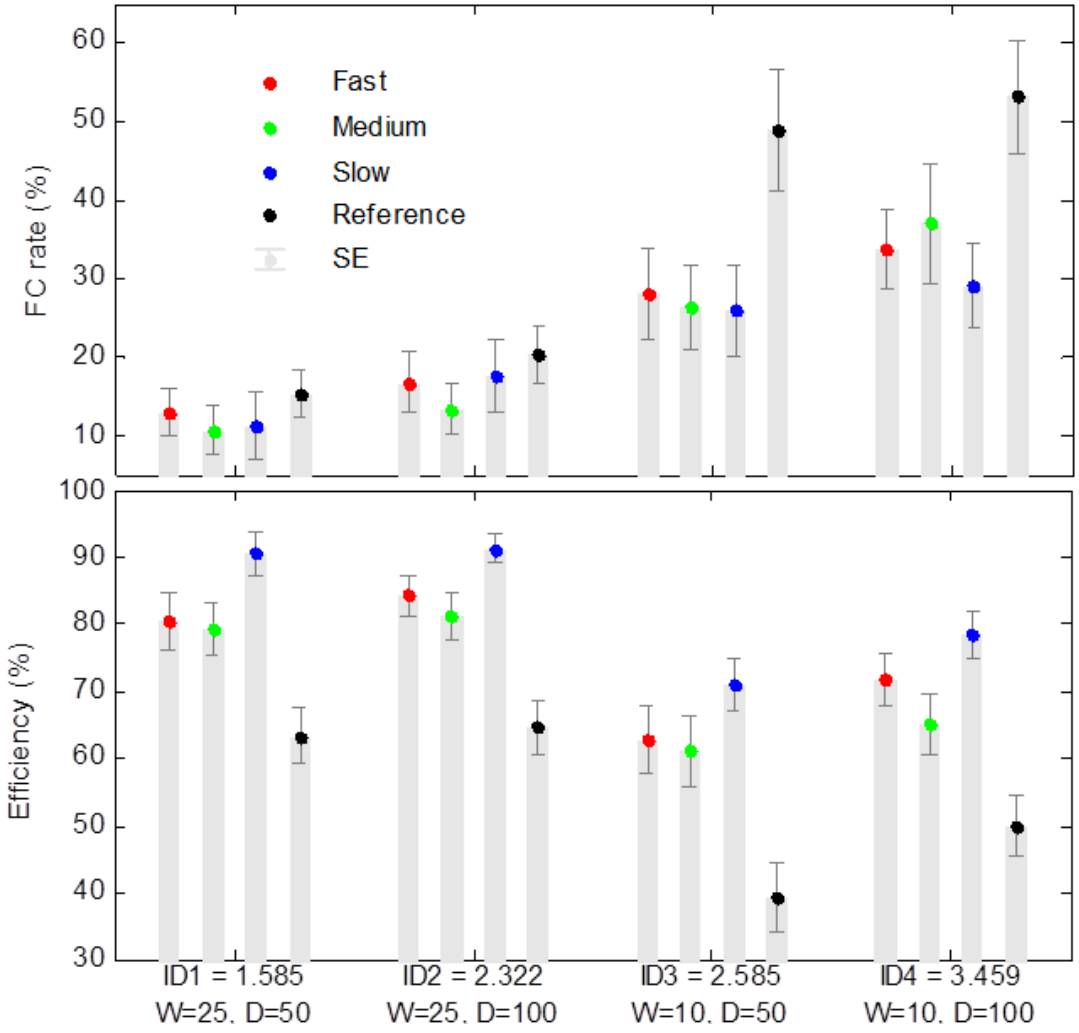

Figure 5-12: Averaged failed completion rate (FC rate, top plot) and efficiency (bottom plot) for each ID obtained using the different system. These values were computed for the 18 tasks of each ID. These values are average across subjects for each system and ID. The vertical bars represent the standard error (SE).

These results correspond to what the users reported anecdotally; most subjects preferred the slow system when selecting small targets, but when the target distance became longer, they complained about the system being too slow or unresponsive to what they were trying to do and preferred the fast or medium systems.

The increased FC rate observed as the task difficulty increased is clearly undesirable; it confounds the use of Fitts' Law which does not expect varying degrees of incomplete trials based on ID. In a conventional Fitts' Law test, the only effect that an increase in index of difficulty should have on the results is an increase in the time it takes to complete the task. To partially compensate for this, failed tasks were assigned a value of 15 seconds (the length of the timeout). Given this assumption, the average motion time for each ID along with the Fitts' law linear regression, throughput, and the coefficient of determination (R^2) are illustrated in Figure 5-13.

The proposed systems obtained higher coefficient of determination R^2, lower b-slope, and higher throughput than the reference system. Also, it is evident from Figure 5-13 that the vertical-intercept of each system is non-zero.

The fast, medium, slow, and reference systems closely followed Fitts' law with R^2 = .97, .99, .96, and .87 respectively. The higher R^2 obtained using the proposed systems suggest that these may have provided a more natural form of control than the reference system as they follow the expected linear relationship between motion time and task difficulty more closely.

The average b-slope and throughput obtained for each trial and system along with the standard error across subjects are shown in Figure 5-14. The throughput increased significantly ($p<0.05$) from trial-to-trial for all systems, supporting the earlier identified

learning effect. The reference system obtained significantly ($p < .05$) lower throughput than the other systems for all trials. The b-slope of the reference system significantly increased from trial-to-trial whereas the b-slope of the fast and slow systems became significantly smaller. The b-slope of the proposed system was significantly better than the reference system for trial#2 and #3.

It is unclear why the throughput and b-slope have different trends from trial-to-trial; however, the literature [121] states that the throughput is unstable when a system's linear regression has a non-zero a-intercept (Figure 5-13). The throughput results suggest a learning effect for all systems, whereas the b-slope suggests a learning effect for the slow and fast systems only.

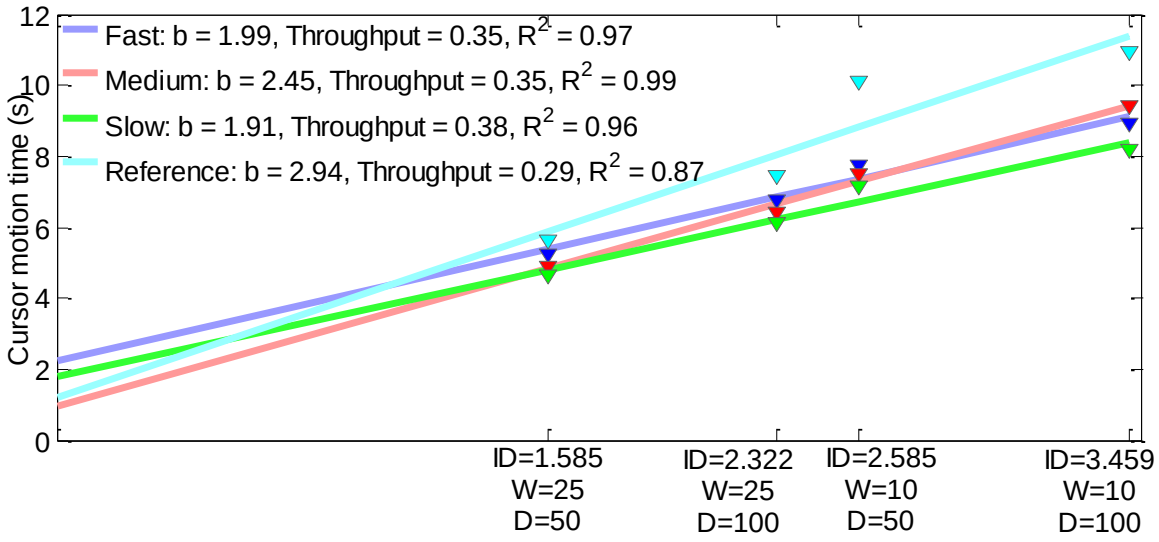

Figure 5-13: Averaged motion time (MT) for each ID, and Fitts' Law linear regression b-slope and R^2 coefficient of determination, as well as the throughput for each system. For each system, the MT is averaged across the tasks with ID = 1.585, the tasks with ID = 2.322, the tasks with ID = 2.585, and the tasks with ID = 3.459. The *a*-intercept and *b*-slope parameters that best fit the data are estimated using least-square methods. The throughput is computed as the ratio of the average ID and the averaged MT.

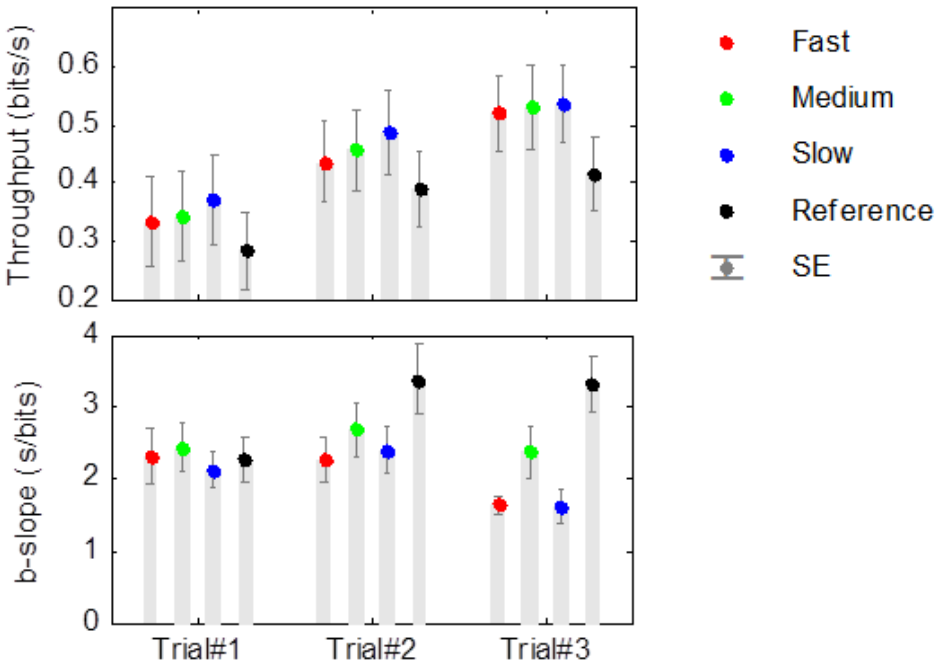

Figure 5-14: Averaged throughput (left plot) and b-slope (right plot) for the first, second and third trial obtained using each system. For each test, the MT is averaged across the 6 tasks with ID = 1.585, the 6 tasks with ID = 2.322, the 6 tasks with ID = 2.585, and the 6 tasks with ID = 3.459. The *b*-slope parameter that best fit the data are estimated using least-square methods. For each test, the throughput is computed as the ratio of the average ID and MT. For each trial and system, values are averaged across subjects. Vertical bars represent the standard error (SE).

Although training data were collected at the beginning of each trial, some physiological factors such as muscle fatigue or sweat on the surface of the skin can affect the user's ability to control the system even after retraining [80]. The degradation in the reference system's b-slope suggests that the proposed systems were more resilient to changes that occur over time. This is due to the fact that feature conditioning prevents features from navigating too far away from the trained class means.

Regardless, the results suggest that the proposed systems obtained better overall performance and a better speed-accuracy tradeoff than the reference system.

5.6 Chapter Summary

The real-time performance of various state-based control methods was compared to the real-time performance of a reference system based on TD feature and LDA classification [82] in order to evaluate the usability of the proposed methods. This was done by conducting a Fitts' Law target acquisition test similar to that proposed by Scheme *et al*. [8].

The results showed that the proposed systems allowed users to complete a significantly larger number of tasks than the reference system. As predicted, the proposed systems allowed users to complete tasks in less time overall. Somewhat surprisingly, the slow responding system obtained the best motion time. This appears to be due to the higher path efficiency, resulting in fewer inadvertent class activations that required compensation. This agrees with the literature that states that extraneous motions require users to elicit additional corrective motions, increasing motion times, and user frustration [90]. However, this does not agree with the results in the pilot study of Appendix J which showed that the state-dependent proportional control system obtained better motion time and speed-accuracy tradeoff than the system that rejected indeterminate decisions. The discrepancies in the results may be due to two factors:

1) The subjects that participated in the study of Appendix J had more experience in EMG pattern recognition controls allowing them to use more complex control methods, and;
2) The tasks performed during the experiment of Appendix J were more complex. This may result in a larger amount of unintentional activations

which would result in a less responsive state-based classification method that defaults to no-motion while in indeterminate state.

Further investigation revealed that the success rate significantly degraded when performing tasks with smaller targets. The proposed systems, however, allowed users to complete tasks with smaller targets with better overall success than the reference system. This suggests that more conservative systems (which defaulted to rest during indeterminate states) allowed users to perform tasks that required greater accuracy (i.e. small targets) with better success. This agrees with Fitts' law which assumes that, as the target width becomes smaller, users must slow down in order to be able to accurately select the target [18].

The target distance had a noticeable impact on the users' ability to complete tasks with larger targets while using the slow system as those with larger distances had lower success rate. This is likely caused by users trying to reach the target faster resulting in a high level of inadvertent classes and therefore over-rejection. Systems that reduce the cursor's speed while in an indeterminate state may be an alternative to rejection methods as they allowed users to complete a comparable amount of tasks without the added risk of over-rejection. Although state-dependent proportional control did not allow users to perform tasks as rapidly as the rejection method, it obtained significantly better motion time, throughput and b-slope than the reference system.

In other words, the results obtained while users performed tasks with targets of varying width and distances suggest that a system that performs at higher speeds when trying to reach a target but at slower speeds when trying to select the target may allow users to complete more tasks. This is in accordance with Woodworth's [120] findings. He

suggested that aiming movements are composed of two phases where the first is an impulse phase that would bring the limb into the vicinity of the target, and the second is a homing phase which uses visual feedback to make any adjustment to the movement trajectory necessary to bring the limb to rest on the target. During the first phase, people tend to put more importance in the speed at which they conduct the motion, while in the second phase they put more importance in the accuracy of the movements. This also agrees with the proportional control method used at the Rehabilitation Institute in Chicago described in [97].

The increases observed in FC rate as the task difficulty increased were undesirable. To account for failed tasks when evaluating the speed-accuracy tradeoff (i.e. b-slope) and the overall performance (throughput), failed tasks were assigned a motion time equal to the total amount of time users were given to perform a task. After such adjustments, results showed that as users gained experienced, the proposed systems resulted in better speed-accuracy tradeoff and overall performance than the reference system. An interesting result obtained from the Fitts' law analysis is that, as users became more experienced, although they were capable of completing more tasks, the speed-accuracy tradeoff (i.e. the b-slope) of the reference system degraded. This may be attributed to the fact that small changes in surface EMG characteristics over time degraded the performance of the reference system, but not the proposed systems [80]. This suggests that the proposed system may be more resilient to changes over time.

The results also revealed a learning effect. Most participants were un-experienced EMG users, but as they became more experienced, the results obtained using the different control methods started to converge. This suggests that with time, users may be able to

perform smooth class transitions without the need for assistance by a system that rejects motions that may be unintentional.

Chapter 6 – Conclusion

6.1 Introduction

The purpose of this work was to improve the performance of pattern recognition based myoelectric control systems during dynamic use. It was motivated by the fact that, although such systems have been shown to be accurate in controlled laboratory experiments, they are often unreliable when used in real-world settings due to unreliable performance during the dynamic contractions required for functional use. Dynamic contractions are any contractions that result in changes in patterns and these can be quite complex. This work focused on a specific type of dynamic contractions: class transitions between intended classes as the literature reports that these can often result in unintended and erroneous activation of the prosthesis.

To solve this problem, this work investigated methods to determine ***when*** users are transitioning and methods to determine ***how*** the system should respond to class transitions. These investigations, performed offline, allowed the development of a state-based classification approach with novel feature conditioning methods that force features towards the class means. The state-based classification method was then evaluated in real-time (user-in-the-loop) task to verify if it improved usability, with promising results.

6.2 Empirical Findings

The first part of this work investigated methods to determine *when* users were in class transitions. This problem was two-fold: first, it was important to be able to estimate the features on short FL so that the features were able to capture class transitions and second, it was important to develop a method to characterize class transitions from the steady-state to be able to implement distinct decision methods for both.

The solution to this problem was the development of a novel feature conditioning method combined with a state-evaluator. The feature conditioning methods forced the features to converge towards the class means, improving the SNR of the features (reducing the variance) and also minimizing the number of patterns that were near the class boundaries. This not only improved the classifier confidence (suggesting fewer decisions that where in an indeterminate state), but it also facilitated the detection of features that were due to transitions and that could result in inadvertent activations. Class transitions were detected using a state-evaluator that partitioned features as either being in a steady-state (when they were near a class center), or in an indeterminate state (while traveling between the class centers). It was found that class transitions were not the only time that features were in an indeterminate state. This suggested that indeterminate state control rather than class transition control is a more general approach to solving the second part of this work: determining *how* to control features that were in class transitions, or more generally, in an indeterminate state.

Both the offline and real-time results showed that the solution to this problem is not a typical classification solution, but rather one that allows users to transition smoothly between contraction types with minimal lag. In other words, reaching for a new class or target is not about finding the exact path to get there, but rather about finding the simplest way of achieving it, with minimal effort. In fact, the state-based classification method that defaults to rest allowed users to perform the largest number of real-time tasks in the least amount of time. This is due to the fact that it is the method that required the users to perform fewer corrective motions to complete the tasks. Although slowing down the control speed rather than rejecting while in an indeterminate state allowed users to

perform the same number of successful tasks as rejection, these took longer as they often resulted in corrective motions. This confirms, as the literature suggests [89], that it is better to do nothing when an error is possible, than to select an incorrect class (i.e. unintended by the user).

Post-processing methods showed promising offline results but poor real-time results due to lag in the decision stream, and therefore user feedback, resulting in user frustration.

The real-time results also revealed a learning effect. Most participants were inexperienced EMG users, but as they became more experienced, the results obtained using the different control methods started to converge. This type of motor learning has been observed in grasping studies [122] and this result extends to pattern recognition. This suggests that with time, users may be able to perform smooth class transitions without the need for assistance by a system that overtly rejects decisions. This concurs with the observation that manual reaching tasks are undertaken in a conservative manner, favoring accuracy over speed [123]. Because subjects were inexperienced, the learning effect may also be a reflection of the power law of practice, which states that there is a decreasing exponential relationship between the time it takes users to react to a particular task and the number of practice trials taken [124]. In other words, learning eventually plateaus after several trials. This suggests that performing more trials could have prevented a learning effect. It is worth mentioning that performing more than three trials would result in a long experimental protocol (more than 2 hours) increasing the likelihood of noticing the effects of muscle fatigue [73] on the results which is undesirable.

Regardless, the proposed method could be used as an assistive control method while users learn to actuate control methods that offer more complex manipulation. It might also serve as a more appropriate control system for users that are challenged by the myoelectric control task. Clearly more work is required to understand this.

6.3 Implication and Contribution

Improving myoelectric pattern recognition control robustness under dynamic conditions has been a problem visited by several authors [45] [67] [72] [79] [86] [87] [88] [89] [90] [91]. There are several approaches to this solution proposed in the literature:

1) Provide the classifier with a training set that allows more variability by including class transitions (and varying intensities, as well as patterns while users are in different positions) in the training data [2] [71] [77] [85];
2) Post-processing of the decision stream [42] [94];
3) Use a classification approach, such as HMM [67] [79], that could handle the temporal evolution of the features while in class transitions;
4) Use a state-based or hybrid approaches that detects class transitions [45] [72] [79] [86] [87] [88];
5) Reject unfamiliar or confounding patterns [89] [90] [91] [92].

Since the first three approaches attempt to make correct or corrective decisions during transitions, they have the potential of generating involuntary actions which, as shown in this work, strongly affect the user's ability to perform real-time tasks.

As with the state-based and hybrid approaches suggested in the literature, the state-based classification approach proposed in this work attempts to detect class transitions.

Unlike previous methods described in the literature that conditioned the signal in order to detect the onset of class transitions, this work conditioned the features using a novel feature conditioning method. Contrary to these previous methods, this novel feature conditioning method does not depend on a threshold that varies across subjects, and does not increase latency.

This work supported, as the literature suggests, that rejecting unknown patterns (such as those produced during transitions between classes) is beneficial because it requires fewer corrections from the user. However, it is important to know when to perform rejection otherwise the system may be overly inactive. Previous attempts have performed rejection by evaluating the class decisions confidence; the multiple binary classifier [89] and 1-vs-1 [90] compare the outputs of different classifiers and reject those that do not obtain a unanimous vote, whereas the confidence rejection method [92] evaluates the posterior class probability of an LDA classifier and rejects any decision with confidence below a specified threshold. Unlike these methods which evaluate the class decisions to determine rejection, this work evaluates the conditioned features to determine if rejection should occur. This has the effect of preventing over-rejection which may occur due to a poorly trained class. In fact, the choice to rejection a pattern is made before classification; hence, it is independent of the class decision. Another advantage of the proposed method is that, although it requires a threshold as with other rejection methods in the literature, its threshold is subject-independent, and can be preset without training.

Another implication of this work is the improvement in the feature quality due to feature conditioning. The literature states that a high-quality feature should [9]:

1) be separable between classes;

2) be repeatable, as selected features should provide consistent patterns between repetitions of a given contraction type, and preserve their repeatability during the non-deal conditions encountered during practical use of a prosthesis, and;
3) be extracted from a small window size to allow the control system to be responsive.

Feature conditioning improved the features' class decision confidence and the SNR of the features suggesting that it also improved the repeatability of the features. Additionally, the proposed feature conditioning method allowed the use of shorter FL.

Finally, this work showed a rapid learning effect in the real-time study. It is well known that, in myoelectric pattern recognition control systems, there is a learning effect [122] and subjects get discouraged by these high learning curves without proper training or guidance. The rapid learning effect reported in this real-time study suggests that the proposed method could be used to assist beginners in learning how to control myoelectric pattern recognition control systems with greater success.

6.4 Limitations and Recommendations for Future Research

The real-time test performed in this study had several limiting factors that are worth mentioning: 1) the tasks were simple and required a single contraction to be completed (in the ideal case), 2) participants were still learning when measurements were taken, 3) the experimental work was perform in a virtual environment which did not use physical devices, and 4) the algorithms were not tested by amputee end-users. The control scheme that rejects unknown patterns performed the best during the real-time test. The tasks, however, were simple as users could complete them with a single contraction unless an errant motion was activated requiring compensation. Therefore, these tasks may have

been biased towards the rejection method tested; the system that generated the fewest errant motions would likely allow users to perform the tasks in the least amount of time. Future work should consider evaluating more complex tasks as well as increasing the number of trials or using experienced users to minimize the learning effect on the results. Future work should also consider participants with upper limb absence.

Future work should also consider alternatives to rejecting features in an indeterminate state. This work considered state-dependent proportional speed control as an alternative, which also showed promising results. However, more in depth studies should be performed to determine the best proportional control method while in an indeterminate state. These could be similar to the velocity ramp technique proposed by Simon *et al.* [94].

Future studies should also consider evaluating the system's tradeoff between the real-time accuracy and responsiveness more closely. This work showed that unintentional class activations had a strong impact on the system's speed and accuracy tradeoff. However, it also assumed that shorter FL would allow better responsiveness; although it was the case offline, the real-time results showed otherwise as the system with shortest FL did not allow users to perform the tasks in less time. Determining the exact factors that affect the speed and accuracy tradeoff of the control system would facilitate the development of a better control method while in an indeterminate state.

There was evidence in this work that feature conditioning improved the quality of the features. Conditioned features should be more resilient to not only dynamic contractions such as those with varying intensity and varying positions, but also to physical and physiological changes in the patterns that may degrade the performance of the system.

Future work should evaluate the effects of pattern changes on conditioned features as well as evaluate the use of feature conditioning to help select features to adapt the system to pattern changes.

Finally, this work showed a strong learning effect. Future work should investigate this learning effect more closely by determining if the proposed system could assist users in either learning more complex control methods or in attaining a suitable level of usability. It would also be important to verify which system offers the best control once the learning has plateaued.

References

[1] R.N. Scott and P.A. Parker, "Myoelectric Prostheses: State of the Art," *Journal of Medical Engineering & Technology*, vol. 12, no. 4, pp. 143-151, July-August 1988.

[2] B. Hudgins, P.A. Parker, and R.N. Scott, "A New Strategy for Multifunction Myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82-94, January 1993.

[3] K.B. Englehart, Signal Representation for Classification of the Transient Myoelectric Signal, Ph.D. Dissertation, Department of Electrical Engineering, Univerisity of New Brunswick, Fredericton, New Brunswick, 1998.

[4] P. Parker and R.N. Scott, "Myoelectric Control of Prostheses," *CRC Critical Reviews in Biomedical Engineering*, vol. 13, no. 4, pp. 283-309, 1986.

[5] B.A. Lock, K. Englehart, and B. Hudgins, "Real-Time Myoelectric Control in a Virtual Environment to Relate Usability vs. Accuracy," in *Proceedings of the 2005 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2005.

[6] B.A. Lock, Design and Interactive Assessment of Continuous Multifunction, M.S. Thesis, Department of Electrical Engineering, Univerisity of New Brunswick, Fredericton, New Brunswick, 2005.

[7] T.A. Kuiken et al., "Targeted Muscle Reinnervation for Real-time Myoelectric Control of Multifunction Artificial Arms," *Journal of the American Medical Association*, vol. 301, no. 6, pp. 619-628, February 2009.

[8] E.J Scheme and K.B Englehart, "Validation of a Selective Ensemble-Based Classification Scheme for Myoelectric Control Using a Three Dimensional Fitts' Law Test," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 4, pp. 616-623, July 2013.

[9] R. Merletti and P.A. Parker, *Electromyography: Physiology, Engineering, and Non-Invasive Applications*. New Jersey, USA: John Wiley & Sons, 2004.

[10] R.N. Scott, "Myoelectric Control of Prostheses: A Brief History," in *Proceedings of the 1992 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 1992.

[11] L. Hargrove, Towards Clinically Robust Pattern Recognition Based Myoelectric Control, Ph.D. Dissertation, Department of Electrical Engineering, Univerisity of New Brunswick, Fredericton, New Brunswick, 2008.

[12] P. Parker, J. Stuller, and R.N. Scott, "Signal Processing for the Multistate Myoelectric Channel," *Proceedings of the IEEE*, vol. 65, no. 5, pp. 662-674, May 1977.

[13] J.V. Basmajian and C.J. De Luca, *Muscles Alive: Their Functions Revealed by Electromyography*. Baltimore, USA: Williams & Wilkins, 1985.

[14] C.K. Battye, A. Nightingale, and J. Whillis, "The Use of Myoelectric Currents in the Operation of Prostheses," *Journal of Bone and Joint Surgery*, vol. 37, no. 3, pp. 506-510, August 1955.

[15] A.H. Bottomley, "Myoelectric Control of Powered Prostheses," *Journal of Bone and Joint Surgery*, vol. 47, no. 3, pp. 411-415, August 1965.

[16] A.H. Bottomley, "Working Model of a Myoelectric Control System," in *Proceedings of the International Symposium of Applications in Automatic Control in Prosthetics Design*, Belgrade, Yugoslavia, 1962, pp. 37-45.

[17] R.D. Rothchild and R.W. Mann, "An EMG Controlled, Force Sensing, Proportional Rate, Elbow Prosthesis," in *Proceedings of the 1966 Biomedical Engineering Symposium*, vol. 1, Milwaukee, Wisconsin, USA, 1966, pp. 106-109.

[18] H.B. Evans, Z. Pan, P.A. Parker, and R.N. Scott, "Signal Processing for Proportional Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. BME-31, no. 2, pp. 207-211, February 1984.

[19] N. Hogan and R.W. Mann, "Myoelectric Signal Processing: Optimal Estimation Applied to Electromyography -Part 1: Derivation of the Optimal Myoprocessor," *IEEE Transactions on Biomedical Engineering*, vol. 27, no. 7, pp. 382-395, July 1980.

[20] A. Fougner, O. Stavdahl, P. Kyberd, Y. Losier, and P. Parker, "Control of Upper Limb Prostheses: Terminology and Proportional Myoelectric Control - A review," *IEEE Transactions on Neural Systems Rehabilitation Engineering*, vol. 21, no. 1, pp. 663-677, September 2012.

[21] F.Q. Xiong and E. Shwedyk, "Some Aspects of Nonstationary Myoelectric Signal Processing," *IEEE Transactions on Biomedical Engineering*, vol. 34, no. 2, pp. 166-172, February 1987.

[22] S. Meek and S. Fetherston, "Comparison of Signal-to-Noise Ratio of Myoelectric Filters for Prosthesis Control," *Journal of Rehabilitation Research*, vol. 29, no. 4, pp. 9-20, September 1992.

[23] N. Hogan and R.W. Mann, "Myoelectric Signal Processing: Optimal Estimation Applied to Electromyography - Part II: Experimental Demonstration of Optimal Myoprocessor Performance.," *IEEE Transactions on Biomedical Engineering*, vol. 27, no. 7, pp. 396-410, July 1980.

[24] N. Hogan and R.W. Mann, "Limitations of Existing Proportional EMG Processors," in *Proceedings of the 1974 Conference on Engineering Devices in Rehabilitation*, Boston, Massachusetts, USA, 1974, pp. 78-81.

[25] T. St-Amant, D. Rancourt, and E. Clancy, "Effect of Smoothing Window Length on RMS EMG Amplitude Estimates," in *Proceedings of the 1996 Annual Conference of the IEEE Northeast Bioengineering Conference*, New Brunswick, New Jersey, USA, 1996, pp. 93-94.

[26] T.W. Calvert and A. Chapman, "The Relationship Between the Surface EMG and Force Transients in Muscle: Simulation and Experimental Studies," *Proceedings of the IEEE* , vol. 65, no. 5, pp. 682-689, May 1977.

[27] T. Kiryu, Y. Saitoh, and K. Ishioka, "Investigation on Parametric Analysis of Dynamic EMG Signals by a Muscle-Structured Simulation Model," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 3, pp. 280-288, March 1992.

[28] R.W. Norman and P.V. Komi, "Electromechanical Delay in Skeletal Muscle Under Normal Movement Conditions," *Journal of Acta Physiologica*, vol. 106, no. 3, pp. 241-248, July 1979.

[29] T.R. Farrell and R.F. Weir, "The Optimal Controller Delay for Myoelectric Prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 111-118, March 2007.

[30] S.C. Jacobsen, S.G. Meek, and R.R. Fullmer, "An Adaptive Myoelectric Filter," in *Proceedings of the IEEE/Engineering in Medicine and Biology Society Annual Conference*, New York, New York, USA, 1984, pp. 592-596.

[31] E.A. Clancy, "Electromyogram Amplitude Estimation with Adaptive Smoothing Window Length," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 6, pp. 717-729, June 1999.

[32] H. Miyano, T. Masuda, and T. Sadoyama, "A Note on the Time Constant in Low-Pass Filtering of Rectified Surface EMG," *IEEE Transaction on Biomedical Engineering*, vol. 27, no. 5, pp. 274-278, May 1980.

[33] D. Graupe and W.K. Cline, "Functional Seperation of EMG Signals via ARMA Identification Methods for Prosthesis Control purposes," *IEEE Transactions on Systems, Man, and Cybernetics* , vol. 5, no. 2, pp. 252-259, March 1975.

[34] P.S. Maybeck, "The Kalman Fitler: An Introduction to Concepts," in *Autonomous Robot Vehicles*. New York, USA: Springer, 1990, pp. 194-204.

[35] P. Parker, K. Englehart, and B. Hugins, "Myoelectric Signal Processing for Control of Powered Limb Prostheses," *Journal of Electromyography and Kinesiology*, vol. 16, no. 6, pp. 541-548, December 2006.

[36] J.L. Segil and R.F. Weir, "Novel Postural Control Algorithm for Control of Multifunctional Myoelectric Prosthetic Hands," *Journal of Rehabilitation Research and Development*, vol. 52, no. 4, pp. 449-466, 2015.

[37] F.R. Finley and R.W. Wirta, "Myocoder Studies of Multiple Myopotential Responses," *Journal of Archives of Physical Medicine and Rehabilitation*, vol. 48, no. 11, pp. 598-601, October 1967.

[38] P. Lawrence, P. Herberts, and R. Kadefors, "Experiences with a Multifunctional Hand Prosthesis by Myoelectric Patterns," in *Proceedings of the 1975 International Symposium on External Control of Human Extremities*, Dubrovnik, Croatia, 1975,

pp. 47-65.

[39] J.H. Lyman, A. Freedy, and R. Prior, "Fundamental and Applied Research Related to the Design and Development of Upper-Limb Externally Powered Prostheses," *Bulletin of Prosthetics Research*, vol. 13, pp. 184-195, 1976.

[40] E.J. Scheme and K.B. Englehart, "Electromyogram Pattern Recognition for Control of Powered Upper-Limb Prostheses: State of the Art and Challenges for Clinical Use," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 643-660, 2011.

[41] M.F. Kelly, P.A. Parker, and R.N. Scott, "The Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 3, pp. 221-230, March 1990.

[42] K. Englehart and B. Hudgins, "A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848-854, July 2003.

[43] K.B. Englehart, B. Hudgins, and P.A. Parker, "A Wavelet-Based Continuous Classification Scheme for Multifunction Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302-311, March 2001.

[44] A. Simon and L. Hargrove, "A Comparison of the Effects of Majority Vote and a Decision-Based Velocity Ramp on Real-Time Pattern Recognition Control," in *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, USA, 2011, pp. 3350-3353.

[45] T. Lorrain, N. Jiang, and D. Farina, "Surface EMG Classification during Dynamic Contractions for Multifunction Transradial Prostheses," in *Proceedings of the 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, 2010, pp. 2766-2769.

[46] L.H Smith, L. Hargrove, B. Lock, and T. Kuiken, "Determining the Optimal Window Length of Pattern Recognition Based Myoelectric Control: Balancing the Competing Effects of Classification and Control Delay," *IEEE Transaction on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 186-192, April 2011.

[47] L. Liu, P. Liu, E. Clancy, E. Scheme, and K. Englehart, "Signal Whitening Preprocessing for Improved Classification Accuracies in Myoelectric Prostheses," in *Proceedings of the 2011 Annual Conference of the IEEE Northeast Bioengineering Conference*, Troy, New York, USA, 2011, pp. 1-2.

[48] G. Hefftner, W. Zucchini, and G. Jaros, "The Electromyogram (EMG) as a Control Signal for Functional Neuromuscular Stimulation - Part I: Autoregressive Modeling as a Means of EMG singiature Discrimination," *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 4, pp. 230-237, April 1988.

[49] L. Hargrove, G. Li, K. Englehart, and B. Hudgins, "Principal Components Analysis Preprocessing for Improved Classification Accuracies in Pattern-Recognition-Based Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1407-1414, May 2009.

[50] J. Chu and Y. Lee, "Conjugate-Prior-Penalized Learning of Gaussian Mixture Models for Multifunction Myoelectric Hand Control," *IEEE Transactions on Neural Systems and Rehabilitation*, vol. 17, no. 3, pp. 287-297, February 2009.

[51] M. Khezri and M. Jahed, "Real-Time Intelligent Pattern Recognition Algorithm for Surface EMG Signals," *Biomedical Engineering Online*, vol. 6, no. 45, December 2007.

[52] F. Chan, Y. Yang, F. Lam, Y. Zhang, and P.A. Parker, "Fuzzy EMG Classification for Prosthesis Control," *IEEE Transactions on Biomedical Engineering*, vol. 8, no. 3, pp. 305-311, September 2000.

[53] B. Karlik, M. Tokhi, and M. Alci, "A Fuzzy Clustering Neural Network Architecture for Multifunction Upper-Limb Prosthesis," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 11, pp. 1255-1261, October 2003.

[54] S. Park and S. Lee, "EMG Pattern Recognition Based on Artificial Intelligence Techniques," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 4, pp. 400-405, August 1998.

[55] A. Soares, A Andrade, E. Lamounier, and R. Carrijo, "The Development of Virtual Myoelectric Prosthesis Controlled by an EMG Pattern Recogniotion System Based on Neural Networks," *Journal of Intelligent Information Systems*, vol. 21, no. 2, pp. 127-141, September 2003.

[56] S.P. Lee, J.S. Kim, and S.H. Park, "An Enhanced Feature Extraction Algorithm for EMG Pattern Classification," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 439-443, Dec 1996.

[57] K. Englehart, B. Hudgins, P. Parker, and M. Stevenson, "Time-Frequency Representation for Classification of the Transient Myoelectric Signal," in *Proceedings of the 1998 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Honk Kong, China, 1998, pp. 2627-2630.

[58] J. Chu, I. Moon, Y. Lee, S. Kim, and M. Mun, "A Supervised Feature-Projection-Based Real-Time EMG Pattern Recognition for Multifunction Myoelectric Hand Control," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 282-290, June 2007.

[59] M. Zardoshhti-Kermani, B. Wheeler, K. Badie, and R. Hashemi, "EMG Feature Selection for Movement Control of a Cybernetic Arm," *International Journal of Cybernetics and Systems*, vol. 26, no. 2, pp. 189-210, August 1995.

[60] S.H. Park and S.P. Lee, "EMG Pattern Recognition Based on Artificial Intelligence Techniques," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 4, pp.

400-405, December 1998.

[61] M.F. Kelly, P.A. Parker, and R.N. Scott, "Myoelectric Signal Analysis using Neural Networks," *IEEE Engineering in Medicine and Biology Magazine*, vol. 9, no. 1, pp. 61-64, May 1990.

[62] C. Castellini and P. Smagt, "Surface EMG in Advanced Hand Prosthetics," *Journal of Intelligent Information Systems*, vol. 21, no. 1, pp. 127-141, January 2003.

[63] J. Chu, I. Moon, and M. Mun, "A Real-Time EMG Pattern Recognition System Based on Linear-Nonlinear Feature Projection for Multifunction Myoelectric Hand," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2232-2239, November 2006.

[64] K. Englehart, B. Hudgins, P. Parker, and M. Stevenson, "Classification of The Myoelectric Signal using Time-Frequency Based Representation," *Medical Engineering & Physics*, vol. 21, no. 6-7, pp. 431-438, September 1999.

[65] A.B. Ajiboye and R.F. Weir, "A Heuristic Fuzzy Logic Approach to EMG Pattern Recognition for Multifunctional Prosthesis Control," *IEEE Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 280-291, September 2005.

[66] Y. Huang, K.B. Englehart, B. Hudgins, and A.D.C. Chan, "A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp. 1801-1811, November 2005.

[67] A. Chan and K. Englehart, "Continuous Myoelectric Control for Powered Prostheses Using Hidden Markov Models," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 121-124, January 2005.

[68] S. Moon and H. Qi, "Effective Dimensionaliry Reduction based on Support Vector Machine," in *Proceedings of the 2010 International Conference on Pattern Recognition*, Istanbul, 2010, pp. 173-176.

[69] S. Bitzer and P. Smagt, "Learning EMG Control of a Robotic Hand: Towards Active Prosthesis," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, 2006, pp. 2819-2823.

[70] M. Yoshikawa, M. Mikawa, and K. Tanaka, "A Myoelectric Interface for Robotic Hand Control Using Support Vector Machine," in *Proceedings of the IEEE/RSJ Interational Conference on Intelligent Robots and Systems*, San Diego, California, USA, 2007, pp. 2723-2728.

[71] T. Lorrain, N. Jiang, and D. Farina, "Influence of the Training Set on the Accuracy of Surface EMG Classification in Dynamic Contractions for the Control of Multifunction Prostheses," *Journal of NeuroEngineering and Rehabilitation*, vol. 8, no. 25, May 2011.

[72] T. Lorrain, N. Jiang, and D. Farina, "A State-Based, Proportional Myoelectric

Control Method: Online Validation and Comparison with the Clinical State-of-the-Art," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 110, Jully 2014.

[73] K. Biron, EMG Pattern Recognition Adaptation, 2010, M.S. Thesis, Department of Electrical Engineering, Univerisity of New Brunswick, Fredericton, New Brunswick, 2010.

[74] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm Motions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 210-211, April 2003.

[75] D. Tkach, H. Huang, and T. Kuiken, "Study of Stability of Time-Domain Features for Electromyographic Pattern Recognition," *Journal of NeuroEngineering and Rehabilitation*, vol. 7, no. 21, May 2010.

[76] L. Hargrove, K. Englehart, and B. Hudgins, "A Training Strategy to Reduce Classification Degradation due to Dlectrode Displacemenets in Pattern Recognition Based Myoelectric Control," *Journal of Biomedical Signal Processing and Control*, vol. 3, no. 2, pp. 175-180, April 2008.

[77] E. Scheme, K. Biron, and K. Englehart, "Improving Myoelectric Pattern Recognition Postional Robustness Using Advanced Training Protocol," in *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, USA, 2011, pp. 4828-4831.

[78] A. Fougner, E. Scheme, A. Chan, E. Englehart, and O. Stavdahl, "Resolving the Limb Position Effect in Myoelectric Pattern Recognition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 6, pp. 644-651, December 2011.

[79] J. Kwon, D. Lee, S. Lee, N. Kim, and S. Hong, "EMG Signals Recognition for Continuous Prosthetic Arm Control Purpose," in *Proceedings of the 1996 Asia Pacific Conference of the IEEE on Circuits and Systems*, Seoul, South Korea, Nov 1996, pp. 365-368.

[80] M.A. Oskoei and H. Hu, "Myoelectric Control Systems - A survey," *Journal of Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275-294, October 2007.

[81] M. Oskoei and H. Huosheng, "Support Vector Machine-Based Classification Scheme for Myoelectric Control Applied to Upper Limb," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1956-1964, August 2008.

[82] L. Hargrove, Y. Losier, B. Lock, K. Englehart, and B. Hudgins, "A Real-Time Pattern Recognition Based Myoelectric Control Usability Study Implemented in a Virtual Environment," in *Proceedings of the 2007 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Lyon, France, 2007, pp. 4842-4845.

[83] J. Sensinger, B. Lock, and T. Kuiken, "Adaptive Pattern Recognition of

Myoelectric Signals: Exploration of Conceptual Framework and Practical Algorithms," *IEEE Transactions on Neural Systems Rehabilitation Engineering*, vol. 17, no. 3, pp. 270-278, June 2009.

[84] B.A. Lock, A. Simon, K. Stubblefield, and L. Hargrove, "Prosthesis-Guided Training for Practical use of Pattern Recognition Control of Prostheses," in *Proceedings of the 2011 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2011.

[85] E. Scheme and K. Englehart, "Training Strategies for Mitigating the Effect of Proportional Control on Classification in Pattern Recognition Based Myoelectric Control," *Journal of Prosthetics and Orthodics*, vol. 25, no. 2, pp. 76-83, April 2013.

[86] Y. Al-Assaf, "Surface Myoelectric Signal Analysis: Dynamic Approaches for Change Detection and Classification," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 11, pp. 2248-2256, November 2006.

[87] S. Solnik, P. Rider, K. Steinweg, P. DeVita, and T. Hortobagyi, "Teager-Kaiser Eenergy Operator Signal Conditioning Improves EMG Onset Detection," *European Journal of Applied Physiology*, vol. 110, no. 3, pp. 489-498, June 2010.

[88] T. Lorrain, I.K. Niazi, N. Jiang, and D. Farina, "Movement Onset Detection in Various Positions for State-Based Myo-Control Scheme," in *Proceedings of the 2011 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2011.

[89] L. Hargrove, E.J. Scheme, K.B. Englehart, and B. Hudgins, "Multiple Binary Classifications via Linear Discriminant Analysis for Improved Controllability of a Powered Prosthesis," *IEEE Transactions on Neural Systems Rehabilitation Engineering*, vol. 18, no. 1, pp. 49-57, January 2010.

[90] E. Scheme, K. Englehart, and B. Hudgins, "Selective Classification for Improved Robustness of Myoelectric Control Under Nonideal Conditions," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1698-1705, June 2011.

[91] L. Xiao-Hui and C. Chin-Seng, "Rejection of Non-Meaningful Activities," in *Proceedings of the 2005 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2005, pp. 122-126.

[92] E.J. Scheme, B.S. Hudgins, and K.B. Englehart, "Confidence-Based Rejection for Improved Pattern Recognition Myoelectric Control," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 6, pp. 1563-1570, January 2013.

[93] E.J. Scheme and K.B. Englehart, "A Comparison of Classification Based Confidence Metrics for use in the Design of Myoelectric Control Systems," in *Proceedings of the 2015 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Milan, Italy, 2015.

[94] A.M. Simon, L. Hargrove, B.A. Lock, and T. Kuiken, "A Strategy for Minimizing

the Effect of Misclassification during Real-Time Pattern Recognition Myoelectric Control," in *Proceedings of the 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, Minnesota, USA , 2009, pp. 1327-1330.

[95] L. Hargrove, P. Zhou, K. Englehart, and T. Kuiken, "The Effect of ECG Interference on Pattern-Recognition-Based Myoelectric Control for Targeted Muscle Reinnervated Patients," *IEEE Transactions on Biomedical Engineering*, vol. 9, no. 56, pp. 2197-2201, September 2009.

[96] A.M. Simon, K. Stern, and L. Hargrove, "A Comparison of Proportional Control Methods for Pattern Recognition Control," in *Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, Massachusetts, USA, 2011, pp. 3354-3357.

[97] E. Scheme et al., "Motion Normalized Proportional Control for Improved Pattern Recognition-Based Myoelectric Control," *IEEE Transactions on Neural Systems and Rehabilitations Engineering*, vol. 22, no. 1, pp. 149-157, January 2014.

[98] T.A. Kuiken, G.A. Dumanian, R.D. Lipschutz, L.A. Muller, and K.A. Stubblefield, "The User of Targeted Muscle Reinnervation for Improved Myoelectric Prosthesis Control in a Bilateral Shoulder Disarticulation Amputee," *Journal of Prosthetics and Orthotics International*, vol. 28, no. 3, pp. 245-253, December 2004.

[99] V. Mathiowetz, G. Volland, N. Kashman, and K. Weber, "Adults Norms for the Box and Blocks Test of Manual Dexterity," *American Journal of Occupational Therapy*, vol. 39, no. 6, pp. 386-391, June 1985.

[100] A. Simon, L. Hargrove, B. Lock, and T. Kuiken, "Target Achievement Control Test: Evaluating Real-Time Myoelectric Pattern-Recognition Control of Multifunctional Upper-Limb Prostheses," *Journal of Rehabilitation Research and Development*, vol. 48, no. 6, pp. 619-628, 2011.

[101] S. Zhai, J. Kong, and X. Ren, "Speed-Accuracy Trade-off in Fitts' Law Tasks - On the Equivalency of Acutal and Nominal Pointing Precision," *International Journal of Human-Computer Studies*, vol. 61, no. 6, pp. 823-856, December 2004.

[102] L. Liu, P. Liu, E. Clancy, E.J. Scheme, and K.B. Englehart, "Electromyogram Whitening for Improved Classification Accuracy in Upper Limb Prosthesis Control," *IEEE Transactions on Neural Systems and Rehabilitation*, vol. 21, no. 5, pp. 767-774, March 2013.

[103] E. Clancy and K. Farry, "Adaptive Whitening of the Electromyogram to Improve Amplitude Estimation," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 6, pp. 709-719, June 2000.

[104] Y. Huang, K. Englehart, B. Hudgins, and A. Chan, "A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 11, pp.

1801-1811, November 2005.

[105] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35-45, March 1960.

[106] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Massachusetts, USA: MIT Press, 2005.

[107] A. Chan and K.B. Englehart, "Continuous Classification of Myoelectric Signals for Powered Prostheses using Gaussian Mixture Models," in *Proceedings of the 2003 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Cancun, Mexico, 2003, pp. 2841-2844.

[108] E.J. Scheme and K.B. Englehart, "A Flexible user Interface for Rapid Prototyping of Advanced Real-Time Myoelectric Control Schemes," in *Proceedings of the 2008 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2008.

[109] W.A. Wilson, Y.G. Losier, P.A. Parker, and D. Lovely, "A Bus-Based Smart Myoelectric Electrode/Amplifier—System Requirements," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3290-3299, October 2011.

[110] R. Boostani and M.H. Mohammad, "Evaluation of the Forearm EMG Signal Feature for the Control of a Prosthetic Hand," *Journal of Physiological Measurement*, vol. 24, no. 2, pp. 309-319, March 2003.

[111] E. Clancy and N. Hogan, "Multiple Site Electromyography Amplitude Estimation," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 2, pp. 203-211, February 1995.

[112] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond Accuracy, F-score and ROC: A Family of Discriminant Measures for Performance Evaluation," in *Proceedings of the 2006 Australian Joint Conference on Artificial Intelligence*, Hobart, Australia, 2006, pp. 1015-1-21.

[113] K.H. Zou, J. O'Malley, and L. Mauri, "Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models," *Journal of American Heart Association*, vol. 115, no. 5, pp. 654-657, February 2007.

[114] C. Metz, "Basic Principles of ROC Analysis," *Journal of Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283-298, October 1978.

[115] T.R. Farrell and R. Weir, "Analysis Window Induced Controller Delay for Multifunctional Prostheses," in *Proceedings of the 2008 Myoelectric Controls Symposium*, Fredericton, New Brunswick, Canada, 2008, pp. 13-15.

[116] K. Englehart, B. Hudgins, and A. Chan, "Continuous Multifunction Myoelectric Control Using Pattern Recognition," *Journal of Technology and Disability*, vol. 15, no. 2, pp. 95-103, August 2003.

[117] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communications*.

Urbana, Illinois, USA: University of Illinois Press, 1949.

[118] S. MacKenzie, "Fitts' Law as a Research and Design Tool in Human-Computer Interaction," *Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 91-139, March 1992.

[119] D. Beamish, S.A. Bhatti, S. MacKenzie, and J. Wu, "Fifty Years Later: A Neurodynamic Explanation of Fitts' Law," *Journal of The Royal Society Interface*, vol. 3, no. 10, pp. 649-654, April 2006.

[120] R.S. Woodworth, "The Accuracy of Voluntary Movement," *The Psychological Review: Monograph Supplements*, vol. 3, no. 3, pp. 1-114, July 1899.

[121] S. Zhai, "Characterizing Compuer Input with Fitts' Law Parameters - The Information and Non-Information Aspects of Pointing," *International Journal of Human Computer Studies*, vol. 61, no. 6, pp. 791-809, December 2004.

[122] H. Bouwsema, C.K. Van der Sluis, and R. M. Bongers, "Changes in Performance Over Time While Learning," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 16, pp. 1-16, February 2014.

[123] N. Qian, Y. Jiang, Z.-P. Jiang, and P. Mazzoni, "Movement Duration, Fitts's law, and an Infinite-Horizon Optimal Feedback Control Model for Biological Motor Systems," *Journal of Neural Computation*, vol. 25, no. 3, pp. 697-724, March 2013.

[124] S.K. Card, P.M. Thomas, and N. Allen, *The Psychology of Human-Computer-Interaction*. Mahwah, New Jersey, USA: Lawrence Erlbaum Associates, 1983.

[125] A. Webb, *Statistical Pattern Recognition*, 2nd ed. Chichester, UK: John Wiley & Sons Ltd, 2002.

[126] G.C. Goodwin and K. Sang Sin, *Adaptive Filtering Prediction and Control*, 1st ed. Englewood Cliffs, New Jersey, USA: Prentice-Hall, 1984.

[127] C.E. Thomaz, Maximum Entropy Covariance Estimate for Statistical Pattern Recognition, 2004, Ph.D. Dissertation, Department of Computing, Univerisity of London, London, England, 2004.

Appendix A – Features

A.1 Time Domain (TD) Features

Time domain features [110] include: the mean absolute value (MAV), the wavelength (WL), the zero crossing (ZC) and the turns (TURNS). Given a segment of data with frame length of *T* samples, denoted $\boldsymbol{\chi}_t = \{\chi_{t-T}, \dots, \chi_t\}$, the TD feature at time *t* may be defined as follows:

$$MAV_t = \frac{\sum_i |\chi_i|}{T} \quad for\ i = t - T, \dots, t \tag{A-1}$$

$$WL_t = \frac{\sum_i (|\chi_i - \chi_{i-1}|)}{T - 1} \quad for\ i = t - (T - 1), \dots, t \tag{A-2}$$

$$ZC_t = \frac{\sum_i ((\chi_i \le 0\ \wedge \chi_{i-1} \ge 0) \vee ((\chi_i \ge\ 0\ \wedge \chi_{i-1} \le 0))}{T - 1} \quad for\ i = t - (T - 1), \dots, t \tag{A-3}$$

$$TRUNS_t = \frac{\sum_i ((\chi_i \ge\ \chi_{i-1}\ \wedge \chi_i \ge \chi_{i+1}) \vee ((\chi_i \le\ \chi_{i-1}\ \wedge \chi_i \le \chi_{i+1}))}{T - 2} \quad for\ i = t - (T - 2), \dots, t \tag{A-4}$$

An extracted feature set of TD features contains *p* features and is defined as follows:

$$\begin{aligned} \vec{\boldsymbol{x}} &= [MAV_{ch1}\ WL_{ch1}\ ZC_{ch1}\ TURNS_{ch1} \dots\ MAV_{chN}\ WL_{chN}\ ZC_{chN}\ TURNS_{chN}] \\ &= [\hat{x}_1\ \hat{x}_2\ \hat{x}_3 \dots \hat{x}_p] \end{aligned} \tag{A-5}$$

where *chN* is the total number of channels and the total number of features *p* is equal to the number of features multiplied by the number of channels (i.e. $p = 4\text{x}chN$).

A.2 Euclidian Distance Features

To define the Euclidian distance (ED) features, some parameters for a system with N_C classes need to be defined:

$$\vec{\boldsymbol{x}}^{\boldsymbol{TD}} = [x_1^{TD}\ x_2^{TD}\ x_3^{TD}\ , \dots x_p^{TD}] \qquad \text{(A-6)}$$

$$\boldsymbol{m_i} = \frac{1}{T_i}\sum_{n=1}^{T_i} \boldsymbol{x_n^{TD}} = [m_{i,1}\ m_{i,2}\ \dots m_{i,p}] \qquad \text{(A-7)}$$

$$\vec{\boldsymbol{x}}^{\boldsymbol{ED}} = [x_1^{ED}\ x_2^{ED}\ x_3^{ED}\ , \dots x_{N_C}^{ED}] \qquad \text{(A-8)}$$

where $\boldsymbol{x^{TD}}$ is a TD feature set with *p* features; $\boldsymbol{m_i}$ is the mean of class *i* obtained from T_i training samples; $m_{i,1}\ m_{i,2}\ \dots m_{i,p}$ are the mean of feature *1,2,…p* for class *i*, and; $\vec{\boldsymbol{x}}^{\boldsymbol{ED}}$ is the Euclidian distance feature set with Euclidian distance ED_i from class *i* computed as:

$$x_i^{ED} = \sqrt{\sum_{p=1}^{P}(x_p^{TD} - m_{p,i})} \quad , \ i = 1,2 \dots N_c \qquad \text{(A-9)}$$

Appendix B – LDA Classifier

B.1 Definition

To define the linear discriminant analysis (LDA) classifier, some parameters for a system with N_C classes need to be defined:

$$\vec{\boldsymbol{x}} = [x_1\ x_2\ \ x_3\ , \dots x_p] \tag{B-1}$$

$$g_i(\cdot), \quad i = 1,2, \dots N_C \tag{B-2}$$

where $\vec{\boldsymbol{x}}$ is a feature set pattern and $g_i(\cdot)$ is the discriminant function for class *i*. The LDA assumes that patterns can be discriminated between classes by evaluating their linear combination. For each class $i = 1,2, \dots N_C$ a linear combination of the feature set $\vec{\boldsymbol{x}} = \left[x_1, x_2 \dots, x_p\right]^T$ is evaluated: [125]

$$g_i(\vec{\boldsymbol{x}}) = \boldsymbol{w}_i^T\vec{\boldsymbol{x}} + \ w_{i0} = \sum_j^p w_{i,j}x_j + w_{i0} \tag{B-3}$$

where $g_i(\vec{\boldsymbol{x}})$ is the linear discriminant function of class i, and $\boldsymbol{w_i} = \{w_1, w_2, \dots w_p\}$ are the parametric weights. The pattern $\vec{\boldsymbol{x}}$ is assigned to class i if:

$$g_i(\vec{\boldsymbol{x}}) = \max_j\ g_j(\vec{\boldsymbol{x}}) \tag{B-4}$$

B.2 Training of Parametric Weights

The parametric weights are obtained from a priori information obtained from training samples. Consider a training set per independent class i with T_i feature set samples $\{\vec{\boldsymbol{x}}_{\boldsymbol{1}}, \dots \vec{\boldsymbol{x}}_{\boldsymbol{T}}\}$ where $\vec{\boldsymbol{x}} = [x_1\ \ x_2 \dots x_p]$. The *sample mean* $\overrightarrow{\boldsymbol{m}}_i$ and *pooled covariance matrix* $\boldsymbol{\Sigma}$ are defined as [125]:

$$\overrightarrow{\boldsymbol{m}}_{\boldsymbol{i}} = \frac{1}{T_i}\sum_{k=1}^{T_i}\vec{\boldsymbol{x}}_{\boldsymbol{k}} \tag{B-5}$$

$$\boldsymbol{S}_i = \frac{1}{T_i}\sum_{k=1}^{T_i}(\vec{\boldsymbol{x}}_{\boldsymbol{k}} - \vec{\boldsymbol{m}}_i)(\vec{\boldsymbol{x}}_{\boldsymbol{k}} - \vec{\boldsymbol{m}}_i) \tag{B-6}$$

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_C}\frac{T_i}{T}\boldsymbol{S}_i \tag{B-7}$$

where T is the total number of training samples (i.e. $T = T_1 + T_2 + \cdots T_{N_C}$). It can be shown that the parametric weights are defined as (please refer to [125] for the derivation):

$$\boldsymbol{w}_{\boldsymbol{i}} = \boldsymbol{\Sigma}^{-1}\,\vec{\boldsymbol{m}}_{\boldsymbol{i}} \tag{B-8}$$

$$w_0 = \log\big(p(i)\big) - \frac{1}{2}\vec{\boldsymbol{m}}_i^T\boldsymbol{\Sigma}^{-1}\,\vec{\boldsymbol{m}}_i \tag{B-9}$$

where $p(i)$ is defined as:

$$p(i) = \frac{T_i}{T} \tag{B-10}$$

B.3 Confusion Matrix

The confusion matrix makes it easy to determine if the classifier is commonly mislabeling one class for another. Given K patterns with class labels $L_{1:K}$ and class decisions $\Theta_{1:K}$, the classifier confuses class i with class j if patterns labeled in class i obtain decisions of class j. This may be defined as:

$$b(i|j) = \frac{\sum_{\mathrm{k}}^{K}(\mathrm{L_k} == i \wedge \Theta_{\mathrm{k}} == j)}{\sum_{\mathrm{k}}^{K}(\mathrm{L_k} == i)} \tag{B-11}$$

Given such a definition, the confusion matrix may be defined as:

$$Confusion(i|j) = \begin{bmatrix} b(1,1) & \dots & b(1,N_C) \\ \vdots & \ddots & \vdots \\ b(N_C,1) & \dots & b(N_C,N_C) \end{bmatrix} \tag{B-12}$$

where N_c is the total number of classes.

Appendix C – Random Theory

C.1 Continuous Random Variables with Gaussian Distribution

Let X denote a random variable that can take on any real number in a continuum:

$$X \in \mathbb{R} \quad \text{(C-1)}$$

The probability that X takes on a specific value x over a finite interval, [a, b], of the continuum is defined as:

$$P(x) = p\big(x \in (a,b)\big) = \int_a^b p(x)\,dx \quad \text{(C-2)}$$

From the law of total probability:

$$P(X) = P(x_1) + P(x_2) + \ldots + P(x_n) = 1 \quad \text{(C-3)}$$

$$\int_{-\infty}^{\infty} p(x)\,dx = \int_{a_1}^{b_1} p(x_1)\,dx_1 + \int_{a_2}^{b_2} p(x_2)\,dx_2 + \cdots + \int_{a_n}^{b_n} p(x_n)\,dx_n = 1 \quad \text{(C-4)}$$

The expected value ($E[X]$) and the variability (σ) of a continuous random variable may be defined as:

$$\mu = E[X] = \int xp(x)dx \quad \text{(C-5)}$$

$$\sigma = E[(X-\mu)^2] = \int (x-\mu)^2 p(x)dx \quad \text{(C-6)}$$

The probability of random events with Gaussian distribution can be fully characterized by their mean (μ) and variance (σ^2):

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)}{2\sigma^2}}, \qquad p(x) \sim N(\mu, \sigma^2) \quad \text{(C-7)}$$

Therefore, the probability of event $X = x$ being on the interval of x_i is defined as:

$$p(X = x|x_i) = p(x|x_i) = \frac{1}{\sqrt{2\pi\sigma_i}} e^{\frac{-(x-\mu_i)}{2\sigma_i^2}} \quad \text{(C-8)}$$

C.2 Discrete Random Variables with Gaussian Distribution

Let X denote a random variable that can take on a discrete number of *n* values:

$$X \in \{x_1, \dots, x_n\} \tag{C-9}$$

The probability of X being a specific value x_i is defined as:

$$P(x_i) = P(X = x_i) = \frac{the\ number\ of\ events\ were\ X = x_i}{the\ total\ number\ of\ events} \tag{C-10}$$

From the law of total probability:

$$P(X) = P(x_1) + P(x_2) + \dots + P(x_n) = 1 \tag{C-11}$$

The expected value ($E[X]$) and the variability (σ) of a discrete random variable may be defined as:

$$\mu = E[X] = \sum_{i=1}^{n} x_i P(x_i) \tag{C-12}$$

$$\sigma = E[(X - \mu)^2] = \sum_{i=1}^{n} (x_i - \mu) P(x_i) \tag{C-13}$$

Appendix D – Kalman Filter Theory

D.1 Definition

Data can be collected from various measurement devices (sensors, radars, etc.). It is common to filter signals obtained from measurement devices to eliminate noise variance. Smoothing filters will perform temporal statistics to obtain the best signal estimate. Common smoothing filters are the moving mean average filter or the median mean average filter. The Kalman filter (KF) [105] differs from smoothing filters as it not only uses the information from the measurement device, but it also incorporates information known from the system's state evolution through time. In the case of a falling object, for example, a radar measurement may provide the KF with information about its position. Instead of averaging temporal radar measurements to eliminate noise variances in the measurement (as with the case of averaging filters), the KF will supplement the measurement with the equation of falling objects to obtain the best position estimate.

This is done through careful measurement and motion modeling. The KF assumes that the system's state is hidden but a noisy measurement can be obtained. The *measurement model* may be defined as:

$$y_t = Cx_t + d_t \quad , \quad d_t \sim N(0, \delta^2) \tag{D-1}$$

where y_t is the measurement, C is the measurement model matrix, x_t is the system's hidden state, and d is Gaussian noise with variance δ^2 to account for any inaccuracy in the measurement. A system's *motion model* through time could be defined:

$$x_t = Ax_{t-1} + e_t \quad , \quad e_t \sim N(0, \sigma_e^2) \tag{D-2}$$

where x_t is the hidden state, *A* is the motion model matrix, and *e* is Gaussian noise with variance σ_e^2 which accounts for variability in the motion model. Although the hidden state is unknown, from the motion model, a *predicted state* is obtained. The predicted state is defined as the expected value (see random theory in Appendix A):

$$E[x_t] = E[Ax_{t-1}+e_t] = \bar{x}_{t-1} + 0 \rightarrow \quad E[x_t] = A\bar{x}_{t-1} \tag{D-3}$$

where $E[x_t]$ is the predicted state which will be denoted $\bar{x}_p$, and $\bar{x}_{t-1}$ is the previous *state estimate*. The KF assumes that the current estimate $\bar{x}_t$ is somewhere between y_t and $A\bar{x}_{t-1}$. In other words, it assumes that the current state estimate is somewhere between the current measurement and the previous estimate. Hence it performs estimate *updates* as follows:

$$\bar{x}_t = A\bar{x}_{t-1} + K(y_t - C(A\bar{x}_{t-1})) \tag{D-4}$$

where *K* is the Kalman gain and the distance between y_t and $A\bar{x}_{t-1}$ (i.e. $y_t - C(A\bar{x}_{t-1})$) is commonly referred to as the innovation Δ. The Kalman gain (*K*) is determined by evaluating the variability in the measurement and motion model. If the measurement model has higher variability then the motion model (i.e. $\delta^2 > \sigma^2$) the estimate should not update towards the measurement and K <0.5, whereas if the motion model has higher variability then the measurement model (i.e. $\sigma^2 > \delta^2$) the estimate should update towards the measurement and K > 0.5.

It is not uncommon for the measurement and the motion to be in different space (for example one may be in rectangular coordinates while the other may be in triangular coordinates). In such a case, the measurement model matrix *C* "rotates" the motion model in measurement model space to compute the innovation, and then the Kalman gain "rotates" the innovation into motion model space. This will not be discussed in detail, but

the generalized KF algorithm for single and multiple dimensional problems are listed in Table D-1 and derived in details in [106]:

Table D-1: Linear Kalman filter algorithm summary

	One dimensional state x	D-dimensional state M-dimensional measurement $\vec{\boldsymbol{x}}_{\boldsymbol{t}} = [x_1 \dots x_D]$ $\vec{\boldsymbol{y}}_{\boldsymbol{t}} = [y_1 \dots y_M]$
Modeling Motion Model:	$x_t = A_t x_{t-1} + e_t \quad e_t \sim N(0, \sigma^2)$	$\vec{\boldsymbol{x}}_t = A_t \vec{\boldsymbol{x}}_{t-1} + e_t \quad e \sim N(0, R_t)$
Measurement Model:	$y_t = C_t x_t + d_t \quad d \sim N(0, \delta^2)$	$\boldsymbol{y}_t = C_t \boldsymbol{x}_t + d_t \quad d \sim N(0, Q_t)$
Step 1 Prediction update:	$\bar{x}_p = A_t \bar{x}_{t-1}$ $\bar{\sigma}_p^2 = A_t^2 \bar{\sigma}_{t-1}^2 + \sigma^2$	$\vec{\bar{\boldsymbol{x}}}_p = A_t \vec{\bar{\boldsymbol{x}}}_{t-1}$ $\bar{\boldsymbol{\Sigma}}_{\boldsymbol{p}} = A_t \bar{\boldsymbol{\Sigma}}_{t-1} A_t{}^T + R_t$
Innovation:	$\Delta = Y_t - C_t \bar{x}_p$	$\Delta = Y_t - C_t \vec{\bar{\boldsymbol{x}}}_p$
Kalman gain:	$K_t = \bar{\sigma}_p^2 C_t (C_t^2 \bar{\sigma}_p^2 + \delta^2)^{-1}$	$K_t = \bar{\boldsymbol{\Sigma}}_t C_t^T (C_t \bar{\boldsymbol{\Sigma}}_t C_t^T + Q_t)^{-1}$
Step 2 Estimate update:	$\bar{x}_t = \bar{x}_{t-1} + K_t(\Delta)$ $\bar{\sigma}_t^2 = (1 - K_t C_t)\bar{\sigma}_p^2$	$\vec{\bar{\boldsymbol{x}}}_t = \vec{\bar{\boldsymbol{x}}}_{t-1} + K_t(\Delta)$ $\bar{\boldsymbol{\Sigma}}_t = (I - K_t C_t)\bar{\boldsymbol{\Sigma}}_p$
Generalized equation:	$\bar{x}_t = \bar{x}_{t-1} + K_t(Y_t - C_t(A_t \bar{x}_{t-1}))$ $\bar{\sigma}_t^2 = (1 - K_t C_t)(A_t^2 \bar{\sigma}_{t-1} + \sigma^2)$	$\vec{\bar{\boldsymbol{x}}}_t = \vec{\bar{\boldsymbol{x}}}_{t-1} + K_t(Y_t - C_t(A_t \vec{\bar{\boldsymbol{x}}}_{t-1}))$ $\bar{\boldsymbol{\Sigma}}_t = (I - K_t C_t)(A_t \bar{\boldsymbol{\Sigma}}_{t-1} A_t{}^T + R_t)$

$A \equiv Motion\ model\ variable$
$\sigma^2 \equiv Motion\ noise\ variance$
$C \equiv Measurement\ model\ variable$
$\delta^2 \equiv Measurement\ noise\ variance$
$K \equiv Kalman\ gain\ variable$
$\Delta \equiv Innovation$
$\bar{x}_p \equiv E[x]$

$A \equiv Motion\ model$ MxD $matrix$
$R \equiv Motion\ noise\ covariance$
$C \equiv Measurement$ DxM $model\ matrix$
$Q \equiv Measurement\ noise\ covariance$
$I \equiv identity\ matrix$
$K \equiv Kalman\ gain$ MxD $matrix$

Note that the initial conditions about the estimate $\bar{x}_{t=0}$ must be known or the KF may diverge and/or be unstable. The algorithm is the same for multi-dimensional problems or problems where the motion and measurement models are in different spaces. The Kalman

filter is quite simple; however, the motion and measurement models can be difficult to design and they are different from one system to the next. The design difficulties are often due to the inability of obtaining a good estimate of the noise variance (σ^2 and δ^2) or covariance matrices (*R* and *Q*). The KF can be highly unstable but this is mainly due to poor or inadequate modeling.

D.2 Example of a Falling Object

It is well known that the height of a free falling object obeys the following equation:

$$h = h_0 - \frac{1}{2} g t^2 \qquad \text{(D-5)}$$

where h is the height of the projectile, h_o is the height of the projectile at time $t=0$, g = 9.8 m/s is the constant of gravity, and t it the time the projectile has traveled. The state evolution of the object's height through time could be modeled as:

$$x_t = Ax_{t-1} + u_t + e_t = x_{t-1} - \frac{1}{2} g(\Delta)^2 + e_t \quad , \quad e_t \sim N(0, \sigma_e^2) \qquad \text{(D-6)}$$

where A=1, x_t is the hidden high of the object, u_t is the system's input which accounts for the gravitational force, and let Δt = 0.1 s be the time elapsed between time t and $t-1$. Let the motion model variance be constant and σ^2 = 0.0025 m^2.

Assuming that a radar can provide the position of the object with some accuracy, than the measurement can be related to the height of the projectile using the following equation:

$$y_t = Cx_t + d_t = x_t + d_t, \quad d_t \sim N(0, \sigma_d^2) \qquad \text{(D-7)}$$

where C=1, y_t is the radar reading, x_t is the unknown high (i.e. hidden state) of the projectile, and d is Gaussian noise with variance δ^2 = 0.0016 m^2 to account for any inaccuracy in the radar.

As a result, the KF algorithm can be modeled as:

$$\begin{gathered} Prediction: \ \bar{x}_p = \bar{x}_{t-1} - \frac{1}{2}9.8(0.1)^2 \ , \ \ \bar{\sigma}_p^2 = \ \sigma_{t-1}^2 + 0.0025 \\ Update: \ \bar{x}_t = \bar{x}_{t-1} + K_t\left(y_t - \bar{x}_p\right), \quad \sigma_t^2 = (1 - K_t)\bar{\sigma}_p^2 \end{gathered} \tag{D-8}$$

where the Kalman gain is defined as:

$$K_t = \frac{\sigma^2}{\sigma^2 + \delta^2} = \frac{.0025}{(.0025 + .0016)} \tag{D-9}$$

Let the initial position of the projectile be x_o = 1.5 m from the ground with an initial variance of σ^2 = 0.02 m^2. The prediction, measurement, and estimate for the first 1.4 s of the falling object are plotted in Figure D-1.

For every time step, the estimate fell between the prediction and the measurement. Figure D-2 illustrates the estimate update process for t = 0.6.

Multiplying the prediction and measurement probabilities result in an estimate that peaks where the probability distribution of the prediction and measurement overlap and drop when either the prediction or measurement distribution is low.

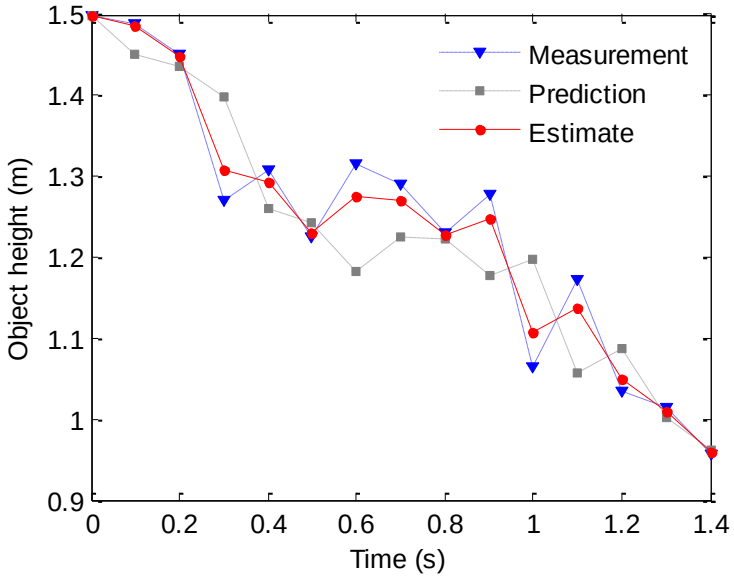

Figure D-1: Measurement y_t, prediction $\bar{x}_p$, and estimate $\bar{x}_t$ of a falling object for its first 1.4 s. The estimate falls between the prediction and the measurement.

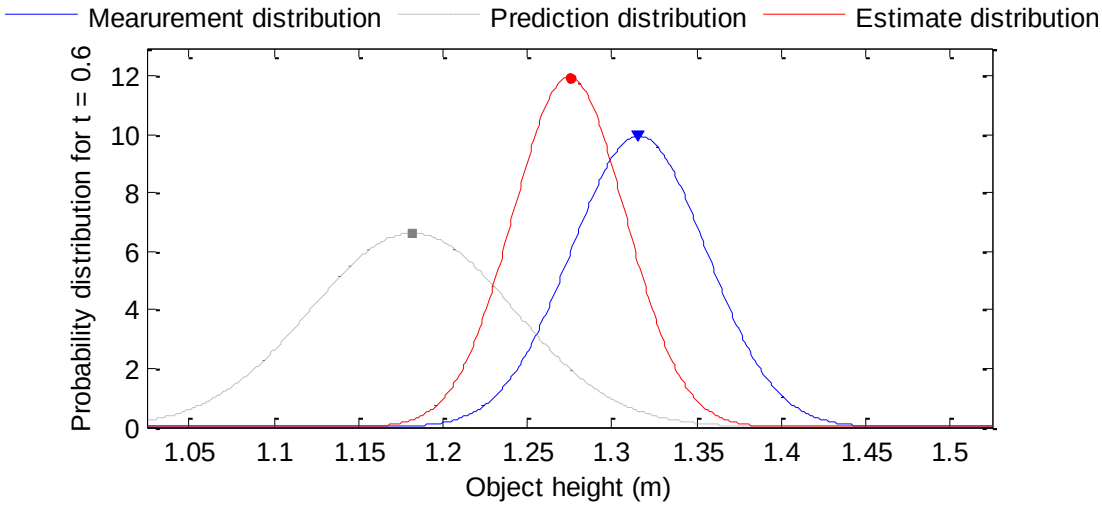

Figure D-2: Estimate update of the falling object for *t = 0.6*.

D.3 Kalman Filter Assumptions

Note the assumptions that were made about the system:

1) The motion model was linear with additive Gaussian noise:

$$x_t = Ax_{t-1} + e_t \quad , \quad e_t \sim N(0, \sigma_e^2) \tag{D-10}$$

2) The measurement model was also linear with additive Gaussian noise:

$$y_t = Cx_t + d_t \quad , \quad d_t \sim N(0, \delta^2) \tag{D-11}$$

3) The initial condition was normally distributed and known. Since the current state is dependent on the previous state, a normally distributed initial condition insures that the states at any time $t > t_0$ will be normally distributed. If the initial conditions are unknown, the KF may be unstable as the system may never converge towards a desirable state.

To apply a KF to a system, these conditions must be true [106] [126]. However, often time the first and second conditions can be relaxed; that is the noise may not always be additive Gaussian as long as the confidence in the motion and measurement models can be defined. Also the system does not need to be linear. In such case, an Extended Kalman Filter (EKF) should be used. For a detailed explanation of this algorithm please refer to Thrun *et al.* [106], however it is not required for this work.

Appendix E – KF Feature Conditioning

E.1 Mathematical Derivation

All the feature conditioning methods are similar in the sense that they all require a motion model, a measurement model, a prediction, and an estimate update (see Appendix D for KF theory). However, they differ on how their measurement model variances are designed and computed. It is also worth mentioning that, unlike Kalman filters that attempt to obtain an estimate of a system's true state (see Appendix D for Kalman filter theory), the feature conditioning methods aim at obtaining compliant features that would allow the system to operate with less variability while in steady-state and better behavior while in class transitions. These compliant features are unknown; however, the feature conditioning methods attempt to obtain the best estimate for these features. Motion and measurement model

The motion model makes assumptions about the features based on their evolution in time. Features are extracted from data segments of processed EMG with fixed window frame length and increment and it is assumed that the EMG within a window is stationary (i.e. time independent). Therefore, given small window frame increments, the features were assumed stationary:

$$\vec{\boldsymbol{x}}_t = \vec{\boldsymbol{x}}_{t-1} = \vec{\boldsymbol{x}}_{t-2} = \cdots = \vec{\boldsymbol{x}}_o \tag{E-1}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} = \boldsymbol{\Sigma}_{t-2} = \cdots = \boldsymbol{\Sigma}_o \tag{E-2}$$

where $\vec{\boldsymbol{x}}_t$ is the current feature set vector, $\vec{\boldsymbol{x}}_{t-1} \ldots \vec{\boldsymbol{x}}_o$ are all the previous feature sets and $\boldsymbol{\Sigma}_t, \boldsymbol{\Sigma}_{t-1} \ldots = \boldsymbol{\Sigma}_o$ are the features covariance. However, the EMG is non-stationary therefore the compliant features and their covariance may not be stationary. The features time evolution can be modeled as:

$$\vec{\boldsymbol{x}}_t = \vec{\boldsymbol{x}}_{t-1} + \boldsymbol{e}_t\,, \quad \boldsymbol{e}_t \sim N(0, \boldsymbol{R}_t) \tag{E-3}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_{t-1} + R_t \tag{E-4}$$

were *e* is Gaussian noise with covariance $\boldsymbol{R}_t$.

The measurement model makes assumptions about the compliant features given an observable state. In current pattern recognition systems, class decisions are obtained using an LDA classifier (Appendix B) which assumes that the classes have Gaussian distribution with trained mean and covariance. The user observes the class decisions; this observation is the measurement. Given a class decision, from Random theory (Appendix C), the features position is most likely at the class means:

$$\boldsymbol{E}[\vec{\boldsymbol{y}}_t] = \boldsymbol{E}[\vec{\boldsymbol{x}}_t] = \overrightarrow{\boldsymbol{m}}_t \tag{E-5}$$

where $\overrightarrow{\boldsymbol{m}}$ is the class mean (Appendix B). Even though the user cannot observe variability, in feature space there is some variability and the compliant features may not always fall at the class means. There is some uncertainly to the measurement; therefore, the measurement model can be defined as:

$$\vec{\boldsymbol{y}}_t = \vec{\boldsymbol{x}}_t + \boldsymbol{d}_t \;\;,\;\; \boldsymbol{d}_t \sim N(0, \boldsymbol{Q}_t) \tag{E-6}$$

where *d* is Gaussian noise with covariance $\boldsymbol{Q}_t$ to account for any feature variability or uncertainty in the class decision.

The motion and measurement models results in the following generalized KF equation for multi-dimensional problems (see Table D-1)[3]:

$$K_t = (\overline{\boldsymbol{\Sigma}}_{t-1} + R_t) * ((\overline{\boldsymbol{\Sigma}}_{t-1} + R_t) + Q_t)^{-1} \tag{E-7}$$

[3] The motion and measurement model matrices (*A* and *C* respectively) were designed as identity matrices and have no effect on the KF algorithm.

$$\vec{\bar{x}}_t = \vec{\bar{x}}_{t-1} + K_t(y_t - \vec{\bar{x}}_{t-1}) \quad \text{(E-8)}$$

$$\bar{\Sigma}_t = (I - K_t)(\bar{\Sigma}_{t-1} + R_t) \quad \text{(E-9)}$$

where $\vec{\bar{x}}_t$ and $\bar{\Sigma}_t$ are the feature estimates and covariance estimates (i.e. the KF outputs). From these equations, it is noticeable that the current estimate $\vec{\bar{x}}_t$ is somewhere between the previous estimate and the current class decision mean (i.e. $\vec{y}_t = \vec{m}$). If the current class decision is more confident than the previous estimate (i.e. $Q_t < R_t$), the current estimate shifts towards the current class mean otherwise the feature estimate is closer to the previous estimate. In other words, the feature estimates are constant until a confident class decision change occurs. The challenge to Kalman filter design is to determine how to model the motion and measurement model variance (R_t and Q_t respectively).

E.1.1 Motion Model Variance

The motion and measurement model noise are often assumed Gaussian, but this assumption can be relaxed. In fact, the motion and measurement model variance are a confidence measure (see Appendix D on Kalman filter theory); the model with less variance is most confident. This is especially true for feature conditioning as the goal is to find the most compliant features.

The goal is to determine if the compliant features from time *t-1* to *t* are most likely to obey the motion model (i.e. be constant) or the measurement model (i.e. the users intent is represented by the class decision). The motion model variance is assumed proportional to the extracted features variability. In other words, the compliant features are assumed to be hidden behind the extracted features variance. Therefore the $\boldsymbol{R}_t$ can be estimated by computing the covariance of the *extracted features* $\vec{\hat{x}}$ at time *t* and *t-1*:

$$\boldsymbol{R}_t = cov(\vec{\hat{\boldsymbol{x}}}_{\boldsymbol{t}}, \vec{\hat{\boldsymbol{x}}}_{\boldsymbol{t-1}}) \tag{E-10}$$

Since features with high variability are characteristic to contraction changes, such a definition of the model variability would results in low confidence in the motion model while users perform non steady-state contractions. However, this definition may result in singular or badly scaled covariance matrix due to the small sample size used to compute the covariance. A singular or badly scaled matrix is not invertible or has a highly unstable inverted matrix. The covariance must be invertible in order to compute the Kalman gain (Equation (E-10)). Various covariance estimation methods to obtain invertible matrices have been suggested [127], in this work the diagonal covariance estimator method is used. This method prevents the KF gain from being unstable due to singular or badly scaled matrices, and minimizes computational complexity. The motion model variance is computed as follows:

$$\begin{aligned} \boldsymbol{R}_t &= \begin{bmatrix} cov(\hat{x}_{1,t}, \hat{x}_{1,t-1}) & 0 & \dots \\ 0 & \ddots & 0 \\ \vdots & 0 & cov(\hat{x}_{P,t}, \hat{x}_{P,t-1}) \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2 & 0 & \dots \\ 0 & \ddots & 0 \\ \vdots & 0 & \sigma_P^2 \end{bmatrix}, P: \#\ of\ dimensions\ in\ \boldsymbol{x} \end{aligned} \tag{E-11}$$

Although the extracted features are a random variable, the equation defined above is computed on two samples of extracted features; therefore, it can be simplified as the distance between the extracted features taken at time *t* and *t-1*:

$$\boldsymbol{R}_t = \begin{bmatrix} (\hat{x}_{1,t} - \hat{x}_{1,t-1})^2 & (\hat{x}_{2,t} - \hat{x}_{2,t-1})^2 & \dots & (\hat{x}_{p,t} - \hat{x}_{p,t-1})^2 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix} \tag{E-12}$$

E.1.2 Measurement Model Variance

For a given subject, it is possible to evaluate the probability of the class decision i to be true given the previous class decision j. This is denoted as:

$$p(i|j) \tag{E-13}$$

The $p(i|j)$ can be computed by evaluating the LDA confusion matrix (Appendix B) obtained from training samples. A highly confused class will have low probability of being true. Such information can be used to determine if the measurement is confident; if the selected class at time t has low probability, the measurement should have low confidence (i.e. high variance), whereas if the selected class has high probability the measurement should have high confidence (i.e. low variance). The measurement confidence (i.e. variance) cannot be directly proportional to the class probability because, in this work, the motion and measurement model variances must be computed on the same range, otherwise this could result in a system that is bias or unstable; the Kalman gain cannot compute the ratio between the two models' variance without biasing the model with variance computed on higher ranges. Also, a zero variance could make the Kalman gain unstable. To prevent the KF from being bias or unstable, the probability of observing a class was mapped to values of variance that ranged from values close to zero up to one (the same range as the motion model). The mapping function was built to allow decisions which obtained high probabilities (>80%) to obtain a variance close to zero, whereas decisions with low probabilities (<15%) were mapped to obtain variance values close to one. The mapping function was an algebraic function with horizontal asymptotes at values of 1 and zero, and with 0.50 variance when the class probability was 50%.

The mapping function between the probability $p(i|j)$ and the variance $q_{i,j}$ is shown in Figure E-1:

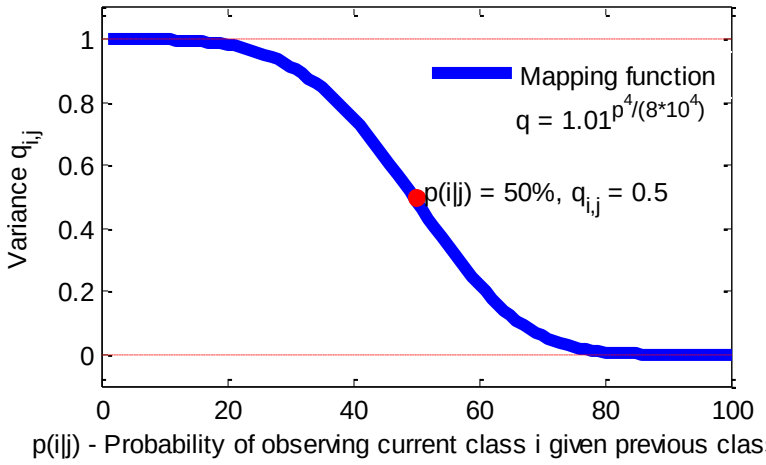

Figure E-1: Mapping of the probability of observing the current class with the previous class to variance values.

Therefore the variance $\boldsymbol{Q_t}$ was defined as the following P by P diagonal matrix (where P is the total number features):

$$\boldsymbol{Q_t} = \begin{bmatrix} q_{i,j} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q_{i,j} \end{bmatrix}, \qquad \begin{matrix} i: Selected\ class\ at\ time\ t \\ j: Selected\ class\ at\ time\ t-1 \end{matrix} \tag{E-14}$$

The measurement is confident if the LDA class decision at time t is not confused with the previous LDA class decision at time *t-1*.

Another measurement model variance was evaluated; however, it was found to be less effective as the former. It is worth mentioning this method although it was not presented in this work. This model assumed that the measurement model variance was proportional to the extracted features distance from the class means; therefore, the closer the extracted features fall to the class mean, the more confident the measurement model. Hence the covariance $\boldsymbol{Q}_t$ was modeled to capture the discrepancies between the class center and the extracted features' position. This was done by computing the covariance between the extracted features $\vec{\hat{\boldsymbol{x}}}$ at time t and the mean of the selected class (or measurement) at time t. However, as seen previously this would results in a singular or

badly scaled matrix due to the small sample size used to compute the covariance. It was assumed that, in such a case, the covariance can be generalized to its diagonal form. Hence:

$$\boldsymbol{Q_t} = \begin{bmatrix} var(y_{1,t}, \hat{x}_{1,t}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & var(y_{p,t}, \hat{x}_{1,t}) \end{bmatrix}, \quad \begin{matrix} P: \#\ of\ dimensions\ in\ \boldsymbol{y} \\ i: Selected\ class \end{matrix} \tag{E-15}$$

Such a definition can be simplified as the distance between the extracted features and the class decision mean.

E.1.3 Initial State

It is important to have a well-defined prior (i.e. initial state) or the KF may never converge towards the system's true state (Appendix D). Assuming users begin in the rest class, the initial state was defined as:

$$\vec{\boldsymbol{x}}_o = \vec{\boldsymbol{m}}_{\boldsymbol{rest}} \tag{E-16}$$

It is best to under estimate the initial state covariance. Setting a low initial covariance allows the system to choose if the motion or measurement is more confident, whereas setting a high initial covariance can result in a system that is bias towards the measurement which is undesirable. Unfortunately, setting the initial covariance to zero could result in an unstable Kalman filter as the Kalman gain cannot handle small numbers due to the matrix inversion. It was found, through empirical, observations that the initial covariance could be defined as:

$$\bar{\boldsymbol{\Sigma}}_{\boldsymbol{0}} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{E-17}$$

E.1.4 Conservative Updates

A change in the output can either be due to high variability in the extracted features or a poor measurement (i.e. misclassification). It was assumed that features were temporally close (i.e. $\boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{e}_t$), therefore from time *t-1* to time *t* the change in the features should be small, meaning that it should not be large enough to change the LDA output. This assumption also holds while users perform class transition where the features' variance may increase, but since the time differential is small this increase in variance should also be small. If the variability in the features creates a change in the LDA output from time *t-1* to *t*, this variability may mainly be due to noise and the estimate should not update toward the current class decision mean. To prevent the estimate from updating towards the measurement, the latter's confidence must be low. In other words, both the motion and measurement model confidence should be low. Unfortunately, from Equation D-9, the KF gain shifts the estimate towards the most confident even though both have low confidence.

It is possible to account for such situations given the following KF algorithm:

$$K_t = (\overline{\boldsymbol{\Sigma}}_{t-1} + R_t) * ((\overline{\boldsymbol{\Sigma}}_{t-1} + R_t) + Q_t)^{-1} \tag{E-18}$$

$$\overline{\boldsymbol{x}}_t = \overline{\boldsymbol{x}}_{t-1} + \beta K_t (y_t - \overline{\boldsymbol{x}}_{t-1}) \tag{E-19}$$

$$\overline{\boldsymbol{\Sigma}}_t = (I - K_t)(\overline{\boldsymbol{\Sigma}}_{t-1} + R_t) \tag{E-20}$$

where β is a Boolean operator which is null if the LDA output at time *t* is different from the LDA output at time *t-1*.

E.1.5 Interpretation

The motion model determines the position of the features based on all previous information, while the measurement gives information about the most probable position

of the current features. The KFs allow the features' estimate to change and migrate toward the mean of the current class if users vary their contraction level and if the current features have fallen near the center of the selected class.

When the LDA output changes from time *t-1* to *t* , the conservative KFs assumed that the variability in the data is not natural and therefore mainly noise. In such case, neither the motion nor measurement models are robust making it impossible to update the estimate with certainty. This system is able to track changes in the EMG features by allowing feature variability during class transitions, is able to minimize variability during steady-state class, and it is able to minimize unwanted or unreasonable changes in the features.

E.2 Simple and Kalman Filter Feature Conditioning Example

A simple (S(td,a)) and Kalman filter (KF(prob,td,a)) feature conditioning methods were applied to 6 window frames of simulated data. The feature estimate and measurement while using the S(td,a) and KF(prob,td,a) results are shown in Figure E-2 and Figure E-3 respectively:

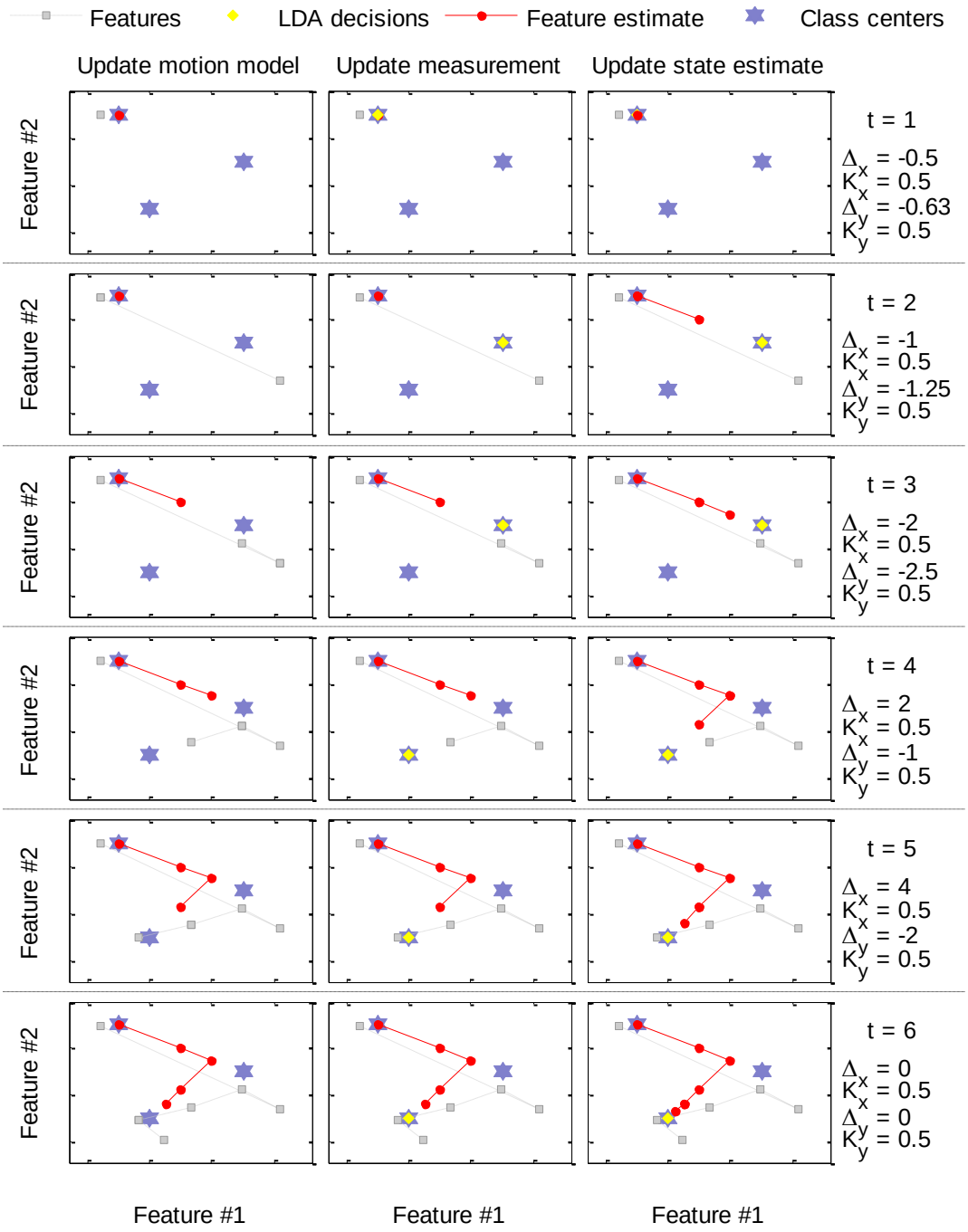

Figure E-2: Iterative example for the simple aggressive methods (S(td,a))

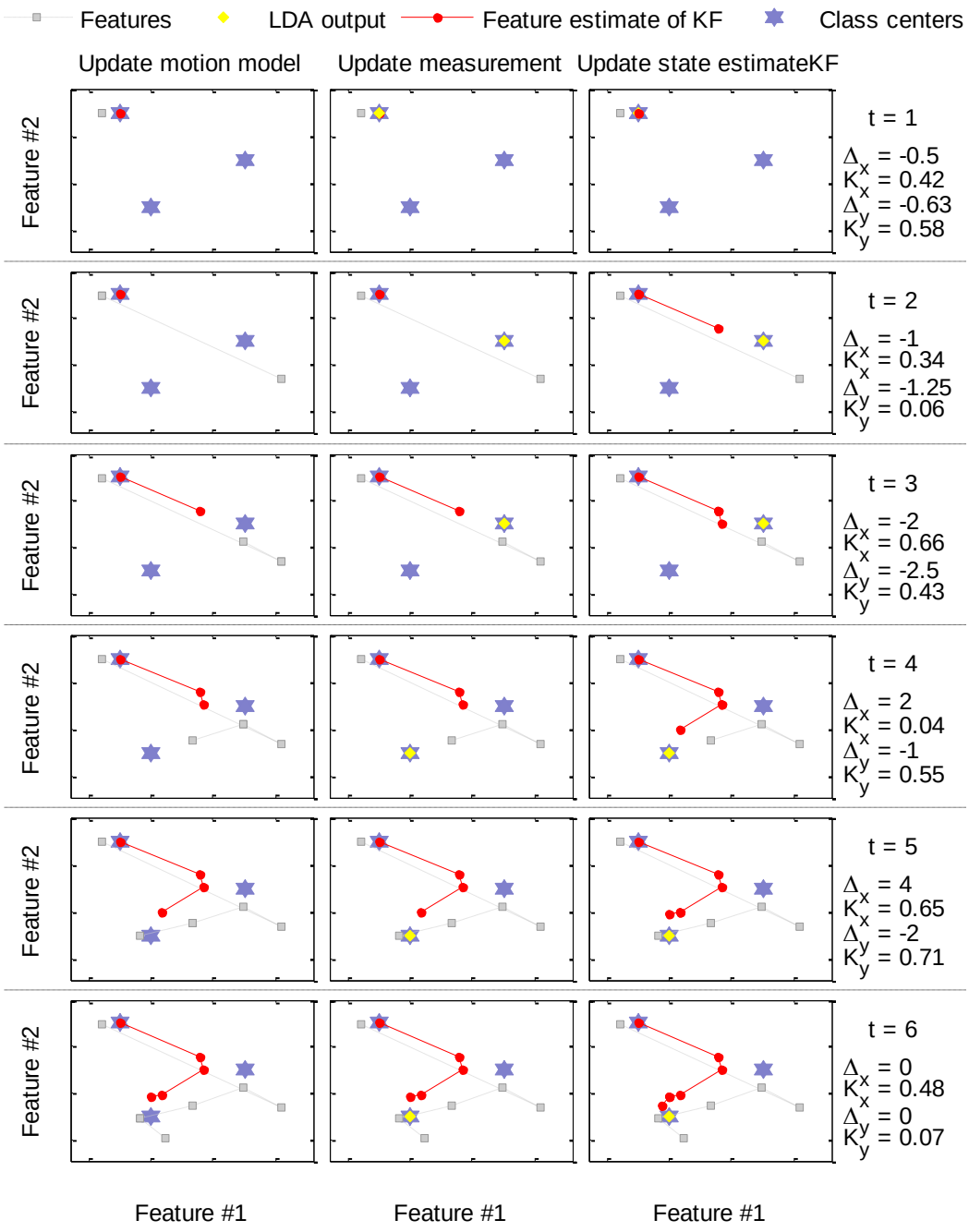

Figure E-3: Iterative example for the KF aggressive methods with td features (KF(prob,ed,a))

Appendix F – Evaluation of the Number of Consecutive Windows

A transition was considered failed if subjects were unable to sustain a prompted class for a specified *N* number of consecutive windows. The choice of *N* has a direct impact on the FC rate. For each 45 trials (3 trials x 15 subjects), the failed completion rate (FC rate as defined in Section 3.3.4.1) was computed on the class decision obtained in Section 3.3.3 for *N* values ranging between 1 and 600 (i.e. maximum number of windows in a single 3 second prompted contraction).

Figure F-1 illustrates the average FC rate obtained across the trials using different feature conditioning methods (and without features conditioning) for increasing values of *N*. A single plot illustrates the results obtained using a specific feature conditioning method (or no-conditioning) which is listed in the table's title. On each plot, various curves represent the results obtained using the different frame lengths (FL). The results show that as N increases, the failed completion rate increases as well. This is due to the fact that it is more difficult for users to select an increasing number of consecutive windows. The results also show that for a given value of N > 30, the failed completion rate is lower as the FL increases. This can be explained by the fact that increasing the FL reduces feature variability; fewer feature variability should facilitate the users' ability to sustain a given contraction type. For smaller values of N (<30), results are comparable across FL (in most cases except for the configuration for S(td,a) or No-cond(td)). Since it requires at least $(5 * N + FL)$ ms of time for users to correctly select a contraction for the transition to be completed, these results suggest that for short sustaining times, the effects of increasing the number of windows N is not significant. Finally, the results show that for N > 30, the failed completion rate increases beyond 20%. Such a high failed

completion rate may be explained by the fact that the data collection was highly dynamic making it difficult for users to complete tasks.

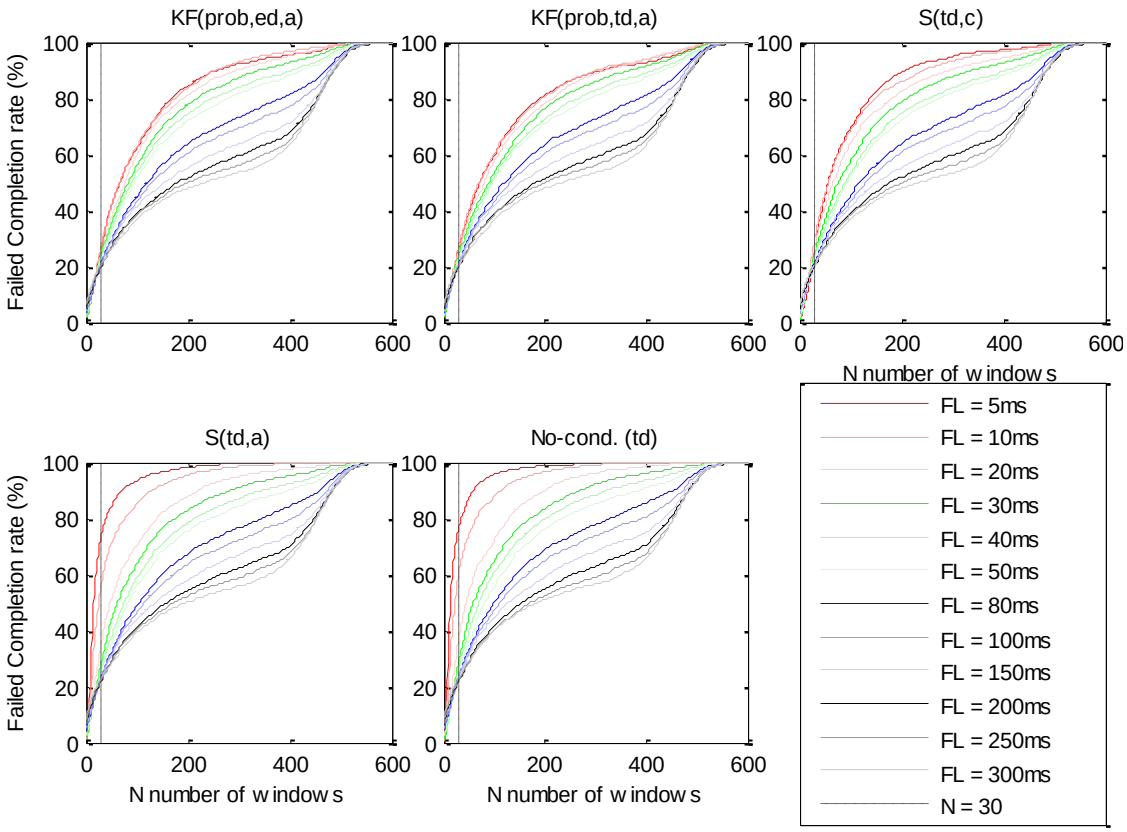

Figure F-1: Effects of the number of consecutive windows on the failed completion rate. Each plot represents a different feature conditioning (or no-conditioning) method. The different frame lengths (FL) are represented by different curves on the plot.

In this work *N* was chosen to be 30 because:

1) The effects of increasing the FL on the FC rate are minimized which facilitates evaluating the effects of feature conditioning on the results, and;
2) The FC rate is high for $N > 30$.

Appendix G – State-Evaluator Performance Evaluation

G.1 Assessment Metrics

Different statistical measures are derived from the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) of the outputs obtained using binary configurations. In this work, for example, the state-evaluator's ability to detect class transitions is of interest. It is anticipated that the features are in indeterminate state while in class transitions. Since the state-evaluator is positive(1) when detecting features that are in an indeterminate state, and negative(0) when in steady state class, the definition of TP, TN, FP and FN would be defined as in Table G-1:

Table G-1: Definition of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) to assess the performance of the state-evaluator.

	Definition
TP:	The output of the state-evaluator is positive(1) and the user is in class transition
TN:	The output of the state-evaluator is negative(0) and the user is in steady-state class
FP:	The output of the state-evaluator is positive(1) and the user is in steady-state class
FN:	The output of the state-evaluator is negative(0) and the user is in class transition

Class transitions and steady state frames were segmented and labeled visually as in Appendix H.

From these definitions, the *accuracy*, *precision*, *sensitivity*, and *specificity* [112] [113] [114] can be measured.

Accuracy measures the binary configurations' ability to correctly identify each frame. It is defined as the percentage of correct outputs over all outputs[4]:

$$accuracy = \frac{\#\ of\ correct\ outputs}{\#\ of\ outputs} = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN} \qquad (G\text{-}1)$$

[4] THe $\sum(\cdot)$ notation indicates the total number of TP, TN, FP or FN over an entire trial.

Precision measures the binary configuration's ability to repeat the same result frame after frame. It is computed as the ratio of correct positive(1) outputs overall positive(1) outputs:

$$precision = \frac{\#\ of\ correct\ positive\ outputs}{\#\ of\ positive\ outputs} = \frac{\sum TP}{\sum TP + \sum FP} \qquad (G\text{-}2)$$

Sensitivity measures the binary configurations' ability to correctly identify frames that were labeled in class transitions. It is the ratio of correct positive(1) outputs overall the number of frames labeled in the class transition:

$$sensitivity = \frac{\#\ of\ correct\ positive\ outputs}{\#\ of\ frames\ labeled\ in\ class\ transition} = \frac{\sum TP}{\sum TP + \sum FN} \qquad (G\text{-}3)$$

Specificity measures the binary configurations' ability to correctly identify the frames that were labeled in steady state class. It is the ratio of correct negative decisions overall the number of class decisions not labeled in the state evaluated:

$$specificity = \frac{\#\ of\ correct\ negative\ decisions}{\#\ of\ frames\ not\ labeled\ in\ state\ evaluated} = \frac{\sum TN}{\sum TN + \sum FP} \qquad (G\text{-}4)$$

G.2 Receiver Operating Characteristics (ROC)

A perfect binary configuration would have 100% sensitivity (i.e. frames labeled as in class transitions were identified as positive(1)) and 100% specificity (i.e. frames labeled in steady state class were identified as negative (0)). However, there is usually a tradeoff between the two; as the error rate increases, either the number of positive (1) output error increases or the number ofnegative (0) output error increases.

The thresholds are optimized by evaluating the Receiver Operating Characteristics (ROC) [113] [114] which shows the tradeoff between true positive rate (i.e. sensitivity) versus false positive rate (i.e. 1-specificity) for varying threshold values. An arbitrary example for threshold coefficients **thrs** = [$thrs_1$ $thrs_2$ … $thrs_{th}$] that obtained true positive rates **TPR** = [TPR_1 TRP_2 … TPR_{th}] and false positive rates **FPR** = [FPR_1 FRP_2 … FPR_{th}] is illustrated in Figure G-1:

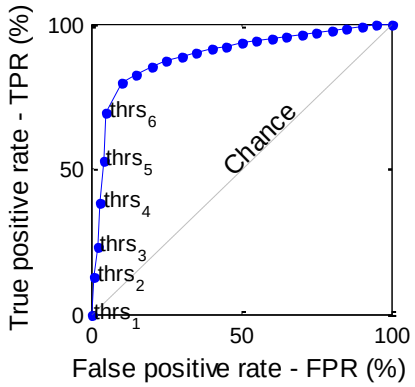

Figure G-1: Arbitrary example of an ROC plot.
Each point on the ROC curve shows the TPR_i and FPR_i obtained using different threshold coefficients $thrs_i$. Any point on the diagonal line from (0,0) to (100, 100) represents chance; there is a 50% chance that the binary configuration identifies a frame as positive or negative.

The threshold is optimized by selecting the point i on the ROC curve which approaches the (0,100) point where the sensitivity and specificity are 100%. This point is determined by maximizing the differential between the **TPR** and the **FPR**:

$$i = \arg \max_{n=1:th} (|TPR(n) - FPR(n)|) \quad \text{(G-5)}$$

where $thrs_i$ is the best threshold. A perfect system would have TPR(i)-FPR(i)=100.

G.3 ROC Plots of all Frame Lengths

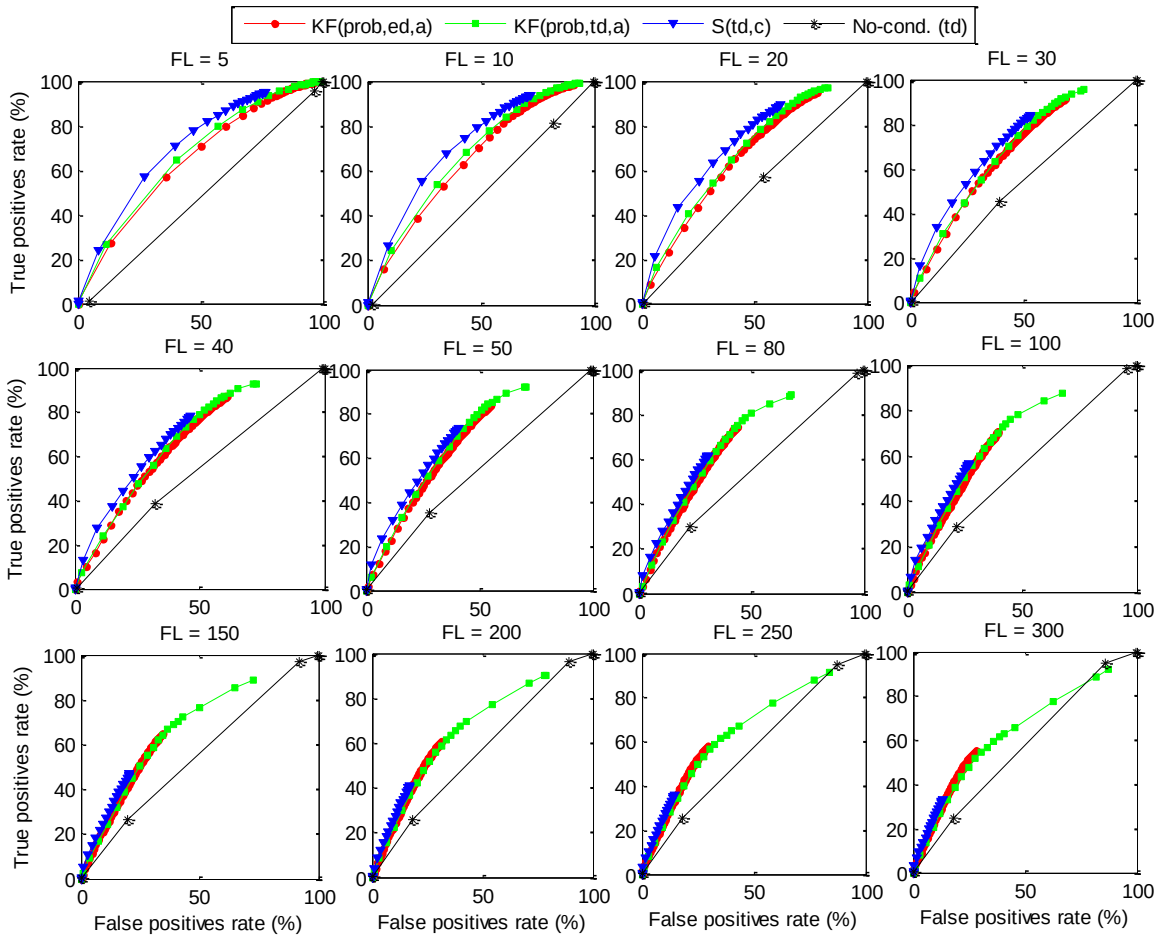

Figure G-2: ROC plots of the state-evaluator using different feature conditioning methods (and without features conditioning) for varying frame length (FL).

G.4 ROC Plots while Detecting Class Changes

When features travel between two trained classes they are in an indeterminate state. This can results in a class change or several class changes in the decision stream. The state-evaluator's ability to detect class changes in the decision stream was evaluated by comparing the output of the state-evaluator to class change labels. These labels were obtained by evaluating the LDA classifier outputs obtained in Section 3.3.2 while using

the conditioned and unconditioned features. The class change labels were positive (1) when a class change occurred in the decision stream. A class change was defined as any instant when:

$$output_{n-1} \neq output_n \quad \text{(G-6)}$$

where *n* is the window of the most current decision. Figure G-3 illustrates the ROC plots of the state-evaluators ability to detect class changes in the decision stream. In these plots, a true positive is defines as the state-evaluator being positive (1) and the class change label being positive (1), whereas a false positive is defined as the state-evaluator being positive (1) and the class change being negative (0).

These plots show that the state-evaluator is capable of detecting class changes in the decision stream when using with feature conditioning. They also show that this is especially true for longer FL. As the FL increases, the amount of class changes in the decision stream caused by high variability in the features is reduced making it easier for the state-evaluator to detect class changes. In fact, Figure G-4 illustrates the percentage of class changes that occur in the decision stream which are due to users performing class transitions. This percentage augments as the FL increases suggesting that for longer FL, class changes are most likely due to class transitions.

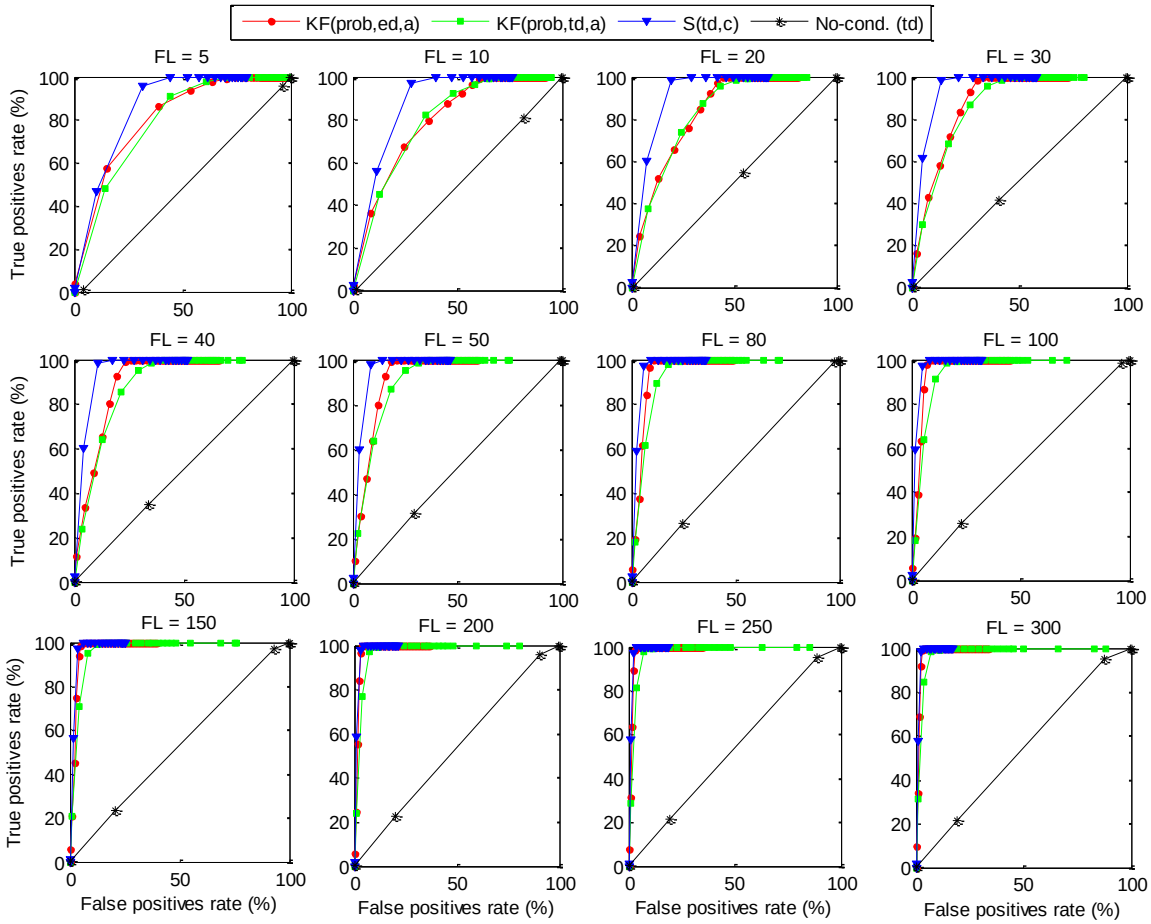

Figure G-3: ROC plots of the state-evaluator while detecting class changes in the decision stream and using different feature conditioning methods (and without features conditioning) for varying frame length (FL).

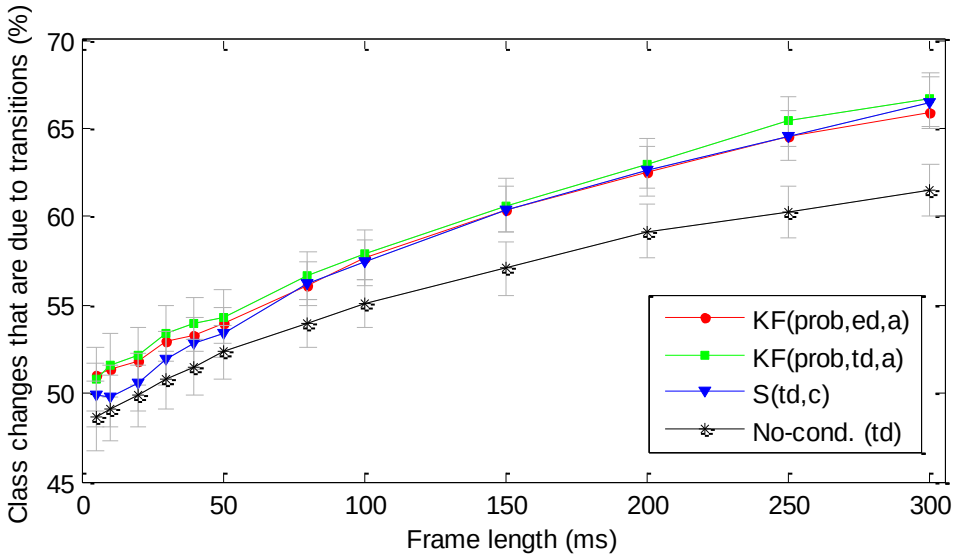

Figure G-4: Percentage of class changes due to class transitions for varying FL.

Appendix H – Visual Segmentation of the Signal

The EMG signal was visually segmented into steady-state contraction, and class transitions. To do so, the raw EMG was notched filtered, band-passed filtered and rectified resulting in a processed EMG. The processed EMG for each channel along with a low-pass filtered processed EMG are illustrated in Figure H-1. The low signal illustrated was obtained by averaging the data obtained from an analysis window of 150 ms frame length and 5 ms frame increments progressing across the signal.

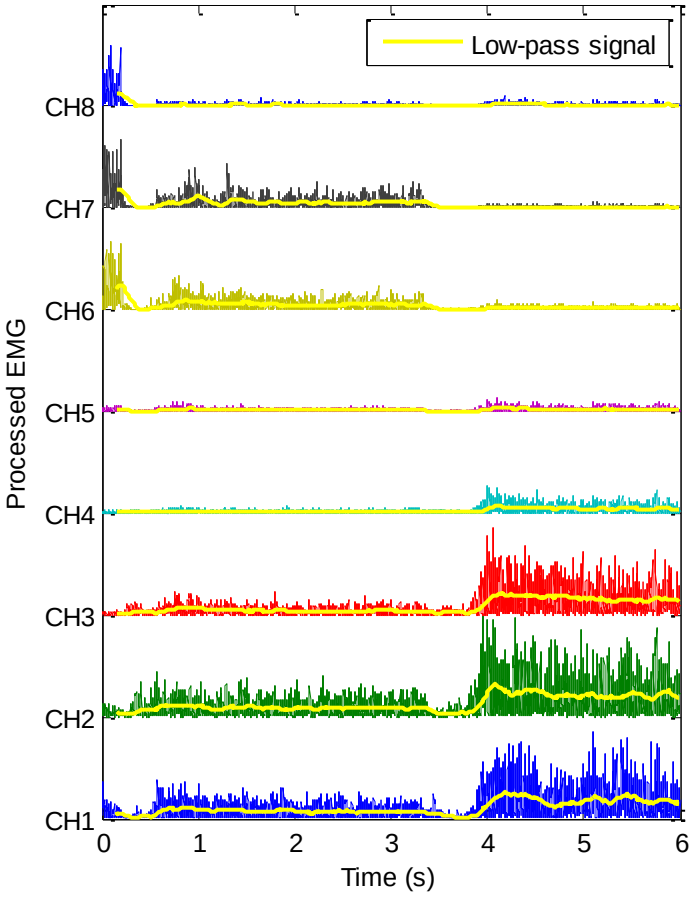

Figure H-1: Processed EMG signal and low-pass filtered signal. To obtain the processed EMG, the raw EMG was notched filtered, band-passed filtered and rectified. The low-pass signal averages 150 ms of data every 5 ms.

The processed EMG of each channel was then passed through a spatial filter by averaging across channels. This signal was used for visual inspection. In Figure H-2, the segments of class transitions are captured in the grey shaded areas. For clarity, the prompted class and the output class are also illustrated in the figure. The prompted class was sampled at the same rate as the data (that is 1000 Hz), whereas the output class were obtained using an LDA to classify TD features extracted from windows of 150 ms frame length and 5 ms frame increment (as in Section 3.3.2). During visual segmentation of the signal, the prompted and output classes were not displayed to avoid biasing the segmentation process. For every class transition, a *time stamp* was obtained for the start and end times. Visual inspection of each data set was done three times. The time stamps were averaged to avoid any inaccuracies in the visual inspection.

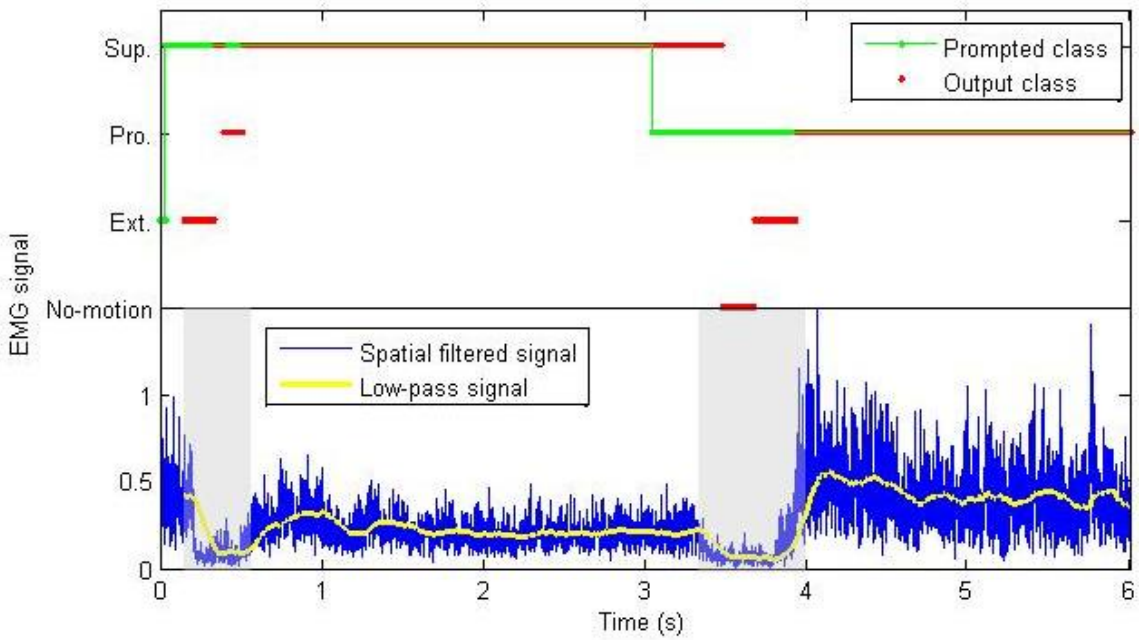

Figure H-2: The upper plot illustrates the prompted class obtained at 1000 Hz frequency, output class obtained from features extracted on windows of 150 ms frame lengths and 5 ms increments. The bottom plot shows the spatial filtered signal, the low-pass signal (with 150 ms windows and 5 ms frame increment), and the segments of class transitions (shaded in grey).

In Figure H-2, it can be observed that at the start and end of class transition segments the low-pass filtered signal is delayed. This is expected because it was obtained using windows of 150 ms of data (i.e. it does not represent instantaneous instances of data in time). The figure also illustrates the user delay; after a class has been prompted it can take some time for the user to begin changing his or her contraction.

Appendix I – State-Based Controls Performance using Class Transition Visual Labels

The state-based classification method with rejection was evaluated using the visual class transition labels to determine the features state. This had for effect of allowing evaluation of the effects of rejecting class transitions on the system performance assuming that all class transitions were detected.

The features conditioned using the KF_{euc} and S_{cons}, and the unconditioned features (as obtained in Section 3.3.2) were classified using the NN_C, the LDA, and the LDA respectively. These were trained as described in Section 4.3.1.1. Class decisions that were labeled as in class transitions were rejected. The mean and standard error of the resulting steady-state A_{CC}, FC rate, and τ_{30} are illustrated in Figure I-1, Figure I-2, and Figure I-3 respectively. A single plot in the figures represents the results obtained using the KF_{euc} (left hand plot), using the S_{cons} (center plot), and using unconditioned features (right hand plot). Because of the high variability in the results between the different contraction types, the average and standard error displayed in these figures were taken across median values computed for every trial. There was 42 contractions performed in a given trial, and a total of 45 trials (15 subjects each performed 3 trials). Therefore, the average and SE illustrated in these figures were taken over 45 values. The grey shaded area represents the results obtained while using the reference configuration; the dotted line is the mean results and the width of the shaded area the standard error. This grey area spans all figures for comparison purposes.

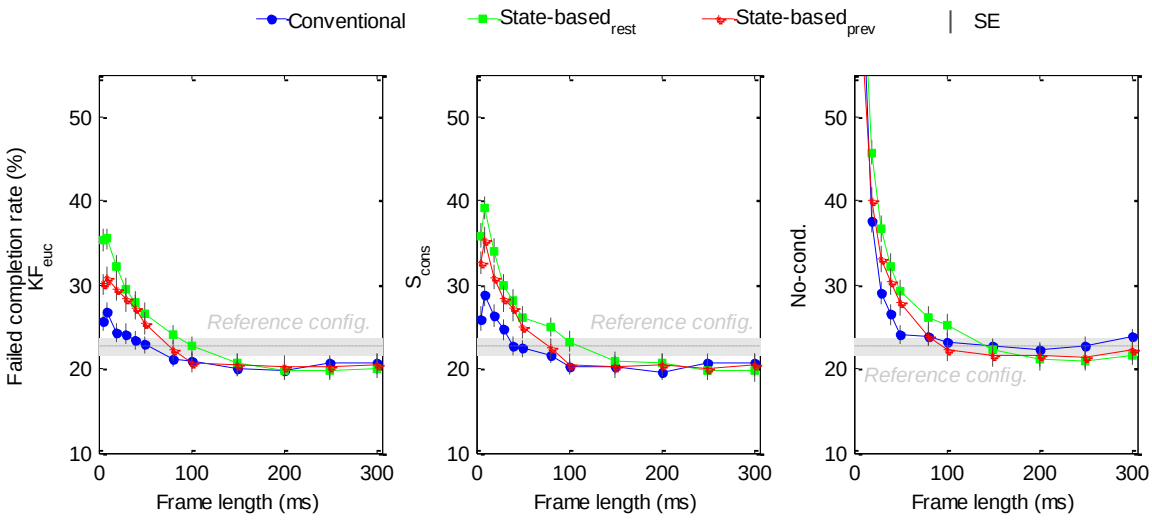

Figure I-1: Classification accuracy obtained using different state-based methods compared to various conventional classification methods. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. Each plot represents different feature conditioning and classification settings listed in on the figures y-label.

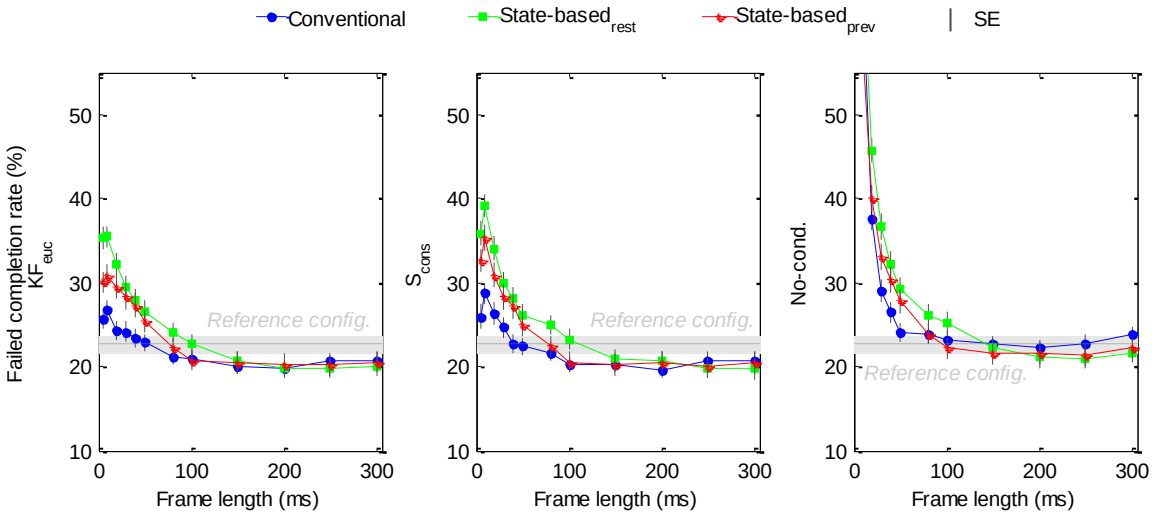

Figure I-2: Failed completion rate obtained using different state-based methods compared to various conventional classification methods. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. Each plot represents different feature conditioning and classification settings listed in on the figures y-label.

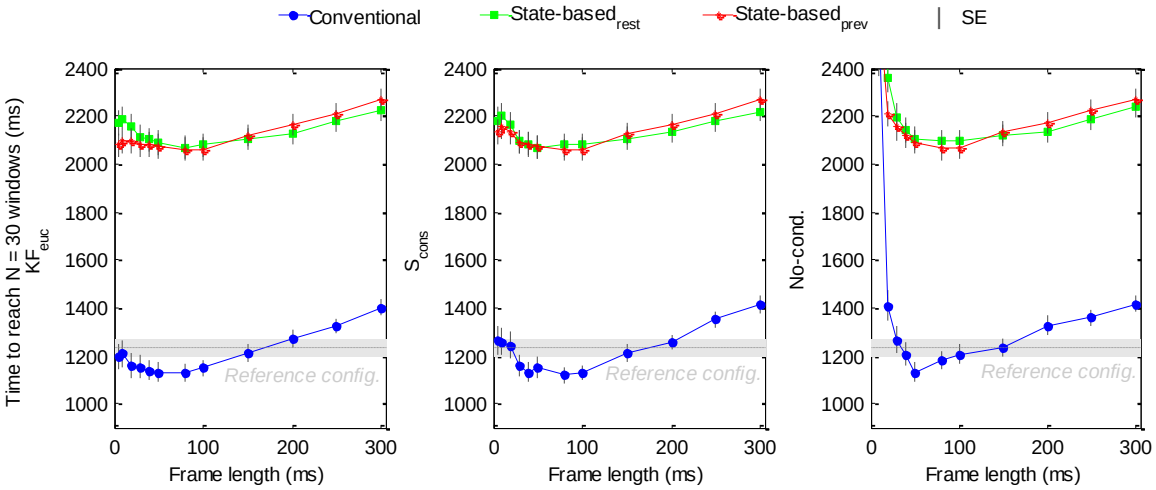

Figure I-3: Time to reach N = 30 consecutive obtained using different state-based methods compared to various conventional classification methods. The state-based method either rejects a decision (state-based$_{rest}$) or stays in the previous (state-based$_{prev}$) class when the state-evaluator detects a class transition. Each plot represents different feature conditioning and classification settings listed in on the figures y-label.

.

These results show significant degradation in steady-state A_{CC} and τ_{30} for all FL regardless of feature conditioning, but no significant changes in FC rate. Such results suggest that rejection causes delay in class transitions. These delays are so large that they degrade classification accuracy. However, such delay did not prevent users from completing the transitions.

Appendix J – Real-Time Pilot Study

J.1 Overview

In Chapter 4, offline analyses were performed to determine which configurations were most likely to provide users with better real-time usability. The following control settings were retained:

Table J-1: Summary of optimal settings during offline analysis.

Type	Feat.	Feature cond.	Classifier	FL
Conv. classification with MV post-processing	**TD**	n/a	LDA	30-40 ms
State-base classification with rejection	**TD**	S_{cons}	LDA	40-150 ms
State-base classification with rejection	**ED**	KF_{euc}	NN_C	40-150 ms

A real-time pilot study was conducted to validate majority voting (MV) as the literature [96] states that MV improves offline performance but this performance improvement is not noticed in real-time. Additionally, the pilot study evaluated the effects of, rather rejecting indeterminate state decisions, slowing down the system's speed which could not be evaluated offline. This is referred to as *state-dependent proportional control*. During this pilot study, empirical observations regarding the best FL to be used amongst the ranges listed in Table J-1 and regarding the use of S_{cons} or KF_{euc} were also performed.

J.2 Methodologies

Nine subjects ranging in age from 23 to 55 with no known motor or sensory impairment participated in the experiment. These were student, staff, and faculty at the Institute of Biomedical Engineering (IBME) at the University of New Brunswick (UNB); therefore, they had some level of experience in EMG pattern recognition controls. All

experiments were approved by the UNB Ethics Review Board under the Surface EMG Data Collection REB #2008-083.

The subjects performed a Fitts' law experiment similar to the one described in Section 5.4.1, however the target locations and the configurations evaluated were different. In fact, unlike the experiment described in Section 5.4.1 which required subjects to perform a single contraction to complete the tasks, the target locations in this experiment required subjects to perform at least two contractions to complete the tasks. Twelve target locations were evaluated; these are illustrated in Figure J-1:

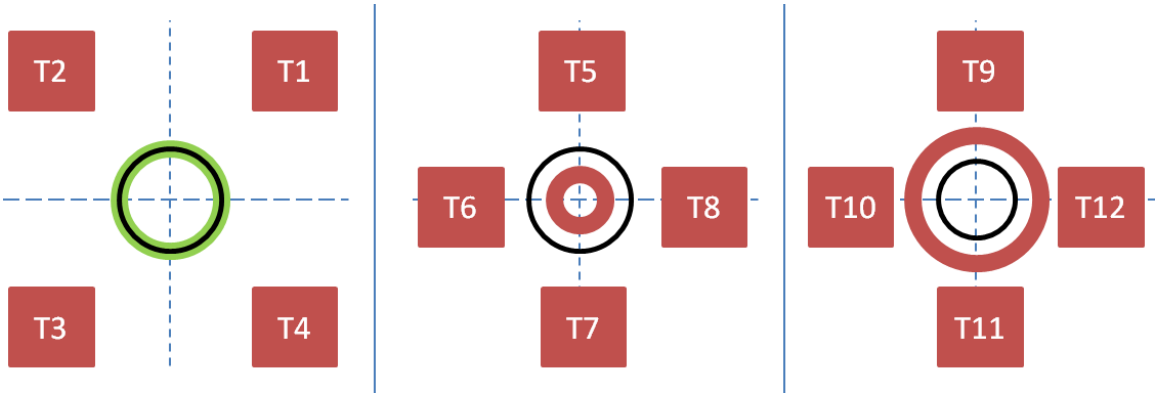

Figure J-1: The 12 possible strategic tasks. For tasks T1 to T4 the target box was in either quadrant, and for tasks T5 to T12 the target circle diameter was larger or smaller than the cursor's diameter and the target box was on either axis. If no errant motions were activated, users could complete tasks T1 to T5 with wrist motions (flexion, extension, supination, and pronation), and T5 to T12 with hand (hand open and close) and wrist motions. In all cases, if no errant motions were activated users could complete the tasks by performing two contractions.

Therefore, subjects performed 48 tasks per trial for a total of 1296 tasks (4 ID x 12 positions x 9 subjects x 3 trials).

The configurations evaluated were the following:

Table J-2: Systems evaluated in real-time pilot study..

Referred to as:	System#1 **Rejection**	System#2 **Proportional**	System#3 **MV**	System#4 **Reference**
Type	State-based*	State-based*	Conventional	Conventional
Feature cond.	KF_{euc}	KF_{euc}	n/a	n/a
FL	40 ms	40 ms	40 ms, 300 ms of MV	150 ms
FI	16 ms	16 ms	16 ms	16 ms
Classifier	NN_C	NN_C	NN_C	LDA
Prop. speed	MAV**	MAV**	MAV**	MAV**
Indet. state	Reject to rest	State-dependent prop. 50% MAV**		

*The state-evaluator threshold was the same for each subject and system (.004)
** Mean-absolute value (MAV) or all EMG channels.

The configurations evaluated (with the exception of the reference configuration) had a FL of 40 ms because (as reported in Chapter 4) it is a good choice for all configurations listed in Table J-1. Appendix K describes the threshold selection for System#1 and System#2. As shown in this appendix, the average threshold for the state-evaluator with KF_{euc} and FL = 40 ms is of the order of $\sim 10^{-3}$ and ranges between $\sim 10^{-2}$ and $\sim 10^{-4}$. As described in Appendix K, the discrepancies in the state-evaluator's threshold obtained with Chapter 4 are likely due to the fact that the offline data used in the analysis of Chapters 3 and 4 were acquired using a different data acquisition system than the system used while acquiring real-time data. The data acquired for the threshold determination of Appendix K was performed using the same system used in the real-time analyses (Chapter 5 and this appendix). Once the data acquired, data processing was performed as in Section 5.4.2.

Although the configurations listed in Table J-2 were fixed, empirical observations were performed to evaluate the effects of varying FL and varying feature conditioning method. Prior to performing the various trials, participants were asked to perform several target acquisition tasks in the virtual environment. They were informed that the control

configurations may change during this initial stage, but they did not know when nor how the settings varied. During this initial stage of the experiment, the configurations FL setting varied between 40-150 ms for Systems #1 and #2, and between 30-40 ms for System #3 listed in Table J-2 (FL ranges were selected based on those listed in Table J-1). A FL = 150 ms was also evaluated for System #3 for comparison purposed. The feature conditioning method of Systems #1 and #2 listed in Table J-2 varied between the KF_{euc} and the S_{cons}. The experimental analyst noted comments from the participants.

J.3 Results

The average failed completion rate, efficiency, and motion time are illustrated in the left hand plots of Figure J-2, Figure J-3, and Figure J-4 respectively. The average motion time was computed for the rejection, proportional, MV, and reference systems (with 1141, 1110, 1014, and 1027 completed tasks, respectively). The right hand plot of these figures illustrates the average difference in results when compared to the reference configuration. The average difference in motion time was computed over the 1027 tasks completed while using the reference configuration.

These results suggest that performing state-dependent proportional control while in an indeterminate state may be a good alternative to rejection. Performing MV had no significant effect on the results when compared to the reference system.

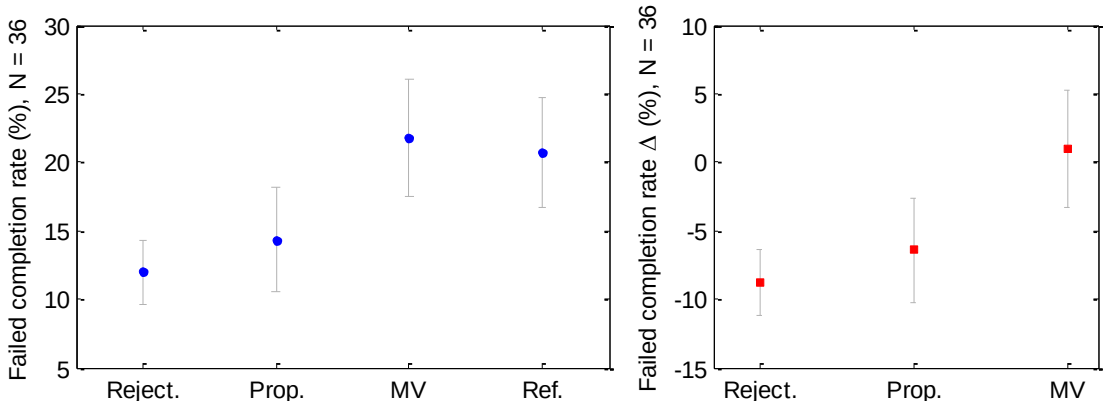

Figure J-2: Average failed completion rate overall trial (left hand plot) and difference in failed completion rate when compared to the reference configuration (right hand plot). Averages were computed over N = 27 trials.

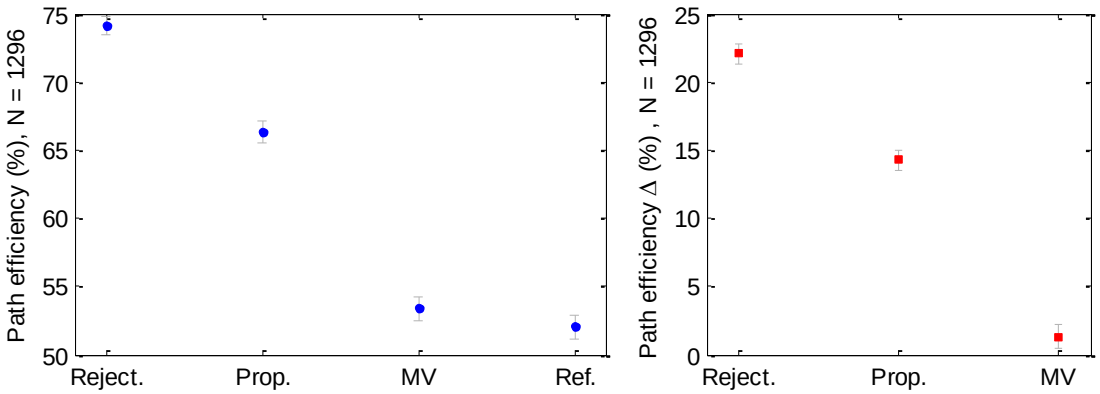

Figure J-3: Average path efficiency overall tasks (left hand plot) and difference in path efficiency when compared to the reference configuration (right hand plot). Averages were computed over N = 1296 tasks.

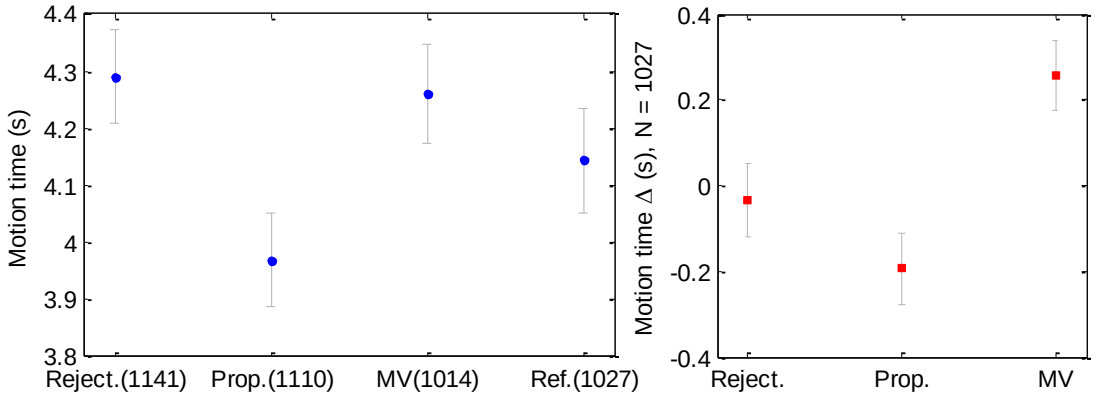

Figure J-4: Average motion time overall tasks (left hand plot) and difference in motion when compared to the reference configuration (right hand plot). The average motion time was computed over N = 1141, N = 1110, N = 1014, and N = 1027 completed tasks for the rejection, proportional, MV, and reference systems respectively. The average in the motion time difference with respect to the reference configuration was computed over the N=1027 completed tasks while using the reference system.

The FC rate and efficiency are displayed per trial in Figure J-5. From these plots it can be seen that, for all systems, the FC rate significantly ($p < .05$) reduced from trial#1 to trial#2. Although there was no significant difference in the FC rate from trial#2 to trial#3 there was a slight increase in FC rate. The results also show an improvement in efficiency from trial-to-trial but this improvement was not significant. Finally, the results indicate that for all trials the state-based classification systems (i.e. the rejection and proportional systems) obtained significantly lower FC rate than the reference system. Performing MV had no significant effect on the results when compared to the reference system. These results suggest that there was a learning effect; however, by the third trial users fail more tasks. The later suggests fatigue indicating that the experimental design may have been too long.

The average FC rate and efficiency obtained for each ID and system (across all three trials) along with the standard error (SE) are shown in Figure J-6. These plots show that, for all systems, the FC rate and efficiency was significantly ($p < .05$) worst for smaller targets (ID3and ID4) than larger targets (ID1 and ID2). When compared to the reference system, the state-based classification systems (rejection and proportional system) significantly improved the FC rate and efficiency for ID3 and ID4 ($p<0.05$) but obtained comparable results for ID1 and ID2. Performing MV had no significant effect on the results when compared to the reference system. These results suggest that the state-based classification systems allowed users to reach and stay within smaller targets more frequently.

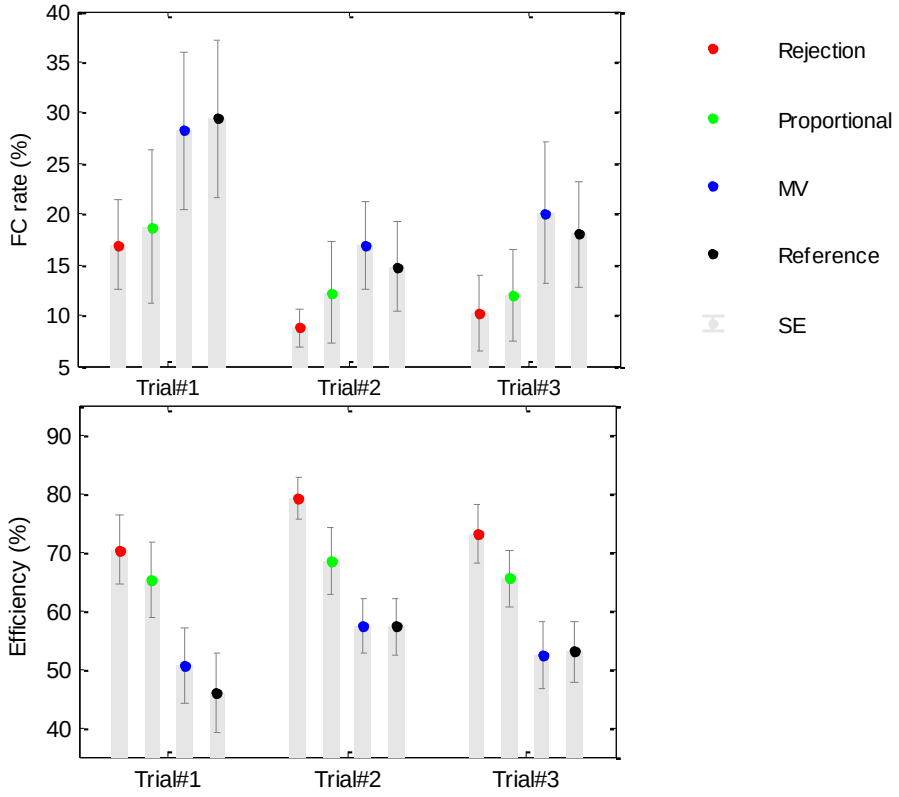

Figure J-5: Averaged failed completion rate (FC rate, top), and efficiency (bottom plot) for the first, second and third trials obtained using each system. Values were averaged across subjects for each system and trial. The vertical bars represent the standard error (SE).

The average motion time for each ID along with the Fitts' law linear regression, throughput, and the coefficient of determination (R^2) are illustrated in Figure J-7. The average throughput and b-slope obtained for each trial and system along with the standard error across subjects are shown in Figure J-8. Note that in these figures, failed tasks were assigned a motion time of 15s (the total amount of time allowed to complete a task).

In Figure J-7, the state-based classification systems obtained higher coefficient of determination R^2, lower b-slope, and higher throughput than the reference system. These results suggests that the state-based classification methods followed the expected linear

relationship between motion time and task difficulty more closely, and had better speed and accuracy tradeoff than the reference system.

The throughput increased significantly ($p<0.05$) from trial#1 to trial#2 for all systems, supporting the earlier identified learning effect. Only the state-based classification method with state-dependent proportional control obtained significantly ($p<.05$) better throughput then the reference system (this is true for all trials). There was no significant difference in the results reported for the b-slope.

The results suggest that the state-based classification method with state-dependent proportional control obtained better overall performance and a better speed-accuracy tradeoff than the reference system. They also suggest that MV does not provide any significant improvement in performance (contrary to what offline analyses showed in Chapter 4).

While evaluating the FL and feature conditioning effects by empirical observations, the experimental analyst noted the following:

- The FL had no (or very little) effect reported by the users when using the system with MV. This is likely due to the fact that MV was performed on 300ms regardless the FL;
- When the FL was decreased for the state-based systems (i.e. systems #1 and #2 in Table J-2) these was faster but harder to control;

When varying the feature condition method used for the state-based systems (i.e. systems #1 and #2 in Table J-2) some subjects reported that the S_{cons} was more responsive when using state-dependent proportional control than the KF_{euc} (but harder to control), and several reported that it was less responsive than the KF_{euc} when using

rejection. This agrees with the offline results of Chapter 4 which showed that the S_{cons} allowed users to perform tasks with shortest τ_{30}but that the state-evaluator performed more poorly while using the S_{cons}.

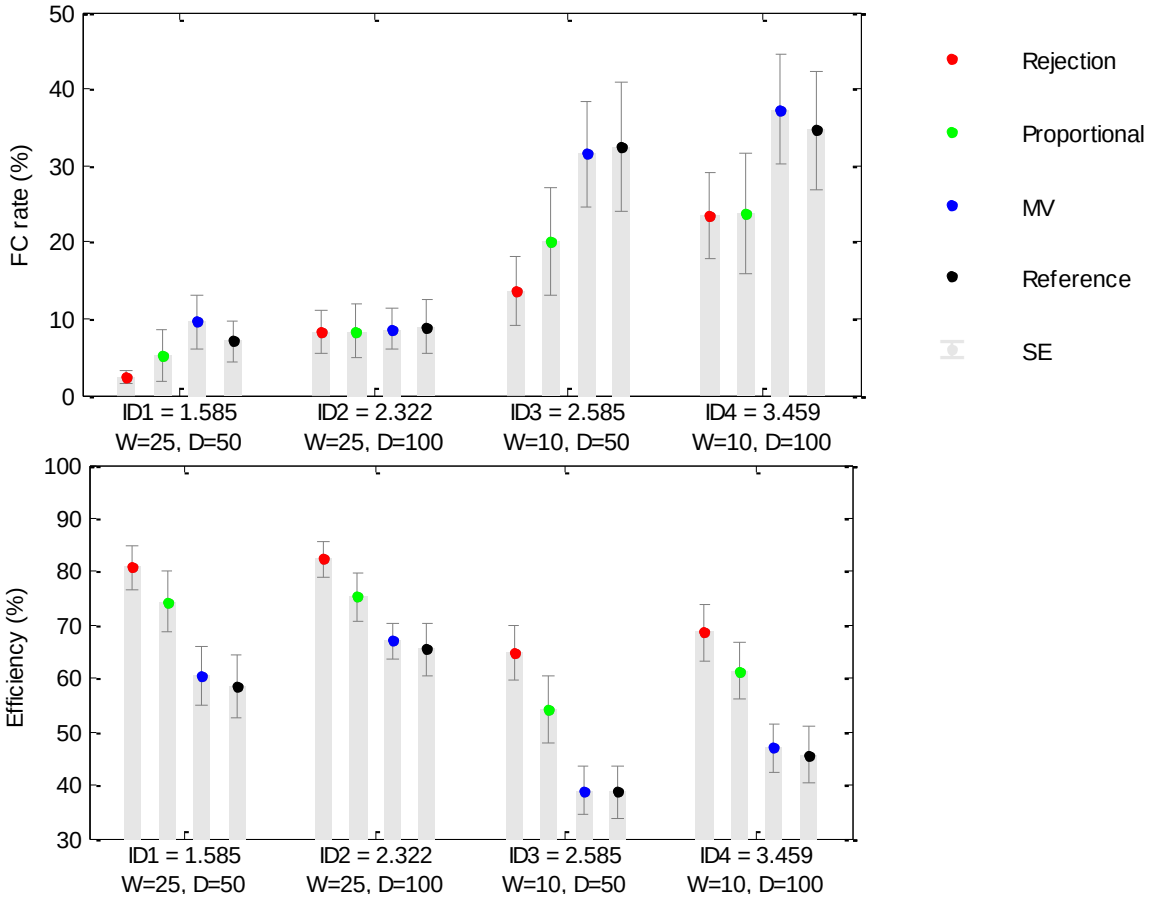

Figure J-6: Averaged failed completion rate (FC rate, top plot) and efficiency (bottom plot) for each ID obtained using the different system. These values were computed for the tasks of each ID. These values are average across subjects for each system and ID. The vertical bars represent the standard error (SE).

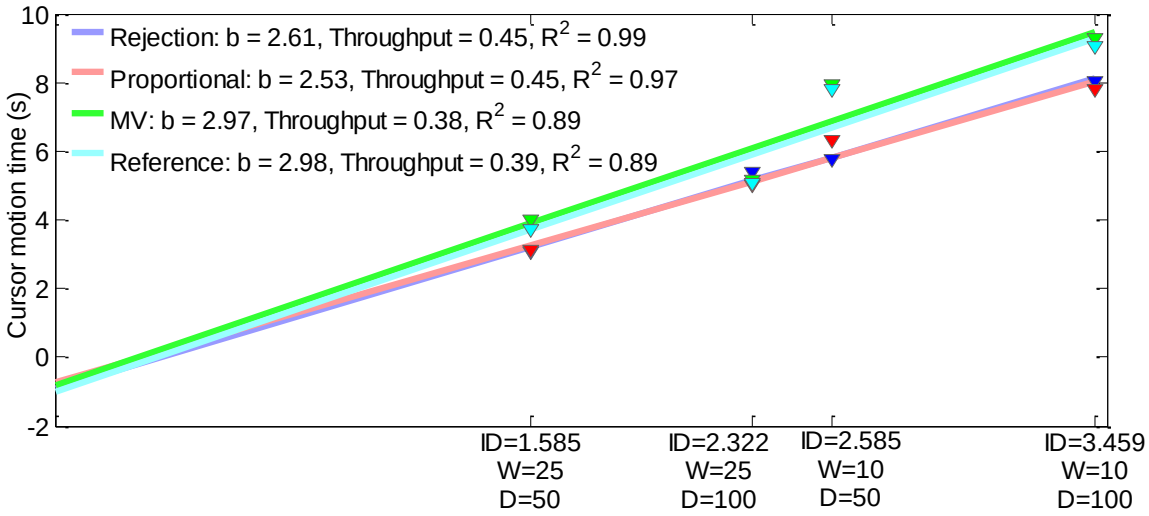

Figure J-7: Averaged motion time (MT) for each ID, and Fitts' Law linear regression b-slope and R^2 coefficient of determination, as well as the throughput for each system. For each system, the MT is averaged across the tasks with ID = 1.585, the tasks with ID = 2.322, the tasks with ID = 2.585, and the tasks with ID = 3.459. The *a*-intercept and *b*-slope parameters that best fit the data are estimated using least-square methods. The throughput is computed as the ratio of the average ID and the averaged MT.

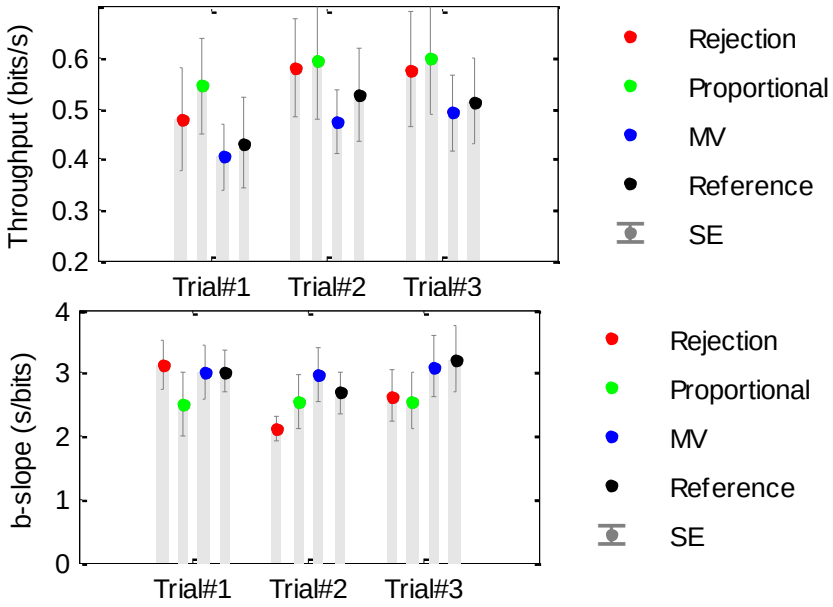

Figure J-8: Averaged throughput (top plot) and b-slope (bottom plot) for the first, second and third trial obtained using each system. For each test, the MT is averaged across the 12 tasks with ID = 1.585, the 12 tasks with ID = 2.322, the 12 tasks with ID = 2.585, and the 12 tasks with ID = 3.459. The *b*-slope parameter that best fit the data are estimated using least-square methods. For each test, the throughput is computed as the ratio of the average ID and MT. For each trial and system, values are averaged across subjects. Vertical bars represent the standard error (SE).

Appendix K – State-Evaluator Threshold for Real-Time Analysis

K.1 Overview

The real-time data acquisition was not performed with the same system as the data collection for the offline study presented in Chapters 3 and 4. It was quickly noticed that the state-evaluator threshold was affected by this change; therefore, an offline pilot study was conducted to determine the state-evaluator's threshold to use while performing the real-time study.

K.2 Methodologies

Four subjects participated in this brief offline data collection to determine the state-evaluator's threshold. These were student, staff, and faculty at the Institute of Biomedical Engineering (IBME) at the University of New Brunswick (UNB). All experiments were approved by the UNB Ethics Review Board under the Surface EMG Data Collection REB #2008-083.

Subjects were fitted with a cuff made of thermo formable gel which contained eight equally spaced pairs of stainless steel dome electrodes. The cuff was placed on the dominant forearm at the area of largest muscle bulk. A silver-silver chloride electrode was placed on the back of the hand as a reference electrode. While users performed various contractions, the eight channels of analog EMG were differentially amplified and low pass filtered. Data were sampled with a sampling frequency of 1000Hz using a 16-bit analog-to-digital converter.

Subjects were asked to perform wrist flexion, wrist extension, wrist supination, wrist pronation, chuck grip, hand open, and no motion. Training data were collected while subjects held steady-state contractions for three seconds with a two second break in

between prompts. Each motion class was repeated twice. During testing, users were prompted to start in no-motion, then to transition to a series of active classes. Subjects were given six seconds to transition and reach steady-state. This was repeated three times resulting in 18 transitions between active classes per trial. Each subject conducted three trials. Subjects conducted two sessions of three trials. Therefore, there were a total of six trials (108 transitions, not including transitions to no-motion) of wrist and hand motions per subject.

Only four of the eight channels were used for analysis. Data processing was performed as in Section 4.2.2.

K.3 Results and Discussion

The ROC curves and the threshold values for best performance are illustrated in Figure K-1 and Figure K-2 respectively. These results suggest that for FL < 150 ms while using the KF_{euc} the threshold ranges from 10^{-1} to 10^{-5} and for FL > 150 ms the threshold is close to zero. Similarly for the S_{cons}; for FL < 100 ms the threshold ranges from 10^{-1} to 10^{-5} and for FL > 150 ms the threshold is close to zero.

It is worth mentioning that the results obtained in Figure K-2 differ from those obtained in Figure 4.3; in fact, the results in Figure K-2 obtained higher thresholds especially for longer FLs. This is likely due to the fact that the data acquisitions were performed using different systems. As mentioned in Chapter 4, the features SNR has a direct impact on the choice of the threshold; configurations with higher FL, which result in features with higher SNR, can use a state-evaluator threshold closer to zero. In this case, it can be assumed that the system used to acquire the data in this pilot study (as well as the real-time studies of Chapter 5 and Appendix J) resulted in features with lower SNR

than those acquired in Chapter 4. Unfortunately, these data acquisitions were performed on different subjects making it difficult to directly compare the signal and/or feature noise of the data acquisition systems.

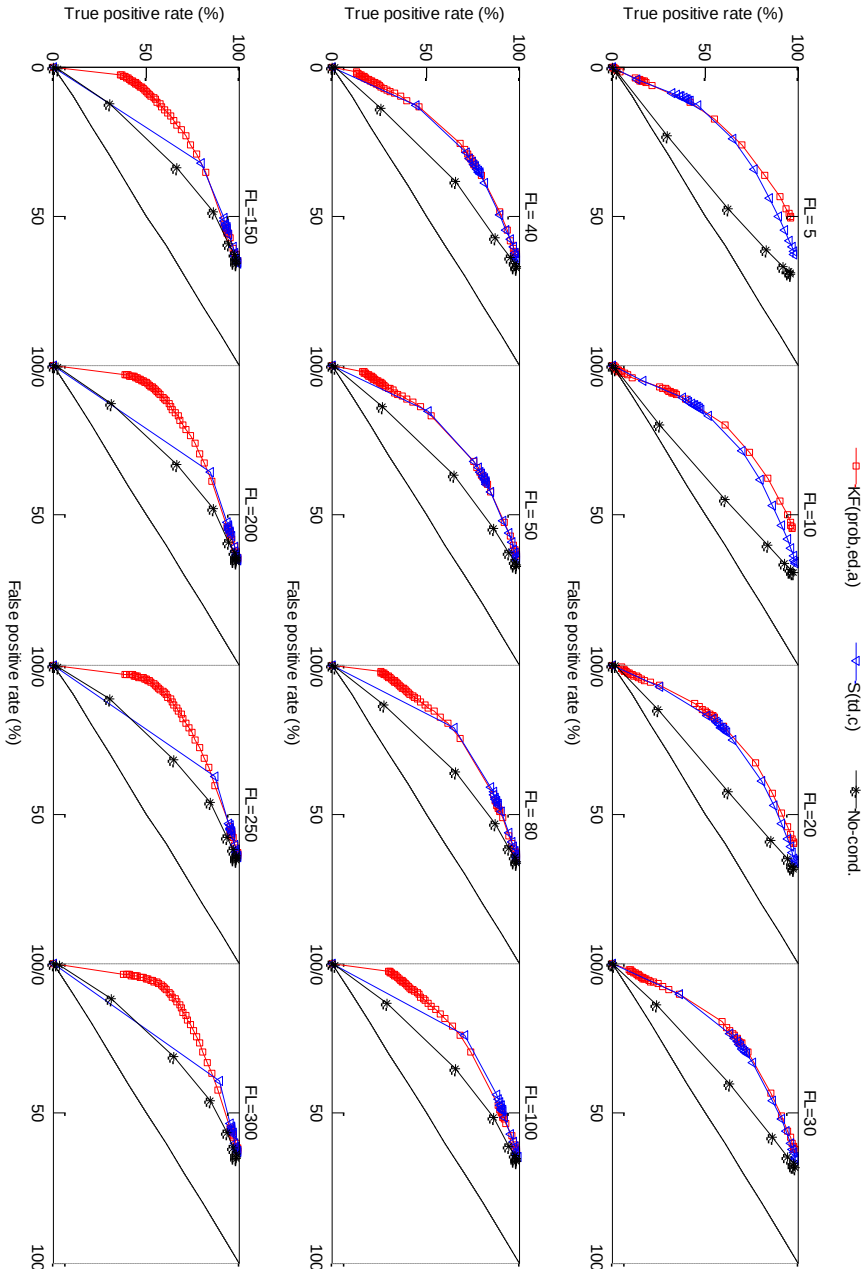

Figure K-1: ROC plots of the state-evaluator using different feature conditioning methods (and without features conditioning) for varying frame length (FL).

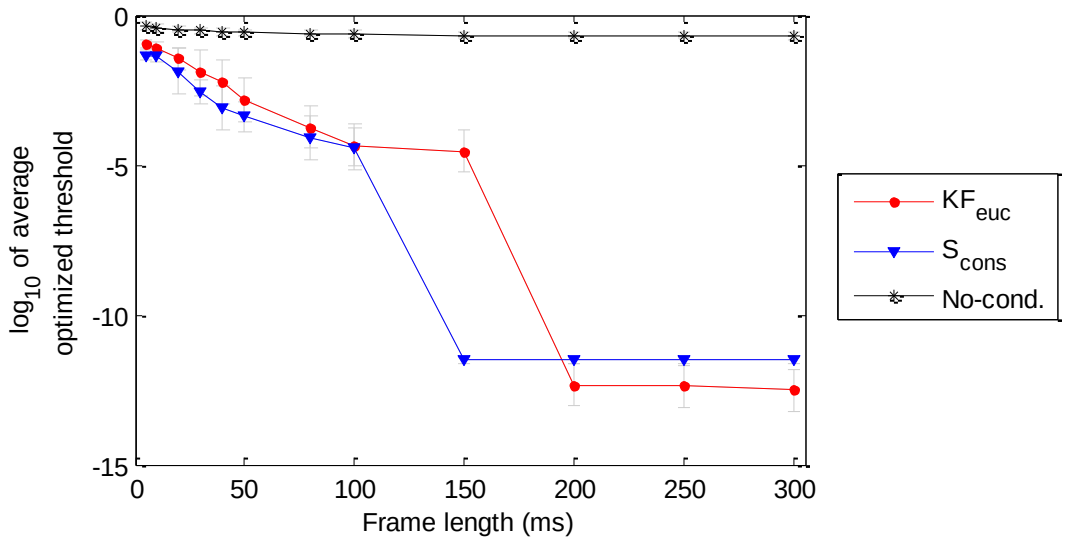

Figure K-2: Log_{10} of the average threshold value. Values are averaged across each trial than the log_{10} of this average value is computed. The markers represent the log_{10} average whereas the vertical bars show the range of the threshold (i.e. log_{10}(average - standard deviation) - log_{10}(average + standard deviation)).

Appendix L – Additional Information about the Data

Table L-1 lists detailed information regarding the data used in this work. Please contact Katerina Biron at katbiron@gmail.com or the Institute of Biomedical engineering at the University of New Brunswick to obtain copies of data or more information about the data.

Table L-1: Details about the data used in this work.

Data type	Number of subjects	Number of trials*	Number of testing sets**	Total number of (training/testing) data sets	Dates taken	Data size (MB)
Offline	15	2	3	(30/90)	Feb-2014	238
Real-time data	12	3	4	(36/144)	Mar-2013	66.9
Real-time pilot data	9	3	4	(27/108)	Oct-2012	100

*A *trial* consists of training data and several sets of testing data.
** There are several testing *sets* within a trial. For the real-time data, every testing set within a trial is collected using a different control configuration.

Curriculum Vitae

Katerina Biron
Gatineau, Qc

Universities attended

Ph.D. in Electrical Engineering, *University of New Brunswick, Fredericton NB, January 2017*
Major: Signal Processing
Minor: Biomedical Engineering

M.Sc. in Electrical Engineering, *University of New Brunswick***,** *Fredericton NB, March 2010*
Thesis Title: EMG Pattern Recognition Adaptation.

B.S in Mechanical Engineering, *Old Dominion University, Norfolk VA, December 2007*
Concentration: Aerospace Engineering
Minor: Computer Engineering

Diplôme d'Études Collégiales, *Collège Jean-de-Brébeuf, Montréal Qc, May 2003*
Field of study: Applied Sciences

Publications

Biron, K., "Ship Wake Analysis for RARDARSAT-2 Images", MDA, Richmond, BC, Rep. TBD, Issued in 2015.

Biron, K., "Velocity Bunching, Tilt Modulation, and Hydrodynamic Tilt Simulation of Wakes", MDA, Richmond, BC, Rep. Rx-Rp-53-6955, Issued in 2014.

Biron, K., and Englehart, K. (2012) Improving the response time of myoelectric prostheses: A Study of Kalman Filtering, The 35th Canadian Medical and Biological Engineering Conference (PhD work).

Scheme, E., Englehart, K., and **Biron, K**. (2011) Improving Myoelectric Pattern Recognition Positional Robustness Using Advanced Training Protocols, 33rd Annual International IEEE EMBS Conference.

Biron, K., and Englehart, K. (2010) EMG Pattern Recognition Adaptation**,** The 18th Congress of the International Society of Electrophysiology and Kinesiology Conference (M.Sc work).

Yu, S., Kim, T., **Biron, K.M.**, and Yuh, J. (2006) Development of Agent Based Docking System for Autonomous Underwater Vehicles. Proceedings of the Sixteenth International Offshore and Polar Engineering Conference. 1: 221-228 (Undergraduate work).

Conference presentations

EMG Pattern Recognition Adaptation, ISEK, Aalborg, Denmark, June 2010

Improving the response time of myoelectric prostheses: A study of Kalman filtering, Halifax, June 2012