Use of Evolutionary Algorithms to Select Filters for Evoked Potential Enhancement

Scott Turner

Thesis submitted for the degree of Doctor of Philosophy
Leicester University

University College Northampton September 2000

Acknowledgement

I would like to thank Professors Phil Picton and Jackie Campbell for their supervision.

Dr Robin Crockett for support as Research Degrees Tutor.

Thank you to Justine, for support, encouragement, proofreading and not least of all putting up with my irritability at times.

Abstract

Evoked potentials are electrical signals produced by the nervous system in response to a stimulus. In general these signals are noisy with a low signal to noise ratio. The aim was to investigate ways of extracting the evoked response within an evoked potential recording, achieving a similar signal to noise ratio as conventional averaging but with less repetitions per average. In this thesis, evolutionary algorithms were used in three ways to extract the evoked potentials from a noisy background.

First, evolutionary algorithms selected the cut-off frequencies for a set of filters. A different filter or filter bank was produced for each data set. The noisy signal was passed through each filter in a bank of filters; the filter bank output was a weighted sum of the individual filter outputs. The goal was to use three filters ideally one for each of the three regions (early, middle and late components), but the use of five filters was also investigated. Each signal was split into two time domains: the first 30ms of the signal; and the region 30 to 400ms. Filter banks were then developed for these regions separately.

Secondly, instead of using a single set of filters applied to the whole signal, different filters (or combinations of filters) were applied at different times. Evolutionary algorithms are used to select the duration of each filter, as well as the frequency parameters and weightings of the filters. Three filtering approaches were investigated.

Finally, wavelets in conjunction with an evolutionary algorithm were used to select particular wavelets and wavelet parameters.

A comparison of these methods with optimal filtering methods and averaging was made. Averages of 10 signals were found suitable, and time-varying techniques were found to perform better than applying one filter to the whole signal.

Words in the main body of the text approx. 28200

Table of Contents

Table of Contents (cont.)

Table of Contents (cont.)

Table of Figures

Table of Figures (cont.)

Table of Figures (cont.)

Table of Figures (cont.)

1 Introduction

In this chapter bioelectrical signals are introduced and evoked potentials in particular. Also considered are some of the problems of evoked potential recordings and the aims of this work.

1.1 Basis of Electrophysiology

A basic principle behind electrophysiological measurements is that living neural cells produce electrical signals. The form of these signals that most people are familiar with is the electrocardiogram (ECG), measuring the electrical activity of the heart. This is not the only type of measurement that is possible. Electrical activity of muscles can be recorded, as can activity of the nerves, spinal cord and brain.

Nerves can be stimulated and the nervous system's response to the stimulus can then be recorded. This is an evoked response and the form of measurement of interest for this work. Before discussing what an evoked response is, it is worth considering where these signals come from.

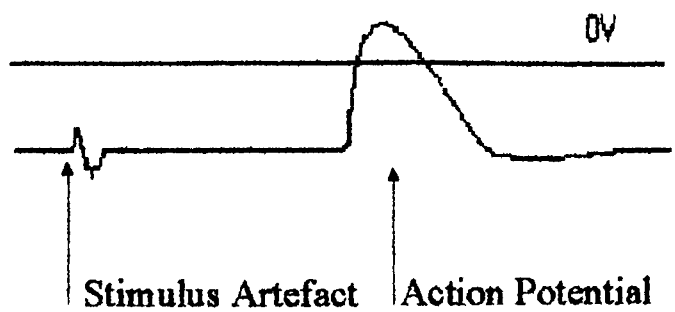

Figure 1-1 Representation of an action potential

All living cells maintain a potential difference across the cell membrane, due to imbalances of positive ions outside the cell and negative ions inside the cell. The potential difference across the cell membrane is called the membrane potential, the magnitude of which indicates the charge on the inside surface of the membrane (when a neuron is not conducting impulses, it is said to be resting). At rest the membrane potential is typically maintained at around –70 mV, so this is known as the resting potential. When a stimulus is applied to the membrane, the membrane or a portion of the membrane increases its permeability. This is achieved by opening sodium channels in the membrane, allowing sodium to diffuse into the cell and causing a small rise in the membrane potential (depolarisation). If the membrane potential increases above a certain threshold level, typically –59 mV, even more sodium channels open. These sodium channels stay open for about 1 ms, allowing the same amount of sodium ions to diffuse in each time, thereby producing the same magnitude of response. This response is called an action potential (figure 1-1) and is an all-or-nothing response. If the threshold level is exceeded the full peak of the action potential is always reached (approximately 30 mV). If the threshold potential is not exceeded, then no action potential is produced. Once the action potential's peak has been reached, the voltage starts to decrease (re-polarisation). Exceeding the threshold potential not only triggers the opening of sodium channels, but also voltage-sensitive potassium channels. The voltage-sensitive potassium channels are slower to respond than the sodium channels and do not open until the diffusion of sodium ions has caused a membrane potential of around +30 mV. Once the potassium channels open, potassium diffuses out of the cell and the potential starts to decrease. The potassium channels remain open when the membrane potential reaches the resting potential; an excess of potassium can diffuse out of the cell. This can cause a brief period where the action potential voltage drops below the resting potential (hyperpolarisation) before sodium-potassium pumps in the membrane return the ion channels to their resting state.

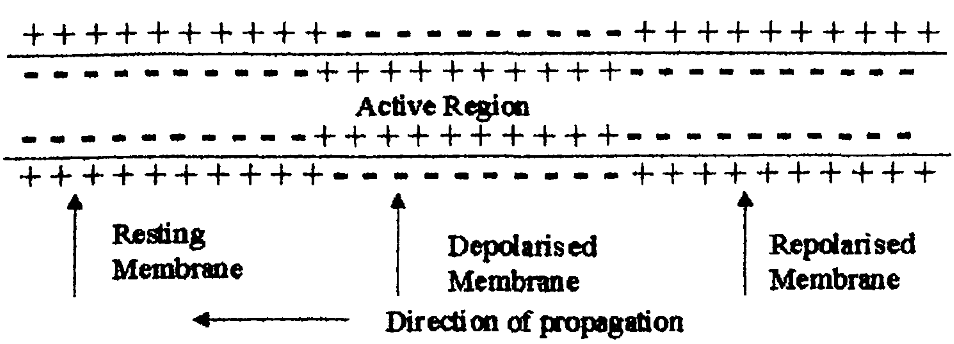

Figure 1-2 Action potential propagation

In approximately the first half millisecond after the threshold potential is surpassed, no matter how large the stimulus, the stimulated portion of membrane will not respond to a further stimulus. This is the absolute refractory period. The relative refractory period occurs in the few milliseconds after the absolute period and is the time it takes for the resting potential to be achieved. A strong stimulus is needed to re-stimulate the membrane during the relative refractory period.

When an action potential occurs in a region of the nerve membrane, it acts as a current source, causing an adjacent portion of the membrane to increase in potential, thereby initiating an action potential. This process is repeated and so an impulse seems to move along the nerve (figure 1-2). Because a portion of the membrane that produced an action potential goes into a refractory period after the action potential has been produced, it is inhibited from being re-stimulated by the action potential further along the nerve. This means that the nerve impulse moves along the nerve in one direction.

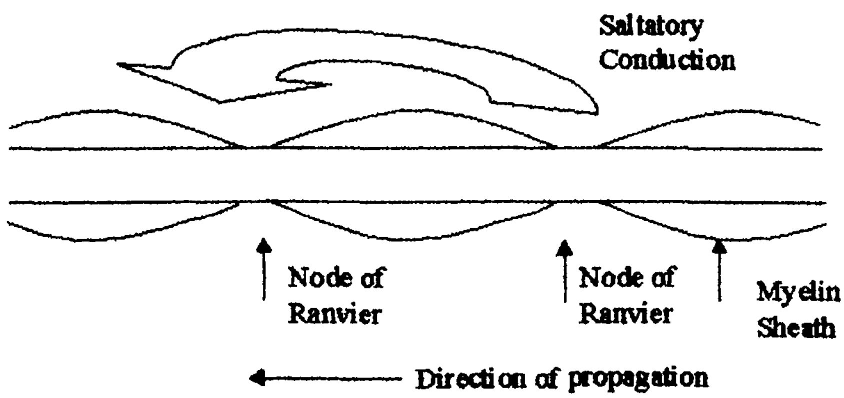

Figure 1-3 Propagation along a myelinated fibre

All except the smallest nerve fibres in the human body are insulated by a covering called myelin. The previous description of impulse conduction refers to unmyelinated fibres, but the principle of an action potential causing others parts of the nerve fibre to be stimulated is still true for myelinated fibres. In myelinated fibres, sodium and potassium channels are densely clustered around the gaps between the myelin sheaths, called nodes of Ranvier. The difference between the way myelinated and unmyelinated fibres conducts an impulse is that myelinated fibres propagate an impulse by a sequentially activating the nodes of Ranvier. The same basic idea of the nerve locally polarising and depolarising as in the unmyleinated fibre is still true. The action potential appears to leap up the nerve fibre (fig 1-3), so this is sometimes called saltatory conduction (from the Latin *saltare*, "to leap or dance") Clark (1992). Conduction along a myelinated fibre is faster (approximately 20 times) that of conduction along unmyelinated fibre.

1.2 What is an evoked potential?

If a nerve is stimulated, then differences in the electrical activity in the brain or spinal cord are produced in response to the stimulus. These are responses evoked by a stimulus and are called therefore evoked responses or evoked potentials. The types of response relevant to this work are somatosensory evoked potentials (SEPs), which are usually produced by electrical stimulation of sensory nerves. Electrodes placed on the scalp, or near the spinal cord, can be used to record these responses. The usual way to produce these responses is to apply a short electrical pulse over the nerve and record the response under the electrodes at the recording sites.

Somatosensory evoked potentials have a variety of applications.

- During spinal operations, such as to correct spinal curvature, SEP monitoring helps to avoid paralysis, which can be a possible complication of the operation.
- To provide information of dysfunction (Rossini et al, 1981). Spinal cord tumours may cause abnormal evoked potentials (Aminoff and Eisen, 1999).
- As a prognostic guide for coma and spinal injuries. In spinal injuries, if stimulation is carried out below the injury site, a response can be looked for at the scalp. If a response can be recorded after the injury, or the response returns soon after, this is taken as a good indication that the prognosis is good.
- MacLennan and Lovely (1995) discussed the use of somatosensory evoked potentials to test nerve conduction, which can be used in the diagnosis of Multiple Sclerosis (MS). The use of somatosensory evoked potentials for nerve conduction velocity studies and as an aid for MS diagnosis are two of the uses of SEPs that have been used the longest. Abnormalities in somatosensory evoked potentials have been found in 80% of patients diagnosed with Multiple Sclerosis (Aminoff (1999)).
- Depth of anaesthesia (Angel et al (1999), Nayak and Roy (1995,1998)).
- Braun et al (1996) used time-frequency analysis to detect temporal and spectral changes in somatosensory evoked potentials due to neurological injury such as from lack of oxygen (hypoxia). A form of time-frequency analysis, wavelet analysis, has been used to characterise changes in the shape of evoked potentials due to neurological injury (Thakor et al, (1993a, 1993b)).

The term 'latency' is used throughout the literature and is an important consideration of any work on evoked potentials. Latency is the measurement of time to the occurrence of a peak after a stimulus. A number of papers (e.g. Rossini et al (1981), Maccabee et al. (1992)) have commented that the earlier (or shorter) latency components are considered more stable than the longer latency components. Anziska and Cracco (1981) discussed the source of the various positive and negative peaks. Short latency SEPs recorded by scalp electrodes do not reflect the signal travelling up the nerve, as it approaches and passes under the electrode, but are far field effects. These are generated mainly in the fibre tracts (Anziska and Cracco 1981). Rossini et al (1981) concluded that the small stable early components come from subcortical processes. Maccabee et al (1992) state that earlier components have higher frequency contents than later components, which means that the assumption that the signal is stationary is not valid. In other words, the signal's frequency properties are not constant throughout the signal. Late (or long) latency components are associated with cortical process and the path the responses can take are more varied, spreading the peaks in the signal, thereby decreasing the stationarity of these components.

1.3 What are the problems?

A variety of problems exists in measuring evoked potentials (EP). Low Signal to Noise ratio (SNR) (often <<1), due to the low signal amplitude in comparison to the amplitude of the background activity (Harrison et al., 1995) is considered the main problem in extracting evoked responses. In the next chapter, methods to improve the SNR will be considered, including the most commonly used method of averaging a large number of responses. The background activity in part comes from other sources of electrical activity within the body. An example of that includes electrical artifacts produced by muscles, which can have a very detrimental effect on the SNR (Cadwell and Villarreal, 1999). Activity produced by respiratory muscles and the heart can interfere with measurements, e.g. during scolosis corrective surgery (Choudhry et al. (1998)). Electrical activity from the brain [electroencephalograph (EEG)] that is not directly evoked by stimulus, is a difficulty found in scalp recordings. Table 1.1 shows examples of the signal properties of other bioelectric signals. The range of frequency values and size of voltages of evoked potentials (ranging from a few microvolts to hundreds of microvolts depends on where the recording electrodes are placed.

		Freq. (Hz)	Voltage (approx.)
Electrocardiogram(ECG) – Recordings of electrical activity of the heart	Min	0.05	5 mV
	Max	500	10 mV
Electroencephalography (EEG) – Recordings of the electrical activity within the brain.	Min	0.1	2 μV
	Max.	100	200 μV
Action potentials	Min	DC	0.01mV
	Max	10000	3 mV
Electromyography (EMG) - Recordings of the electrical activity of muscles.	Min.	DC	25 mV
	Max.	10000	5 mV

Table 1-1 Other Bioelectric signals: Taken from various sources (Khandpur (1987), Olson (1992))

Table 1-1 shows the approximate absolute frequencies and voltages for the four common sources of the electrical activity (and artifacts) in the body. The investigation site (e.g. scalp or spinal recordings) will have an effect on the response by the different artifacts produced (Harrison and Lovely, 1995). The term noise used here is taken to mean the unwanted components of the signal, with respect to the evoked potentials. It should be borne in mind that some of these unwanted components such as those in table 1-1 do have uses outside of this work. Other factors such as fluctuations in vigilance state and conduction delays, change the amplitude and latency of waves within the recording session and therefore alter the signal.

Interference can also come from the recording equipment such as electrical noise from the amplifiers used and from the environment. Line frequency or mains hum is one such problem; this is interference from the mains supply. A filter that attenuates frequency components at the line frequency and its harmonics can help to reduce this problem. This type of filter is used at the recording equipment while the signal are being collected or after the signal have been collected, may also remove components of the evoked response at the same time. The signal, like many biological signals, will vary (i.e. be nonstationary) over a long time interval and most of the more commonly used signal processing techniques assume stationarity (e.g. linear filtering).

It is timely at this point to explain some conventions for such recordings used in this work. One convention in electrophysiological recordings is for a positive peak to be down the page (-y axis) and negative to be up (+y axis). In this thesis because of the packages used, the convention of positive as up (+y axis) the page and negative down (-y axis) has been adopted as with most physical science conventions. There is also no clear definition on what is classed as short, mid, or late latency regions, but based on the similar conventions adopted by some authors (e.g. Nishida et al, 1993), less than 30 ms is classed here are short latency, 30-100 ms as mid latency and greater than 100 ms as late latency regions.

In summary, evoked potentials are signals produced in response to an electrical stimulus and are often recorded at the scalp or spine. These signals have a certain amount of time variance between individual recordings. Recordings are often corrupted with noise from within the body, as well as susceptible to noise from outside the body. The noise power is often greater than the signal power; i.e. often have low signal-to-noise ratios (SNR).

1.4 Aims

As will be discussed later, conventional averaging (section 2.1) is the usual method for improving the SNR, by using large numbers of responses. The overall aim is to investigate ways of extracting the evoked response within an evoked potential recording, achieving a similar SNR as conventional averaging but with less repetitions per average (see section 2.1 for advantages of this). This thesis investigates whether the use of evolutionary algorithm can be used to achieve this by the selection of filters or wavelets. Evolutionary algorithms (see chapter 5) use the principles of biological evolution to find solutions to problems. An evolutionary algorithm approach allows the number of assumptions made at the start to be limited. One of the reasons for choosing a filter bank approach is that clinicians already use filters routinely, so applying a set of filters to these signals is not a great change in their usual practice.

1.5 Structure of the Thesis

After this introductory chapter, there follows a chapter (Chapter 2) introducing the range of techniques that have been investigated previously and those in current usage. Chapter 3 discusses the recording methodology and includes a description of the data sets used. Chapter 4 looks at various ways of analysing these signals and includes two techniques to enhance the extraction of evoked potential: - one by looking at the power spectrum of the signal, the other by building *a Posteriori* optimal filters discussed in chapter 2. Chapters 5-8 show evolutionary algorithms being used to select sets of filters or wavelets applied to a novel application of enhancing the extraction of evoked potentials from noisy recordings. Chapter 9 contains comparisons of the methods. The final chapter (chapter 10) includes conclusions of the work and areas of possible work leading off from this research presented in chapters 5-8.

2 Review of Current Techniques

After discussing, in the last chapter, some of the problems of recording evoked potentials, this chapter reviews techniques currently used.

From a signal-processing viewpoint, evoked potentials have one advantage over some other forms of recorded biological activity (e.g. EEG). The start of the signal is known, due to the stimulus evoking the response at a known point in time. The beginning of the evoked response is considered to start when the stimulus is applied. This leads to the use of techniques such as averaging, without the need for extra processing to detect the start of the signal. Further processing is needed though to improve the SNR.

2.1 Averaging

Two forms of averaging are considered: ensemble (or coherent) averaging and weighted averaging. Ensemble averaging, is the usual method for improving SNR (Harrison and Lovely, 1995) in evoked potential studies. The assumptions and some of the properties of averaging are also relevant to other techniques and so will be discussed in detail here.

2.1.1 Ensemble averaging

Ensemble averaging is where the mean value across all the signals at each point in time, is calculated. At the end of the process one signal is produced, an averaged signal. For averaging to be valid, assumptions about the signals have to be made (Glaser and Ruchkin 1976).

- **The recorded waveform is a linear sum of the noise and the evoked response.**
- **The shape of the waveform is attributable solely to the stimulus and is the same for each repetition.** The evoked response is assumed not to change between each repetition. All the components of the evoked potentials are considered to be locked to the stimulus. Variations in the position of the features are known to be possible between responses (e.g. Rossini et al., 1981, Maccabee et al. 1992).
- **The contribution of noise to the observed data is sufficiently irregular that it can be considered statistically independent samples of a random process.** In

other words, the noise is considered random and uncorrelated with the response. If applying a stimulus alters the noise, then this change is part of the response. Usually the noise is considered to be 'white' noise (zero mean value, with a Gaussian distribution).

In the ideal situation, it can be expected at some time, t, for a similar peak will be present in all the signals. The noise is assumed a random signal. As the number of signals (N) used in the average increases, the contribution to the overall signal from noise increases proportionally to the square root of the number of signals. The underlying signal has the same value at a point in time in all the signals. As the number of signals averaged increases, the value in the averaged signal increase proportionally to N. Signal to noise ratio therefore increases proportionally to the square root of N.

Consideration of how these techniques are used is needed. To be of use a large number of responses is often needed, sometimes over 1000 (Morin et al, 1987, Maccabee et al, 1986) for small components. For late components that are often relatively large 50-256 signals per average have been used (Carlton and Katz (1986)). There is a practical limit to the number of stimulations per second. This limit is due to the possibility of components of a previous response overlapping with components in the current response. Therefore increasing the stimulation rate is problematic. In somatosensory evoked potentials this is often a maximum of 2-5 stimuli per second (MacLennan and Lovely, 1995). The large number of responses needed and the limitation on the stimulation rate, means in order to acquire a suitable average, several minutes (1 to 5 minutes) worth of recordings are often needed. Remembering that each signal is produced by a stimulus to the subject, this is a technique where a person is having a nerve artificially stimulated for several minutes. Collecting the evoked potentials quickly is useful, as these techniques may be carried out as part of other activities (such as surgery), or there is an increased risk of the subject getting bored and therefore moving more, introducing artefacts. Connected with this, if the peaks in each response are shifted in time, this can lead to the 'smoothing out' of peaks of the response (leBron et al, 1995). Therefore, what is needed is to either collect fewer signals or do more with the signals collected.

2.1.2 Weighted averaging

Davilla and Mobin (1992) have investigated a method in which each response is weighted individually. The value of weights were derived from the set of signals to be averaged x and the ensemble average, $\bar{x}$, of the set of signals, by the equation shown below, where x^T is the transpose of x.

$$w = \frac{\bar{x}x^T}{\left\| \bar{x}x^T \right\|}$$ **Equation 2-1**

The Davilla and Mobin method was applied to auditory evoked potentials and showed increased signal to noise ratio compared with ensemble averaging. Using two estimates of SNR, the authors show an 8 to 21% improvement using weighted averaging compared to conventional ensemble averaging. The percentage increase in SNR for each subject was considered independent of the estimator used.

Darragh et al (1995) also compared conventional averaging with weighted averaging extending the results of Davilla and Mobin. The data used in their study was simulated using three sets of signals based on exponentially weighted sinusoids. Each signal was translated along the time axis randomly to simulate random variations in latency of peaks ('jitter'). White noise was added to simulate background activity. They showed a decrease in SNR as the standard deviation of the 'jitter' increased. The problem with this is that the data is based on a further set of assumptions to those already discussed, such as the noise being white, or that the signals can be realistically be modelled as a sets of exponentially weighted sinusoids. It is questionable that the noise of background activity is white, but coloured.

Bezerianos et al (1995) applied a different form of weighted averaging; they called it data dependent weighted averaging (DDWA), to visual evoked potentials. Two forms of this approach were used: one based on suppressing results that differ substantially from the rest of the data and a more successful method based on the signal to noise ratios (the estimator of Coppla et al (1978)). The authors claimed good results for both methods but no comparison with Davilla and Mobin's method was given; though reference to the paper was made. One problem with these claims is the SNR estimator used to test both methods is the same one used to form the weights of the second method. A method of estimating SNR that is independent of how the weights were produced would have led to greater confidence in the test method.

Both Davilla and Mobin's approach and that of Bezerianos et al were applied to actual evoked potential recordings. What is not clear is the suitability of these techniques for signals with SNR significantly lower than the auditory and visual evoked potentials used.

2.2 Linear Filtering

Filtering is often used to remove certain frequency components of a signal, e.g. the removal of noise components or investigating the signal over a narrow band of frequencies. Four basic types of filters are considered low-pass, high-pass, bandpass and bandstop filters. A low-pass filter attenuates signals above a certain frequency (cut-off frequency), but frequencies below this cut off frequency pass through relatively unattenuated. A high-pass filter is the opposite of the previous filter, attenuating signal components below the cut off frequency. The third type of filter is a bandpass filter that has two cut-off frequencies. Frequencies between these two cut-off frequencies are passed through relatively unattenuated. All other frequencies are attenuated. A bandstop filters is the opposite of the previous filter, all the frequencies between the two cut-off frequencies are attenuated, all others frequencies are passed through relatively unattenuated.

Bandpass and high-pass filtering methods have been applied to evoked potential recordings, often to remove low frequency components that would otherwise dominate the signal. Several groups have used bandpass filters with relatively high cut-off frequencies (2 kHz - 3 kHz), e.g. Maccabee et al (1983), Rossini et al (1981). All of these groups applied filtering to signals averaged with large numbers of responses. Removing the low frequency signal components was found to make detection of other components at higher frequencies possible, especially for the early components. Table 2.1 (at the end of the chapter) contains a list of several groups and the filters they used.

Analogue filters may alter the phase of the filtered signals and thereby alter the position of signal peaks; this is phase or latency shifting. The greater the rate of attenuation of the filter (therefore the higher the order of the filter), the greater the phase shift (Kriss, 1985). New ‘peaks’ can also be generated by the differential effects of analogue filters (Campbell and Leandri, 1984). Digital filters can avoid this problem, as techniques are available to produce digital implementations of the analogue filters that do not produce any phase shifting in the filtered signals. A digital band pass filter, 200-1500 Hz, has been applied to median nerve stimulated responses (Green, 1986). These results show these filters avoided the introduction of latency shifting. Eisen et al (1984) used a digital band-pass filter (300-2500 Hz) to remove the effects of the larger low frequency components, similar to the analogue filtering of others (e.g. Maccabee et al 1983). Their results show no phase shifting was produced on the signal peaks and the filter attenuated the peaks. Bessel filters are analogue filters that have a constant transmission delay, that means the effect of phase shifting filtering can be accounted for. Bessel Filter do have a disadvantage of having less effective magnitude characteristics than some of the filter types that were commonly used (e.g. Butterworth and Chebyshev filters).

A positive property of digital filtering methods is they can be implemented in software and several commercially available packages (e.g. MATLAB and LabView) include commands to implement them. This leads to the possibility of investigating the effects of altering properties of the filters and combinations of filters quickly, simply and cheaply, with the possibility of later being implemented is hardware for quicker operation

Filters set between 20-2000 Hz and 200-2000 Hz have been applied to epidural spinal cord stimulated evoked potentials (Paradiso et al, 1995). Surface spinal recordings of peroneal nerve stimulated signals have been made using a bandpass filter 10 Hz - 1 kHz (Morin et al, 1987). These surface recordings showed little significant contribution above 500 Hz. Maccabee et al (1986) found that in scalp evoked potentials due to stimulating the median nerve, most of the spectral energy was below 125 Hz, with lower energy components extending up to 500-700 Hz. One group (Maccabee et al, 1986) highlighted that filters can produce artifacts of their own. For a clinical correlation, they suggested that the filtered signal must be compared with the all pass signal. This was defined as the unfiltered signal. Only those peaks that appear in both the all pass and filtered signals were considered authentic. The problem with this is how to decide that what is present is a peak of the evoked potential and not noise. Therefore, this kind of comparison is only possible after a signal has been processed with a technique such as averaging a large number of signals, to reduce the noise.

A common feature of most of these groups work is that they do not justify why they used these particular frequencies for their filter. Bandpass filters acting as high-pass filters are justified on the grounds that these can be used to enhance the extraction of low amplitude high-rate components, while suppressing long-latency components (Maccabee and Hassan (1992)). What is not usually discussed is why those particular values were selected. A second point is the 'high-pass' filter approach is unsuitable if both short and long latency components are of interest. In the paper by Maccabee et al (1986), some justification was provided by showing a power spectrum of a scalp recording due to median nerve stimulation. The spectrum showed different regions of the spectra, Maccabee et al (1986) commented that high frequency components extended up to 500-700 Hz.

One of the problems with this linear filtering is it that the spectrum of the both the noise and the evoked potential occupy similar regions (Karajalainen et al., 1999). This means that a single linear filter is unlikely to extract the whole of the signal.

A related approach (Nishida et al., 1993) was to pass the evoked potentials through, three bandpass filters at different times and combine the outputs of the filters, to form a single signal. Scalp recordings of a somatosensory evoked potential were considered to contain three frequency ranges. The first range is a high frequency range (64.5 Hz to 300 Hz) occurring at the beginning of the signal. A second range, a mid frequency range (17.5 to 64.5 Hz) occurs in the middle of the signal. The final region was a low frequency range (5 to17.5 Hz) that occurs at the end of the signal. The outputs of the filters are combined depending on where along the time axis the signal components are. In the range 0 to 25 ms, the high frequency filter provides the output of the signal. In the time between 35 and 80 ms, the mid-frequency filter provides the output. For the period greater than 90 ms, the low frequency filter provides the output. There were no abrupt changes between the various time segments and filters; a 10 ms transition region was included. During the transition period, the earlier filter's contribution decreased as the next filter's contribution increased, until only the later filter is providing the signal for the output. A different filter at different times is an extension of the analogue and digital filters discussed earlier. The group also added an EEG reducing algorithm they developed at the output of the low frequency filter. The selection of these frequency ranges were justified by looking at the frequencies in the power spectra of a scalp recorded evoked potential. What is not justified is the time segment sizes. Power spectra do not show when an event occurred; an assumption is made when using a power spectrum that the frequency components are present throughout the signal. The assumption that this group's work was based upon different filters for different regions puts into question the validity of using a power spectrum as the basis of selecting frequencies for the filters. What has not be shown by these authors, or those carrying out the analogue and digital filtering work is whether there are two or three distinct regions and if each region is adjacent to the last in terms of frequencies. The interesting part of Nishida's paper is combining the outputs of different filters at different times, to enhance the extraction of the evoked response.

LeBron Paige et al (1996) compiled an ensemble of averaged EP waveforms in a 2-dimensional (2D) array, for monitoring evoked potentials during surgery. In a 2D form, peaks and troughs in the evoked potential provide information about the variation within a response and between responses. If a 'ridge' is seen to shift during an operation, one possible reason could be that something is affecting the nerves. A 2D representation also means that some of the methods used in image processing may be applicable, such as spatial filtering, increasing the range of possible techniques that can be applied. Low-pass filtering between potentials (i.e. vertically) has been investigated, the assumption being that the desired waveform variations are small from one response to another (i.e. low frequency components) and high frequency components are more likely to contain noise. One problem with assuming high frequency components can be filtered out, is making sure that early components are not lost. In intraoperative electrophysiological monitoring (IEM), one objective is to obtain interpretable EPs as rapidly as possible. The main problem, as has been discussed earlier (see section on averaging), is the length of acquisition time; many individual responses are needed to produce a single average with a high SNR. During the time it takes to produce the average the characteristics of the response may have changed. This group's solution was to apply the 2D array to rapidly acquired EP, with fewer EPs in each averaged response, therefore lower SNR than an averaged response with more responses. LeBron Paige et al (1996) believed the added benefits of a 2D approach meant that the SNR can be raised above the low SNR.

2.3 Adaptive Filtering

Adaptive filtering techniques automatically adjust the filter based on the input and past output values. These techniques have been widely used in improving the extraction of evoked potentials. Usually more than one input channel was used so that some reference to the noise in the channels can be made. Thakor (1987) produced a two-channel system where the primary channel (the input to be filtered) and the reference channel were assumed to have the same signal, but with different uncorrelated noise imposed on top. The problem with this approach is that the signal looked for is assumed to be the same each time. Noise power is also assumed to be lower than the signal power (i.e. a SNR>1), but somatosensory evoked potentials often have a SNR<<1 (Harrison et al (1995)).

Parsa et al (1994) looked at the cancellation of unwanted electrical activity from muscles (myoelectrical activity) in recording evoked potentials. They used an array of electrodes on the forearm, on the same side as the stimulation, with one electrode over the median nerve and four others over the forearm. The responses were evoked by stimulation of the index finger. Myoelectric activity meant the SNR for the evoked potential was low. As the filter is adaptive, the aim is to follow the nonstationarities of the signal. The stimulus itself produces artifacts; Parsa et al (1998) investigated the use of adaptive filters, both non-linear and linear, to attenuate stimulus artifacts. Their results show that a non-linear adaptive filter produced better cancellation of the stimulus artifact than the linear adaptive filter. This they believed was due to the nature of the stimulus artifacts generation also being non-linear. Several recording channels are needed with these approaches to provide references to extract the signal.

All these techniques either need a good version of the signal or need several channels of data including one that contains background activity or reference signal. Approaches that require several channels have practical limitations in clinical situations and also can not always be applied retrospectively to historically recorded data where there is no way to control how the data were recorded and the number of channels used. One possible way to provide the noise data would be to use recordings, which were recorded immediately before the evoked potential recordings, but without any stimulation. A problem with this is that a further assumption has to be made that the properties of the background activity are not changed between recording the noise and the stimulated activity. This is an extension of assuming noise properties do not change with time; i.e. the noise is a stationary process.

Doncarli et al (1992) used a Kalman Filter to improve the evoked potential SNR. The somatosensory evoked potentials were recorded from the scalp, in response to stimuli to the tibial nerve at the ankle. An ensemble of evoked potentials was produced in the form of a 2D array, in a similar way to LeBron Paige et al (1996). The approach processes the data vertically to monitor for slow changes between the responses. The results were shown for both visual and somatosensory evoked potentials. Only the first 100 ms were considered, so whether this approach is applicable to latencies above 100 ms is unknown. The authors did suggest that to get a good estimate of a positive peak around 100 ms (P100) it is best to use 16 signals in an average, rather than 128 signals. This suggests there is a problem with variation between the signals of the position of the peaks with time. All this brings the discussion back to late latency components being more time variant than earlier components.

2.4 Signal Modelling and Prediction

A moving average filter uses previous input values with appropriate weights (filter coefficients) to estimate the evoked potential at the output of the filter. Expanding a binomial expression and using these as the coefficients of a moving average filter, Wastell (1979) produced a low-pass filter with no phase shift. This filter was then applied to visual evoked potentials. The low-pass filter removed the high frequency noise observed in the unfiltered signal, the expected features being clear of the noise. Care must be taken that peaks are not smoothed out or removed by this process. Challis and Kitney (1990) describe ensemble averaging (section 2.1) as a moving average filter (see Appendix B).

Autoregressive (AR) methods use previous output values from the filter to estimate the current signal value. The problem with AR modelling is that either the output values of the filter need to be known beforehand, or the signal to noise ratio needs to be sufficiently large. A signal with a large SNR can provide a good estimate of the signal to be used. Lange et al (1996, 1997) used an AR method to produce a single trial (single evoked potential) estimate of evoked potentials this time for movement related potentials by adapting a template of the signal. Norcia et al (1986) applied linear predictive methods to evoked potential recordings. Predicted output values y (n) were calculated by convoluting the prediction coefficients with past output values. Regions of the spectrum where resonance-like peaks occur are heavily weighted by the Linear Prediction Coefficients (LPC) model and flat regions de-emphasised. A combined form of time and frequency analysis was carried out using LPC (Norcia et al, 1986). The aim was not to extract evoked potentials, but to track variations in frequency components with time. Muthuswamy and Thakor (1998) used an AR process to form a spectrum of the signal. This approach produced a spectrum again with the advantages discussed in Norcia et al (1986).

Autoregressive Moving Average (ARMA) techniques as the name suggests are a combination of both autoregressive and moving average techniques. Hansson (1996) used a form of autoregressive-moving average, the Prony Method, to form a filter for filtering single evoked potential. In this approach, the signal is modelled as sum of damped sinusoids. This work was based on extracting parameters for the linear prediction of the signals from additive white noise, but assumes the signals are linear and time invariant. There is mounting evidence that these signals are not time-invariant (e.g. Parker and Goplan (1987)).

Wiener filters, developed in the 1940s by Norbert Wiener (Wiener, 1949), are optimal filters. Yu and McGillem (1983) used Wiener Filtering to model evoked potentials. Comparing time-invariant and time-varying Wiener filters, they found that time-varying filters have superior performance to time invariant filters. Time-variant filters are better able to deal with effects such as time 'jitter' (variations in the position on the time axis of a particular peak of the signal, relative to the start of the signal). Their other main conclusion was that the covariance matrix of the desired signal is needed, produced either from experimental data, or from a good signal model.

2.5 *A Posteriori* "Wiener" Filtering

Wiener filtering refers to techniques, as discussed previously, in which the mean squared error of the estimate of the signal and the desired signal is minimised, by the filter coefficients (Hayes, 1996). Usually, as discussed previously, a target signal (or some knowledge of the signal's characteristics) needs to be known beforehand. Methods have been developed that can produce a model from the signals themselves

(*A Posteriori*). Walter (1969) produced a transfer function by dividing the spectrum of the typical record, by the average of the spectrum of the signals. This transfer function can be used as a filter.

$$Sr(\omega) = \frac{N}{N-1} S\bar{x}(\omega) - \frac{1}{N-1} \overline{Sx(\omega)} \qquad \textbf{Equation 2-2}$$

$$H(\omega) = \frac{Sr(\omega)}{\overline{Sx(\omega)}} \qquad \textbf{Equation 2-3}$$

Where $Sr(\omega)$ is the spectrum of the desired response, $S\bar{x}(\omega)$ is the power spectrum of the average response and $\overline{Sx(\omega)}$ is the average of the individual power spectra. To recover the average response spectral components, the transfer function $H(\omega)$ is multiplied by $\overline{Sx(\omega)}$.

Doyle (1975) suggested that Walter's approach was more suitable for individual visual evoked response (VEPs) and used a variation of the above equation for averaged VEPs. In Doyle's methods a transfer function is formed by dividing $Sr(\omega)$ by the spectrum of the desired response and the noise spectra shown below: -

$$H(\omega) = \frac{Sr(\omega)}{Sr(\omega) + nn(\omega)} \qquad \textbf{Equation 2-4}$$

Where

$$nn(\omega) = \frac{1}{N}(\overline{Sx(\omega)} - Sr(\omega)) \qquad \textbf{Equation 2-5}$$

Walter used very simple signals, as his calculations were mostly done by hand and simulated background noise as 'white noise'. Doyle considered the filter to be accentuating the frequency components of the signal where the desired response is strong and noise is weak (high SNR) and suppressing the frequency components where the response is weak (low SNR). Both Walter and Doyle used optimal filtering on signals where the noise does not dominate the signal, which is not the case in somatosensory evoked potential.

Carlton and Katz (1980) compared four optimal filtering approaches for somatosensory evoked potentials recorded from anaesthetised monkeys. These approaches were the average of optimally filtered sweeps (Walter 1969); optimally filtering the average of sweeps (Doyle 1975); and recursive versions of the last two approaches. The recursive approaches were not found to give accurate estimates, so were not considered further. The average of the signal and the filtered signal over the same number of signals were correlated with the average of a large number of samples (256 unfiltered sweeps). Both optimal filtering methods were considered to preserve the components present in the average. It was found that there was no significant improvement in the correlation coefficient for the same number of samples between the unfiltered average and optimally filtered average. A factor that was suggested for this lack of improvement is that the filter assumes that the response and the background activity are constant. This does not account for variations between responses. The use of a correlation coefficient is limited if the signals looked for vary between the signals.

Dobie et al (1990) applied optimal filtering to auditory evoked responses using Walter's method (Walters 1969). Dobie et al and Wastell (1981) both raised the same criticism of optimal filters; optimal filter performance is superior to signal averaging-alone, but only at signal levels where the conventional approach (averaging) already yields a working SNR. Wastell (1981) concluded that given the computational overheads, this form of filtering is doubtful as a good technique. This last criticism is less important with computing speed and cost now relatively low. Furst and Blau (1991) took the work of Yu and McGillem and formed a suboptimal *a posteriori* version, using autocorrelation of a set of signals. This approach used several assumptions that the noise is stationary, zero-mean valued process and the signal is deterministic. Furst and Blau did point out that the later components of evoked potentials are less deterministic. With this in mind, it would explain why they applied this method to Brainstem Auditory Evoked potentials (BAEP) which are stable for early components. This still does not get around the problem of coping with low signal-to-noise ratio signal.

The above approaches all assume that the signals are stationary, but as discussed previously, the signals are considered to be nonstationary as they do not have the same frequency components throughout the signals. An *a posteriori* filter that has time-varying properties has been applied to evoked potentials (de Weerd and Kap 1981a, 1981b, 1981c). The time-varying property works in a similar way to the Nishida's linear filter bank approach (Nishida et al, 1993). The ratio of the central frequency and bandwidth of the filters is constant; filters with a wide bandwidth extract the later components, whereas filters with a narrow bandwidth extract the early components, with overlap between filters. The assumption is made that early components contain high frequency components and the later components contain predominantly low frequency ones. The signal is passed through these filters, which filter at different frequency ranges and a time-varying function is applied so that the filters contribution to the overall signal varies with time. In the de Weerd approach, the time-varying signal at the output of the filters is determined by the signals. Some criticism has been made of the work by Bertrand et al (1994) that the theory of the approach was lacking. De Weerd et al admit this, both groups agree that it did appear to work for the signals used, but all the signal used had relatively large SNR. This approach is a sensible direction because it enables time variation of the signal properties to be included in the modelling process.

2.6 Wavelets

Wavelets and time-frequency techniques are being increasingly used in the analysis and processing of biomedical and biological recordings. Using wavelets to extract evoked potentials from noisy averages has been investigated (Lim et al, 1995; and Bartnik et al 1992a, 1992b). This technique uses the wavelet's ability to decompose a signal into several signals, then to remove or alter some of the signals and then recompose the signals. Lim et al. (1995) applied this to evoked potentials caused by stimulation of respiratory muscles and found for these signals only a few wavelets are needed to extract the features they were interested in. These methods were able to extract some peaks but not all. Bartnik's group (Bartnik et al 1992a, 1992b) used auditory evoked potentials, obtaining a representation of the signal using one of the decomposed wavelets produced (in the range 2-8 Hz), though it does appear as if small components were being lost.

The fact that the wavelet approach produces both frequency related and time related components has been used by Journee et al (1995), to build a time-varying filter (a filter whose spectral properties vary with the signal). In their approach the averaged signal is processed into wavelet components, each of which approximately related to a different, adjacent, bandpass filtered version of the signal. The signal represented as a set of filtered signals and each value in the signals is a component. Instead of treating these as a set of filtered signals, the components were treated as measures of how well this signal at that time 'matched' a particular size of wavelet. A representation of how the frequency components of the signal vary with time can be formed. In other words as different 'sizes' of wavelets represent different bandpass filters, the components can be thought of as relating how a range of frequencies varies with time. The maximum absolute value of all the components was found and any component whose value is less than 5% of the maximum value is set to zero. This is based on the assumption that noise components are from a white noise (e.g. for EEG (Cadwell and Villarreal, 1999)) and likely to be smaller than signal components. Therefore, by setting the smaller components to zero the noise components should be reduced. A signal to be filtered is then processed into time and frequency related components and multiplied by the previously developed filter. The inverse of the wavelet transforms is applied to the resulting signal, acting as if the signal has been filtered. In all three groups, an assumption was made about what the desired signal should look like.

A modified form of Doyle's *a posteriori* optimal filter (1975), where the power spectra are replaced by squared discrete wavelet transform components, has been investigated by Bertrand (1994). The 'optimal' filter now has time-varying properties, which can be seen as an extension of the work of deWeerd (1981a) into time-varying optimal filters. What is not clear in the paper is how much of an improvement over just averaging this is, and what the effect would be of using single evoked potential recordings, or a smaller number of evoked potentials, in a subaverage. As a concept it is of interest because it provides a possible method of producing a time-varying filter, without having an ideal signal known beforehand (see section 2.5). Geva et al (1995, 1997), used 'wavelet-like' analysis to model the spatial and temporal characteristics of neurological signal generators. The 'wavelet' they chose to use was the Hermite function, which is based on the first and second derivatives of the Gaussian function. The reason for choosing this wavelet was its shape, as it resembles monophasic and biphasic shapes found in evoked potentials. The shape of the wavelet can improve the ability of wavelets to extract a signal. Samar et al (1996) used a modified form of a Meyer wavelet to match the shape the features of an auditory evoked potential. The method was applied to the early components of the signal. Isoglu et al (1998) used wavelets to decompose an evoked potential, in a similar ways to that of Bartnik et al (1992a). Lewalle et al (1995) used wavelets to analyse olfactory nerve response to a stimulus. Saatchi et al (1997) used a combination of wavelet analysis and artificial neural networks to filter evoked potentials.

2.7 Artificial Neural Networks

Neural networks have been used to filter visual evoked potentials (VEPs) (Fung et al, 1995 and 1996). A multilayer perception (MLP) network was used as a filter. The contents of a moving window formed the inputs to the neural network and the output is a non-linear combination of the inputs. The neural network estimates the VEP presented by estimating the components that can be determined and removing noise that is uncorrelated with the stimulus. The training signals have SNR of approximately -5 dB. The target signal is an ensemble average of 100 responses and has a higher SNR. Simulated results shows that the system can improve the SNR of single VEP, so reducing the number of responses per stimulus. No inter-subject analysis has been carried out to see if one filter can be used on several people, or if you need to get an average of 100 responses from each person.

Another form of neural network, the Hopfield network, was used to produce a robust moving average (Laskaris et al., 1997). The network was used to implement cluster analysis, in which the core of the cluster acts as an estimate of an instantaneous visual evoked potential signal. Tian et al (1997) used a neural network to estimate the latency of auditory brainstem response. The filter produced was developed to extract every peak possible without introducing a shift in the position of the peak. Amplitude distortion was considered irrelevant, as the estimation of the latency contains the medically significant information. The network was implemented as a four layer MLP, where the input is in the principal component projection values of the training and test data sets (15 principal component were used). The output of the network is compared with the results of the same data as assessed by an audiologist, with the results showing good correspondence between the two. The paper raises some interesting points such as how much importance should be attached to preserving relative amplitude values, if the important information is in the latencies of the peaks.

A neural network approach has also been investigated by Grieve et al. (1995) to remove the stimulus artifact. The network was trained with a reference signal that was not recorded over the recording site but near by, as the input to the network and the primary signal from the recording site as the target for the network. The network attempts to predict artifacts in the primary signal using artifacts in the reference. Their initial work suggests that the network was able to cancel stimulus artifacts.

The neural network approaches have been mainly developing non-linear filters, which are reported to have some success. What the resulting model does is not always as easily interpreted as either a single or a set of linear filters.

2.8 Evolutionary Algorithms

Evolutionary algorithms are a range of techniques including genetic algorithms (see chapter 5), and genetic programming, which are based on the concepts of evolving solutions to problems. Laskaris et al (1996) used a genetic algorithm to estimate a signal by taking the average of selected signals. Possible solutions were encoded as a binary sequence, with a bit per signal in the set. If the bit is '1' then the signal was included, if '0' then it was excluded. The algorithm was used to select the sets of signals whose average produced the maximum value for the equation they used. Evolutionary algorithms were used because they have proved a robust search method in a large space, starting from an initial set of possible solutions, they create sets of better solutions. This work is an extension of this group's work discussed previously into weighted averaging (Bezerianos et al, 1995) and it seems to be the first use of an evolutionary algorithm approach to enhance the extraction of evoked potentials. They themselves point out the problem that more work is needed to extract more 'subtle' peaks in the signals.

2.9 Summary

What these papers suggest is that there is a need for a method that can extract key clinical features of the signal using a small set of signals. This is needed to avoid the problem of a detrimental smoothing effect, due to the shifting of the latency of the peaks with time.

Search techniques are being increasingly used to extract or enhance evoked potentials, including evolutionary algorithms. The nature of evolutionary techniques, such as genetic algorithms, has potential for limiting and possibly testing the assumption made, at the same time as enhancing the SNR of the evoked potentials. At the same time new techniques such as wavelets are being increasingly used.

Many assumptions are made about these signals, some which conflict and are often subjective or made with little justification. A method that can extract the evoked potential with limited assumptions would be useful, not only as a technique to improve SNR but to produce a better understanding of the properties of evoked potentials. Evolutionary algorithms provide a way of limiting the assumptions made, by searching for solutions from an initially random set of solutions.

Author	Stimulated Nerve	Recording Site	Range (Hz) Min	Max.	Comment
Paradiso et al (1995)	Epidural spinal	Scalp and Cervical Spine	20	2000	20-2000 is the 'open bandpass' range and 200-2000 is for early latency component enhancement
			200	2000	
Nishida et al (1993)	Median	Scalp	5	17.5	Filter bank
			17.5	64.5	
			64.5	300	
Morin (1987)	Peroneal	Spinal	10	1000	Digital filter applied to surface recordings.
Maccabee et al(1986)	Median	Scalp	30	3000	Digital filter, 30-3000 Hz used for all latency components, 150-3000 Hz and 300-3000 Hz, act As high-pass filters for short latency components.
			150	3000	
			300	3000	
Green et al (1986)	Median	Scalp	200	1500	Digital filter used to enhance early latency components. Acting as a high-pass filter.
Rossini et al (1985)	Median, peroneal	Spinal	200	5000	
Eisen et al (1984)	Peroneal	Scalp	300	2500	Digital filter used to enhance early latency Components. Acting as high-pass filters.
	Median	Spine	300	2500	
Nuwer & Dawson(1984)	Peroneal	Spinal	30	3000	
Maccabee et al(1983)	Median	Scalp	5	3000	Analogue filter, 5-3000 Hz used for all latency components, 150-3000 Hz and 300-3000 Hz, act As high-pass filters for short latency components.
			150	3000	
			300	3000	
Rossini et al(1981)	Peroneal	Scalp	150	3000	Analogue filter used to enhance early latency components. Acting as a high-pass filter.

Table 2-1 Analogue and digital filters used to filter somatosensory evoked potentials.

3 Data Sets

All signals in this study were historical recordings, recorded either during spinal operations or from scalp recordings, were recorded by Prof. J. Campbell at the Walton Hospital, Liverpool. The recording methodology is described in Campbell (1985) and is briefly described below. A description of the data sets is included, also included are power spectra of background activity, single evoked potentials, and averaged signals.

3.1 Recording Methodology

Recordings were made using surface electrodes on the scalp or by electrodes placed within the anterior quadrant of the spinal cord.

3.1.1 Apparatus and storage.

Electrodes were connected to the inverting input of a differential pre-amplifier in a MEDELEC MS6 EMG machine. Pre-amplification by a factor of 15 was applied to the signals. All the signals were filtered by a bandpass filter with a passband of 0.016 to 750 Hz. Signals were then stored on FM tape (BASF super-ferro LH), using a STORE 4 FM tape recorder (Racal Recorders, UK) recorded at tape speed of 3.75 inches per second. A model of the system is shown in figure 3-1. Both the intraspinal and scalp recorded evoked potentials were produced by electrical stimulation of the median nerve in the left arm. A stimulation rate of two pulses per second was selected to provide a sufficient interval between evoked responses.

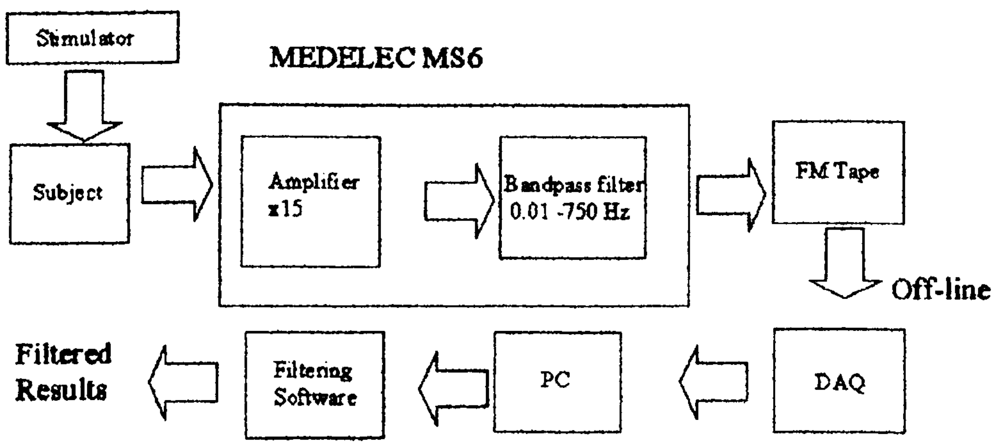

Figure 3-1Outline of the system

3.1.2 Data Collection

Data were collected from the tape using a Gateway 2000 Pentium P90 computer, via an interface card and data acquisition software (PC30F, Eagle Technology). Signals were stored as a set of binary data files by the data acquisition software, collecting three channels of data at a time, 2 recording channels and a third channel indicating when the stimulations were produced. A set of routines was developed in the MATLAB (MathWorks Inc., USA) programming language to convert the signals from a binary format into a set of single responses 400 ms long, stored in MATLAB's file format for ease of handling. The signal amplitude was left unaltered.

3.1.3 Power spectrum

Power spectra were used as a method of gaining further understanding of some of the frequency characteristics of the signals and background activity. The MATLAB command PSD was used to produce power spectral density estimates. The default values for this command were used, with the sampling frequency set at 7.5 kHz. The number of samples by which the sections overlap is zero and a Hanning window was applied. The power spectral density was estimated using Welch's method (Hayes, 1996). This approach was chosen because it is the standard MATLAB method for calculating power spectral density. Other options were considered but this was decided upon as being both appropriate to the task and convenient. There are difficulties with using power spectra to analyse signals, as it assumes the spectral components of the signal are constant throughout the signal. This is not necessarily true for all signals. Even with these problems, power spectra are used to gain some understanding of the signals, as well as a starting point for further analysis.

3.2 Assumptions about the signals

The assumptions made about the signals were:

- Noise is assumed independent of the response, but not necessarily white noise source. The possibility that the noise is structured is allowed for.
- The signal looked for has a similar shape to the target signals.
- Noise and signal are assumed to be linearly summated.
- Averaging can make some improvement in SNR.

- The averaged signal of a large number of signals is a good representation of the responses.

3.3 Intraspinal Recordings

3.3.1 Data Set 1

Recorded data consisted of 222 responses, from a single subject, collected from the tape. Thirty-eight responses were excluded from the experiments. The exclusion criteria were the same for all the data sets. If the signals contained artifacts such as 'clipping', flat responses, or very abrupt changes in the baseline within the signal, the response was excluded. These artifacts were taken to be due to the recording set-up and are not due to the responses. Using the remaining 184 recordings, two subsets with 92 responses in each were formed into a training and test subsets. The target signal (ideally the signals in the data set should have peaks in similar places to those in the target signal) was produced from an average of the 184 responses (see figure 3-2c and d).

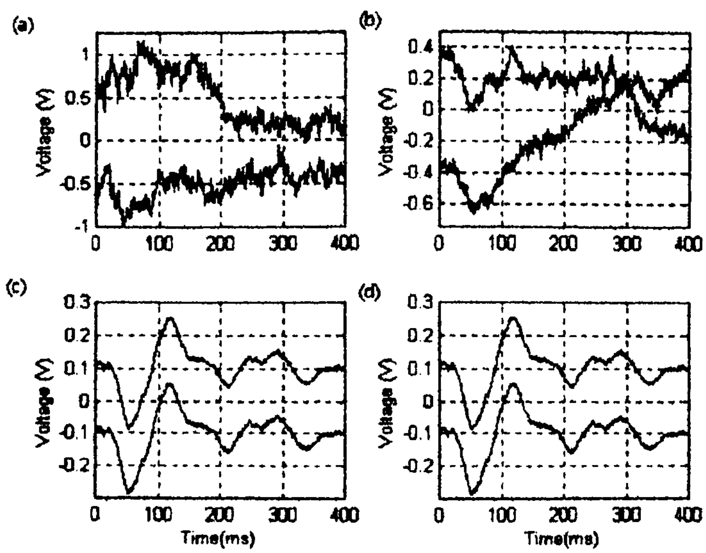

Figure 3-2 (a) 20 and 40th signals in the test subset of data set1, (b) 20 and 40th signals in the test subset of the simulated data set. (c and d) are the target signals.

Figure 3-2 shows one of the problems with these signals, that of low frequency components obscuring some of the features of the responses. The upper left plot (figure 3-2(a)) shows two examples of responses of a spinal recording.

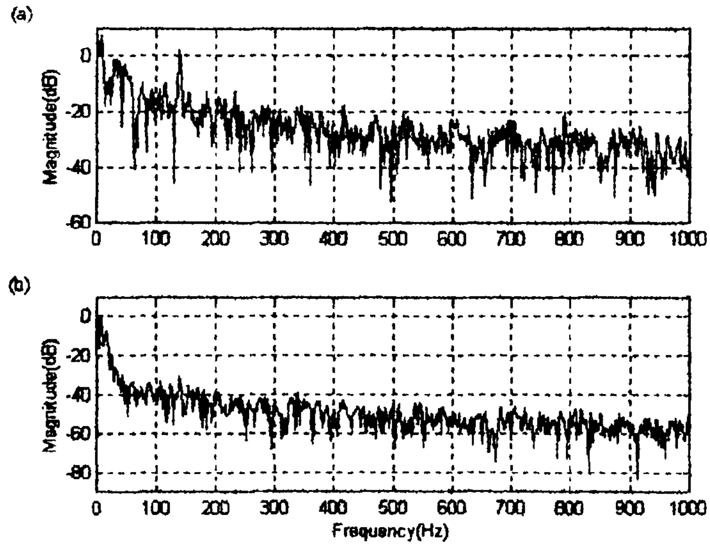

Figure 3-3 (a) unfiltered recorded signal, (b) target signal

In figure 3-3 the power spectral density (PSD) of an example of unfiltered signals and the target signals are shown. Both of these spectra are both dominated by the low frequency components.

3.3.2 Simulated Set

Pre-stimulus recordings, i.e. electrical activity recorded just before stimulation occurred, were used as a source of background activity. The background activity was added to a repeated set of the reference signal to create a set of simulated recordings comprising a known signal and noise. This simulated data (target signal and noise) set was split into a training set (55 responses) and a test set (56 responses). The pre-stimulus recordings were taken from the same site as the evoked potential recorded in data set 1. Activity recorded without the stimulation was considered a model of the noise that would be present during a recording of evoked potentials. The result was a set of signals, where the underlying signal was the same in all examples, but was corrupted by noise, which is different in each example. The simulated data sets used in some of the previous techniques have been a combination of a signal and white noise. The difference here is that the noise used in combination with the signal is the electrical activity when no stimulus is applied – no assumption are being made that the noise is a Gaussian noise source, only that it is reasonably representative.

Both the training and test subsets were subaveraged (i.e. split into groups of a set number of signals and each group produced an averaged signal) and the whole data set was used to produced an averaged signal. In figure 3-2, the upper right plots show two simulated signals, which are like those of the two example of recorded response, show a signal where the response is obscured by noise.

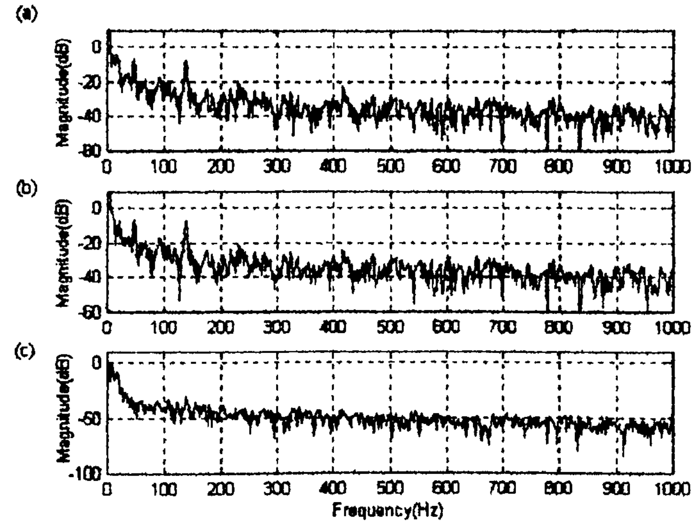

Figure 3-4 simulated data (a) 20^{th} signal, (b) noise and (c) target

In figures 3-4, the PSDs for a corrupted signal, noise and the target signal are shown. MATLAB command PSD was used with values as used previously. Comparing by inspection, the spectra of the noise and the noisy signal, it is difficult to see a considerable difference between the two spectra. The target signal and noisy signal are different, so this suggests noise dominate the simulated signal.

Figure 3-5 shows a comparison of an example of the recorded noise and a randomly produced signal. As there is a pre-filtering stage during recording, the noise is band-limited. Even considering this, there is a clear difference between the two signals, suggesting the noise can not be considered to have a Gaussian distribution which is an assumption that is often made.

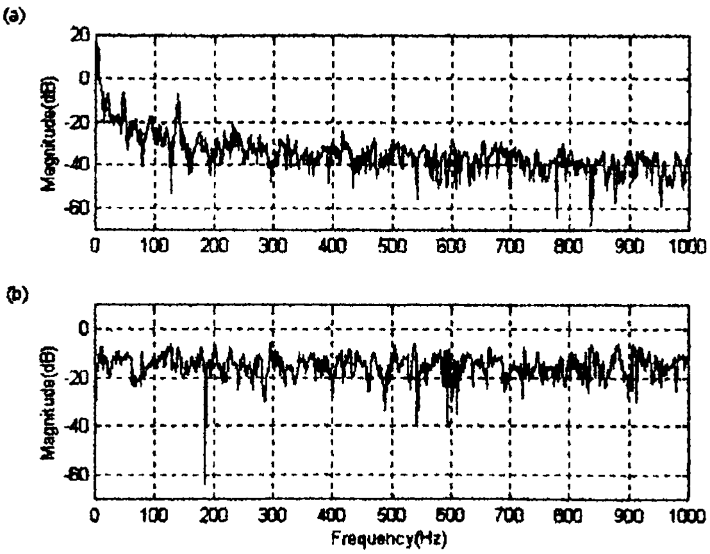

Figure 3-5 (a) recorded noise (b) random noise

3.3.3 Data set two

A second set of spinal recordings was produced from a different subject in the same way as the previous data set. Test set has 117 responses and the training set has 116 responses. A spectrum of an example is shown in figure 3-6.

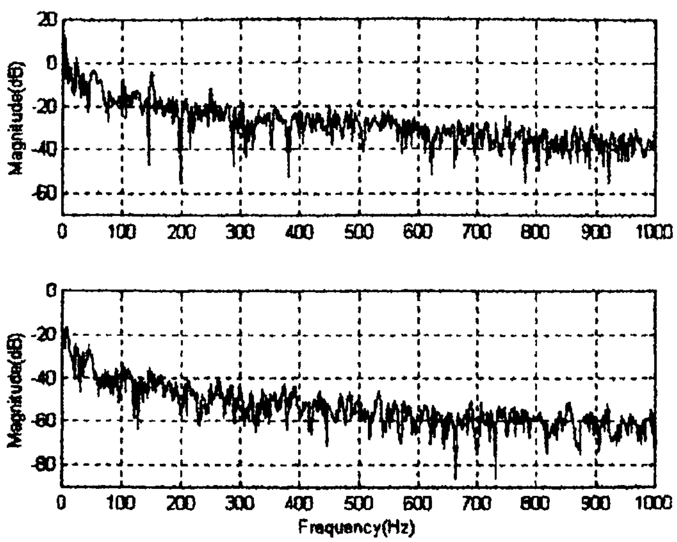

Figure 3-6 Example of the PSD of data set 2 signal (b) the target signal for data set 2

3.4 Scalp recordings

Two data sets were recorded from the scalp. Data set 3 contains 83 signals in training set and 83 in the test set. Data set 4 contains 122 signals in the training set and 121 signal in the test set. The target signals were formed from all the signals in each data set. Two examples of the power spectral density plots for signals from two data sets (figure 3-7 and 3-8) data set 3 and data set 4 respectively. The upper plot in each is an example of a single response, arbitrarily selected as the 20th in each test data set. The lower plots are the power spectral density plots of the target signals for each data set.

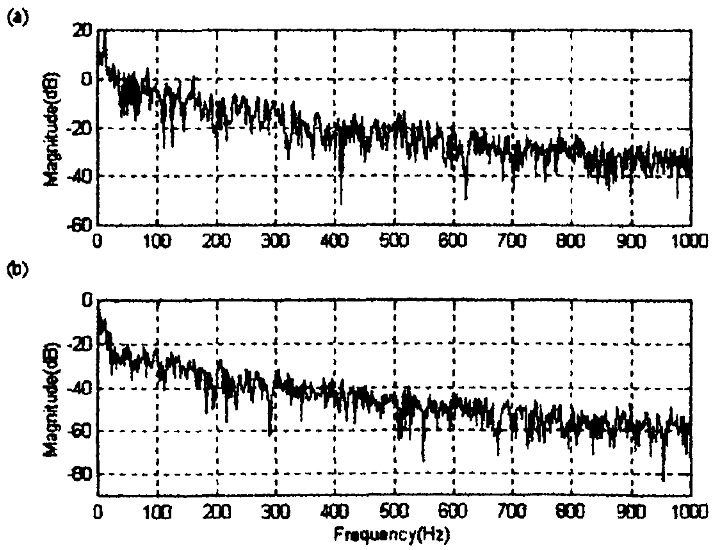

Figure 3-7 An example of the PSD of data set 3 signal (b) the target signal for data set 3

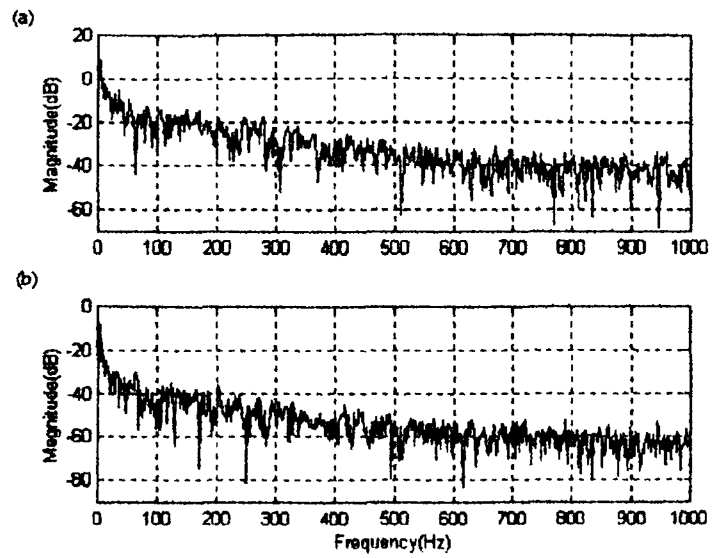

Figure 3-8 An example of the PSD of data set 4 signal (b) the target signal for data set 4

As with the spinal recordings, the differences between the PSD of the signals and the target signals are not clear apart from an overall decrease in the PSD magnitude for the target signal.

3.5 Discussion

Five data sets have been produced. Three data sets for spinal recordings (one of which is a simulated data set) and two data sets for scalp recordings. The spectra show that signal energy was concentrated at the lower end of the frequency spectrum. This is in agreement with some of the comments made by other groups about the use of high-pass filtering to remove low frequency components of the late latency signal components. The problem is that power spectra can not be used to confirm this, as information about where in time a particular feature occurs is lost with a power spectrum.

4 Signal Analysis and Filtering

In this chapter, the selection of an initial set of linear filters for the signals will be investigated. In the previous chapter, signal analysis was carried out in the form of the power spectrum. The problems of power spectra for selecting filters and alternative method that contains time information will also be discussed.

4.1 Filter Selection

The first approach to filter selection used the power spectrum of the averaged signal (Turner et al (1997) see Appendix D).

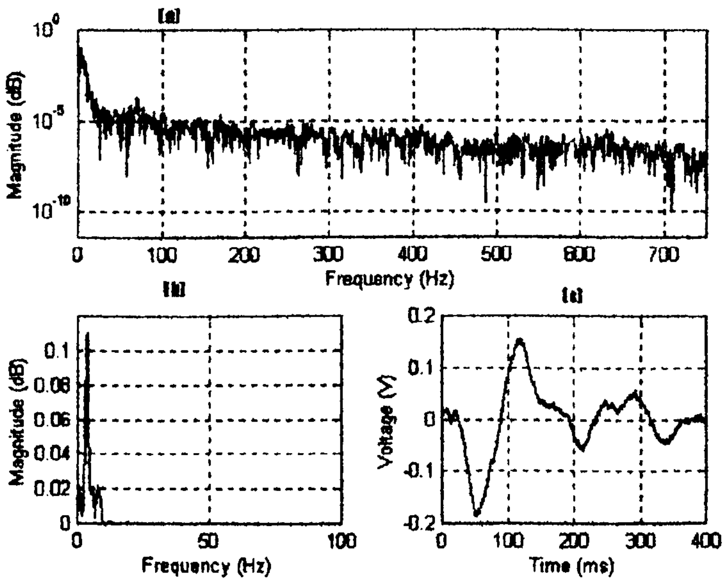

Figure 4-1 The target signal for the simulated data set and data set 1 (a) Power spectra of the averaged signal (target signal) (b) the low frequency components of the target signal, (c) the averaged signal.

The power spectrum of the target signal for two of the data sets (simulated and data set 1) is shown in figure 4-1. The upper graph shows the power spectrum of the target signal over the range 0 to 750 Hz. It can be seen that most of the signal power is in the lower frequency components. The lower spectrum shows that the majority of the signal's power is below 25 Hz.

Before looking at what the signals after filtering, it is worth considering what the signals look like before filtering. Figures 4-2 and 4-3 show test signals for the simulated and data set 1 respectively, both show three different sets of test signals.

The first is a set of examples where there has been no processing, in the second all signals were pre-processed as a set of averages of four signals and the third shows averages of 10 signals. From the theory of averaging, the more signals in an average, the better the SNR. This improvement can be seen by a comparison of the unaveraged and averaged signals, with that of the target signal (figure 4-1). There is a greater similarity between the target and the larger averaged results, than for the unaveraged test signals. Visual inspection is used here and throughout the thesis, because the assumption that will be made throughout the thesis is that the filtered signal must be similar in terms of position in time of peaks to the target signal. Inspection is a way to check this as well as demonstrating what the signals will look like.

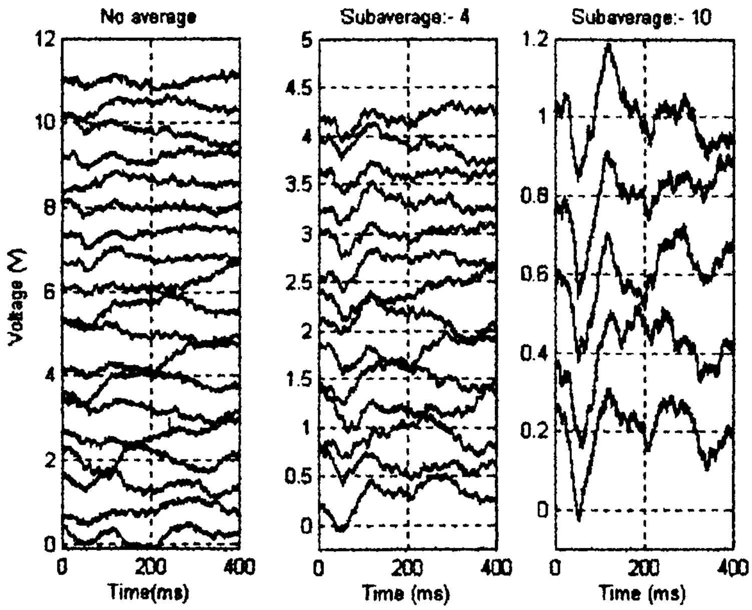

Figure 4-2 Set of simulated test signals before filtering.

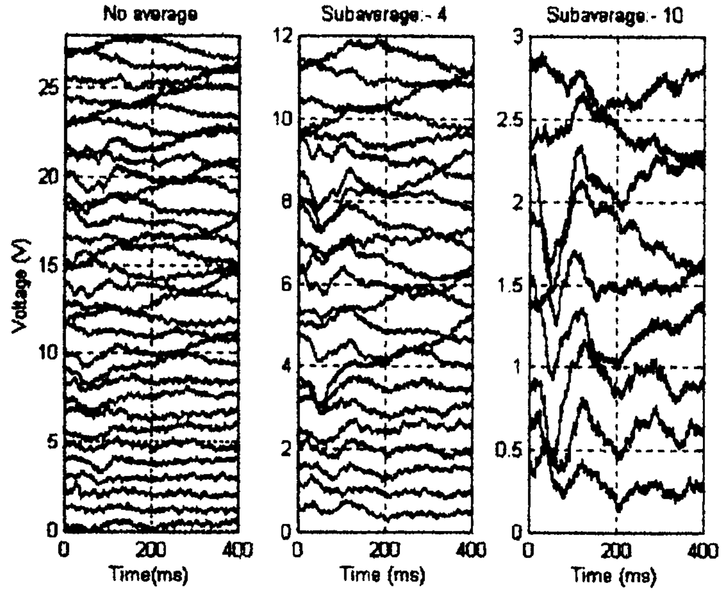

Figure 4-3 Set of test signals from data set 1 before filtering.

Initially a low-pass filter with a cut-off frequency of 30 Hz was used on these two test sets. Based on a visual inspection, the filtered results appeared to have unwanted low frequency components compared to the target signal. The literature was reexamined, to consider what the frequency components of other group's filters were, as a way of finding possible cut-off frequencies. Maccabee et al (1983) used a bandpass filter with a passband from 5 to 3000 Hz for what they called their 'open-pass filter'. In this study, the lower frequency (5 Hz) was adopted for this data set, but 30 Hz was used as the upper limit. Visual inspection is an appropriate method for comparing signals since the end-use are typically clinicians who make diagnosis based on visual inspection of the signal, looking for appropriate key features. Later in this chapter an additional numerical methods will be introduced which is also used to compare the waveforms.

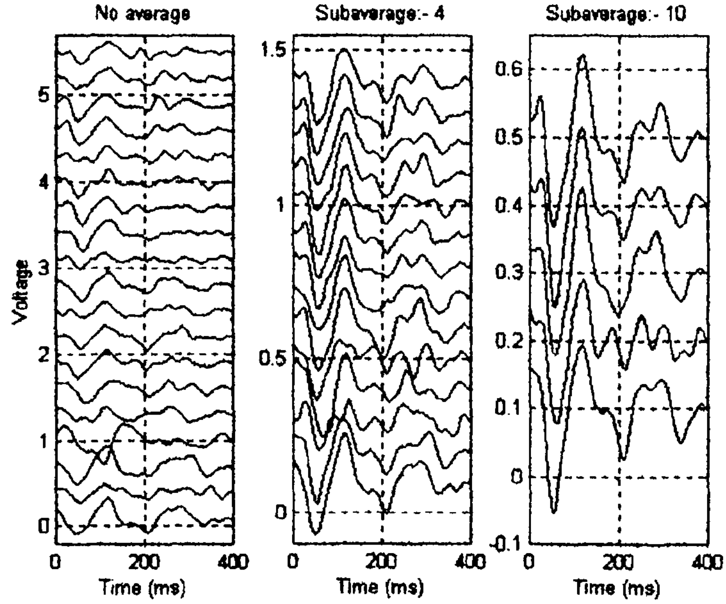

Figure 4-4 Filtered with a bandpass filter 5-30 Hz for simulated test set

Figures 4-4 and 4-5 show results of applying this filter to these two data sets. In both figures, the performance for the later large components of the signal indicates the filter's ability to improve the clarity of these peaks. However, the smaller early components are lost using these filters.

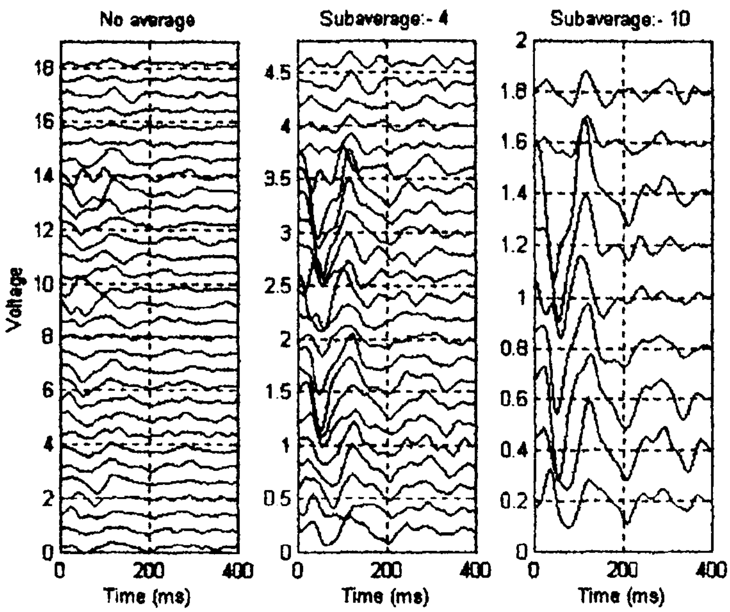

Figure 4-5 Filtered with a bandpass filter 5-30 Hz for the test set of data set 1

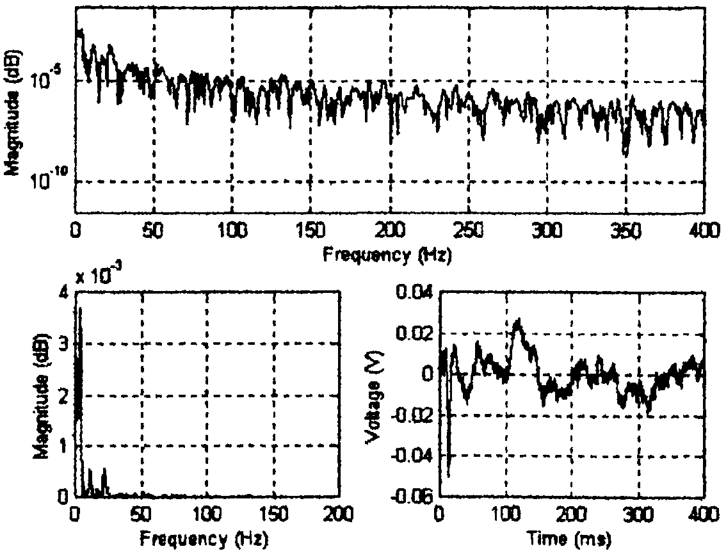

Figure 4-6 Power spectra for the target signal of data set 2 and the target signal

Figure 4-6 shows the power spectra of the target signal for data set 2. The spectra again show power that is predominantly in the low frequency range of the signals, up to around 60 Hz, so a bandpass filter with cut-off frequencies at 5 and 60 Hz was selected. Again, the lower cut-off frequency was selected from the work by Maccabee et al (1983). Figure 4-7 shows the results before filtering has been applied and there is not a great deal of similarity between these signals and the target signal.

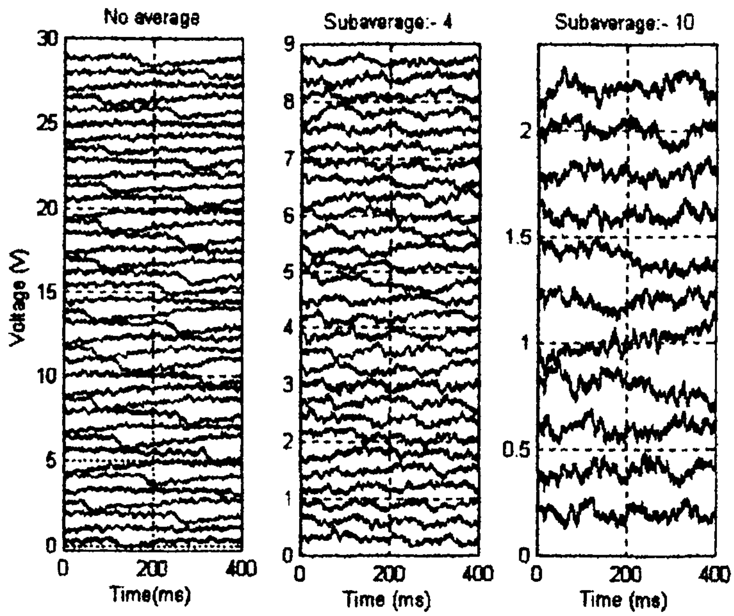

Figure 4-7 Set of test signals from data set 2 before filtering.

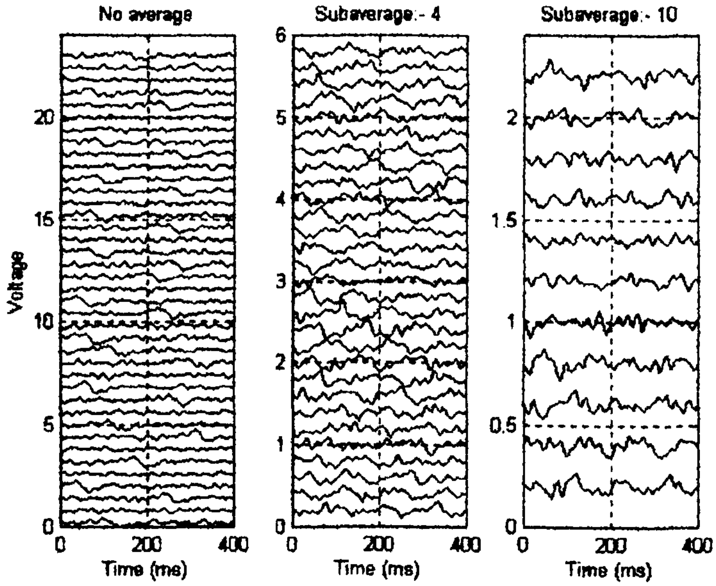

Figure 4-8 Filtered with a bandpass filter 5-60 Hz for the test set of data set 2

As could be expected, the filter (figure 4-8) has removed some of the noise but it is difficult to see similarities between these signals and the target.

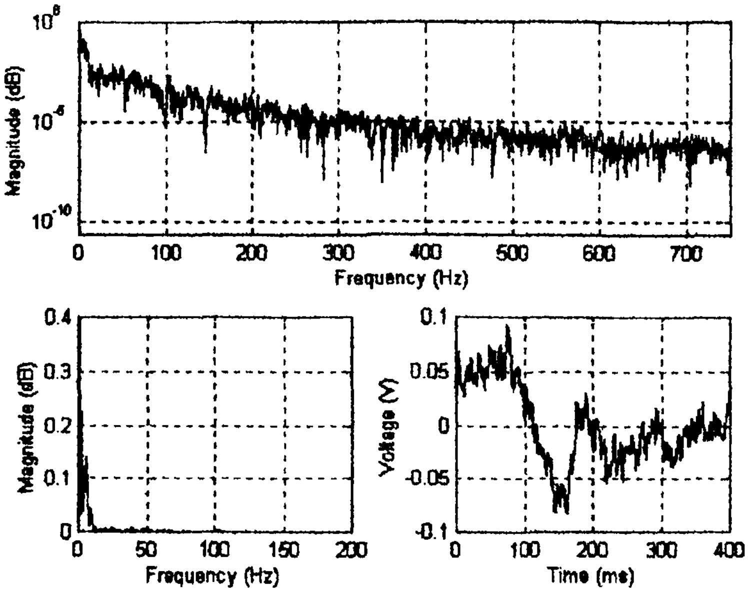

Figure 4-9 Power spectra of the target signal for data set 3 and the target signal.

Figure 4-9 shows the power spectra for the target signal of data set 3. Two possible filters were investigated. First, a low-pass filter up to 60 Hz because of some of the small components up to and around this frequency value (figure 4-11). The second low-pass filter has a cut-off frequency at 20 Hz, as most of the signal's energy occurs at frequencies below 20 Hz (figure 4-12). Visually the results of filtering with the 60Hz filter (figure 4-11) were 'noisier' than those of the 20 Hz filter (figure 4-12).

The peaks in the 20 Hz were similar to smoothed versions of those in the target signal (figure 4-9).

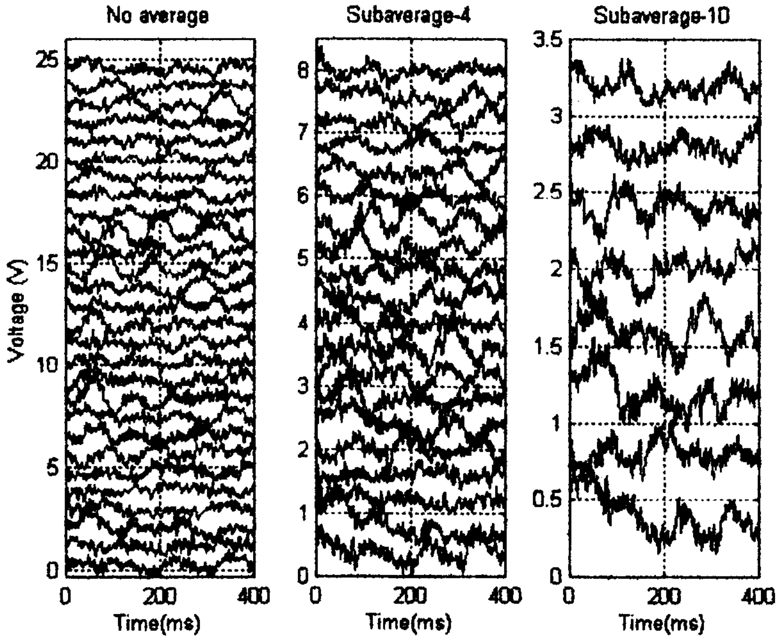

Figure 4-10 unfiltered data set 3 test signals

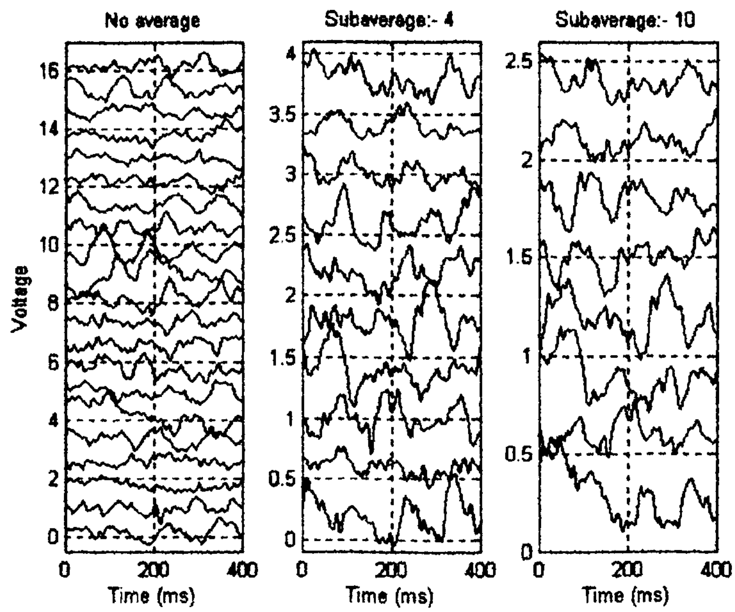

Figure 4-11 Filtering the test signals of data set 3 with a low-pass filter (60 Hz).

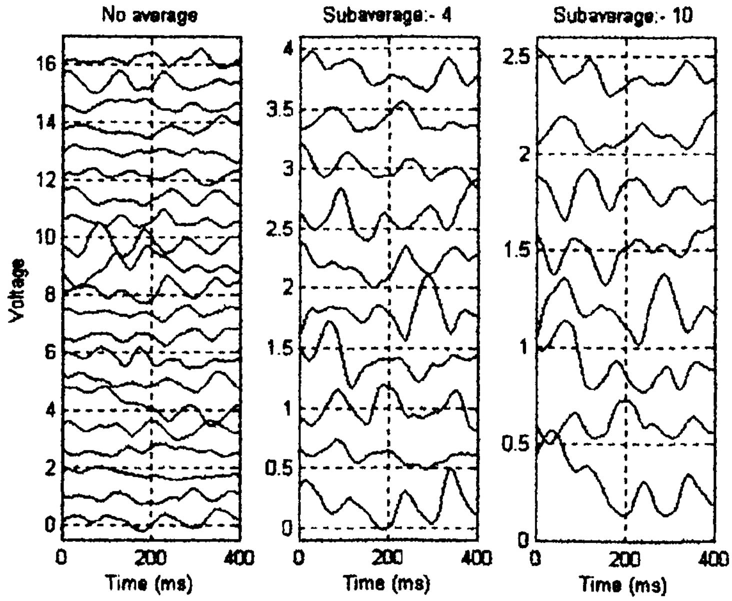

Figure 4-12 Filtering the test signals of data set 3 with a low-pass filter (20 Hz).

Figure 4-13 shows the target signal and its power spectrum of the target signal of data set 4 and the similar illustrations for the unfiltered test signal are shown in figure 4-14. Again, the signal energy is predominantly below 20 Hz as in data set 3.

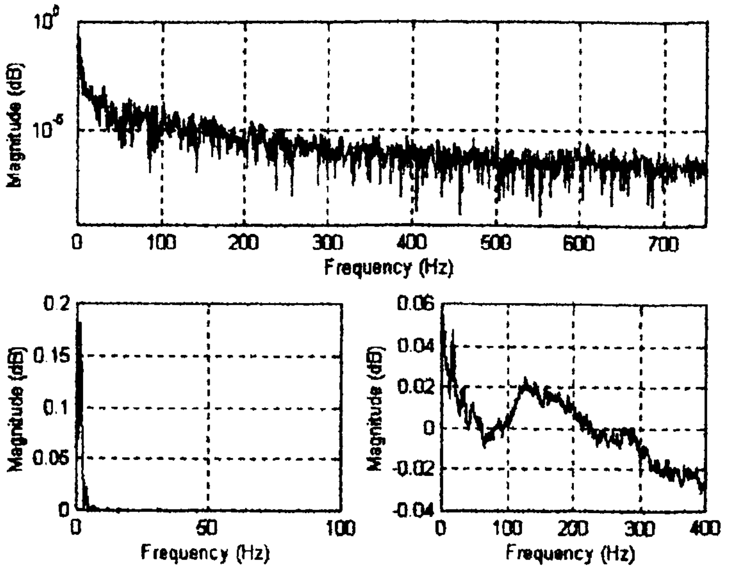

Figure 4-13 Power spectra for the target signal of data set 4 and the target signal for data set 4

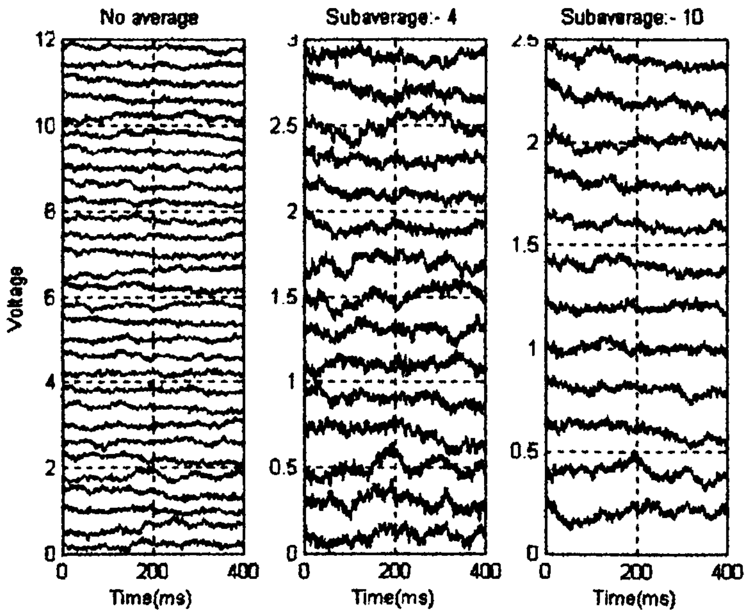

Figure 4-14 Unfiltered data set 4 test signals

Looking at the filtered results for this data set (figure 4-15) using a 20Hz low-pass filter in the subaverages of 10 signals, there appears to be a common peak to many of the signals in this set is between 100-200 ms. This can be seen in the some of the unfiltered signals results (figure 4-14), but not as clearly as in the filtered signal.

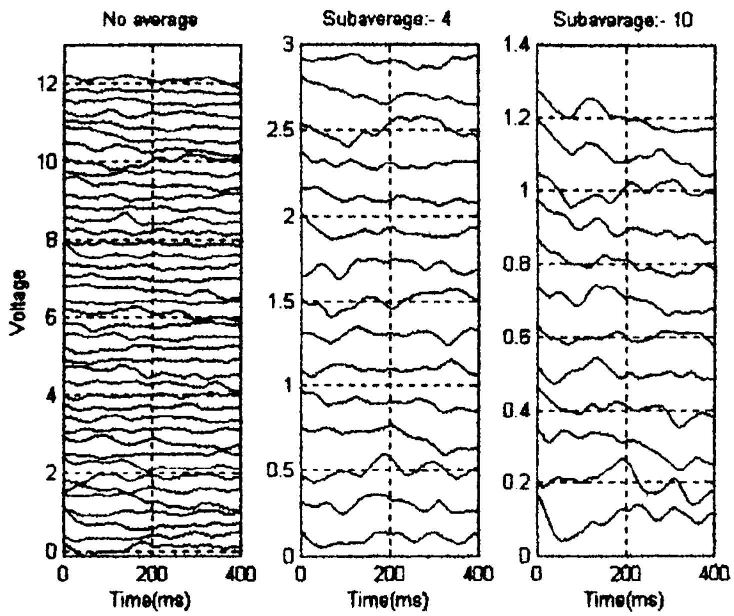

Figure 4-15 Filtered test signals of data set 4

Looking at figures for the filtered results, relatively small early components are often lost. This loss is probably due to the fact that the early components generally include high rate peaks than the later components, as these filters have removed high rate (and therefore usually higher frequency) components. One possible way around this is to build a bank of filters that occur at different times. This was the approach of Nishida et al (1993). They looked at the power spectrum of the target signal and suggested that there were three regions in the spectrum that occurred at three different times during the signal. There are problems with that approach, such as assuming that each spectral region is adjacent to the next. This raises the question of when does one region start and the previous region finish, something that cannot be derived from the power spectrum. This approach of making filter selections by power spectrum or the approach of Nishida et al (1993) therefore was not investigated further, due to this inability of the power spectrum to represent when the transition between regions occurs.

Mean squared error (equation 4-1) between the filtered signal and the target as a measure of the success of selecting filters using power spectra and averaging-alone. It requires only the assumptions that have already been made.

$$MSE = \frac{1}{N}\sum_{k=1}^{N} e(k)^2 \qquad \textbf{Equation 4-1}$$

Looking at the MSE values for the unfiltered averages (table 4-1) and for the filtered averages (table 4-2), it can be seen that filtering did produce lower MSE values.

4.2 Spectrograms

Spectral analysis using PSD assumes that the spectral properties are the same throughout the signal. An alternative is to use techniques that display variation in frequency with time. Spectrograms are one such technique, computing a windowed discrete-time Fourier transform of a signal using a sliding window.

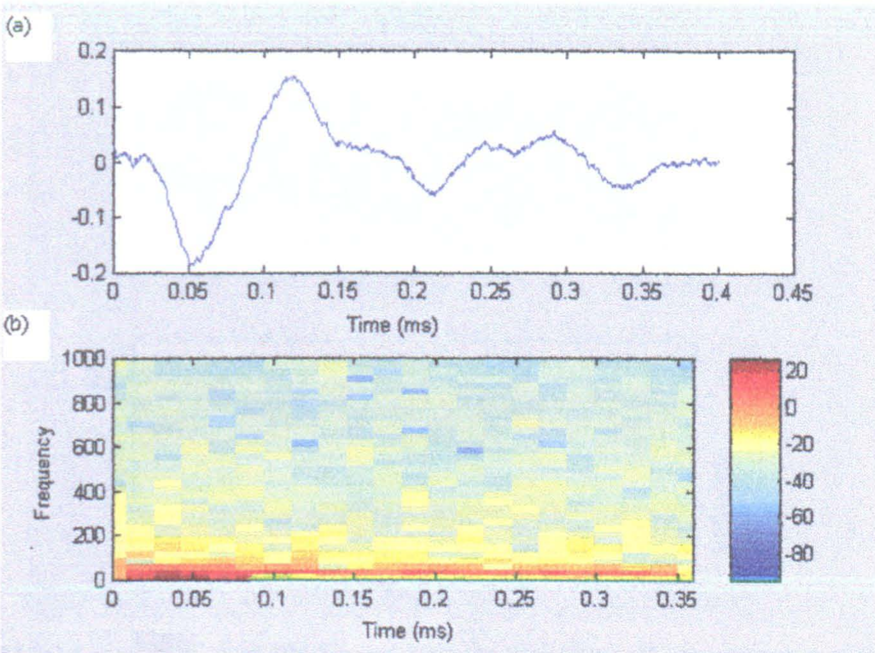

Figure 4-16 (a) target signal (data set 1), spectrogram of the target signal (a)

In figure 4-16, the upper plot shows the target signal for data set 1 and the lower graph shows the spectrogram of the same signal. The spectrogram shows that the signal contains predominantly low frequency components (<100 Hz), but that there is variation in the magnitude of frequency components with time. To illustrate the effect, a single continuous set of frequencies that is the same all through the signal is shown in figure 4-17, using two sine waves at 500 Hz and 2500 Hz. These were selected to show the spectrogram's ability to show the different sinusoids, both of equal magnitude.

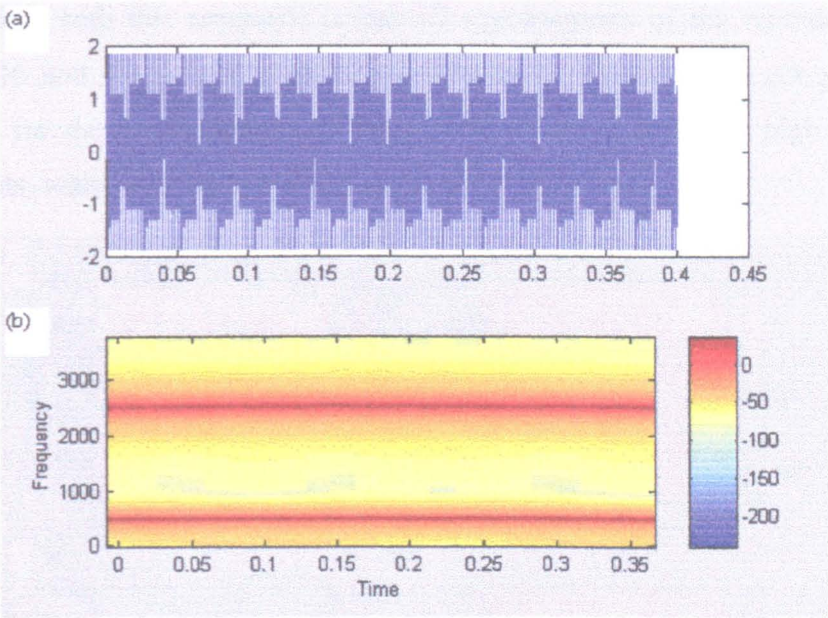

Figure 4-17 (a) a combination of 500 Hz and 2500 Hz sine waves, (b) spectrogram of (a)

Comparing figure 4-16 with 4-17 there is not a strong peak going through all the plots of figure 4-16, but for the low frequencies, there are relatively high values. Comparing the spectrogram of the background activity (figure 4-18) with that of the averaged signal for the same subject (figure 4-16). The most noticeable difference is that the background activity (figure 4-18) has a notable amount of high frequency activity (>400 Hz) compared to the target signal (figure 4-16).

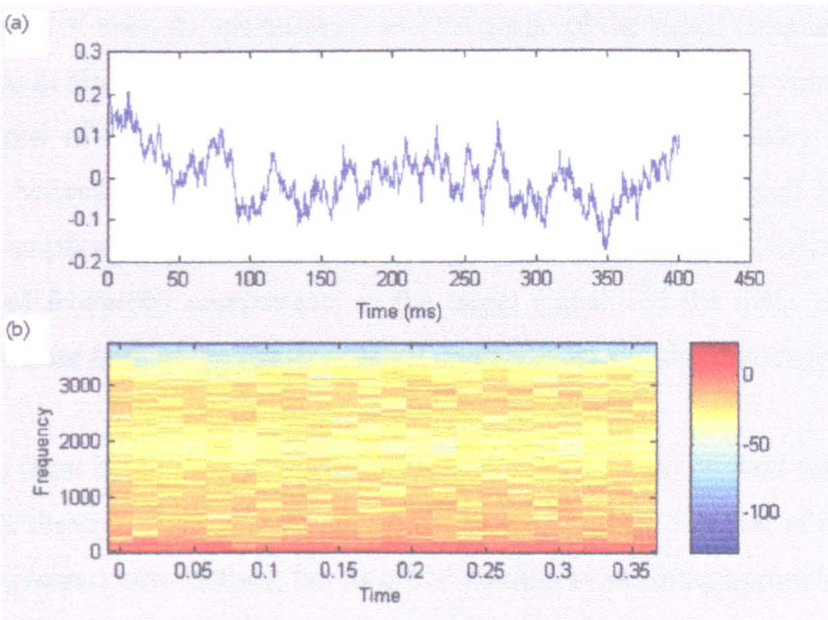

Figure 4-18 (a) An example of background activity (b) spectrogram of (a)

One problem with this approach is that all spectrograms of the recorded activity (figure 4-16 and 4-18) have a lot of low frequency activity. It is not possible to 'zoom-in' on the lower frequency (figure 4-19) more than the high frequency components, without losing the time resolution.

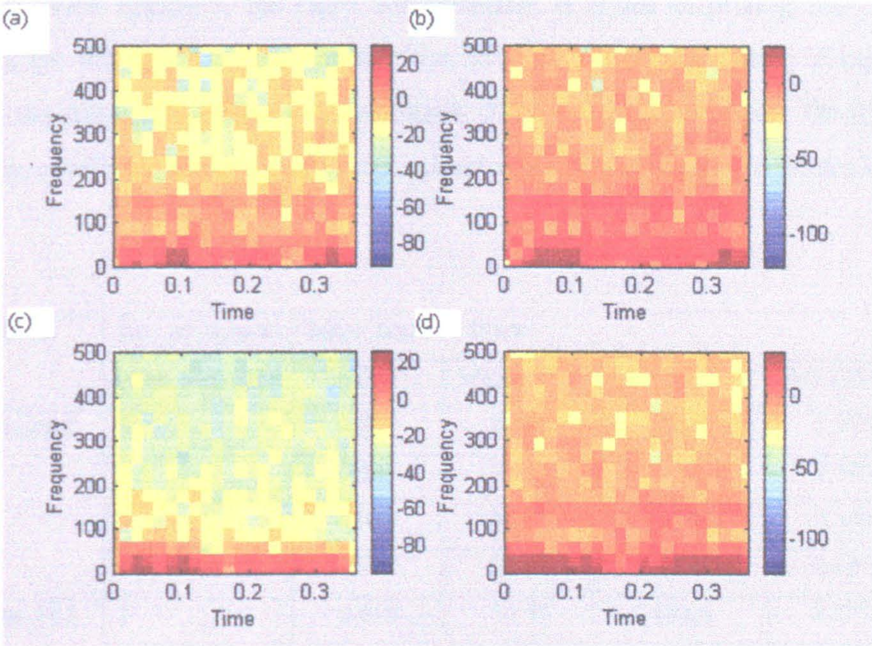

Figure 4-19 Spectrograms of (a) the single trial evoked potential (individual responses) , (b) and (d) two examples of background activity (noise), (c) of an averaged response.

4.3 Conclusions

Consideration of both the spectrogram and the shape of the signal (smaller high rate components at the beginning of the signal and larger lower frequency components at later portions of the signal), suggests these signals are nonstationary. Often the difference between the unfiltered signal spectra and the target signal is a –20dB change in amplitude. Looking at the individual recorded responses (figure 4-19(a)), the range of frequency components in the target signal and the noise do overlap. This can be seen both in the power spectral density plots and the spectrograms.

Selecting a filter based on a power spectrum has been a useful method for exploring some of the disadvantages and advantages of linear filtering. The aim of this section was not to claim a new method, but as a first attempt at selecting appropriate filters. A significant disadvantage is that care must be taken when interpreting the results as being peaks of an evoked response and not as noise. The advantages are that linear filters are easy to implement and understand. This work also justifies some of the decisions made by other researchers in their selection of cut-off frequencies for their filters to remove long latency (late components) effects. The filters generally seemed to be attempting to extract the late components. The region containing the late components is the biggest region of the signal often with the most dominant features (compared to the region of the early components). It is not surprising that a method that looks for where in a spectrum the majority of the signal energy occurs in the extracted the dominant features in the signal. This method works best for the signals such as the simulated and data set 1 test subset where averaging also works well.

Data set	No .of signals in an Average	Mean Squared Error			
		Min (10^{-2})	Max (10^{-2})	Mean (10^{-2})	Std (10^{-2})
Simulated	1	0.389	9.482	2.656	1.962
	2	0.140	7.710	1.830	1.590
	4	0.079	1.620	0.781	0.504
	10	0.097	0.295	0.181	0.086
Data set 1	1	0.896	38.37	10.66	8.277
	2	0.694	23.90	8.093	5.842
	4	0.746	9.851	4.261	2.971
	10	0.329	2.050	0.994	0.542
Data set 2	1	0.472	10.100	3.564	2.112
	2	0.456	3.952	1.78	0.963
	4	0.201	0.823	0.396	0.143
	10	0.071	0.267	0.141	0.073
Data set 3	1	1.766	33.890	6.666	4.622
	2	1.051	15.27	3.353	2.412
	4	0.619	4.841	1.688	0.971
	10	0.376	1.470	0.798	0.343
Data Set 4	1	0.199	6.630	0.651	0.714
	2	0.122	1.309	0.323	0.228
	4	0.055	0.372	0.159	0.084
	10	0.028	0.123	0.058	0.026

Table 4-1 Unfiltered results

Data Set	Filter (Hz)	No. of signals in an Average	Mean Squared Error			
			Min (10^{-2})	Max (10^{-2})	Mean (10^{-2})	std (10^{-2})
Simulated	5-30	1	0.049	1.630	0.251	0.256
		2	0.053	0.507	0.149	0.108
		4	0.652	0.234	0.105	0.048
		10	0.065	0.098	0.077	0.015
Data set 1	5-30	1	0.176	3.111	0.959	0.662
		2	0.163	2.432	0.645	0.467
		4	0.189	1.001	0.426	0.223
		10	0.168	0.538	0.280	1.180
Data set 2	5-60	1	0.178	1.587	0.627	0.284
		2	0.115	0.617	0.342	0.114
		4	0.103	0.297	0.176	0.049
		10	0.029	0.084	0.055	0.017
Data set 3	<60	1	0.837	32.023	5.343	4.441
		2	0.484	14.511	2.711	2.351
		4	0.325	4.513	1.381	0.942
		10	0.296	1.322	0.684	0.317
Data set 3	<20	1	0.354	30.112	4.344	4.105
		2	0.243	13.803	2.232	2.231
		4	0.209	4.191	1.154	0.887
		10	0.247	1.203	0.589	0.294
Data set 4	<20	1	0.053	6.412	0.489	0.703
		2	0.051	1.181	0.244	0.219
		4	0.026	0.319	0.121	0.081
		10	0.017	0.104	0.045	0.024

Table 4-2 Filtered results

5 Evolutionary Algorithms and Single Filters.

In the previous chapter a method was considered to aid the extraction of evoked potentials from noisy recordings by selecting a filter based on an averaged signal's power spectrum. In this chapter, an evolutionary algorithm is used to select a filter for each data set. The evolutionary algorithm is used to select the upper and lower cut-off frequencies of the filter's passband, as well as to select a weighting factor that is applied to the filter's output.

5.1 What is an Evolutionary Algorithm?

Evolution by natural selection is one of the most important and probably the most debated ideas in science. Computer scientists have looked at the idea of how processes based on evolution could be used as an optimisation tool for solving engineering problems. Goldberg (1989) provides a historical overview of evolutionary computation, suggesting a field that has a relatively long history, but which is still developing.

The most widely known of these are genetic algorithms, invented by John Holland in the late 1960s and developed further in the 1970s (Holland 1995). Genetic algorithms are based on the application of evolutionary concepts of natural selection, mutation and reproduction to select solutions to problems. Survival of the fittest solutions is the aim and is achieved by letting the fittest candidate solutions (or parts of the solution) pass into the next population of solutions. A set of solutions evolves over time, with 'fitter' individual solutions (individuals) adapting to their 'environment'. Some of the language of biology has crossed over into these applications. A population is a set of possible individuals. A chromosome is an individual in the population (i.e. a possible solution).

Genetic algorithms are often represented by a sequence of bits. Binary sequences are not the only possibility; the use of integers and floating point numbers is also possible. To minimise any confusion the term genetic algorithm is used here to refer to an algorithm based on binary sequences. The more general term, evolutionary algorithm is used here to refer to an algorithm based on non-binary sequences. This is discussed further in section 5.1.2.

Evolutionary and genetic algorithms have been used in an ever-growing variety of applications. Some related applications in signal processing include the selection of signals which will contribute to form an averaged signal (Laskaris et al., 1996), or modelling sources of electrical activity from scalp recordings (McNay et al, 1996, Aguiar et al, 2000). This approach has also been used to select parameters of a filter for glucose monitoring using infrared spectra (Schafer et al., 1996).

5.1.1 General overview of evolutionary algorithms

The design of an evolutionary algorithm involves

- Encoding the individual. Selecting the way the parameters are encoded within the individuals.
- Choosing a fitness function. The aim of using an evolutionary algorithm is usually to end up with the best individual, the algorithm needs to include a mechanism to determine this.
- Selecting how individuals will 'breed' and appropriate rates of mutation. Natural selection is not only about keeping the 'best' individuals, it is also about the individuals passing on their traits into the next population. So a mechanism is needed to select individuals and combine parts of these to produce a new individual. Changes (mutations) can also be introduced randomly to the sequence to produce new individuals.

Using an evolutionary algorithm involves

- Testing the fitness of the individuals in the population.
- Selection and crossover of sequences to form new individuals.
- 'Mutating' some of the elements in an individual.
- Iterating this process, until a certain condition is met. The 'best' individual is unlikely to be found in the first population, but by iterating through this process refining the individuals, a good individual can often be found.

5.1.2 Coded Sequence

The first stage of the process is to construct a set of coded sequences. A genetic algorithm is usually a sequence of binary digits or Gray code, but other representations are possible. It is usual to produce the initial population randomly. A binary representation allows the crossover to occur within a binary number. Crossover within a number allows the possibility of the value of a section of code to change considerably just by changing a single bit. Genetic algorithms using binary representation (when single-point crossover and binary mutation are used) can be considered robust algorithms (Davis 1991).

It has been stated that floating-point values are "intuitively closer to the problem space" (Michaelwicz 1996, pp. 106). Applications sometimes require the range of possible numbers to be large, at the same time maintaining precision. If implemented as a binary representation this could lead to long sequences of binary digits, so a floating point representation may be more appropriate. There is opposition to using floating point numbers; one of the arguments put forward by Goldberg (1989) is that it is distant from any biological precedent, especially from the notion of simple mutation. In terms of an analogy of modelling the chromosome as a string of coded letters, binary sequences (Holland, 1995) are closer to the idea of a gene, using a limited alphabet to encode the sequence. From this point of view, floating point representation is not really a 'genetic algorithm' as in the sense of the work of Holland (1995). However, it is the idea of selecting the best individuals in a group, breeding and mutating to solve the problem that is required, not to produce a model of how nature does the same thing. Goldberg did not question the usefulness of the floating-point representations, just their similarity to a biological system. Michalewicz (1996) conducted experiments comparing floating point and binary representations and concluded that floating point representation is faster, more consistent from run-to-run and had higher precision. Consequently, floating-point representations were used with this project.

5.1.3 Fitness Functions

A test is needed to measure the suitability of an individual sequence. In evolutionary terms, the test is a fitness function. The selection of the fitness function is an important factor in the success of the approach. In some applications, the choice of fitness function is obvious. For example if the global maximum of a function is needed, the fitness value is just the magnitude of the function. At other times it is not obvious what the function should be.

5.1.4 Selection and Crossover

Based on the fitness value for each sequence (individual), parts of one sequence are swapped with those of another. The fitter the individual the more likely that part or all of the individual will be passed onto the next generation. The roulette wheel approach is one method for selecting individuals; it gives a larger portion of the wheel to individuals that produce better fitness values. Therefore the larger the portion of the wheel, the higher the probability of the individual 'breeding' and passing on their 'genetic information' to the next generation. A pair of individuals is selected and at a randomly chosen point in the individual sequences, the pair swaps information, creating two new individuals. The process repeats until the required number of individuals forms a population. In an elitist strategy a number of the highest scoring (that is fittest) individuals go through to the next generation unchanged, the rest are formed using the selection and crossover operations. Figure 5.1 illustrates this function. The roulette wheel approach is not the only selection process, other options include tournament selection where selection is based on select from a small subset of the population each time.

5.1.5 Mutation

Mutation operates by randomly selecting an element in an individual and changing its value. In a binary sequence, mutation works by inverting a bit, a '1' becomes '0' and vice-versa. Mutation must also be applied when floating point data is used. Davis (1991) discussed two methods. The first was to replace the floating-point number in the chromosome with a randomly produced floating-point number. The second method is real number creep. An assumption is made that an individual that is reproducing is likely to be in a reasonably good position in the search space with respect to the rest of the population. What real number creep does is to change the value by a small random amount if the condition for mutation is met. The effect on the search was to enable the investigation of possible solutions close in the search space to those already present in the population.

```
3 5 0.1 9 20|0.5 10 90 0.1        Individual 1      -    Generation n
1 2 0.5 3 60|0.1 30 45 0.3        Individual 2
<——x1——>^<——x2——>
            |
        Crossover point

3 5 0.1 9 20 0.1 30 45 0.3        New individual 1  -    Generation n+1
1 2 0.5 3 60 0.5 10 90 0.2        New individual 2
                         ^
                         |__ Mutation
```

Figure 5-1 Outline of the evolutionary algorithm methodology

5.2 General Experimental Methodology

A description of the methodology of how the data sets were formed was included in chapter 3. After the recordings were formed into data sets, one half of the data sets (the training subset) were used to develop sets of filters via evolutionary algorithms. The other half of the data set is used to test the filters produced. All the evolutionary algorithms were developed and implemented in MATLAB (MathWorks, USA) on a Gateway 2000 Pentium P90.

5.2.1 The Filters

Each filter was a 4^{th} order Butterworth bandpass filter, implemented using the MATLAB command FILTFILT, which produces a zero-phase-shifting filter. Butterworth filters were selected for their relatively smooth pass-band (compared to that of the Chebyshev filter). A filter with a smooth pass-band was selected to lessen the effect of an artifact being introduced by the filter. The filters were set up so that initially the low frequency cut-off was within the range 0-200 Hz and the high frequency cut-off was selected to be 0-300 Hz higher than the low frequency cut-off. Averaging (or subaveraging) using a small number of single responses, was performed to reduce the noise level. By combining the signals into a set of averaged signals, the total number of signals used during training and testing phases were reduced, speeding up these processes. The aim of this phase of the work was to investigate the use of a single filter selected by an evolutionary algorithm to enhance the extraction of the evoked potentials.

5.2.2 Fitness functions

Measuring how well a particular sequence in a population performs is central to the evolutionary algorithm approach. Two methods were investigated to measure the similarity of filtered signals to the target signal. The first method was the correlation coefficient between the filtered signals and the target signal (an example of an averaged signal is shown in figure 5-2). The second method was the mean squared error (MSE) between the filtered signal and the target (equation 4-1). Both were considered because they are relatively simple methods, they require only the assumptions that have already been made. Mean square error was ultimately selected instead of correlation coefficient for two reasons. First, the scale of the signal is not taken in account by the nature of the correlation coefficient (see appendix A for a comparison using filter banks); in mean squared error, scale is maintained. Second, it can be argued there is a relationship between MSE and signal-to-noise ratio (see Appendix B.3).

In the training process, every example in the subaveraged training set had the filter specified by an individual sequence applied to it. The fitness of a particular sequence is the mean of all the fitness values of the examples for that particular sequence.

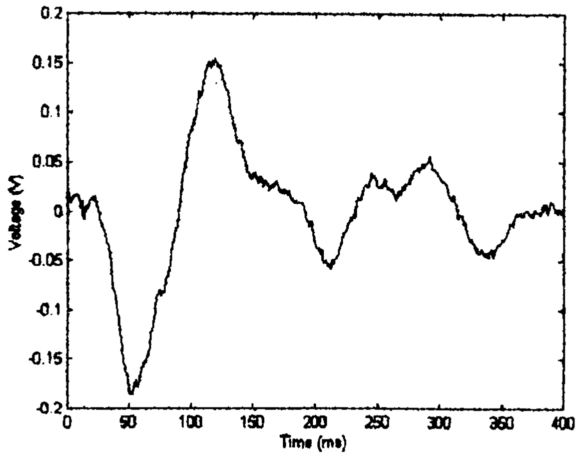

Figure 5-2 an example of an averaged signal, used as the target for data set 1 and the simulated data set.

5.2.3 Selection

A quarter of the original population goes through unchanged, selected as those with the best fitness values. A selection process produced the remaining three-quarters of the population. The selection process used in this work is the roulette wheel. Pairs of random numbers, ranging from one to the sum of all portions of the roulette wheel, were used to select the sequences that were the 'parents' of the next generation. A third random number was produced, that determines where along the sequence the swapping occurs, so the two original sequences produce two new sequences.

5.2.4 Mutation

A second operation was carried out that randomly selected elements of an individual sequence to alter. A randomly produced value (ranging from 0 to 1) was produced for each element in the population. If this value was less than or equal to the mutation rate then the corresponding element in the population is selected for mutation. The value in the population matrix was altered by between +/- 12% of the current value, with a 5% probability that an element would mutate (mutation rate=0.05). All sequences in the new population, except the sequence with the highest fitness value in the previous population, were subject to possible mutation.

5.2.5 Size of Population

In this chapter, a population size of 80 individuals was selected. This size was selected for various reasons. Firstly, a feature of the way the algorithm was implemented meant that the population had to be multiple of eight. Though there does not seem to be clear rules to how big the initial population needs to be, Haupt (1995) produced a rule of thumb for genetic algorithms that the population should be an order magnitude larger than the number of genes (or parameters). Grefenstette (1986) investigated population size, crossover rate and mutation rate for genetic algorithms, coming up with population sizes of 30 to 110 individuals depending on the application. Because of the nature of evolutionary algorithms, a large (relative to the number of parameters) population takes longer to process whereas a small population could cause the algorithm to converge too early. A population of 80 was selected as a trade off between these.

5.3 Single Filtering

Filters were developed for two groups of data. In the first group, none of the data sets were averaged (single trial). In the second group, the data was averaged in groups of 10 signals (subaveraging), done to investigate whether averaging ability to reduce the SNR can be useful in this work. Table 5-1 and 5-2 show the MSE of the filtered and unfiltered test subsets for both groups. Comparing the results of filtering without averaging (Table 5-1) and filtering with subaveraging in sets of 10 responses (Table 5-2), the mean MSE for the data set are lower when subaveraging was used. This effect can be seen in both the unfiltered and unfiltered results (Tables 5-1 and 5-2). Subaveraging lowers SNR, so it was included in the process, to improve the SNR of the training data before training.

	No. of	Frequency (Hz)		Weigh	Mean Squared Error (10^{-2})			
Data set	filters	Fx	Fx+1	t	Min	Max	Mean	Std
Simulated	0	-	-	-	0.3893	9.4816	2.6463	1.9622
	1	5.2342	14.0689	0.7746	0.0775	0.9855	0.1935	0.1494
1	0	-	-	-	0.8960	38.365	16.665	0.2272
	1	5.1211	11.9849	0.363	0.0748	0.8973	0.2981	0.1571
2	0	-	-	-	0.4724	10.111	3.5639	2.1122
	1	5.9414	133.8074	0.0104	0.0080	0.0093	0.0087	0.0003
3	0	-	-	-	1.7658	33.887	6.6581	4.6192
	1	0.6798	95.6960	0.0182	0.1044	0.1664	0.1282	0.0097
4	0	-	-	-	0.1992	6.6298	0.6508	0.7141
	1	57.8492	78.2683	0.0318	0.0240	0.0243	0.0241	0.0001

Table 5-1 Single filter single response

	No. of	Frequency (Hz)		Weight	Mean Squared Error (10^{-2})			
Data set	filters	Fx	Fx+1		Min	Max	Mean	Std
Simulated	0	-	-	-	0.0970	0.2954	0.1805	0.0859
	1	3.4687	27.5156	0.8896	0.0369	0.0804	0.0567	0.0211
1	0	-	-	-	0.3294	2.0501	0.9943	0.5429
	1	3.653	21.6969	0.6034	0.0522	0.3804	0.1739	0.1143
2	0	-	-	-	0.0707	0.2671	0.1412	0.0732
	1	5.0093	163.3893	0.1002	0.0073	0.0100	0.0082	0.0008
3	0	-	-	-	0.3761	1.4710	0.7983	0.3432
	1	3.744	7.0710	0.6613	0.0737	0.2225	0.1196	0.0542
4	0	-	-	-	0.0284	0.1230	0.0587	0.0263
	1	58.5554	77.8876	0.2744	0.0238	0.0242	0.0241	0.0001

Table 5-2 single filtering subaverages of 10 signals

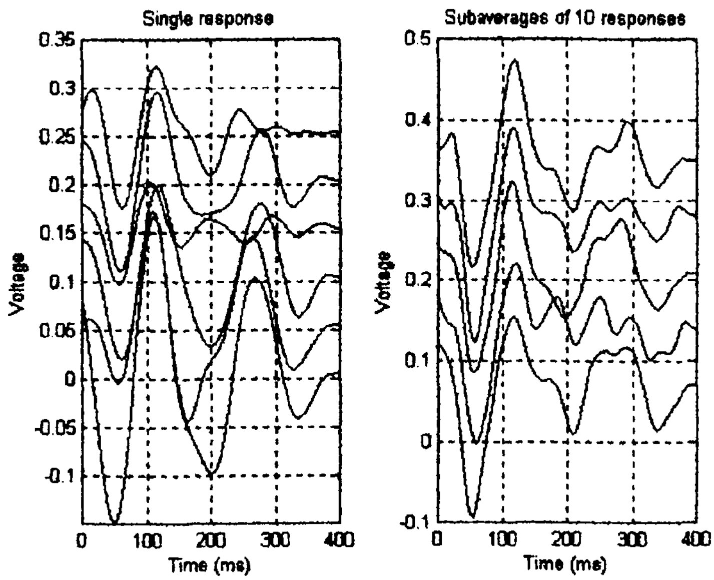

Figure 5-3 Single filter - simulated test data

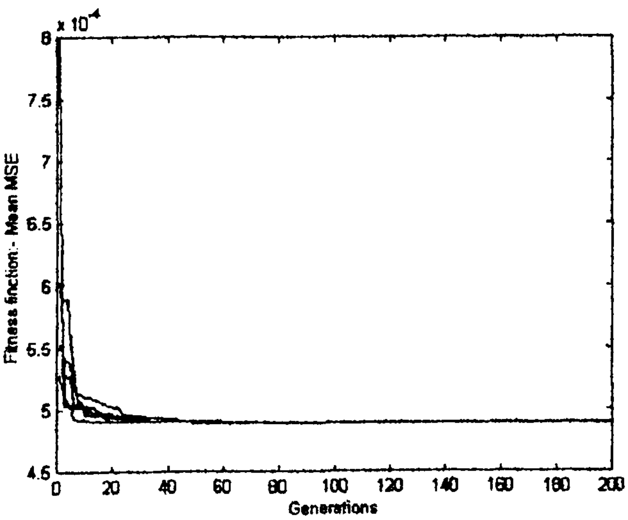

Figure 5-4 the variation in the fitness functions with the number of generations, for five runs for a single filter with the simulated data set.

As in chapter 4, a single filter did not extract all of the features of the evoked responses, but did enhance the extraction of some of the features in an evoked response. These results and those of chapter 4 showed there is an improvement when the number of responses averaged each time is increased. Examining tables 5.1 and 5.2, the results filtered using the filters selected by the evolutionary algorithm produced lower MSE values than when averaging-alone (unfiltered results) was used. An improvement with increasing numbers of signals in an average is in line with the theory of averaging, the SNR improves as the number of signals in an average increases. Ten responses per average were selected as the final size, as a trade off between possible improvements in the MSE with increasing the number of signals averaged and the number of examples in the test and training sets. Figure 5-4 shows how, the fitness values varied with the number of generation during five runs of the evolutionary algorithm for the simulated training data. This figure showed the results converged to similar fitness values for this set of data. Figures 5-5 to 5-7 and 5-9 show filtered response developed using an evolutionary algorithm for single and subaverages of 10 responses for data sets 1 to 4 respectively. Figures 4-2,4-3,4-7, 4-10, 4-15, from the previous chapter, are unfiltered results for test sets for the simulated data set and data sets 1 to 4 respectively. For both sets of results, filtering produced a clearer response in two of the spinal data sets (simulated data set and data set 1). Clarity is used here to mean that the key features that the clinicians are looking for are present, this usually supported by lower mean MSE values. In the scalp recordings and data set 2, the features extracted were not as clear as those observed in data set 1.

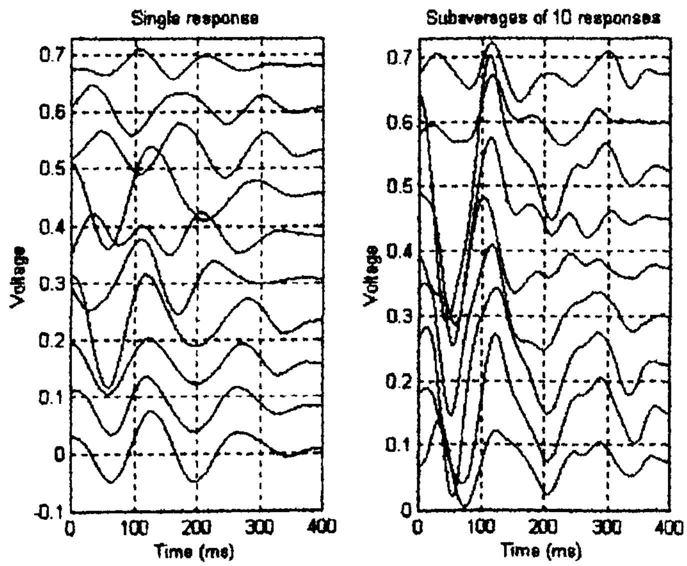

Figure 5-5 Single filter – data set 1 test set

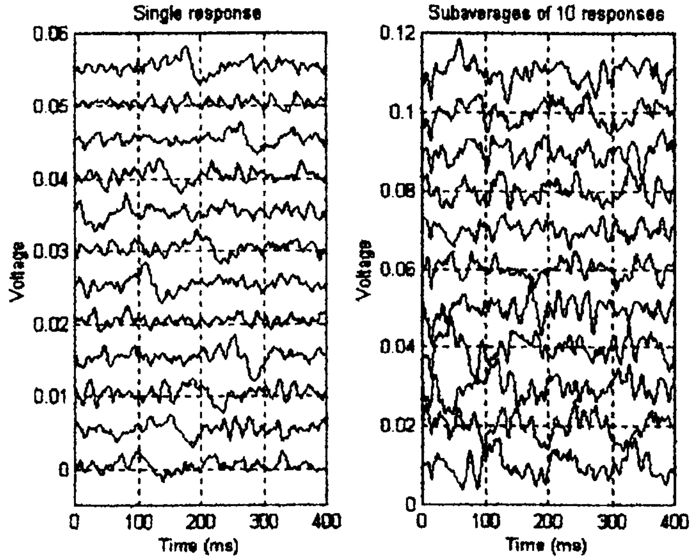

Figure 5-6 Single filter – data set 2 test set

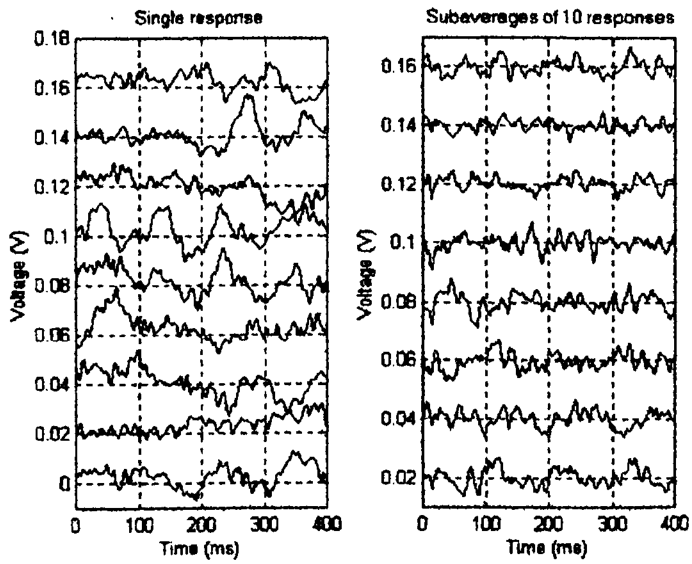

Figure 5-7 Single filter – data set 3 test set

Figure 5-8 shows in a similar way to figure 5-4 how the fitness function varied with the number of generations for the five runs of data set 3 training data. Unlike figure 5-4 all the runs did not converge to the same fitness values, but found two different values. One possible explanation is that the results are ending up in local minima in the search space. Figure 5-9 shows the filtered results for the test subset of data set 4. This is an example of the limitation of the single filter approach: the filter is trying to do too much; trying to have both low and high frequency components and not succeeding. Alternative strategies that may improve this are included in section 10.2.2.

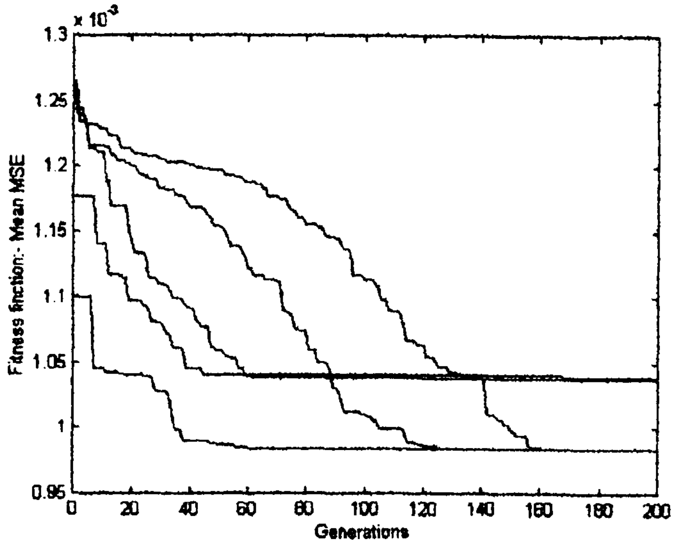

Figure 5-8 the variation in the fitness functions with the number of generations, for five runs for a single filter to filter data set 3.

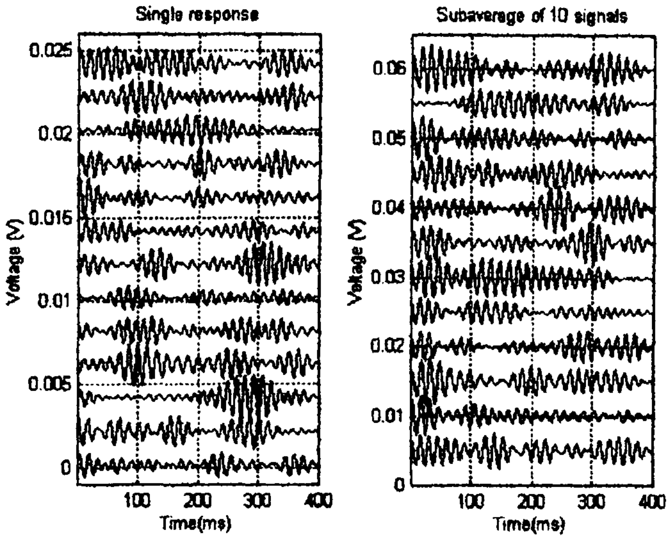

Figure 5-9 single filter - data set 4

	No. of	Mean Squared Error (10^{-2})			
Data set	filters	Min	Max	Mean	Std
Simulated	0	0.005	0.0455	0.0250	0.0179
	1	0.007	0.0207	0.0124	0.0056
1	0	0.031	2.1269	0.3659	0.0674
	1	0.0088	0.0319	0.0152	0.0075
2	0	0.0059	0.0654	0.0367	0.0212
	1	0.0185	0.0219	0.0204	0.0012
3	0	0.1104	0.8474	0.3155	0.2758
	1	0.0091	0.0093	0.0092	0.0001
4	0	0.0036	0.2531	0.0159	0.0094
	1	0.0156	0.0164	0.0160	0.0003

Table 5-3 first 30 ms

	No. of	Mean Squared Error (10^{-2})			
Data set	filters	Min	Max	Mean	Std
Simulated	0	0.094	0.2885	0.1819	0.0874
	1	0.0353	0.1005	0.0688	0.0287
1	0	0.3183	2.0462	0.9789	0.5538
	1	0.0447	0.4236	0.1696	0.1250
2	0	0.0707	0.2599	0.1341	0.0654
	1	0.0062	0.0093	0.0072	0.0009
3	0	0.3664	1.2566	0.7668	0.3005
	1	0.0759	0.2206	0.1104	0.0523
4	0	0.0294	0.1298	0.0564	0.0284
	1	0.0193	0.0194	0.0194	0.0001

Table 5-4 30-400 ms

Applying the filters to the first 30 ms and 30-400 ms as two separate signals shows the mean MSE values as lower for filtered results, the exception being data set 4. The second region 30-400 ms all produced lower MSE values for the filtered signals than averaging-alone.

Data Set	Selection method	Filter		No. of signals in an Average
		Fl (Hz)	Fh (Hz)	
Simulated	PS	5.00	30.00	1 and 10
	EA	5.23	14.07	1
	EA	3.47	25.52	10
Data set 1	PS	5.00	30.00	1 and 10
	EA	5.12	11.98	1
	EA	3.65	21.70	10
Data set 2	PS	5.00	60.00	1 and 10
	EA	5.94	133.81	1
	EA	5.00	163.39	10
Data set 3	PS	0.00	20.00	1 and 10
	EA	0.68	95.72	1
	EA	3.744	7.071	10
Data set 4	PS	0.00	20.00	1 and 10
	EA	57.84	78.27	1
	EA	58.55	77.89	10

Table 5-5 Comparison of the frequencies selected by using the power spectra (PS) and using evolutionary algorithms (EA).

In table 5-5 for the spinal recordings (simulated, data set 1 and data set 2) the filters selected by the power spectra method and those selected by an evolutionary algorithm (the results of the lowest scoring run for each of the data sets) had similar values for the lower limits. For the simulated data set and data set 1, the upper limits to the passband were also similar. The results for the scalp recordings show a difference in the frequencies select between the filters produced using the power spectra method and those produced using the evolutionary algorithm. Power spectra based approaches will extract the dominant features. The evolutionary algorithm approaches will also try to extract high-rate components that are small but can be observed in the averaged signal. The evolutionary algorithm developed filters tries to balance maintaining these components but filtering out of the high-frequency components due to noise.

6 Evolutionary Algorithms and Time-Invariant Filter banks

Previously, an evolutionary algorithm based approach was considered as a method to aid the extraction of evoked potentials from a set of noisy signals. In this chapter, that work is developed further by an evolutionary algorithm being used to select the parameters for a combination of filters, i.e. a filter bank.

6.1 Introduction

Some groups (e.g. Nishida et al 1983) suggest there may be three regions within the signal (section 2.2). A region at the beginning of the signal, with relatively stable components, has the highest frequency components of the three regions. A second region, following the first, has more variation in the position of features between responses and lower frequency components than the first region. In the third region, there is more variation in the position of the features than the previous regions and the frequency components are even lower than the other two regions. For this reason, the evolutionary approach was extended to three weighted filters, applied to the whole signal.

6.2 Whole Signal Filter Banks

6.2.1 Aim

The aim of this evolutionary algorithm is to select sets of filters, which enhance the extraction of evoked potentials from sets of noisy recordings, by filtering out the noise. A different filter bank will be produced by the algorithm for each data set, depending on the individual characteristics of those data sets. The noisy signal was passed through each filter in a bank of filters and output of the filter bank was a weighted sum of the individual filter outputs (Figure 6-1). The goal is to use three filters, ideally one for each of the three regions (early, middle and late components). To investigate whether additional filters would be an advantage the use of a bank of five filters is also considered.

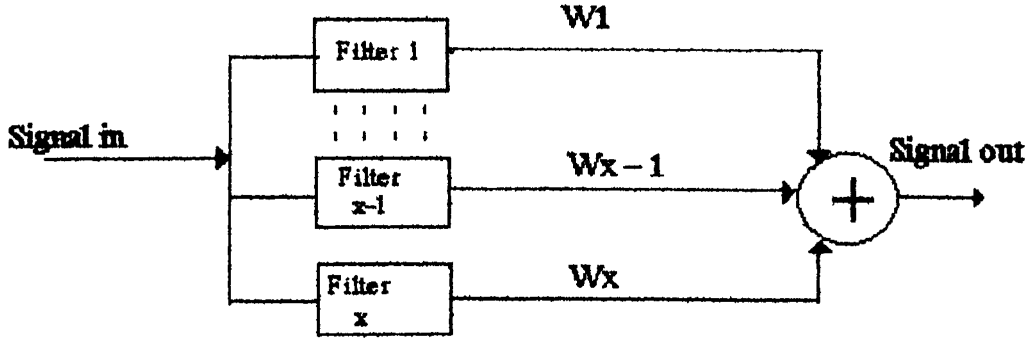

Figure 6-1 modelling the response as a set of x parallel filters

Each filter in the filter-bank was a 4^{th} order Butterworth filter of the same type as the filters used in chapter 5. As in chapter 5, initially data sets containing single trial and averages of 10 responses were employed. The results of using five filters in a filter bank were also investigated using simulated data and data set 1.

6.2.2. Preliminary study

A preliminary study of this approach was performed to determine whether three or five filters are needed and whether the use of filter banks was a valid direction for investigation. In the preliminary study, two data sets were used with both three and five filters (simulated data set and data set 1). The results (Table 6-1) showed that using five filters did not improve MSE values enough to be worth the extra processing of two further filters and weights. From these results, filter banks of three filters were chosen as the appropriate size of filter-bank for this approach. The use of fewer filters in the filter bank was not considered, as there are believed to be three regions.

Data set	No. of filters	Frequency (Hz)		Weighting	Mean Squared Error ($x10^{-2}$)			
		F_L	F_H		Min	Max	Mean	Std
Simulated	0	-	-	-	0.0970	0.2954	0.1805	0.0859
	3	4.02	26.97	0.9307	0.0430	0.0804	0.0603	0.0180
		66.75	576.81	0.2848				
		133.40	637.09	0.0563				
	5	199.18	209.62	0.2971	0.0420	0.0806	0.0605	0.0180
		72.83	91.66	0.4007				
		80.29	200.64	0.1753				
		4.04	27.08	0.9353				
		140.48	1796.10	0.2170				
1	0	-	-	-	0.3294	2.0501	0.9943	0.5429
	3	3.71	18.43	0.6253	0.0540	0.3769	0.1731	0.1124
		52.79	55.65	0.3097				
		42.55	260.14	0.0864				
	5	124.00	167.57	0.0984	0.0551	0.3754	0.1730	0.1121
		312.06	409.82	0.1387				
		178.09	231.15	0.141				
		238.14	303.94	0.1428				
		3.79	19.25	0.6196				
2	0	-	-	-	0.0707	0.2671	0.1412	0.0732
	3	13.13	15.35	0.156	0.0079	0.0083	0.0083	0.0003
		152.54	155.55	0.63				
		20.54	231.27	0.109				
3	0	-	-	-	0.3761	1.4701	0.7983	0.3432
	3	158.73	161.73	0.1478	0.0927	0.3153	0.1842	0.0722
		162.95	801.79	0.0702				
		0.68	89.95	0.4500				
4	0	-	-	-	0.0294	0.1298	0.0564	0.0284
	3	58.48	77.55	0.2670	0.0238	0.0242	0.024	0.0001
		107.08	522.14	0.0640				
		272.20	274.88	0.2180				

Table 6-1 Preliminary filtered and unfiltered results for the whole signals for all test subsets

Data set	No. of filters	Mean Squared Error (10^{-2})			
		Min	Max	Mean	Std
Simulated	0	0.0050	0.0455	0.0252	0.0179
	3	0.0055	0.0198	0.0116	0.0057
1	0	0.0311	2.1269	0.3659	0.6744
	3	0.0082	0.0173	0.0125	0.0038
2	0	0.0059	0.0654	0.0367	0.0212
	3	0.0186	0.0216	0.0202	0.0010
3	0	0.1104	0.8474	0.3155	0.2758
	3	0.0105	0.1212	0.0461	0.0465
4	0	0.0036	0.2531	0.0159	0.0094
	3	0.0151	0.0163	0.0158	0.0003

Table 6-2 First 30 ms using filters banks selected during the preliminary study.

Often clinicians look at these signals on two time scales, the early components and the late components. The filter banks and single filter (chapter 4) were applied to the first 30 ms of the test signals, to look at the early components. An improvement in

MSE values of the filter bank approach (Table 6-2) compared to only using a single filter (see table 5-3), was seen in four out of the five data sets using this shorter region. The single filtered signals predominantly contained low frequencies. Using three filters showed further features that can also be seen in the target signal are included (for example figure 6-2). This improvement is therefore believed to be due the extra filters enabling more than one band of frequencies to be included, so a combination of low and high frequency components can be included in the same filtered signals. In figure 6-3 the data set (data set 3) is shown in which this improvement in MSE value for the first 30 ms was not seen is shown. Both filtering methods did not extract anything that looked like the target. This data set is challenging, by examining the signals before filtering (see figure 6-13), little similarity can be seen with the target signal. These results suggest the three-filter approach has shown potential in extracting the signal's earlier components.

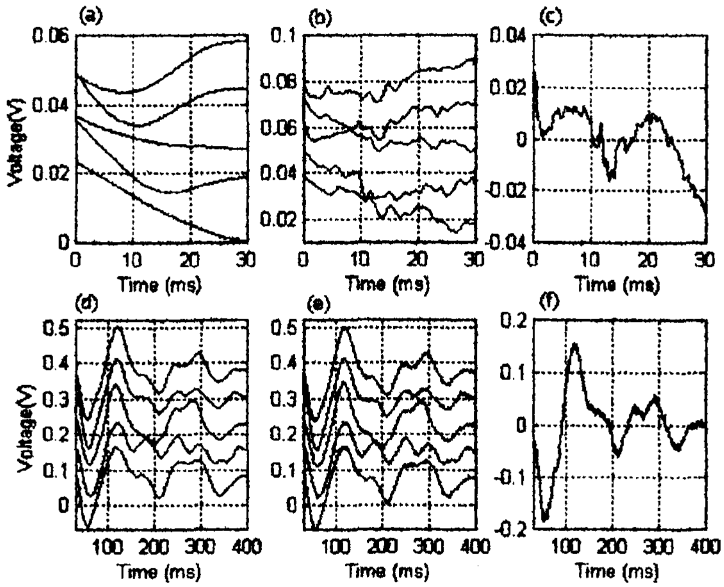

Figure 6-2 simulated preliminary results versus single filters. The first 30 ms (a) using a single filter (b) using a filter bank of three filters, (c) of a target signal. The region for the 30-400 ms (d) using a single filter (e) using a filter bank of three filters, (f) of a target signal.

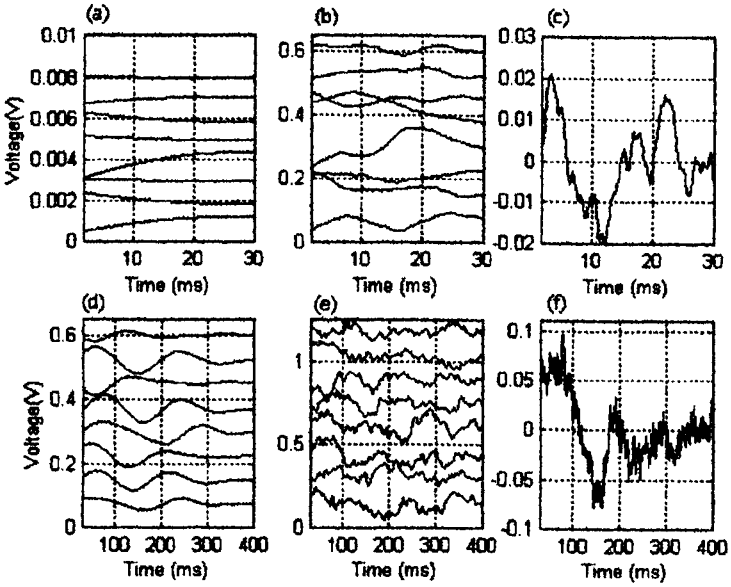

Figure 6-3 Data set 3 preliminary results versus single filters. The first 30 ms (a) using a single filter (b) using a filter bank of three filters, (c) of a target signal. The region for the 30-400 ms (d) using a single filter (e) using a filter bank of three filters, (f) of a target signal.

Data set	No. of filters	Mean Squared Error (10^{-2}) Min	Max	Mean	Std
Simulated	0	0.0940	0.2885	0.1819	0.0874
	3	0.0403	0.1007	0.0710	0.0261
1	0	0.3183	2.0462	0.9789	0.5538
	3	0.0458	0.4199	0.1682	0.1233
2	0	0.0707	0.2599	0.1341	0.0654
	3	0.0069	0.0081	0.0073	0.0003
3	0	0.3664	1.2566	0.7668	0.3005
	3	0.0922	0.3388	0.1176	0.0794
4	0	0.0294	0.1298	0.0564	0.0284
	3	0.0192	0.0194	0.0193	0.0001

Table 6-3 30-400 ms using filters banks selected during the preliminary study.

The results of the three-filter bank (Table 6-3) and single filters (Table 5-4) were similar (change of between 0.8-6% in mean MSE compared to the mean MSE when the whole signal was filter) for 30-400 ms region of the signals.

6.2.3. Filter banks applied to the whole signal

Based on the preliminary results a large trial was performed. This time five runs for each test subset were performed, going up to 400 generations, with a population of 200 individuals. Tables 6-4 and 6-5, show the results for single trial and subaveraged (10 signals) respectively. One goal of this study was to investigate the number of signals used in a subaverage. As has been previously observed increasing the number of signals in an average can improve the extraction of the responses and this was observed for this process as well.

Table 6-5 shows the results of using subaverages of 10 signals and the MSE between the filtered test subsets and their respective target signals. The unfiltered results have higher MSE values than the filtered results (Table 6-5). This suggests that the filtering process is producing signals that are closer to the target signal. This was also shown by a visual inspection of the results for the filtered signal (figure 6-4, 6-5 and 6-6). The mean value of each averaged signal had its mean amplitude subtracted from each value, to rule out the possibility of the larger MSE being due a DC level in the averaged signals.

Data Set	No of filters	Frequencies (Hz)		Weights	Best mean MSE 10^{-2}				Variations in mean MSE 10^{-2}			
		F_L	F_H		Min	Max	Mean	Std	Min	max	mean	std
Simulated	0	-	-	-	0.3893	9.4816	2.6463	1.9622	-	-	-	-
	3	5.87	7.41	3.2744	0.055	1.237	0.1578	0.1749	0.1562	0.1686	0.1602	0.0049
		9.78	16.29	0.6558								
		11.26	60.24	0.0596								
1	0	-	-	-	0.896	38.3685	10.6555	8.277	-	-	-	-
	3	5.54	7.97	0.6437	0.0615	0.9944	0.2834	0.1724	0.2834	0.3002	0.2947	0.0069
		162.40	215.51	0.0156								
		9.27	17.46	0.2045								
2	0	-	-	-	0.4724	10.1105	3.5639	2.1122	-	-	-	-
	3	5.70	47.68	0.1036	0.0088	0.0222	0.0151	0.0036	0.0151	0.02	0.0165	0.0021
		164.01	527.88	0.0547								
		63.36	141.36	0.1735								
3	0	-	-	-	1.7658	33.8873	6.658	4.619	-	-	-	-
	3	1.42	3.06	0.1877	0.0684	0.277	0.1243	0.028	0.1243	0.1301	0.1279	0.0022
		147.27	148.65	0.1963								
		149.43	286.29	0.0064								
4	0	-	-	-	0.1992	6.6298	0.6508	0.7141	-	-	-	-
	3	0.03	8.78	0.017	0.0196	0.0336	0.0236	0.0019	0.0236	0.0241	0.0239	0.0002
		80.23	273.31	0.0094								
		60.18	282.58	0.0194								

Table 6-4 Whole signal model filtering whole signal (single trial)

Data Set	No of filters	Frequencies (Hz)		Weights	Best mean MSE 10^{-2}				Variations in mean MSE 10^{-2}			
		F_L	F_H		Min	Max	Mean	Std	min	Max	mean	std
Simulated	0	-	-	-	0.097	0.2954	0.1805	0.0858	-	-	-	-
	3	3.04	4.42	1.7682	0.0269	0.0806	0.0521	0.0218	0.0521	0.0674	0.0599	0.0058
		0.11	64.90	0.1096								
		5.60	20.02	0.8996								
1	0	-	-	-	0.3294	2.0501	0.9943	0.5424	-	-	-	-
	3	3.59	23.42	0.4260	0.0334	0.3686	0.1705	0.1114	0.1705	0.1733	0.1726	0.0012
		58.37	397.29	0.0900								
		5.56	7.76	0.5780								
2	0	-	-	-	0.0707	0.2671	0.1412	0.0732	-	-	-	-
	3	5.90	15.61	0.1293	0.0072	0.0098	0.0081	0.0008	0.0081	0.0084	0.0082	0.0001
		18.81	240.90	0.1006								
		83.73	85.01	0.2230								
3	0	-	-	-	0.3761	1.47	0.7983	0.3432	-	-	-	-
	3	1.45	2.66	2.3792	0.0384	0.2036	0.1061	0.0580	0.1061	0.1125	0.1100	0.0034
		25.57	329.72	0.0543								
		4.69	7.73	0.4549								
4	0	-	-	-	0.03	0.12	0.06	0.03	-	-	-	-
	3	58.163	77.21	0.2772	0.0206	0.0249	0.0223	0.0013	0.0223	0.0239	0.0229	0.0006
		5.57	6.25	2.7908								
		105.24	525.68	0.0633								

Table 6-5 Whole signal model filtering whole signal (averages of 10 signals)

Data Set	No of filters	first 30 ms (10^{-2})				30-400 ms (10^{-2})			
		Min	Max	Mean	Std	Min	max	mean	std
Simulated	0	0.005	0.0455	0.025	0.0179	0.094	0.2885	0.18193	0.0874
	3	0.0063	0.0158	0.0102	0.0036	0.0235	0.0862	0.0596	0.027
1	0	0.031	2.1269	0.3659	0.6744	0.3183	2.0462	0.9789	0.5538
	3	0.0089	0.0196	0.0132	0.0043	0.0334	0.4091	0.1601	0.1209
2	0	0.0059	0.0654	0.0367	0.0212	0.0706	0.2599	0.1341	0.0654
	3	0.0188	0.0217	0.0205	0.001	0.0063	0.0088	0.0071	0.0008
3	0	0.1104	0.8474	0.3155	0.2758	0.3664	1.2566	0.7688	0.3005
	3	0.0065	0.0114	0.009	0.0016	0.0360	0.1871	0.096	0.0524
4	0	0.0036	0.031	0.0159	0.094	0.02941	0.1298	0.0564	0.0284
	3	0.0151	0.0163	0.0158	0.0003	0.0192	0.0195	0.0193	0.0001

Table 6-6 Whole signal model filtering partials

In figure 6-4 the process was shown to be able to extract approximations of the target signal for the simulated data. In this section, the same process is applied to the four sets of recorded activity, each of which is averaged to form a target signal and split into two subsets for training and testing. To look at differences in the recording sites, two spinal recordings and two scalp recordings were used. All signals were subaverages of ten signals in accordance with the results of the trials using two sizes of averages (Tables 6-4 and 6-5).

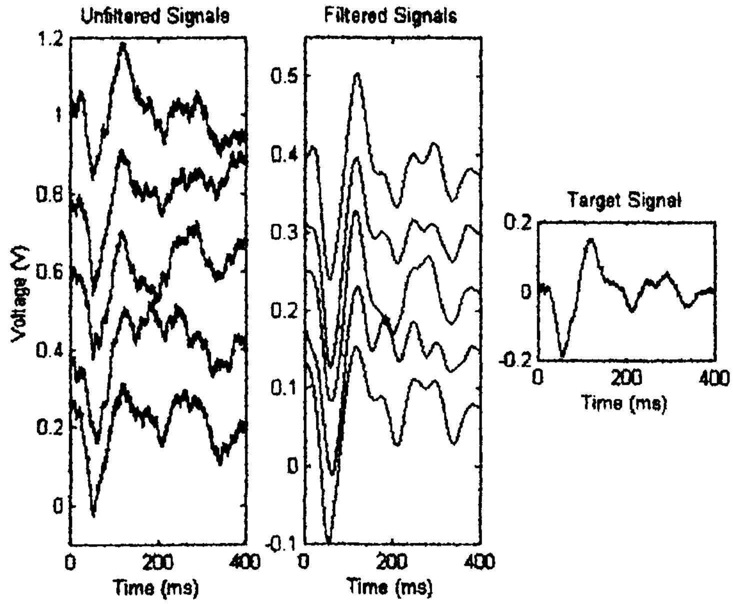

Figure 6-4 simulated data, whole signal model.

Data set 1 was used to form the target signal (Figure 6-5) and the underlying signal in the simulated data set, so a comparison between the two can be made, as they both have the same target signal. The results of the simulated data and results of the filter-bank produced using data set 1, show some common features (figure 6-5). The larger features are present in both. Smaller features at the beginning of the signals and near the end of the signal between 200 and 300 ms are more attenuated in data set 1 (figure 6-5) than those observed in the simulated data sets results (figure 6-4). There are two likely reasons for this. The first is time variation in the position of the features. Looking at the position of the largest positive feature between 100 and 200 ms, it is not always in the same place in each signal. To produce an average signal, variations in the position of a feature can lead to smoothing of smaller features. The second reason is low frequency noise which is not completely removed by this approach, can partially obscure the features.

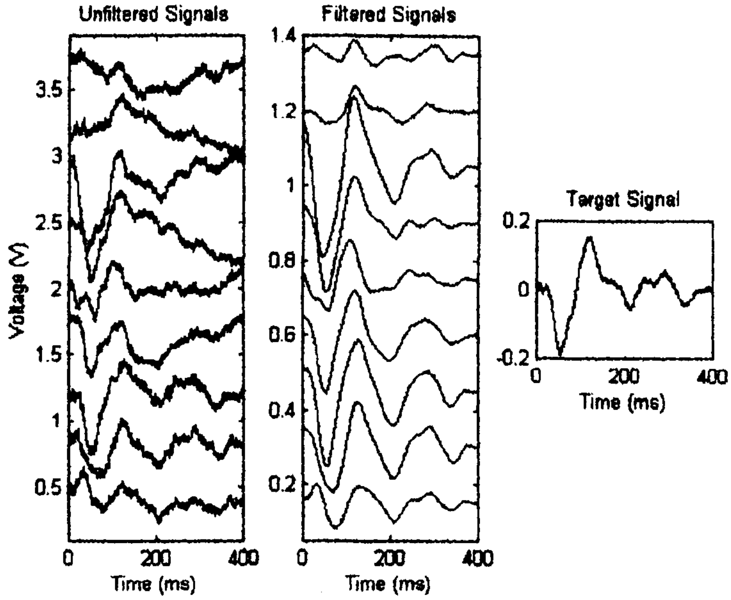

Figure 6-5 Data set 1, whole signal model

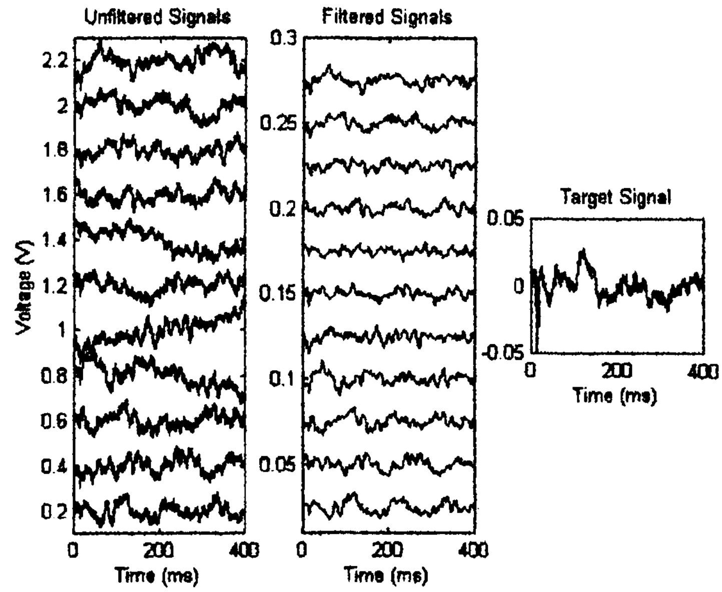

Figure 6-6 data set 2, whole signal model

Data set 2 was also produced from spinal recordings. This time the average signal is noisier than that seen in the previous two data sets. Of interest in this data set, were the large features occurring at the beginning of the signal, as opposed to the later components in data set 1. Filtering the test subset using the filter developed for this data set can be seen to produce noisy results in figure 6-6. Examining the first 30 ms of the signals (figure 6-7), this time show the large dominant features being extracted, but the subtler features are lost.

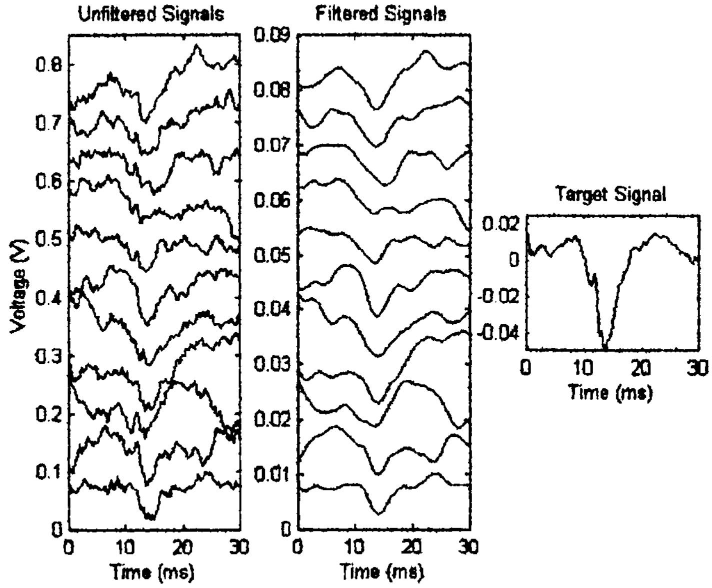

Figure 6-7 data set 2, first 30 ms using the whole signal model

The scalp recorded set of signals, data set 3, shows the filters acting as a relatively low frequency bandpass filter (figure 6-8). The weightings and the frequency parameters for this filter-bank shown in table 6-5 also show this. In this data set, the larger features occur in the later components (figure 6-8). The final data set, data set 4, is another set of scalp recordings. The filtered signals for this data set (figure 6-9) show no similarity to the target signal (figure 6-9).

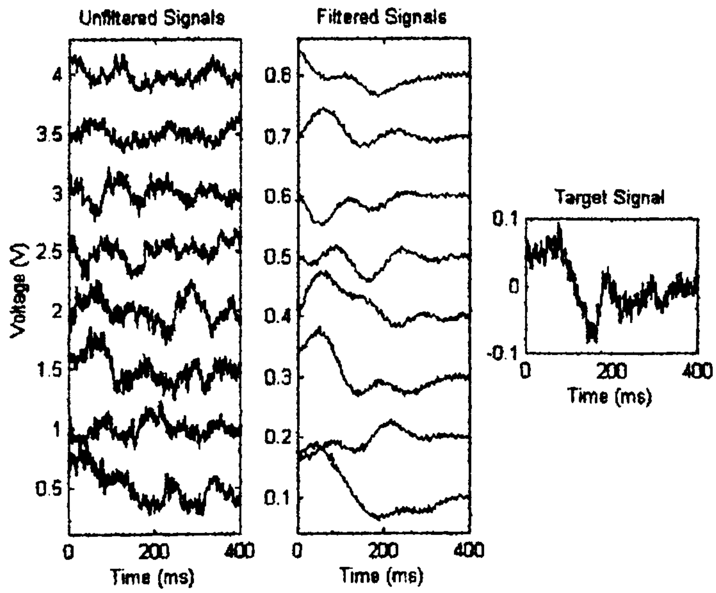

Figure 6-8 data set 3, whole signal model

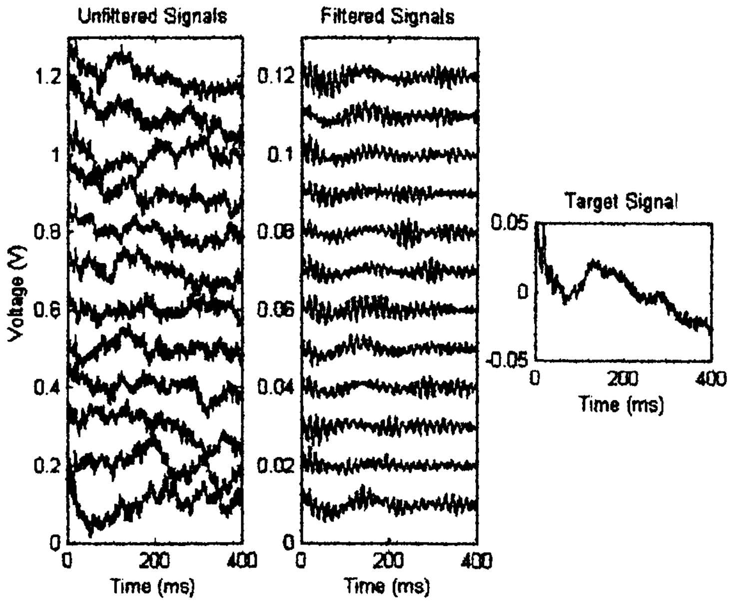

Figure 6-9 Data set 4, whole signal model

6.3 Limiting the signal size

The previous results suggest that the evolutionary algorithm usually extracted the dominant features of a signal, often at the expense of the smaller features. The variation in the position of features between signals is one contributing factor to this effect. A further likely factor, due to the nonstationary nature of these signals, is that the signal has different requirements for a filter bank at different parts of the signal. A particular filter-bank may work well for one region (e.g. first 30 ms of the signal), but may not work as well for another region (e.g. 30-400 ms). These different requirements may be too much for a single filter bank and the evolutionary algorithm may produce a compromise between the requirements of the separate regions. To examine this effect and to fit into the way clinicians use these signals, filter banks were separately developed for two smaller regions of the signals. The first region is the first 30 ms looking at the short latency components. The second region, 30 to 400 ms, is used to investigate mid and late latency components; from now on, for simplicity, these will be termed late components.

6.3.1 Short Latency (first 30 ms)

Many authors (e.g. Rossini et al (1981), Maccabee et al. (1992)) believe that earlier signal components are more stable than the late components. Restricting the signal, by developing a filter bank just for this early region, was investigate as a way of improving the extraction of these earlier components.

Figure 6-10 shows the simulated test subset filtered with a filter-bank developed using the simulated training data in this region. These filtered signals do resemble, in terms visual inspection of key clinical features, the target signal more than the unfiltered signal does. A comparison with the results of figure 6-4 show there is an improvement visually in filtering the first 30 ms, when the algorithm is trained specifically for this region, as compared to using a filter-bank developed to filter the whole signal. This improvement can be seen by comparing the results for the filter bank in Table 6-6, with that in Table 6-7 where there is a lower mean MSE.

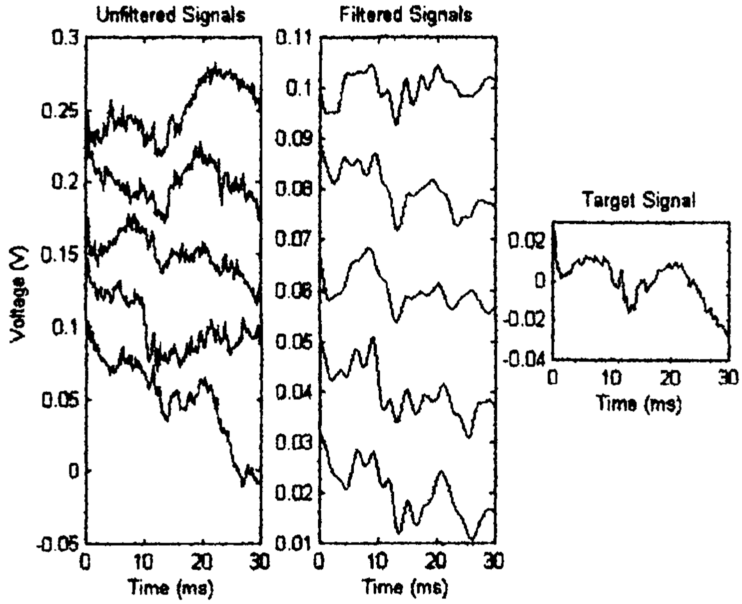

Figure 6-10 simulated data set using the filter bank trained specifically for this region (first 30 ms)

Figure 6-11 shows results from the test set of data set 1 after passing through a set of filters produced after training using only the first 30 ms of signal. The results seemed to have extracted a negative 'trough' around 13-16 ms but the peaks seen at around 12 ms and 16 ms in the target are lost. In the majority of the signals a smoothed outline of the signal is formed. Comparing the filtered and unfiltered signal, in the filtered signals key features are observed, which can not be observed in the unfiltered signals. The MSE values are lower for this filter-bank (Table 6-7) compared to whole signal filter-bank results (Table 6-5).

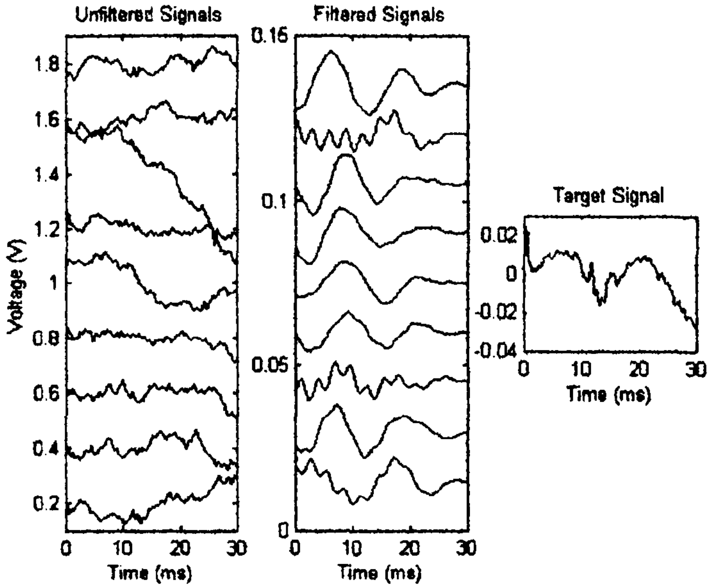

Figure 6-11 data set 1 using the filter bank trained specifically for this region (first 30 ms)

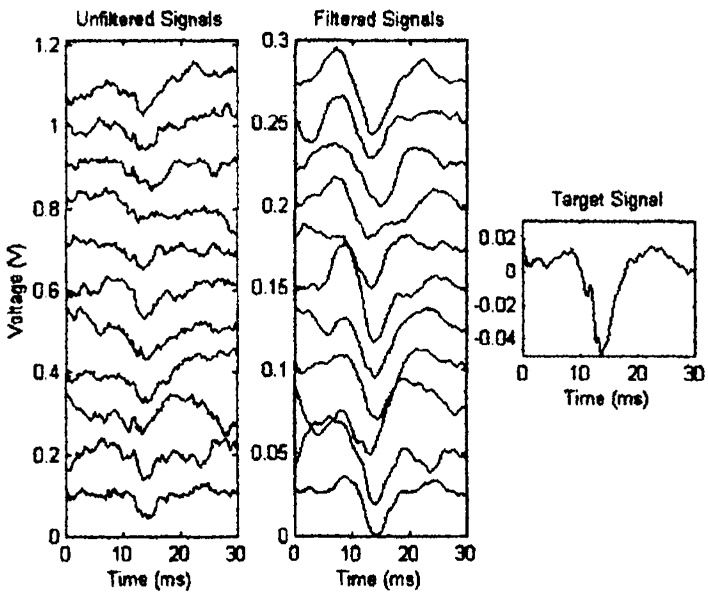

Figure 6-12 data set 2 using the filter bank trained specifically for this region (first 30 ms)

Figure 6-12 shows the results of filtering the test data of data set 2. The filtered signals do bear a resemblance, in terms of key features, to the target signal, more so than the unfiltered signals, with the largest features of the signals being extracted. As in the previous data set, the smaller positive features, in this data set at approximately 12 ms and 18 ms, have been attenuated by these filtering operations. Unlike the previous data set, these signals are similar to those filtered with filter-bank produced using the whole of the signal (figure 6-7) over the same region. The filtered signals are extracting the larger dominant components in the signal, but unable to extract the high rate, smaller features. In this signal, the largest feature (see figure 4-6) of the signal appears in this early region, hence the similarity (mean MSE of 0.0085 and a standard deviation of 0.0038) in the results.

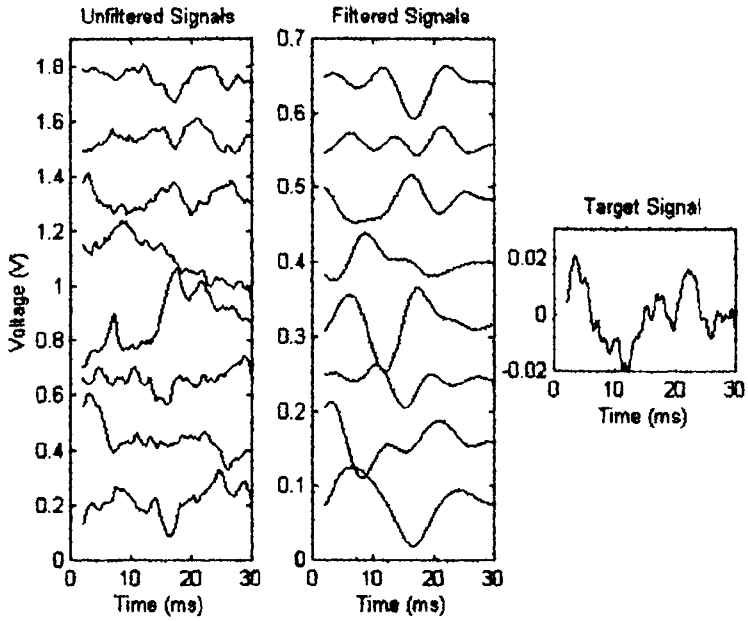

Figure 6-13 data set 3 using the filter bank trained specifically for this region (first 30 ms)

Figure 6-13 shows results from data set 3 (scalp recording) for 2 to 30 ms. The exclusion of the signal components below 2 ms was done to remove a large stimulation artifact. Again the smaller features have been lost in the filtering process. Large components are present but not always in the same place as those in the target signal (figure 6-13). In figure 6-14 data set 4 was filtered extracting the underlying shape of the signal, but in the unfiltered results these features can often be seen.

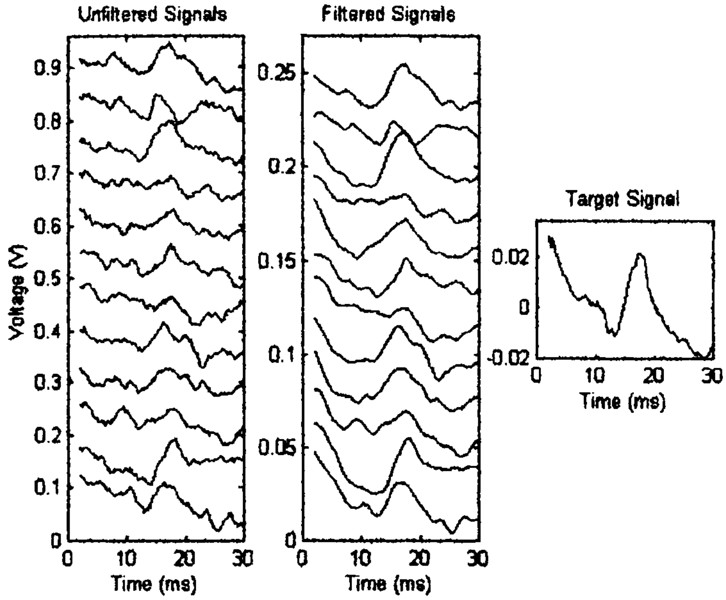

Figure 6-14 data set 4 using the filter bank trained specifically for this region (first 30 ms)

6.3.2 Mid and Long Latency Components (30-400 ms)

The remainder of the signals can also be filtered in a similar way to the early components. Filtering the simulated data set, data set 1 and data set 3 respectively produced similar results (figure 6-15, 6-16 and 6-18) to those observed when filtering the whole signal (Figures 6-4, 6-5 and 6-7 respectively). There was a small increase in the mean MSE values compared to the filter bank developed for the whole signal, for this region (<5% of the value of the equivalent whole signal mean MSE (Table 6-6)). A common feature of these three data sets is they all have relatively large dominant late component features. The rise in MSE is due to the inclusion of noisy components. The results of the filtering data set 2 are improved visually (figure 6-17) by only using 30-400 ms region of the data and looking at Tables 6-6 and 6-8 the MSE shows an increased mean value when using this narrower region. The difference in the mean MSE values for this data set between the two filter banks was negligible. Filtering data set 4 (figure 6-19) with narrower signal length had a similar effect to that seen in figure 6-8 for data set3, a noisy low-pass filter. This result can also be seen in the weightings and frequency parameters for this data set in table 6-8. The result is due to the filtering process not having to try to model the larger components at the beginning of the signals. This time the mean MSE (Table 6-8) is lower using this filter than when the filter developed for the whole signal was used (Table 6-6).

Data Set	No of filters	Frequencies (Hz)		Weights	Best mean MSE 10^{-2}				Variations in mean MSE 10^{-3}			
		F_L	F_H		Min	Max	Mean	Std	Min	Max	mean	std
Simulated	0	-	-	-	0.0050	0.0455	0.025	0.0179	-	-	-	-
	3	0.31	7.00	2.3741	0.0060	0.0105	0.0076	0.0017	0.0760	0.1060	0.0830	0.0130
		66.00	225.20	0.4987								
		310.45	398.69	0.653								
1	0	-	-	-	0.0310	2.1269	0.3659	0.6744	-	-	-	-
	3	52.78	108.87	0.3996	0.00960	0.01277	0.01137	0.00103	0.1137	0.1144	0.1142	0.0003
		335.92	362.00	0.8477								
		142.70	142.77	0.0883								
2	0	-	-	-	0.0059	0.0654	0.0367	0.0212	-	-	-	-
	3	93.80	135.4	0.6496	0.0036	0.016	0.0085	0.0038	0.0850	0.0880	0.0860	0.0010
		29.50	84.40	0.6703								
		69.70	839.30	0.1601								
3	0	-	-	-	0.1104	0.8474	0.3155	0.2758	-	-	-	-
	3	54.45	73.22	0.4837	0.0051	0.0124	0.009	0.0027	0.0900	0.0930	0.0910	0.0010
		98.29	205.44	0.0544								
		164.64	151.10	0.1875								
4	0	-	-	-	0.0036	0.031	0.0159	0.0940	-	-	-	-
	3	39.16	84.16	0.7994	0.0014	0.0129	0.0068	0.0028	0.0680	0.0710	0.0690	0.0010
		0.17	20.20	0.5440								
		106.17	157.38	0.5960								

Table 6-7 partial signal (first 30 ms) model filtering partial signal (first 30 ms)

Data Set	No of filters	Frequencies (Hz)		Weights	Best mean MSE 10^{-2}				Variations in mean MSE 10^{-2}			
		F_L	F_H		Min	Max	Mean	Std	Min	max	Mean	std
Simulated	0	-	-	-	0.094	0.2885	0.1819	0.0874	-	-	-	-
	3	6.33	25.80	0.8251	0.0231	0.0933	0.0608	0.0313	0.0608	0.0624	0.0617	0.0006
		61.65	273.37	0.2765								
		0.77	9.96	0.4792								
1	0	-	-	-	0.3183	2.0462	0.9789	0.5538	-	-	-	-
	3	2.67	11.09	0.3338	0.0544	0.4299	0.1672	0.1222	0.1672	0.1675	0.1674	0.0001
		5.22	20.72	0.4631								
		62.19	206.86	0.064								
2	0	-	-	-	0.0706	0.2599	0.1341	0.0654	-	-	-	-
	3	5.86	15.08	0.1155	0.0063	0.0086	0.0070	0.0008	0.0070	0.0072	0.0071	0.0001
		13.25	44.16	0.0802								
		78.73	1361.50	0.0507								
3	0	-	-	-	0.3664	1.2566	0.7688	0.3005	-	-	-	-
	3	0.97	7.90	0.3125	0.0399	0.1983	0.1003	0.0492	0.1003	0.1151	0.1046	0.0066
		13.61	290.32	0.0596								
		244.24	2089.50	0.0613								
4	0	-	-	-	0.02941	0.1298	0.0564	0.0284	-	-	-	-
	3	0.19	3.43	0.2765	0.0064	0.0287	0.0158	0.0074	0.0158	0.0181	0.0168	0.0009
		131.28	266.14	0.0008								
		2.2772	324.82	0.0944								

Table 6-8 partial signals (30-400 ms) model filtering partial signal (30-400 ms)

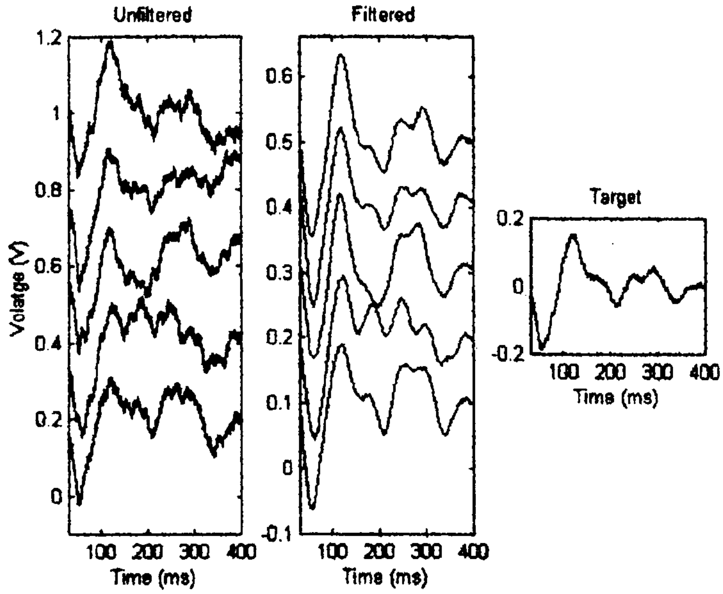

Figure 6-15 simulated data set using the filter bank trained specifically for this region (30-400 ms)

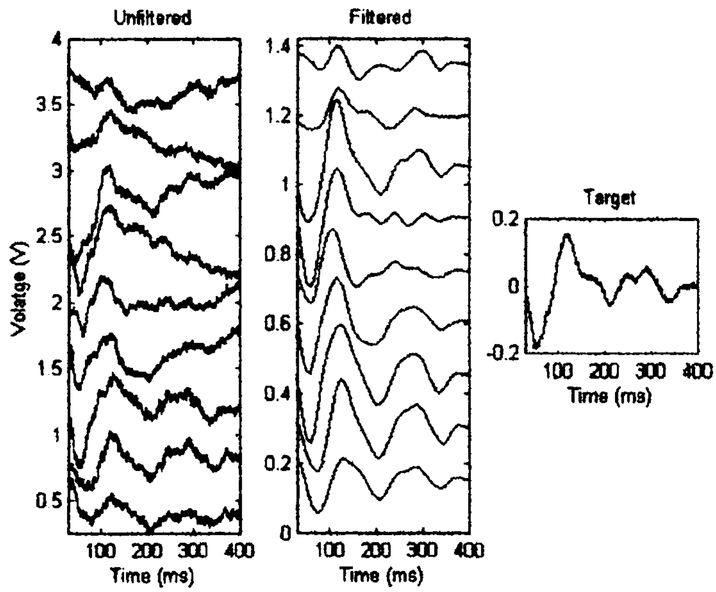

Figure 6-16 data set 1 using the filter bank trained specifically for this region (30-400 ms)

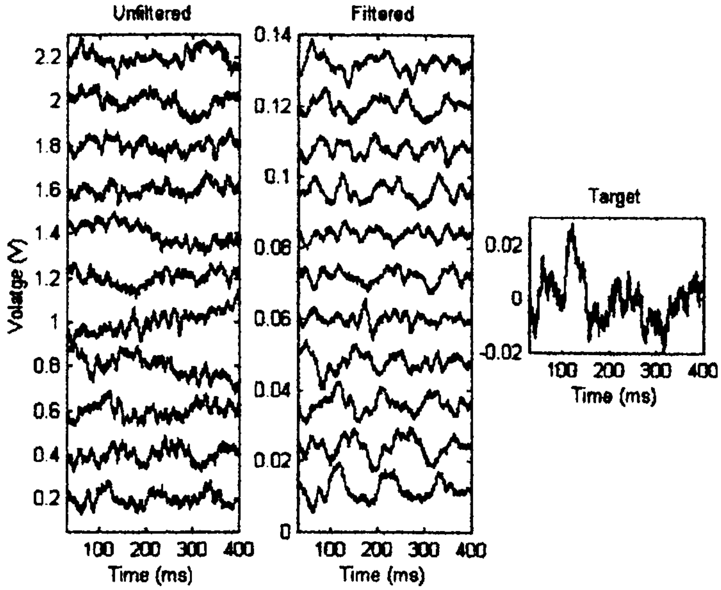

Figure 6-17 data set 2 using the filter bank trained specifically for this region (30-400 ms)

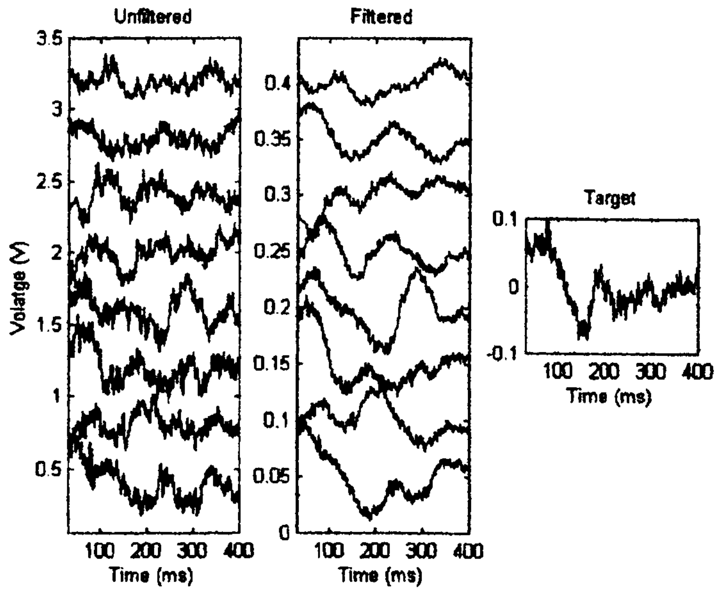

Figure 6-18 data set 3 using the filter bank trained specifically for this region (30-400 ms)

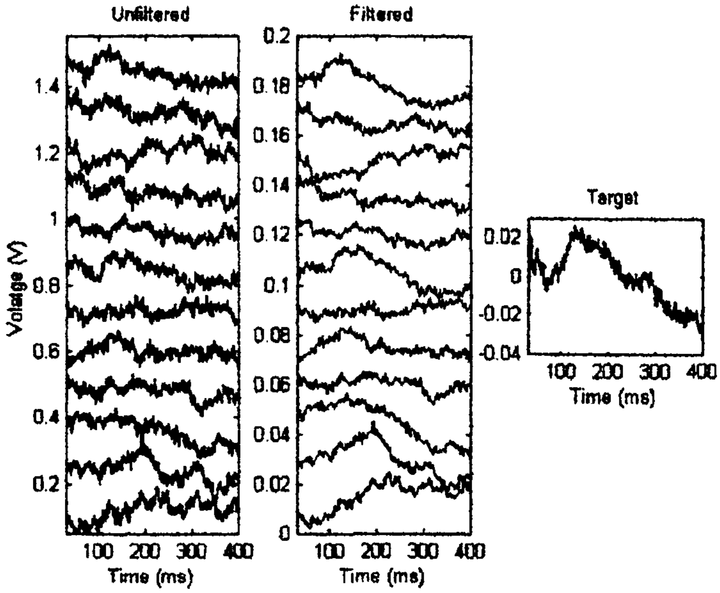

Figure 6-19 data set 4 using the filter bank trained specifically for this region (30-400 ms)

6.4 Comparison of time-invariant methods

In this and the previous chapter time-invariant methods were developed, but now a comparison of their effectiveness is needed. The MSE of each of the filtered signals with the target signal forms the basis of the measurement. The mean of these MSE values is selected as the representative value for the test set for that filtering method. A further consideration for this work is that there should be minimal averaging (ideally none). As the number of signals in each subaverage is varied, a mean MSE value is produced. The goal is to produce filtering techniques with the lowest MSE values, for a small number of signals in each subaverage. As the size of the subaverage increases, the MSE value is expected to decrease as it gets closer to all the signals being averaged and therefore closer to the target signal. Therefore, the effectiveness is shown by how the mean MSE varies with an increase in the number of signals per subaverage, for each filtering method (similar to the method shown in Thakor, 1987). In all the tests, results of averaging-alone are included for comparison as this is the standard technique for processing these types of signals.

6.4.1 Simulated data set

Figure 6-20 shows that the filters developed using subaveraged training data produce better results (lower mean MSE value), for both single filter and bank of three filters. These results were better than the equivalent filters developed using non-averaged training data. All the above filters were developed using the whole signal. In figures

6-21 and 6-22 where smaller regions are considered, a 3 filter model trained on this region (represented by squares) produces an improved response over those trained over the whole signal. In figure 6-21 the single filter models performed as well as the three filter models developed for the whole signal. Below 20 signals per average the filtering methods produce better (lower mean MSE values) results than averaging-alone. Below 35 signals per average the three-filter model produced for this region produces the lowest mean MSE values, after this averaging-alone is the better approach. Figure 6-22 shows the results of applying filtering to the region of the signal occupying the period 30-400 ms. A similar set, in terms of mean MSE values, of responses to figure 6-20 is shown, except there is, in addition, a filter developed using subaveraged training data of the same region. As in figure 6-20, the filtering methods developed using subaverages of 10 signals produced lower mean MSE values than averaging-alone. The filter specifically developed for this period produced lower mean MSE value than the other methods as the size of subaverage increased above 20 signals per average. When the size of subaverages were between 10 and 20 signals per average then mean MSE value were similar to those of the bank of three filtered developed with subaveraged data.

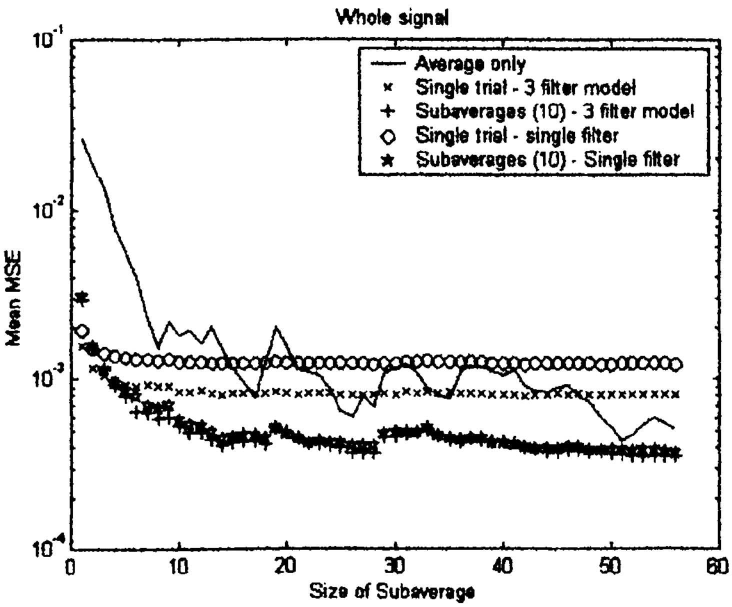

Figure 6-20 Comaprison of the filtering methods for simulated test subset using the whole of the signal.

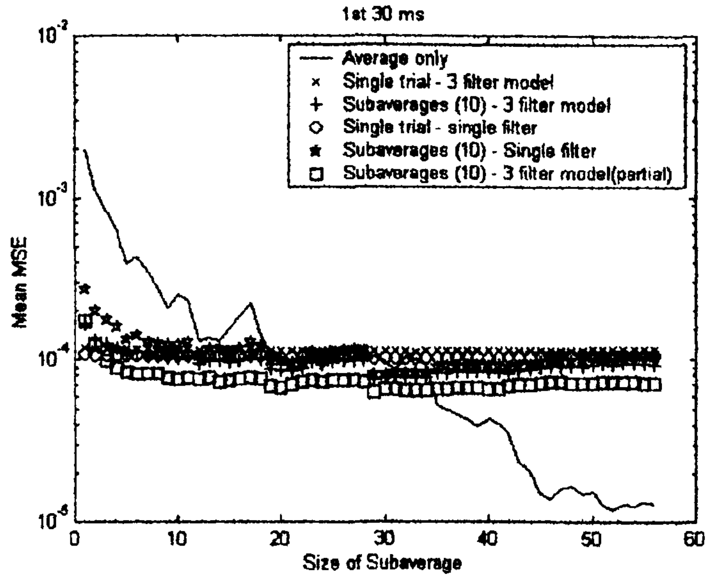

Figure 6-21 Comparison of filtering methods for the first 30 ms of the simulated test subset

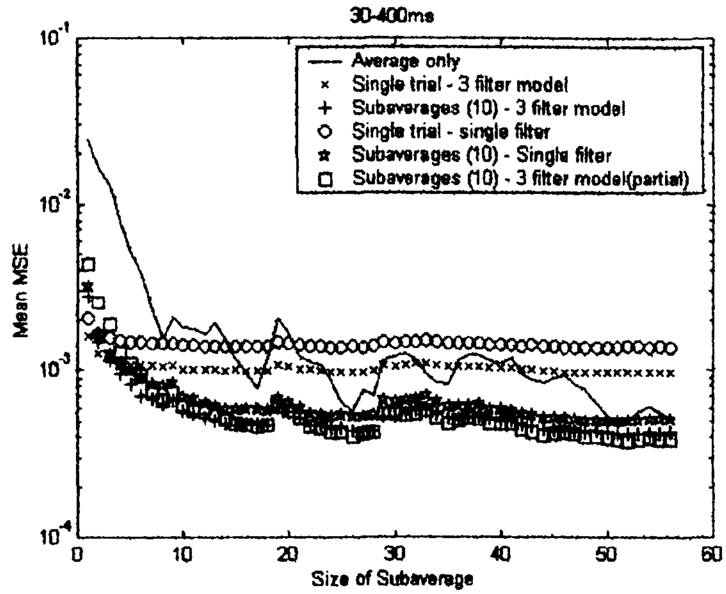

Figure 6-22 Comparison of filtering methods for the region 30-400 ms of the simulated test subset

6.4.2 Data set 1.

Figure 6-23 shows the same effect seen in figure 6-20, with the filters produced from subaveraged data performing better than those filters produced using a single example. There is some improvement with using a three-filter model for both the single and subaveraged data, though in both cases the improvement is relatively small. Figure 6-24 shows the results of filtering the first 30 ms of the signals. In this figure, all the techniques produce better mean MSE values than averaging-alone.

This changes briefly between subaverages of 47 to 65 signals, when the number of signals averaged switches to one. In the region 20 to 60 signals per average, the difference between the experimental techniques is relatively small. Figure 6-25 shows results for filtering the signals between 30 to 400 ms; the results are similar to figure 6-23, with a small improvement for larger sizes of subaverage using the filter bank developed for this region.

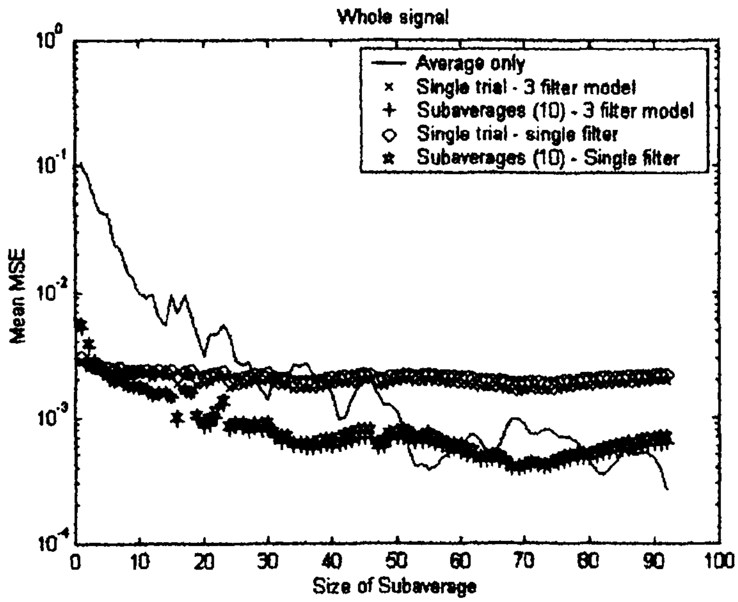

Figure 6-23 comparison of filtering methods for the test subset of data set 1

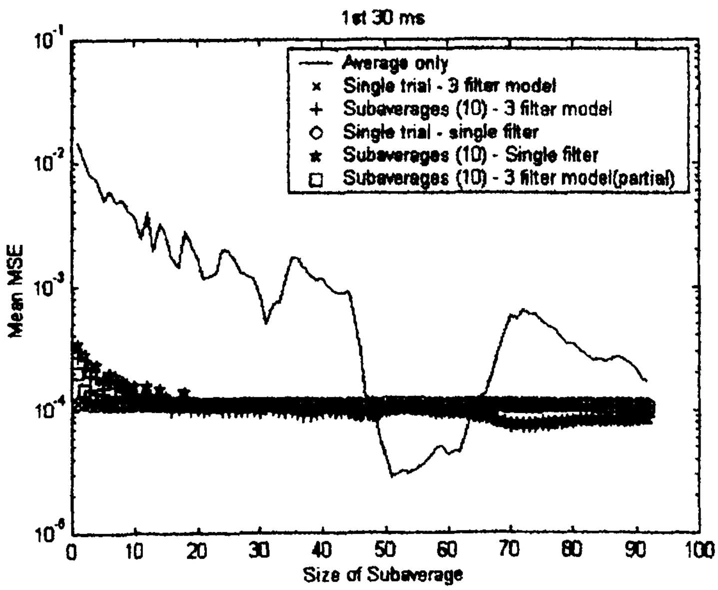

Figure 6-24 comparison of filtering methods for the first 30 ms of the test subset of data set 1

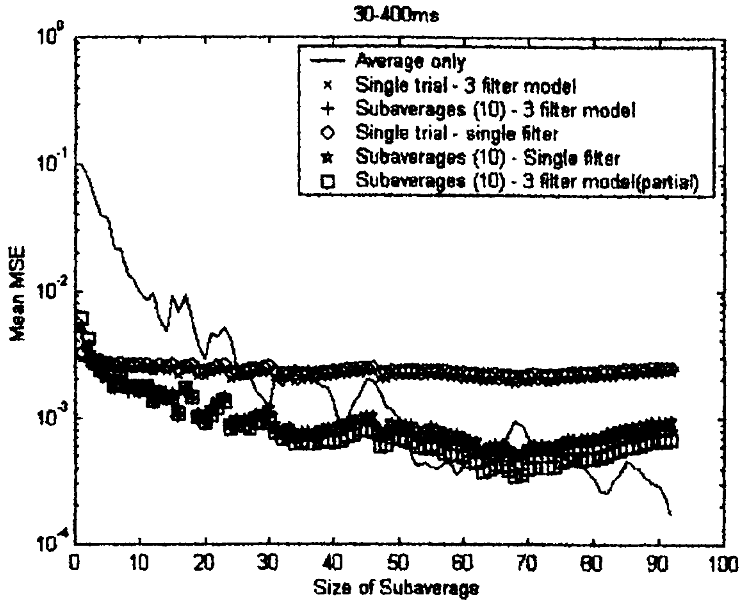

Figure 6-25 comparison of filtering methods for the region 30-400 ms of data set 1 test subset

6.4.3 Data set 2

Figure 6-26 shows that all the techniques produced lower mean MSE values than averaging-alone with smaller sizes of subaverages. Figure 6-27 shows filtering methods applied to the first 30 ms of this test subset, the three filter bank developed on this smaller region perform better than the other methods and better than averaging-alone up to around 40 signals per subaverage. Figure 6-28 shows the filtering methods applied to the region 30 to 400 ms, indicating that the filtering methods all produce better results than averaging-alone, for smaller sizes of subaveraging.

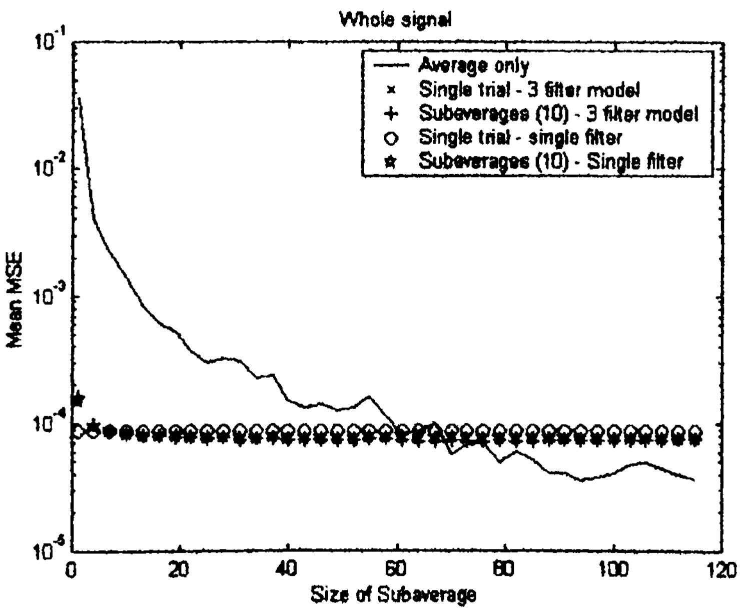

Figure 6-26 comparison of filtering methods for data set 2 test subset

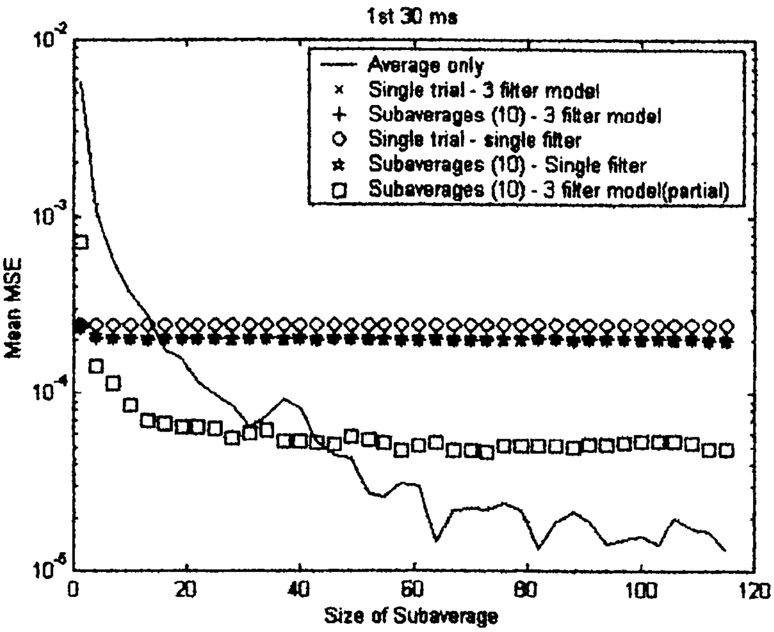

Figure 6-27 comparison of filtering methods for the first 30 ms of data set 2 test subset

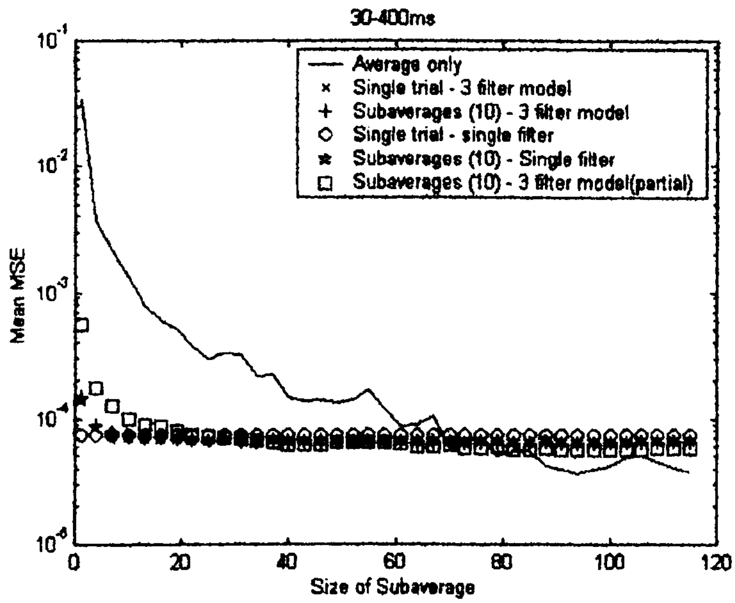

Figure 6-28 Comparison of filtering methods for the region 30-400 ms of data set 2 test subset

6.4.4 Data set 3

Figure 6-29 shows that initially the single trial single filter produced the best results but, when ten signals are used in each average the filter bank developed using subaveraged data performs better than a single filter. Later averaging-alone produced the lowest mean MSE values, with the next best being the model trained on the region of the data. Again, the results for the whole signal show the filters trained with the larger subaverages produced better results as the size of subaverages (number of signals per average) increase. At the smaller subaverages, single trial results work best, the transition occurs at around subaverages of four signals. Figure 6-30 shows results for filtering the smaller region first 30 ms models trained on this region show initially worse results than the other filtering methods, but better results than averaging-alone. A transition occurs at around 10 signals per average; from there up to around 60 signals per average the model produced for this region performed best. Figure 6-31 shows the results of filtering signals in the period 30 to 400 ms. The results are similar to figure 6-29 the filter developed specifically for this region did not, for larger subaverages, perform as well as the filter bank developed over the whole signal.

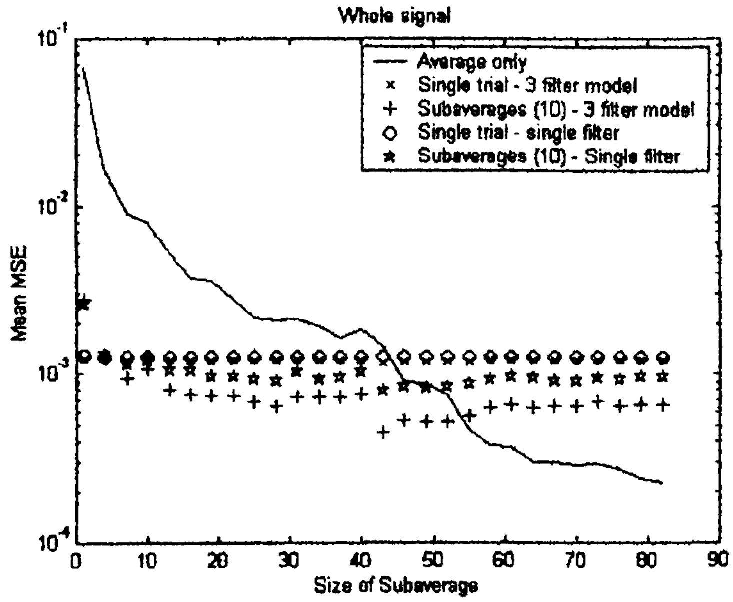

Figure 6-29 comparison of filtering methods for data set 3

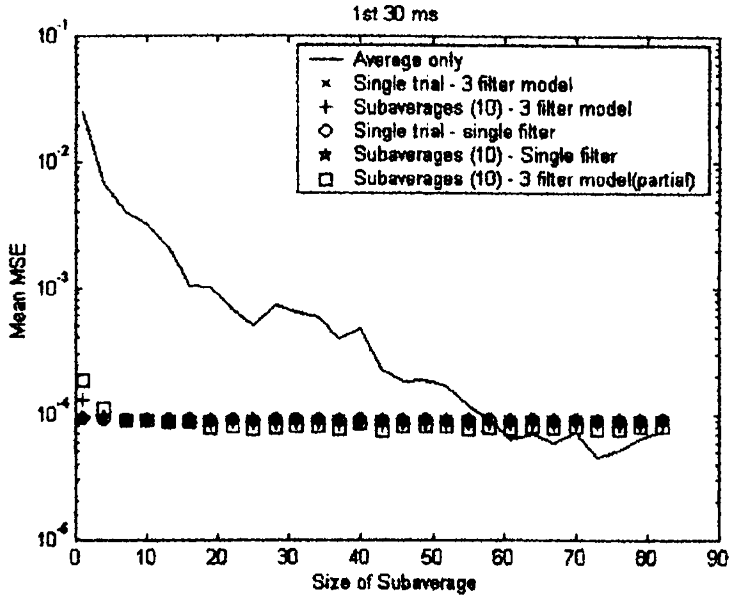

Figure 6-30 comparison of filtering methods for the first 30 ms of data set 3 test subset

.

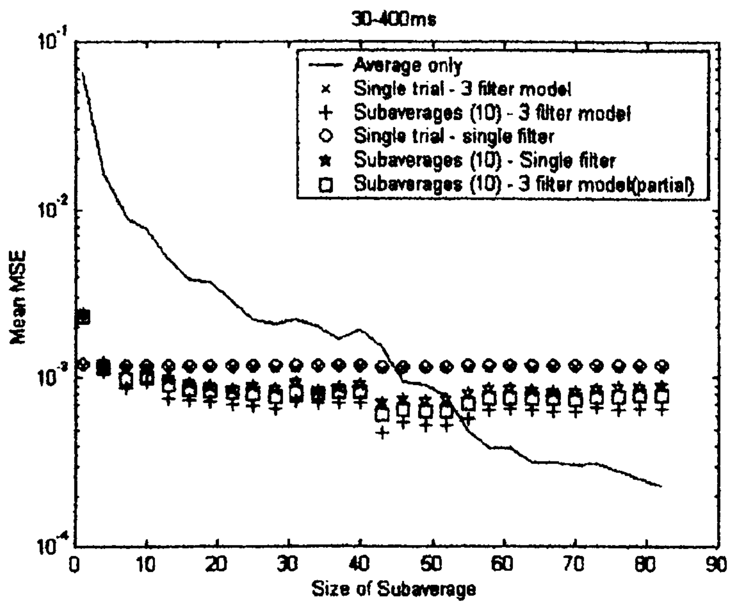

Figure 6-31 comparison of filtering methods for the region 30-400 ms of data set 3 test subset

6.4.5 Data set 4

All the methods (except averaging-alone) applied to the whole of the signal are similar (figure 6-32). Using the shorter region of the first 30 ms (figure 6-33), the 3 filter bank developed using subaverages of 10 signals for this region, performed better than the other methods up to 20 signals per average, then averaging-alone produced the better results. All the filtering methods produced better results (figure

6-34) than averaging-alone below 30 signals per average, with the filters developed for 30-400 ms.

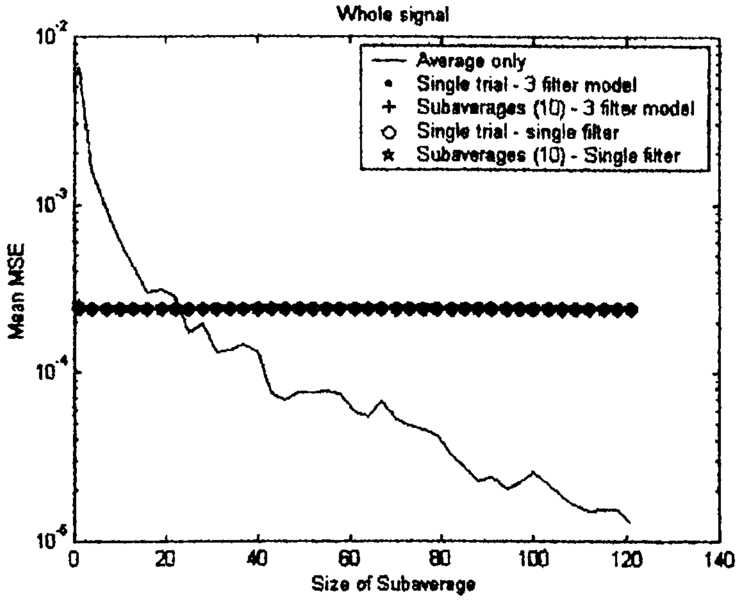

Figure 6-32 comparison of filtering methods for data set 4

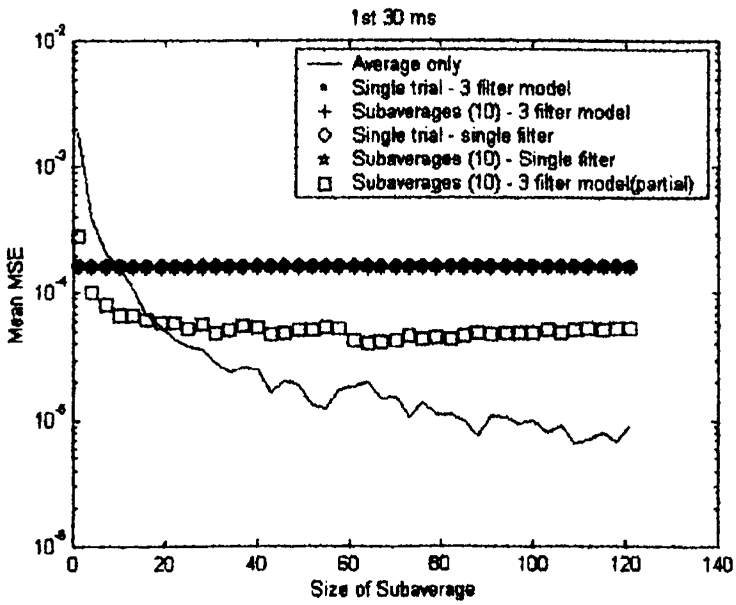

Figure 6-33 comparison of filtering methods for the first 30 ms of data set 4 test subset

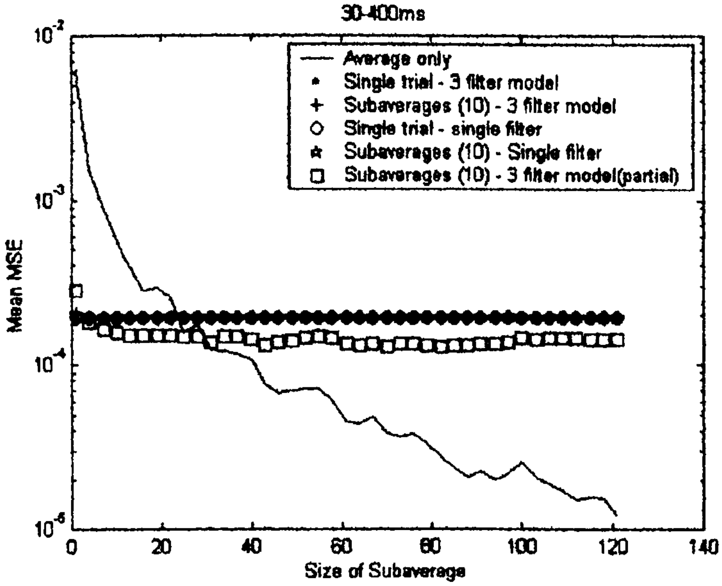

Figure 6-34 comparison of filtering methods for the region 30-400 ms of data set 4 test subset

6.4.6 Overall

In general, filtering methods developed using subaveraged data produced better (lower mean MSE values) results for both single filter and banks of three filters, producing lower mean MSE values than either of the methods developed using non-averaged data (single trial). When the whole signal is being filtered, all the filtering methods for small numbers of signal per average (fewer than 20 signals in a subaverage) produced lower mean MSE values than averaging-alone (see figures 6-20, 6-23, 6-26, 6-29, 6-32).

When the filtering methods for the whole signal are grouped as those developed using data that were not averaged and those developed using data that were formed as subaverages of 10 signals each, the difference between the results of the single filtered and filter bank filtered results was often small. Usually the three filter bank methods in both groups produced the lower mean MSE results.

Overall, filtering developed specifically for the first 30 ms performed better than the filters developed for the whole signal. This effect can be most clearly seen in figure 6-27 and 6-33. The common feature of these two signals is that when the test data are subaveraged (see figures 6-12 and 6-14), some features of the response can be seen in the unfiltered signals. Initially these often had a higher mean MSE for the very small size of subaverages (<5 signals per average), or similar results to the other methods.

The results for the whole signal are similar to those for the region 30-400 ms. The results using the filter developed for the whole signal produced higher mean MSE values (except in data set 3) than those of the filter bank developed specially for this region.

The developed filtering methods responses with increasing the number of signals per average in terms of mean MSE values 'flattens' out earlier than those of averaging-alone. This effect is mostly to do with the relative scale of the averaging-alone results and the filtering methods. The range of values of mean MSE for averaging-alone can be nearly three orders of magnitude. The filtering methods range of mean MSE values vary by up to one order of magnitude. Variations in the relative values of the filtering methods were less obvious than averaging-alone, therefore appeared to have a smoother response.

As the number of signals in an average increased, averaging-alone in most of the recording produced lower MSE values than the filtering methods, but usually for smaller averages, the filtering methods produced the better results. Apart from averaging-alone, three filter methods performed better than the single filter.

6.5 Conclusions

The time-invariant methods developed can extract the largest features in the signals. Smaller features (relative to the largest features in a signal) such as observed in the first 30 ms can be attenuated with averaging and these techniques, the filtering processes may also distort the signal. The differences between a five-filter and three-filter bank were small: the advantage to using a three-filter bank over the five-filter bank was speed, less processing needed and fewer parameters to 'adapt'.

Simulated responses show lower MSE than those of the recorded responses. This is due to the simulated response being time-invariant as they were produced by taking the target response and adding recorded noise. This means that the underlying response does not change. In the recorded data, the underlying response can vary; for example, a peak at 15 ms in one responses could be at 16 ms in another or 14 ms in a third. In averaging the assumption is that the signals do not vary between the responses. In these filtering approaches the assumption is modified to be that the variation between the responses is assumed to be small. Training a filter on the first 30 ms improves the extraction of the earlier components. This is due to the greater stability of these earlier components as compared to the later components of most of the data sets. The results for filtering the signal in the region 30 to 400 ms were often similar to those of filtering the whole signal, though for three of the data sets there was a small rise in mean MSE values for the filter bank developed specially for this region.

If the signals to be filtered have both high signal-to-noise ratio (>1) and stability between the signals, then filter banks developed for a specific signal region (e.g. the first 30 ms of the signal) can produce a larger improvement in mean MSE than the other techniques, for smaller numbers of signals per average than are currently used.

7 Time-varying Filter Banks

In the previous chapter, evolutionary algorithms were used to select the cut-off frequencies and weights for a bank of filters. In this chapter the previous work is extended to produce time-varying filters. Time-varying filters have been applied to evoked potentials before (e.g. deWeerd (1981a, 1981b), Nishida et al., 1993) and were found to perform better than time-invariant filters (Yu et al., 1983). Instead of using a single set of filters applied to the whole signal, in this chapter different filters (or combinations of filters) are applied at different times. Evolutionary algorithms are used to select the active duration of each filter, as well as the frequency parameters and weightings of the filters. Three filtering approaches will be investigated.

7.1 Modification to Nishida's Approach

Nishida et al (1993) used three bandpass filters to extract evoked potentials, with a time-varying function on the output of each filter. This function enabled each filter to be applied to a different segment of the signal. A high frequency bandpass filter, relative to the other two filters, was used for the first portion of the response. A medium frequency bandpass filter was used for the middle portion of response and a bandpass filter of lower frequencies for the remainder of the response (figure 7-1). This has been discussed previously in chapters 2 and 6.

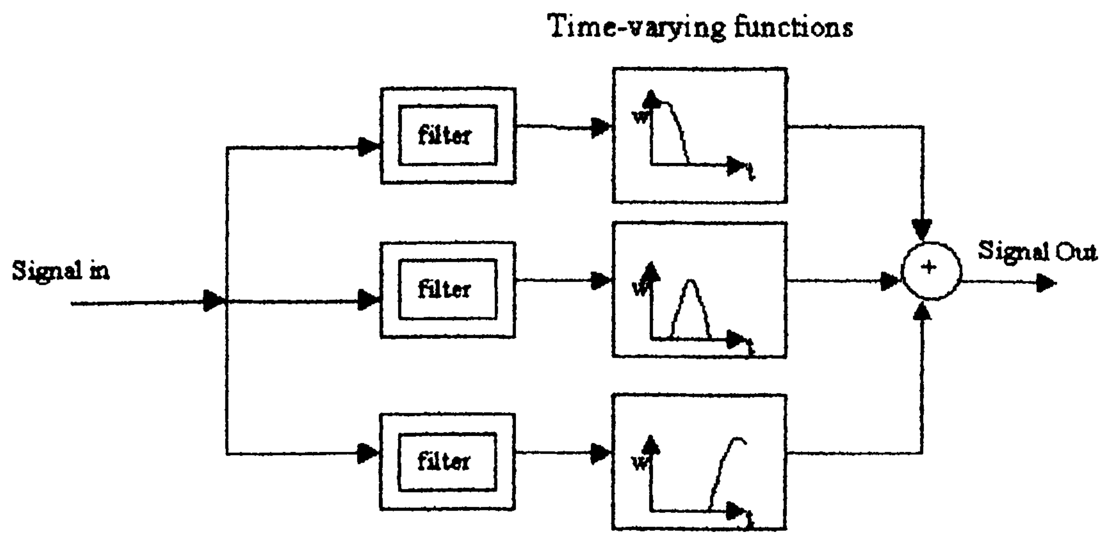

Figure 7-1 Idealised model of the Nishida approach

A modified version of the Nishida method was investigated. Instead of trying to get this information from a power spectrum an evolutionary algorithm selects the four frequencies used in the three filters, as well as the duration of the filters. The active duration of the filters was selected by specifying the ends of the period for the high frequency and medium frequency filters. A 10 ms transition period between the end of one filter and the start of the next filter's period was included as in Nishida et al, to limited abrupt changes between the active region of one filter and another. The fitness function was, as in the time-invariant filter bank, the mean squared error between the test responses and the averaged response, for each 'individual' in the population.

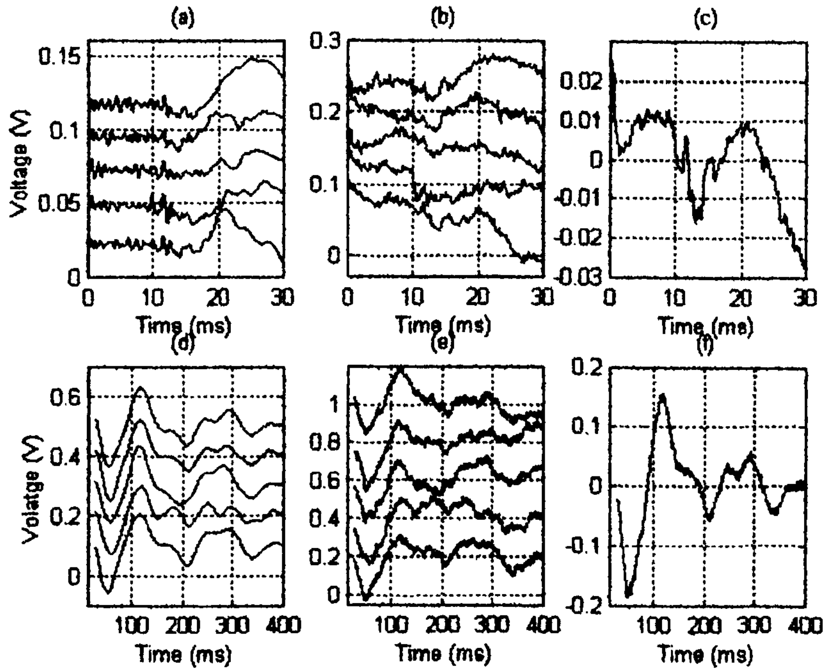

Figure 7-2 Simulated test subset filtered by the modified Nishida approach. (a) The filtered short latency results, (b) the unfiltered short latency test signals (c) the target for the short latency, (d) the filtered late latency signals (e) the unfiltered late latency test signals (f) the target signal for the late latency signals.

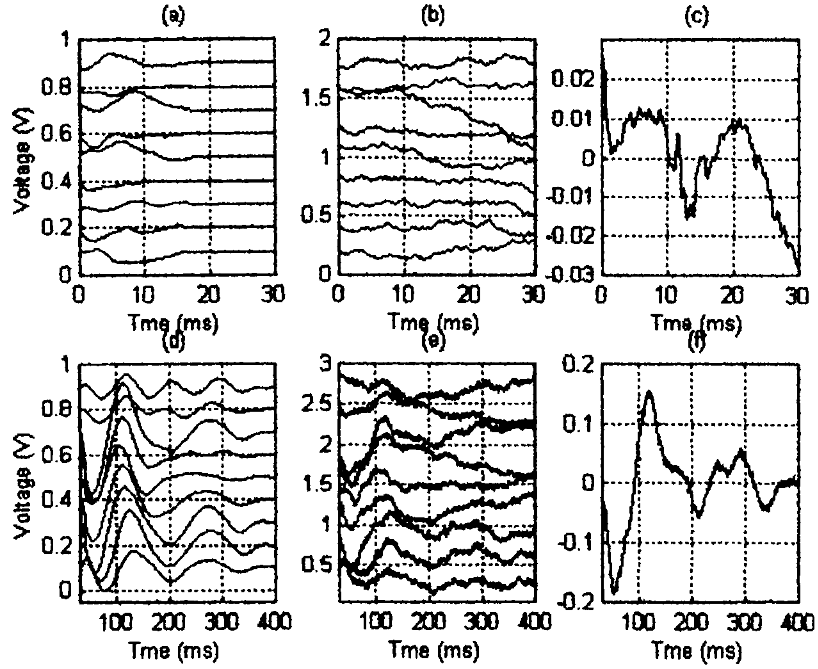

Figure 7-3 Modified Nishida approach applied to the data set 1 test subset. (a) The filtered short latency results, (b) the unfiltered short latency test signals (c) the target for the short latency, (d) the filtered late latency signals (e) the unfiltered late latency test signals (f) the target signal for the late latency signals.

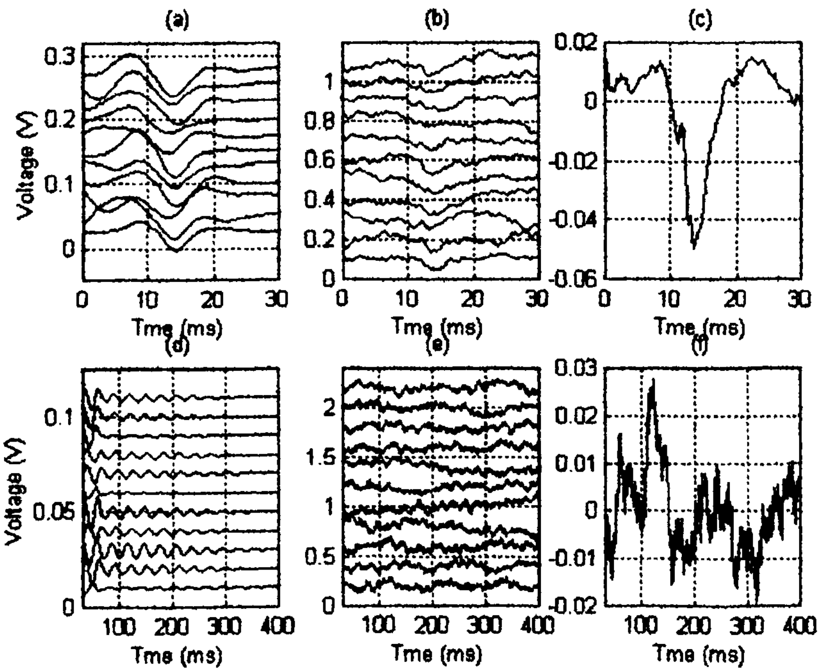

Figure 7-4 Modified Nishida approach applied to the data set 2 test subset. (a) The filtered short latency results, (b) the unfiltered short latency test signals (c) the target for the short latency, (d) the filtered late latency signals (e) the unfiltered late latency test signals (f) the target signal for the late latency signals.

In only figures 7-2 (simulated) and 7-4 (data set 2) has this method extracted some of the peaks of the first 30 ms seen in the target signal. In the others (figures 7-3, 7-5 and 7-6) the extraction of the peaks was not effective enough to see these features. As in the time invariant methods (chapter 6) for the later components of the simulated data (figure 7-2) and data set 1 (figure 7-3) some of the peaks observed in the target signal were extracted. The extraction of the later components in the other data sets were insufficient for clinical usage.

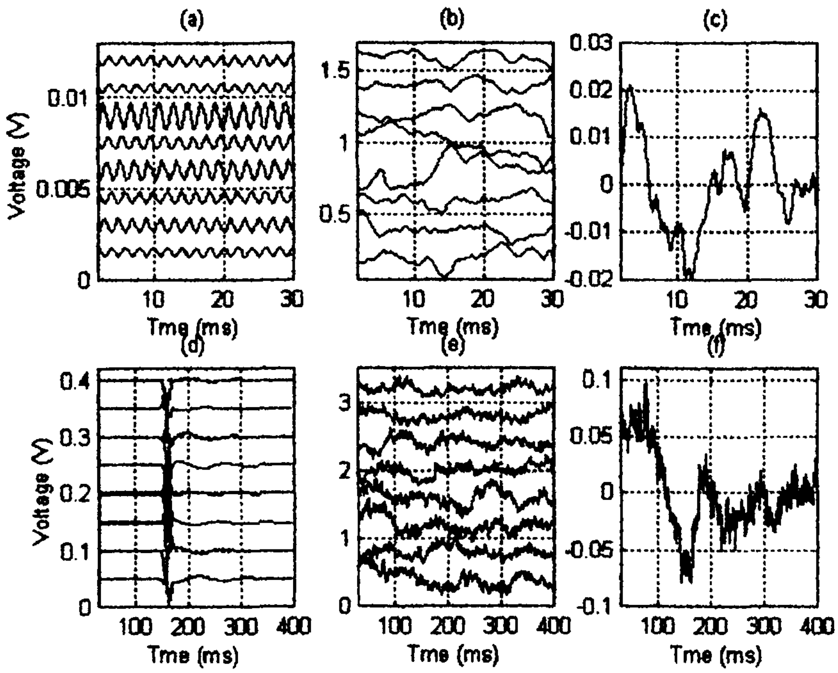

Figure 7-5 Modified Nishida approach applied to the data set 3 test subset. . (a) The filtered short latency results, (b) the unfiltered short latency test signals (c) the target for the short latency, (d) the filtered late latency signals (e) the unfiltered late latency test signals (f) the target signal for the late latency signals.

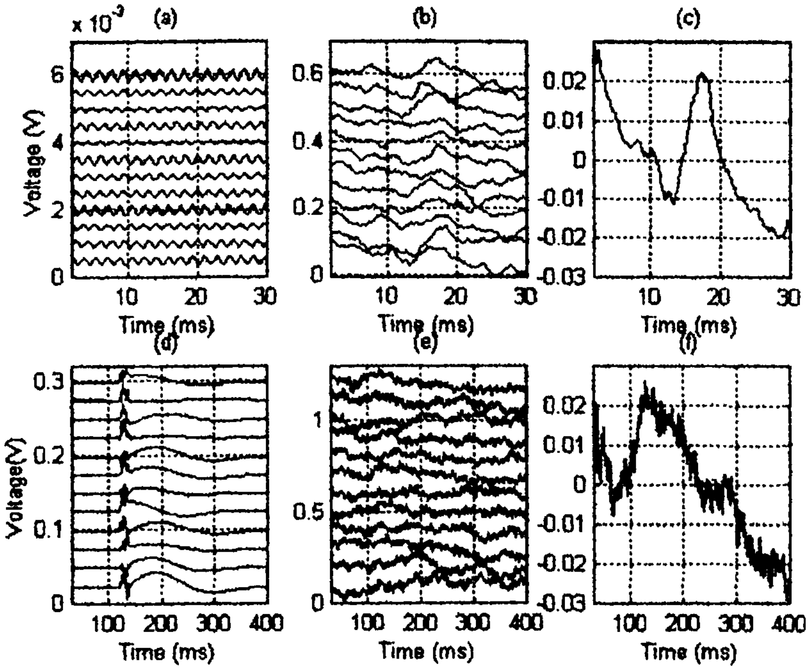

Figure 7-6 Modified Nishida approach applied to the data set 4 test subset. (a) The filtered short latency results, (b) the unfiltered short latency test signals (c) the target for the short latency, (d) the filtered late latency signals (e) the unfiltered late latency test signals (f) the target signal for the late latency signals.

Data Set	Lowest				Range of Mean Values			
	Mean Squared Error (10^{-3})				Mean Squared Error (10^{-3})			
	Min	Max	Mean	Std	Min	Max	Mean	Std
Simulated	0.3977	0.7644	0.5474	0.1777	0.5474	0.6885	0.5764	0.0627
1	1.0891	3.1925	1.9835	0.7987	1.9835	2.0782	2.0404	0.0361
2	0.0685	0.0821	0.0761	0.0043	0.0761	0.1033	0.0852	0.0113
3	1.1979	1.3257	1.2565	0.0564	1.2565	1.3307	1.2938	0.0341
4	0.2288	0.2485	0.2338	0.0058	0.2077	0.243	0.2322	0.0144

Table 7-1 MSE values for the modified Nishida approach

Overall, this method produced MSE values that were unacceptable and did not preserve enough key features in a visual inspection and was therefore deemed to be insufficient for clinical usage.

7.2 Extending the number of filters

The previous approach assumed that the regions were adjacent to each other in the frequency spectrum, giving low, middle and high frequency regions, with no overlap between the regions. This was then modified in three ways. First, though the signals were split into three regions, each region was a weighted combination of the output of two filters, enabling more than one set of frequency components in the same portion of signal. Second, the filters selected were not directly dependent on the filters in the other regions. The passband frequencies of each filter did not necessarily have any frequencies in common with the other two regions. Thirdly, overlapping pass bands of the filters within a region were possible.

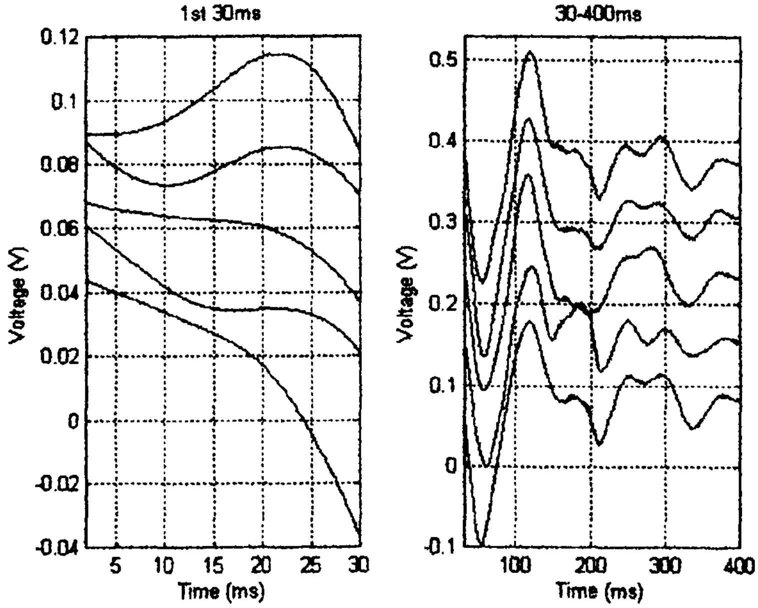

Figure 7-7 filtering using multiple filters for the test subset of simulated data set 1

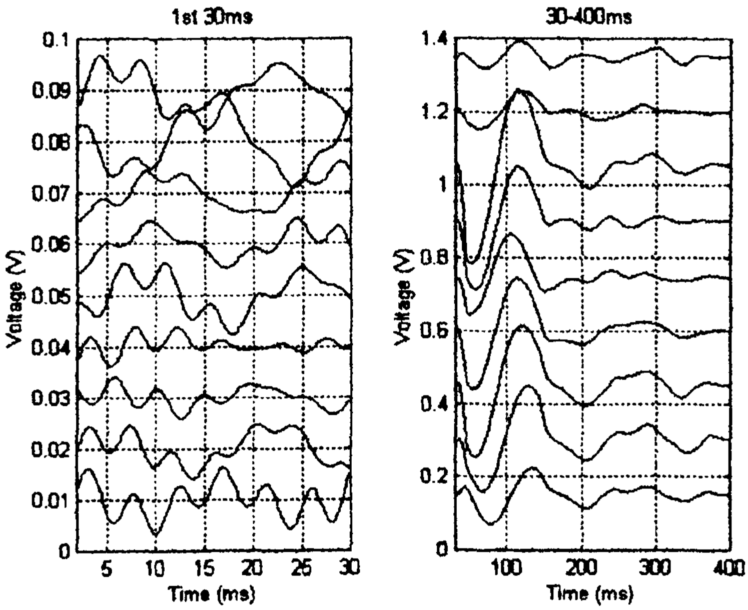

Figure 7-8 filtering using multiple filters for the test subset of data set 1

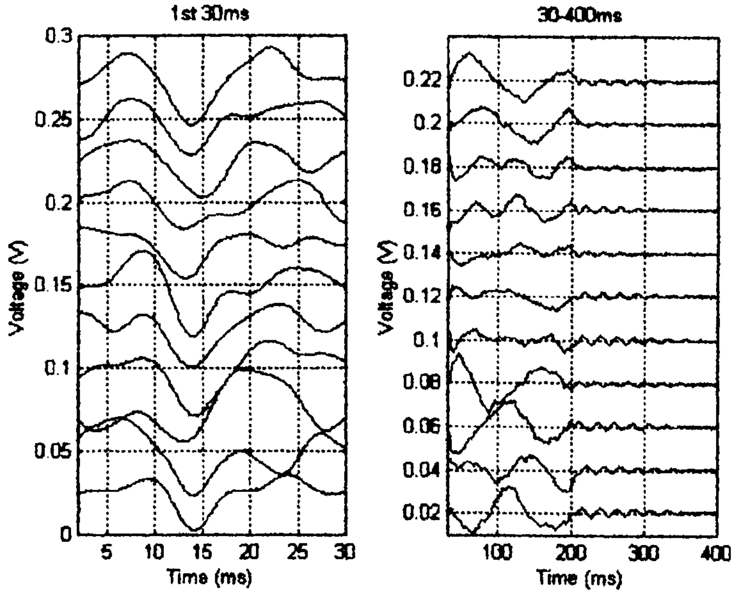

Figure 7-9 filtering using multiple filters for the test subset of data set 2

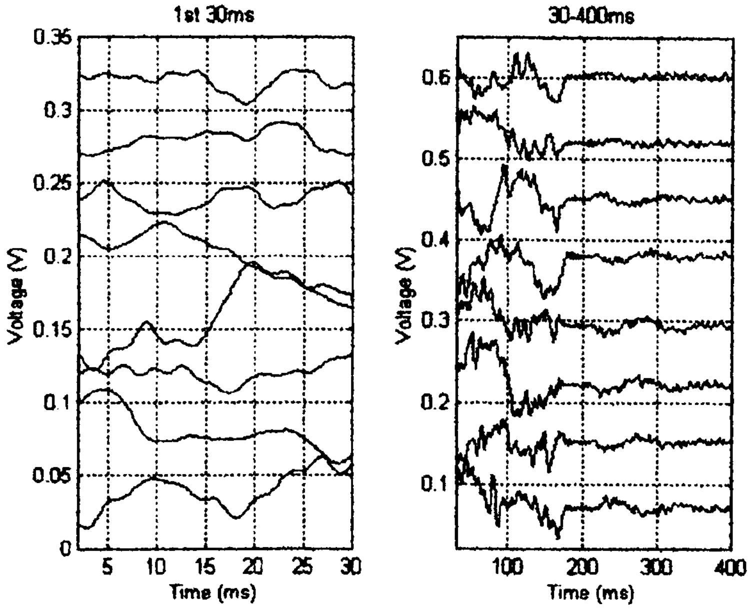

Figure 7-10 filtering using multiple filters for the test subset of data set 3

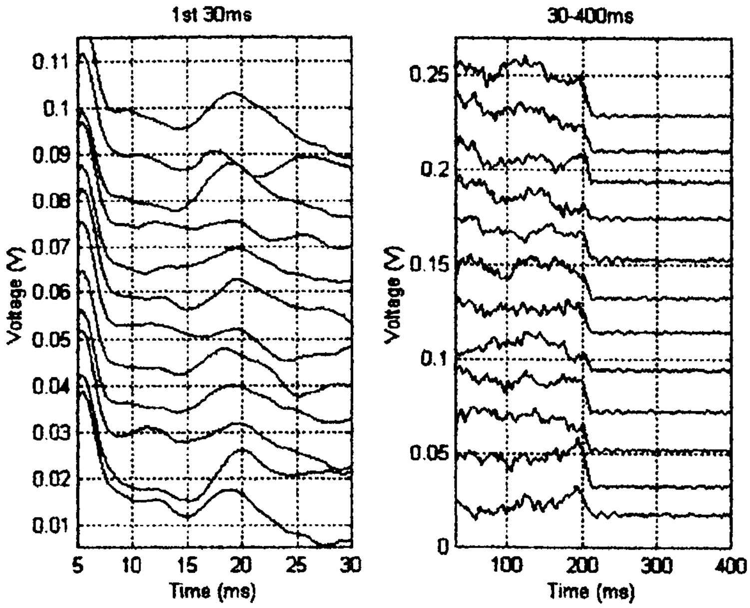

Figure 7-11 filtering using multiple filters for the test subset of data set 4

The results of the first 30 ms for the simulated data set (figure 7-7), data set 1 (figure 7-8) and data set 3 (figure 7-10) are shown above. It can be seen that the technique was unable to extract features from the noisy signal (data set 3). However, this technique was able to extract certain key features of the first 30 ms of data set 2 (e.g. a negative feature at 14 ms in figure 7-9) and data set 4 (e.g. a positive feature at approximately 20 ms in figure 7-11). The unfiltered signals for the first 30 ms of these signals show some of the features that are to be extracted, in the other signals the features are less visible. The results for the first 30 ms of data set 4 also included a large artifact at the beginning of the signal. It is unclear what the exact cause of the artifact is, but it is most likely to be a stimulus artifact. Late components are extracted in the simulated data set (figure 7-7) and data set 1 (figure 7-8), whereas in the other data sets the effectiveness is less clear.

7.3 Splitting the signals into early and late components

In the previous chapter, splitting the signal into two separate signals improved the effectiveness of the filters for the early components. This work and the work carried out by other groups suggest that there are at least two different regions. Early components are more stable than the later components and have higher frequency components (Maccabee et. al., 1983). The late component region is likely to vary between signals and has predominantly low frequency components. Splitting the signal into two signals based on these regions (the first 30 ms and 30-400 ms) means that the two regions are filtered independently of each other using the method described previously (section 7.2). In other words the method described previously is applied to the first 30 ms and in parallel applied to the second signal (30-400 ms), treating each as two independent signals. The fitness value was changed to the mean value of the MSE values of the two separate regions. In tables 7-2 and 7-3 values shown are for the lowest scoring combined MSE values, not necessarily the lowest scoring first 30 ms or 30-400 ms results.

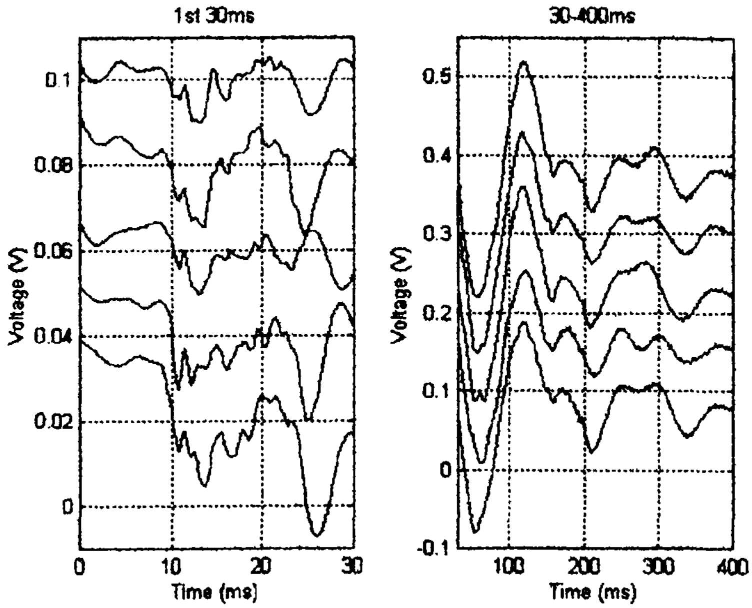

Figure 7-12 The filtered results when splitting the simulated signals

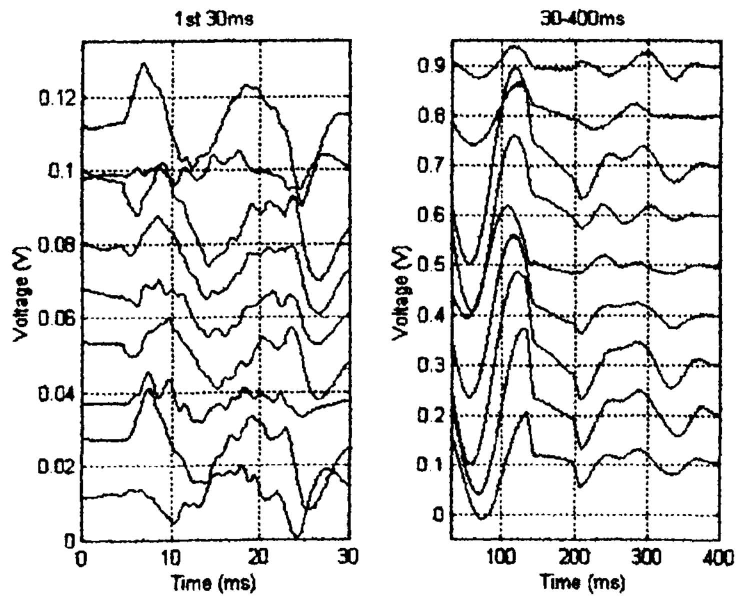

Figure 7-13 The filtered results when splitting data set 1 into two parts

In both the simulated data set (figure 7-12) and data set 1 (figure 7-13), features were extracted for the late components. In figure 7-12, the first 30 ms of simulated data has extracted a combination of large and small components, in figure 7-13 in six out of the nine signals for the early components the larger components are extracted. For data set 2 (figure 7-14) the dominant early component is extracted, but there appears to be an artifact around 100 ms probably due to the filtering process.

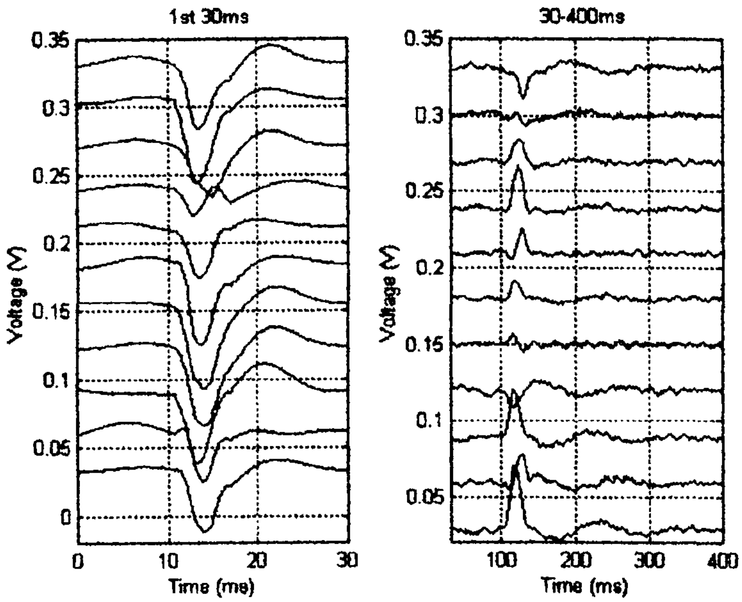

Figure 7-14 The filtered results when splitting the signals in data set 2 into two parts

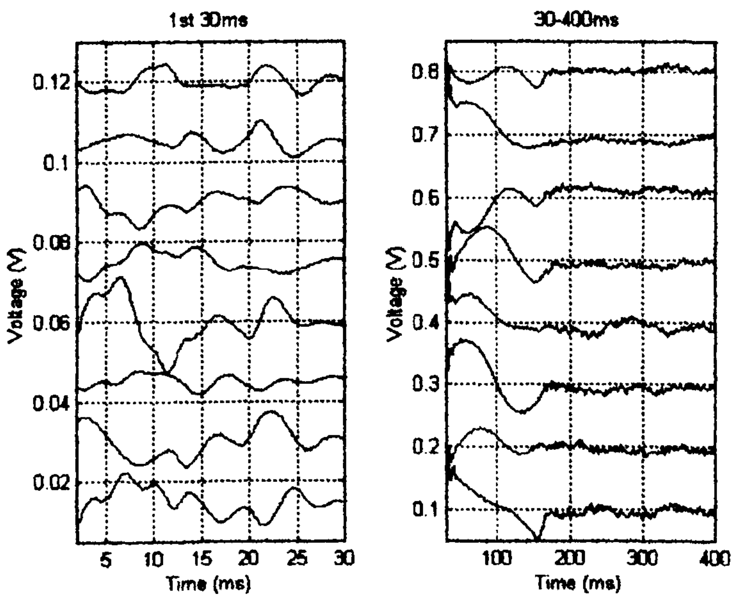

Figure 7-15 Splitting the signals in data set 3 into two parts

For data set 3 (figure 7-15) it is difficult to see any clear features in either of the two regions. The results of data set 4 (figure 7-16) show that the dominant early components are extracted, but the late components were not.

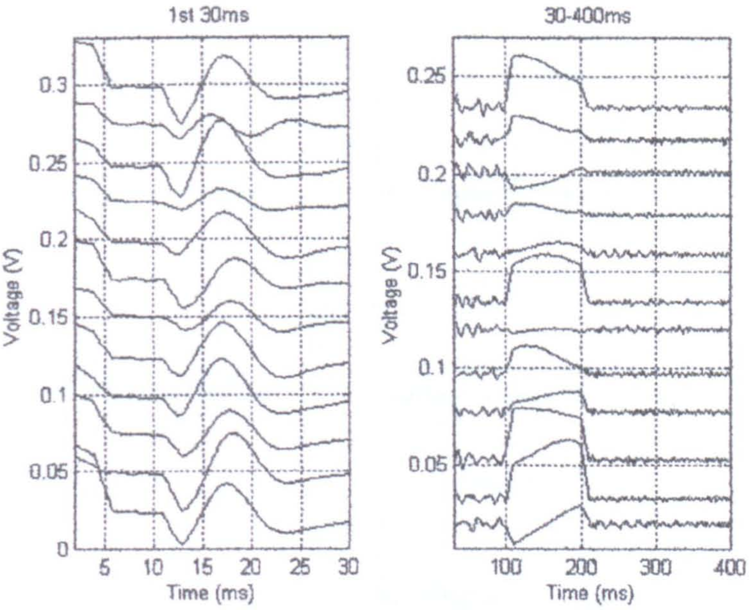

Figure 7-16 The filtered results when splitting the signals of data set 4 into two parts

This technique improved the extraction of early components. In some of the data sets, late components have also been extracted.

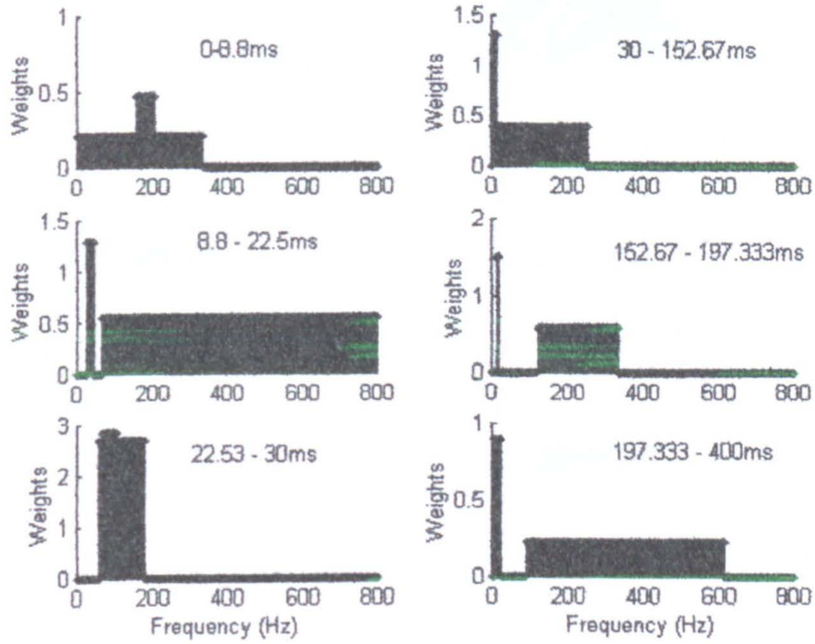

Figure 7-17 Filters selected for the simulated test set

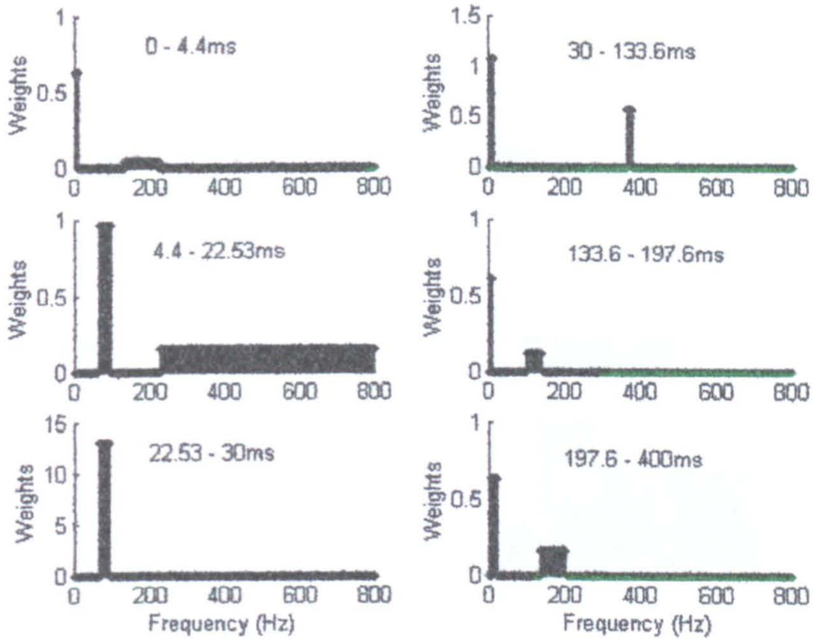

Figure 7-18 Filters selected for the test subset of data set 1

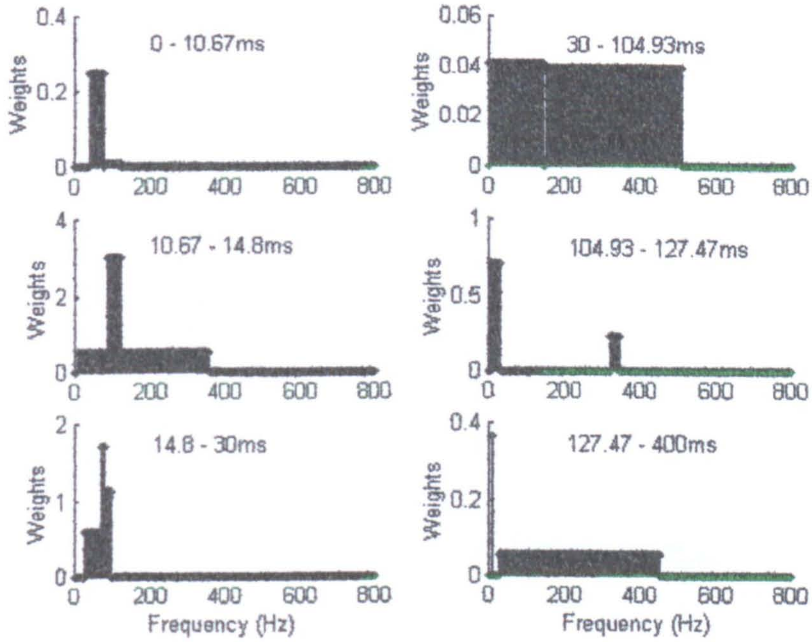

Figure 7-19 Filters selected for the test subset of data set 2

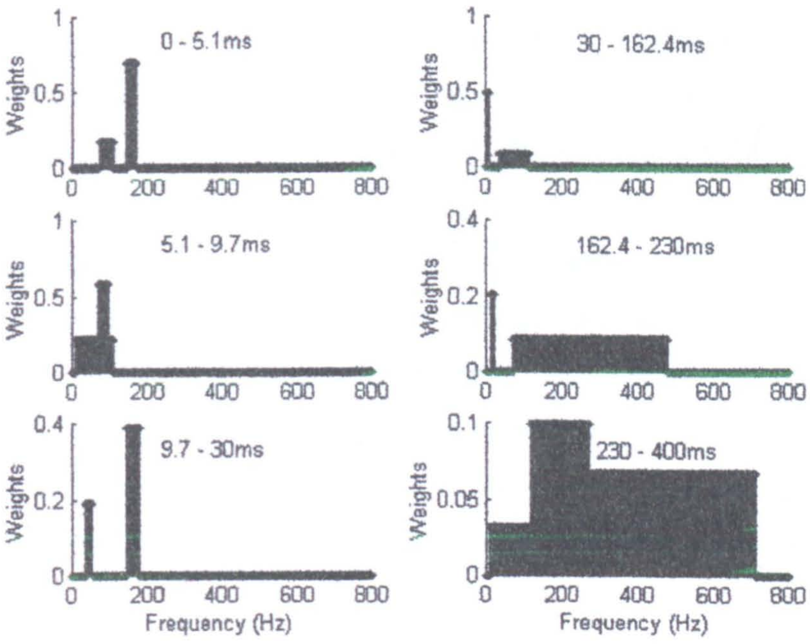

Figure 7-20 Filters selected for the test subset of data set 3

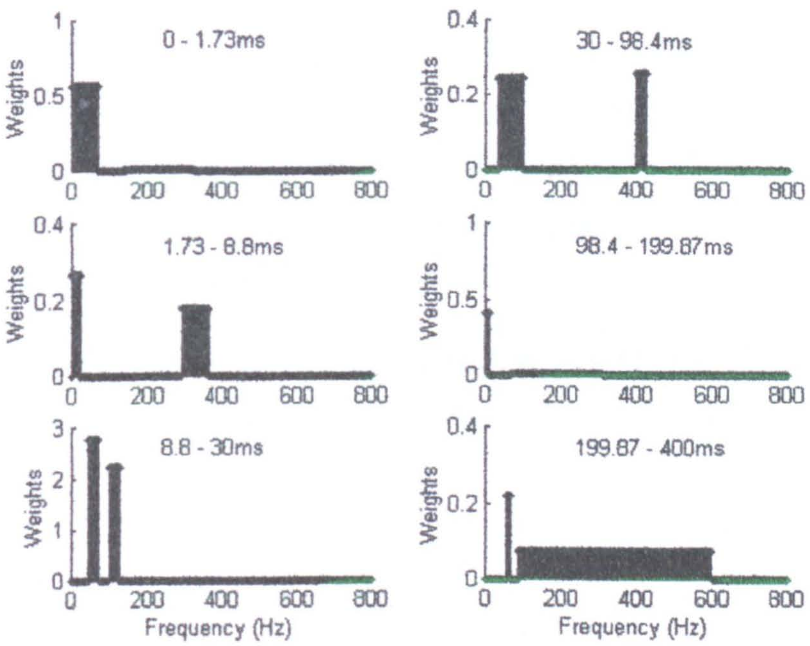

Figure 7-21 Filters selected for the test subset of data set 4

For the early latency components, the bandpass frequencies with the largest weighting were generally at higher frequencies than those selected for the late latency components. These results fit with some of the previous work on linear filters that discussed the early components having higher frequency components than the later components. The late components of both data set 1(figure 7-18) and the simulated data set results (figure 7-17) are dominated by filter with a narrow range of relatively low frequency filters between 4 and 27 Hz. For the early components of data set 2 (figure 7-19) in general, the frequency components are below 200 Hz. The exception to this is a short region of time 10.67 to 14.8 ms where a wider filter with lower weighting extending up to 400 ms. One possible explanation of this is that the filter is attempting to extract the small feature visible in this region on the target signal (figure 7-4c). Data set 3 (figure 7-20) used a relatively narrow range of low frequencies, up to around 200 Hz for early components. For data set 4 (figure 7-21) for early components the filters are narrow band filters with frequency below 200 Hz. In the region 1.73-8.8 ms, the second filter has a pass band approximately between 300-400 Hz. In this region of the target signal (figure 7-6c) small features are present. One possible explanation is that this filter is attempting to extract those features.

7.4 Comparison of methods

In chapter 6 a comparison of time-invariant methods showed that the results (lower mean MSE values) were improved when the signals were split into a short signal (the first 30 ms of the signal) and a longer signal (30-400 ms of the signal). Figures 7-22 to 7-26 show a similar comparison of the three methods used in this chapter, comparing the mean MSE of the filtered signals from the target signals, as the number of signals per subaverage increase.

In all the figures, the best (i.e. the lowest) mean MSE values were those of the third approach, of splitting the signals for smaller sizes of subaverages.

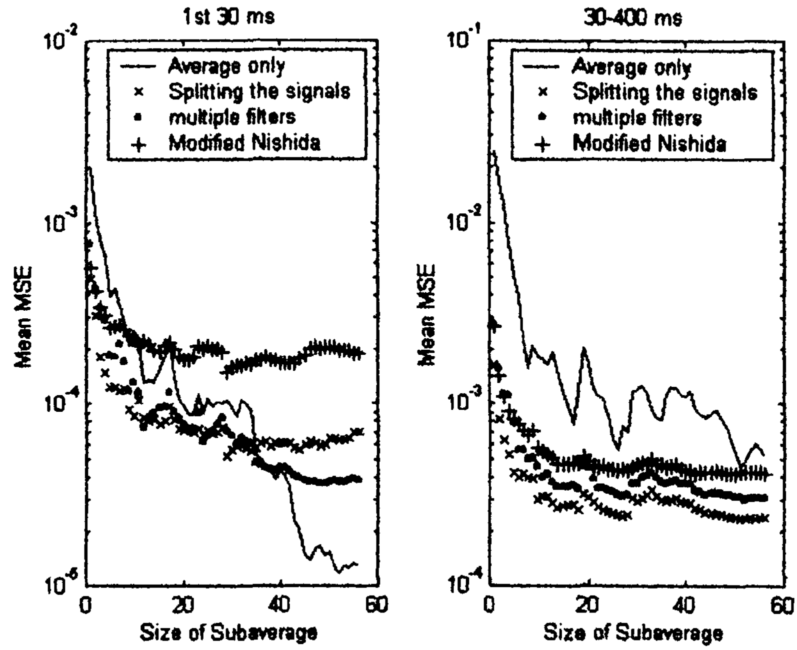

Figure 7-22 Comparison of time-varying methods for simulated test data

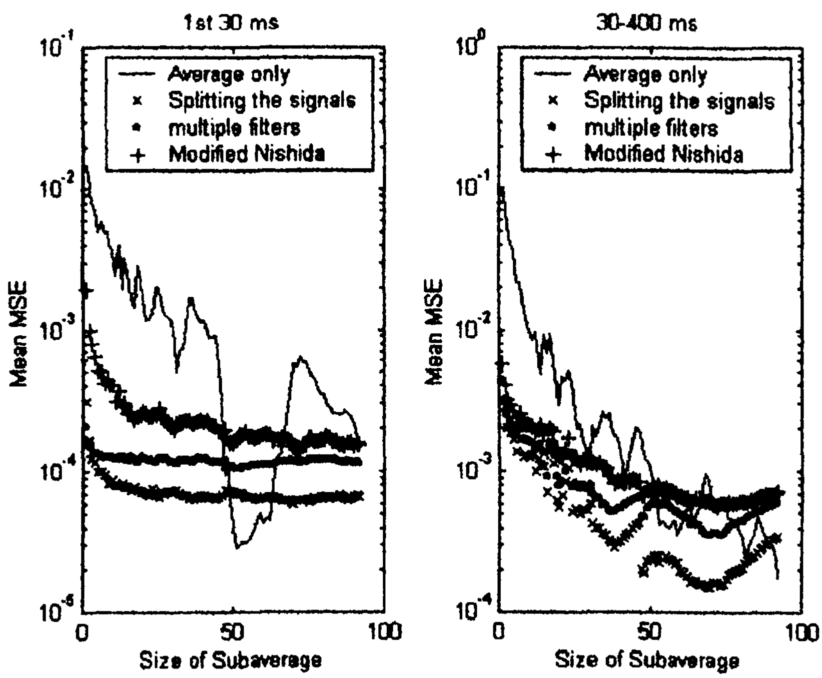

Figure 7-23 Comparisons of time-varying methods for data set 1

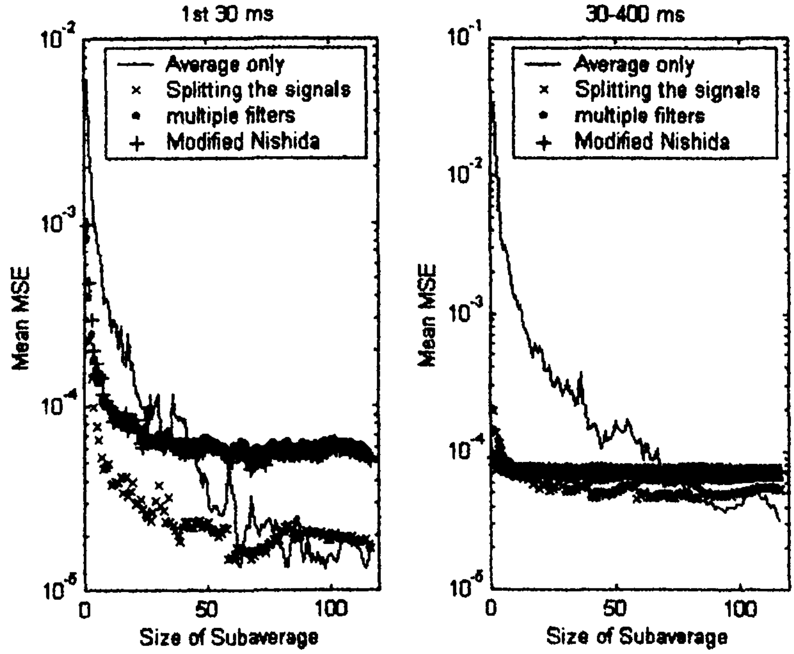

Figure 7-24 Comparisons of time-varying methods for data set 2

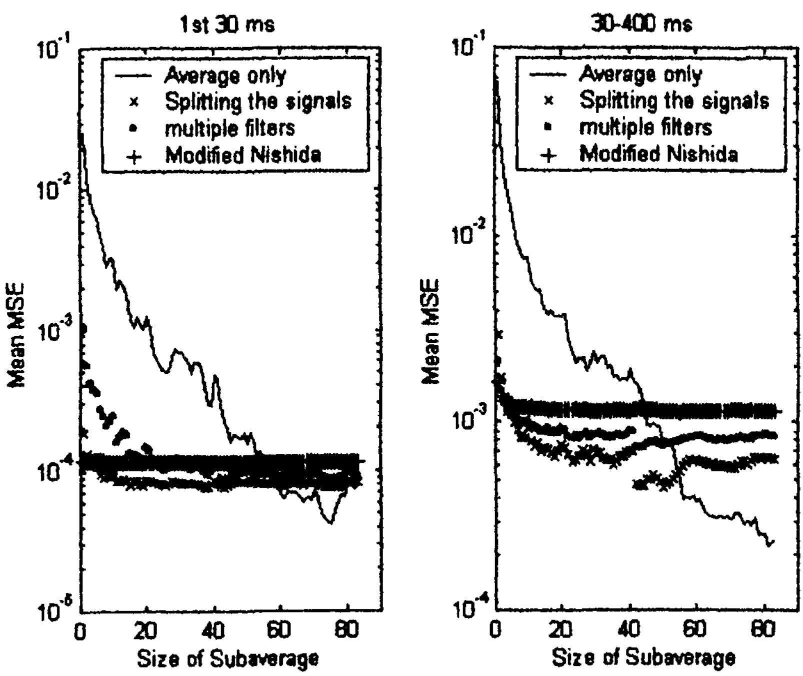

Figure 7-25 Comparisons of time-varying methods for data set 3

The first 30 ms of the simulated data set (figure 7-22) produced an unusual result as compared to all the others (figures 7-23 to 7-26) in that the multiple filter results produce the lower MSE values for larger sizes of subaveraging (20 to 40 signal per average). Compared with the other data sets where the approach of splitting the signals produced the lower MSE values, this effect is not significant, as the aim of this work is to use a small number of signals in a subaverage to enhance the signals.

In figure 7-26 the difference between three approaches for the region of 30-400 ms was small. The effects shown in figure 7-26 were also shown in table 7-3.

The modified Nishida approach in general has been the least effective for all the signals, except for the first 30 ms of data set 4 (figure 7-26).

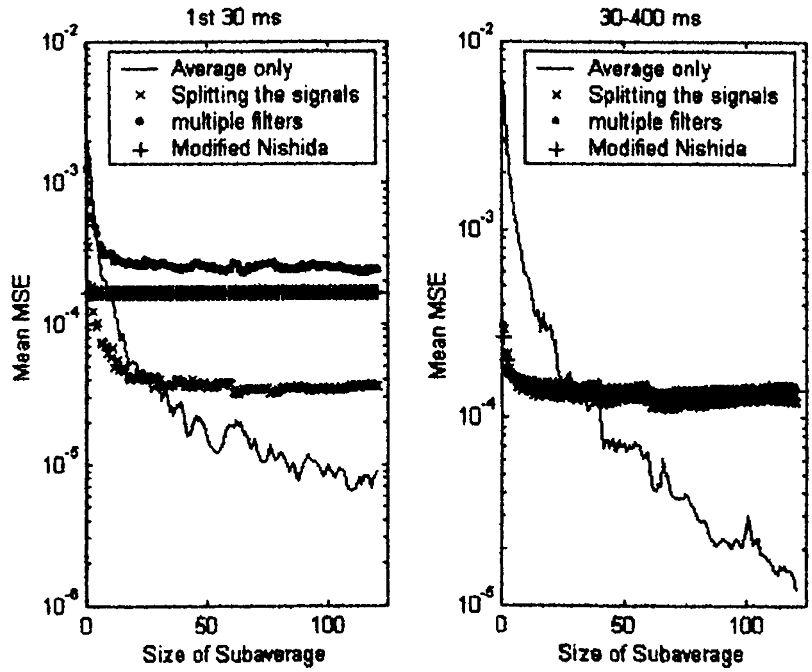

Figure 7-26 Comparisons of time-varying methods for data set 4

7.5 Conclusions

Splitting the signal into two separate signals and using three regions per separate signal, with two filters per region, produces lower mean MSE values than using three regions per signal with two or one filter per region. This confirms some of the results seen in the previous chapter where the results for the early components are improved by splitting the signals into early and late components. In this chapter the late components mean MSE values are also improved by the time-varying filters. This improvement is seen in both in terms of the lower mean MSE values shown in figures 7-12 to 7-16, but also in the extraction of peaks not previously extracted by the other methods.

All the techniques showed variations in the position of the largest peak in the region 30-400 ms for the test subset of data set 1. As the simulated subsets were constructed from the target signal of data set 1, a direct comparison between the effect of the methods can be made. This variation in the latency of the peak does not appear in the simulated data where variation in latency was not included, but it can be seen in results for data set 1. This suggests that the system can cope with relatively small variation of latency in the dominant peak. When looking at the smaller late latency components between 200-300 ms, these peaks did not always appear in the filtered results of data set 1, using a single filter per region. When the second filter was added these components were usually present. The addition of this extra filter allows peaks to be extracted from a different region to the first filter, again suggesting that using a single filter will not going to solve the problem of extracting evoked potentials from a noisy signal. The filters are still assuming stationarity, but now only over the part of the signals that each combination of filters is applied to.

For smaller number of averages (<20 signals/average) both splitting and multiple filters were often better than averaging-alone, the exception being the multiple filter results for data set 4.

A direct comparison with some groups work (e.g. such as Paradiso et al, 1995, Maccabee et al (1986), or Rossini et al, 1981) for short latency components is not entirely meaningful as they used filters to discard low frequency components in order to extract the higher frequency components of some of the smaller features. Looking at the representation of the frequencies of the filters and weighting for the time-varying technique of splitting the signals, in figure 7-17 to 7-18 for simulated and data set 1, a filter is shown for the largest segment of this region that is able to keep high frequency components, but a larger weighting is given for the a narrow lower frequency filter. The results for data set 2 showed a wide band frequency components starting at low frequencies and going up to 400 Hz, but this time the dominant filters were within this range but also relatively narrow. This fits in with the shape of the target signal where there is a dominant low frequency component with smaller features superimposed. Figure 7-20 to 7-21, the scalp recordings did not show a filter for a wide band of high frequency components. Instead the results showed filters with narrow bands. The later components do not appear to be better than the frequencies suggested by other groups except in part Maccabee et al (1986), who used a 5-3000 Hz bandpass filter and the late latency part of Nishida (1993) (5 – 17.5Hz). In general bandpass filters with low frequency components were the dominant filters.

	Modified Nishida Mean Squared Error ($x10^{-3}$)			Multiple filters/Region Mean Squared Error ($x10^{-3}$)			Splitting the Signal Mean Squared Error ($x10^{-3}$)		
Data Set	Min	Max	Mean	min	max	Mean	Min	Max	Mean
Simulated	0.1126	0.3987	0.2281	0.0422	0.2293	0.1311	0.0627	0.1103	0.0863
1	0.2176	0.5542	0.4014	0.0872	0.1666	0.1251	0.0515	0.1539	0.0866
2	0.0522	0.1601	0.1010	0.0470	0.2176	0.1006	0.0168	0.1497	0.0482
3	0.0915	0.0922	0.0918	0.0842	0.4960	0.2111	0.0346	0.1315	0.0914
4	0.1640	0.1642	0.1641	0.0792	0.2272	0.1288	0.0311	0.1155	0.0572

Table 7-2Results for the first 30 ms

	Modified Nishida Mean Squared Error ($x10^{-3}$)			Multiple filters/Region Mean Squared Error ($x10^{-3}$)			Splitting the Signal Mean Squared Error ($x10^{-3}$)		
Data Set	Min	Max	Mean	min	max	Mean	min	Max	Mean
Simulated	0.4192	0.8107	0.5719	0.1863	0.7515	0.4593	0.1139	0.5700	0.2983
1	1.1563	3.4327	2.102	0.4382	3.4062	1.6211	0.3274	3.3581	1.2555
2	0.0692	0.0801	0.0739	0.0553	0.1197	0.0761	0.0391	0.1195	0.0666
3	1.1112	1.2396	1.1714	0.5995	1.8312	1.0159	0.3895	2.6819	0.9252
4	0.1221	0.1912	0.1514	0.0804	0.234	0.1429	0.0884	0.2411	0.1441

Table 7-3 Results for the region 30-400 ms

8 Wavelets and Evoked Potentials

In the previous chapters linear filters were used, with different methods applied to making a filter bank have time-varying properties. Wavelets are a group of techniques that fit the need for filter banks and the ability to process nonstationary data. In this chapter the aim is not to perform a detailed analysis of wavelets for processing SEPs, as there is a growing body of work in this area (e.g. Bartnik et al. (1992); Bertrand et al (1994); Blinowska and Durka (1997); and Samar et al. (1995,1996,1999)). The aim of this chapter is to investigate whether evolutionary algorithms can select wavelets and coefficients to extract evoked responses.

8.1 Introduction to Wavelets

Wavelet analysis has become a widely used set of techniques to analyse and process signals and images. A general outline of these techniques is presented here, as well as placing wavelets in the context of other signal processing techniques. More detail introductions are available (e.g. Bentley and McDonnell (1994); and Strang and Nguyen (1997) and from a mathematical perspective, Daubechies (1988,1990) and Mallat (1998)).

The most commonly used method of signal analysis is Fourier analysis, in which the signal is broken down into its constituent sinusoids of different frequencies and relative phase. In other words, Fourier analysis is a mathematical approach to transform the signal from a time-based representation to a frequency-based one. The technique for doing this is the Fourier Transform.

$$F(\omega) = \int_{\infty}^{\infty} f(t)\ \exp(j\omega t)\ dt \qquad \textbf{Equation 8-1}$$

This is the sum over all time of the signal f (t) multiplied by a complex exponential. Complex exponentials can be broken down into real and imaginary sinusoidal components, hence the reason for sine waves being the basis of the Fourier analysis.

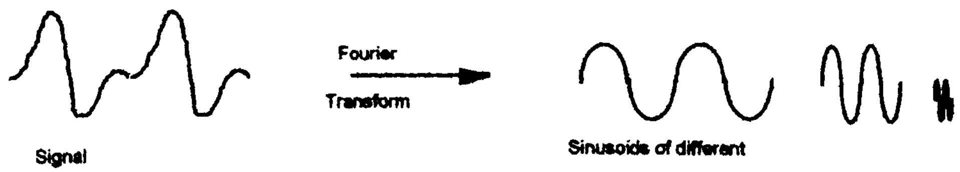

Figure 8-1 Principles behind Fourier transforms

The results of the transform are the Fourier coefficients F (ω) of the signal, which when multiplied by a sinusoid of frequency ω produce the sinusoidal components of the original signal

This technique is useful for many signals, as the frequency components of the signal can contain important information. However, there is a drawback to this approach. When transforming from a time-based signal to a frequency-based representation, time information is lost. So when looking at the transformed signal, it is not possible to tell when a particular event took place. If a signal does not change significantly over time it is stationary and so this drawback is not important. If the signal contains components that vary over time, such as drift, or abrupt changes, the signal is nonstationary and where in the signal a particular event occurred is lost. These characteristics are often important, so Fourier Analysis is not suited to for use with nonstationary signals.

8.1.1 Short Time Fourier Transforms

Trying to correct this weakness, the Fourier Transform was adapted to look at small section of the signal at a time. This process is called the Short-Time Fourier Transform (STFT). This transform maps the signal onto a function not only of frequency, but also of time. The transform works by only carrying out the Fourier Transform on a signal enclosed within a window. The window moves along the time axis and then performs the Fourier Transform on the signal within this new window. These windows often overlap. This process continues until the whole signal has been analysed.

This process provides some information about both when and with what frequency components a signal event occurred. This only obtains information with limited precision and the precision is determined by the window size.

8.1.2 Wavelet Analysis

Wavelet analysis provides a more flexible approach. One advantage of wavelet analysis is its ability to perform localised analysis, i.e. to analyse a small area of a signal. Therefore, what is wavelet analysis and what are wavelets? A wavelet is a waveform of effectively limited duration that has an averaged value of zero. Fourier analysis works by 'breaking' the signal into sine waves of varying frequencies. In wavelet analysis, the signal splits into components of scaled and shifted versions of a basic original (or mother) wavelet.

8.1.3 What is a wavelet?

One definition of a wave is an oscillating function, such as a sinusoid, it is periodic so theoretically it can continue for infinite amount of time, so has infinite energy. A wavelet is a 'small wave', it may oscillate but it does not repeat and as time increases, it tends to zero, therefore it has finite energy. This property means that if it is used as a window function, signal components in time away from the window are going to have no significant contribution to the analysis until the 'window' is used to analyse them. Fourier analysis is based on sinusoids, so it can be thought of as wave analysis, the whole signal contributes to the analysis. Because of their finite range wavelets provide more localised analysis.

8.1.4 Continuous Wavelet Transform

The continuous wavelet transform (CWT) is the sum over all time of the signal

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\psi(scale, position, time)dt \qquad \textbf{Equation 8-2}$$

multiplied by a scaled, shifted version of a wavelet ψ.

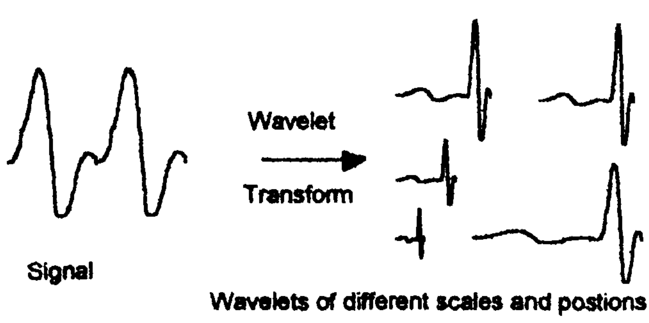

Figure 8-2 Basic principles of wavelets

The results of the CWT are wavelet coefficients C, which are functions of scale and position. Multiplying each coefficient by the appropriate scaled and shifted wavelet yields the component wavelets of the original signal. Another way of thinking about the value C, is that it is a measure of how well the section of signal under investigation correlates with the wavelet, the higher the value of C the greater the similarity. Two terms have been mentioned, shifting and scaling, but yet not discussed. Shifting a wavelet means either delaying or hastening its onset, so if a wavelet has a function $\psi(t)$, a delayed wavelet by a time interval k has a function $\psi(t-k)$. Scaling a wavelet means to stretch or compress it. A scaling factor is usually signified by **a**.

8.1.5 Scalogram

One point to note is that in wavelet analysis there is no frequency component but a scale component. In a similar way to the time-frequency plots formed using an STFT, wavelets produce a different view of the signals properties in the form of a scale-time plot. The higher scale values of **a**, correspond to more stretched wavelets. The longer the wavelet, the longer the portion of the signal that is being compared (similar to the larger the window in STFT, the larger the portion of the signal considered each time.) and the coarser the signal peaks measured. For a low scale value of **a** (i.e., a compressed wavelet), the shorter the portions of the signal considered, so the finer the details of the signals compared.

8.1.6 Why use wavelets?

An advantage of wavelets is the ability to analyse a signal at a localised level. This ability comes from scaling the wavelet, compressing the wavelet and, performing the analysis of the signal at progressively smaller area. Wavelets are essentially looking at how similar the portion of the signal is to a scaled wavelet, so wavelet analysis can reveal aspects of the signal (or images) such as discontinuities and self-similarity (fractal). Wavelet analysis has also been used to compress and de-noise signals and images. In **appendix A.3** scalogram of evoked potentials and background activity (noise) are included to show the nonstationary nature of the signals and properties of wavelets.

8.1.7 Discrete Wavelet Analysis

So far, continuous wavelet transformations have been considered. The continuous part of the name comes from the ability of the CWT to operate at every scale. It does not mean that the signal must be continuous. The signals by their very nature, since the signals are being processed on a computer, are discrete. Having to calculate wavelet coefficients for every scale is computer intensive, producing a considerable amount of data. If the scales and position selection are based on powers of two (or dyadic), the analysis is more efficient and just as accurate (Strang and Nguyen, 1997). This type of analysis is called the Discrete Wavelet Transform (DWT). The Mallat algorithm (Mallat, 1989) is an efficient way to implement this and is based on the idea of two channel subcoding, or pyramidal decomposition. The wavelet coefficients quickly emerge, so it is sometimes also called the fast wavelet transform.

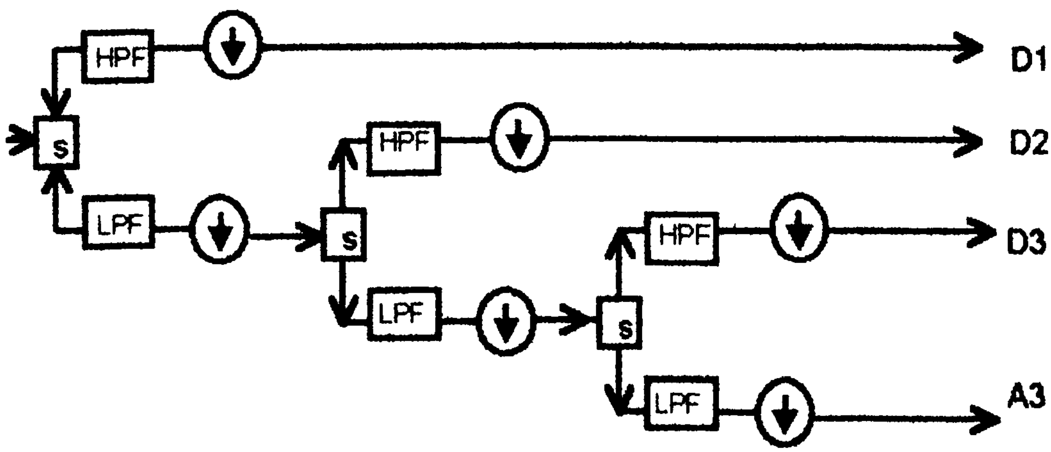

Figure 8-3 Decomposing a signal into details and an approximation, HPF- High-pass filter, LPF- low-pass filter.

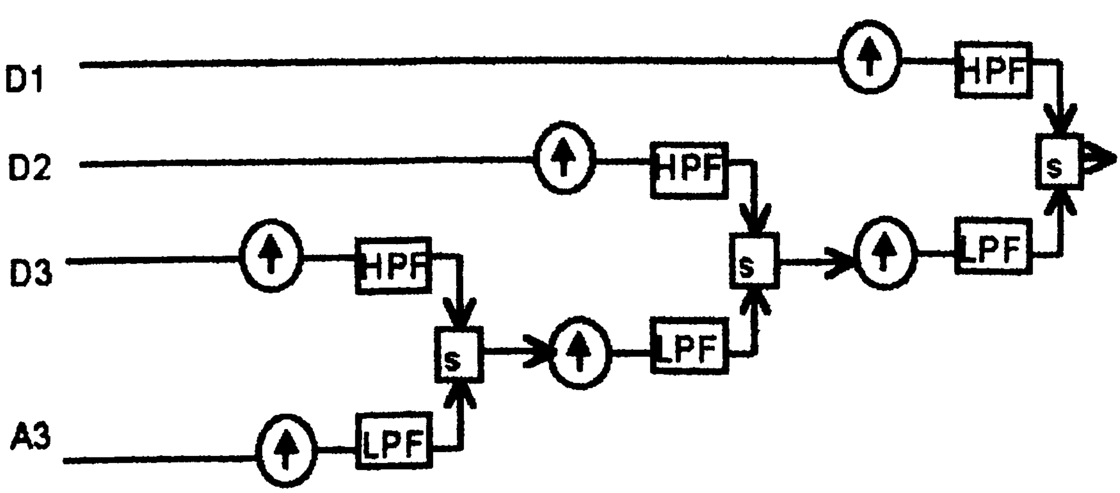

Figure 8-4 Reconstructing a signal from the details and approximation. HPF- High-pass filter, LPF- low-pass filter.

The signal goes through a low-pass filter and a high-pass filter and emerges as two signals (figure 8-3). The output of the high-pass filter is the detailed signal and the low-pass filter output is an approximation. The high-pass filters used are wavelets and the low-pass filters are the scaling functions. To reduce the number of points in each signal and therefore avoiding having a lot of data stored, the detail and approximations are down sampled. Down sampling here means discarding every second data point. The process is repeated several times with the approximation forming the input to the next stage, so the signal is broken down into many lower resolution components. The number of stages (or levels) could theoretically be continued until a single pixel or sample point is formed, but this is not practicable. In practice, the number of levels selected is usually based on the nature of the signal. To reconstruct the signal (figure 8-4) the process is carried out in reverse with the inputs being the detail signal and the final approximation and up sampling this time. Up sampling is the process of lengthening the signal by adding zeros between each sample. The equation for the reconstructed signal, using the notation shown in figure 8-4, is: -

$$S=A3+D1+D2+D3 \qquad \textbf{Equation 8-3}$$

So far, in the discussion about the DWT, the use of low-pass and high-pass filters was been discussed, but no mention of wavelets. The high-pass filters are the wavelets, and the low-pass filters form the approximation.

8.1.8 Denoising

Techniques to reduce the noise in a signal have been developed for decomposed signals. Donoho and Johnstone (1994) made the assumption that the large detail coefficients of a noise corrupted signal will be those of the signal and the others will be components of the noise. This assumes that the noise is white noise. The denoising technique starts by setting a threshold, so that if the absolute value of a coefficient is below the threshold the coefficient is set to zero. This is known as hard thresholding. A second method, soft thresholding, is an extension of the previous method. As well as reducing coefficients that are less than the threshold value to zero, the other coefficients are reduced by the threshold value (Donoho (1995), Donoho et al. (1995)).

8.2 Combined evolutionary algorithm and wavelet approaches

Approaches that combine evolutionary algorithms and wavelets have been investigated previously for other applications; Lankhorst and Lann (1994) used spline wavelets and an evolutionary algorithm to approximate signals. They selected parameters such as dilation, translations and amplitude for each wavelet. A different method is investigated here, in line with the idea of splitting the signal into two portions (first 30 ms and 30 to 400 ms) and then processing them separately as in sections 6.5 and 7.3. Here the filters are replaced with wavelets. The discrete wavelet transform using Mallat's method decomposes the signal (figure 8.3) into a set of 'details' and an approximation of the signal at the lowest level is examined. This can be viewed as a set of bandpass filters (the filters may be high-pass filters but in combination with downsampling they become bandpass filters) and a single low-pass filter, in other words a filter bank. The aim of the evolutionary algorithm this time is to weight the outputs of these filters (figure 8-5) and select which wavelet to use (from a list of 46 wavelets). The fitness function was the same as in section 7.3, the mean value of the MSE values of the two regions.

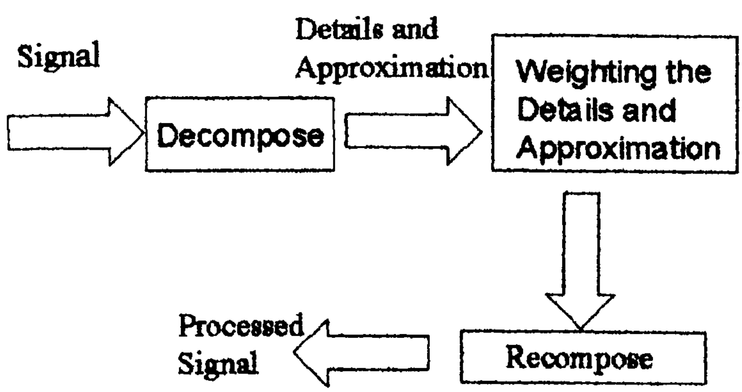

Figure 8-5 Outline of the combined wavelet and evolutionary algorithm approach.

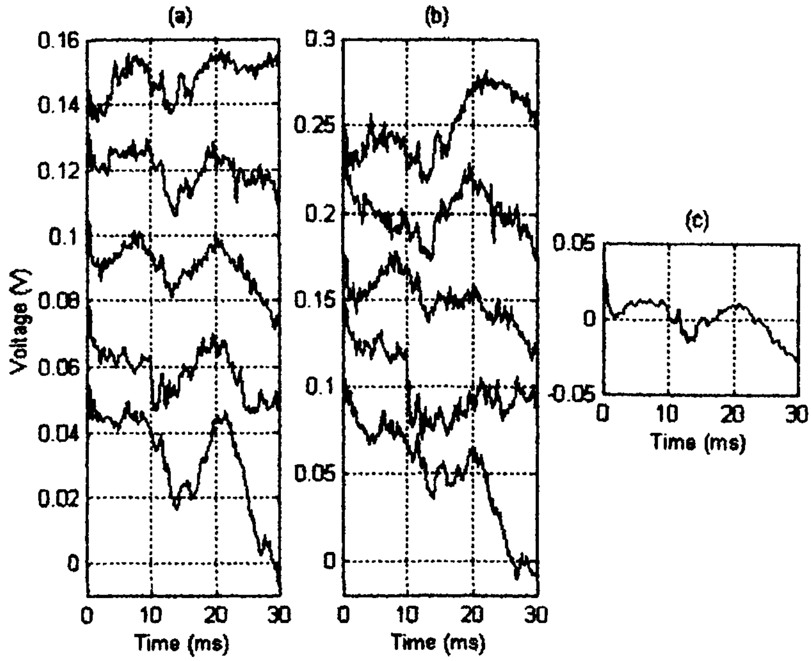

Figure 8-6 Wavelet filtering approach applied to the test subset of the simulated data set (first 30 ms). (a) After filtering, (b) before filtering, (c) target signal.

In figures 8-6 to 8-9 the filtered signals are noisier than the methods in chapter 6 and 7 but the smaller features can be seen that appear in the target signal but were lost by the other methods

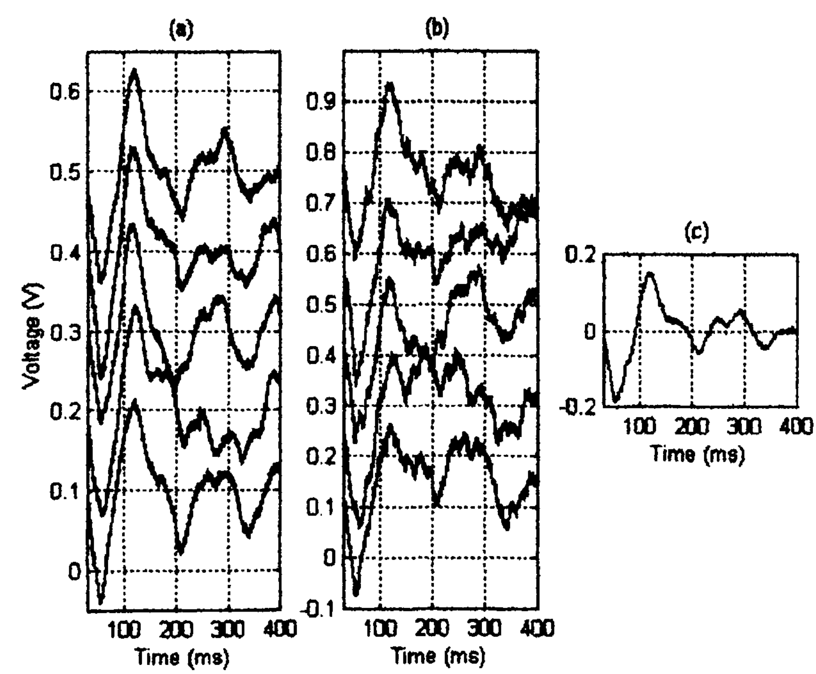

Figure 8-7 Wavelet filtering approach applied to the test subset of the simulated data set (30-400 ms). (a) After filtering, (b) before filtering, (c) target signal.

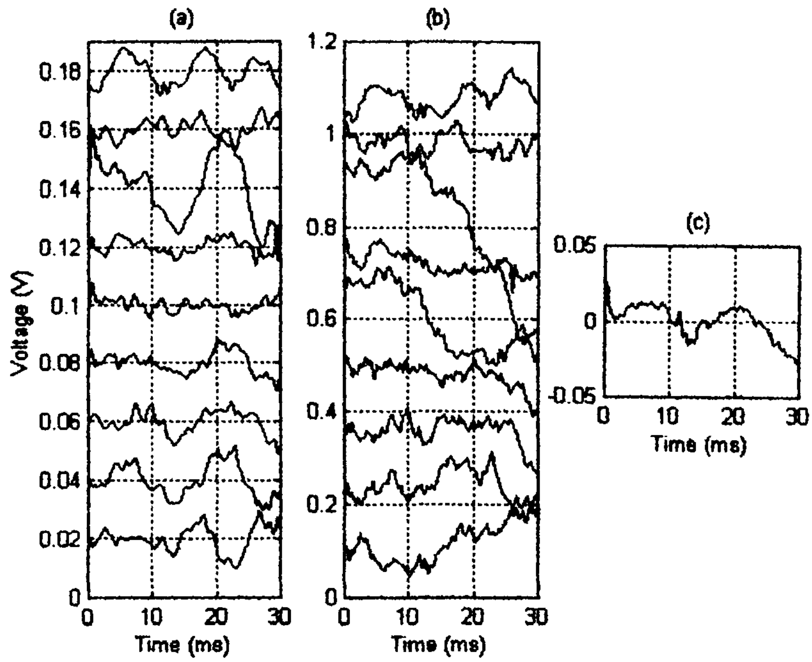

Figure 8-8 the wavelet filtering approach applied to the test subset of data set 1. (a) After filtering, (b) before filtering, (c) target signal.

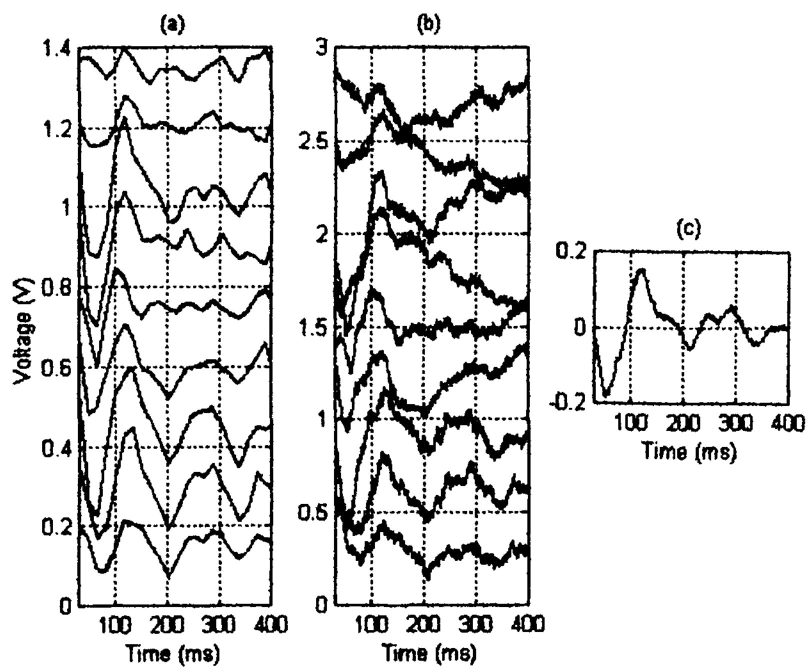

Figure 8-9 Wavelet filtering approach applied to the test subset of data set 1 (30-400 ms). (a) After filtering, (b) before filtering, (c) target signal.

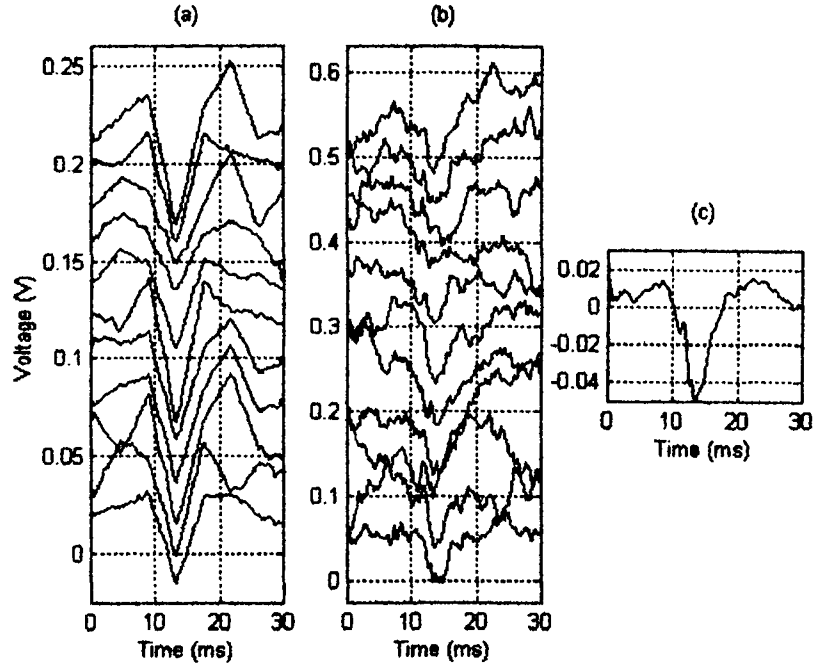

Figure 8-10 the wavelet filtering approach applied to the test subset of data set 2 (first 30 ms). (a) After filtering, (b) before filtering, (c) target signal.

Figure 8-10 shows a problem using wavelets where the larger features of the signal fit the shape of the wavelet and the smaller features do not, so the smaller features are lost.

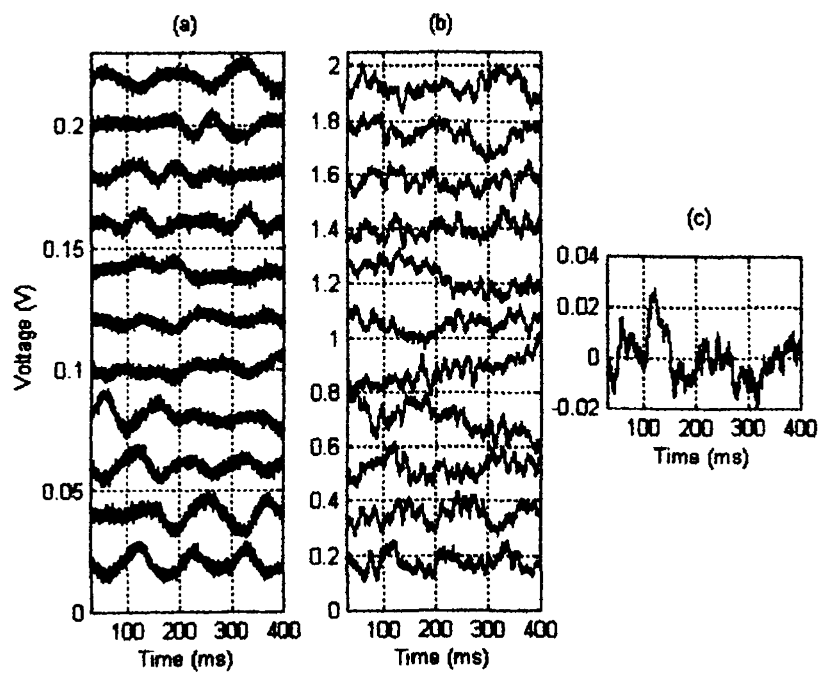

Figure 8-11The wavelet filtering approach applied to the test subset of data set 2 (30-400 ms). (a) After filtering, (b) before filtering, (c) target signal.

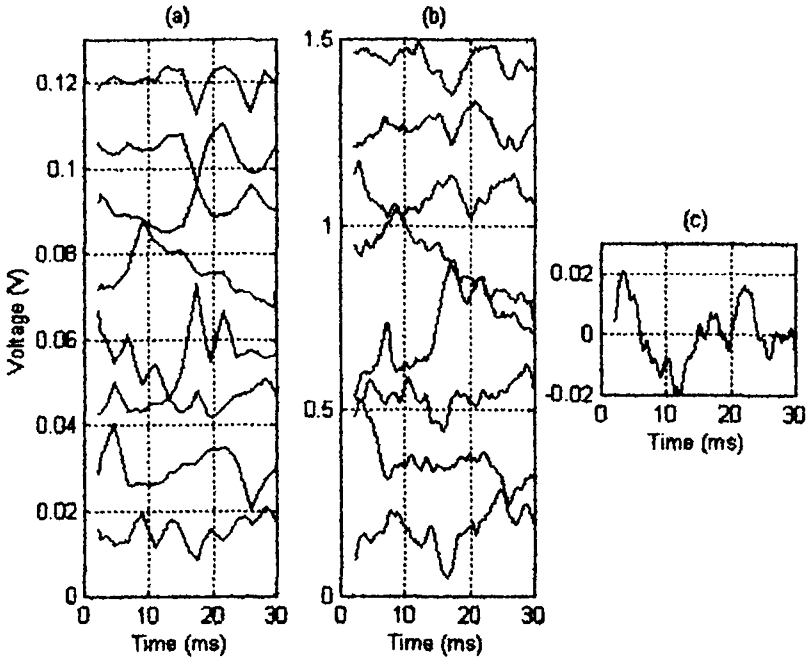

Figure 8-12 the wavelet filtering approach applied to the test subset of data set 3. (a) After filtering, (b) before filtering, (c) target signal.

Figure 8-11 shows results for the late components of data set 2, showing signals with low frequency content with a small amount of noise, similar to the target signal. As observed in the previous chapter the results for data set 3 (figures 8-12 and 8-13) show many of the key features of the target signals.

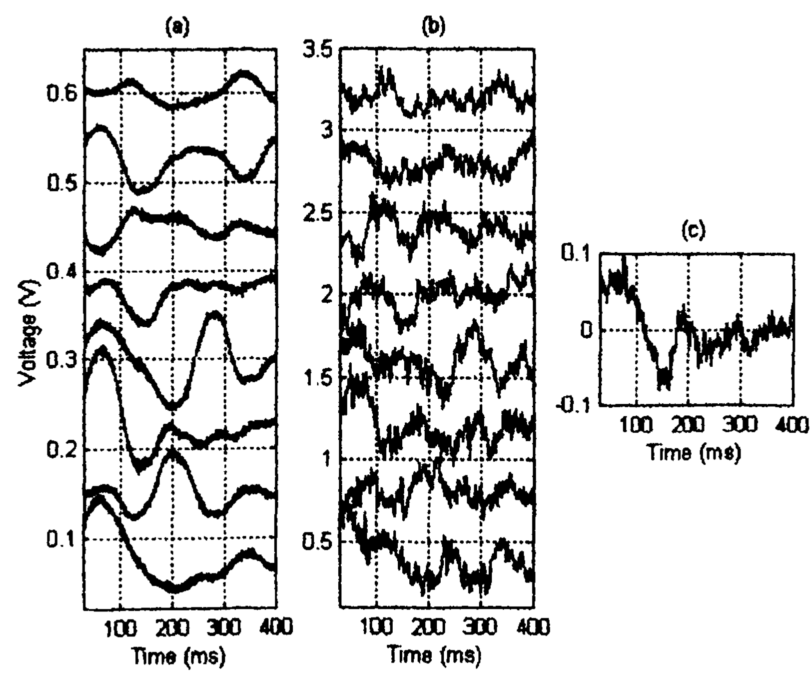

Figure 8-13 The wavelet filtering approach applied to the test subset of data set 3 (30 to 400 ms). (a) After filtering, (b) before filtering, (c) target signal.

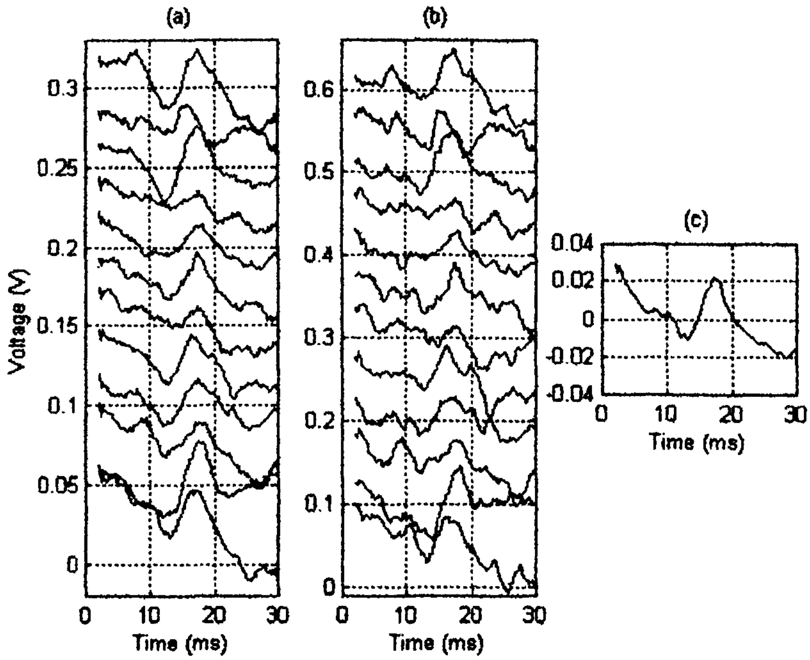

Figure 8-14 the wavelet filtering approach applied to the test subset of data set 4. (a) After filtering, (b) before filtering, (c) target signal.

Both the unfiltered and filtered early components of data set 4 (figure 8-14) were similar to the target signal, with the filtered signal producing visually greater similarity to the target signal and also in terms of MSE (see figure 8-25).

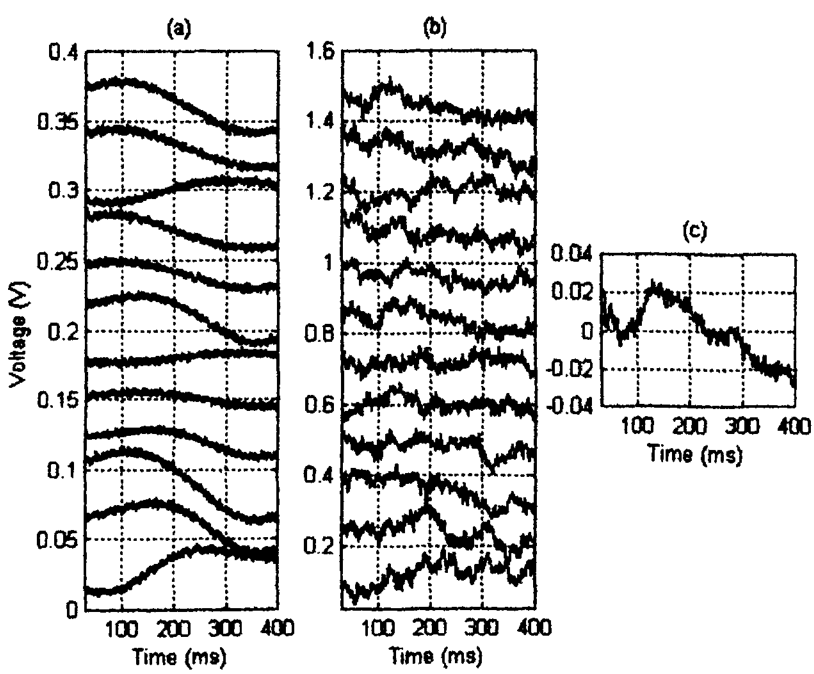

Figure 8-15The wavelet filtering approach applied to the test subset of data set 4 (30-400 ms). (a) After filtering, (b) before filtering, (c) target signal.

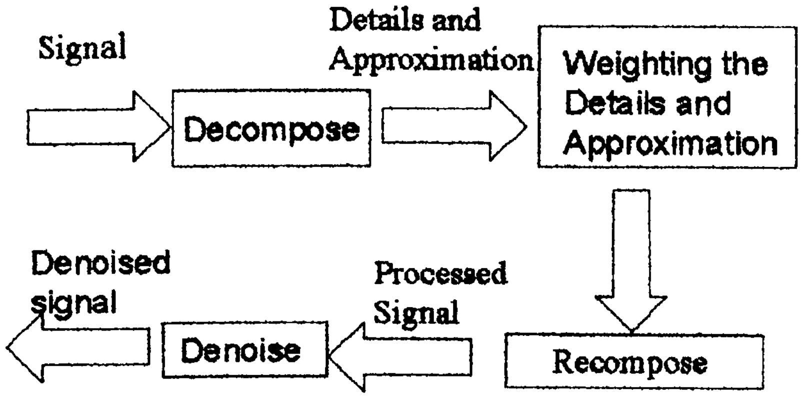

Figure 8-16 Outlines of the wavelet and evolutionary algorithm with denoising as a post-processing stage

As has already been mentioned (section 8.1.8) wavelets can be used to denoise the signals. This was applied as a further stage to this process (figure 8-16). After the evolutionary algorithms have selected the wavelet and weighting for the levels, whilst processing the test data the signal were further processed using a denoising algorithm and the wavelet selected to lower the noise. This process is separate to the evolutionary algorithm, performed after processing with the model developed using evolutionary algorithms.

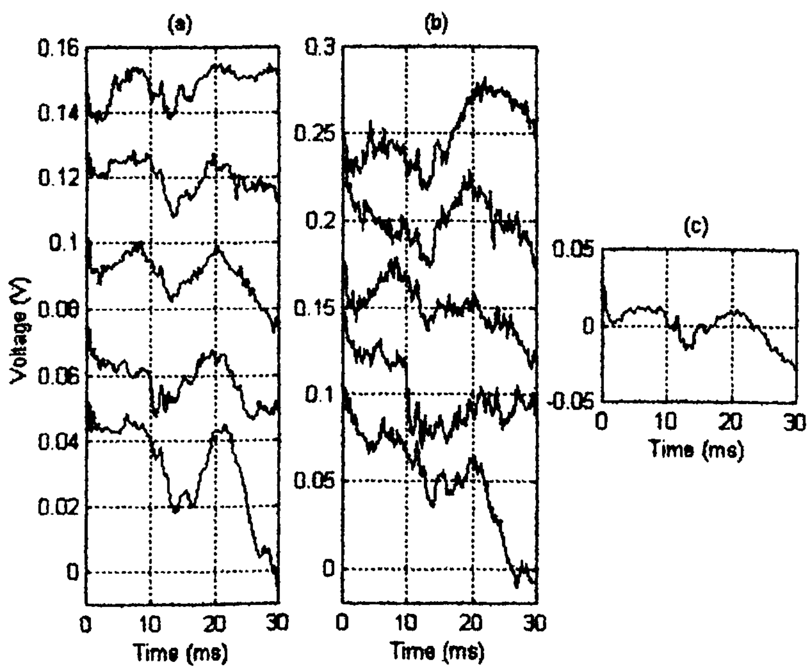

Figure 8-17 Denoising applied to the first 30 ms of the simulated test data. (a) After filtering, (b) before filtering, (c) target signal.

Comparing the effect of the extra denoising stage (figures 8-17 and 8-18), this extra stage is applied to the first 30 ms of the test sets of the simulated data and data set 4, with the results of the wavelet approach (figures 8-6 and 8-14 respectively). What is shown here is that the denoising algorithm has only a small effect on noise removal, removing relatively low magnitude high frequency components (but can improve the visual representation of the signals).

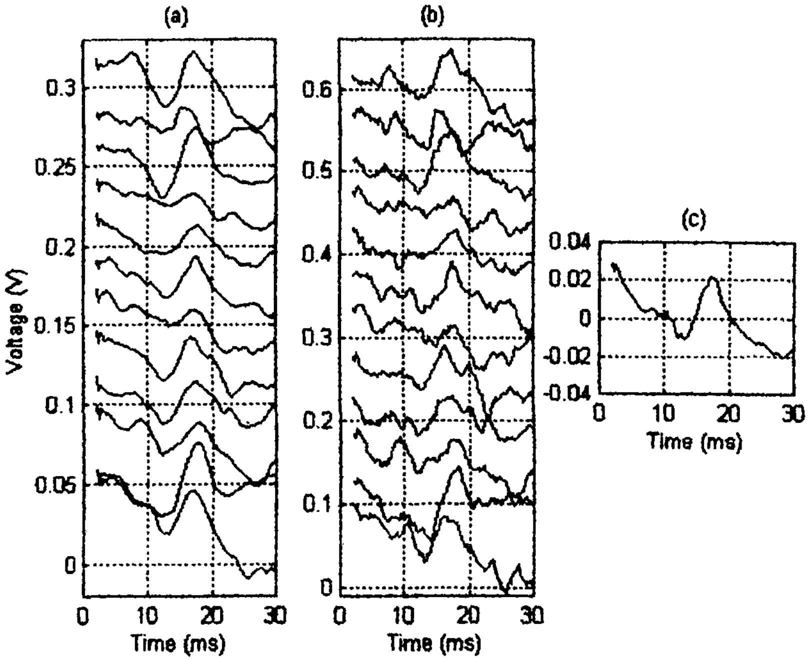

Figure 8-18 Denoising applied to the first 30 ms of data set 4. (a) After filtering, (b) before filtering, (c) target signal.

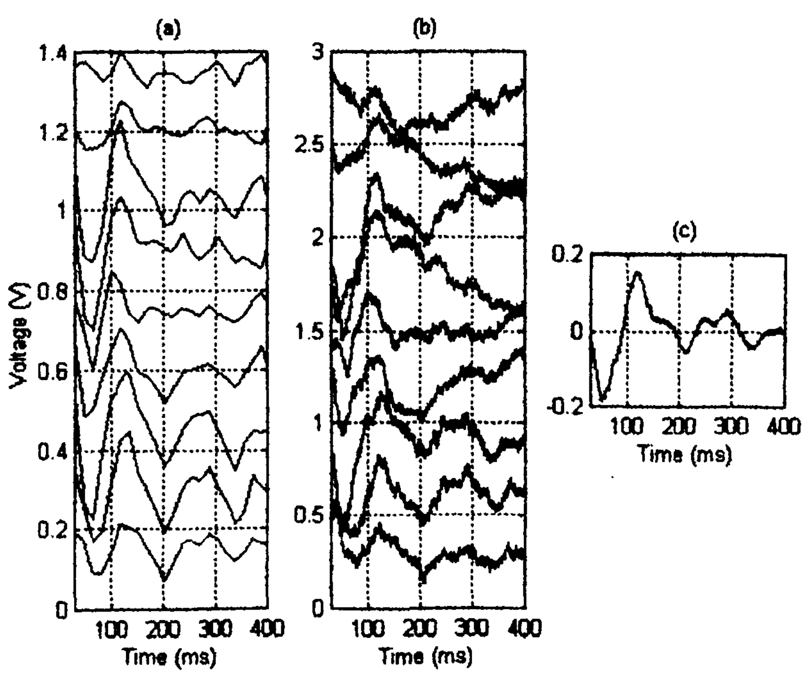

Figure 8-19 Denoising applied to the region of 30-400 ms of data set 1. (a) After filtering, (b) before filtering, (c) target signal.

The effect of denoising is more noticeable when applied to the late components (30 - 400 ms). Comparing the non-denoised results of data set 1 and data set 2 (figures 8-9 and 8-11 respectively) with the signals after denoising (figure 8-19 and 8-20). The denoising stage removes what appears to be the random noise left after filtering, which would fit in with the theory of denoising (Donoho et al. 1995).

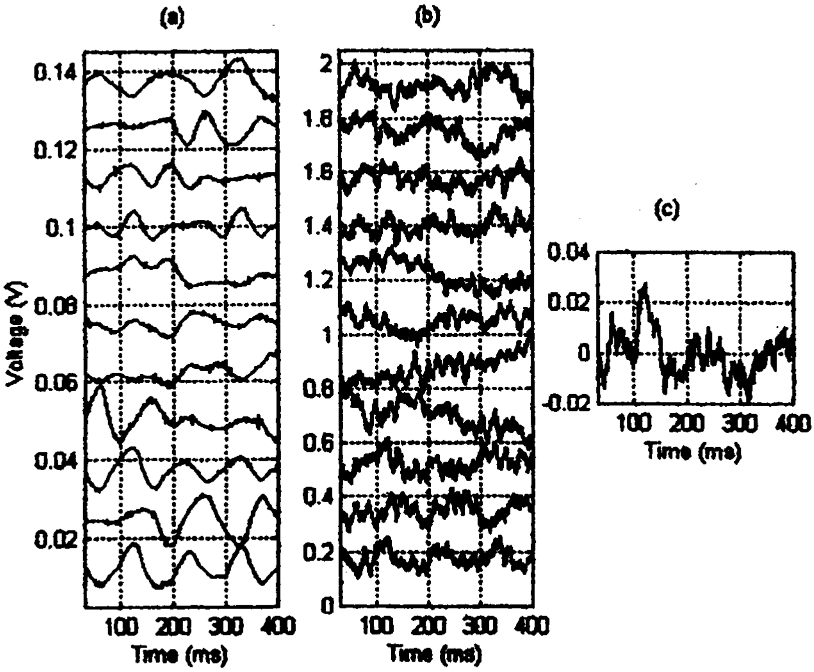

Figure 8-20 Denoising applied to the region 30-400 ms of data set 2. (a) After filtering, (b) before filtering, (c) target signal.

Comparisons of the wavelet methods with and without the extra denoising stage are shown in figure 8-21 to 8-25. In terms of mean MSE there is little difference between the two methods.

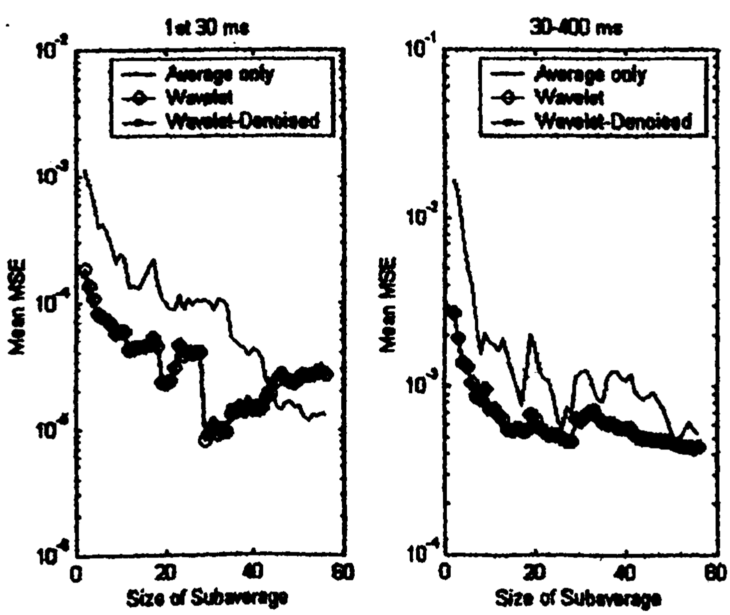

Figure 8-21 A comparison of the wavelet method and the wavelet method with a denoising stage for the early (first 30 ms) and late components (30-400 ms) of the test data of the simulated data set.

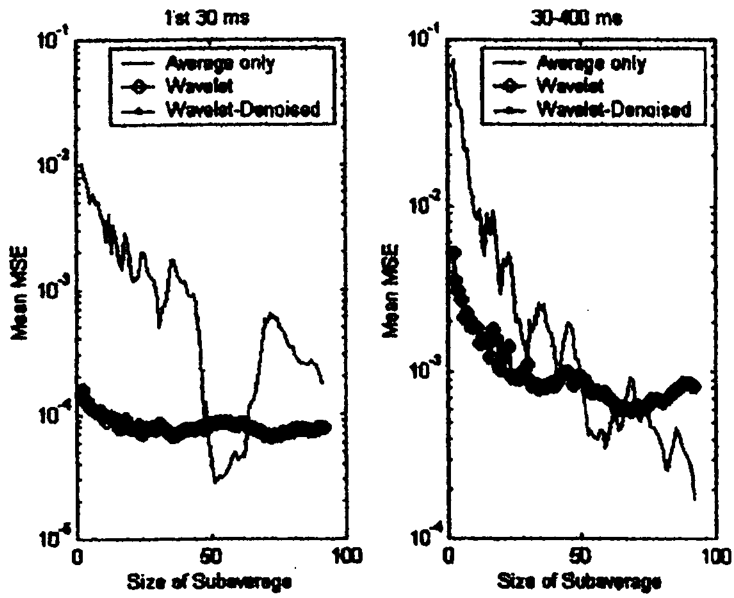

Figure 8-22 A comparison of the wavelet method and the wavelet method with a denoising stage for the early (first 30 ms) and late components (30-400 ms) of the test data of data set 1.

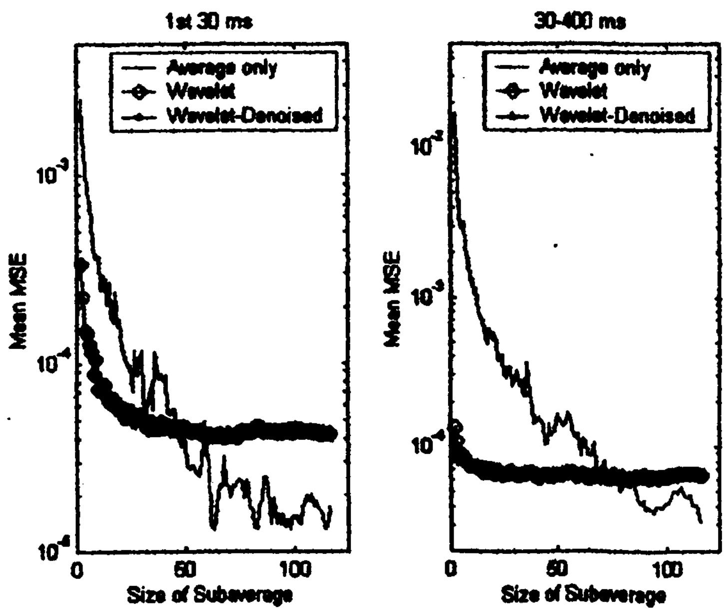

Figure 8-23 A comparison of the wavelet method and the wavelet method with a denoising stage for the early (first 30 ms) and late components (30-400 ms) of the test data of data set 2.

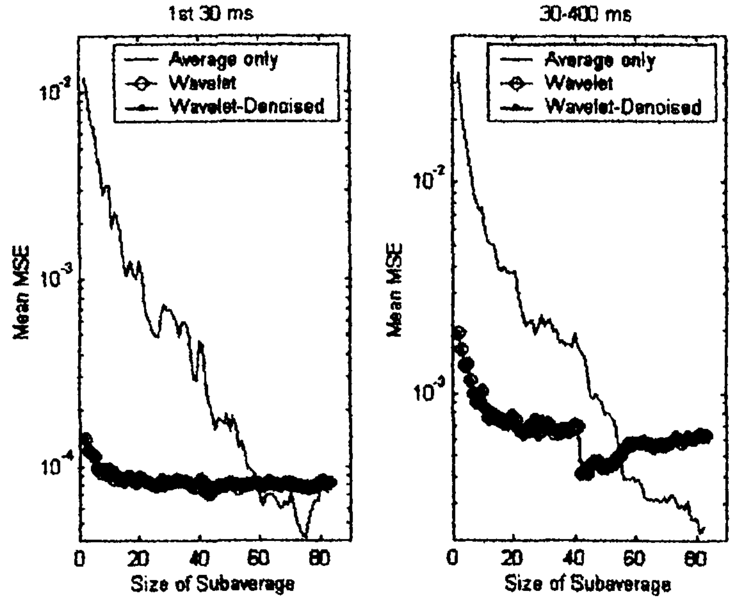

Figure 8-24 A comparison of the wavelet method and the wavelet method with a denoising stage for the early (first 30 ms) and late components (30-400 ms) of the test data of data set 3.

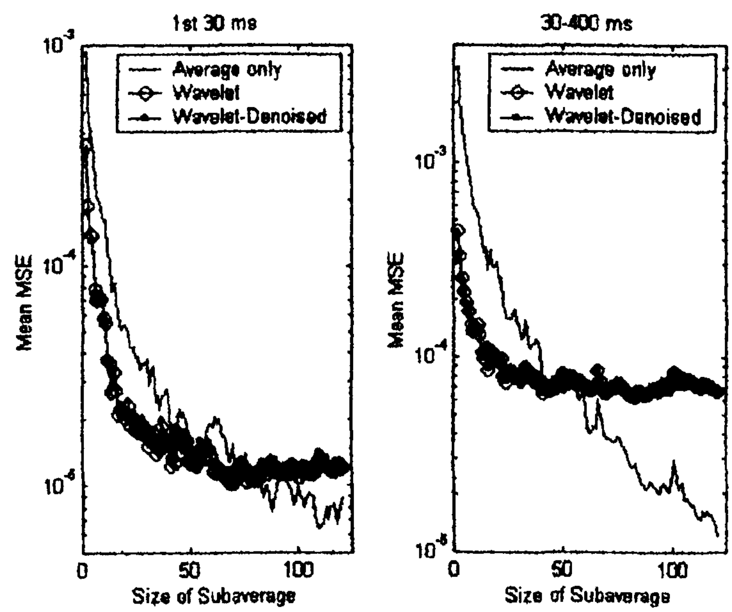

Figure 8-25 A comparison of the wavelet method and the wavelet method with a denoising stage for the early (first 30 ms) and late components (30-400 ms) of the test data of data set 4.

In comparison with the time-varying methods of chapter 7 the results are similar to those of splitting the signals (section 7.3). Averaging-alone was found to only better or similar to these filtering methods when the size of the subaveraging is relatively large (greater than 40 to 50 signals per average).

8.3 Discussion

Wavelets are appropriate because they are suited to dealing with nonstationary signals. Here wavelets were used to act in a similar way to a set of bandpass filters with the output of the filters, the wavelets, weighted, which was similar to the approaches in the previous chapters.

This technique is broadly similar to the approach used by several other groups for late components (e.g. Bartnik et al 1992a, Quiroga, 2000), where the levels of the wavelets were selected for inclusion in the reconstructed signal. The differences however are that the evolutionary algorithm selects the wavelet used and by adjusting the weights, the evolutionary algorithm selects how much a particular level contributes to the result. This technique differs from the approach of Lankhorst and Lann (1994) in two ways. The technique used here, again, selects which wavelet to use and the dilation and translation is taken care of by the decompostion and reconstruction stages, the algorithm contains the amplitude weighting for the wavelets. Second, Lankhorst and Lann (1994) method was developed to approximate the signals it was trained upon and so only used single examples of signals uncorrupted by noise, unlike the signals used with this study.

Both small and large features can be extracted by wavelets (see figure 8-17), but when features due to noise are similar to some of the features in the target signal then these noise components are likely to be included in the reconstructed signal.

This technique often selected two different wavelets for the two separate regions (see Appendix A). This difference is believed to be due to the difference in requirement for the two regions. Early components are more likely to contain high frequency components and the later components lower frequency components, leading to the wavelet used in the reconstruction of the late components usually being smoother than those used to reconstruct the early components. The denoising algorithms used made small visual improvements in the signals, but in terms of MSE the difference was insignificant.

9 Comparison of Methods and Discussion

In chapters 6 to 8 the methods developed were compared with averaging and for some of the methods averaging was not as good as these techniques until the size of subaverages was relatively large (greater than 40 signals per subaverage). In this chapter a comparison of the filtering methods for smaller sizes of subaverages will be produced.

9.1 Comparison of Methods

As measures of the effectiveness of the techniques developed during this work, three repeatedly used techniques have been included for comparison. The first is Wiener optimal filtering (Doyle, 1975), introduced as one benchmark. The second benchmark is ensemble averaging, which is selected because it is the most commonly used technique for extracting these types of signals. Finally, a comparison with Bertrand's wavelet optimal filtered method is also made (Bertrand et al, 1996). Bertrand's method is based on Doyle's method with the spectra replaced by wavelets and is included because it is both a wavelet method and a method of optimal filtering (see chapter 2).

9.1.1 Whole signal

A further technique using the frequencies and the time domain regions described by Nishida et al (1983) was included as a comparison with the modified Nishida approach (section 7.1). The effects of the various filtering methods developed to extract the whole signal are shown in figure 9-1. Four methods were considered: a filter bank of three filters (section 6.2), single filter (chapter 5), multiple filters (section 7.2) and the Modified Nishida approach (section 7.1).

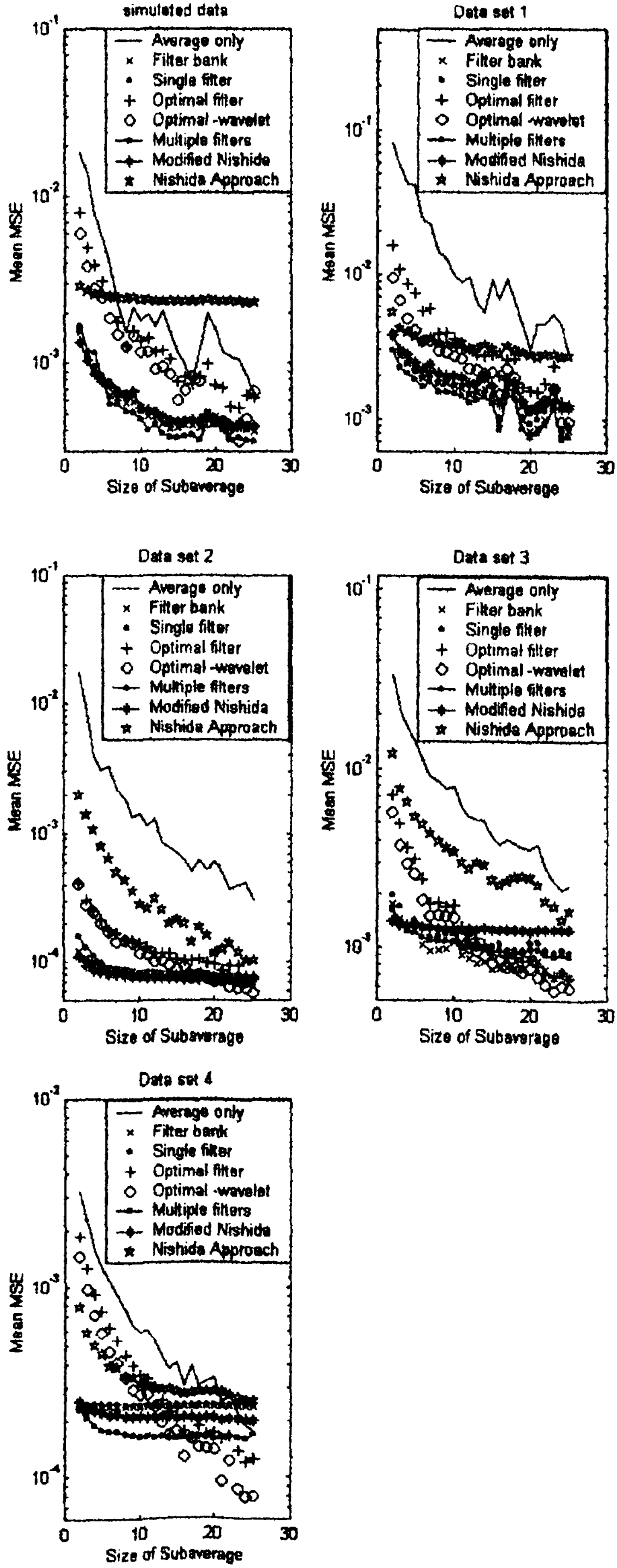

Figure 9-1 Comparison of whole signal methods

When the number of signals per average was small (less than 10), the evolutionary algorithm based methods are all better (lower mean MSE values) than optimal filtering methods and averaging-alone. For the spinal recordings, in general, optimal filtering performed worse or similarly, in terms of mean MSE, to the evolutionary algorithm filtering methods. Using multiple filters per region produced the best results for small sizes of subaveraging in three out of the five data sets. The exceptions being data sets 2 and 3. In data set 2 the performance of this technique was not as good initially, but as the size of subaverages increased, the difference between this and the other evolutionary algorithm based filtering methods became marginal. In the scalp recordings, the optimal filtering methods produced lower mean MSE than the other methods for smaller numbers of signals.

The modified Nishida and single filtering were the least effective methods. The common feature between these two approaches is that only one filter at a time is used. The main difference between them is the modified Nishida approach uses different filters at different times through the signal, whereas the single filter approach applies the same filter all through the signal. For all the data sets, as the number of signals in the average increased, the wavelet optimal filter (Bertrand et al, 1996) performed better than the optimal Wiener filtering approach (Doyle, 1975). A possible explanation of this is that as the sizes of subaverage increases, the 'high rate' features in the signal due to noise are reduced. The remaining features are more likely to be those of the signals and the wavelet can then extract them. Averaging-alone, currently most widely used, had a higher MSE in 4 out of 5 of the data sets. Its performance improves only with the simulated data set in which the underlying signal was the averaged signal.

9.1.2 Partial signal

The filtering methods were first developed and applied to the whole signal. Techniques were then developed to be applied to the signals in two parts: the first 30 ms and the region 30 to 400 ms of the signals, including the optimal filtering methods. To this aim figures 9-2 to 9-6 show the results in a similar way to figure 9-1 how the mean MSE values for these techniques varied as the size of subaverage increased. See Appendix A (Table A-4) for example mean MSE values of the various techniques.

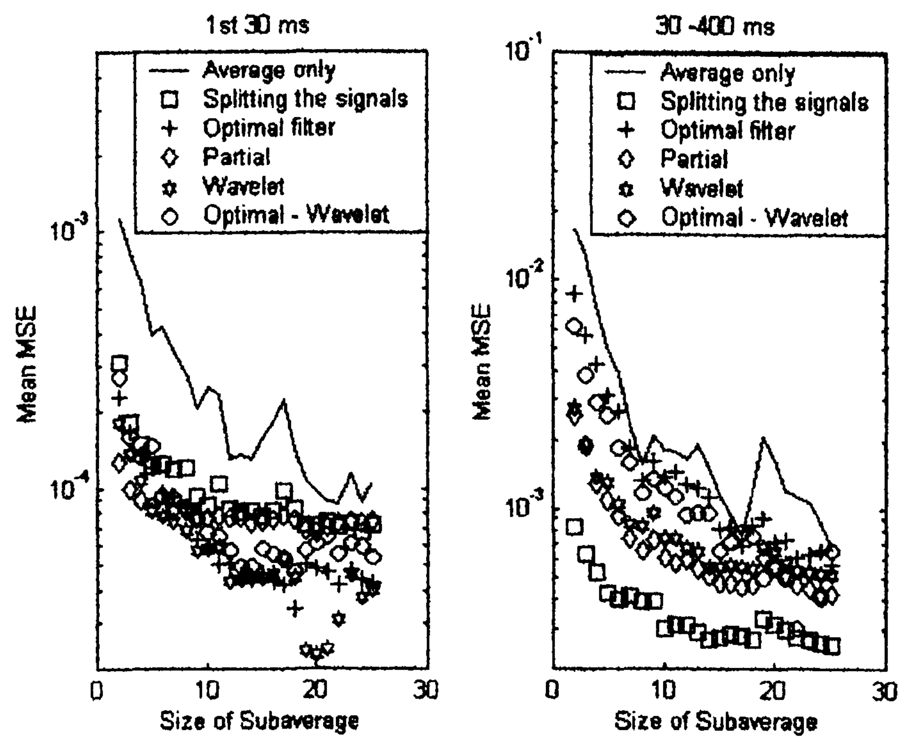

Figure 9-2 Comparison of the filtering methods for the simulated data set

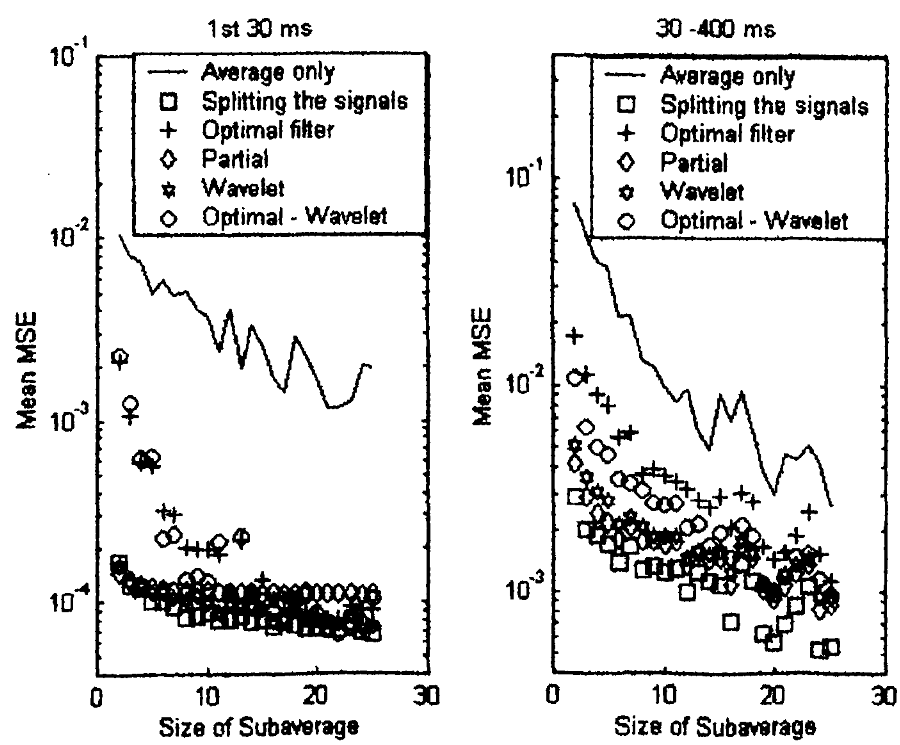

Figure 9-3 Comparison of the filtering methods for data set 1

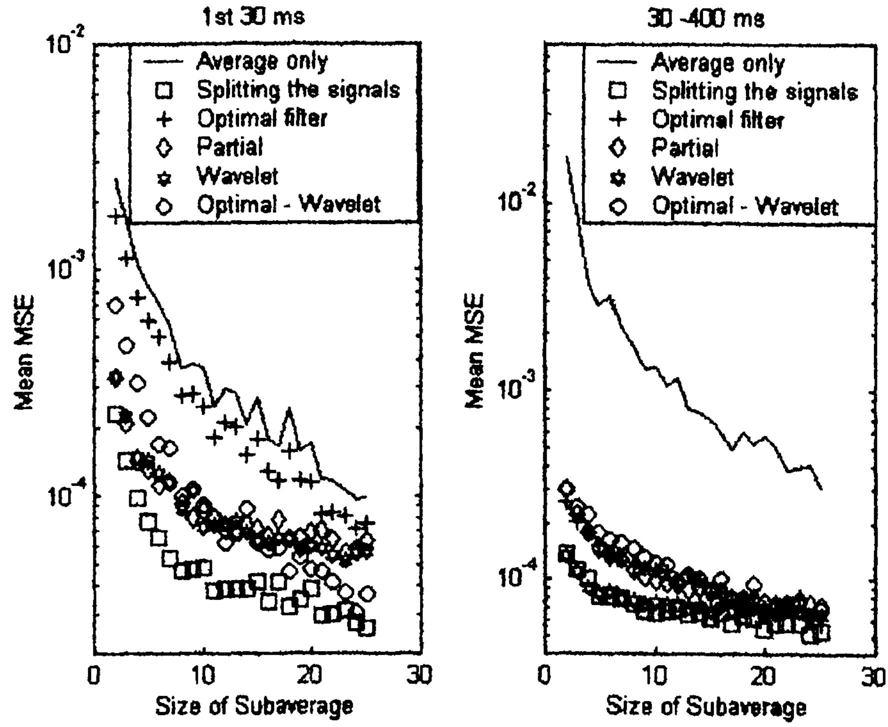

Figure 9-4 Comparison of filtering methods for data set 2

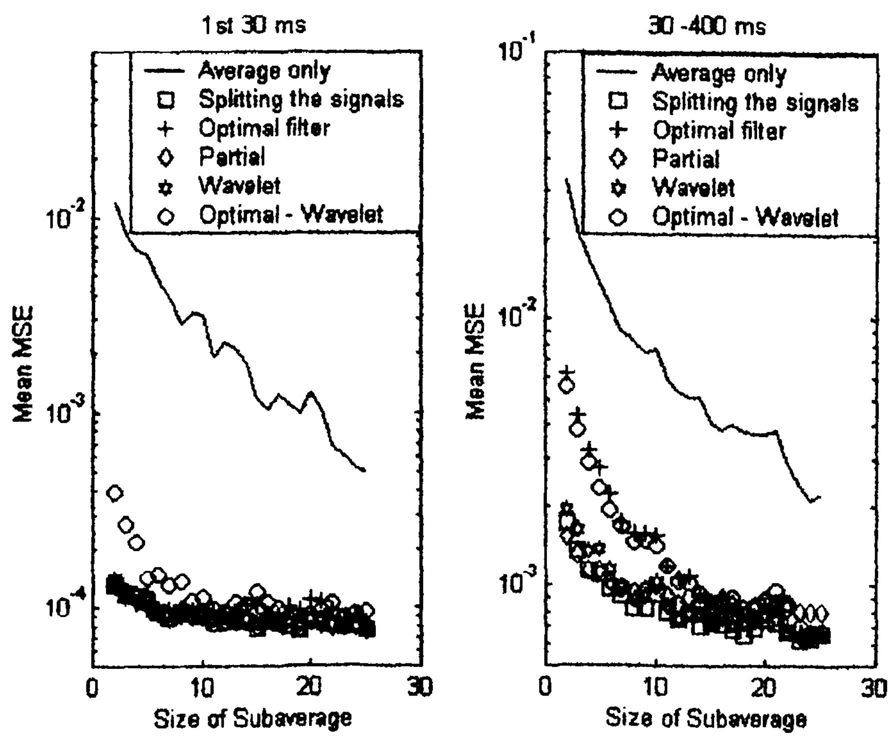

Figure 9-5 Comparison of filtering methods for data set 3

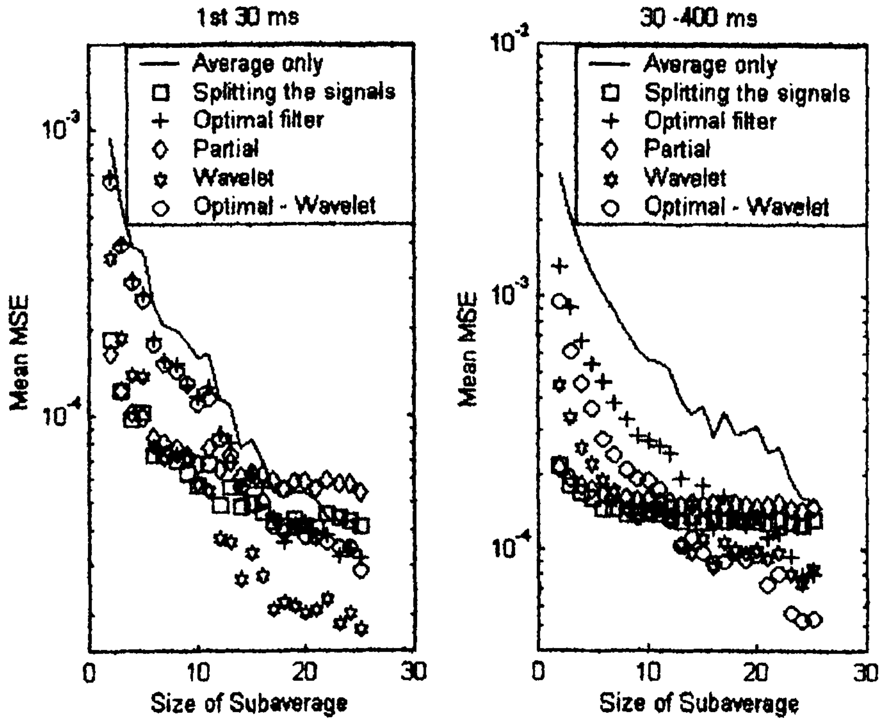

Figure 9-6 Comparison of filtering methods for data set 4

	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet
	Change %	Change %	Change %	Change %	Change %
simulated	26.5306	50.0000	30.6122	76.5306	37.7551
1	95.8186	95.1119	93.4629	95.3475	94.9941
2	77.1930	32.7485	59.0643	64.9123	72.5146
3	93.2163	91.3009	93.1346	91.0615	92.7374
4	23.2319	21.8957	-8.3917	62.2313	29.9864
mean	63.1981	58.2114	53.7365	78.0166	65.5975

Table 9-1 For the first 30 ms the percentage decrease in mean MSE of the techniques compared to the values for average alone, for subaverages of 20 signals using the equation (1-(techniques mean MSE - averaging-alone mean MSE)//averaging-alone mean MSE)

	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet
	Change %	Change %	Change %	Change %	Change %
simulated	81.3590	57.2459	65.8449	60.9140	67.3482
1	80.8749	51.6446	68.8708	65.2764	67.1075
2	90.5923	87.8049	86.7596	88.5017	88.1533
3	79.7530	77.1751	77.4973	79.0548	76.3963
4	57.5397	55.5525	51.6096	67.1724	68.6915
mean	78.0238	65.8846	70.1164	72.1839	73.5394

Table 9-2 For the 30 – 400 ms the percentage decrease in mean MSE of the techniques compared to the values for average alone, for subaverages of 20 signals using the equation (1-(techniques mean MSE - averaging-alone mean MSE)/averaging-alone mean MSE)

For the first 30 ms, in all but the simulated data set and data set 4 (figure 9-2 and 9-6) the method of splitting the signal into three regions with two filters per region (section 7-3) produced lower mean MSE (the largest percentage change for these data sets in table 9-1) values than the other techniques. In the simulated data set and data set 4, a similar method where the filters were replaced by wavelets (section 8-2) produce the lower mean MSE. For the early components, the approaches that produced the lowest mean MSE values were both time-varying techniques using more than one filter at a time. This included the results of data set 4 (figure 9-6) where the wavelet can be considered as a set of filters.

For the region of the signal between 30 and 400 ms, in all but data set 4 (figure 9-6) the method of splitting the region into three sub-regions with two filters each produced the lowest mean MSE values (see Table 9-2). The improvements in mean MSE was small (<5% difference) in both data set 2 and 3. The results of evolutionary algorithms filtering methods, for the late components, were better than the optimal filtering results, except in data set 4 (figure 9-6). The time-invariant filter bank developed by splitting the signals into these two regions in the spinal recordings was more successful for the 30-400 ms and first 30 ms, than the filter bank approach developed for the whole signal (figure 9-1). This approach work well especially for data set 2 (figure 9-4).

9.2 Discussion

The simulated data set has been useful for checking that the techniques are practical and extract some of the features. There is a lack of realistic models evoked potentials, which meant that only a time-invariant simulated data set was generated. Time-varying models for simulated data based on exponentially weighted sinusoids have been suggested by some authors (e.g. Darragh et al, 1995), without any clear justification as being realistic models. Therefore a time-invariant simulated data set and four recorded data sets were used. Though the simulated data set was based on a time invariant response, it is an averaged signal from a recorded data set (data set 1). The noise was recorded at the same site as the signals used in the averaged signal. Both the averaged signal and the noise are based on actual recorded electrical activity.

The evolutionary algorithm based techniques work best for short latency components. This is not entirely surprising as the method works by comparing the filtered signals to a fixed target signal. Short latency has less variability in position than the late components. Therefore, the filtered and target signals are more likely than the late components, to have features that are approximately (<2 ms) in the same position in time as those in the target signal. The approaches in general found the large features. If the relative positions in time of the components do not vary greatly between subaverages, these approaches could still extract the feature. This can be seen in filtered results of the late components of data set 1 (see figure 6-16). For the early components, the likely positions of the features are known, so by looking to see if these features are present in the filtered signal it can be said that the features have been extracted. For later components, because of the variations in position of features, it is not always easy to say what is and what is not a feature of a response.

Time-invariant methods were able to show that the overall shape of the signal could often be extracted, but the small details were often lost. Splitting the signal into different regions, in this case two subregions, the short latency region performed better (lower mean MSE) than the models developed using the whole signal applied to the same region. Artifacts were not produced due to phase shift, as these filters do not introduce any phase shift (see chapter 5).

Splitting the signals and using multiple filters in each region has been shown (in the comparison of methods section chapter 7) to be the best time-varying method for some data sets and in others the equivalent wavelet based results were better.

Looking at the simulated data set for splitting of the signals, the methods ability to enhance the extraction of features can be seen. It has extracted features that are present in the target, that the other time-varying and time invariant techniques (other than the wavelet based result) have not been able to extract. The assumption that there is an underlying signal throughout the data set, which is the basis of averaging, is flawed. The results suggest that time-varying filters produced better results which is in agreement with the results of some other authors (e.g. deWeerd et al, 1981a). The nonstationary nature of the signal means that improvements in the extraction of the signal are possible when the different filters operate at different portions of the signal. Wavelet analysis of recorded background activity, an averaged signal and a random signal, suggest that the background activity and the averaged evoked response show greater similarity to each other than the example of random noise (see Appendix A.3). These techniques have worked better for regions of the signals where the signal looked for is relatively stable; these are often the early components of the signal due to the nature of its production. Overall the first 30 ms could be extracted in three out of the five data sets with fewer of the smaller features.

The signals are nonstationary which is the logic behind using time-varying techniques. Wavelets, by their nature, are suitable for nonstationary signals. Their shape means that the signals they filtered are in effect windowed and the signal's energy localised. One of the advantages of wavelets is their ability to extract discontinuities in the signals making it suitable for extracting sharp features in the signals. The disadvantage of this is that noise can also produce high-rate features that can also be extracted. This could explain the appearance of the wavelet results in chapter 8.

It is important to use more than one run, and given the time constraints it was decided to run each evolutionary algorithm five times for each test. The use of only five runs for each evolutionary algorithm means that statistical analysis of the effectiveness of the techniques is limited. In this application the requirement was to achieve low mean MSE for a set of evoked responses, with a small number of runs. The techniques are slow and having to wait several days to get an answer was not appropriate. One of the difficulties with these approaches was shown in figure 5-8 where in five runs the MSE values converged to two different values. In other words, non-optimal solutions in terms of MSE, were possible. It has been concluded that the parameters of the evolutionary algorithm (e.g. weightings and frequencies of the filters) were not independent.

10 Conclusions and Scope for Future Developments

The aims of the research have been met. The claims for this work is evolutionary algorithms can be used to select sets of filters or wavelets applied to enhance the extraction of evoked potentials from noisy recordings. Using the approaches developed, evoked potentials can be extracted using fewer signals per average than is typically used in ensemble averaging. Effectiveness is measured in terms of visual inspections and MSE.

10.1 Conclusions

- Using an evolutionary algorithm to choose filters or wavelets is effective both in terms of visual comparison between the resulting signals and the respective target signals and MSE values. With current processor speeds this has to used as an off-line process.
- Techniques with a time-varying element worked better than applying the same set of filters to the whole signal.
- These worked most effectively for early (or short latency) components, or when the variability of the late components was small, as in the simulated data set and data set 1.
- Time-varying techniques were better (in terms of MSE values) than optimal filtering methods for small sizes of subaverages.
- The wavelet approach and the equivalent time-varying filter bank approach (splitting the signals) gave comparable.
- Results showed some common features such as filters with narrow bandwidth were in general selected for the early components, often with no overlapping components.
- Wavelets proved to be the better approach to extracting early components of evoked responses from a noisy signal for two reasons. Firstly, denoising can be applied as a post processing stage, which improves the visual representation of the signal. As the wavelet approach has the ability to handle discontinuous features in a signal such as small features in an evoked response, it can keep features in the results that may be lost with other techniques. The wavelet results were often better than, or similar to, the results of splitting the signal. However, the wavelet approach is very computationally intensive.

10.2 Scope for Future Development with these data sets

As a direction for future investigation there is further scope in the idea of an evolutionary algorithm being used to select filter banks or wavelets.

10.2.1 Alternative target

To develop the techniques further, an alternative to the averaged signal as the target would be potentially useful. The alternative should take into account nonstationary nature of the evoked potential signals or by its nature be relatively independent of the position of the signal features.

10.2.2 Wavelet Techniques

An extension to the wavelet techniques is to use a combination of different wavelets. Wavelet packets are an extension of the pyramidal decomposition method (Mallat et al, 1989) where instead of decomposing the approximated signal into details and approximations, the detail signals are also decomposed. This means that signal can be reconstructed by combining these decomposed details and approximations and permits further control by including or excluding certain details and approximations. Coifman and Wickerhauser (1992), used wavelet packets to model evoked potentials. One possible direction for investigation is to use the methods in chapter 8 but replace the decomposition of wavelets by a wavelet packet. Matching pursuit (Mallat and Zhang, 1993) is a similar technique to wavelet packets but a dictionary of possible waveforms is used. Akay and Daubenspeck (1999) used matching pursuit to separate bioelectric noise sources and EEG activity under certain conditions and may be worth investigating to separate evoked potentials from noise sources. Wavelet Networks (Zhang and Benveniste 1992) are a combination of wavelets and artificial neural networks. These have recently been applied to a closely related set of signals, event-related potentials, for analysis (Heinrich et al., 1999).

10.2.3 Modification to the evolutionary algorithm approaches

Evolutionary algorithm methods could be modified to extract the small high frequency components in the early latency region. The system would preprocess the signals (and the target signal) with a high-pass filter, removing the often larger low frequency components that dominate the signal. Maccabee et al (1986) used a high-pass filter cut-off frequency set at around 150 to 300 Hz in a similar way to filter averages of large number of signals. This would mean losing the lower frequency components, but increases the potential to extract the smaller high-rate components.

Alternative strategies are possible for mutation and selection, such as increasing the mutation rate when the population converges to an early solution, so other areas of the search space can be searched. These may avoid the searching becoming too localised in the search space, missing possible optimal solutions.

10.2.4 Noise modelling

Instead of treating the background activity as noise, modeling it to see if realistic models can be produced and then use these as part of the process to remove the noise from the recorded signals. A model that varies with time is likely to be effective, due to the time-varying nature of the noise (as shown by the spectrograms in chapter 4).

10.3 Scope for future development with alternative data sets

All the method discussed in 10.2 were for the historical data sets used here and could be used for other signals recorded in a similar method. Further techniques as well as those above are available if more channels of data were collected.

Independent component analyses (ICA) are a range of techniques that take signals from several recording sources and extract a set of signals. Makeig et al (1997) applied this technique to represent the sources generating the signals on the scalp due to the Auditory Event Related responses. This technique was not available to this research, as only two or three channels (including one used as a stimulus marker) were available.

A Background Experimental Work.

A.1 Choice of fitness function

A measure of how well a particular sequence in a population performs is needed. The fitness function initially considered here was the correlation coefficient between the results of the evolutionary algorithm and a known target response.

$$r = \frac{\frac{1}{n-1}\sum_{k=1}^{n}(x(k) - x_{mean})(y(k) - y_{mean})}{\sigma_x \sigma_y} \qquad \textbf{Equation A-1}$$

The above equation shows the correlation coefficient where x (k) is the kth value in the resulting response after using the operations encode by an individual sequence and y (k) was the kth value of the target response. The standard deviations of the two signals σ_x and σ_y were included. This is the Pearson's Correlation coefficient and the values of r range from -1 for an exact negative correlation, to +1 for an exact positive correlation, with zero being no correlation between the two signals.

$$mse = \frac{1}{N}\sum_{k=1}^{N} e(k)^2 \qquad \textbf{Equation A-2}$$

The above equation shows the Mean Squared Error, where the difference between the target signal and the test sequence (e (k)) at a point in time in squared and the results is the mean value of these values.

Table A-1 shows a comparison of the two fitness functions for the simulated data set and data set 1, with the functions being used to compare the two methods. Both produce similar mean correlation coefficients, which are higher than those for unfiltered signals, but different MSE values. When MSE was used as a fitness function, their MSE values were lower than those of correlation coefficient are fitness functions. The difference is due to the correlation coefficient being independent of scale, where MSE is not.

Data set	Fitness Function	Frequency (Hz)		Weighting	Measurements	
		F_L	F_H		MSE	Correlation Coefficient
Simulated	Unfiltered	-	-	-	0.8313	0.0018
	MSE	4.022	26.9667	0.9307	0.9423	0.0006
		66.751	576.816	0.2848		
		133.398	637.093	0.0563	0.9422	0.0031
	Correlation Coefficient	1.11	53.05	0.678		
		4.75	22.56	1.348		
		105.46	367.72	0.35		
1	Unfiltered	-	-	-	0.5405	0.0099
	MSE	3.7097	18.4259	0.6253	0.7727	0.0017
		52.787	55.6489	0.3097		
		42.5453	260.1425	0.0864		
	Correlation Coefficient	4.41	23.77	0.623	0.7734	0.003
		4.06	19.07	0.211		
		55.61	198.11	0.069		

Table A-1Preliminary filtered and unfiltered results for the whole signals for all test subsets comparing filters developed using MSE and correlation coefficient as fitness functions

The unfiltered subaveraged signals for the two data sets are shown in figure A-1. The filtered results of the two fitness functions are shown in figures A-2 and A-3. There is little difference between the results of the two methods in terms of morphology, but the magnitudes are different

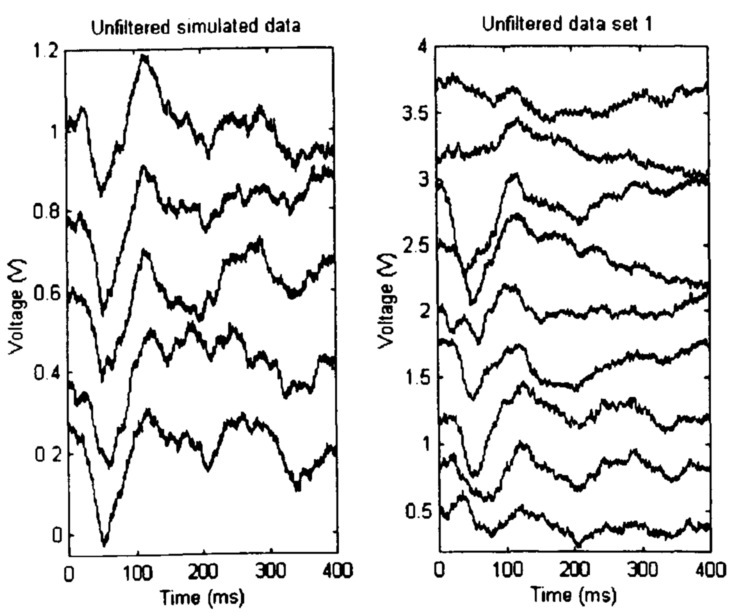

Figure A-1 unfiltered simualted and data set 1

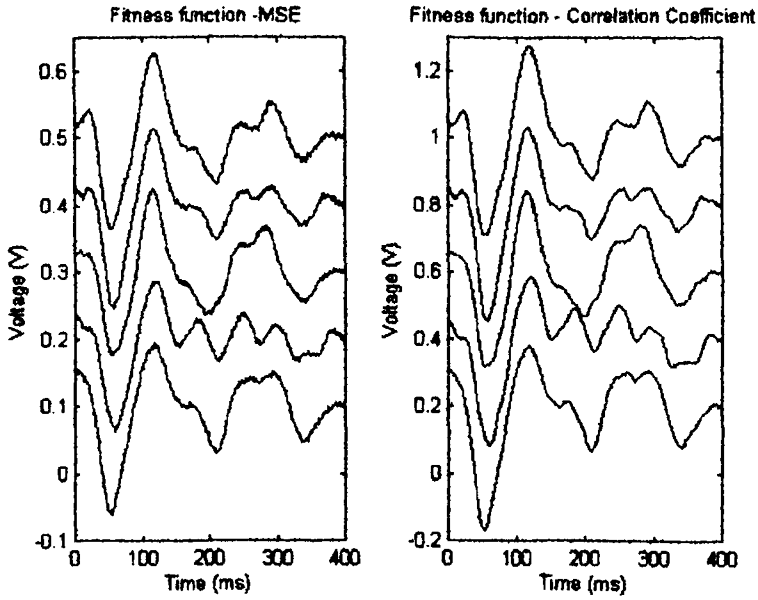

Figure A-2 simulated data sets results using the two fitness functions

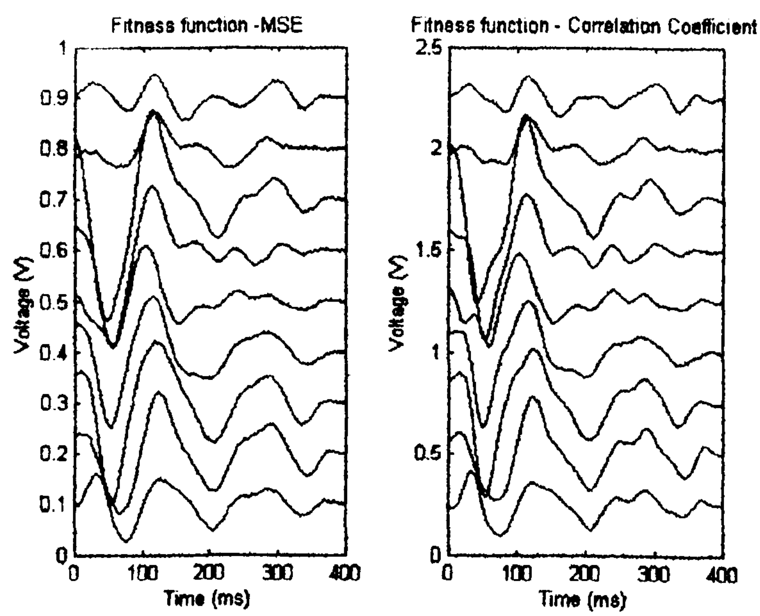

Figure A-3 Data set 1 results for the two fitness functions

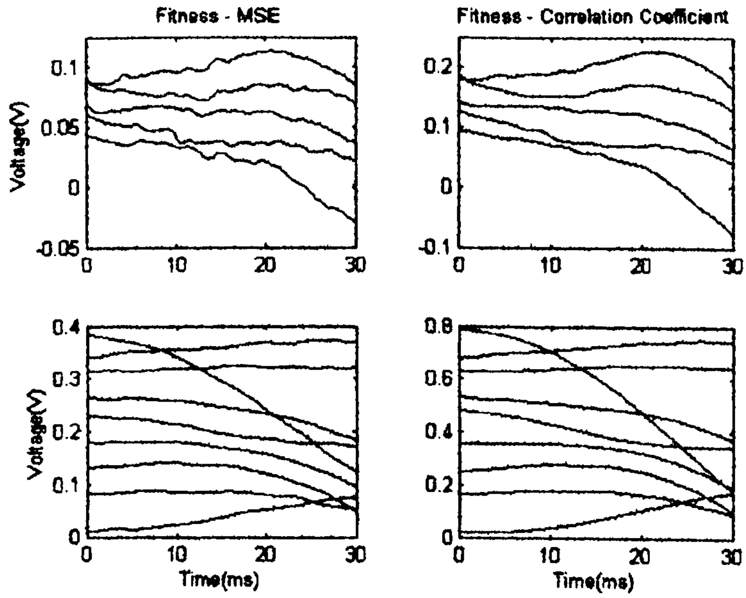

Figure A-4 Comparing the first 30 ms of the two data sets for the two fitness functions

The only other difference between the two methods is that the MSE based results produced slightly noisier results. Looking at the early (first 30 ms) components of these signals show in the simulated data set that the MSE based results (figure A-4) produced results that which are closer to the target. Than those of the correlation coefficient based results. Due to the results of MSE as the fitness functions to produce good correlation coefficient results as well as MSE values, this was selected as the fitness function.

A.2 Power spectra of data sets

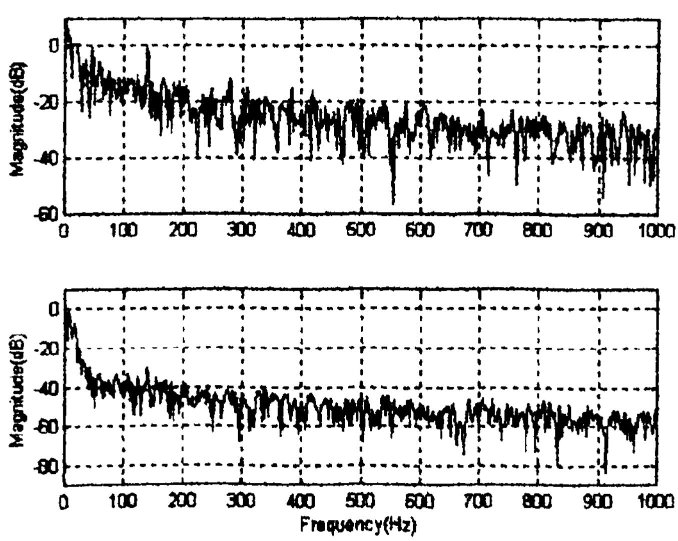

Figure A-5 20th of the test subset of recorded signal (data set 1), (b) target

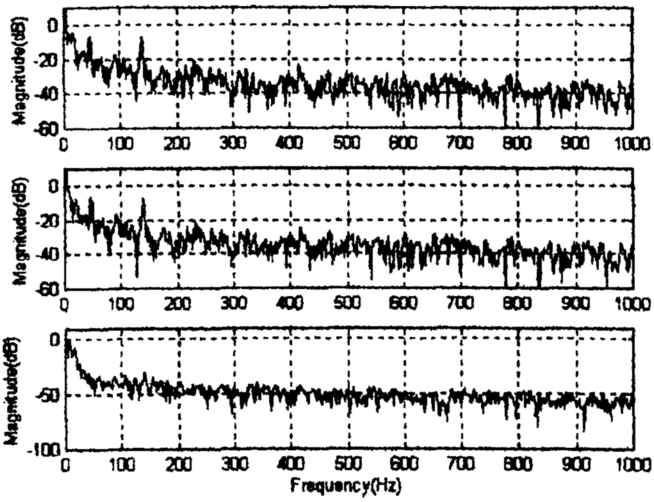

Figure A-6 simulated data (a) 20th signal, (b) noise, (c) target

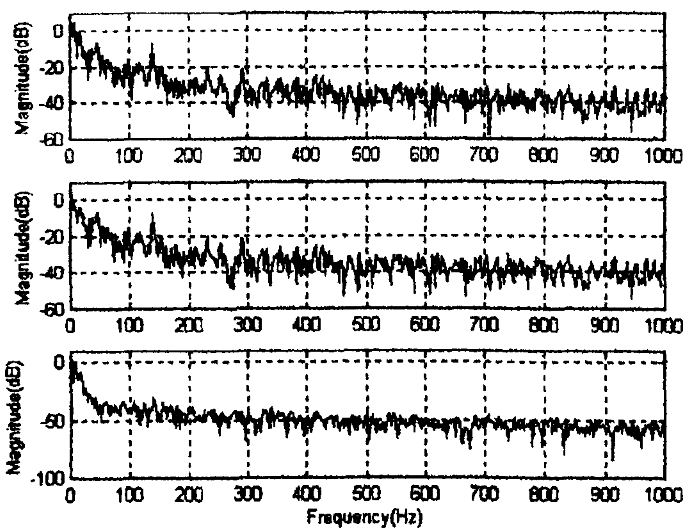

Figure A-7 simulated (a) 40th signal (b) noise (c) target

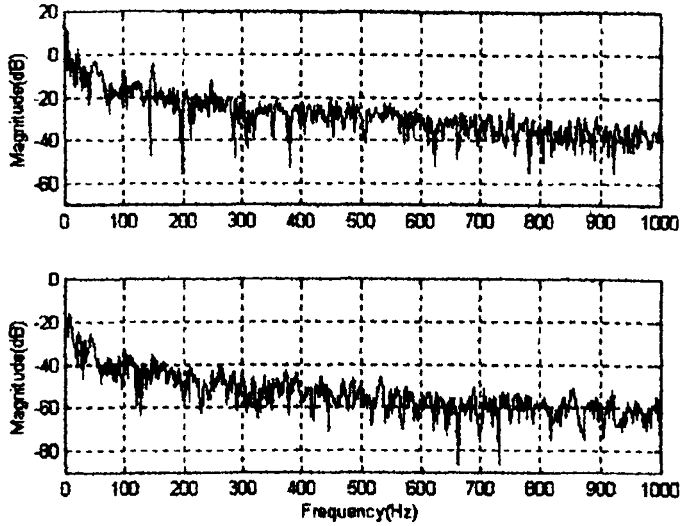

Figure A-8 Example of data set 2 signal (b) the target signal for data set 2

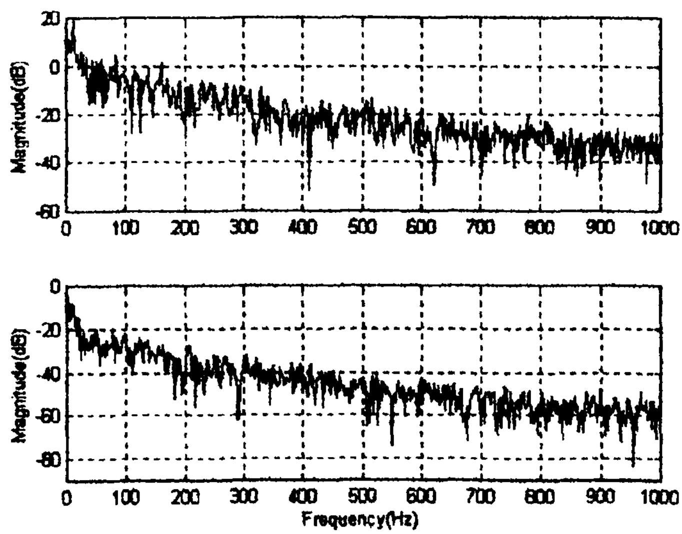

Figure A-9 an example of data set 3 signal (b) the target signal for data set 3

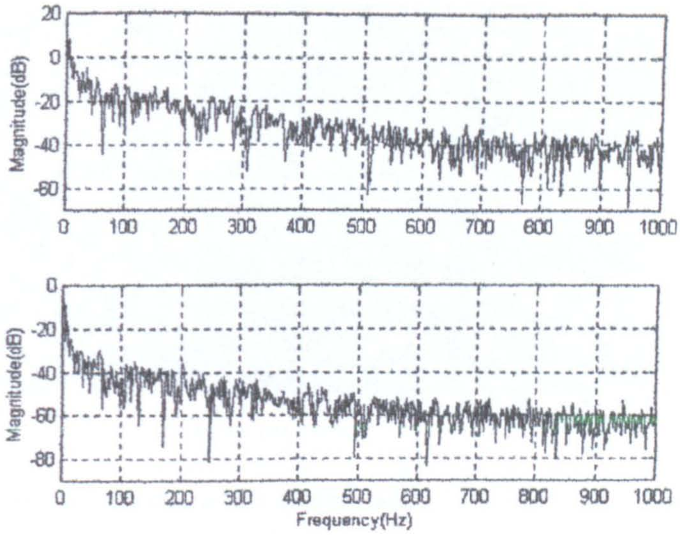

Figure A-10 an example of data set 4 signal (b) the target signal for data set

A.3 Continuous Wavelet Analysis of Evoked Potential

As has already been discussed, wavelets can be used to analyse how the frequencies (or components related to frequency) vary with time. The examples below were processed using the symlet-4 wavelet because of its approximate similarity to an action potential.

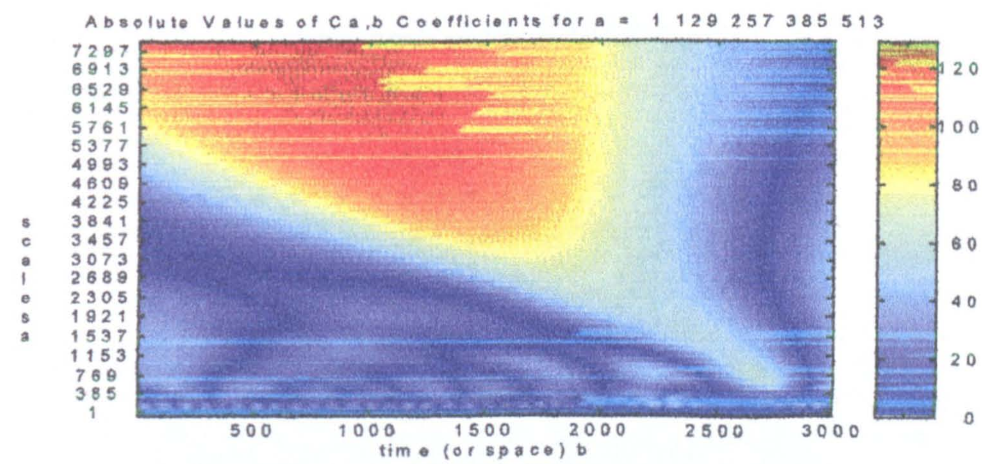

Figure A-11 Continuous Wavelet Analysis of a portion of background activity

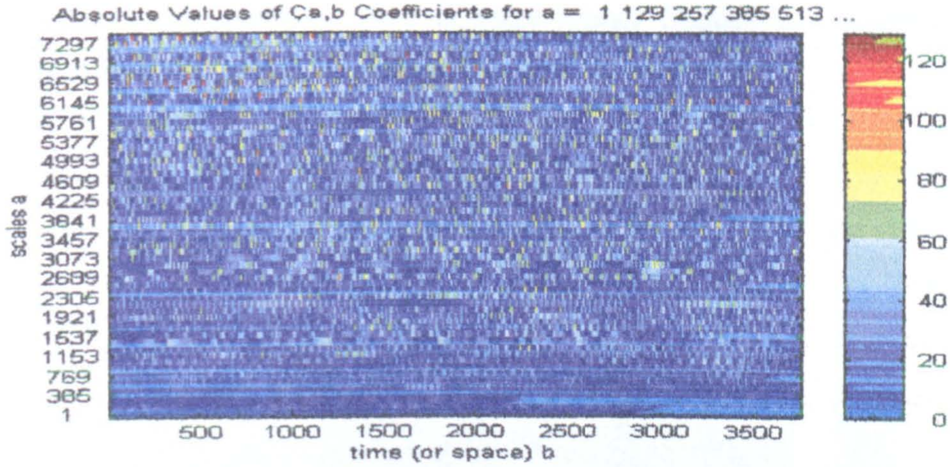

Figure A-12continuous wavelet analysis of white noise, used as a comparison

In figures A-11, a sample of background activity has been processed using a continuous wavelet, showing the distribution of frequency components with time. The vertical axis is scale, the higher the frequency component the lower its value. What can be seen here is that the signal is dominated by low frequency. As a comparison, in figure A-12 a random signal (white noise) was used and scalogram was produced. What is clear here is that there is no structure within the signal. This is to be expected, as a high scale value is as likely as a low scale value, if the signal was randomly produced.

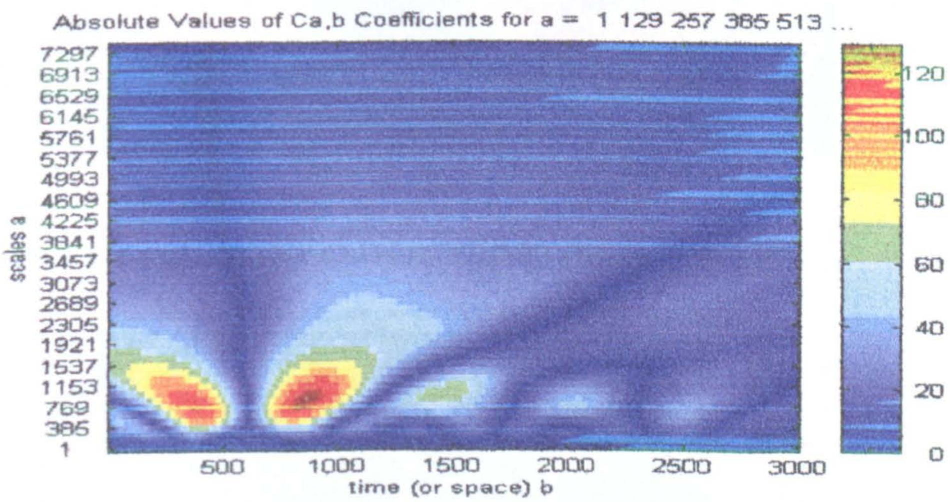

Figure A-13 continuous wavelet analysis of an averaged spinal recording of an evoked potential

Figure A-13 shows the results of an average of 222 evoked potentials, processed using wavelet analysis. In contrast to figure A-11, where the very low frequency components are dominant, there is some higher frequency components that dominate at different points in the signal.

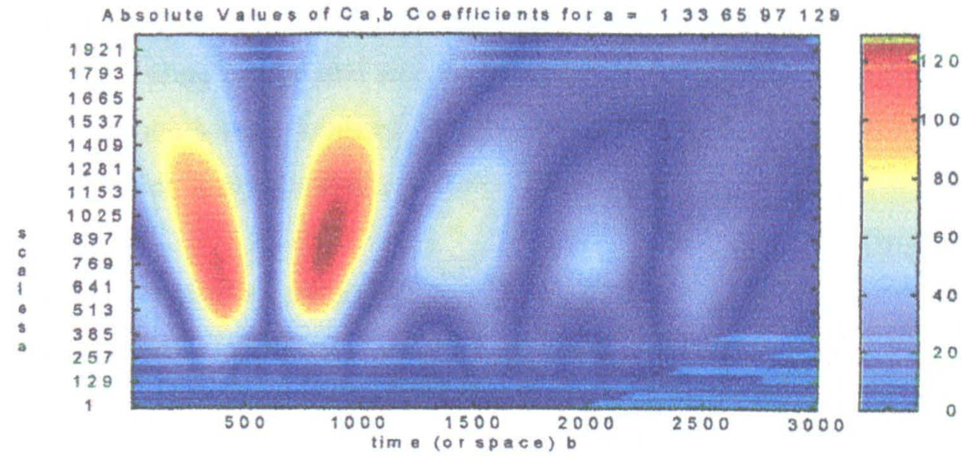

Figure A-14 Looking at the high frequency components of the wavelet analysis of the averaged signal

Figure A-14 shows the low scale values of the averaged signal and shows one of the properties of wavelet analysis, the ability to look at low scale (high frequency) components in time, to localise the analysis.

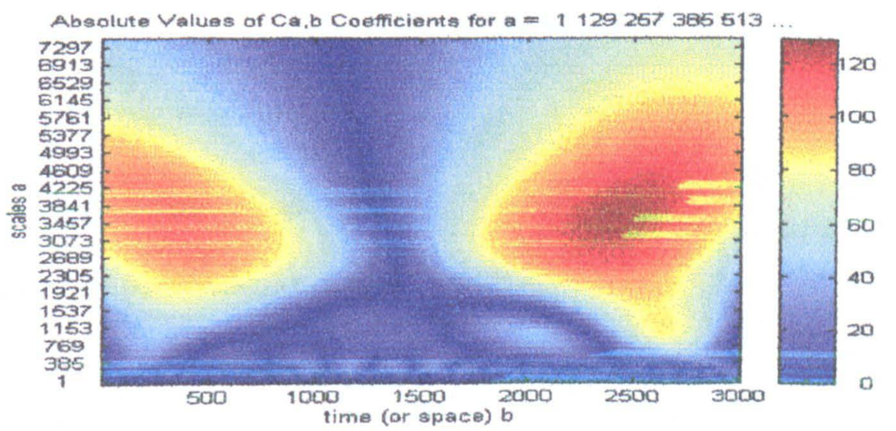

Figure A-15 continuous wavelet analysis of a single spinal recording of an evoked potential

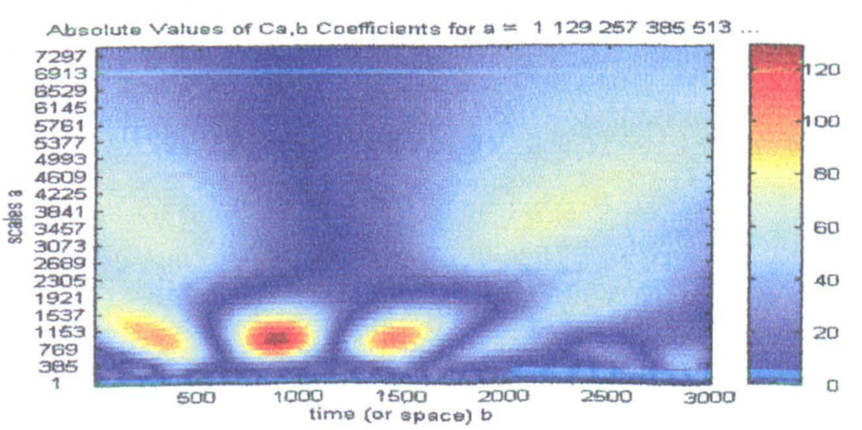

Figure A-16 continuous wavelet analysis of a single spinal recording of an evoked potential

The properties of the signals vary with time, as has been seen in the previous igures, but there are also changes between different samples of the signal, even within the same person and site. In figures A-15 and A-16, the differences are shown. Figure 8-15 is dominated by a low frequency signal, whereas in figure A-16 higher frequency components than in figure A-15 are dominant.

Continuous wavelets are useful for showing the signal as nonstationary, but is computationally intensive.

A.4 Wavelets

Included here are the scaling function (produces the approximations) and the wavelet functions (produces the details) select for by the evolutionary algorithms in chapter 8.

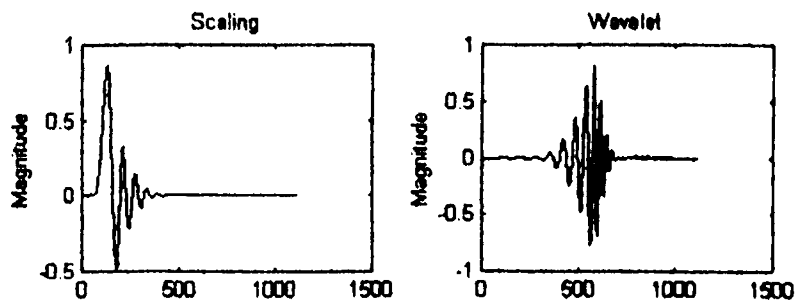

Figure A-17 Daubechies-18 first 30 ms of simulated

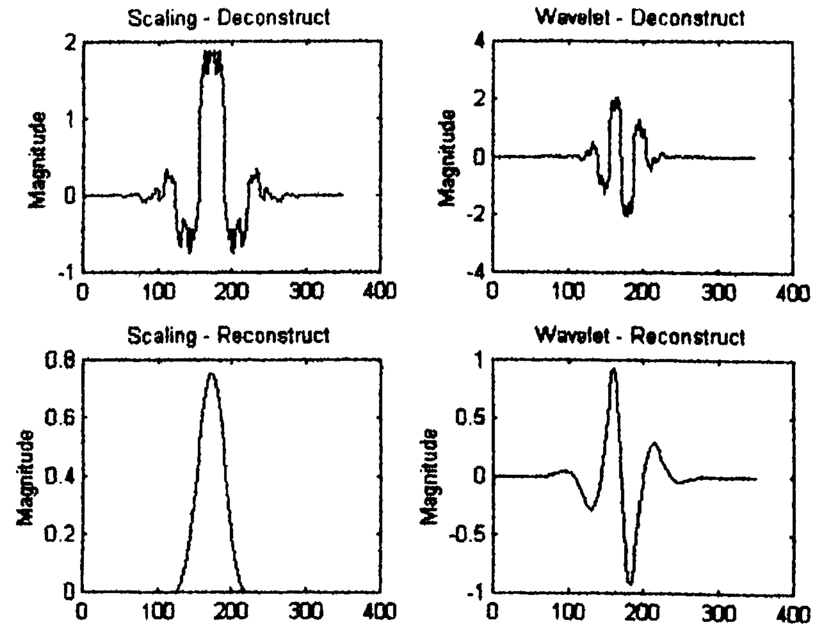

Figure A-18 bior3.5 30-400 ms of simulated data

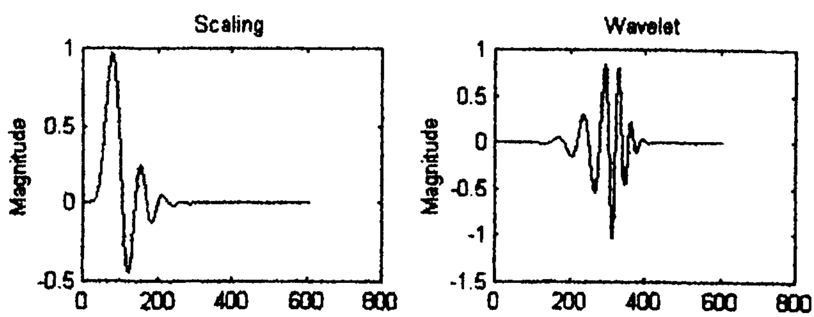

Figure A-19 Daubechies-10 for first 30 ms of data set 1

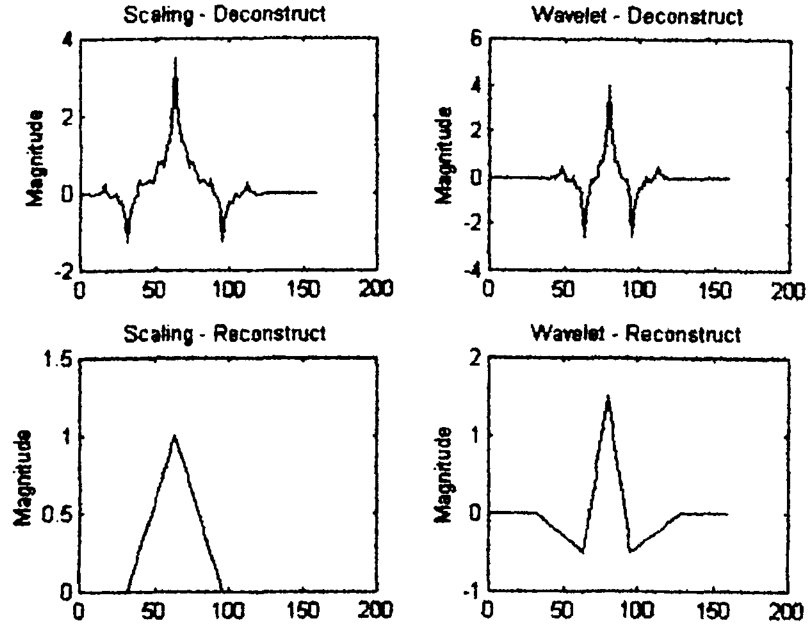

Figure A-20bior2.2 for 3-400 ms of data set 1 and for first 30 ms of data set 2 and 3.

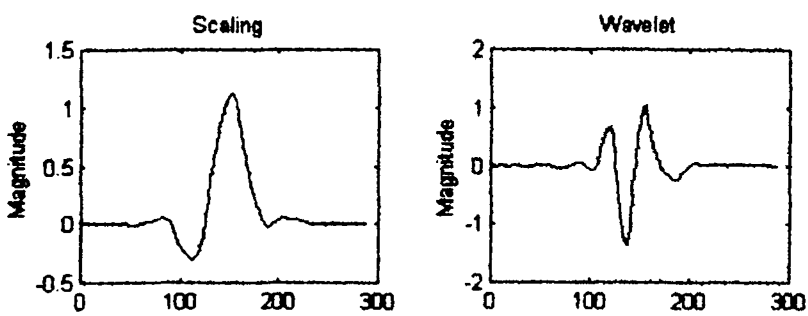

Figure A-21 sym-5 for 30-400 ms of data set 2

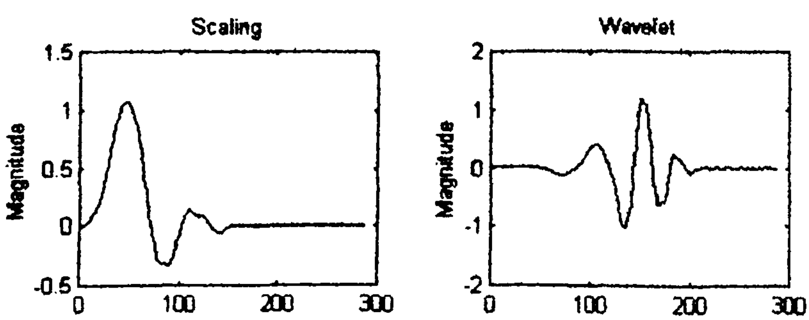

Figure A-22 db5 30-400 ms of data set 3

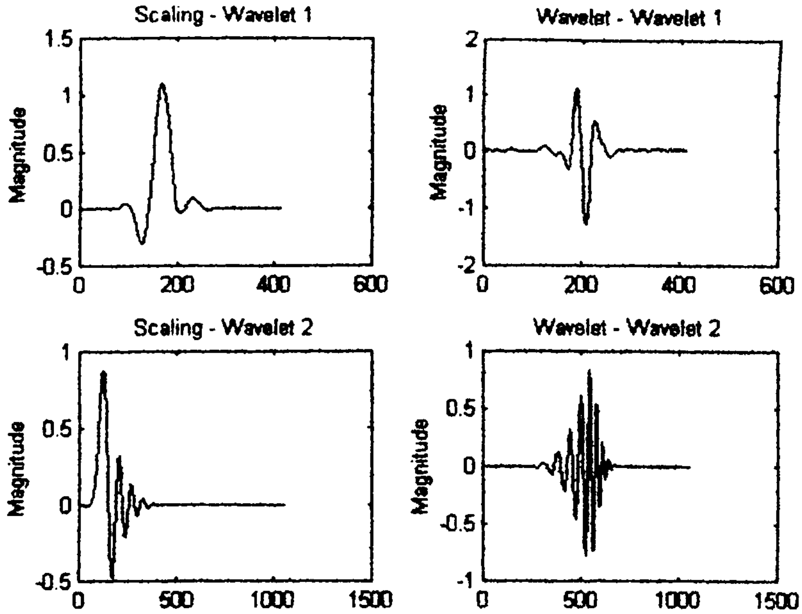

Figure A-23 wavelet 1 –sym7 for first 30 ms of data set 4 and wavelet 2 – db17 for 30-400 ms of data set 4

An extension of the previous approach was investigated. As well as selecting the weightings for the levels and the wavelet as before, the parameters used for de-noising: thresholding method, whether hard or soft thresholding is to used and scaling (no extra scaling, scaling based on one level, scaling each level separately) were selected. The results of applying this approach to the simulated data were found to produce distorted signals, losing more of the features than in the previous techniques. This extended approach was not developed further.

A.5 Effect of the filters on random and biological noise

The following figures (figures A-24 to A-26 for background activity and A-28 to A-30 for randomly produced signal) show the effect of passing subaveraged noise through the time-varying splitting the signals method, time-invariant filter bank applied to the two separate regions and the wavelet method. As a comparison the unfiltered signals are shown in figures A-27 and A-31.

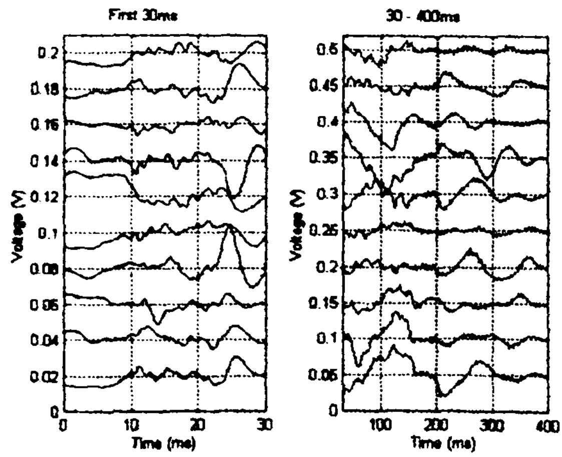

Figure A-24 Background activity passed through a filtered developed using splitting the signal.

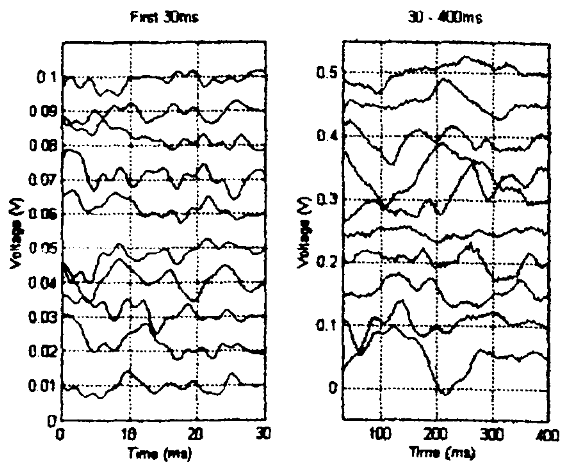

Figure A-25 Background activity passed through a filtered developed using separate signals.

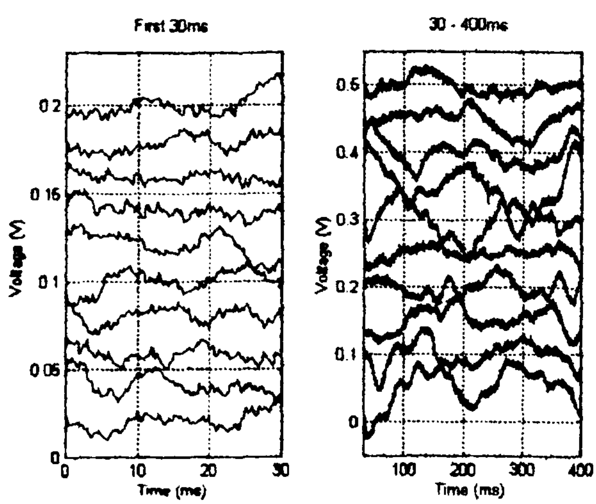

Figure A-26 Background activity passed through a filtered developed using the wavelet method.

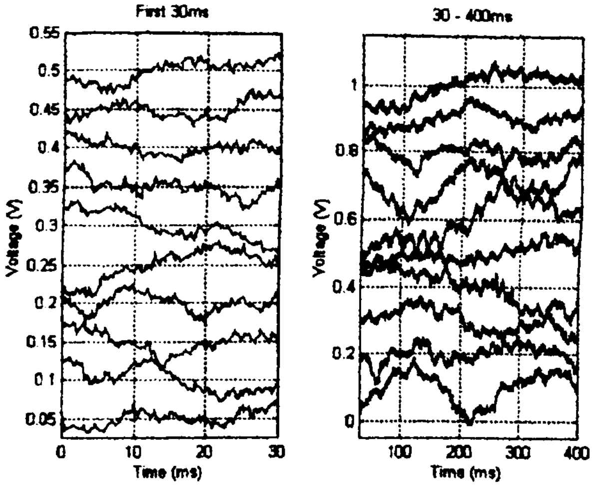

Figure A-27 Unfiltered background activity

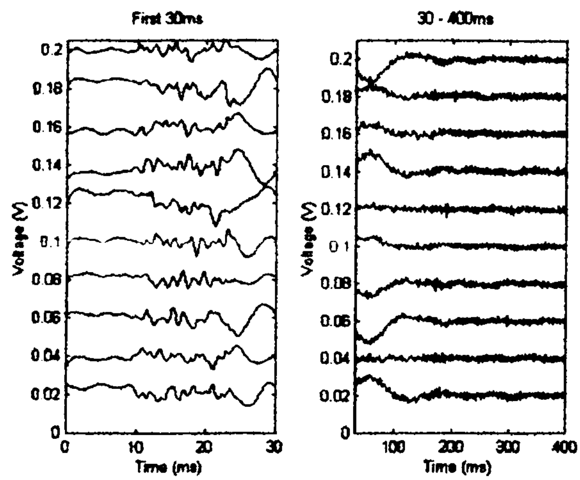

Figure A-28 Random noise passed through a filtered developed using splitting the signal.

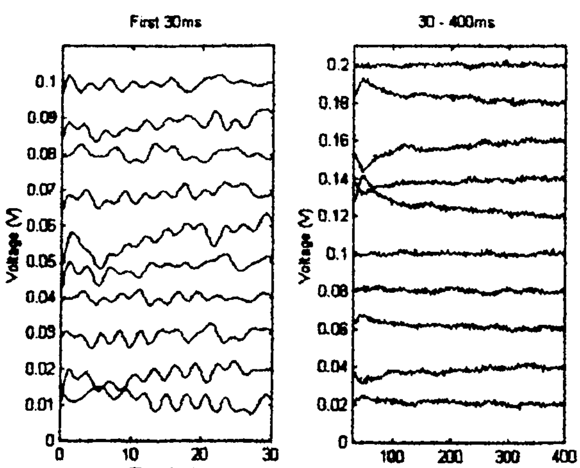

Figure A-29 Random noise passed through a filtered developed using separate signals.

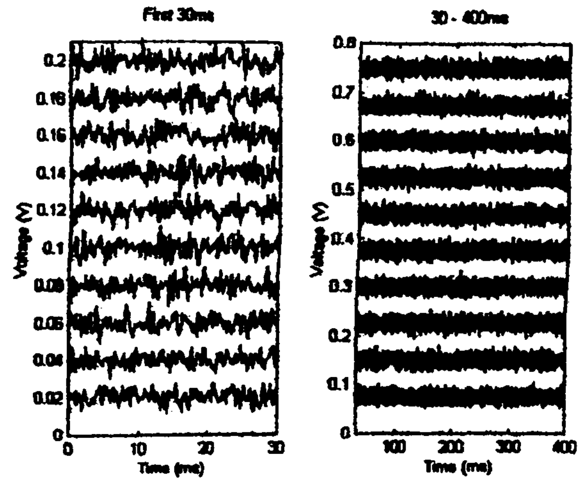

Figure A-30 Random noise passed through a filtered developed using wavelet method

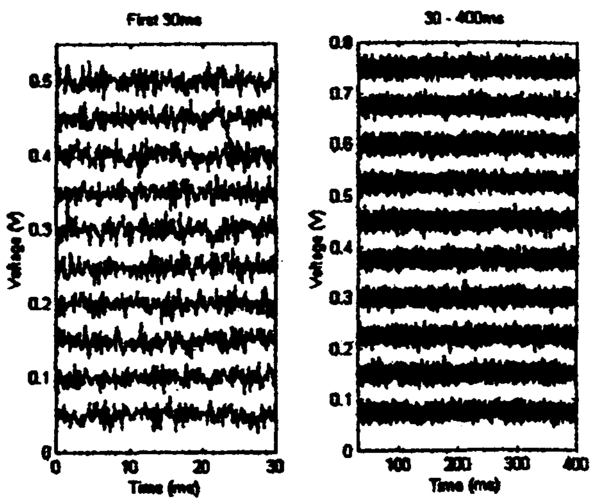

Figure A-31 unfiltered random signals (averaged)

These methods did not produce replica of evoked potentials, this was especially true of the wavelet method filtering subaveraged random noise, which produce signals that bear no resemblance to evoked potentials.

	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet
	Change %	Change %	Change %	Change %	Change %
simulated	26.5306	50.0000	30.6122	76.5306	37.7551
1	95.6186	95.1119	93.4629	95.3475	94.9941
2	77.1930	32.7485	59.0643	64.9123	72.5146
3	93.2163	91.3009	93.1346	91.0615	92.7374
4	23.2319	21.8957	-8.3917	62.2313	29.9864
mean	63.1981	58.2114	53.7365	78.0166	65.5975

Table A-2 For the first 30 ms the percentage decrease in mean MSE of the techniques compared to the values for average alone, for subaverages of 20 signals using the equation (1-(techniques mean MSE - averaging-alone mean MSE)/ /averaging-alone mean MSE)

	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet
	Change %	Change %	Change %	Change %	Change %
simulated	81.3590	57.2459	65.8449	60.9140	67.3482
1	80.8749	51.6446	68.8708	65.2764	67.1075
2	90.5923	87.8049	86.7596	88.5017	88.1533
3	79.7530	77.1751	77.4973	79.0548	76.3963
4	57.5397	55.5525	51.6096	67.1724	68.6915
mean	78.0238	65.8846	70.1164	72.1839	73.5394

Table A-3 For the 30 – 400 ms the percentage decrease in mean MSE of the techniques compared to the values for average alone, for subaverages of 20 signals using the equation (1-(techniques mean MSE - averaging-alone mean MSE)/averaging-alone mean MSE)

		First 30 ms – Mean MSE (10^{-2})						30-400 ms – Mean MSE (10^{-2})					
Size of average	Data set	Averaging -alone	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet	Averaging -alone	Splitting the signals	Optimal filter	Partial	Wavelet	Optimal-wavelet
10	simulated	0.0250	0.0086	0.0058	0.0076	0.0059	0.0067	0.1820	0.0298	0.1393	0.0608	0.0748	0.1236
	1	0.3659	0.0087	0.0198	0.0114	0.0098	0.0129	0.9784	0.1256	0.3657	0.1671	0.1858	0.2632
	2	0.0367	0.0048	0.0248	0.0086	0.0073	0.0090	0.1341	0.0067	0.0113	0.0099	0.0075	0.0123
	3	0.3155	0.0091	0.0096	0.008	0.0123	0.0113	0.7665	0.0925	0.1544	0.0971	0.1031	0.1404
	4	0.0159	0.0057	0.0117	0.0068	0.0140	0.0110	0.0564	0.0144	0.0268	0.0158	0.0159	0.0190
20	simulated	0.0098	0.0072	0.0049	0.0068	0.0023	0.0061	0.1663	0.0310	0.0711	0.0568	0.0650	0.0543
	1	0.1698	0.0071	0.0083	0.0111	0.0079	0.0085	0.2949	0.0564	0.1426	0.0918	0.1024	0.0970
	2	0.0171	0.0039	0.0115	0.0070	0.0060	0.0047	0.0574	0.0054	0.007	0.0076	0.0066	0.0068
	3	0.1253	0.0085	0.0109	0.0076	0.0112	0.0091	0.3724	0.0754	0.085	0.0838	0.0780	0.0879
	4	0.0054	0.0041	0.0042	0.0058	0.0140	0.0038	0.0307	0.0130	0.0136	0.0148	0.0145	0.0096

Table A-4 Comparison of time varying techniques when 10 and 20 signals per subaveraging was used

B Background Theory

B.1 Signal Modelling

Methods that model evoked potentials have been investigated. The basics of signal modelling are considered.

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k) \qquad \textbf{Equation B-1}$$

In the equation y (n) is the current output (or an estimation of the output value) of a linear filter and x (n) is the input to the filter. In the above equation, the output is predicted from past output values, and past and present inputs. The input is a combination of the desired signal d (n) and noise v (n), such that x (n)=d (n)+v (n),

$$y(n) + \sum_{k=1}^{p} a(k)y(n-k) = \sum_{k=0}^{q} b(k)x(n-k) \qquad \textbf{Equation B-2}$$

Where ideally d (n)=y (n).

The coefficients a (k) and b (k) are called the filter coefficients. Separating all the components of y (n) and x (n).

Taking the z-transform of the above equation

$$Y(z)(1 + \sum_{k=1}^{p} a(k)z^{-k}) = X(z)\sum_{k=0}^{q} b(k)z^{-k} \qquad \textbf{Equation B-3}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{q} b(k)z^{-k}}{1 + \sum_{k=1}^{q} a(k)z^{-k}} \qquad \textbf{Equation B-4}$$

H (z) is the pole-zero representation of the transfer function of the predictive filter.

There are three forms of models:

(1) If a (0)=1 and a (k)=0 for k>0, then H (z) is an all-zero model known as the Moving Average (MA) process.

$$H(z) = Y(z) = \sum_{k=0}^{q} b(k) z^{-k}$$ **Equation B-5**

$$y(n) = \sum_{k=0}^{q} b(k)x(n-k)$$ **Equation B-6**

The output is predicted only by the past and present inputs and is often used in functions to smooth the signal.

(2) If b (0)=1 and b (k)=0 for k>0, the H (z) only contain poles, this is know as an autoregressive (AR) process.

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a(k) z^{-k}} = Y(z)$$ **Equation B-7**

$$y(n) = \sum_{k=1}^{q} a(k)y(n-k)$$ **Equation B-8**

The output is predicted from past output values. The AR process is widely used. In some applications the signal generated will result in an all-pole model or filter, but in others it has been found that AR produce sufficiently accurate representation of many types of signal. The problem with the AR modelling are that the output values of the filter need to be known before hand, or the signal to noise ratio needs to be sufficiently large enough that a good estimate of the signal can be used.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{q} b(k) z^{-k}}{1 + \sum_{k=1}^{q} a(k) z^{-k}}$$ **Equation B-9**

(3) The final case is where both poles and zeros are used, this is called an Autoregressive Moving Average (ARMA) process

This has the properties of both the moving average and autoregressive processes, leading to methods that can predict output values based on previous and present values of inputs and previous output values. The point of these three cases is to predict what the value of the output signal is going to be, by modelling the signal.

B.2 Averaging

Ensemble averaging can be considered as a moving average (Challis and Kitney, 1990).

$$H(z) = \frac{1 + z^{-k} + z^{-2k} + ... + z^{-(N-1)k}}{N} = \frac{1}{N}\frac{1 - z^{-Nk}}{1 - z^{-1}} \qquad \textbf{Equation B-10}$$

If there are N responses each of k samples long, then the z-transform of the filtering operation is shown in equation (2) and this is a moving average low-pass filter.

B.3 Relating Mean-Squared Error (MSE) and Signal-to-Noise Ratio (SNR)

The assumption has been that the filtered signal (filtered) is the target signal plus noise and as the signal and target are voltage so the ratio of the signal (target) and noise power is: -

$$SNR = \frac{\sum t\arg et^2}{\sum (filtered - t\arg et)^2} \qquad \textbf{Equation B-11}$$

and

$$MSE = \frac{1}{N}\sum (filtered - t\arg et)^2 \qquad \textbf{Equation B-12}$$

$$SNR = \frac{\sum t\arg et^2}{N.MSE} \qquad \textbf{Equation B-13}$$

As N, the number of samples in the filtered signal and the power of the target signal are constants for the data set used, then

$$SNR \propto \frac{1}{MSE}$$

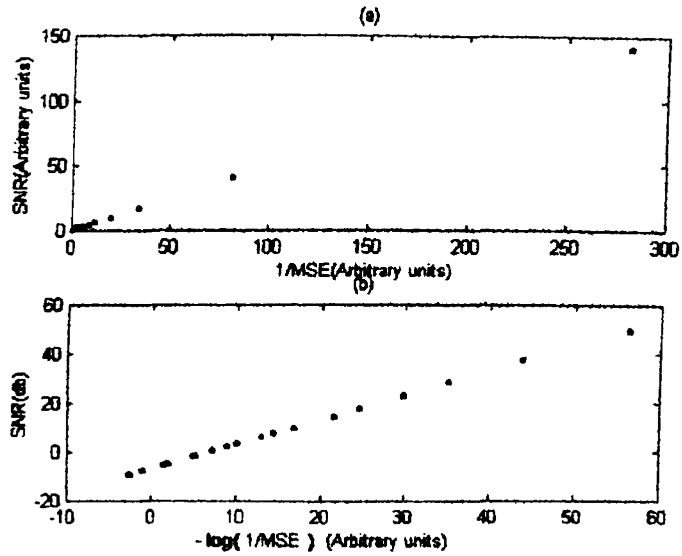

Figure B-1 (a) Relating SNR to MSE, (b) Relating SNR in decibels to –log (MSE)

C Selection of MATLAB programs

C.1 Time invariant filter bank (Chapter 6)

```
function [temprx,rrx]=gamselx(dat,av,rp,sz2,mu,R)

fs=3750; %Half the sampling frequency in Hz
sz1=max(size(dat));
sz3=min(size(dat));
sz4=3;
%sz1 - size of the sequence
%sz2 - size of the initial population defined as input argument
sz2
%sz3 - the number of examples
%sz4 - The number of parameters

%Selecting the filter cut-off frequencies
f(:,1)=100.*rand([sz2,1]);     f(:,2)=300.*rand([sz2,1])+f(:,1);
%Selecting weights
w=rand([sz2,1]);

%Removing mean values from the target signal
av=detrend(av,0);

%Producing the subaverages
rnddl=floor(sz3/R);
lxx=1;rrx=0;
for loopl=0:(rnddl-1)

avv(:,loopl+1)=detrend(mean(dat(:,loopl.*R+1:(loopl+1).*R)')',0);
end;

%A set of target signal same size as the subaverages
for loop=1:rnddl,
   avvx(:,loop)=av.*ones([sz1,1]);
end;


%While the number of generations is less than the set value
%iterate
while((lxx<=rp))
   clear cmn;
   for ll=1:sz2,
      %Producing the filtered results
      [bl,al]=butter(2,[f(ll,1)/fs f(ll,2)/fs]);
      yy=filtfilt(bl,al,avv).*w(ll);
      %Calculating the MSE values and Mean MSE value for the
      %individual solution
      e=yy-avvx;
      cmnl=mean(e.*e);
      cmn(ll)=mean(cmnl);
   end;
r=[f w];

vn=[cmn' (1:sz2)'];

%Sorting the sequences in fitness function descending order
[s4,sl]=sort(vn(:,1));

%select individual to produce new generation
```

```
s4=1./s4;
ss4=round(s4./10)'+1;sum1=sum(abs(ss4));
clear selc
c=0;
for loop=1:sz2,
    selc(c+1:c+(ss4(loop)))=s1(loop).*ones([ss4(loop),1])';
    c=c+ss4(loop);
end;
sz=max(size(selc));
check1=zeros([sz2,1]);
gx(1:sz2/4,:)=r(s1(1:sz2/4),:);
r1=floor(rand([3.*(sz2/8),2]).*(sz-ss4(loop)))+1;

for loop=1:3.*(sz2/8),
    r2=round((sz4-2).*rand)+1;
gx((sz2/4)+loop,:)=[r(selc(r1(loop,1)),1:r2),r(selc(r1(loop,2)),r
2+1:sz4)];

gx((sz2/4)+3.*(sz2/8)+loop,:)=[r(selc(r1(loop,2)),1:r2),r(selc(r1
(loop,1)),r2+1:sz4)];
 end;
pp(:,:)=abs(rand(sz2,sz4-1)); %Randomly generated probabilistic
values
for loop=2:sz2,
   for loop2=1:sz4-1,
      if (pp(loop,loop2)<=mu) %is pp<=mu
         r5=0.25.*(rand-0.5);
         gx(loop,loop2)=gx(loop,loop2)+r5.*gx(loop,loop2);
      end;
   end;
end;
temprr=r;
r=gx;f=r(:,1:2);w=r(:,3);
rrx(lxx)=1./s4(1);
lxx
[1./s4(1)]
temprr(s1(1),1:3)
lxx=lxx+1;
end;
temprx=temprr(s1(1),:);
```

C.2 Combined evolutionary algorithm and wavelets approach (Chapter 8)

```
function [finall,rrx]=gn5(inp,av,sz1,rp,mu,R,xz1,loopout)

[sz2,sz3]=size(inp);          %Calculates length of sequence sz2
and

%Select the level of the sequence to avoid unnecessary processing
Lev=11;
sz4=Lev+4;
szx=0;%Value one less than the first individual to be tested

%Averaging the input signals into rnndl signals
rnddl=floor(sz3/R);
lxx=1;rrx=0;
for loop1=0:(rnddl-1),
   dat1(:,loop1+1)=mean(inp(:,loop1.*R+1:(loop1+1).*R)')';
end;
dat11=detrend(dat1(xz1:225,:),0);av1=detrend(av(xz1:225),0);
dat12=detrend(dat1(225:sz2,:),0);av2=detrend(av(225:sz2),0);
```

```
%Weightings individual ranging from -1 to 1
GAx1(:,:,:)=5.*((rand(sz1,Lev+2,2))-0.5);
GAx2=round(45.*rand(sz1,1,2))+1;
GA=[GAx2 GAx1 zeros([sz1,1,2])];

%lxx is the generation number.
lxx=1;

%Main loop counting the number of generations from 1 to the value
rp
while (lxx<=rp),
   %loop to process szx+1 individuals in the population
   for loop2=1:sz1,
      [wn1]=selwavelet(GA(loop2,1,1));
      [wn2]=selwavelet(GA(loop2,1,2));
%wn1 and wn2 are the wavelets for the region <30 ms and 30-400 ms
%respectively
      [c,LL]=wavedec(av1,7,wn1);
      [cc,LLL]=wavedec(av2,11,wn2);
      nn(1)=0;nnn(1)=0;
      for loop1=2:9,
         nn(loop1)=nn(loop1-1)+LL(loop1-1);
      end;
      for loop11=2:13,
         nnn(loop11)=nnn(loop11-1)+LLL(loop11-1);
      end;

      for loop=1:rndd1,
         [c,LL]=wavedec(dat11(:,loop),7,wn1);
         [cc,LLL]=wavedec(dat12(:,loop),11,wn2);
         for loop1=2:9,
            cx((nn(loop1-1)+1):nn(loop1))=
            c((nn(loop1-1)+1):nn(loop1)).*GA(loop2,loop1,1);
         end;
         for loop11=2:13,
            ccx((nnn(loop11-1)+1):nnn(loop11))=
            cc((nnn(loop11-
1)+1):nnn(loop11)).*GA(loop2,loop11,2);
         end;
         %Reconstruct the signal by combining the GA sequence and
the wavelet coefficient
         xx=waverec(cx,LL,wn1);
         cc1=xx'-av1;
%error between the reconstructed signal and target signal
         cmnx1(loop)=mean(cc1.*cc1); %MSE
         xx1=waverec(ccx,LLL,wn2);
         ccc1=xx1'-av2;
%error between the reconstructed signal and target signal
         cmnx2(loop)=mean(ccc1.*ccc1); %MSE
      end;
      %mean MSE for the individual solution
      mn1=mean(cmnx1);
      cmn1x(loop2)=mn1;
      mn2=mean(cmnx2);
      cmn2x(loop2)=mn2;
   end;

   vn1=[cmn1x' (1:sz1)'];
   vn2=[cmn2x' (1:sz1)'];
   %Sorting the sequences in fitness function descending order
   [s41,s11]=sort(vn1);
   [s42,s12]=sort(vn2);
   s1=[1:sz1]';
```

```
    s4=(s41(:,1)+s42(:,1))./2;
    GA1(:,:,1)=GA(s11(:,1),:,1);
    GA1(:,:,2)=GA(s12(:,1),:,2);

    %Setting up the roulette wheel
    ss4=round(0.01./s4)'+1;
    sum1=sum(ss4);
    clear selc
    xc=0;
    for loop=1:sz1,
       selc(xc+1:(xc+ss4(loop)))=s1(loop).*ones([ss4(loop),1])';
       xc=xc+ss4(loop);
    end;

    sz=max(size(selc));
    clear gx;

    %Fittest quarter of the population go through to the next
 population
    gx(1:sz1/4,:,:)=GA1([1:sz1/4],:,:);

    %selecting where on the 'wheel' the sequence to crossover
 occur
    r1=floor(rand([3.*(sz1/8),4]).*(sz-ss4(loop)))+1;

    %Crossover the sequences
    for loop=1:3.*(sz1/8),
       r2=round((sz4-1).*rand)+1;
       gx((sz1/4)+loop,:,1)=
[GA1(selc(r1(loop,1)),1:r2,1),GA1(selc(r1(loop,2)),r2+1:sz4,1)];
       gx(5.*(sz1/8)+loop,:,1)=
[GA1(selc(r1(loop,2)),1:r2,1),GA1(selc(r1(loop,1)),r2+1:sz4,1)];
       gx((sz1/4)+loop,:,2)=
[GA1(selc(r1(loop,3)),1:r2,2),GA1(selc(r1(loop,4)),r2+1:sz4,2)];
       gx(5.*(sz1/8)+loop,:,2)=
[GA1(selc(r1(loop,4)),1:r2,2),GA1(selc(r1(loop,3)),r2+1:sz4,2)];
    end;

    %Mutation
    ppr(:,:,:)=abs(rand(sz1,sz4-1,2));
    %Randomly generated probablistic values
    for loop4=1:2
       if loop4==1,mm1=10;end;
       if loop2==2,mm1=14;end
       for loop=2:sz1,
          for loop2=1:mm1,
             if (ppr(loop,loop2,loop4)<=mu) %is pp<=mu
                if loop2>=2,
                gx(loop,loop2,loop4)=
                gx(loop,loop2,loop4).*(1+(0.25.*(rand-0.5)));
                end;
                if loop2==1,
                   gx(loop,loop2,loop4)=round(45.*rand)+1;
                end;
             end;
          end;
       end;
end;

%Temporary store for the current population
tempr=GA1;

%New population becomes current population
GA=gx;
```

```
    rrx(lxx)=s4(1);%Store the maximum fitness for population
    tempr(1,1,:)
    lxx=lxx+1;%increment generation counter
end;
%Final sequences
finall=[tempr(1,:,:)];
```

C.3 Splitting the signal

```
function [tempr,rrx,stemp]=nisind2d2(dat,av,sz2,rp,mu,R)

sz4=21;fs=3750;sxxz=0;
[sz1,sz3]=size(dat); %size of the sequence

%Selecting the filter cutoff frequencies
f(:,1,:)=50.*rand([sz2,1,2]);
    f(:,2,:)=50.*rand([sz2,1,2])+f(:,1,:);
f(:,3,:)=300.*rand([sz2,1,2]);
    f(:,4,:)=300.*rand([sz2,1,2])+f(:,3,:);
f(:,5,:)=50.*rand([sz2,1,2]);
    f(:,6,:)=50.*rand([sz2,1,2])+f(:,5,:);
f(:,7,:)=300.*rand([sz2,1,2]);
    f(:,8,:)=300.*rand([sz2,1,2])+f(:,7,:);
f(:,9,:)=50.*rand([sz2,1,2]);
f(:,10,:)=50.*rand([sz2,1,2])+f(:,9,:);
f(:,11,:)=300.*rand([sz2,1,2]);
f(:,12,:)=300.*rand([sz2,1,2])+f(:,11,:);

w1=190.*rand([sz2,2]);
w2=750*rand([sz2,2]);
w(:,:,1)=w1;w(:,:,2)=w2;
ww=rand([sz2,6,2]);
w=round(w);
rnddl=floor(sz3/R);
lxx=1;rrx=0;
   for loop1=0:(rnddl-1)
      dat1(:,loop1+1)=mean(dat(:,loop1.*R+1:(loop1+1).*R)')';
   end;
   av1=detrend(av(16:225),0);
   av2=detrend(av(226:3001),0);
  while((lxx<=rp)&(rrx<0.99))
   for ll=sxxz+1:sz2,
      clear tt1;clear tt2;clear yy;
      for loopc1=1:2
      for loopff=1:2
         for loopf=1:2:11,
         if f(ll,loopf,loopff)>=f(ll,loopf+1,loopff)
            tempf=f(ll,loopf+1,loopff);
            f(ll,loopf+1,loopff)=f(ll,loopf,loopff).*1.1;
            f(ll,loopf,loopff)=tempf.*0.9;
         end;
         if (f(ll,loopf,loopff)>1000),
            f(ll,loopf,loopff)=900;
         end;
         if (f(ll,loopf+1,loopff)>1000),
            f(ll,loopf+1,loopff)=900;
         end;
      end;end;end;
      [b1,a1]=butter(2,[f(ll,11,1)/fs f(ll,12,1)/fs]);
      [b2,a2]=butter(2,[f(ll,9,1)/fs f(ll,10,1)/fs]);
      [b3,a3]=butter(2,[f(ll,7,1)/fs f(ll,8,1)/fs]);
```

```
        [b4,a4]=butter(2,[f(l1,5,1)/fs f(l1,6,1)/fs]);
        [b5,a5]=butter(2,[f(l1,3,1)/fs f(l1,4,1)/fs]);
        [b6,a6]=butter(2,[f(l1,1,1)/fs f(l1,2,1)/fs]);
        for loopwx=1:2
        if ((w(l1,1,1)+17)>=w(l1,2,1))
           tempw=w(l1,2,1);
           w(l1,2,1)=w(l1,1,1)+32;
           w(l1,1,1)=tempw;
        end;
        if ((w(l1,1,1)<2)), w(l1,1,1)=2;end;
        if w(l1,2,1)>=170, w(l1,2,1)=16;end;
        end;
        t1=(1/15).*([w(l1,1,1):1:w(l1,1,1)+14]'-w(l1,1,1));
        t2=(1/15).*([w(l1,2,1):1:w(l1,2,1)+14]'-w(l1,2,1));
        y11(:,:)=filtfilt(b1,a1,dat1(16:225,:));

y21(:,:)=filtfilt(b2,a2,dat1(16:225,:));y1=ww(l1,1,1).*y11+ww(l1,
2,1).*y21;
        y31(:,:)=filtfilt(b3,a3,dat1(16:225,:));

y41(:,:)=filtfilt(b4,a4,dat1(16:225,:));y2=ww(l1,3,1).*y31+ww(l1,
4,1).*y41;
        y51(:,:)=filtfilt(b5,a5,dat1(16:225,:));

y61(:,:)=filtfilt(b6,a6,dat1(16:225,:));y3=ww(l1,5,1).*y51+ww(l1,
6,1).*y61;
        for lx2=1:rnddl,
           tt1=((1-
t1).*y1(w(l1,1,1)+1:(w(l1,1,1)+15),lx2))+(t1.*y2(w(l1,1,1)+1:(w(l
1,1,1)+15),lx2));
           tt2=((1-
t2).*y2(w(l1,2,1)+1:(w(l1,2,1)+15),lx2))+(t2.*y3(w(l1,2,1)+1:(w(l
1,2,1)+15),lx2));

yy=[y1(1:w(l1,1,1),lx2);tt1;y2(w(l1,1,1)+16:w(l1,2,1),lx2);tt2;y3
(w(l1,2,1)+16:225-15,lx2)];
           if max(size(yy))~=(225-15),yy=yy(1:225-15);end;
           e=av1-yy;cmnxz(lx2)=mean(e.*e);
        end;
        clear tt1;clear tt2;clear yy;
        cmnx(l1)=mean(cmnxz);
        [b11,a11]=butter(2,[f(l1,11,2)/fs f(l1,12,2)/fs]);
        [b21,a21]=butter(2,[f(l1,9,2)/fs f(l1,10,2)/fs]);
        [b31,a31]=butter(2,[f(l1,7,2)/fs f(l1,8,2)/fs]);
        [b41,a41]=butter(2,[f(l1,5,2)/fs f(l1,6,2)/fs]);
        [b51,a51]=butter(2,[f(l1,3,2)/fs f(l1,4,2)/fs]);
        [b61,a61]=butter(2,[f(l1,1,2)/fs f(l1,2,2)/fs]);
        for loopxw=1:2
        if ((w(l1,1,2)+17)>=w(l1,2,2))
           tempw=w(l1,2,2);
           w(l1,2,2)=w(l1,1,2)+32;
           w(l1,1,2)=tempw;
        end;
        if ((w(l1,1,2)<2)), w(l1,1,2)=2;end;
        if w(l1,2,2)>=0.5.*sz1, w(l1,2,2)=200;end;
        end;
        t1=(1/75).*([w(l1,1,2):1:w(l1,1,2)+74]'-w(l1,1,2));
        t2=(1/75).*([w(l1,2,2):1:w(l1,2,2)+74]'-w(l1,2,2));
        y111(:,:)=filtfilt(b11,a11,dat1(226:3001,:));

y211(:,:)=filtfilt(b21,a21,dat1(226:3001,:));yx1=ww(l1,1,2).*y111
+ww(l1,2,2).*y211;
        y311(:,:)=filtfilt(b31,a31,dat1(226:3001,:));
```

```
y411(:,:)=filtfilt(b41,a41,dat1(226:3001,:));yx2=ww(l1,3,2).*y311
+ww(l1,4,2).*y411;
        y511(:,:)=filtfilt(b51,a51,dat1(226:3001,:));

y611(:,:)=filtfilt(b61,a61,dat1(226:3001,:));yx3=ww(l1,5,2).*y511
+ww(l1,6,2).*y611;
        for lx2=1:rndd1
            tt1=((1-
t1).*yx1(w(l1,1,2)+1:(w(l1,1,2)+75),lx2))+(t1.*yx2(w(l1,1,2)+1:(w
(l1,1,2)+75),lx2));
            tt2=((1-
t2).*yx2(w(l1,2,2)+1:(w(l1,2,2)+75),lx2))+(t2.*yx3(w(l1,2,2)+1:(w
(l1,2,2)+75),lx2));

yy=[yx1(1:w(l1,1,2),lx2);tt1;yx2(w(l1,1,2)+76:w(l1,2,2),lx2);tt2;
yx3(w(l1,2,2)+76:(sz1-225),lx2)];
            if max(size(yy))~=3001-225,yy=yy(1:(3001-225));end;
            e=av2-yy;cmnxzz(lx2)=mean(e.*e);
        end;
        cmnxz1(l1)=mean(cmnxzz);
    end;
    if (lxx>1),
        cmnx(1:sxxz)=cmn(s1(1:sxxz),1);
    end;
    cmn(:,1)=cmnx';
    if (lxx>1),
        cmnxz1(1:sxxz)=cmn(s1(1:sxxz),2);
    end;
    cmn(:,2)=cmnxz1';

r=[f w ww zeros([sz2,1,2])];
vn1=[cmn(:,1)];
vn2=[cmn(:,2)];

%Sorting the sequences in fitness function descending order
[s41,s11]=sort(vn1);
[s42,s12]=sort(vn2);
%[s41 s42]
r(1:sz2,:,1)=r(s11(:,1),:,1);
r(1:sz2,:,2)=r(s12(:,1),:,2);

s4=(s41(:,1)+s42(:,1))./2;
s1=[1:sz2];
ssx4=1./s4(:,1);
for loop=1:sz2
    if ((ss4(loop)<10^10)&(ss4(loop)>1)),
        ss4(loop)=ss4(loop);
    else
        ss4(loop)=1;
    end
end;
%Call to the function to produce new generation
ss4=round(0.01.*ssx4)'+1;sum1=sum((ss4));
 clear selc
 c=0;
 for loop=1:sz2,
     selc(c+1:c+(ss4(loop)))=s1(loop).*ones([ss4(loop),1])';
     c=c+ss4(loop);
 end;
 sz=max(size(selc));
 clear gx;
 gx(1:sz2/4,:,:)=r(1:sz2/4,:,:);
 r1=floor(rand([3.*(sz2/8),2]).*(sz-ss4(loop)))+1;
```

```
 for loop=1:3.*(sz2/8),
     r2=round((sz4-2).*rand)+1;

gx((sz2/4)+loop,:,1)=[r(selc(r1(loop,1)),1:r2,1),r(selc(r1(loop,2
)),r2+1:sz4,1)];

gx((sz2/4)+3.*(sz2/8)+loop,:,1)=[r(selc(r1(loop,2)),1:r2,1),r(sel
c(r1(loop,1)),r2+1:sz4,1)];

gx((sz2/4)+loop,:,2)=[r(selc(r1(loop,1)),1:r2,2),r(selc(r1(loop,2
)),r2+1:sz4,2)];

gx((sz2/4)+3.*(sz2/8)+loop,:,2)=[r(selc(r1(loop,2)),1:r2,2),r(sel
c(r1(loop,1)),r2+1:sz4,2)];
 end;
 ppr(:,:,:)=abs(rand(sz2,sz4-1,2)); %Randomly generated
pobablistic values
 for loop4=1:2
for loop=sz2./4:sz2,
 for loop2=1:sz4-1,
  if (ppr(loop,loop2,loop4)<=mu) %is pp<=mu
     r5=0.25.*(rand-0.5);

gx(loop,loop2,loop4)=gx(loop,loop2,loop4)+r5.*gx(loop,loop2,loop4
);
     end;
   end;
end;end;

tempr=r;tempr(:,13:14,:)=round(tempr(:,13:14,:));

r=gx;f=abs(r(:,1:12,:));w=abs(round(r(:,13:14,:)));ww=r(:,15:20,:
);
rrx(lxx)=s4(1);
lxx,[s41(1) s42(1) s4(1)]
sxxz=0;
%tempr(s1(1),:)
stemp=tempr(s1(1),:);
 lxx=lxx+1;
end;
```

C.4 Wavelet Optimal Filter (Bertrand et al, 1996)

```
function [out1,dat_m,H]=dfilterw1(inp,L)

[sz2,sz3]=size(inp);
R=floor(sz3/L);
   for loop1=0:(R-1),
      dat_m(:,loop1+1)=mean(inp(:,loop1.*L+1:(loop1+1).*L)')';
   end;
mav=mean(inp')';
[c1,Len1] = wavedec(mav,11,'dmey');
%Decomposing the grand averaged signal into details and
%approximations
frq1=c1.*c1;
for loop=1:R
   [c2,len2] = wavedec(dat_m(:,loop),11,'dmey') ;
   %Decomposing each averaged signal into details and
   %approximations
   frq2(:,loop)=c2.*c2;
end;
```

```
mav2=mean(frq2')';
[szz1,szz2]=size(mav2);

%Avoiding dividing by zero errors by setting any components that
%equal zero to a small number several orders of magnitude smaller
%than the number likely as wavelets components
for loop=1:szz1
   if frq1(loop)==0,frq1(loop)=1e-14;end;
   if mav2(loop)==0,mav2(loop)=1e-14;end;
end;

%The transfer function
H=(R./(R-1)).*(1-((1/R).*(mav2./frq1)));

%Filtering the averaged signals
for loop=1:R
   [c3,len3] = wavedec(dat_m(:,loop),11,'dmey') ;
   out1(:,loop)= waverec(c3.*H,len3,'dmey');
end;
```

D Published Work

Turner SJ, Campbell JA, Picton PD (1997) Improved Signal to Noise Ratio in Somatosensory Evoked Potentials *3rd Annual National Conference of the Institute of Physics and Engineering in Medicine* 2-4 September, Dundee pp. 161.

Campbell JA, Turner S, Picton P (1999) Filter selection for evoked potentials using genetic algorithm techniques Eds. Moglia A, Zappoli F, Sirabella G *Atti del VII Meeting Invernale di Scienze Neurologiche* 8-12 February pp 15-25.

Turner SJ, Picton PD, Campbell JA (1999) Selecting Filter Banks to Enhance Evoked Potentials Recordings Using Evolutionary Algorithms Eds. Poli R, Voigt HM, Cagnoli S, Corne D, Smith GD, Fogary TC *Evolutionary Image Analysis, Signal; Processing and Telecommunications* May Lecture Notes Springer - Verlang 1596 pp 101-110.

Turner SJ, Picton PD, Campbell JA (2000) Use of Evolutionary Algorithms to Enhance the Extraction of Short Latency Evoked Potentials *4th Conference of the Systemics, Cybernetics and Informatics (SCI'2000)* Orlando, July 23-26 pp. 641-643.

Società Italiana di Neurochirurgia

Associazione Italiana di Neuroradiologia

Atti del VII Meeting Invernale di Scienze Neurologiche

Selva di Valgardena (Bz), 8-12 Febbraio 1999

a cura di
A. MOGLIA, F. ZAPPOLI, G. SIRABELLA

Filter selection for evoked potentials using genetic algorithm techniques

J A Campbell, S Turner, P Picton

University College Northampton UK

Introduction

The recording of potentials evoked by different parts of the human nervous system in response to specific stimuli can lead to vital information on neurophysiological mechanisms and function and are an important technique in neurological diagnosis and research. The main problem with these techniques however is the presence of electrical noise from other sources, which can mask or distort the underlying signal. This unwanted activity originates from sources such as other parts of the body (e g muscle, heart, other neural structures), recording equipment, or the local electrical environment (Harrison and Lovely 1995). Ensemble averaging is the most commonly used method of reducing the noise in evoked potential recordings. This technique calculates the algebraic sum of a large number of repeated, triggered responses. The main disadvantages of this method are that the signals need to be collected over a relatively long period of time, which may be undesirable or impractical, and that the signals may vary with time, leading to distortion of features within the signal.

This paper describes a method that aims to reduce the number of stimuli needed to produce an evoked response comparable to a 'good' averaged response and reports the results to date of this on-going work. The objectives of this project were to design a system that uses post-recording digital filters to reduce the noise components to such an extent that the number of responses needed to be averaged could be substantially reduced with no significant degradation of the final averaged response. The frequency spectrum of the noise component of evoked potentials overlaps with that of the signal spectrum and is different for each recording situation. It was not considered realistic therefore to apply a fixed set of filters to all recordings. An evolutionary approach to individual selection of the most appropriate filters for each set of data was therefore taken.

Methods

All the signals were recorded on FM tape, using a STORE 4 FM tape recorder (RACAL Recorders), using analogue filters with a bandpass of 0.016-750Hz. The evoked potentials used in this work were recorded intraspinally at C_{1-2} in response to stimulation of the median nerve at the wrist. The data was collected using a Gateway 2000 Pentium P90 computer via an interface card and data acquisition software (PC30F, Eagle Technology). All the filters and evolutionary algorithms were developed and implemented in MATLAB (MathWorks, USA).

Recorded data consisting of 222 responses were collected from the tape. A total of 38 responses were excluded from the experiments as they were found to contain artefacts such as 'clipping'. Using the remaining 184 recorded responses, two sets of data with 92 responses in each were formed into a test and a training set. These sets are referred to here, as the recorded data. An average of the 184 recorded responses was used to form a reference signal which was the target signal that the filters aimed to extract. Pre-stimulus recordings, i.e. electrical activity recorded just before stimulation occurred, were used as a source of background noise which was also added to the reference signal to create simulated recordings with a known signal and typical noise characteristics This simulated data (target signal+noise) set was split into a training set (55 responses) and a test set (56 responses).

The arrangement of the filter bank is shown in Figure 1. Each signal was passed through each filter separately. The output of this system was the response produced by a weighted sum of the individual filter outputs.

The results shown in this paper are those obtained using 3 filters in the filter bank. The filters were 4_{th} order Butterworth bandpass filters, implemented using the MATLAB command FILTFILT. This command produced a zero-phase shift filter, which means that the filter itself did not produce a phase shift in the signal. Butterworth filters were selected because of their relatively smooth pass-band.

All the filters were set up randomly so that initially the low frequency cut-off was within the range 0-200Hz, and the high frequency cut-off was selected to be up to 300Hz higher than the low frequency cut-off. Sub-averages (averaging small sets of signals) of the input sets was performed to reduce the noise level. Again, the results shown in this paper are those obtained when 10 responses were used in each sub-average. The stimulation rate was set at two stimulations per second, so sub-averages of 10 responses equate to 5 seconds worth of evoked responses. In the training process every example in the sub-averaged training set was used to measure the fitness of the 'individual' set of filters and weightings in the population of possible solutions. The mean of all the example fitness values for that individual solution was used Both simulated data and recorded data were used to develop and test the filter banks

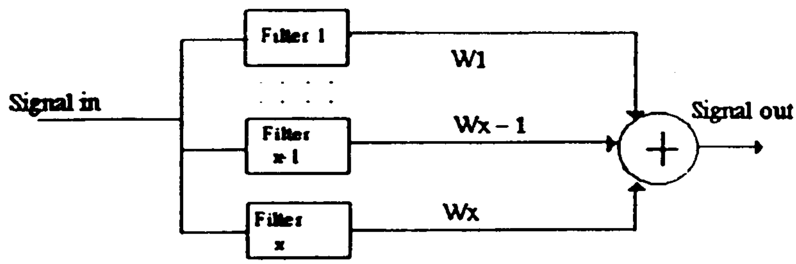

Fig. 1 - Arrangement of the set of x parallel filters

A genetic algorithm was used to select the optimum filter settings for each set of data. This is a computer-based problem solving system, which uses some of the mechanisms of evolution as key elements in their design and implementation. The computer creates a population of structures that evolve according to specified rules of selection. The evolutionary mechanisms involve procedures such as crossover, mutation and fitness-proportionate reproduction and therefore mirror such processes in the natural world. Fitness for purpose measures are then used to assess the success of this generation of structures and the process is iterated until a satisfactory end-point is reached.

The genetic algorithm for filter selection comprised of procedures for creation of initial filter parameters, evaluation of goodness-of-fit against the known signal and selection and mutation of filter parameters which were in turn evaluated for fit and so on until a satisfactory fit was obtained

Two sets of data were used initially for this work. To simulate the situation whereby the underlying signal is unknown, the test and training sets of data were used. The reference signal + noise data was used to assess the performance of the system against a known signal.

A population of random 3-filter 'chromosomes' was created (n=80), each encoded as a sequence of floating point numbers on the chromosomes. This representation was selected as it closer to the values expected and the methods used. Michaelwicz (1996) suggests that floating-point values are "intuitively closer to the problem space". Each chromosome comprised of 9 numbers, representing the low frequency limit, the high frequency limit and the weighting factor for each of the three filters.

The Mean Squared Error (MSE) between the results of the evolutionary algorithm and the known target response was used to assess goodness of fit.

$$mse = \frac{1}{N}\sum_{k=1}^{N} e(k)^2$$

The error is the difference between the target signal and the test sequence at a point in time, *e(k)*. The fitness function for the evolutionary algorithm was the mean MSE for all the training signals.

The next generation of chromosomes was generated by first selecting the 25% of chromosomes, which had the best fit and passing them unchanged to the next generation. The remaining 75% were produced by a selection process using crossover. The selection process used in this work was the roulette wheel approach where the higher the fitness of an individual sequence, the greater the probability that the sequence's genes will be used in the next generation. A set of pairs of random numbers, ranging from 1 to the sum of all portions of roulette wheel, was used to select the sequences that were the 'parents' of the next generation. A third random number was produced, that determines where along the sequence the swapping occurs, so the two original sequences produce two new sequences. A further random mutation was made in one gene of 5% of the next generation of chromosomes. These random mutations were limited to ±12.5% of the previous gene value.

Results

Simulated data

Figure 2 shows the averaged signal used both as the underlying signal of the simulated data sets and as the target signal.

This spinal recording was chosen as it has small but important early components and much larger later components which were more time invariant. It therefore combines many of the features that need to be taken into account in the extraction of the evoked potentials from the background noise.

Five unfiltered sub-averages of simulated test data and the effects of filtering are shown in Figure 3. The signals have been shifted along the voltage axis to aid the visual presentation. Three filters were developed using the simulated training set Comparing the results of the set of filters and the target signal, resemblance between the target signal and these filters can be seen

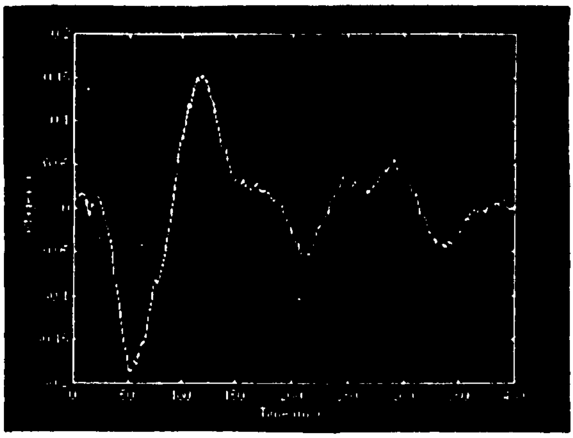

Figure 2: Target signal

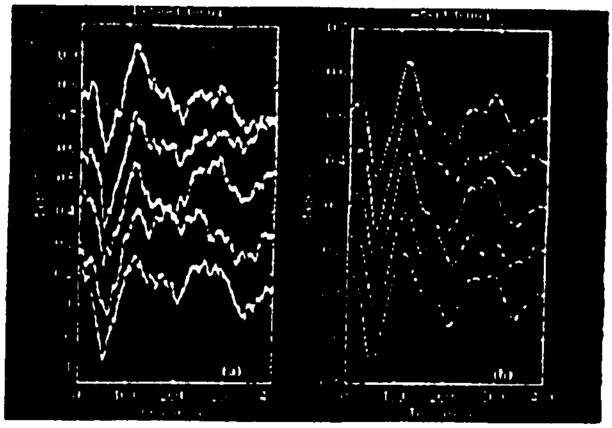

Figure 3: (a) Unfiltered data comprising averages (n=10) of simulated data (target signal + recorded noise), (b) The signals in (a) after being passed through a set of filters whose parameters were selected by a genetic algorithm. (Filter characteristics: 4-27 Hz, weighting factor w= 0.98, 67-577 Hz w= 0.28, 133-637 Hz w=0.05)

The most noticeable feature of these filtered signals is that they have negative peaks at around 50ms and 200ms. Two positive peaks in the region 100-200ms were also observed. These peaks are present in the target signal (Figure 2) Later components around 250, 300, 350ms do not appear in the majority of the signals. At the beginning of the signal, features are not present or have been 'flattened'. This data set contains stationary (time invariant) signals and it can be seen that this technique gives good noise reduction although there is a reduction in amplitude of the small early components in this instance

Recorded data (signal unknown)

This process was repeated using input signals which were averages of 10 sweeps each of consecutively recorded evoked responses The training set used by the genetic algorithm to define the filter settings was an average of the preceding 92 recorded responses (training set). The input and output signals are shown in Figure 4.

A noticeable feature of all the processed responses, whether from simulated or recorded data, was that features at the beginning of the response were not as large as they were in the target signal. The reason for this was that these components were small compared with the rest of the response, and have higher frequency components than those in the rest of the response. The dominant features were therefore these larger components, and this was the feature the algorithms have preferentially found. Changes in the larger low frequency components produced larger changes in MSE than the higher frequency components of the smaller early components. Both sets of filters illustrated above have a high weighting on low frequencies, which would help to explain why components at the beginning of the response were smoothed out or reduced. This fits with groups such as Rossini et al. (1981), who used a bandpass filter with relatively high frequency parameters (e.g. 150-3000 Hz) to extract short latency components (early components of the responses). Increasing the number of filters was investigated, but the results were no better, and so did not justify the extra processing needed. A range of different number of responses per average for the input signal was used, but the optimum number of pre-processing sweeps was found to be 10.

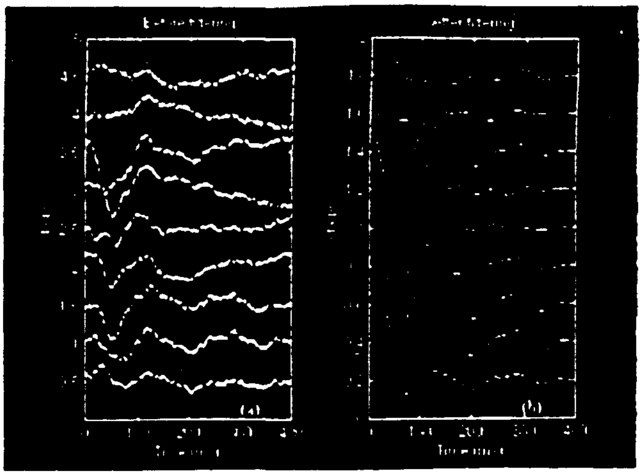

Figure 4: (a) Unfiltered data comprising averages (n=10) of recorded data. (b) The signals in (a) after being passed through a set of filters whose parameters were selected by a genetic algorithm. (Filter characteristics: 4-18 Hz, weighting factor w = 0.62, 53-56Hz w = 0.31, 42-260 Hz w = 0.05)

Conclusions

Filtering stimulated responses produced better results than filtering recorded responses. This was believed to be due to the simulated response being time invariant. They were produced by taking the target responses, repeating it several times, and adding recorded noise. This means that the underlying response was not changing between the responses. In the recorded data, the underlying response can vary between the responses. All averaging procedures assume that the signal is stationary and hence that the frequency components are the same throughout the response. A visual inspection of the responses suggests this is not true, as does the work by various groups using high-pass filtering to extract short latency components (e.g. Rossini et al.(1981), McCabee et al. (1983)). A possible way around this problem is to allow the filters to contribute to the overall final response at different times. These are time-varying filters, and work is on-going to investigate these.

References

Harrison SAB, Lovely DF (1995) Identification of noise sources in surface recording of spinal somatosensory evoked potentials. Medical & Biological Engineering & Computing pp. 299-305.

McCabee PJ, Pinkhasov EI, Cracco RQ (1983) Short latency somatosensory evoked potentials to median nerve stimulation effect of low frequency filter. Electroencephalography and Clinical Neurophysiology Vol. 55 pp 34-44.

Michaelwicz Z (1996) Genetic Algorithms + Data Structures = Evolution Programs 3rd Edition, Springer.

Rossini PM, Cracco RQ, Cracco JB, House WJ (1981) Short latency somatosensory evoked potentials nerve stimulation: scalp topography and the effect of different frequency filters Electroencephalography and Clinical Neurophysiology vol. 52 pp 540-552

Use of Evolutionary Algorithms to Enhance the Extraction of Short Latency Evoked Potentials

Scott J Turner, Phil D Picton, Jackie A Campbell.
Faculty of Applied Sciences, University College Northampton,
Northampton, NN2 6JD, UK.

ABSTRACT

An evolutionary algorithm has been used to determine the cut-off frequencies for a bank of band-pass filters. The output of the filter bank is a weighted sum of the outputs of the individual filters, where the weights are also determined by the evolutionary algorithm. These filters process evoked potential signals recorded at the spine or scalp of a patient. These signals are very noisy, and it is common practice to use ensemble averaging to remove the noise, which usually requires a large number of responses. The aim of this work is to reduce the number of responses required by filtering sub-averages.

Keywords: Evoked potentials, evolutionary algorithms, filter design , short latency.

1. INTRODUCTION

Evoked potentials (or evoked responses) are electrical signals recorded from a human body in response to stimuli to the nervous system. Somatosensory evoked potentials are recorded at sites such as the scalp or spine, usually due to direct electrical stimulation of nerves in arms or legs. The features looked for are negative or positive peaks at certain known values of time, e.g. at 20msec. The main problem with evoked potentials is the presence of noise from, for example, other sources within the body, and recording equipment [1]. There are several difficulties with this noise, one of which is that the spectral components of the noise overlap with those of the evoked potential. This means that just applying a bandpass filter will not extract the evoked potential, and the noise components are often much larger than those of the evoked potential. Ensemble averaging is the most commonly used method of reducing the noise in evoked potential recordings. The main disadvantage of this method is that to produce a reasonably noise-free signal, a large number of signals need to be averaged. Collection of a large number of signals means that signals need to be collected over relatively long periods of time. Taking a long time to collect the data may be undesirable for the subject under going the tests, or even impractical, and variations between signals can lead to distortion of features in the averaged signal. After ensemble averaging, a single bandpass filter is often applied [2].

The aim of this work is to investigate the use of a set of bandpass filters to process the signals after ensemble averaging in order to reduce the number of signals that are needed to extract the evoked potential. The searching abilities of evolutionary algorithms were used to select appropriate filter parameters and weights. Previous work has concentrated on extracting the entire response from the noisy signal which lasts for around 400ms [3]. It is believed by many authors e.g. Maccabee et al. [4] that the earlier components (the first 30 ms) are more stable than the late components, and that a great deal of useful information can be provided from this early component. This paper therefore describes work which concentrates only on the extraction of the earlier components.

2. METHOD

Equipment and Data

All the signals were recorded on FM tape, using a STORE 4 FM tape recorder (RACAL Recorders), in response to stimulation of the median nerve at the wrist. The data was collected using a Gateway 2000 Pentium P90 computer via an interface card and data acquisition software (PC30F, Eagle Technology). All the filters and evolutionary algorithms were developed in MATLAB (MathWorks, USA). Before being stored on tape, the signals were passed through a bandpass filter (0.016-750Hz). Data set 1 consists of 184 responses split equally into a test set and a training set. An average of the 184 recorded responses was used to form a reference signal which was the target signal that the filters aim to extract. In addition it was possible to make simulated data by adding noise to this reference signal. Electrical activity recorded just before stimulation occurred was the source of the added noise. This was chosen as it represents electrical activity recorded at the same site as where the evoked responses were to be recorded and should therefore contain similar kinds of electrical activity as the background noise on the evoked response recordings. This simulated data (target signal + noise) set was split into a training set (55 responses) and a test set (56 responses). Data set 2 consists of a test set with 117 responses, and a training set with 117 responses. Data set 3 consists 122 signals in the training set and 121 signals in the test set. Data sets 1 and 2 were spinal recordings, and data set 3 was a scalp recording.

Filter Banks

The arrangement of the filter bank is shown in Figure 1. The signal was passed through each filter separately. The output of this system was the response produced by a weighted sum of the individual filter outputs.

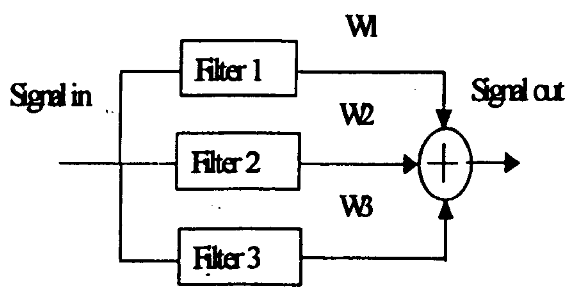

Figure 1 Modelling the response as a set of 3 parallel filters

The results shown in this paper are those obtained using 3 filters in the filter bank. The filters were 4th order Butterworth bandpass filters, implemented using the MATLAB command FILTFILT. This command produced a zero-phase shift filter, which means that the filter itself did not produce a phase shift in the signal. Butterworth filters were selected because of their relatively smooth pass-band.

All the filters were set up randomly so that initially the low frequency cut-off was within the range 0-200Hz, and the high frequency cut-off was selected to be up to 300Hz higher than the low frequency cut-off. Sub-averages (averaging small sets ignals) of the input sets were created to reduce the noise d. Again, the results shown in this paper are those obtained n 10 responses were used in each sub-average. The ulation rate was set at two stimulations per second, so sub-ages of 10 responses equate to 5 seconds worth of evoked ponses. In the training process every example in the sub-aged training set was used to measure the fitness of the ividual' set of filters and weightings in the population of sible solutions. The mean of all the example fitness values that individual solution was used. Both simulated data and orded data were used to develop and test the filter banks.

olutionary Algorithm

e filter parameters were encoded as a sequence of floating point mbers on the chromosomes. Michalewicz [5] suggests that ating point values are "intuitively closer to the problem ace."

- **Fitness Function:** The fitness function used here was e Mean Squared Error (MSE) between the results of the olutionary algorithm and the known target response. The error is e difference between the target signal and the test sequence at a int in time, $e(k)$.

$$mse = \frac{1}{N}\sum_{k=1}^{N} e(k)^2 \qquad \text{Equation (1)}$$

Selection and Mutation: After the fitness of each of the lter banks has been calculated the top quarter of the original opulation go through to the next generation unchanged i.e. those ith the highest fitness (i.e. lowest MSE values). The remaining ree quarters of the population in the next generation were roduced by a selection process using crossover. The selection rocess used in this work is the roulette wheel approach where the igher the fitness of an individual sequence, the greater the robability that the sequence's genes will be used in the next eneration. A set of pairs of random numbers, ranging from 1 to he sum of all portions of roulette wheel, was used to select the equences that were the 'parents' of the next generation. A third andom number was produced that determines where along the equence the swapping occurs, so the two original sequences roduce two new sequences. A second matrix was formed that was he same size and shape as the population matrix, and contained values in the range 0 to 1. If the value in the matrix was less than or equal to the mutation rate, then a change was made to the value in the population matrix at the corresponding position. The value in the population matrix was altered by up to +/- 12.5% of the current value. The population size was chosen as 80 individuals and the mutation rate was set at 0.05. The evolutionary algorithms were all stopped after 400 generations.

3. RESULTS

Figure 2 shows how the MSE varied during training as the number of generations increased for all four data sets. The MSE converged to a stable value after 400 generations.

Figures 3 to 6 show results of the trained filter banks for each data set on the test data. Table 1 compares the mean squared error between the signal and the target signal, when the sub-averages are unfiltered, filtered with a filter bank developed for the same data set but for the whole of the signal (400ms) and the filter bank trained on only the 1[st] 30ms of the signal. Visually there is an improvement when using these filter banks over the unfiltered signals. Table 1 shows the lowest MSE values were produced when the filters were developed for this smaller region of the signal.

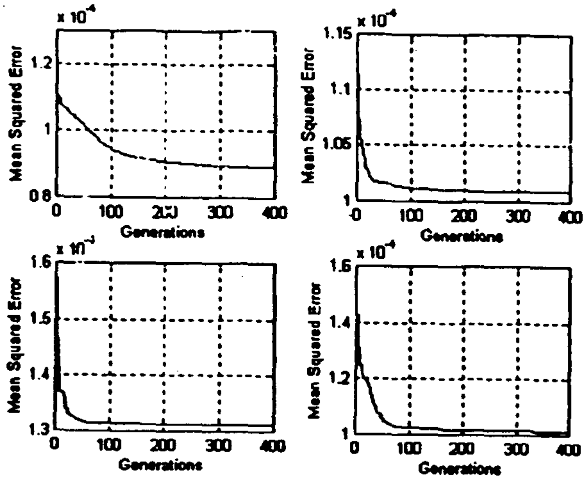

Figure 2 The variation in mean squared error against the number of generations during the training process for (a) simulated data set (b,c, and d) data set 1,2,3 respectively.

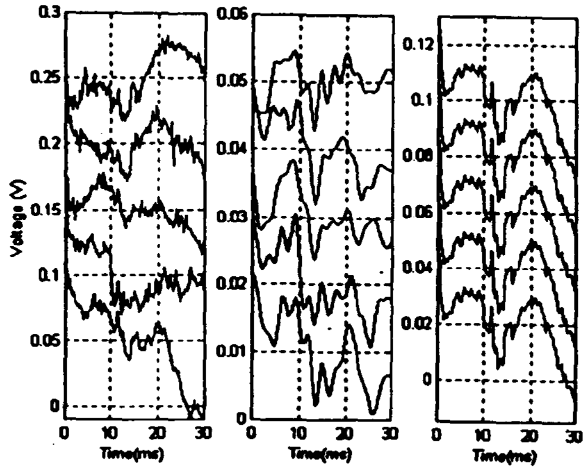

Figure 3 The results of a bank of three filters filtering the region 1[st] 30ms of the simulated test subset, each signal a sub-average of ten signals. The filters were trained on the 1[st] 30ms of the training subset for the data set. (a) The unfiltered sub-averages. (b) The sub-average signals after filtering. (c) The target signals.

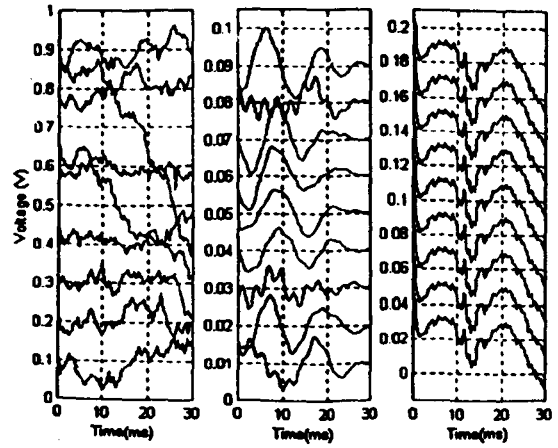

Figure 4 The results of a bank of three filters filtering the region 1st 30ms of data set 1's test subset, each signal a sub-average of ten signals. The filters were trained on the 1[st] 30ms of the training subset for the data set. (a) The unfiltered sub-averages. (b) The sub-average signals after filtering. (c) The target signals.

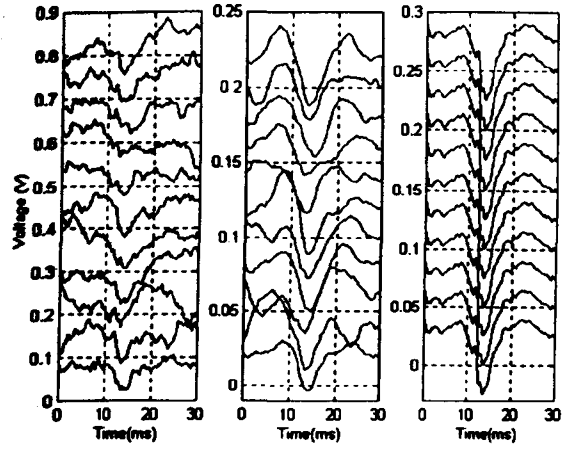

Figure 5 The results of a bank of three filters filtering the region 1st 30ms of data set 2's test subset, each signal a sub-average of ten signals. The filters were trained on the 1st 30ms of the training subset for the data set. (a) The unfiltered sub-averages. (b) The sub-average signals after filtering. (c) The target signals.

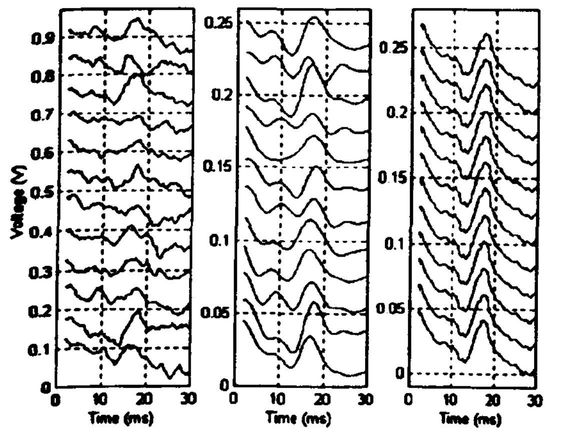

Figure 6 The results of a bank of three filters filtering the region 2 to 30ms of data set 3's test subset, each signal a sub-average of ten signals. The filters were trained on the 2-30ms of the training subset for the data set. (a) The unfiltered sub-averages. (b) The sub-average signals after filtering. (c) The target signals.

	Mean Squared Error (10^{-3})		
DATA SETS	unfiltered	Whole	1st 30ms
Simulated	0.1805	0.1008	0.0764
Data set 1	3.6593	0.1302	0.1143
Data set 2	0.3669	0.2047	0.0851
Data set 3	0.1593	0.1578	0.0675

Table 1 A comparison of the filter bank produced and applied to the 1st 30ms, and when no filtering is applied, and when filter developed for 400ms were used on this region.

4. DISCUSSION

Evolutionary algorithms enable less specific assumptions to be made about the frequency properties of the signal beforehand. Using an evolutionary algorithm, the algorithm can select cut-off frequencies for the filter, and weights, based on how well the filtered signal matches the shape of the target signal. A filter bank approach was selected so that spectral areas that are not important to extracting the evoked potential are less likely to be included in the filtered result.

A noticeable feature seen in figures 4 and 5 is the sharper features in the signals are not being extracted, and in figure 3 these are being extracted but so are some artefacts.

5. CONCLUSIONS AND FURTHER WORK

Previous work [2,4,6] used ensemble averaging followed by a single band-pass filter. This paper has shown that fewer samples can be used for ensemble averaging when a more sophisticated filter bank is used as a post processor. The cut-off frequencies and weights were optimised using an evolutionary algorithm. It was noted that filtering simulated responses produced better results than filtering recorded responses (Table 1). This was believed to be due to the simulated response being time invariant. They were produced by taking the target responses, repeating it several times, and adding recorded noise. This means that the underlying response was not changing between the responses. In the recorded data, the underlying response can vary between the responses. An assumption has to be made about the response that the frequency components were the same throughout the response, i.e. that it is stationary. A visual inspection of the responses suggests this is not true, as does the work by various groups using high-pass filtering to extract short latency components (e.g. Rossini et al. [6], Maccabee et al. [2]). Therefore, further work is needed to design time variant filters, which could also be done using evolutionary algorithms possibly in combination with wavelets.

6. REFERENCES

[1] S. A. B. Harrison, D. F. Lovely, "Identification of Noise Sources in Surface Recording of Spinal Somatosensory Evoked Potentials", Medical & Biological Engineering & Computing, 1995, pp. 299-305.

[2] P. J. Maccabee, E. I. Pinkhasov, R. Q. Cracco, "Short Latency Somatosensory Evoked Potentials to Median Nerve Stimulation Effect of Low Frequency Filter" Electroencephalography and Clinical Neurophysiology, 55, 1983, pp. 34-44.

[3] S. J. Turner, P. D. Picton, J. A. Campbell, "Selecting Filter Banks to Enhance Evoked Potentials Recordings Using Evolutionary Algorithms", Proc. First European Workshops, EvoIASP'99 and EuroEcTel'99, Goteborg, Sweden, May 1999, pp. 101-109.

[4] P. J. Maccabee, N. F. Hassan, "AEEM Minimonograph #39: Digital Filtering: Basic Concepts and Applications to Evoked Potentials", Muscle and Nerve, 15, 1992, pp. 865-875.

[5] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd Edition, Springer, 1996.

[6] P. M. Rossini, R. Q. Cracco, J. B. Cracco, W. J. House, "Short Latency Somatosensory Evoked Potentials Nerve Stimulation: Scalp Topography and the Effect of Different Frequency Filters" Electroencephalography and Clinical Neurophysiology, 52, 1981, pp. 540-552.

Poster 5

IMPROVED SIGNAL TO NOISE RATIO IN SOMATOSENSORY EVOKED POTENTIALS

[?] Turner, JA Campbell, PD Picton, Nene College of Higher Education, St George's Avenue, Northampton NN2 6JD

One of the largest challenges in the processing of evoked potentials is overcoming their low signal-to-noise ratio (SNR). The aim of this research is to improve the SNR by the use of filtering and estimation techniques.

The poster shows the distribution of spectral energy of potentials from scalp recordings, both with and without synchronised stimulation of Sensory nerves. This information is then used to design lowpass filters that have zero-phase shift between filtered and unfiltered signals, customised to the individual recording situation. This gives an increase in SNR, but does not overcome the problem that there are spectral components of the unstimulated recordings, which is considered as background noise, in the same region as stimulated responses.

The results of a variety of estimation techniques performed on the evoked signals and simulated signals are presented to compare their effects on SNR.

* * *

Poster 6

EQUIPMENT SELECTION & SUPPORT OF INEXPENSIVE OSCILLOMETRIC ELECTRONIC NON-INVASIVE BLOOD PRESSURE (NIBP) SPHYGMOMANOMETERS

PD Lee, Medical Equipment Management Department, King's Mill Centre, Mansfield Road, Sutton-in-Ashfield NG17 4JL

Inexpensive electronic oscillometric sphygmomanometers are becoming increasingly prevalent in the healthcare setting because of the toxicity of mercury, the [illegible] "white coat" hypertension on limited budgets [illegible] technical note addresses [illegible]

Selecting Filter Banks to Enhance Evoked Potentials Recordings Using Evolutionary Algorithms

SJ Turner, PD Picton, JA Campbell
University College Northampton, Northampton, NN2 6JD, UK
scott.turner@nene.ac.uk

Abstract. Evoked potentials are electrical signals produced by the body in response to a stimulus. In general these signals are noisy with a low signal to noise ratio. In this paper a method is proposed that uses sets of filters, whose cut-off frequencies are selected by an evolutionary algorithm. An evolutionary algorithm was investigated to limit the assumptions that were made about the signals. The set of filters separately filter the evoked potentials, and are combined as a weighted sum of the filter outputs. The evolutionary algorithm also selects the weights. Inputs to the filters are sets of averaged signal, 4 or 10 signals per average. Even though there is likely to be variations between the signals, this process can improve the extraction of potentials.

1. Introduction

Evoked potentials (or evoked responses) are electrical signals recorded from a human body in response to a stimulus to the nervous system. Somatosensory evoked potentials in particular are recorded at sites such as the scalp or spine, ordinarily due to direct electrical stimulation of the nerves in the arms or legs. The features looked for are negative or positive peaks at certain known values, e.g. at 20msec or 300msec. The main problem with evoked potentials is the presence of noise from, for example, other sources within the body, recording equipment, or the local environment [1]. Noise can dominate the recorded signal, leading to a very low signal to noise ratio. There are several difficulties with this noise, one of which is that the spectral components of the noise overlap the same region as those of the evoked potential. This means that just applying a bandpass filter will not extract the evoked potential, and the noise components are often larger than those of the evoked potential. Ensemble averaging is the most commonly used method of reducing the noise in evoked potential recordings. The main disadvantage of this method is that to produce a reasonably noise-free signal, a large number of signals need to be averaged. Collection of a large number of signals means that signals need to be collected over relatively long periods of time. Taking a long time to collect the data may be undesirable for the subject under going the tests, or even impractical, and variations between signals can lead to distortion of features in the averaged signal. After ensemble averaging, a single bandpass filter is often applied. The aim of this work is to investigate using a set of bandpass filters to reduce the number of signals that are needed to extract the evoked potential. The searching abilities of evolutionary algorithms were used to select appropriate filter parameters and weights.

2. Method

2.1 Equipment And Data

All the signals were recorded on FM tape, using a STORE 4 FM tape recorder (RACAL Recorders), from spinal recorded evoked potentials in response to stimulation of the median nerve at the wrist. The data was collected using a Gateway 2000 Pentium P90 computer via an interface card and data acquisition software (PC30F, Eagle Technology). All the filters and evolutionary algorithms were developed and implemented in MATLAB (MathWorks, USA). Before being recorded the signals were passed through a bandpass filter (0.016-750Hz).

Recorded data consisting of 222 responses were collected from the tape. A total of 38 responses were excluded from the experiments as they were found to contain artifacts such as 'clipping.' Using the remaining 184 recorded responses, two sets of data with 92 responses in each were formed into a test and a training set These sets are referred to here, as the recorded data. An average of the 184 recorded responses was used to form a reference signal which was the target signal that the filters aim to extract. In addition it was possible to make simulated data by adding noise to this reference signal. Pre-stimulus recordings, i.e. electrical activity recorded just before stimulation occurred, was the source of the added noise. This was chosen as it represents electrical activity recorded at the same site as where the evoked responses were to be recorded and should therefore contain similar kinds of electrical activity as the background noise on the evoked response recordings. This simulated data (target signal + noise) set was split into a training set (55 responses) and a test set (56 responses).

2.2 Filter Banks

The arrangement of the filter bank is shown in Figure 1. The signal was passed through each filter separately. The output of this system was the response produced by a weighted sum of the individual filter outputs.

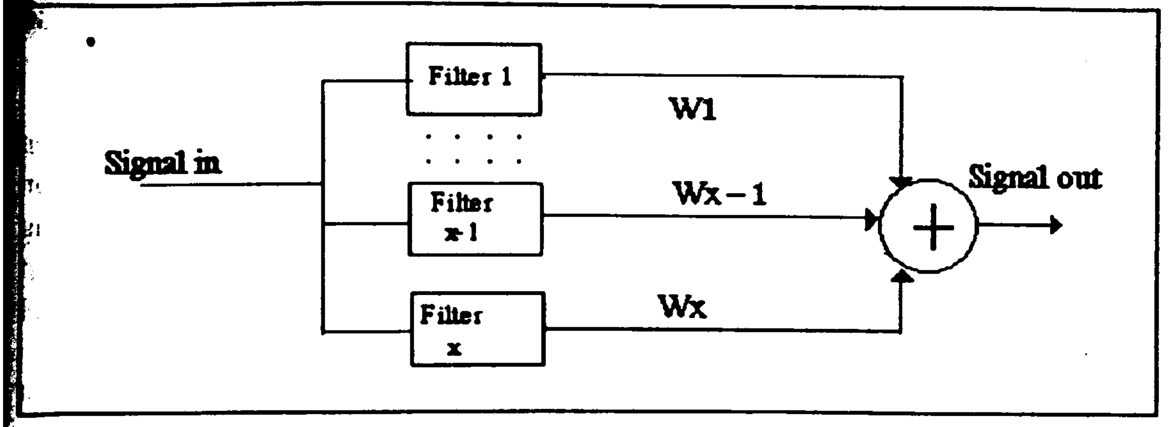

Fig. 1. Modeling the response as a set of x parallel filters

The results shown in this paper are those obtained using 3 filters in the filter bank. The filters were 4th order Butterworth bandpass filters, implemented using the MATLAB command FILTFILT. This command produced a zero-phase shift filter, which means that the filter itself did not produce a phase shift in the signal. Butterworth filters were selected because of their relatively smooth pass-band.

All the filters were set up randomly so that initially the low frequency cut-off was within the range 0-200Hz, and the high frequency cut-off was selected to be up to 300Hz higher than the low frequency cut-off. Subaverages (averaging small sets of signals) of the input sets were created to reduce the noise level. Again, the results shown in this paper are those obtained when 10 responses were used in each sub-average. The stimulation rate was set at two stimulations per second, so sub-averages of 10 responses equate to 5 seconds worth of evoked responses. In the training process every example in the sub-averaged training set was used to measure the fitness of the 'individual' set of filters and weightings in the population of possible solutions. The mean of all the example fitness values for that individual solution was used. Both simulated data and recorded data were used to develop and test the filter banks.

2.3 Evolutionary Algorithm

The filter parameters were encoded as a sequence of floating point numbers on the chromosomes. Michalewicz [3] suggests that floating point values are "intuitively closer to the problem space."

Fitness Function. The fitness function used here was the Mean Squared Error (MSE) between the results of the evolutionary algorithm and the known target response.

$$mse = \frac{1}{N}\sum_{k=1}^{N} e(k)^2 \qquad (1)$$

The error is the difference between the target signal and the test sequence at a point in time, $e(k)$.

Selection and Mutation After the fitness of each of the filter banks has been calculated the top quarter of the original population go through to the next generation unchanged i.e. those with the highest fitness (i.e. lowest MSE values). The remaining three quarters of the population in the next generation were produced by a selection process using crossover. The selection process used in this work is the roulette wheel approach where the higher the fitness of an individual sequence, the greater the probability that the sequence's genes will be used in the next generation. A set of pairs of random numbers, ranging from 1 to the sum of all portions of roulette wheel, was used to select the sequences that were the 'parents' of the next generation. A third random number was produced that determines where along the sequence the swapping occurs, so the two original sequences produce two new sequences. A second matrix was formed that was the same size and shape as the population matrix, and contained values in the range 0 to 1. If the value in the matrix was less than or equal to the mutation rate, then a change was made to the value in the population matrix at the corresponding position. The value in the population matrix was altered by up to +/- 12.5% of the current value. The population size was chosen as 80 and the mutation rate was set at 0.05. The evolutionary algorithms were all stopped after 200 generations.

3. Results

Figure 2 shows the averaged signal used both as the underlying signal of the simulated data sets and as the target signal.

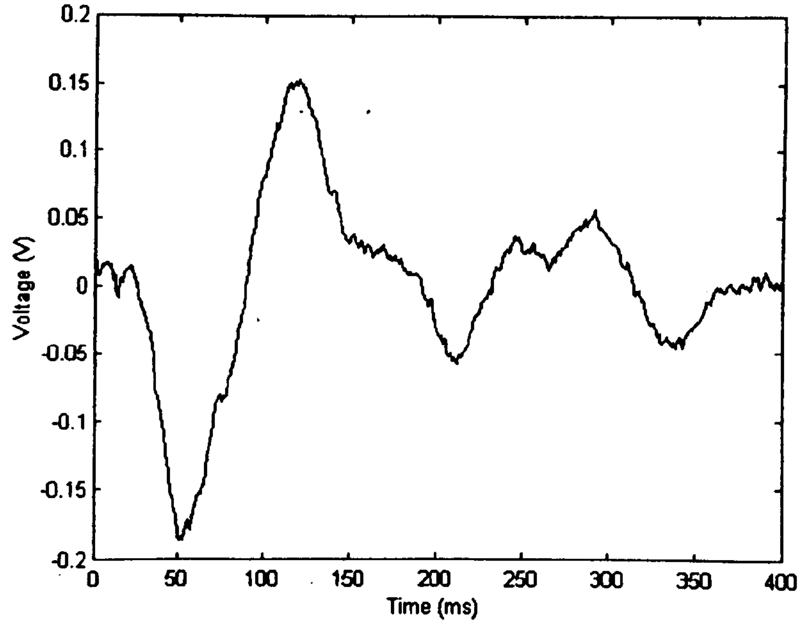

Fig. 2. Target signal

This spinal recording was chosen as it has small but important early components and much larger later components which were more time invariant. It therefore combines many of the features that need to be taken into account in the extraction of the evoked potentials from the background noise.

Five unfiltered sub-averages of simulated test data and the effects of filtering are shown in Figure 3. The signals have been shifted along the voltage axis to aid the visual presentation. Three filters were developed using the simulated training set. Comparing the results of the set of filters and the target signal, resemblance between the target signal and these filters can be seen. The most noticeable feature of these filtered signals is that they have negative peaks at around 50ms and 200ms. Two positive peaks in the region 100-200ms were also observed. These peaks are present in the target signal (Figure 2). Later components around 250, 300, 350ms do not appear in the majority of the signals. At the beginning of the signal, features are not present or have been 'flattened.'

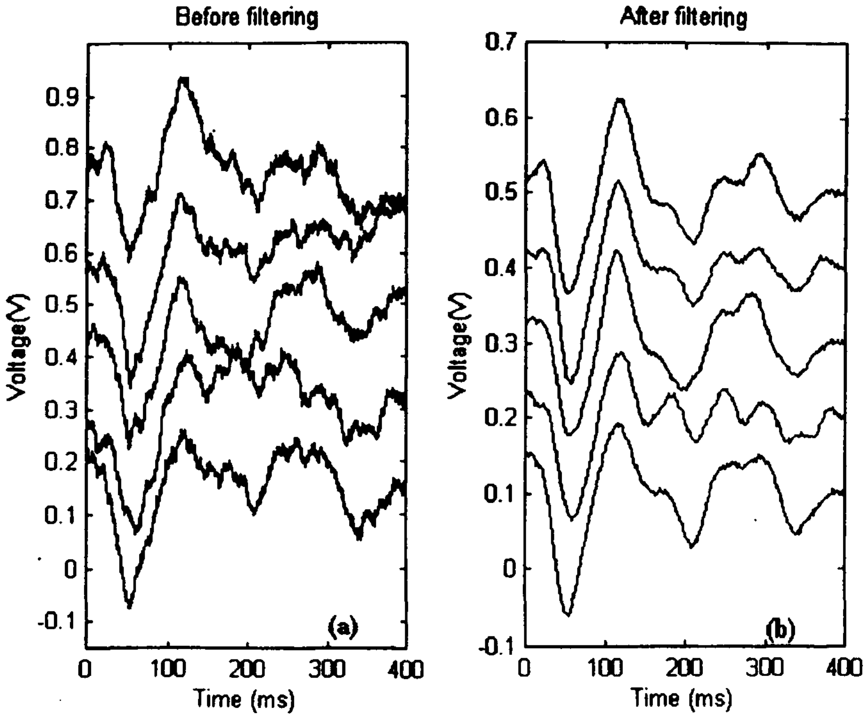

Fig. 3. (a) Averages of simulated activity (10 simulated evoked response per average). (b) The signals in (a) after being passed through a set of filters whose parameters were selected by an evolutionary algorithm, based on training with a different set of simulated responses.

Recorded evoked potentials were passed through the filters used previously. Figure 4 shows both the unfiltered and filtered responses. As in Figure 3, some of the features can be seen, but the similarities with the target signal are not as clear as when the simulated test data were filtered. Figure 5 shows the unfiltered averaged test recorded data again, but this time the signals are passed through a set of filters developed using the recorded data training set. In comparison with Figure 4, these are essentially the same shape but smoother. Table 1 contains the filter parameters and weightings for both sets of filters. Table 2 contains the minimum, maximum, mean and standard deviations of the MSE values.

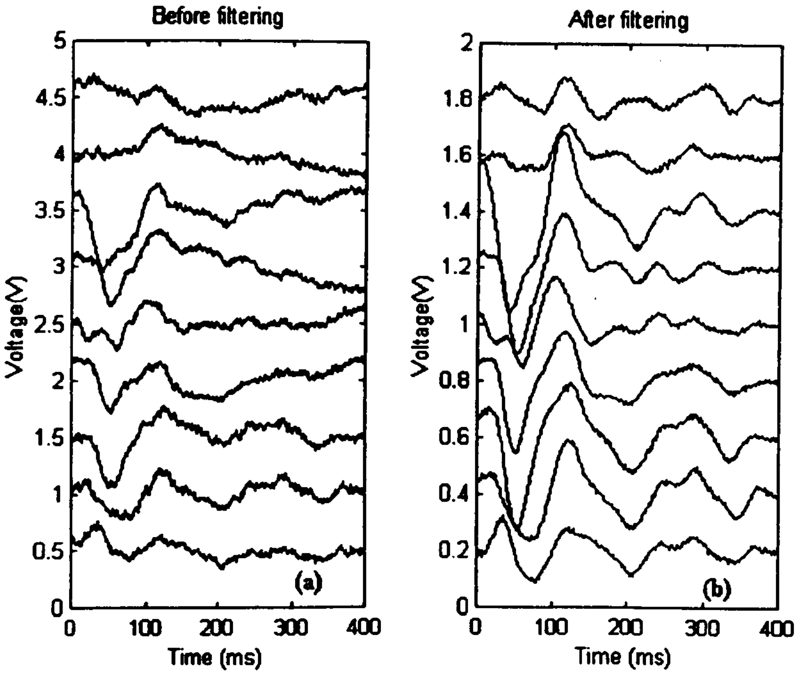

Fig. 4. (a) Averages of recorded activity (10 recorded evoked response per average). (b) The signals in (a) after being passed through a set of filters whose parameters were selected by an evolutionary algorithm (same filters as used in figure 3.)

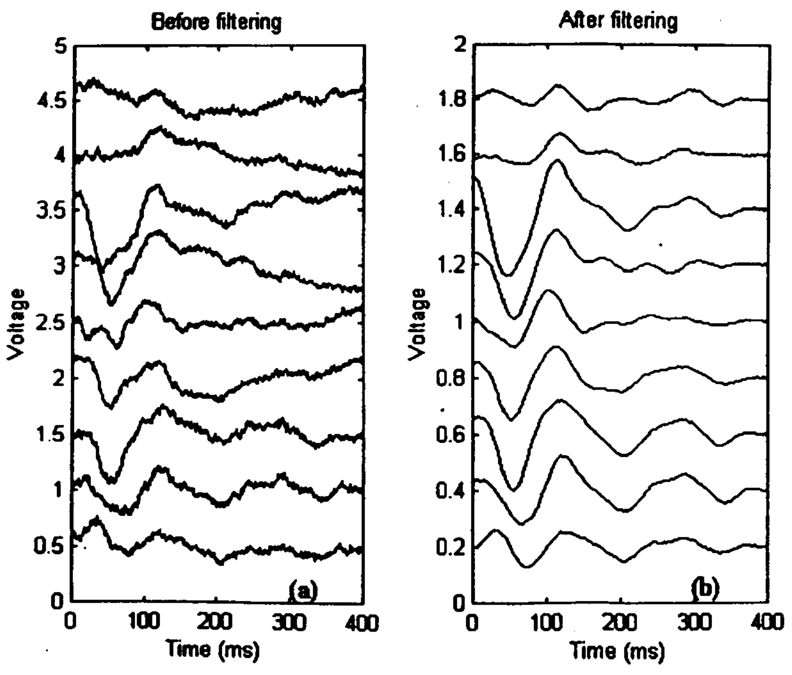

Fig. 5. (a) Averages of recorded activity (10 recorded evoked response per average). (b) The signals in (a) after being passed through a set of filters whose parameters were selected by an evolutionary algorithm, based on training with a different set of recorded responses.

	Cut-off frequencies fx	(Hz) fx+1	weight
3 filter bank (simulated)	4.022	26.9667	0.9307
	66.751	576.816	0.2848
	133.398	637.093	0.0563
3 filter bank (recorded)	3.71	18.4259	0.6253
	52.788	55.649	0.31
	42.545	260.143	0.086

Table 1. Filter parameters selected using the evolutionary approach

Training Data	Test Data	min 10^{-3}	max. 10^{-3}	mean 10^{-3}	STD 10^{-3}
simulated	simulated	0.39	0.81	0.61	0.18
	recorded	1.6	5.9	2.7	1.4
recorded	simulated	1.0	1.4	1.2	0.2
	recorded	0.6	3.8	1.7	1.1

Table 2. Mean Squared Error values for the two filters.

4. Discussion

Evolutionary algorithms enable less specific assumptions to be made about the frequency properties of the signal beforehand. Using an evolutionary algorithm the algorithm can select cut-off frequencies for the filter, and weights, based on how well the filtered signal matches the shape of the target signal. A filter bank approach was selected so that spectral areas that are not important to extracting the evoked potential are less likely to be included in the filtered result.

A noticeable feature of all the processed responses, whether from simulated or recorded data, was that features at the beginning of the response were not as large as they are in the target signal. The reason for this was that these components were small compared with the rest of the response, and have higher frequency components than those in the rest of the response. The dominant features were therefore these larger components, and this was the feature the algorithms have found. Changes in the larger low frequency components produced larger changes in MSE than the higher frequency components of the smaller early components. In Table 1, a list of the cut-off frequencies of the filter banks developed are given. Common to both set of filters is a high weighting on low frequencies, which would help to explain why components at the beginning of the response were smoothed out or reduced. This fits with groups such as Rossini et al. (1981) [4], who used a bandpass filter with relatively high frequency parameters (i.e. 150-3000 Hz) to extract short latency components (early components of the responses). The idea of a bandpass filter to extract these components does therefore seem relevant. Increasing the number of filters was investigated, but the results were no better, and so did not justify the extra processing needed. A reduction in the number of responses per average was tried, but the combination of the 'noisier' inputs signals and filters produced were not as effective as those where 10 responses were used.

5. Conclusions

Filtering simulated responses produced better results than filtering recorded responses (Table 2). This was believed to be due to the simulated response being time invariant. They were produced by taking the target responses, repeating it several times, and adding recorded noise. This means that the underlying response was not changing between the responses. In the recorded data, the underlying response can vary between the responses. The results of the filter developed using recorded data suggest that it was better than the filter developed using simulated data, for filtering recorded data (Table 2), at least for the later components of the signals. An assumption has to be made about the response that the frequency components were the same throughout the response, i.e. that it is stationary. A visual inspection of the responses suggests this is not true, as does the work by various groups using high-pass filtering to extract short latency components (e.g. Rossini et al. (1981) [4], McCabee et al. (1983)[2]). A possible way around this problem is to allow the filters to contribute to the overall final response at different times. These are time-varying filters, and work is on-going to investigate these. The effects of using other sets of intraspinal recordings and scalp recordings are also needed to investigate the effects of variations in recordings between subjects. A particular problem area is that the later components of the signals are likely to vary more than the earlier components. It is possible that other data sets may have signals that vary more than these, but this would also be a problem for the conventional ensemble average method.

Refe

1. Ha
rec
En

2. Mc
evc
Ele

3. Mi
Pr

4. Ro
sor
eff
Ne

References

1. Harrison SAB, Lovely DF (1995) "Identification of noise sources in surface recording of spinal somatosensory evoked potentials" *Medical & Biological Engineering & Computing* pp. 299-305.

2. McCabee PJ, Pinkhasov EI, Cracco RQ (1983) "Short latency somatosensory evoked potentials to median nerve stimulation effect of low frequency filter" *Electroencephalography and Clinical Neurophysiology* Vol. 55 pp 34-44.

3. Michalewicz Z (1996) *Genetic Algorithms + Data Structures = Evolution Programs* 3rd Edition, Springer.

4. Rossini PM, Cracco RQ, Cracco JB, House WJ (1981) "Short latency somatosensory evoked potentials nerve stimulation: scalp topography and the effect of different frequency filters" *Electroencephalography and Clinical Neurophysiology*

References

Akay M and Daubenspeck JA (1999) Investigating the Contamination of Electroencephalograms by Facial Muscle Electromyographic Activity using Matching Pursuit *Brain and Language* Vol. 66 p. 184-200.

Aguiar P, David A, Palo S, Rosa A (2000) EEGSolver-Brain Activity and Evolutionary algorithms [on-line] http://theta.ist.utl.ptl/~142549/files/paper/paper.html Accessed on 7/2/2000

Aminoff MJ (1999) Electrophysiology Ed. Goetz CG, Pappert EJ *Textbook of Clinical Neurophysiology* Saunders p. 436-457.

Aminoff MJ and Eisen A (1999) Somatosensory Evoked Potentials Ed. Aminoff MJ *Electrodiagnosis in Clinical Neurology* Churchill Livingston 4th Edition p. 513-536.

Angel A, Linkens DA, Ting CH (1999) Estimation of Latency Changes and Relative Amplitudes in Somatosensory Evoked Potentials Using Wavelets and Regression *Computers and Biomedical Research* Vol. 32 p. 209-251.

Anziska BJ, Cracco RQ (1981) Short latency SEPs to median nerve stimulation: Comparison of recording methods and origin of components *Electroencephalography and Clinical Neurophysiology* Vol. 52 p. 531-539.

Bartnik EA, Blinowska KJ, Durka PJ (1992a) Single evoked potential reconstruction by means of wavelet transform *Biological Cybernetics* Vol. 67 p. 175-181.

Bartnik EA, Blinowska KJ (1992b) Wavelets - A new method of Evoked Potential Analysis *Medical and Biological Engineering and Computing* Vol. 30 p. 125-126.

Bentley PM, McDonnell JTE (1994) Wavelet Transforms: An Introduction *IEE Electronics & Communication Engineering Journal* p. 175-186.

Bertrand O, Bohorquez J, Pernier J (1994) Time-frequency Digital Filtering Based on an Invertible Wavelet Transform: An Application to Evoked Potentials *IEEE Transactions of Biomedical Engineering* Vol. 41 No 1 p.77-88.

Bezerianos A, Laskaris N, Fotopoulos S, Papathanasopoulos (1995) Data dependent weighted averages for recording evoked potential signals *Electroencephalography and Clinical Neurophysiology* Vol. 96 p. 468-471.

Blinowska KJ, Durka PJ (1997) Introduction to wavelet analysis *British Journal of Audiology* Vol. 31 No. 6 p. 449-459.

Braun JC, Hanley DF, Thakor NV (1996) Detection of Neurological Injury Using Time-Frequency Analysis of Somatosensory Evoked Potentials *Electroencephalography and Clinical Neurophysiology* Vol. 100 No. 4 p. 310-

leBron Paige A, Ozdamar O, Delegado RE (1996) Two dimensional spectral processing of sequential evoked potentials *Medical & Biological Engineering & Computing* Vol. 34 p. 239-243.

Cadwell JA and Villarreal (1999) Electrophysiological Equipment and Electrical Safety Ed. Aminoff MJ *Electrodiagnosis in Clinical Neurology* Churchill Livingston 4th Edition p. 15-33.

Campbell JA (1985) Observations on Somatosensory Evoked Potentials Recorded From Within the Human Spinal Cord , PhD Thesis, University of Liverpool.

Campbell JA and Leandri M (1986) The effects of High-pass Filters on Computer Reconstructed Evoked Potentials *Electroencephalography and Clinical Neurophysiology* Vol. 57 pp 99-101.

Carlton EH, Katz S (1980) Is Wiener Filtering an Effective Method of Improving Evoked Potential Estimation *IEEE Transactions of Biomedical Engineering* Vol. 27 No. 4 p. 187-192.

Challis RE, Kitney RI (1990) Biomedical Signal Processing (in four parts) Part 1 Time Domain Methods *Medical and Biological Engineering and Computing* Vol. 28 p. 509-524.

Choudhry DK, Stayer SA, Rehman MA, Schwartz RE (1998) Electrocardiographic artifact with SSEP monitoring unit during scolios surgery *Paediatric Anaesthesia* Vol. 8 p. 341-343.

Clark (1992) The Origin of Biopotentials *Medical Instrumentation: Application and Design* Ed. Webster JG 2nd Edition Houghton Muffin: Boston

Coifman RR, Wickerhauser MV (1992) Entropy-based Algorithms for best Basis Selection *IEEE Transactions on Information Theory* Vol. 38 No.2 p. 713-719.

Coppla R, Tabor R, Buchsbaum MS (1978) Signal To Noise Ratio Variability Measurements in Single Trial Evoked Potentials *Electroencephalography and clinical Neurophysiology* Vol. 4 p. 214-222

Darragh GA, Ifeachor EC, Rosen K (1995) Optimal Signal Enhancement of EPs: A Pilot Study *Proceedings of the International Workshop on Medical and Biological Signal Processing* p. 47-54.

Daubechies I (1988) Orthonormal Bases of Compact Supported Wavelets *Communications on Pure & Applied Mathematics* Vol. XLI p. 909-996.

Daubechies I (1990) The Wavelet Transform, Time-frequency Localization and Signal Analysis *IEEE Transactions on Information Theory* Vol. 36 No. 5 p. 961-1005.

Davilia CE, Mobin MS (1992) Weighted Averaging of Evoked Potentials *IEEE Transactions of Biomedical Engineering* Vol. 39 No. 4 p. 338-345.

Davis L (1991) Hybridisation and Numerical Representation Ed. Davis L *Handbook of Evolutionary algorithms* p. 61 -71.

Demiralp T, Yordanova J, Kolev V, Ademoglu A, Devrim M, Samar VJ (1999) Time-Frequency Analysis of Single-Sweep Event-Related Potentials by Means of Fast Wavelet Transforms *Brain and Language* Vol. 66, p. 129-145.

Dobie RA, Wilson MJ (1990) Optimal ('Wiener') digital filtering of auditory evoked potentials: use of coherence estimates *Electroencephalography and Clinical Neurophysiology* Vol. 77 p. 205-213.

Doncarli C, Goering L, Guiheneuc P (1992) Adaptive smoothing of evoked potentials *Signal Processing* Vol. 28 p. 63-76.

Donoho DL (1995) De-Noising by Soft Thresholding *IEEE Transactions on Information Theory* Vol. 41 No. 3 p. 613-627.

Donoho DL, Johnstone IM (1994) Ideal spatial Adaptation by Wavelet Shrinkage *Biometrika* Vol. 81 p. 425

Donoho DL, Johnstone IM, Kerkyacharian G, Picard D (1995) Wavelet Shrinkage: Asymptopia ? *Journal of the Royal Statistical Society B* Vol. 57 No. 2 p. 301-369.

Doyle DJ (1975) Some comments on the use of Wiener filtering for estimation of evoked potentials *Electroencephalography and Clinical Neurophysiology* Vol. 38 p. 533-534.

Eisen A, Roberts F, Low M, Hoirch M, Lawerence P (1984) Questions Regarding the Sequential Neural Generators Theory of the Somatosensory Evoked Potentials Biased by Digital Filtering *Electroencephalography and Clinical Neurophysiology* Vol. 59 p. 388-395.

Fung KSM, Chan FHY, Lam K, Poon PWF, Liu JG(1995) Visual Evoked Potential Estimator By Artificial Neural Network Filter: Comparison With The Ensemble Averaging Method *Proceedings of the IEEE Engineering in Medicine and Biology 17th Annual Conference* p. 805-806.

Fung KSM, Chan FHY, Lam K, Liu JG, Poon PWF (1996) Visual Evoked Potential Enhancement by an Artificial Neural Network Filter *Bio-Medical Materials and Engineering* Vol. 6 p. 1-13.

Furst M, Blau A (1991) Optimal A Posteriori Time Domain Filter for Average Evoked Potentials *IEEE Transactions on Biomedical Engineering* Vol. 38 No. 9 p. 827-833.

Geva AB, Pratt H, Zeevi YY (1995) Spatio-temporal multiple source localization by wavelet-type decomposition of evoked potentials *Electroencephalography and Clinical Neurophysiology* Vol96 p. 278-286

Geva AB, Pratt H, Zeevi YY (1997) Multichannel wavelet-type decomposition of evoked potentials: model-based recognition of generator activity *Medical & Biological Engineering & Computing* Vol. 35 No. 1 p. 40-45.

Glaser EM and Ruchkin DS (1976) *Principles of Neurobiological Signal Analysis* Academic Press: New York.

Goldberg DE (1989) *Evolutionary algorithms in Search, Optimisation & Machine Learning* Addison-Wesley: Reading,Mass.

Green JB, Nelson AV, Michael D (1986) Digital zero-phase-shift filtering of short-latency somatosensory evoked potentials *Electroencephalography and Clinical Neurophysiology* Vol. 63 p. 384-388.

Grefenstette JJ (1986) Optimisation of control parameters for genetic algorithms *IEEE Transactions on Systems, Man and Cybernetics* Vol. 16 No. 1 p. 122-128.

Grieve RCW, Parker PA, Hudgins B (1995) Adaptive stimulus Artifact Cancellation in Biological Signals using Neural Networks *Proceedings of the IEEE Engineering in Medicine and Biology 17th Annual Conference* p. 801-802.

Hansson M, Gansler T, Salomonseen G (1996) Estimation of Single Event-Related Potentials Ultizing the Prony Method *IEEE Transactions on Biomedical Engineering* Vol. 43 No. 10 p. 973-981.

Harrison SAB, Lovely DF (1995) Identification of noise sources in surface recording of spinal somatosensory evoked potentials *Medical & Biological Engineering & Computing* p. 299-305.

Haupt RL (1995) An introduction to Genetic Algorithms for Electromagnetics *IEEE Antennas and Propagation Magazine* Vol. 37 No. 2 p. 7-15.

Hayes MH (1996) *Statistical Digital Signal Processing and Modelling*, John Wiley and Sons, first Edition, p. 335-390.

Heinrich H, Dickhaus H, Rothenberger A, Heinrich V, Moll GH (1999) Single-Sweep Analysis of Event-Related Potentials by Wavelet Networks – Methodological Basis and Clinical Application *IEEE Transactions on Biomedical Engineering* Vol. 45 No. & p. 867 –879.

Holland JH (1995) *Adaptation in Natural and Artificial Systems: An Introduction Analysis with Applications to Biology, Control and Artificial Intelligence* MIT Press: Cambridge, Mass.

Isoglu-Alkac U, Baser-Eroglu C, Ademoglu A, Demiralp T, Miener H, Stadler M (1998) analysis of the Electroencephalographic activity during the necker cube reversals by means of the wavelet Transform *Biological Cybernetics* Vol. 79 p. 437-442.

Journee HL, van der Worp PE, Zeinstra E, Buchtal A, Mooji JJA (1995) Enhancement of the Acquisition Speed of Somatosensory Evoked Potentials By a Response Tuned Wavelet Filter *Proceedings of the IEEE Engineering in Medicine and Biology 17th Annual Conference* p. 1065-1066.

Karajalainen PA, Kaipio JP, Koistinen AS, Vauhkonen M (1999) Subspace Regularization Method for the Single-Trial Estimation of Evoked Potentials *IEEE Transactions on Biomedical Engineering* Vol. 48 No. 7 pp. 849-860.

Khandpur RS (1987) *Handbook of Biomedical Instrumentation* p.5 Tata McGraw Hill: New Dehli.

Kriss A (1985) Setting up an Evoked Potential (EP) Laboratory Ed. Halliday AM *Evoked Potentials in Clinical Testing* p. 1-44.

Lange DH, Inbar GF (1996) A Robust Parametric Estimator for Single-Trial Movement Related Brain Potentials *IEEE Transactions on Biomedical Engineering* Vol. 43 No. 4 p. 341-347.

Lange DH, Pratt H, Inbar GF (1997) Modelling and Estimation of Single Evoked Brain Potential Components *IEEE Transactions on Biomedical Engineering* Vol. 44 No. 9 p. 791-799.

Lankhorst MM, van der Lann MD(1994) Wavelet-based Signal Approximation with Evolutionary algorithms [on-line] ftp://ftp.cs.rug.nl/pub/cs-reports/wavelets.ps.gz Accessed on 24/4/1999

Laskaris N, Bezarianos A, Fotopoulos S, Papathanasopoulos P (1996) A Evolutionary algorithm For the Multidimensional Filtering Process of Visual Evoked Potential Signals *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* p. 1553-1554.

Laskaris N, Fotopoulous S, Papathanasopoulos P, Bezerianos A (1997) Robust moving averages, with Hopfield neural network implementation, for monitoring evoked potential signals , *Electroencephalography and Clinical Neurophysiology* Vol. 104 p. 151-156.

Lewalle J, Peek FW, Murphy SJ (1995) Wavelet Analysis of Olfactory Nerve Responses to Stimulus *Journal of Theoretical Biology* Vol. 177 No. 3 p. 215-236.

Lim LM, Akay M, Doubenspeck JA (1995) Identifying Respiratory-Related Evoked Potentials *IEEE Engineering in Medicine and Biology Magazine* March/April p. 174-178.

Maccabee PJ, Pinkhasov EI, Cracco RQ (1983) Short latency somatosensory evoked potentials to median nerve stimulation effect of low frequency filter *Electroencephalography and Clinical Neurophysiology*Vol 55 p. 34-44.

Maccabee PJ, Hassan NF, Cracco RQ, Schiff JA (1986) Short latency somatosensory evoked potentials: power spectra comparisons between high-pass analog and digital filter *Electroencephalography and Clinical Neurophysiology*Vol 65 p. 177-187.

Maccabee PJ, Hassan NF (1992) AEEM Minimonograph #39: Digital Filtering: Basic Concepts and Applications to Evoked Potentials *Muscle & Nerve* Vol. 15 p. 865-875.

MacLennan AR and Lovely DF (1995) Reduction of evoked potential measurement time by a TMS320 based adaptive matched filter *Med. Eng. Phys.* Vol. 17 (4) p. 248-256.

Makeig S, Jung TP, Bell AJ, Ghahreman D, Sejinowski TJ (1997) Blind Separation of Auditory Event Related Responses into Independent Components *Proceedings of the National Academy of Science USA* Vol. 94 p. 10979-10984.

Mallat SG (1989) A Theory for Multiresolution Signal Decomposition: The wavelet Representation *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 11 No. 7 p. 674-693.

Mallat SG (1998) *A Wavelet Tour of Signal Processing* Academic Press: San Diego

Mallat SG and Zang Z (1993) Matching pursuit with time-frequency dictionaries *IEEE Transactions on Signal Processing* Vol. 41 No. 12 p. 3397-3415.

McNay D, Michielesen E, Rodgers RL, Taylor SA, Akhatari M, Sutherling WW (1996) Multiple Source Localization using Genetic Algorithms *Journal of Neuroscience Methods* Vol. 64 pp 163-172.

Michalewicz Z (1996) *Evolutionary algorithms+Data Structures = Evolution Programs* 3rd Edition, Springer: Berlin

Misiti M, Misiti Y, Oppenheim G, Poggi JM Wavelet Toolbox For Use with MATLAB *The MathWorks Inc.*

Morin E, Scott RN, Parker PA (1987) Characterization and Processing of Surface Recorded Spinal Somatosensory Evoked Potentials *Electroencephalography and Clinical Neurophysiology* Vol. 68 pp.40-44

Muthuswamy J, Thakor NV (1998) Spectral analysis methods for neurological signals *Journal of Neuroscience Methods* Vol. 83 p. 1-14.

Nayak A, Roy RJ (1995) Neural Networks for Predicting Depth of Anesthia From Auditory Evoked Potentials: Comparison of the Wavelet Transform with Autoregressive Modelling and Power Spectrum *Proceedings of the IEEE Engineering in Medicine and Biology 17th Annual Conference* p. 797-800.

Nayak A, Roy RJ (1998) Anaesthesia Control Using Midlatency Auditory Evoked Potentials *IEEE Transactions on Biomedical Engineering* Vol. 45 No. 4 p. 409-421.

Nishida S, Nakamura M, Shibasaki H (1993) Method for single-trial Recording of somatosensory evoked potentials *Journal of Biomedical Engineering* Vol. 15 p. 255-262.

Norcia AM, Sato T, Shinn P, Mertus J (1986) Methods for identification of evoked response in frequency and combined time/frequency domains *Electroencephalography and Clinical Neurophysiology* Vol. 65 p. 212-226.

Nuwer MR, Dawson E (1984) Intraoperative evoked potential monitoring of spinal cord: enhanced stability of cortical recordings *Electroencephalography and clinical Neurophysiology* Vol. 59 p. 318-327

Olson WH Basic Concepts of Medical Instrumentation Ed. Webster JG *Medical Instrumentation Application and Design* 2nd Edition p. 10-11, Houghton Mifflin Co: Boston

Paradiso C, De Vito L, Rossi S, Setacci C, Battisini N, Cioni R, Passero S, Giannini F, Rossini PM (1995) Cervical and scalp recorded short latency somatosensory evoked potentials in response to epidural spinal cord stimulation in patients with peripheral vascular disease *Electroencephalography and Clinical Neurophysiology* Vol. 96 p. 105-113.

Parker PA, Gopalan R (1987) Evoked Potential Signal and System Nonlinearity *IEEE Transactions on Biomedical Engineering* Vol. 34 No.10 p. 771-778.

Parsa V, Parker PA (1994) Multireference Adaptive Noise Cancellation Applied to Somatosensory Evoked Potentials *IEEE Transactions on Biomedical Engineering* Vol. 41 No. 8 p. 792-800.

Parsa V, Parker PA, Scott RN (1998) Adaptive Stimulus Artifact Reduction in Noncortical Somatosensory Evoked Potential Studies *IEEE Transactions on Biomedical Engineering* Vol. 45 No. 2 p. 161-179.

Quiroga RQ (2000) Obtaining single stimulus evoked potentials with Wavelet Denoising [on-line] http://www.physio.mu-luebeck.de/user/rq/denoising1.ps.gz accessed on 3/7/2000.

Rossini PM, Cracco RQ, Cracco JB, House WJ (1981) Short latency somatosensory evoked potentials to peroneal nerve stimulation: scalp topography and the effect of different frequency filters *Electroencephalography and Clinical Neurophysiology* Vol. 52 p. 540-552.

Rossini PM, Basciani M, DiStefano E, Febbo A, Mercuri N (1985) Short Latency scalp somatosensory evoked potential and central spine to scalp propagation characteristics during peroneal and median nerve stimulation in multiple sclerosis *Electroencephalography and clinical Neurophysiology* Vol. 60 p. 197-206.

Saatchi Mr, Gibson C, Rowe JWK, Allen EM (1997) Adaptive Multiresolution Analysis Based Evoked Potential Filtering *IEE Proceedings on Science, Measurement and Technology* Vol. 144 No. 4 pp. 149-155.

Samar VJ, Swartz KP, Raghuveer (1995) Multiresolution Analysis of Event-Related Potentials by Wavelet Decomposition *Brain and Cognition* Vol. 27 p 398-438.

Samar VJ, Begleiter H, Chapa JO, Raghuveer MR, Orlando M, Chorlian D (1996) Matched Meyer Neural Wavelets for Clinical and Experimental Analysis of Auditory and Visual Evoked Potentials *Signal Processing: VIII Theories and Applications* p. 387-390.

Samar VJ, Bopardikar, Raghuveer R, Swartz K (1999) Wavelet Analysis of Neuroelectric Waveforms: A conceptual Tutorial *Brain and Language* Vol. 66 p. 7-60.

Schafer RE, Small GW, Arnold A (1996) Evolutionary algorithm-Based Protocol for Coupling Digital Filtering and Partial Least-Squares Regression: Application to the Near-Infrared Analysis of Glucose in Biological Matrices *Analytical Chemistry* Vol. 68 p. 2663-2675.

Strang G and Nguyen T (1997) *Wavelets and Filter Banks* Wellesley-Cambridge Press Revised Edition.: Wellesley, MA

Thakor NV (1987) Adaptive filtering of Evoked Potentials *IEEE Transactions of Biomedical Engineering* Vol. 32 N0. 1 p. 7-12.

Thakor NV, Guo X, Vaz RA, Laguna P, Jane R, Caminal P, Rix H, Hanley DF (1993a) Orthonormal (Fourier and Walsh) Models of Time-varying Evoked Potentials in Neurological Injury *IEEE Transactions of Biomedical Engineering* Vol. 40 No. 3 p. 213-221.

Thakor NV, Xin-rong G, Yi-Chun S, Hanley DF (1993b) Multiresolution Wavelet Analysis of Evoked Potentials *IEEE Transactions of Biomedical Engineering* Vol. 40 No.11 p. 1085-1094.

Tian J, Juhola M, Gronfors T (1997) Latency Estimation of Auditory Brainstem Response By Neural Networks *Artificial Intelligence in Medicine* Vol. 10 p. 115-128.

Walter DO (1969) A Posteriori Wiener Filtering of Averaged Evoked Responses *Electroencephalography and Clinical Neurophysiology* suppl 27 p. 61-70.

Wastell DG (1979) The application of low-pass filtering to evoked potential data: Filtering without phase distortion *Electroencephalography and Clinical Neurophysiology* Vol. 46 p. 355-356.

Wastell DG (1981) When wiener filtering is less than optimal: An illustrative application to the brain stem evoked potential *Electroencephalography and Clinical Neurophysiology* Vol. 51 p. 678-682.

de Weerd JPC, Kap JI (1981a) *A Posteriori* Time-varying Filtering of Averaged Evoked Potentials I Introduction and Conceptual Basis *Biological Cybernetics* Vol. 41 p. 211-222.

de Weerd JPC, Kap JI (1981b) *A Posteriori* Time-varying Filtering of Averaged Evoked Potentials I Mathematical and Computational Aspects *Biological Cybernetics* Vol. 41 p. 223-234.

de Weerd JPC, Kap JI (1981c) Spectro-temporal representation and time-varying spectra of evoked potentials *Biological Cybernetics* Vol 41 p 101-117

Wiener N (1949) *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications* cited by Proakis JG and Manolakis DG (1996) *Digital Signal Processing Principles, Algorithms and Applications* Prentice-Hall:New Jersey 3rd Edition pp. 715.

Yu KB, McGillem CD (1983) Optimum filter for estimating evoked potential waveforms *IEEE Transactions on Biomedical Engineering* Vol 30 No 11 p 730-373.

Yu XH, Zhang YS, He ZY (1994) Peak component latency-corrected average method for evoked potential waveform estimation *IEEE Transactions on Biomedical Engineering* Vol. 41 No. 11 p. 1072-1082.

Zhang Q and Benveniste A (1992) Wavelet Networks *IEEE Transactions on Neural Networks* Vol. 2 p. 889-898