



139
192
THS

This is to certify that the
thesis entitled

Methods for Neural Signal Processing and Analysis

presented by

Yasir Suhail

has been accepted towards fulfillment
of the requirements for the

M.S. degree in Electrical Engineering


Major Professor's Signature

08/23/05
Date

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE
JAN 12 2013 FEB 12 2013		

Methods for Neural Signal Processing and Analysis

By

Yasir Suhail

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

2005

ABSTRACT

Methods for Neural Signal Processing and Analysis

By

Yasir Suhail

High-density microelectrode arrays offer indispensable tools for neuroscience research and potential benefits for neuroprosthetic applications at the micro and nano scales. Advances in microfabrication and microelectronics have enabled integration of large number of electrodes on a single device with modest signal processing capabilities, which in turn triggered many new scientific discoveries in the neuroscience community. From a system viewpoint, these rapid developments present numerous challenges to the associated data processing and analysis stages. In this work, we present algorithms and performance analyses of a neural interface system designed for large scale processing of the neural data obtained. We evaluate the performance and tradeoffs of the system's computational complexity versus data precision and signal fidelity during real-time telemetry transmission. We also evaluate performance tradeoffs between single and multiple channel signal detection techniques. On the data mining aspect, we present a new multiscale processing and clustering technique to infer functional interdependency between populations of neurons from the recorded mixture to ease the identification of local and global biological neural circuits.

Copyright © by

Yasir Suhail

2005

To my parents and my sister

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis advisor Dr. Karim G. Oweiss for his constant guidance, support and encouragement throughout my two years of graduate study. I must credit him with finding time whenever it was needed for discussing any results and progress. His support has been indispensable not only for this work, but also for my graduate study and career.

I also wish to record the support and camaraderie of my lab mates and fellow graduate students, notably Kyle Thomson, Lance Slifka, and April Thompson which made it easier to get through the work and enjoy it. I have benefitted greatly from the company of many people at Michigan State, most notably Professors Hayder Radha and Selin Aviyente who have taught me most of what I know of Signal Processing, Information Theory, and Statistics. Professor Rong Jin has been very helpful with his collaboration in the area of probabilistic clustering. Michigan State University hosts one of the friendliest people to learn from and spend time with. A diverse community of wonderful students within the Department and the University in general have been a bedrock of support and have made my stay memorable.

I would not have been able to achieve any of this without the unconditional love, support, and good wishes of my parents and my little sister. I do not have the

words to express my feelings and appreciation to them. Most of all, I thank the Almighty for bestowing on me these blessings so I could comfortably engage myself in the pursuit of knowledge that I have enjoyed so much.

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	x
1 Wavelet Transform Block for Implantable Neural Interfaces	1
1.1 Introduction	1
1.2 Theory	4
1.2.1 The Traditional Mallat's Algorithm	4
1.2.2 The Lifting Approach	5
1.2.3 Integer Lifting DWT	8
1.2.4 Quantization of the Filter Coefficients	8
1.2.5 Wavelet Thresholding	9
1.2.6 Computation and Data Rate Savings	12
1.3 Methodology	12
1.4 Results	13
1.5 Conclusion	17
2 Spike Detection	19
2.1 Introduction	19
2.2 Data model	21
2.3 Wavelet domain detection	22
2.3.1 The Stationary Wavelet Packet Tree	23
2.4 Analysis of the Bayesian Tests for Spike Detection	29
2.4.1 Likelihood Ratio Test for the known signal case	29

2.4.2	Likelihood Ratio Test for the unknown signal case	36
2.5	Results	44
2.6	Conclusions	49
3	Clustering Neural Spike Trains	51
3.1	Introduction	51
3.1.1	Background	52
3.2	Simulation Method	54
3.2.1	Proposed Technique	55
3.3	Multi-resolution Approach	56
3.4	A Principal Components Approach for Multi-scale Clustering	65
3.5	Graph Theoretic Clustering	66
3.6	Clustering Results	69
3.7	Conclusions	74
	BIBLIOGRAPHY	76

LIST OF TABLES

1.1	Update and Prediction filter coefficients for <i>symmlet 4</i>	8
2.1	Signal to Noise Ratio in the different SWPT nodes.	29
3.1	Eigenvalues associated with the Principal Components	66
3.2	Clustering Errors obtained in the k-means algorithm	72

LIST OF FIGURES

1.1	The basic analysis block of the pyramidal algorithm proposed by Mallat[10] for 3 decomposition levels.	5
1.2	Lifting scheme for computing a single level of DWT decomposition.	6
1.3	Comparison of denoising performance for various precisions of filter coefficients for a sample channel of a 4-channel array with the data quantized to 10 bits.	14
1.4	Denoising performance for various wavelet transform computations.	15
1.5	Performance in the distance metric of the signal subspace for data denoised using various wavelet transform computations.	15
1.6	Performance in the distance metric of the signal subspace for data denoised using various wavelet transform computations.	16
1.7	Example spike waveforms obtained with DWT computation with different filter coefficient bit widths.	16
1.8	Effect of round off and quantization errors on the signal fidelity as a function of multiplier complexity.	17
2.1	The Stationary Wavelet Packet Tree.	25
2.2	A single channel represented in different SWPT nodes.	28
2.3	Receiver operating Characteristics for Array and single channel detection.	41
2.4	Receiver operating Characteristics for Array and single channel detection.	43
2.5	ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 7.6024.	45

2.6	ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 8.7623.	45
2.7	ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 10.857.	46
2.8	ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 8.1177.	46
2.9	ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 6.9579.	47
2.10	ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 10.2124.	47
2.11	Noisy data on one channel used to detect spikes.	48
2.12	The clean spike data to be detected.	48
2.13	A clean and noisy spike example.	49
3.1	The Haar basis approximation function.	57
3.2	The Haar basis detail function.	58
3.3	The Wavelet Tree representation to 5 levels.	59
3.4	Correlations in different subbands.	64
3.5	Principal Components of correlations.	67
3.6	Fused profile of the first 4 principal components.	70
3.7	The true clusters depicted in the first two components resolved from the fused matrix D	71
3.8	Result of the k-means algorithm applied to the first two principal components of the fused matrix.	73

CHAPTER 1

Wavelet Transform Block for Implantable Neural Interfaces

1.1 Introduction

Electrical recordings from neural tissue give direct observations of the neural activity. Compared to Electroencephelogram (EEG) recordings or chemical detection of neurotransmitters and neuromodulators, invasive recordings from electrodes placed inside the neural tissue provide data with high temporal (of the order of 30 kHz) and spatial (micron range) resolution. Cellular recordings from neurons have been used to study the neural systems such as the motor, somatosensory, and visual cortical areas of the brain.

With the advances in micro fabrication, it is possible to manufacture dense

arrays of microelectrodes[1]. These arrays can record from a large number of neurons and provide high volumes of data for further analysis. For artificially grown neural tissue, it is possible to have relatively long term cellular recordings from individual neurons through techniques such as patch clamping[2] and directed neural growth[3]. In-situ recordings from animals, however is more complicated for various reasons. The recorded data is corrupted to a large extent with noise from the neighboring tissue, and the signal conditions evolve over time. Another problem is the actual transmission and processing of this information. The traditional technique relies on amplifiers situated on the recording probe relaying the recorded signals over wires to a bank of filters, analog to digital converters, and a computer for signal processing. This complicates things for experiments in live, freely behaving animals. It is not always possible to have bulky wired connections to a large data processing system. The opportunities for clinical neural prosthetic devices also present similar challenges. These devices have the potential to help patients with motor disorders and spinal cord injuries lead more normal lives.

Our motivation is the development of on-chip signal processing and wireless data transmission for high density, implantable neural recording probes. Researchers have tried to move towards wireless telemetry[4] to make the neural probes less obtrusive and suitable for both long term behavior experiments and prosthetic devices. It is clear that complex signal processing is required on this data for compression, detection of action potentials and local field potentials, separation of neural sources, and decoding the neural signals. Passive probes transmitting all the

data wirelessly face a deluge of data. A modestly sized array with 96 electrodes recording at a sampling rate of 25kHz per channel to a precision of 12bits/sample, required bit rate would be 29 Mbps. Further complicating this problem is that the biological environment places stringent limits on the size and power availability for such devices.

We focus on the development of on-chip signal processing prior to data transmission to extract some valuable information and reduce the telemetry bandwidth requirements. Studies have shown that on-chip processing may reduce the data throughput to approximately 75% for bursting activity and 85-90% for spontaneous activity and reasonable bit rates are achieved for a 32 channel device sampling at 25 kHz, with 8 bit sample quantization[5]. This technique relied heavily on the Discrete Wavelet Transform (DWT)[6].

The Discrete Wavelet Transform is also the basis of many algorithms for spike detection[7], noise reduction[8], spike sorting[9] of multichannel microelectrode recordings, including those presented in this thesis. On-chip DWT computation of the recorded data will be an essential component of any implantable electrode recording array with intelligent wireless transmission of data. With the stringent power and area limitations imposed on the system, a suitable implementation of the DWT is paramount. The use of the *lifting* technique for DWT computation results in considerable savings in computational hardware, memory, and bandwidth requirements.

1.2 Theory

The compaction property of the wavelet transform, which expresses the signal in a *few* larger coefficients while expressing the noise in *many* small coefficients is used in most of the algorithms for neural signal processing. Discarding the smaller coefficients results in denoising[8]. Also, the fact that the signal of interest (neural spikes) occupy a small number of coefficients is beneficial for compression strategies.

1.2.1 The Traditional Mallat's Algorithm

The traditional Mallat's algorithm[10] for evaluating the wavelet transform of a given signal involves recursively convolving the signal through two decomposition filters h and g , and decimating the result to obtain the approximation and detail coefficients at every decomposition level j as illustrated in Fig.1.1.

These filters l and h are the low pass scaling function and high pass wavelet functions respectively. The approximation and detail coefficients at any arbitrary level are given by

$$\begin{aligned} a^j(i) &= \sum_{k=0}^{K-1} a^{j-1}(2i-k)l_k \\ d^j(i) &= \sum_{k=0}^{K-1} a^{j-1}(2i-k)h_k \end{aligned} \tag{1.1}$$

where K is the length of the filters, also known as the support of the wavelet

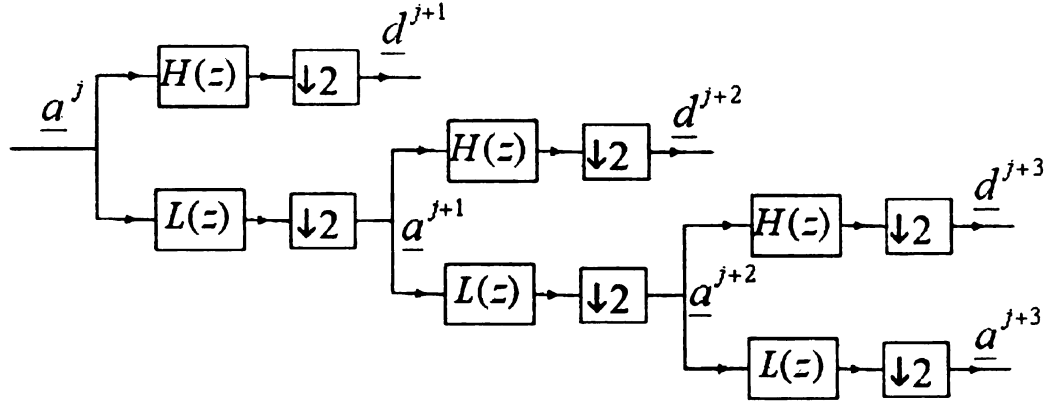


Figure 1.1. The basic analysis block of the pyramidal algorithm proposed by Mallat for 3 decomposition levels.

function.

1.2.2 The Lifting Approach

From the computational standpoint, the lifting scheme for DWT computation has shown numerous benefits over traditional DWT computation[11]. It is a fast and efficient mechanism for implementing the DWT. For asymptotically long filters, the computational cost of the lifting algorithm is one half of the cost of the standard algorithm, and it requires only *in-place* computations. It can also be adapted to a lossless integer-to-integer transform which does not require any floating point computations. The lifting scheme relies on factorizing the l and h filters to obtain

the filters S_N and T_N as

$$\begin{aligned}
 P(z) &= \begin{bmatrix} \frac{h(\sqrt{z}) + h(\sqrt{-z})}{2} & \frac{l(\sqrt{z}) + l(\sqrt{-z})}{2} \\ \frac{h(\sqrt{z}) - h(\sqrt{-z})}{2} & \frac{l(\sqrt{z}) - l(\sqrt{-z})}{2} \end{bmatrix} \\
 &= \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} 1 & S_N(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_N(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & S_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_1(z) & 1 \end{bmatrix}
 \end{aligned} \tag{1.2}$$

In this scheme, there is no need to compute the two different discrete convolutions for the whole data record. The memory required is the same as that required to store the data, while Mallat's algorithm requires twice as much memory as the initial data to compute the wavelet transform. In addition to these savings, an integer approximation of the data and quantization of the filter coefficients further reduces the computational load.

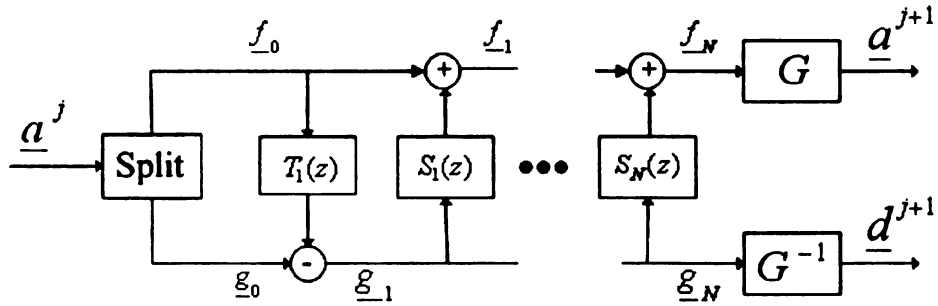


Figure 1.2. Lifting scheme for computing a single level of DWT decomposition.

The lifting scheme of Fig.1.2 replaces one stage of the Mallat's algorithm shown

in Fig.1.1. It is based on three steps: *splitting* the data into even and odd samples, *predicting* the odd samples from the even samples, and *updating* the even samples to obtain the approximation coefficients.

For our hardware implementation with strict power and area constraints, we consider an integer to integer wavelet transform[12]. Here, the first factor in Eq. (1.3) is discarded, and the realized transform is in fact a scaled version of the original wavelet and scaling functions. The *symmlet4* wavelet basis has been found suitable for processing of neural recordings[9]. Factoring the scaling and wavelet functions for this basis results in the following steps to be used in the scheme of Fig.1.2.

$$\begin{aligned}
\text{step1 : } g_1[i] &= \lfloor g_0[i] + C_1 f_0[i] \rfloor \\
\text{step2 : } f_1[i] &= \lfloor f_0[i] + C_2 g_1[i] + C_3 g_1[i + 1] \rfloor \\
\text{step3 : } g_2[i] &= \lfloor g_1[i] + C_4 f_1[i] + C_5 f_1[i - 1] \rfloor \\
\text{step4 : } a[i] &= \lfloor f_1[i] + C_6 g_2[i] + C_7 g_2[i - 1] \rfloor \\
\text{step5 : } d[i] &= \lfloor g_2[i] + C_8 a[i + 1] \rfloor
\end{aligned} \tag{1.3}$$

where the input registers $g_0[i]$ and $f_0[i]$, are the pair of even and odd samples of the signal, $a[i]$ and $d[i]$ are a pair of approximation and detail coefficients computed, and $\lfloor . \rfloor$ denotes the floor function. Note that we truncate the result at every stage, since we wish to implement the integer version of the transform.

The constants $C_1, C_2, \dots, C_8$, which are the coefficients of the filters $T_n(z)$ and

Table 1.1. Update and Prediction filter coefficients for *symmlet 4*.

Coefficient	Value	Coefficient	Value
C_1	0.39114	C_5	0.16203
C_2	-0.12439	C_6	0.43128
C_3	-0.33924	C_7	0.43128
C_4	-1.41951	C_8	-1.04925

$S_n(z)$ are given in Table 1.1.

1.2.3 Integer Lifting DWT

The data recorded at any channel is scaled and quantized to obtain data samples within a 1-bit integer precision. The DWT is then computed using the lifting scheme described in Eq. (1.3) to the required number of decomposition levels. The last step (of the corresponding floating point lifting) of multiplication by the constants K and K^{-1} (in Eq 1.3) is omitted. Hence, the dynamic range of the transform at each level will change by K .

1.2.4 Quantization of the Filter Coefficients

The filter coefficients C_1 through C_8 have real values given in Table 1.1. However, since we are working with 10-bit integer data, we wish to remove all the floating point operations from the DWT computation strategy. The filter coefficients are therefore quantized to a fixed number of bits. We have experimented with 4, 6, 8, 10, and 12-bit precision of the lifting factorized coefficients. Quantization to a

small number of bits results in a significant reduction in the number of additions and shift operations required for each multiplication in Eq. (1.3).

1.2.5 Wavelet Thresholding

The DWT representation of neural signals is suited for noise reduction as the signal components are compactly represented by few large amplitude coefficients because of the inherent similarity between the wavelet basis functions and the spike waveforms[13], and the noise is spread across all decomposition levels in many small amplitude coefficients. This allows thresholding[14] the small coefficients to take place and simultaneously reduces the number of coefficients to code for extracutaneous transmission. The coefficients that are thresholded can be *run-length* encoded to give significant savings in bandwidth[5].

We formulate a measures to quantitatively assess the performance of our DWT module for typical neural signal processing problems. The Signal to Noise Ratio of the recorded data at any stage of processing is defined as

$$\text{SNR (dB)} = 20 \log_{10} \frac{\text{Average Spike Peak to Peak Amplitude}}{\text{Noise RMS}}. \quad (1.4)$$

This definition of the SNR is relevant for spike detection in noisy data as discussed in Chapter 2.

We also wish to evaluate a measure of neural source separability in multiunit

array recordings. The wavelet based MASSIT algorithm[9] for spike sorting relies on the invariance of the signal subspace in the “best” subbands of the DWT expansion. The approximations performed in the DWT module design should not compromise the accuracy of the spike sorting stage.

To compute the eigenvectors for the MASSIT spike sorter, we model the observed spike waveform across multiple channels as

$$\mathbf{Y} = \mathbf{X} + \mathbf{Z} \quad (1.5)$$

where $\mathbf{Y} \in \mathfrak{R}^{M \times N}$ denotes the observed spike in the interval $[1, \dots, N]$ across an array of M channels, $\mathbf{X} = \mathbf{a_p s_p^T}$ is the noise free observed spike across the array such that the column vector $\mathbf{a_p} \in \mathfrak{R}^{M \times 1}$ describes the array response to the spike waveform $\mathbf{s_p} \in \mathfrak{R}^{N \times 1}$ elicited by neuron p , and $\mathbf{Z} \in \mathfrak{R}^{M \times N}$ denotes a zero mean additive noise component with a non-diagonal spatial covariance matrix $\mathbf{R_Z} \in \mathfrak{R}^{M \times M}$.

The column vector $\mathbf{a_p}$ spans the same subspace as the first eigenvector $\mathbf{u}_1$ of the spatial covariance matrix of $\mathbf{Y}$, computed as

$$\mathbf{R_Y} = E[\mathbf{Y Y}^T] = \mathbf{U_Y D_Y U_Y^T} = \sum_{m=1}^M \delta_m \mathbf{u}_m \mathbf{u}_m^T \quad (1.6)$$

Therefore, a qualitative assessment of the neural source separability is a distance metric between the eigenvector obtained with floating-point standard Mallat DWT

algorithm and the eigenvector obtained using fixed-point integer lifting DWT computation. The distance between both vectors, in a mean square sense is computed as

$$d_p = \|\mathbf{u}_1 - \mathbf{a}_p\|^2 \quad (1.7)$$

where $\|\cdot\|$ is the Euclidean norm.

This distance metric computed after thresholding in the Integer Lifting DWT module, and the regular DWT module provides a qualitative measure of the performance in spike sorting.

The distortion in the reconstructed spike waveform due to the quantization and approximation errors in the DWT module is also considered. Faithful reconstruction of the spike waveform is essential for all spike sorting algorithms, whether for single channel[15] or array data[9]. We compute the distortion as

$$\text{Distortion SNR (dB)} = 10 \log_{10} \frac{\mathbf{X}\mathbf{X}^T}{(\mathbf{X} - \mathbf{Y}_r)(\mathbf{X} - \mathbf{Y}_r)^T} \quad (1.8)$$

where $\mathbf{Y}_r \in \mathbb{R}^{1 \times N}$ is the reconstructed signal after wavelet thresholding for the DWT module, and $\mathbf{X} \in \mathbb{R}^{1 \times N}$ is the true neural signal for any single channel.

This distortion metric can be evaluated for the Integer Lifting DWT module with arbitrary data and filter quantization, and compared to the “ideal” distortion obtained with the traditional floating point DWT module.

1.2.6 Computation and Data Rate Savings

The lifting scheme for DWT computation requires less computation than the traditional Mallat’s algorithm, even without any data or filter quantization. For the *symmlet4* orthogonal wavelet with a filter length of eight ($|h| = 8, |l| = 8$), the number of floating point operations (*flops*) required for the standard algorithm is 34, while the lifting algorithm requires only 20 *flops* [11].

Further, the quantization of the data and filter coefficients $C_1, C_2, \dots, C_8$ reduce the complexity of the multipliers used in the lifting scheme computations (Eq. (1.3)). If the data is quantized to N_d bits and the lifting filter coefficients are quantized to N_f bits, the product will be truncated to N_d bits. Hence, we estimate the total number of 1-bit addition steps required for one multiplication in the lifting module to be $N_d \times N_f$. The required precision of the data is directly proportional to the data rate needed for transmission or storage on on-chip buffers.

1.3 Methodology

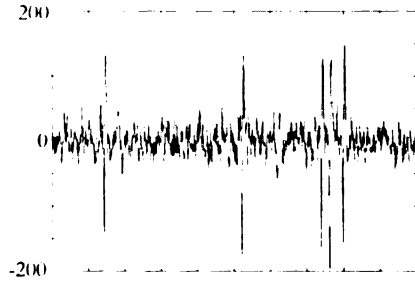
A template spike waveform from a single neural source was used to generate a simulated spike train across a four channel array. The inter-spike interval times were realizations of an exponentially distributed random process. Experimental neural noise from a closely spaced array recording was superimposed on the spike train. The noise level was varied to obtain various SNR conditions.

To evaluate the expected performance in source separation, we used Monte Carlo simulations for various choices of mixing matrices for different single units. The dominant eigenvectors were computed from the denoised data and their distance metric was evaluated from the original mixing matrix $\mathbf{A}$. The distortion in the spike waveform in an individual channel as a function of the multiplier complexity and transmission data rate are also plotted

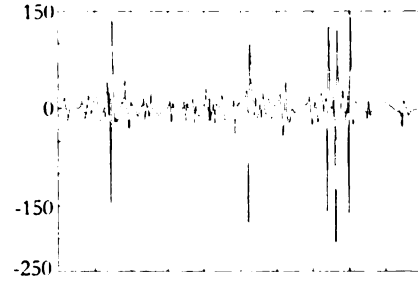
1.4 Results

Figure 1.3(a) illustrates the noisy signal, and its denoised versions using the traditional floating point DWT, Integer Lifting DWT with 10-bit filter coefficients, and Integer Lifting DWT with 4-bit filter coefficients. Figure 1.4 illustrates the denoising performance comparison of the floating point DWT, Integer Lifting DWT with 10-bit and 4-bit filter coefficients on 10-bit data. The interesting result here is that the performance of the Integer Lifting DWT module with 4-bit filter coefficients is close to that of the 10-bit DWT module, which translates into tremendous computational and memory savings.

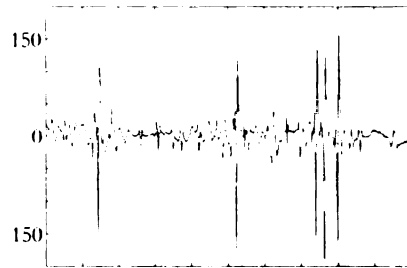
The performance of the source separation distance metric was evaluated and illustrated in Figure 1.6. The plot indicates that the signal subspace is maintained when replacing the floating point DWT to the Integer to Integer Lifting based DWT. Figure 1.7 shows that the reconstructed waveform after denoising for the different cases is very similar. We can locate an optimal operating point for minimum



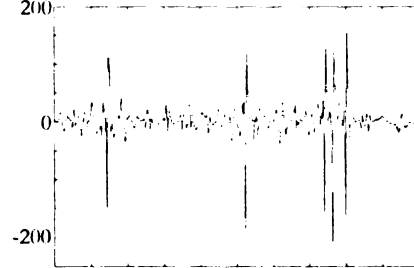
(a) A single channel of noisy data.



(b) Wavelet thresholded data using the traditional DWT



(c) Wavelet thresholded data with Integer Lifting DWT with 10-bit filter coefficients.



(d) Wavelet thresholded data with Integer Lifting DWT with 4-bit filter coefficients.

Figure 1.3. Comparison of denoising performance for various precisions of filter coefficients for a sample channel of a 4-channel array with the data quantized to 10 bits.

computational complexity while giving good signal fidelity from Figure 1.8. We see that using 10-bits to represent the data, and constructing the DWT module with 4-bit filter coefficients gives very little waveform distortion, while huge savings on the hardware are obtained.

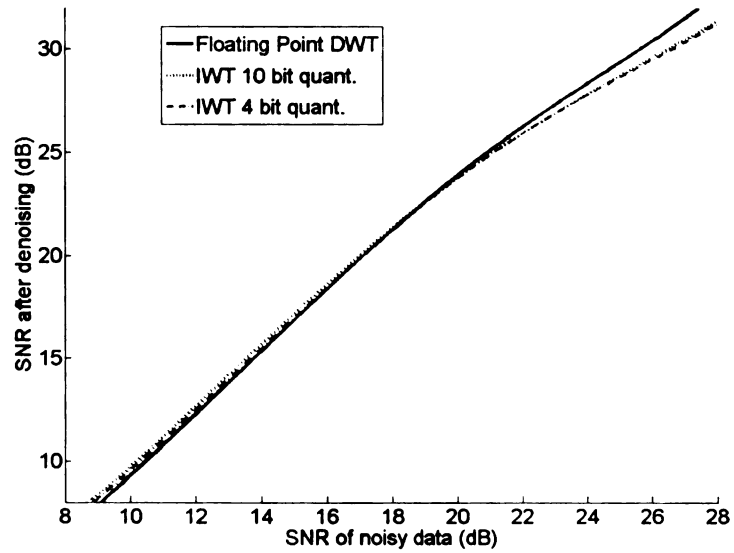


Figure 1.4. Denoising performance for various wavelet transform computations.

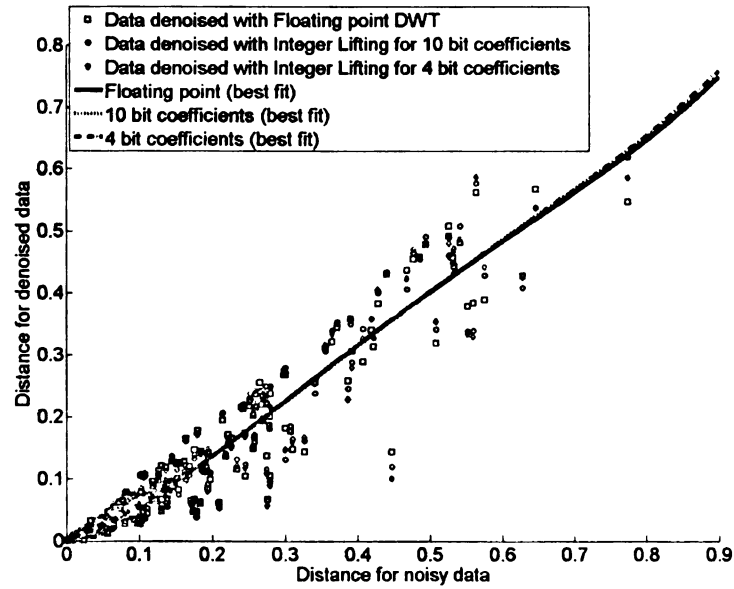


Figure 1.5. Performance in the distance metric of the signal subspace for data denoised using various wavelet transform computations.

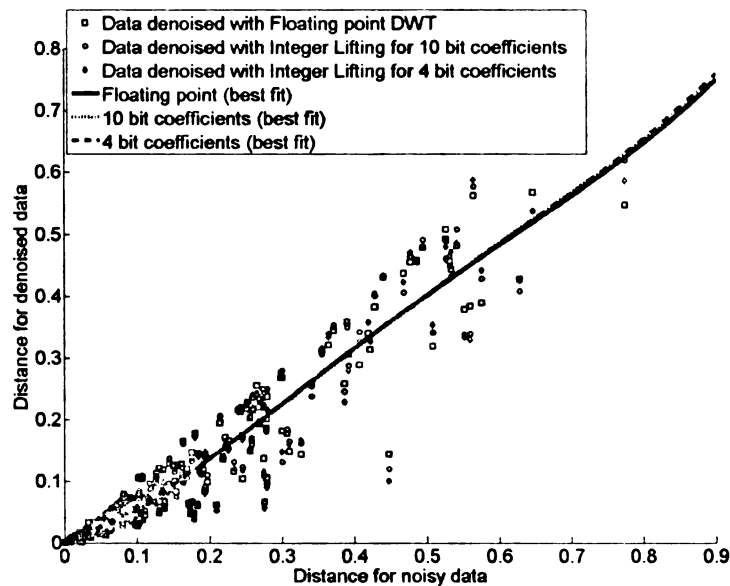


Figure 1.6. Performance in the distance metric of the signal subspace for data denoised using various wavelet transform computations.

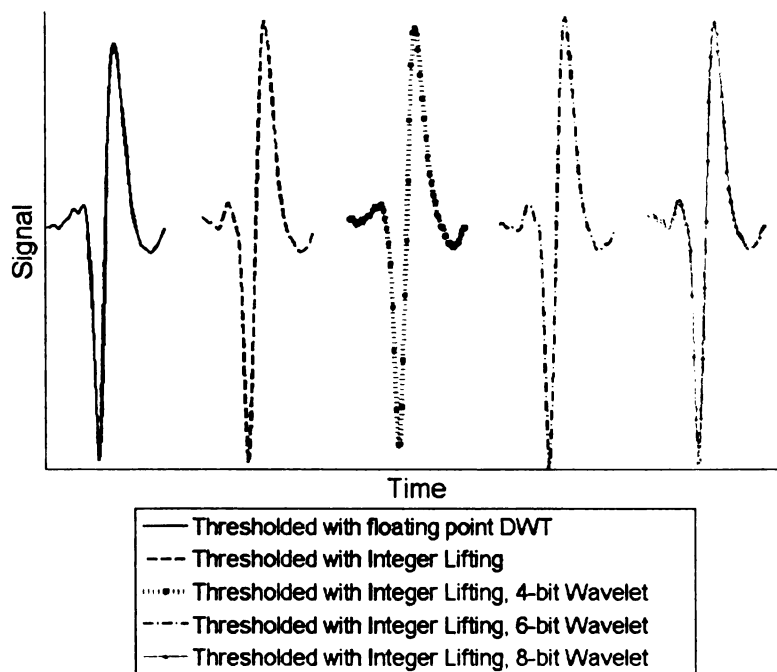


Figure 1.7. Example spike waveforms obtained with DWT computation with different filter coefficient bit widths.

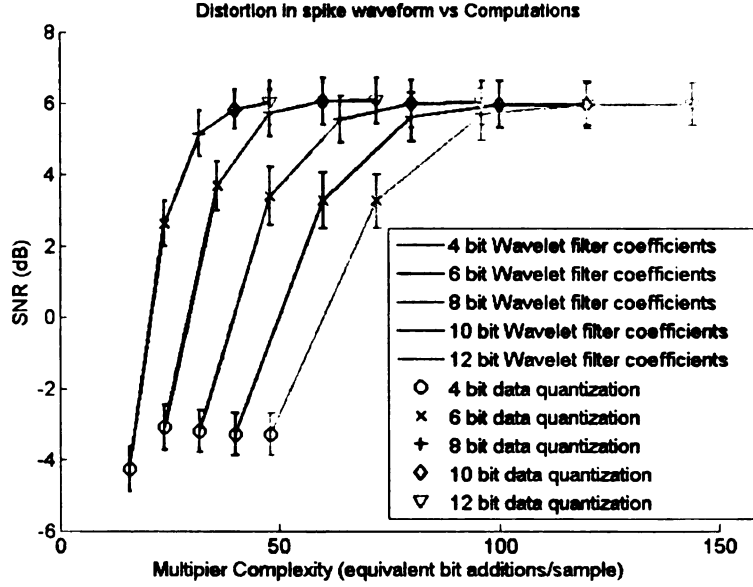


Figure 1.8. Effect of round off and quantization errors on the signal fidelity as a function of multiplier complexity.

1.5 Conclusion

A robust, computationally simple module with anticipated major reduction in chip area and power consumption for implementing the DWT has been demonstrated. The proposed design was motivated by the need to efficiently process neural recordings from high-density implantable neural interface modules in the nervous system. The ability of the DWT to preserve the signal characteristics in the temporal, spectral and spatial domains intuitively makes it a desirable signal processing tool to pipeline the neural data with reduced bandwidth in any associated telemetry system. The anticipated reduction in chip area and power consumption without reducing the performance to a noticeable degree strongly suggests that the proposed

module design can be a key processing stage in the next generation of neuroprosthetic devices.

A circuit design and simulation of the proposed module has indeed confirmed the savings in chip area and power requirements in addition to the versatility of the design for real-time and streaming data processing[16].

CHAPTER 2

Spike Detection

2.1 Introduction

Spike Detection refers to the detection of action potentials generated at the membrane of neurons from electrical signals recorded from neural electrodes. There is generally a noise process from the surrounding neural tissue corrupting the recorded signal. The neural spikes or action potentials are transient signals of small duration that appear as local voltage surges. The wavelet transform appears as a natural way to represent data containing transient signals due to its localization properties in both time and frequency. We have demonstrated use of the wavelet transform to do low level processing on the recorded data for denoising[17] and compression[18]. Therefore, the wavelet transform computations would need to be performed in our framework of a modular design of the prosthetic devices, with the detection algorithms adding very little overhead.

The algorithms presented here detect the spikes through the peaks in signal amplitude occurring at any point over its waveform. For wavelet bases which might be correlated with the signal waveform to some extent, the ratio of the signal peaks associated with the spike to the noise variance increases in some wavelet subbands. The wavelet transform essentially gives a higher SNR for detection in the relevant subbands.

The waveform of spikes generated by a neuron may change over time, and in general cannot be assumed known. For chronic experiments or neural prosthetic devices, active neurons whose action potentials are observed at the electrodes may appear and go due to electrode movement, and changes in neural tissue. For these reasons we assume no knowledge of the spike waveform or amplitude. Nenadic[19] also gives a wavelet based method for spike detection heavily based on Oweiss[20]. However, Nenadic's method implicitly assumes knowledge of the spike waveform by choosing the wavelet function and decomposition levels which are highly correlated with the spike waveform. Further, the estimation of signal strength and signal probability makes the detection susceptible to changes in signal conditions. These assumptions and estimations will run into more problems when action potentials from more than one neural source are to be detected, and the problem becomes coupled with that of estimating the number of sources. Nenadic's approach also relies on a continuous wavelet transform which is highly computationally expensive. Our emphasis is evaluating approaches for a very blind signal detection scenario with very little assumptions about the signal and noise conditions. The work presented

here uses algorithms for single electrode channel and array observations similar to those presented in [20] and derives results for performance in different cases.

2.2 Data model

We assume the following data model for the an M electrode array (Eq. (2.1)).

$$\begin{aligned}\mathbf{Y} &= \mathbf{AS} + \mathbf{Z} \\ &= \mathbf{X} + \mathbf{Z}\end{aligned}\tag{2.1}$$

where $\mathbf{S} \in \mathbb{R}^{P \times N}$ is the neural signal, $\mathbf{A} \in \mathbb{R}^{M \times P}$ is the mixing matrix, $\mathbf{Z} \in \mathbb{R}^{M \times N}$ is the noise and $\mathbf{Y} \in \mathbb{R}^{M \times N}$ is the observed data, N being the number of time samples, and P being the number of neurons.

For simplicity, we assume stationarity and iid distributions at each time sample. Hence, for any time snapshot we get Eq. (2.2).

$$\begin{aligned}\mathbf{y} &= \mathbf{As} + \mathbf{z} \\ &= \mathbf{x} + \mathbf{z}\end{aligned}\tag{2.2}$$

where $\mathbf{s} \in \Re^{P \times 1}$ is the neural signal, $\mathbf{z} \in \Re^{M \times 1}$ is the noise and $\mathbf{y} \in \Re^{M \times 1}$ is the observed data.

$$\mathbf{z} \sim N(\mathbf{0}, \mathbf{R}) \quad (2.3)$$

We assume the noise (Eq. (2.3)) to be temporally white Gaussian, with the spatial correlation structure given by $\mathbf{R} \in \Re^{M \times M}$.

2.3 Wavelet domain detection

The compaction property of the wavelet transform condenses the energy of the signal into a few coefficients, while the noise (assumed white) is distributed across all the nodes. Hence, it should be easier to detect the signals in the wavelet domain.

Our approach is to perform a multi level stationary wavelet packet decomposition of the observed data. The stationary (or undecimated) wavelet transform is performed because the truncated wavelet transform reduces the number of samples at each subsequent level by half. Though it takes less memory or registers, the truncated wavelet transform reduces the temporal resolution for transient signal detection. The wavelet packet tree decomposes both the approximations and details at each level, as opposed to decomposing only the approximations as in a wavelet tree.

2.3.1 The Stationary Wavelet Packet Tree

The Stationary Wavelet Transform seeks to represent the signal in a redundant, shift-invariant basis with uniform time-scale tiling. To aid the understanding of the transform as a matrix operation, consider the following definition of a translation operator[21].

Definition 2.1 Suppose $\mathbf{z}$ is a vector. Define $(R_k z)_n = z_{n-k}$. R_k is a translation operator.

If the low pass filter function (also called the *father wavelet*) be designated as $\mathbf{u} \in \mathbb{R}_{1 \times f}$, and the high pass filter function (also called the *mother wavelet*) designated $\mathbf{v} \in \mathbb{R}_{1 \times f}$, where f is the length of the filters (compact support of the wavelet), then the matrices for the approximations and detail decomposition of a signal of length N at any level can be written as in Eq. (2.4) and Eq. (2.5).

$$\mathbf{W}^{\mathbf{a}} = [R_1 \mathbf{u} \ R_2 \mathbf{u} \ \cdots \ R_N \mathbf{u}] \quad (2.4)$$

$$\mathbf{W}^{\mathbf{d}} = [R_1 \mathbf{v} \ R_2 \mathbf{v} \ \cdots \ R_N \mathbf{v}] \quad (2.5)$$

Here, $\mathbf{W}^{\mathbf{a}} \in \mathfrak{R}_{N \times N}$ is the transformation matrix for the approximations, and $\mathbf{W}^{\mathbf{d}} \in \mathfrak{R}_{N \times N}$ is the transformation matrix for the details. The approximations $\mathbf{s}^{\mathbf{a}} \in \mathfrak{R}_{1 \times N}$ and details $\mathbf{s}^{\mathbf{d}} \in \mathfrak{R}_{1 \times N}$ for the signal $\mathbf{s} \in \mathfrak{R}_{1 \times N}$ can be obtained as in Eq. (2.6).

$$\begin{aligned}\mathbf{s}^{\mathbf{a}} &= \mathbf{s} \mathbf{W}^{\mathbf{a}} \\ \mathbf{s}^{\mathbf{d}} &= \mathbf{s} \mathbf{W}^{\mathbf{d}}\end{aligned}\tag{2.6}$$

Note that at every level of decomposition the amount of data (or memory requirement) is doubled for a stationary wavelet packet tree. The multilevel Stationary Wavelet Packet Tree can be represented as in Fig.2.1.

The node 1 in this tree corresponds to the un-processed time domain signal. We can therefore write the transformation matrix for obtaining the signal at any node as a product of $\mathbf{W}^{\mathbf{a}}$ and $\mathbf{W}^{\mathbf{d}}$. For example, to obtain the wavelet transformed signal at the node 5 we have,

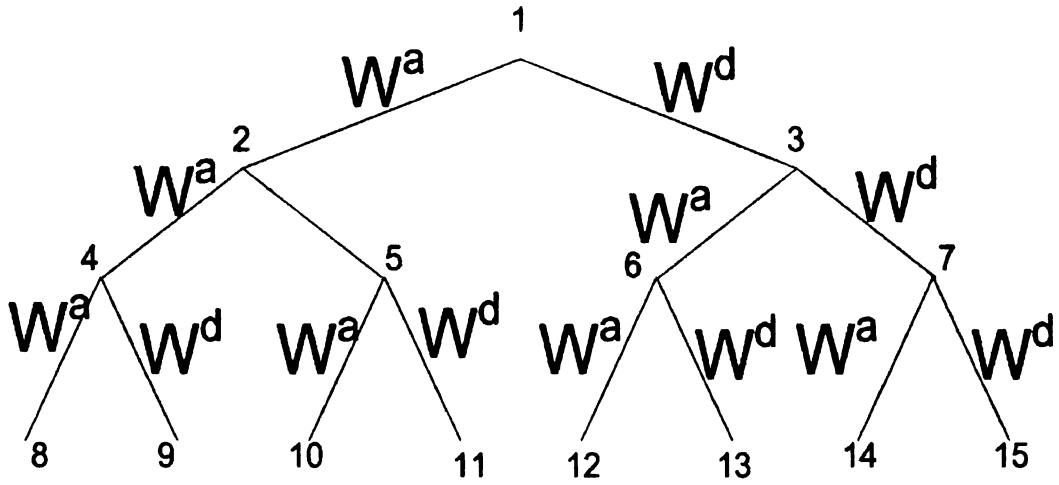


Figure 2.1. The Stationary Wavelet Packet Tree.

$$\begin{aligned}
 s^{(5)} &= s^{(2)} W^d \\
 &= s W^a W^d \\
 &= s W^{(5)} \\
 \Rightarrow W^{(5)} &= W^a W^d.
 \end{aligned} \tag{2.7}$$

Looking back at the data model for the observations, we formulate the wavelet transform equations for the array data

$$\begin{aligned}
\mathbf{Y} &= \mathbf{AS} + \mathbf{Z} \\
&= \mathbf{X} + \mathbf{Z} \\
\mathbf{Y}^{(j)} &= \mathbf{YW}^{(j)} \\
\mathbf{Y}^{(j)} &= \mathbf{XW}^{(j)} + \mathbf{ZW}^{(j)} \\
\mathbf{Y}^{(j)} &= \mathbf{X}^{(j)} + \mathbf{Z}^{(j)}
\end{aligned} \tag{2.8}$$

where $\mathbf{Y}^{(j)} \in \mathbb{R}^{M \times N}$ is the wavelet transform in the j^{th} node of the observations $\mathbf{Y} \in \mathbb{R}^{M \times N}$.

The observations are assumed to be sampled at 20kHz, and since the spike waveforms are usually concentrated at bandwidths less than 10kHz, the first level details (labelled Node 3 in Fig.2.1) and its offspring do not include data of interest. Hence, to save computations the complete wavelet tree is not computed. Rather, only Node 2 and its children are computed. The detection tests are run only on these subbands.

Similar to Eq. (2.2), we describe a snapshot of the data at a single time sample as,

$$\mathbf{y}^{(j)} = \mathbf{x}^{(j)} + \mathbf{z}^{(j)}. \tag{2.9}$$

Here, $\mathbf{y}^{(j)} \in \mathbb{R}_{M \times 1}$ is simply one column of the $\mathbf{Y}$ matrix, corresponding to the time instant of interest. The algorithms considered in this work operate on single time snapshots. The spike detection tests are carried out independently across all the wavelet sub-bands and the decisions are logically OR'ed. For convenience and reducing visual clutter, the subscript corresponding to the node (subband) of the Wavelet Packet Tree will be dropped, and all the data will be understood to be from a particular wavelet subband.

It is shown in the sections on detection performance evaluation, that the detection performance is a function of the Signal to Noise Ratio of a channel. The wavelet transform helps in the detection performance because neural spike waveform, being localized in time would be correlated to some extent with the wavelet functions. This in effect, increases the equivalent SNR at some wavelet subbands. Previous work[8],[17] has shown the *symmlet 4* to be a good wavelet basis for neural spike signals, and this is the wavelet basis used in this study. As an illustration, we have plotted the signal in various nodes of the Stationary Wavelet Packet Tree for a single channel of data containing 4 spikes in Fig.2.2. The red dots denote the instant of the largest absolute peak for a spike, which is to be detected. As we can see, some of the nodes, for example Nodes 8 and 17 are well tuned for capturing the spike energy. The SNR's in the different SWPT nodes for this example are tabulated in Table 2.1.

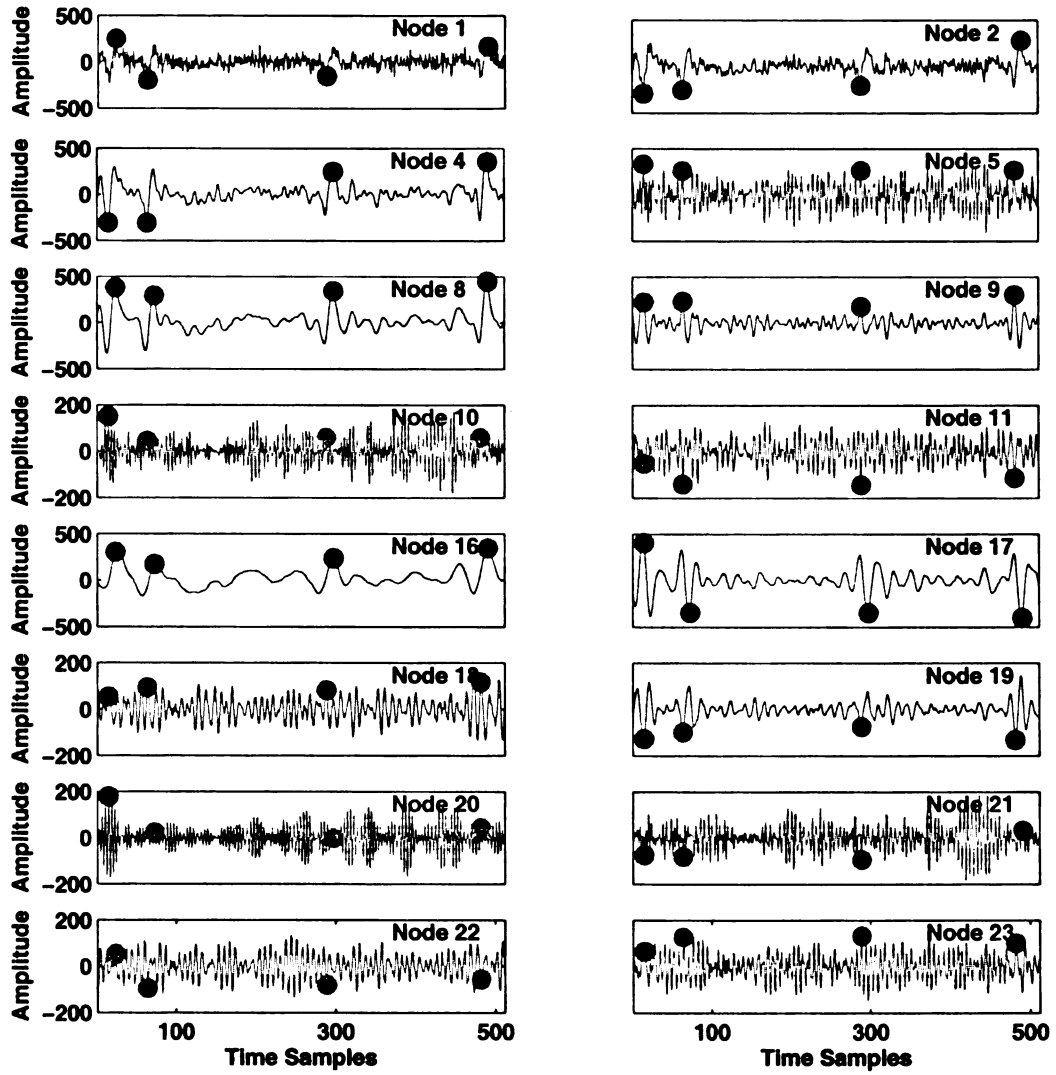


Figure 2.2. A single channel represented in different SWPT nodes.

Table 2.1. Signal to Noise Ratio in the different SWPT nodes.

SWPT Node	Signal to Noise Ratio
1	10.4799
2	13.3108
4	14.8127
5	6.1543
8	15.5377
9	13.8415
10	3.3308
11	7.2197
16	11.9609
17	16.7398
18	5.9863
19	14.3290
20	4.1822
21	1.2984
22	3.0082
23	6.4636

2.4 Analysis of the Bayesian Tests for Spike Detection

2.4.1 Likelihood Ratio Test for the known signal case

For simplicity, we consider a single neural source impinging on an array. In this case, we have a binary hypothesis. Under hypothesis H_0 , no signal is present and $\mathbf{s} = \mathbf{0}$. Under the alternative hypothesis H_1 , the signal (spike) is present and $\mathbf{As} = \mathbf{x} = \boldsymbol{\mu}$.

The LRT then becomes,

$$\frac{\frac{1}{\sqrt{2\pi}|\mathbf{R}|^{-N/2}}e^{\frac{-(\mathbf{y}-\boldsymbol{\mu})^T\mathbf{R}^{-1}(\mathbf{y}-\boldsymbol{\mu})}{2}}}{\frac{1}{\sqrt{2\pi}|\mathbf{R}|^{-N/2}}e^{\frac{-\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y}}{2}}} \leq_{H_0}^{H_1} \eta \quad (2.10)$$

where N is the number of electrode channels and η is the detection threshold.

This reduces to

$$\mathbf{y}^T\mathbf{R}^{-1}\boldsymbol{\mu} \leq_{H_0}^{H_1} \log \eta + \frac{\boldsymbol{\mu}^T\mathbf{R}^{-1}\boldsymbol{\mu}}{2} \equiv \gamma \quad (2.11)$$

We therefore define the Sufficient Statistic T for the array spike detection with Eq. (2.12). For a vector case, this reduces to a matched filter.

$$T = \mathbf{y}^T\mathbf{R}^{-1}\boldsymbol{\mu} \quad (2.12)$$

To compute this Sufficient Statistic, we require the noise covariance $\mathbf{R}$. In the absence of any information about the noise structure (the case in real spike detection applications), it will be estimated from the data. This is simply the sample covariance of the observations (Eq. (2.13)). We will assume this noise covariance estimate for all subsequent expressions.

$$\hat{\mathbf{R}} = \frac{1}{N} \mathbf{Y} \mathbf{Y}^T \quad (2.13)$$

Since the Sufficient Statistic T is a linear function of a Gaussian random variable $\mathbf{Y}$, it is also a Gaussian. We can determine the pdf of the Sufficient Statistic T for the two hypothesis in terms of the means and variances.

$$\begin{aligned} H_0 : \quad E[T] &= E[\mathbf{y}^T \mathbf{R}^{-1} \boldsymbol{\mu}] = 0 \\ E[(T - \bar{T})^2] &= E[(\mathbf{y}^T \mathbf{R}^{-1} \boldsymbol{\mu})^T (\mathbf{y}^T \mathbf{R}^{-1} \boldsymbol{\mu})] = \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \\ H_1 : \quad E[T] &= E[\mathbf{y}^T \mathbf{R}^{-1} \boldsymbol{\mu}] = \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \\ E[(T - \bar{T})^2] &= \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \end{aligned} \quad (2.14)$$

Therefore, we can evaluate the performance of the detection in terms of these numbers.

$$\begin{aligned} P_D &= P_{H_1}(T > \lambda) = \frac{1}{2} \left[1 - Q \left(\frac{\lambda - \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}}{\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \sqrt{2}} \right) \right] \\ P_F &= P_{H_0}(T > \lambda) = \frac{1}{2} \left[1 - Q \left(\frac{\lambda}{\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \sqrt{2}} \right) \right] \end{aligned} \quad (2.15)$$

Correspondingly, we have similar expressions for single channel spike detection too.

$$\begin{aligned} P_D &= P_{H_1}(T > \lambda) = \frac{1}{2} \left[1 - Q \left(\frac{\lambda - \frac{\mu_j^2}{\sigma_j^2}}{\frac{\mu_j^2}{\sigma_j^2} \sqrt{2}} \right) \right] \\ P_F &= P_{H_0}(T > \lambda) = \frac{1}{2} \left[1 - Q \left(\frac{\lambda}{\frac{\mu_j^2}{\sigma_j^2} \sqrt{2}} \right) \right] \end{aligned} \quad (2.16)$$

Hence, for a Gaussian distributed Sufficient Statistic T , the distance metric between the hypotheses is simply $\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}$. For a single channel case, this is simply the energy of the signal divided by the variance of the noise.

To compare the performance of the array detection vs. the single channel detection, we observe the distance measure between the hypotheses in the two scenarios. The array processing will give better performance if the following condition (Eq. (2.17)) holds true.

$$\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} > \frac{\mu_j^2}{\sigma_j^2} \quad (2.17)$$

where μ_j is the strength of the signal at the the channel j where we are detecting the signal, and σ_j is the variance of the noise at that channel.

Hence, we can describe $\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}$, as an equivalent Signal to Noise Ratio for the array case. The observations (array or channel) with a better Signal to Noise Ratio give better detection performance.

Performance Comparison for Spatially Uncorrelated Noise

If we assume the noise covariance to be diagonal (Eq. (2.18)), we can get results on the equivalent SNR's for a single channel and the array data. Eq. (2.19) shows that the equivalent SNR, and hence the performance of a detector using the array data will be better than that working on any single channel.

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_M^2 \end{bmatrix} \quad (2.18)$$

$$\begin{aligned} \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} &= \boldsymbol{\mu}^T \begin{bmatrix} \sigma_1^{-2} & \dots & 0 \\ & \ddots & \\ 0 & \dots & \sigma_M^{-2} \end{bmatrix} \boldsymbol{\mu} \\ &= \sum_{i=1}^M \frac{\mu_i^2}{\sigma_i^2} > \frac{\mu_i^2}{\sigma_i^2} \quad \forall i \in \{1, 2 \dots M\} \end{aligned} \quad (2.19)$$

Performance Comparison for Spatially Correlated Noise

In this case, the noise covariance $\mathbf{R}$ is not diagonal. Since we have already have the expressions for the detection performance in various cases, our purpose here is simply to qualitatively describe the effect of the correlation. For mathematical simplicity, we assume a two channel array. The signal and noise covariances are described in Eq. (2.20).

$$\begin{aligned}\boldsymbol{\mu} &= \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \\ \mathbf{R} &= \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{bmatrix}\end{aligned}\tag{2.20}$$

An inequality condition (Eq. (2.21)) can be derived for the array and channel SNRs. By symmetry, a similar condition holds for channel 2. Hence, we conclude that the array performance is better than a single channel performance.

$$\begin{aligned}\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} - \frac{\mu_1^2}{\sigma_1^2} &= \frac{\sigma_2^2 \mu_1^2 - 2\mu_1 \mu_2 \sigma_1 \sigma_2 \rho + \mu_2^2 \sigma_1^2}{\sigma_1^2 \sigma_2^2 - (\sigma_1 \sigma_2 \rho)^2} - \frac{\mu_1^2}{\sigma_1^2} \\ &= \frac{(\mu_2 \sigma_1)^2 \left(1 - \frac{\mu_1 \sigma_2 \rho}{\mu_2 \sigma_1}\right)^2}{\sigma_1^2 \sigma_2^2 (1 - \rho^2)} > 0\end{aligned}\tag{2.21}$$

We can also try to understand the effect of the noise correlation on the performance. For some fixed signal conditions, and fixed channel noise variances, the array SNR performance improves with the noise correlation coefficient opposite in sign to the signal correlation, and deteriorates when the noise correlation coefficient is of the same sign as the signal.

2.4.2 Likelihood Ratio Test for the unknown signal case

Here, we construct the generalized likelihood ratio test because we do not know the signal waveform or strength. The generalized estimate for the signal therefore is the observed data itself.

$$\frac{\frac{1}{\sqrt{2\pi}|\mathbf{R}|^{-N/2}} e^{\frac{-(\mathbf{y} - \mathbf{y})^T \mathbf{R}^{-1} (\mathbf{Y} - \mathbf{y})}{2}}}{\frac{1}{\sqrt{2\pi}|\mathbf{R}|^{-N/2}} e^{\frac{-\mathbf{y}^T \mathbf{R}^{-1} \mathbf{y}}{2}}} \leq_{H_0}^{H_1} \eta \quad (2.22)$$

This reduces to

$$\mathbf{y}^T \mathbf{R}^{-1} \mathbf{y} \leq_{H_0}^{H_1} 2 \log \eta \quad (2.23)$$

This is essentially a blind energy detector. We therefore define the Sufficient Statistic T for the array spike detection with Eq. (2.24).

$$T = \mathbf{y}^T \mathbf{R}^{-1} \mathbf{y} \quad (2.24)$$

The Sufficient Statistic in this case is χ^2 distributed with degrees of freedom equal to the dimension of $\mathbf{y}$. The χ^2 statistics can be proven if we consider the singular value decomposition of the noise covariance (Eq. (2.25)).

$$\mathbf{R} = \mathbf{U}\mathbf{D}\mathbf{U}^T \quad (2.25)$$

The Sufficient Statistic can thus be expressed as the norm of the whitened observations $\tilde{\mathbf{y}}$.

$$\begin{aligned} T &= \mathbf{y}^T \mathbf{R}^{-1} \mathbf{y} \\ &= \mathbf{y}^T (\mathbf{U}\mathbf{D}\mathbf{U}^T)^{-1} \mathbf{y} \\ &= (\mathbf{D}^{-1/2} \mathbf{U}^{-1} \mathbf{y})^T (\mathbf{D}^{-1/2} \mathbf{U}^{-1} \mathbf{y}) \\ &= \tilde{\mathbf{y}}^T \tilde{\mathbf{y}} \end{aligned} \quad (2.26)$$

Now, $\tilde{\mathbf{y}} = \mathbf{D}^{-1/2} \mathbf{U}^{-1} \mathbf{y}$, being a linear relation on a Gaussian Random variable $\mathbf{y}$, is also Gaussian. The norm of a multidimensional Gaussian is χ^2 distributed, and hence the Sufficient Statistic (Eq. (2.26)) for unknown signal detection in an array is χ^2 distributed. We can calculate some statistics for the two hypotheses (Eq. (2.27)) in terms of the whitened observations.

$$\begin{aligned}
H_0 : \quad \bar{\bar{\mathbf{Y}}} &= E \left[\tilde{\mathbf{Y}} \right] &= E \left[\mathbf{D}^{-1/2} \mathbf{U}^{-1} \mathbf{y} \right] \\
& &= \mathbf{0} \\
E \left[(\tilde{\mathbf{y}} - \bar{\bar{\mathbf{y}}})(\tilde{\mathbf{y}} - \bar{\bar{\mathbf{y}}})^T \right] &= \mathbf{I} \\
\bar{\bar{\mathbf{y}}}^T \bar{\bar{\mathbf{y}}} &= \mathbf{0}^T \mathbf{0} &= 0
\end{aligned} \tag{2.27}$$

$$\begin{aligned}
H_1 : \quad \bar{\bar{\mathbf{y}}} &= E [\tilde{\mathbf{y}}] &= E \left[\mathbf{D}^{-1/2} \mathbf{U}^{-1} \mathbf{y} \right] \\
& &= \mathbf{D}^{-1/2} \mathbf{U}^{-1} \boldsymbol{\mu} \\
E \left[(\tilde{\mathbf{y}} - \bar{\bar{\mathbf{y}}})(\tilde{\mathbf{y}} - \bar{\bar{\mathbf{y}}})^T \right] &= \mathbf{I} \\
\bar{\bar{\mathbf{y}}}^T \bar{\bar{\mathbf{y}}} &= (\mathbf{D}^{-1/2} \mathbf{U}^{-1} \boldsymbol{\mu})^T (\mathbf{D}^{-1/2} \mathbf{U}^{-1} \boldsymbol{\mu}) = \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu} \\
& \equiv \lambda
\end{aligned}$$

Therefore, we can construct the probability density functions and cumulative distribution functions for the Sufficient Statistic T in terms of the non-central χ^2 distribution functions.

$$\begin{aligned}
H_0 : \quad p(T) &= p_n(T, 0) = \frac{T^{n/2-1} e^{-T/2}}{\Gamma(\frac{1}{2}n) 2^{n/2}} \\
F(T) &= F(T|n, 0)
\end{aligned} \tag{2.28}$$

$$\begin{aligned}
H_1 : \quad p(T) &= p_n(T, \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}) = p_n(T, \lambda) \\
&= \frac{T^{(n/2-1)} e^{-(T+\lambda)/2}}{2^{n/2}} \sum_{k=0}^{\infty} \frac{(\lambda T)^k}{2^{2k} k! \Gamma(k + \frac{1}{2}n)} \\
F(T) &= F(T|n, \lambda)
\end{aligned} \tag{2.29}$$

where $F(T|n, \lambda)$ is the cumulative distribution function of a non-central χ^2 distributed random variable with n degrees of freedom and non-centrality parameter λ .

These statistics (Eq. (2.28), Eq. (2.29)) can help us calculate the performance of the detection as follows.

$$\begin{aligned}
P_D &= P(T > 2 \log \eta | H_1) = 1 - F_{H_1}(2 \log \eta) \\
&= 1 - F(2 \log \eta | n, \lambda) \\
&= 1 - F(2 \log \eta | n, \boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}) \\
P_F &= P(T > 2 \log \eta | H_0) = 1 - F_{H_0}(2 \log \eta) \\
&= 1 - F(2 \log \eta | n, 0)
\end{aligned} \tag{2.30}$$

where P_D is the probability of detection, and P_F is the probability of false alarm.

To compare the performance of the array detection with that of the single channel detection, we write the corresponding probabilities of detection and false

alarm for the single channel case.

$$\begin{aligned} P_D &= 1 - F_{H_1}(2 \log \eta) = 1 - F(2 \log \eta | 1, \frac{\mu_j^2}{\sigma_j^2}) \\ P_F &= 1 - F_{H_0}(2 \log \eta) = 1 - F(2 \log \eta | 1, 0) \end{aligned} \quad (2.31)$$

where μ_j is the mean of the signal at the channel j where we are detecting the signal.

In this case of unknown signal detection, we do not have one expression similar to Eq. (2.17) to compare the performance of array and single channel detection. However, it is possible to compare the performance in different cases numerically using the tabulated values of various χ^2 cumulative distribution functions.

Let us consider an example of a three channel electrode array. Assume the signal strength at the 3 electrodes in the presence of a spike given by Eq. (2.32), and the noise covariance given by Eq. (2.33).

$$\mathbf{As} = \mathbf{z} = \boldsymbol{\mu} = \begin{bmatrix} 2.2 \\ 2 \\ 1.8 \end{bmatrix} \quad (2.32)$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0.4 & 0.3 \\ 0.4 & 1 & 0.2 \\ 0.3 & 0.2 & 1 \end{bmatrix} \quad (2.33)$$

The detection performance in terms of the Receiver Operating Characteristics are plotted for this case in Fig.2.3.

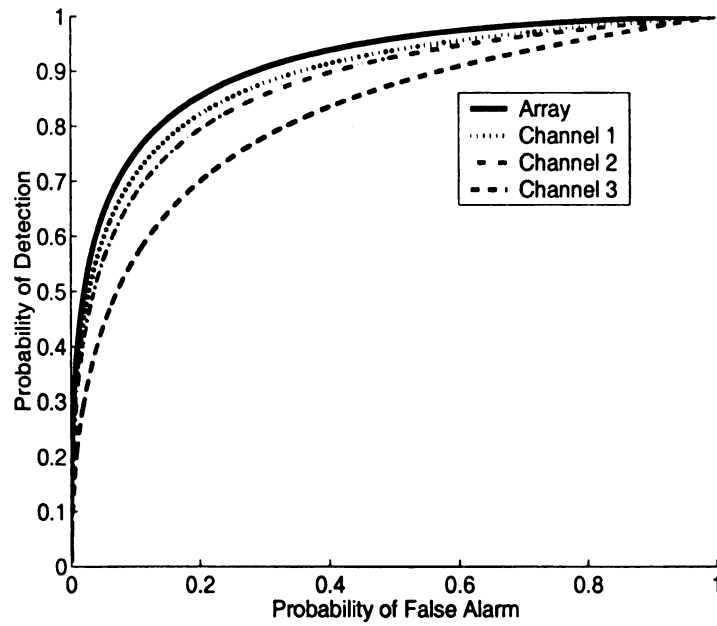


Figure 2.3. Receiver operating Characteristics for Array and single channel detection.

It is seen that in this case, spike detection using the array data gives better performance.

Let us now consider a slightly different case with the same noise covariance(Eq. (2.33)), but the signal strengths given by Eq. (2.34).

$$\mathbf{As} = \mathbf{x} = \boldsymbol{\mu} = \begin{bmatrix} 2.48 \\ 2.1 \\ 1.8 \end{bmatrix} \quad (2.34)$$

The detection performance in terms of the Receiver Operating Characteristics are plotted for this case in Fig.2.4.

On close examination, it is revealed that for low values of Probability of False Alarm, Channel 1 detection gives better performance compared to detection with the Array data. The situation is reversed for higher values of Probability of False Alarm. We thus observe that for the generalized likelihood ratio detection, we can not derive a Uniformly Most Powerful property for array vs. channel detection.

It will be worthwhile to calculate the values of the Signal to Noise Ratio at channel 1, and the equivalent array Signal to Noise Ratio we had derived in the known signal detection case (Eq. (2.35)).

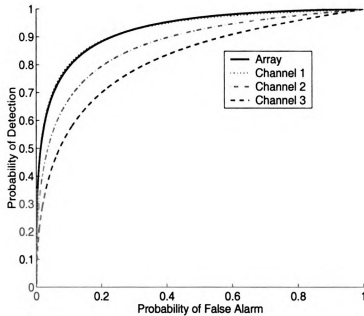


Figure 2.4. Receiver operating Characteristics for Array and single channel detection.

$$\begin{aligned}\text{Channel 1 SNR} &= 10\log_{10}\left(\frac{\mu_1^2}{\sigma_1^2}\right) = 7.889 \text{ dB} \\ \text{Equivalent Array SNR} &= 10\log_{10}\left(\boldsymbol{\mu}^T \mathbf{R}^{-1} \boldsymbol{\mu}\right) = 9.3516 \text{ dB}\end{aligned}\quad (2.35)$$

Channel 1 data and the Array data give comparable detection performance in this case. However, if we had known the signal (and applied the test of Eq. (2.10), the performance of the Array detection would have been much better due to its

higher equivalent SNR. Thus, we can conclude that the performance gain in array detection is more pronounced in the case of known signals.

2.5 Results

We run Monte Carlo simulations for spike detection of unknown signals for single channel and array observations at various SNR conditions. The performance of spike detection in the time domain and multiresolution (wavelet) domain is compared.

Fig.2.5 shows the performance of detection for single channel detection in multiresolution and time domain for an SNR of 7.6024. As we can see, the multiresolution detection performs much better than the time domain detection. Figs.2.6 and 2.7 also show similar improvements in detection performance due to the multiresolution analysis.

Figs.2.8, 2.9, and 2.10 show ROC curves for multiresolution and time domain detection at various SNR conditions. An example of the noisy data and the real locations of spikes for an SNR of 8.117 is shown in Figs.2.11 and 2.12.

At these noise levels, the spikes are well masked by the noise so as to make detection by visual inspection quite difficult. The level of masking can be seen in

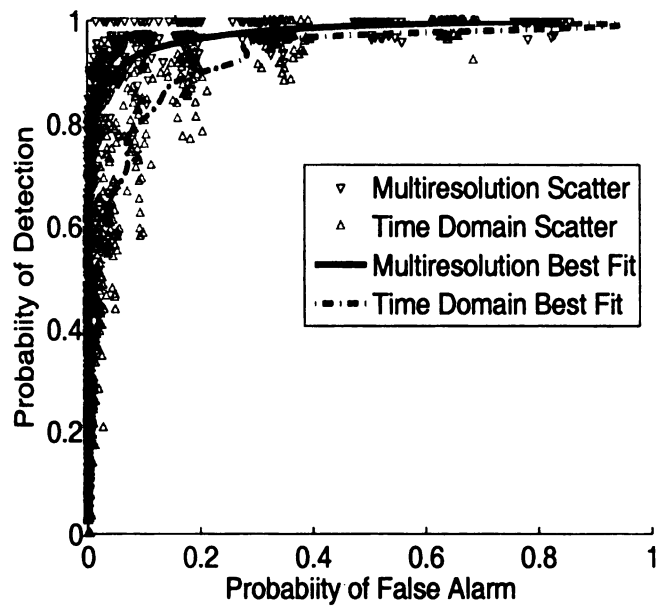


Figure 2.5. ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 7.6024.

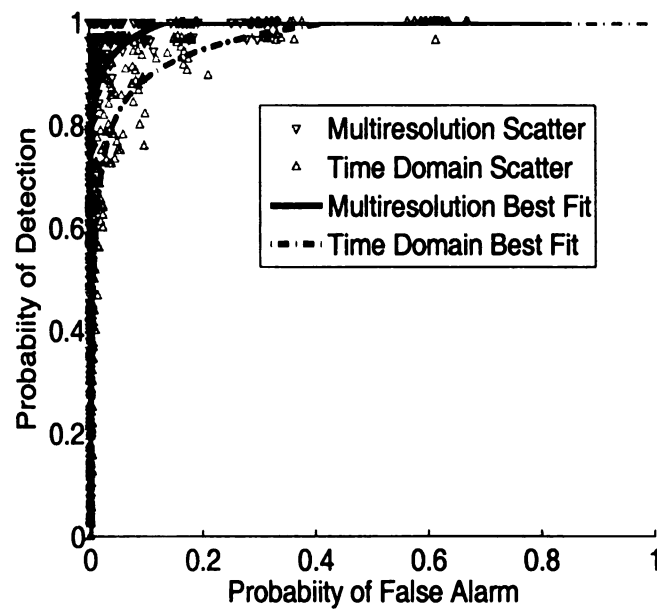


Figure 2.6. ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 8.7623.

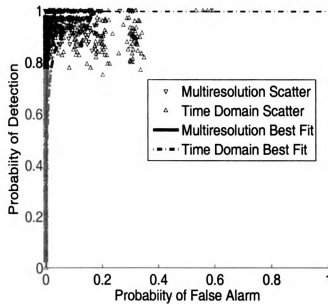


Figure 2.7. ROC curves for Multiresoution and Time domain detection for a single channel data at an SNR of 10.857.

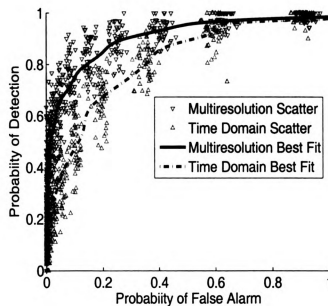


Figure 2.8. ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 8.1177.

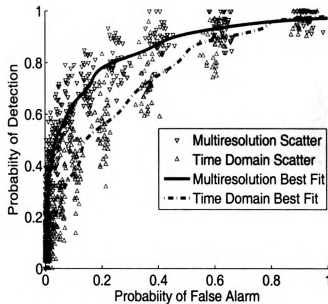


Figure 2.9. ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 6.9579.

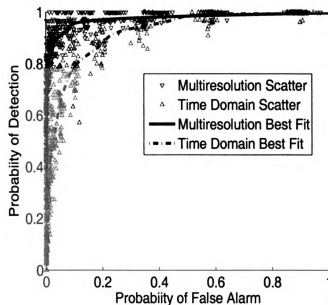


Figure 2.10. ROC curves for Multiresoution and Time domain detection for a two channel array at an SNR of 10.2124

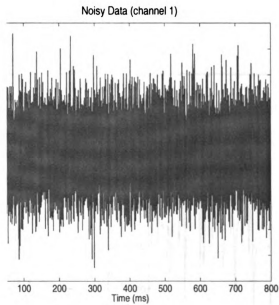


Figure 2.11. Noisy data on one channel used to detect spikes.

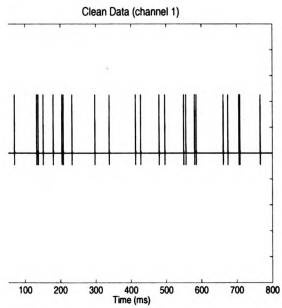


Figure 2.12. The clean spike data to be detected.

the spike which is shown in detail in Fig.2.13.

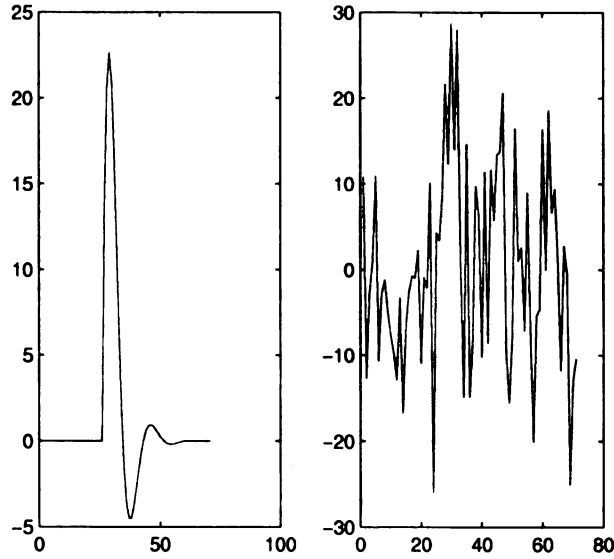


Figure 2.13. A clean and noisy spike example.

2.6 Conclusions

We have described Bayesian detection tests for spike detection in single electrode and array data. Analytical results for the Receiver Operating Characteristics have been derived for the known and unknown signal cases. Detection using the multi-electrode array data gives potentially better performance than single channel spike detection. A Uniformly Most Powerful property for the array detection versus the single channel detection is proved for the known signal case in terms of an “equiva-

lent array SNR". The effect of the spatial correlation of the noise on the performance of the array detection is also investigated. In the case of an unknown signal case, the array detection gives better performance than the single channel detection for certain signal and noise cases, but it may not be Uniformly Most Powerful. We have concentrated on the unknown signal case, as we assume spike detection to occur in the preliminary stages of the neural signal processing before any signal conditions are estimated. Transforming the data in a multi-resolution space like the wavelet transform and running the tests on the various time scales improves detection because the wavelet compresses the signal energy in a small number of coefficients. The performance gain in wavelet domain detection has been experimentally verified for various Signal to Noise Ratio conditions in single channel and array cases.

CHAPTER 3

Clustering Neural Spike Trains

3.1 Introduction

Ensemble coding of stimuli[22], motor commands[23], and memory[24] has been demonstrated by neuroscientists and is a subject of contemporary research. We consider high dimensional electrode arrays recording from subthalamic regions of the central nervous system. After the preliminary signal processing stages of compression, telemetry, spike detection, and spike sorting, we obtain spike trains of individual neurons. The recorded data usually comprise spike trains of a large number of neurons. While such neural yield presents rich data sets for understanding complex neural circuits and coding mechanisms, it makes inferring relationships between neurons quite a formidable task. However, not all the recorded neurons might be responding to a common stimulus under study. Our goal here is to cluster the data into groups of neurons which have correlated neural activity. This re-

duces the data dimensions for any secondary processing of the recorded spike trains.

3.1.1 Background

The spike trains, which are a series of action potentials, are characterized as discrete-event random processes. The exact statistical characteristics of the spike trains is still a research topic in the neuroscience community, with most studies subscribing to either a rate coding or temporal coding mechanism in the neural system.

The Poisson process has been traditionally used to model the arrival times of spike events. Alternatively, the inter spike interval generally obeys an exponential density. If the spike train is modelled as a homogenous Poisson process, it will be described completely by its mean firing rate. However, a homogenous Poisson assumption implies stationarity. Since neural signals encode time varying stimuli, commands, and higher brain functions, inhomogenous Poisson processes appear as a more natural way of modelling spike trains. An inhomogenous Poisson process is completely characterized by a time varying rate function. The rate function describes the probability of an action potential in a unit time interval, as a function of time. In this study we assume all neural spike trains to be inhomogenous spike trains. Various methods exist to characterize a spike train's Poisson rate function. The simplest way to do so is to count the number of spikes occurring in a time

window of some duration δT . This is known as binning the data with a bin width of size δT .

Inferring relationships between neural spike trains is an integral part of any multi unit spike train analysis[25]. Presently a number of methods exist to analyze these associations. The cross-corelogram is the cross variance of two spike trains calculated for different time lags. Cross-intensity is a similar relation that characterizes the firing rate of one neuron as a function of the firing time of the other neuron. These methods can be characterized as time domain methods[26].

Another widely used approach is the characterization of the firing activity of two or more neurons after the presentation of a stimulus. The Joint Peri-Stimulus Time Histogram (JPSTH) for two neurons is a matrix of the firing rate of one neuron at a time t_1 after the application of the stimulus, and firing rate of the other neuron at a time t_2 after the application of the stimulus. Computing the JPSTH requires repeatedly recording the same neurons under repetitive controlled application of the stimulus.

Analogous to these time domain methods, there are various frequency domain methods that infer relationships between spike trains. Computing the correlation between the Fourier transforms of spike trains leads to measures of coherence.

Operating in the frequency domain, these methods are not bin size dependent but assume stationarity on the data. Spectrograms and coherograms can be seen as the frequency domain counterparts of the JPSTH as they characterize the joint frequency domain firing rates of neurons in the presence of stimuli[27].

Multiple spike trains can also be modelled as jointly random discrete event processes. The joint probability functions can thus be characterized by a number of parameters, depending on the form of the probability function assumed. Standard statistical techniques can be used to estimate these parameters like maximum likelihood[28] and minimum errors. However, this method requires *a priori* knowledge of the parametric form of the multidimensional point process probability function, and sufficient data to estimate the parameters.

3.2 Simulation Method

Our simulation consists of 120 spike trains with 4 groups of 30 neurons each having related firing rates within the group, and independent of neurons outside the group. Every group of neurons had a basic firing rate function. For example, the k^{th} group $\mathbf{S}_{\mathbf{k}}$ is associated with a firing rate function $\mathbf{f}_{\mathbf{k}} \in \mathbb{R}_{N \times 1}$ sampled at N points in time. A basic firing rate function was selected independently for each of the four groups.

For each group $\mathbf{S}_{\mathbf{k}}$, firing rate functions $\mathbf{f}_{\mathbf{m}} \in \mathbb{R}_{N \times 1}$ were generated for all neurons $m, m \in \mathbf{S}_{\mathbf{k}}$ with linear and delay relations as in Eq. (3.1).

$$f_m[n] = \alpha f_k[n - \tau] + \beta \quad (3.1)$$

where α and β are randomly generated parameters for a linear relation, τ is a randomly generated delay parameter.

For each neural firing rate $\mathbf{f}_{\mathbf{m}}$, a spike train $\mathbf{s}_{\mathbf{m}}$ is obtained as a realization of an in-homogenous Poisson process. Spike trains were thus generated for 30 neurons $m_1, m_2, \dots, m_{30}$ based on the basic firing rate function of each group k , where $m_1, m_2, \dots, m_{30} \in \mathbf{S}_{\mathbf{k}}$. For 4 groups $k = \{1, 2, 3, 4\}$, a total of 120 spike trains are thus generated. Our aim is to cluster these 120 neuronal spike trains into the 4 natural clusters.

3.2.1 Proposed Technique

In order to overcome the difficulties in predetermining the bin size in time domain methods, the stationarity assumptions in frequency domain methods, and the data requirements of the the parametric methods, we rely on a novel method to cluster these spike trains.

A discrete wavelet transform representation of the spike train is obtained to 8

levels. The correlations among the neuronal spike trains in different time scales are calculated and plotted in Fig.3.4. For visual clarity, each group of related neural spike train is numbered consecutively. Hence, a cluster appears as a continuous band of 30 neurons. We see that different clusters appear with different strengths in different time scales.

Graph theoretic and k-means clustering results with features extracted by Principal Component Analysis are reported.

3.3 Multi-resolution Approach

We assume that functionally interdependent neurons due to synaptic connections or shared stimuli will have related firing rates. The relations in the firing rate can be characterized in the pair-wise correlations of the estimated firing rates.

We have indicated that time domain methods like cross-corelograms require a fixed bin width depending on the assumed frequency content of the signal. Increasing the bin width gives a more efficient estimate of the firing rate in the sense of a smaller variance of the estimate. However, the temporal resolution in the estimation of a varying firing rate would degrade with larger bin widths. Therefore,

this results in a variance-resolution trade off in the firing rate estimation. The optimal bin width will depend on the firing rate, its rate of variation, and its relation to the stimulus or signal being encoded by the neural signals. Frequency domain methods are not bin-size dependent but assume stationarity on the signals.

Multi-resolution estimation of the firing rates resolves these problems by decomposing the spike trains with a wavelet transform (Fig.3.3). We propose the use of the Haar wavelet basis to represent the observed spike trains. The use of the Haar is well-suited for the spike trains because the wavelet coefficient at each level j of the wavelet tree can be related to the firing rate computed at a certain bin width 2^j . The Haar basis approximation $u[n]$ and detail $v[n]$ functions are shown in Figs.3.1, and 3.2.

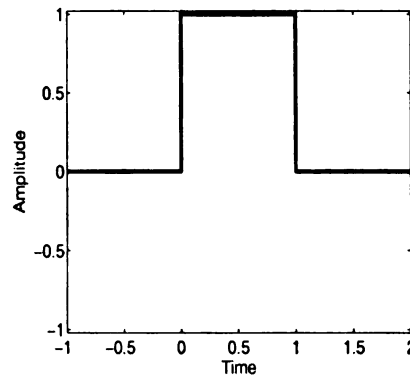


Figure 3.1. The Haar basis approximation function $u[n]$.

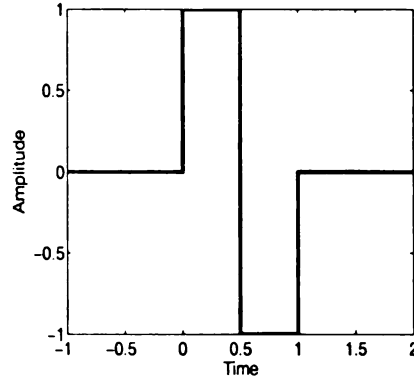


Figure 3.2. The Haar basis detail function $v[n]$.

At any level j , the details coefficients $d[n]$ of a signal $s[n]$ in the wavelet transform are obtained as $d[n] = \frac{1}{2^j} \langle v[2^j(r - n)]s[r] \rangle$, and the approximations coefficients $a[n]$ by $a[n] = \frac{1}{2^j} \langle u[2^j(r - n)]s[r] \rangle$. The Haar basis functions have compact support, and the constant absolute magnitude within the support does not scale the data as in the case of any higher order basis function. With the wavelet functions for the Haar basis, the details coefficient $d[n]$ is simply the change in the firing rate at the n^{th} bin of size 2^{j-1} . The approximation coefficient $a[n]$ is the firing rate at the n^{th} of size 2^j .

A wavelet representation of the neural spike trains to characterize processes as Poisson or non-Poisson has been reported by Cao[29] for decoding neural spike trains in neural prosthetic devices. However, our purpose here is to have a convenient representation for the neural spike train firing rate for inferring clusters of neurons with correlated activity.

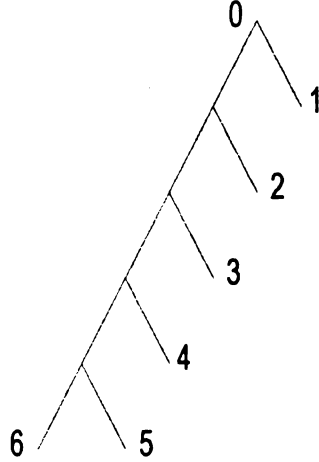


Figure 3.3. The Wavelet Tree representation to 5 levels.

The wavelet transformation at level j can be represented as a lumped matrix $\mathbf{W}^{(j)} \in \mathbb{R}_{\frac{N}{2^j} \times N}$ where N is the length of the spike train being decomposed[21]. We initially assume the spike trains to be binned to a very small bin width δT where within δT no more than one event can occur. The spike train expressing the neural activity during a duration $N\delta T$ will be a vector of length N . Therefore, the wavelet details of the m^{th} spike train $\mathbf{s}_m$ at level j can be computed as in Eq. (3.2).

$$\mathbf{s}_m^j = \mathbf{W}^{(j)} \mathbf{s}_m \quad (3.2)$$

where $\mathbf{s}_m^j \in \mathbb{R}_{\frac{N}{2^j} \times 1}$ is the representation of $\mathbf{s}_m$ at node j .

The representation of the M spike trains $\mathbf{S} \in \mathbb{R}_{N \times M}$ at node j of the wavelet tree is given by $\mathbf{S}^j \in \mathbb{R}_{\frac{N}{2^j} \times M}$ (Eq. (3.3)).

$$\mathbf{S}^j = \mathbf{S} \mathbf{W}^{(j)} = \begin{bmatrix} \mathbf{s}_1^j & \mathbf{s}_2^j & \dots & \mathbf{s}_M^j \end{bmatrix} \quad (3.3)$$

We can assume the spike train vector at bin width δT to be a noisy estimate of its underlying firing rate $\mathbf{f}_m = [f_m[1] \ f_m[2] \ \dots \ f_m[N]]^T$ sampled at intervals of δT . Hence, for a neuron m with firing rate given by $\mathbf{f}_m \in \mathbb{R}_{N \times 1}$ (sampled at N intervals δT time apart), the spike train data can be represented as in Eq. (3.4).

$$\mathbf{s}_m = \mathbf{f}_m \delta T + \mathbf{q}_m \quad (3.4)$$

where $\mathbf{q}_m$ expresses a random process associated with the uncertainty due to using a single realization of the random process (spike train $\mathbf{s}_m$) to estimate the parameter (firing rate $\mathbf{f}_m$). Note that we chose δT small enough so that not more than a single spike can occur in one bin (due to the refractory period). This implies that $\mathbf{f}_m \delta T < 1$.

We can derive some properties of the "noise" term $\mathbf{q}_m$. At a particular instance of the discrete time, p , the probability of observing n spikes due to its Poisson nature, is given by Eq. (3.5).

$$P[\mathbf{s}_m[p] = n] = \frac{(\mathbf{f}_m[p]\delta T)^n e^{-(\mathbf{f}_m[p]\delta T)}}{n!} \quad (3.5)$$

We thus observe that $\mathbf{q}_m[p]$, at any time instance p is given by Eq. (3.6). Writing out its probability distribution function(Eq. (3.7)), we observe that it is a zero mean, non-stationary, temporally white process which is uncorrelated across neurons(Eq. (3.8)).

$$\mathbf{q}_m[p] = \mathbf{s}_m[p] - \mathbf{f}_m[p]\delta T \quad (3.6)$$

$$P[\mathbf{q}_m[p] = m] = \frac{(\mathbf{f}_m[p]\delta T)^{(m+\mathbf{f}_m[p]\delta T)} e^{-(\mathbf{f}_m[p]\delta T)}}{(m + \mathbf{f}_m[p]\delta T)!}, \quad (3.7)$$

$$\forall m - \mathbf{f}_m[p]\delta T \in \{\dots, -1, 0, 1, 2, \dots\}$$

$$E[\mathbf{q}_m[p]] = 0$$

$$E[(\mathbf{q}_m[p])^2] = \mathbf{f}_m[p]\delta T - (\mathbf{f}_m[p]\delta T)^2 \quad (3.8)$$

$$E[\mathbf{q}_m[p_1]\mathbf{q}_m[p_2]] = 0$$

$$E[\mathbf{q}_{m_1}[p]\mathbf{q}_{m_2}[p]] = 0$$

The property of the noise process being white across neurons is because we had assumed neurons in the same group to have related firing rates but no second probabilistic dependence. This may or may not hold in practice depending on the biological mechanism producing the correlated firing activity. Note that this ap-

proximation (Eq. (3.4)) of the binned spike train eases further analysis. We assume each of the M neurons as belonging to any one of the K groups ($\mathbf{S}_1 \mathbf{S}_2 \dots \mathbf{S}_k \dots \mathbf{S}_K$) with some related firing rate. We assume the underlying basic firing rate function of the k^{th} group of neurons to be $\mathbf{f}_k$. Within group S_k it is assumed that m_k neuronal firing rates are linearly related by

$$\mathbf{f}_m = \sum_{k=1}^K a_{mk} \mathbf{f}_k \quad (3.9)$$

where a_{mk} are coefficients describing the degree of belongingness of neurons to the different groups with $a_{mk} = 0 \forall m \notin \mathbf{S}_k$, and $a_{mk} \neq 0 \forall m \in \mathbf{S}_k$.

The wavelet representation at level j of the spike train can thus be expressed as

$$\mathbf{s}_m^j = a_{mk} \mathbf{W}^{(j)} \mathbf{f}_k \delta T + \mathbf{W}^{(j)} \mathbf{q}_m, \forall m \in (S_k) \quad (3.10)$$

To infer relations between neural spike trains, a classical technique is to compute the correlation of the estimated firing rates at a certain bin width[30]. By taking the discrete wavelet transform of the spike trains the multiple resolutions obtained correspond to multiple bin widths. We now compute the sample cross-correlation between different neurons at the wavelet subband j as

$$\begin{aligned} c_{m_1, m_2}^{(j)} &= \frac{1}{N} \mathbf{s}_{m_1}^j T \mathbf{s}_{m_2}^j = \frac{1}{N} \sum_{n=0}^N s_{m_1}^j[n] s_{m_2}^j[n] \\ &\simeq E \left[s_{m_1}^j[n] s_{m_2}^j[n] \right] \end{aligned} \quad (3.11)$$

The complete correlation matrix at level j , $\mathbf{C}^{(j)} \in \mathbb{R}_{M \times M}$ will have as its elements $c_{m_1, m_2}^{(j)}$.

If we assume the underlying basic firing rates $f_k, k = 1, \dots, K$ of the different neural groups ($\mathbf{S}_1 \dots \mathbf{S}_K$) to be independent, we can derive the following relations on the expectation of the correlation.

$$\begin{aligned}
E \left[c_{m_1, m_2}^{(j)} \right] &= \frac{1}{N} E \left[\mathbf{s}_{\mathbf{m}_1}^j{}^T \mathbf{s}_{\mathbf{m}_2}^j \right] \\
&= \frac{1}{N} E \left[\left(a_{m_1 k} \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} \delta T + \mathbf{W}^{(j)} \mathbf{q}_{\mathbf{m}_1} \right)^T \left(a_{m_2 k} \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} \delta T + \mathbf{W}^{(j)} \mathbf{q}_{\mathbf{m}_2} \right) \right] \\
&= \frac{1}{N} \left(\sum_{k=1}^K a_{m_1 k} \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} \right)^T \left(\sum_{k=1}^K \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} a_{m_2 k} \right) (\delta T)^2 \\
&= 0 \quad \forall m_1 \in \mathbf{S}_{\mathbf{k}_1}, m_2 \in \mathbf{S}_{\mathbf{k}_2}, k_1 \neq k_2 \\
&= \frac{1}{N} a_{m_1 k} a_{m_2 k} \mathbf{f}_{\mathbf{k}}^T \mathbf{W}^{(j)T} \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} (\delta T)^2 = a_{m_1 k} a_{m_2 k} e_k^j \quad \forall m_1, m_2 \in \mathbf{S}_{\mathbf{k}} \quad (3.12)
\end{aligned}$$

where $e_k^j = \frac{1}{N} \mathbf{f}_{\mathbf{k}}^T \mathbf{W}^{(j)T} \mathbf{W}^{(j)} \mathbf{f}_{\mathbf{k}} (\delta T)^2$ represents the energy of the k^{th} groups firing rate $\mathbf{f}_{\mathbf{k}}$ at time scale j .

This is interpreted as the expectation of the correlation between neurons m_1

and m_2 at any time scale j being zero if the neurons do not have correlated firing within that choice of bin width. The expectation of the correlation is $a_{m_1 k} a_{m_2 k} e_k^j$, with e_k^j being a scaling coefficient describing the spectral content of the firing rate of the neural group k at time scale j , and a_{mk} , being some unknown relations on the firing rates with $a_{mk} \neq 0$ describing the membership of neuron m in the group k .

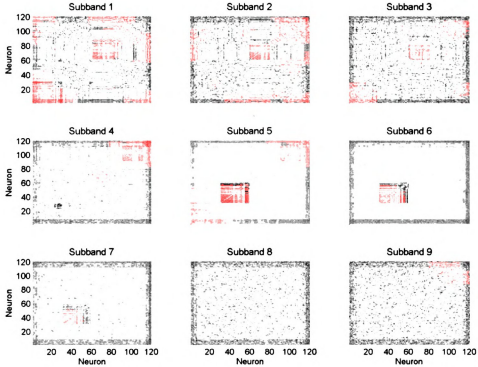


Figure 3.4. Correlations in different subbands.

3.4 A Principal Components Approach for Multi-scale Clustering

Eq. (3.11) hints that some clusters can be identified in certain time scales with different strengths e_k^j . Correlated neural activity at time scale j appears as non zero correlation among the neurons at that time scale. Consider a cluster correlation $\mathbf{A}^{(\mathbf{k})} \in \mathbb{R}_{M \times M}$ for every cluster $k = 1, \dots, K$ with the m_1, m_2^{th} element being $\mathbf{A}^{(\mathbf{k})}[m_1, m_2] = a_{m_1 k} a_{m_2 k}$. The cluster correlations $\mathbf{A}^{(\mathbf{k})}$ can be thought of as the essential features making up the correlation $\mathbf{C}^{(j)} = \sum_{k=1}^K \mathbf{A}^{(\mathbf{k})} e_k^j$ at any arbitrary time scale j . Therefore, Principal Component Analysis[31] should resolve the cluster energies in the first few dominant principal components, with the noise energy distributed among the smaller principal components. The correlation profiles in different principal components are given in Fig.3.5. Figures in this thesis appear in color. The dominant principal components express the energy of the correlated firing rates in them, while the independent noise occupy the less dominant principal components. The eigenvalues associated with the principal components are tabulated in Table 3.1. The first four principal components capture 75.5% of the energy, and we assume these to capture the required information for all the clusters. We form a fused matrix from the principal components to be used for clustering the neurons. The entries of this matrix $\mathbf{D}$ are calculated as in Eq. (3.13) using the 4 most dominant principal components.

Table 3.1. Eigenvalues associated with the Principal Components

Principal Component	Eigenvalue
1	0.2537
2	0.1926
3	0.1739
4	0.1349
5	0.1136
6	0.0598
7	0.0326
8	0.0226
9	0.0162

$$d_{m_1, m_2} = \sum_{p=1}^4 |c_{m_1, m_2}^{(p)}| \lambda^{(p)} \quad (3.13)$$

where $c_{m_1, m_2}^{(p)}$ is the correlation between the neurons m_1 and m_2 in principal component p , and $\lambda^{(p)}$ is the eigenvalue associated with the p^{th} principal component.

This resulting matrix has a profile plotted in Fig.3.6. As we can see, all the four natural clusters are well represented and distinctly visible in this matrix.

3.5 Graph Theoretic Clustering

A graph theoretic clustering approach[32] is needed in our application because the neurons to be clustered are related through pair-wise similarity measures

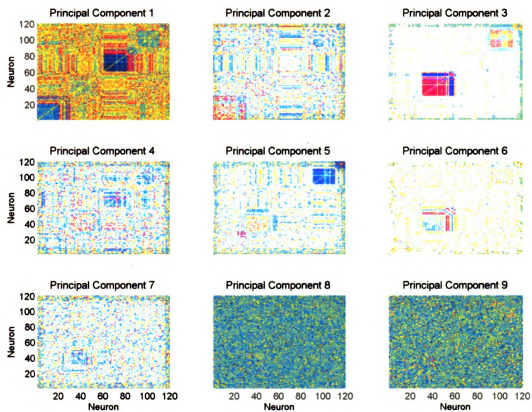


Figure 3.5. Principal Components of correlations.

rather than a low dimensional vector associated with each neuron as required in vector clustering algorithms[33]. The true clusters of neurons are plotted in the space of two components resolved from the edge weight matrix of Eq. (3.13) are plotted in Fig.3.7. While our choice of the edge weights computed from the principal components of the correlation matrices in different subbands offers a good representation of the clusters, the data cannot be satisfactorily clustered with any of the classical “vector data” algorithms like k-means or k-Nearest Neighbors.

The neural spike trains are represented as the vertices of an undirected graph. The weight, or capacity of an edge connecting any two neurons m_1 and m_2 is a measure of correlation among the neurons. The weights of the edges are assigned from the $\mathbf{D}$ matrix, with the edge weight between neurons m_1 and m_2 given by Eq. (3.13).

We use a minimum cut type of algorithm on our graph representation to separate the neurons in clusters. Probabilistic clustering, with “soft” membership functions to characterize the membership of a neuron in a cluster is used. Probabilistic clustering is a technique to reduce the computational complexity of an otherwise NP-hard clustering problem[34]. We wish to find clusters that have maximum edge weights between neurons within a cluster, and minimum edge weights between neurons in two different clusters, or across a cut. With the $p_{m_1,k}$ denoting the membership of neuron m_1 in cluster k , we form an objective

function[35] described by Eq. (3.14). We note that intuitively, we want $p_{m,k}$ to have a high value if a_{mk} is statistically non-zero.

$$l = \sum_{k=1}^K \frac{\sum_{i=1}^M \sum_{j=1}^M p_{i,k} p_{j,k} d_{i,j}}{\sum_{i=1}^M \sum_{j=1}^M p_{i,k} d_{i,j}} \quad (3.14)$$

Maximizing the objective function l attempts to minimize the edge weights (which are measures of correlated firing) across cuts separating the clusters. Finally, crisp decisions on the cluster memberships are calculated by assigning the neuron to the cluster in which it has the highest probabilistic membership.

3.6 Clustering Results

To compare the performance of our probabilistic graph clustering with traditional methods, we tried to use the first few resolved components from the fused matrix $\mathbf{D}$ to perform clustering with the k-means algorithm. As we can see in Fig.3.7, neural groups 1 and 3 are well mixed, and can not be linearly separated in the first two component space. Fig.3.8 shows a k-means result for the first two principal components of the fused matrix. As expected, this gives quite a significant clustering error of 32.5% We define the clustering error as the number of incorrectly

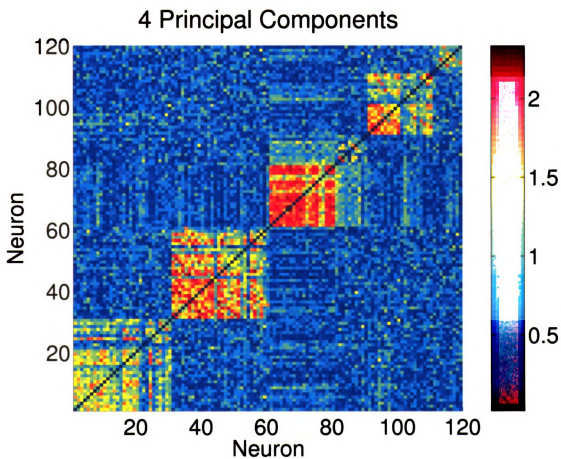


Figure 3.6. Fused profile of the first 4 principal components.

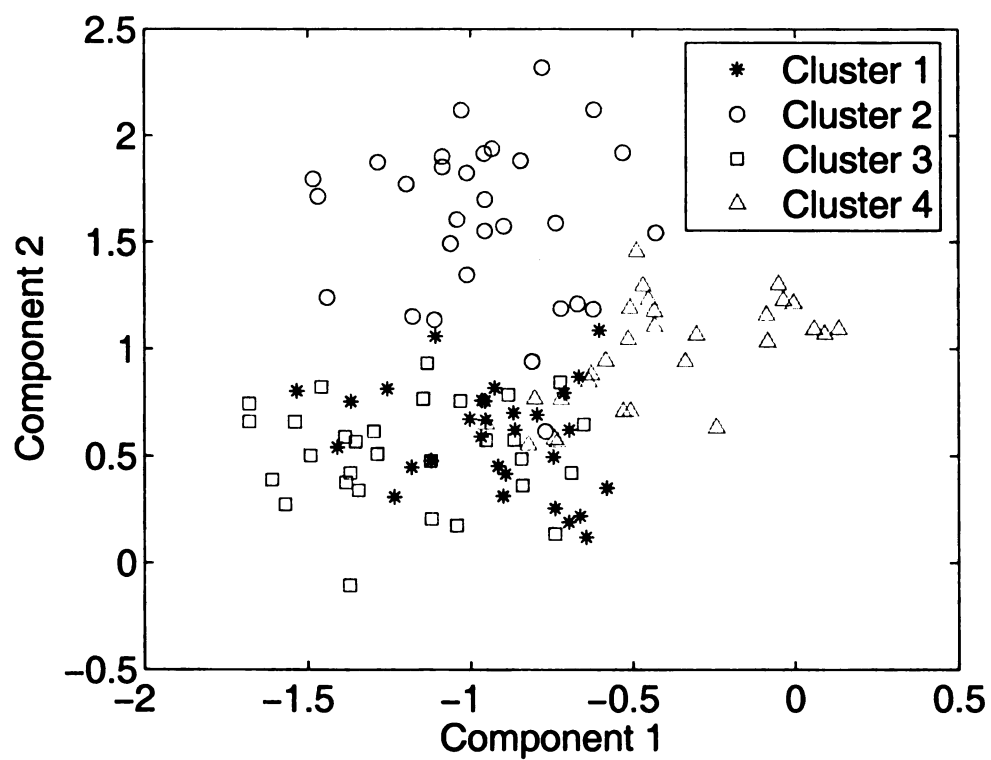


Figure 3.7. The true clusters depicted in the first two components resolved from the fused matrix **D**.

Table 3.2. Clustering Errors obtained in the k-means algorithm

No. of Principal Components	Clustering Error
2	33.21%
3	27.49%
4	14.61%
5	16.87%
6	16.93%
7	17.57%
15	16.41%

classified neurons as a fraction of the total number of neurons. Since the results of the k-means algorithm depend on the initial guess of cluster centers, we run the k-means in 1000 iterations and compute the mean error. Table 3.2 shows the mean clustering errors for the k-means algorithms applied to different numbers of principal components. As we see, the clustering error does not decrease uniformly by increasing the number of principal components. Though increasing the number of principal components captures more useful information, it also adds statistical noise to the data. This shows that the performance of the k-means algorithm is limited due to the inherent nature of the pair-wise similarity measures in our problem, which lends itself better to a graph theoretic representation.

Probabilistic clustering with the edge weights of Eq. (3.13) gave a clustering error of 2.5% *i.e.*, only 3 neurons out of the 120 being classified in the wrong cluster.

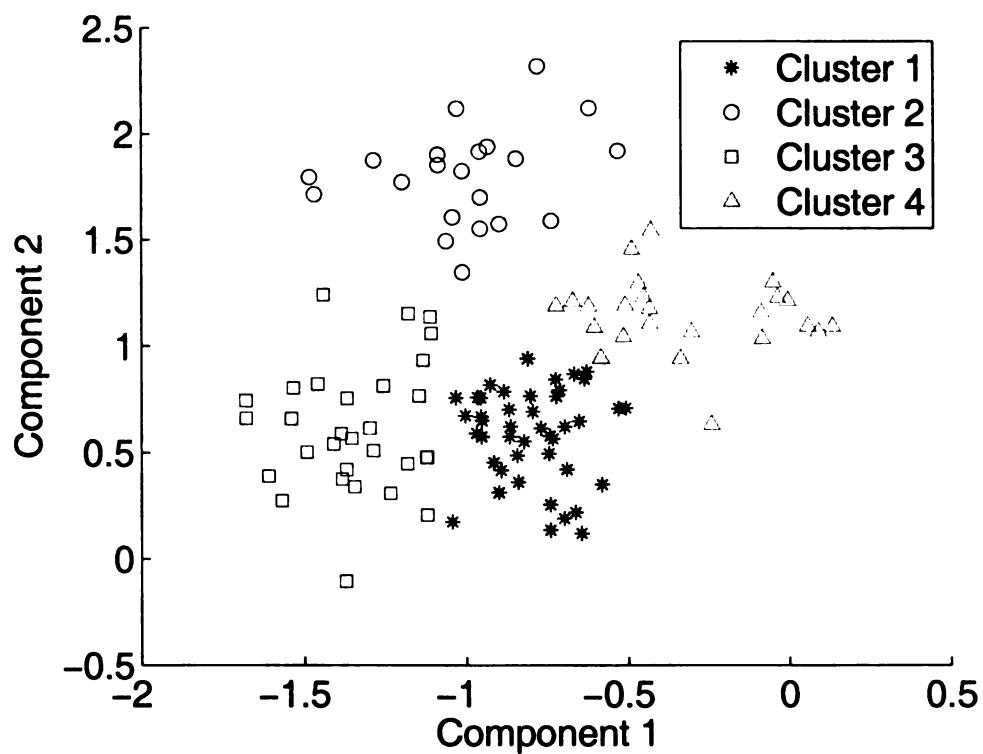


Figure 3.8. Result of the k-means algorithm applied to the first two principal components of the fused matrix.

3.7 Conclusions

We have demonstrated the use of a novel multi-scale clustering algorithm for partitioning a large set of neural spike trains into groups with correlated firing. A multi-scale or wavelet domain representation of the neural spike train eliminates the need of fixing a bin width and assuming stationarity, which are the limitations of traditional time domain and frequency domain methods respectively. We have demonstrated that the probabilistic graph theoretic clustering approach outperforms traditional algorithms like k-means for our application. We believe that this algorithm can go a long way in enabling neuroscientists to analyze large scale neural recordings and give a better systems understanding of neural activities. In future, we need to test this approach on data sets generated from a more biologically faithful representation of neural circuits or on real large scale data sets.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] K.D Wise, D.J. Anderson, J.F. Hetke, D.R. Kipke, and K. Najafi. Wireless implantable microsystems: High-density electronic interfaces to the nervous system. In *Proc. of the IEEE*, volume 92, pages 76–97, 2004.
- [2] F.A. Edwards, A. Konnerth, B. Sakmann, and T. Takahashi. A thin slice preparation for patch clamp recordings from neurones of the mammalian central nervous system. *Pflugers Archive: European Journal of Physiology*, 414(5):600–612, September 1989.
- [3] J.C. Chang, G.J. Brewer, and B.C. Wheeler. Microelectrode array recordings of patterned hippocampal neurons for four weeks. *Biomedical Microdevices*, 2(4):245–253, December 2000.
- [4] M. Ghovanloo and K. Najafi. A wideband frequency-shift keying wireless ink for inductively powered biomedical implants. *IEEE Transactions on Circuits and Systems I*, 51(12):2374–2383, December 2004.
- [5] K.G. Oweiss, D.J. Anderson, and M.M. Papaefthymiou. Optimizing signal coding in neural interface system-on-a-chip modules. In *Proceedings of the 25th IEEE Conference of EMBS*, pages 2016–2019, September 2003.
- [6] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Publishers, 2 edition, 1999.
- [7] K. G. Oweiss and D. J. Anderson. A multiresolution generalized maximum likelihood approach for the detection of unknown transient multichannel signals in colored noise with unknown covariance. In *Proceedings of the ICASSP'2002*, volume 3, pages 2993–2996, May 2002.
- [8] K.G. Oweiss and D.J. Anderson. Noise reduction in multichannel neural recordings using a new array wavelet denoising algorithm. *Neurocomputing*, 38-40:1687–1693, July 2001.
- [9] K.G. Oweiss and D.J. Anderson. Massit-multiresolution analysis of signal subspace invariance technique: A novel algorithm for blind source separation. In *Proc. of the IEEE 35th Asilomar Conference on Signals, Systems and Computers*, pages 819–823, November 2001.

- [10] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- [11] I. Daubechies and W. Swedens. Factoring wavelet transform into lifting steps. *Journal of Fourier Anaysis Apllications*, 41(3):247–269, 1998.
- [12] W. Sweldens, A.R. Calderbank, I. Daubechies, and B.L. Yeo. Wavelet transforms that map integers to integers. *Applied Computational Harmonic Analysis*, 5(3):332–369, 1998.
- [13] K. G. Oweiss and D. J. Anderson. A unified framework for advancing array signal processing technology of multichannel microprobe neural recording devices. In *Proceedings of the 2nd IEEE Conference on Microtechnology in Medicine and Biology*, pages 245–250, May 2002.
- [14] David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [15] Michael S. Lewicki. A review of methods for spike sorting: the detection and classification of neura action potentials. *Network: Computational Neura Systems*, 9:R53–R78, 1998.
- [16] Karim Oweiss, Andrew Mason, Kyle Thomson, Jian Li, and Yasir Suhail. A scalable wavelet transform vlsi architecture for real-time neural signal conditioning in implantable multichannel neuroprosthetic devices. Under review, for *IEEE Transactions on Circuits and Systems*.
- [17] Y. Suhail and K.G. Oweiss. A reduced complexity integer lifting wavelet-based module for real-time processing in implantable neural interface devices. In *Conference Proceedingsof the 26th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 4552 – 4555, Sept. 2004.
- [18] K. Oweiss, K. Thomson, , and D. Anderson. A systems approach for real-time data compression in advanced brain-machine interfaces. In *Conference Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering.*, 2005.
- [19] Z. Nenadic and J. W. Burdick. Spike detection using the continuous wavelet transform. *IEEE Transactions on Biomedical Engineering*, 52(1):74–87, January 2005.
- [20] K. G. Oweiss. *Multiresolution Analysis of Multichannel Neural Recordings in the Context of Signal Detection, Estimation, Classification and Noise Suppression*. PhD thesis, University of Michigan, Ann Arbor, 2002.
- [21] Michael W. Frazier. *An Introduction to Wavelets Through Linear Algebra*. Springer-Verlag, 1999.

- [22] Paul M. Gochin, Michael Colombo, A. Gerald Dorfman., George L. Gerstein, and Charles G. Gross. Neural ensemble coding in inferior temporal cortex. *Journal of Neurophysiology*, 71(6):2325–2327, June 1994.
- [23] Mark Laubach, Johan Wessberg, and Miguel A. L. Nicolelis. Cortical ensemble activity increasingly predicts behaviour outcomes during learning of a motor task. *Nature*, 405:567–571, June 2000.
- [24] H Eichenbaum, SI Wiener, ML Shapiro, and NJ Cohen. The organization of spatial coding in the hippocampus: a study of neural ensemble activity. *Journal of Neuroscience*, 9:2764–2775, August 1989.
- [25] Emery N. Brown, Robert E. Kass, and Partha P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, May 2004.
- [26] Z. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. ii. simultaneous spike trains. *Biophysics Journal*, 7(4):419–40, July 1967.
- [27] M. R. Jarvis and Mitra P. P. Sampling properties of the spectrum and coherency of sequences of action potentials. *Neural Computation*, 13:717–749, 2001.
- [28] E.S. Chornoby, L.P. Schramm, and A. F. Karr. Maximum likelihood estimation of neural point processes. *Biological Cybernetics*, 59:265–275, 1988.
- [29] Shiyao Cao. *Spike Train Characterization and Decoding for Neural Prosthetic Devices*. PhD thesis, California Institute of Technology, 2003.
- [30] Y. Hata, T. Tsumoto, H. Sato, and H. Tamura. Horizontal interactions between visual cortical neurones studied by cross-correlation analysis in the cat. *Journal of Physiology*, 441:593–614, September 1991.
- [31] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2 edition, October 2002.
- [32] D. W. Matula. *Classification and Clustering*, chapter Graph theoretic techniques for cluster analysis algorithms, pages 95–129. Academic Press, 1977.
- [33] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
- [34] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14, 2001.
- [35] K. Oweiss, R. Jin, and Y. Suhail. Scale-space processing and clustering for efficient multi-electrode data analysis of large-size neuronal ensembles. In *Conference Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering*, 2005.