## 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 powerlaw 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  $\sigma^2$ . Then from [39], it follows that

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

where  $M\{\mathbf{S}\}$  is the sample median of the sequence  $\mathbf{S}$  and  $\bar{\mathbf{X}} \stackrel{\text{def}}{=} 1/N \sum_{i=1}^{N} x_i$ . 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  $\sigma^2$  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  $\sigma^2$ . (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  $3\sigma$ . 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  ${f W}$  is initially set to be entire input. converge  $\leftarrow$  false; iter  $\leftarrow 1$ ; while not converge or iter  $\leq$  maxIter do iter  $\leftarrow$  iter + 1; //  $ar{\mathbf{W}}$  is the sample mean  $\hat{\mu} \leftarrow \bar{\mathbf{W}}$ ;  $\hat{\sigma} \leftarrow$  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  $|W| - |W^*| \le 0.0027 |W|$  then converge  $\leftarrow$  true; else $\mathbf{W} \leftarrow \mathbf{W}^*;$ end end





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:

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

where  $MEP_i(t)$ , defined over  $[0, L_i]$ , and  $\delta_i$  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  $j = 1, 2, \dots, N_i$ , 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:

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

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

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

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:

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

where  $\sigma^2$  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  $\sigma^2$  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:

$$x(t) = \sum_{i=1}^{N} A_i(t - p_i) + w_{\sigma}(t)$$
(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  $\sigma^2$ .

The goal is to estimate all the peak locations:  $\mathcal{P} \stackrel{\text{def}}{=} \{p_i, i = 1, 2, \dots, N\}$ . 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 multiresolution 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:

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

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

$$X(s,u) = \int_{-\infty}^{+\infty} P(t)\psi_{s,u}(t)dt + \int_{-\infty}^{+\infty} B(t)\psi_{s,u}(t)dt + \int_{-\infty}^{+\infty} C\psi_{s,u}(t)dt$$
(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  $\theta_s(t) \stackrel{\text{\tiny def}}{=} (1/\sqrt{s})\theta(t/s)$  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:

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

$$\mathcal{S} \stackrel{\text{\tiny def}}{=} \{s_0, s_1, \dots, s_j, \dots, s_J\}, \qquad s_i < s_j \qquad \text{for} \quad i < j \qquad (3.3.2.1)$$

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  $\infty$ ), 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" dis 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:

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

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:

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

$$\frac{s_{i+1}}{s_i} = \frac{2d \cdot C_{\psi} + BW_{\psi}}{2d \cdot C_{\psi} - BW_{\psi}}$$
(3.3.2.5)  
where:  $0 \le i \le J - 1, \quad i \in \mathbb{Z}$ 

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

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

where:

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

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  $\lambda$ . 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  $\lambda$ , "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:

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

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

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

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  $\lambda$ . Eq. (3.3.2.11) shows that a smaller value of  $\lambda$  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  $Q_{\psi}$  from Eq. (3.3.2.8)), the set of scales, 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



(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:

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

In Eq. (2.4.3.3), suppose the support of  $\psi_s$  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:

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

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

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



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  $\sigma^2$ :

$$W_s[k] \sim \mathcal{N}(0, \sum_{n=B_L}^{B_R} \psi_s^2[n] \cdot \sigma^2)$$
 (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  $\{x[n], n = 0, 1, 2, ..., N\}$ , with choice of mother wavelet and scales from previous sections, as **X**:

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

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

where  $X_{j,k}$  is the element of **X** at row j, column k, and  $s_j$  is derived from Eq. (3.3.2.1). **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:

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

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

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  $\gamma$  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  $\sigma$  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  $\gamma$  are removed. The remaining local maxima in the wavelet coefficients are called "peak maxima".

Define the parameter "threshold coefficient", c, as:

$$c \stackrel{\text{\tiny def}}{=} \frac{\gamma}{\sigma}; \qquad c \in \mathbb{R}^+ \tag{3.3.4.5}$$

So:

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

where  $\sigma$  is the standard deviation of WGN,  $\gamma$  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 **X** (Refer to Eq. (3.3.4.1)), all the peak maxima produced by the binary hypothesis testing at scale  $s_i$  are:

$$\mathcal{P}_{j} \stackrel{\text{\tiny def}}{=} \{k \mid X_{j,k} > \gamma, \quad X_{j,k} > X_{j,k-1}, \quad X_{j,k} > X_{j,k+1}\}, \qquad j = 0, 1, 2, \dots J$$
(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:

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

$$(3.3.4.8)$$

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

$$m_j \stackrel{\text{\tiny def}}{=} |\mathcal{P}_j|, \quad j = 0, 1, 2, \dots J$$
 (3.3.4.9)

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

$$\mathcal{P} \stackrel{\text{\tiny def}}{=} \{ \mathcal{P}_j \mid j = 1, 2, \cdots, J \}$$
(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 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  $s_i$ :

for 
$$\forall i \in \{1, 2, \dots, m_j\}$$
  
if  $\exists k \in \{1, 2, \dots, m_{j+1}\} : |p_{j,i} - p_{j+1,k}| < w,$  (3.3.5.1)  
then:  $p_{j,i}$  is connected to  $p_{j+1,k}$ 

Otherwise:  $p_{j,i}$  is regarded as the start of a new ridge.

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:

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

where 
$$k_{ridge} \in \mathbb{R}^+$$
: ridge length threshold coefficient (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  $\tau_i$  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  $\sigma^2$ .

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  $k = 0, 1, \dots, N$ , 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  $i = 1, 2, \dots, N_p$ , such that  $p_i \in \{0, 1, 2, \dots, N\}$ .

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],  $k = 0, 1, 2, \dots, N$ , with Mexican Hat mother wavelet,  $\psi[n]$ and a choice of scales,  $s_j$ ,  $j = 0, 1, 2, \dots, J$ , calculated from Eq. (3.3.2.6), (3.3.2.9), and (3.3.2.11), to obtain a matrix of wavelet coefficients **X**.

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

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

2. Identify peak maxima by performing GLRT on local maxima of  $X_{j,:}$  for all  $j = 0, 1, 2, \dots, J$ , 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:

$$\mathcal{P}_{j} \stackrel{\text{\tiny def}}{=} \{k \quad | \quad X_{j,k} > \gamma, \quad X_{j,k} > X_{j,k-1}, \quad X_{j,k} > X_{j,k+1}\}, \qquad j = 0, 1, 2, \dots J$$

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

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

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

$$m_j \stackrel{\text{\tiny def}}{=} |\mathcal{P}_j|, \quad j = 0, 1, 2, \dots J$$

 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  $p_{J,i}$ ,  $i = 1, 2, \dots, m_J$ . For  $j = J - 1, J - 2, \dots, 1, 0$ , at scale  $s_j$ :

for 
$$\forall i \in \{1, 2, ..., m_j\}$$
  
if  $\exists k \in \{1, 2, ..., m_{j+1}\} : |p_{j,i} - p_{j+1,k}| < w,$  (3.3.6.1)  
then:  $p_{j,i}$  is connected to  $p_{j+1,k}$ 

Otherwise:  $p_{j,i}$  is regarded as the start of a new ridge.

The result is a set of ridges, denoted as  $\{\mathcal{R}_i, i = 1, 2, 3, \dots, N_R\}$ , where  $N_R$  is the total number of ridges detected.  $\mathcal{R}_i$  is a set of pair of indices:  $\mathcal{R}_i = \{(rx_{i,j}, ry_{i,j}), j = 1, 2, \dots, L_i\}$ , 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:

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

Then:

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



(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 dis 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 doublethreshold 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  $1\mu$ V.)

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.

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

- 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.

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

- 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 resultsFigure 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:

$$x[n] \stackrel{\text{\tiny def}}{=} \sum_{i=1}^{N_p} P[n-p_i] + w_{\sigma}[n]$$
(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  $\sigma^2$ .  $p_i$  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 - 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:

$$P_f[n] \stackrel{\text{def}}{=} A \cdot \sin(\pi \cdot \frac{n}{L_f - 1}), \qquad n = 0, 1, 2, \cdots, L_f - 1$$
 (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  $1\mu$ V.)

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.



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  $1\mu$ V.

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:

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

where  $[s_1, s_2]$  is the support of the transient signal s[n],  $\sigma^2$  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  $\sigma$  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  $\tau_i$  from interval  $[1 + (i 1) \cdot M/(1 \text{sparsity}) + \text{buffer}, i \cdot M/(1 \text{sparsity})]$  for  $i \in \{1, 2, 3, \dots, N\}$ . buffer is used to make sure two transient signals don't overlap with each other.
- 6. Synthesize WGN  $w_{\sigma}[n]$  with variance of  $\sigma^2$  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  $TP \stackrel{\text{\tiny def}}{=} |\mathcal{P} \cap \mathcal{O}|$ , the number of EMG peaks that are detected.

- False positive  $FP \stackrel{\text{def}}{=} |\mathcal{O} \setminus \mathcal{P}|$ , the number of non-EMG peaks (peaks from noise) that are detected (number of false alarms).
- False negative  $FN \stackrel{\text{\tiny def}}{=} |\mathcal{P} \setminus \mathcal{O}|$ , 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:

$$Recall = \frac{TP}{TP + FN}$$
$$Precision = \frac{TP}{TP + FP}$$

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  $\beta$  is:

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

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

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

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

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,  $\omega$ , (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  $\omega$  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:

## $f_s = 2000$ sparsiy = 80%

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

density: d = 3; (3.4.3.3a)

- threshold coefficient: c = 2.2; (3.4.3.3b)
  - time threshold: w = 5; (3.4.3.3c)
- ridge length threshold coefficient:  $k_{ridge} = 0.6;$  (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:

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

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 tradeoff 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



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



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 re-call 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.



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.



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.



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.