Electromyographic Signal Processing With Application To Spinal Cord Injury

Thesis by

Zhao Liu

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

Caltech

California Institute of Technology
Pasadena, California

2016
(Defended March 21, 2016)

To Teresa, for her endless support and love.

Acknowledgments

It gives me great pleasure to acknowledge the many people whose work, advice and support have helped me during my Ph.D. over the last six years.

Foremost, I would like to thank my advisor, Prof. Joel Burdick. He gave me the opportunity of working on Electromyogram (EMG) processing, which made me realize how much l Iike programming and algorithm development. I still clearly remember the “brick and wall” example he used to illustrate how to approach a problem when I first started working in his group. I find that really useful in many different areas of my life, not just research. During my time working with him, Prof. Burdick has always been very supportive, kind and generous. Without him, I couldn’t have made it here.

I would also like to thank my research group for numerous discussions on my research. I would like to thank Jeffrey Edlund in particular because he showed me to the world of EMG processing when I first started. I’ve learnt a lot from him, especially about programming, and he contributed many of the codes used in EMG processing.

Thank you to my collaborators at University of Louisville at Kentucky for their support and discussions on my research and I’m especially grateful for their warm reception during my visit there.

I’m very thankful for all the suggestions from my committee members, Prof. Hyuck Choo, Prof. David Rutledge, Prof. Pietro Perona, Prof. Changhuei Yang and Prof. Reggie Edgerton.

I would also like to thank my parents. Without them, I couldn’t come to pursue graduate study at Caltech. Even though they are far away in China, their support and love have been invaluable to me during my life at Caltech.

Last but not least, I’d like to thank my fiancee, Teresa Liu, for her great and constant support for my Ph.D. life at Caltech. She helped me solve numerous problems I encountered during my road to graduation. Without her love and encouragement, I definitely wouldn’t be finishing this thesis.

Abstract

An Electromyogram or Electromyographic (EMG) signal is the recording of the electrical activity produced by muscles. It measures the electric currents generated in muscles during their contraction. The EMG signal provides insight into the neural activation and dynamics of the muscles, and is therefore important for many different applications, such as in clinical investigations that attempt to diagnose neuromuscular deficiencies. In particular, the work in this thesis is motivated by rehabilitation for patients with spinal cord injury. The EMG signal is very important for researchers and practitioners to monitor and evaluate the effect of the rehabilitation training and the condition of muscles, as the EMG signal provides information that helps infer the neural activity in the spinal cord. Before the work in this thesis, EMG analysis required significant amounts of manual labeling of interesting signal features. The motivation of this thesis is to fully automate the EMG analysis tasks and yield accurate, consistent results.

The EMG signal contains multiple muscle responses. The difficulty in processing the EMG signal arises from the fact that the transient muscle response is a transient signal with unknown arrival time, unknown duration, and unknown shape. In addition, the EMG signal recorded from patients with spinal cord injury during rehabilitation is very different from the EMG signal of normal healthy people undergoing the same motions. For example, some of the muscle responses are very weak and thus hard to detect. Because of this, general EMG processing tools and methods are either not applicable or insufficient.

The primary contribution of this thesis is the development of a wavelet-based, double-threshold algorithm for the detection of transient peaks in the EMG signal. The application of wavelet transform in the detection of transient signals has been studied extensively and employed successfully. However, most of the theories assume certain knowledge about the shapes of the transient signals, which makes it hard to be generalized to the transient signals with arbitrary shapes. The proposed detection scheme focuses on the more fundamental feature of most transient signals (in particular the EMG signal): peaks, instead of the shapes. The continuous wavelet transform with Mexican Hat wavelet is employed. This thesis theoretically derived a framework for selecting a set of scales based on the frequency domain information. Ridges are identified in the time-scale space to combine the wavelet coefficients from different scales. By imposing two thresholds, one on the wavelet coefficient and one on the ridge length, the proposed detection scheme can achieve both high recall and high precision. A systematic approach for selecting the optimal parameters via simulation is proposed and demonstrated. Comparing with other state-of-the-art detection methods, the proposed method in this thesis yields a better detection performance, especially in the low Signal-to-Noise-Ratio (SNR) environment.

Based on the transient peak detection result, the EMG signal is further segmented and classified into various groups of monosynaptic Motor Evoked Potentials (MEPs) and polysynaptic MEPs using techniques stemming from Principal Component Analysis (PCA), hierarchical clustering, and Gaussian mixture model (GMM). A theoretical framework is proposed to segment the EMG signal based on the detected peaks. The scale information of the detected peak is used to derive a measure for its effective support. Several different techniques have been adapted together to solve the clustering problem. An initial hierarchical clustering is first performed to obtain most of the monosynaptic MEPs. PCA is used to reduce the number of features and the effect of the noise. The reduced feature set is then fed to a GMM to further divide the MEPs into different groups of similar shapes. The method of breaking down a segment of multiple consecutive MEPs into individual MEPs is derived.

A software with graphic user interface has been implemented in Matlab. The software implements the proposed peak detection algorithm, and enables the physiologists to visualize the detection results and modify them if necessary. The solutions proposed in this thesis are not only helpful to the rehabilitation after spinal cord injury, but applicable to other general processing tasks on transient signals, especially on biological signals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An Electromyogram or Electromyographic (EMG) signal is the recording of the electrical activity produced by muscles. It measures the electric currents generated in muscles during their contraction. The EMG signal provides insight into the neural activation and dynamics of the muscles, and is therefore important for clinical investigations that attempt to diagnose neuromuscular deficiencies such as those caused by stroke and Parkinson's disease [50]. Other potential important applications exist in areas such as aging, exercise physiology, space medicine, and ergonomics, where it is of interest to understand whether the control of muscles is altered as a consequence of aging, exercise, exposure to microgravity, fatigue, and excessive and prolonged force production [38]. The EMG signal can also be used to identify hand gestures and finds its application in controlling prosthetic devices for rehabilitation or wearable devices for gaming [57]. In particular, the work in this thesis is motivated by rehabilitation for patients with spinal cord injury. The EMG signal is very important for researchers and practitioners to monitor and evaluate the effect of the rehabilitation training and the condition of muscles [24], as the EMG signal provides information that helps infer the neural activity in the spinal cord. Before the work in this thesis, EMG analysis in [24] requires significant amount of manual labeling of interesting signal features. The motivation of this thesis is to fully automate the EMG analysis tasks and yield accurate, consistent results.

The processing of the EMG signal is a hard problem in general. Similar to action potentials and electrocardiogram (ECG), the useful information (the actual muscle response other than noise) in an EMG recording is classified as a transient signal, as it is contained within a short time ($\sim 10-50ms$). Every EMG recording contains multiple muscle responses. The difficulty in processing EMG arises from the fact that the transient muscle response is a transient signal with unknown arrival time, unknown duration, and unknown shape. In addition, the EMG signal recorded from patients with spinal cord injury during rehabilitation is very different from the EMG signal of normal healthy people undergoing the same motions. For example, some of the muscle responses are very weak and thus hard to detect. Because of this, traditional EMG processing tools and methods are either not applicable or insufficient. In this thesis, two EMG processing tasks were addressed and effective methods were derived. The solutions proposed in this thesis are not only helpful to the rehabilitation after spinal cord injury, but applicable to other general processing tasks on transient signals (biological signals in particular).

The first task is to detect useful information (the transient muscle response) from EMG recording. The detection of transient signals with unknown arrival time, unknown duration, and unknown shape is still an active research problem. There is not a universal optimal detector. Usually, different detection schemes are employed to tackle transient signals from different applications. In this thesis, a peak-based transient detection algorithm was proposed. The methodology consists of a combination of several techniques stemming from multi-resolution wavelet decomposition, statistics, and detection theory. The detection method is totally unsupervised, and it's especially good at detecting trasient signals with low signal-to-noise ratio (SNR). Simulation results show that this method works much better than existing methods found in the literature. In addition, the simulation shows that this method is applicable to general transient signals.

The other task addressed in this thesis is to automatically segment and classify the EMG signal into different areas of interest. A theoretical framework is proposed to segment the EMG signal based on the detected peaks. The scale information of the detected peaks is used to derive a measure for its effective support. Several different techniques have been adapted together to solve the clustering problem. An initial hierarchical clustering is first performed to obtain most of the monosynaptic Motor Evoked Potentials (MEPs). Principal component analysis (PCA) is used to reduce the number of features and effect of the noise. The reduced feature set is then fed to a Gaussian mixture model (GMM) to further divide the MEPs into different groups of similar shapes. The method of breaking down a segment of multiple consecutive MEPs into individual MEPs is also derived.

1.1 Spinal Cord Injury

The work of this dissertation is motivated by the field of *spinal cord injury* (SCI). The goal of the spinal cord injury research is to help patients with severe spinal cord injury to recover their ability to stand, walk, and have voluntary movements [24]. According to the report published by the National Spinal Cord Injury Statistical Center in 2012, the number of people in the United States who have SCI has been estimated to be approximately 270,000, with approximately 12,000 new cases each year. In particularly, about 30% of the cases are *complete* SCI, which means all the sensation and voluntary control of some parts of the body is lost [58]. Figure 1.1-1 shows that SCI is mainly caused by motor vehicle crashes, followed by falls and violence. Because of this, SCI primarily affects young

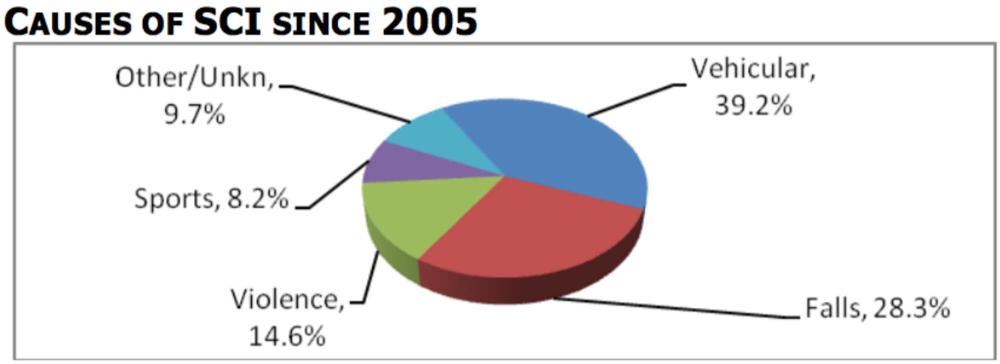


Figure 1.1-1: Causes of SCI since 2005 [58]

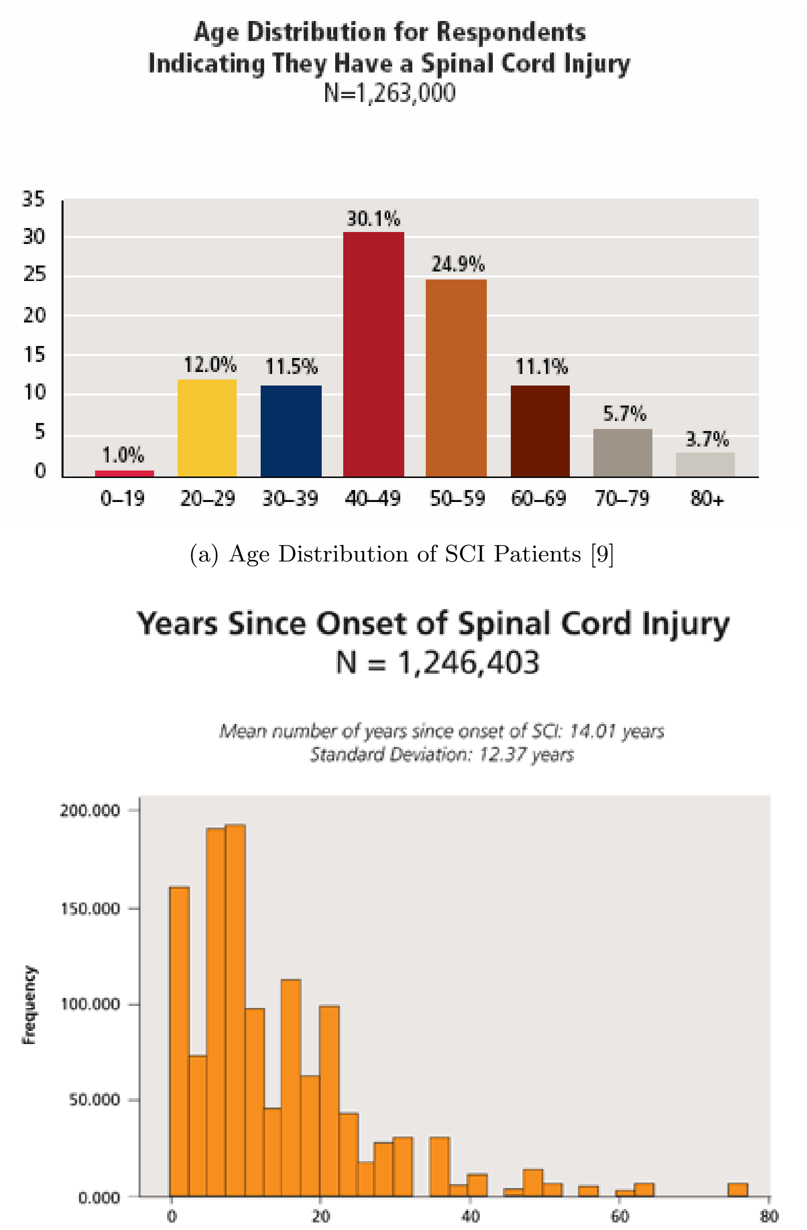
adults. According to the data from the Christopher & Dana Reeve Foundation, the average age of those who reported being paralyzed due to a spinal cord injury was 48, and the average length of time since the spinal cord injury occurred was 14 years, as shown in Figure 1.1-2 [9]. This indicates that those sustaining SCI tend to be young individuals suffering traumatic injury at the prime of their personal lives and economic earning potentials.

Right now, SCI cannot be cured, and rehabilitation can be very difficult. Besides the pain and extreme high cost of lifetime medical care, people with severe SCI cannot currently stand or walk, which can impact their social and work life. The work described in this thesis supports a collaborative effort between Caltech, University of Louisville, and UCLA to provide new therapies for SCI [24].

1.2 Recovery of Spinal Cord Injury

The spinal cord is the main pathway for information connecting the brain and the rest of the body. The human spinal cord is divided into 31 different segments: different segments connect to different parts of the body, as shown in Figure 1.2-1. When the spinal cord is injured, this pathway is blocked, and patients can lose sensation and voluntary control of some parts of the body depending on which level of the spinal cord is injured. In addition to the role of transmission of neural signals, the spinal cord also contains neural circuits that can independently control numerous reflexes and the central pattern generator. In the case of SCI, the spinal circuitry below the lesion remains intact, but loses supraspinal modulation from the brain.

Although there are no fully restorative treatments for SCI, various rehabilitative, cellular, and molecular therapies have been tested in animal models. In the cellular therapies, cells are transplanted, for example, to replace dead cells. In the molecular therapies, different pharmacological agents are used to restore biochemical imbalance after SCI. Rehabilitative training, such as loco-



(a) Age Distribution of SCI Patients [9]

(b) Years since onset of SCI [9]

Figure 1.1-2: Some statistics of SCI patients: age distribution as in (a); years since onset as in (b)

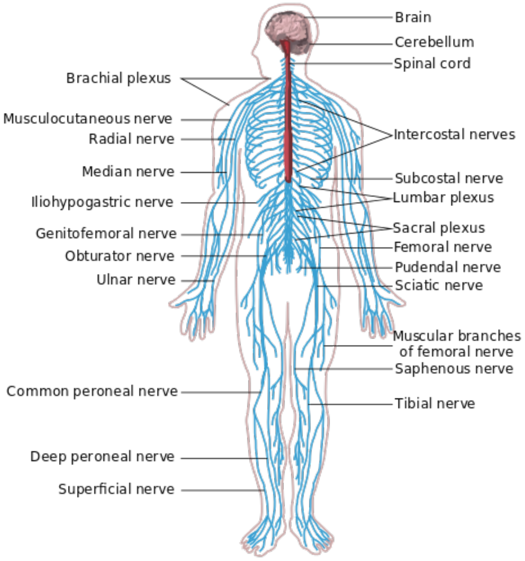


Figure 1.2-1: Human Spinal Cord [63]

motor training, improves the locomotor function. For a complete review, please see [59] and the references within. Our strategy for recovering from spinal cord injury is derived mainly from two important facts about the spinal circuitry.

First, basic posture and locomotion control depends significantly on the spinal circuitry, not the brain. In fact, one can stand without much conscious effort. The spinal circuitry needs to achieve some crucial level of excitability in order to work properly. For healthy humans, the brain sends neural signals (train of action potentials) down to the spinal cord to raise the excitation level of the spinal circuitry. For people with complete SCI, this modulation is lost, and hence the spinal circuitry cannot function properly to generate locomotion [12]. Our strategy is to use electrical stimulation to reactivate previously silent spinal circuitry.

The second fact is the plasticity of the spinal circuitry. The spinal circuitry can be trained to adapt to the loss of modulation from the brain after SCI. Therefore, task-specific training is performed to let the spinal circuitry adapt to the new modulation from the electrical stimulation [24].

Epidural Electrostimulation (EES) involves electrically stimulating the spinal cord via an electrode or multi-electrode array placed in the epidural space of the vertebral canal. To date, human studies of EES in [24] have been based on an epidural spinal cord stimulation unit from Medtronic (RestoreADVANCED, Medtronic, Minneapolis, MN, USA), a FDA-approved commercial product for back pain management. The electrode array was implanted over spinal cord segments L1–S1, the lumbosacral enlargement, which is responsible for lower-body movement.

The electrode array dimension is about 49mm by 10mm, and it contains 16 electrodes. Different

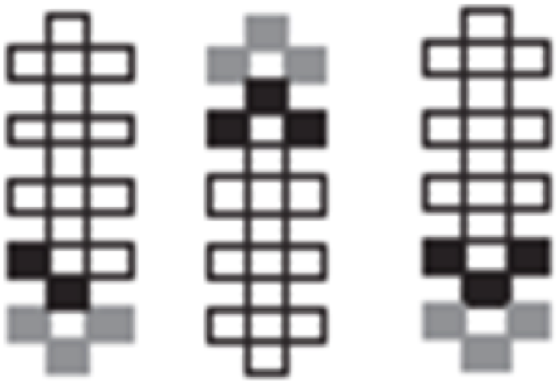


Figure 1.2-2: Examples of electrode configurations: anodes are black, and cathodes are gray, while the rest are floating electrodes (from Harkema [24]).

patterns of the stimulating electric field can be chosen by assigning anodes and cathodes to different electrodes (See Figure 1.2-2). The stimulation signal is a pulse train whose parameters, such as frequency, pulse width, and amplitude, can be adjusted.

EES must be coupled with physical training for maximum effect. The training currently employed includes standing, stepping, and voluntary movement control while lying supine. Various sensors are used during the training to analyze the performance of a subject under EES, and guide future training. In particular, surface electrodes are placed on various major muscle groups of the lower body, and the electric signal from the muscle, formally known as the Electromyogram or Electromyographic (EMG) signal, is recorded during the training. The EMG signal is one of the most important clinical data obtained in this project and many other neuromuscular research efforts. My research is to develop new tools for analyzing and processing the EMG signal.

1.3 Electromyographic (EMG) Signal Processing: Objective Statement

The Electromyographic (EMG) Signal is the electric potential generated by the muscle cells. In general, the EMG signal is very useful in many areas, such as detection of the medical abnormalities, and analysis of the biomechanics of movement. For spinal cord research, the EMG signal is one of the most important quantities to evaluate motor performance and training. Ideally, one would like to know how neurons in the spinal cord react to the epidural electrical stimulation. In practice, it's almost impossible to directly measure the neural activities in the spinal cord. On the other hand, when motor neurons in the spinal cord are activated, they send action potentials down to the specific muscles and activate the muscle fibers. EMG is a measurement of the electric activity happening around the muscle fibers. Therefore, the EMG signal can be used to infer the activity of the motor neurons in the spinal cord. In Section 2.1.1, a detailed review of the physiology of the EMG signal is presented. In summary, if one thinks of the spinal cord as a black box, by checking its output (the EMG signal), one can often infer the neural activity inside the spinal cord.

A major effort in the EMG processing with the application to the diagnosis of neuromuscular disorder is to decompose the EMG signal into its continuant Motor Unit Action Potential Trains (MUAPTs) [53, 38]. A fully description of MUAPTs is given in Section 2.1.1. Basically, EMG decomposition involves mainly two steps: Firstly, detect/segment the raw EMG signal to obtain individual Motor Unit Action Potentials (MUAPs), and then group MUAPs from the same motor unit to obtain MUAPTs. Another research topic on EMG processing is to detect the onset of muscle activity [50, 57]. In this problem, waveforms of individual MUAPs are no long desired; instead, the EMG signal is viewed on a higher level, and active periods of the EMG signal are determined.

In the following, a brief description of the two specific EMG processing tasks to be addressed in this thesis is given first. For detailed descriptions, please refer to Chapter 3 and Chapter 4, respectively. After that, a brief discussion on the challenges and motivations is presented. For full description on the challenges, please read Section 2.2. For a complete literature review, please read the sections within Chapter 3 and Chapter 4.

1.3.1 Peak-based Detection

There are many different kinds of processing we can do on the EMG signal. One very important aspect of the EMG signal is to measure the level of activation. More specifically, we'd like to detect the *Motor Evoked Potential* (MEP). MEPs are the action potentials along the muscle fibers generated by active motor neurons as a result of EES. MEPs are transient signals, and are corrupted by noise in the recording. Our goal is to detect all the peaks of the MEPs in the EMG signal (See Figure 1.3-1). The MEP peaks can potentially tell us what motor neurons are active, when they are active, and how strong the activation is. Throughout the thesis, EMG signal detection refer to the detection of the MEP peaks.

1.3.2 Segmentation and Classification

Another Important task is to segment and classify the EMG signal into various intervals of interest. Segmentation refers to the task of dividing the given EMG signal into various sections, and classification is to label the segmented sections according to their nature. In the example of the EMG signal obtained from EES, segmentation is supposed to find the sections of data that only contain

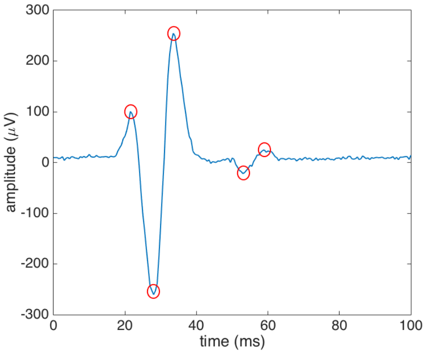


Figure 1.3-1: An example showing the detection of the peaks of MEPs in the EMG signal: red circles show the peaks of MEPs. (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

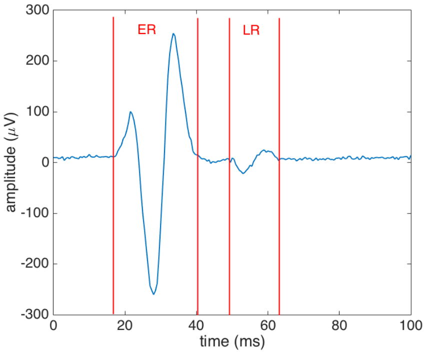


Figure 1.3-2: An example showing the segmentation of the EMG signal: ER (short for Early Response) is the label for the early MEP; LR (short for Late Response) is the label for the late MEP. (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

MEP waveforms. In addition, the MEPs can be further classified into early MEPs and late MEPs (See Figure 1.3-2). An early MEP is the direct response after an electrical stimulus, and so it's normally strong. A late MEP is the indirect response after an electrical stimulus. It comes after the early MEP and is normally weak. Both early MEPs and late MEPs are important, and they correspond to different underlying biophysics. Therefore, it's crucial to identify and differentiate them. With this information, one can better infer the neural activities in the spinal cord. Please refer to Section 2.1.2 for a full description of early MEPs and late MEPs.

Before my work, EMG processing was entirely carried out manually by many technicians at University of Louisville. It's laborious to manually label all the features (including peaks, early and late MEPs), since the rehabilitation training generates enormous amount of data from every experiment. Typically, a training session lasts a couple of hours. There are about 20 EMG channels, each of which measures a specific muscle group in the lower body. The sampling rate is 2000Hz. As a result, the EMG data file from one experiment easily exceeds 100MB. In addition, manual inspection and processing of the EMG signal are very subjective and the quality of the results depends on the experience and the knowledge of the individuals. Therefore, an automatic tool to process the EMG signal will save lots of time for the practitioners, and give reliable, accurate results.

Another motivation behind our work is to facilitate the machine learning algorithm developed to optimize the stimulation parameters [11]. The EMG signal is a very important input signal to the algorithm. The machine learning algorithm is supposed to automatically adjust the stimulation parameters from all observations. Thus, an unsupervised, accurate EMG processing algorithm is needed to enable on-line learning and adjustment of the array stimuli.

There are mainly two challenges behind the EMG processing work in this thesis.

First of all, the EMG signal is in nature hard to process automatically. The muscle response (the signal carrying useful information other than noise) is a transient signal with unknown arrival time, unknown duration, and unknown shape. The characteristics of the muscle responses vary from time to time, from subject to subject. As a result, most of the EMG processing in clinical research is still carried out with lots of human supervision, and the methods employed for detection and segmentation are still rudimentary [34, 37, 15, 8, 7, 27]. See Section 3.1 for a complete review of the literature on EMG detection.

Second of all, the EMG processing tasks and the characteristics of the EMG signal in the research on spinal cord injury (in particular, for rehabilitation with epidural electro-stimulation as in [24]) are quite different from those in typical research on neuromuscular disorder. As a result, no existing tools can be used to solve the problems encountered in this project, and new tools and methodologies need to be invented.

1.4 Thesis Outline and Contributions

The primary contribution of this thesis is the development of a set of automatic, unsupervised tools for the analysis of Electromyogram (the EMG signal) for the purpose of studying electrical stimulation based rehabilitation on patients with spinal cord injuries. A wavelet-based, double-threshold algorithm was developed for the detection of the transient peaks in the EMG signal (Chapter 3). Based on the transient peak detection result, the EMG signal is further segmented and classified into various groups of monosynaptic MEPs and polysynaptic MEPs using techniques stemming from Principal Component Analysis (PCA), hierarchical clustering, and Gaussian mixture model (Chapter 4). A software with graphic user interface has been implemented in Matlab. The software implements the proposed peak detection algorithm, and enables the physiologists to visualize the detection results and modify them if necessary.

Chapter 2 gives an introduction to the necessary background in order to understand the thesis. It first talks about the physiology of the generic EMG signal and the EMG signal from patients with spinal cord injuries undergoing rehabilitation. With that, the characteristics of the EMG signal and the challenge of processing it are explained. After that, a basic introduction to the classical detection theory and the wavelet transform is given, because these two are the main theories behind the transient peak detection algorithm proposed in Chapter 3. In particular, two classical detectors, the matched filter and the energy detector, are introduced because they normally serve as the upper bound and the lower bound of any given detector. Frequency properties of the wavelet transform are studied here in order to help understand the derivation of the choices of scales later in Chapter 3.

Chapter 3 extends existing theories in the transient detection field. The application of the wavelet transform in the detection of transient signals has been studied extensively and employed successfully. However, most of the theories assumes certain knowledge about the shapes of the transient signals, which makes it hard to be generalized to the transient signals with arbitrary shapes. The proposed detection scheme focuses on the more fundamental feature of most of the transient signals (in particular the EMG signal): peaks. The continuous wavelet transform with Mexican Hat wavelet is employed. This thesis theoretically derived a framework for selecting a set of scales based on the frequency domain information. Ridges are identified in the time-scale space to combine the wavelet coefficients from different scales. By imposing two thresholds, one on the wavelet coefficient and one on the ridge length, the proposed detection scheme can achieve both high recall and high precision. A systematic approach for selecting optimal parameters via simulation is proposed and demonstrated. Comparing with other state-of-the-art detection methods, the proposed method in this thesis yields better detection performance, especially in the low Signal-to-Noise-Ratio (SNR) environment.

In Chapter 4, a method for automatically segmenting and clustering the EMG signal is derived. A theoretical framework is proposed to segment the EMG signal based on the detected peaks. The scale information of the detected peaks is used to derive a measure for its effective support. Several different techniques have been adapted together to solve the clustering problem. An initial hierarchical clustering is first performed to obtain most of the monosynaptic Motor Evoked Potentials (MEPs). Principal component analysis (PCA) is used to reduce the number of features and the effect of the noise. The reduced feature set is then fed to a Gaussian mixture model (GMM) to further divide the MEPs into different groups of similar shapes. The method of breaking down a segment of multiple consecutive MEPs into individual MEPs is derived.

Chapter 5 reviews the work and the contributions of this thesis and talks about some future directions and applications.

Chapter 2

Background

This chapter gives an overview of the background knowledges in order to understand the work in this thesis in Chapter 3 and Chapter 4.

Firstly, the physiology of the EMG signal is introduced. This helps the readers to understand the characteristics of the EMG signal, and the correlation between the EMG signal and the neural activity of the spinal cord. With this understanding, readers can have a clear view of the challenges and motivations behind the work in this thesis. In addition, the assumptions made in order to derive the new detection and segmentation techniques in Chapter 3 and Chapter 4 are based on the understanding of the physiology and observation of the EMG signal.

After that, the background knowledge in order to understand the detection technique in Chapter 3 is presented. The methodology consists of a combination of several techniques stemming from multi-resolution wavelet decomposition, robust statistics, and the detection theory. Since the task is to detect transient muscle responses, it's apparent to review the classical detection theory. The classical detection theory cannot be applied directly to solve the problem, but it helps readers understand the proposed methods from a theoretical point of view. In particular, the proposed method involves the use of the generalized likelihood ratio test. Therefore, the classical likelihood ratio test (formally known as the Neyman-Pearson Theorem) is introduced. The use of the wavelet transform was inspired by many literatures in the transient detection field. Detection of the transient signals with unknown structure is generally hard, but employing certain transformation on the signal can yield good detection result, given that the transformation exposes the unique structure of the signal. Therefore, a review on the wavelet theory is given, and specifically a study of the frequency properties of the wavelets is presented, since the proposed detection method makes use of them.

2.1 Physiology of the Electromyographic (EMG) Signal

Electromyography, or EMG for short, is one of the electrophysiological recording methods. Many people are probably more familiar with other electrophysiological recording methods: Electroencephalography (EEG), the recording of electrical activity along the scalp, and Electrocardiography (ECG, or EKG), the recording of the electrical activity of the heart, etc. Similarly, EMG is a technique for evaluating and recording the electrical activity produced by skeletal muscles. The instrument for performing EMG is called Electromyograph, and the record produced is called Electromyogram, or the Electromyographic signal (the EMG signal for short)

The EMG signal is essentially the voltage fluctuation resulting from ionic current flows across the membranes of the muscle cells, when these cells are electrically or neurologically activated. Therefore, from the EMG signal, one can analyze the underlying biological processes of muscles. From that, one can further infer the neural activity of the spinal cord and potentially the central nervous system. The EMG signal can be analyzed to diagnose neuromuscular deficiencies such as caused by stroke and Parkinson's disease [50], the biomechanics of human or animal movement.

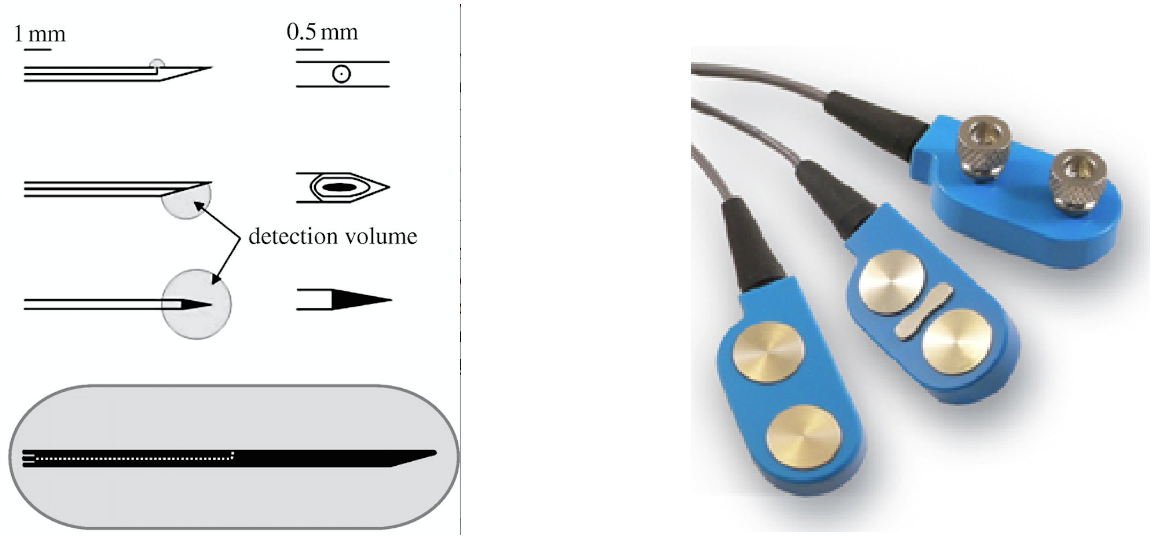
EMG can be categorized into two kinds, *surface EMG* and *intramuscular EMG*, based on the electrodes being used (See Figure 2.1-1). In Surface EMG (*sEMG* for short), a pair of electrodes or a more complex array of multiple electrodes is placed on the surface of the skin above the muscle; while in intramuscular EMG, (*iEMG* for short), typically either a monopolar or concentric needle electrode is inserted through the skin into the muscle tissue. To perform iEMG, special treatment needs to be taken, while sEMG is a non-intrusive, relatively simple approach. On the other hand, iEMG electrodes can be placed much closer to the muscle of interest, while the sEMG signal is influenced by the depth of the under-skin tissue at the site of the recording. Because of this difference, iEMG results in a much more selective, less noisy recording [53].

2.1.1 Physiology of the Generic EMG signal

Stashuk's review paper [53] on the EMG signal decomposition gives a very good explanation of how the EMG signal is generated. The following materials follow the discussion in that paper.

2.1.1.1 Muscle Fiber Action Potential (MFAP)

Muscle fibers are simply the colloquial term for muscle cells, or myocytes, which are the individual components constituting skeletal muscles. Skeletal muscle is subdivided into parallel bundles of stringlike fascicles, which themselves are bundles of even smaller stringlike multinucleated cells, the muscle fibers. Muscle fibers typically have a length of $2 - 6cm$, and a diameter of $50 - 100\mu m$ [26]. Each muscle fiber is normally innervated by only one motor neuron in only one place, usually near its



(a) Schematics of typical intramuscular electrodes [35]

(b) Picture of surface electrodes from Motion Lab Systems [56]

Figure 2.1-1: Intramuscular and surface EMG electrodes

midpoint [26]. *Neuromuscular junction* is the structure through which a motor neuron innervates its muscle fiber. When a muscle fiber is excited, it fires action potentials propagating relatively slowly ($3 - 5\text{m/s}$) in both direction away from the neuromuscular junction, similarly to the propagation of action potentials (AP) along the axons of neurons. This action potential is called a *muscle fiber action potential* (MFAP), and is the fundamental component contributing to the detected EMG signal. The characteristics of MFAPs will depend upon the diameter of the fiber, the conduction velocity, its location relative to the detection site, and the configuration and type of electrodes.

2.1.1.2 Motor Unit Action Potential (MUAP)

The fibers of a muscle are not excited individually. They are controlled together in a group, called the *motor unit*. A motor unit is made up of a single motor neuron and the skeletal muscle fibers innervated by that motor neuron. A typical muscle is controlled by about 100 large motor neurons [26]. A motor unit can innervate anywhere from 100 to 1000 muscle fibers scattered over a substantial part of the muscle. All of the muscle fibers innervated by the same motor neuron respond faithfully and synchronously to each action potential of the motor neuron [26]. As a result, individual MFAPs are normally not detected. Instead, a summation of all of a motor unit's MFAPs is detected, known as a *motor unit action potential* (MUAP).

Let $\text{MFAP}_i(t)$ be the waveform of a muscle fiber action potential from the i-th fiber of a motor unit. Let $\text{MUAP}_j(t)$ be the electrical potential from the j-th motor unit, which arises as a sum of all MFAPs:

$$\mathrm{MUAP}_j(t) = \sum_{i=1}^{N_j} \mathrm{MFAP}_i(t - \tau_i) s_i \qquad (2.1.1.1)$$

where τ_i is the temporal offset of $\mathrm{MFAP}_i(t)$, and N_j is the number of fibers in motor unit j. The binary variable s_i represents the neuromuscular junction function that has a value of 1 if fiber i fires and 0 if not.

τ_i depends on the location of the neuromuscular junction and the conduction velocity of the muscle fiber. N_j represents the size of the motor unit. As pointed out before, $N_j \sim 100 - 1000$. Because a single action potential in a motor neuron can activate hundreds of muscle fibers in synchrony, the resulting currents sum to generate an electrical signal that is readily detectable outside the muscle itself [26]. Because of the attenuation of MFAP with distance to the detection electrode, the size of the MUAP is in practice often dependent on the location and diameter of the closest few muscle fibers. Figure 2.1-2 depicts the composition of a MUAP as the summation of individual MFAPs.

In general, MUAP waveforms will vary in shape due to variations in the delays of the fiber potentials (affecting τ_i), possible changes in the position of the electrode relative to the muscle fibers (affecting MFAP_i), and the possibility of a particular fiber failing to fire (affecting s_i). These variations are the source of stochastic biological variability in the MUAP waveform [53].

2.1.1.3 Motor Unit Action Potential Train (MUAPT)

In order to maintain or increase the force generated by a muscle, the specific motor neuron must fire a temporal sequence of action potentials, called a *spike train.* As discussed in last section, one action potential from a single motor neuron results in one MUAP. Therefore, this spike train, when arriving at the neuromuscular junctions of all muscle fibers of this motor unit, results in a temporal sequence of MUAPs, called *Motor Unit Action Potential Train* (MUAPT) [53].

$$\mathrm{MUAPT}_j(t) = \sum_{k=1}^{M_j} \mathrm{MUAP}_{jk}(t - \delta_{jk}) \qquad (2.1.1.2)$$

where $\mathrm{MUAPT}_j(t)$ is the MUAPT of the j-th motor unit, $\mathrm{MUAP}_{jk}(t)$ is the MUAP generated during the k-th firing of the j-th motor unit, M_j is the number of times the j-th motor unit fires, and δ_{jk} is the k-th firing time of the j-th motor unit.

2.1.1.4 Composite EMG signal

When more than minimal force is required, many motor neurons generate an asynchronous barrage of action potentials. Due to the property of superposition of electric fields, an electrode, either inserted into a muscle or on the surface of the skin, measures the spatial and temporal sum of MUAPTs contributed from all recruited motor units within the “listening sphere”. The result is a

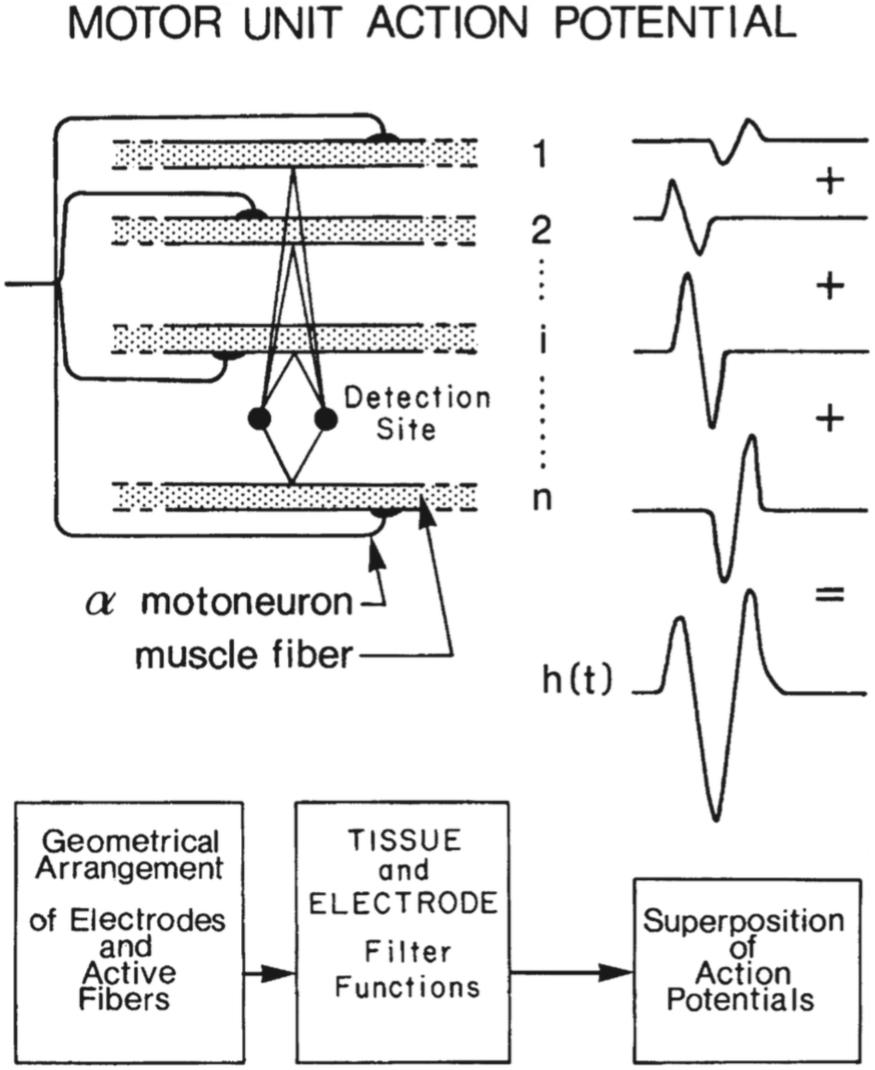

MOTOR UNIT ACTION POTENTIAL
1
2
i
n
+
+
+
=
h(t)
Detection Site
α motoneuron
muscle fiber
Geometrical Arrangement of Electrodes and Active Fibers
TISSUE and ELECTRODE Filter Functions
Superposition of Action Potentials

Figure 2.1-2: A MUAP is composed of the summation of the MFAPs of its component muscle fibers. (from Stashuk [53])

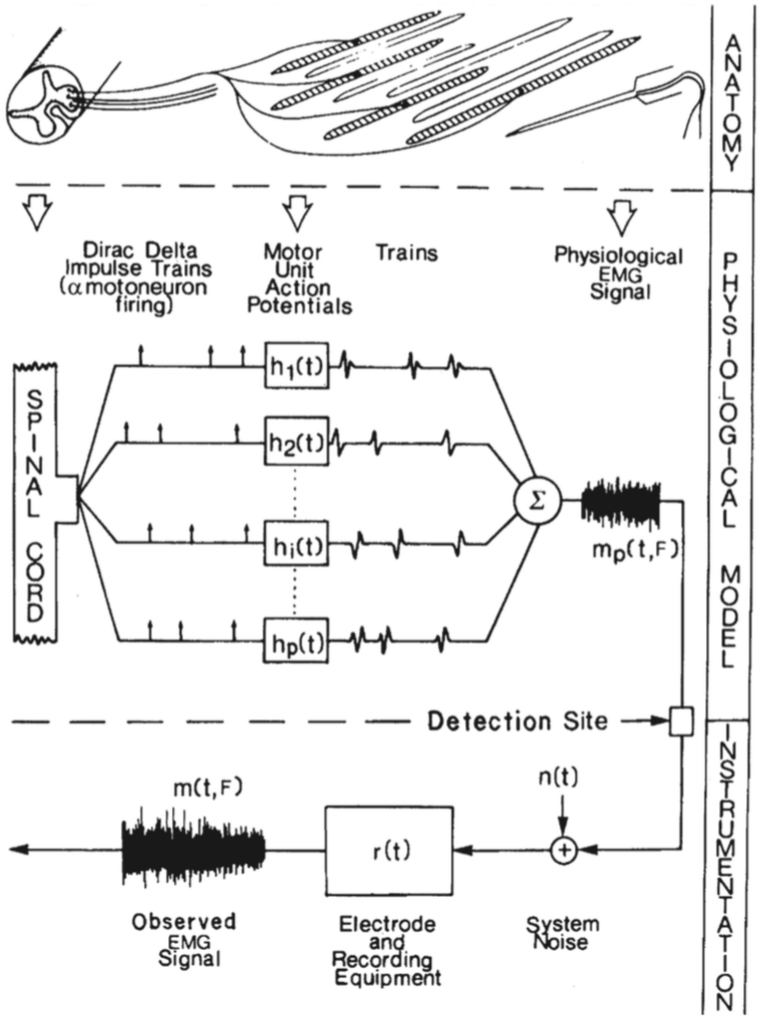


Figure 2.1-3: Physiological and mathematical model for the composition of a detected EMG signal (from Stashuk [53]).

complex pattern of electric potentials (typically in the order of $100\mu V$ in amplitude) that is called the *composite* EMG signal [26]. Figure 2.1-3 presents both an anatomical and physiological model of an EMG signal.

$$\text{EMG}(t) = \sum_{j=1}^{N_m} \text{MUAPT}_j(t) + n(t) \tag{2.1.1.3}$$

where $\text{MUAPT}_j(t)$ is the j-th MUAPT, N_m is the number of active motor units, and $n(t)$ is the background instrumentation noise.

Normally, more motor units are recruited as the muscle force increases. Different motor units are recruited at different times and stay active for different lengths of time. In addition, each MUAPT has its own characteristics of firing intervals, and this firing interval changes within each MUAPT, too. A general research direction is to decompose the detected EMG signal into its MUAPTs from

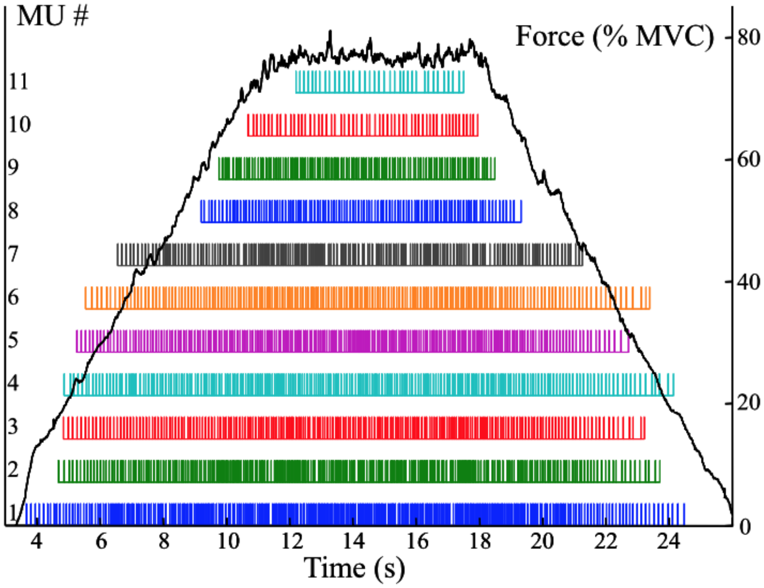


Figure 2.1-4: Bar plot for the firing times obtained via the decomposition method in [38]. MU: Motor Unit; MVC: Maximum Voluntary Contraction). (from Nawab [38])

different motor units. EMG decomposition is normally performed on the iEMG signal, since iEMG measures a few MUAPTs while sEMG detects many more, making decomposition very difficult. An example of the decomposition result on the iEMG signal from Nawab's paper [38] is shown in Figure 2.1-4.

The sEMG signal can reveal important muscle excitation information about underlying limb movement. As a result, a typical research direction is to detect muscle activation intervals in the sEMG signal. Figure 2.1-5 gives an example of muscle activity onset detection using an energy detector in [50].

As mentioned before, the shape of MUAPs depends on many different factors, such as the position of the electrode relative to the active muscle fibers, the physical characteristics and configuration of the electrodes. In addition, an EMG signal is composed of temporal overlapping of different MUAPs. As a result, it's hard to predict the actual shape of the EMG signal. This property is the major challenge in EMG processing. However, the good news is that no matter how variable the shapes can be, the effective bandwidth of the EMG signal can be assumed as prior knowledge of the physiology of EMG, as shown in Figure 2.1-6. This prior knowledge will be used in developing the EMG detection method.

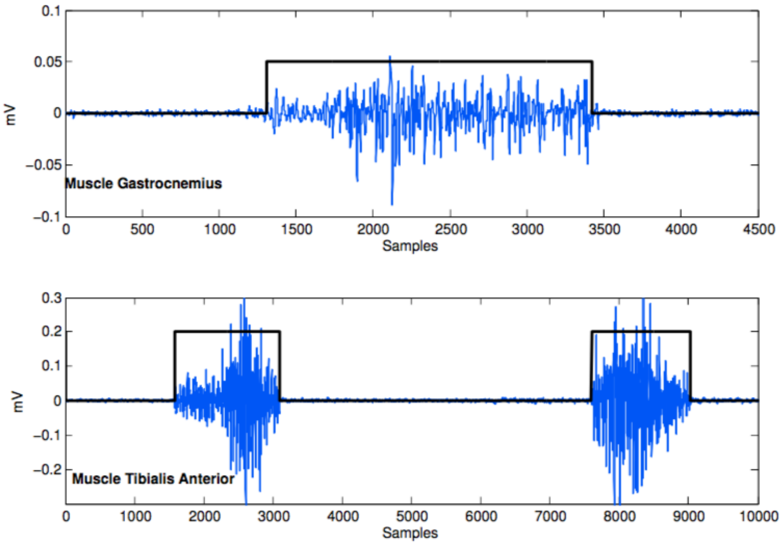


Figure 2.1-5: Muscle activity onset detection result for clinical EMG signal (from Rasool [50])

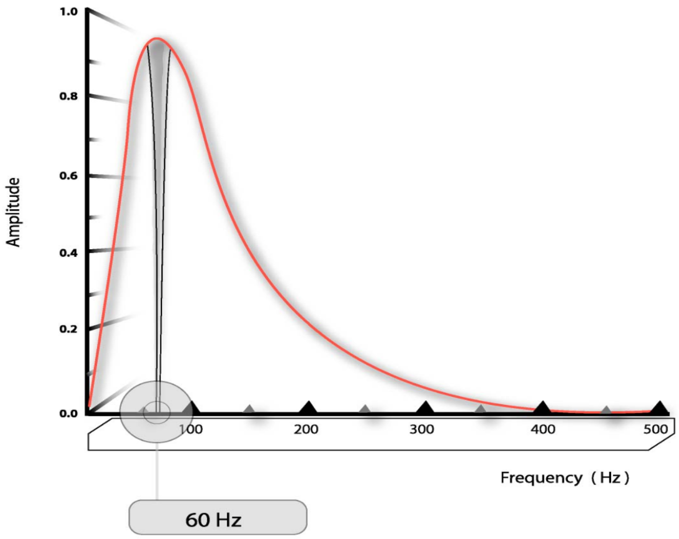


Figure 2.1-6: Schematic representation of a typical sEMG power spectrum (from Day [10])

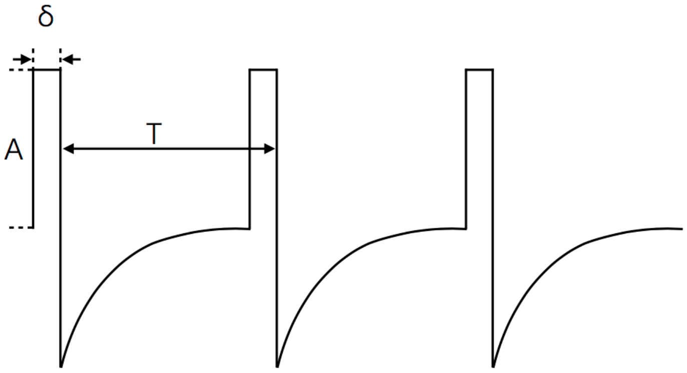


Figure 2.1-7: Schematic of electric stimuli (the actual shape of the stimulus may look different.)

2.1.2 EMG signal Resulting from Electro-stimulation: Motor Evoked Potentials

The EMG signal obtained from patients with SCI in rehabilitation training is different from the EMG signal of normal healthy spinal cords undergoing the same motions, although the fundamental physiology is similar. Recall from the previous discussion, patients with complete SCI lose all sensation and voluntary movement control below the injury level. This is because the information pathway is blocked between the brain and the neurons of the spinal cord below the lesion. As a result, the brain can no longer send or receive information from certain parts of body. Although certain locomotion control, such as stepping and standing, is governed in part by the neural circuitry within the spinal cord, this neural circuitry becomes silent after the SCI because it needs modulation and stimulation from the brain to function properly. Electro-stimulation (ES) therapy is based on the belief that this neural circuitry is intact and can resume working if given proper electrical stimulation and rehabilitation training due to plasticity of the neurons of the spinal cord. Specifically, an electrode array was implanted over the spinal cord segments to stimulate the spinal cord neurons. The electric signal can be thought of as a spike train, similar to the action potential train found in neurons. Figure 2.1-7 gives a schematic of the electric stimuli. The actual shape of one stimulus, though it may differ from the drawing, is a biphasic waveform. Each stimulus is a very short pulse, and it is repetitive with a given frequency. Many parameters associated with the stimulation can be adjusted, such as the pulse width δ, the frequency $1/T$, the amplitude A, the electrodes configuration, and electrode polarity (shown in Figure 1.2-2).

To provide some biology background in the following discussion, electric signaling in neurons is first explained.

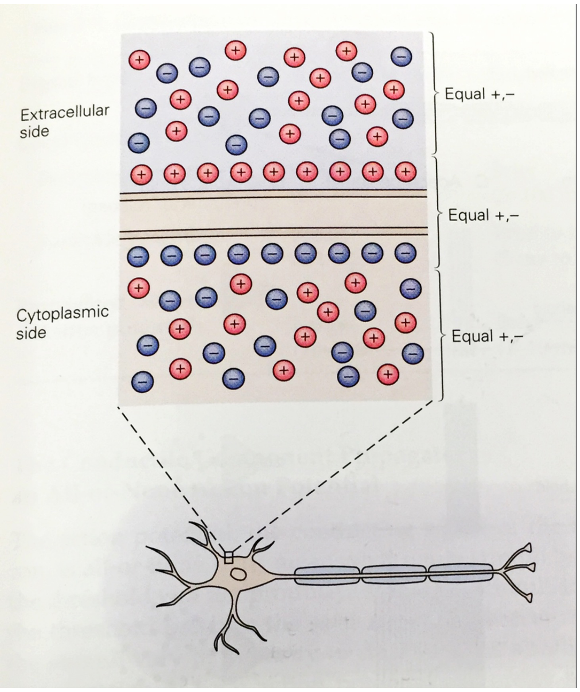


Figure 2.1-8: The membrane potential of a cell results from a difference in the net electric charge on either side of its membrane. When a neuron is at rest, there is an excess of positive charge outside the cell and an excess of negative charge inside it. (from Kandel [26])

2.1.2.1 Signaling in Neurons

At rest, all cells, including neurons, maintain a difference in the electric potential across the cell membrane. This is called the *resting membrane potential*. At rest, there are more negative charges at the cytoplasmic side, while there are more positive charges at the extracellular side (See Figure 2.1-8). By default, the membrane potential is defined as the difference obtained by subtracting extracellular potential from cytoplasmic potential. Hence, the resting membrane potential is a negative value (typical value for neurons is -65mV, typical value for muscle cells is -90mV) [26].

Excitable cells, such as neurons and muscle cells, differ from other cells in that their membrane potentials can be significantly and quickly altered; this change can serve as a signaling mechanism. The change in the membrane potential can be either a decrease or increase from the resting potential. The resting membrane potential provides the baseline: a reduction in membrane potential is called *depolarization*. Because depolarization enhances a cell's ability to generate an action potential, it is *excitatory*; an increase in membrane potential is called *hyperpolarization*. Hyperpolarization makes a cell less likely to generate an action potential and is therefore *inhibitory*. There are typically four components associated with the electric signaling in neurons and muscle cells. The four components in the list below are only an abstraction of the four functionality. Different cells have different structures and mechanisms. Figure 2.1-9 show an example of the signaling in a sensory neuron.

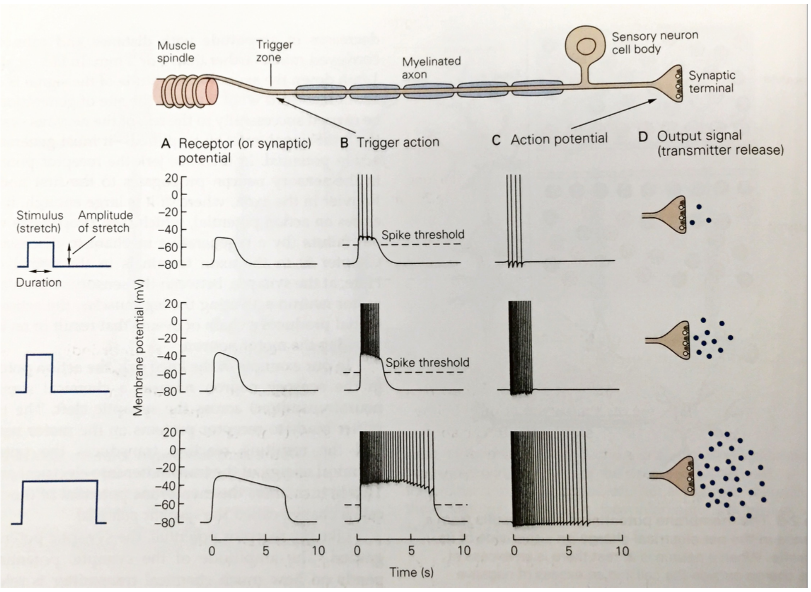


Figure 2.1-9: A sensory neuron transforms a physical stimulus (a stretch in this example) into electric signals in the neuron. Each of the neuron's four signaling components produces a characteristic signal. (from Kandel [26])

Input : Input component produces graded local signals. This signal passively propagates to other parts of the cell.

Trigger : Trigger component takes consideration of all input signals, and then makes the decision whether or not to generate action potentials.

Conduction : Conductive component actively propagates the action potentials down to the other parts of the cells. Active propagation means the amplitude of the action potentials doesn't diminish over time or distance.

Output : Output component passes the action potentials to other neurons or muscle cells. A *synapse* is a structure that permits a neuron to pass an electrical or chemical signal to another cell (a neuron or muscle cell).

There are 3 main functional groups of neurons in the spinal cord[26]:

Sensory neurons : carry information from the body's periphery into the nervous system for the purpose of perception and motor coordination.

Motor neurons : carry commands from the brain or the spinal cord to muscles and glands.

Interneurons : constitute by far the largest class, consisting of all nerve cells that are not sensory or motor neurons. They form complex neural network that enable complicated logic and decision making.

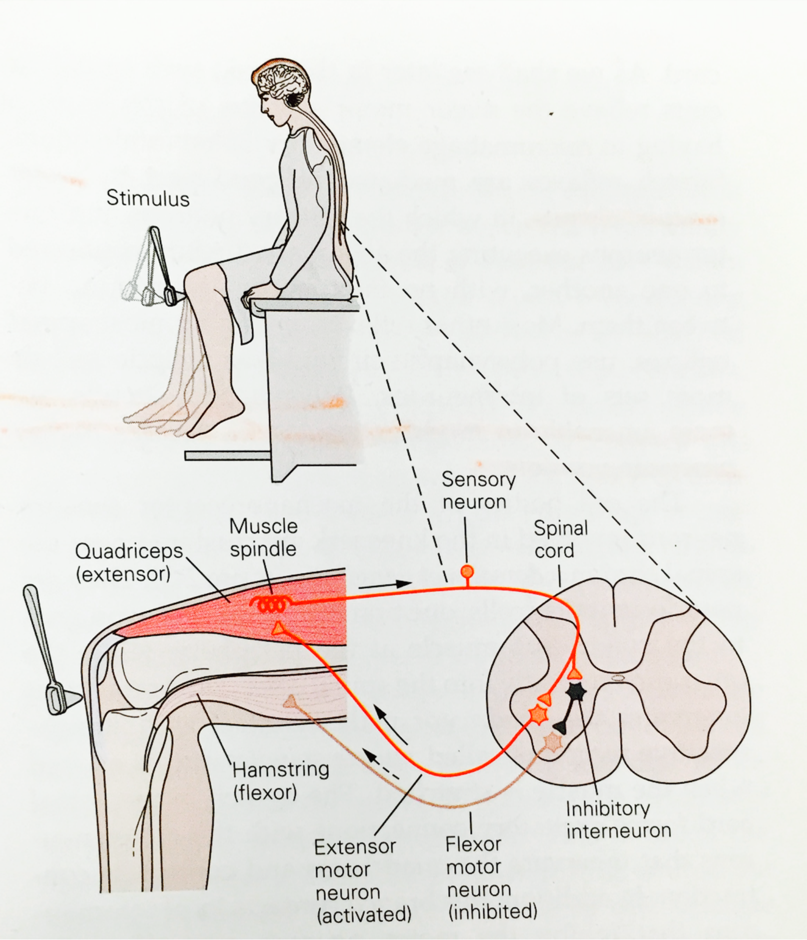


Figure 2.1-10: The knee jerk is an example of a monosynaptic reflex system, a simple behavior controlled by direct connections between sensory and motor neurons. (from Kandel [26])

There are 2 types of neural circuitry in the spinal cord [26]:

Monosynaptic circuits : the sensory neurons and motor neurons executing the action are directly connected to one another, with no interneuron intervening between them.

Polysynaptic circuits : include one or more sets of interneurons; are more amenable to modifications by the brain's higher processing centers.

Figure 2.1-10 shows the reflex mechanism of knee jerk. In this example, The extensor motor neuron is connected directly to the sensory neuron, thereby forming a *monosynaptic* circuit. It becomes active when sensory neuron is active. On the other hand, the flexor motor neuron is connected to the sensory neuron via an inhibitory interneuron, thereby forming a *polysynaptic* circuit. As a result of the inhibitory interneuron, the flexor motor neuron becomes inhibited (or inactive) when the sensory neuron is active. Overall, the extensor and the flexor motor neurons are coordinated by interneurons.

2.1.2.2 Motor Evoked Potential

Now let's come back to the EMG signal generated from patients with SCI under electrical stimulation. What happens to the neurons in the spinal cord under electrical stimulation is still an ongoing research. Here, only the fundamentals are introduced.

From previous discussion, it is shown that the action potentials in neurons can be generated when the membrane potential of the trigger zone or axon of a neuron depolarize to a certain threshold. The external electric field can affect different parts of the neurons in order to drive action potentials. Here let's focus the discussion on axons, as this is described in [26]. Again, this is only a postulate. At the presence of the external electric field, the current needs to pass through the cell membrane in order to drive a cell to threshold. In the vicinity of the positive electrode, current flows across the membrane into the axon. It then flows along the axoplasmic core, eventually flowing out through more distant regions of axonal membrane to the negative electrode in the extracellular fluids. Not all currents pass through the cell membranes; in fact, a lot more of the stimulating current move instead through the low-resistance pathway provided by the extracellular fluid. The axons with lower axial resistance to the flow of longitudinal current can pass more currents, and as a result can depolarize more efficiently. Normally, axons with larger diameters have lower axial resistance. If an axon depolarizes beyond threshold, it will then fire and propagate action potentials. This resulting action potential is called *compound action potential*.

If the external electric field *directly* excites a motor neuron (e.g., by depolarizing its axon), then the motor neuron fires an action potential and propagates it down to its muscle fibers. The result is one MUAP. I borrow the terminology from reflex physiology and call it a *monosynaptic* MUAP. If the external electric field *indirectly* excites a motor neuron, either by exciting its presynaptic interneurons (the interneurons that transmit signals to this motor neuron), or by modulating the presynaptic input signals from sensory neurons, then the motor neuron also fires an action potential and results an MUAP. I call it a *polysynaptic* MUAP.

Usually, the external electric field directly excites more than one motor neuron. All excited motor neurons will approximately fire action potentials *synchronously* (synchronized with the electrical stimuli). This is the key difference between the EMG signal from patients with SCI under electrical stimulation and the generic EMG signal. In a healthy spinal cord, when multiple motor units are recruited, they fire action potentials *asynchronously* because each motor neuron is modulated via its own complex neural circuitry formed by a large number of interneurons. Regarding the EMG signal that resulted from the electrical stimulation, because of the synchrony, all of the monosynaptic MUAPs from multiple motor units overlap with each other, and produce one large response, which I call the *monosynaptic response*.

The indirect excitation of a motor neuron is harder than direct excitation, because when a motor neuron is excited via all the synapses from all presynaptic interneurons, there need to be enough interneurons, and all the interneurons need to be coordinated properly. For example, normally, when a motor neuron is excited, its excitatory presynaptic interneuron needs to be active while its inhibitory presynaptic interneuron needs to be inactive. If both the inhibitory and excitatory interneurons are active, the motor neuron won't be excited. In the case of external electric stimulation, it's typically hard to coordinate this kind of activity among the interneurons. This is why various configurations of electrodes and different parameters of the stimuli are chosen in order to achieve certain neural activity within the spinal cord. The overlapping response from all the polysynaptic MUAPs is called a *polysynaptic response*, and is much weaker than the monosynaptic response, since much fewer motor neurons are indirectly excited. Because of the complex neural pathway between the origin of the compound action potential to the motor neuron, a polysynaptic response arrives later than a monosynaptic response, and is less synchronous with the electrical stimulus. Thereby, a monosynaptic response is also sometimes referred to as an *early response*; a polysynaptic response is referred to as a *late response*.

Collectively, both the monosynaptic response and the polysynaptic response are called the *Motor Evoked Potential* (MEP). A MUAP is no longer a proper term in the EMG signal in this thesis, as both monosynaptic and polysynaptic responses are somehow synchronized to the external electric stimuli, and there is hardly an individual MUAP in the resulting EMG signal. In this thesis, MEP refers specifically to the compound response from the patients with SCI under the electrical stimulation. Figure 1.3-2 gives an example of both an early response and a late response after one electrical stimulus (the stimulus is at the beginning of the plotted signal). As you can see from the example, the early (monosynaptic) response is much stronger than the late (polysynaptic) response.

2.2 Characteristics of MEPs and Challenges of Processing Them

The first part of this section shows some of the major characteristics of the EMG signal with example figures. With the physiology background discussed above, the readers can have a deep understanding of the characteristics. Next, the challenges arising from these characteristics will be listed, along with the major insufficiencies of some prior work.

2.2.1 Characteristics of MEPs

The biggest challenge is the randomness of the MEPs. The parameters of a MEP are not deterministic, and must be modeled by random variables. The actual probability model and its parameters

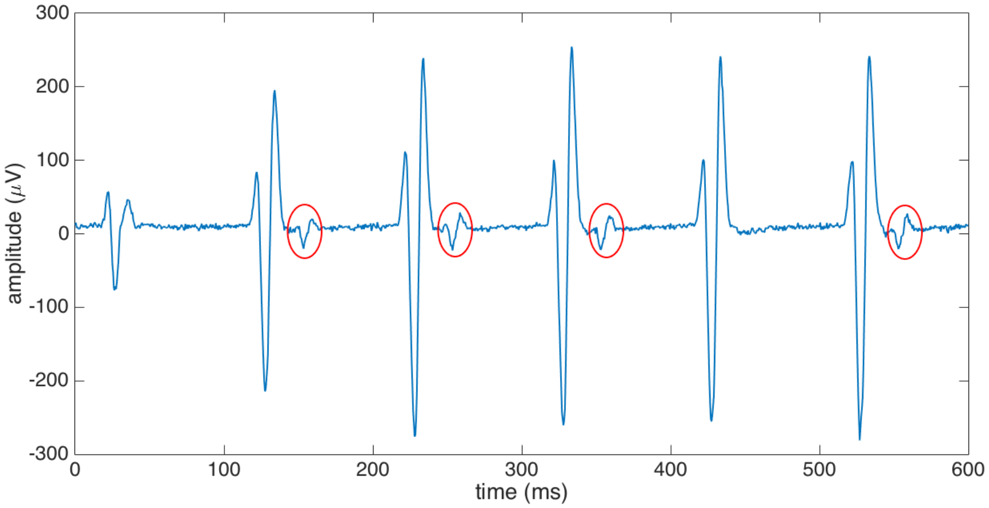


Figure 2.2-1: Example EMG signal containing MEPs with low SNR (marked by red circles). (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

are also unknown, which makes the problem even harder.

Figure 2.2-1 shows that the arrival times of the MEPs are random. The MEPs in the red circles are polysynaptic ones (the weak ones), while the rest are monosynaptic ones (the strong ones). The arrival times of the monosynaptic MEPs, although still random, show certain regular pattern, so a reasonably good probability model can be sufficient. However, the arrival times of the polysynaptic MEPs have limited pattern, and a proper probability model is therefore hard to determine. As pointed in the physiology section of this chapter, the arrival time of a MEP depends on many different factors, such as the conduction of the motor neuron axons, the detection location relative to the neuromuscular junctions, and the overall conduction speed of the muscle fibers. Moreover, in the case of polysynaptic MEPs, the neural pathway between the source of the compound action potential and the excited motor neuron is very complex and totally unpredictable.

Figure 2.2-2 shows different waveforms of the MEPs from different muscles of one training session. Figure 2.2-3 shows different waveforms of the MEPs from one single muscle of one training session. The two figures show that the durations and the shapes of the MEPs vary a lot. The lack of information on the structure of a signal leads to great difficulty in processing it. Later in this section, this difficulty will be elaborated within the context of prior work. There are many different factors contributing to the varying shapes of the MEPs. As shown in the physiology section, every MUAP consists of multiple MFAPs. MUAP waveforms vary in shape due to variations in the delays of the MFAPs, and the number of muscle fibers that fire. In addition, there is a lot of substances between the muscle fibers and the detection site, including a layer of fat tissues and skin. All these substances

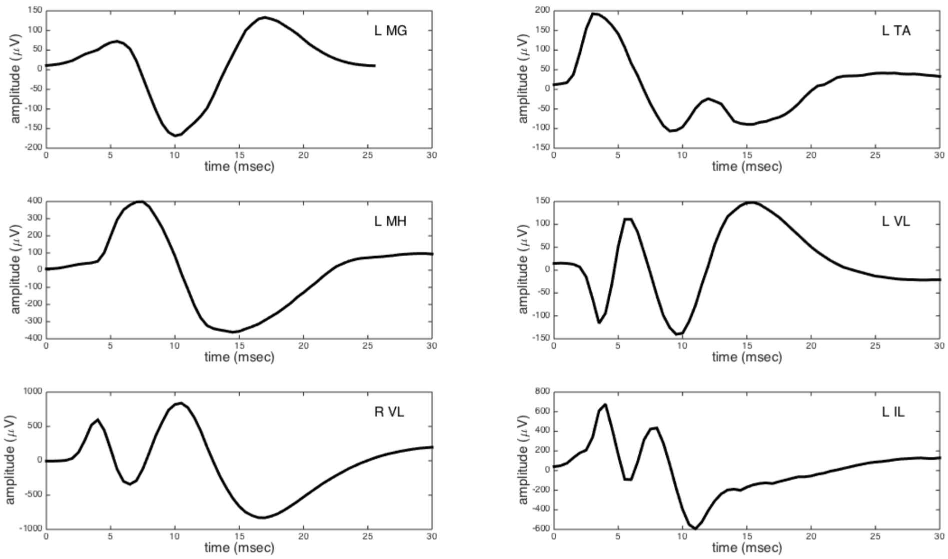


Figure 2.2-2: Examples of MEP waveforms from different muscles of one patient under one rehabilitation session. The muscle from which each MEP waveform is from is shown in its short name on the upper right corner of each subplot. For the full names of the muscles, refer to Appendix A. (The example EMG signal is from various muscles while the patient is lying in supine position under EES.)

degrade the MUAPs significantly. The configuration of the electrodes can also alter the waveform in an unpredictable way. Totally, the shapes of the resulting MUAPs are completely random and unpredictable. Moreover, an MEP is a superposition of multiple MUAPs. In the rehabilitation training with electrical stimulation, the spinal cord is damaged, and hence there is little to know about how many motor neurons are excited and which they are. This further adds to the complexity of the shapes of the MEPs. The polysynaptic MEPs have a even less regular structure than the monosynaptic ones, because the complex neural pathway results in a complicated, asynchronous overlapping of the MUAPs. The only common feature from all the MEPs is that they all contain multiple transient peaks, although the shapes of the peaks, such as the widths and heights, are still random. Also the number of peaks within one MEP is unknown.

Figure 2.2-1 and Figure 2.2-4 show that the polysynaptic MEPs have an extremely low signal-noise ratio (SNR). This is another major difference between MEPs found in the electro-stimulation induced EMG signal and the MUAPs found in the generic EMG signal. Due to the spinal cord injury, lots of the neurons in the spinal cord below the lesion are inactive. A lot less motor neurons can fire from the excitation of their presynaptic interneurons. As a result, the polysynaptic MEPs are very weak.

The last characteristic of the EMG signal is shown in Figure 2.2-5. As commonly seen in the recording of any electrical signals, the EMG signal suffers from the baseline fluctuation.

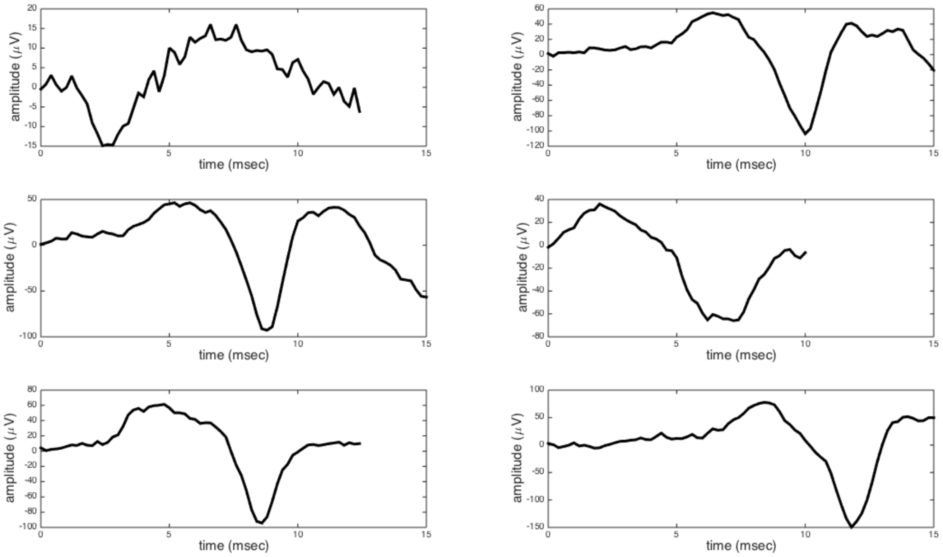


Figure 2.2-3: Examples of MEP waveforms from the same muscle. (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

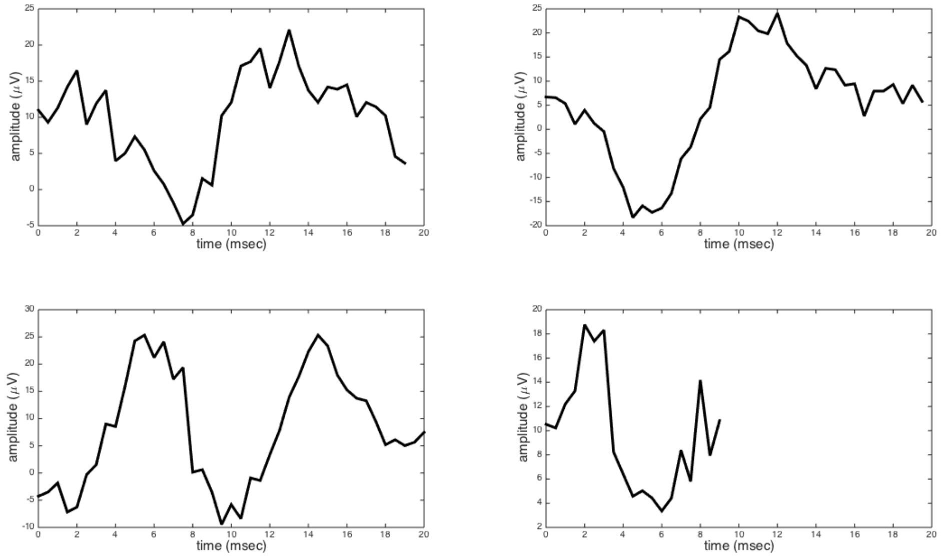


Figure 2.2-4: Examples of MEP waveforms with low SNR. (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

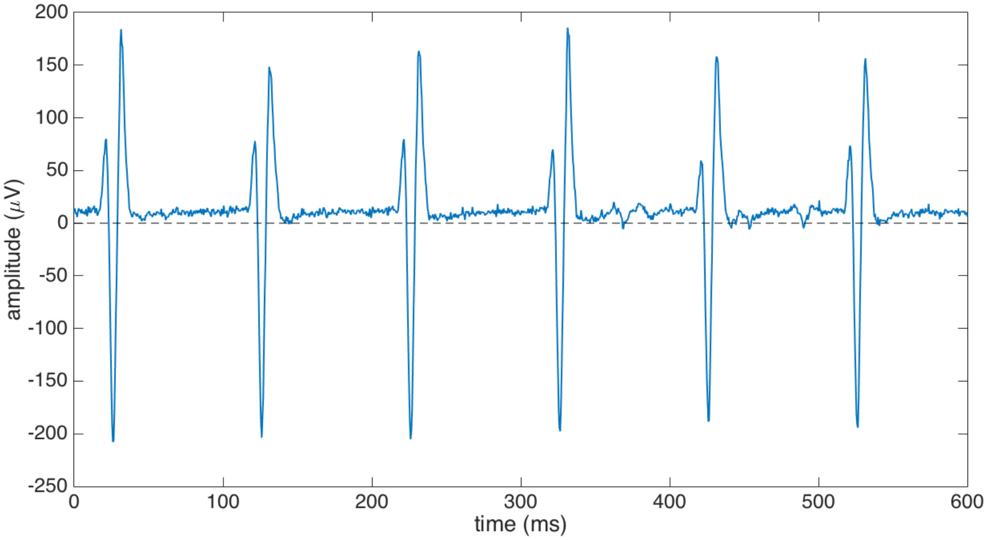


Figure 2.2-5: Examples of the baseline fluctuation in the EMG signal. The baseline deviates from 0 (marked by a dashed horizontal line), and changes slowly over time. (The example EMG signal is from the muscle of left medial gastrocnemius while the patient is lying in supine position under EES.)

All the previous discussion is limited to the variations of the MEPs within one person from one training session. In practice, all the characteristics of the MEPs also vary from session to session, and from person to person. The spinal cord is injured differently in different patients. The strength of the muscles also varies a lot among individuals. Finally, in every training session, the EMG electrodes are placed by practitioners manually, resulting in different detection sites. The location of the detection site also has an impact on the shapes of the MEPs.

2.2.2 Challenges of Processing MEPs

Detailed literature reviews are given within Chapter 3 and Chapter 4. A brief introduction to the prior work is presented here as a context for the discussion of the challenges in the EMG processing.

Traditionally, the detection of signals in a noisy observation is formulated as a binary hypothesis testing problem in the detection theory. When the probability models of the signal and the noise are fully known, an optimal detector with constant false-alarm rate can be formulated as a likelihood ratio test according to Neyman-Pearson Theorem [28]. Please refer to Section 2.3 for an overview of the binary hypothesis testing problem and the likelihood ratio test. For example, when the shape of the signal is completely known, then a matched filter gives the optimal detection performance [28]. Normally, a matched filter is used to give the upper bound on the detection performance. On the contrary, if there is absolutely nothing known about the signal, then an energy detector gives the optimal detection performance [28]. An energy detector normally serves as the lower bound when evaluating a detector. When certain parameters in the model of either the signal or the noise are unknown, then a generalized likelihood ratio test can be formulated with the parameters being their maximum likelihood (ML) estimates [28]. The shapes of the MEPs are unknown, so a matched filter cannot be applied directly. The use of an energy detector is insufficient because it doesn't use the structure of the MEPs at all and hence gives the worst detection performance. The idea is to find a certain representation of the EMG signal, such that the feature or structure of the MEP is exposed in that representation. Then a binary hypothesis testing problem can be formulated in the new representation.

Many different representations have been proposed in the field of transient detection. In particular, the wavelet transform is proven to be successful for a variety of signals. Wavelet transform gives the local feature of a signal rather than a global feature. So it naturally works well on a transient signal. In addition, Wavelet transform can expose features at different levels by specifying different scale values. Some prior work on the transient-signal detection with wavelet transform includes [21], in which the signal is assumed to have a known shape, but unknown arrival time and scaling. In [19], the signal is unknown, but its bandwidth and time-bandwidth-product are assumed to be known. The methods in the prior work are either insufficient because of the strong constraints made on the signal model, or not applicable to the case of peak detection.

The methodology proposed in Chapter 3 combines detection theory with wavelet transform. As a result, a brief review of some background knowledge in the detection theory and wavelet transform is given in Section 2.3 and Section 2.4, respectively.

2.3 Classical Detection Theory

The first task to address in this thesis is to detect the transient muscle responses (more specifically, the MEP peaks). The detection of the signal corrupted by noise is studied in the detection theory. In classical detection theory, the detection of the signal out of noise is formulated as a binary hypothesis testing problem, so I will first review the binary hypothesis testing problem, and I will introduce the fundamental theorem for solving it: the Neyman-Pearson Theorem. The theorem introduced a detection scheme called likelihood ratio test that gives an optimal detector given the full probability models of the signal and the noise. In the proposed detection method, a generalized likelihood ratio test is used to tackle the problem of unknown parameters in the signal model.

In Chapter 3, the proposed detector is compared against other detectors in the literature. As a result, a review of the performance metrics of binary detection is presented in Section 2.3.2. There are many different evaluation metrics in different applications. This thesis uses two statistics called *recall* and *precision*, which are widely used in the field of pattern recognition and machine learning. These two statistics are chosen because they give the most important information about the detection performance in this application: detection of transient MEP peaks from an EMG signal. Basically, from recall one can tell how many MEP peaks are detected among all the true MEP peaks, and precision shows how many detected peaks are true MEP peaks. Recall is important because a practical detector sometimes misses a true signal, and precision is important as any practical detector sometimes detects noise as a signal. Other statistics are either equivalent to recall or precision, or less important to the task.

2.3.1 Binary Hypothesis Testing and Neyman-Pearson Theorem

In classical detection theory, a signal-detection problem is often formulated as a binary hypothesis testing problem, where under the null hypothesis $\mathcal{H}_0$ the signal is not present, and under the alternative hypothesis $\mathcal{H}_1$ both the signal and the noise are present.

Suppose N observations $x[n]$, $n = 0, 1, \cdots, N-1$, are generated depending on the hypothesis:

$$\mathcal{H}_0 \;:\; x[n] = w[n] \qquad\qquad w[n] \sim \mathcal{N}(0, \sigma^2) \quad \textit{i.i.d.} \tag{2.3.1.1a}$$

$$\mathcal{H}_1 \;:\; x[n] = s[n] + w[n] \qquad\qquad w[n] \sim \mathcal{N}(0, \sigma^2) \quad \textit{i.i.d.} \tag{2.3.1.1b}$$

where $x[n]$ represents a noisy observation at a discrete time n, $s[n]$ is the transient signal to be detected and $w[n]$ is the background white noise.

A binary detector maps the observation into either $\mathcal{H}_0$ or $\mathcal{H}_1$. If I use notation $P(\mathcal{H}_i; \mathcal{H}_j)$ to represent the probability of deciding $\mathcal{H}_i$ when $\mathcal{H}_j$ is true, then there are four probability associated with a given binary detector.

- $P(\mathcal{H}_0; \mathcal{H}_0)$ = probability of correct non-detection
- $P(\mathcal{H}_0; \mathcal{H}_1)$ = probability of missed detection = P_M
- $P(\mathcal{H}_1; \mathcal{H}_0)$ = probability of false alarm = P_{FA}
- $P(\mathcal{H}_1; \mathcal{H}_1)$ = probability of detection = P_D

When the full knowledge of the statistics of the signal $s[n]$ and the noise $w[n]$ is given, then an optimal detector exists according to *Neyman-Pearson Theorem*:

Theorem 1 (Neyman-Pearson Theorem) *To maximize P_D for a given $P_{FA} = \alpha$, decide $\mathcal{H}_1$ if:*

$$L(x) \overset{\text{def}}{=} \frac{p(x; \mathcal{H}_1)}{p(x; \mathcal{H}_0)} > \gamma \tag{2.3.1.2}$$

where the threshold γ is found from:

$$P_{FA} = \int_{\{x:L(x)>\gamma\}} p(x;\mathcal{H}_0)dx = \alpha \tag{2.3.1.3}$$

The Eq. (2.3.1.2) is called *likelihood-ratio test (LRT)*, because the left-hand side $L(x)$ is the ratio of the data likelihood under $\mathcal{H}_1$ over $\mathcal{H}_0$. The detector given by the Neyman-Pearson Theorem is also referred to as the *Constant False Alarm Rate* (CFAR) detection, as the detector maintains a constant P_{FA}.

The Neyman-Pearson theorem can be applied when the statistics of the signal $s[n]$ and the noise $w[n]$ are fully known, so that the likelihood ratio can be analytically derived. When the statistics of the signal $s[n]$ are not completely known, then the *generalized likelihood ratio test* (GLRT) can be formulated as follows.

Suppose the statistics of the signal $s[n]$ depends on the parameter vector $\boldsymbol{\theta}$, then the likelihood ratio is:

$$L(x) \overset{\mathsf{def}}{=} \frac{p(x;\hat{\boldsymbol{\theta}},\mathcal{H}_1)}{p(x;\mathcal{H}_0)} > \gamma \tag{2.3.1.4}$$

where $\hat{\boldsymbol{\theta}}$ is the *maximum likelihood* (ML) estimate of $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(x;\boldsymbol{\theta},\mathcal{H}_1) \tag{2.3.1.5}$$

Two classical detectors will be derived based on Neyman-Pearson theorem. The first one is called *matched filter*, which is derived when the signal is fully known. The other one is called *energy detector*, which is derived when nothing is known about the signal. As a result, matched filter is normally considered as an upper bound of the detection performance of any given detector, while energy detector is used as a lower bound.

In both cases, assume there are N observations $x[n]$, $n = 0, 1, \cdots, N-1$, with noise, $w[n]$, being white gaussian noise with variance σ^2.

2.3.1.1 Matched Filter

When the signal $s[n]$ is deterministic:

$$p(x;\mathcal{H}_1) = \frac{1}{(2\pi\sigma)^{N/2}} \exp\Big[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-s[n])^2\Big] \tag{2.3.1.6}$$

$$p(x;\mathcal{H}_1) = \frac{1}{(2\pi\sigma)^{N/2}} \exp\Big[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n]\Big] \tag{2.3.1.7}$$

Therefore:

$$L(x) = \exp\Big[-\frac{1}{2\sigma^2}\Big(\sum_{n=0}^{N-1}(x[n]-s[n])^2 - \sum_{n=0}^{N-1}x^2[n]\Big)\Big] > \gamma \tag{2.3.1.8}$$

Take the logarithm on both sides and simple steps yield the log likelihood ratio test:

$$\ln(L(x)) = \frac{1}{\sigma^2}\sum_{n=0}^{N-1}x[n]s[n] - \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}s^2[n] > \ln\gamma$$

Since $s[n]$ is known:

$$T(x) \overset{\text{def}}{=} \sum_{n=0}^{N-1}x[n]s[n] > \sigma^2\ln\gamma + \frac{1}{2}\sum_{n=0}^{N-1}s^2[n] \overset{\text{def}}{=} \gamma\prime$$

Or:

$$T(x) = \sum_{n=0}^{N-1}x[n]s[n] > \gamma\prime \tag{2.3.1.9}$$

$T(x)$ is called the *test statistic*, as used in statistical hypothesis testing. The test statistic in Eq. (2.3.1.9) is obtained by correlating a known signal, or template, with the observation, and is therefore called the *matched filter*. It is also sometimes referred to as the *replica-correlator*.

2.3.1.2 Energy Detector

When nothing is known about the signal $s[n]$, the parameter θ as in the generalized likelihood ratio test is the signal itself, $\boldsymbol{\theta} = \mathbf{s}$. As a result:

$$\begin{aligned}\hat{\mathbf{s}} &= \arg\max_{\mathbf{s}} p(x;\mathbf{s},\mathcal{H}_1)\\ &= \arg\max_{\mathbf{s}} \frac{1}{(2\pi\sigma)^{N/2}}\exp\Big[-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-s[n])^2\Big]\\ &= \mathbf{x}\end{aligned}$$

It follows:

$$p(x;\hat{\mathbf{s}},\mathcal{H}_1) = \frac{1}{(2\pi\sigma)^{N/2}} \tag{2.3.1.10}$$

$p(x;\mathcal{H}_0)$ is the same as in Eq. (2.3.1.7). The generalized likelihood ratio test follows:

$$L(x) = \exp\Big[\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n]\Big] > \gamma \tag{2.3.1.11}$$

Take the logarithm on both sides and simple steps yield the log likelihood ratio test:

$$\ln(L(x)) = \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n] > \ln(\gamma)$$

$$T(x) \overset{\text{def}}{=} \sum_{n=0}^{N-1}x^2[n] > 2\sigma^2\ln\gamma \overset{\text{def}}{=} \gamma\prime$$

Or:

$$T(x) = \sum_{n=0}^{N-1}x^2[n] > \gamma\prime \tag{2.3.1.12}$$

The test statistic $T(x)$ in Eq. (2.3.1.12) is obtained from the energy of the observation $x[n]$, and is therefore called the *energy detector.*

2.3.2 Performance Metrics of Binary Detection

The detector of a binary hypothesis testing problem is also sometimes called a binary classifier or predictor. There are many different metrics that can be used to measure the performance of a binary classifier. Different metrics are used in different fields due to different goals. Sometimes, the same metrics are given different names in different applications. This section first gives a general overview of the fundamental metrics. After that, some of the metrics that are used throughout the thesis are highlighted.

In Eq. (2.3.1.1), the observation in which a signal is absent (e.g., null hypothesis $\mathcal{H}_0$ is true), is often called a *negative*; while the observation in which a signal is present (e.g., alternative hypothesis $\mathcal{H}_1$ is true) is often called a *positive*. The detector classifies the observation as either from the null hypothesis $\mathcal{H}_0$ or the alternative hypothesis $\mathcal{H}_1$. To evaluate the detector, one compares the classification results to the ground truth and cross tabulates the data into a 2x2 contingency table or *confusion matrix* [54].

	$\mathcal{H}_0$ true	$\mathcal{H}_1$ true
predict $\mathcal{H}_1$	False Positive	True Positive
predict $\mathcal{H}_0$	True Negative	False Negative

Table 2.1: Confusion matrix of a binary classifier

One can then evaluate the detector by counting the following 4 numbers:

- **FP:** number of false positives
- **TP:** number of true positives
- **TN:** number of true negatives
- **FN:** number of false negatives

There are 8 possible ways to evaluate the detection performance by dividing each number by its row sum and column sum. However, only 4 of them are independent. The other 4 are just their ones' complements.

In the field of detection theory [28], following two statistics are often used:

- **Detection rate:** $\frac{TP}{TP+FN}$, the percentage of true positives that are labeled as positives.
- **False alarm rate:** $\frac{FP}{FP+TN}$, the percentage of true negatives that are labeled positives.

Ideally, the detection rate should be 1, while the false alarm rate is 0. For a practical detector, there is always a trade-off between the detection rate and the false alarm rate. To compare two different detectors, normally people draw the *Receiver Operating Characteristic (ROC)* [28]. The ROC plot illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting detection rate against the false-alarm rate at various threshold settings.

In the field of pattern recognition or machine learning, two statistics are mostly often used:

- **Recall:** $\frac{TP}{TP+FN}$, the percentage of true positives that are labeled as positives.
- **Precision:** $\frac{TP}{TP+FP}$, the percentage of labeled positives that are true positives.

Recall is equivalent to detection rate in the detection theory. Ideally, both recall and precision are 1. In practice, there is always a trade-off between precision and recall: increasing recall normally decreases precision and vice versa. By choosing a good detector, one can achieve both high recall and precision. One can plot recall vs. precision, a plot similar to the ROC.

When precision and recall are used to quantify the performance of a classifier, it's hard to compare the performance of two different classifiers, since one classifier could have higher recall but lower precision. To compare the overall performance by incorporating both recall and precision, people commonly use the *F-score*. The traditional F-measure or balanced F-score (F_1 score) is the harmonic mean of the precision and the recall:

$$F_1 \overset{\text{def}}{=} 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.3.2.1}$$

The general formula for positive real β is:

$$F_\beta \overset{\text{def}}{=} (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{2.3.2.2}$$

By choosing different values of β, the F-score puts different weights on precision and recall: a larger β means more emphasis on recall while a smaller β means more emphasis on precision.

There are many other statistics defined for other applications. Usually they are either equivalent to each other, or you can find one from the other (e.g., Sum of the two is 1). For a complete list of all the statistics, please refer to [54].

2.4 Wavelet Transform

The detection of a transient signal with unknown arrival time, unknown duration, unknown shape is difficult to solve, and there is no universal optimal detectors. A lot of transient detection work explored different models, transformation, or representation of the signal in order to expose its innate, distinct structure. A proper representation of the signals that takes advantage of the prior knowledge about the structure of the signals normally yields better detection performance [20, 49, 18]. The use of wavelet transform as a multi-resolution decomposition technique in the field of transient detection has been proven successful [19, 21, 33]. The continuous wavelet transform is used by the proposed detection methodology in Chapter 3, and therefore reviewed here. In particular, the frequency properties of the wavelets are derived, because the choice of scales in the proposed detector depends on them, and will be discussed in details in Section 3.3.2 of Chapter 3.

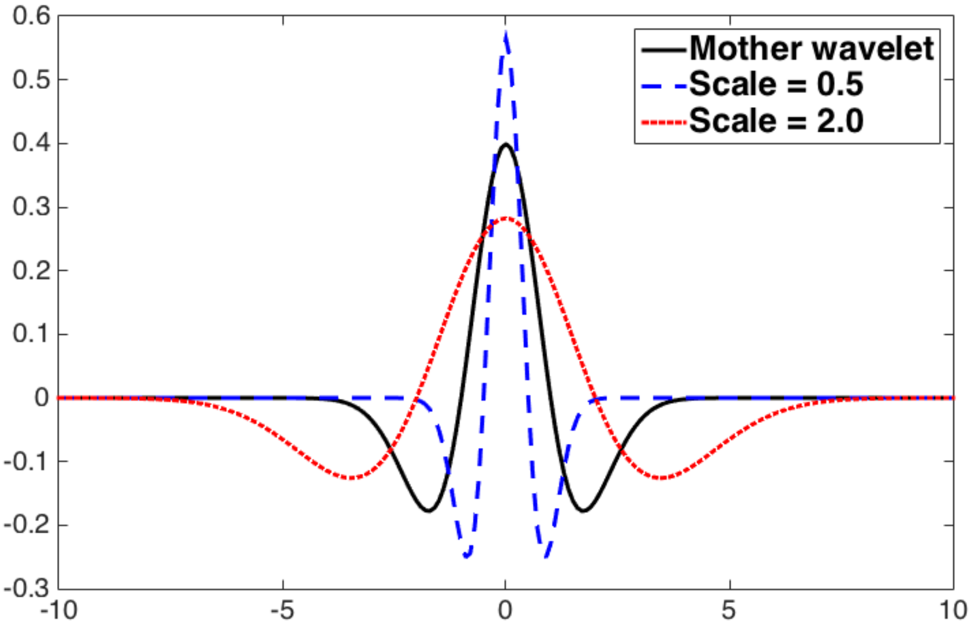


Figure 2.4-1: Mexican Hat Wavelets at different scales. Mexican Hat mother wavelet is defined by Eq. (3.3.1.1)

2.4.1 Mother Wavelet and Wavelets

A mother wavelet $\psi(t)$ has the following two properties:

$$\int_{-\infty}^{+\infty} \left|\psi(t)\right|^2 dt = 1 \tag{2.4.1.1a}$$

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \tag{2.4.1.1b}$$

For a mother wavelet: $\psi(t)$, the wavelet with scale s and translation u is:

$$\psi_{s,u}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-u}{s}) \tag{2.4.1.2}$$

Eq. (2.4.1.2) indicates that the wavelet becomes wider (has larger support) as scale s increases, and narrower (smaller support) as scale s decreases (see Fig. 2.4-1).

You can easily prove that all wavelets satisfy the properties of mother wavelet as in Eq. (2.4.1.1):

$$\int_{-\infty}^{+\infty} \left|\psi_{s,u}(t)\right|^2 dt = 1 \tag{2.4.1.3a}$$

$$\int_{-\infty}^{+\infty} \psi_{s,u}(t) dt = 0 \tag{2.4.1.3b}$$

2.4.2 Continuous Wavelet Transform

The continuous wavelet transform (CWT) of a function $x(t)$ is defined as:

$$X(s,u) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} x(t)\bar{\psi}_{s,u}(t)dt \qquad (2.4.2.1)$$

where $\bar{\psi}_{s,u}(t)$ is the complex conjugate of the wavelet $\psi_{s,u}(t)$.

From the definition, the wavelet transform gives the inner product between the function $x(t)$ and wavelet $\psi_{s,u}(t)$. Since $\psi_{s,u}(t)$ is only non-vanishing in the neighborhood of u, $X(s,u)$ gives the local information of $x(t)$ around u. Furthermore, $X(s,u)$ measures the resemblance between function $x(t)$ around u and $\psi_s(t)$, the mother wavelet scaled by s,

$$\psi_s(t) \overset{\text{def}}{=} \psi_{s,0}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t}{s}) \qquad (2.4.2.2)$$

From Eq. (2.4.1.2) and Eq. (2.4.2.2), the follow equation can be derived:

$$\psi_{s,u}(t) = \psi_s(t-u)$$

Using $\psi_s(t)$ instead of $\psi_{s,u}(t)$, Eq. (2.4.2.1) can be rewritten as (by substituting t with $t+u$):

$$X(s,u) = \int_{-\infty}^{+\infty} x(t+u)\bar{\psi}_s(t)dt \qquad (2.4.2.3)$$

Here is another way to view the CWT when the mother wavelet is a *symmetric*, *real-valued* function.

$$\begin{aligned}
X(s,u) &= \int_{-\infty}^{+\infty} x(t)\psi_{s,u}(t)dt \\
&= \int_{-\infty}^{+\infty} x(t)\psi_s(t-u)dt \\
&= \int_{-\infty}^{+\infty} x(t)\psi_s(u-t)dt \\
&= (x * \psi_s)(u) \qquad (2.4.2.4)
\end{aligned}$$

From Eq. (2.4.2.4), CWT can also be viewed as a convolution between the function $x(t)$ and scaled mother wavelet $\psi_s(t)$, when the mother wavelet is a *symmetric*, *real-valued* function.

2.4.3 CWT on Discrete-time Signals

When implementing CWT on a computer, one need to adapt above equations to their discrete versions. When performing the continuous wavelet transform to discrete-time signals, both signal $x(t)$ and wavelets $\psi_{s,u}(t)$ become their sampled versions $x[n]$ and $\psi_{s,k}[n]$, respectively, for $n \in (\mathbb{Z})$,

$k \in (\mathbb{Z})$.

Define:

$$\psi_s[n] \overset{\text{def}}{=} \psi_{s,0}[n] \tag{2.4.3.1}$$

which is the scaled version of the discrete-time mother wavelet. Eq. (2.4.2.1) and Eq. (2.4.2.3) become:

$$X_s[k] = \sum_{n=-\infty}^{+\infty} x[n]\bar{\psi}_{s,k}[n] \tag{2.4.3.2}$$

$$= \sum_{n=-\infty}^{+\infty} x[n+k]\bar{\psi}_s[n] \tag{2.4.3.3}$$

For real-valued, symmetric wavelets, Eq. (2.4.2.4) becomes:

$$X_s[k] = \sum_{n=-\infty}^{+\infty} x[n]\psi_s[k-n] = (x * \psi_s)[k] \tag{2.4.3.4}$$

In all equations above, scale s can still be an arbitrary, positive real number.

2.4.4 Frequency-domain Properties of Wavelets

Suppose the Fourier transform of $\psi(t)$ is $\hat{\psi}(\omega)$, and the Fourier Transform of $\psi_{s,u}(t)$ is $\hat{\psi}_{s,u}(\omega)$:

$$\psi(t) \xrightarrow{\mathscr{F}} \hat{\psi}(\omega)$$

$$\psi_{s,u}(t) \xrightarrow{\mathscr{F}} \hat{\psi}_{s,u}(\omega)$$

where $\mathscr{F}$ is the Fourier transform.

From Eq. (2.4.1.2):

$$\hat{\psi}_{s,u}(\omega) = \sqrt{s}\hat{\psi}(s\omega)e^{-i2\pi u\omega} \tag{2.4.4.1}$$

The mother wavelet $\psi(t)$ is essentially a band-pass filter that is centered at C_ψ, and has the bandwidth of BW_ψ

From Eq. (2.4.4.1):

$$C_\psi(s) \overset{\text{def}}{=} \text{center frequency of wavelet } \psi_{s,u}(t)$$

$$= \frac{1}{s}C_\psi \tag{2.4.4.2a}$$

$$BW_\psi(s) \overset{\text{def}}{=} \text{bandwidth of the wavelet } \psi_{s,u}(t)$$

$$= \frac{1}{s}BW_\psi \tag{2.4.4.2b}$$

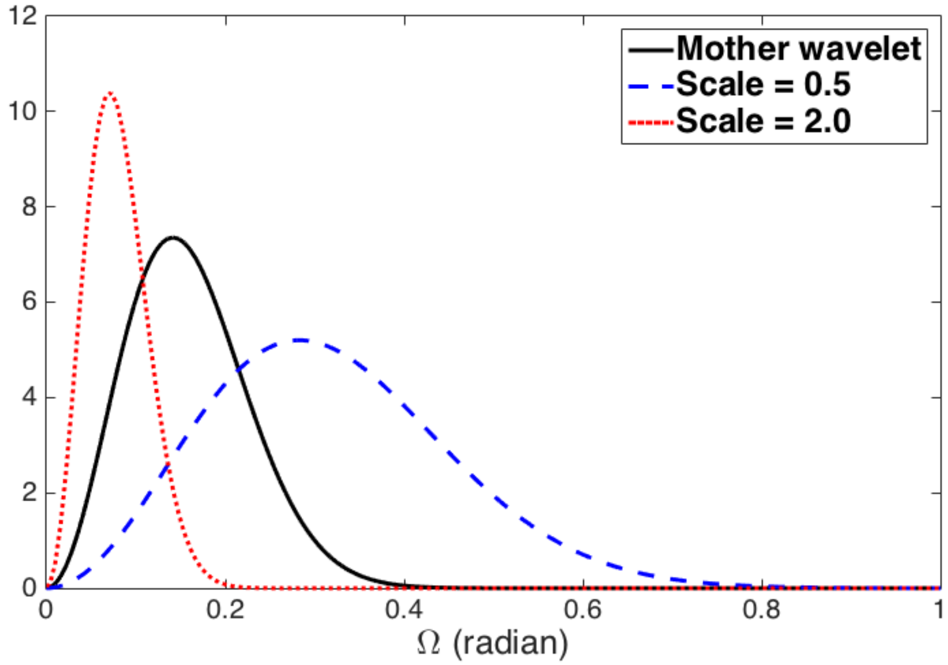


Figure 2.4-2: Discrete-time Fourier Transform of Mexican Hat Wavelets at different scales

Therefore, every wavelet at scale s is again a band-pass filter, although with a different center frequency and bandwidth. $\hat{\psi}_{s,u}(\omega)$ is centered at $C_\psi(s) = \frac{1}{s}C_\psi$, and has a bandwidth of $BW_\psi(s) = \frac{1}{s}BW_\psi$. Fig. 2.4-2 shows the magnitude of the Discrete-time Fourier Transform (DTFT) of the Mexican hat wavelets at different scales. The DTFT is periodic with period of 2π. For real-valued signals, the magnitude of the DTFT is also symmetric. Hence, in Fig. 2.4-2, only the positive frequency part of the DTFT is plotted with focus on the frequency from 0 to 1 for a better view, since all the DTFTs vanish quickly beyond 1. In addition, the frequency shown in the DTFT is in units of radians, rather than Hz. If one wants to interpret the DTFT in Hz, one needs to incorporate the sampling rate of the signal.

Chapter 3

Peak-based EMG Detection Via CWT

3.1 Existing Methods

In the EMG signal detection problem, one of the main tasks is to identify transient peaks of the muscle responses, or Motor Evoked Potentials (MEPs) in the application of spinal cord injury. There are two reasons why the peaks must be detected. First, the number, locations, and amplitudes of EMG peaks are crucial for assessing the response to the treatment. Second, although MEPs can have various shapes, all MEPs contain transient peaks. By making use of the peak characteristics of MEPs, detection performance can be potentially increased. Therefore, the MEP detector involves both transient detection and peak detection. Transient detection is a theoretical field with many theories being proposed. On the other hand, peak detection is more of a practical problem encountered in various applications. In the following, the prior literature on EMG detection is first reviewed, and its shortcomings are pointed out. Then, theoretical developments on transient signal detection are reviewed. After that, the peak detection literature is reviewed with focus on applications. One goal of this work is to build upon theories of transient signal detection and algorithms from peak detection, and combine them and adapt it to the MEP detection.

3.1.1 EMG processing

Most of the methods used in the EMG community rely upon some variation of amplitude thresholding, often based on empirical formulas [34, 37, 15, 8, 7, 27]. In [15], a threshold cursor is set by an operator visually at a level to distinguish spike potentials from noise. In [7], a threshold is manually adjusted (decreasing from its maximum value 10% at each iteration) to obtain good motor unit firing rates. In [8] and [27], a threshold is calculated from the maximum value and the mean absolute value of the EMG signal as measured over a lengthy interval. In both [34] and [37], the EMG signal is first low-pass filtered to reduce the noise. In [34], a threshold is set to a coefficient (typically 3.5) times the standard deviation of the baseline noise. In [37], a trigger level is set to 10% of the maximum value. These methods could work practically, but not for the EMG application with large dynamic range of EMG and the need for unsupervised processing. An improvement over amplitude-based thresholding is to compute the signal variation and set a threshold based on that quantity [22, 42]. Both of these papers use different formulas to calculate the empirical variation of the EMG signal within a window and set a threshold. Still, many of these formulas are empirical and lack theoretical support. In [3], amplitude envelope or local energy is obtained to detect the onset of muscle activity. This is essentially a linear filtering or energy detector which will be discussed later in the review of transient signal detection methods. Other methods are based on more advanced techniques, such as matched filtering [14], *Maximum A Posteriori* (MAP [30]) estimation and wavelet transforms [36]. However, these approaches either require some kind of template, or assume that different muscle responses are the scaled versions of a prototype function. Therefore the methods can't be generalized to solve the MEP detection problem, since the MEPs have various, unpredictable shapes.

In summary, most detection techniques developed in the EMG community are relatively simple, and involve manual adjustment of parameters by human supervision. Manual operation is useful in some practice, since the operators can use their experiences and knowledge to improve performance. However, it can be very laborious for large data sets. Since the quality of the results largely depends on the skill level of the individual, it's hard to consistently get good results. Some other detection techniques use simple empirical formulas. Since the algorithm is simple, it's easy to implement and runs fast. However, they lack theoretical support, and the empirical formulas are usually only successful on specific data. In addition, most of the techniques require a window. It's not a good idea to choose a fix window for detection of transient peaks, as the peaks are really short and have various widths. A large window would lose the resolution and thus miss some transient peaks, while a small window normally yields poor detection performance. Some advanced techniques have been developed recently. However, they either suffer from the baseline and the noise, or put strong assumptions on the waveforms of the EMG signal, which make them hard to generalize and apply on an actual EMG signal. After all, all algorithms are still spike-detection in essence, but the MEPs are far more complicated than spikes. In addition, almost all the techniques' detection performance deteriorate sharply for EMG signal with low signal-to-noise ratio (SNR).

3.1.2 Transient Signal Detection

In classical detection theory, a detector is given a record of observation samples and decides whether the observation contains, in addition to noise, a signal of interest or not. This type of problem arises in many different areas, such as communication systems, radar systems, and medical diagnosis. In applications such as underwater acoustics, seismic surveillance, and EMG signal processing as in this thesis, the signal to be detected is a transient signal, a signal of short duration. In most practical applications, the complete knowledge about the location, shape, or strength of the transient signals is unknown.

There are many different techniques proposed for the detection of transient signals. Theoretically, if the statistics of the noise and the signal are completely known, then the likelihood ratio test (LRT) is formulated to obtain a constant false alarm rate (CFAR) detector. For example, when the exact shape of the signal is a known priori, then the optimal detector is given by a matched filter (MF). When some parameters (either the signal or the noise or both) are unknown, a generalized likelihood ratio test (GLRT) is then formulated with the maximum likelihood estimates (MLE) of the unknown parameters [28].

The detection problem becomes particularly challenging in the case of EMG-like transient signals, because the signal typically has unknown shape, unknown strength, and unknown location. Classical techniques are not guaranteed to work. Many different methods have been proposed to tackle the transient detection problem. Generally, if some prior information about the signals to be detected is known, then the detector, which makes good use of the prior knowledge, would generally perform well on those signals [20]. In [20], Frisch shows that a good detection performance can be achieved if the prior information about the signal is translated into a proper signal representation. Meantime, a detector based on some signal assumptions will not perform well on signals that don't satisfy the assumptions. As a result, some prior work has focused on general detectors which make few assumptions on the transient signals. For example, a plug-in power-law detector was proposed in [62].

Many methods formulate test statistics based on the notion of signal "energy" or "power". When nothing is known about the signal, the optimal detector is an energy detector [28, 60]. In an energy detector, the energy of the signal (sum of squares of the discrete-time samples) is calculated and compared against a threshold. However, the performance of the energy detector is typically not good, as it doesn't use any prior knowledge about the signal. Hence, energy detectors usually serve as a lower bound when evaluating the detection performance of any given detector. An improvement over the energy detector is the power-law detector [43, 44, 45, 62]. Nuttall formulated the power-law detector in the frequency domain via preprocessing the data by the magnitude-square DFT, and studied its performance extensively in his technical reports [43, 44, 45]. Wang, in [62], extended Nutall's idea and proposed a series of variations on the power-law detectors. In particularly, one of her power-law detectors deals with unknown, colored noise. By making use of the structure of the transient signals (real transient signals tend to aggregate their energy in a band), her power-law detector combines contiguous DFT bins. She also explored the use of wavelet transform to take advantage of the fact that real transient signals also tend to aggregate energy locally in time. Another extension to the energy detection was proposed in [55]. Instead of calculating the power or energy, the second-order statistics, a higher order statistics (in particularly, the sum of the absolute values of the third power of the observed samples) is used.

Another group of methods are based on “Page’s test” [46]. Page’s test was originally developed to detect a sudden change in the statistics of the data as quickly as possible. A transient signal can be seen as a two-sided change: when a signal s_1 arrives after an interval of signal s_0, the probability of the data switches from $p(s_0)$ to $p(s_1)$; when signal s_1 disappears at a later time, the probability returns to $p(s_0)$. Hence, Page’s test has been used for transient detection and shown to be quite useful [2, 23, 1, 5].

Other transient detection investigations explored different models, transformation or representation of the signal in order to expose its innate, distinct structure. A proper representation of the signals that takes advantage of the prior knowledge about the structure of the signals normally yields good detection performance [20, 49, 18]. In [40, 48], a transient signal is modeled as the impulse response of a rational transfer function. Many different kinds of linear transforms have been explored for transient signal detection, such as the short-time Fourier transform (STFT) [64], the Gabor transform [17], and the wavelet transform [19, 21, 33].

There are other various techniques that try to solve the transient detection problem. In [55], an approach coined as “hyperparameter estimation” is proposed, in which unknown parameter are assumed to be drawn from an underlying probability distribution, and unknown parameters about the *meta*-distribution (so called *hyperparameters*) are estimated. In [16], data is ordered before processing it, since an approach based on order statistics is robust.

3.1.3 Peak Detection

Peak detection has also been studied quite extensively in various application domains [13, 25, 41, 47, 51]. In [47], various *peak functions* were proposed in order to capture the structure of a peak. In [13, 51], a multiscale-based peak detection algorithm was proposed. Reference [51] uses *local maxima scalogram* (LMS) for the purpose of detecting periodic or quasi-periodic peaks. Reference [13] uses the continuous wavelet transform (CWT) to find peaks in the mass spectrum. Du’s wavelet idea is similar to the solution proposed in this thesis, although the proposed wavelet-based peak detection algorithm was independently developed. Compared to [13], this thesis provides more theoretical support for peak detection. In [41], peak detection based methods were used in ECG signals in cardiac MRI. In [39], the continuous wavelet transform is employed to detect action potentials in neural recordings. The technique developed in this thesis was inspired by [39]. However, action potentials have a stereotyped shape, while MEPs in the EMG signal can have various shapes, which makes the problem studied in this thesis much harder than that in [39].

Overall, the proposed peak detection algorithm employs the GLRT from detection theory, the continuous wavelet transform, and uses a double-threshold method to enhance the detection performance.

3.2 Preprocessing: Robust Estimation of Noise

The proposed peak detection algorithm assumes that noise is a stationary additive white Gaussian noise (AWGN) with a known variance. It's proper to assume the noise is stationary when the processing is carried out on a short interval (several seconds). However, variance is not a known priori in real applications. The estimation of the noise variance in a raw EMG signal is in particular difficult because the noise and the MEPs are mixed together. In most applications, noise variance is calculated in a supervised fashion, in which an operator selects a segment of signal-free data and calculates its variance. This method has two major drawbacks. First of all, it relies on human supervision, and can't run automatically. In addition, visual inspection is not reliable and consistent. Secondly, this said operation is normally only applied once at the beginning of the processing, because doing it repeatedly on every short intervals is impractical. As a result, this said manual operation does not work when the noise is non-stationary over a long time period, which is typical in most real systems.

In [39], noise variance is calculated based on the theory of robust estimation, assuming the signal samples are outliers. Let $\mathbf{X} = \{x_i\}_{i=1}^{N}$ be a sequence of N independent identically distributed Gaussian random variables with variance σ^2. Then from [39], it follows that

$$\hat{\sigma} = M\{|x_1 - \bar{\mathbf{X}}|, |x_2 - \bar{\mathbf{X}}|, \cdots, |x_N - \bar{\mathbf{X}}|\}/0.6745 \tag{3.2.0.1}$$

where $M\{\mathbf{S}\}$ is the sample median of the sequence $\mathbf{S}$ and $\bar{\mathbf{X}} \overset{\text{def}}{=} 1/N \sum_{i=1}^{N} x_i$. From robust statistics, the median of a random variable is less sensitive to outliers than its variance. As a result, indirect estimation of variance from the median is much more accurate than directly computing the variance of the sequence containing outliers, especially when outliers, in the case of signal samples, have much larger amplitudes than noise.

However, the above estimation is only useful when the outliers (signal samples in the case of noise estimation) are "sparse": the outliers only occupy a small portion of the observed data. Otherwise, the estimation accuracy deteriorates rapidly, as shown by the simulation results in Figure 3.2-1. *sparsity* is defined as the fraction of signal-free samples (noise-only samples) over the total number of samples. The signal model is one period of sine wave. The exact signal model has little effect on the results, because the sparsity is the dominating factor. The sine wave is chosen because it contains the typical characteristics of MEPs (having transient peaks). Below is a summary of how the simulation experiment is carried out to generate the plots in Figure 3.2-1.

1. Construct signal: one period of sine wave (with length w and amplitude A) (e.g., $w = 20, A = 100$)

2. Choose a sparsity value s and a SNR value.

3. Calculate the variance σ^2 of the noise from Eq. (3.4.1.4) based on the chosen SNR value and the signal model.

4. Suppose the simulated data contains N signals. Then generate $N * [w/(1-s)]$ samples of WGN with σ^2. (e.g., $N = 200$ to yield accurate results)

5. Divide the data into N chunks, each contains $[w/(1-s)]$ samples. In each chunk, randomly choose the location of the signal (the signal needs to be completely contained in each chunk to avoid overlapping). Add signal samples to the WGN at selected locations.

6. Estimate the variance of the noise from Eq. (3.2.0.1), denoted as $\hat{\sigma}^2$.

7. Calculate the error: $e = \|(\hat{\sigma} - \sigma)/\sigma\|$.

Perform above procedure for multiple values of SNR ($10 \sim 100$) and sparsity ($50\% \sim 98\%$) and plots the Error vs. Sparsity curves from different SNR values (each curve coded by a unique color). The point average curve is also plotted as a black dashed line. Figure 3.2-1 shows that data with different SNRs have similar errors when estimating the variance of the noise using Eq. (3.2.0.1). Due to the relatively large-amplitudes of the signals, variance is always over estimated. The error is small for a large sparsity, but increases rapidly when the sparsity drops. Sparsity needs be above 90% in order to have an error below 10%. In a practical EMG signal, the sparsity of the MEPs is at most 80%, which means the error is above 25%.

To solve the problem in estimating variance using Eq. (3.2.0.1) when signals are not sparse, an iterative estimation algorithm is proposed, as shown in Algorithm 1. The idea behind the proposed algorithm is to iteratively obtain a better estimation of signal-free (noise-only) samples $\mathbf{W}$ from given samples $\mathbf{X}$. Signal-free samples are estimated assuming that signal samples mostly have larger amplitudes than 3σ. At the same time, assuming noise is Gaussian, then 99.73% of the noise samples are preserved during the step $\mathbf{W}^* \leftarrow \{x \in \mathbf{X} : (x - \hat{\mu}) < \hat{\sigma} * 3\}$. In the while loop, if the initial estimation of signal-free samples $\mathbf{W}$ contains some signal samples, then the estimated variance is larger than the true value. As a result, the estimated signal-free samples will contain almost all of the signal-free samples while having less signal samples than before. Eventually, when the signal samples become sparse enough, the robust estimation given by Eq. (3.2.0.1) is very close to the true value, and $\mathbf{W}$ will contain mostly just signal-free samples. When $\mathbf{W}$ is purely Gaussian noise, then after an update to get $\mathbf{W}^*$, the condition $|\mathbf{W}| - |\mathbf{W}^*| \leq 0.0027\,|\mathbf{W}|$ will hold, and thus the loop is

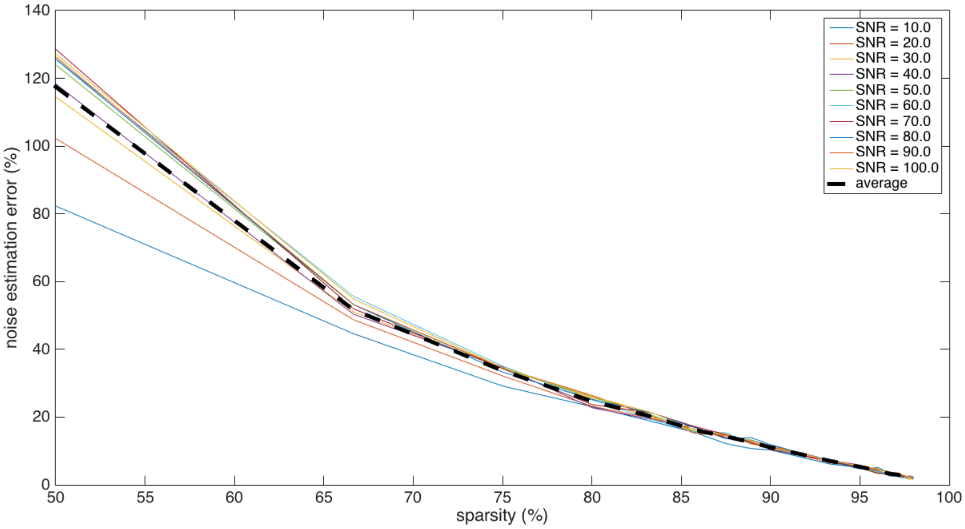


Figure 3.2-1: Simulation results on noise estimation using Eq. (3.2.0.1) based on robust statistics: Error vs. Sparsity curves from different SNR values (each curve coded by a unique color). The point average curve is also plotted as black dashed line.

```
Data: maxIter, $\mathbf{X} = [x_1, x_2, \cdots, x_N]$ that is a sum of the transient signal and the Gaussian
      noise with variance $\sigma^2$
Result: Estimated noise standard deviation $\hat{\sigma}$
/* Initialization                                                              */
$\mathbf{W} \leftarrow \mathbf{X}$ ;                     // Noise $\mathbf{W}$ is initially set to be entire input.
converge ← false;
iter ← 1;
while not converge or iter ≤ maxIter do
    iter ← iter + 1;
    $\hat{\mu} \leftarrow \bar{\mathbf{W}}$ ;                                       // $\bar{\mathbf{W}}$ is the sample mean
    $\hat{\sigma} \leftarrow$ result from Eq. (3.2.0.1);
    $\mathbf{W}^* \leftarrow \{x \in \mathbf{X} : (x - \hat{\mu}) < \hat{\sigma} * 3\}$ ; // Update noise samples from current estimated
    stats
    if $|\mathbf{W}| - |\mathbf{W}^*| \leq 0.0027\,|\mathbf{W}|$ then
        converge ← true;
    else
        $\mathbf{W} \leftarrow \mathbf{W}^*$;
    end
end
```

Algorithm 1: Iterative Robust Noise Estimation

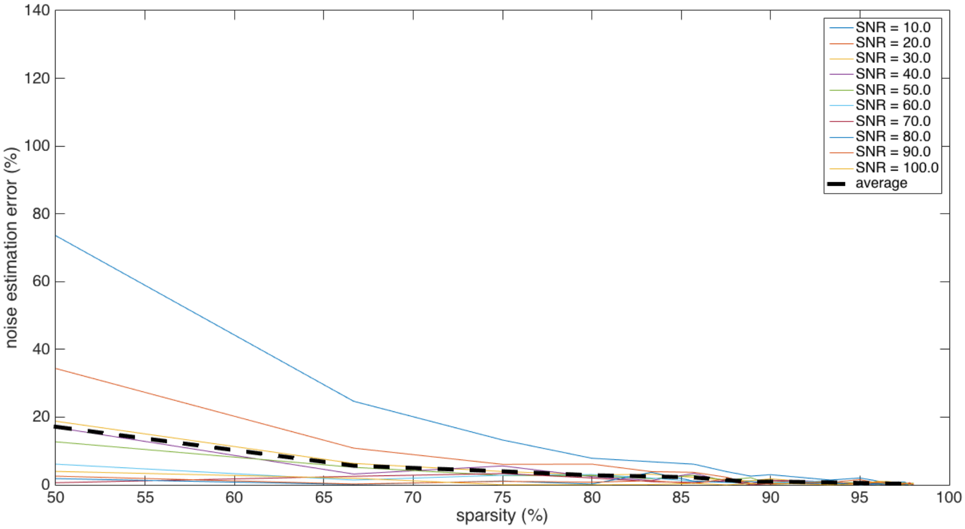


Figure 3.2-2: Simulation results on noise estimation using the iterative algorithm as in Algorithm 1: Error vs. Sparsity curves from different SNR values (each curve coded by a unique color). The point average curve is also plotted as black dashed line.

terminated as $\mathbf{W}$ converges to the signal-free samples in the original samples $\mathbf{X}$.

Figure 3.2-2 and Figure 3.2-3 show the results when using the proposed algorithm to estimate the noise variance. The conditions are the same as in Figure 3.2-1. Comparing results in Figure 3.2-1 and Figure 3.2-2, the iterative algorithm greatly boosts the estimation accuracy. For example, at the sparsity of 80%, the average error is only 2%, reduced from 25% as in simple one-pass estimation.

The potential drawback of using the iterative method is the increase in the computation time. The cost of the computation depends on the number of iterations it takes for convergence. Figure 3.2-3 shows the number of iterations it took to converge when running the simulation to get Figure 3.2-2. It shows that it took on average $2 \sim 5$ iterations, and the maximum of all is 7 iterations. Since each iteration takes linear time $O(n)$ to compute, the computation cost of the iterative algorithm is again linear $O(n)$ with a slightly larger constant.

3.3 Peak Detection Via Wavelets

It's assumed that the noise is white Gaussian noise (WGN) with known, constant variance (assume noise is stationary). In the EMG signal, the noise is not stationary and the statistics of the noise is not well known. However, the noise can be approximated as a stationary process in a given short interval (noise is locally stationary). Within the given short interval, the variance of the noise can

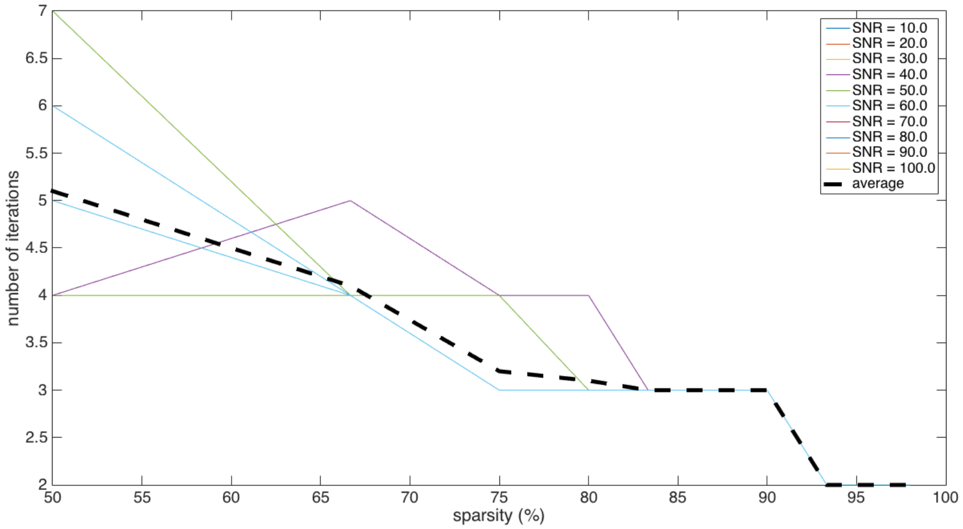


Figure 3.2-3: Simulation results on noise estimation using the iterative algorithm as in Algorithm 1: # of Iterations vs. Sparsity curves from different SNR values (each curve coded by a unique color). The point average curve is also plotted as black dashed line.

be assumed to be constant, and can be estimated in a preprocessing step, before peak detection.

As discussed in Chapter 1 and Chapter 2, the goal is to find all the peaks of Motor Evoked Potentials (MEPs). MEPs are the actual electric potential generated by muscle cells. The EMG signal includes the useful MEPs as well as noise. Throughout this chapter, EMG peaks mean MEP peaks. It should be clear to the readers since peaks of the noise are clearly not of interest.

Before diving into the algorithm, let's first define the terminology to be used through out the discussion.

Let $EMG(t)$, $t \in [t_1, t_2]$ denote the EMG signal voltage over a recording interval. This recorded signal is assumed to contain M MEPs:

$$EMG(t) = \sum_{i=1}^{M} MEP_i(t - \delta_i) + w_\sigma(t) \tag{3.3.0.1}$$

where $MEP_i(t)$, defined over $[0, L_i]$, and δ_i are the waveform and the onset of the i-th MEP, respectively. $w_\sigma(t)$ is the background noise.

Every $MEP_i(t)$ contains multiple peaks. Let $p_{i,j}$ denote the location of the j-th peak of $MEP_i(t)$, for $j = 1, 2, \cdots, N_i$, where N_i is the number of peaks from $MEP_i(t)$. N_i typically takes values from 2 to 5. All the peaks from all the MEPs in a given EMG signal are collectively denoted as:

$$\mathcal{P} = \{p_{i,j} + \delta_i, \qquad j = 1, 2, \cdots, N_i; \quad i = 1, 2, \cdots, M\} \tag{3.3.0.2}$$

The total number of peaks in the given recording $EMG(t)$ is:

$$N \overset{\text{def}}{=} |\mathcal{P}| = \sum_{i=1}^{M} N_i \tag{3.3.0.3}$$

The goal is to find $\mathcal{P}$ from a given EMG signal $EMG(t)$.

Throughout the thesis, signal-to-noise ratio (SNR) is used to discuss the noise level. Generally speaking, SNR is defined as the ratio of the power of the signal to the power of the noise. There is no universal definition of SNR. The definition of SNR is subject to the nature of the signal and the application domain. In this thesis, SNR is defined as the power of one MEP waveform to the power of the local noise. Then the SNR for $MEP_i(t)$ is:

$$SNR \overset{\text{def}}{=} \frac{1/L_i \int_0^{L_i} (MEP_i(t))^2}{\sigma^2} \tag{3.3.0.4}$$

where σ^2 is the variance of the background noise in the neighborhood of $MEP_i(t)$. This definition is suitable for transient signals because it measures the local signal strength or noise level. When there are no MEPs in the signal, SNR is simply 0. When there is a MEP, over the short period of the MEP and its neighborhood, noise can be assumed to be stationary, and so it's valid to simply use σ^2 to denote the power of noise. From this definition, SNR is only comparable between two MEPs of the same shape. This implies that for one MEP, a larger SNR will always yield better detection performance. However, for two MEPs of different shapes, a larger SNR doesn't necessarily yield better detection performance.

The goal is to locate the peaks of the transient EMG signal (more specifically, the transient MEPs) corrupted by white Gaussian noise with known statistics. There are two particular challenges as discussed in great detail in the Section 2.2. The first challenge is that the peaks of the EMG signal have different shapes: some peaks are wide while some are narrow; the second challenge is that some peaks of the EMG signal have very low SNR, making the signal barely recognized by visual inspection. To tackle these two challenges, the algorithm is mainly composed of two steps, which are broadly summarized below.

Step 1

To tackle the challenge of various shapes of the peaks, wavelet transform is employed to perform multi-resolution decomposition of the EMG signal. The idea is that a peak of a particular shape will result in large wavelet coefficients at one or more scales. Therefore, a binary hypothesis testing is performed at each scale, where the test statistic is derived from the local maximum of the wavelet coefficients. Then, the local maximum that are above certain threshold are selected as "peak candidates". This is better than a simple digital filtering, since the continuous wavelet transform is essentially a filter bank in the sense that one wavelet at each single scale corresponds to one band-pass filter [61]. So, wavelet transform at each scale is more selective to peaks of certain frequency than a general digital filter with a wide passband.

Step 2

To tackle the challenge of low SNR, peak candidates from different scales are combined. In particular, ridges in the time-frequency space are identified by linking the peak candidates at each scale. The idea is that a real EMG peak results in a relatively long ridge, while a peak from noise doesn't. Therefore, the threshold defined in Step 1 can be reduced in order to increase the recall. However, lower threshold also decreases precision, especially in a low SNR signal. To increase the precision while not hurting recall too much, ridges with short length are considered as from the noise.

The peak detection problem is formulated as a general peak detection problem regardless of the EMG signal for two reasons. First of all, a precise model of the EMG signal is not available. Secondly, it's desirable to develop a general peak-detection algorithm that is applicable to other application areas. In particular, this algorithm works well on the actual EMG signal recorded from spinal cord research subjects. Here is the formal definition of the problem to be solved.

Suppose a given data recoding $x(t), t \in [t_1, t_2]$ contains N peak signals:

$$x(t) = \sum_{i=1}^{N} A_i(t - p_i) + w_\sigma(t) \tag{3.3.0.5}$$

where $A_i(t)$ is the i-th peak signal containing one peak at 0, and p_i is the location of the i-th peak in $x(t)$, and $w_\sigma(t)$ the background noise, which is assumed as stationary zero-mean WGN with known variance σ^2.

The goal is to estimate all the peak locations: $\mathcal{P} \overset{\text{def}}{=} \{p_i, \quad i = 1, 2, \cdots, N\}$. The number of peaks N and the locations p_i are not a priori known. The peak signal $A_i(t)$ is assumed to have only one local maximum, which is the peak. The exact shapes of the peak signals $A_i(t)$ are unknown. But the bandwidth of the peak signals is assumed to be prior knowledge from the nature of the signals.

The proposed methodology consists of a combination of several techniques stemming from multi-resolution wavelet decomposition, statistics, detection theory. For convenience, the five major steps of the algorithm are briefly stated up-front. Each step will be explained in detail in subsequent sections.

1. Perform multi-scale decomposition of the signal using an appropriate wavelet basis with ap-

propriate scales.

2. Perform binary hypothesis testing at every scale to find peak candidates.
3. Identify “ridges” in the time-frequency space by linking peak candidates detected in Step 2.
4. Remove false positive peaks by removing wavelet extrema ridges with short lengths.
5. Map wavelet extrema ridges to peaks: each ridge represents one peak.

3.3.1 Mother Wavelet For Peak Detection

Wavelet transformation (WT) methods can be categorized as the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT). In CWT, both scales and translations are continuous while the DWT uses dyadic scales and translations. The DWT is often used in data compression applications due to its efficiency in exact reconstruction of data from wavelet coefficients. On the contrary, the CWT is a very redundant representation of the data. Redundancy is not good for storage efficiency, but redundancy often leads to accuracy if it’s used properly. The redundancy of the CWT makes the features of a real peak more visible. Therefore, the CWT is widely used in pattern matching. The proposed methodology makes use of the pattern matching ability of the CWT in the peak detection process.

More specifically, from the definition of CWT (Eq. (2.4.2.1)), the wavelet coefficients are found from the inner product between the signal $x(t)$ and the wavelets $\psi_{s,u}(t)$. Therefore, the wavelet coefficients indicate the resemblance between the signal at translation index u and wavelet at scale s. A larger coefficient magnitude means better pattern matching. For peak detection, the goal is to match the different shapes (heights and widths) of the peaks by scaling the mother wavelet $\psi(t)$ at different scales. Therefore, the mother wavelet should preferably have the basic features of a peak. In this work, the Mexican hat wavelet was selected as the mother wavelet due to the resemblance of its shape to a peak.

The Mexican hat wavelet is the Ricker wavelet in mathematics and numerical analysis, and defined as:

$$\psi(t) = \frac{1}{\sqrt{2\pi}} \left(1 - t^2\right) e^{\frac{-t^2}{2}} \tag{3.3.1.1}$$

It is the negative normalized second derivative of a Gaussian function. It is usually referred to as the *Maxican hat wavelet* because its shape looks like that of a sombrero (See Figure 3.3-1).

As pointed out by Du’s [13], the CWT with the choice of Mexican Hat wavelets naturally removes the baseline from the raw data, so that baseline removal in the raw data in the preprocessing steps is not necessary. In fact, any symmetric mother wavelet will have this benefit. The following

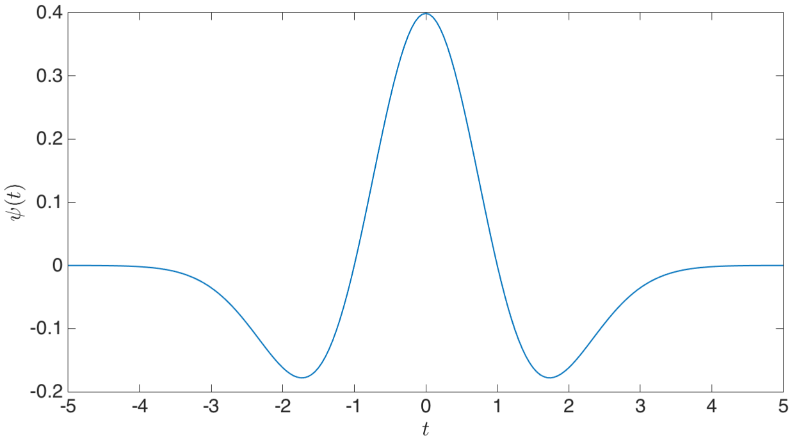


Figure 3.3-1: Mexican Hat Wavelet $\psi(t)$ given by Eq. 3.3.1.1

derivation follows that in [13].

Suppose each peak in the raw data, $P_{raw}(t)$, can be represented as follows:

$$P_{raw}(t) = P(t) + B(t) + C, \qquad t \in [t_1, t_2] \tag{3.3.1.2}$$

where $P(t)$ is the signal peak component, $B(t)$ is the baseline function with zero mean, C is a constant and $[t_1, t_2]$ is the support region of the signal peak. Since the wavelet support is small compared to the time scale of the raw data, it can be assumed that the baseline is slowly changing and monotonic in the wavelet support region. As a result, the baseline can be decomposed into an odd function and a constant.

Based on Eq. (3.3.1.2), the CWT coefficients of the peak can be calculated:

$$X(s,u) = \int_{-\infty}^{+\infty} P(t)\psi_{s,u}(t)dt + \int_{-\infty}^{+\infty} B(t)\psi_{s,u}(t)dt + \int_{-\infty}^{+\infty} C\psi_{s,u}(t)dt \tag{3.3.1.3}$$

where $\psi_{s,u}(t)$ is the scaled and translated wavelet function. Note that in Eq. (3.3.1.3) the wavelet is not taken as the complex conjugate, because the Mexican Hat wavelet is a real-valued function.

Because the wavelet function $\psi_{s,u}(t)$ has a zero mean, the third term in Eq. (3.3.1.3) will be zero. For symmetric wavelet functions, like the Mexican Hat wavelet, the second term will also be zero. Thus, only the term with real peak $P(t)$ is left in Eq. (3.3.1.3). That is to say, as long as the baseline is slowly changing and locally monotonic in the wavelet support region, it will be automatically removed in the CWT coefficients.

There are other reasons to choose the Mexican Hat wavelet. When there is a peak-shape function, its 2nd order derivative has maximum at the peak location. Since the Mexican Hat wavelet function is proportional to the 2nd-order derivative of a Gaussian function, the CWT coefficient of a function $f(t)$ is proportional to the 2nd-order derivative of $f(t)$ smoothed by a Gaussian function [32].

A Gaussian function is a smoothing function, since it is the impulse response of a low-pass filter. The convolution of a function $x(t)$ with a Gaussian function attenuates part of its high frequencies without modifying the lowest frequencies and hence smooths $x(t)$. As a result, the CWT with the Mexican Hat wavelet is essentially a low-pass filtering of the signal. Let $\theta(t)$ be a Gaussian-like function (Gaussian function multiplied by some constant), and then Mexican Hat mother wavelet can be written as:

$$\psi(t) = \frac{d^2\theta(t)}{dt^2}$$

Let $\theta_s(t) \overset{\text{def}}{=} (1/\sqrt{s})\theta(t/s)$ denote the dilation of $\theta(t)$ by a scale factor s. Then from Eq. (2.4.2.2):

$$\begin{aligned}\frac{d^2\theta_s(t)}{dt^2} &= \frac{d^2(\frac{1}{\sqrt{s}}\theta(\frac{t}{s}))}{dt^2} \\ &= \frac{1}{\sqrt{s}}\frac{1}{s^2}\psi(\frac{t}{s}) \\ &= \frac{1}{s^2}\psi_s(t)\end{aligned}$$

The last step makes use of the Eq. (2.4.2.2). From Eq. (2.4.2.4), one can derive that:

$$X(s,u) = (x * \psi_s)(u) = (x * s^2\frac{d^2\theta_s(t)}{dt^2})(u) = s^2\frac{d^2}{du^2}(x * \theta_s)(u) \tag{3.3.1.4}$$

Hence, $X(s,u)$ is proportional to the 2nd-order derivative of $x(t)$ smoothed by $\theta_s(t)$. The 2nd-order derivative of a peak-like signal normally also reaches its maximum near (if not at) the peak location. So, by searching for local maxima in the wavelet coefficients at one proper scale, one can estimate the locations of the peaks in the time domain.

In summary, the choice of the Mexican Hat wavelet as the mother wavelet gives following benefits:

- Its peak shape gives excellent pattern-matching results for peak detection.
- It naturally removes baseline from the raw signal.
- It naturally smooths the raw signal, removing high frequency noise.

3.3.2 Choice of Scales

In the continuous wavelet transform, scale s is a positive real number, which takes continuous values. In practice, a set of discrete scales have to be chosen. In the following, scales will be chosen from both the time-domain and the frequency-domain point of views.

The intuition behind the proposed methodology is "template matching". From the time-domain point of view, if a wavelet at certain scale, s, $\psi_{s,u}(t)$ has "similar shape" to a peak in the signal at certain location u, then the wavelet coefficient at scale s and translation u will be significantly different from zero. This is because a wavelet coefficient is the inner product of the signal and the wavelet, and it measures the "resemblance" between the signal at location (or translation) u and the wavelet at scale s. In the case of peak detection using the Mexican Hat wavelet, the shape can be characterized by the curve (or slope, roughly speaking) on both sides of the peak. Therefore, the scales need to be chosen, such that wavelets at selected scales have similar shapes as the peaks of interest.

From the frequency point of view, each wavelet is essentially a band-pass filter. Wavelets with different scales have different passbands and bandwidths. The wavelet transform at different scales is similar to a filter bank. So, if a signal at location u has energy within the passband of a wavelet with scale s, then the wavelet coefficient at scale s and translation u will be significantly different from zero. Therefore, the scales need to be chosen, such that the passbands of the wavelets at selected scales cover the frequency components of the signal of interest.

The two points of view are essentially equivalent. Although this method is inspired by the idea of "template matching", it's hard to quantify the "shape" information. On the other hand, frequency is a well-defined concept. Therefore, the scales will be determined from the frequency point of view.

Let's denote the set of scales (in ascending order)to be:

$$\mathcal{S} \overset{\text{def}}{=} \{s_0, s_1, \ldots, s_j, \ldots, s_J\}, \qquad s_i < s_j \qquad \text{for} \quad i < j \tag{3.3.2.1}$$

Suppose the peaks of interest have frequency components within the range $[\omega_L, \omega_U]$, then the corresponding scale limits s_0 and s_J can be calculated from Eq. (2.4.4.2):

$$\omega_U = C_\psi(s_0) = \frac{1}{s_0} C_\psi$$
$$\omega_L = C_\psi(s_J) = \frac{1}{s_J} C_\psi$$

Solving for s_0 and s_J yields:

$$s_0 = \frac{C_\psi}{\omega_U} \tag{3.3.2.2a}$$
$$s_J = \frac{C_\psi}{\omega_L} \tag{3.3.2.2b}$$

Now the task is to find a useful set of distribution of the intermediate values between s_0 and s_J. Let me first introduce the concept of "*effective coverage*" of a wavelet on the frequency domain.

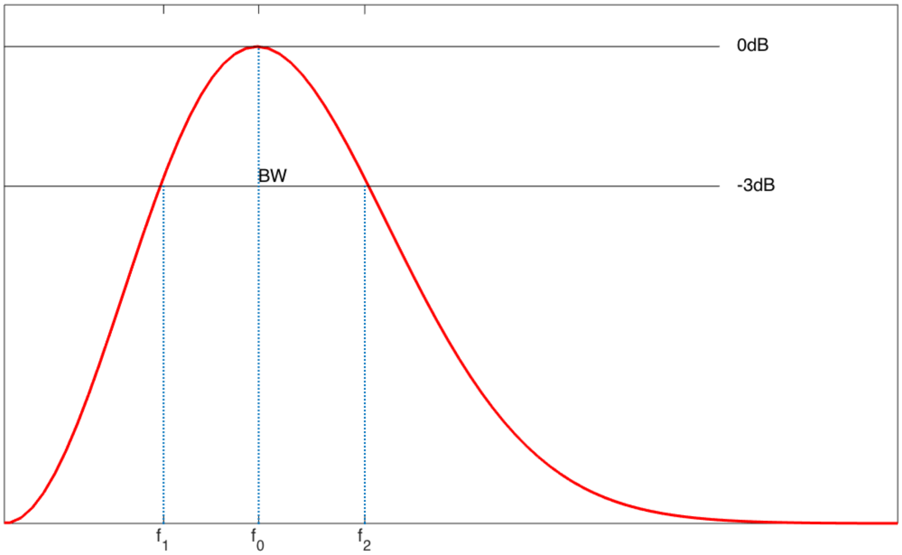


Figure 3.3-2: Illustration of bandwidth defined on the Mexican hat mother wavelet. The red curve is the Fourier transform of the Mexican hat mother wavelet. f_0 is the center frequency. f_1 and f_2 are the cutoff frequencies. BW is the bandwidth.

The coverage of a wavelet at scale s is its Fourier Transform. Although its Fourier Transform is non-vanishing over the entire frequency domain, it's only significantly different from zero over a short interval. There are two extreme cases. If the coverage of the wavelet is considered to be the entire frequency domain (coverage is ∞), then only one scale value is needed. This case will then degenerate to simple band-pass filtering. If the coverage of the wavelet is considered to be its peak location (coverage is 0), then scales need to take all positive real values to cover the entire frequency domain. This reverts to the continuous wavelet transform. In practice, one seeks a positive, efficient coverage for wavelet.

As shown in Figure 3.3-2, the bandwidth $BW = f_2 - f_1$ is the range where the Fourier Transform is above the half-energy point (-3dB point). So, bandwidth can be used to describe the "effective coverage" of the wavelet. However, to allow more flexibility, another parameter called "density" d is introduced. The meaning of d will become clear later on.

The interval centered at $C_\psi(s)$, with length of $BW_\psi(s)/d$, is defined as the "effective coverage" or "EC" of a wavelet $\psi_{s,u}(t)$, or more formally:

$$\text{EC} \overset{\text{def}}{=} \left[C_\psi(s) - \frac{1}{2} \cdot \frac{BW_\psi(s)}{d}, C_\psi(s) + \frac{1}{2} \cdot \frac{BW_\psi(s)}{d} \right], \qquad \text{where} \quad d \in \mathbb{R}^+ \quad \text{is the density} \quad (3.3.2.3)$$

where $C_\psi(s)$ and $BW_\psi(s)$ are the center frequency and the bandwidth of $\hat{\psi}_{s,u}$, respectively, and $\hat{\psi}_{s,u}$ is the Fourier transform of the Mexican hat wavelet $\psi_{s,u}$.

To realize a uniform “effective coverage” of the frequency domain, the following equation must be satisfied:

$$C_\psi(s_i) - C_\psi(s_{i+1}) = \frac{1}{2} \cdot \frac{BW_\psi(s_i)}{d} + \frac{1}{2} \cdot \frac{BW_\psi(s_{i+1})}{d} \tag{3.3.2.4}$$

Substituting Eq. (2.4.4.2) into Eq. (3.3.2.4) yields:

$$(\frac{1}{s_i} - \frac{1}{s_{i+1}})C_\psi = \frac{1}{2d} \cdot (\frac{1}{s_i} + \frac{1}{s_{i+1}})BW_\psi$$

Rearranging this equation yields:

$$\frac{s_{i+1}}{s_i} = \frac{2d \cdot C_\psi + BW_\psi}{2d \cdot C_\psi - BW_\psi} \tag{3.3.2.5}$$

$$\text{where:} \qquad 0 \le i \le J-1, \quad i \in \mathbb{Z}$$

From Eq. (3.3.2.5), scales need to be a geometric series in order to uniformly cover the frequency domain. Therefore,

$$s_n \overset{\text{def}}{=} s_0 \lambda^n, \qquad s_0 \in \mathbb{R}^+, \quad n = \{0, 1, \cdots, J\} \tag{3.3.2.6}$$

where:

$$\lambda \overset{\text{def}}{=} \frac{2d \cdot C_\psi + BW_\psi}{2d \cdot C_\psi - BW_\psi} \tag{3.3.2.7}$$

Compared with dyadic scaling, in which scales are based on the power of 2, the scales from Eq. (3.3.2.6) are based on the power of λ. I call the scaling defined in Eq. (3.3.2.6) “generalized dyadic scaling”, or “exponential scaling”, as scales grows “exponentially”. Both dyadic scaling and “exponential scaling” are powers with different choices of the base. By choosing different values of λ, “Exponential scaling” gives much more flexibility to the choice of the scales than dyadic scaling, while the “exponential” nature still keeps the benefit of “efficiency” from dyadic scaling, in the sense that the frequency interval of interest is covered uniformly by selected set of scales.

The quality factor (or Q factor) is:

$$Q \overset{\text{def}}{=} \frac{f_0}{f_2 - f_1}$$

from which, the quality factor of the mother wavelet is:

$$Q_\psi \overset{\text{def}}{=} \frac{C_\psi}{BW_\psi} \tag{3.3.2.8}$$

Eq. (3.3.2.7) can be further simplified to:

$$\lambda = \frac{2d \cdot Q_\psi + 1}{2d \cdot Q_\psi - 1} \tag{3.3.2.9}$$

From Eq. (3.3.2.6), the value of J can be found as:

$$J = \frac{\log \frac{s_J}{s_0}}{\log \lambda} \tag{3.3.2.10}$$

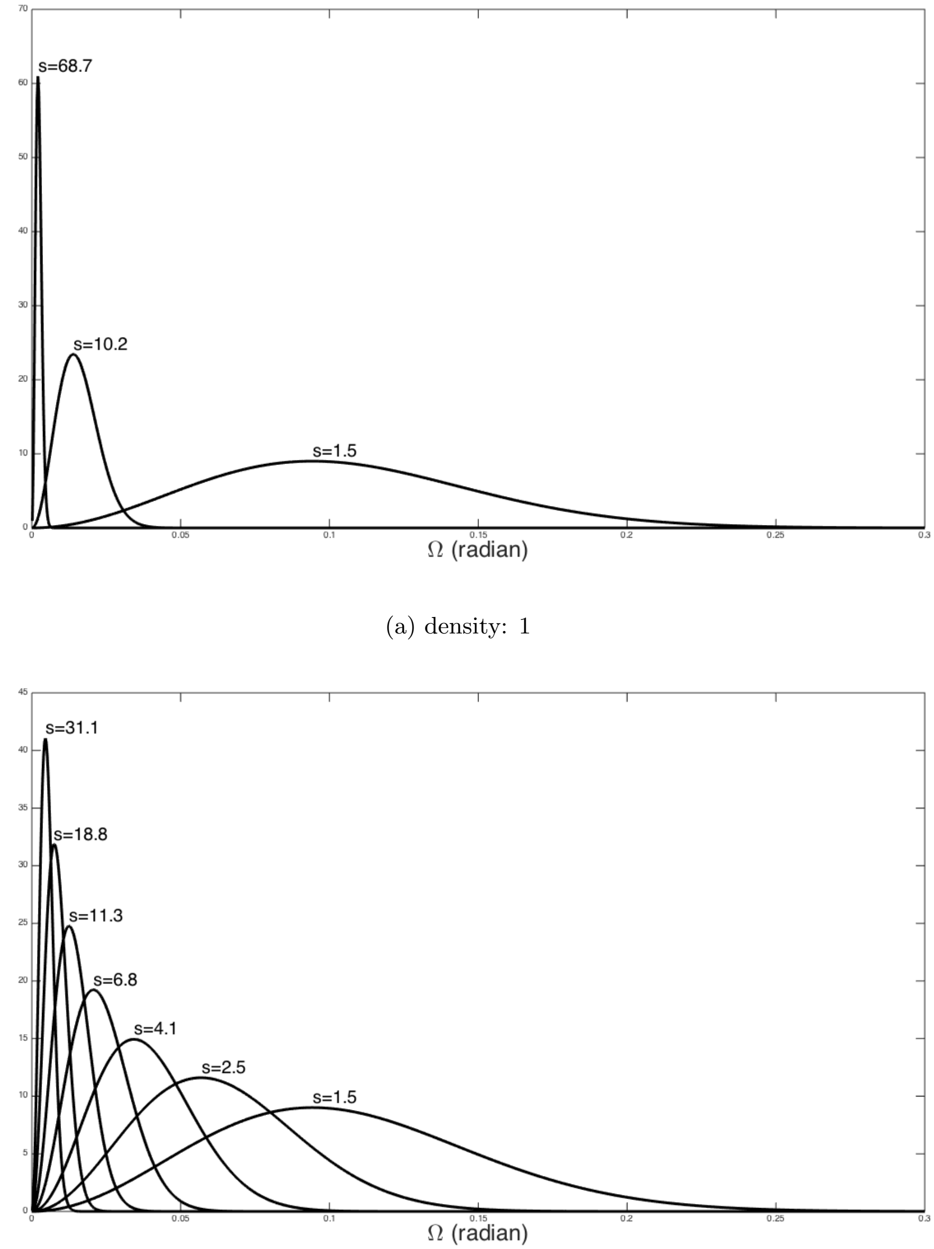
Substituting Eq. (3.3.2.2) yields:

$$J = \frac{\log \frac{\omega_U}{\omega_L}}{\log \lambda} \tag{3.3.2.11}$$

Now consider the role of the parameter d. Eq. (3.3.2.9) shows that a larger value of d translate to a smaller value of λ. Eq. (3.3.2.11) shows that a smaller value of λ gives a larger value of J. Overall, a larger value of d gives a larger value of J. Therefore, a larger d implies more band-pass filters within a fixed frequency interval $[\omega_L, \omega_U]$. That's the meaning of "density" as I define it: it's the density of band-pass filters within a given frequency interval. For example, Fig. 3.3-3 plots the Discrete-time Fourier Transform (DTFT) of the wavelets with scales calculated using two different density values. Fig. 3.3-3a shows the DTFTs of all the wavelets with scales calculated with density of 1; Fig. 3.3-3b depicts the DTFTs of all the wavelets with scales calculated with density of 3. Each wavelet at a given scale is essentially a bandpass filter as shown in the figure. From the two plots, it's clear to see that a higher density value gives a denser coverage of a given frequency interval by the wavelets.

In Du's [13], linear scaling are used instead of "exponential scaling" as defined in Eq. (3.3.2.6). To compare the difference between these two choices, the DTFTs of the wavelets for linear scaling are plotted in Fig. 3.3-4. Let's compare it with Fig. 3.3-3b. In both cases, a total of 7 wavelets covers roughly the same frequency interval. However, linear scaling provides an uneven coverage: more wavelets cover the low frequency area than high frequency area. On the contrary, exponential scaling gives a uniform coverage: the wavelets spread out more or less evenly along the frequency range of interest. Unless it is a priori known that signal peaks have a higher probability to occur in the low frequency range, it's better to cover the frequency space uniformly, yielding consistent detection performance for peaks with both high and low frequency components within the frequency range of interest.

In summary, for a user-given frequency interval $[\omega_L, \omega_U]$, and given mother wavelet $\psi(t)$ (so that one can calculate Q_ψ from Eq. (3.3.2.8)), the set of scales, $\mathcal{S}$, are calculated from Eq. (3.3.2.6), Eq. (3.3.2.9), and Eq. (3.3.2.11). One must carefully choose the parameter *density* (d) in order to optimize the peak detector's performance. In this thesis, parameter density d is chosen by running



(a) density: 1

(b) density: 3

Figure 3.3-3: Example of the effect of density on the scales: In both the two plots, DTFTs of the wavelets with scales calculated from a specific density are plotted. Since the DTFT is symmetric, only the positive part of the DTFT is shown, and only the area of interest is zoomed in. Scales are calculated from Eq. (3.3.2.11), Eq. (3.3.2.9), and Eq. (3.3.2.6)

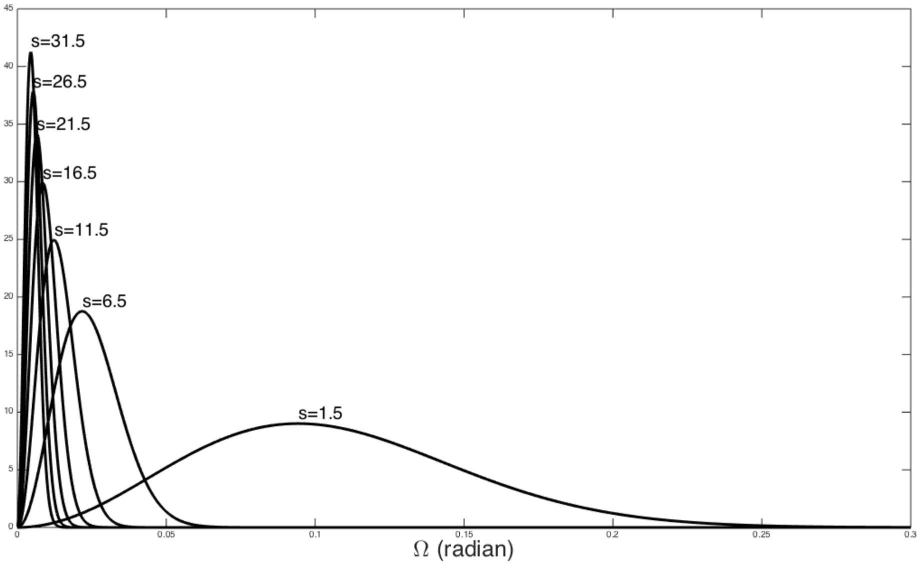


Figure 3.3-4: Example of wavelets with linear scales

experiments on simulated data with artificial peak signals. Section 3.4.3 talks about how to choose all the algorithm-related parameters from simulation in detail.

3.3.3 The Statistics of Wavelet Coefficients of Noise

At the beginning of this chapter, it is assumed the noise corrupting the signal is a zero-mean White Gaussian Noise (WGN). This implies that each noise sample, $w[n]$, follows a Gaussian distribution, and all N noise samples are mutually independent:

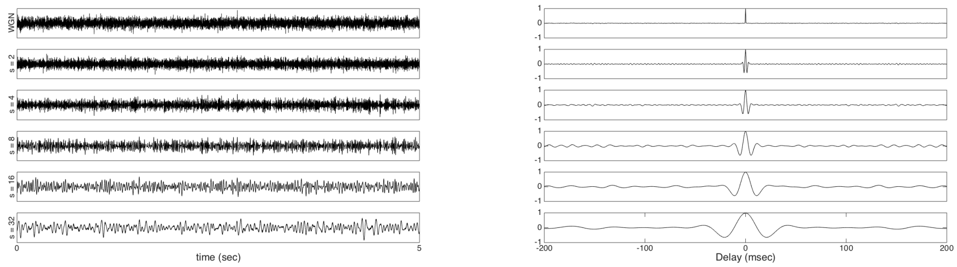
$$w[n] \sim \mathcal{N}(0,\ \sigma^2) \quad i.i.d, \quad \text{for} \quad n \in \{1, 2, \cdots, N\} \tag{3.3.3.1}$$

In Eq. (2.4.3.3), suppose the support of ψ_s has a finite time duration, denoted $[B_L, B_R]$, with $B_L, B_R \in \mathbb{Z}$. Also, assume wavelets are real-valued, such as the Mexican Hat wavelets. Then Eq. (2.4.3.3) becomes:

$$X_s[k] = \sum_{n=B_L}^{B_R} x[n+k]\psi_s[n] \tag{3.3.3.2}$$

When the signal is WGN, $w[n]$, the wavelet coefficients of WGN, $W_s[k]$, is:

$$W_s[k] \overset{\text{def}}{=} \sum_{n=B_L}^{B_R} w[n+k]\psi_s[n]$$



(a) WGN and its CWTs

(b) ACVS of WGN and CWTs

Figure 3.3-5: ACVS of the wavelet coefficients of WGN: Fig. 3.3-5a (left) shows a WGN with unit variance and its CWTs with different scales. The top one is the WGN, below which are CWTs with scales 2, 4, 8, 16, 32 from top to bottom. Fig. 3.3-5b (right) shows the normalized ACVS of the sequence on the left. The WGN is a 5 second sequence. The scales are chosen so that they are in the same range as the actual implementation. The sampling rate is 2000Hz as in the spinal cord injury application.

Because $w[n]$ are jointly independent Gaussian random variable with the zero mean and the same variance σ^2:

$$W_s[k] \sim \mathcal{N}(0, \sum_{n=B_L}^{B_R} \psi_s^2[n] \cdot \sigma^2) \tag{3.3.3.3}$$

The discrete-time version of Eq. (2.4.1.3a) is:

$$\sum_{n=B_L}^{B_R} \psi_s^2[n] = 1$$

Therefore, each wavelet coefficient of WGN is again a Gaussian random variable with zero mean and the same variance as the WGN:

$$W_s[k] \sim \mathcal{N}(0, \sigma^2) \tag{3.3.3.4}$$

Of course, the wavelet coefficients jointly at each scale s are no longer *white*; they are *colored* Gaussian noise. However, the wavelet support is relatively small (compared to the signal length), especially at smaller scales. So, the correlation between the wavelet coefficients only exists locally. For the wavelet coefficients at larger scales, WGN is still a good approximation to model the wavelet coefficients of noise. This is further verified from the distribution and the auto-covariance sequences (ACVS) of the wavelet coefficients.

From Fig. 3.3-5, the ACVS of the wavelet coefficients at different scales resemble the Dirac function, indicating that the coefficients are approximately uncorrelated. Also note that the white noise approximation is less valid at larger scales, due to a significant amount of overlap in the basis

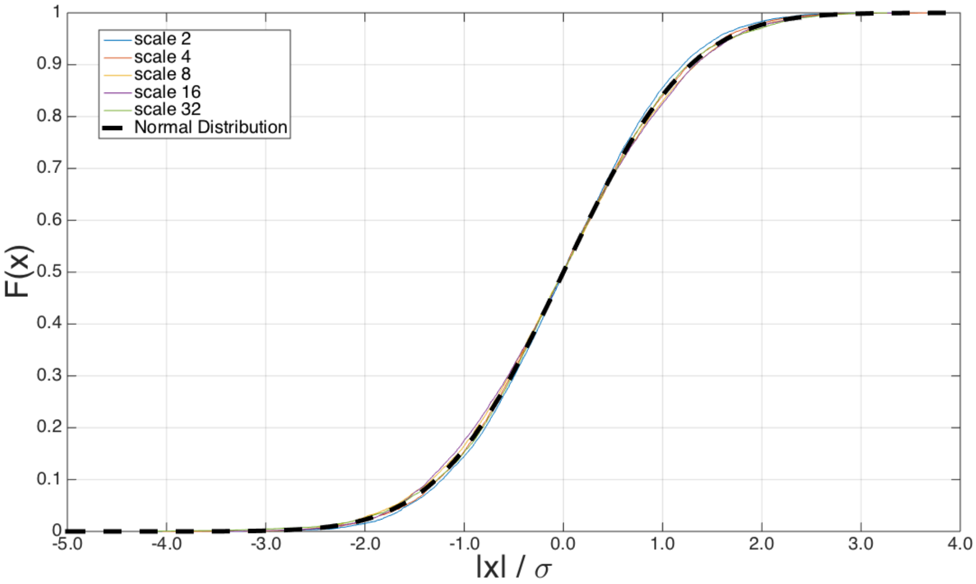


Figure 3.3-6: Empirical CDF of the CWT of WGN: black dashed line is the theoretical CDF of a Gaussian random variable. All other color lines are the empirical CDFs of the CWTs of the WGN as in Fig. 3.3-5a

functions. With the choice of scales and sampling rate, the ACVS roughly vanishes beyond 50ms for scale of 32. The data to be processed is much longer than 50ms. Hence, it's a safe to assume that the wavelet coefficients of WGN is still white.

From the empirical Cumulative Distribution Function (CDF) of the CWT of the WGN in Fig. 3.3-6, the CWT of WGN has almost the same CDF as a Gaussian distribution. This further justifies the assumption of the statistics of the CWT of WGN.

3.3.4 Detection at a Single Scale

After performing the CWT on the EMG signal, the next step is to find peak candidates from the wavelet coefficients at each single scale. For a set of discrete scales, the CWT results in a matrix of coefficients. Denote the CWT result of a discrete-time signal $\{x[n], n = 0, 1, 2, \ldots, N\}$, with choice of mother wavelet and scales from previous sections, as $\mathbf{X}$:

$$X_{j,k} \overset{\text{def}}{=} (\mathbf{X})_{j,k} = \sum_{n=-\infty}^{+\infty} x[n]\psi_{s_j,k}[n] \tag{3.3.4.1a}$$

$$= \sum_{n=-\infty}^{+\infty} x[n+k]\psi_{s_j}[n] \qquad j = 0, 1, \ldots, J, \quad k = 0, 1, \ldots, N; \tag{3.3.4.1b}$$

where $X_{j,k}$ is the element of $\mathbf{X}$ at row j, column k, and s_j is derived from Eq. (3.3.2.1). $\mathbf{X}$ is a $J+1$ by $N+1$, matrix where $J+1$ is the total number of scales, and $N+1$ is the total number of samples in the data of interest.

Because the Mexican Hat mother wavelet is used, Eq. (3.3.1.4), shows that if there is a peak signal at location t_0, then the CWT should achieve a local maxima in the neighborhood of t_0. However, due to the existence of noise in the recorded EMG signal, not all local maxima correspond to real peaks. In fact, for a data interval containing short-duration transients, most of the local maxima are likely to be noise. So, a binary hypothesis testing on the wavelet coefficients at every scale is performed to differentiate signals from noise:

$$\mathcal{H}_0 \;:\; x[n] = w[n] \qquad\qquad w[n] \sim \mathcal{N}(0, \sigma^2) \tag{3.3.4.2a}$$

$$\mathcal{H}_1 \;:\; x[n] = A + w[n] \qquad -\infty < A < \infty, \quad w[n] \sim \mathcal{N}(0, \sigma^2) \tag{3.3.4.2b}$$

where A is the local maxima of the CWT of an actual peak signal. Since the strength of the peak signal is not a priori known in the EMG peak detection problem, A is modeled as a constant with unknown magnitude, and the observation length is one sample period.

$w[n]$ is the wavelet coefficient of white Gaussian noise at n. From Section 3.3.3, it is known that the wavelet coefficient itself is a Gaussian random variable with zero-mean, and with the same standard deviation as the noise in time domain, although, collectively, the wavelet coefficients of white Gaussian noise are no longer white.

For the above binary hypothesis testing problem, no universal optimal detector exists. However, a sub-optimal detector can be formulated as a *Generalized Likelihood Ratio Test (GLRT)* [28]. The GLRT decides $\mathcal{H}_1$ if:

$$|x[n]| > \gamma \tag{3.3.4.3}$$

where the threshold γ is found from

$$P_{FA} = Q(\frac{\gamma}{\sigma}) \tag{3.3.4.4}$$

where P_{FA} is the probability of False Positive, and $Q(x)$ is the *Q-function*, which is the *complementary cumulative distribution function* of the standard normal distribution, and σ is the standard deviation of the WGN. In practice, one can choose a desired P_{FA}, then a threshold can be determined from Eq. (3.3.4.4). For example, by choosing threshold $\gamma = 3 \cdot \sigma$, one can get the theoretical P_{FA} of 0.13%.

Practically, the detection process on a single scale is slightly different from the theory above.

All of the local maxima in the wavelet coefficients are found first. Then, the local maxima that are below the selected threshold γ are removed. The remaining local maxima in the wavelet coefficients are called “peak maxima”.

Define the parameter “threshold coefficient”, c, as:

$$c \overset{\mathsf{def}}{=} \frac{\gamma}{\sigma}; \qquad c \in \mathbb{R}^{+} \tag{3.3.4.5}$$

So:

$$\gamma = c \cdot \sigma, \tag{3.3.4.6}$$

where σ is the standard deviation of WGN, γ is the threshold used in the binary hypothesis testing (Eq. (3.3.4.3)), and c is a parameter that can be adjusted to yield different detection performance.

Formally, from the CWT result $\mathbf{X}$ (Refer to Eq. (3.3.4.1)), all the peak maxima produced by the binary hypothesis testing at scale s_j are:

$$\mathcal{P}_j \overset{\mathsf{def}}{=} \{k \quad | \quad X_{j,k} > \gamma, \quad X_{j,k} > X_{j,k-1}, \quad X_{j,k} > X_{j,k+1}\}, \qquad j = 0, 1, 2, \ldots J \tag{3.3.4.7}$$

Sort the elements in $\mathcal{P}_j$ in ascending order, and denote the i-th element as $p_{j,i}$, such as:

$$p_{j,k_1} < p_{j,k_2}, \quad \text{for} \quad 1 < k_1 < k_2 < m_j \tag{3.3.4.8}$$

$p_{j,i}$ is the column index of the i-th peak maxima at row, s_j, in matrix $\mathbf{X}$, and m_j is the total number of elements in $\mathcal{P}_j$:

$$m_j \overset{\mathsf{def}}{=} |\mathcal{P}_j|, \quad j = 0, 1, 2, \ldots J \tag{3.3.4.9}$$

Define the set $\mathcal{P}$ as the collection of the binary hypothesis testing results performed at all scales:

$$\mathcal{P} \overset{\mathsf{def}}{=} \{\mathcal{P}_j \quad | \quad j = 1, 2, \cdots, J\} \tag{3.3.4.10}$$

3.3.5 Combine Peak Candidates across Scales: Identify Ridges

If a proper density d (See Section 3.3.2) is chosen, then there will be a peak maxima across multiple adjacent scale values in $\mathcal{S}$ (See Eq. (3.3.2.1)) around the occurrence of a peak signal. On the contrary, when there are no peak signals, even if a false positive (local maxima in wavelet coefficients of noise) is picked up during the binary hypothesis testing at a single scale in Section 3.3.4, it’s very unlikely that false positives are produced across multiple adjacent scales at the same neighborhood, due to the randomness of noise. In the example shown in Fig. 3.3-7, there is one real peak signal at time index 86. This figure shows that after identifying all the peak maxima, a real peak signal results in peak maxima across multiple scales. There are other peak maxima identified due to the noise (around time indices 90, 100, and 130), but they don't span multiple scales.

Therefore, if peak maxima across the scales are connected to form ridges in the time-frequency domain, then each peak in the time domain is represented as a ridge in the time-frequency domain (shown in Figure 3.3-8c). Real peak signals will have relatively long ridges, while peaks from noise will have relatively short ridges. Hence, false positives can be further eliminated by removing short ridges.

Given the peak candidates $\mathcal{P}$, all the peak candidates are examined from the largest scale to the smallest scale. The procedure of ridge identification is as follows:

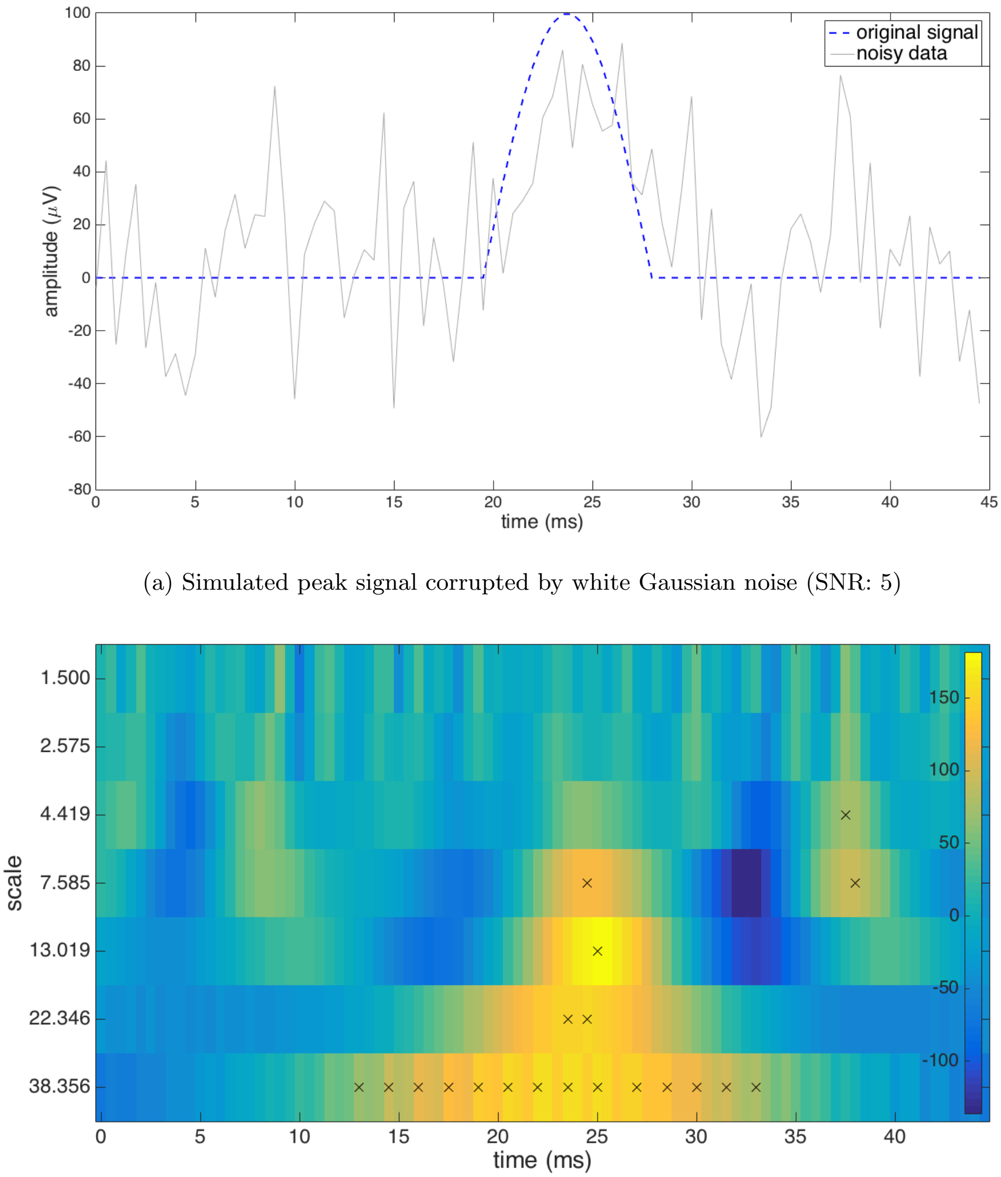
1. Starts at largest scale. Let $j = J$. Every peak maxima at the largest scale starts a new ridge.
2. Go to the next scale by updating scale index: $j = j - 1$. Go over all the peak maxima at the scale s_j. If the peak maxima is close to a peak maxima at scale s_{j+1}, it will be connected to the existing ridge containing that peak maxima at scale s_{j+1}. If it is not close to any peak maxima at scale s_{j+1}, then a new ridge is created starting at this peak maxima.
3. Repeat Step 2 until it finishes processing peak maxima at the smallest scale.

There is one parameter that needs to be chosen in order to link the peak maxima to form ridges, and this parameter is called the *time threshold*, and denoted as w. It determines if a new peak maxima is connected to an existing ridge, or is regarded as the start of a new ridge.
More specifically, in Step 2 at scale s_j:

$$\begin{aligned}
&\text{for} \quad \forall\ i \in \{1, 2, \ldots, m_j\} \\
&\quad \text{if} \quad \exists\ k \in \{1, 2, \ldots, m_{j+1}\} : |p_{j,i} - p_{j+1,k}| < w, \\
&\qquad \text{then:} \quad p_{j,i} \text{ is connected to } p_{j+1,k} \\
&\quad \text{Otherwise:} \quad p_{j,i} \text{ is regarded as the start of a new ridge.}
\end{aligned} \tag{3.3.5.1}$$

where w is the threshold for connecting two peak maxima from adjacent two scales during ridge identification.

The length of a ridge is defined as the number of peak maxima along the ridge. Define a parameter called "*ridge length threshold*", l_{thres}. Then ridges with length no larger than l_{thres} are removed to



(a) Simulated peak signal corrupted by white Gaussian noise (SNR: 5)

(b) CWT with identified peak candidates (marked by X)

Figure 3.3-7: Example of identified peak candidates for one peak signal

further remove the false positives. The parameter, ridge length threshold, is dependent on another parameter *density* as defined in Eq. (3.3.2.3). When *density* is higher, ridges tend to be longer, because there are more scales at which peak maxima could be produced. On the contrary, lower *density* yields shorter ridges on average. Therefore, the *ridge length threshold* should be positively correlated to *density.* For simplicity, assume they are linearly correlated:

$$l_{thres} \overset{\text{def}}{=} k_{ridge} * d \tag{3.3.5.2a}$$

$$\text{where} \quad k_{ridge} \in \mathbb{R}^{+} : \text{ridge length threshold coefficient} \tag{3.3.5.2b}$$

In the remaining ridges, each ridge corresponds to one peak signal. The location of the peak is estimated from the location of the peak maxima on the ridge that has the largest wavelet coefficient. The idea is that the wavelet coefficient measures the pattern matching between a real peak and a wavelet function. The location of the wavelet function which gives the best pattern matching naturally gives the best estimation of the location of the peak signal.

3.3.6 The Overall Detection Algorithm

This section first summarizes the overall peak detection algorithm by repeating some of the important steps from previous sections, and then discusses in more details the parameters associated with the algorithm.

Suppose a given data recoding $x(t), t \in [t_1, t_2]$ contains N_p peak signals:

$$x(t) = \sum_{i=1}^{N_p} A_i(t - \tau_i) + w_\sigma(t)$$

where $A_i(t)$ is the i-th peak signal containing one peak at 0, and τ_i is the location of the i-th peak in $x(t)$, and $w_\sigma(t)$ the background noise, which is assumed as stationary zero-mean WGN with known variance σ^2.

This signal is sampled at a sampling rate $f_s = 1/T_s$, where T_s is the sampling period. This results in a sequence of signal samples, the discrete-time signal, $x[k] = x(t_k)$, for $k = 0, 1, \cdots, N$, where $N = \lfloor (t_2 - t_1)/T_s \rfloor$. As a result, the locations of the peak signals in the discrete-time case become p_i, for $i = 1, 2, \cdots, N_p$, such that $p_i \in \{0, 1, 2, \cdots, N\}$.

Suppose the peak signals have frequency components within the range $[\omega_L, \omega_U]$, and a set of parameters associated with the algorithm have been chosen :

- **density,** d**:** Density of the scales as defined in Eq. (3.3.2.3).

- **threshold coefficient, c:** The coefficient for calculating the threshold used in GLRT, as defined in Eq (3.3.4.6).
- **time threshold, w:** The threshold for connecting peak maxima from two adjacent scales during ridge identification, as defined in Eq. (3.3.6.1).
- **ridge length threshold coefficient, k_{ridge}:** The coefficient for calculating the length threshold when removing short ridges, as defined in Eq. (3.3.5.2).

The overall procedure for peak detection:

1. Perform CWT on raw data, $x[k], \quad k = 0, 1, 2, \cdots, N$, with Mexican Hat mother wavelet, $\psi[n]$ and a choice of scales, $s_j, \quad j = 0, 1, 2, \cdots, J$, calculated from Eq. (3.3.2.6), (3.3.2.9), and (3.3.2.11), to obtain a matrix of wavelet coefficients $\mathbf{X}$.

   $$X_{j,k} \overset{\text{def}}{=} (\mathbf{X})_{j,k} = \sum_{n=-\infty}^{+\infty} x[n]\psi_{s_j,k}[n] \qquad j = 0, 1, \ldots, J, \quad k = 0, 1, \ldots, N;$$

   Accordingly, the wavelet coefficients at scale s_j is denoted as $X_{j,:}$, the j-th row of the matrix $\mathbf{X}$.

2. Identify peak maxima by performing GLRT on local maxima of $X_{j,:}$ for all $j = 0, 1, 2, \cdots, J$, as illustrated in Eq. (3.3.4.7) and (3.3.4.6), to obtain a set of peak maxima at scale s_j, whose column indices are denoted as:

   $$\mathcal{P}_j \overset{\text{def}}{=} \{k \quad | \quad X_{j,k} > \gamma, \quad X_{j,k} > X_{j,k-1}, \quad X_{j,k} > X_{j,k+1}\}, \qquad j = 0, 1, 2, \ldots J$$

   Sort the elements in $\mathcal{P}_j$ in ascending order, and denote the i-th element as $p_{j,i}$, such as:

   $$0 \leq p_{j,k_1} < p_{j,k_2} \leq N, \quad \text{for} \quad 1 < k_1 < k_2 < m_j$$

   $p_{j,i}$ is the column index of the i-th peak maxima at row, s_j, in matrix $\mathbf{X}$, and m_j is the total number of elements in $\mathcal{P}_j$:

   $$m_j \overset{\text{def}}{=} |\mathcal{P}_j|, \quad j = 0, 1, 2, \ldots J$$

3. Identify all the ridges by linking neighboring peak candidates across scales, as described in Section 3.3.5.
   Starting at scale s_J, initiate a new ridge for every peak maxima $p_{J,i}, \quad i = 1, 2, \cdots, m_J$.
   For $j = J - 1, J - 2, \cdots, 1, 0$, at scale s_j:

$$\begin{aligned}
&\text{for} \quad \forall\, i \in \{1, 2, \ldots, m_j\} \\
&\quad \text{if} \quad \exists\, k \in \{1, 2, \ldots, m_{j+1}\} : |p_{j,i} - p_{j+1,k}| < w, \\
&\qquad \text{then:} \quad p_{j,i} \text{ is connected to } p_{j+1,k} \\
&\quad \text{Otherwise:} \quad p_{j,i} \text{ is regarded as the start of a new ridge.}
\end{aligned} \tag{3.3.6.1}$$

The result is a set of ridges, denoted as $\{\mathcal{R}_i, \quad i = 1, 2, 3, \cdots, N_R\}$, where N_R is the total number of ridges detected. $\mathcal{R}_i$ is a set of pair of indices: $\mathcal{R}_i = \{(rx_{i,j}, ry_{i,j}), \quad j = 1, 2, \cdots, L_i\}$, where $rx_{i,j}$ and $ry_{i,j}$ give the row and column index (as in $\mathcal{X}$) of the j-th peak maxima along ridge $\mathcal{R}_i$, respectively, and $L_i = |\mathcal{R}_i|$ is the number of peak maxima along the ridge.

4. Remove ridges whose length, L_i are no larger than l_{thres}.

5. Estimate peak location from the remaining ridges.
   For each $\mathcal{R}_i$, the location of the peak signal associated with it is estimated as follows:

$$(r\hat{x}_i, r\hat{y}_i) = \underset{(x,y)\in\mathcal{R}_i}{\arg\max}\, X(x, y) \tag{3.3.6.2}$$

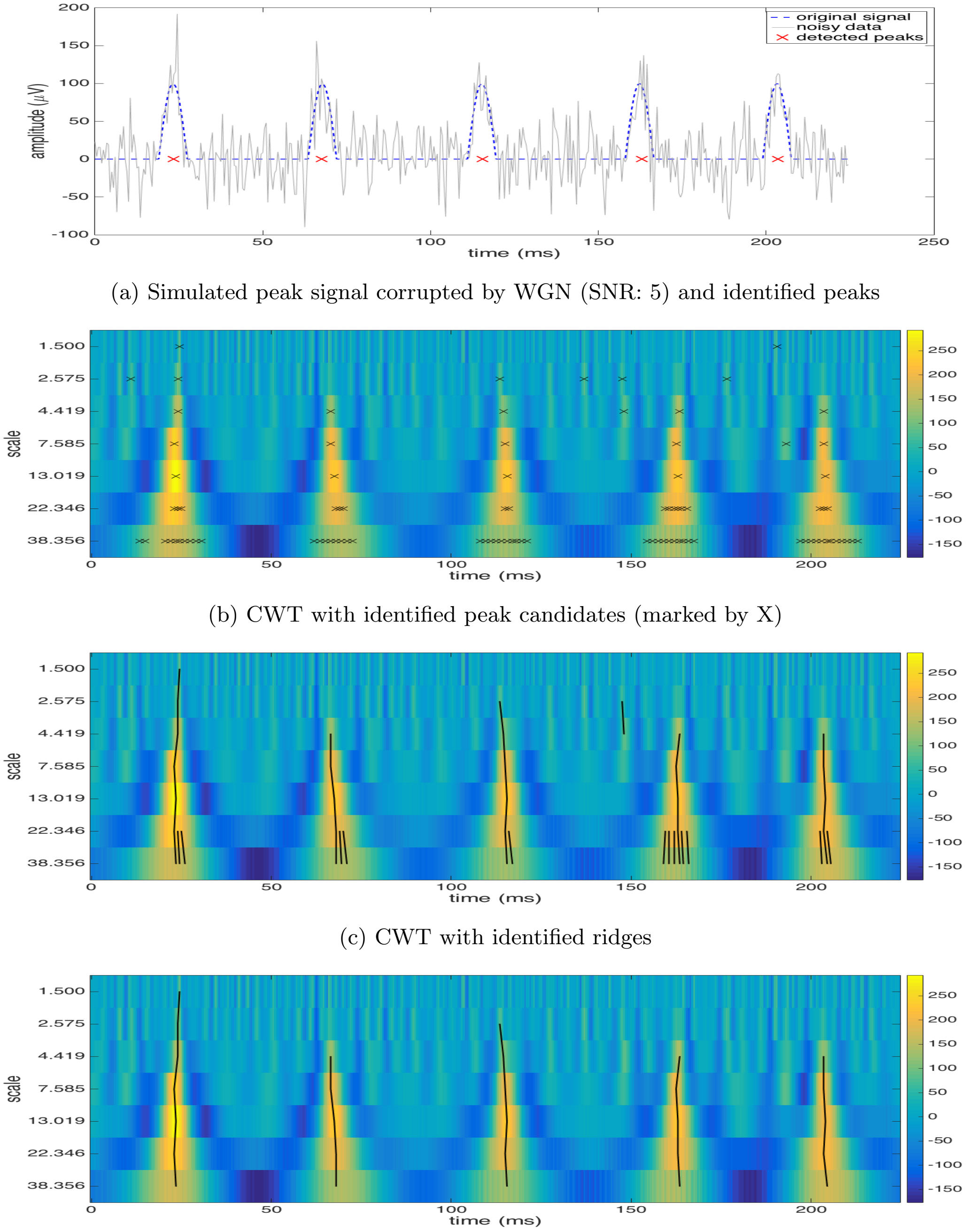
   Then:

$$\hat{p}_i = r\hat{y}_i, \qquad \text{for} \quad i = 1, 2, \cdots, N_R \tag{3.3.6.3}$$

Fig. 3.3-8 shows an example of the peak detection process.

There are totally four parameters associated with the peak detection algorithm as mentioned above. Once the four parameters are chosen, the algorithm is fully specified and can run by itself. The four parameters needs to be chosen carefully, as they can have a big effect on the detection performance. The remaining of this section discusses the roles of the parameters and their effect on the detection performance.

The *density*, d, determines the relative length of the ridges identified in the algorithm. If d is very small, then a peak signal wouldn't have large wavelet coefficients (and hence peak maxima) across multiple scales, so the ridges for peak signals are short. As a result, the length of the ridge of a true peak signal does not differ too much from that of noise. Therefore, when removing short ridges, true peak signals are removed, too, resulting in a poor detection rate. If d is very large, then the ridges from noise could potentially be long. Despite that, the biggest issue is the computation



(a) Simulated peak signal corrupted by WGN (SNR: 5) and identified peaks

(b) CWT with identified peak candidates (marked by X)

(c) CWT with identified ridges

(d) CWT with identified ridges after removing short ridges

Figure 3.3-8: Overall peak detection process based on the wavelet method

time. The computation time linearly increases as the number of scales. In summary, it's preferable to choose a *density* that is as small as possible, and yet gives very good detection rate.

The *threshold coefficient*, c, is the first threshold in what's effectively a double-threshold method, and has direct impact on the detection results on a single scale. At one scale, when c is large, although false alarm rate can be small, a weak peak signal is likely to be missed: no peak maxima produced at the occurrence of the peak signal. This results in short ridges or even no ridges for a true peak signal. When c is small, more noise components are picked up as peak maxima. This results in long ridges for noise and more computation time, as the ridge identification process is positively correlated to the number of peak candidates at every scale. For WGN, $c = 3$ theoretically removes 99.9% of noise samples as shown in Eq. (3.3.4.4). However, this also removes lots of real peak signals when signals are weak. The advantage of this double-threshold method is that a smaller c (smaller than 3) can be chosen to achieve better detection rate for weak peak signals, and later the false positives are further removed by removing short ridges.

The *time threshold*, w, has effect on the accuracy of the ridge identification process. If w is too large, then peak maxima from different peak signals are likely to be connected as one ridge. In addition, peak maxima from noise will be connected to form a long ridge. When w is too small, then even peak maxima from the same true peak signal are not connected, resulting in multiple short ridges for one true peak signal. If they are short enough to be removed in the following step, then a true real peak will be missed. This parameter is related to the choice of *density d*. When d is larger, then peak maxima from adjacent scales are closer to each other, and hence only a smaller w is needed to connect peak maxima from the same peak signal. When d is smaller, a large w is needed to ensure peak maxima from the same real peak signal are connected to form one ridge.

The *ridge length threshold coefficient*, k_{ridge}, is the second threshold in the proposed double-threshold method, and has a direct impact on the ridge-removing process. For a large k_{ridge}, less noise is falsely detected, resulting in small false alarm rates; however, less true peak signals are found, resulting in small detection rates as well. Hence, a proper k_{ridge} needs to be chosen for best overall detection performance.

Each of the four parameters has different effect on the overall detection performance in a rather complicated way, and they are correlated in a complicated way, too. Therefore, the optimal parameters are chosen from running experiments on simulated data sets.

3.3.7 Detection of Peaks and Troughs in the EMG signal

Before showing simulation results, let's first consider how to apply the peak detection algorithm to the EMG signal. The EMG signal have both peaks and troughs, as shown in Fig. 3.3-9. The goal in EMG peak detection is to detect both the peaks and the troughs, which can be done as follows.

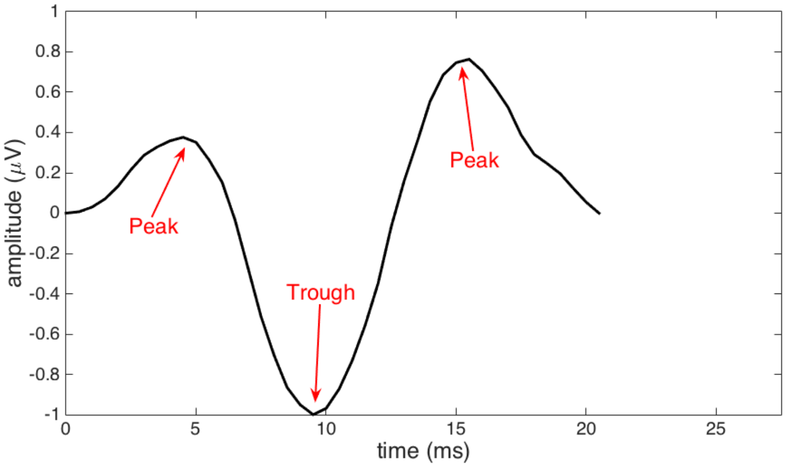


Figure 3.3-9: Peaks and Troughs of an EMG template waveform (The amplitude of the waveform is scaled so that the maximum absolute value is 1μV.)

Assuming the EMG signal is corrupted by zero-mean WGN, the follow procedure detects both the peak and the troughs of the EMG signal. An example process is shown in Fig. 3.3-10.

1. rectify the raw EMG signal $x[n]$ to obtain the positive part $x_p[n]$. See Fig. 3.3-10b.

$$x_p[n] = \begin{cases} x[n] & \text{if } x[n] \geq 0 \\ 0 & \text{if } x[n] < 0 \end{cases}$$

2. perform wavelet-based peak detection on $x_p[n]$ to obtain positive peaks $\{\hat{p}_i^+\}$.

3. obtain the negative part of the data $x_n[n]$ by flipping the raw EMG signal and rectifying the EMG signal $x[n]$. See Fig. 3.3-10c.

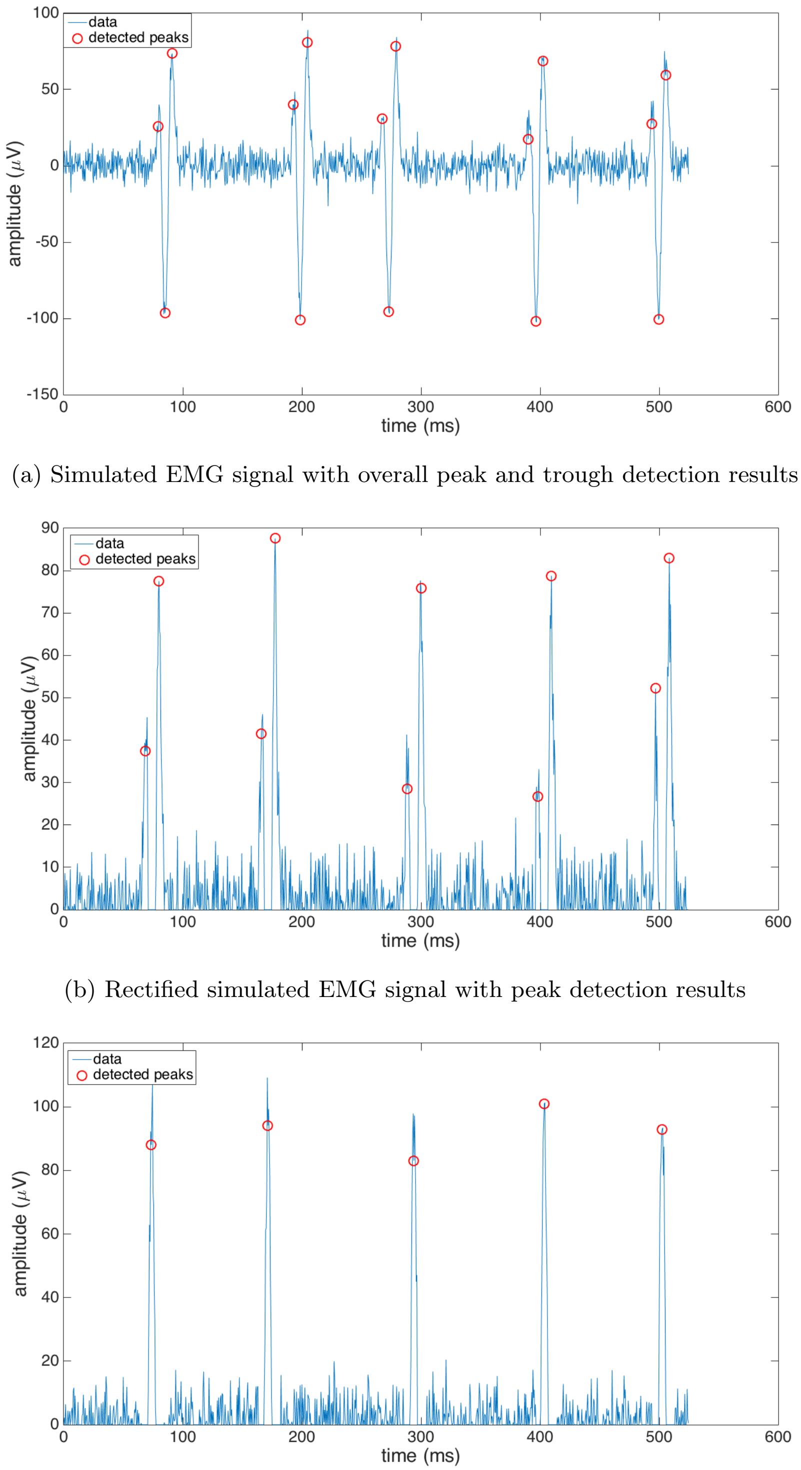
$$x_n[n] = \begin{cases} -x[n] & \text{if } x[n] \leq 0 \\ 0 & \text{if } x[n] > 0 \end{cases}$$

4. perform wavelet-based peak detection on $x_n[n]$ to obtain negative peaks, or troughs $\{\hat{p}_i^-\}$.

5. combine results from Step 2 and Step 4 to get all peaks $\{\hat{p}_i\} = \{\hat{p}_i^+\} \cup \{\hat{p}_i^-\}$. See Fig. 3.3-10a.

3.4 Simulation

Simulations are used for two goals: one, to choose optimal algorithm parameters, and two, to evaluate the peak detection algorithm on "ground truth" data, and compare it against other methods.



(a) Simulated EMG signal with overall peak and trough detection results

(b) Rectified simulated EMG signal with peak detection results

(c) Flipped and rectified simulated EMG signal with trough detection results

Figure 3.3-10: Example process of detecting peaks and troughs of the EMG signal

This section first describes the simulated data sets. Then, several different ways to evaluate the performance of a peak detection algorithm are introduced. After that, the idea behind the selection of the optimal algorithm parameters from simulations is explained. Lastly, to demonstrate the effectiveness of the proposed peak detection algorithm, other peak detection methods are introduced and compared against it.

3.4.1 Simulated Ground Truth Data

The peak detection algorithm is designed to detect transient peak signals from a noisy recording, assuming noise is known WGN. Therefore, the simulated data is the summation of a series of transient peak signals and WGN. The simulated signal $x[n]$ can be expressed as:

$$x[n] \overset{\text{def}}{=} \sum_{i=1}^{N_p} P[n - p_i] + w_\sigma[n] \tag{3.4.1.1}$$

where $P[n]$ is a transient peak signal with peak at time index 0, p_i is the location of the i-th transient signal peak, N_p is the number of transient peak signals, and $w_\sigma[n]$ is the WGN with variance of σ^2. p_i is chosen randomly and depends on the sparsity of the transient signals. *sparsity* is defined as the fraction of signal-free samples (noise-only samples) over the total number of samples. If signal $P[n]$ has length M, then every $M/(1 - \text{sparsity})$ samples of data contains exactly one transient signal on average. Hence, the total length of data $x[n]$ is $M/(1 - \text{sparsity}) \cdot N_p$.

Two kinds of simulated data are generated. The first one is a generic peak signal embedded in WGN, since the algorithm is designed to work for any generic peak signal. The second one is a simulated EMG signal, which is used to demonstrate the effectiveness of the peak detection algorithm on the EMG signal. In both cases, the same equation (Eq. (3.4.1.1)) is used, with difference choices of peak signal model $P[n]$.

3.4.1.1 Simulated Generic Peak Signals

A half-period sine wave is used as the peak signal model, so that a precise description of the frequency component of the peak signal is possible. This allows a good control over the performance of the peak detection. In addition, the half-period sine wave naturally resembles a peak. Consider the following peak signal model:

$$P_f[n] \overset{\text{def}}{=} A \cdot sin(\pi \cdot \frac{n}{L_f - 1}), \qquad n = 0, 1, 2, \cdots, L_f - 1 \tag{3.4.1.2}$$

$$L_f = \frac{f_s}{2f} + 1 \tag{3.4.1.3}$$

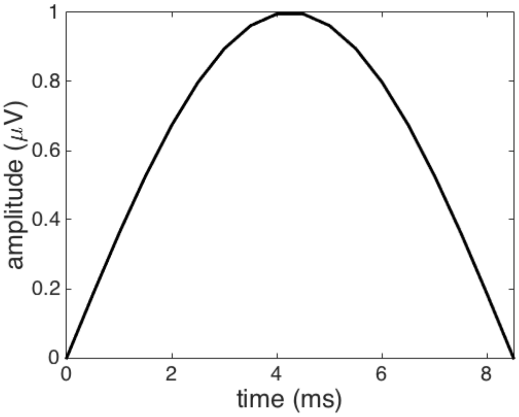
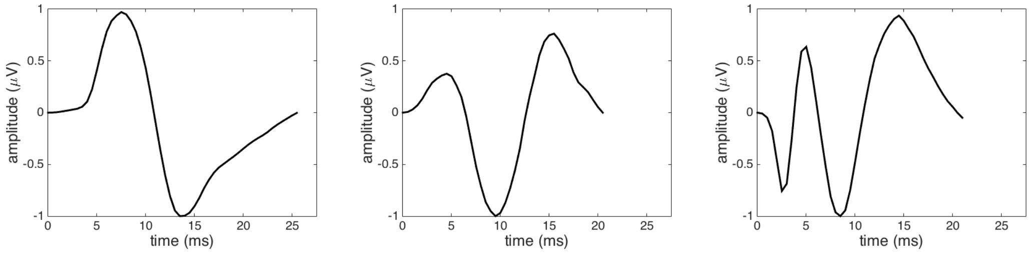


Figure 3.4-1: Template waveform of the peak signal model. (The amplitude of the waveform is scaled so that the maximum absolute value is 1μV.)

where A is the amplitude of the peak, L_f is the length of the peak signal, which can be calculated from the sampling rate f_s and the frequency of the sine wave f in Eq. (3.4.1.3). The extra 1 in Eq. (3.4.1.3) is needed since both ends of the one half-period need to be included. From the above equations, one can see that the frequency information or the sampling rate are not needed in order to obtain simulated data; all are needed is the length of the peak signal. However, in practice, the discrete data is obtained with certain sampling rate, and the frequency information of the signal is specified independently of the sampling rate. Therefore, in practice, Eq. (3.4.1.3) is used to calculate the length of the peak signal. When synthesizing peak signals, Eq. (3.4.1.1) is used by substituting $P[n]$ with $P_f[n]$ with given f. Fig. 3.4-1 shows the waveform of the peak signal model.

3.4.1.2 Synthesized EMG Signals

Real EMG signal is not good for evaluating the peak detection performance, because it's hard to determine the ground truth. Although experts can select true EMG peaks from the noisy recording based on their expertise and years of experience, manual detection is subjective, and leads to bias and inaccuracy. In particular, the proposed detection algorithm focuses on signal with such low SNR that it's almost impossible for human to detect by visual inspection. In addition, a real EMG signal suffers from unknown characteristics of noise, which can affect the detection performance significantly. When evaluating the detection performance of the peak-detection algorithm, it's preferable to single out the effect of the algorithm itself on the detection performance. Otherwise, when detection performance is bad, it's impossible to determine if the algorithm is not good or estimation of the noise is inaccurate. Therefore, synthesized EMG signal is used instead of real EMG signal. When applying peak detection algorithm to a real EMG signal with unknown characteristics of noise, the noise characteristics from real EMG signal is first estimated, and then the peak detection algorithm is carried out with the estimation result.



(a) R MH (b) L MG (c) L VL

Figure 3.4-2: MEP template waveforms used in the synthesized EMG signal. The MEP waveforms are picked from the real EMG signals of different muscles. The muscle name is given in its short form under each MEP waveform. For the complete names of the muscles from their short names, please refer to Appendix A. The amplitude of the waveform is scaled so that the maximum absolute value is 1μV.

The synthesized EMG signal uses actual EMG waveforms hand-picked from clinical EMG recordings, and therefore contain characteristics of the real EMG signal. However, the real EMG signal suffers from noise, so that the hand-picked EMG waveforms are noisy. To remove the noise, a point-average of several EMG waveforms with similar shapes is constructed. A Matlab program with graphic user interface to facilitate this process is implemented. This graphic tool is used to hand-pick several typical EMG waveforms from the real EMG signal. When synthesizing the EMG signal using Eq. (3.4.1.1), one simply substitutes $P[n]$ with a chosen EMG waveform $W_{EMG}[n]$. Fig. 3.4-2 shows three recorded EMG waveforms which will be used to synthesize the EMG signal in the subsequent experiments. These three waveforms were chosen since they differ from each other in the number of peaks, and they represent a large body of the real EMG waveforms.

3.4.1.3 Definition of Signal-to-Noise Ratio (SNR)

The peak detection algorithm gives different performance for signals with different SNRs. When comparing performance of different algorithms, a fixed SNR is maintained. Since the signal is a transient signal, the traditional definition of SNR doesn't apply since the energy or power of the signal depends on the density of the signals: energy or power increases if there are more signals within a fixed time interval. Therefore, the notion of transient average power is used in the definition of SNR:

$$SNR \overset{\text{def}}{=} \frac{P_{signal}}{P_{noise}} = \frac{\frac{1}{s_2-s_1+1} \cdot \sum_{n=s_1}^{s_2} s[n]^2}{\sigma^2} \tag{3.4.1.4}$$

where $[s_1, s_2]$ is the support of the transient signal $s[n]$, σ^2 is the variance of the WGN. For WGN, the power is the variance.

This definition gives a qualitative description of the noise level in a signal: a large SNR means a relatively low noise level. For a fixed signal shape, a higher-SNR signal will give better detection performance. However, for signals with different shapes, the detection performances under different

SNRs are non-comparable: A signal with higher SNR doesn't necessarily give better detection performance than another signal with lower SNR. Hence, this definition is used to quantitatively measure the noise level for a fixed signal model.

3.4.1.4 Procedure of Synthesizing Simulated Data

Simulated ground truth data was synthesized as follows:

1. The sampling rate, f_s, is selected to be 2000, which is the sampling rate of the clinical EMG signal of interest.
2. Choose the number of peak signals N, and sparsity. In this thesis, N is chosen to be 300, and sparsity is chosen to be 80%.
3. Choose either the generic peak signal model or the EMG signal model, and obtain peak signal model $P[n]$. For generic peak signal model, choose amplitude A and frequency f, and use Eq. (3.4.1.2) to calculate $P_f[n]$, and let $P[n]$ be $P_f[n]$. For EMG signal model, choose a waveform, and let $P[n]$ be $W_{EMG}[n]$. In this thesis, A is chosen to be 100, and f is chosen to be 60Hz because that's the major frequency component of a typical EMG signal.
4. Choose SNR and calculate σ from Eq. (3.4.1.4). Typical SNR takes values from 1 to 100.
5. Based on the sparsity and the signal length M, randomly choose τ_i from interval $[1 + (i-1) \cdot M/(1 - \text{sparsity}) + \text{buffer}, i \cdot M/(1 - \text{sparsity})]$ for $i \in \{1, 2, 3, \cdots, N\}$. *buffer* is used to make sure two transient signals don't overlap with each other.
6. Synthesize WGN $w_\sigma[n]$ with variance of σ^2 and length $M/(1 - \text{sparsity}) \cdot N$.
7. Obtain synthesized data $x[n]$ from Eq. (3.4.1.1).

3.4.2 Performance Evaluation Methods

To evaluate the performance of the peak detection algorithm, the terminology from pattern matching or machine learning in the case of binary classification is borrowed (See Section 2.3.2).

Suppose the ground truth of locations of signal peaks is denoted as a set $\mathcal{P}$. For any given peak detector, its output is a set of peak locations denoted as a set $\mathcal{O}$. Accordingly, the following 3 quantities can be defined:

- **True positive** $TP \overset{\text{def}}{=} |\mathcal{P} \cap \mathcal{O}|$, the number of EMG peaks that are detected.

- **False positive** $FP \overset{\text{def}}{=} \left|\mathcal{O} \setminus \mathcal{P}\right|$, the number of non-EMG peaks (peaks from noise) that are detected (number of false alarms).
- **False negative** $FN \overset{\text{def}}{=} \left|\mathcal{P} \setminus \mathcal{O}\right|$, the number of EMG peaks that are *NOT* detected (number of missed detections).

To quantify detection performance, terms from pattern recognition as discussed in Section 2.3 are employed.

- **Recall** is the fraction of real EMG peaks that are detected
- **Precision** is the fraction of detected peaks that are actual EMG peaks

Or formally:

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \end{aligned}$$

Recall is also called *Detection Rate* in detection theory or *hit rate* in the signal processing field, or *sensitivity* in statistics. A good detector has a high recall (close to 1). *Precision* is directly related to *False Discovery Rate (FDR)* as used in statistics: Precision = $1 -$ FDR. The concept of precision describes another very important aspect of the peak detection algorithm. A high recall only means most of the true signal peaks were found, it doesn't imply anything about the "accuracy" or "precision" about this detection method. Along with all the true signal peaks, "fake peaks" or noise could also be labeled as peak signals. Consider an extreme case. If a detector labels all peaks as signal peaks, then the recall can be as high as 1. However, the precision will be extremely bad: close to 0 for transient peak signals.

When comparing two detectors, one detector is better if it has both higher recall and precision. However, it's not easy to compare two detectors if one detector has higher recall but lower precision: it depends on which quantity is more important to the specific application. For example, in radar detector for military purpose, one wants to have extremely high recall, but just a decent precision.

Therefore, different emphasis or weights can be placed on recall and precision when measuring the performance of a detector according to the specific application. *F-score* is defined to quantitatively describe the overall performance. The general formula for F-score with a positive real β is:

$$F_\beta \overset{\text{def}}{=} (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{3.4.2.1}$$

By choosing different values of β, the F-score puts different weights on the precision and recall: a larger β means more emphasis on recall while a smaller β means more emphasis on precision.

In this thesis, F_1 score, the harmonic mean of precision and recall is used:

$$F_1 \overset{\text{def}}{=} 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.4.2.2}$$

Both precision and recall change as the SNR of the data changes, so the performance of the detector is evaluated under different SNR environments, especially under a low SNR environment.

In traditional detection theory, a Receiver Operating Characteristic (ROC) curve is plotted for a binary detector. The ROC curve is a plot of the relationship between the detection rate (P_D) and the false alarm rate (P_{FA}), when the threshold takes different values. Different threshold of the detector yields different pairs of (P_D, P_{FA}). Typically, a higher threshold yields smaller P_D and smaller P_{FA}. A ROC curve is used to compare between different detectors. The ROC curve of a better detector is closer to the upper-left corner of the graph, as that region gives high P_D and low P_{FA}. In this thesis, recall vs. precision curve is plotted. The curve of recall vs. precision is a similar plot to the traditional ROC curve. In fact, recall is the same as the detection rate P_D. In the peak detection problem, false-alarm rate P_{FA} is ill-defined. As a result, the closely related quantity, precision, is used instead. A detector with a small false-alarm rate normally gives a large precision. In recall vs. precision curve, a better detector yields a curve closer to the upper-right corner of the graph, as that region gives both high precision and recall.

More specifically, precision vs. recall curve is plotted by varying the threshold coefficient c in the proposed peak detector under a very low SNR environment. The threshold coefficient c is varied because it's used in the binary hypothesis testing during peak maxima detection at each scale.

In summary, following plots will be generated based on the detection results of a given peak detector.

- fix the detector parameter; find the precision, recall, and F-score of the detector for different SNRs. Evaluate the following relationships:
  - precision vs. SNR
  - recall vs. SNR
  - F-score vs. SNR
- fix the SNR of the data; find the recall and precision of the detector with different detector thresholds. Evaluate the following relationships:
  - recall vs. precision

  - recall vs. threshold
  - precision vs. threshold

3.4.3 Choose Algorithm Parameters from Simulation

There are four parameters associated with the peak detection algorithm. It's impossible to formulate an analytical formula that relates the detection performance with the four parameters. Hence, I propose to choose the parameters by experimenting on simulated data that is representative of the application.

As discussed briefly in the beginning of Section 3.3, the problem is formulated as a generic peak detection problem regardless of the specific shapes of the EMG signal. Therefore, the simulated generic peak signal is used rather than the simulated EMG signal for the purpose of selecting algorithm parameters. Please note that a set of "optimal parameters" is not necessarily the best, but the best for the given data. In fact, choosing the algorithm parameters from experiments is rather a subjective process. In this thesis, I just chose one that gives an overall satisfying and seemingly best detection performance.

To find the parameters that work well for various shapes of peaks, the detection performance for peak signals with various frequencies is tested. Also, a relatively large sparsity in generating the simulated data is used, in order to prevent a situation where the wavelet coefficients from two close peaks interfere with each other. In practice, real signals, such as the EMG signal, will often have two peaks very close to each other. However, when choosing the algorithm parameters, I want to focus on the ability of the algorithm at detecting single peak signal from noisy environment. The next section will demonstrate the ability of the algorithm to detect EMG peaks, and shows that the peak detection algorithm works well even when peaks are close to each other.

In general, the algorithm parameters are chosen by brute-force search. The peak detection performance is checked for different combinations of the 4 parameter values and visually inspect the recall vs. precision curve and the F-score vs. SNR curve to choose the best one. Firstly, the ranges of reasonable values for all the parameters are chosen. Then several candidate values within the range are chosen. If many candidate values are chosen for every parameter, then there will be a lot of 4-parameter pairs, and hence the time to run the peak detection for all parameter pairs is enormous. To avoid this, at first stage, only a few candidate values are roughly chosen, as this step only aims to narrow down the range of the parameters. In the next stage, candidate values are finely selected in the new range. In this way, a set of good parameters can be quickly found.

When initially checking the performance of the detector, the F-score vs. SNR curve is used to find a range of candidate parameter pairs, and then the recall vs. precision curve is used to select the best one among the candidates. The F-score is a quick way to select good candidate values, since those parameters that give very low F-score (F-score < 0.9) can be automatically filtered. After that, the parameter selection process focuses on the low-SNR data (SNR $= 1 \sim 10$), and select the one parameter pair that gives the best recall vs. precision curve.

From observation, the optimal parameters are different for different SNRs, but the detection performance doesn't differ too much within a finite range of SNRs. For example, it's good enough to have just one set of parameters for low-SNR (SNR < 10) data, and one set of parameters for high-SNR (SNR > 10) data. In practice, one can choose a proper set of parameters based on the actual noise levels in the application. The EMG application focuses on low-SNR data, since this is the most challenging and interesting problem.

From the physiology of the EMG signal, the frequency range of interest is $0 \sim 400$Hz, and the major component is at 60Hz. Hence:

$$f_L = 20 \tag{3.4.3.1}$$

$$f_U = 400 \tag{3.4.3.2}$$

The lower frequency bound, f_L cannot be 0, since this implies an infinite range of scales in the CWT. 20Hz is a good lower bound, as most of the frequency components in a EMG signal is above 20Hz.

Note that in the theoretic derivation, angular frequency, ω, (with unit of rad/sec) is used to represent frequency, but in practice, people most often use frequency, f, (with unit of Hz). The previous equations using ω to present frequency can be adapted to frequency in Hz with simple relation:

$$\omega = 2\pi \cdot f$$

Other parameters used in the simulation are:

$$f_s = 2000$$

$$\text{sparsiy} = 80\%$$

With the above parameters, the best algorithm-parameter pair selected is:

$$\text{density:} \quad d = 3; \tag{3.4.3.3a}$$

$$\text{threshold coefficient:} \quad c = 2.2; \tag{3.4.3.3b}$$

$$\text{time threshold:} \quad w = 5; \tag{3.4.3.3c}$$

$$\text{ridge length threshold coefficient:} \quad k_{ridge} = 0.6; \tag{3.4.3.3d}$$

3.4.4 Experimental Results

3.4.4.1 Methods for Comparison

This section describes a couple of peak detection methods widely used in the literature, and they will be implemented for comparison with the proposed method.

The most straightforward, widely used in EMG clinics, method for peak detection is a simple *amplitude thresholding*, in which only peaks passing certain threshold are identified as real peaks; peaks below the threshold are considered noise. This method cannot work well, if the sampling rate is high or the SNR is low. In either case, the problems occur when noise results in peaks on the rising and falling side of a true EMG peak. These peaks normally have large amplitude such that they pass the threshold, but arise from noise.

A better method is to incorporate some prior knowledge about the peak shapes: only peaks with certain width are considered real peaks, assuming peaks from noise have small width. Therefore, the simple thresholding method can be improved by eliminating peaks from noise that are near a true peak. This way, the peaks on the rising and falling sides are shadowed by the true peak. Specifically, a threshold on the minimum separation between two peaks is set: if two peaks are within the separation threshold, the smaller one is removed, and only the larger one is kept. This method is referred to as the *improved amplitude thresholding* in this thesis. In the implementation, the separation threshold is set to be 5 samples which is 2.5ms, considering the data sampling rate is 2000Hz. This threshold is based on the fact that the simulated peaks have a base frequency of 60Hz. However, this method puts too much restrictions on the width of the peaks. If the peak signals have a wide range of widths, then the performance of this method would be affected a lot.

A further improved method is to use digital filtering before peak detection. By digitally *low-pass filtering* the data, peaks from noise will be removed, and this works for a wide range of shapes of peaks, unlike the method above. I used Matlab's digital filter design tool to implement this filter with passband of 400Hz and stopband of 450Hz. However, digital filtering can only suppress noise to a certain level. Because a digital filter must allow all peaks of interest to pass, it must have a relatively wide passband. Therefore, for a peak with a particular frequency component, the filtering performance is not very good. In other words, digital filtering is not very selective to the peak signals. In fact, for very low SNR data, digital filtering wouldn't separate real peaks from noise very well. This is where the wavelet method shines. CWT at each scale is similar to digital filtering with a specific passband. Hence, for every peak signal, with a proper choice of density value, there would exists a couple of selected scale values, such that the passband of the wavelets at those scales cover the major frequency components of the peak signal. As a result, the CWT is much more selective than simple digital filtering. In addition, the additional process of ridge identification further separates the peak signals from the noise.

To compare with classical transient detection methods, Nuttall's classical power-law detector [43] was selected as a benchmark. Consider a window size of $2w + 1, w \in \mathbb{Z}^+$, then the power at sample i given data $x[k], k \in \mathbb{Z}$ is defined as:

$$p[i] \overset{\text{def}}{=} \frac{1}{2w+1} \sum_{k=i-w}^{i+w} x[k]^2$$

The threshold on the power is chosen to be a coefficient times the power of the noise (the variance). This method is termed *Nuttall's power detector.*

To compare with other peak detection method using the wavelet transform, Du's wavelet-based detector [13] was selected for comparison. Du's paper offers very good peak detection results in mass spectroscopy when compared to other classical peak detection algorithms. But the proposed wavelet-based double thresholding method in this thesis offers more theoretical support and allows users to fine tune the parameters specifically for their applications. It will be seen that this detector offers better detection results compared to Du's method. I call this method *Du's wavelet detector.*

As a upper bound for the peak detection performance, I also proposed a method similar to the "Matched Filtering" in classical detection theory, termed the *matched filter.* This method uses the peak signal itself as the digital filter, and find the correlation (by running convolution) between data and the peak signal (See Eq. (2.3.1.9). After that, peaks above a certain threshold are selected to be actual peaks. This method makes use of full knowledge of the signals, so it naturally gives the optimal performance, and serves as a upper bound for evaluating the peak detection algorithm.

In summary, I have implemented the following 7 methods (6 from literature for comparison; "wavelet" refers to the wavelet-based double-thresholding method proposed in this thesis), and have performed experiments on simulated data with these methods. The legends on the resulting plots will match the method names given in the following list.

- amplitude thresholding
- improved amplitude thresholding
- low-pass filter
- power-law detector
- Du's paper
- wavelet (developed in this thesis)
- matched filter

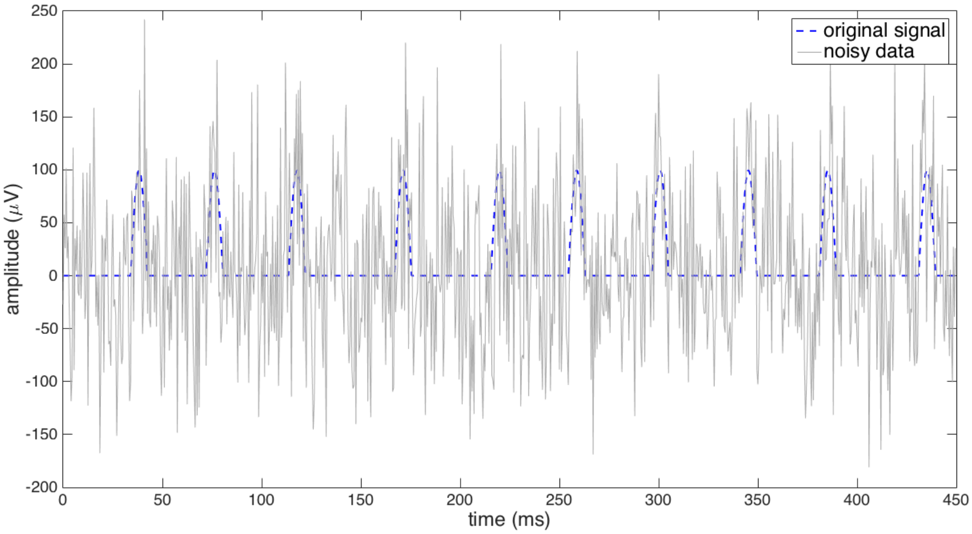
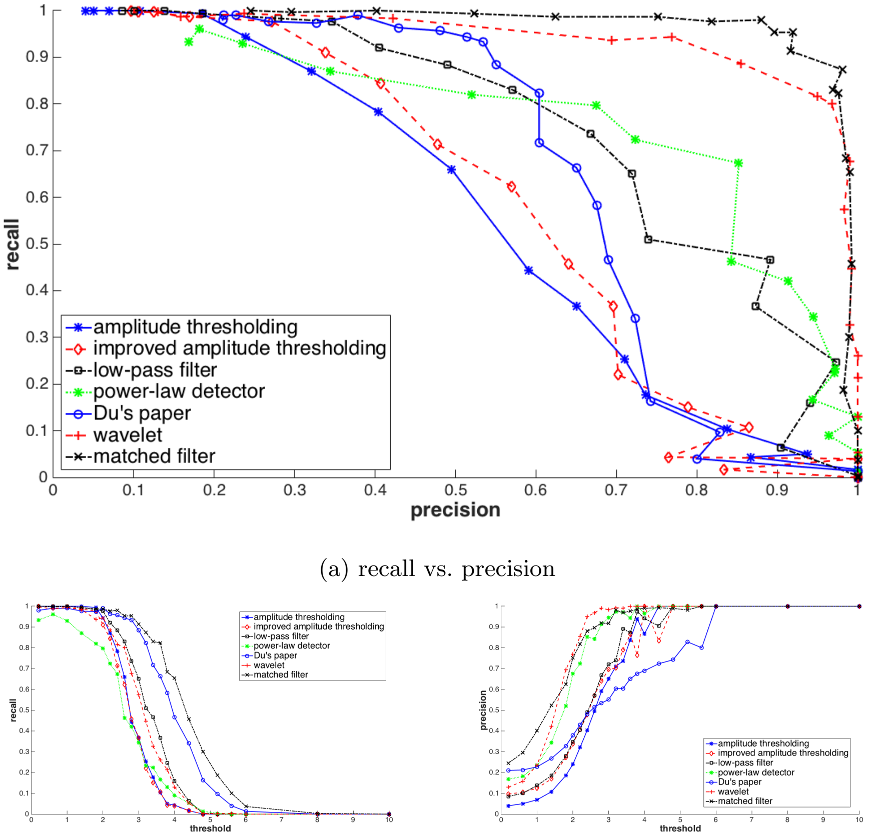


Figure 3.4-3: Example of simulated data with 10 peak signals (60Hz peak with SNR = 1)

3.4.4.2 Test Results on the Generic Peak Signal

Fig. 3.4-3 shows a section of an example simulated peak signals with SNR being 1. This plot shows that for data with very low SNR (e.g., SNR = 1), it's almost impossible to visually detect the real signals. Since the EMG signal has most of its frequency component around 60Hz, peak detection will be evaluated on the generic peak signal of 60Hz. Fig. 3.4-4 compares the recall vs. precision curves of all seven methods previously described in Section 3.4.4.1. In addition, the recall vs. threshold curves and the precision vs. threshold curves are plotted to give the readers more insights. Figure 3.4-4a verifies the basic intuition about the 7 methods: amplitude thresholding, being the simplest method, performs worst; improved amplitude thresholding gives slightly better results; more complex methods, such as low-pass filtering, power-law detector, and Du's method give even better performance. The wavelet-based algorithm performs significantly better than all of those methods. In fact, the performance of the wavelet-based algorithm is close to the upper bound given by the matched filtering. As can be seen from Fig. 3.4-4b, the recall for the wavelet-based method is similar to that of low-pass filtering, but lower than Du's method. However, the precision evaluation in Fig. 3.4-4c shows that the wavelet-based method has the highest precision among all methods. Even though the power-law detector gives close precision, its recall is much worse. In general, there is always a trade-off between precision and recall: a method giving good recall usually suffers from bad precision, vice versa. The wavelet method is designed to give a good overall performance (good recall vs. precision curve) without sacrificing too much on either recall or precision. The fact that the wavelet method gives best precision and reasonable recall comes from the intuition behind the method: by imposing double-thresholding, the amount of false positives can be reduced, hence increasing precision, while



(a) recall vs. precision

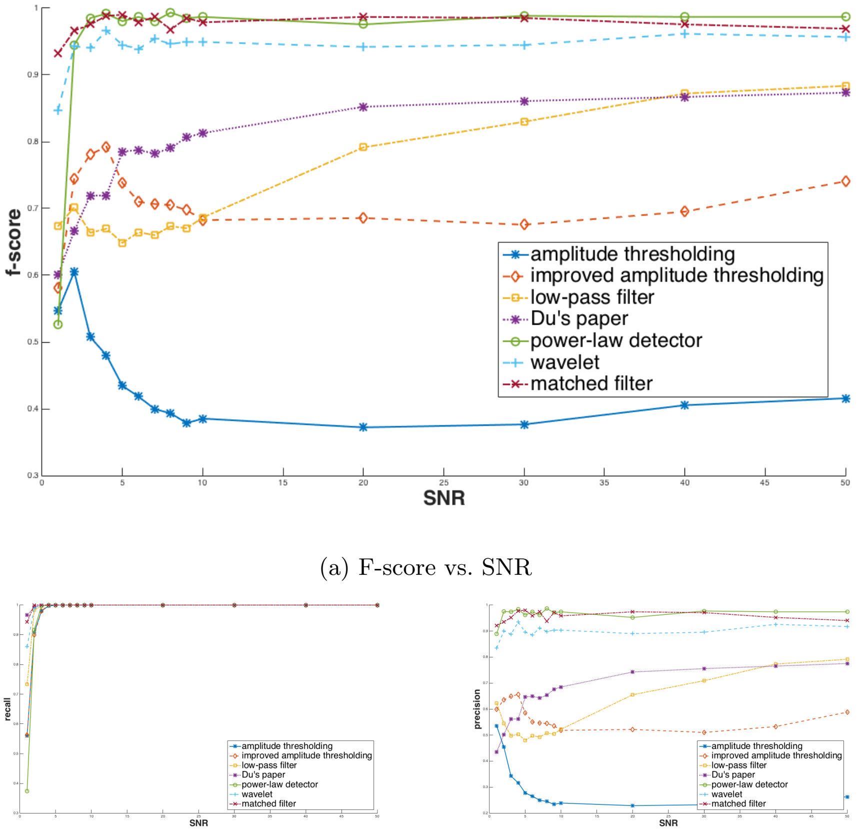
(b) recall vs. threshold

(c) precision vs. threshold

Figure 3.4-4: Experimental Results on the generic peak signal. Recall and precision are calculated under different thresholds for the proposed wavelet method (labeled as "wavelet" in the plot), as well as other six methods for comparison. For detailed descriptions on what methods the labels refer to, please see Section 3.4.4.1. The frequency of the peak signal is 60Hz, and the SNR is 1. (a) gives the recall vs. precision curves; (b) gives the recall vs. threshold curves; (c) gives the precision vs. threshold curves.

not hurting too much the detection rate, or recall.

Now compare the 7 methods in terms of their F-scores. Fig. 3.4-5 calculates the recall, precision, and F-score of the 7 methods for peak signals with SNR ranging from 1 to 50. Naturally, a higher SNR signal gives better detection performance, as shown in all three plots. From Fig. 3.4-5a, the wavelet-based method performs much better than all other methods except the power-law detector, and almost as well as the matched filtering method, the theoretical upper bound. The wavelet method gives slightly lower F-score compared to the power-law detector in high-SNR range (which is less interesting in the application), but if one zooms in on the low-SNR range, then the power-law detector drops its F-score significantly. Since the goal of this thesis is to detect low-SNR EMG peaks, the wavelet method is much better than the power-law detector. If looking at the recall



(a) F-score vs. SNR

(b) recall vs. SNR

(c) precision vs. SNR

Figure 3.4-5: Experimental results on the generic peak signal. Recall and precision are calculated under different SNRs for the proposed wavelet method (labeled as "wavelet" in the plot), as well as other six methods for comparison. For detailed descriptions on what methods the labels refer to, please see Section 3.4.4.1. The frequency of the peak signal is 60Hz, and the SNR is 1. (a) gives the F-score vs. SNR curves; (b) gives the recall vs. SNR curves; (c) gives the precision vs. SNR curves.

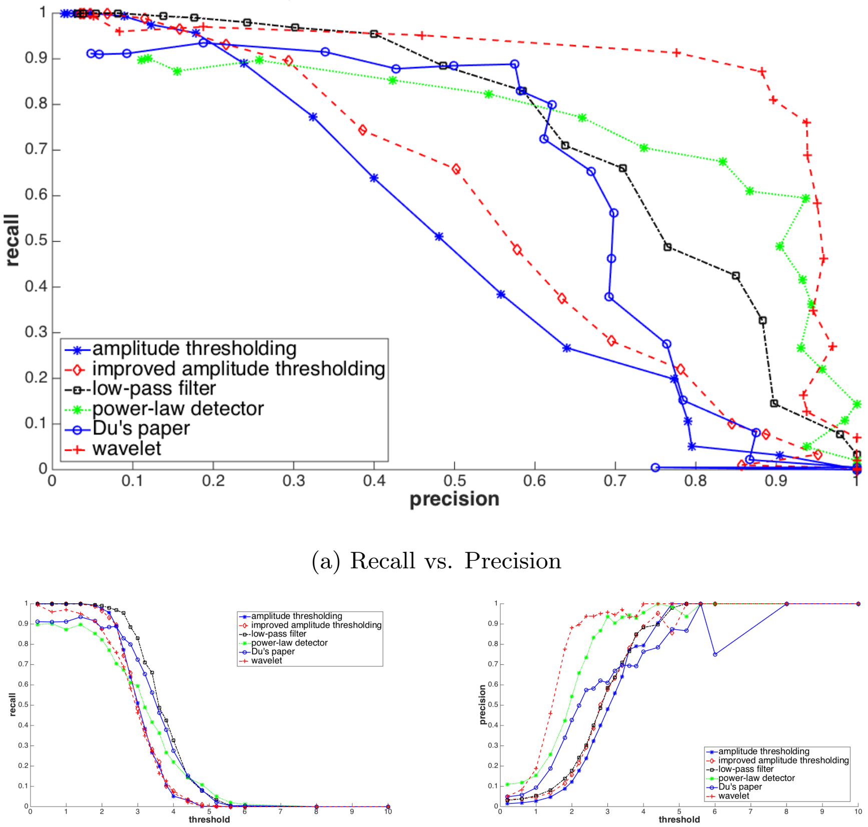
and precision plots, one can tell that the wavelet-based method has slightly lower recall as Du's method, but much better precision over Du's method. This again justifies my observation in the recall vs. precision curves: Du's method gives good recall while the power-law gives good precision, but the wavelet method gives the best overall performance.

3.4.4.3 Test Results on the EMG signal

To demonstrate that the wavelet-based peak detection algorithm works for various shapes of EMG signals, I synthesized three simulated EMG signals. Each simulated EMG signal uses one real EMG waveform from Fig. 3.4-2. As can be seen from that figure, the EMG waveforms are very different from each other. In total, peak detection is performed on three synthesized EMG signals. In the following, I will present the experimental results on all three synthesized EMG signals.

For the EMG signal, there doesn't exist a matched filtering method. I can only compare the wavelet-based method with the rest of methods from literature. Also, from last section, the F-score isn't highly discriminative since the focus is on the low-SNR range, and hence only the recall vs.precision curves are plotted for comparison.

As can be seen from Figure 3.4-6, Figure 3.4-7 and Figure 3.4-8, the detection performance is consistent among all three synthesized EMG signals. The wavelet-based method always has better overall recall vs. precision curves than other methods for all three types of synthesized EMG signals. As predicted, the wavelet-based method has a high precision due to its double thresholding approach, without hurting the detection rate performance too much. Since this detection performance is similar to the one realized in the generic peak signal of the last section, I will not analyze every curve in details. The results are given to demonstrate the effectiveness of the wavelet-based method on the EMG signal. The performances on three different EMG signals of very different shapes demonstrate that the wavelet method can tackle the main challenge in the detection of the EMG signal.

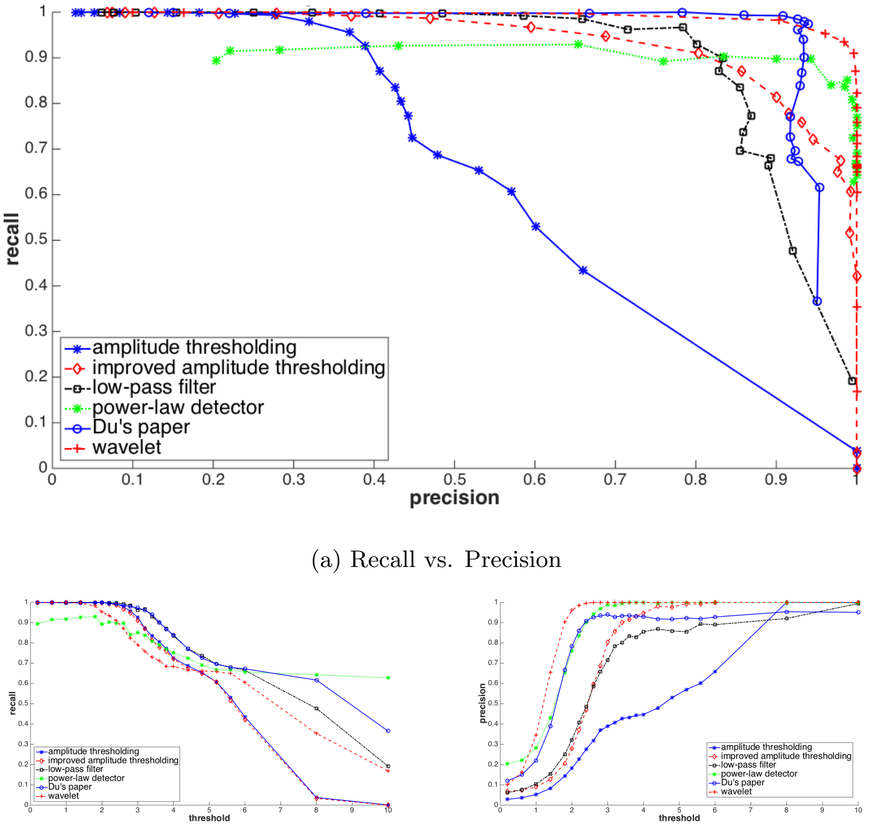


(a) Recall vs. Precision

(b) Recall vs. Threshold

(c) Precision vs. Threshold

Figure 3.4-6: Experimental results on the synthesized EMG signal with EMG waveform from Fig. 3.4-2a. The SNR is 1. Recall and precision are calculated under different thresholds for the proposed wavelet method (labeled as “wavelet“ in the plot), as well as other five methods for comparison. For detailed descriptions on what methods the labels refer to, please see Section 3.4.4.1. (a) gives the recall vs. precision curves; (b) gives the recall vs. threshold curves; (c) gives the precision vs. threshold curves.

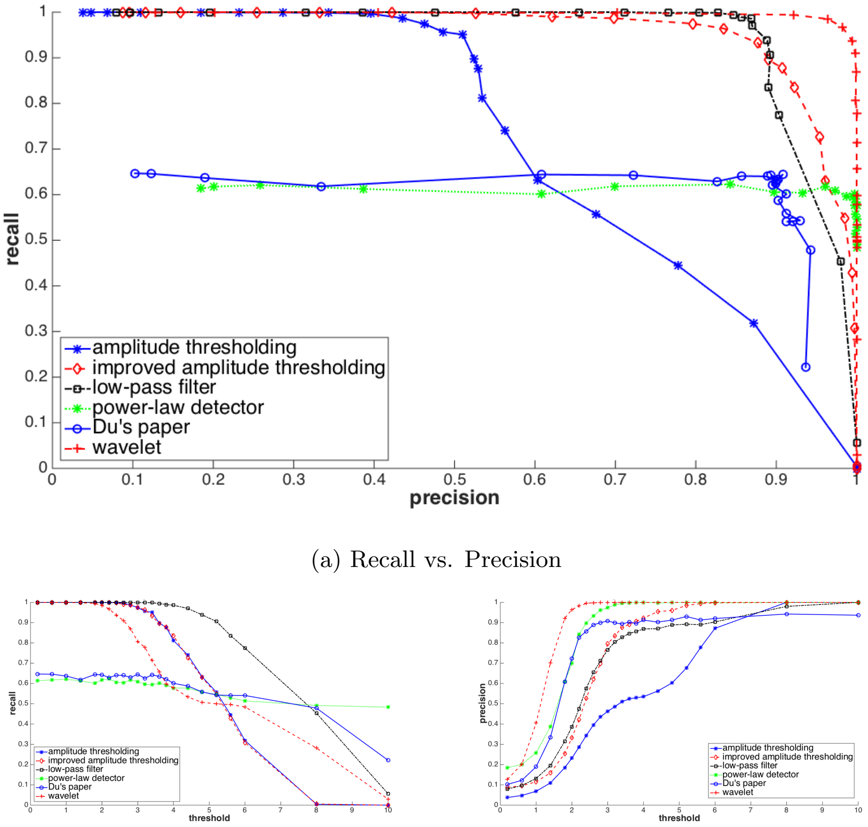


(a) Recall vs. Precision

(b) Recall vs. Threshold

(c) Precision vs. Threshold

Figure 3.4-7: Experimental results on the synthesized EMG signal with EMG waveform from Fig. 3.4-2b. The SNR is 10. Recall and precision are calculated under different thresholds for the proposed wavelet method (labeled as “wavelet“ in the plot), as well as other five methods for comparison. For detailed descriptions on what methods the labels refer to, please see Section 3.4.4.1. (a) gives the recall vs. precision curves; (b) gives the recall vs. threshold curves; (c) gives the precision vs. threshold curves.



(a) Recall vs. Precision

(b) Recall vs. Threshold

(c) Precision vs. Threshold

Figure 3.4-8: Experimental results on the synthesized EMG signal with EMG waveform from Fig. 3.4-2c. The SNR is 10. Recall and precision are calculated under different thresholds for the proposed wavelet method (labeled as "wavelet" in the plot), as well as other five methods for comparison. For detailed descriptions on what methods the labels refer to, please see Section 3.4.4.1. (a) gives the recall vs. precision curves; (b) gives the recall vs. threshold curves; (c) gives the precision vs. threshold curves.

Chapter 4

Segmentation and Clustering of MEPs

This chapter describes a series of steps for processing the EMG signal in order to extract useful information. The EMG signal was recorded from patients with spinal cord injury during their rehabilitation described in [24]. For a complete description on the physiology and characteristics of the said EMG signal, please refer to Section 2.1 and 2.2.

During the rehabilitation training in [24], electrical stimulation is applied to the spinal cord. The stimulation parameters (such as amplitude, frequency, and electrode configurations) are varied manually by the researchers in order to study the different effects of the electrical simulation on the spinal neurons, and to find the optimal set of parameters. The stimulation with one fixed set of stimulation parameters is called one "event". Every event lasts for a couple of seconds. As a result, the processing of the EMG signal is carried out on every event individually. There are several benefits in doing so. Firstly, by performing EMG processing on a short interval, it is safe to assume the statistics of the noise are stationary. Secondly, the EMG signal from different events is likely to show different statistics, and the performance is good when the signal in regard has a stationary statistics. Last but not least, it's desired to compare the characteristics of the EMG signal between different events.

The muscle responses in the EMG signal are called Motor Evoked Potentials (MEPs). MEPs can be classified into two subgroups: monosynaptic MEPs and polysynaptic MEPs. They are also sometimes referred to as the early responses and late responses, respectively, due to the differences of their arrival times after the electrical stimulus. After the peaks of MEPs are detected using the peak detection methods described in Chapter 3, a series of interesting tasks can be further carried out in order to expose finer structures of the MEPs. This information can assist the neurophysiologists in assessing the underlying neural activities of the spinal cord.

The fundamental two tasks are to segment and cluster the MEPs. In this thesis, the goal of segmentation is to identify MEP segments from the EMG signal sequence. More specifically, the boundaries of an MEP need to be obtained. Suppose an EMG sequence of N samples, $\mathbf{X} = \{x_1, x_2, \cdots, x_N\}$, contains one MEP. The goal of segmentation is to obtain index a and b ($b > a$) , such that segment, $\mathbf{w} = \{x_a, x_{a+1}, \cdots, x_b\}$ contains all the samples from the MEP, but no other samples. Clustering, in general, is the task of partitioning a set of objects into different groups, so that two objects within one group are more similar than two objects from two different groups. In the task of MEP clustering, the goal is to group similar MEPs together. This is motivated by the fact that similar MEPs are likely to come from the same neural physiological process. By identifying different clusters of MEPs, different neural processes can be inferred by the physiologists. In particular, monosynaptic MEPs are significantly different from polysynaptic MEPs because of the distinct neural pathways. As a result, the clustering is carried out in a sequential manner: firstly monosynaptic MEPs are separated from polysynaptic MEPs via an initial, crude clustering; then clustering is performed within monosynaptic MEPs and polysynaptic MEPs so as to discover finer structures. The overall procedure is first stated here. Detailed explanations on these step are described in the following sections.

1. Find segments of MEPs from the detected peaks in the EMG signal.

2. Find majority of the monosynaptic MEPs via hierarchical clustering on all segments of MEPs.

3. Further cluster monosynaptic MEPs into subgroups based on clustering with Gaussian mixtures.

4. Decompose segments that contain both polysynaptic MEPs and monosynaptic MEPs

5. Clustering on polysynaptic MEPs with Gaussian mixtures

To demonstrate the proposed method, the same event of the EMG signal is used throughout this chapter. The EMG signal was recorded from left medial gastrocnemius (L MG) of a patient with clinically motor complete spinal cord injuries while lying in the supine position with electrical stimulation applied to the spinal cord [24]. The stimulation amplitude is 7.2V and frequency is 10Hz. This event lasts 30 seconds with 300 simulation intervals (A simulation interval is the time period between two stimuli).

4.1 Segmentation of MEP Waveforms

After the peaks of the MEPs are detected from the EMG signal, MEP waveforms are segmented based on peak information. As observed from the EMG signal, MEPs are composed of consecutive transient peaks. As a result, if two peaks are "next to" each other, then they are likely to be peaks from the same MEP, or vice versa. It's not a good idea to choose a hard threshold because the distance between two peaks varies for different shapes of MEPs, and is not a known priori. A better way is to incorporate the shape information from the wavelet transform of the peaks.

In the proposed wavelet-based peak detection algorithm in Chapter 3, every detected peak corresponds to one ridge in the wavelet space (the time-scale space). The wavelet used is the Mexican hat wavelet, which naturally resembles a peak. The peak location is estimated from the translation of the maximum wavelet coefficient along the ridge, as the magnitude of the wavelet coefficient measures the resemblance between the signal and the wavelet function (See Eq. (3.3.6.2) and Eq. (3.3.6.3)). For convenience, the equations are copied here.

$$(\hat{rx_i}, \hat{ry_i}) = \arg\max_{(x,y)\in\mathcal{R}_i} X(x, y)$$

$$\hat{p}_i = \hat{ry_i}, \qquad \text{for} \quad i = 1, 2, \cdots, N_R$$

where $X(i, j) : (i, j) \mapsto X_{i,j}$, which is the wavelet coefficient matrix defined by:

$$X_{j,k} \overset{\mathsf{def}}{=} (\mathbf{X})_{j,k} = \sum_{n=-\infty}^{+\infty} x[n]\psi_{s_j,k}[n] \qquad j = 0, 1, \ldots, J, \quad k = 0, 1, \ldots, N;$$

$\{\mathcal{R}_i, \quad i = 1, 2, 3, \cdots, N_R\}$ is the set of ridges detected, where N_R is the total number of ridges detected, and $\mathcal{R}_i$ is a set of pair of indices: $\mathcal{R}_i = \{(rx_{i,j}, ry_{i,j}), \quad j = 1, 2, \cdots, L_i\}$, where $rx_{i,j}$ and $ry_{i,j}$ give the row and column index (as in $\mathbf{X}$) of the j-th peak maxima along ridge $\mathcal{R}_i$, respectively, and $L_i = |\mathcal{R}_i|$ is the number of peak maxima along the ridge.

The scale of the maximum coefficient along the ridge $\mathcal{R}_i$ is therefore:

$$\hat{rs_i} = s_{\hat{rx_i}}$$

since $\hat{rx_i}$ gives the index in the scale set $\mathcal{S} \overset{\mathsf{def}}{=} \{s_0, s_1, \ldots, s_j, \ldots, s_J\}$.

Now the effective support of the peak at $\hat{p}_i$ is defined with respect to the wavelet function at scale $\hat{rs_i}$. Although the theoretical mother wavelet function is not time-limited, its most energy is confined only within a small region around the origin. Define the effective support of the Mexican Hat mother wavelet in time domain as w (as shown in Figure 4.1-1), then the Mexican Hat wavelet at any scale s has effective support of $w \cdot s$. There are multiple choices of the values of w. In this thesis, w is chosen to be $[-4, 4]$, and the signal energy within the support occupies almost 100% of the total energy.

The effective support of a detected peak at $\hat{p}_i$ is estimated to be the same as the wavelet at scale $\hat{rs_i}$:

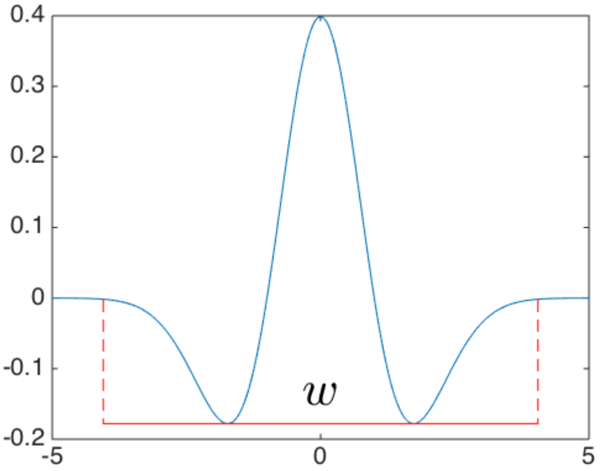


Figure 4.1-1: Mexican Hat mother wavelet $\psi(t)$ (given by Eq. 3.3.1.1) and its effective support (denoted by red line and w)

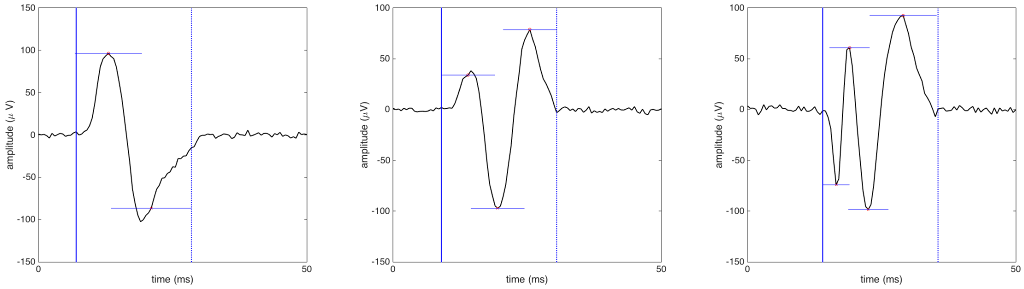
$$w(p_i) = w \cdot \hat{rs}_i \tag{4.1.0.1}$$

If the supports of two consecutive peaks overlap, then these two peaks are claimed to be from the same MEP. More formally, $\hat{p}_i$ and $\hat{p}_{i+1}$ are judged to be from the same MEP, if and only if:

$$\hat{p}_i + w(p_i)/2 > \hat{p}_{i+1} - w(p_{i+1})/2 \tag{4.1.0.2}$$

Example results are shown in Figure 4.1-2. The figure contains three EMG slices containing different MEPs to demonstrate the effectiveness of this method on different shapes of MEPs. For every peak estimated, an estimated effective support is calculated using Eq. (4.1.0.1). After this is done for all the peaks, peaks are grouped into MEPs based on the criterion in Eq. (4.1.0.2). The onset of an MEP is estimated as the beginning of the effective support of its first peak, while the end of an MEP is estimated as the end of the effective support of its last peak, as shown in Figure 4.1-2.

Given the EMG signal of N samples: $\mathbf{X} = \{x_1, x_2, \cdots, x_N\}$. The result of this segmentation procedure is a set of MEPs: $\mathcal{W}_{MEP} = \{\mathbf{w}_i\}_{i=1}^{N_M}$, where $\mathbf{w}_i$ is the waveform i-th MEP: $\mathbf{w}_i = w_{i,1}, w_{i,2}, \cdots, w_{i,D_i}$, and N_M denotes the total number of MEPs. Extra information about the MEPs are also obtained to assist further analysis. The onset locations of MEPs are denoted as $\mathbf{b}_{MEP} = \{b_1, b_2, \cdots, b_{N_M}\}$ where $b_i \in \{1, 2, \cdots, N\}$ is the index of $w_{i,1}$ as in $\mathbf{X}$. If one MEP is caused by the electrical stimulus, then this stimulus is said to be "associated" with this MEP. This stimulus associated with MEP $\mathbf{w}_i$ is simply obtained by finding the closest stimulus that comes before it. Given an array of the locations of N_s stimuli: $\mathbf{T}_{stim} = \{t_1, t_2, \cdots, t_{N_s}\}$ where $t_i \in \{1, 2, \cdots, N\}$



(a) R MH

(b) L MG

(c) L VL

Figure 4.1-2: Examples of segmenting MEPs from effective supports of peaks: Each subplot shows one piece of the EMG signal simulated from one of the MEP waveforms in Figure 3.4-2. The black line is the simulated EMG signal containing one MEP. The red dots/crosses are estimated positive/negative peak locations from the proposed wavelet method. The effective support of every peak is plotted as a short blue horizontal line centering at the peak. The blue vertical line denotes the onset of an MEP, while the dotted vertical line denotes the end. R MH = right medial hamstrings. L MG = left medial gastrocnemius. L VL = left vastus lateralis.

gives the index of i-th stimulus in $\mathbf{X}$, then the stimulus associated with $\mathbf{w}_i$, a_i follows:

$$a_i = \underset{t \in \mathbf{T}_{stim} \cap t < b_i}{\arg\max} \; t - b_i \tag{4.1.0.3}$$

The result is a set of indices $\mathbf{a} = \{a_i\}_{i=1}^{N_M}$.

Running the peak detection algorithm proposed in this thesis followed by the above segmentation method on the example event of the EMG signal yields 469 MEPs. The EMG signal is recorded from left medial gastrocnemius (L MG) of a patient with clinically motor complete spinal cord injuries while lying in the supine position with electrical stimulation applied to the spinal cord. The stimulation amplitude is 7.2V and frequency is 10Hz. This event lasts 30 seconds with 300 simulation intervals.

4.2 Identify Cluster of Monosynaptic MEPs

As discussed in Section 2.1.2.2, MEPs can be classified as monosynaptic MEPs and polysynaptic MEPs. Monosynaptic MEPs are usually the direct responses of muscles after electrical stimulation, while polysynaptic MEPs are typically indirect responses. It takes different times for the effect of the electrical stimulation to show up in the muscle cells. The *latency* of an MEP is defined as the time between the electrical stimulation and the onset of the MEP. An example of the latencies of two MEPs is shown in Figure 4.2-1.

The latency of a monosynaptic MEP is normally larger than that of a polysynaptic one. Hence,

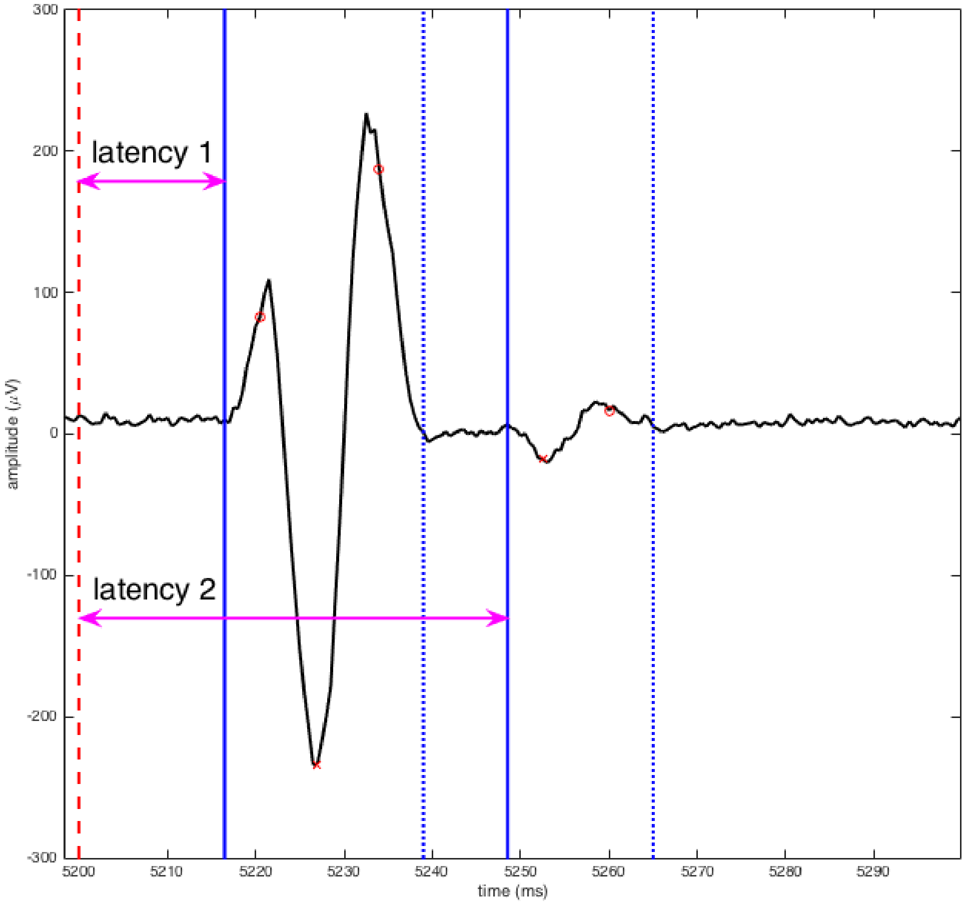


Figure 4.2-1: Example of latency of an MEP when occurrence of MEP is defined as its onset: The vertical red dotted line on the far left indicates the occurrence of one stimulus. *latency 1* is the latency of the monosynaptic MEP, while *latency 2* is the latency of the polysynaptic MEP.

sometimes monosynaptic MEPs are also called early responses while polysynaptic MEPs are called late responses. Monosynaptic MEPs typically have similar latencies among them, while monosynaptic MEPs don't. In addition, monosynaptic MEPs are normally much stronger than polysynaptic MEPs. For a complete explanation of the underlying physiology, please refer to Section 2.1.2.2.

The ultimate goal is to divide MEPs into monosynaptic ones and polysynaptic ones, and perform further analysis (such as clustering) within each group. To achieve this goal, the cluster that contains the majority of the monosynaptic MEPs is identified in this section, because monosynaptic MEPs have a relatively predictable structure, and hence easier to identify. Then the remaining MEPs are processed to extract remaining monosynaptic MEPs, and absorb them to the cluster of monosynaptic MEPs. The final remaining MEPs are grouped into the cluster of polysynaptic MEPs.

4.2.1 Feature Extraction

Based on these prior knowledge on the key differences between polysynaptic MEPs and monosynaptic MEPs, the following features are extracted for every MEP:

Strength: the strength of i-th MEP $\mathbf{w}_i$ is defined as its energy:

$$strength(i) = \|\mathbf{w}_i\|^2 = \sum_{j=1}^{D_i} w_{i,j}^2 \tag{4.2.1.1}$$

Latency: the latency of i-th MEP[1].

$$latency(i) = b_i - a_i \tag{4.2.1.2}$$

Duration: the length of i-th MEP: $duration(i) = D_i$

As a result, each MEP $\mathbf{w}_i$ is associated with a vector v_i in its feature space, comprised of its strength, latency and duration: $v_i = [strength(i) \quad latency(i) \quad duration(i)]$. The set of all features are $\mathbf{v} = \{v_i\}_{i=1}^{N_M}$. Denote the k-th features of all MEPs as $\mathbf{v^k} = \{v_{i,k}\}_{i=1}^{N_M}$.

The features need to be scaled to have a common value range so that all features have equal impact on the computation of the distance. Two scaling schemes are employed.

[1] The neurophysiologists define latency to be $b_i - a_i$, as shown in Figure 4.2-1. Alternatively, the weighted mean (defined in Eq. (4.3.1.8) as τ_i, and used as the alignment mark) is less sensitive to the noise and the non-stationarity of MEPs than the onset of the MEP, b_i. As a result, weighted mean is instead used to calculate the latency feature in the hierarchical clustering.

Interval scaling: the interval-scaled feature of $\mathbf{v^k}$ is:

$$\hat{v_{i,k}} = \frac{v_{i,k} - \min_i v_{i,k}}{\max_i v_{i,k} - \min_i v_{i,k}} \tag{4.2.1.3}$$

Ratio scaling: the ratio scaling is performed simply by doing a log transform followed by an interval scaling:

$$\hat{v_{i,k}} = \frac{\log(v_{i,k}) - \min_i \log(v_{i,k})}{\max_i \log(v_{i,k}) - \min_i \log(v_{i,k})} \tag{4.2.1.4}$$

Both scaling are used for numeric features. The interval scaling is the most common one, and ratio scaling is used when the feature takes exponential values. In the proposed method, the latency and duration features are interval-scaled. The strength feature is ratio-scaled so as to minimize the distance between monosynaptic MEPs as the energy of an MEP takes value in a wide range.

Different weights can be put on different features so as to put more emphasis on certain features. Suppose the weights vector is $\mathbf{c} = [c_1\ c_2\ c_3]^T$, then the k-th feature is replaced by $\mathbf{v^k}$ multiplied by c_k. In the proposed method, $c_1 = 1, c_2 = 1, c_3 = 2$ to highlight the difference of duration between monosynaptic MEPs and polysynaptic MEPs.

4.2.2 Hierarchical Clustering

The objective of the clustering is to partition MEPs into two groups: the monosynaptic MEPs and the polysynaptic MEPs. There are a number of clustering techniques employed for the EMG signal, such as hierarchical clustering [31, 42, 15, 8], k-means clustering [52]. k-means clustering is simple and efficient, but it is sensitive to outliers and initial seeds, and it's not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres). Since the detection and segmentation are not perfect, there will be outliers. In addition, as one can see when the features are plotted, the clusters of monosynaptic MEPs are not hyper-spheres at all. Because of the weaknesses of k-means clustering, hierarchical clustering is chosen for this task. Although hierarchical clustering is less efficient, the number of MEPs detected from one event is not big (typically $100 - 1000$), so efficiency is not a concern in this application.

Hierarchical clustering involves following three steps:

1. compute the distance between all pairs of objects, and stores the result in a distance matrix.
2. group the objects into a binary, hierarchical cluster tree, called *dendrogram.* Initially every object forms a cluster. Then the two closest pair of clusters gets merged into one new cluster. Repeat the linking until only one cluster remains, resulting a binary tree with root being the final cluster with all objects and leaves being every individual objects.

3. determine where to cut the hierarchical tree into clusters.

Each step can be implemented in various ways. In the proposed method, the following options are chosen.

- the distance measure between two objects is chosen to be the L_1 distance. A L_1 distance between two vectors $\mathbf{p}$ and $\mathbf{q}$ is defined as:

  $$d_1(\mathbf{p},\mathbf{q}) = \|p - q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{4.2.2.1}$$

  This distance is chosen so as to minimize the distance within monosynaptic MEPs while maximizing the distance between monosynaptic MEPs and polysynaptic MEPs.

- In the linking stage, the distance between two clusters is defined as the shortest distance between any two objects in the two clusters, one from each cluster.

- When cutting the hierarchical tree, the criterion is chosen to be the distance between the clusters. If the distance between two clusters are above the threshold, then they are two separate clusters in the final result. The distance is chosen to be 0.1.

The time complexity for the proposed clustering is $O(N_M^2)$ for N_M MEPs.

From the physiology of MEPs, monosynaptic MEPs tend to have similar latencies and durations, and much larger energy than polysynaptic ones. The cutoff threshold is intentionally chosen to be a small value to ensure that there is at least one cluster full of pure monosynaptic MEPs. The potential drawback is that other clusters may contain monosynaptic ones, too. This problem will be solved later. The cut of the dendrogram results in many clusters. One of them contains only the monosynaptic MEPs. The next task is to identify the largest cluster that contains only the monosynaptic MEPs. That cluster of monosynaptic MEPs are plotted with their associated stimuli aligned in Figure 4.2-2. The method of finding this cluster is described as follows.

The idea is to assign a score to every cluster that measures the effective standard deviation of the latencies of monosynaptic MEPs within the said cluster. Then the cluster with the lowest effective standard deviation is chosen to be the cluster of monosynaptic MEPs. Suppose there are N_C clusters $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^{N_C}$, where $\mathbf{C}_i = \{c_{i,1}, c_{i,2}, \cdots, c_{i,n_i}\}$ gives the set of indices of MEPs in the i-th cluster ($c_{i,1} \in \{1, 2, \cdots, N_M\}$ are indices in the set $\mathcal{W}_{MEP}$). The latency feature is $\mathbf{v^2} = \{v_{i,2}\}_{i=1}^{N_M}$, the second features of all MEPs in the feature space. Denote the total number of events as N_E. Then the effective standard deviation of i-th cluster, $\sigma(i)$, is calculated following:

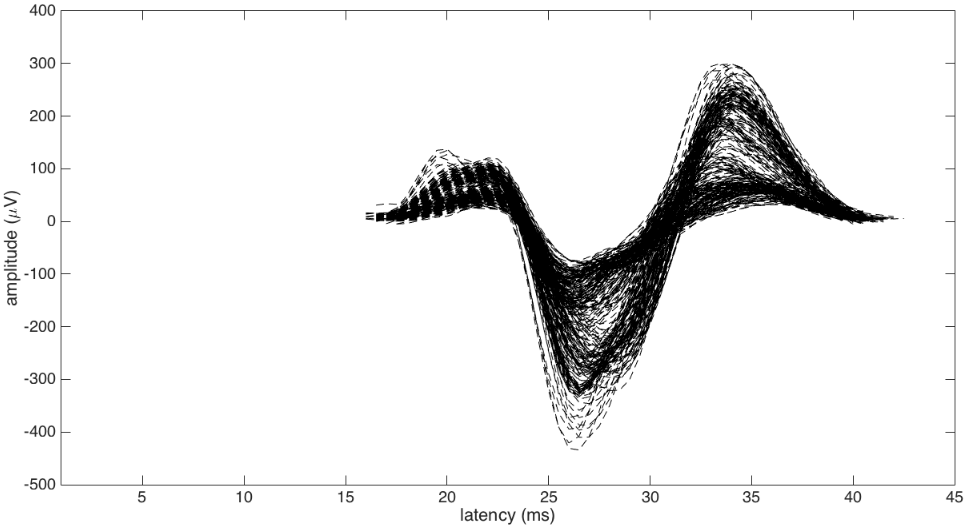


Figure 4.2-2: Example of a cluster with 285 monosynaptic MEPs from the initial hierarchical clustering of 469 MEPs detected and segmented from 30 seconds of the EMG signal. The MEP waveforms are aligned to their associated electrical stimuli at the origin, so the x axis is the latencies of the MEPs. The EMG signal was recorded from left medial gastrocnemius (L MG) of a patient with clinically motor complete spinal cord injuries while lying in the supine position with EES. The stimulation amplitude is 7.2V and frequency is 10Hz.

$$\sigma(i) = std(\{v_{c_{i,j},2}\}_{j=1}^{n_i}) \cdot \frac{|N_E - n_i|}{N_E} \tag{4.2.2.2}$$

where $std(\mathbf{S})$ gives the standard deviation of the set $\mathbf{S}$. The above formula is formulated with following rationale. The monosynaptic MEPs have very consistent latencies, so the standard deviation of the latencies of the cluster of monosynaptic MEPs should be very small (close to 0). However, the standard deviation alone does not always work because small clusters could have smaller standard deviations than the large cluster of pure monosynaptic MEPs. Based on physiology, there is approximately 1 monosynaptic MEP in one event, so theoretically there should be N_E monosynaptic MEPs in the cluster. As a result, the closer that n_i is to N_E, the higher confidence that $\mathcal{C}_i$ contains the monosynaptic MEPs. As a result, a factor $\frac{|N_E - n_i|}{N_E}$ is multiplied to the standard deviation to get rid of the small clusters. The index of the identified cluster of monosynaptic MEPs is:

$$\hat{I}_M = \underset{i \in \{1,2,\cdots,N_C\}}{\arg\min} \ \sigma(i) \tag{4.2.2.3}$$

In the example shown in Figure 4.2-2, the identified cluster of monosynaptic MEPs contain 285 MEPs and have a effective standard deviation of 0.000371. MEPs of the second largest cluster from hierarchical clustering is also plotted in Figure 4.2-3 to give readers some insight on the hierarchical clustering result. It contains 46 polysynaptic MEPs.

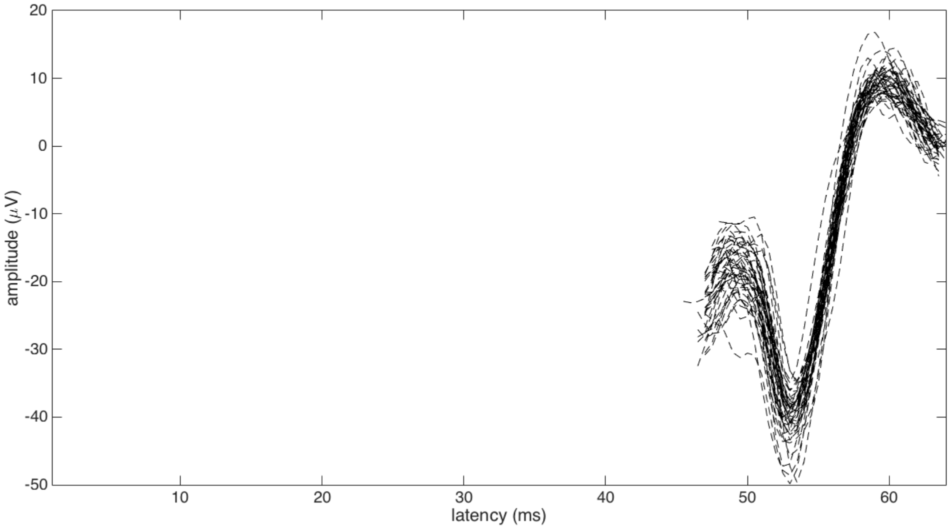


Figure 4.2-3: Example of the second largest cluster with 46 MEPs from the initial hierarchical clustering of 469 MEPs detected and segmented from 30 seconds of the EMG signal. The MEP waveforms are aligned to their associated electrical stimuli at the origin, so the x axis is the latencies of the MEPs. The EMG signal was recorded from left medial gastrocnemius (L MG) of a patient with clinically motor complete spinal cord injuries while lying in the supine position with EES. The stimulation amplitude is 7.2V and frequency is 10Hz.

4.3 Clustering of Monosynaptic MEPs

In the hierarchical clustering, the features are chosen so as to maximize the difference between monosynaptic MEPs and polysynaptic MEPs, while at the same time minimizing the difference within monosynaptic MEPs. Because both monosynaptic and polysynaptic MEPs can have various waveforms and the exact shapes are not a known priori, shape information is not used in the hierarchical clustering. After the majority of the monosynaptic MEPs are extracted during the hierarchical clustering, a further clustering within them is performed to find intrinsic structures of the monosynaptic MEPs.

As discussed in the physiology of MEPs in Section 2.1.2.2, MEPs are the activities of the muscle cells in response to the electrical stimulation provided by the implanted electrodes on the spinal cord. The neurons in the spinal cord are excited and their activities are modulated by the electric field. Different neural pathways would result in different MEPs shown in the EMG signal of the lower-body muscles. The goal of the clustering within monosynaptic MEPs is to infer the different neural pathways or activities from the different structures or shapes of the MEPs.

4.3.1 Feature Extraction: Principal Component Analysis

In order to cluster with the shape information, the entire waveform of an MEP should be used. However, the MEP waveforms contain about $\sim 10 - 100$ samples, and many of them are highly redundant. The redundant features add unnecessary complexity and computation time to the clustering task, and would shadow the most differentiating features. As a result, the first step before performing any kinds of clustering is to reduce the number of features, and principal component analysis (PCA) is the most common way of doing this.

PCA is a technique that is widely used for applications such as dimensionality reduction, lossy data compression, feature extraction, and data visualization. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the *principal subspace*, such that the variance of the projected data is maximized [4]. Suppose the data is $\{\mathbf{x}_n\}$, where $n = 1, 2, \cdots, N$, and $\mathbf{x}_n = [x_{n1}\ x_{n2}\ \cdots\ x_{nD}]^T$ is a (column) vector in the Euclidean space with dimensionality D. Assume the mean has already been subtracted from the data such that $\bar{\mathbf{x}} = 0$. The subtraction of the mean is a preprocessing of the data before performing PCA. With $\bar{\mathbf{x}} = 0$, the meaning of maximum variance in projected data is achieved.

The goal is to find a subspace with dimensionality $M < D$ such that the projected data contains most of the variance in the original data. Suppose the subspace is given by M orthonormal vectors $\{\mathbf{u}_i\}$ where $i = 1, 2, \cdots, M$. $\mathbf{u}_i = [u_{i1}\ u_{i2}\ \cdots\ u_{iD}]$ is a column vector in original space with following properties:

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \tag{4.3.1.1}$$

$$\|\mathbf{u}_i\| = 1 \tag{4.3.1.2}$$

where δ_{ij} is the Kronecker delta function.

The projection of a given data $\mathbf{x}_n$ onto the subspace $\{\mathbf{u}_i\}_{i=1}^M$ is therefore $\mathbf{z}_n = [z_{n1}\ z_{n2}\ \cdots\ z_{nM}]^T$ where $z_{ni} = \mathbf{x}_n^T \mathbf{u}_i$. Let $\mathbf{U} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_M]$ be the matrix with columns being the orthonormal vectors $\mathbf{u}_i$. Then, the projection of $\mathbf{x}_n$ can compactly be expressed as:

$$\mathbf{z}_n = \mathbf{U}^T \mathbf{x}_n \tag{4.3.1.3}$$

Let matrix $\mathbf{X} = [\mathbf{x}_1\ \cdots\ \mathbf{x}_N]^T$ represent the N data with rows being the observations $\mathbf{x}_n$, and let matrix $\mathbf{Z} = [\mathbf{z}_1\ \cdots\ \mathbf{z}_N]^T$ represent the N projections with rows being the projections $\mathbf{z}_n$. Then:

$$\mathbf{Z} = \mathbf{X}\mathbf{U} \tag{4.3.1.4}$$

From the projection $\mathbf{Z}$, original data can be reconstructed with:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{U}^T \tag{4.3.1.5}$$

$\tilde{\mathbf{X}}$ would be the same as $\mathbf{X}$ if $M = D$. For $M < D$, there is a distortion or error. The distortion measure is the squared distance between the original data and the reconstructed data.

$$J = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\| \tag{4.3.1.6}$$

It turns out PCA minimizes J for a fixed $M < D$ with respect to different choice of the M orthonormal vectors $\{\mathbf{u}_i\}_{i=1}^{M}$ [4].

Theoretically, PCA is achieved by evaluating the covariance matrix $\mathbf{S}$ of the data set $\mathbf{X}$ and then finding the M eigenvectors of $\mathbf{S}$ corresponding to the M largest eigenvalues. The covariance matrix is calculated by: $\mathbf{S} = 1/N \sum_{n=1}^{N} \mathbf{x}_n^T \mathbf{x}_n$.[2] In practice, many algorithms have been developed to compute the M eigenvectors efficiently.

The mathematical derivation can be found in [4]. Here is the summary of the conclusions. Suppose $\{\lambda_k\}$ and $\{\mathbf{u}_k\}$ are the eigenvalues and eigenvectors of $\mathbf{S}$, where $k = 1, 2, \cdots, D$. If $\{\mathbf{u}_k\}_{k=1}^{M}$ are used as the bases of the subspace, then the variance of the projected data onto this subspace is given by $\sum_{k=1}^{M} \lambda_k$. In addition, the distortion measure J can also be derived as:

$$J = \sum_{k=M+1}^{D} \lambda_k \tag{4.3.1.7}$$

Therefore, by choosing the M eigenvectors of $\mathbf{S}$ corresponding to the M largest eigenvalues, the variance of the projected data is maximized while the distortion measure is minimized.

In practice, the choice of M depends on how much of the total variance is desired to be kept in the principal subspace. 90% is typically considered to be a good approximation. In fact, beyond that, the remaining 10% of the variance is mainly composed of noise, so PCA also helps reduce the effect of the noise.

One typical issue when applying PCA to the waveforms is alignment. Suppose $\mathbf{x}_n = [x_{n1}\ x_{n2}\ \cdots\ x_{nD}]^T$ gives the feature vector of length D. Then, the i-th feature x_{ni} should be the same type of feature for all $\mathbf{x}_n$ where $n = 1, 2, \cdots, N$. Unfortunately, MEP waveforms don't satisfy this. First of all, the segmented waveforms of MEPs don't have the same length. Secondly, the amplitude at every sampled time is simply a measurement of the voltage, and there is no information about how the

[2] This is why data $\{\mathbf{x}_n\}$ needs to be centered. Only when $\bar{\mathbf{x}} = 0$ will $\mathbf{S}$ given by above equation give the covariance matrix.

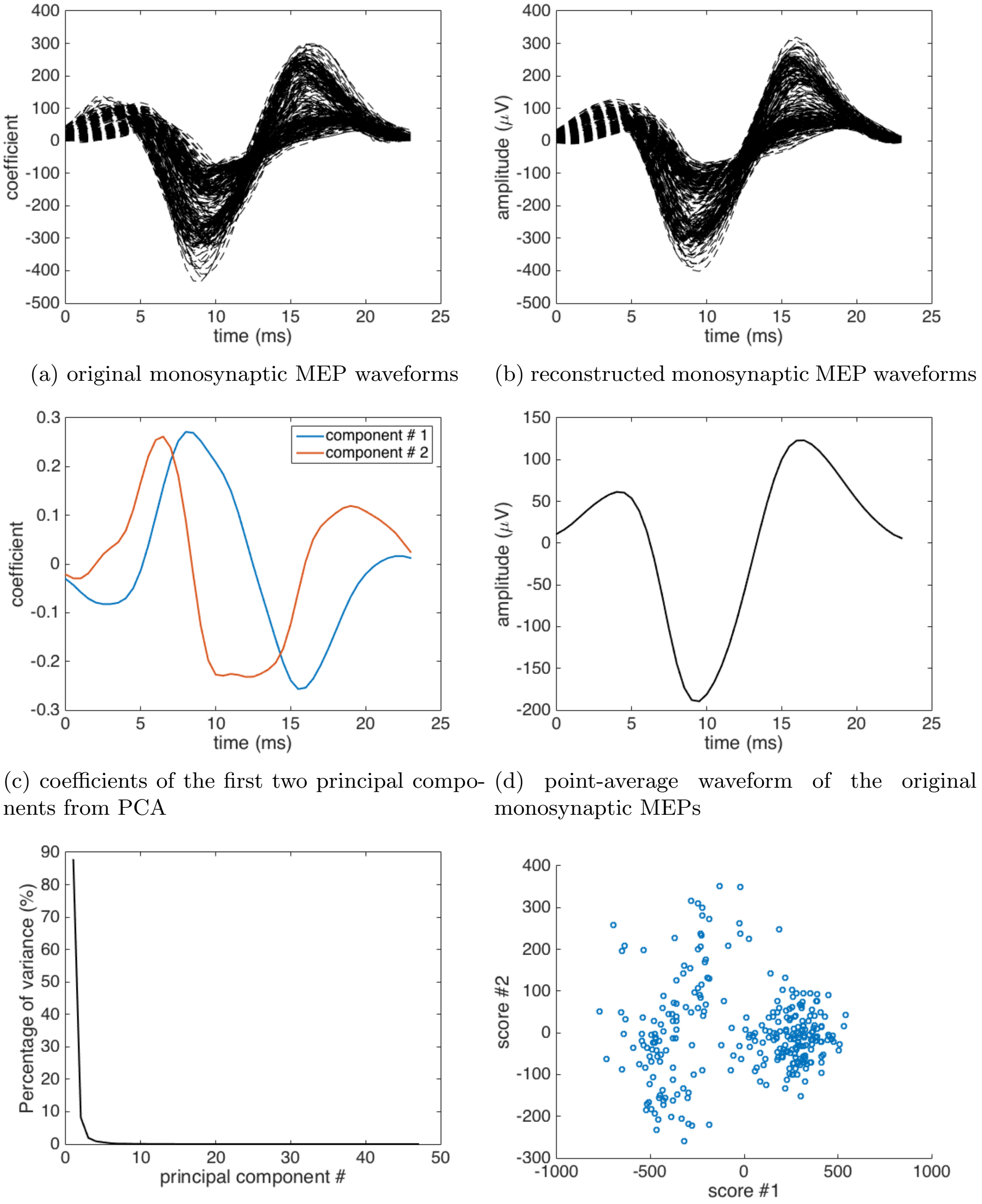
waveforms of two MEPs are correlated with each other. Normally, the alignment is performed by aligning the most significant feature of the waveform. For example, the largest peak of a MUAP is used as the alignment mark. This is not possible for MEPs because one MEP can have multiple significant peaks, and there is no single most significant peak. As a result, MEPs can't be aligned based on their peaks.

In this thesis, the alignment mark is chosen to be the weighted average of the time indices with weight being the amplitude square, as defined in Eq. (4.3.1.8). This definition removes the dependency on the peak locations of the MEPs, and therefore can be applied to MEPs with various shapes. In addition, this definition makes use of the shape information so that similar MEPs will be aligned correctly and consistently. After all the MEPs are aligned with the weighted time mean, the MEPs are adjusted to have the same length. The length is chosen to be the average of all MEPs. Hence, longer MEPs are truncated while shorter MEPs are padded with 0s.

$$\tau_i = \frac{\sum_{j=1}^{D_i} j * w_{i,j}^2}{\sum_{j=1}^{D_i} w_{i,j}^2} \tag{4.3.1.8}$$

where $\tau_i \in \{1, 2, \cdots, N\}$ is the alignment mark of the i-th MEP, of which the time indices are $\{1, 2, \cdots, D_i\}$, and the waveform is $\mathbf{w}_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,D_i}\}$.

To illustrate the PCA process applied to the monosynaptic MEPs, the MEPs in Figure 4.2-2 are used for demonstration. The original MEP waveforms, aligned with respect to their weighted time means, are plotted in Figure 4.3-1a. Figure 4.3-1e shows the percentage of the variance each principal component (eigenvector) explains. The percentage of variance is simply the ratio of the eigenvalue corresponding to each eigenvector over the sum of all eigenvalues. As shown in Figure 4.3-1e, the first two eigenvectors already explain more than 95% of the total variance, so two principal components are good enough to retain most of the useful information in the original waveforms (See Figure 4.3-1c). Please note that the PCA is performed on the centered data (eg. data with mean subtracted). To reconstruct the original data, the mean has to be added back (See Figure 4.3-1d). The projection of the centered, aligned monosynaptic MEPs onto the principal subspace of dimension 2 is plotted in Figure 4.3-1f. This shows a side benefit of PCA, which is to help visualize high-dimensional data. Finally, the reconstructed waveforms are plotted in Figure 4.3-1b. Compare it with Figure 4.3-1a, one can tell the reconstructed waveforms are very close to the original waveforms, and hence a two-dimensional subspace is enough to capture most of the useful information in the original high-dimensional space.



(a) original monosynaptic MEP waveforms

(b) reconstructed monosynaptic MEP waveforms

(c) coefficients of the first two principal components from PCA

(d) point-average waveform of the original monosynaptic MEPs

(e) percentage of the variance explained by every component from PCA

(f) projection of the original monosynaptic MEPs

Figure 4.3-1: PCA result on the 285 MEPs in the monosynaptic MEP cluster obtained from the initial clustering. (a) gives the original MEP waveforms aligned with respect to their weighted time means. The common length of all waveforms is 47 samples. (b) gives the MEP waveforms reconstructed from their PCA projections with a two-dimensional principal subspace. (c) gives the coefficients of the first two principal components. (d) gives the point-average waveform of the original MEP waveforms. (e) gives the percentage of the variance explained by every principal component. (f) gives the projection of the original MEP waveforms onto the two-dimensional principal subspace. Score 1 and score 2 give the coordinates in the principal subspace when projecting the original waveforms onto the first and second component, respectively.

4.3.2 Clustering with Gaussian Mixure Model

Clustering with Gaussian Mixture Model (GMM), or Mixtures of Gaussians, is a very popular and powerful technique. K-means clustering is actually a special case of GMM clustering. Unlike K-means clustering, which only works well if clusters of data form hyper-spheres, GMM can model clusters with hyper-ellipsoids of various orientations. In addition, K-means gives hard thresholding while GMM assigns the probability to every data point that measures the likelihood that each cluster explains the data. GMM has been used to cluster neural signals successfully [65]. A Gaussian distribution can accounts for the variability in the MEP waveforms. As a result, GMM has been used to cluster monosynaptic MEPs in this thesis.

A Gaussian Mixture Model is a probability distribution which is a linear superposition of Gaussian distributions with different weights:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.3.2.1}$$

where $\mathbf{x}$ is a random vector, π_k are called *mixing coefficients* and satisfy $\sum_{k=1}^{K} \pi_k = 1$, and $\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ gives the multivariate normal distribution with mean μ_k and covariance matrix Σ_k. The above GMM has K components. For convenience, let $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{K}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^{K}$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_k\}_{k=1}^{K}$. Given a fixed number of components K, the set of parameters of a GMM is therefore $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

The goal is to estimate the parameters of a GMM from a given set of observations $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$. The most common approach is to maximize the log of the likelihood function (commonly referred as Maximum Likelihood, or ML) with respect to the model parameter $\boldsymbol{\Theta}$:

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{4.3.2.2}$$

Maximizing the above function is a very complex task and does not yield a closed form solution. The most commonly used alternative approach for finding maximum likelihood solutions to GMM is called the *expectation-maximization* algorithm, or *EM* algorithm [4]. Various constraints can be put on the parameter $\boldsymbol{\Theta}$ when maximizing the likelihood function. The most common one is the free form optimization as described in [4], in which no constraints are put on the parameters. For covariance matrix, a parsimonious model can be used to apply different constraints, as discussed in details in [6]. For example, the covariance matrix $\boldsymbol{\Sigma}$ can be constrained to be diagonal. The parameters of all components can be constrained to be the same. Adding constraints to the model reduces the model complexity, and therefore reduces overfitting when the amount of data is small.

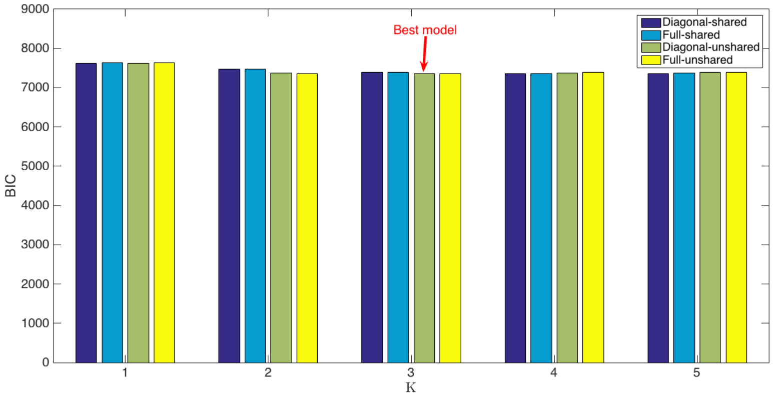


Figure 4.3-2: BIC for various K and Σ choices: Run clustering with GMM on the monosynaptic MEPs from Figure 4.2-2. The best model has 3 Gaussians ($K = 3$) with diagonal covariance matrix and non-shared parameters. The BIC associated with the best model is 7360, the smallest of all.

Model Class

In this thesis, the following models of Gaussian mixtures are considered:

- The number of Gaussian mixture components $K = 1$, 2, 3, 4, or 5.
- The covariance matrix $\mathbf{\Sigma}_k$ is diagonal or full.
- The parameters $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ are shared or not.

Overall there are $5 \cdot 2 \cdot 2 = 20$ models to choose from.

The more complex the model is, the better it fits the data, and the bigger the likelihood is. However, a complex model tends to overfit the data. As a result, a penalty has to be given based on the complexity of the model. *Bayesian information criterion* (BIC) is commonly used as a criterion for model selection among a finite set of models. The model with the lowest BIC is preferred. The BIC is defined as:

$$\mathrm{BIC} = -2\ln(\hat{L}) + M\ln(N) \tag{4.3.2.3}$$

where $\hat{L}$ is the maximized value of the likelihood function of the model, M is the number of free parameters to be estimated, and N is the number of data points. The BIC calculated under different models for the clustering of monosynaptic MEPs is shown in Figure 4.3-2. For that particular data set, the best model has three Gaussians, with all having diagonal covariance matrices and non-shared parameters.

After the model is identified from the data, the clustering is done by evaluating the *responsibility*,

γ_{nk}, that component k takes for explaining data $\mathbf{x}_n$, as defined by:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4.3.2.4}$$

$\mathbf{x}_n$ is assigned to cluster k if cluster k has the largest responsibility for explaining it. For a total of K clusters, the cluster index c_n of data $\mathbf{x}_n$ is given by:

$$c_n = \underset{k \in \{1,2,\cdots,K\}}{\arg\max}\ \gamma_{nk} \tag{4.3.2.5}$$

The clustering result is shown in Figure 4.3-3. From simple visual inspection, the clustering result looks good as it finds the subtle differences between different clusters of MEPs, and MEPs from every cluster share similar shapes. Of course, to accurately interpret the clustering result, one needs to consult neurophysiologists for expertise.

4.4 Decomposition of Overlapping MEPs

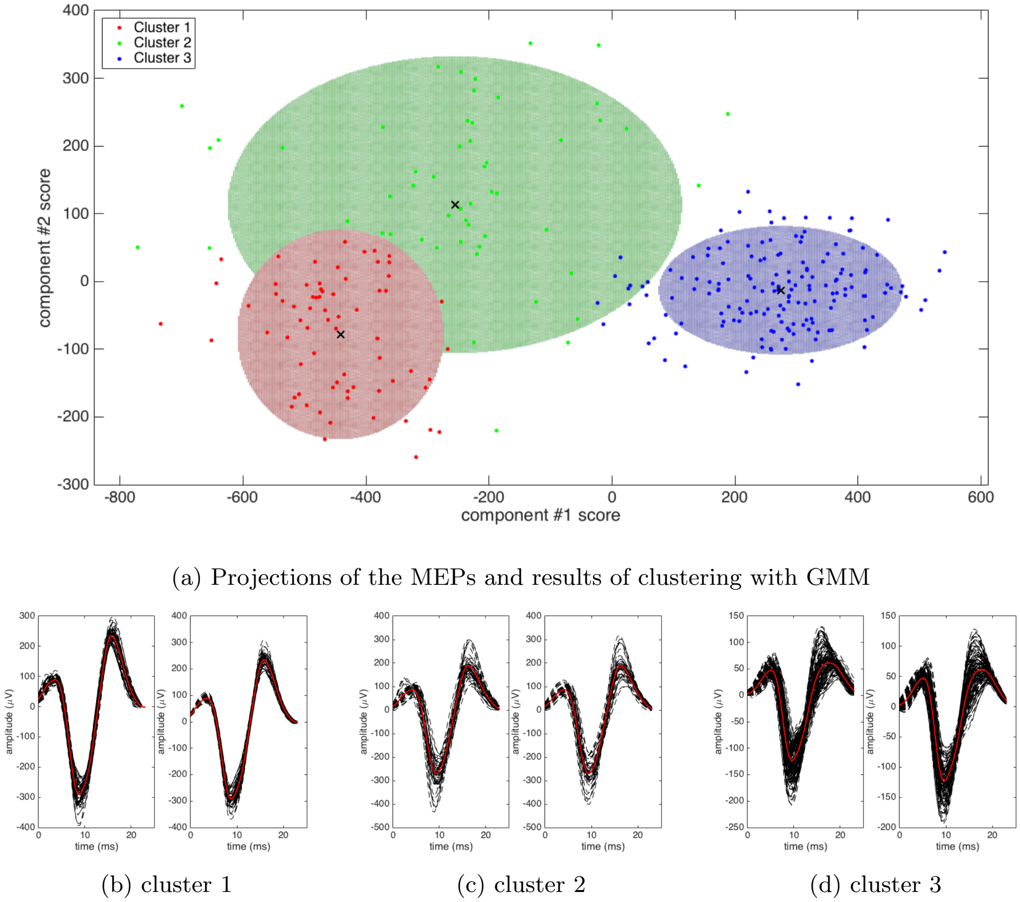
In the initial hierarchical clustering, the cutoff threshold is intentionally chosen to be small so that there is at least one cluster of only monosynaptic MEPs. As a result, some of the monosynaptic MEPs are "left out". This is especially true when a monosynaptic MEP is very close to, or even overlaps with, a polysynaptic MEP. In this case, the segmentation step produces a long segment with both monosynaptic MEPs and polysynaptic MEPs. The task in this step is to extract the remaining monosynaptic MEPs when two or more MEPs overlap in time.

The overall procedure is first stated, followed by a detailed discussion of every step. For every remaining MEP,

1. Find a segment of the waveform that gives the best matching with the cluster of monosynaptic MEPs. This segment is regarded as a candidate monosynaptic MEP.

2. Calculate the likelihood of given candidate MEP being from the clusters of monosynaptic MEPs using information of the shape and the latency. If the likelihood is above a certain threshold, the candidate MEP is labeled as a monosynaptic MEP.

3. Classify the newly added monosynaptic MEP using the Gaussian mixture model identified in Section 4.3.2.



(a) Projections of the MEPs and results of clustering with GMM

(b) cluster 1

(c) cluster 2

(d) cluster 3

Figure 4.3-3: Clustering result on monosynaptic MEPs with GMM: the monosynaptic MEPs are from Figure 4.2-2. PCA is performed to reduce the dimensionality before clustering (See Figure 4.3-1f). The GMM has three Gaussians with diagonal covariance matrices. There are 285 MEPs in total: cluster 1 has 163 MEPs; cluster 2 has 51 MEPs; cluster 3 has 71 MEPs. (a) gives the scatter plot of the projections of the MEPs onto a two-dimensional principal subspace. Each projection is color-coded to give its membership. For every component, the Gaussian distribution is also plotted with mean given by cross (X) and covariance given by ellipsoids that specifies a 99% probability threshold for confidence region; (b) - (d) Monosynaptic MEPs in Cluster 1 - 3 with original waveforms on the left and reconstructed waveforms on the right. The points average is plotted (in red) within each group of waveforms.

4.4.1 Identify Candidate Monosynaptic MEPs

The waveform of a remaining MEP is denoted as a vector of length D, $\mathbf{w} = [w_1 \; w_2 \; \cdots \; w_D]^T$. From the means of the K components in the GMM, cluster template waveforms can be reconstructed by Eq. (4.3.1.5). Denote the K cluster template waveforms as $\mathbf{w}^{(k)}$, where $k = 1, 2, \cdots, K$, and:

$$\mathbf{w}^{(k)} = \mathbf{U}\boldsymbol{\mu}_k$$

where $\boldsymbol{\mu}_k$ is the mean of the k-th component in GMM, and $\mathbf{U}$ is the base of principal subspace. Note that all vectors $\mathbf{w}^{(k)}$ and $\boldsymbol{\mu}_k$ are column vectors.

Cross-correlate $\mathbf{w}$ with $\mathbf{w}^{(k)}$ and find the best matching point, n_k, the point where the cross-correlation coefficient takes its maximum value, ρ_k.

$$n_k = \underset{n \in \mathbb{Z}}{\arg\max}\, (\mathbf{w} \star \mathbf{w}^{(k)})[n] \tag{4.4.1.1}$$

$$\rho_k = (\mathbf{w} \star \mathbf{w}^{(k)})[n_k] \tag{4.4.1.2}$$

where $(f \star g)[n]$ is the cross-correlation between two real-valued discrete signals $f[n]$ and $g[n]$:

$$(f \star g)[n] = \sum_{m=-\infty}^{+\infty} f[m+n]g[m] \tag{4.4.1.3}$$

which gives the dot product of $f[m]$ with lag n and $g[m]$.

Then find the component $\hat{k}$ that gives the maximum value of ρ_k: $\hat{k} = \arg\max_{k=1,2,\cdots,K} \rho_k$. Suppose the length of the aligned monosynaptic MEPs is $\bar{D}$, then the candidate monosynaptic MEP is $\mathbf{w}' = \{w_j\}_{j=n_{\hat{k}}}^{n_{\hat{k}}+\bar{D}-1}$. The other segments of $\mathbf{w}$ might contain polysynaptic MEPs, so create new MEP segments, $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$:

$$\mathbf{w}^{(1)} = \{w_j\}_{j=n_1}^{n_{\hat{k}}-1} \tag{4.4.1.4}$$

$$\mathbf{w}^{(2)} = \{w_j\}_{j=n_{\hat{k}}+\bar{D}}^{D} \tag{4.4.1.5}$$

where $\mathbf{w}^{(1)}$ is the segment before $\mathbf{w}'$, while $\mathbf{w}^{(2)}$ is the segment after $\mathbf{w}'$. If there are none or very few samples on either sides, then the candidate intervals $\mathbf{w}^{(1)}$ or $\mathbf{w}^{(2)}$ are not meaningful. Figure 4.4-1 shows an example of breaking down a long segment into a candidate monosynaptic MEP and a candidate polysynaptic MEP. In that case, no segments exist before $\mathbf{w}'$, and only one segment exists after $\mathbf{w}'$. In this example, $\mathbf{w}'$ is successfully accepted to the cluster of monosynaptic MEPs in next

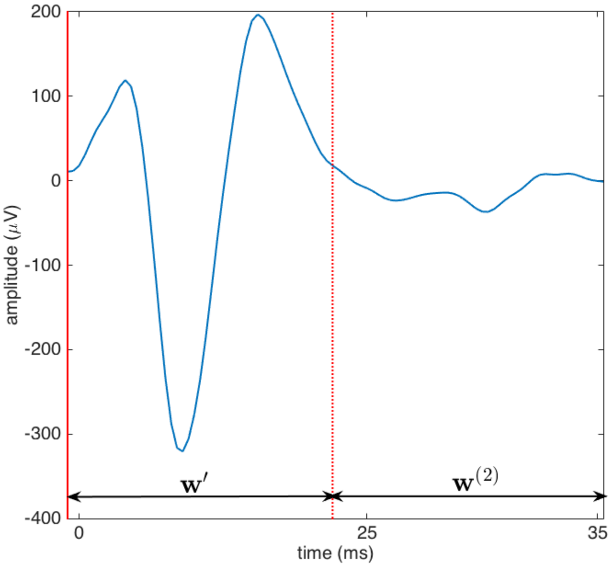


Figure 4.4-1: Breaking segment into candidate monosynaptic and polysynaptic MEP: $\mathbf{w}'$ is the candidate monosynaptic MEP found by cross-correlation. There is not a segment before $\mathbf{w}'$. $\mathbf{w}^{(2)}$ is the remaining segment after $\mathbf{w}'$, and will be analyzed as a potential polysynaptic MEP: red solid line and red dotted line indicate the location of $n_{\hat{k}}$ and $n_{\hat{k}} + \bar{D} - 1$, the beginning and end of the candidate monosynaptic MEP $\mathbf{w}'$, respectively

step.

4.4.2 Determine if the Candidate is a Monosynaptic MEP

After a segment of candidate MEP is identified, the likelihood of it being from the monosynaptic MEP cluster needs to be quantitatively evaluated. The Gaussian mixture model can give the likelihood based on the waveform of the MEP, but it does not use the latency information, which is crucial in determining the type of the MEP. In the following, the waveform and the latency information are combined to give a quantitative measurement of the likelihood.

For convenience, some of the notations introduced in Section 4.1 are restated here, together with some new notations. An EMG signal takes the format of N samples (discrete-time signal): $\mathbf{X} = \{x_1, x_2, \cdots, x_N\}$. Suppose the cluster of monosynaptic MEPs contains N_M MEPs, $\mathcal{W}_{MEP} = \{\mathbf{w}_i\}_{i=1}^{N_M}$, where $\mathbf{w}_i$ is the waveform of the i-th MEP of length D_i: $\mathbf{w}_i = w_{i,1}, w_{i,2}, \cdots, w_{i,D_i}$. The timing of the stimulus which is presumed to elicit the response in $\mathbf{w}_i$ is denoted as $a_i \in \{1, 2, \cdots, N\}$ that represents the (discrete-time) index in the EMG signal $\mathbf{X}$. The latency of the i-th MEP, which is the time from the occurrence of the electrical stimulus (a_i) to the MEP ($\mathbf{w}_i$), is denoted as d_i. The complete information about the i-th MEP is thus $\mathcal{M}_i = \{\mathbf{w}_i, a_i, d_i\}$, which includes the waveform, the associated stimulus, and its latency to the stimulus.

The probability model for the waveform can be given by the Gaussian mixture model, $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. The probability model for the latency is given by the Gaussian distribution, $\boldsymbol{\Theta}_d = \{\mu_d, \sigma_d^2\}$ where subscript d indicates latency. The principal subspace of the MEP waveform space (the Euclidean space with the dimension of the common length of the MEP waveforms) is given by $\mathbf{U}$ (Refer to Section 4.3.1 for feature extraction via PCA). Assume the waveform is independent of the latency, and then the probability of a given MEP being a monosynaptic MEP is:

$$p(\mathbf{w}_i, d_i; \boldsymbol{\Theta}, \boldsymbol{\Theta}_d, \mathbf{U}) = p(\mathbf{U}^T \mathbf{w}_i | \boldsymbol{\Theta}) \cdot \mathcal{N}(d_i | \mu_d, \sigma_d^2) \tag{4.4.2.1}$$

where $p(\mathbf{U}^T \mathbf{w}_i | \boldsymbol{\Theta})$ is the Gaussian mixture model given by Eq. (4.3.2.1). Then the log likelihood is:

$$\ln \left\{ p(\mathbf{w}_i, d_i; \boldsymbol{\Theta}, \boldsymbol{\Theta}_d, \mathbf{U}) \right\} = \ln \left\{ p(\mathbf{U}^T \mathbf{w}_i | \boldsymbol{\Theta}) \right\} + \ln \left\{ \mathcal{N}(d_i | \mu_d, \sigma_d^2) \right\} \tag{4.4.2.2}$$

The parameters of the Gaussian distribution of the latency $\boldsymbol{\Theta}_d$ are estimated by:

$$\hat{\mu}_d = \frac{1}{N_M} \sum_{i=1}^{N_M} d_i \tag{4.4.2.3}$$

$$\hat{\sigma}_d = \sqrt{\frac{1}{N_M - 1} \sum_{i=1}^{N_M} (d_i - \hat{\mu}_d)^2} \tag{4.4.2.4}$$

which gives the unbiased estimation of the population mean and the population variance.

The threshold for accepting a given MEP as a monosynaptic MEP is given by:

$$T = \gamma \left(\min \left\{ \ln \left\{ p(\mathbf{U}^T \mathbf{w}_i | \boldsymbol{\Theta}) \right\} \right\} + \min \left\{ \ln \left\{ \mathcal{N}(d_i | \mu_d, \sigma_d^2) \right\} \right\} \right) \tag{4.4.2.5}$$

where γ is a factor that can be adjusted to either tighten or relax the threshold. For example, in the thesis, $\gamma = 1.5$ is found to yield good results. Note that the log likelihood is a negative number, so a $\gamma > 1$ relaxes the threshold.

For a given new MEP $\mathcal{M}_{new} = \{\mathbf{w}_{new}, a_{new}, d_{new}\}$, $\mathcal{M}_{new}$ is said to be monosynaptic MEP if:

$$\ln \left\{ p(\mathbf{w}_{new}, d_{new}; \boldsymbol{\Theta}, \boldsymbol{\Theta}_d, \mathbf{U}) \right\} \geq T \tag{4.4.2.6}$$

4.4.3 Classify the New Monosynaptic MEP

The newly added monosynaptic MEP, $\mathbf{w}_{new}$, is classified to one of the K clusters determined by the K components of Gaussian mixtures. The posterior probability of the given MEP being from the k-th component is γ_k:

$$\gamma_k = \frac{\pi_k \mathcal{N}(\mathbf{U}^T \mathbf{w}_{new} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{U}^T \mathbf{w}_{new} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4.4.3.1}$$

This is the adaptation of Eq. (4.3.2.4) by replacing $\mathbf{x}_n$ with the PCA projection of the new MEP waveform, $\mathbf{U}^T \mathbf{w}_{new}$.

$\mathbf{w}_{new}$ is assigned to the component with the largest posterior probability for the observation.

$$\hat{k} = \underset{k \in \{1,2,\cdots,K\}}{\arg\max} \; \gamma_k \tag{4.4.3.2}$$

In Section 4.1, 469 MEPs were detected and segmented from 30 seconds of the EMG signal which was recorded from left medial gastrocnemius (L MG) of a patient with clinically motor complete spinal cord injuries while lying in the supine position with EES. The stimulation amplitude is 7.2V and frequency is 10Hz. Among the 469 MEPs, 285 MEPs were placed to the monosynaptic MEP cluster in the initial clustering in Section 4.2.2. After running the above decomposition procedure on the remaining 184 MEPs, additional 15 MEPs were placed to the monosynaptic MEP cluster, so the monosynaptic MEP cluster has 300 MEPs. This is expected as there are 300 stimulation intervals from 30 seconds of the EMG signal with stimulation frequency being 10Hz, and normally there is one monosynaptic MEP within one stimulation interval. The remaining 184 MEPs are considered candidate polysynaptic MEPs and will be clustered via GMM in the next section.

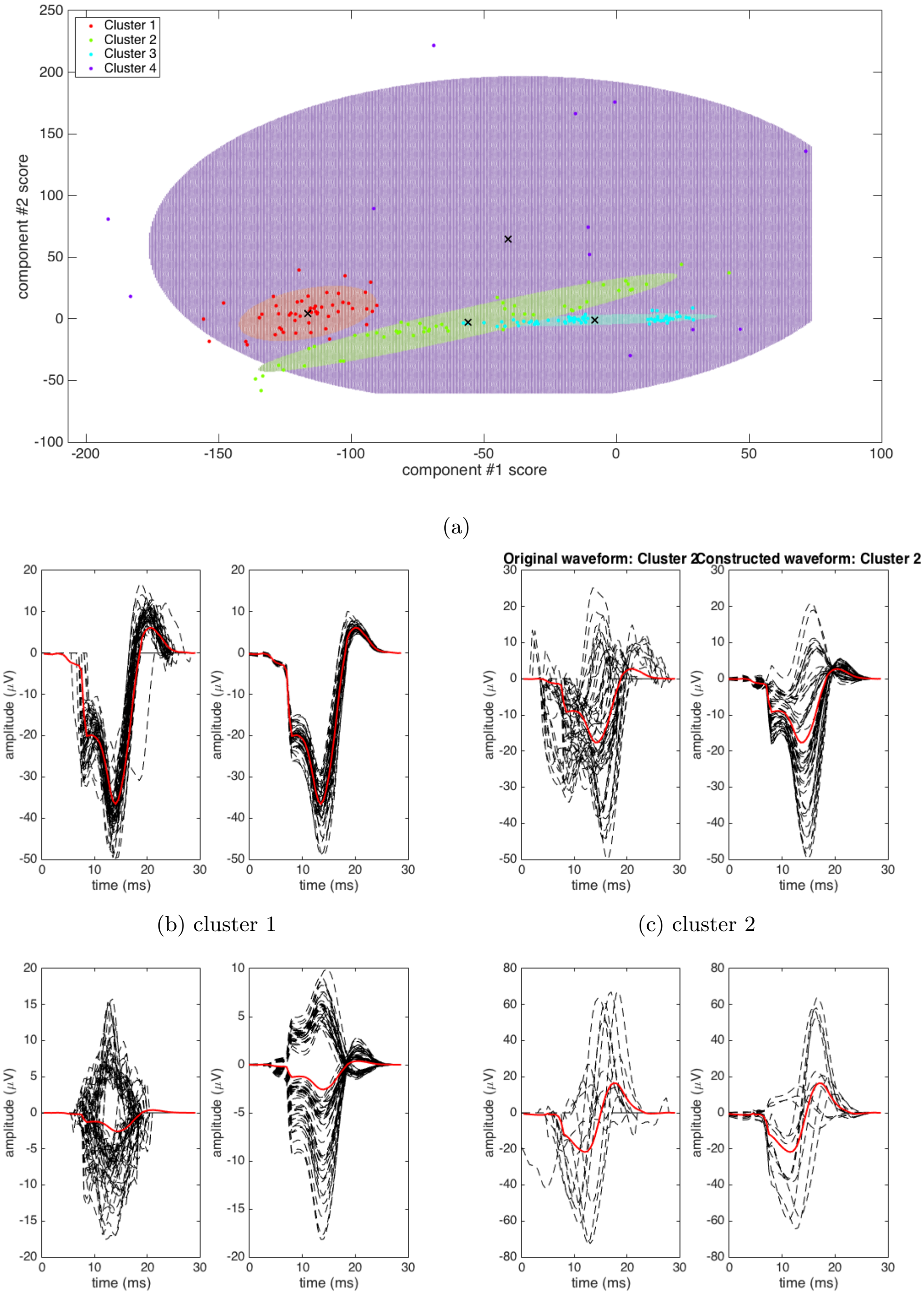
4.5 Clustering of Polysynaptic MEPs

Clustering of polysynaptic MEPs undergoes the same set of procedures as the monosynaptic MEPs as described in Section 4.3. Here is a brief recap of the procedures.

1. Align the MEP waveforms based on its weighted time mean. Run PCA and project waveform onto the principal subspace (with dimensionality of 2).
2. Perform maximum likelihood (ML) optimization on a set of Gaussian mixture models via EM algorithm. Select the best model.
3. Cluster the PCA projections of the MEPs using the selected Gaussian mixture model.

An example clustering result is shown in Figure 4.5-1. PCA is performed to reduce the dimensionality before clustering. The best model has four Gaussians with full covariance matrices. There are 184 polysynaptic MEPs in total: cluster 1 has 52 MEPs; cluster 2 has 52 MEPs; cluster 3 has 68 MEPs; cluster 4 has 12 MEPs.

Although polysynaptic MEPs are very weak, and do not have consistent latencies or shapes even within an event, the clustering successfully partitions them into different groups (See Figure 4.5-1). Because the SNR of the polysynaptic MEPs is low, the real MEP waveforms are tempered by



(a)

(b) cluster 1

(c) cluster 2

(d) cluster 3

(e) cluster 4

Figure 4.5-1: Clustering result on polysynaptic MEPs with GMM: the polysynaptic MEPs are from the same EMG signal as the monosynaptic MEPs in Figure 4.2-2. (a) gives the scatter plot of the projections of the MEPs onto a two-dimensional principal subspace. Each projection is color-coded to give its membership. For every component, the Gaussian distribution is also plotted with mean given by cross (X) and covariance given by ellipsoids that specifies a 99% probability threshold for confidence region; (b) - (d) Monosynaptic MEPs in Cluster 1 - 4 with original waveforms on the left and reconstructed waveforms on the right. The points average is plotted (in red) within each group of waveforms.

noise severely, which make them visually hard to analyze by physiologists. The use of PCA not only reduces the feature dimensionality, but also reduces the effect of the noise. In Figure 4.5-1b to Figure 4.5-1e, the reconstructed waveforms are shown on the right hand side of the original waveforms. The reconstructed waveforms are much smoother, and the structure or shape of the MEP waveforms is more prominent.

The clustering result is especially satisfying considering that these clusters don't visually separate from each other in the scatter plot. As a result, the K-means clustering would obtain completely wrong results. This example shows the flexibility and powerfulness of the GMM.

Because polysynaptic MEPs are weak, there are more false positives in detecting transient peaks of the polysynaptic MEPs. The error in the peak detection will be carried over to the segmentation phase and eventually the clustering phase. The threshold in the peak detection step is intentionally chosen to be small enough to yield a high recall, but a low precision. A high recall guarantees that most of the true MEPs are detected. A low precision implies more noise samples are picked up as MEPs, but this can be corrected in the following clustering phase, because noise samples have similar statistics and tend to show up in one cluster. The cluster of noise samples can be removed upon inspection by the physiologists.

Chapter 5

Conclusion

5.1 Summary of Thesis Contributions

The primary contribution of this thesis is the development of a set of automatic, unsupervised tools for the analysis of Electromyogram or the EMG signal for the purpose of studying electrical stimulation based rehabilitation on patients with spinal cord injuries. A wavelet-based, double-threshold algorithm was developed for the detection of transient peaks in the EMG signal (Chapter 3). Based on the transient peak detection result, EMG signals are further segmented and classified into various groups of monosynaptic MEPs and polysynaptic MEPs using techniques stemming from Principal Component Analysis (PCA), hierarchical clustering, and Gaussian mixture model (Chapter 4). A software with graphic user interface has been implemented in Matlab. The software implements the proposed peak detection algorithm, and enables the physiologists to visualize the detection results and modify them if necessary.

Although there exist many different sets of tools to analyze the EMG signal, most of them rely greatly on the human supervision. One significant aspect of the contributions of this thesis is that all the proposed analysis methods are completely automatic and unsupervised. This is particularly important when the amount of data is huge, or real-time processing is desired. The EMG signal of interest in this thesis was recorded from patients with spinal cord injuries during the rehabilitation under electrical stimulation. As a result, the EMG signal is very different in nature from most of the signals in the EMG community. The EMG signal dealt with in this thesis has more complicated shapes, and the shape information is not a known priori. As a result, the set of methods developed in this thesis made no assumptions on the shapes of the signals, and therefore can be applied to any generic transient signals, as long as the transient signals are composed of peaks, which is the case in most practical systems, such as ECG signals, and mass spectrum.

Chapter 3 extends existing theories in the transient detection field. The application of wavelet transform in the detection of transient signals has been studied extensively and employed successfully. However, most of the theories assumes certain knowledge about the shapes of the transient signals, which makes it hard to be generalized to transient signals with arbitrary shapes. The proposed detection scheme focuses on the more fundamental feature of most transient signals (in particular the EMG signal) – peaks, instead of the shapes. The continuous wavelet transform with Mexican Hat wavelet is employed. This thesis theoretically derived a framework for selecting a set of scales based on the frequency domain information. Ridges are identified in the time-scale space to combine the wavelet coefficients from different scales. By imposing two thresholds, one on the wavelet coefficient and one on the ridge length, the proposed detection scheme can achieve both high recall and high precision. A systematic approach for selecting optimal parameters via simulation is proposed and demonstrated. Comparing with other state-of-the-art detection methods, the proposed method in this thesis yields a better detection performance, especially in the low Signal-to-Noise-Ratio (SNR) environment.

In Chapter 4, a method for automatically segmenting and clustering the detected EMG signal is derived. A theoretical framework is proposed to segment the EMG signal based on the detected peaks. The scale information of the detected peaks is used to derive a measure for its effective support. Several different techniques have been adapted together to solve the clustering problem. An initial hierarchical clustering is first performed to obtain most of the monosynaptic Motor Evoked Potentials (MEPs). Principal component analysis (PCA) is used to reduce the number of features and effect of the noise. The reduced feature set is then fed to a Gaussian mixture model (GMM) to further divide the MEPs into different groups of similar shapes. The method of breaking down a segment of multiple consecutive MEPs into individual MEPs is derived.

In order to make the processing completely unsupervised, the statistics of the underlying noise must be estimated automatically and accurately. Assume the noise is White Gaussian Noise (WGN), from robust statistics, the variance of the noise can be estimated if the signals are outliers. In the case that signals are not outliers, the performance deteriorates rapidly. An iterative algorithm has been proposed to improve the accuracy of the estimation even when the signals are not outliers, and simulated experiments show great boost in the accuracy.

5.2 Opportunities for Future Work

The key difficulty in processing the EMG signal is the complex structure of the transient signals (MEPs in the case of the EMG signal in this thesis). The lack of prior knowledge on the shape of the transient signals makes the detection and further processing very difficult. In this thesis, transient signals are modeled as consecutive transient peaks. Based on this model, detection and segmentation methods are derived. In the future work, better representation of the transient signals can be explored and employed to formulate new detection and segmentation scheme. For example, Le's proposed idea of using L-spline functions to model transient signals is promising [29].

The parameters in the proposed peak detection method are selected based on simulated experiments. In the future work, the parameters can be theoretically derived. In the segmentation, the effective support of the mother wavelet is chosen to be the width of the its most significant peak. Other choices may yield better results, and is subject to more theoretical development. A Bayesian clustering can be derived in order to incorporate the prior knowledge for better clustering results. Lastly, the noise is assumed to be additive white Gaussian. Other noise models may be explored.
The proposed peak detection method is formulated as a generic detector with little assumptions made on the transient signals. As a result, the work in this thesis can potentially be applied to any transient signals that are composed of transient peaks, especially biological signals such as the electrocardiogram (ECG).

Appendix A

Muscle Names

Throughout the thesis, short names of different muscles are used to represent the EMG channels. In the short names, “L” means *left* while “R” means *right*. For every type of muscle, EMG from both the left and right ones are recorded. Every muscle is represented by 2 or 3 letters, as shown in Table A.1

<table>
<tr><td colspan="2">Commonly Acquired EMG during supine, sitting, standing and stepping experiments</td></tr>
<tr><td>SOL</td><td>soleus</td></tr>
<tr><td>MG</td><td>medial gastrocnemius</td></tr>
<tr><td>TA</td><td>tibialis anterior</td></tr>
<tr><td>MH</td><td>medial hamstring</td></tr>
<tr><td>VL</td><td>vastus lateralis</td></tr>
<tr><td>RF</td><td>rectus femoris</td></tr>
<tr><td>GL</td><td>gluteus maximus</td></tr>
<tr><td>GM</td><td>gluteus medius</td></tr>
<tr><td>AD</td><td>adductor</td></tr>
<tr><td colspan="2">Additional Acquired EMG during voluntary control experiments</td></tr>
<tr><td>EDL</td><td>extensor digitorum longus</td></tr>
<tr><td>EHL</td><td>extensor hallucis longus</td></tr>
<tr><td>IC</td><td>intercostals</td></tr>
<tr><td colspan="2">Acquired EMG for stimulation artifact</td></tr>
<tr><td>PS</td><td>paraspinals</td></tr>
</table>

Table A.1: Commonly Acquired EMG channels

Appendix B

EMG Peak Detection Software

To help the practitioners at University of Louisville at Kentucky to process the EMG signal for the spinal cord injuries research, a MATLAB program with graphic user interface (GUI) is developed. It is called *EmgPackage*, and its core functionality is to detect EMG peaks reliably and accurately by implementing the peak detection algorithm described in Chapter 3. It also allows the users to adjust the parameters to yield the best results for their specific data and applications. Since no detection algorithm is perfect and there will be missed detections or false positives sometimes, the software allows the users to manually adjust the detection results by adding or deleting peaks.

Figure B.0-1 shows the interface when EmgPackage is first opened. There are mainly five sections:

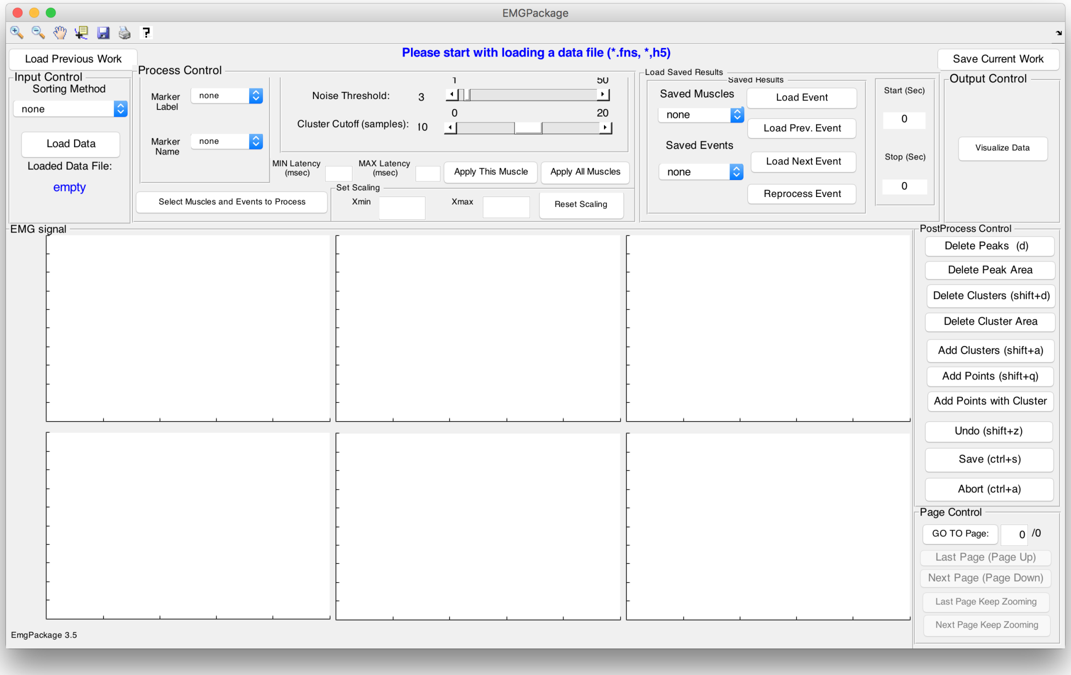
Input Control Select data for processing

Process Control Select muscles and stimulation events, and algorithm-related parameters. The default values of the parameters already give the best overall performance, but if the users particularly want a low false positive rate or a low false negative rate, then parameters can be adjusted to yield the desired performance.

Load Results After processing, the peak detection results are saved and can be loaded and further modified.

Postprocessing Control Users can manually modify the peak detection results by adding or deleting a single peak or a group of similar peaks.

EMG Signal Six consecutive stimulation intervals are shown on each page (starting at the upper left one and goes horizontally to the upper right one, which is followed by the lower left and ending at the lower right one). Users can go to the next or the previous page, or jump to any specific page. Every of the six sub-windows shows one stimulation interval, and the horizontal axis represents the latency. The red circles show the detected peaks. The numbers next to them are their cluster numbers.



Please start with loading a data file (*.fns, *,h5)

Sorting Method: none

Loaded Data File: empty

Marker Label: none

Marker Name: none

Noise Threshold: 3 (1–50)

Cluster Cutoff (samples): 10 (0–20)

Saved Muscles: none

Saved Events: none

Start (Sec): 0

Stop (Sec): 0

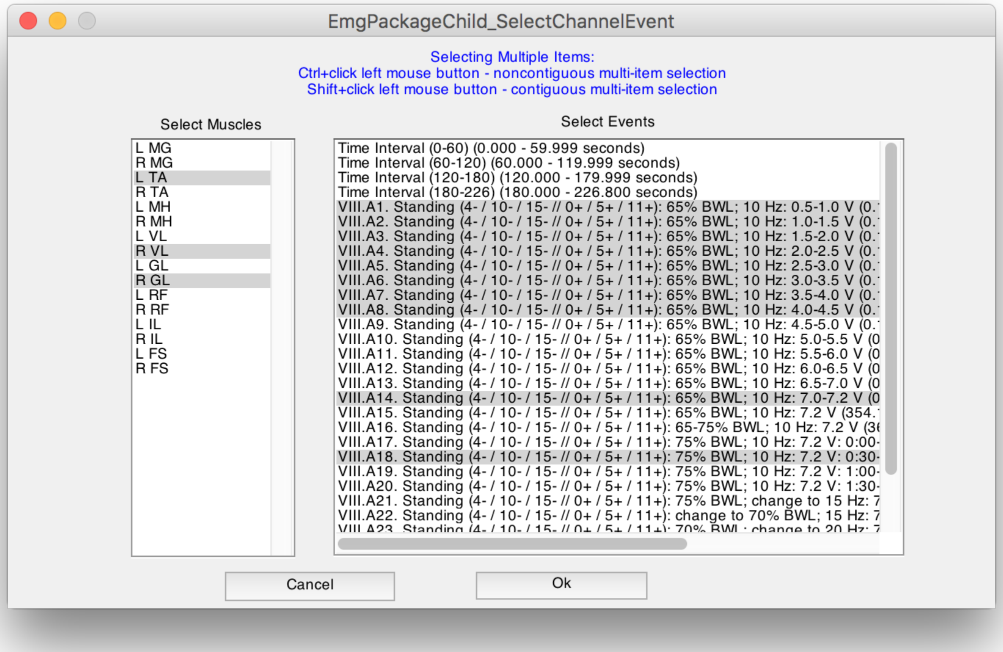
GO TO Page: 0 /0

Figure B.0-1: EmgPackage's main GUI when it's first opened.

The detected peaks are clustered based on their latencies using hierarchical clustering. The idea is that many of the similar peaks across different stimulation intervals have similar latencies. By grouping similar peaks together, users can modify a group of similar peaks instead of one by one. This can greatly reduce their work load.

Figure B.0-2 shows the sub-GUI when users click the button "Select Muscles and Events to Process". Users can select arbitrary muscles and stimulation events that they are interested in for peak detection.

Figure B.0-3 shows an example of the peak detection results.

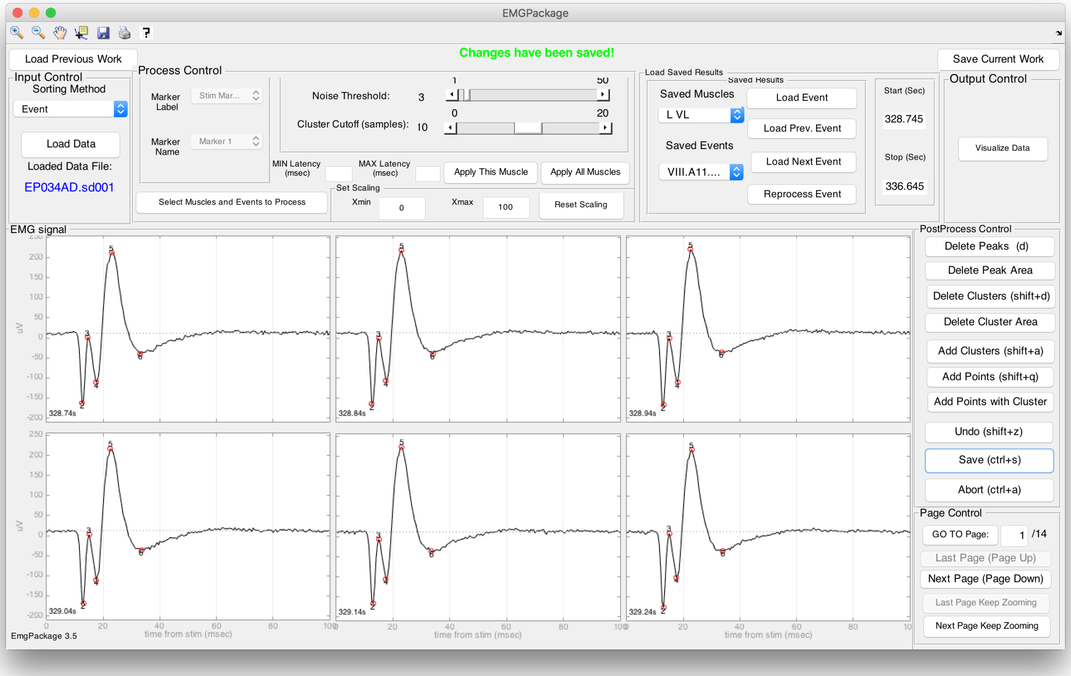


Selecting Multiple Items:
Ctrl+click left mouse button - noncontiguous multi-item selection
Shift+click left mouse button - contiguous multi-item selection

```
L MG
R MG
L TA
R TA
L MH
R MH
L VL
R VL
L GL
R GL
L RF
R RF
L IL
R IL
L FS
R FS
```

```
Time Interval (0-60) (0.000 - 59.999 seconds)
Time Interval (60-120) (60.000 - 119.999 seconds)
Time Interval (120-180) (120.000 - 179.999 seconds)
Time Interval (180-226) (180.000 - 226.800 seconds)
VIII.A1. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 0.5-1.0 V (0.
VIII.A2. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 1.0-1.5 V (0.
VIII.A3. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 1.5-2.0 V (0.
VIII.A4. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 2.0-2.5 V (0.
VIII.A5. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 2.5-3.0 V (0.
VIII.A6. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 3.0-3.5 V (0.
VIII.A7. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 3.5-4.0 V (0.
VIII.A8. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 4.0-4.5 V (0.
VIII.A9. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 4.5-5.0 V (0.
VIII.A10. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 5.0-5.5 V (0
VIII.A11. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 5.5-6.0 V (0
VIII.A12. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 6.0-6.5 V (0
VIII.A13. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 6.5-7.0 V (0
VIII.A14. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 7.0-7.2 V (0
VIII.A15. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65% BWL; 10 Hz: 7.2 V (354.
VIII.A16. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 65-75% BWL; 10 Hz: 7.2 V (3
VIII.A17. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 75% BWL; 10 Hz: 7.2 V: 0:00-
VIII.A18. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 75% BWL; 10 Hz: 7.2 V: 0:30-
VIII.A19. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 75% BWL; 10 Hz: 7.2 V: 1:00-
VIII.A20. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 75% BWL; 10 Hz: 7.2 V: 1:30-
VIII.A21. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 75% BWL; change to 15 Hz: 7
VIII.A22. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): change to 70% BWL; 15 Hz: 7
VIII.A23. Standing (4- / 10- / 15- // 0+ / 5+ / 11+): 70% BWL; change to 20 Hz: 7
```

Figure B.0-2: Sub-GUI for selection of different muscles and stimulation events



Changes have been saved!

Sorting Method: Event

Loaded Data File: EP034AD.sd001

Noise Threshold: 3 (1–50)

Cluster Cutoff (samples): 10 (0–20)

Xmin: 0

Xmax: 100

Saved Muscles: L VL

Saved Events: VIII.A11....

Start (Sec): 328.745

Stop (Sec): 336.645

GO TO Page: 1 /14

Figure B.0-3: An example of peak detection results from using EmgPackage: every one of the six sub-plots is one stimulation interval with the horizontal axis being the latency. The beginning of every sub-plot is the occurrence of one stimulus. The red circles show the detected peaks. The numbers next them indicate their cluster numbers.

Bibliography

[1] Douglas A. Abraham. A Page test with nuisance parameter estimation. *IEEE transactions on information theory*, 42(6):2242–2252, 1996.

[2] Michele Basseville and Igor V. Nikiforov. *Detection of Abrupt Changes: Theory and Application.* Prentice Hall information and system sciences series. Prentice Hall, April 1993.

[3] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE transactions on speech and audio*, 13(5):1035–1047, 2005.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 1 edition, October 2007.

[5] Bruce Broder and Stuart Schwartz. Quickest detection procedures and transient signal detection. Technical Report 21, Information Science & Systems Laboratory, Princeton University, Princeton, NJ 08544, June 1990.

[6] Gilles Celeux and G. Govaert. Gaussian parsimonious clustering models. Technical report, INRIA, 1993.

[7] E. Chauvet, O. Fokapu, J. Y. Hogrel, D. Gamet, and J. Duchene. Automatic identification of motor unit action potential trains from electromyographic signals using fuzzy techniques. *Medical & Biological Engineering & Computing*, 41:646–653, 2003.

[8] Christodoulos I. Christodoulou and Constantinos S. Pattichis. Unsupervided pattern recognition for the classification of EMG signals. *IEEE Transactions on Biomedical Engineering*, 46(2):169–178, 1999.

[9] Christopher & Dana Reeve Foundation. Paralysis and Spinal Cord Injury in the United States, 2015. [Online; accessed 16-July-2015].

[10] Scott Day. Important factors in surface EMG measurement.

[11] Thomas Desautels, Andreas Krause, and Joel Burdick. Parallelizing exploration–exploitation trade- offs with Gaussian process bandit optimization. In *29th International Conference on Machine Learning*, 2012.

[12] V. Dietz and Susan J. Harkema. Locomotor activity in spinal cord-injured persons. *Journal of Applied Physiology*, 96:1954–1960, 2004.

[13] Pan Du, Warren A. Kibbe, and Simon M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006.

[14] Gene Dwyer, Yasuaki Noguchi, and Hazel H. Szeto. Emg burst waveform recognition procedure. In *Bioengineering Conference*, 1989.

[15] Jianjun Fang, Gyan C. Agarwal, and Bhagwan T. Shahani. Decomposition of multiunit electromyographic signals. *IEEE transactions on biomedical engineering*, 46(6):685–697, 1999.

[16] E. Fishler and H. Messer. Detection and parameter estimation of a transient signal using order statistics. *IEEE transactions on signal processing*, 48(5):1455–1458, 2000.

[17] Benjamin Friedlander and Boaz Porat. Detection of transient signals by the Gabor representation. *IEEE transactions on acoustics, speech, and signal processing*, 37(2):169–180, 1989.

[18] Benjamin Friedlander and Boaz Porat. Performance analysis of transient detectors based on a class of linear data transforms. *IEEE transactions on information theory*, 38(2):665–673, 1992.

[19] Mordechai Frisch and Hagit Messer. The use of the wavelet transform in the detection of an unknown transient signal. *IEEE transactions on information theory*, 38(2):892–897, 1992.

[20] Mordechai Frisch and Hagit Messer. Transient signal detection using prior information in the likelihood ratio test. *IEEE transactions on signal processing*, 41(6):2177–2192, 1993.

[21] Mordechai Frisch and Hagit Messer. Detection of a known transient signal of unknown scaling and arrival time. *IEEE transactions on signal processing*, 42(7):1859–1863, 1994.

[22] Andreas Gerber, Roland M. Studer, Rui J. P. De Figueiredo, and George S. Moschytz. A new framework and computer program for quantitative emg signal analysis. *IEEE transactions on biomedical engineering*, BME-31(12):857–863, 1984.

[23] Chunming Han, Peter K. Willett, and Douglas A. Abraham. Some methods to evaluate the performance of Page's test as used to detect transient signals. *IEEE transactions on signal processing*, 47(8):2112–2127, 1999.

[24] Susan Harkema, Yury Gerasimenko, Jonathan Hodes, Joel Burdick, Claudia Angeli, Yangsheng Chen, Christie Ferreira, Andrea Willhite, Enrico Rejc, Robert G Grossman, and V Reggie Edgerton. Effect of epidural stimulation of the lumbosacral spinal cord on voluntary movement, standing, and assisted stepping after motor complete paraplegia: a case study. *Lancet*, 377:1938–1947, 2011.

[25] Valentin T. Jordanov, Dave L. Hall, and Mat. Kastner. Digital peak detector with noise threshold. In *IEEE Nuclear Science Symposium Conference Record*, 2002.

[26] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill, fourth edition, 2000.

[27] C.D. Katsis, Y. Goletsis, A. Likas, D.I. Fotiadis, and I. Sarmas. A novel method for automated EMG decomposition and MUAP classification. *Artificial Intelligence in Medicine*, 37:55–64, 2006.

[28] Steven Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Prentice Hall, 1 edition, February 1998.

[29] Hung Tri Le. A functional analytic approach to transient signal detection and estimation, 1989.

[30] Ronald S. LeFever and Carlo J. De Luca. A procedure for decomposing the Myoelectric signal into its constituent action potentials-part i: Technique, theory, and implementation. *IEEE transactions on biomedical engineering*, BME-29(3):149–157, 1982.

[31] G. H. Loudon, N. B. Jones, and A. S. Sehmi. New signal processing techniques for the decomposition of EMG signals. *Medical and Biological Engineering and Computing*, 30:591–599, 1992.

[32] Stephane Mallat. Zero-crossings of a wavelet transform. *IEEE transactions on information theory*, 37(4):1019–1033, July 1991.

[33] Stephen Del Marco and John Weiss. Improved transient signal detection using a wavepacket-based detector with an extended translation-invariant wavelet transform. *IEEE transactions on signal processing*, 45(4):841–850, 1997.

[34] Kevin C. McGill, Kenneth L. Cummins, and Leslie J. Dorfman. Automatic decomposition of the clinical electromyogram. *IEEE transactions on biomedical engineering*, BME-32(7):470–477, 1985.

[35] Roberto Merletti and Dario Farina. Analysis of intramuscular electromyogram signals. *Philosophical Transactions of the Royal Society A*, 367:357–368, 2009.

[36] Andrea Merlo, Dario Farina, and Roberto Merletti. A fast and reliable technique for muscle activity detection from surface EMG signals. *IEEE transactions on biomedical engineering*, 50(3):316–323, 2003.

[37] Sanjeev D. Nandedkar, Paul E. Barkhaus, and Alison Charles. Multi-motor unit action potential analysis (MMA). *Muscle & Nerve*, 18:1155–1166, 1995.

[38] S. Hamid Nawab, Robert P. Wotiz, and Carlo J. De Luca. Decomposition of indwelling EMG signals. *Journal of Applied Physiology*, 105:700–710, May 2008.

[39] Zoran Nenadic and Joel W. Burdick. Spike detection using the continuous wavelet transform. *IEEE transactions on biomedical engineering*, 52(1):74–87, 2005.

[40] Philippe Nicolas and Dieter Kraus. Detection and estimation of transient signals in coloured gaussian noise. In *International Conference on Acoustics, Speech, and Signal Processing*, 1988.

[41] G. M. Nijm, A. V. Sahakian, S. Swiryn, and A. C. Larson. Comparison of signal peak detection algorithms for self-gated cardiac cine mri. In *IEEE Computers in Cardiology*, 2007.

[42] Mile Nikolic, John Aasted Sorensen, Kristian Dahl, and Christian Krarup. Detailed analysis of motor unit activity. In *19th International Conference -IEEE/EMBS*, 1997.

[43] Albert H. Nuttall. Detection performance of power-law processors for random signals of unknown location, structure, extent, and strength. TR 10751, NUWC-NPT, New London, Connecticut 06320-5594, September 1994.

[44] Albert H. Nuttall. Near-optimum detection performance of power-law processors for random signals of unknown location, structure, extent, and arbitrary strength. TR 11123, NUWC-NPT, New London, Connecticut 06320-5594, April 1996.

[45] Albert H. Nuttall. Performance of power-law processor with normalization for random signals of unknown structure. TR 10760, NUWC-NPT, New London, Connecticut 06320-5594, May 1997.

[46] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[47] Girish Keshav Palshikar. Simple algorithms for peak detection in time-series. In *1st International conference on Advanced Data Analysis, Business Analytics and Intelligence*, 2009.

[48] Boaz Porat and Benjamin Friedlander. Adaptive detection of transient signals. *IEEE transactions on acoustics, speech, and signal processing*, ASSP-34(6):1410–1418, 1986.

[49] Boaz Porat and Benjamin Friedlander. Performance analysis of a class of transient detection algorithms-a unified framework. *IEEE transactions on signal processing*, 40(10):2536–2546, 1992.

[50] Ghulam Rasool and Kamran Iqbal. Muscle activity onset detection using energy detectors. In *34th Annual International Conference of the IEEE EMBS*, 2012.

[51] Felix Scholkmann, Jens Boss, and Martin Wolf. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. *Algorithms*, pages 588–603, 2012.

[52] D. Stashuk and Y. Qu. Adaptive motor unit action potential clustering using shape and temporal information. *Medical and Biological Engineering and Computing*, 34:41–49, 1996.

[53] Dan Stashuk. EMG signal decomposition: how can it be accomplished and used? *Journal of Electromyography and Kinesiology*, 11:151–173, 2001.

[54] Stephen Stehman. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62:77–89, 1997.

[55] Roy L. Streit and Peter K. Willett. Detection of random transient signals via hyperparameter estimation. *IEEE transactions on signal processing*, 47(7):1823–1834, 1999.

[56] Motion Lab Systems. Emg preamplifiers — motion lab systems, 2015. [Online; accessed 17-July-2015].

[57] Mojgan Tavakolan, Zhen Gang Xiao, Jacob Webb, and Carlo Menon. EMG processing for classification of hand gestures and regression of wrist torque. In *The Fourth IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2012.

[58] the National Spinal Cord Injury Statistical Center. Spinal cord injury facts and figures at a glance. Technical report, the National Institute on Disability and Rehabilitation Research, February 2012.

[59] Sandrine Thuret, Lawrence D. F. Moon, and Fred H. Gage. Therapeutic interventions after spinal cord injury. *Nature Reviews Neuroscience*, 7:628–643, 2006.

[60] Harry Urkowitz. Energy detection of unknown deterministic signals. *Proceedings of the IEEE*, 55(4):523–531, 1967.

[61] Martin Vetterli and Cormac Herley. Wavelets and filter banks: Theory and design. *IEEE transactions on signal processing*, 40(9):2207–2232, September 1992.

[62] Zhen Wang and Peter K. Willett. All-purpose and plug-in power-law detectors for transient signals. *IEEE transactions on signal processing*, 49(11):2454–2466, 2011.

[63] Wikipedia. Spinal cord — wikipedia, the free encyclopedia, 2015. [Online; accessed 16-July-2015].

[64] J. J. Wolcin. Maximum likelihood detection of transient signals using sequenced short-time power spectra. TM 831138, Naval Underwater Systems Center, August 1983.

[65] Michael T. Wolf. *Target Tracking Using Clustered Measurements, with Applications to Autonomous Brain–Machine Interfaces.* PhD thesis, California Institute of Technology, 2008.