MYOELECTRIC HUMAN COMPUTER INTERACTION USING RELIABLE TEMPORAL SEQUENCE-BASED MYOELECTRIC CLASSIFICATION FOR DYNAMIC HAND GESTURES

A Dissertation

by

SUNGTAE SHIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Reza Langari
Co-Chair of Committee,	Reza Tafreshi
Committee Members,	Won-Jong Kim
	Yoonsuck Choe
Head of Department,	Andreas A. Polycarpou

December 2016

Major Subject: Mechanical Engineering

ABSTRACT

To put a computerized device under human control, various interface techniques have been commonly studied in the realm of Human Computer Interaction (HCI) design. What this dissertation focuses on is a *myoelectric interface,* which controls a device via neuromuscular electrical signals. Myoelectric interface has advanced by recognizing repeated patterns of the signal (*pattern recognition-based myoelectric classification*). However, when the myoelectric classification is used to extract multiple discrete states within limited muscle sites, there are robustness issues due to external conditions: limb position changes, electrode shifts, and skin condition changes. Examined in this dissertation is the robustness issue, or drop in the performance of the myoelectric classification when the limb position varies from the position where the system was trained.

Two research goals outlined in this dissertation are to increase reliability of myoelectric system and to build a myoelectric HCI to manipulate a 6-DOF robot arm with a 1-DOF gripper. To tackle the robustness issue, the proposed method uses *dynamic motions* which change their poses and configuration over time. The method assumes that using dynamic motions is more reliable, vis-a-vis the robustness issues, than using *static motions*. The robustness of the method is evaluated by choosing the training sets and validation sets at different limb positions. Next, an HCI system manipulating a 6-DOF robot arm with a 1-DOF gripper is introduced. The HCI system includes an inertia measurement unit to measure the limb orientation, as well as EMG sensors to acquire muscle force and to classify dynamic motions. Muscle force and the orientation of a forearm are used to generate velocity commands. Classified dynamic motions are used to change the manipulation modes. The performance of the myoelectric interface is measured in terms of real-time classification accuracy, path efficiency, and time-related measures.

In conclusion, this dissertation proposes a reliable myoelectric classification and develops a myoelectric interface using the proposed classification method for an HCI application. The robustness of the proposed myoelectric classification is verified as compared to previous myoelectric classification approaches. The usability of the developed myoelectric interface is compared to a well-known interface.

With love and honor,

this dissertation is dedicated to

my precious family who are my true supporters,

and my respectful mentors who advise my life and research.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Reza Langari, and my co-chair, Dr. Reza Tafreshi, for their dedicated help, guidance, encouragement and advice during my PhD degree. Their advice has led to the success of my research. All of my thought, critical ideas and scientific writing skills have been developed by their guidance and support. Furthermore, I appreciate the kindness, valuable comments, and support of my committee members: Dr. Won-jong Kim and Dr. Yoonsuck Choe. Last but not least, I would like to express my uncountable gratitude to my family: my mother (Eun-Soon Park), my father (Jai-Soon Shin), my younger sister, her husband, and my adorable nephew and niece. Moreover, I need to say many thanks to my friends: Dr. Han-Ul Yoon, Eun-Kyeong Jung, Youngho Kim, Hyundoo Jeong, and Dr. Sanghyun Lee for their assistance and advice during my Ph.D. degree. Finally, thank you God for your guidance and blessing in my life. I will always remember that you are by my side. Thank you, Lord.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

1.1 Background

As computer technology and robotics have become popular, the need for controlling robots or computerized devices intuitively and based on human intent has increased. As a result, various interface techniques recognizing human intent have been studied in the Human Computer Interaction (HCI) and Human Machine Interface (HMI) realms. Among many proposed approaches such as electroencephalography (EEG), vision, and inertial measurement, the recognition of human motion intent by using electromyographic (EMG) signals has been actively studied since EMG signals can represent human muscle activities quite well [1]. With this in mind, this dissertation focuses on myoelectric interface (or myoelectric control), which is a strategy to control a device via myoelectric signals generated when humans use gestures or move a limb can be used to: 1) to control powered prosthetics for amputees and robots based on human intent, and 2) to recognize human gestures for other applications.

Myoelectric interface has advanced from simply using the amplitude of myoelectric signals (conventional myoelectric interface) to recognizing repeated patterns of the signal by using pattern analysis and machine learning techniques (pattern recognition-based (PR) myoelectric interface). A problem the conventional myoelectric interface has, however, is that the interface needs more muscle sites to augment controllable states. In order to solve this limitation, PR myoelectric interface classifies the myoelectric signal patterns that can identify certain gestures. Due to the development of pattern analysis and machine learning techniques, PR myoelectric interface extracts more independent discrete states than the conventional myoelectric interface with limited muscle sites. Even though PR myoelectric interface enhances the scope of recognizable states, it is still not sufficiently robust due to external conditions: limb position changes, electrode shifts, and skin condition changes.

1.1.1 Electromyogram

Electromyogram (EMG) signal (or myoelectric signal) is an electrical signal superposing the motor unit action potentials (MUAP) caused from muscle contraction. An EMG signal has an amplitude between ±5 mV and the informative content ranges from 6 to 500 Hz [2]. In general, an unprocessed EMG (or surface EMG) signal is called a raw EMG (or sEMG) signal. In general, the EMG signal can be used for controlling prosthetic devices, assisting rehabilitation tasks, and recognizing human motions in HCI applications because of its capability to represent the human motion intention [3-5]. Human motions mostly involve muscle contraction. The EMG signal is a suitable bio-signal source to stand for the muscle contraction.

To use the EMG signal for prosthetic devices and rehabilitation tasks, finding a relationship between the EMG signal and muscular force is essential. In order to find the relationship between EMG signals and muscular forces, many signal processing methods have been studied. Well-known approaches are full-wave rectification, linear envelope, and root mean square (RMS) techniques [2]. Briefly speaking, the full-wave rectification converts all negative EMG signals to positive ones. The linear envelope technique passes the raw EMG signal through a low-pass filter (typically 3~6 Hz) after full-wave rectification. RMS looks similar to the linear envelope, but RMS is more popular because of its inherent meaning (the power of the signal). The above techniques are widely used in conventional myoelectric interface methodologies. In *myoelectric classification* using pattern recognition, more features from the raw EMG signal are extracted and used. The brief explanation of EMG features used in the myoelectric classification can be found in the next section.

In EMG signal processing, there still exist major issues: 1) crosstalk, 2) EMG changes with varying joint angles, 3) movements of the innervation zone with respect to the skin, 4) varying distance from newly recruited motor units to electrodes (biological tissues act as low-pass filters), and 5) difference in the relationship of the firing rate, EMG amplitude, and force. These are well-known reasons causing non-stationarity and wide-variance of the EMG signals.

Generally speaking, a skeletal muscle is a cluster of motor units and a motor unit is a set of muscle fibers that are connected to the same motor neuron. Based on the structure of the muscles, an electrode measuring the EMG signal cannot pick up just a motor unit action potential (MUAP) of a given motor unit at a certain muscle. MUAPs from surrounding active muscles are picked up together by the electrode. This is called *cross-talk*, which is a reason of the variation in EMG signals. Moreover, when a human moves, the pose, role, and status of muscles are also changed, these changes vary the EMG signal as well. Moreover, in a perspective of collecting EMG signals, electrodes are located on the skin surface. One fact related to this is that the skin and the muscles beneath the skin are not fixed together. They move relative to each other when humans change their pose. This movement between the skin and the muscles affects the collected EMG signals.

The other issues are related to the nature of the skeletal muscles in generating muscle forces. There are two ways to increase muscle force: recruiting more motor units (MU) and firing more stimulation, or action potentials (AP). When recruiting more MUs, the smallest MUs are firstly recruited and the largest MUs are lastly enrolled (referred to as the size principle) [6]. Moreover, newly required MUs can be recruited from different locations compared to the previous ones. In the way of generating more muscle force, the strength and frequency of EMG signals varies according to the distance between the newly recruited MUs and the electrode. Besides, the relationship between the firing rate-force and the EMG amplitude-force is variable. That means that EMG signals vary based on the location of MUs and the respective firing rate.

The major issues noted above make EMG signal processing difficult. To tackle the difficulties, many approaches have been studied in the past. One approach is myoelectric classification using well-developed techniques in the computer science literature such as machine learning and pattern analysis.

1.1.2 Myoelectric Classification

The classification process of myoelectric signals to find particular gestures (or motions) is called myoelectric classification, and is the basis for myoelectric control (also known as myoelectric interface [7]). Roughly speaking, myoelectric control is a control strategy using EMG signals to translate the human motion intent into pre-defined control commands. An EMG signal is a prominent bio-signal source to put a prosthesis or an exoskeleton under human control [3-5, 7] because EMG signals can represent the human movement intent originating in the brain. Well-known examples of myoelectric control using EMG signals are prosthetics for amputees [8-11], exoskeletons for rehabilitation tasks [12], and augmenting human power [13] in military and industrial setting.

The early myoelectric control methods, called *conventional myoelectric controls* [5], have exploited the amplitude of the EMG signal to control one- or two- degrees of freedom prosthetic devices. Under conventional myoelectric control, large number of EMG electrodes are needed to improve the functionality of a prosthetic device. However, the number of electrodes is restricted because the area of human skin where electrodes can be attached is limited. One approach to solve this limitation is *pattern recognition-based myoelectric control* to find repeated patterns in the EMG signal [3, 4]. It increases the number of functions per electrode compared to conventional myoelectric control [3, 5].

Pattern recognition-based myoelectric control works through *pattern-based myoelectric classification (PMC)*. PMC is based on the assumption that each human motion has a highly-correlated, unique, and repeated pattern in the EMG signal [5]. Assuming this and using well-developed pattern recognition techniques, PMC extracts distinct EMG patterns in each motion and uses them as control commands.

PMC generally consists of *pre-processing*, *segmentation*, *feature extraction*, *dimension reduction*, *classification*, *post-processing*, and *command generation* as depicted in Figure 1. In pre-processing, raw EMG signals are filtered to reduce noise; generally, a notch filter is used for power-line noise (50 ~ 60 Hz), and low-pass and high-pass filters are also used to remove movement artifacts and meaningless noise in high frequency ranges [2]. Segmentation is a process to make a segment, which is usually a 10~300 millisecond time-window of the given EMG signal, to extract features. Feature extraction translates the EMG signals into features capable of distinguishing EMG patterns. Dimension reduction finds highly informative features in the broad feature space. This reduces the size of the feature space and can increase the performance of the classification process. Classification can be divided into two steps: *training* and *prediction*. In training, a classifier learns from training sets how to distinguish motion classes. During prediction, the trained classifier identifies the class of an input motion based on the knowledge obtained during the training process. Post-processing step usually generates commands for prosthetic robots or interfacing computers according to the class of input motion.

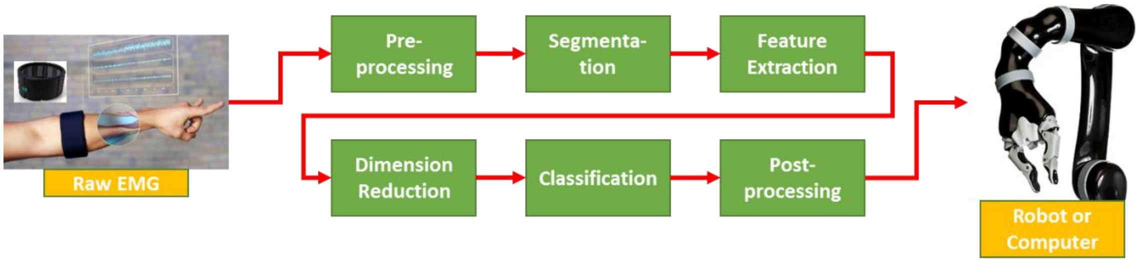


Figure 1 Process diagram of pattern recognition-based myoelectric classification

Feature extraction is a process to extract representative features from raw EMG signals because it is difficult to directly interpret the raw EMG signals for the purpose of the classification. For this reason, a variety of features have been introduced. In general, features used in myoelectric classification are categorized into three groups: time domain (TD), frequency (or spectral) domain (FD), and time-frequency (or time-scale) domain (TFD) features [3, 7]. TD features are basically obtained from the amplitude of EMG signals such as integral of EMG (IEMG), wavelength (WL), variance (VAR), Mean Absolute Value (MAV), Zero Crossings (ZC), and Slope Sign Changes (SSC). An advantage of TD features is fast in calculation because of no need for mathematical transformation, but TD features are less robust to noise. On the contrary, FD features generally need Fourier transform to be calculated. Examples include cepstral coefficients, autoregressive coefficients (AR), and power spectral density (PSD). FD features are possibly used for muscle fatigue [14], EMG/force relations [15] and MU recruitment and firing pattern analysis [3]. A limitation of Fourier transform is that data after applying the transform does not indicate its temporal information even though the temporal information can be retrieved by inverse Fourier transform. To overcome this limitation, TFD features are calculated by using time-frequency analyses such as short time Fourier transform [16], wavelet transform [16, 17], wavelet packet transform [16], stationary wavelet transform [16], and empirical mode decomposition [18, 19]. Wavelet analysis and empirical mode decomposition can handle the non-stationary property of EMG and expand available feature spaces.

Englehart and Hudgins [20] used four TD features (MAV, ZC, SSC, and WL) at each of four channels for a real-time multi-function myoelectric control. In [21], the authors tested the reliability of fifty EMG features with EMG datasets collected during 21 days. They mentioned that sample entropy (SampEn) outperformed other features when using a linear discriminant analysis classifier. The average classification accuracy was 93.37%. Zhaojie, et al. [22] introduced the nonlinear features, DET (a predictability measure) and ENTR (a Shannon entropy of the frequency distribution). The authors compared the classification accuracies of the nonlinear features and other TD and FD features. The top accuracy was 96.7% when using Fuzzy Gaussian Mixture Model based classifier and a feature set associated with the Willison Amplitude (WAMP) and DET.

With the increase of number of EMG electrodes and the dimension of features, the dimension reduction has become a necessary process to enhance the classification accuracy. Various dimension reduction techniques for myoelectric classification have been studied: principal components analysis (PCA), linear discriminant analysis (LDA), self-organizing feature map [23], and orthogonal fuzzy neighborhood discriminant analysis [24]. Chu, et al. [23] studied a linear-nonlinear feature projection using a collaboration of PCA and self-organizing feature map (SOFM) for dimensionality reduction and nonlinear mapping of the features. Khushaba, et al. [24] proposed an extension of LDA, called as orthogonal fuzzy neighborhood discriminant analysis (OFNDA). OFNDA used advantages of both of uncorrelated LDA (ULDA) and fuzzy LDA (FLDA) for the decorrelation of data and the maximum separation in a fuzzy manner. Phinyomark, et al. [25] compared the performance of PCA, LDA, ULDA, OFNDA, and orthogonal LDA (OLDA) in the myoelectric classification of eight upper-limb movements. They tested the five dimensionality reduction methods with seven different feature sets and concluded that ULDA, OLDA and OFNDA are suitable for myoelectric classification. Khushaba and Kodagoda [26] proposed Mutual Components Analysis (MCA) to combine both feature selection and feature projection. MCA first removes noisy and redundant features before applying PCA. They tested MCA with EMG datasets including fifteen finger motions collected from eight human subjects. The results showed that MCA performs better than PCA.

The classification process, undoubtedly, plays a crucial rule in myoelectric classification because its ability and capability is directly connected to the performance of myoelectric systems; the process starting with feature extraction and dimension reduction is completed via classification algorithms. For this reason, significant efforts have focused on classification algorithms. There are a number of studies related to classification algorithms for myoelectric systems: linear discriminant analysis [20, 27, 28][45], k-nearest neighbors [29], multi-layer perceptrons [30], time-delayed neural network [20, 31], extreme learning machine [32], fuzzy [33], fuzzy logic [34], adaptive neuro-fuzzy inference system [35] NEURO-fuzzy [36], fuzzy ARTMAP networks [37], fuzzy-MINMAX networks [38], fuzzy Gaussian mixture models [22], wavelet neural network [39], hidden Markov models [40], Gaussian mixture models [41-43], and support vector machines [44, 45].

From the perspective of moving body parts for gestures, hand and wrist motions using major muscles, such as hand open/close, wrist flexion/extension, and forearm pronation/supination, were frequently used at the beginning stage in myoelectric classification because they are more convenient to detect the difference of EMG signal patterns. However, due to the need for more dexterity in myoelectric control, many finger motion classification methods have focused on to increase the number of discriminable motions. Khushaba, et al. [46] classified 10 individual and combined finger motions with an average classification accuracy of about 90%. They used a post-processing, Bayesian data fusion, to enhance the classification accuracy. EMG datasets were collected from two electrodes located on a forearm from eight participants. Al-Timemy, et al. [47] used a combination of orthogonal fuzzy neighborhood discriminant analysis for dimensionality reduction and linear discriminant analysis for classification. The myoelectric classification method recognized fifteen finger motions at 98% accuracy with ten intact subjects and twelve finger motions at 90% accuracy with 6 amputated subjects. Tenore, et al. [30] discriminated 10 finger motions (individual flexion and extension of each finger) with over 90% accuracy in a transracial amputee. Moreover, they showed that there is no statistically significant difference ($p<0.05$) between a transracial amputee and five intact subjects.

On the other hand, there are studies focusing on the classification of dynamic motions via myoelectric signals for the purpose of diversifying the motion type the myoelectric classification recognizes. With the perspective of Mitra and Acharya [48], gestures can be divided into two types: *static motion* and *dynamic motion* (see Figure 2). In general, a static motion has an invariant pose and configuration in time which means that it can be represented by relatively repeated and constant multi-dimensional EMG features with negligible changes in time (called hereafter *a pattern*). For example, "Hand Close" motion in Figure 2 (a) is a static motion because a formation of a hand and fingers is fixed and does not change in time. However, dynamic motions change their poses and configurations over time. In Figure 2 (b), “Finger snapping” is an example of a dynamic motion; evidently, during a finger snapping motion, the fingers move and RMS values of EMG signals change in a noticeable manner. This means that a dynamic motion needs to be expressed by a temporal sequence of multi-dimensional EMG features that are changed during the motion (called hereafter *a sequence*). In myoelectric classification, a pattern (an EMG feature vector) is used to identify a static motion, whereas a sequence (a series of EMG feature vectors) represents a dynamic motion.

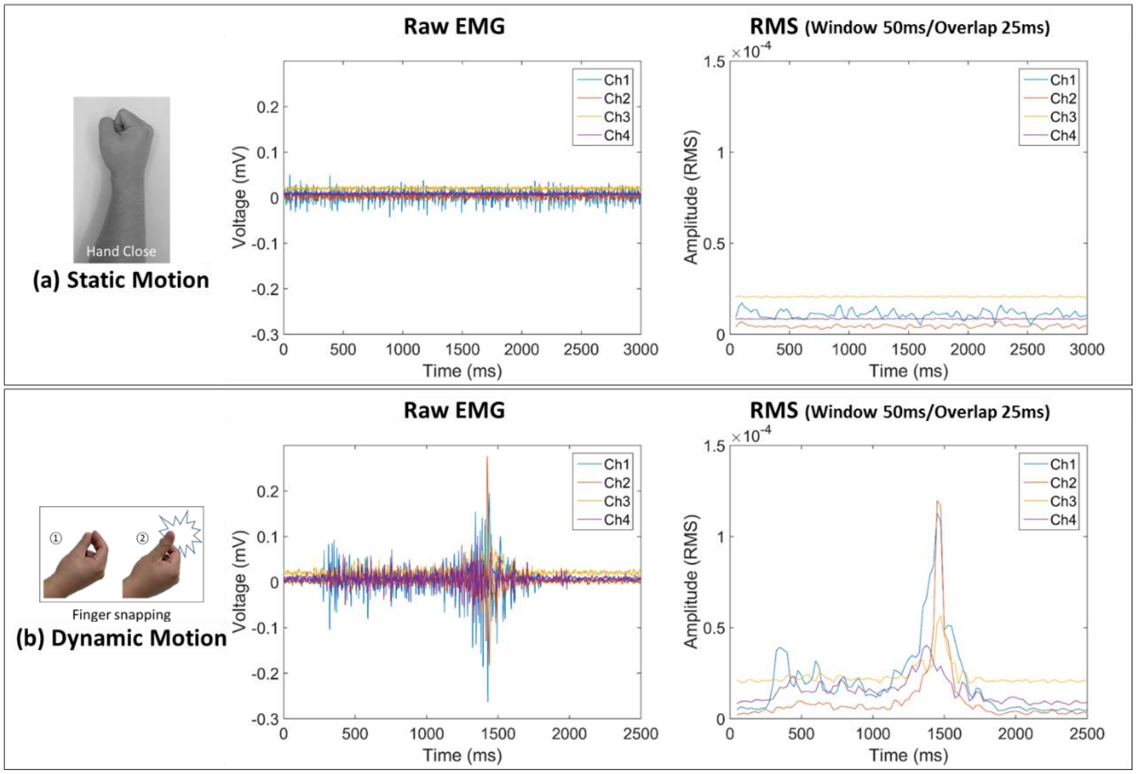


Figure 2 (a) Static motion vs. (b) dynamic motion; Root Mean Square (RMS) in a static motion (Hand Close) is relatively steady, whereas RMS in a dynamic motion (Finger Snap) is changed over time.

To classify dynamic motions, there are vision-based [49], accelerometer-based [50, 51], and EMG-based [52] approaches, as well as their combinations. Shen, et al. [49] recognized dynamic hand gestures via a vision-based system. They used a template matching approach based on their proposed motion divergence fields, which is an image representation method using a gray-scale. Akl, et al. [50] recognized dynamic motions with a single 3-axis accelerometer. They generated each template of motions by employing dynamic time warping and affinity propagation. Wang, et al. [53] utilized a three-axis accelerometer, three sEMG sensors and a vision sensor to recognize dynamic hand motions. They used EMG signals for two purposes: 1) to identify the start and end of a motion by monitoring the intensity of EMG signals and 2) to classify four static motions as commonly done in myoelectric classification. The dynamic motions were determined by acceleration and visual trajectories. Wolf, et al. [54] developed the *"BioSleeve"* to interpret static and dynamic gestures using EMG sensors and an inertial measurement unit (IMU) sensor. In the classification of static gestures, the IMU was only exploited to figure out elevation angles of an arm. For the dynamic gesture classification, both IMU and EMG data were used via a dynamic programming technique. AbdelMaseeh, et al. [52] classified ten dynamic motions using only EMG signals. They used multi-dimensional dynamic time warping (MD-DTW) to classify dynamic motions as finding the best match among all labeled trajectories from the training phase. Lu, et al. [55] also focused on the myoelectric classification of dynamic motions. They described time-varying dynamic motions by using the expectation maximization and HMM. They compared the performance of their proposed method with two classifiers (artificial neural network and support vector machine).

1.1.3 Myoelectric Interface

With wide-spreading computer technology, computerized devices and intelligent systems are right beside human life. In this circumstance, the need for interfacing devices conveniently and easily also has increased. To satisfy this need, numerous interfacing approaches in HCI have been studied. This dissertation focuses on *myoelectric interface*, which is an HCI strategy using myoelectric signals to transfer human intent to a device. Many studies have focused on myoelectric interface because myoelectric signal is suitable to represent human movements. There are several benefits of myoelectric interface: 1) human is more familiar with gestures, 2) there is no limitation for outdoor activities, 3) this is no interference when using a hand, and 4) information of muscle force can be used for other purpose. In general, human beings, especially, elderly persons and amputees, are more familiar with acting gestures than manipulating a joystick or a controller box with tens of buttons. Moreover, myoelectric interface has no limitation for outdoor activities compared to vision-based gesture recognition. Compared to vision-based systems, myoelectric interface is more suitable to recognize diverse and dexterous finger and hand motions because myoelectric interface is not affected by difficulties the vision-based systems have such as blocking fields of vision and various strengths of light. Although data gloves can recognize dexterous finger and hand motions, there is a challenge due to interference with hand movements and tactile feedback caused from the skin covered by the data gloves. However, myoelectric interface is free from hindering hand movements and tactile feedback. This advantage is valuable for many applications. In addition, the information of muscle force estimated from EMG sensors is beneficial as well. The recognized muscle force can be used for more skillful and user-friendly applications.

There are a number of studies related to myoelectric interface. Fukuda, et al. [42] developed an EMG teleoperation system consisted of a hand and wrist control part and an arm control part. The hand and wrist control part handled the end effector of a robot manipulator through myoelectric classification and the arm control part manipulated the arm section of the robot manipulator based on a three-dimensional position sensor. Artemiadis and Kyriakopoulos [56] proposed the switching regime (SR) model to manipulate a robot arm with adjusting the changes of EMG signals over time. The SR decoding model shifts to another decoding model that can express the relationship between changed EMG signals and arm motions. Xu, et al. [57] presents a sign language recognition system based on the data fusion of a three-axis accelerometer and multi-channel EMG sensors. For the system, a decision tree and multi-stream hidden Markov models were utilized. Liu and Young [58] developed a myoelectric interface for a 6-DOF robot manipulator. The myoelectric interface used four EMG sensors and classified upper limb statuses by using empirical mode decomposition for the feature extraction and adaptive neuro-fuzzy inference system for the classification. Wolf, et al. [59] and Wolf, et al. [54], built the BioSleeve consisting of an IMU and 16 EMG sensors. In [59], they used the IMU sensor to determine the level of a forearm. The level information was used to increase the number of recognized gestures. They used hand gestures to manipulate two robot arms and two robot hands. In [54], the IMU information was used for classifying dynamic motions. They basically recognized seventeen static and nine dynamic gestures. Min Kyu, et al. [60] manipulated a robot via IMU and sEMG sensors. They estimated human arm motions by using IMU to generate commands for an arm of the robot and determined hand postures and grasping force by using sEMG signals for a hand of the robot. Haque, et al. [61] proposed the *"Myopoint"* system that can replace the function of a computer mouse using IMU and EMG sensors by using a Myo™ armband. Georgi, et al. [62] worked on the session- and person-independent myoelectric classification. The authors used twelve gestures collected from five subjects during five sessions at different days and modeled each of gestures by using hidden Markov models (HMM) and Gaussian mixture models (GMM). The authors mentioned that the classification accuracy in session-independent was 97.8% and the accuracy in person-independent was 74.3%.

1.1.4 Needs for Robustness in Pattern-based Myoelectric Classification

PMC still has weaknesses to be commercially used although it has improved functionality of myoelectric systems compared to the conventional myoelectric approaches [5, 63]. In fact, a crucial weakness of PMC is lack of robustness against variations of EMG signals over time [4]. EMG signals can vary due to different reasons, such as electrode shift [64, 65], long-term use [21, 66-68], limb position change [69-71], fatigue [72], and skin condition [73]. This requires frequent re-calibrations of the classifier to perform properly. This cumbersome re-calibration task is a key factor that has prevented pervasive use of myoelectric devices [5]. The robustness issue of PMC should be addressed in order to decrease the gap between academic research and commercial uses. Without robustness, usability of myoelectric system will be limited. Farina, et al. [5] well explained the needs for the reliability of myoelectric system in the variations of EMG signal patterns related to changing positions of electrodes caused by re-wearing, limb position changes (range of activities), electrode/skin impedance changes, and long-term use.

To solve the robustness issues, numerous studied have been conducted. For the limb position changes, Scheme, et al. [74] and Fougner, et al. [70] used additional sensors (accelerometers) that have different modality from EMG sensors in order to take limb positions into account. Geng, et al. [69] proposed a cascade classification scheme using a position classifier and multiple motion classifiers in charge of different limb positions. Khushaba, et al. [71] developed robust myoelectric features that are invariant to the translation and scaling of EMG signals. They used the features to tackle unreliable issues caused from limb position changes. Masters, et al. [75] also tried to increase the classification accuracy due to limb position changes by training a classifier with data from different limb positions. Their approach needed no additional hardware and sensors. They tested four different configurations with an individual classifier (trained at a certain limb position), an aggregate classifier (trained by data collected from all limb position), and a position classifier (figure out a current limb position). Park, et al. [76] developed a proportional myoelectric interface that performed reliably with respect to arm position changes as estimating the likelihood of arm positions.

For long-term uses, Phinyomark, et al. [21] benchmarked a group of EMG features in terms of their robustness. The result indicated that the sample entropy EMG feature (SampEn) showed the best performance with a linear discriminant analysis classifier.

Kaufmann, et al. [67] collected EMG signals of 10 motions during 21 days and used 5 different classification methods. The result indicated that most of the classifiers needed re-training to prevent decreasing over time in the classification accuracy. Amsuss, et al. [68] recorded EMG patterns of 8 motions from 5 human subjects during 5 consecutive days. The authors concluded that the accuracy of myoelectric pattern classification decreased on the long term behavior.

As far as electrode shift, muscle fatigue, and various muscle force levels, López, et al. [77] used two data fusion algorithms, Variance Weighted Average (VWA) and Decentralized Kalman Filter (DKF), to correct failures of a myoelectric control system caused from electrode shift and various noise sources. Tkach, et al. [78] investigated the stability of EMG features with respect to electrode shift, muscle fatigue, and various muscle contraction levels. They mentioned that the effect of muscle fatigue is less than the others (electrode shift and various muscle contraction levels). Tkach, et al. [79] measured the robustness of TD features in terms of force variations (low and high force levels), muscle fatigue, and electrode location shift. Young, et al. [64] and Young, et al. [80] compared the classification errors with respect to shifting electrodes, varying distance between positive and negative poles of bipolar channel sEMG sensors, and changing electrode size and orientation of electrode poles. Al-Timemy, et al. [81] also tested the robustness of EMG features with three different muscle force levels (low, medium, and high force levels). They stated that muscle force variation caused a degradation in the classification accuracy. In their result, TD features performed slightly better than AR coefficients and RMS EMG features. In addition, when training a classifier with all force levels, the classification error dropped to about 17%.

1.2 Objectives of the Dissertation

Research goals in this dissertation are: 1) increasing the reliability of a myoelectric system by using dynamic motions and 2) building an HCI with a reliable myoelectric interface to manipulate a robot arm. To do this, the proposed myoelectric interface uses dynamic motion, which is a posture changing its shape, pose, and configuration over time. In general, a dynamic motion can be represented by a sequence of myoelectric signals during a certain period (usually, from the start of the motion to the end of it). Compared to dynamic motions, static motions such as “opened hand” and “closed hand” have no temporal information because they do not change their shapes and poses over time. This simplicity is an advantage when modeling static motions and a reason why the myoelectric interface has widely used static motions as recognizable gestures. The proposed approach assumes that using dynamic motions are more reliable when dealing with the robustness issue, particularly limb position changes, than using static motions. Some studies used dynamic motions to expand the variety of motions that can be recognized by myoelectric signals. However, to the author’s knowledge, previous studies have not used dynamic motions for limb position changes.

In here, the term, reliability (or robustness) of myoelectric interface means that the interface recognizes human gestures without noticeable decrease in the classification accuracy against the non-stationary condition of myoelectric signals caused by the external condition changes. The term reliability (or robustness) is used widely in the literature. The authors in [5] addressed the robustness issues of myoelectric control; that is, the myoelectric system needs to be robust to limb position, electrode/skin impedance changes, and changes in EMG signals over different days of use. The robustness of a myoelectric classification method is usually shown experimentally, not mathematically. For example, the authors in [70] used additional accelerometers to tackle reducing the performance of myoelectric classification when the limb position changes. They explained the robustness via classification accuracy of their methods. Furthermore, the authors in [71] proposed EMG features for the robustness of the myoelectric classification against the impact of changing limb positions. They also showed the robustness of the method through experimental results.

Among many robustness issues, the dissertation mainly focuses on the limb position changes, which describes the drop in the performance of myoelectric classification when the limb position varies from a position where the system was previously trained. Lack of robustness related to the limb position changes is mainly caused by the discordance between what a classifier learned from the training sets and the actual EMG patterns in service. When the limb position varies from where a classifier was initially trained, the classification accuracy suffers due to this difference. The proposed method classifies dynamic motions with robustness to limb position changes. The robustness of the proposed method is evaluated by choosing the training sets and validation sets at different limb positions.

As an application of the proposed myoelectric interface, an HCI system manipulating a 6-DOF robot arm with a 1-DOF gripper is introduced. The HCI system includes an IMU sensor to measure limb orientation and eight EMG sensors to acquire muscle force and to classify dynamic motions. Varying muscle forces controls the activation modes (waiting for a dynamic motion and moving a robot arm). Gradients of a limb orientation give directions of movements to the robot arm. Classified dynamic motions are used to change the control states of the HCI system such as Arm mode (translation), Wrist mode (rotation), and Finger mode (gripper). Each dynamic motion is mapped to a specific function like buttons in a joystick.

1.3 Organization of the Dissertation

The subsequent chapters consist of an explanation and comparison of general pattern-based myoelectric classifications (Chapter II), robustness to the limb position changes in the proposed sequence-based myoelectric classification (Chapter III), the manipulation of a robot arm via IMU and EMG signals with the proposed myoelectric classification scheme (Chapter IV), and the summarization and conclusion of this dissertation (Chapter V).

In Chapter II, the comparison of various methods for traditional myoelectric classification with finger and hand motions is presented. In Chapter III, the difference between dynamic and static motions is considered. Furthermore, the proposed sequence-based myoelectric classification for classifying dynamic motions is presented and discussed. The robustness of the proposed myoelectric classification to the limb position changes was evaluated and compared to PMC. The static and dynamic motion sets were collected at four different limb positions. In Chapter IV, a myoelectric interface was developed to manipulate a 6-DOF robot arm (with a 1-DOF gripper). The proposed sequence-based myoelectric classification was used to handle robot control modes. Muscle forces and the orientation of a forearm are used to generate movement commands for the robot arm. To measure the performance of the myoelectric interface, a predefined task was repeated with five participants. Finally, Chapter V concludes the dissertation with a brief summarization of the above works and recommendation for future research in this area.

CHAPTER II

A PERFORMANCE COMPARISON OF CONVENTIONAL PATTERN-BASED MYOELECTRIC CLASSIFICATION[*]

2.1 Introduction

With increasing interests in human computer interaction, an approach using EMG signals has been widely studied because EMG signals account for human muscle activities. Several approaches, such as proportional control, threshold control, onset analysis, and finite state machines, have used EMG to reflect the human motion intents. Even though these approaches have worked properly on reflecting the human motion intent in their system, myoelectric pattern classification approaches, or pattern-based myoelectric classification (PMC), have been developed to extract more detailed information about the human motion intent from myoelectric signals [3]. Myoelectric classification is a way of classifying EMG signal patterns, which are correlated with particular human motions. Applications that can benefit from EMG pattern recognition include assistive robotics and artificial prosthetics to enhance the quality of life for amputees and the elderly [82].

The aim of this chapter is to compare well-known methods for feature extraction, dimension reduction, and classification in myoelectric classification to estimate the performance of the methods in myoelectric classification: real-time operation, acceptable

[*] Reprinted with permission from "A Performance Comparison of EMG Classification Methods for Hand and Finger Motion" by Sungtae Shin, et al., 2014 Dynamic Systems and Control Conference (DSCC), 2014 (Copyright © 2014, ASME)

accuracy, and convenience of training procedures. For comparison, not only *classification accuracy* but also *training time* as well as *validation time* were considered as the performance evaluation criteria. For the purpose of generality, two published EMG datasets (hand and finger motions) were used. In addition, three feature extraction techniques (time domain, empirical mode decomposition, and discrete wavelet transform), two dimension-reduction methods (principal component analysis and linear discriminant analysis), and six classification algorithms (naïve Bayes, k-nearest neighbor, quadric discriminant analysis, multi-layer perceptrons, support vector machines, and extreme learning machines) were adopted in this chapter.

2.2 Methods

This section describes public gesture EMG datasets (hand & finger motion EMG datasets) to test the performance of methods used in the study. Moreover, the methods (feature extraction, dimensionality reduction, and classification techniques) are explained.

2.2.1 EMG Gesture Datasets

In order to evaluate the performance of methods used for PMC, two public EMG gesture datasets are chosen in this study. One includes 10 hand-motions and the other one contains 15 finger-motions, as detailed next.

Hand Motion EMG Dataset (HED)

The *"10mov4chUntargetedForearm"* dataset from BioPatRec [83] was chosen for hand-motion classifications. The dataset includes ten hand *motion classes*, shown in Figure 3, measured by four bipolar EMG electrodes, which were located around the forearm. Each motion had a 3-second contraction time and a 3-second relaxation time, and was repeated three times. In total, twenty subjects participated in the process.

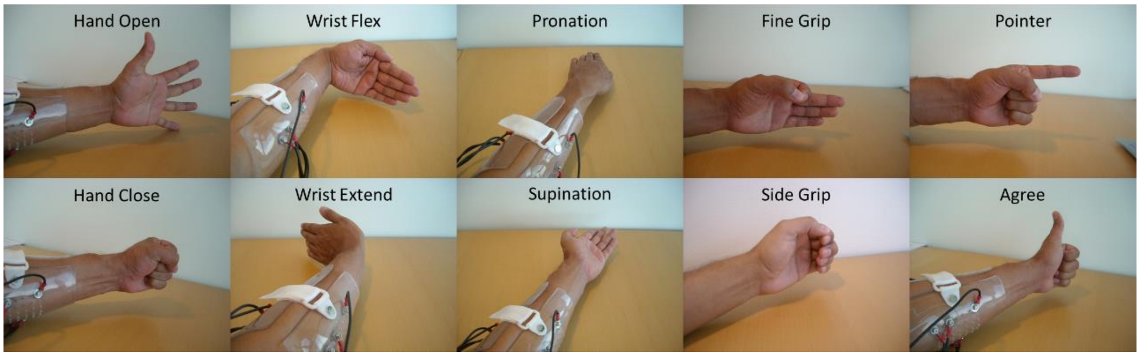


Figure 3 Hand and wrist motions in BioPatRec dataset [83]

Finger Motion EMG Dataset (FED)

For finger-motion classifications, the Khushaba and Kodagoda dataset [26] was used. This dataset contains fifteen individual and combined finger motion classes measured by eight electrodes which were placed across the circumference of the forearm. These motions are depicted in Figure 4. Each motion was held for 20 seconds and recurred three times. The first 5-second data of each motion was used in our study. Eight subjects were involved in this dataset.

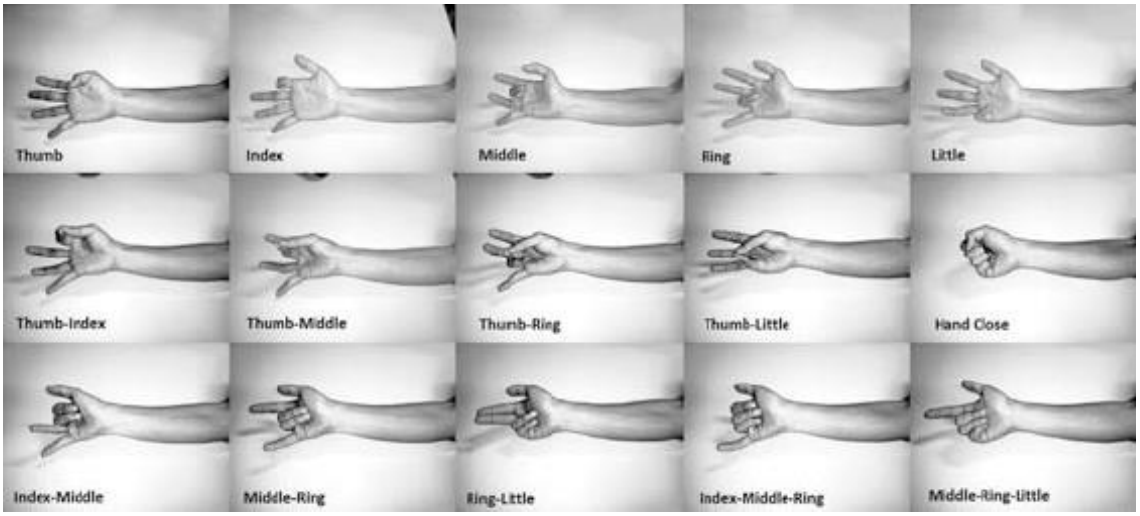


Figure 4 Fifteen finger motions in Khushaba and Kodagoda dataset [26]

2.2.2 Feature Extraction

This section introduces the methods of feature extraction studied in this work. Feature extraction has an important role in the performance of the classification algorithm. Strictly speaking, raw EMG signals appear noise-like; in fact, it is hard to capture meaningful features directly from the raw EMG signal. Feature extraction is a sort of translation process from raw EMG signals, which seem to be meaningless, to meaningful features, which represent the muscle activities. Therefore, extracting representative features helps to enhance the performance of the classification algorithm.

Time-domain features were used to extract informative feature vectors from the EMG signals. In addition, signal analysis methods were used to expand the possible feature space (discrete wavelet transform and empirical mode decomposition).

Time Domain Features

Eight different time-domain features were used here: Mean Absolute Value, Root Mean Square, Integrated EMG, Waveform Length, Zero Crossing, Slope Sign Change, Skewness, and sixth-order auto-regressive coefficients. Their equations are presented in Table 1. The dimension of the feature set is totally thirteen. This feature set will be called here as TDAR6.

Table 1 Time domain features[84][84][84]

Feature Extraction Name	Definition
Mean absolute value	$\text{MAV} = \frac{1}{N}\sum_{i=1}^{N} \lvert x_i \rvert$
Root mean square	$\text{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Integrated EMG	$\text{IEMG} = \sum_{i=1}^{N} \lvert x_i \rvert$
Waveform length	$\text{WL} = \sum_{i=1}^{N-1} \lvert x_{i+1} - x_i \rvert$
Zero crossing	$\text{ZC} = \sum_{i=1}^{N-1} [sgn(x_i \times x_{i+1}) \cap \lvert x_i - x_{i+1} \rvert \geq th]$
Slope sign change	$\text{SSC} = \sum_{\text{i}=2}^{N-1} [sgn((x_i - x_{i+1}) \times (x_i - x_{i+1})) \cap (\lvert x_i - x_{i+1} \rvert$ $\geq th \cup \lvert x_i - x_{i-1} \rvert \geq th)]$
Skewness	$\text{SKW} = \frac{\frac{1}{N}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{N}\sum_{i=1}^{n}(x_i - \overline{x_\imath})^2\right)^{\frac{3}{2}}}$ ($\bar{x}$ is sample mean)
Auto-regressive coefficients (a_i)	$x_k = \sum_{i=1}^{P} a_i x_{k-i} + e_k$ (a_i: AR coefficients, e_k: an error term, P: the order of AR)
N: the size of samples $th =$ threshold	$sgn(x) = \begin{cases} 1, if\ x \geq th \\ 0, othewise \end{cases}$

Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) is a wavelet transform to deal with discrete data. Wavelet transform analyzes both time and frequency information of the given data (unlike Fourier transform which focuses on the frequency domain). This property offers a significant advantage for EMG signals because they are usually *non-stationary*, which means that their properties changes over time.

In general, DWT decomposes a signal into a group of *approximation coefficients* and a group of *detail coefficients*, by using a low-pass filter and a high-pass filter, respectively, as shown in Figure 5. From each group of coefficients one can reconstruct the original signal by Inverse DWT.

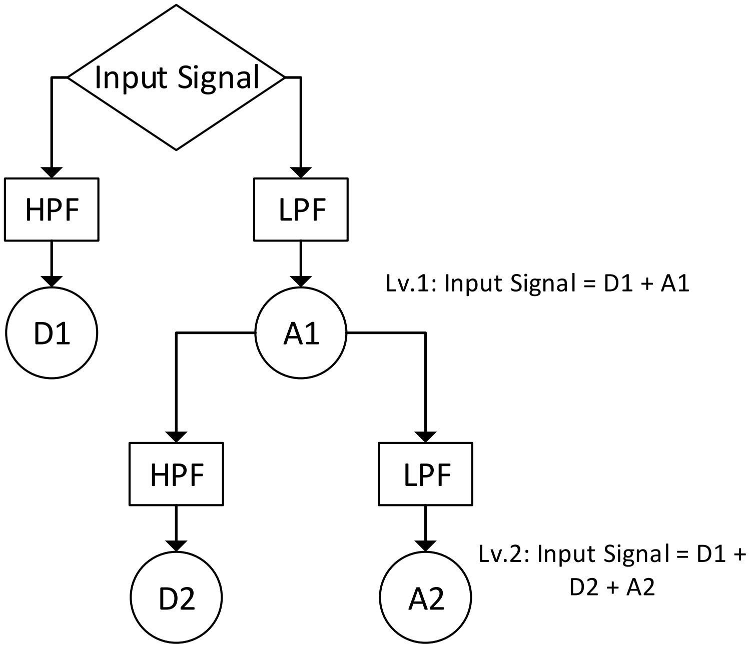

Input Signal
HPF
LPF
D1
A1
Lv.1: Input Signal = D1 + A1
HPF
LPF
D2
A2
Lv.2: Input Signal = D1 + D2 + A2

Figure 5 Schematic diagram of DWT (HPF: high pass filter, LDP: low pass filter, Dx: detail coefficients of level x, Ax: approximation coefficients of level x)

A fifth order Symlets wavelet with four decomposition levels ($\mathrm{DL} = 4$) were used to analyze EMG signals. After the transformation, the last approximation coefficient, $\{\mathrm{cA}_4\}$, and four detail coefficients, $\{\mathrm{cD}_1, \mathrm{cD}_2, \mathrm{cD}_3, \mathrm{cD}_4\}$, were calculated. These five coefficients were separately reconstructed by inverse DWT. With the reconstructed signal set from the coefficients, $\mathrm{X} = \{\mathrm{x}_i, 1 \leq i \leq \mathrm{DL} + 1\} = \{\mathrm{A}_4, \mathrm{D}_1, \mathrm{D}_2, \mathrm{D}_3, \mathrm{D}_4\}$, each reconstructed signal was used to extract a total of nine features (the eight time-domain features and Relative Signal Energy (RSE)). RSE is the ratio of energy of a signal at a certain frequency band to the total energy of the signal, and is inspired by the relative wavelet packet energy [85]. RSE is defined by:

$$\mathrm{RSE}_i = \frac{\sum_{\mathrm{k}} |\mathrm{x}_i(\mathrm{k})|^2}{\sum_{\mathrm{n}=1}^{\mathrm{DL}+1} \sum_{\mathrm{k}} |\mathrm{x}_{\mathrm{n}}(\mathrm{k})|^2} \tag{1}$$

where i is the reconstructed signal index and k is the number of samples of the reconstructed signal x_i.

Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) [18] is also a tool to analyze non-stationary signals. Input signals (X) can be decomposed by EMD into a group of intrinsic mode functions (IMFs) and a residue, as shown in Equation (2). These IMFs contain contents of the signal in different frequency bands.

$$\mathrm{X}(\mathrm{t}) = residue(t) + \sum_{i=1}^{n} \mathrm{IMF}_i(t) \tag{2}$$

The EMD process produces IMFs in the ascending frequency order, which is called *sifting*. This process can be summarized as follows:

1. Find all the local maxima and minima in the input signal.

2. Link all the local maxima as the upper envelope.
3. Link all the local minima as the lower envelope.
4. Calculate their mean between the upper envelope and the lower envelope.
5. Subtract the mean (m) from the input signal (X). The result becomes an IMF as shown in Equation (3).
6. For the next IMF, the mean $\mathrm{m}_1(t)$ becomes the next input signal. Now, repeating the above procedure with the next input signal, the next IMF can be calculated.
7. This procedure will be stopped when the stoppage criterion (SD, Equation (4)) [18] is smaller than a pre-defined value.

$$\mathrm{X(t)} - \mathrm{m}_1(t) = \mathrm{IMF}_1(t) \tag{3}$$

$$\mathrm{SD_k} = \frac{\sum_{t=0}^{T} |h_{k-1}(t) - h_k(t)|^2}{\sum_{t=0}^{T} h_{k-1}^2(t)} \tag{4}$$

In general, most valuable information in the EMG signal is located between 20Hz and 500Hz [86]. With this in mind, the first four IMFs and the residue were chosen to calculate features. The time-domain features and RSE were extracted from the IMFs and the residue, respectively.

2.2.3 Dimension Reduction

In order to increase the classification accuracy in the EMG pattern recognition, the number of EMG electrodes and the size of extracted features from raw EMG signals are usually increased. However, with increasing the size of the data, the computational time and power also increases. To remedy this problem, dimension reduction is indispensable in EMG pattern recognition.

In this study, two well-known and basic dimension reduction techniques, i.e., PCA [87] and LDA [88], also known as Fisher's linear discriminant, were adopted. In general, PCA maximizes the representation of data by maximizing the variance of the data. Likewise, LDA maximizes the separation of the data using the class label in a supervised learning manner, as shown in Figure 6. The classification accuracy results with dimension reduction were compared to the results without the dimension reduction.

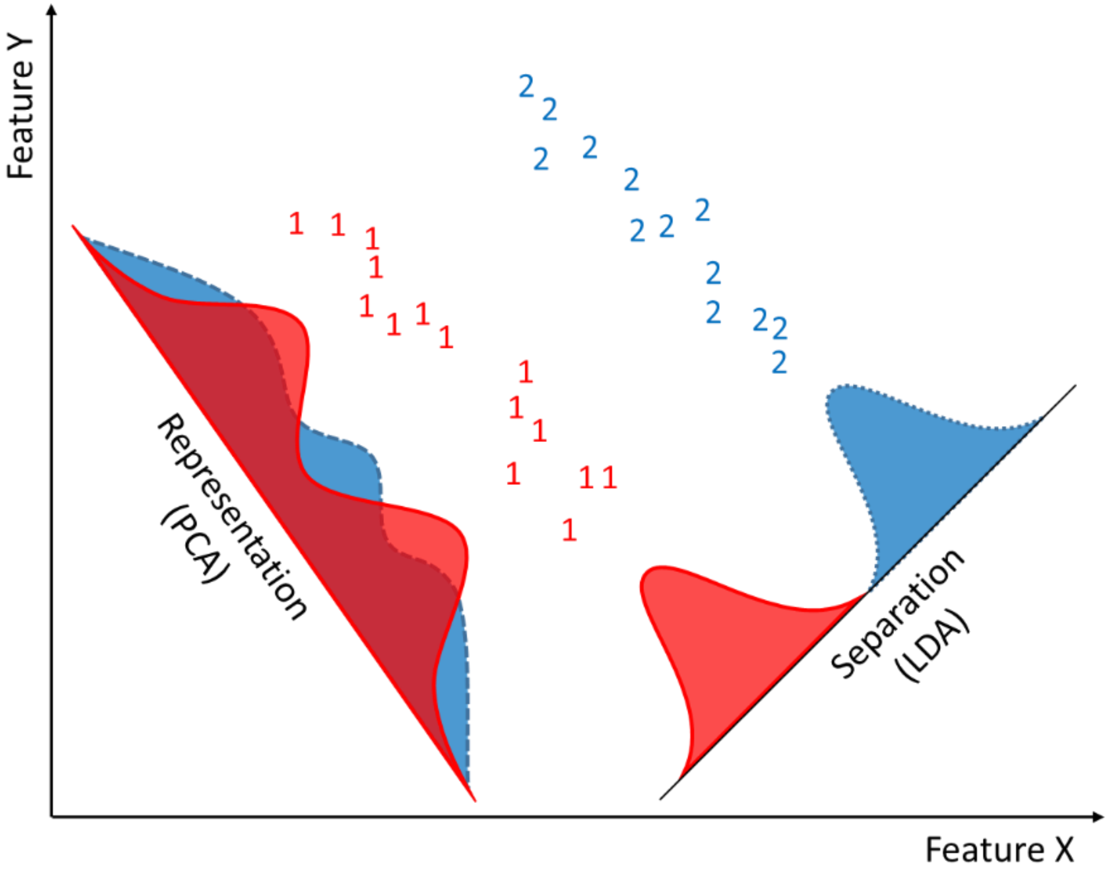


Figure 6 PCA vs. LDA

2.2.4 Classification

In this study, six classification methods were chosen to classify EMG gesture. The six methods (Naïve Bayes, K-Nearest-Neighbor, Quadratic Discriminant Analysis, Multi-Layer Perceptron, Support Vector Machine, and Extreme Learning Machines) are briefly explained next.

Naïve Bayes

Naïve Bayes (NB) is a statistical classifier using Bayes' theorem with an assumption that random variables are statistically independent of each other. NB can be modeled as the posterior probability, $\mathrm{p}(\mathrm{C}|\mathrm{X}_1, \dots, \mathrm{X}_\mathrm{n})$ with a class variable C and input variables X_1 through X_n. The posterior probability is proportional to the product of the prior probability, $p(C)$, and the likelihood probabilities, $\prod_{i=1}^{n} p(X_i|C)$, according to Bayes' theorem and the independence assumption:

$$\mathrm{p}(\mathrm{C}|\mathrm{X}_1, \dots, \mathrm{X}_\mathrm{n}) = \frac{p(C)p(X_1, \dots, X_n|C)}{p(X_1, \dots, X_n)} \propto p(C) \prod_{i=1}^{n} p(X_i|C) \qquad (5)$$

NB finds the prior probabilities and the likelihood probabilities from training data during the training procedure. Then, for classification, NB decides a target class, which maximizes the posterior probability with input variables.

K nearest neighbor

K nearest neighbor (KNN) classifies input features as finding the K nearest neighbors from its repository and then, determining the appropriate class from among these neighbors. The schematic explanation of KNN is described well in Figure 7. Due to the nature of KNN, it does not need training, but requires relatively more computational effort in the classification stage. For the studies, the k parameter was adopted as five by cross validation.

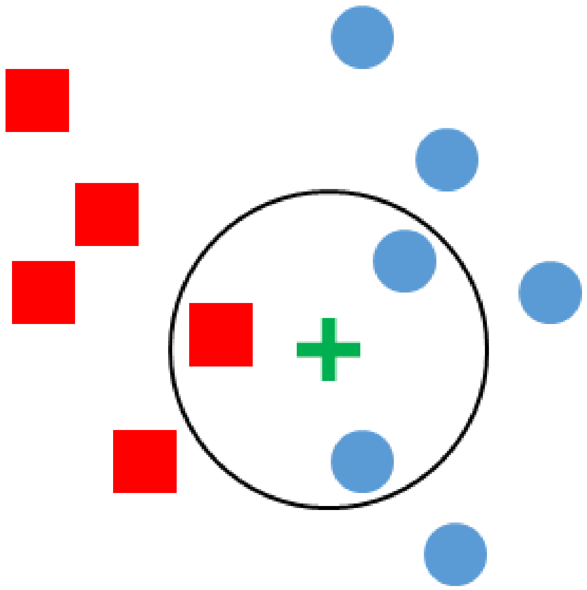


Figure 7 KNN classification (k=3, the green cross is classified as a blue circle)

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a Gaussian model based statistical classification method. This means that QDA assumes all observations are normally distributed and these observations can be modeled as a Gaussian distribution with mean (μ) and covariance (Σ). Furthermore, QDA assumes that every class has a different mean and covariance. Then, a quadratic discriminant function can be expressed as Equation (6), where x is the input variable, μ_i is the mean of class i, Σ_i is the covariance of class i, $P(\omega_i)$ is the prior probability of class i.

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i) - \frac{1}{2}\log|\Sigma_i| + \log P(\omega_i) \qquad (6)$$

The discriminant function (DF) of an observation x within class i, $g_i(x)$, is calculated for all classes. The calculated DFs are used for the decision rule (assigning x to class i if $g_i(x) > g_j(x)\ \forall j \neq i$).

Multi-Layer Perceptron

Multi-Layer Perceptron (MLP) is a feedforward neural network having multiple layers (one input and one output layer, and one or more hidden layers) as shown in Figure 8. Each perceptron (σ) is calculated by Equation (7), a weight from i to j (w_{ij}) and bias (b), activation function (ϕ).

$$y_j = \phi(\sum_i w_{ij} x_i + b) \qquad (7)$$

Single-hidden layer feedforward network (SLFN), a type of MLP, was used for the experiment. The size of neurons in the hidden layer was 15 in this study. The size was determined by cross validation technique. For the activation function of the hidden and output layers, the tan-sigmoid function is chosen. Furthermore, Levenberg-Marquardt back-propagation algorithm was considered for the training of the network.

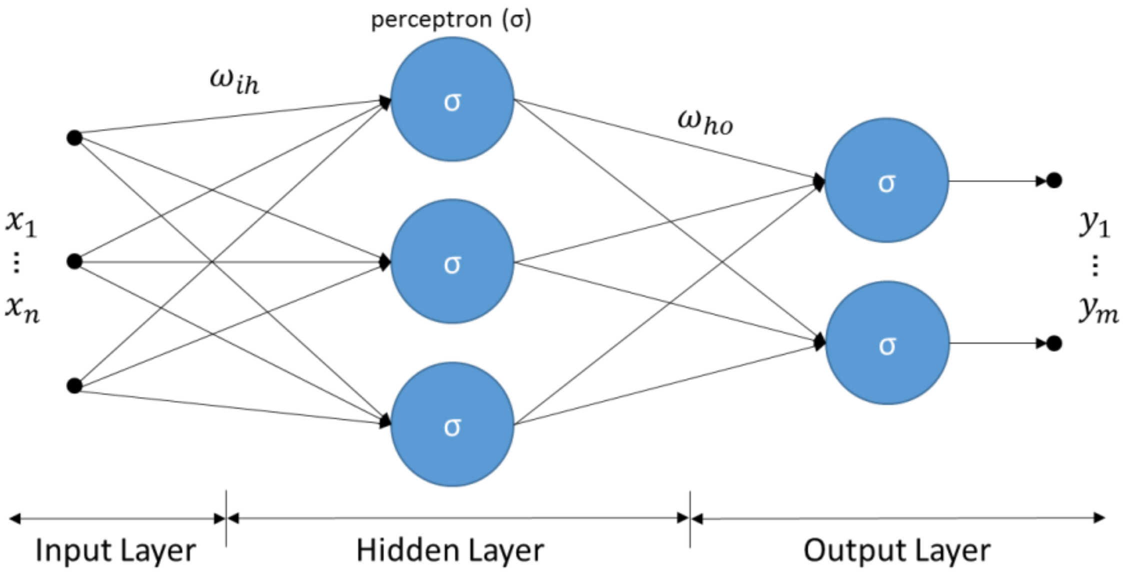


Figure 8 Multi-layer perceptron

Support Vector Machine

Support Vector Machine (SVM) is a discriminative model that generates *hyperplanes* based on *support vectors* to classify unobserved variables, as shown in Figure 9. The goal of SVM is to maximize a margin, which defined as the distance between two hyperplanes. This optimization problem can be formulated as:

$$L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j k(x_i, x_j)$$
$$\text{Subject to } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0 \qquad (8)$$

where α is the Lagrange multiplier, N is the number of samples, y is a class label (-1 or 1), x is a feature vector, C is the soft margin parameter, which is the margin allowing a certain degree of misclassification in SVM, and k(·) is the kernel function representing the nonlinearity. With an assumption that samples are normally distributed, Gaussian

Radial Basis Function (RBF) shown in Equation (9) was used as the kernel function, where γ is the parameter for adjusting RBF.

$$\mathrm{k}(\mathrm{x}_\mathrm{i}, \mathrm{x}_\mathrm{j}) = \exp\left(-\gamma\left\|\mathrm{x}_\mathrm{i} - \mathrm{x}_\mathrm{j}\right\|^2\right), \text{for } \gamma > 0 \qquad (9)$$

To solve this optimization problem, libSVM package [89] was used. Basically, SVM find a hyperplane separating two classes (*binary classification*). However, this study focuses on a problem classifying multiple gestures (*multiclass classification*). To solve the multiclass classification problem via multiple binary classification approaches, *one-versus-the-rest* approach [90] was implemented. The parameters, C (32) and γ (0.0313), were found by the grid search approach [89].

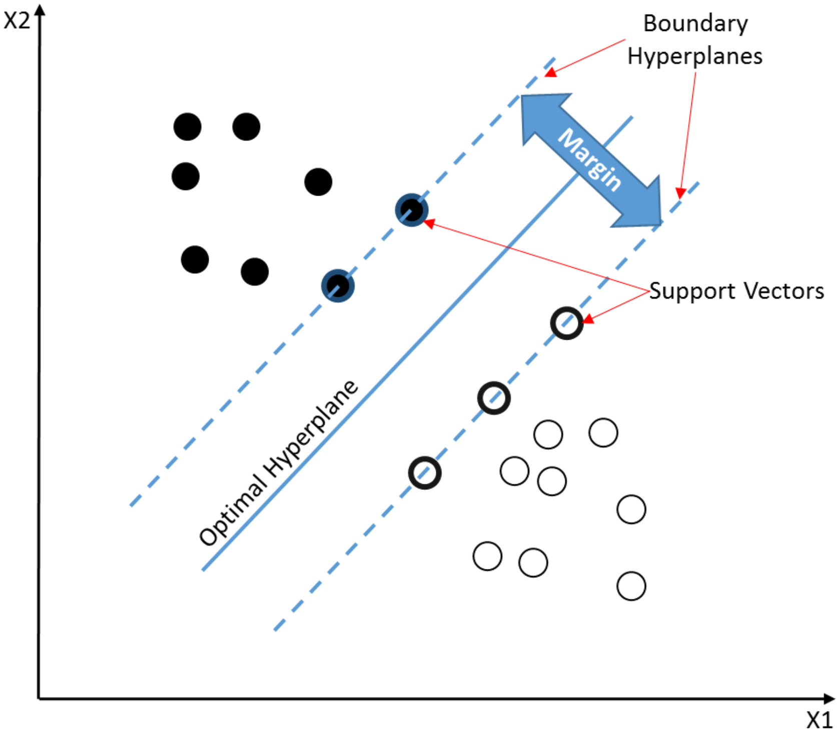


Figure 9 Support vector machine conceptual schema; the sold line is the optimal hyperplane separating two classes (solid circles and empty circles) with the maximum margin, and samples on the boundary hyperplanes (the dot lines) are called the support vectors (the bold circles).

Extreme Learning Machines

Extreme Learning Machines (ELM), introduced by Huang [91], is a learning algorithm for single-hidden layer feedforward network (SLFN). Its distinguishable property is that it does not need to be trained iteratively unlike other gradient-based learning algorithms for multilayer perceptrons. This is an important advantage for real-time operation.

For given train sets $\mathrm{T} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ where $\mathbf{x}_\mathrm{i} = [\mathrm{x}_{i1}, \dots, \mathrm{x}_{in}\ ]^\mathrm{T} \in \mathbf{R}^n$ and $\mathbf{y}_i = [\mathrm{y}_{i1}, \dots, \mathrm{y}_{im}\ ]^\mathrm{T} \in \mathbf{R}^m$, SLFN with the number of neurons in the hidden layer, L, can be expressed as:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_\mathrm{i} g\big(\mathbf{w}_i \cdot \mathbf{x}_j + \mathrm{b}_i\big) = \mathbf{y}_j\ , j = 1, \dots, N \qquad (10)$$

where $\mathbf{w}_i = [\mathrm{w}_{i1}, \dots, \mathrm{w}_{in}]^\mathrm{T}$ is the weight vector between the ith hidden node and the input nodes, $\boldsymbol{\beta}_i = [\beta_{i1}, \dots, \beta_{im}]^\mathrm{T}$ is the weight vector between the ith hidden node and the output nodes, b_i is the bias for the ith hidden node, and $g(\cdot)$ is the activation function (e.g., sigmoid function). The number of neurons in the hidden layer, L, was 100 in this study and it was determined by cross validation technique. The polynomial form of Equation (10) can be rewritten in a matrix form:

$$\mathbf{H\beta} = \mathbf{Y},$$

where

$$\mathrm{H}(\mathbf{w}_1, \dots, \mathbf{w}_L, \mathrm{b}_1, \dots, \mathrm{b}_L, \mathbf{x}_1, \dots, \mathbf{x}_N) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + \mathrm{b}_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + \mathrm{b}_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + \mathrm{b}_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_\mathrm{N} + \mathrm{b}_L) \end{bmatrix}_{N\times L}, \quad (11)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^\mathrm{T} \\ \vdots \\ \beta_L^\mathrm{T} \end{bmatrix}_{L\times m}, \text{and } \mathbf{Y} = \begin{bmatrix} \mathrm{y}_1^\mathrm{T} \\ \vdots \\ \mathrm{y}_N^\mathrm{T} \end{bmatrix}_{N\times m}$$

From this matrix notation, ELM algorithm trains the SLFN structure as follows:

1. Assign randomly the hidden weights ($\mathbf{w}_i$) and the hidden node bias (b_i), *i*=1,…,*L*
2. Calculate the hidden layer output matrix **H**
3. Calculate the output weight matrix $\widehat{\boldsymbol{\beta}} = \mathbf{H}^\dagger\mathbf{Y}$
4. Where $\mathrm{H}^\dagger$ is the Moore-Penrose generalized inverse of the hidden layer output matrix **H.**

2.3 Results

This section shows the evaluation results in terms of classification accuracy, training time, and classification (or prediction) time. For training and classification time, only the results of TDAR6 feature set are stated because TDAR6 feature set showed the best classification accuracy. The window size used in the segmentation is 256ms length and 128ms increment. Finally, statistical analysis of the results is described. The size of the training and test sets in the hand motion dataset were 440 and 220, respectively. In the finger motion dataset, the size of the training set was 1140, and the size of the test set was 570.

2.3.1 Classification Accuracy

The results in this section shows the classification accuracy of the myoelectric classification methods for the hand motion and the figure motion datasets.

Hand Motion

The effects of different dimension reductions and feature extractions in the hand motions are analyzed. The results of the dimension reduction methods with TDAR6 feature set are depicted in Figure 10. The error bars in the figure display the standard deviation of the total 20 subjects with 3 trials of each hand motion. As one can see in the figure, LDA shows higher accuracies regardless of the classification methods. The highest accuracy in the results is 88.9% of NB with LDA. The lowest accuracy was 67.5% of QDA with *No-DR* (no dimension reduction). In SVM and MLP, the accuracy of both LDA and No-DR were similar. The other results indicated that PCA did not show better performance than No-DR.

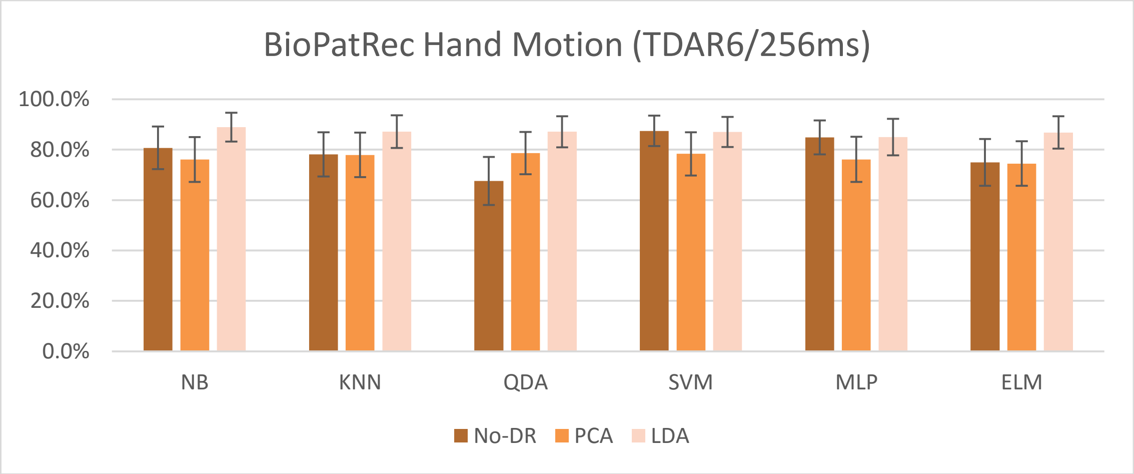


Figure 10 Classification accuracy of the BioPatRec hand motion dataset without DR

The results with the different feature extractions are displayed in Figure 11. In this figure, the results of applying LDA was only displayed because LDA outperformed PCA in the overall results. Moreover, MLP did not work without dimension reduction due to "out of memory" problem. As shown in Figure 11, TDAR6 were more accurate than EMD and DWT in the all results. The results of EMD and DWT were very similar.

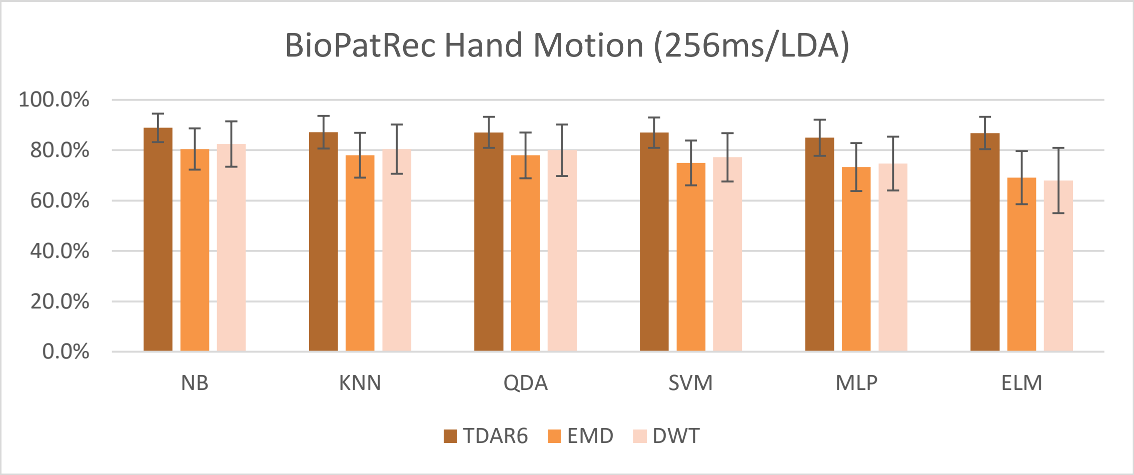


Figure 11 Classification accuracy of the BioPatRec hand motion dataset with LDA

Finger Motion

The results of dimension reductions and feature extractions in the finger motions were analyzed. Furthermore, the results were compared with the hand motion results. The effects of different dimension reduction methods can be seen in Figure 12. Much like the results of HED, LDA performed well in FED compared to PCA and No-DR. The results of LDA showed analogous accuracies regardless of the classification method. Although PCA showed similar accuracy results with the different classifiers, LDA was more accurate than PCA. The results of No-DR seemed to depend on the classification method. Moreover, the classification accuracy of all classification methods with LDA were over 80%. However, the accuracies of other methods were less than this except for SVM and MLP with No-DR.

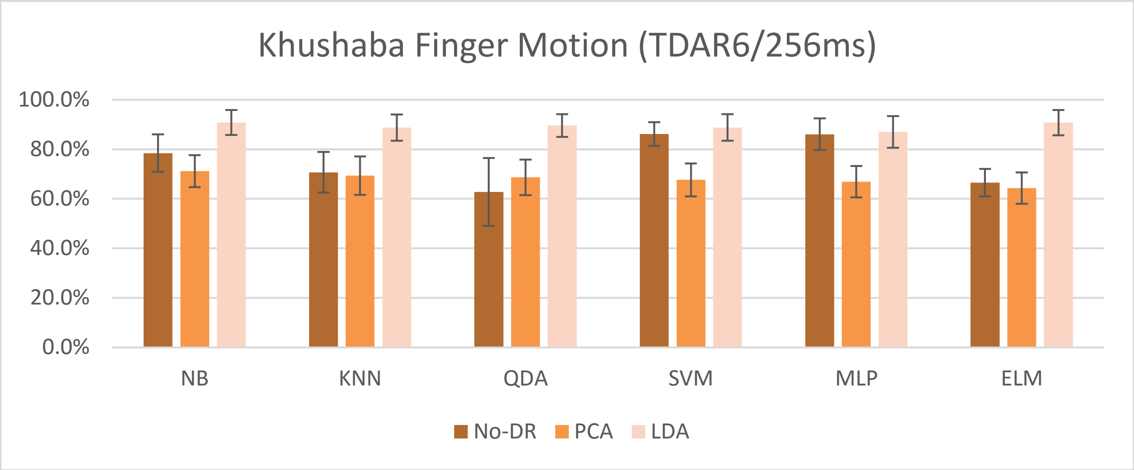


Figure 12 Classification accuracy of the Khushaba finger motion dataset without DR

We can see the comparison of the different feature extraction schemes in Figure 13. Similar to the hand motion results, TDAR6 achieved better results than EMD and DWT. One point to consider is that the results of the finger motions are more accurate than the results of the hand motions.

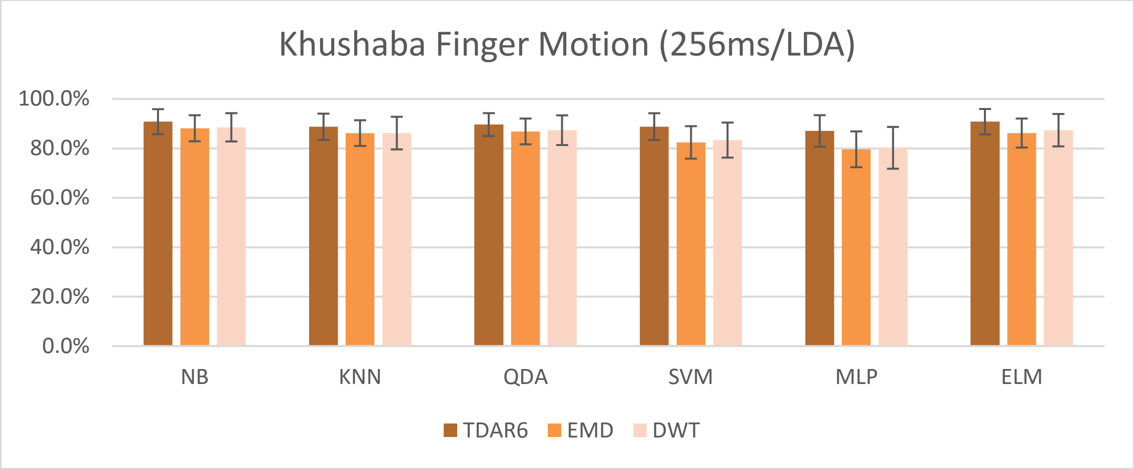


Figure 13 Classification accuracy of the Khushaba finger motion dataset with LDA

2.3.2 Training Time

This section discusses the training time of the myoelectric classification methods within the hand motion and the figure motion datasets.

Hand Motion

The training time of the hand motion studies with the TDAR6 feature set were evaluated to compare their performance for different classification methods and dimension reduction techniques. Here, we only considered TDAR6 feature set because it showed less computational effort and more accuracy in classification. The results are tabulated in Table 2. As one can see, the training time with the dimension reductions were reduced considerably compared to No-DR because dimensionality reduction decreases the size of feature space a classifier needs to learn. Nevertheless, MLP needed the longest training time in all cases as it uses the time-consuming backpropagation training. The repeated backpropagation training needs more time to propagate training data over a multi-layer neuron network. For this heavy training process, the dimensionality reduction narrowing searching scopes of the feature space plays a vital role in decreasing the training time of the MLP. Moreover, the advantage of using dimensionality reduction is also found in SVM. The reduced feature space is advantageous for solving the optimization problem of SVM in the training process.

Table 2 Training time of the BioPatRec hand motion dataset with TDAR6

Training Time
(BPR/TDAR6/256ms)

		DR		
		None	**PCA**	**LDA**
Classification	**NB**	0.005s	0.003s	0.003s
	KNN	0.007s	0.007s	0.007s
	QDA	0.098s	0.083s	0.084s
	SVM	0.014s	0.006s	0.003s
	MLP	18.075s	2.612s	2.178s
	ELM	0.005s	0.005s	0.005s

Finger Motion

Table 3 shows the results of training time in finger motion studies according to the classification methods and the dimension reduction techniques. Because of the different sizes of training sets (the training size of the finger motions was about two times more than of the hand motions), their training time would not be same, but, their trend is co-related. As one can see in the hand and finger motion studies, MLP and SVM need dimension reduction for practical real-time applications. Furthermore, LDA was more effective than PCA in SVM with regard to the training time. As described in Figure 6, while PCA maximizes the variance of the data, LDA aims to maximize class-separability on a lower feature dimensional space. The pre-processed separation of data relaxes the complexity of the optimization problem of SVM. This causes to decrease the training time of SVM when using LDA compared to PCA.

Table 3 Training time of the Khushaba finger motion dataset with TDAR6

Training Time
(Khu/TDAR6/256ms)

		DR		
		None	**PCA**	**LDA**
Classification	**NB**	0.011s	0.004s	0.004s
	KNN	0.007s	0.006s	0.006s
	QDA	0.241s	0.173s	0.173s
	SVM	0.191s	0.038s	0.014s
	MLP	366.271s	14.521s	30.086s
	ELM	0.007s	0.006s	0.006s

2.3.3 Classification Time

This section discusses the classification (prediction) time of the myoelectric classification methods within the hand motion and the figure motion datasets.

Hand Motion

The classification time of a classifier is the time needed for classifying the validation set. It is an important factor for real-time operation of a classifier. Table 4 displays the results of the classification time of hand motions. From the results, the effectiveness of the dimension reduction could be verified to be similar to the training cases. In addition, we can see that SVM with LDA showed better performance than with PCA. Moreover, NB and ELM resulted in faster classification times. All said, the classification time of MLP was the largest in all cases. The size of the validation set in HED was 220.

Table 4 Classification time of the BioPatRec hand motion dataset with TDAR6

Classification Time
(BPR/TDAR6/256ms)

		DR		
		None	**PCA**	**LDA**
Classification	**NB**	0.005s	0.003s	0.004s
	KNN	0.006s	0.002s	0.002s
	QDA	0.004s	0.003s	0.003s
	SVM	0.005s	0.002s	0.001s
	MLP	0.006s	0.006s	0.006s
	ELM	0.001s	0.001s	0.001s

Finger Motion

The results of the classification time in the finger motions is presented in Table 5. The size of the validation set in FED was 570. There is no remarkable difference between the results of the hand and finger motions with regard to the classification time. The only difference is that the classification time of the finger motion is longer than of the hand motion because the number of the validation set of the finger motion is larger than of the hand motions.

Table 5 Classification time of the Khushaba finger motion dataset with TDAR6

Classification Time
(Khu/TDAR6/256ms)

		DR		
		None	**PCA**	**LDA**
Classification	**NB**	0.015s	0.006s	0.006s
	KNN	0.053s	0.012s	0.012s
	QDA	0.011s	0.005s	0.005s
	SVM	0.057s	0.020s	0.009s
	MLP	0.008s	0.006s	0.006s
	ELM	0.002s	0.002s	0.001s

2.3.4 Statistical Analysis

Statistical analysis is conducted to determine whether 1) the classification accuracies of classifiers in three cases of using No-DR, PCA, and LDA are significantly different; and 2) the computational times (training and classification times) of the three cases (using No-DR, PCA, and LDA) are significantly different. For statistical analysis of the results, one-way ANOVA (analysis of variance) was conducted in HED and FED. The null hypothesis $\{\mathcal{H}_0\colon \mu_1 = \mu_2 = \cdots = \mu_n\}$, which means that the population means, $\forall \mu_i$ for $1 \leq i \leq n$, of the results of n different groups are the same, is rejected with 95% significance level (p-value < 0.05).

First of all, the effect of dimension reduction in the classification accuracy was analyzed with TDAR6 feature set. The classification accuracies of the six classifiers (NB, KNN, QDA, SVM, MLP, and ELM) with No-DR are significantly different in the studies (p=0.0 (HED) and p=0.0 (FED)). With PCA, there are no significant differences (p=0.63 (HED) and p=0.46 (FED)). With LDA, there are also no significant differences (p=0.56

(HED) and *p*=0.72 (FED)). This analysis means that all classification methods shows similar performance in the classification accuracy after applying dimensionality reduction.

Next, for the analysis of the effect of dimension reduction in the computational times, the dimension reduction affects the computational times because it reduces the size of feature vectors that need computational calculations. However, the effect of using dimension reduction varies based on the characteristic of classification algorithm. Interestingly, NB, KNN, QDA, and ELM have no meaningful differences between the results of using LDA and PCA in HED and FED. NN has differences, but their tendencies in HED and FED are not consistent. However, SVM has a consistent trend in HED and FED; LDA outperformed PCA in all cases (the training and classification times in HED and FED). The training and classification times between PCA and LDA with SVM classifications have significant differences, as shown in Table 6. A possible explanation about this trend is that LDA projection makes the optimization problem of SVM easier than what PCA does because LDA projection maximizes class-separability on a lower feature dimensional space.

Table 6 Training and classification times of SVM and NN in HED and FED

			SVM			***NN***		
			No-DR	**PCA**	**LDA**	**No-DR**	**PCA**	**LDA**
Hand	Training Time	Mean	14.4ms	6.0ms	3.4ms	18.075s	2.612s	2.178s
		Std	1.9ms	1.0ms	0.5ms	13.201s	1.218s	0.875s
	Classification Time	Mean	5.3ms	1.9ms	1.0ms	6.2ms	6.1ms	6.2ms
		Std	0.7ms	0.4ms	0.0ms	0.4ms	0.2ms	0.5ms
Finger	Training Time	Mean	190.5ms	37.9ms	13.5ms	366.27s	14.52s	30.09s
		Std	37.9ms	2.3ms	1.2ms	144.62s	7.15s	16.56s
	Classification Time	Mean	57.0ms	19.9ms	9.0ms	7.8ms	6.1ms	6.0ms
		Std	6.1ms	1.0ms	0.8ms	0.7ms	0.4ms	0.0ms

2.4 Discussion

The aim of this chapter was to compare the performance of methods used for EMG pattern recognition. The basic belief is that we can identify proper methods for real-time applications from these comparisons. Many papers have showed their methods performed well with their datasets [30, 34, 40, 41, 44]. Nevertheless, we cannot be sure without having a comparison upon the same dataset how well these methods perform, and which methods we have to select for our application. A number of review papers [3, 63] have surveyed the classification accuracies of previous researches as a comparable factor. However, these did not consider practical performances such as real-time operation and short training for user-convenience.

Considering only the classification accuracy is insufficient to analyze the performance of methods for the practical applications. To compare the performance of different methods, we chose three performance criteria: the classification accuracy, the training time, and the classification time. Because considering only the classification accuracy seems to be insufficient to analyze the performance of methods for the practical applications. Among three main categories in the EMG pattern recognition such as feature extraction, dimension reduction, and classification, various methods used in the literature were considered in this study.

In grouping *weak-load classifiers* (NB, KNN, QDA, and ELM) and *heavy-load classifiers* (SVM and MLP) in terms of the required computational power, one finding is that the heavy-load classifiers performed better in the classification accuracy than the weak-load classifiers in the results without dimension reduction but with the cost of computational time. However, with an appropriate DR, the weak-load classifiers performed better or equal to the heavy-load classifiers without the need to sacrifice computational time. This result seemed to show that DR methods are indispensable for EMG pattern recognition. This is why researchers have developed various DR methods [23, 24].

Moreover, LDA showed better performance in terms of classification accuracy than PCA because LDA maximizes the separation of classes in dimensionality reduction. However, in the results of computational times (both the training and classification times), NB, KNN, QDA, and ELM showed similar training and classification times, regardless of LDA and PCA. In contrast, when using SVM as a classifier, LDA outperformed PCA in both the training and classification times. A possible explanation of this result is that LDA makes optimization problems of SVM simpler. This advantage decreases the computational times of SVM.

In the study, two signal processing techniques, EMD [58] and DWT [17], were used to increase the number of EMG features. The improvement of the classification accuracy because of this increase was also examined. The results showed that increasing the feature space did not improve the performance which means that the time domain features seems to be enough for EMG pattern recognition. In addition, statistical analysis showed that the EMD and DWT were not significantly different.

2.5 Conclusion

EMG pattern recognition technique has immeasurable potentials and numerous applications which can change our life style and assist disabled or elderly people who need assistive-robot technology for enhancing their activities of daily living. This chapter aimed to compare the performance of the well-known methods. This study categorized the methods in three different sections: feature extractions, dimension reductions, and classification algorithms. Furthermore, the performance of different combinations were evaluated by three performance criteria: classification accuracy, training time, and classification time. TDAR6 showed the best performance in terms of classification accuracy and computational efforts compared to other feature sets. Also, it is confirmed that a dimension reduction technique is necessary for the increased size of feature space and the augmented number of electrode channels. Another notable finding is that after applying dimension reduction, heavy-load algorithms do not have any benefits related to weak-load algorithms; in fact, the weak-load algorithms showed better training/classification time with similar classification accuracies compared to the heavy-load algorithms. In conclusion, in a number of possible methods for EMG pattern recognition, the weak-load classification algorithms, less-computational and less-complex features, and appropriate dimension reduction methods are suitable for the real application, rather than heavy, high computational approaches.

CHAPTER III

ROBUSTNESS OF SEQUENCE RECOGNITION-BASED MYOELECTRIC CONTROL IN THE LIMB POSITION EFFECT*

3.1 Introduction

Myoelectric classification is a technique for discriminating a pattern of myoelectric signals caused by muscular movements. This technique has been widely used for manipulating prosthetic robots and interfacing a device for HCI. In fact, it is believed that myoelectric signal, also called electromyography (EMG) signal, is an appropriate bio-signal to infer human gestures as it carries information of muscular activities causing human motions [1]. With well-developed pattern recognition techniques and an assumption that myoelectric signals produced by similar human muscular activities are similar as well [5], pattern-based myoelectric classification, PMC, has been able to classify human motions. However, there exists a robustness issue when a limb position is changed from where it was trained. This issues is called as the limb position effect or the limb position changes that decreases the classification accuracy [5].

In order to improve the robustness of myoelectric classification—especially, against the limb position effect—we employ dynamic motions. The hypothesis is that dynamic motions are more robust to the limb position effect than static motions. When the

* Reprinted with permission from "Robustness of Using Dynamic Motions and Template Matching to the Limb Positon Effect in Myoelectric Classification" by Sungtae Shin, et al., ASME Journal of Dynamic Systems, Measurement, and Control, 2016 (Copyright © 2016, ASME) and "Myoelectric pattern recognition using dynamic motions with limb position changes" by Sungtae Shin, et al., 2016 American Control Conference (ACC), 2016 (Copyright © 2016, IEEE)

limb position alters from where PMC was initially trained, patterns of static motions changes from what PMC remembers because the configuration of muscles is transformed as the limb position shifts. However, traits of relative events (peak, increment, and decrement) within a dynamic motion is less affected by the limb position changes because changing the configuration of muscles does not mean that the relative traits of a dynamic motion themselves change.

PMC usually uses static motions because a static motion can easily express a pattern. However, we propose a myoelectric classification distinguishing dynamic motions with increasing robustness on limb position changes. It is hereafter called *sequence-based myoelectric classification (SMC)* for the comparison to PMC. The proposed method assumes that reliable information to the limb position effect can be extracted by comparing temporal sequences of EMG features of dynamic motions. Changing the limb position causes the shift of the baselines of EMG signals because the configuration of muscles is changed to keep the altered pose. However, specific relative traits identifying a dynamic motion in the temporal sequences are maintained even when the baselines of EMG signals are changed. In order to show the robustness of dynamic motions, the classification accuracies between SMC and PMC were compared when training sets and validation sets were chosen at different limb positions. Statistical analysis of Results section of this chapter shows that the classification accuracies of SMC with dynamic motions are significantly better than of PMC with static motions over all subjects. It is then concluded that SMC is more robust than PMC to the limb position changes.

Due to different attributes of static and dynamic motions, the classification of dynamic motions and the classification of static motions have to be separately dealt with. For example, to classify static motions, supervised learning methods for non-temporal pattern recognition such as Naïve Bayes, multilayer perceptron, and support vector machine can be exploited. On the other hand, temporal pattern recognition is needed to recognize dynamic motions because dynamic motions are represented by temporal sequences. For temporal pattern recognition, dynamic time warping with template matching, hidden Markov models, and time-delay neural network can be used.

3.2 Methods

In this section, a sequence-based myoelectric classification is proposed and its key processes (onset/offset detection, alignment, and template matching) are described. Furthermore, details for pattern-based myoelectric classification, which is compared with SMC in terms of robustness on limb position changes, are described. Experimental protocol and data collection are also explained.

3.2.1 Pattern-based Myoelectric Classification

For extracting features of static motions, eight statistics were used as described in Table 1. The total number of features for each segment was 13 (the first seven statistics plus six AR coefficients associated with 6th order autoregressive model). The window size for the segmentation was 200ms with 50% overlap. Fisher's Linear Discriminant Analysis was applied to the extracted features from eight EMG channels to reduce dimensionality. For the classification of static hand motions, Naïve Bayes classifier was chosen. These methods with the time domain features (Table 1) showed reasonable performance for PMC [84].

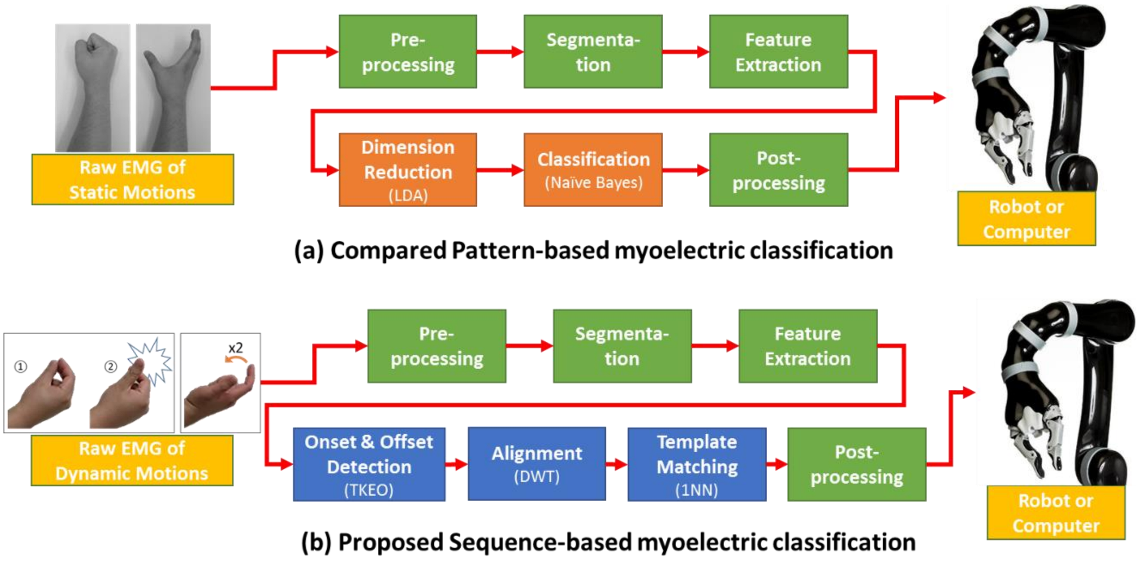

Raw EMG of Static Motions
Pre-processing
Segmentation
Feature Extraction
Dimension Reduction (LDA)
Classification (Naïve Bayes)
Post-processing
Robot or Computer
(a) Compared Pattern-based myoelectric classification
①
②
x2
Raw EMG of Dynamic Motions
Pre-processing
Segmentation
Feature Extraction
Onset & Offset Detection (TKEO)
Alignment (DWT)
Template Matching (1NN)
Post-processing
Robot or Computer
(b) Proposed Sequence-based myoelectric classification

Figure 14 Pattern-based myoelectric classification (PMC) and the proposed sequence-based myoelectric classification (SMC)

3.2.2 Proposed Sequence-based Myoelectric Classification

The proposed SMC focuses on classifying dynamic motions. Similar to PMC, SMC uses pre-processing, segmentation, feature extraction, and post-processing. However, three processes are vital for the classification of dynamic motions: *onset/offset detection*, *alignment*, and *template matching* as shown in Figure 14 (b). Teager–Kaiser Energy Operator (TKEO) [92] was used to detect onset/offset (start and end points) of dynamic motions. Dynamic Time Warping (DTW) [93] was employed to align different time-length dynamic motions before the classification (Figure 15). Template matching uses the Pearson product-moment correlation coefficient as a similarity metric to classify the dynamic motion.

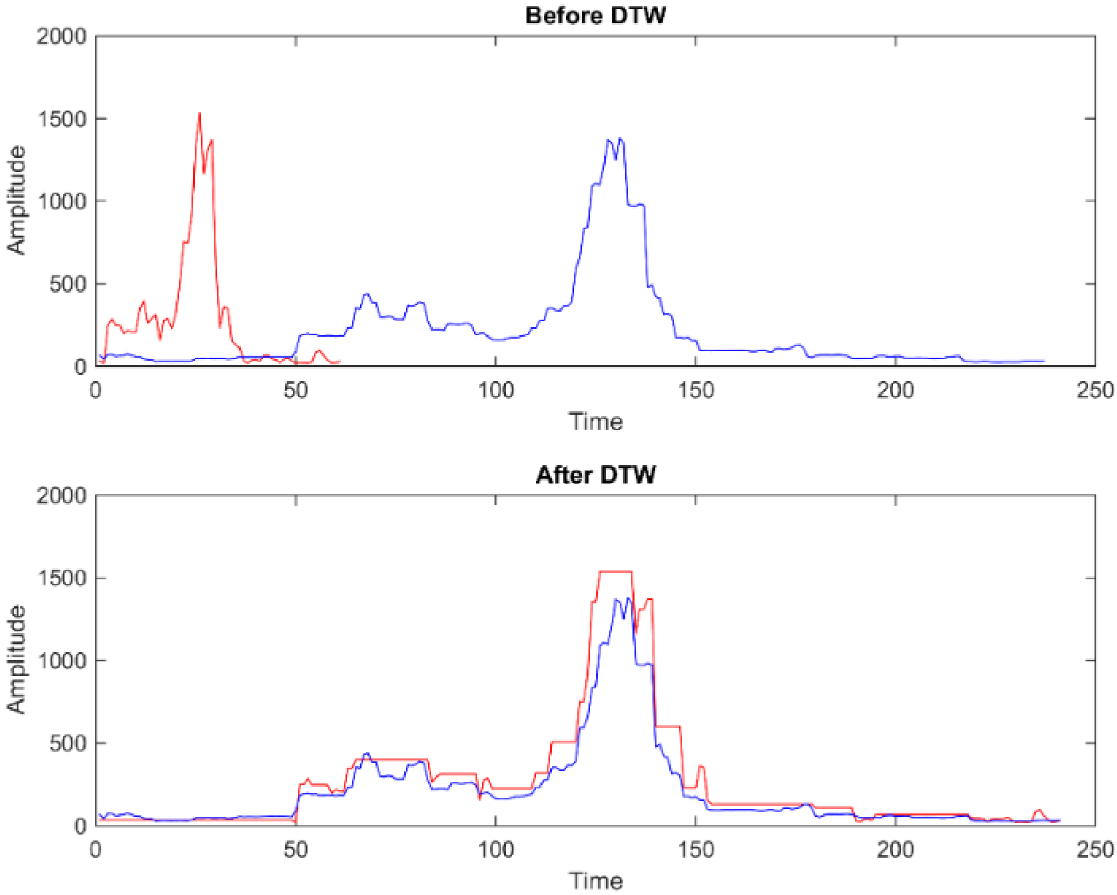


Figure 15 RMS values of two dynamic motions (the same motion class but different lengths) before and after alignment by DTW

For feature extraction, seven features (MAV, RMS, IEMG, WL, ZC, SSC, and SKW) in Table 1 were used. At a preliminary test, classification accuracies of SMC with the seven features and with all eight features (including 6th order AR coefficients) were compared. However, there was no enhancement when using 6th order AR coefficients. The window size for the segmentation was 50ms with 50% overlap. We set a smaller window size for SMC than PMC because SMC needs more detailed shapes of EMG feature vectors to distinguish a dynamic motion.

Onset and Offset Detection

Detecting the start and end of a motion is needed to classify dynamic motions. To do this, TKEO was used. TKEO has been widely employed for muscle activity detection [94-96] because of its ability to identify the amplitude and instantaneous frequency of EMG signals [92].

The discrete TKEO Ψ_{d} is defined for a discrete signal $\mathbf{x}(\mathrm{n})$ as:

$$\Psi_{\mathrm{d}}[\mathbf{x}(\mathrm{n})] = \mathbf{x}^2(\mathrm{n}) - \mathbf{x}(\mathrm{n}+1)\mathbf{x}(\mathrm{n}-1) \tag{12}$$

Means and standard deviations (SD) of TKEO of EMG signals at rest (for all EMG channels) were used to determine threshold values of each EMG channel, using:

$$\begin{aligned} &\mathrm{TK}_{ch}(\mathrm{n}) = \Psi_{\mathrm{d}}[\mathrm{EMG}_{ch}(\mathrm{n})], \\ &\mathrm{Threshold}_{ch} = \overline{\mathrm{TK}_{ch}} + h \times \mathrm{SD}(\mathrm{TK}_{ch}) \end{aligned} \tag{13}$$

Here, $\mathrm{EMG}_{ch}(\mathrm{n})$ is a raw value of an EMG channel, ch, at time n. $\overline{\mathrm{TK}_{ch}}$ is mean of signal TK_{ch} and $\mathrm{SD}(\mathrm{TK}_{ch})$ is standard deviation of signal TK_{ch}. The factor h was set to 3 [2]. If TKEO values of all EMG channels are higher than threshold values and are kept for a certain period of time τ, then we consider this point as an onset of motion. If TKEO values of all EMG channels stay below the threshold values for the time period of τ, this point is considered as the offset of motion. In this study, τ was heuristically chosen as 300ms based on an acceptable motion detection delay (less than 1s in this study).

Alignment

All dynamic motions have different time-lengths. In order to compare these dynamic motions in different lengths, a temporal alignment process is required. DTW [93] was used for this purpose. DTW is a distance function that estimates a minimal distance of two time sequences $\mathrm{A} \coloneqq (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_\mathrm{N})$ of length $\mathrm{N} \in \mathbb{N}$ and $\mathrm{B} \coloneqq (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_\mathrm{M})$ of length $\mathrm{M} \in \mathbb{N}$ after the optimal alignment of two sequences. DTW can also be used for the alignment of two different length sequences as depicted in Figure 15. The process is described below.

A warping path $\mathrm{W} \coloneqq (\mathrm{w}_1, \mathrm{w}_2, \dots, \mathrm{w}_\mathrm{L})$ with $\mathrm{w}_\ell = (\mathrm{n}_\ell, \mathrm{m}_\ell)$ for $\mathrm{n}_\ell \in [1\!:\mathrm{N}]$, $\mathrm{m}_\ell \in [1\!:\mathrm{M}]$, and $\ell \in [1\!:\mathrm{L}]$, $\max(\mathrm{N}, \mathrm{M}) \leq \mathrm{L} < \mathrm{N} + \mathrm{M}$ has three conditions as follows:

(i) Boundary condition:

$$\mathrm{w}_1 = (1,1) \text{ and } \mathrm{w}_\mathrm{L} = (\mathrm{N}, \mathrm{M}) \tag{14}$$

(ii) Monotonicity condition:

$$\mathrm{n}_1 \leq \mathrm{n}_2 \leq \cdots \leq \mathrm{n}_\mathrm{L} \text{ and } \mathrm{m}_1 \leq \mathrm{m}_2 \leq \cdots \leq \mathrm{m}_\mathrm{L} \tag{15}$$

(iii) Step size condition:

$$\mathrm{w}_{\ell+1} - \mathrm{w}_\ell \in \{(1,0), (0,1), (1,1)\} \text{ for } \ell \in [1\!:\mathrm{L} - 1] \tag{16}$$

The total cost $\mathrm{c_W}(\mathrm{A}, \mathrm{B})$ of a warping path W between A and B can be defined as:

$$\mathrm{c_W}(\mathrm{A}, \mathrm{B}) \coloneqq \sum_{\ell=1}^{L} d\left(\mathbf{a}_{\mathrm{n}_\ell}, \mathbf{b}_{\mathrm{m}_\ell}\right) \tag{17}$$

The distance function $d(\mathbf{a}, \mathbf{b})$is a Euclidean distance defined as:

$$d(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{K} \sqrt{(\mathrm{a}_i - \mathrm{b}_i)^2} \tag{18}$$

Before calculating a Euclidean distance of multi-dimensional vectors, the normalization is required to offset different magnitude of each feature. In this study, z-score normalization was used.

With an *optimal warping path* W^* between A and B, DTW distance of A and B, $\mathrm{dtw}(\mathrm{A},\mathrm{B})$, is then defined as:

$$\begin{aligned}\mathrm{dtw}(\mathrm{A},\mathrm{B}) &:= \mathrm{c}_{W^*}(\mathrm{A},\mathrm{B}) \\ &= \min\{\mathrm{c}_{\mathrm{W}}(\mathrm{A},\mathrm{B})|\ \mathrm{W}\text{ is a warping path}\}\end{aligned} \tag{19}$$

In order to find an optimal warping path W^*, DTW uses a dynamic programming technique [97] via *accumulated cost matrix* D $\in\ \mathbb{R}^{\mathrm{NxM}}$ defined by:

$$\mathrm{D}(\mathrm{n},\mathrm{m}) := \mathrm{dtw}\big(\mathrm{A}(1\colon\mathrm{n}),\mathrm{B}(1\colon\mathrm{m})\big) \tag{20}$$

The accumulated cost matrix D can be calculated by:

$$\mathrm{D}(\mathrm{n},\mathrm{m}) = d(\mathbf{x}_\mathrm{n},\mathbf{y}_\mathrm{m}) + \min\begin{Bmatrix}\mathrm{D}(\mathrm{n}-1,\mathrm{m}-1)\\ \mathrm{D}(\mathrm{n}-1,\mathrm{m})\\ \mathrm{D}(\mathrm{n},\mathrm{m}-1)\end{Bmatrix}\text{ for }\ \mathrm{n}\in[2\colon\mathrm{N}],\mathrm{m} \\ \in[2\colon\mathrm{M}] \tag{21}$$

with $\mathrm{D}(\mathrm{n},1)=\sum_{\mathrm{k}=1}^{\mathrm{n}}\mathrm{d}\big(\mathbf{x}_\mathrm{k},\mathbf{y}_1\big)$ for $\mathrm{n}\in[1\text{:}\mathrm{N}]$, $\mathrm{D}(1,\mathrm{m})=\sum_{\mathrm{k}=1}^{\mathrm{m}}\mathrm{d}\big(\mathbf{x}_1,\mathbf{y}_\mathrm{k}\big)$ for $\mathrm{m}\in[1\text{:}\mathrm{M}]$. From this iterative technique, an optimal warping path and DTW distance can be calculated in a time complexity of $\mathrm{O}(\mathrm{NM})$ [97].

Template Matching

Template matching labels an input dynamic motion as matching a template predefined in the training phase. It finds the closest templates to the input dynamic motion by a similarity metric; in fact, this decision-making process is the same as 1-Nearest Neighbor (1NN) classification [98].

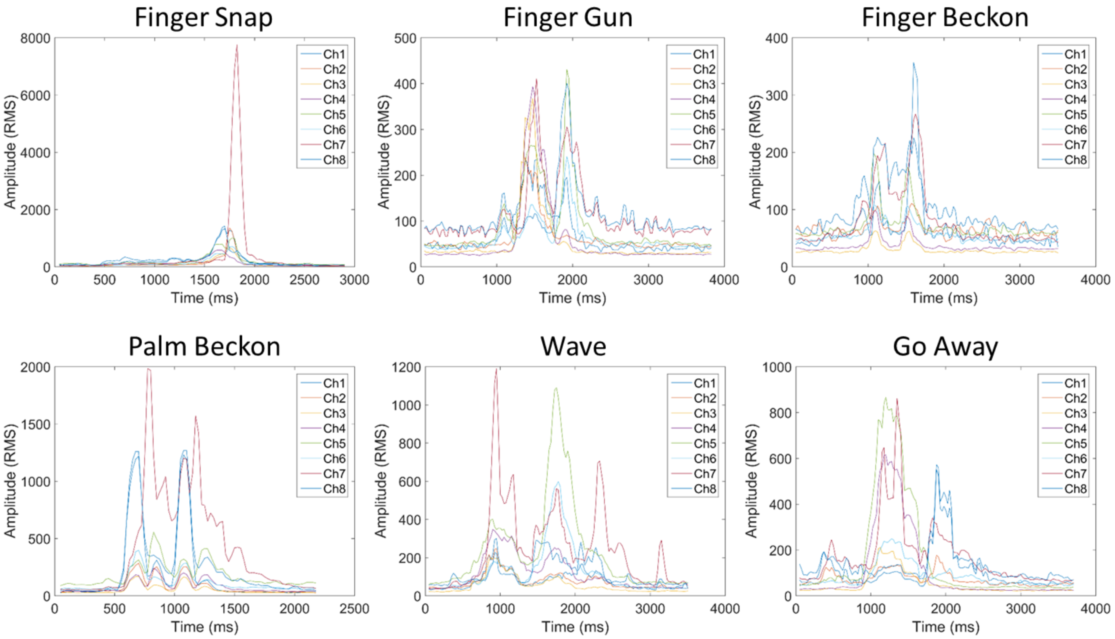


Figure 16 Six templates of dynamic motions

In the training section, a template of each dynamic motion was generated from a training set. To make the collected dynamic motions mutually comparable, training samples of a dynamic motion were aligned using DTW. After the alignment, mean feature profile of the aligned motion samples was calculated as a template. This process was repeated for all dynamic motions. The templates were used as representatives of each dynamic motion for the classification. Figure 16 instantiates templates of six dynamic motions.

In the prediction section, after the alignment of a queried motion and the templates by DTW, a correlation coefficient, r, of the queried motion *x* and a template *y* is calculated as a similarity metric, defined by:

$$\mathrm{r}(\mathbf{p},\mathbf{q}) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{p_i}-\bar{\mathrm{p}})(\mathrm{q_i}-\bar{\mathrm{q}})}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{p_i}-\bar{\mathrm{p}})^2}\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{q_i}-\bar{\mathrm{q}})^2}} \tag{22}$$

Here, $\bar{\mathrm{p}}$ and $\bar{\mathrm{q}}$ are mean values of signals **p** and **q**, respectively, and n is the length of signals. Correlation coefficient can compare the similarity of sequences of dynamic motions.

A motion class of the queried motion was then decided by 1NN classification by:

$$\text{Predicted Motion Class} = \arg\max_{c}\ \mathrm{r}\left(\text{Motion}_{\text{query}}, \text{Template}_{c}\right) \tag{23}$$

Here, Template_c is the template of dynamic motion *c*. Note that the more similar two signals are, the higher a correlation coefficient r is. Hence, Equation (23) compares the queried motion with six templates of dynamic motions and finds the most similar template as the predicted motion class for the queried motion.

3.2.3 Data Collection

For evaluating the performance of PMC and SMC, seven static motions and six dynamic motions (Figure 17) were collected at four different limb positions. For collecting EMG signals of motions, eight Trigno™ Wireless EMG sensors were located on upper limb muscles, as shown in Figure 18. Raw EMG signals were sampled at 2 kHz and filtered by a band-pass filter between 10-400 Hz. The hand motions were collected from eight subjects. The experiments were approved by the Texas A&M University Institutional Review Board and informed consent forms were signed by all subjects.

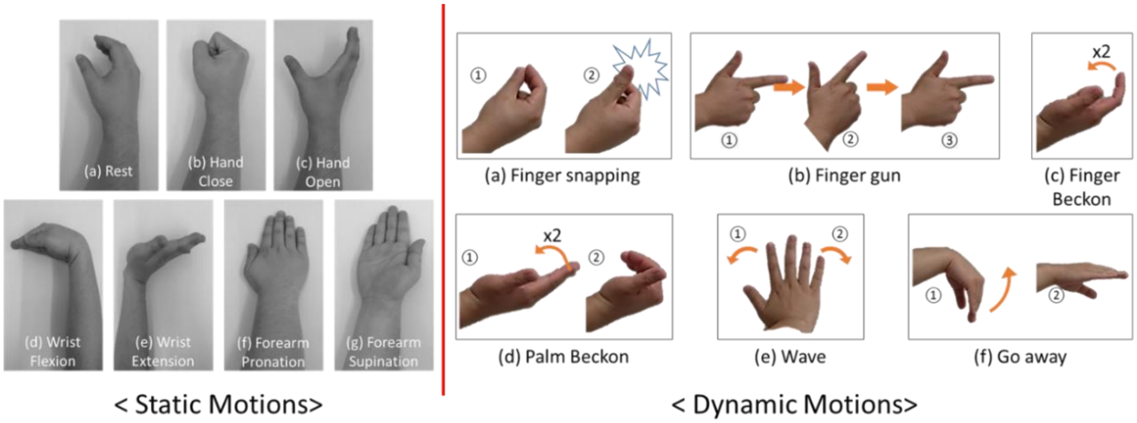


Figure 17 Static motions (left), the shape of a motion is unchanged over time; Dynamic motions (right), the shape of a motion is changed over time

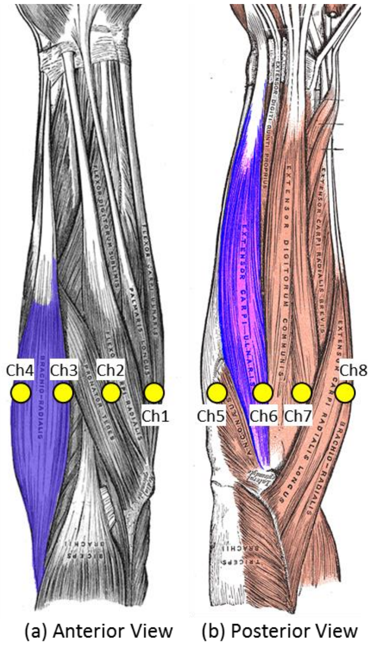


Figure 18 Location of surface EMG electrodes to collect EMG data

Six dynamic motions were repeated 15 times with different time lengths (1-3 seconds), whereas seven static motions were repeated 5 times, each held for 3 seconds. Note that SMC needs a whole dynamic motion as a training sample or a validation sample, whereas, in PMC, plentiful training samples or validation samples can be extracted from a static motion by segmenting the same motion. In order to produce sufficient samples of SMC, dynamic motions were repeated. Furthermore, because SMC uses a feature profile, not a feature value, for training and validation, it needs more motion samples than PMC.

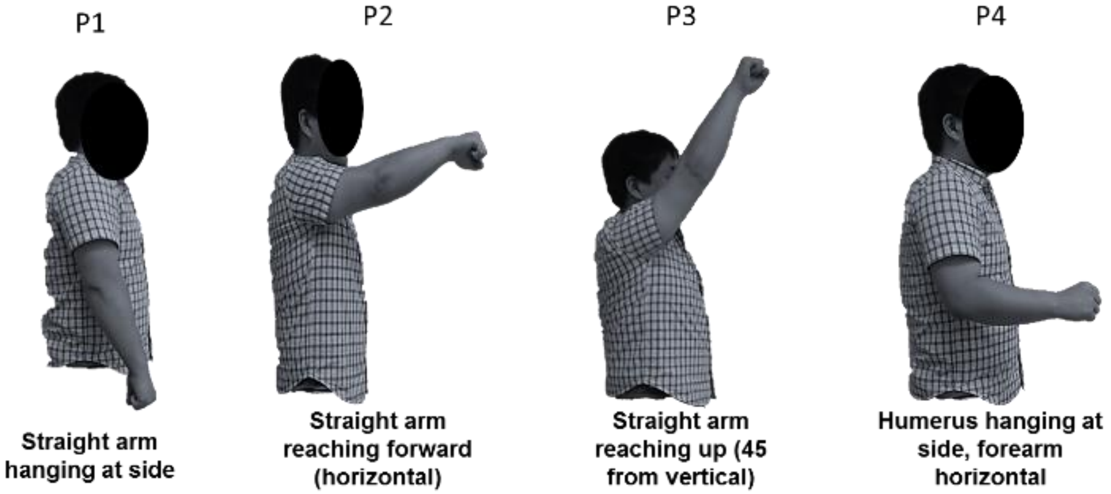

P1
P2
P3
P4
Straight arm hanging at side
Straight arm reaching forward (horizontal)
Straight arm reaching up (45 from vertical)
Humerus hanging at side, forearm horizontal

Figure 19 Four different limb positions, 1) straight arm hanging at side (P1), 2) straight arm reaching forward (P2), 3) straight arm reaching up (P3), and 4) humerus hanging at side and forearm horizontal (P4)

In order to collect static and dynamic motions at different poses, four upper limb positions were chosen as depicted in Figure 19. The positions are: (1) Straight arm hanging at side (or P1 for short), (2) Straight arm reaching forward (P2), (3) Straight arm reaching up (P3), and (4) Humerus hanging at side and forearm horizontal (P4). The dynamic and static motions at these limb positions were collected repeatedly to evaluate the robustness of PMC and SMC to the limb position effect.

3.3 Results

In order to evaluate the proposed approach, classification accuracies of SMC and PMC with choosing training and validation sets at four different limb poses are compared. Furthermore, statistics of the experimental results are analyzed.

3.3.1 Experimental Results

To evaluate the classification accuracy of SMC, a 5-fold cross validation (CV) technique was chosen. Among 15 repeated samples of each dynamic motion, 12 were used for making a template of each dynamic motion for training, and the other 3 were used to validate the classification accuracy. Because there were five possible combinations to choose different training sets and validation sets, an average of the five iterations was chosen as the final classification accuracy.

In the same manner, for estimating the classification accuracy of PMC, a 5-fold CV technique was also chosen. Four out of five repeated samples of each static motion were used for training and the other was used for validation. In total, a classifier was trained from 4 (repeated samples) x 7 (static motions) = 28 samples and was validated from 1 x 7 = 7 samples. This procedure also repeated 5 times with different training and validation sets.

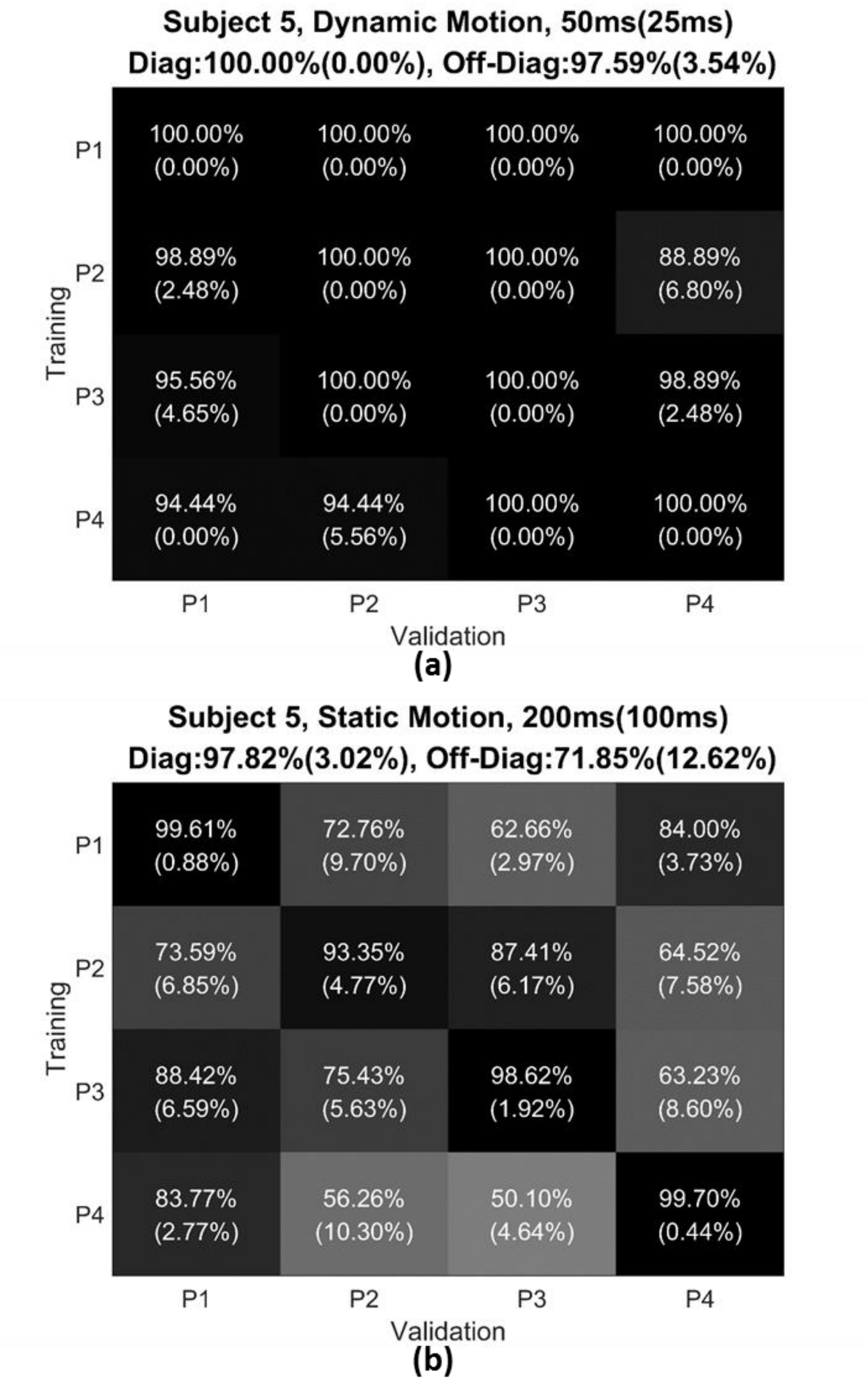
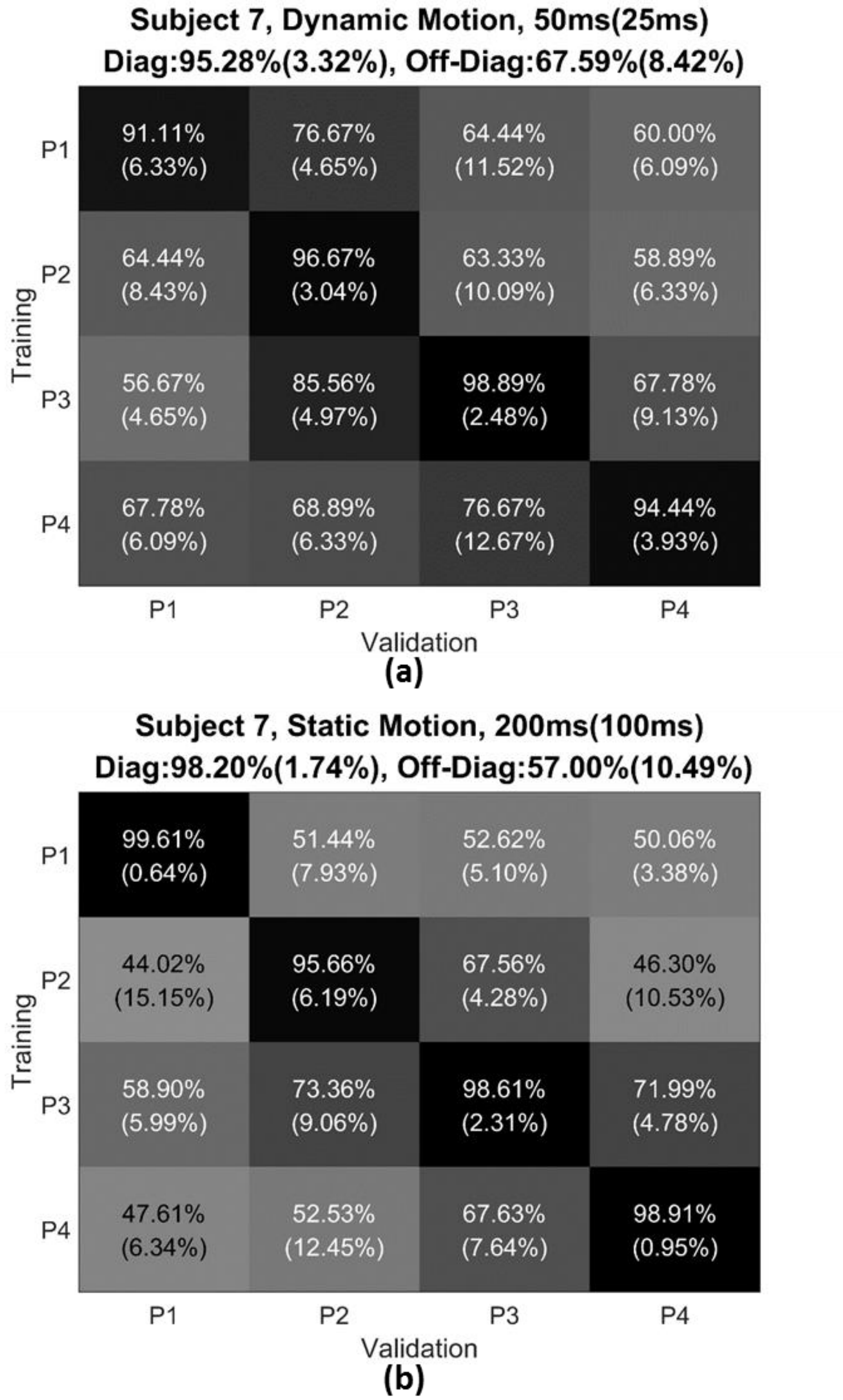


Figure 20 Classification accuracy of Subject 5 (the best case); (a) SMC, (b) PMC. Standard deviations are in the parentheses. 50ms window size with 25ms overlap for dynamic motion and 200ms window size with 100ms overlap for static motions were chosen.



Subject 7, Dynamic Motion, 50ms(25ms)
Diag:95.28%(3.32%), Off-Diag:67.59%(8.42%)

Training \ Validation	P1	P2	P3	P4
P1	91.11% (6.33%)	76.67% (4.65%)	64.44% (11.52%)	60.00% (6.09%)
P2	64.44% (8.43%)	96.67% (3.04%)	63.33% (10.09%)	58.89% (6.33%)
P3	56.67% (4.65%)	85.56% (4.97%)	98.89% (2.48%)	67.78% (9.13%)
P4	67.78% (6.09%)	68.89% (6.33%)	76.67% (12.67%)	94.44% (3.93%)

(a)

Subject 7, Static Motion, 200ms(100ms)
Diag:98.20%(1.74%), Off-Diag:57.00%(10.49%)

Training \ Validation	P1	P2	P3	P4
P1	99.61% (0.64%)	51.44% (7.93%)	52.62% (5.10%)	50.06% (3.38%)
P2	44.02% (15.15%)	95.66% (6.19%)	67.56% (4.28%)	46.30% (10.53%)
P3	58.90% (5.99%)	73.36% (9.06%)	98.61% (2.31%)	71.99% (4.78%)
P4	47.61% (6.34%)	52.53% (12.45%)	67.63% (7.64%)	98.91% (0.95%)

(b)

Figure 21 Classification accuracy of Subject 7 (the worst case); (a) SMC, (b) PMC. Standard deviations are in the parentheses. 50ms window size with 25ms overlap for dynamic motion and 200ms window size and 100ms overlap for static motions were chosen.

In order to measure the robustness to the limb position effect, a limb position for extracting a training set and a different limb position for extracting a validation set were chosen. Figure 20 and Figure 21 show the classification accuracy for Subject 5 and Subject 7, respectively, for both SMC and PMC schemes. Each cell in figures represents an experimental result with a pair of limb positions. The percent value in each cell is the average classification accuracy and the value in the parenthesis is SD from the 5-fold CV. For example, the cell (1, 1) at the left-top corner of Figure 20 (a) illustrates 100% classification accuracy with 0% SD when using a training set at the limb position P1 and a validation set at the same position. This case—the limb positions of training and validation sets are same—is called the *same limb position.* Otherwise, the limb positions of training and validation sets are different; this is called the *different limb position.* The color-codes display the classification accuracy of each cell ranging from 100% (black color) to 0% (white). Among eight subjects, Subject 5 showed the highest classification accuracy of SMC, and subject 7 showed the lowest.

As shown in Figure 20, SMC and PMC worked properly with high classification accuracies (>90%) at the same limb position. However, SMC worked better at the different limb position than PMC; the classification accuracies of SMC at the different limb position were ranging from 88.89% to 100% while PMC showed the classification accuracy ranging from 50.1% to 88.42%. Even for subject 7 that had the lowest classification accuracy, SMC at the different limb position still shows better performances (from 76.67% to 56.67%) than PMC (from 73.36% to 44.02%).

3.3.2 Statistical Analysis

In order to statistically validate the experimental results, t-test was used. There were three different conditions: hand motion types (static vs. dynamic), the number of features (single vs. multiple), and similarity metrics (normalized DTW distance vs. correlation coefficient). Each condition included twofold results at the same and different limb position. A particular point was that the first condition (different hand motion types) was analyzed by two-sample t-test because their results were independent; they resulted from PMC and SMC. The other two conditions were analyzed by a paired t-test. The significance level of all statistical analysis was set as 5% ($p < 0.05$).

Table 7 represents the classification accuracy of both SMC and with training and validation sets came from the same limb position. SMC classified dynamic hand motions and PMC classified static hand motions. Based on the results of Table 7, the classification accuracies of SMC were greater or equal to of PMC at the same limb position. Table 7 indicates that there was no significant difference in the classification accuracy between PMC and SMC except for Subject 2 and 6 from whom SMC showed significantly higher accuracies than PMC.

Table 8 shows the classification accuracy of both SMC and PMC with training and validation sets came from different limb positions. The statistical results show that SMC significantly outperformed PMC from all subjects. This indicates that classifying dynamic motions is more robust to the limb position effect than PMC (classifying static motions).

Table 7 Classification accuracy with different motion types at the same limb position

Subject	Classification Accuracy (standard deviation)		Two-sample t test	p value (<0.05)
	PMC (Static Motion)	SMC (Dynamic Motion)		
1	97.97% (1.58%)	96.67% (1.81%)	t(6) = 1.08	p = 0.322
2	96.53% (0.84%)	98.89% (0.91%)	t(6) = -3.81	**p = 0.009**
3	99.70% (0.41%)	99.44% (0.64%)	t(6) = 0.68	p = 0.524
4	98.91% (1.56%)	99.17% (1.06%)	t(6) = -0.27	p = 0.794
5	97.82% (3.02%)	100.00% (0.00%)	-[a]	-
6	97.77% (1.40%)	99.72% (0.56%)	t(6) = -2.60	**p = 0.040**
7	98.20% (1.74%)	95.28% (3.32%)	t(6) = 1.56	p = 0.170
8	98.93% (0.92%)	99.72% (0.56%)	t(6) = -1.48	p = 0.190
a.	**Some experiments resulted same classification accuracies during 5-fold CV like Subject 5. (As variance is ~0, t-test cannot be applied.)**			

Table 8 Classification accuracy with different motion types at the different limb position

Subject	Classification Accuracy (standard deviation)		Two-sample t test	p value (<0.05)
	PMC (Static Motion)	SMC (Dynamic Motion)		
1	54.74% (17.43%)	75.28% (8.15%)	t(22) = -3.70	**p = 0.001**
2	56.48% (12.54%)	90.46% (4.28%)	t(22) = -8.89	**p = 0.000**
3	80.91% (8.01%)	89.35% (6.99%)	t(22) = -2.75	**p = 0.012**
4	77.15% (10.79%)	86.11% (6.57%)	t(22) = -2.46	**p = 0.022**
5	71.85% (12.62%)	97.59% (3.54%)	t(22) = -6.80	**p = 0.000**
6	64.65% (9.18%)	96.57% (3.62%)	t(22) = -11.20	**p = 0.000**
7	57.00% (10.49%)	67.59% (8.42%)	t(22) = -2.73	**p = 0.012**
8	70.57% (14.29%)	91.67% (9.73%)	t(22) = -4.23	**p = 0.000**

Table 9 and Table 10 represent the classification accuracies of the proposed SMC at the same limb position and at the different limb position, respectively, when using multi-dimensional EMG features (7 time-domain features) and when using only RMS feature. According to the statistical analysis displayed in Table 9 (at the same limb position), there was no significant difference between 7 time-domain features and only RMS feature. However, the statistical analysis for Subject 2,3,6,7, and 8 shown in Table 10 (at the different limb position) states that the classification accuracy of SMC with 7 time-domain features was significantly higher than with only RMS feature. This means that multi-dimensional EMG features perform better for SMC at the different limb position. In addition, for no effect of using multi-dimensional EMG features at the same limb position, we can speculate that the classification accuracy was already enough high with using only RMS feature.

Table 9 Classification accuracy with different feature sizes at the same limb position in SMC

Subject	**Classification Accuracy (standard deviation)**		**Paired t test**	**p value (<0.05)**
	RMS	7 Features		
1	96.11% (2.13%)	96.67% (1.81%)	t(3) = -1.00	p = 0.391
2	96.94% (3.19%)	98.89% (0.91%)	t(3) = -1.48	p = 0.235
3	98.61% (1.06%)	99.44% (0.64%)	t(3) = -1.19	p = 0.319
4	99.17% (1.06%)	99.17% (1.06%)	t(3) = 0.00	p = 1.000
5	100.00% (0.00%)	100.00% (0.00%)	-	-
6	99.44% (0.64%)	99.72% (0.56%)	t(3) = -1.00	p = 0.391
7	93.89% (2.31%)	95.28% (3.32%)	t(3) = -1.67	p = 0.194
8	98.61% (1.40%)	99.72% (0.56%)	t(3) = -2.45	p = 0.092

Table 10 Classification accuracy with different feature sizes at the different limb position in SMC

Subject	**Classification Accuracy (standard deviation)**		**Paired t test**	**p value (<0.05)**
	RMS	7 Features		
1	73.24% (5.98%)	75.28% (8.15%)	t(11) = -1.51	p = 0.158
2	86.67% (6.34%)	90.46% (4.28%)	t(11) = -2.76	**p = 0.019**
3	86.57% (6.20%)	89.35% (6.99%)	t(11) = -3.74	**p = 0.003**
4	84.91% (7.57%)	86.11% (6.57%)	t(11) = -0.83	p = 0.422
5	95.65% (6.74%)	97.59% (3.54%)	t(11) = -1.69	p = 0.120
6	93.15% (4.92%)	96.57% (3.62%)	t(11) = -3.05	**p = 0.011**
7	64.17% (10.12%)	67.59% (8.42%)	t(11) = -3.46	**p = 0.005**
8	85.19% (11.71%)	91.67% (9.73%)	t(11) = -3.47	**p = 0.005**

Likewise, the results shown in Table 11 and Table 12 represents the comparison of classification accuracy between when using the correlation coefficient and when using the normalized DTW as a similarity metric in template matching of SMC; at the same limb position and at the different limb position, respectively. At the same limb position (Table 11), there was no significant difference except Subject 7 from whom using the correlation coefficient significantly performed better than using the normalized DTW. Moreover, at the different limb position (Table 12), there were significant improvements when using the correlation coefficient (for Subject 1, 2, 6, 7, and 8). The other subjects (Subject 3, 4, and 5) showed similar accuracies. It means that using the correlation coefficient performed better than or equal to using the normalized DTW for SMC regardless the limb position.

Table 11 Classification accuracy with different similarity metrics at the same limb position in SMC

Subject	**Classification Accuracy (standard deviation)**		**Paired t test**	**p value (<0.05)**
	Normalized DTW	Corr. Coeff.		
1	94.44% (2.03%)	96.67% (1.81%)	t(3) = -1.36	p = 0.267
2	97.50% (3.67%)	98.89% (0.91%)	t(3) = -0.78	p = 0.492
3	99.17% (1.06%)	99.44% (0.64%)	t(3) = -0.52	p = 0.638
4	99.44% (1.11%)	99.17% (1.06%)	t(3) = 0.29	p = 0.789
5	100.00% (0.00%)	100.00% (0.00%)	-	-
6	99.72% (0.56%)	99.72% (0.56%)	-	-
7	85.83% (6.31%)	95.28% (3.32%)	t(3) = -3.54	**p = 0.038**
8	98.89% (2.22%)	99.72% (0.56%)	t(3) = -1.00	p = 0.391

Table 12 Classification accuracy with different similarity metrics at the different limb position in SMC

Subject	**Classification Accuracy (standard deviation)**		**Paired t test**	**p value (<0.05)**
	Normalized DTW	Corr. Coeff.		
1	67.87% (7.98%)	75.28% (8.15%)	t(11) = -4.90	**p = 0.000**
2	87.50% (5.47%)	90.46% (4.28%)	t(11) = -2.60	**p = 0.025**
3	88.33% (8.17%)	89.35% (6.99%)	t(11) = -0.90	p = 0.387
4	85.93% (6.24%)	86.11% (6.57%)	t(11) = -0.13	p = 0.900
5	96.11% (5.75%)	97.59% (3.54%)	t(11) = -1.39	p = 0.191
6	95.09% (3.68%)	96.57% (3.62%)	t(11) = -2.68	**p = 0.021**
7	56.39% (7.51%)	67.59% (8.42%)	t(11) = -6.27	**p = 0.000**
8	85.65% (12.84%)	91.67% (9.73%)	t(11) = -3.71	**p = 0.003**

Statistical analysis of experimental results attested that SMC overall performed well with high accuracy, and was more robust to the limb position effect than PMC. Furthermore, using multi-dimensional EMG features and correlation coefficient in SMC were shown improvements compared to the case using only RMS feature and normalized DTW. However, there still exists a need to improve SMC over all subjects in terms of a relatively-wide gap between the best case and the worst case, e.g., the classification accuracy from 67.6% to 97.6%.

3.4 Discussion

Myoelectric classification has been broadly studied for prosthetic and rehabilitation devices. A number of studies have been published to improve the performance of myoelectric classification. Pattern-based myoelectric classification is also one of these achievements. However, PMC still has drawbacks in terms of robustness and the discrepancy between the laboratory and real-life settings. In order to diminish the gap and increase the robustness, this study focused on using dynamic motions that has been less considered in the myoelectric classification literature. Basically, the assumption was that comparing the similarity of dynamic motions is more robust to the changes in EMG signals when moving a limb than comparing statistical measures (means and standard deviations) of static motions. Figure 22 describes this idea. The limb position change causes to alter the level and amplitude of RMS values of EMG signals. After the change, statistical values of the static motion are different, however, RMS profiles of the dynamic motion have similar shapes just with a slight shift.

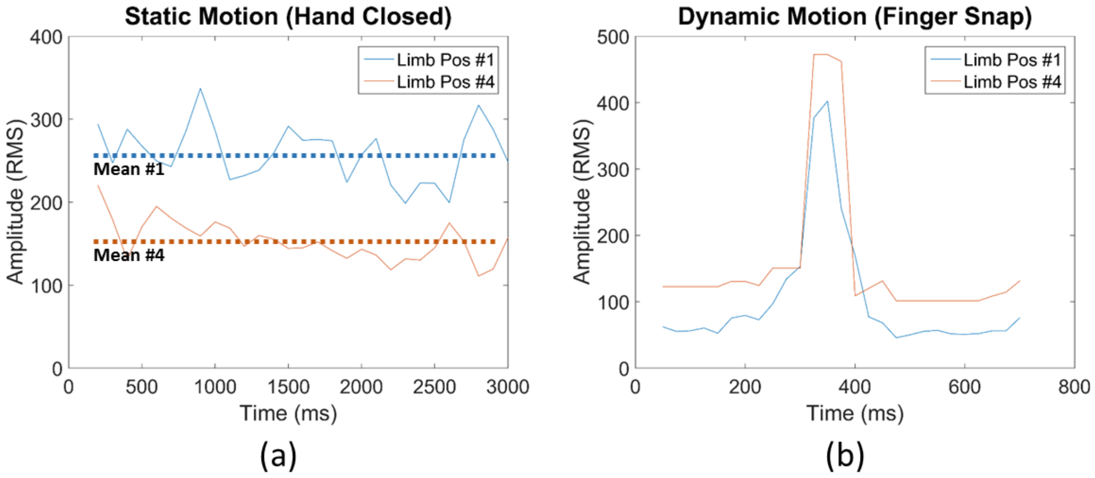


Figure 22 (a) Static motion (Hand Closed) vs. (b) dynamic motion (Finger Snap) at the different limb position; after changing a limb position, the same static motions are totally different, however, the same dynamic motions keep their similar RMS profiles even their RMS values are slightly shifted.

We proposed Sequence-based myoelectric classification to classify dynamic hand motions by comparing their profiles of EMG feature vectors (sequences). Furthermore, to evaluate the robustness to the limb position effect the classification accuracy of SMC was examined at different limb positions. The proposed approach seems to be similar to [52], however, there are distinct points: (1) correlation coefficient was used as a similarity metric between two temporal sequences compared to using normalized DTW distance in [52], (2) TKEO was adopted to improve the detection of onset and offset of EMG signals, and (3) multi-dimensional EMG features were used to increase the classification accuracy. This study showed the robustness to the limb position effect when using dynamic motions compared to using static motions. Despite several attempts to solve the limb position effect [69-71, 74, 75], to the authors' knowledge the approach taken in this study, i.e., classifying dynamic motions to enhance the robustness of the myoelectric system, has not been tried before.

In addition, an advantage SMC have is its ability to increase the number of classifiable motions compared to PMC because of unlimited temporal information that dynamic motions have, but static motions do not. This means that the temporal characteristic of dynamic motions can be used to diversify discriminable motions, for example, differentiating a fast “wave” motion from a slow “wave” motion.

A significant difference between myoelectric approaches, such as SMC proposed here, and non-myoelectric approaches (vision [99], accelerometer [100], and gyroscope [101]) for gesture recognition can be found at the form of gestures they are focusing on (hand movements themselves or trajectories of a hand in a 3D space). Non-myoelectric approaches usually track trajectories of a hand of a subject to recognize human gestures, but myoelectric approaches recognize hand movements because EMG signals can represent accurate wrist and finger motions. For example, if a subject draws a pre-defined trajectory (a circle, a triangle, or a rectangle) with his/her hand in front of a camera, a vision-based gesture recognition system will recognize it and then conduct a command associated with the recognized gesture. However, a myoelectric gesture recognition system can recognize wrist and finger movements such as snapping fingers and a beckoning palm.

In terms of the workload and the size of space for gestures, myoelectric approach is preferable because it can recognize precise and concise gestures with small actions (finger and wrist movements). With the user’s perspective, repeated gestures with large movements can easily make users tired. Gestures (trajectories of a hand) for non-myoelectric approaches relatively need long and large movements compared to finger and wrist movements. This means that myoelectric approaches for gesture recognition are more convenient for users.

3.5 Conclusion

This chapter introduced the sequence-based myoelectric classification to classify dynamic hand motions. Also, it was examined at four different limb positions to show its robustness against the limb position effect compared to the pattern-based myoelectric classification. The robustness issues caused from limb position change, electrode shift, long-term use, fatigue, and skin condition are crucial for reliable uses in myoelectric classification systems. In order to enhance the reliability of the systems, this study considered classifying dynamic motions instead of static motions. It was shown that comparing the similarity of temporal sequences of dynamic motions is more robust to the changes of myoelectric signals than comparing constant values of static motions. This is based on the fact that each dynamic motion has its own temporal sequence which will be less affected by the limb position effect.

In order to evaluate the robustness of dynamic and static motions, hand motions were collected at four different limb positions and used for training and validation sets. The experimental results showed that, for all eight subjects, dynamic motions had higher classification accuracy than static motions when training and validation sets were chosen from different limb positions. For example, Subject 5 showed 97.59% classification accuracy of dynamic hand motions at different limb positions, while the classification accuracy of static hand motions was 71.85%.

From the experimental results, we could confirm that the dynamic motions are more robust to the limb positions effect than the static motions. Even though the approach of classifying dynamic motions need to be improved further to increase the classification accuracy, we can highly expect a more robust performance of dynamic motions with issues such as long term use and skin condition.

CHAPTER IV

REAL-TIME MYOELECTRIC INTERFACE USING DYNAMIC HAND GESTURES FOR A MULTIPLE-DOF ROBOT ARM

4.1 Introduction

The quality of human life keeps increasing via introducing computerized technologies ranging from smartphones to autonomous vehicles. There are uncountable examples of computerized technologies that support human life. With our reliance on the technology, humans want to communicate more intuitively with and to control more easily these devices. The needs have accelerated Human Computer Interaction studies. One approach in HCI, *myoelectric interface*, is the topic of this chapter. Myoelectric interface is an approach to send human intentions to computers, robots, and gaming devices by interpreting EMG signals generated by muscular activities. EMG signals represent electrical activities of muscles where electrodes are attached. Because all human movements involve muscular activities, the movements can be translated by understanding EMG signals well.

Based on the needs of interfacing with the devices supporting human life, we propose a dexterous myoelectric interface to control a 6-DOF robot manipulator with a 1-DOF gripper via the orientation of a forearm, muscle force, and dynamic hand motions via IMU and EMG signals. Orientation, angular velocities, and linear accelerations in a 3D space of a forearm of a human operator are estimated by an IMU sensor. Muscle force and dynamic hand motions are recognized from EMG signals. In this research, orientation of a forearm of an operator and proportional muscle force are used to generate continuous commands for the robot manipulator. Dynamic hand motions are classified to change manipulation modes. For example, a 6-DOF Kinova robot manipulator with a 1-DOF gripper needs mainly three manipulation modes: arm mode (translation), wrist mode (rotation), and finger mode (opening\closing a gripper). The dynamic hand motions are responsible for changing manipulation modes for the robot manipulator. Moreover, estimated muscle force is exploited to be aware of the activation of continuous commands as calculated from orientation of the IMU sensor.

4.2 Methods

In this study, a myoelectric interface using the dynamic motion classification introduced in the previous chapter is proposed. To estimate the performance of the proposed myoelectric interface, performance measures such as *real-time classification accuracy*, *time-measures*, and *path efficiency* are introduced. The performance measures of the myoelectric interface are compared to those of *GUI button-based jog interface* that is commonly used for controlling a robot manipulator in industrial and laboratorial applications. It is assumed here that the jog interface performs desirably in controlling a robot manipulator. The assumption is quite valid because numerous applications in the industries and researches have widely used this type of the jog interface. The performance of the proposed myoelectric can be relatively estimated by comparing it with the performance of the jog interface. The proposed myoelectric interface, the jog interface, the experimental setup, and protocol details are described below.

4.2.1 Proposed Myoelectric Interface

The aim of the proposed myoelectric interface is to control a robot manipulator via human gestures. Figure 23 shows a brief schema of the proposed myoelectric interface. In general, to control a robot manipulator, there are two types of commands: discrete commands to handle manipulation modes like the speed and movement type of the robot manipulator and continuous commands to represent velocity or position commands. Figure 23 represents discrete commands determined by classifying dynamic hand motions and continuous commands calculated from muscle force and orientation in a 3D space of a forearm. Each dynamic hand motion is used as an action to change states defined in a finite state machine. The muscle force activates/inactivates the interface system and the orientation (roll, pitch, and yaw) of a forearm generates 3-axis velocity commands.

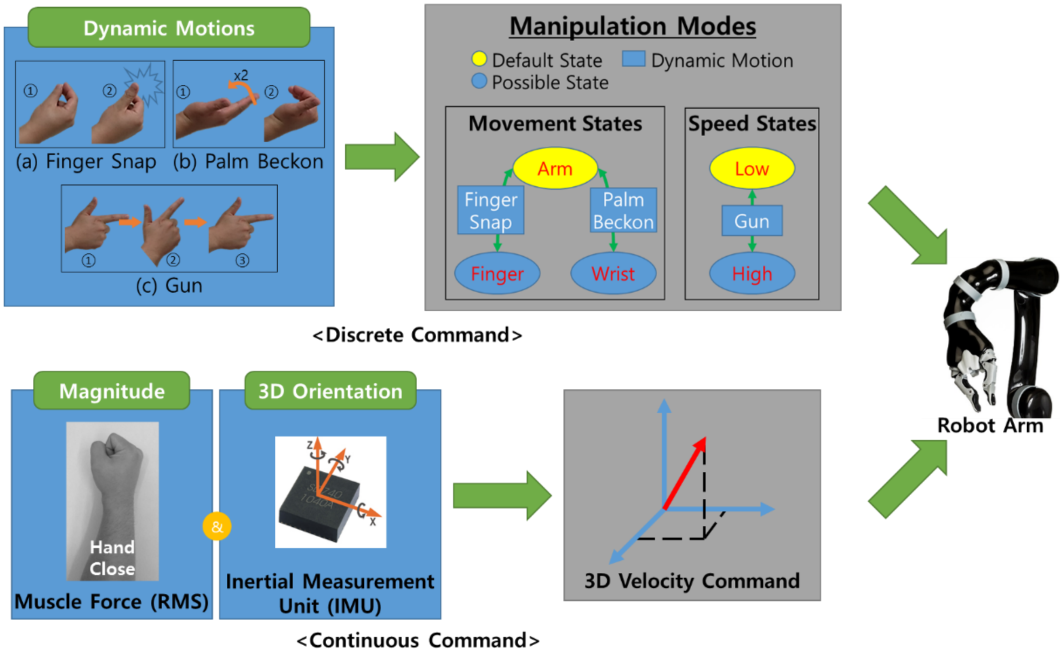


Figure 23 Schema of the proposed myoelectric interface

Definitions of a dynamic motion

A dynamic motion may be defined as a motion changing its pose and orientation over time. In order to distinguish a dynamic motion in real-time, we define constraints of the dynamic motion. First, to detect the dynamic motion in real-time, the dynamic motion has *rest-regions* before and after the motion. Two rest-regions are called as the *before-rest-region* and the *after-rest-region* as shown in Figure 24. The rest-region means that the mean of AVR RMS (average root mean square), $\mu_{\text{AVR RMS}}$, and the standard deviation of AVR RMS, $\sigma_{\text{AVR RMS}}$, are under the rest thresholds, μ_{THrest} and σ_{THrest}, respectively, during a rest time period (t_{rest}). AVR RMS is defined by:

$$\text{AVR RMS}(n) = \frac{1}{ch}\sum_{i=1}^{ch} \text{RMS}_i(n) \;\; for \text{ sample index } n \tag{24}$$

$\mu_{\text{AVR RMS}}$ and $\sigma_{\text{AVR RMS}}$ are calculated by:

$$\mu_{\text{AVR RMS}}(n) = \frac{1}{n}\sum_{i=1}^{n} \text{AVR RMS}(n) \;\; for\ 1 \le n \le n_c \tag{25}$$

$$\sigma_{\text{AVR RMS}}(n) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\text{AVR RMS(n)} - \mu)^2} \;\; for\ 1 \le n \le n_c \tag{26}$$

If a rest-region is placed before the abrupt change point (n_c), or the start point of a dynamic motion, the rest-region is the before-rest-region. The after-rest-region is placed after the start point of a dynamic motion. Once an abrupt change is detected, the proposed algorithm identifies the zone before the abrupt change to find a *before-rest-region*. If the *before-rest-region* exists, the proposed algorithm starts to find an *after-rest-region*. If an optimal solution of the problem (27) is feasible, a *before-rest-region* exists and b is the starting point of a dynamic motion.

$$\begin{array}{ll} \underset{b}{\text{maximize}} & b \\ \text{subject to} & t_{rest} \leq b \leq n_c \\ & \mu_{\text{AVR RMS}}(b) < \mu_{THrest} \\ & \sigma_{\text{AVR RMS}}(b) < \sigma_{THrest} \end{array} \tag{27}$$

An *after-rest-region* exists as well, provided that an optimal solution of the problem (28) is a feasible solution.

$$\begin{array}{ll} \underset{c}{\text{minimize}} & c \\ \text{subject to} & n_c + t_{DynMin} + \ t_{rest} \leq c \leq n_c + t_{DynMax} \\ & \mu_{\text{AVR RMS}}(c) < \mu_{THrest} \\ & \sigma_{\text{AVR RMS}}(c) < \sigma_{THrest} \end{array} \tag{28}$$

Note that the time length of the dynamic motion is limited to t_{DynMin} (1 second) to t_{DynMax} (3 seconds). Figure 24 shows the ideal dynamic motion as described.

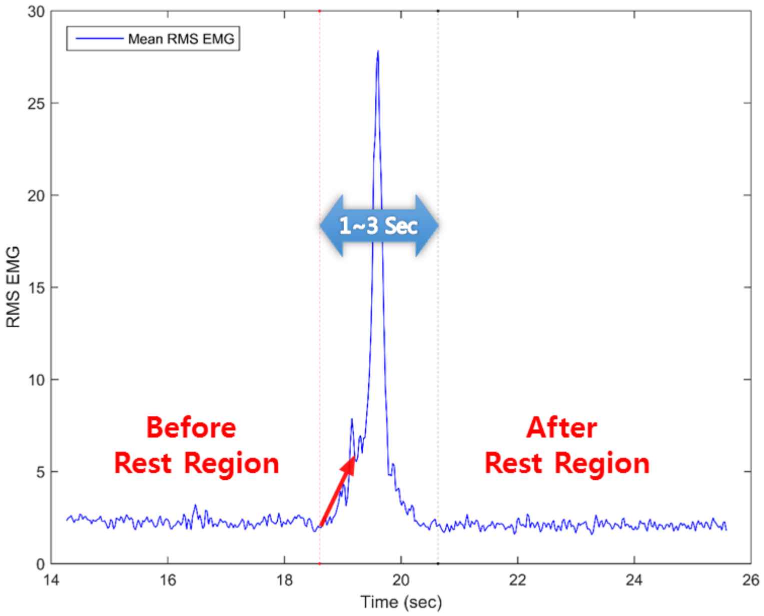


Figure 24 An ideal dynamic motion

Processes of the proposed myoelectric interface

The proposed myoelectric interface generates two types of commands: discrete and continuous. The discrete commands are used to change *manipulation modes* shown in Figure 25 for the robot manipulator. The manipulation modes are considered as a finite state machine to control a robot manipulator with multiple degrees of freedom. To generate discrete commands, dynamic hand motions are classified by the sequence-based myoelectric classification (SMC) described in Chapter 3. On the other hand, the continuous commands give variations ranging from zero to the maximum velocity in the robot movements. Muscle force and orientation (roll, pitch, and yaw) of a forearm of an operator are exploited to yield continuous commands.

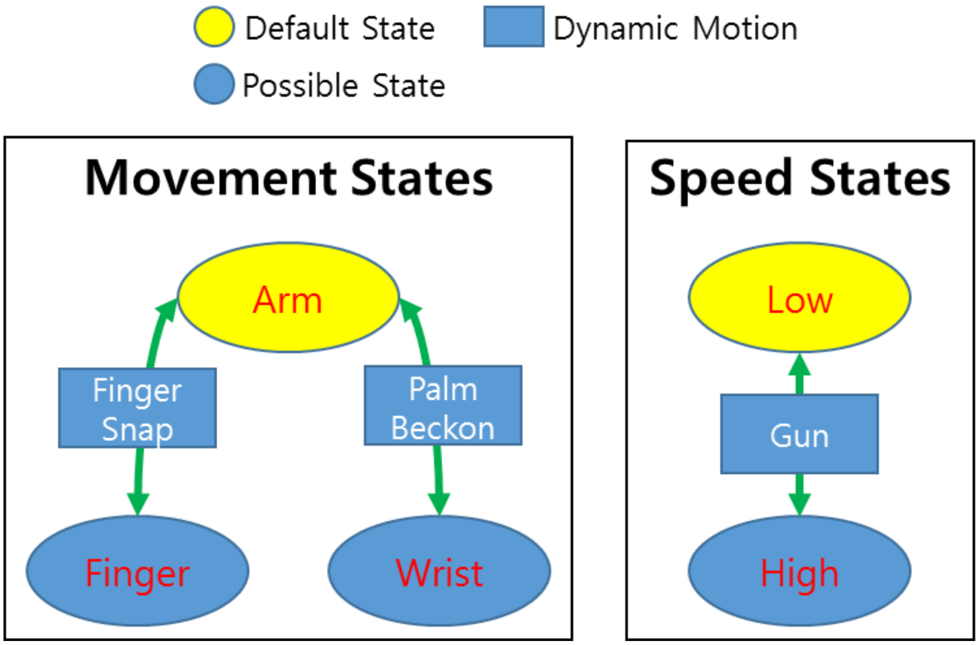


Figure 25 Manipulation modes described by a finite state machine in the proposed myoelectric interface. Each certain dynamic motion is responsible for changing a specific state; ‘finger snap’ dynamic motion changes moving states (between ‘Arm’ and ‘Finger’ mode), ‘palm beckon’ changes moving states (between ‘Arm’ and ‘Wrist’ mode), and ‘Gun’ switches speed states between ‘Low’ and ‘High’.

The manipulation modes consist of two types; *movement states* and *speed states*. For the movement states, there are *arm state* for translation, *wrist state* for wrist rotation, and *finger state* for opening and closing of a gripper. For the speed states, *high* and *low* states exist. The maximum velocity of the continuous commands is the full speed of the preset maximum speed in the high state and it becomes the half speed (50%) of that in the low state. To control the manipulation modes, the proposed myoelectric interface uses three dynamic hand motions; *Finger Snap* (FS), *Palm Beckon* (PB), and *Gun* (GU) as shown in Figure 26. The FS motion changes states between the arm and finger states. The PB motion controls the arm and wrist states. The GU motion handles the speed states, both the high and low states.

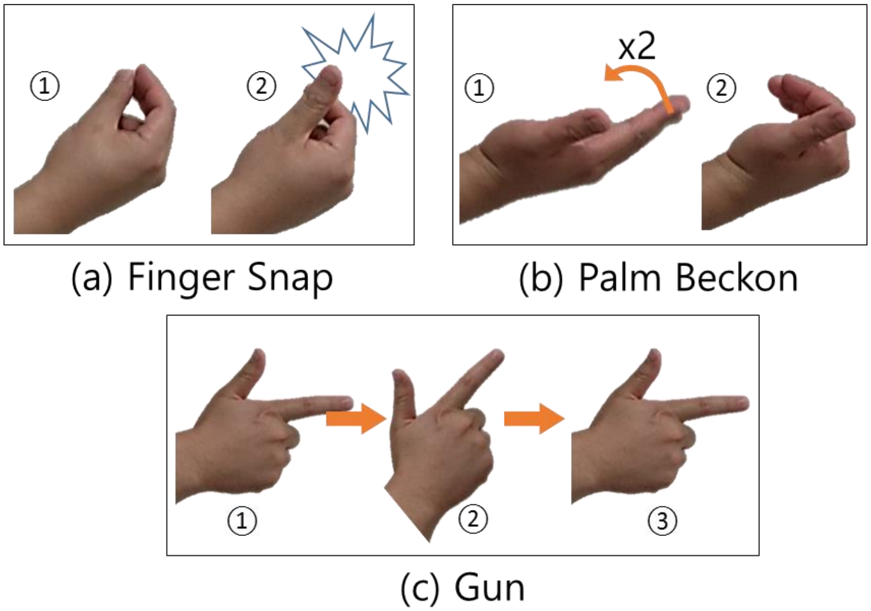


Figure 26 Three dynamic motion used in the proposed myoelectric interface; (a) Finger Snap, (b) Palm Beckon, and (c) Gun

To recognize the dynamic motions in real-time, the myoelectric interface has a process hereafter called *Discrete Command Generator* (DCG) shown in Figure 27. There are three main tasks in the DCG; 1) catching a dynamic motion (called as *CDM* process) in real-time, 2) classifying the caught dynamic motion into FS, PB, GU, and *unknown* motions, and 3) changing the manipulation modes by using the classified dynamic motion. The flowchart of the three tasks is described in Figure 27. In order to reduce the chance of mal-operations caused by classifying wrong dynamic motions, the unknown motion, which the classifier yields with low confidence, is introduced in the DCG. The unknown motion is determined by limiting the maximum correlation coefficient calculated by the sequence-based myoelectric classification, SMC. If the maximum correlation coefficient is under a pre-defined level (here 0.5 is chosen), the classified output of the SMC is stated as the unknown motion.

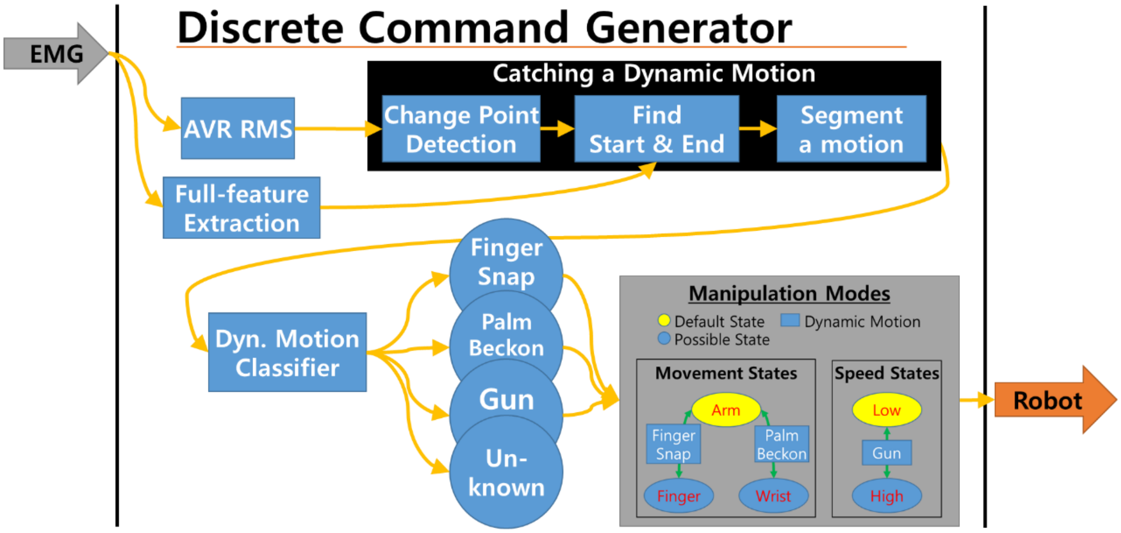


Figure 27 A process of discrete command generator (DCG)

Another key process, called here *Continuous Command Generator* (CCG), in the proposed myoelectric interface is a process of generating continuous commands with information from the EMG and IMU sensors. Figure 28 shows the brief workflow of the process to generate continuous commands. In the process, RMS values from 8-EMG sensors are averaged to determine the muscle force in a proportional manner. The mean of the RMS values is then used to activate or inactivate the CCG process with *hysteresis*. If a mean RMS value is above the active line, the CCG is activated and if a mean RMS value is below the inactive line, the CCG is inactivated as shown in Figure 29. The hysteresis checks four states, activation/inactivation, positive trigger (states changing from inactivation to activation), and negative trigger (states changing from activation to inactivation). The CCG generates continuous commands only while it is activated. In the event of a positive trigger, the CCG remembers the roll, pitch, and yaw (RPY) at that time as the initial RPY to calculate changes of RPY in orientation compared to the initial RPY. The changes of RPY are used to generate continuous commands. In the event of a negative trigger, CCG stops the robot manipulator. A particular point in the processing of RPY from the IMU sensor is that raw RPYs have discontinuity caused from the indiscrimination of -180 and +180 degrees in the IMU sensor. The discontinuity is mediated before using the RPY for continuous commands. The IMU quantities (RPY) removing the discontinuity, $\widehat{\mathrm{IMU}_n}$, are calculated by:

$$\widehat{\mathrm{IMU}_n} = \begin{cases} \mathrm{IMU}_n + 360^\circ & , if\ \Delta\mathrm{IMU}_n < -180^\circ\ and\ \Delta\mathrm{sgn}_n \neq 0 \\ \mathrm{IMU}_n - 360^\circ & , if\ \Delta\mathrm{IMU}_n > 180^\circ\ and\ \Delta\mathrm{sgn}_n \neq 0 \\ \quad \mathrm{IMU}_n & , otherwise \end{cases} \tag{29}$$

where sgn(·) is a sign function, IMU_n is one of measured IMU quantities (roll, pitch, and yaw) at the current time n, $\Delta\mathrm{IMU}_n$= IMU_n-$\mathrm{IMU}_{n\text{-}1}$, and $\Delta\mathrm{sgn}_n$=sgn(IMU_n)-sgn($\mathrm{IMU}_{n\text{-}1}$).

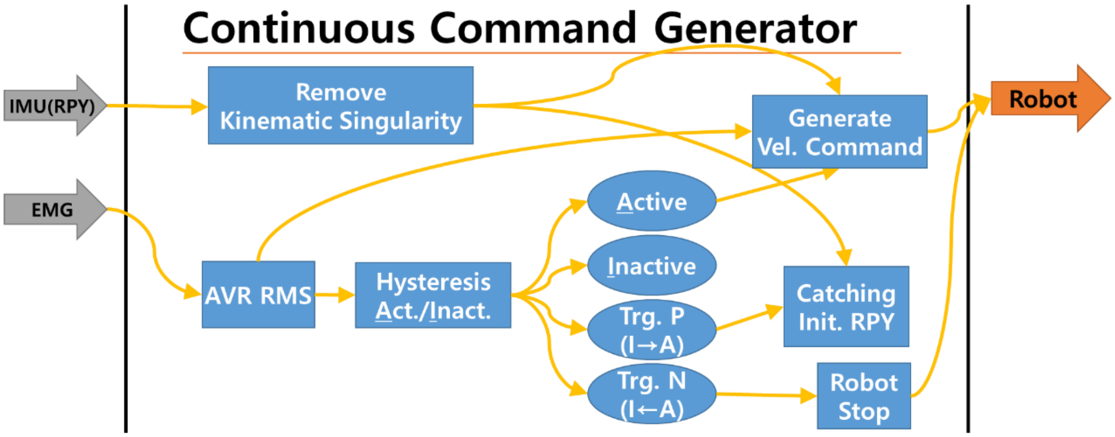


Figure 28 A process of continuous command generator (CCG)

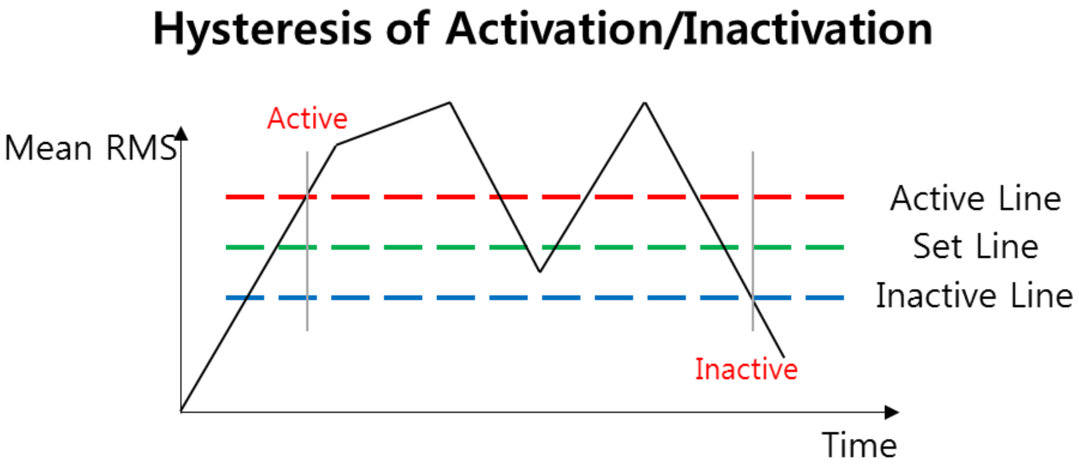


Figure 29 Hysteresis band check

Change-point Detection

Change-point detection is a well-developed field to find whether observations of a random signal still keep following a distribution or an abrupt change has occurred after a certain point [102]. A well-known method in the change-point detection, cumulative summation (CUSUM) algorithm, is modified to detect the starting point of the dynamic motion among many possible abrupt changes in the mean RMS values of EMG signals in an online manner. This method is called *catching-a-dynamic-motion* (CDM) in the proposed myoelectric interface.

Let $X = \{x_1, \dots, x_n\}$ be a discrete random signal with independent and identically distributed samples until the current time n. Each sample follows a probability density function (PDF), $p(x, \theta)$, with a deterministic parameter θ. In order to determine whether an abrupt change at a time n_c between the first time to the current time n exists, there are two hypotheses, *no change* hypothesis ($\mathcal{H}_0$) and *one change at a time* n_c hypothesis ($\mathcal{H}_1$). The PDF of X under the $\mathcal{H}_0$ hypothesis is given by:

$$p_{X|\mathcal{H}_0} = \prod_{k=1}^{n} p(x_k, \theta_0) \tag{30}$$

The PDF of X under the $\mathcal{H}_1$ hypothesis is given by:

$$p_{X|\mathcal{H}_1} = \prod_{k=1}^{n_c-1} p(x_k, \theta_0) \prod_{k=n_c}^{n} p(x_k, \theta_1) \tag{31}$$

Based on setting the hypotheses, the CUSUM algorithm requires two calculation steps: *detection step* using the likelihood ratio test to decide whether the abrupt change has occurred and the *estimation step* using the maximum likelihood estimation for the PDF of

X under $\mathcal{H}_1$ to find an estimated abrupt change time $\widehat{n_c}$. In the detection step, the *log-likelihood ratio* L_X is used:

$$L_X(n, n_c) = \ln\left[\frac{p_{X|\mathcal{H}_1}(n, n_c)}{p_{X|\mathcal{H}_0}(n)}\right] = \sum_{k=n_c}^{n} \ln\left[\frac{p(x_k, \theta_1)}{p(x_k, \theta_0)}\right] \tag{32}$$

With the definition of the log-likelihood ratio, the *generalized log-likelihood ratio* G_X is defined by:

$$G_X(n) = \max_{1 \leq n_c \leq n} L_X(n, n_c) = \max_{1 \leq n_c \leq n} \sum_{k=n_c}^{n} \ln\left[\frac{p(x_k, \theta_1)}{p(x_k, \theta_0)}\right] \tag{33}$$

If $G_X > h$, $\mathcal{H}_1$ is selected, otherwise $\mathcal{H}_0$ is chosen, where *h* is a threshold parameter. Once an abrupt change is detected in the detection step, an estimated abrupt change time $\widehat{n_c}$ is calculated, which maximizes the PDF of X under $\mathcal{H}_1$ by:

$$\begin{aligned} \widehat{n_c} = \arg\max_{1 \leq n_c \leq n} p_{X|\mathcal{H}_1}(n, n_c) = \arg\max_{1 \leq n_c \leq n} L_X(n, n_c) = \\ = \arg\max_{1 \leq n_c \leq n} \sum_{k=n_c}^{n} \ln\left[\frac{p(x_k, \theta_1)}{p(x_k, \theta_0)}\right] \end{aligned} \tag{34}$$

In order to write the steps above in a recursive form, the *instantaneous log-likelihood ratio* at time *n*, s_n, and the cumulative sum of the instantaneous log-likelihood ratio, S_n, are defined by:

$$s_n = L_X(n, n) = \ln\left[\frac{p(x_n, \theta_1)}{p(x_n, \theta_0)}\right] \tag{35}$$

$$S_n = \sum_{k=0}^{n} s_k = S_{n-1} + s_n \tag{36}$$

With the definition (36), the log-likelihood ratio defined in (32) can be written as:

$$L_X(n, n_c) = S_n - S_{n_c - 1} \tag{37}$$

The generalized log-likelihood ratio is rewritten as:

$$G_X(n) = \sup\{G_X(n-1) + s_n, 0\} \tag{38}$$

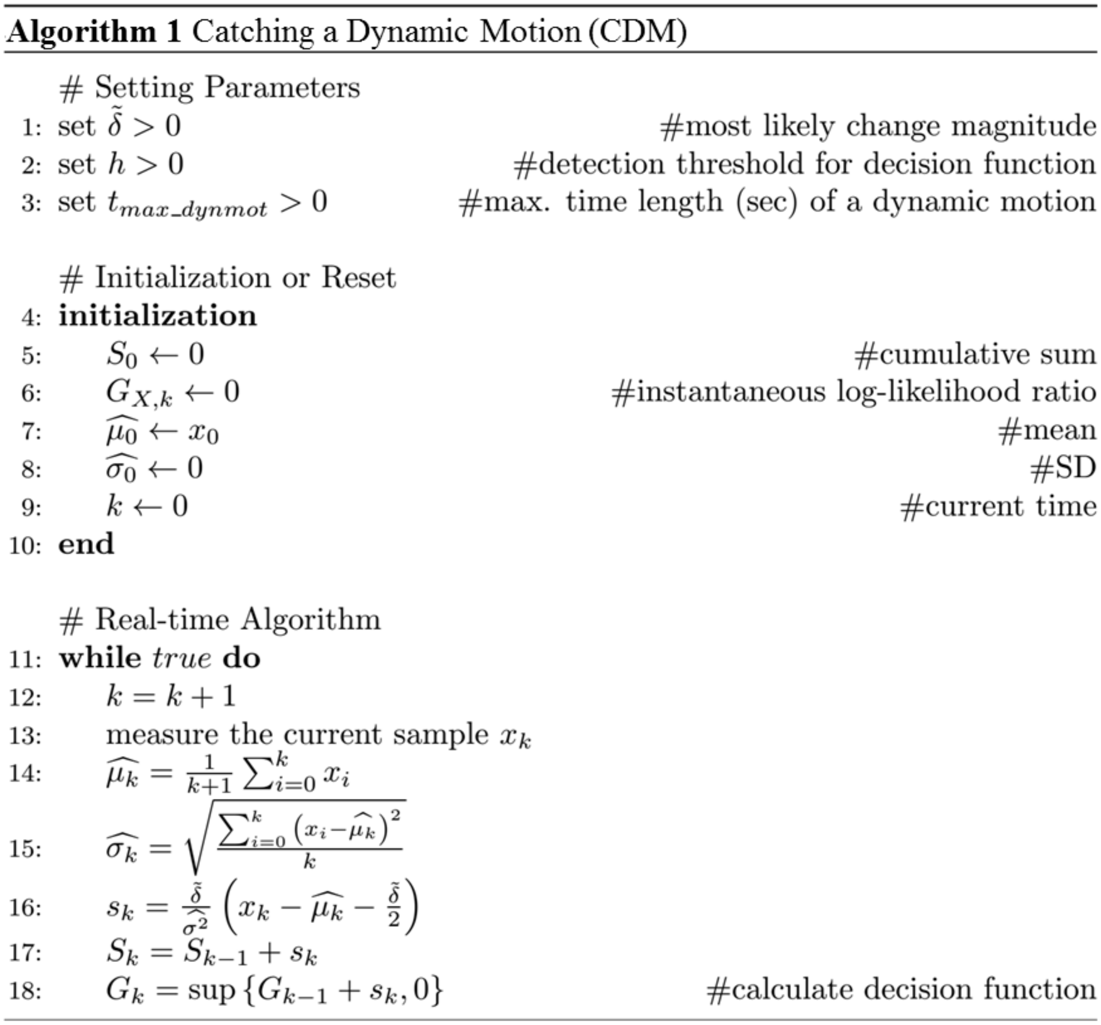
The estimated abrupt change time is now calculated by:

$$\widehat{n_c} = \underset{1 \le n_c \le n}{\arg\max}\, S_{n_c - 1} \tag{39}$$

More details about the CUSUM algorithm are available in [103].

In order to implement the DCG process in the proposed myoelectric interface, detecting a dynamic motion in real-time is a key task. Based on the change-point detection algorithm explained above, a pseudocode of the CDM process with an assumption that the samples $x_n$of AVR RMS values follow a Gaussian distribution is described in Table 13. By using CDM, change points that can be a possible starting point of the dynamic motion are detected. If a change point is detected, CDM looks for a before-rest-region prior to the change point. If CDM finds the before-rest-region, then it starts to find an after-rest-region of a dynamic motion while keeping watch to prevent infinite waiting for an after-rest-region. In the definition of a dynamic motion in the proposed myoelectric interface, the minimum and maximum time lengths of the dynamic motion are set as 1 sec and 3 sec, respectively. If the time waiting for an after-rest-region is more than the maximum time length of the dynamic motion, the CDM cancels the wait for an after-rest-region. Or if the CDM finds an after-rest-region, it cuts a segment of EMG feature vectors calculated from the *full-feature extraction* block in DCG from the start to the end of the dynamic motion, and passes the segment to the dynamic motion classifier.

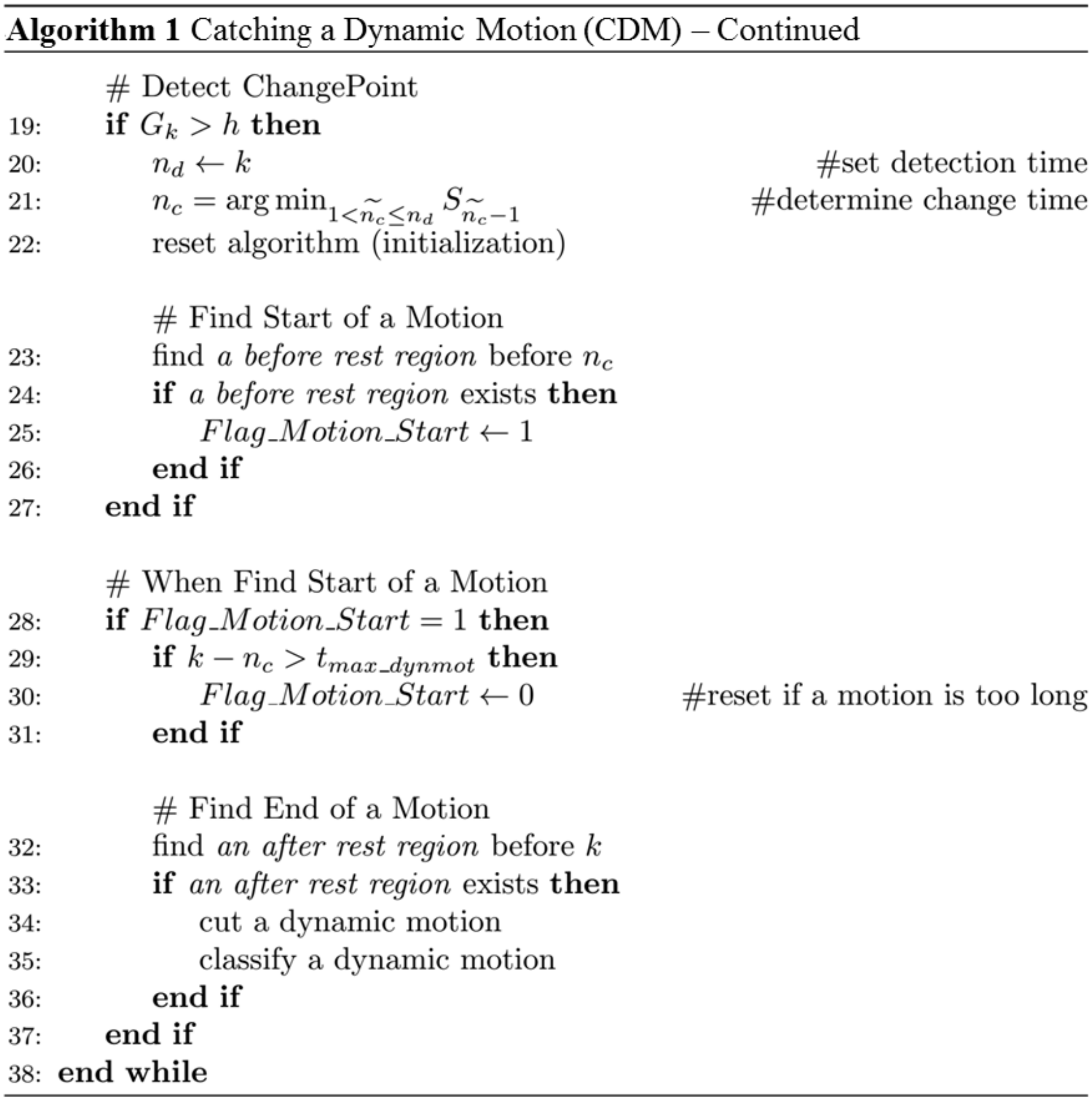
Table 13 Pseudocode of the catching-a-dynamic-motion (CDM)



Algorithm 1 Catching a Dynamic Motion (CDM)

```
# Setting Parameters
set $\tilde{\delta} > 0$                    #most likely change magnitude
set $h > 0$                                 #detection threshold for decision function
set $t_{max\_dynmot} > 0$                   #max. time length (sec) of a dynamic motion

# Initialization or Reset
initialization
    $S_0 \leftarrow 0$                      #cumulative sum
    $G_{X,k} \leftarrow 0$                  #instantaneous log-likelihood ratio
    $\widehat{\mu_0} \leftarrow x_0$        #mean
    $\widehat{\sigma_0} \leftarrow 0$       #SD
    $k \leftarrow 0$                        #current time
end

# Real-time Algorithm
while $true$ do
    $k = k + 1$
    measure the current sample $x_k$
    $\widehat{\mu_k} = \frac{1}{k+1} \sum_{i=0}^{k} x_i$
    $\widehat{\sigma_k} = \sqrt{\frac{\sum_{i=0}^{k} \left(x_i - \widehat{\mu_k}\right)^2}{k}}$
    $s_k = \frac{\tilde{\delta}}{\widehat{\sigma^2}} \left(x_k - \widehat{\mu_k} - \frac{\tilde{\delta}}{2}\right)$
    $S_k = S_{k-1} + s_k$
    $G_k = \sup\{G_{k-1} + s_k, 0\}$        #calculate decision function
```

Table 13 Continued



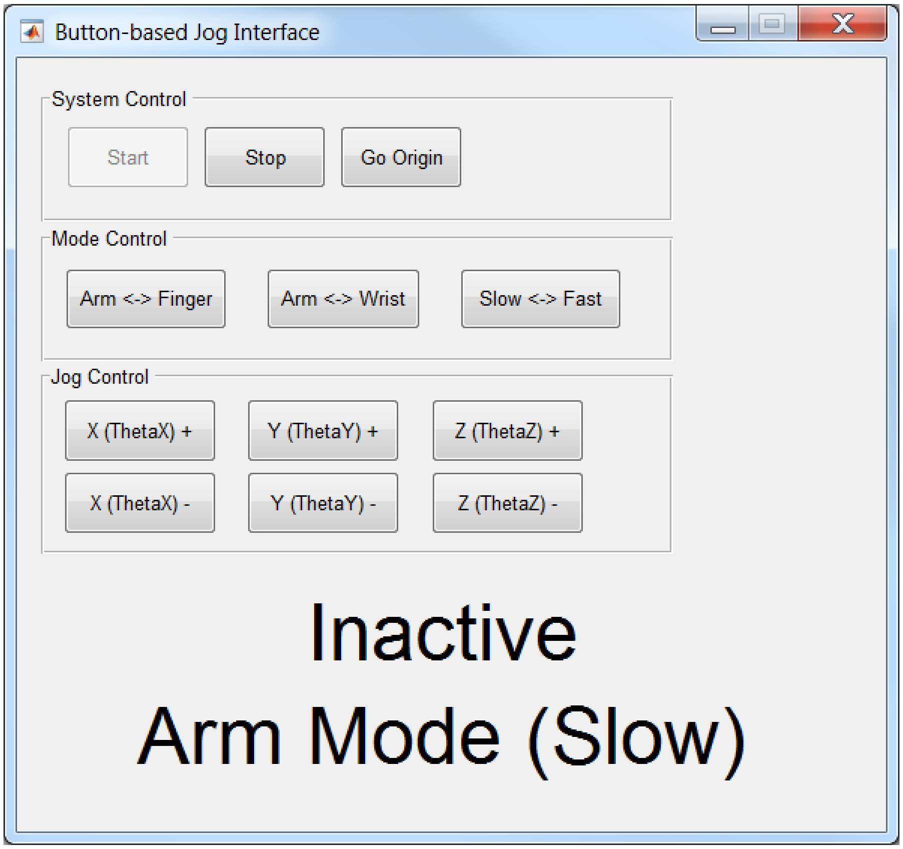
Algorithm 1 Catching a Dynamic Motion (CDM) – Continued

```
    # Detect ChangePoint
    if $G_k > h$ then
        $n_d \leftarrow k$                                                    #set detection time
        $n_c = \arg\min_{1<\widetilde{n_c}\leq n_d} S_{\widetilde{n_c}-1}$    #determine change time
        reset algorithm (initialization)

        # Find Start of a Motion
        find *a before rest region* before $n_c$
        if *a before rest region* exists then
            $Flag\_Motion\_Start \leftarrow 1$
        end if
    end if

    # When Find Start of a Motion
    if $Flag\_Motion\_Start = 1$ then
        if $k - n_c > t_{max\_dynmot}$ then
            $Flag\_Motion\_Start \leftarrow 0$                                #reset if a motion is too long
        end if

        # Find End of a Motion
        find *an after rest region* before $k$
        if *an after rest region* exists then
            cut a dynamic motion
            classify a dynamic motion
        end if
    end if
end while
```

4.2.2 GUI Button-based Jog Interface

For comparison purposes, a GUI button-based jog interface was developed using Matlab (Figure 30). The jog interface has all the required functions to control the robot manipulator as the proposed myoelectric interface; three buttons that can control three movement states (arm, wrist, and finger states) and two speed states (high and low states);

and six buttons that can control the direction of the 3 DOFs for arm translation/wrist rotation (or 1 DOF for a gripper) at each movement state (positive and negative X, Y, and Z in the arm state, Theta X, Theta Y, and Theta Z in the wrist state, or opening and closing a gripper in the finger state). The “Go Origin” button moves the robot manipulator to the pre-defined starting position when it is pushed.



Inactive

Arm Mode (Slow)

Figure 30 GUI button-based jog interface made by Matlab GUI

4.2.3 Protocol of Experiments

In order to estimate the performance of the proposed myoelectric interface and the GUI button-based jog interface, a pre-defined task, a *go-and-pickup* task, is repeated. This task consists of three sections: 1) approaching to the target object from a pre-defined start point to an object to grasp (*Go* motion), 2) grasping the object using the robot gripper (*Grasp* motion), and 3) picking up the object until the final goal (*Pickup* motion). The task takes approximately a minute and is repeated 30 times for evaluating the performance of both interfaces. Before starting the experiment, participants have to practice about 10 minutes to become familiar with each of the interfaces. For the experiment, healthy intact participants (who had no diseases or disorders in hand gestures and were from 18 to 40 years old) were recruited to be involved in the experiment. The experiments were approved by the Texas A&M University Institutional Review Board (IRB) and informed consent forms were signed by all participants.

In the proposed myoelectric interface, a training procedure is necessary for the classifier to learn dynamic motions of a participant. To train the classifier, three dynamic motions (FS, PB, and GU) are repeated 3 times respectively. After the training, in order to validate the classifier performance (classification accuracy in real-time), a validation procedure repeating three dynamic motions 10 times each is carried out. This validation procedure is conducted before repeating the pre-defined task to determine the performance of DCG. The real-time classification accuracy is presented in the Results section of this chapter.

To measure the orientation of a forearm and myoelectric signals in the proposed myoelectric interface, a Myo armband designed by Thalmic Labs was used. The Myo armband includes eight EMG sensors and one IMU sensor. The sampling rates of the Myo armband are 200 Hz (for the EMG sensors) and 50 Hz (for the IMU sensor). The Myo armband was located on the forearm the participants preferred. For the robot manipulator in the experiment, a MICO robot arm produced by Kinova was used. The MICO robot arm includes 7 DOFs, 3 DOFs for a robot arm (arm translation), 3 DOFs for a robot wrist (wrist rotation), and 1 DOF for a gripper (open/close). The experimental setting is shown in Figure 31.

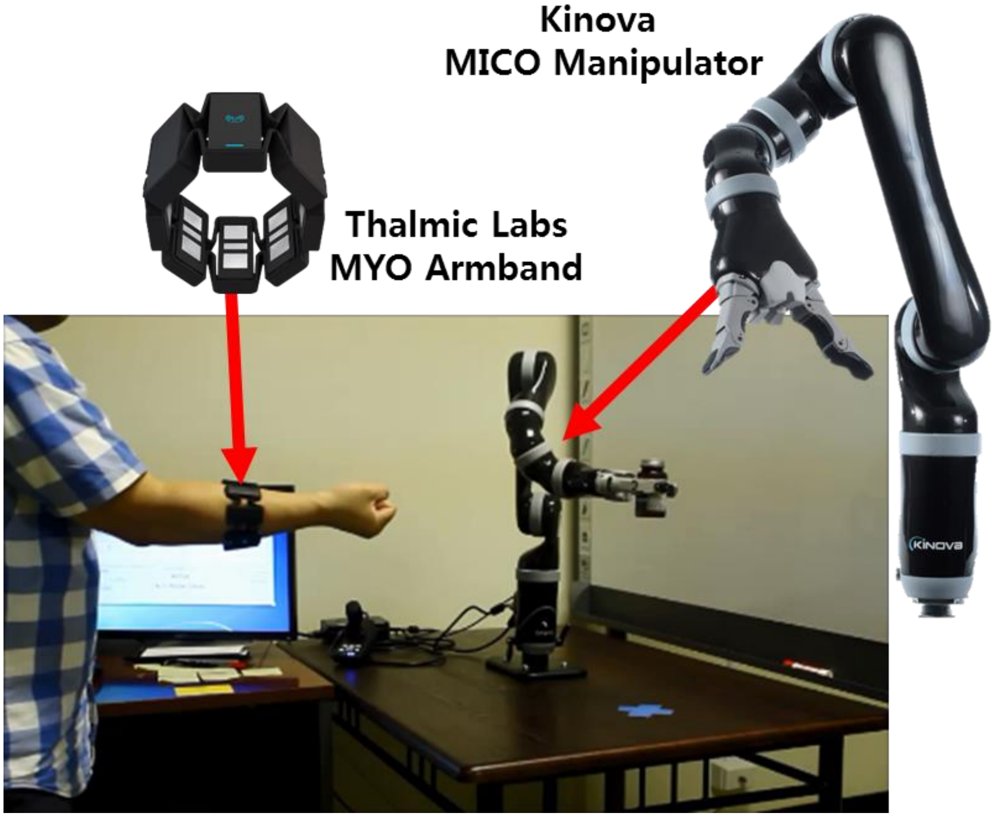


Figure 31 Experimental setting for the myoelectric and jog interfaces

4.2.4 Performance Measures

Real-time Classification Accuracy

In order to estimate the performance of the dynamic hand motion classification in real-time, a validation task was performed. Each of the three dynamic motions was repeated 10 times, and the results were recorded. The number of *target classes* we performed is 3. However, the number of classified *output classes* is 4 because the classifier categorizes an incoming dynamic motion into 4 groups (three dynamic motions and an unknown motion).

Time-Measures

In order to estimate the performance of each interface, five time-measures are introduced: *Experiment Time*, *Move Time*, *Stop Time*, *State-change Time*, and *Non-state-change Time*. Figure 32 shows a plot of normalized positions of the robot manipulator during a go-and-pickup task done by a subject. The positions are normalized by their mean values and standard deviations because only the moving states of the robot manipulator are important to define the time-measures. The experiment time is defined by a time period from starting a movement of the robot manipulator, (a), in Figure 32, to finishing a pickup motion, (f). The finishing time of a go-and-pickup task, (f), is defined as a time when the robot manipulator passes an empirically-determined goal position (0.15 m in Z-axis of a robot coordinate system) during a pickup motion after grasping the object.

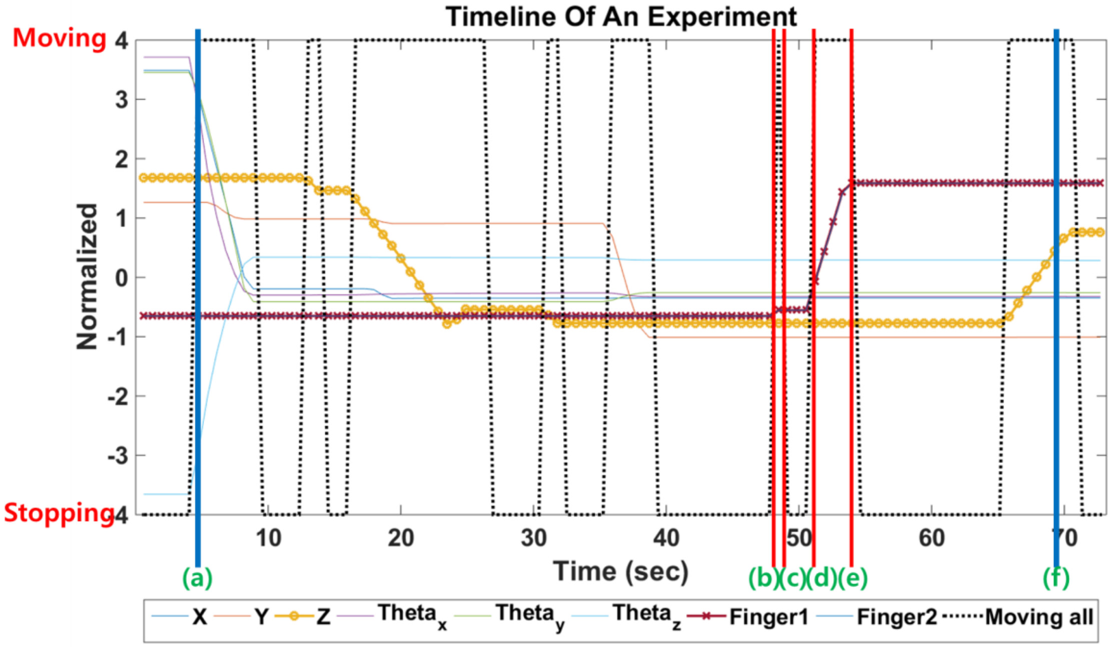


Figure 32 Normalized position information of the robot arm; (a) start of an experiment, (b) start of 1st finger movement, (c) end of 1st finger movement, (d) start of 2nd finger movement, (e) end of 2nd finger movement, and (f) end of the experiment (Z-axis position is above 0.15 (m))

The other time-measures (move time, stop time, state-change time, and non-state-change time) are defined with regard to the changes of the manipulation mode as shown in Figure 33. The moving time is the total time the robot manipulator moves during a task. It is simply calculated by summing up time periods of all moving states; (a), (b), (c), (d), (e), (f), (g), and (h) in Figure 33. The stop time is the total time when the robot manipulator is stopped and is calculated by subtracting the move time from the experiment time. The state-change time is the time spent to change the manipulation mode. The state-change time is calculated by summing up all time periods when the robot manipulator is stopped to change the manipulation mode as performing a dynamic motion; (A) and (B) in Figure 33 are the time periods for the changes of the manipulation mode. The non-state-change time is easily calculated by subtracting the state-change time from the stopping time.

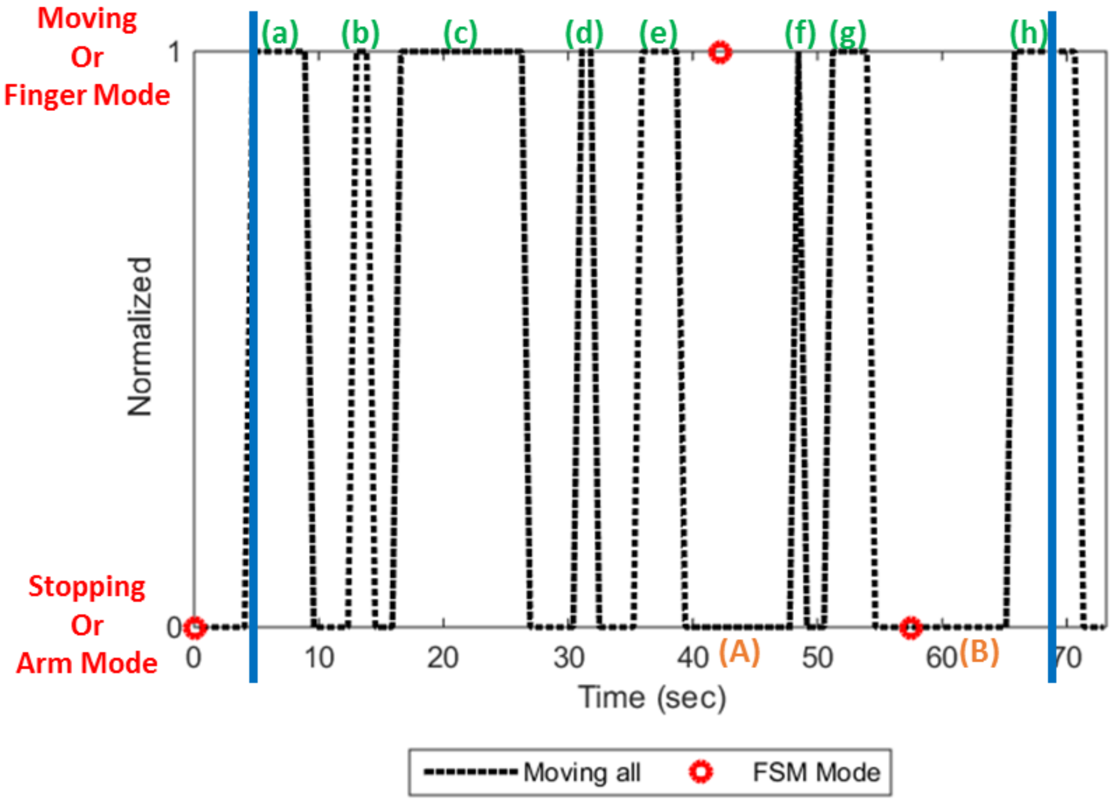


Figure 33 Changes of the manipulation mode over an experiment; (a), (b), (c), (d), (e), and (h) are time periods of moving the arm part; (f) and (g) are time periods of moving the finger part; (A) and (B) are time periods of stopping the robot manipulator

Path Efficiency

In order to determine the performance of moving paths the robot manipulator follows to complete a go-and-pickup task via the different interfaces, a *Path Efficiency* is introduced [104]. The path efficiency is calculated by a ratio of the straight line between the start position of the robot manipulator and the position of the target object over an actual moving path of the robot manipulator to the target object. The path efficiency is defined by:

$$\text{Path Efficiency (\%)} = \frac{\text{Dist}_{Straight}}{\text{Dist}_{Actual}} \times 100 \tag{40}$$

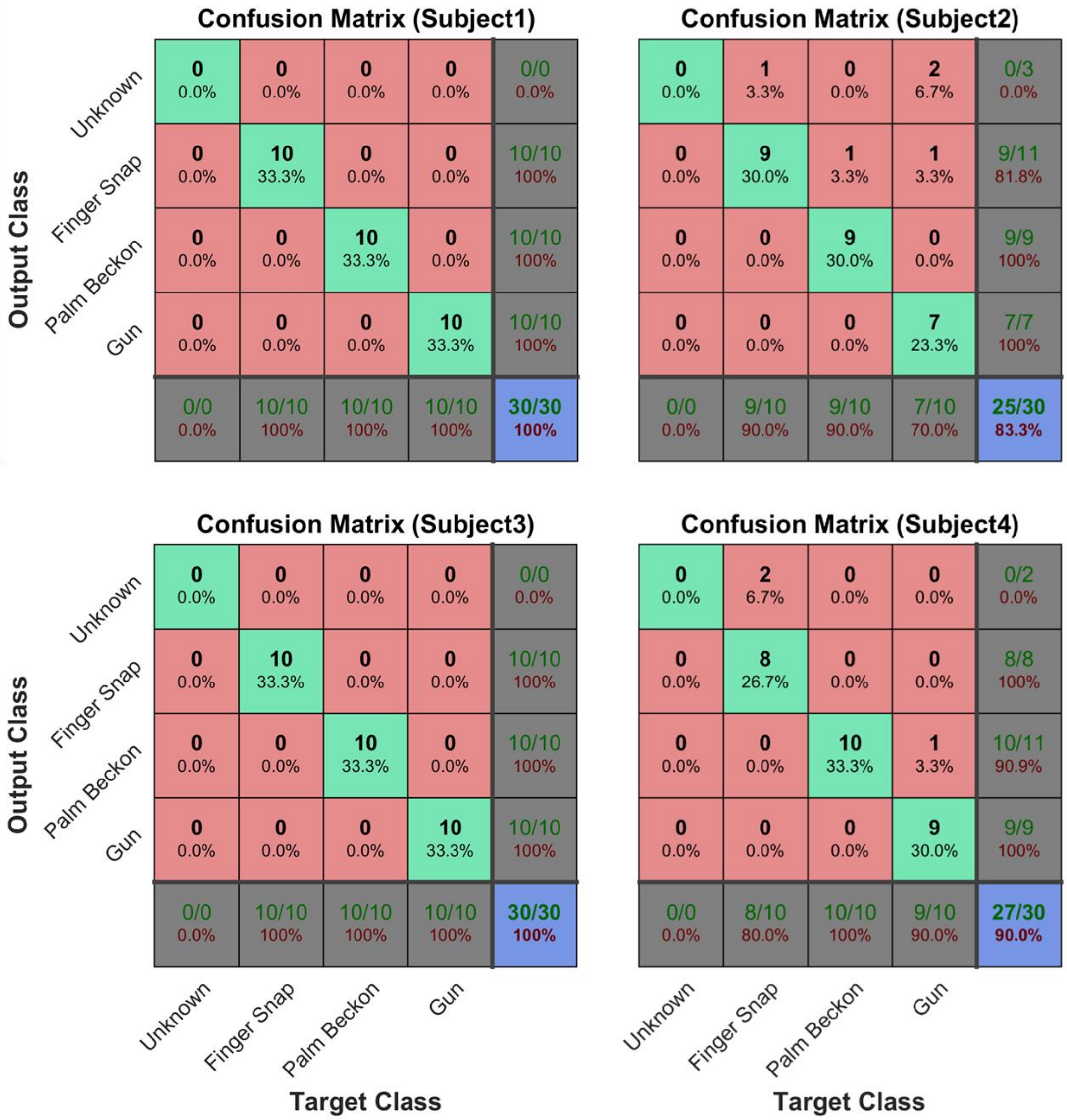
As comparing path efficiencies of the myoelectric and jog interfaces, the similarity of moving paths that the robot manipulator moves is indirectly inferred.

4.3 Results

In the experiment, each human subject repeated the go-and-pickup task 30 times with the jog interface and the myoelectric interface, respectively. In total, seven human subjects participated in the experiment. In general, human beings have their own learning curve to adapt to an interface method [105]. With this tendency, lastly-and-well-performed 10 trials among 30 trials are selected for the analysis of the time-measures and the path efficiency.

4.3.1 Real-time Classification Accuracy

In order to verify the performance of the discrete command generator, DCG, in real-time, before starting the experiments with the myoelectric interface, the real-time classification accuracy of DCG was tested. To do so, a total of 30 dynamic motions (10 FSs, 10 PBs, and 10 GUs) were performed in real-time for each subject.



Confusion Matrix (Subject1)

Target Class

Output Class	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0/0 0.0%
Finger Snap	0 0.0%	10 33.3%	0 0.0%	0 0.0%	10/10 100%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	0 0.0%	10/10 100%
Gun	0 0.0%	0 0.0%	0 0.0%	10 33.3%	10/10 100%
	0/0 0.0%	10/10 100%	10/10 100%	10/10 100%	30/30 100%

Confusion Matrix (Subject2)

Target Class

Output Class	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	1 3.3%	0 0.0%	2 6.7%	0/3 0.0%
Finger Snap	0 0.0%	9 30.0%	1 3.3%	1 3.3%	9/11 81.8%
Palm Beckon	0 0.0%	0 0.0%	9 30.0%	0 0.0%	9/9 100%
Gun	0 0.0%	0 0.0%	0 0.0%	7 23.3%	7/7 100%
	0/0 0.0%	9/10 90.0%	9/10 90.0%	7/10 70.0%	25/30 83.3%

Confusion Matrix (Subject3)

Target Class

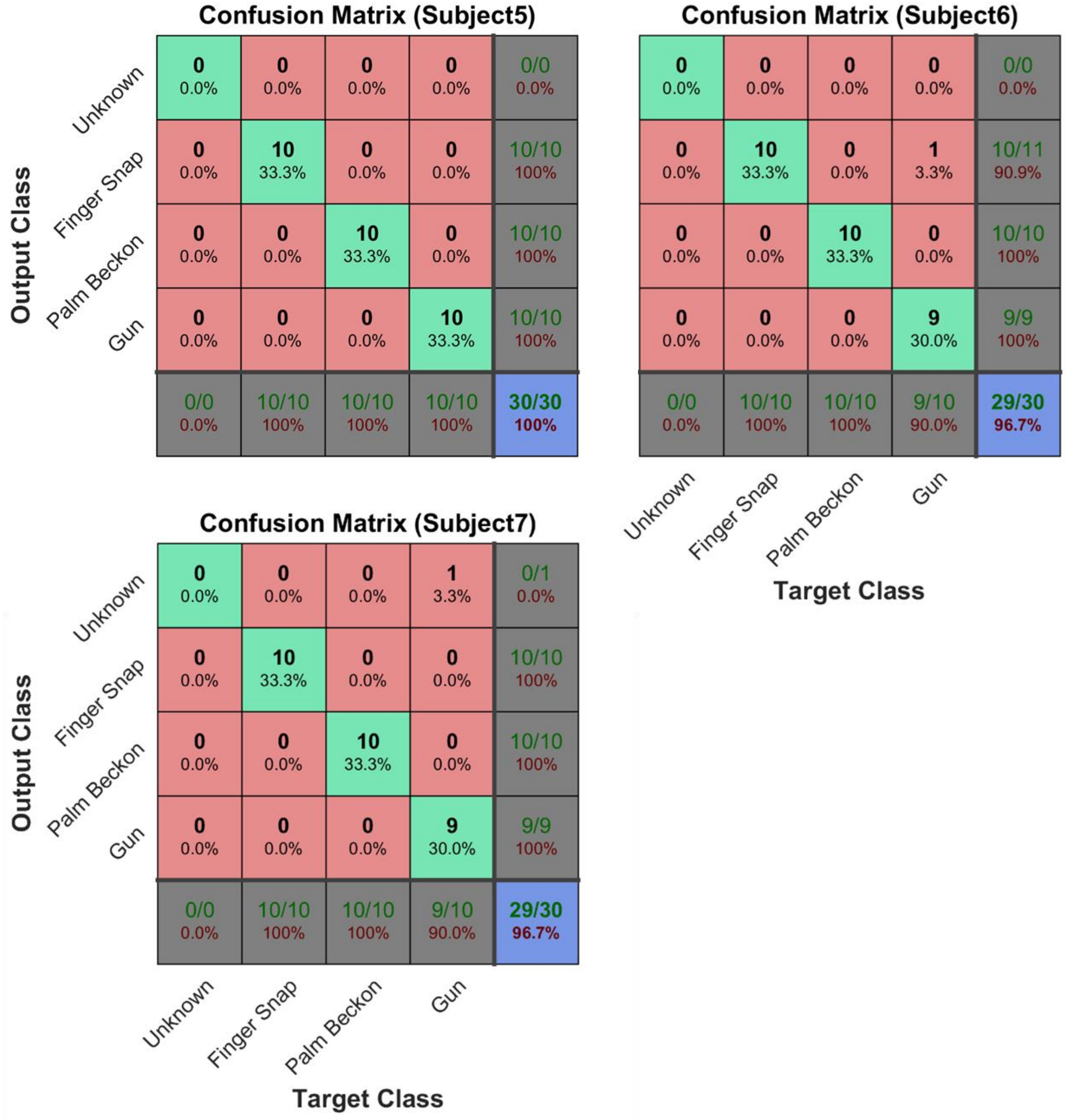
Output Class	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0/0 0.0%
Finger Snap	0 0.0%	10 33.3%	0 0.0%	0 0.0%	10/10 100%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	0 0.0%	10/10 100%
Gun	0 0.0%	0 0.0%	0 0.0%	10 33.3%	10/10 100%
	0/0 0.0%	10/10 100%	10/10 100%	10/10 100%	30/30 100%

Confusion Matrix (Subject4)

Target Class

Output Class	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	2 6.7%	0 0.0%	0 0.0%	0/2 0.0%
Finger Snap	0 0.0%	8 26.7%	0 0.0%	0 0.0%	8/8 100%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	1 3.3%	10/11 90.9%
Gun	0 0.0%	0 0.0%	0 0.0%	9 30.0%	9/9 100%
	0/0 0.0%	8/10 80.0%	10/10 100%	9/10 90.0%	27/30 90.0%

Figure 34 Confusion matrix of DCG in real-time



Confusion Matrix (Subject5)

Output Class

	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0/0 0.0%
Finger Snap	0 0.0%	10 33.3%	0 0.0%	0 0.0%	10/10 100%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	0 0.0%	10/10 100%
Gun	0 0.0%	0 0.0%	0 0.0%	10 33.3%	10/10 100%
	0/0 0.0%	10/10 100%	10/10 100%	10/10 100%	30/30 100%

Confusion Matrix (Subject6)

	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0/0 0.0%
Finger Snap	0 0.0%	10 33.3%	0 0.0%	1 3.3%	10/11 90.9%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	0 0.0%	10/10 100%
Gun	0 0.0%	0 0.0%	0 0.0%	9 30.0%	9/9 100%
	0/0 0.0%	10/10 100%	10/10 100%	9/10 90.0%	29/30 96.7%

Target Class

Confusion Matrix (Subject7)

Output Class

	Unknown	Finger Snap	Palm Beckon	Gun	
Unknown	0 0.0%	0 0.0%	0 0.0%	1 3.3%	0/1 0.0%
Finger Snap	0 0.0%	10 33.3%	0 0.0%	0 0.0%	10/10 100%
Palm Beckon	0 0.0%	0 0.0%	10 33.3%	0 0.0%	10/10 100%
Gun	0 0.0%	0 0.0%	0 0.0%	9 30.0%	9/9 100%
	0/0 0.0%	10/10 100%	10/10 100%	9/10 90.0%	29/30 96.7%

Target Class

Figure 34 Continued

Figure 34 shows the confusion matrices of DCG. The target class means the dynamic motion performed by a human subject, and the output class means the dynamic motion predicted from DCG, respectively. There are no unknown motions performed intentionally. Subjects 1, 3, and 5 showed 100% in the classification accuracy. Subjects 6 and 7 showed 96.7% with only a misclassification. The other subjects, Subject 2 and 4, showed 83.3% and 90%, respectively. The real-time classification accuracy of DCG can be affected by the proficiency of the human subjects. The effect of the classification accuracy of the DCG tested here can be verified from the number of unknown motions in Table 14 as well.

Table 14 The number of unknown motions and average time of a dynamic motion during 30 tasks

Subject No.	**# of Unknown Motions**	**Average Time of a Dynamic Motion (Sec)**
1	3	2.09 s
2	9	2.11 s
3	2	2.03 s
4	12	2.03 s
5	2	2.08 s
6	2	2.02 s
7	1	2.04 s

Table 14 shows the number of *unknown* motions during all 30 tasks in the proposed myoelectric interface. The number of the unknown motions is the total count of dynamic motions caught and classified as unknown (excluding, of course, the known motions: finger snap, palm beckon, and gun dynamic motions). Also, the average time length of a dynamic motion over all 30 tasks is calculated. In Table 14, the time length of a dynamic motion for all subjects is about 2 seconds. The numbers of unknown motions differ by subject because of the variance of their proficiency in executing dynamic motions. The number of unknown motions is highly correlated to the real-time classification accuracy in Figure 34.

4.3.2 Time-Measures

A simple way to measure the performance of an interface method is to calculate the required time to finish a task by using that interface. To estimate the performance of the proposed myoelectric interface, the five time-measures are calculated from the 10 trials out of the total repeated go-and-pickup task (30 trials). Moreover, the performance of the myoelectric interface was compared with the performance of the well-known jog interface. The performance of the jog interface is the goal of the myoelectric interface because the jog interface has a simple and widely-used structure for controlling a robot manipulator. The five time-measures of the jog interface are calculated as well.

The five time-measures (experiment time, move time, stop time, state-change time, and non-state-change time) of the proposed myoelectric interface and the jog interface are plotted for comparison in Figure 35. In general, there is a subject-dependent learning curve to be accustomed to the interfaces. In Figure 35, the red left bar (circle) of each time-measure is the result of the proposed myoelectric interface and the blue right bar (triangle) of each time-measure is the result of the jog interface. The difference between the myoelectric and jog interfaces are calculated by subtracting the result of the jog interface from that of the myoelectric interface.

From the results of all subjects, the experiment times of the jog interface shows that it outperforms of the myoelectric interface. The differences in the experiment time range from 6.1s to 11.9s. The main cause of the difference in the experiment time come from the stop time; in fact, the difference in the move time is under 2 seconds. Most portion of the difference in the experiment time is caused from the stop time. The stop time includes the process of DCG and the time of acting dynamic motions. Therefore, the stop time of the myoelectric interface must be greater than that of the jog interface even if the myoelectric interface performs as good as it can.

The stop time consists of the state-change time and the non-state-change time. The state-change time is a total time to change the manipulation mode. The non-state-change time is the time excluding the state-change time from the stop time; for example, the non-state-change time includes a pause time to decide the next commands and a delay time to activate CCG as increasing the muscle force above the active line. Differences in the state-change time between the myoelectric and jog interfaces range from 4.1s to 7.0s as shown in the Figure 35. Since at least two changes of the manipulation mode are needed to finish a go-and-pickup task, the myoelectric interface needs at least 4 seconds to perform two dynamic motions during a go-and-pickup task; as shown in Table 14, an action of a dynamic motion takes 2 seconds on average. From the results of the difference in the state-change time, the myoelectric interface needs more time periods (0.1s to 3s) to handle the manipulation system than the jog interface except the time of acting a dynamic motion itself.

To see details of the time-measures of each interface, the calculated time-measures when using the myoelectric and jog interfaces are tabulated in Table 15 and Table 16 each.

The mean values and the standard deviation (values in the parentheses) of each time-measure are listed in the tables.

Table 15 Time-measures of the proposed myoelectric interface; standard deviations (SD) are in the parentheses.

Myoelectric Interface

Subject No.	**Exp. Time**	**Move Time**	**Stop Time**	**State-Change Time**	**Non-State-Change Time**
1	33.37s (2.14s)	21.15s (0.97s)	12.22s (2.12s)	7.94s (2.12s)	4.28s (1.75s)
2	37.72s (1.96s)	22.55s (1.13s)	15.16s (2.02s)	9.98s (1.71s)	5.19s (0.89s)
3	35.62s (3.77s)	21.64s (1.53s)	13.98s (3.10s)	9.91s (2.32s)	4.07s (2.09s)
4	34.64s (3.39s)	22.54s (1.43s)	12.10s (2.14s)	6.18s (0.65s)	5.92s (1.98s)
5	34.28s (1.60s)	21.41s (1.24s)	12.87s (1.60s)	9.83s (1.74s)	3.05s (0.84s)
6	32.57s (2.16s)	22.31s (1.09s)	10.26s (1.43s)	6.38s (1.27s)	3.88s (0.80s)
7	32.03s (2.11s)	20.71s (0.76s)	11.31s (1.77s)	8.14s (1.12s)	3.18s (0.89s)

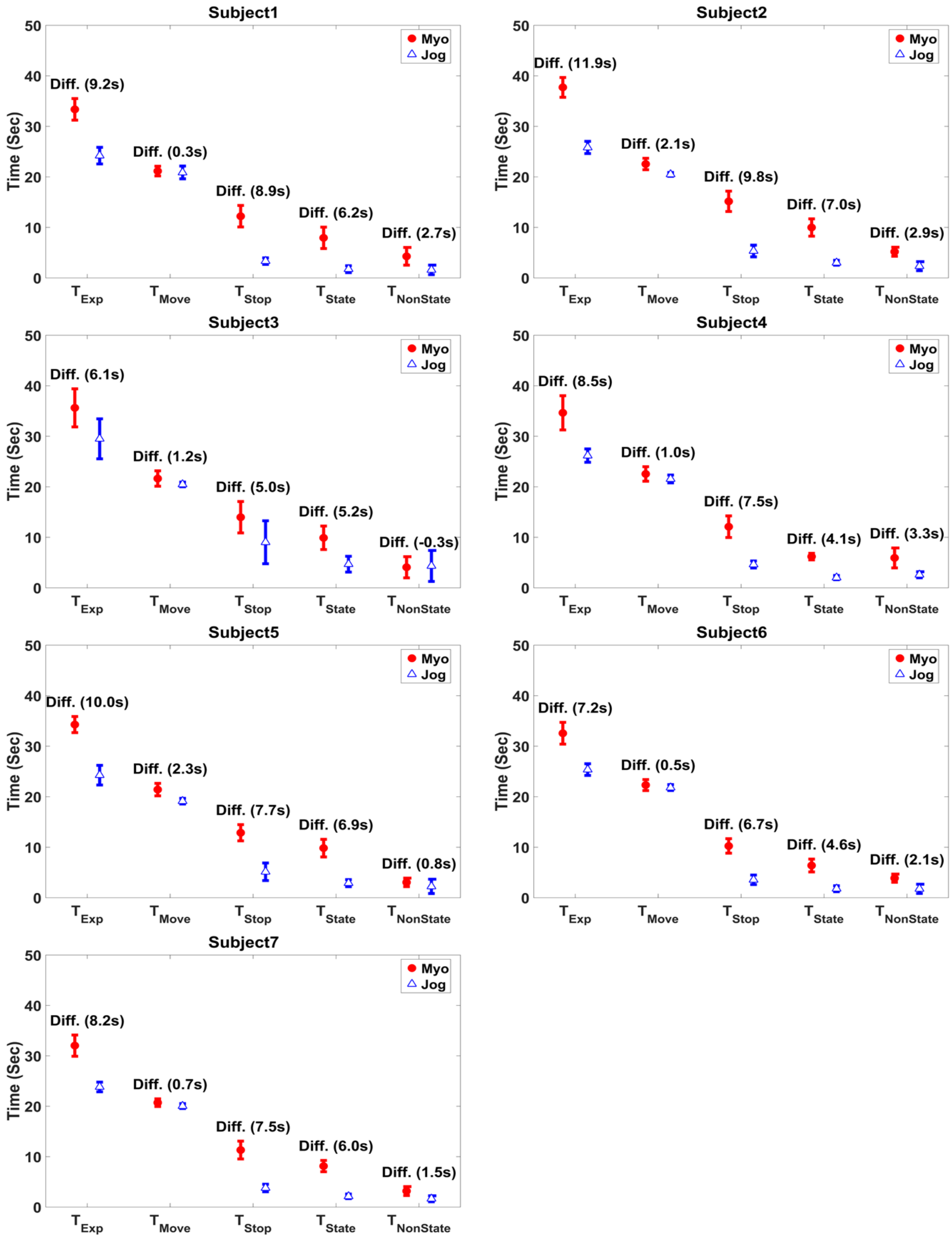


Figure 35 Time-measure plots of myoelectric and jog interface (these plots were drawn from the 10 trials among 30 trials); the left bar (circle) of each time-measure is a result of the proposed myoelectric interface and the right bar (triangle) of each time-measure is a result of the jog interface. The error bar represents mean and standard deviation values of the 10 trials. The differences between the myoelectric and jog interface (Diff = Myo - Jog) are also displayed.

Table 16 Time-measures of the jog interface; SDs are in the parentheses.

Jog Interface

Subject No.	**Exp. Time**	**Move Time**	**Stop Time**	**State-Change Time**	**Non-State-Change Time**
1	24.21s (1.65s)	20.88s (1.27s)	3.33s (0.64s)	1.73s (0.68s)	1.59s (0.94s)
2	25.83s (1.21s)	20.49s (0.37s)	5.33s (1.15s)	3.01s (0.48s)	2.33s (0.89s)
3	29.50s (3.95s)	20.48s (0.46s)	9.02s (4.25s)	4.69s (1.56s)	4.33s (3.07s)
4	26.18s (1.32s)	21.55s (0.78s)	4.62s (0.70s)	2.03s (0.42s)	2.59s (0.60s)
5	24.26s (1.94s)	19.12s (0.56s)	5.15s (1.74s)	2.89s (0.70s)	2.26s (1.41s)
6	25.37s (1.15s)	21.81s (0.58s)	3.56s (0.94s)	1.79s (0.58s)	1.77s (0.91s)
7	23.83s (0.95s)	20.04s (0.49s)	3.79s (0.75s)	2.13s (0.46s)	1.65s (0.59s)

4.3.3 Path Efficiency

To quantitatively compare the performance in movements of the end-effector when using the myoelectric and jog interfaces, the path efficiency is determined. Table 17 shows the path efficiencies of the myoelectric and jog interfaces. To statistically compare the path efficiencies of the myoelectric and jog interface, two-sample t-tests were performed. Except Subject 1, the path efficiencies of the other subjects show no statistically significant difference. The path efficiency is related to the length of the straight line between the start point and the position of the target object. The higher the path efficiency is, the shorter the end-effector travels to approach to the target object. From the experimental results of the path efficiency, Subjects 2 to 5 show that the traveling path lengths of the myoelectric and jog interfaces are similar. In the case of Subject 1, the jog interface outperformed the myoelectric interface in the path efficiency.

Table 17 Path efficiency

Subject No.	Path Eff. Myo	Path Eff. Jog	Welch's t-test	p value ($p < 0.05$)	Mean Diff. (Myo - Jog)
1	44.54% (1.64%)	46.19% (1.55%)	t(18) = -2.31	0.03	-1.65%
2	44.19% (2.13%)	44.71% (0.71%)	t(11) = -0.73	0.48	-0.52%
3	44.56% (1.73%)	45.14% (1.04%)	t(15) = -0.91	0.38	-0.58%
4	44.88% (2.16%)	45.96% (0.58%)	t(10) = -1.53	0.16	-1.08%
5	49.15% (3.21%)	48.87% (1.13%)	t(11) = 0.26	0.80	0.27%
6	48.50% (2.52%)	45.31% (0.32%)	t(9) = 3.97	0.00	3.18%
7	46.05% (1.19%)	47.36% (1.07%)	t(18) = -2.58	0.02	-1.31%

4.4 Discussion

With increasing dependency on computerized devices in human life, the need for communicating intuitively with them is also growing. As an interface method between a human and a computerized device, a myoelectric interface using dynamic motions to control a 6-DOF robot manipulator with a 1-DOF gripper is proposed here. The myoelectric interface has useful advantages such as 1) more dexterous gestures with a hand and fingers, 2) no limitations for outdoor activities, 3) no interference when using a hand, and 4) continuous information in term of muscle force. These advantages can be compared to vision-based gesture recognition, which is an active research topic for entertainment and surveillance. Recognition of dexterous finger and hand gestures and no limitation of locations for outdoor applications are potential advantages compared to vision-based approaches [106]. Based on the strong points of the myoelectric interface, the proposed myoelectric interface uses the sequence-based myoelectric classification to add the robustness with respect to the limb position changes. The proposed myoelectric interface focuses on controlling a robot manipulator in terms of translation, rotation, and opening/closing a gripper. The proposed myoelectric interface classifies dynamic motions to handle the manipulation mode, and uses orientation of a forearm and averaged muscle force to generate continuous commands.

To estimate the performance of the proposed myoelectric interface, the five time-measures, the path efficiency, and the real-time classification accuracy were calculated from the repeated go-and-pickup tasks. Also, in order to compare the results of the myoelectric interface with a well-known interface method in industry and laboratory related applications, the GUI button-based jog interface was developed using Matlab GUI software, and was used to perform the same go-and-pickup task repeatedly. The time-measures and the path efficiencies of the two interfaces were compared.

In terms of accurate real-time classification, DCG can efficiently classify dynamic hand motions in real-time with an average accuracy of 95.24%. In fact, there were cases that DCG missed a dynamic motion because of lack of proficiency of the human subjects in conducting the dynamic motion as described in the definition of a dynamic motion, i.e., not placing the before- and after-rest-regions and not generating an abrupt change at the beginning of a dynamic motion. The missing rate is not examined here because the missing rate becomes negligible by practicing dynamic motions.

According to the results of the path efficiency, Subjects 2, 3, 4, and 5 showed similar results, but the results of the other subjects, Subjects 1, 6, and 7, show that the myoelectric and jog interfaces performed differently in terms of the path efficiency. In the case of Subject 6, the myoelectric interface outperformed the jog interface by 3.18% in the path efficiency. However, in the cases of Subjects 1 and 7, the jog interface showed better performances of 1.65% and 1.31%, respectively. More than half of the subjects shows that the moving paths of the robot manipulator to complete the task by using the myoelectric and jog interfaces are similar. Although the results of the path efficiency from the other subjects (Subjects 1, 6, and 7) are different in the myoelectric and jog interfaces, Subject 6 shows that the myoelectric interface outperforms the jog interface, and Subjects 1 and 7 show that the mean differences of the path efficiency between the myoelectric and jog interfaces are close to the standard deviation of their path efficiency results. Based on this tendency, these two different interface methods seem to have no difference in controlling a moving path of the robot manipulator even though they are slightly subject-dependent.

From the result of the time-measures, the experiment time shows the jog interface outperforms the myoelectric interface. This result was expected because conducting a dynamic motion needs more time than just pushing a button to change the manipulation mode. From the experimental results, performing a dynamic motion requires about 2 seconds on average as shown in Table 14. Considering the length of the before- and after-rest-regions, the necessary time to change the manipulation mode once in the myoelectric interface is more than 3 seconds. In general, the manipulation mode is changed twice to complete the go-and-pickup task. As compared to the fact that pushing a button in the jog interface needs less than half a second, the myoelectric interface needs about 6 seconds more to finish the go-and-pickup task. The additionally-required time (about 6s) to change the manipulation modes in the myoelectric interface can explain the gap in the state-change times of the myoelectric and jog interface. Even though the jog interface outperforms the myoelectric interface in the time-measures, if the advantages of the myoelectric interface are considered, the more-necessary time can be thought as a trade-off.

Reviewing previous studies related to the myoelectric interface shows that many studies used EMG sensors and additional sensors that have different modalities to EMG so as to classify dynamic motions [54, 57, 59, 62]. In [57], accelerometer and EMG sensors were used to recognize 72 Chinese Sign Language words. In [59], static finger and wrist motions were recognized by EMG signals. The IMU data augmented the number of recognizable gestures by discriminating the arm angles. In [54], the authors used both IMU and EMG sensors to recognize 9 dynamic motions. The discriminative feature vectors of combining the gyroscope, accelerometer, and EMG data over time are generated for each dynamic motion. In [62], IMU and EMG sensors were fused to classify 12 dynamic gestures, which were characterized by continuous density Hidden Markov Models and Gaussian Mixture Models.

Based on the circumstance of using myoelectric signals for HCI and gesture recognition in previous studies, the proposed myoelectric interface catches and classifies dynamic hand motions in real-time via only EMG signals. Moreover, by differentiating the gesture types (static and dynamic motions), the generating source of discrete and continuous commands can be separated. This is particularly convenient because users' understanding of the interface system can be increased and possible confusions when generating the different types of commands can be prevented. Also, the robustness of the proposed myoelectric interface to the limb position changes is improved by using the sequence-based myoelectric classification (comparing the changes in temporal sequences of the feature vectors). In addition, an event-based control schema using dynamic motions as discrete commands can reduce the effort of using muscles for human operators compared to previous myoelectric interfaces using static gestures that are needed to be held in operation.

The myoelectric interface has a huge potential for numerous HCI applications. In this study, a 6-DOF robot manipulator with a 1-DOF gripper was chosen as a target application. However, the myoelectric interface can be applied to a wide range of fields. For examples, Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV) are important supporting devices in military environments for surveillance and hazardous tasks. To control them, the characteristics of the myoelectric interface such as using familiar and dexterous hand gestures, no limitation for outdoor activities, and no interference when using a hand are advantageous in contrast to the data glove-based and vision-based systems [106]. Moreover, the drone industry is quite popular in personal entertainment, i.e., selfie drones. With increasing usability of drones in human life, the ease of controlling a drone is also an interesting issue. In this circumstance, using everyday gestures to control a drone is quite convenient and practical. The myoelectric interface can be used to recognize regular gestures. In addition, for virtual reality (VR) and augmented reality (AR), interfacing techniques to connect humans and virtual environments are also significant. The proposed myoelectric interface can be used practically for VR and AR because the myoelectric interface does not limit human movements and working spaces. Besides, the proportional information coming from estimating muscle force magnifies the usability of the myoelectric interface compared to previous gesture-based interfacing systems using vision and acceleration measurement.

4.5 Conclusion

In this study, a myoelectric interface using dynamic motions is proposed. In Chapter III, we showed that the sequence-based myoelectric classification for dynamic motions is relatively robust to the limb position changes. With this advantage, we developed a myoelectric interface using the sequence-based myoelectric classification. The proposed myoelectric interface utilizes dynamic hand motions, muscle force, and 3-axis orientation (roll, pitch, and yaw) to generate discrete and continuous commands for manipulating a 6-DOF robot manipulator.

The performance of the myoelectric interface was estimated in carrying out a pre-defined task that requires a participant to control a robot manipulator to approach an object, to grasp it, and to pick it up. The performance of the system is evaluated in terms of real-time classification accuracy, time-measures, and path efficiency. To compare with the performance of the myoelectric interface, a well-known interface, the jog interface, used widely in industrial and laboratory settings is also applied.

Based on the experimental results, the myoelectric interface needs more time to complete the task than the jog interface because of the processing time of catching and classifying dynamic motions and the time to execute the dynamic motion itself. However, the path efficiencies of the two interfaces are similar. Considering the benefits that myoelectric interface has such as using familiar and dexterous gestures, no limitations for outdoor activities, no interference when using a hand, and additional information (muscle force), the myoelectric interface shows a significant potential for human computer interaction applications.

CHAPTER V

CONCLUSION AND FUTURE RESEARCH

5.1 Summary of the Work

Among many approaches in the Human Computer Interaction realm, this dissertation focuses on a myoelectric approach to recognize human gestures. There exist numerous advantages in the myoelectric approach compared to other gesture recognition methods such as using vision-based, accelerometer-based, and data-glove-based approaches. First of all, the myoelectric interface can recognize more dexterous finger and hand gestures than vision-based gesture recognition. Moreover, the myoelectric interface does not hinder hand movements and somatosensory systems (touch sensation) in contrast to the data-glove-based approach. The myoelectric interface, also, can be used in outdoor environments because myoelectric signals are not affected by lights and sound noises. Furthermore, the proportional information by varying the muscle force is valuable in terms of increasing the richness of usable resources for applications.

Considering the benefits of the myoelectric interface, this dissertation proposed a sequence-based myoelectric classification scheme that is more robust to the limb position changes (Chapter 3), as well as a dexterous myoelectric interface to control a 6-DOF robot manipulator with a 1-DOF gripper (Chapter 4). The sequence-based myoelectric classification showed that comparing the similarity of temporal sequences of dynamic motions is more robust to changes resulting from the non-stationarity of myoelectric signals than comparing constant feature values static motions have. This is based on the fact that each dynamic motion has its own temporal sequence, which is less affected by the limb position effect. Moreover, the proposed myoelectric interface generated discrete commands by catching and classifying dynamic motions in real-time and continuous commands by interpreting the orientation of a forearm and muscle force. The proposed myoelectric interface is a possible application of using the sequence-based myoelectric classification and can be used for numerous applications such as an HCI for translating human intentions to a device.

5.2 Contributions of the Research

This research mainly deals with using myoelectric signals for gesture recognition. The sequence-based myoelectric classification comes from the effort to reduce the effect of non-stationary characteristics of myoelectric signals caused by limb position changes. To accomplish that, the sequence-based myoelectric classification focuses on using dynamic motions and comparing the similarity of temporal sequences dynamic motions have. In addition, as an application of the proposed myoelectric classification, the myoelectric interface using the sequence-based myoelectric classification in real-time was introduced so as to control a 6-DOF robot manipulator with a 1-DOF gripper. The novel aspects of the sequence-based myoelectric classification and the proposed myoelectric interface are as follows:

- The performances of various conventional pattern-based myoelectric classification approaches are evaluated in terms of classification accuracy, training time, and classification (prediction) time with public EMG datasets of hand and finger motions.

- The robustness of the limb position changes is improved in the proposed sequence-based myoelectric classification (the changes in temporal sequences of feature vectors of dynamic motions are compared).

- The improvement of the sequence-based myoelectric classification is verified by comparing the performance of a conventional pattern-based myoelectric classification using Naïve Bayesian classification and Fisher's Linear Discriminant Analysis dimensionality reduction. The results of two approaches in the limb position changes are also statistically analyzed.

- Dynamic hand motions are captured and classified in real-time via only EMG signals. The satisfactory real-time classification accuracy of the system is determined as a performance value.

- A myoelectric interface system using EMG and IMU is developed as a Human Computer Interaction application. This system generates discrete and continuous commands for controlling a robot manipulator as well as includes advantages of myoelectric system such as 1) recognition of dexterous hand and finger gestures, 2) availability in outdoor activities, 3) no interference in tactile sensitivity of a hand, and 4) continuous information of muscle force.

- The performances of the proposed myoelectric interface and GUI button-based jog interface are compared by five time-measures and path efficiency in a quantitative manner.

5.3 Future Research Work

In this dissertation, mainly two studies have been introduced: the myoelectric classification for dynamic motions and the myoelectric interface as an HCI scheme to control a robot manipulator. Based on the results described in the dissertation, there are possible future works as follow:

- Improving the performance of the sequence-based myoelectric classification by applying advanced approaches, such as Hidden Markov Models and Deep Neural Networks,
- Reducing the effect of subject dependency and sensor-position dependency,
- Improving the process of generating continuous commands by using more than two IMUs to recognize moving trajectories of a forearm and an upper arm rather than using an IMU for roll, pitch, and yaw (orientation) of a forearm,
- Considering muscle fatigue when estimating muscle force from RMG,
- Increasing the number of recognizable dynamic motions,
- Removing the mandatory existence of the before- and after-rest-regions to catch a dynamic motion in real-time, and
- Applying the proposed myoelectric interface to other applications such as controlling drones and interfacing with VR.
- Speeding up the real-time process using a fast processor such as GPU and FPGA.

REFERENCES

[1] D. Farina, A. Holobar, R. Merletti, and R. M. Enoka, "Decoding the neural drive to muscles from the surface electromyogram," *Clinical Neurophysiology,* vol. 121, pp. 1616-1623, Oct 2010.

[2] P. Konrad. (2005). *The ABC of EMG: a practical introduction to kinesiological electromyography (Version 1.4 March 2006 ed.)*.

[3] M. A. Oskoei and H. S. Hu, "Myoelectric control systems-A survey," *Biomedical Signal Processing and Control,* vol. 2, pp. 275-294, Oct 2007.

[4] J. Ning, S. Dosen, K. Muller, and D. Farina, "Myoelectric control of artificial limbs; Is there a need to change focus?," *Signal Processing Magazine, IEEE,* vol. 29, pp. 152-150, 2012.

[5] D. Farina, J. Ning, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, *et al.*, "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on,* vol. 22, pp. 797-809, 2014.

[6] D. A. Winter, *Biomechanics and motor control of human movement*, 4th ed. ed.: John Wiley & Sons, 2009.

[7] M. Hakonen, H. Piitulainen, and A. Visala, "Current state of digital signal processing in myoelectric interfaces and related applications," *Biomedical Signal Processing and Control,* vol. 18, pp. 334-359, 4// 2015.

[8] L. Resnik, S. L. Klinger, and K. Etter, "The DEKA Arm: Its features, functionality, and evolution during the Veterans Affairs Study to optimize the

DEKA Arm," *Prosthetics and Orthotics International,* vol. 38, pp. 492-504, Dec 2014.

[9] L. A. Miller, R. D. Lipschutz, K. A. Stubblefield, B. A. Lock, H. Huang, T. W. Williams III, *et al.*, "Control of a six degree of freedom prosthetic arm after targeted muscle reinnervation surgery," *Archives of physical medicine and rehabilitation,* vol. 89, pp. 2057-2065, 2008.

[10] F. Tenore, R. S. Armiger, R. Jacob Vogelstein, D. S. Wenstrand, S. D. Harshbarger, and K. Englehart, "An embedded controller for a 7-degree of freedom prosthetic arm," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2008, pp. 185-188.

[11] M. Jiaxin, N. V. Thakor, and F. Matsuno, "Hand and wrist movement control of myoelectric prosthesis based on synergy," *Human-Machine Systems, IEEE Transactions on,* vol. 45, pp. 74-83, 2015.

[12] A. Al-Jumaily and R. A. Olivares, "Bio-driven system-based virtual reality for prosthetic and rehabilitation systems," *Signal Image and Video Processing,* vol. 6, pp. 71-84, Mar 2012.

[13] K. Kiguchi and Y. Hayashi, "An EMG-based control for an upper-limb power-assist exoskeleton robot," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* vol. 42, pp. 1064-1071, Aug 2012.

[14] M. Al-Mulla, F. Sepulveda, and M. Colley, *sEMG techniques to detect and predict localised muscle fatigue*: INTECH Open Access Publisher, 2012.

[15] B. Bigland-Ritchie, "EMG/force relations and fatigue of human voluntary contractions," *Exercise and Sport Sciences Reviews,* vol. 9, pp. 75-118, 1981.

[16] K. Englehart, B. Hudgins, and A. D. C. Chan, "Continuous multifunction myoelectric control using pattern recognition," *Technology & Disability,* vol. 15, pp. 95-103 9p, 2003.

[17] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification," *Measurement Science Review,* vol. 11, pp. 45-52, 2011.

[18] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences,* vol. 454, pp. 903-995, 1998.

[19] H. J. Liu and K. Y. Young, "Robot motion governing using upper limb EMG signal based on empirical mode decomposition," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, 2010.

[20] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering,* vol. 50, pp. 848-854, Jul 2003.

[21] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "EMG feature evaluation for improving myoelectric pattern

recognition robustness," *Expert Systems with Applications,* vol. 40, pp. 4832-4840, 9/15/ 2013.

[22] J. Zhaojie, O. Gaoxiang, M. Wilamowska-Korsak, and L. Honghai, "Surface EMG based hand manipulation identification via nonlinear feature extraction and classification," *IEEE Sensors Journal,* vol. 13, pp. 3302-3311, 2013.

[23] J. U. Chu, I. Moon, and M. S. Mun, "A real-time EMG pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand," *IEEE Transactions on Biomedical Engineering,* vol. 53, pp. 2232-2239, Nov 2006.

[24] R. N. Khushaba, A. Al-Ani, and A. Al-Jumaily, "Orthogonal fuzzy neighborhood discriminant analysis for multifunction myoelectric hand control," *Biomedical Engineering, IEEE Transactions on*, vol. 57, pp. 1410-1419, 2010.

[25] A. Phinyomark, H. Hu, P. Phukpattaranont, and C. Limsakul, "Application of Linear Discriminant Analysis in Dimensionality Reduction for Hand Motion Classification," *Measurement Science Review,* vol. 12, pp. 82-89, 2012.

[26] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, 2012, pp. 1534-1539.

[27] K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *Biomedical Engineering, IEEE Transactions on,* vol. 48, pp. 302-311, 2001.

[28] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins, "Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 18, pp. 49-57, Feb 2010.

[29] R. M. G. Tello, T. Bastos-Filho, R. M. Costa, A. Frizera-Neto, S. Arjunan, D. Kumar, *et al.*, "Towards sEMG Classification Based on Bayesian and k-NN to Control a Prosthetic Hand," *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP,* pp. 195-200, 2013 2013.

[30] F. V. G. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *Biomedical Engineering, IEEE Transactions on,* vol. 56, pp. 1427-1434, 2009.

[31] A. T. C. Au and R. F. Kirsch, "EMG-based prediction of shoulder and elbow kinematics in able-bodied and spinal cord injured individuals," *IEEE Transactions on Rehabilitation Engineering,* vol. 8, pp. 471-480, 2000.

[32] J. Shi, Y. Cai, J. Zhu, J. Zhong, and F. Wang, "SEMG-based hand motion recognition using cumulative residual entropy and extreme learning machine," *Medical & Biological Engineering & Computing,* vol. 51, pp. 417-427, Apr 2013.

[33] F. H. Y. Chan, Y. Yong-Sheng, F. K. Lam, Z. Yuan-Ting, and P. A. Parker, "Fuzzy EMG classification for prosthesis control," *Rehabilitation Engineering, IEEE Transactions on,* vol. 8, pp. 305-311, 2000.

[34] A. B. Ajiboye and R. F. Weir, "A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13, pp. 280-291, 2005.

[35] M. Khezri and M. Jahed, "Real-time intelligent pattern recognition algorithm for surface EMG signals," *Biomedical Engineering Online*, vol. 6, Dec 3 2007.

[36] K. Kiguchi, T. Tanaka, K. Watanabe, and T. Fukuda, "Design and control of an exoskeleton system for human upper-limb motion assist," in *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, 2003, pp. 926-931 vol.2.

[37] M. Vuskovic and D. Sijiang, "Classification of prehensile EMG patterns with simplified fuzzy ARTMAP networks," in *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, 2002, pp. 2539-2544.

[38] H. Jeong-Su, Z. Z. Bien, K. Dae-Jin, L. Hyong-Euk, and K. Jong-Sung, "Human-machine interface for wheelchair control with EMG and its evaluation," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, 2003, pp. 1602-1605 Vol.2.

[39] A. Subasi, M. Yilmaz, and H. R. Ozcalik, "Classification of EMG signals using wavelet neural network," *Journal of Neuroscience Methods*, vol. 156, pp. 360-367, Sep 2006.

[40] A. D. Chan and K. B. Englehart, "Continuous myoelectric control for powered prostheses using hidden Markov models," *Biomedical Engineering, IEEE Transactions on,* vol. 52, pp. 121-124, 2005.

[41] Y. H. 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, pp. 1801-1811, Nov 2005.

[42] O. Fukuda, T. Tsuji, M. Kaneko, and A. Otsuka, "A human-assisting manipulator teleoperated by EMG signals and arm motions," *Robotics and Automation, IEEE Transactions on,* vol. 19, pp. 210-222, 2003.

[43] A. D. C. Chan and K. B. Englehart, "Continuous classification of myoelectric signals for powered prostheses using gaussian mixture models," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, 2003, pp. 2841-2844 Vol.3.

[44] P. Shenoy, K. J. Miller, B. Crawford, and R. P. N. Rao, "Online Electromyographic Control of a Robotic Prosthesis," *Biomedical Engineering, IEEE Transactions on,* vol. 55, pp. 1128-1135, 2008.

[45] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *Biomedical Engineering, IEEE Transactions on,* vol. 55, pp. 1956-1965, 2008.

[46] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Systems with Applications*, vol. 39, pp. 10731-10738, Sep 2012.

[47] A. Al-Timemy, G. Bugmann, J. Escudero, and N. Outram, "Classification of finger movements for the dexterous hand prosthesis control with surface electromyography," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, pp. 608-18, May 2013.

[48] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, pp. 311-324, 2007.

[49] X. Shen, G. Hua, L. Williams, and Y. Wu, "Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields," *Image and Vision Computing*, vol. 30, pp. 227-235, 3// 2012.

[50] A. Akl, F. Chen, and S. Valaee, "A novel accelerometer-based gesture recognition system," *Signal Processing, IEEE Transactions on*, vol. 59, pp. 6197-6205, 2011.

[51] N.-D. Long-Van, D. Roggen, A. Calatroni, and G. Troster, "Improving online gesture recognition with template matching methods in accelerometer data," in *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*, 2012, pp. 831-836.

[52] M. AbdelMaseeh, T.-W. Chen, and D. Stashuk, "Multifunction myoelectric control using multi-dimensional dynamic time warping," in *Engineering in*

Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, Chicago, IL, USA, 2014, pp. 4366-4369.

[53] W. H. Wang, X. A. Chen, K. Q. Wang, X. Zhang, and J. H. Yang, "Dynamic gesture recognition based on multiple sensors fusion technology," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, ed, 2009, pp. 7014-7017.

[54] M. T. Wolf, C. Assad, A. Stoica, K. You, H. Jethani, M. T. Vernacchia, *et al.*, "Decoding static and dynamic arm and hand gestures from the JPL BioSleeve," in *Aerospace Conference, 2013 IEEE,* 2013, pp. 1-9.

[55] Y. Lu, Z. Ju, Y. Liu, Y. Shen, and H. Liu, "Time series modeling of surface EMG based hand manipulation identification via expectation maximization algorithm," *Neurocomputing,* vol. 168, pp. 661-668, 11/30/ 2015.

[56] P. K. Artemiadis and K. J. Kyriakopoulos, "A switching regime model for the EMG-based control of a robot arm," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 41, pp. 53-63, 2011.

[57] Z. Xu, C. Xiang, L. Yun, V. Lantz, W. Kongqiao, and Y. Jihai, "A framework for hand gesture recognition based on accelerometer and EMG sensors," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 41, pp. 1064-1076, 2011.

[58] H. J. Liu and K. Y. Young, "Upper-limb EMG-based robot motion governing using empirical mode decomposition and adaptive neural fuzzy inference

system," *Journal of Intelligent & Robotic Systems,* vol. 68, pp. 275-291, Dec 2012.

[59] M. T. Wolf, C. Assad, M. T. Vernacchia, J. Fromm, and H. L. Jethani, "Gesture-based robot control with variable autonomy from the JPL BioSleeve," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 1160-1165.

[60] K. Min Kyu, R. Kwanghyun, O. Yonghwan, O. Sang-Rok, and K. Keehoon, "Implementation of real-time motion and force capturing system for tele-manipulation based on sEMG signals and IMU motion data," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 5658-5664.

[61] F. Haque, M. Nancel, and D. Vogel, "Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3653-3656.

[62] M. Georgi, C. Amma, and T. Schultz, "Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing," in *International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS-2015)*, 2015, pp. 99-108.

[63] B. Peerdeman, D. Boere, H. Witteveen, R. H. in't Veld, H. Hermens, S. Stramigioli*, et al.*, "Myoelectric forearm prostheses: State of the art from a user-

centered perspective," *Journal of Rehabilitation Research and Development*, vol. 48, pp. 719-737, 2011.

[64] A. Young, L. Hargrove, and T. Kuiken, "The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift," *IEEE Transactions on Biomedical Engineering,* vol. 58, pp. 2537 - 2544, 2011.

[65] L. Hargrove, K. Englehart, and B. Hudgins, "A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control," *Biomedical Signal Processing and Control,* vol. 3, pp. 175-180, Apr 2008.

[66] J. He, D. Zhang, X. Sheng, and X. Zhu, "Effects of long-term myoelectric signals on pattern recognition," in *International Conference on Intelligent Robotics and Applications*, 2013, pp. 396-404.

[67] P. Kaufmann, K. Englehart, and M. Platzner, "Fluctuating EMG signals: Investigating long-term effects of pattern matching algorithms," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology,* 2010, pp. 6357-6360.

[68] S. Amsuss, L. P. Paredes, N. Rudigkeit, B. Graimann, M. J. Herrmann, and D. Farina, "Long term stability of surface EMG pattern classification for prosthetic control," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, 2013, pp. 3622-3625.

[69] Y. Geng, P. Zhou, and G. Li, "Toward attenuating the impact of arm positions on electromyography pattern-recognition based motion classification in transradial amputees," *Journal of Neuroengineering and Rehabilitation*, vol. 9, pp. 1-11, 2012.

[70] A. Fougner, E. Scheme, A. D. Chan, K. Englehart, and Ø. Stavdahl, "Resolving the limb position effect in myoelectric pattern recognition," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 19, pp. 644-651, 2011.

[71] R. N. Khushaba, M. Takruri, J. V. Miro, and S. Kodagoda, "Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features," *Neural Networks*, vol. 55, pp. 42-58, 2014.

[72] B. Wan, L. Xu, Y. Ren, L. Wang, S. Qiu, X. Liu, *et al.*, "Study on fatigue feature from forearm SEMG signal based on wavelet analysis," in *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, 2010, pp. 1229-1232.

[73] E. Clancy, E. L. Morin, and R. Merletti, "Sampling, noise-reduction and amplitude estimation issues in surface electromyography," *Journal of Electromyography and Kinesiology*, vol. 12, pp. 1-16, 2002.

[74] E. Scheme, A. Fougner, Ø. Stavdahl, A. Chan, and K. Englehart, "Examining the adverse effects of limb position on pattern recognition based myoelectric control," in *Engineering in Medicine and Biology Society (EMBC), 2010 32th Annual International Conference of the IEEE*, Buenos Aires, Argentina, 2010, pp. 6337-6340.

[75] M. R. Masters, R. J. Smith, A. B. Soares, and N. V. Thakor, "Towards better understanding and reducing the effect of limb position on myoelectric upper-limb prostheses," in *Engineering in Medicine and Biology Society (EMBC), 2014 34th Annual International Conference of the IEEE*, 2014, pp. 2577-2580.

[76] K.-H. Park, H.-I. Suk, and S.-W. Lee, "Position-independent decoding of movement intention for proportional myoelectric interfaces," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. PP, pp. 1-1, 2015.

[77] N. M. López, F. di Sciascio, C. M. Soria, and M. E. Valentinuzzi, "Robust EMG sensing system based on data fusion for myoelectric control of a robotic arm," *Biomedical engineering online*, vol. 8, 2009.

[78] D. Tkach, H. Huang, and T. A. Kuiken, "Research study of stability of time-domain features for electromyographic pattern recognition," *Journal of Neuroengineering and R*

ehabilitation, vol. 7, p. 21, 2010.

[79] D. Tkach, H. Huang, and T. A. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Journal of Neuroengineering and Rehabilitation*, vol. 7, May 2010.

[80] A. Young, L. Hargrove, and T. Kuiken, "Improving myoelectric pattern recognition robustness to electrode shift by changing interelectrode distance and electrode configuration," *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 645 - 652, 2012.

[81] A. H. Al-Timemy, J. E. Guido Bugmann, and N. Nicholas Outram, "A preliminary investigation of the effect of force variation for myoelectric control of hand prosthesis," presented at the Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, Osaka, Japan, 3 - 7 July, 2013, 2013.

[82] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the United States: 2005 to 2050," *Archives of Physical Medicine and Rehabilitation,* vol. 89, pp. 422-429, Mar 2008.

[83] M. Ortiz-Catalan, R. Brånemark, and B. Håkansson, "BioPatRec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms," *Source code for biology and medicine,* vol. 8, p. 11, 2013.

[84] S. Shin, R. Langari, and R. Tafreshi, "A Performance Comparison of EMG Classification Methods for Hand and Finger Motion," in *ASME 2014 Dynamic Systems and Control Conference*, 2014, pp. V002T16A008-V002T16A008.

[85] X. Hu, Z. Wang, and X. Ren, "Classification of surface EMG signal using relative wavelet packet energy," *Computer Methods and Programs in Biomedicine,* vol. 79, pp. 189-195, 2005.

[86] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *Journal of biomechanics,* vol. 43, pp. 1573-1579, 2010.

[87] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* vol. 2, pp. 559-572, 1901.

[88] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.

[89] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.

[90] A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization and beyond*: MIT press, 2002.

[91] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing,* vol. 70, pp. 489-501, 2006.

[92] J. F. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, 1990, pp. 381-384.

[93] J. B. Kruskal and M. Liberman, "The symmetric time-warping problem: from continuous to discrete," in *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, D. Sankoff and J. B. Kruskal, Eds., ed: Addison-Wesley, 1983, pp. 125-162.

[94] X. Li, P. Zhou, and A. S. Aruin, "Teager–Kaiser energy operation of surface EMG improves muscle activity onset detection," *Annals of biomedical engineering,* vol. 35, pp. 1532-1538, 2007.

[95] S. Solnik, P. Rider, K. Steinweg, P. DeVita, and T. Hortobágyi, "Teager–Kaiser energy operator signal conditioning improves EMG onset detection," *European journal of applied physiology*, vol. 110, pp. 489-498, 2010.

[96] A.-C. Tsai, T.-H. Hsieh, J.-J. Luh, and T.-T. Lin, "A comparison of upper-limb motion pattern recognition using EMG signals during dynamic and isometric muscle contractions," *Biomedical Signal Processing and Control,* vol. 11, pp. 17-26, May 2014.

[97] M. Müller, "Dynamic Time Warping," in *Information Retrieval for Music and Motion*, ed: Springer Berlin Heidelberg, 2007, pp. 69-84.

[98] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning* vol. 1: Elsevier, 2006.

[99] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM,* vol. 54, pp. 60-71, 2011.

[100] A. Akl and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing," in *Acoustics, Speech, and Signal Processing (ICASSP), 2010 IEEE International Conference on*, Dallas, TX, USA, 2010, pp. 2270-2273.

[101] D. Yang, J. Zhao, L. Jiang, and H. Liu, "Dynamic hand motion recognition based on transient and steady-state EMG signals," *International Journal of Humanoid Robotics,* vol. 9, p. 1250007, Mar 2012.

[102] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing,* vol. 99, pp. 215-249, 6// 2014.

[103] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes: theory and application*: Prentice-Hall, Inc., 1993.

[104] M. R. Williams and R. F. Kirsch, "Evaluation of head orientation and neck muscle EMG signals as command inputs to a human-computer interface for individuals with high tetraplegia," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 16, pp. 485-496, 2008.

[105] S. M. Radhakrishnan, S. N. Baker, and A. Jackson, "Learning a novel myoelectric-controlled interface task," *Journal of Neurophysiology,* vol. 100, pp. 2397-2408, 2008.

[106] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review,* vol. 43, pp. 1-54, 2015.