University of Kentucky
UKnowledge

University of Kentucky Master's Theses | Graduate School

2006

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

Dharshan C. Medonza
University of Kentucky, DharshanMedonza@gmail.com

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Medonza, Dharshan C., "AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS" (2006). *University of Kentucky Master's Theses*. 271.
https://uknowledge.uky.edu/gradschool_theses/271

This Thesis is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Master's Theses by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF THESIS

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

Currently technologies such as EEG, EMG and EOG recordings are the established methods used in the analysis of sleep. But if these methods are to be employed to study sleep in rodents, extensive surgery and recovery is involved which can be both time consuming and costly. This thesis presents and analyzes a cost effective, non-invasive, high throughput system for detecting the sleep and wake patterns in mice using a piezoelectric sensor. This sensor was placed at the bottom of the mice cages to monitor the movements of the mice. The thesis work included the development of the instrumentation and signal acquisition system for recording the signals critical to sleep and wake classification.

Classification of the mouse sleep and wake states were studied for a linear classifier and a Neural Network classifier based on 23 features extracted from the Power Spectrum (PS), Generalized Spectrum (GS), and Autocorrelation (AC) functions of short data intervals. The testing of the classifiers was done on two data sets collected from two mice, with each data set having around 5 hours of data. A scoring of the sleep and wake states was also done via human observation to aid in the training of the classifiers. The performances of these two classifiers were analyzed by looking at the misclassification error of a set of test features when run through a classifier trained by a set of training features. The best performing features were selected by first testing each of the 23 features individually in a linear classifier and ranking them according to their misclassification rate. A test was then done on the 10 best individually performing features where they were grouped in all possible combinations of 5 features to determine the feature combinations leading to the lowest error rates in a multi feature classifier. From this test 5 features were eventually chosen to do the classification. It was found that the features related to the signal energy and the spectral peaks in the 3Hz range gave the lowest errors. Error rates as low as 4% and 9% were achieved from a 5-feature linear

classifier for the two data sets. The error rates from a 5-feature Neural Network classifier were found to be 6% and 12% respectively for these two data sets.

KEYWORDS: Piezoelectric Sensors, Generalized Spectrum, Pattern Recognition, Neural Network, Linear Classifier

_____

_____

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

By

Dharshan C. Medonza

DR. KEVIN D. DONOHUE
Director of Thesis

DR. YUMING ZHANG
Director of Graduate Studies

December 6th 2005
Date

Copyright © Dharshan C. Medonza 2005

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

By

Dharshan C. Medonza

DR. BRUCE O' HARA
Co- Director of Thesis

DR. YUMING ZHANG
Director of Graduate Studies

December 6th 2005
Date

Copyright © Dharshan C. Medonza 2005

RULES FOR THE USE OF THESIS

Unpublished theses submitted for the Masters degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the graduate School of the University of Kentucky.

A library that borrows this dissertation for use by its patrons is expected to secure the signature of each user.

Name Date

_____

_____

_____

THESIS

Dharshan C. Medonza

The Graduate School
University of Kentucky
2005

Copyright © Dharshan C. Medonza 2005

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in the
College of Engineering
at the University of Kentucky

By

Dharshan C. Medonza

Lexington, Kentucky

Director: Dr. Kevin D. Donohue, Professor of Electrical Engineering

Lexington, Kentucky

2005

Copyright © Dharshan C. Medonza 2005

AUTOMATIC DETECTION OF SLEEP AND WAKE STATES IN MICE USING PIEZOELECTRIC SENSORS

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in the
College of Engineering
at the University of Kentucky

By

Dharshan C. Medonza

Lexington, Kentucky

Co-Director: Dr. Bruce O'Hara, Professor of Biology

Lexington, Kentucky

2005

Copyright © Dharshan C. Medonza 2005

MASTER’S THESIS RELEASE

I authorize the University of Kentucky Libraries
to reproduce this thesis in whole or in part for purposes of research

_____

_____

To my parents

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to my advisor, Dr. Kevin D. Donohue, for providing me with the opportunity and continuous support to undertake this project as well as for offering timely and instructive comments and evaluation at every stage of the thesis process. I would also like to thank Dr. Bruce O'Hara for his support throughout this project and also Dr. Laurence G. Hassebrook for serving on the thesis committee. This research was supported by a DoD DEPSCoR award and an associated Kentucky EPSCoR supplement. Early support was also provided by the University of Kentucky through an NSF EPSCoR and unrestricted funds.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Why study sleep

Humans on average spend around 8-8.5 hours a day on sleep, which means that about a third of our lifetime is spent sleeping. Yet recent estimates suggest that around 40 to 70 million Americans experience either chronic or intermittent sleep related problems [4]. And each year sleep disorders, sleep deprivation and sleepiness add an estimated $15.9 billion to the national health care bill [4]. If progress is to be made in the understanding of what causes these sleep disorders, technology has to play a helping hand. Currently the most popular technologies used in sleep analysis are the Electroencephalograms (EEG), Electromyograms (EMG) and Electrooculography (EOG) recordings [5].

Recently there have been many breakthroughs in genetic studies in humans and rodents which can be used to find the genes that influence sleep disorders by studying sleep patterns in rodents. But if the above stated EEG/EMG/EOG techniques are to be used on rodents to study their sleep patterns it would involve extensive surgery and recovery as genetic studies generally require a large number of animals. This can be both costly and time consuming and would not be practical for large-scale studies. This study was focused on developing a non-invasive method of differentiating between sleep and wake in rodents that is cheaper and easier than surgical methods, and would be ideal for large scale studies. It is cheaper and is also ideal for large scale studies. With this method, rodents with differing sleep patterns can be identified and their genetic variations that underlie these differences studied.

1.2 Introduction to the system

Our technology uses a piezoelectric material as a motion sensor. This material is a thin film that is highly sensitive to any motion on it (a more in-depth description of the piezo film will be presented in the next chapter). The piezoelectric material was placed at the bottom of the mouse cages, with one mouse occupying one cage. During the wake state of the mouse a highly erratic output is generated from the piezoelectric sensor due to

the mouse running about on the sensor, climbing the walls of the cage etc…. But during the sleep state of the mouse, the mouse is laying down on the sensor and the only gross body movement from the mouse is its breathing. These breathing patterns produce a low frequency periodic signal on the sensor. The signal from the sensor is amplified, sampled, and stored for later processing of the data. Once features related to the sleep and wake states were extracted from the data, a linear classifier as well as a Neural Network classifier was used to differentiate between the sleep and wake states (Figure 1.1).

The piezoelectric film was obtained from Measurement Specialties Incorporated. The amplifier was designed as part of this thesis work. The A/D converter was a National Instruments DAQ card (SCB-68), the output of which was read with the aid of National Instruments Labview 7.1 script. The processing of the data, classification and other data analysis was implemented with MATLAB 7.0.

This thesis will discuss a non-invasive automated method for classification of sleep and wake states in mice using features extracted from the data using algorithms such as the Power Spectrum [1], Generalized Spectrum [9] and Autocorrelation [1]. It will first discuss the amplifier that was designed to collect data from the piezo sensor after which it will go through the pre-processing and processing of the signal that was done. The performance of the features extracted from the signal will be examined for a linear classifier along with other parameters such as data segment window sizes and tapering window parameters. As a comparison a nonlinear Neural Network classifier was also used and its performance compared with the linear classifier.

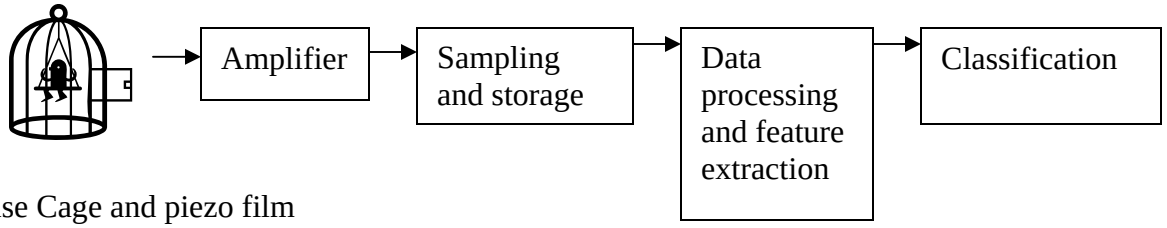


Figure 1.1-Outline of system

1.3 Literature review

Data, such as what was collected, have a random or a stochastic element in its structure. These time signals can be periodic when the mice are sleeping, transient when the mouse is in motion or shifting and stationary noise when the mouse is still and elevated (as is the case when the mouse is hanging on the walls of the cage). Stochastic signals such as this can be either stationary or non-stationary depending on the length of the data segment being analyzed. A stationary time signal is a signal whose mean and variance are constant and do not change with time or position, whereas a non-stationary time signal has changing mean and variance does change with time and position [2]. However, most signals that are outputs of systems such as what was used for this application are more likely to be non-stationary than stationary. The non-stationarities in this applications signal occur mainly due to transients created by the movements of the mice or due to the signals being cyclostationary signals (cyclostationary signals are non-stationary processes whose parameters will vary in a periodic manner [9]). Therefore it is important that the analysis of the data were done for both stationary and non-stationary time series. Methods for processing stationary and non-stationary time series signals were researched and this section will discuss these efforts along with previous studies of mouse behavior using piezo film sensors.

The use of piezo film sensors as transducers to monitor behavioral activity of mice has been done before by Megens and Voetens [6]. In this study the piezo sensors were used to detect the motor activity of the mice to different drugs and different dosages of drugs. In this study, however, the respiratory movements along with the noise were filtered out. The primary signal of interest was from motor movements of the feet of the mice striking the piezo film and the film is sectioned off as to determine the position of the rodent in the cage. But unlike Megens and Voetens work, our study considered both the respiratory movements and the motor movements of the mouse and only noise elements were filtered out.

There have been a number of studies done on detecting periodic behavior in stationary and non-stationary signals. The use of spectral analysis methods such as the Power Spectrum (PS) which is derived from the Discrete Fourier Transform (DFT) for example has been the standard for analyzing stochastic stationary signals in the frequency

domain [1, 2]. But the PS can give ambiguous results when there are non-stationarities in the signal. There are also many alternatives to the Fourier Transform and the power spectrum that have been presented.

For instance there is the Short Time Fourier Transform (STFT) also known as the time-dependent Fourier Transform which analyzes only small segments of the signal at a time [1]. This is done by moving a window of fixed length over the data by a certain increment and at each instance taking the Discrete Fourier Transform (DFT). It would seem like this method would work better for non-stationary signals than the DFT which takes the Fourier Transform for the whole signal. And indeed it is, but this method will be good only for non-stationary time signals whose characteristics will change over time (a chirp signal for instance). But in this application the non-stationarities are mainly due to transients which have no dependence on time and thus the STFT would not be ideal for this application.

There have also been methods such as the Periodicity Transform which was proposed by Sethares and Staley [7]. This algorithm finds its own set of nonorthogonal basis elements (based on the data), rather than assuming a fixed predetermined basis as in the Fourier, Gabor and wavelet transforms. But according to the authors this method performs better than methods such as the DFT only when the periodicity of the signal provides a better explanation of the signal than does frequency [7]. If this is the case then a method such as autocorrelation is a better option as it's more straightforward and easier to use for analysis of a signals periodicity, especially for stochastic signals.

Kanjilal and Palit [8] proposed a scheme for extracting weaker periodic signals in the presence of noise and other stronger periodic waveforms based on the Singular Value Decomposition (SVD). The extraction procedure consists of two steps. First the signal is partitioned into different period lengths. After this matrices are formed with each row having a period of the data (for example one matrix will have rows equal to about 10 samples of data, the next matrix with 20 samples per row and so on). The second step consisted of calculating the Singular values for each matrix. The ratio of the singular values for each matrix will give an idea of periodicity of the signal. The higher the ratio of the singular values, the closer the length of the rows is to the periodicity of that signal. This process was repeated till all the components in the signal were recovered. But

according to the authors this method has its limitations in that if the power of the weakest component happens to be less than that of the background noise or close to it then this method will yield inconclusive results. So in other words this method will likely fail if there isn't a high Signal to Noise (SNR) ratio. For this application the signal out of the piezo film was small (in the 0.5-1mV range) and hence would be susceptible to noise interference. So this method would not have been suitable for this application. This algorithm would also have been computationally expensive as it involved dealing with a large number of matrices which would have taken up a large amount of memory. And considering that this application dealt with large amounts of data, this algorithm again would not have been suitable.

Venkatachalam and Aravena [13] presented a method for detecting periodic behavior in non-stationary signals using Wavelet packet decomposition. This method used the wavelet packets to create low resolution representations of the signal. The assumption that the authors used was that at the low resolutions only the most significant effects of the signal would appear, thus enhancing the periodic behavior and while suppressing the noise and other random effects. After this was completed a processing was done where segments of the data were taken and any outliers were removed (similar to a median filtering of the data). But according to the authors this method worked best only if the signals were highly periodic and would not be suitable for highly irregular signals. For this application, in some instances the data had large and sharp transients which meant that this method would not be ideal.

Gerr and Allen [9] presented the Generalized Spectrum that provides a method for analyzing non-stationary time series by exploiting the phase, particularly in periodic and transient signals. Donohue and Huang [14] used this method with good effect for the classification of ultrasonic backs-scattered echoes resulting from the structures within ultrasonically scanned objects. Donohue and Black [15] used the Generalized Spectrum to detect and estimate pitch in acoustic audio signals. Both these applications [14, 15] used the Generalized Spectrum with much success on non-stationary signals similar to what this application was dealing with. So this method was used in this work in addition to the PS to get complementary information from the frequency domain.

There are also many methods of doing analysis of time series in the time domain of which Autocorrelation [1, 2] is probably the best and most straightforward. Autocorrelation basically measures the correlation between observations at different distances apart and generates autocorrelation coefficients which can be used as a guide to the properties of a time series. Hence for this application Autocorrelation was used as well.

Chapter 2

System Design for Data Acquisition

2.1 Introduction

Data acquisition for this work required an amplifier for reading, filtering and amplifying the data from the piezo film. This chapter gives an introduction to the piezo film and also the amplifier used to obtain the data on which the classifier was built and tested. This chapter also discusses the ideal and non-ideal responses of the amplifier along with issues related to the A/D converter that was used.

2.2 Description of piezo film sensor

The piezo film is made out of a very thin PVDF (Polyvinylidene fluoride**),** polymer. One of the primary uses of this film is to convert mechanical energy to electrical energy and thus was used as a transducer in our application. Its extra sensitivity also made it an ideal sensor for our application. If a vertical force is applied on this film it results in a voltage proportional to the force applied on the film. An electrical model of the film is a voltage source in series with a capacitor as shown in figure 2.1 [11], where Vs is the voltage created by the force on the piezofilm and C is the capacitance of the piezofilm material.

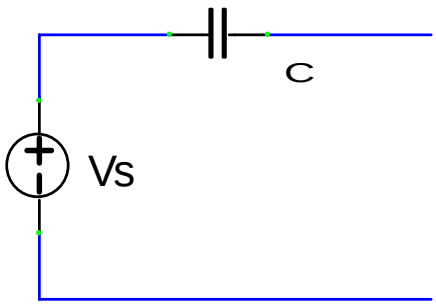


Figure 2.1 - Electrical model of Piezo Film

The capacitance “C” of the piezo film can be calculated using the following formula:

$$C = \varepsilon \frac{A}{t} \qquad (2.1)$$

Where “A” is the area of the film, “t” is the thickness, “C” is the capacitance and ε is the permittivity that can also be expressed as

$$\varepsilon = \varepsilon_r \varepsilon_o \qquad (2.2)$$

where ε_r is the relative permittivity equal to about 12 for PVDF and ε_o is the permittivity of free space (a constant with value 8.854×10^{-12} F/m) [11].

For this application 7.5cm by 15cm piezo films were used. The thickness of the film was 28 microns. Hence the total capacitance of the film was 40nF as calculated from equation 2.2:

$$C = (12 \times 8.854 \times 10^{-12} F/m) \frac{(.075m \times .15m)}{(28 \times 10^{-6} m)} = 40nF$$

To get a larger sensing area two Piezo films were placed side by side and connected in series as shown in figure 2.2.

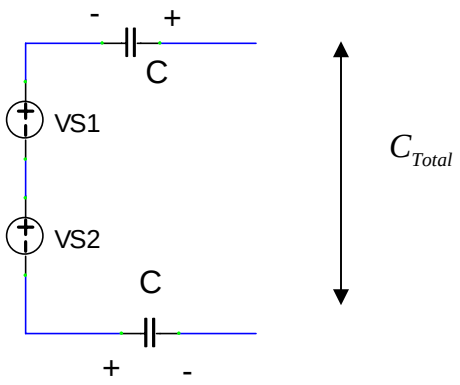


Figure 2.2 – Electrical model for the series connection of two piezo films. The capacitors will have opposite polarities.

Hence the total capacitance C_{total} for the two Piezo film sensors connected in series is:

$$\frac{1}{C_{total}} = \frac{1}{C} + \frac{1}{C} = \frac{2}{C} \Rightarrow C_{total} = \frac{C}{2} = 20nF \qquad (2.3)$$

2.3 Amplifier Design

Figure 2.3 shows the design of the amplifier that was used. It consisted of 4 parts, the piezo film sensor input, a differential amp, a low pass filter and a non-inverting amplifier. The amplifier was designed so that the frequencies of interest were not attenuated in any way. The frequencies of interest were in the 1-10Hz range. This was because the fundamental frequency of the sleep signals from the mice was found to be typically in the 2-3Hz range, and its significant harmonics in the 4-9Hz ranges.

Since the electrical model of a piezo film sensor is a capacitor in series with a voltage source, the sensor exhibits very high output impedance. This is due to the fact that the piezo film sensor has a very low capacitance. In addition, the application requires measuring signals at low frequencies (<20Hz). So the piezo films output impedance, $1/2\pi fc$, will be very large. Hence a high impedance buffer is needed to get good responses at the low frequencies.

For this application a differential operational amplifier was used as the buffer. The main reason for choosing a differential amplifier is because it does not amplify noise induced on both capacitor surfaces as dramatically as the voltage difference between the surfaces resulting from the piezoelectric signal. Noise such as line noise due to cross talk and poor connections as well as thermal noise from the resistors can add common mode noise to the signal from the piezo film. But the differential amplifier suppresses this common mode noise and instead enhances the differential voltage, which is primarily our signal of interest [11].

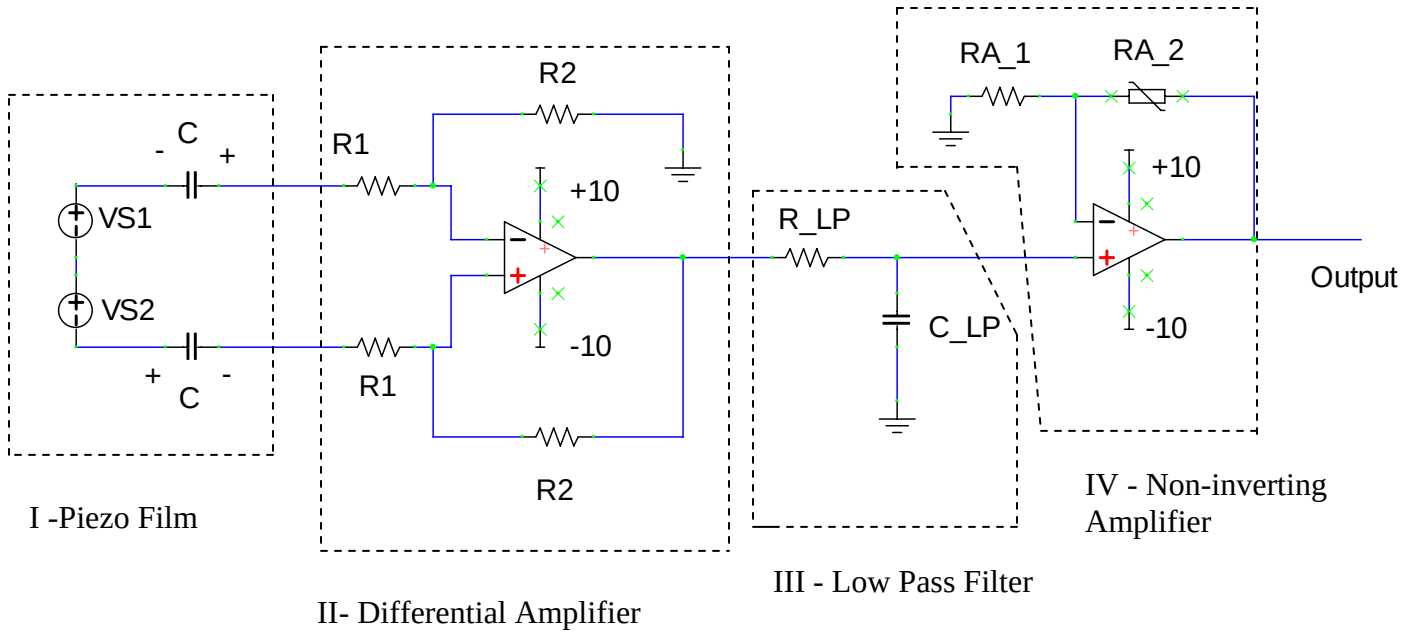


Figure 2.3 – Layout of the amplifier.

As can be seen from figure 2.3, the capacitance of the piezo film and the two input resistances (R1) of the differential amplifier are in series which means that the piezo film and the differential amplifier will also act as a first order high-pass filter with a cutoff of

$$Cutoff\_highpass = \frac{1}{2\pi(R1+R1)C} \quad (2.4)$$

And a gain of

$$Gain = \frac{R2}{R1} \quad (2.5)$$

The input resistances R1 were chosen to be $3\text{Meg}\Omega$ each, so that the total input resistance to the differential amp is $6\text{Meg}\Omega$. This results in a cutoff of the highpass filter at around

$$\frac{1}{2\pi(R1+R1)C} = \frac{1}{2\pi(3\times10^{6}+3\times10^{6})(20\times10^{-9})} = 1.2Hz \quad (2.6)$$

This cutoff value was chosen so that it does not dampen the frequencies in the range 2 - 3Hz, which is the breathing rate of the mouse when sleeping. It was also chosen so that any DC frequency components lower than 1Hz was also dampened. The feedback

resistances R2 were chosen as 10MegΩ which meant a gain of $\frac{R2}{R1} = \frac{10}{3} = 3.33$ coming out of the differential amplifier. A disadvantage of having such large resistor values is that it increases the thermal noise (thermal noise is proportional to resistor values). But the attenuating of this type of noise by the differential amplifier should minimize any unwanted effects from these large resistors.

The piezo film sensor also picked up not only the signal from the mouse but also signals from other sources such as power supplies and fluorescent lighting. Probably the main source of outside interference was from the 60 Hz power supply that was used to power the amplifier. The reason this 60 Hz signal interfered with the piezo signal is due to the weakness of the piezo signal relative to the power supplies 60 Hz ripple (It should be noted that the piezo signal was in the 0.5-1mV range before any amplification). This created a significant spike in the frequency spectrum at the 60Hz point which was clearly visible when the actual impulse response of the amplifier was plotted out (see figure 2.5). To reduce the impact of this 60Hz signal on the quantization and saturation of this analog signal at the A/D conversion, a first order low-pass filter was employed on the signal from the differential amp. The cutoff was chosen as follows:

$$\frac{1}{2\pi(R\_LP)(C\_LP)} = \frac{1}{2\pi(620)(10\times10^{-6})} = 26Hz \quad (2.7)$$

To make the roll off in the transition region steeper a higher order low-pass filter was considered, but this required more circuitry for little signal improvement, and thus only a first order filter was used. This cutoff was also chosen so as not to interfere with harmonics for the critical 2-3 Hz breathing oscillations. This meant that stages I, II and III of the amplifier formed a band-pass filter with cutoffs at 1.2Hz and 26Hz.

The signal from stage III of the amplifier was in the order of about 1-2mV for a sleep signal. Hence a non-inverting amplifier with a gain of about 50-75 (17dB-19dB) was added to amplify the signal before sending it on the cable for the A/D conversion.

Figure 2.4 shows the frequency response of the amplifier obtained by modeling the ideal amplifier in SPICE. The resulting band-pass filter cutoffs are also shown.

Looking at the ideal frequency response it is clear that there is only about a -3dB relative attenuation between the maximum at 6Hz and the cutoffs at 1.2Hz and 26Hz.

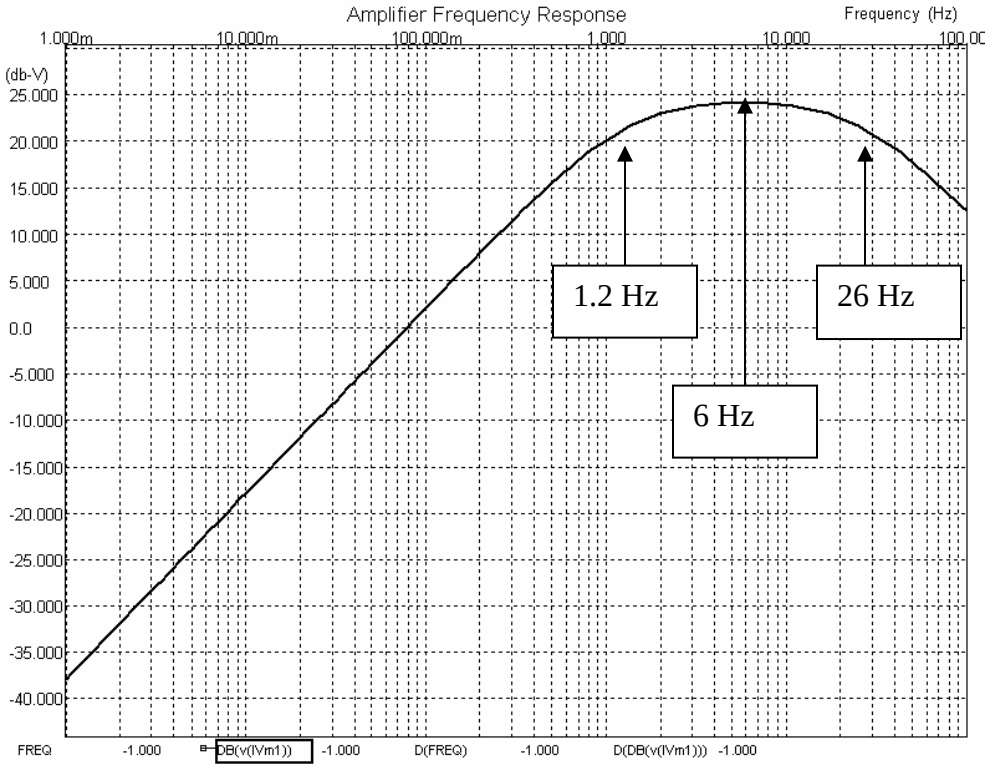
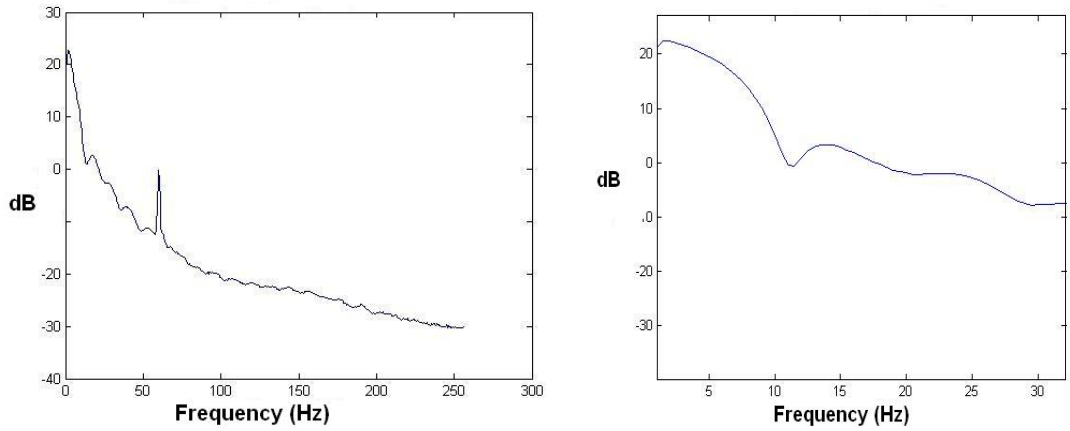


Figure 2.4 - The frequency response of the amplifier in a log-log scale.

But to get an idea of what the response of the actual circuit was we attempted to get an impulse response from the amplifier. This was done by a quick vertical tapping of the piezo film with the tip of a pen, thus simulating an impulse. A power spectral density plot was then done on this data. The result can be seen in Figure 2.5 below.

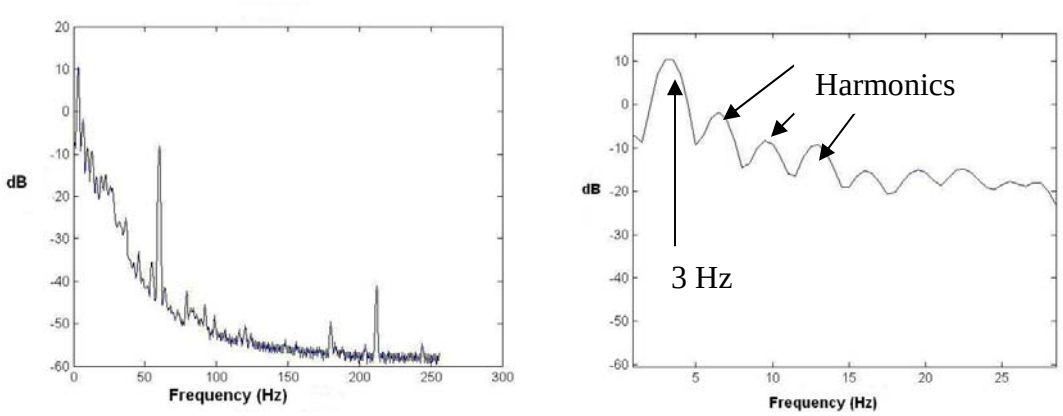


(a) (b)

Figure 2.5 (a) Impulse response of the amplifier (b) Impulse response of amplifier zoomed in to the 0-30 Hz range.

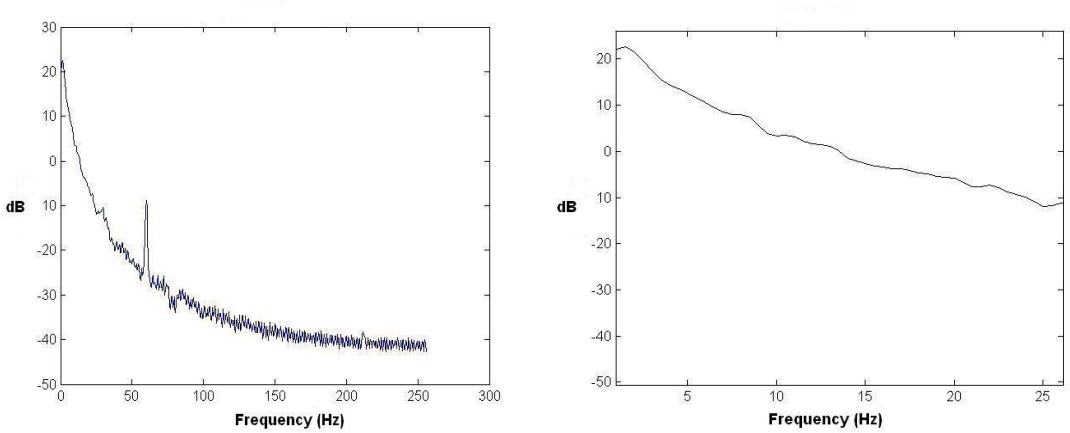
The actual impulse response of the amplifier shows a relatively flat response between 0-6Hz and a rolloff between 6Hz and 26Hz. This is due to the fact that an ideal impulse response cannot be imparted on the piezo film. To verify this, the response from 0-6Hz can be modeled as a sinc function with a main lobe width of 12Hz (-6Hz to +6 Hz). This will translate into the time domain as a square wave with a width of 1/12= 0.08 sec. So this is roughly the width of the impulse that we are imparting on the piezo film. If we are to impart a better impulse which will give a flatter response we will have to give an impulse smaller than 0.08 sec which would be impractical given the inertia of the mechanical elements involved. Due to this limitation we will never be able to get the theoretical frequency response in practice.

As an example Figures 2.6 and 2.7 shows some power spectral density plots of sleep and wake data from a mouse. Figure 2.6 is a typical sleep signal with its 3Hz fundamental frequency and its harmonics whereas in the wake state you will see a flatter spectrum with no peaks.



(a) (b)

Figure 2.6 (a) PSD plot of a 30 second sleep signal which was sampled at 512 Hz (b) The PSD plot zoomed in to the 0-25Hz region.



(a) (b)

Figure 2.7 (a) Power spectral density plot of a 30 second wake signal sampled at 512 HZ (b) PSD plot zoomed into the 0-25Hz range

2.4 A/D converter

The output of the amplifier was run into an A/D converter. The A/D converter that was used was a SCB-68 16 channel National Instruments DAQ. This uses a 16 bit quantizer. The inputs to the DAQ device were configured in single-ended non-referenced mode. The DAQ card was connected to a computer via a National Instruments PCI-6224 PCI card. The DAQ card was read via a Labview 7.10 script. The A/D converter was

sampled at 128 samples/sec. This sampling rate insured that there would be no aliasing from the 60Hz frequency which was the highest significant frequency in the spectrum. Thus the Nyquist rate of (2x60) Hz was met with this sampling rate. This sampling rate also insured that calculations involved in the analysis did not have too many data points. It should be noted that the data being collected was for long periods, maybe 24 to 48 hours at a time. Hence if there were too many data points the calculations would have taken longer and issues such as computer memory and hard disk space would have also arisen. The quantization levels ranged from ± 5 at 16 bits/sample. The sampled signals were then stored in binary format to save hard disk space and then later read in by MATLAB for the analysis.

2.5 Summary

The main purpose was for the amplifier to deliver a good clean signal from the piezo film sensor with less noise as possible. Also it was important that the 60Hz frequency from the power source had little or no effect on the lower frequencies. And this circuit did quite well in this aspect as the low pass filter and the selecting of the proper sampling rate limited this interference. Although in practice the circuit did not give the ideal response, it gave a good response in the frequency range 1-10Hz which was what was needed. The rolloff after this frequency range was more beneficial than harmful as it would attenuate the 60Hz and other noise components even more.

Chapter 3
Algorithms used to extract features

3.1 Introduction

To build a classifier to classify between sleep and wake states, features must be identified and extracted from the data and used to train and test classifiers. Features can be evaluated by their impact on the performance of classifiers in which they are used. This chapter will first discuss how the short data segments were extracted for data analysis, after which it will discuss the nature of the features extracted from these short data segments. It will also examine how these features performed for the sleep and wake states.

When analyzing a stochastic time series signal such as this it is important to take into consideration that the signal may not be stationary over the period from which the features are extracted. In this data analysis, short segments of data were taken in order to increase the likelihood of stationarity. However, if these short data segments have strong transients like when the mouse is running around on the sensor, the assumption of stationarity does not hold, especially for short segments. If this data segment had one or two transients (which can occur when the sleeping mouse makes a slight movement) the segment cannot be considered stationary either. Hence to assume that the data in this segment would be purely stationary or non stationary would be a false premise. Also it should be noted that time series data will be correlated at different time instants, especially for periodic data that the sleep signal gives. So it is important that all these factors are taken into consideration when analyzing a time series signal and extracting features from it. For this application 23 features were extracted from the data for consideration of their use in a classifier. There were 3 primary algorithms that were used to create functions from which these 23 features were extracted. These 3 algorithms are listed below.

1) Power Spectrum (PS) features
2) Generalized Spectrum (GS) features
3) Autocorrelation (AC) features

The Power Spectrum (PS) was used to extract features directly from the frequency domain. These features provided information on which frequencies in the signal contained the most power. During sleep most of the power will be concentrated in the lower harmonic frequencies i.e. in the fundamental frequency and its harmonics, whereas during the wake state it will be spread out over all frequencies (no harmonic patterns). The Generalized Spectrum (GS) was also used to extract features from the frequency lag or difference domain. Unlike the power spectrum, this method uses additional information on the phase of the signal and correlations between the distinct frequency components. The autocorrelation was used to extract features from the time lag or difference domain. Autocorrelation gives an idea of the periodicity of the time domain signal. It also gave information on the correlation of the time signal as a function of time domain lags.

3.2 Extraction of data segments

The processing and the extraction of features from the data were done on 8 second windows of data (the criteria for selecting an 8 second window will be discussed later). This window was then moved by a 4 second increment and another set of 8 second data was processed and features extracted. This process continued till the whole data set was analyzed. So the data was extracted by using an 8 second moving window with 50% overlap.

A test was done to see how many seconds the window should be incremented by before it was decided to use the 4 second increment. This was done by seeing how the 23 extracted features performed individually in the classification procedure (classification will be discussed later) for increments of 1, 2, 3 and 4 seconds. The window size was kept constant at 8 seconds. The performance was measured by the misclassification errors from the classifier. But it turned out that the increment size did not have any impact at all in the classification procedure as the misclassification errors were the same regardless of the increment size. Hence increments of 4 seconds were chosen as the data analysis progressed faster due to less data segments being analyzed.

3.3 The Power Spectrum (PS)

One set of features was derived from the power spectrum of the data where the absolute value of the DFT of the data segment was taken and then normalized by the energy of the spectrum.

The discrete data segment x[n] of length N samples is a concatenation of N/2 samples from the previous data segment and N/2 samples from the current data segment due to the 50% overlap discussed earlier. This data segment is then windowed with a Kaiser Window w[n] also of length N samples (the parameters used for the Kaiser Window will be discussed in a later chapter). The DFT of this windowed data segment is then taken resulting in the DFT vector Y[F]. This process can be represented as follows:

$$Y[F] = \sum_{n=0}^{N-1} \left(w[n] x\left[n - N + \frac{N}{2} \right] \right) e^{-\frac{j2\pi F}{N}} \qquad (3.1)$$

Where F = 0, 1….N

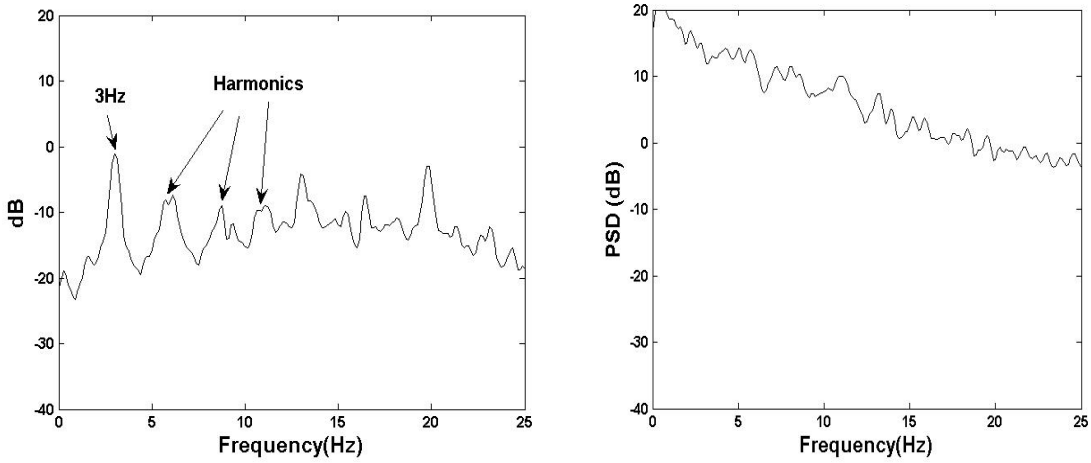
(It should be noted that the DFT is symmetric about DC and hence only one side of the DFT were considered).

Y[F] is then energy normalized. Thus the Energy Normalized Power Spectrum (ENPS) can be written as:

$$Y[F]_{ENPS} = \frac{|Y[F]|}{\sqrt{\Sigma(|Y[F]|)^2}} \qquad (3.3)$$

The reason for the energy normalization was so that different depths of the spectrum will be more uniform across the data segments being analyzed. In other words normalization removes any large scale energy variations that may occur in the DFT values.

Figure 3.1 shows what the energy normalized power spectrum looks like during sleep and wake states. The use of the PS was useful when the data segment being analyzed was stationary.



(a) (b)

Figure 3.1(a) Power spectrum of a sleep signal sampled at 128Hz showing a fundamental frequency at 3Hz and its harmonics. (b) Power spectrum of a wake signal sampled at 128 Hz showing a flatter spectrum and no prominent peaks.

3.4 The Generalized Spectrum and the Collapsed Average.

3.4.1 The Generalized Spectrum (GS)

Spectral analysis of time series is well known and has been widely used throughout the years. But almost all spectral analysis methods are ideal only for frequency domain analysis of stationary signals and can produce ambiguous results for non-stationary signals [9]. A spectral analysis method like the power spectrum for example is the expected value of the squared magnitude of the Fourier Transform of a signal and thus does not preserve phase information of the signal. Whereas phase is irrelevant in the analysis of random stationary signals, it can be an important characterization parameter for random non-stationary signals such as transient or cyclostationary signals (cyclostationary signals are non-stationary processes whose parameters will vary in a periodic manner [9]). The sleep signal from the mouse for instance can be considered as periodically correlated or cyclostationary and the wake signal can be considered a series of transients. Thus a method for separating these non-stationarities from other stationarities such as noise would be useful for this application. The Generalized Spectrum (GS) is different from the PSD in that it preserves the phase

information of the signal and thus more useful for non-stationary analysis. So the GS was used to see if the impact of the phase information provided complimentary information and thus improved detection performance.

The GS is in fact an extension of the power spectral density, but the non-stationarities of the signal can be characterized by using the frequency domain repetition and the phase information. The GS is defined in the bi-frequency plane as follows [12]:

$$G(F_1, F_2) = E\left[Y(F_1).Y^*(F_2)\right], \tag{3.4}$$

Where $Y(F_i)$ is the Discrete Fourier Transform (DFT) of the signal *y(t)* at F_i. The E[.] is the expected value and the * indicates the complex conjugate [12]. This amounts to taking the outer products of the DFT vector and its conjugate which results in an NxN matrix where N is the length of the DFT vector. This multiplication and the resulting matrix G are shown below:

$$G = E\left(\begin{bmatrix} Y(F_N) \\ Y(F_{N-1}) \\ ..... \\ ..... \\ Y(F_2) \\ Y(F_1) \end{bmatrix} .\begin{bmatrix} Y^*(F_1) & Y^*(F_2) & .... & .... & Y^*(F_{N-1}) & Y^*(F_N) \end{bmatrix}\right) \tag{3.5}$$

$$= E\begin{pmatrix} Y(F_N)Y^*(F_1) & Y(F_N)Y^*(F_2) & ... & ... & Y(F_N)Y^*(F_{N-1}) & Y(F_N)Y^*(F_N) \\ Y(F_{N-1})Y^*(F_1) & Y(F_{N-1})Y^*(F_2) & ... & ... & Y(F_{N-1})Y^*(F_{N-1}) & Y(F_{N-1})Y^*(F_N) \\ \vdots & \vdots & & \iddots & \vdots & \vdots \\ \vdots & \vdots & \iddots & & \vdots & \vdots \\ Y(F_2)Y^*(F_1) & Y(F_2)Y^*(F_2) & \dots & \dots & Y(F_2)Y^*(F_{N-1}) & Y(F_1)Y^*(F_N) \\ Y(F_1)Y^*(F_1) & Y(F_1)Y^*(F_2) & \dots & \dots & Y(F_1)Y^*(F_{N-1}) & Y(F_1)Y^*(F_N) \end{pmatrix}$$

Here the main diagonal will have real values (zero phase) and the off diagonals will have complex values. Since the main diagonal results in a square of the DFT and real values (since you are multiplying a complex number by its own conjugate) it represents the PSD of the data segment if the data segment is stationary. On the other hand the off diagonals

will represent the phase coherence between different frequency components and can characterize structure in the data. For example the first off diagonal will show the phase coherence between frequencies separated by 1Hz, the second diagonal between frequencies separated by 2Hz and so on. Also it should be noted that when doing analysis with the GS matrix only the upper half is useful as the lower half is just the complex conjugate.

3.4.2 Normalization of GS

There were two methods of normalization used on the DFT values before the GS was calculated; they are the system normalization and the energy normalization methods [12].

The system normalization method is defined as follows:

$$Y(F_i)_{SN} = \frac{Y(F_i)}{|Y(F_i)|} \quad \text{Where i = 1….N} \qquad (3.6)$$

Where $Y(F_i)$ is the i^{th} component of the DFT vector of the data segment being analyzed. This normalization will scale all the magnitudes to unity, but will still preserve the phase. The main advantage of system normalization is that it reduces the effects of the system on the DFT values and also reduces the energy variation from segment to segment. But at the same time it can scale up the noise levels especially from spectral areas where the initial amplitudes were low. This can degrade the detection of the actual signal and disrupt any phase coherence that may exist. And for this application the system normalization did scale up the noise and in many cases reduced the signal-to-noise ratio as can be seen from figure 3.2 (b). Thus for this application system normalization was not suitable and hence was not used.

Energy normalization is defined as follows:

$$Y(F_i)_{EN} = \frac{Y(F_i)}{\sqrt{\sum_i |Y(F_i)|^2}} \quad \text{For i = 1,2,3…N} \qquad (3.7)$$

Where $Y(F_i)$ is the i^{th} component of the DFT vector and N is the length of the DFT vector. This normalization method reduces the variance of the energy in the power spectrum and makes comparing the spectrum from different data segments easier. But it also has its disadvantages in that it can increase the energy in parts of the spectrum that does not display any peaks and has less energy, which can also degrade the detection of the signal.

But for this application this energy normalization method worked better than the system normalization as can be seen from figure 3.2. For this data segment the GS is well formed with prominent periodic peaks for the energy normalized GS (figure 3.2 (a)) whereas the system normalization (figure 3.2 (b)) has scaled up the noise quite significantly thus drowning out the peaks.

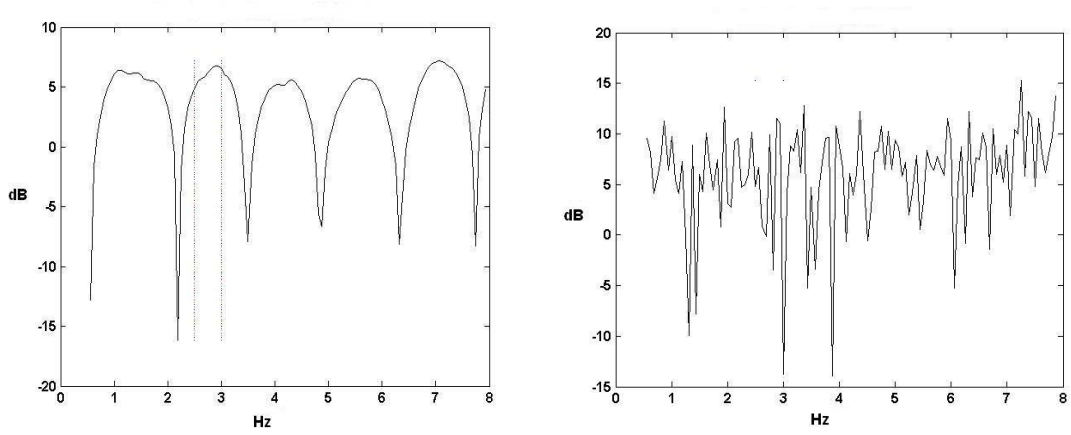
3.4.3 The Collapsed Average (CA)

In typical PSD estimations such as the Periodagram for instance, averaging is employed and likewise averaging can be used for the GS as well. For this application the Collapsed Average (CA) of the GS [12] was calculated as a method of averaging the GS. In cases where the SNR is constant over the spectrum, the CA is used to collapse the 2-dimensional GS matrix into a 1-dimensional vector. This also decreases the computations necessary during the averaging process, as now the CA is only a 1-dimensional vector for each data segment. The CA is calculated by taking the coherent mean along each diagonal of the GS and plotting the magnitude of this as a function of the frequency difference that each diagonal represents. The CA can be described as follows :

$$CA = \frac{\sum_{p-q=j} G(p,q)}{\sum_{p=q} G(p,q)} \qquad (3.8)$$

Where the numerator is the sum of all the values along the diagonal j of the GS matrix and the denominator is the sum along the first diagonal (which is the PSD). So the numerator does an arbitrary normalization of the CA using the PSD values. Figure 3.2

below shows the CA for a data segment with Energy Normalization and System Normalization.



(a) (b)

Figure 3.2 (a) CA of an 8 seconds sleep segment of data that was energy normalized. (b) the same segment of data that was system normalized.

3.4.4 Fast calculation of the CA

The calculation of the CA will still involve a larger number of calculations. First the DFT values had to be calculated and then the GS matrix calculated from which the CA is derived. So a faster calculation method to compute the CA was used [12]. This involved using the relationship between the PSD and autocorrelation. It should be noted that the GS is basically an autocorrelation in the frequency domain (see equation 3.4). Hence the above definition of the CA in equation 3.8 can be written as:

$$CA(F) = \frac{R_{YY}(Y(F))}{[R_{YY}(Y(F)),1]} \qquad (3.9)$$

Where the numerator is the autocorrelation of the DFT vector $Y(f)$ and the denominator is the first element of this autocorrelation vector. But since the definition of autocorrelation in a random process is the inverse Fourier Transform of the PSD, the

numerator of equation 3.9 can also be calculated as follows:

$$R_{YY}(Y(F)) = IFFT(\left|FFT(y(t))\right|^2) \qquad (3.10)$$

Figure 3.2 above shows the CA plot computed using the fast calculation method for a sleep data set.

3.4.5 Result of the CA for sleep data

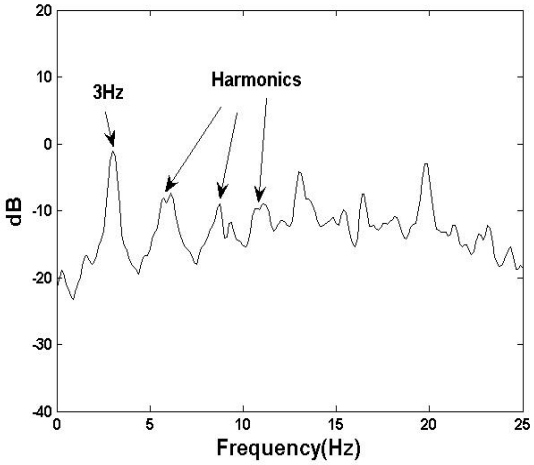
A time series that is periodically correlated can be modeled as a cyclostationary random process. If the data segment being analyzed is purely a sleep signal then this can be classified as a non-stationary cyclostationary signal. The phase of this type of non-stationary signal will be non-linear as can be seen from figure 3.3 (b). For such a cyclostationary signal with period T the CA coefficients have the following properties:

$$CA(F_j) > 0 \quad \text{for} \quad F_j = \frac{k}{T}, \quad k = 1,2,3.... \qquad (3.11)$$

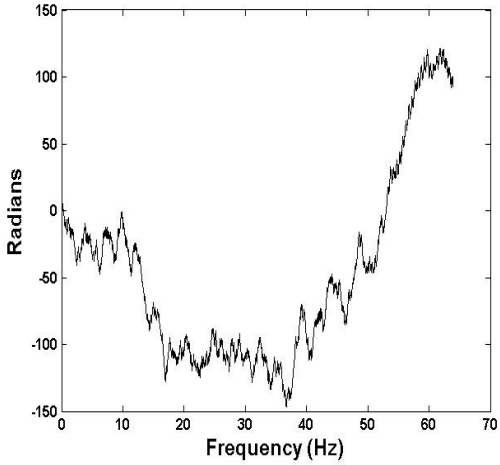
and $CA(F_j) = 0$ for $F_j =$ other frequencies

What this means is that for a cyclostationary sleep signal with period T, the CA will display peak values at the fundamental frequency and its corresponding harmonics thus displaying correlation between these frequencies and coherency in the phase of the signal at these frequencies. It will display zero values or values close to zero at all other frequencies.

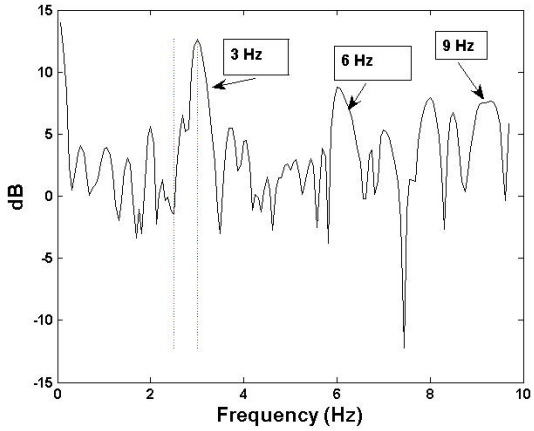
As stated earlier for this application the fundamental frequency of the breathing of the mouse during the sleep state is around 3Hz which means that the CA will have peak values at the frequencies 3, 6, 9 Hz etc… (how prominent these peak values are will depend on how well the spectrum is formed and how many harmonics there are in the spectrum). Figure 3.3 (c) shows the CA for a sleep data segment and the prominent peaks at the 3, 6, 9 Hz. What this signifies is that there is high correlation between frequencies separated by 3, 6 and 9 Hz in the DFT. Looking at Figure 3.3 (a), the fundamental frequency and harmonics clarify this fact.



(a)



(b)



(c)

Figure 3.3 (a) PSD of a sleep signal samples at 128Hz (b) The unwrapped phase of the same signal showing non-linear phase (c) The GS of this sleep segment showing the peaks at 3,6,9 Hz and the correlation between these frequencies.

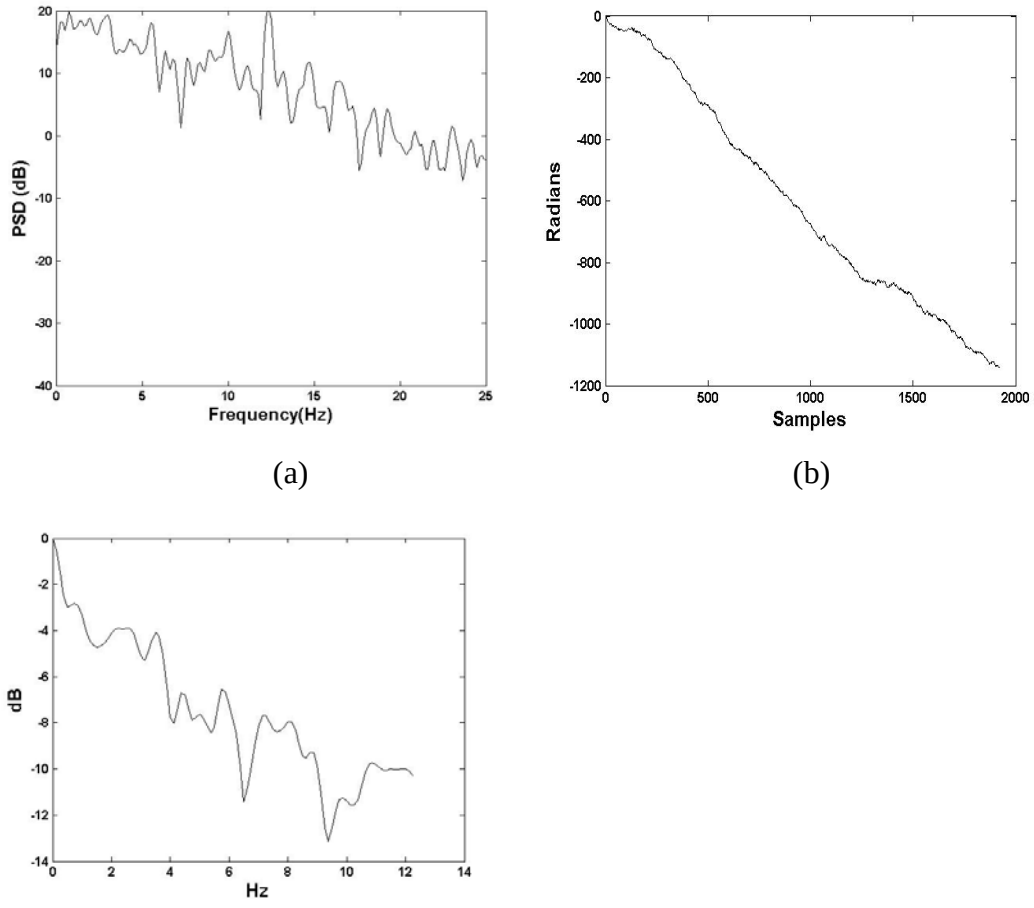
3.4.6 Result of CA for wake data

The wake state of the mouse can be characterized as a series of transient non-stationary signals with the mouse running about on the sensor, climbing the walls etc... But what will the result from the CA be for such a wake signal? Since the wake signal is a series of transients or impulses where the position and strength will be totally random,

the phase of this type of signal will also be linear (see Figure 3.4 (b)) due to the energy of the signal being highly localized. Hence in this case there will be phase coherence between all the frequencies. So the CA will display the following relationship:

$$CA(F_j) = 0 \text{ for any } F_j \qquad (3.12)$$

Figure 3.4(c) below shows the CA for a wake signal without correlation between the peaks. It also shows a rolling off of the CA due to a lack of coherence between any of the frequency components.



(a) (b)

(c)

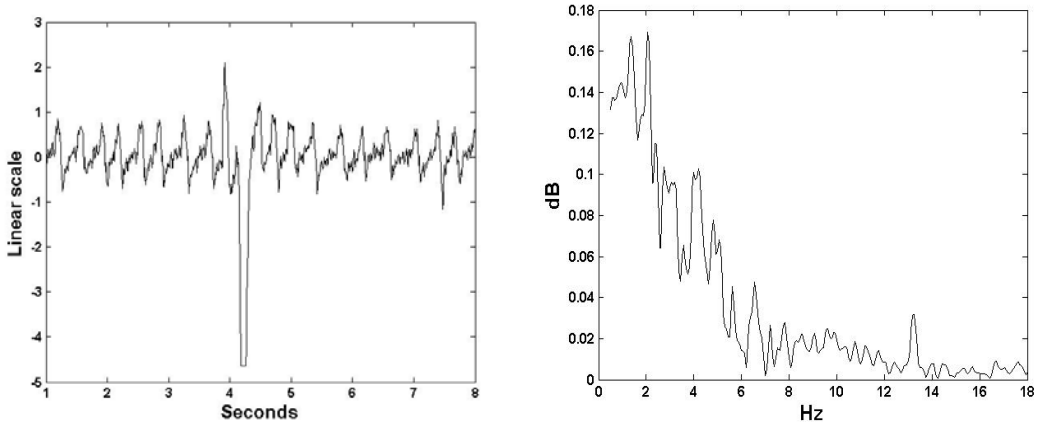
Figure 3.4 (a) PSD of wake signal (b) unwrapped phase of wake signal showing a linear phase (c) The CA of the wake signal

3.4.7 Result of Power spectrum and CA for mix of sleep and wake data

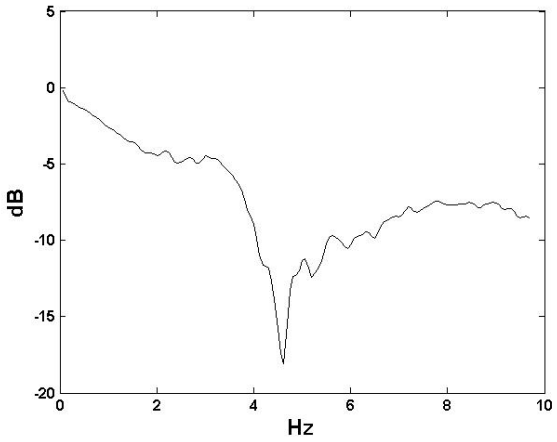
Now the question arises as to what the result will be from the GS and PS if the data segment being analyzed has sleep and wake states in it. Figure 3.5 (a) below shows an eight second sleep data segment with one sharp strong transient probably caused by a sudden movement of the mouse while sleeping.

Sharp transients such as this are going to have a very low frequency. For the data segment in the figure the transient lasts for about .45 seconds which translates to about 2.2 Hz in the frequency domain. Sharp transients such as this will also have very broad frequency spectra which will result in the frequency spectrum spreading out in this 0-2.2 Hz range. This transient will also contain more power compared to the rest of the data segment, which will correspond to higher peaks in this 0-2.2 Hz frequency range. These lower frequencies will then swamp out any frequencies outside this range thus flattening out any peaks in the PS caused by the sleep segment of the data in the. Figure 3.5(b) shows this very clearly. The CA will also show no prominent peaks and a rolling off due to the lack of coherence between any of the DFT frequencies (see Figure 3.5 (c)).

It should be noted however that if the transient has a longer duration in the time domain (not as sharp a transient as shown in Figure 3.5(a)), then the frequency spreading in the lower frequencies will not be that prominent and hence the CA and PS can pick up the 3Hz sleep signal and its harmonics.



(a) (b)



(c)

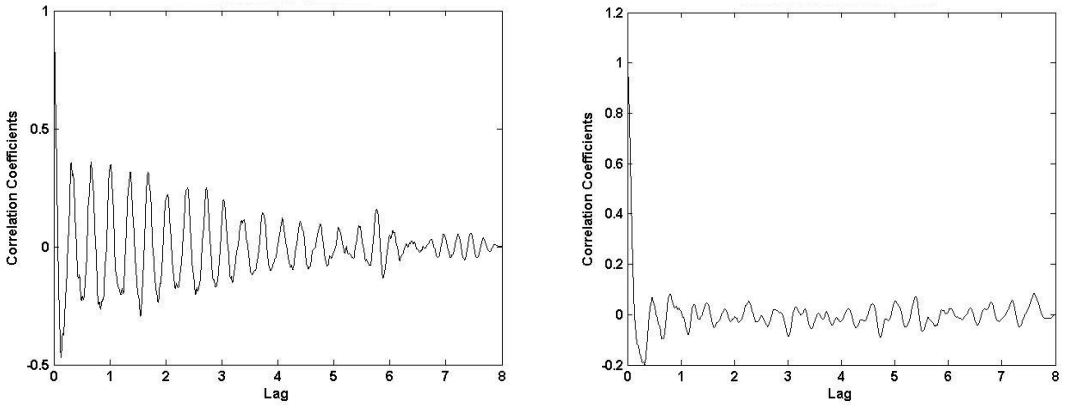
Figure 3.5 (a) 8 seconds sleep data segment with sharp transient (b) The PSD of this data segment (c) The CA of this data segment

3.5 Autocorrelation (AC)

The previous methods of extracting features were all from the frequency domain. So autocorrelation was used as a means of having some features from the time domain. Autocorrelation is used to detect non-randomness in the data and periodicities. It is defined as follows for a time series X(t) [1,2]:

$$R_{XX}(t_1, t_2) = E\{X^*(t_1)X(t_2)\} \quad (3.13)$$

where $R_{XX}(t_1, t_2)$ are the autocorrelation coefficients. Hence when there is a periodic sleep pattern in the data the autocorrelation coefficients will show periodic peaks and show random values for the wake state. Figure 3.6 shows the results of autocorrelation on sleep and wake signals.



(a) (b)

Figure 3.6 (a) Autocorrelation of sleep signal (b) Autocorrelation of wake signal of an 8 second window.

The autocorrelation plot for the sleep signal is typical of a stochastic periodic process in that as the time shifts get longer, the autocorrelation coefficients get smaller and smaller (for a pure deterministic sine wave for example the amplitude and phase will be constant through all the time shifts).

Chapter 4
Processing of signal

4.1 Introduction

Before doing any analysis on empirical data it is always useful to remove outliers, smooth or detrend the data. This chapter will discuss the pre-processing stage that was done on the raw data to remove elements of the signal not related to the sleep and wake signatures. It will also discuss a processing of the DFT data that was done.

4.2 Pre-processing of data

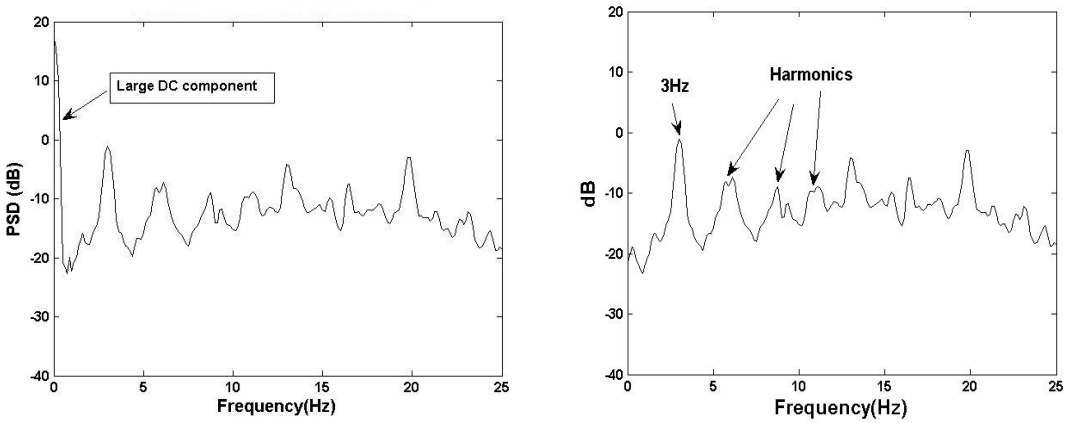
Both the PS and the GS involved taking the DFT of the data segments. But before the DFT of the data was taken steps were taken to reduce phenomenon such as spectral leakage. When a finite segment of data is taken from an infinitely long data stream (as is done in this application), there will be discontinuities at the ends of the data segment. These sharp discontinuities have broad frequency spectra and can cause a spreading out of the signals frequency spectrum. This spreading means that the signal energy that should be concentrated only at one frequency instead leaks into the other adjoining frequencies. This spreading is known as spectral leakage. One way to prevent this is by making the ends of the data segment go to zero instead of having the discontinuities. Applying a tapering window to the data segment will do just this. So to reduce the effects of spectral leakage the data segments were first multiplied by a Kaiser window and then the DFT taken. The Kaiser window was chosen as it gives more flexibility over the main-lobe and sidelobe widths and thus more control over the spectral leakage [1]. The parameters of the Kaiser window used will be discussed in a later chapter.

The data segments were also zero padded before taking the DFT so that the spectrum would be sufficiently oversampled. This oversampling of the DFT will result in the spectrum being evaluated on more grid points, and thus the important peak frequencies will be better resolved on the grid axis.

The data collected from the mice can also have trends in them that can have unwanted effects on the frequencies in the spectrum. The data can have aperiodic non-stationary components that influence the low frequencies of the spectrum. At times the

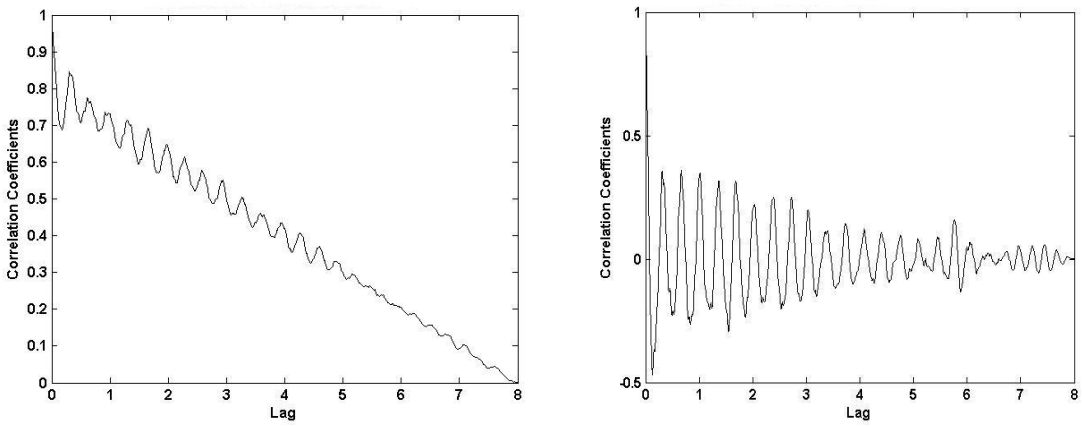
data can also exhibit a drift or offset in it, and for this application this was observable at times (most likely created by the slow response of the op amp filter, whose bandwidth ranged from 1.5 Hz to 10 Hz). This can also introduce a trend in the data. Even if the data does not have anomalous trends from the electronics, if segment of data is taken from a larger data series it can exhibit an oscillation whose period is larger than the segment. This too can introduce trends. For example taking a segment of data from a larger series of sleep data will be like introducing ramp like trends to the data, which can lead to spurious components in the PS and autocorrelation coefficients, especially at low frequencies. So to remove these trends a linear detrending was done on the raw data. For this application removing these linear trends helped remove ramp like trends from the autocorrelation that tended to obscure critical features, and likewise removed strong low frequency artifacts from the PSD that can potentially obscure signal activity in the critical 1 to 4 Hz range. The detrending, however, did not seem to have a significant effect on the CA. Figure 4.1 and Figure 4.2 shows a data segment with and without the detrending for the PS and autocorrelation. Figure 4.1 (a) shows the PS with a very large DC component and Figure 4.1(b) shows the PS with the DC component removed due to the detrending. Figure 4.2 (a) shows the rolling off of the autocorrelation coefficients due to a linear trend and Figure 4.2 (b) shows the coefficients with the trend removed due to detrending. Figure 4.3 clarifies the fact that the data detrending did not have any effect on the CA as the plots are the same with and without detrending.

Applying a bandpass linear digital filter to the raw data was also considered, but this seemed to attenuate some of the lower frequencies in the critical range and hence was not used in the final analysis.



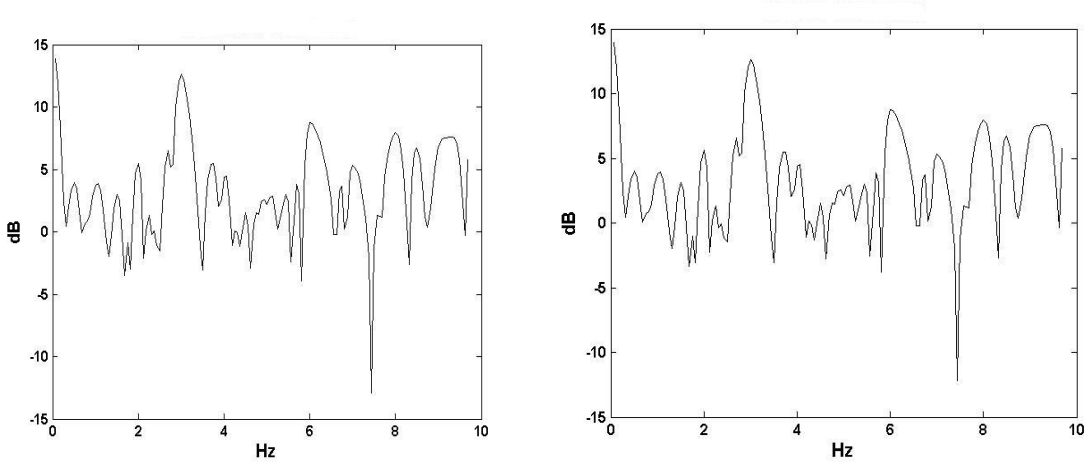
(a) (b)

Figure 4.1 (a) PS of data segment of 8 seconds without detrending (b) PS of detrended data segment.



(a) (b)

Figure 4.2 (a) Autocorrelation without data detrending (b) Autocorrelation with data detrending



(a) (b)

Figure 4.3 (a) CA without data detrending (b) CA with data detrending

4.3 Processing of data

Removal of linear trends from the raw data only removes the low frequency and DC components from the PS and reduce the ramp and parabolic artifacts of the Autocorrelation. But the PS and CA still display a rolloff proportional to 1/f in the frequency domain (see Figure 4.4 (a) and Figure 4.5(a)). This is more evident in the PS than the GS. This frequency domain trend is most likely due to the natural attenuation of high frequencies in natural data and the roll-off of the electronic filters. Thus the lower frequencies tend to attenuate less than the higher frequencies. The sloping of the spectrum can have an inadmissible affect on the location and characterization of spectral peaks, depending on where the peaks are located. So to reduce the effects of this rolloff, a linear trend removal was done on the PS and the CA. Initially a line was fit in a least squares sense to the PS and CA. If the N samples long PS and CA data segment is represented as y[n] for the Y-axis values (amplitude) and x[n] for the X-axis values (frequency) then the least squares estimate line fit of the data segment can be defined as follows [16]:

$$\hat{y}[n] = ax[n] + b \quad \text{for } n = 1, 2, \ldots N \qquad (4.1)$$

where $$a = \frac{N\left(\sum_{n=1}^{N} x[n]y[n]\right) - \left(\sum_{n=1}^{N} x[n]\right)\left(\sum_{n=1}^{N} y[n]\right)}{N\left(\sum_{n=1}^{N} x[n]^2\right) - \left(\sum_{n=1}^{N} x[n]\right)^2} \quad (4.2)$$

and $$b = \frac{\left(\sum_{n=1}^{N} x[n]^2\right)\left(\sum_{n=1}^{N} y[n]\right) - \left(\sum_{n=1}^{N} x[n]y[n]\right)\left(\sum_{n=1}^{N} x[n]\right)}{N\left(\sum_{n=1}^{N} x[i]^2\right) - \left(\sum_{n=1}^{N} x[i]\right)^2} \quad (4.3)$$

Here a and b are the polynomial coefficients that satisfies the least squares line fit. The error or the residuals from this line fit can then be represented as follows:

$$Error = \left(y[n] - \hat{y}[n] \right) \quad (4.4)$$

After the line was fit to the data, 50% of the points that deviate in the positive direction from the line were censored off. The remaining data points can be represented as in equations 4.5 and 4.6 below.

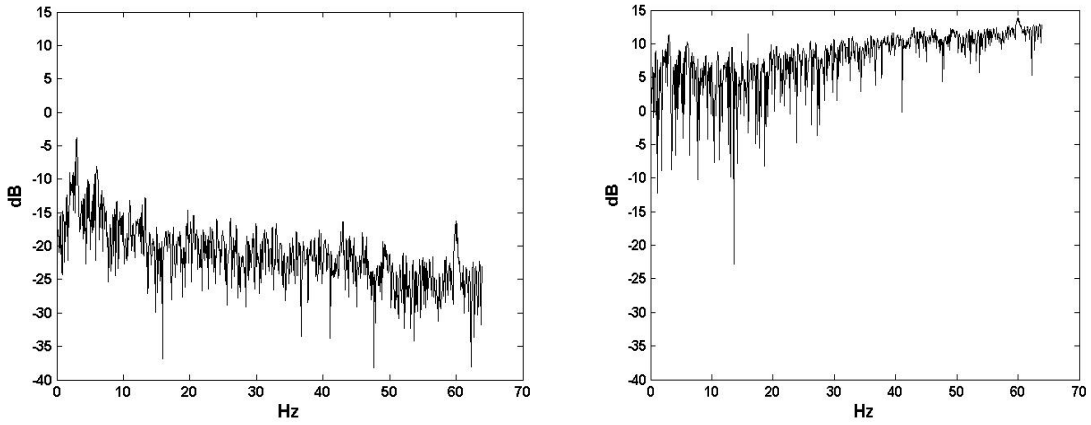
$$\bar{y}[n] = y[n \mid (< median(Error))] \quad (4.5)$$

$$\bar{x}[n] = x[n \mid (< median(Error))] \quad (4.6)$$

The same process was repeated again for these remaining data points $\bar{y}[n]$ and $\bar{x}[n]$ and another 50% of the data points that deviate in the positive direction removed (as in equations 4.1 - 4.6). After this second fitting of the line, the slope and y-intercept of the line that fit these remaining data points was used to remove the linear trend for the final result as shown in equation 4.7 below:

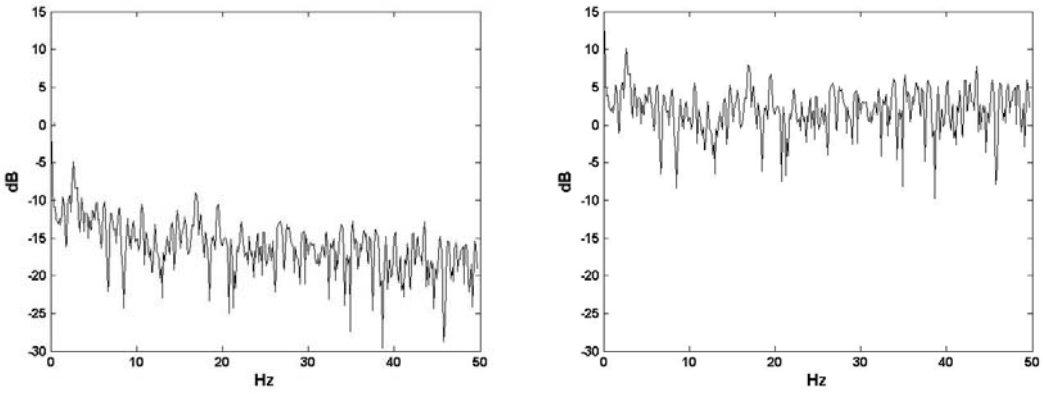
$$y[n]_{FINAL} = y[n] - (ax[n] + b) \quad (4.7)$$

Where $y[n]_{FINAL}$ is the PS and CA data with the linear trend removed. The slope, Y-intercept and the midpoint of this best fit line was also used to extract features for classification purposes. These features will be discussed in more detail later on.



(a) (b)

Figure 4.4 (a) The PS of a sleep segment of 8 seconds without the linear trend removed. (b) The PS of the same sleep segment with the linear trend removed.



(a) (b)

Figure 4.5 (a) CA without the linear trend removed. (b) CA with the linear trend removed.

Chapter 5

Feature extraction

5.1 Introduction

This chapter describes the 23 features used in the classifier and details how they were computed from the data. It will also discuss some of the processing of the features that was done and the reasons for doing it.

5.2 Extraction of features from the data

Table 5.1 below lists all the features that were extracted and tested in the classifiers. The features are categorized based on the signal functions from which they were extracted. The feature categories are the Collapsed Average (CA), Power Spectrum (PS) and Autocorrelation (AC) features. Each of the 23 features that were used was a vector with each element in the vector containing the value of that feature for a single 8 second data segment that was analyzed.

In the sleep state both the PS and the CA display a large peak in the 2-3Hz region along with peaks at the harmonics whereas in the wake state the PS and CA will display a flatter response in this frequency range. Based on this observation the position and the magnitude of the maximum peaks in this range were extracted from the PS and CA. The peaks were detected from the gradients. When the gradient transitioned from a positive to a negative between two points it signified a maximum. An interpolation was then done between these two points to find the position of the peak. If $\nabla[k]$ represents the gradient at the point y[k], x[k] in the PS or CA data segment, then the interpolation to find the position can be represented as follows:

$$Position[k] = \frac{(y[k] - \nabla[k] * x[k]) - (y[k+1] - \nabla[k+1] * x[k+1])}{\nabla(k+1) - \nabla(k)} \qquad (5.1)$$

The magnitude at these positions was found as follows:

$$Magnitude[k] = (\nabla[k] * \text{Position}[k]) + (y[k] - \nabla[k] * x[k]) \qquad (5.2)$$

After all positions and magnitudes of the peaks in the frequency range for the data segment were found, the peak with the largest magnitude was picked and its position noted. There was a set of features that looked at the magnitude and position of the maximum peaks in the frequency range of 0.5-8Hz for both the power spectrum and the CA. This frequency range was selected as the sleep signals fundamental frequency and harmonics lie in this range. Another set of features was also extracted by looking at the position and the magnitude of the maximum peak in the frequency range of 1.5-4.5Hz. This frequency range was selected as the sleep signals fundamental frequency lies in this range. This same method of peak detection was applied to the autocorrelation as well. But for the autocorrelation the full 8 second data segment was used.

Another feature was extracted by looking at the energy in the power spectrum. The energy was calculated for each N samples long data segment as follows:

$$Energy = \sum_{F=0}^{N} \left(|Y[F]|^2 \right) \qquad (5.3)$$

During sleep states the PS typically displays peaks at only the fundamental frequency and harmonics. In addition, the weak pressure on the piezoelectric film of the mouse breath motion (weak relative to the higher pressures of the mouse moving about on the pad) results in low energy overall for sleep states compared to the wake states.

A set of features was also extracted from the least squares line fit to the PS and the CA that was discussed in the previous chapter. The y-intercept, the slope and the midpoint of the line were calculated and used as features. The slope is found from equation 4.2 and the y-intercept is found from equation 4.3. The midpoint was calculated from the mean of the least squares line as follows:

$$Midpnt = \frac{1}{N} \sum_{n=1}^{N} (ax[n] + b) \qquad (5.4)$$

In the wake state the y-intercept and the slope displayed larger values due to rolling off of the PS and CA (see Figure 3.4 (a) and (c)), whereas in the sleep state they displayed smaller values due to the PS and CA having less rolloff (see Figure 3.3 (a) and (c)).

Another feature used was the periodic likelihood of the CA, PS and the Autocorrelation. This was calculated by looking at the total number of all the positive gradients in the CA, PS and Autocorrelation. If $\nabla[l]$ is a vector representing all the gradients and $\nabla[p]$ is all the positive gradients then this calculation can be represented as follows:

$$\nabla[p] = \nabla[l \mid (\nabla[l] > 0)] \quad (5.5)$$

$$Periodic\_likelihood = length(\nabla[p]) \quad (5.6)$$

The periodic likelihood was higher when there was high periodicity (as in the sleep case) and low values (as in the wake case). This feature was extracted only from the 0.5-8Hz frequency range of the CA and PS, but was extracted from the entire 8 second data segment for the Autocorrelation.

The frequencies less than 0.5Hz in the CA also provided some useful information. It was mentioned earlier that in the GS matrix the main diagonal represents the PS of the data segment. This main diagonal corresponds to the low frequencies in the CA. Hence one feature was extracted that gave the frequency content < 0.5 Hz in the CA.

Table 5.1 Description of parameters extracted

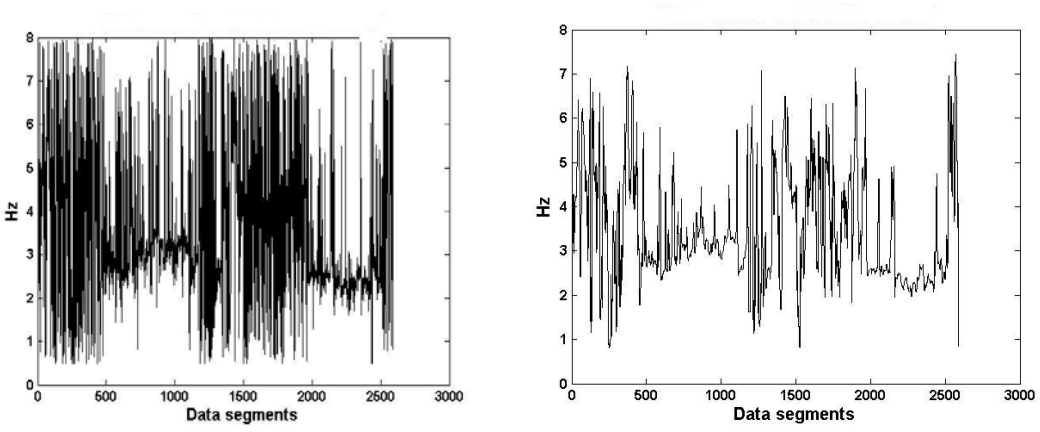
Number	Feature name	Description
1	Ca.slope	Slope of least squares line fit to CA
2	Ca.yintc	Y-intercept of least squares line fit to CA
3	Ca.mdband	Midband of least squares line fit to CA
4	Ca.perlike	Periodic likelihood of CA
5	Ca.lfp	Frequency content <0.5Hz of the CA
6	Ca.sp	Position of maximum peak in 1.5-4.5Hz range of the CA
7	Ca.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the CA
8	Ca.spt	Position of maximum peak in 0.5-8Hz range of the CA
9	Ca.smt	Magnitude of maximum peak in 0.5-8Hz range of the CA
10	Sp.enrg	Energy in PS
11	Sp.slope	Slope of least squares line fit to PS
12	Sp.yintc	Y-intercept of least squares line fit to PS
13	Sp.mdband	Midband of least squares line fit to PS
14	Sp.sp	Position of maximum peak in 1.5-4.5Hz range of the PS
15	Sp.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the PS
16	Sp.spt	Position of maximum peak in 0.5-8Hz range of the PS
17	Sp.smt	Magnitude of maximum peak in 0.5-8Hz range of the PS
18	Sp.perlike	Periodic likelihood of PS
19	Ac.sp	Position of maximum peak in 1.5-4.5Hz range of the AC
20	Ac.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the AC
21	Ac.spt	Position of maximum peak in 0.5-8Hz range of the AC
22	Ac.smt	Magnitude of maximum peak in 0.5-8Hz range of the AC
23	Ac.perlike	Periodic likelihood of AC

5.3 Processing of the features

Once the features were extracted from the whole data set, further processing was done on the features to improve there robustness to anomalous transient activity. The

estimated features are impacted by transients, which can cause classification errors. For example if an 8 second data window (1 data segment) has two or three small mouse movements during sleep, sharp transients spikes can occur in addition to the normal sleep signature. In this case it is likely that the features extracted from this data segment will lean more towards wake rather than sleep (as shown in section 3.4.7). If the adjoining data segments display sleep features then this one wake segment will act as a transient.

Thus a median filter was employed to smooth consecutive feature estimates. A median filter is a one-dimensional, nonlinear filtering technique that applies a sliding window to a data sequence. The median filter replaces the center value in the window with the median value of all the points within the window. The reason a median filter was preferred over a filter such as a mean filter is because of the robustness of the median filtering to the outliers in the data. A mean filter replaces the center value of the data segment with the mean of the whole segment. Thus an outlier can have an undue influence on a mean filter more than a median filter. For this application a median filter window length of 10 was used. This means that the median filter will be applied to 10 samples of the feature vector at a time (these 10 samples correspond to the value of the feature from 10 data segments each 8 seconds long and with 50% overlap). Figure 5.1 below shows the effect of the median filtering on one of the features.



(a) (b)

Figure 5.1 (a) Plot of the Sp.spt feature for 2500 data segments (each 8 seconds long) without the median filtering. (b) Plot of the same feature but with the median filtering.

Chapter 6

The linear classifier and the performance of features

6.1 Introduction

Once the 23 features were extracted and processed they were run through a linear classifier to test their performance in discriminating between sleep and wake. A linear classifier was chosen for comparing feature performance as it's easier to identify the amount of influence each feature has in the classification process. The features were run through the classifier individually and also in combinations and the best performing were selected. This chapter will first discuss how the data was collected for use in the classifiers. After which it will discusses the training and testing of the linear classifier and also the performance of the CA, PS and AC features when run through the classifier. It also examines the data segment size and the Kaiser Window parameters that worked best for the features. The use of the Fisher Discriminant to calculate the weighting of the features and the computation of the threshold for separating sleep from wake is also described. Finally, the performance of the linear classifier was compared to that of a Neural Network classifier in order to determine if non-linear classifiers may improve performance.

6.2 Collection of data

There were two sets of data collected using two different mice on two different days. A total of 9 hours of data was collected. The specifics of the data sets are given below in Table 6.1.

Table 6.1 – Details of data sets used for testing.

Data set name	Length of data (hours)	Number of mice	Time of day	Percentage of sleep and wake
Data set #1	5 hours	1	Day time	55% sleep, 46% wake
Data set #2	4 hours	1	Day time	56% sleep, 44% wake

Along with each of these data sets collected from the piezo sensor, each data set had its own research assistant doing a human scoring of the observed state of the mouse as either wake or sleep.

The human scoring was done via a LabVIEW script which was connected to a Function Generator via a GPIB interface. This script had different buttons for the sleep and wake states. Thus when one of the buttons in the LabVIEW script was pressed it would send a signal to the Function Generator which in turn gave out a DC signal. Each button had a different DC level assigned to it. So when the human observed a sleep or wake state in the mouse he would click on the respective button and get a certain DC level out of the function generator. This DC signal from the Function Generator was read into the A/D converter along with the signal from the piezo sensor. The state when the mouse was laying down on the piezo sensor and had its eyes closed was determined by the human as a sleep state. The wake states were when the mouse was walking, grooming, climbing the cage or standing still.

It should be noted that during the classification process this human scoring is used for the training of the classifiers (this will be discussed in more depth in the following sections). Thus there is a correlation between the performance of the classifiers and the human scoring. But this also turns out to be a limitation on the classification performances. For instance the human scoring may not be accurate for short sleep or wake bouts as a human likely would not record a sleep or wake state unless it persisted for at least 4 or 5 seconds or more.

6.3 The linear classifier

For this application the *classify* function in Matlab was used to design and test the linear classifier for sets of computed features. This is a supervised classification method that first trains the classifier with known sleep and wake features, and then tests it with a different feature set.

6.3.1 Design and testing of the linear classifier

Each of the two data sets were divided into 8 second data segments and 23 features extracted from each data segment using a moving window (as was discussed in section 3.2). This meant that each of the 23 features that were extracted was a vector, with each element in the vector containing the value of that feature for a single 8 second data segment. From these feature vectors 25% of the segments were randomly selected to train the classifier (compute the weights) and 25% of the segments were randomly selected to test the classifier performance. The segments were randomly selected for training and testing using a random number generator such that there were no data in common between the training and testing sets. The misclassification rate for the trained classifier was determined by the performance on the independent test features. A bootstrapping procedure was used where the training and testing features were randomly selected 40 times from the entire feature set to yield 40 performance estimates from which the mean and standard deviations were computed.

6.4 Individual performance of features

Now the question arises as to which features perform the best. To figure this out a test was done on each feature individually. One feature at a time was run through the classifier and the error rates recorded (one feature was used for the training and the same feature was used for testing of the classifier). This testing was done on the two data sets listed above in Table 6.1.

Various data segment sizes and various values for the Kaiser tapering windows β parameter were also tested for each feature to see which combination would give the lowest errors. This β parameter controlled the main-lobe width and the side-lobe attenuation of the Kaiser window in the frequency domain, and thus controlled the frequency resolution and the spectral leakage that the window allowed. The length of the Kaiser window was the same as the data segment size [1].

The following sections describe the tests on these two data sets and show the best performing features in a table format. The summary on individual feature testing that follows (section 6.4.5) will discuss the performance of the features. The errors are also

plotted in a bar chart format for clarity. The features were tested for data segment sizes of 2, 4, 6 and 8 seconds and β values of 0, 2, 4 and 6.

6.4.1 Individual feature test #1

Table 6.2 below gives the lowest error rates for Data set #1 which had 5 hours of data. For this test the feature sequences were not median filtered. This was so that we can get an idea of how robust these features are without the median filtering. Column three shows the best performing data segment sizes and column fours shows the best performing Kaiser Window β parameters for that feature. Column five shows the minimum error achievable with only that one feature used for classification. The features are arranged from the best performing to the worst performing. Figure 6.1 also shows the errors in the form of a bar chart. The 95% confidence intervals are also given along with the errors.

As can be observed from Table 6.2 and Figure 6.1 the CA parameters did not seem to perform too well without the median filtering with 8 of the 9 CA features having errors above 25 %. Whereas 4 of the 9 SP features and 3 of the 5 AC features gave error rates less than 25%. The features that gave less than 25% errors are highlighted in bold.

It should be noted that the majority of the features that performed the best were the features that measured the magnitude of the maximum peak in the PS, CA and AC. So it is clear that the best information for the classifier comes from the magnitude of the peaks in the PS, CA and AC.

Table 6.2 Individual performance of features for data set #1 (without median filtering).

Feature Number	Feature Name	Data Segment Size (sec)	Kaiser Window β Parameter	Minimum Error
15	**Sp.sm**	**4,6,8**	**all**	**14 ± .304 %**
10	**Sp.enrg**	**8**	**all**	**16 ± .259 %**
17	**Sp.smt**	**6,8**	**all**	**17 ± .277 %**
20	**Ac.sm**	**2**	**all**	**19 ± .391 %**
7	**Ca.sm**	**8**	**all**	**21 ± .359 %**
23	**Ac.perlike**	**6,8**	**all**	**22 ± .336 %**
22	**Ac.smt**	**2**	**all**	**24 ± .358 %**
12	**Sp.yintc**	**4,6,8**	**0,2**	**24 ± .410 %**
9	Ca.smt	8	all	25 ± .336 %
13	Sp.mdband	6,8	0,2,4	26 ± .333 %
8	Ca.spt	6,8	all	27 ± .322 %
16	Sp.spt	6,8	all	28 ± .357 %
2	Ca.yintc	8	all	28 ± .438 %
3	Ca.mdband	8	all	29 ± .460 %
5	Ca.lfp	8	all	29 ± .366 %
11	Sp.slope	8	all	30 ± .394 %
18	Sp.perlike	2	2,4,6	30 ± .383 %
1	Ca.slope	8	all	32 ± .369 %
4	Ca.perlike	4	0	32 ± .296 %
6	Ca.sp	2	0,2,4	35 ± .471 %
14	Sp.sp	2	all	36 ± .504 %
19	Ac.sp	6,8	all	39 ± .465 %
21	Ac.spt	6, 8	all	46 ± .492 %

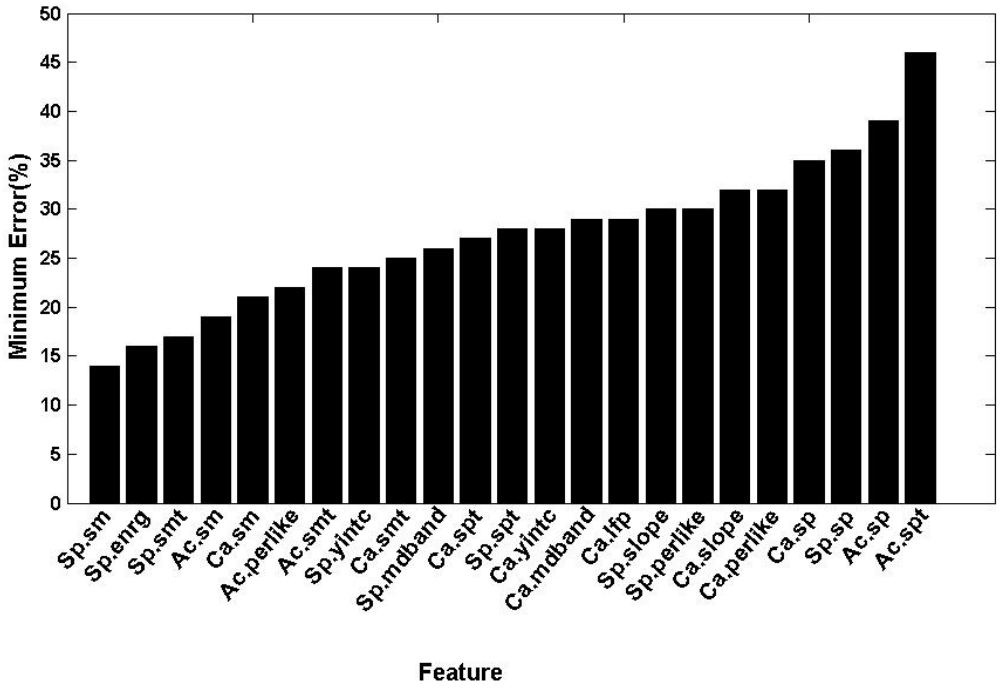


Figure 6.1 Bar plot of classification errors from table 6.2

6.4.2 Individual feature test #2

Table 6.3 below is the performance of the features individually for the same data set #1, but with the median filtering done on the features. The error rates show a markedly decrease with the median filtering. It should be noted that the GS features also showed a big improvement and had error rates almost as good as the SP and AC features. Most of the GS features showed error rates less than 20% .The features that produced less than 20% error are shown in bold. The features are arranged from best performing to worst performing. Figure 6.2 also shows the errors in a bar chart format.

Table 6.3 Individual performance of features for Data set #1 (with median filtering).

Feature Number	Feature Name	Data Segment Size (sec)	Kaiser Window β Parameter	Minimum Error
15	**Sp.sm**	**all**	**all**	**8 ± .230 %**
10	**Sp.enrg**	**all**	**all**	**10 ± .233 %**
17	**Sp.smt**	**all**	**all**	**10 ± .214 %**
20	**Ac.sm**	**4,6**	**all**	**10 ± .262 %**
7	**Ca.sm**	**8**	**4,6**	**11 ± .231 %**
18	**Sp.perlike**	**2**	**2,4,6**	**14 ± .214 %**
9	**Ca.smt**	**6,8**	**4,6**	**15 ± .236 %**
8	**Ca.spt**	**2**	**4,6**	**15 ± .288 %**
22	**Ac.smt**	**4,6**	**all**	**15 ± .212 %**
12	**Sp.yintc**	**4,6,8**	**all**	**16 ± .360 %**
23	**Ac.perlike**	**6,8**	**all**	**16 ± .320 %**
4	**Ca.perlike**	**2**	**0,2**	**16 ± .325 %**
2	**Ca.yintc**	**6,8**	**all**	**18 ± .281 %**
3	**Ca.mdband**	**6,8**	**all**	**18 ± .276 %**
5	**Ca.lfp**	**8**	**4,6**	**19 ± .325 %**
16	Sp.spt	W=2,4,6	all	21 ± .410 %
1	Ca.slope	8	all	21 ± .376 %
13	Sp.mdband	any	all	21 ± .307 %
6	Ca.sp	2	0,2,4	25 ± .402 %
11	Sp.slope	6,8	all	25 ± .366 %
14	Sp.sp	2	all	28 ± .405 %
19	Ac.sp	all	all	35 ± .357 %
21	Ac.spt	6,8	all	39 ± .419 %

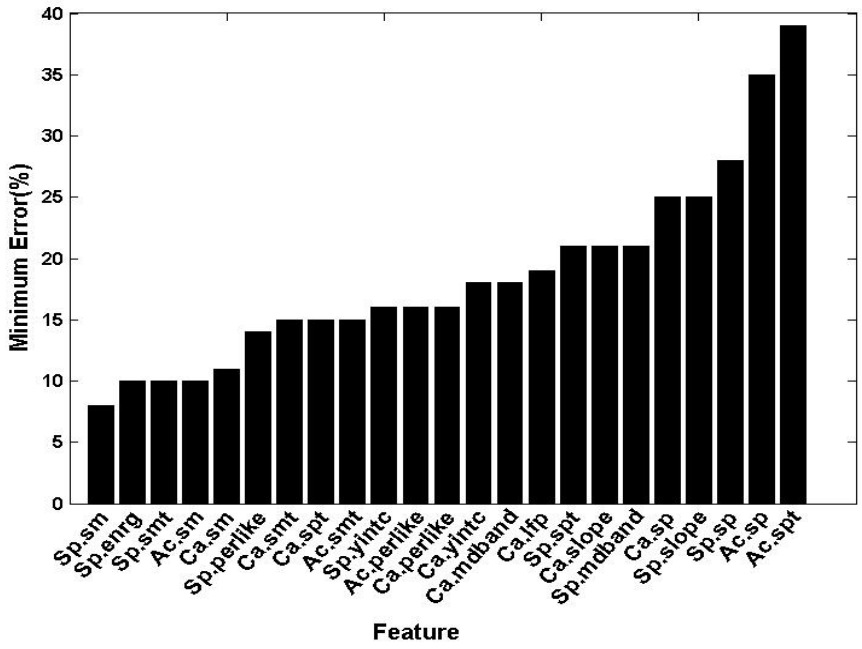


Figure 6.2 Bar plot of classification errors from table 6.2

GS features such as 2, 3, 4, 5 and 7, 8, 9 show an error percentage less than 20%. The SP features 10, 12, 15 and 17 as well as the AC features 20, 22 and 23 give less than 20% error. The minimum error achievable for this data set with just one feature was with the Sp.sm feature which gave 8% error.

It should also be noted that most of the features that performed well gave their best performance for larger data segment sizes such as 6 or 8 seconds and smaller Kaiser window β parameters such as 4 or 6. The reason for this will be discussed later.

6.4.3 Individual feature test #3

This test was for a Data set #2 described in Table 6.1. Here again the features were run individually and the errors recorded. No median filtering was performed on these sequences of features. Table 6.4 shows the best performing window sizes, Kaiser window β parameters and the minimum errors. The features are arranged from best performing to worst performing and the features that gave less than 25% error are highlighted in bold. Figure 6.3 shows the errors in a bar chart format.

Table 6.4 Individual performance of features for Data set #2 (without median filtering).

Feature Number	Feature Name	Data Segment Size (sec)	Kaiser Window β Parameter	Minimum Error
15	**Sp.sm**	**all**	**all**	**19 ± .303 %**
10	**Sp.enrg**	**8**	**all**	**21 ± .396 %**
17	**Sp.smt**	**6,8**	**all**	**21 ± .335 %**
20	**Ac.sm**	**all**	**all**	**21 ± .409 %**
22	**Ac.smt**	**all**	**all**	**22 ± .395 %**
16	**Sp.spt**	**6,8**	**4,6**	**23 ± .463 %**
12	**Sp.yintc**	**6,8**	**4,6**	**23 ± .393 %**
7	Ca.sm	6,8	all	27 ± .378 %
13	Sp.mdband	6,8	all	27 ± .418 %
9	Ca.smt	6,8	all	29 ± .391 %
2	Ca.yintc	6,8	4,6	30 ± .419 %
18	Sp.perlike	2	2,4	30 ± .505 %
11	Sp.slope	6,8	4,6	30 ± .404 %
23	Ac.perlike	all	all	31 ± .427 %
19	Ac.sp	all	all	31 ± .470 %
3	Ca.mdband	6,8	all	32 ± .461%
4	Ca.perlike	2,4	0,2,4	34 ± .525 %
5	Ca.lfp	6,8	4,6	34 ± .430 %
6	Ca.sp	6,8	all	35 ± .623 %
14	Sp.sp	6,8	all	35 ± .477 %
21	Ac.spt	all	all	35 ± .418 %
1	Ca.slope	6,8	4,6	38 ± .463 %
8	Ca.spt	6,8	all	38 ± .502 %

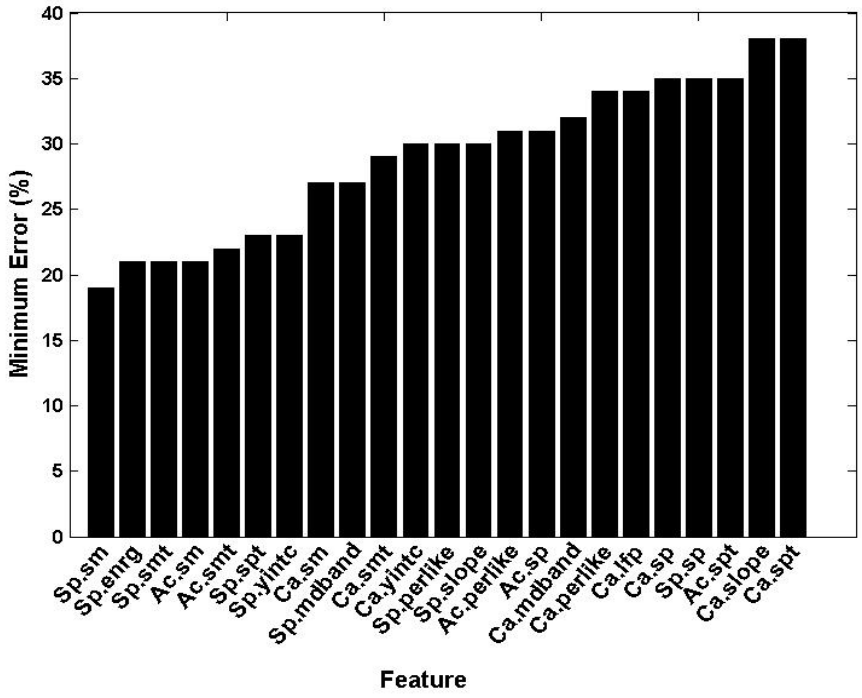


Figure 6.3 Bar plot of classification errors from table 6.3

The results are very similar to the results from individual feature test #1 in that the CA features did not seem to perform as well as the SP and AC without the median filtering. All the CA features showed error rates above 25 %. Just like for individual feature test #1, the features that gave the best information to the classifier were the features that measured the magnitude of the maximum peaks in the PS, CA and AC.

6.4.4 Individual feature test #4

This test was for the same Data set #2 but with the median filtering done on the features. The performance of the features and the error rates are shown in table 6.5 and figure 6.4 below. The results for this data set was very similar to that from individual features test #2 in that almost the same median filtered features from Data set #1 seemed to perform well for Data set #2 after the median filtering. The CA features also showed considerable improvement from non median filtered CA features. The features that gave less than 20% error are highlighted in bold and they are arranged from best performing to worst performing.

Here again the best performance for the features was for a larger data segment size and smaller Kaiser window β parameters.

Table 6.5 Individual performance of features for data set #2 (with median filtering).

Feature Number	Feature Name	Data Segment Size (sec)	Kaiser Window β Parameter	Minimum Error
15	**Sp.sm**	**all**	**all**	**13 ± .269 %**
20	**Ac.sm**	**all**	**all**	**13 ± .335 %**
16	**Sp.spt**	**6,8**	**all**	**14 ± .325 %**
7	**Ca.sm**	**6**	**4,6**	**16 ± .282 %**
10	**Sp.enrg**	**8**	**all**	**16 ± .395 %**
2	**Ca.yintc**	**6,8**	**4,6**	**16 ± .426 %**
12	**Sp.yintc**	**6,8**	**0,2,4**	**16 ± .332 %**
18	**Sp.perlike**	**4,6**	**all**	**16 ± .337 %**
22	**Ac.smt**	**6,8**	**all**	**16 ± .330 %**
17	**Sp.smt**	**6**	**all**	**17 ± .284 %**
9	**Ca.smt**	**6**	**4,6**	**17 ± .366 %**
13	**Sp.mdband**	**any**	**all**	**20 ± .403 %**
3	**Ca.mdband**	**6**	**4,6**	**20 ± .430 %**
4	Ca.perlike	6	2	21 ± .430 %
11	Sp.slope	8	all	23 ± .465 %
5	Ca.lfp	8	4,6	23 ± .400 %
14	Sp.sp	6,8	2,4	25 ± .475 %
6	Ca.sp	6	0,2,4	26 ± .299 %
19	Ac.sp	6,8	all	27 ± .469%
21	Ac.spt	2,4	all	27 ± .407 %
23	Ac.perlike	6,8	all	28 ± .343 %
1	Ca.slope	6,8	4,6	30 ± .452 %
8	Ca.spt	8	4,6	30 ± .512 %

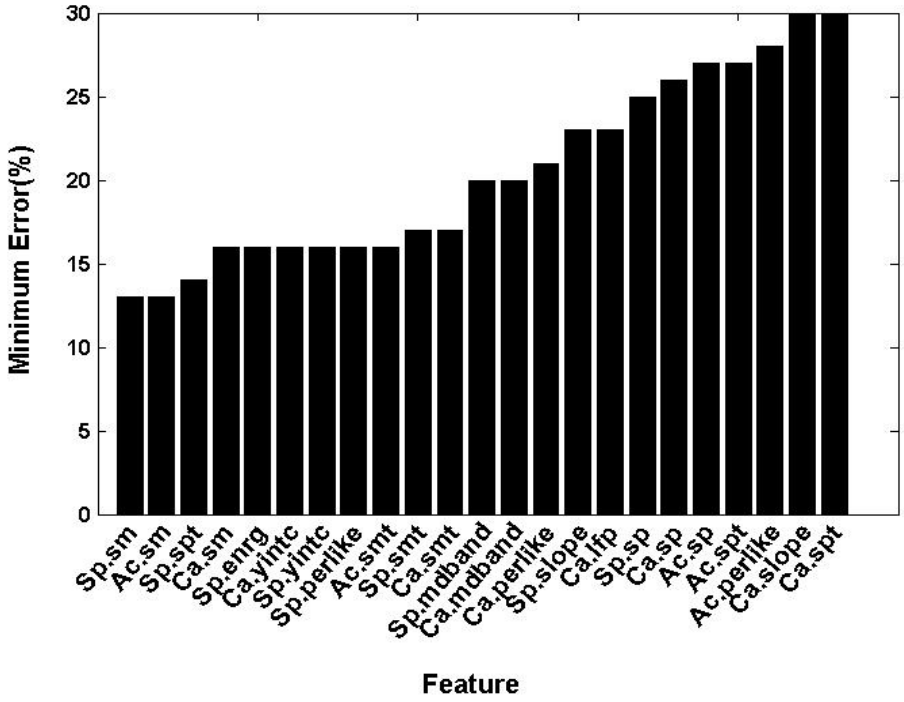


Figure 6.4 Bar plot of errors from Table 6.4

6.4.5 Summary of individual feature testing

The results from the testing of the individual features on the two data sets showed the features performing pretty much the same for both data sets. The CA parameters did not perform well without the median filtering but performed almost on the same level as the PS and AC features with the median filtering. What can be deduced from this is that the CA features were not as robust as first thought compared to the PS and AC features. The CA features seemed to have a lot of transients and outliers which were eventually smoothed out by the median filtering. These transients in the features probably occur due to the analysis window falling on a sleep segment similar to what was discussed in section 3.4.7 where there is a sharp spike like transient or transients in the data in which case the features will reflect a wake state for the data segment even though the data segment was mostly sleep. If a data segment such as this is preceded and followed by purely sleep data segments then the transients in the features are created. The CA seemed to be influenced more by this than the SP. The reason for this is likely that the CA is

more sensitive to detecting transients or spike-like signals in the data, and in the case of weak harmonics, the transients dominated the CA.

The fact that there was a pre-processing and processing of the data done can be another factor why the PS features performed so well. The detrending of the data would have taken out any non-stationarities due to level shift or ramp-like transitions in the data that would have obscured the low-frequency features, thus making it more stable.

Also of interest from these tests was the data segment size and Kaiser window β parameter. The best performances occurred for larger segment sizes such as 6 and 8 seconds and smaller β parameter values such as 4 and 6. A larger Kaiser window in the time domain (the Kaiser window size is the same as the data segment size) and a smaller β value implies a broader tapering window in the time domain corresponding to narrower main-lobe window that is convolved with the spectrum in the frequency domain. This narrower window will results in better resolution in the frequency domain and resolves the frequency components a lot better, thus giving better results for the PS and CA features. As an example Figure 6.5 and Figure 6.6 below shows an 8 second long Kaiser window with β values of 4 and 8 in both the time domain and frequency domain.

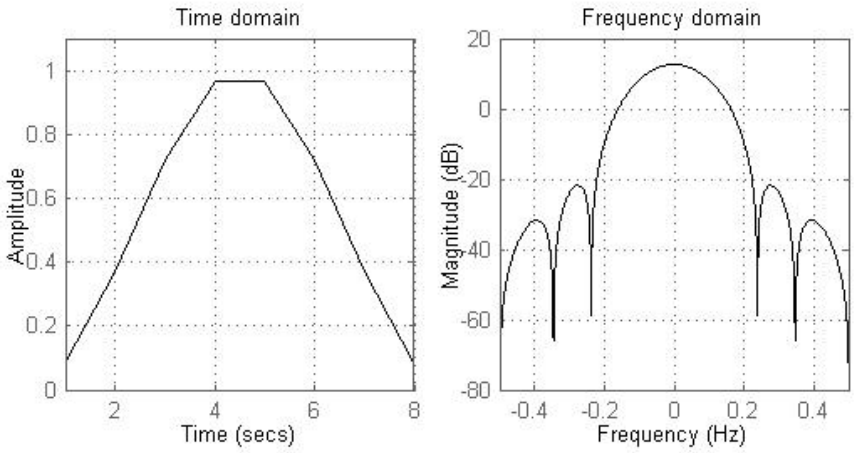


Figure 6.5 Kaiser window which is 8 seconds long with β value of 4.

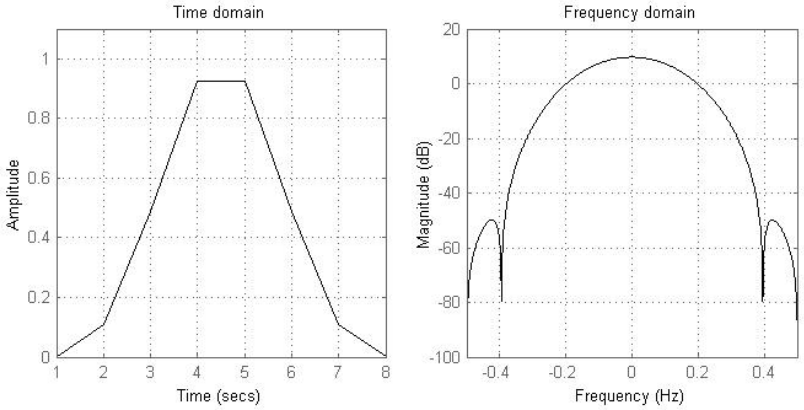


Figure 6.6 Kaiser window which is 8 seconds long with β value of 8

As these figures show a Kaiser window with β = 4 has a narrower main-lobe, but higher side-lobes than the window with β = 8 which has a wider main-lobe, but lower side-lobes. The higher side-lobes for β =4 implies slightly more spectral leakage, but in this application it did not seem to matter too much. It appears that the better resolution from a narrower main lobe was more critical. The fact that β =4 gave a narrower main-lobe in the frequency domain seemed to have a larger impact. So for this application a data segment size and Kaiser window size of 8 seconds and β =4 were chosen.

6.5 Performance of multiple features

Instead of running features individually, running them in combinations can reduce the error rates even more. The reason for this is that additional features can provide additional information to the classifier than just one feature. But then again adding extra features can also degrade classification performance and actually introduce more errors, especially if the feature has a high variance and it is estimated from a small a data segment or used to train the classifier with a small data set. For instance suppose three features are being used for classification and later two new features are added and the errors recorded. Although no features have been deleted, and new information has been added, this may actually yield worse results on the new, more complete set of features than on the original smaller set. The reason is likely that the new features have a high

noise content and that slows down the convergence of the classifier weights in the training sequence. Here the classifier performs particularly well with relatively low variance and highly predictive features, but degrades when noisy features are included. In addition if the noisy features primarily contain redundant information, a lower performance can be expected since the redundant features will only bring noise into the classifier and not add any new information. So it is important that the features are selected so that they give complementary information to the classifier, but without adding noise or adding redundancy.

The following sections describe some tests that were done to test combinations of multiple features. The tests were done on the same Data sets #1 and #2 discussed earlier. But instead of running the features individually, combinations of 5 features at a time were run through the classifier. The classifier was trained with 5 training features and then tested with the same 5 features and the errors recorded.

But not all the features were considered for these combinations tests. Only 10 out of the 23 were chosen. The features selected were only the ones that performed well for the individual testing of the processed (median filtered) features. Features that gave high error rates in the individual median filtered feature testing were thrown out. This limits the possibility of adding noisy features to the classifier.

The features also had to give additional new information. For example both the Sm.sm and Sm.smt features were not considered together even though one gave information from 1.5-4.5Hz and the other 0.5-8Hz. But they both had very similar error rates and performed almost the same which meant that the maximum peak in the 1.5-4.5Hz was also the maximum peak in the 0.5-8Hz range. So the Sm.smt feature did not give any additional new information over the Sm.sm feature. Hence only the best individually performing feature from these two were chosen. This same selection procedure was used for the Ca.smt/Ca.sm, Ca.spt/Ca.sp, Sp.spt/Sp.sp and the Ac.sm/Ac.smt features as well. Table 6.6 below gives the 10 best features that were chosen.

Table 6.6 Features chosen for combination testing.

No.	Feature name	Description
2	Ca.yintc	Y-intercept of least squares line fit to CA
4	Ca.perlike	Periodic likelihood of CA
7	Ca.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the CA
8	Ca.spt	Position of maximum peak in 0.5-8Hz range of the CA
10	Sp.enrg	Energy in PS
12	Sp.yintc	Y-intercept of least squares line fit to PS
15	Sp.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the PS
16	Sp.spt	Position of maximum peak in 0.5-8Hz range of the PS
18	Sp.perlike	Periodic likelihood of PS
20	Ac.sm	Magnitude of maximum peak in 1.5-4.5Hz range of the AC

6.5.1 Combinations of features test #1

This test was done on the Data set #1 mentioned earlier. Median filtering was done on the features before classification. The 10 features listed above were taken in different combinations of 5 (this would mean 252 different combinations) and run through the linear classifier along with the human scored training features. The error from the classifier for each combination was recorded.

From individual feature test #2 (section 6.4.2 and table 6.3) it was clear that the Sp.sm feature alone gave an error of around 8% for this data set. So when running the features in combinations a better error rate should be achieved. And indeed this was the case as a minimum error rate of 4% was achievable with the combinations.

But now the question rises as to which of these 10 features when taken in combinations would give the lowest error rates. To test this, first all the combinations that gave less than 6% error were chosen after which a scoring was done to see how often the individual features occurred in these combinations. This test gave a good indication of what features were working well in combinations. Figure 6.7 below gives the result of

this test. It shows how often each of the 10 features occurred in the combinations that gave less than 6% error.

From the results the features Ca.yintc, Sp.enrg and Sp.sm seem to stand out and are more recurrent in the combinations. From the figure it is clear that features Sp.enrg and Sp.sm occur in 100% of the combinations and feature Ca.yintc occurs in about 55% of the combinations. These features are the Ca.yintc which is the Y-intercept of the CA, Sp.enrg which is the energy of the PS and the Sp.sm which is the spectral magnitude in the 1.5-4.5 Hz range. The rest of the features seem to perform about the same.

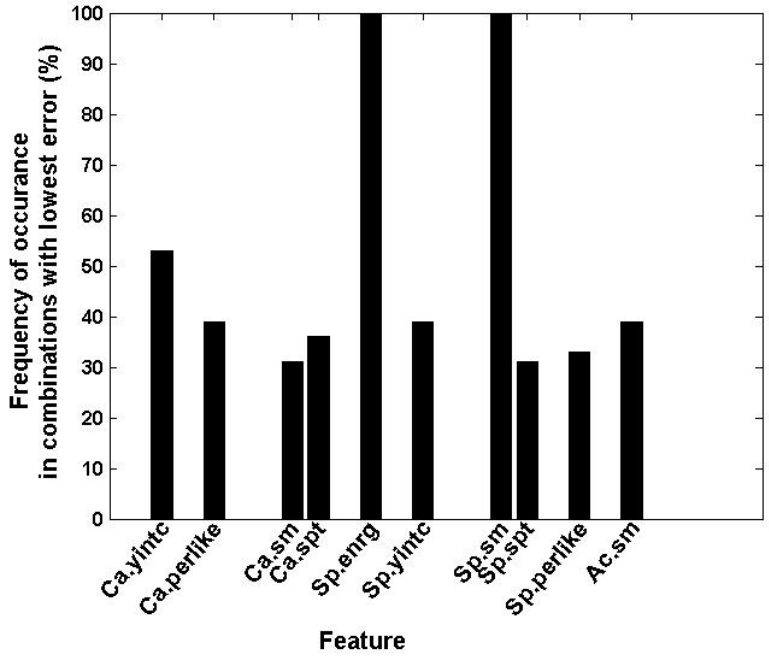


Figure 6.7 - Features that perform best in combinations for Data set #1

6.5.2 Combinations of features test #2

This test was the same as the previous one but for Data set #2. Median filtering was done on the features and the same 10 features were run through the classifier in combinations.

From individual feature test #4 (section 6.4.4 and table 6.5) it should be observed that the Sp.sm and the Ac.sm features alone gave about 13% error. With the

combinations the minimum error achievable for this data set was 9 % which is still an improvement from using just one feature. The same test was then done to see how often the individual features appeared in the combinations (all combinations that gave less than 11% error were chosen and a scoring done to see how often the individual features occurred in the combinations). Figure 6.8 below gives the results.

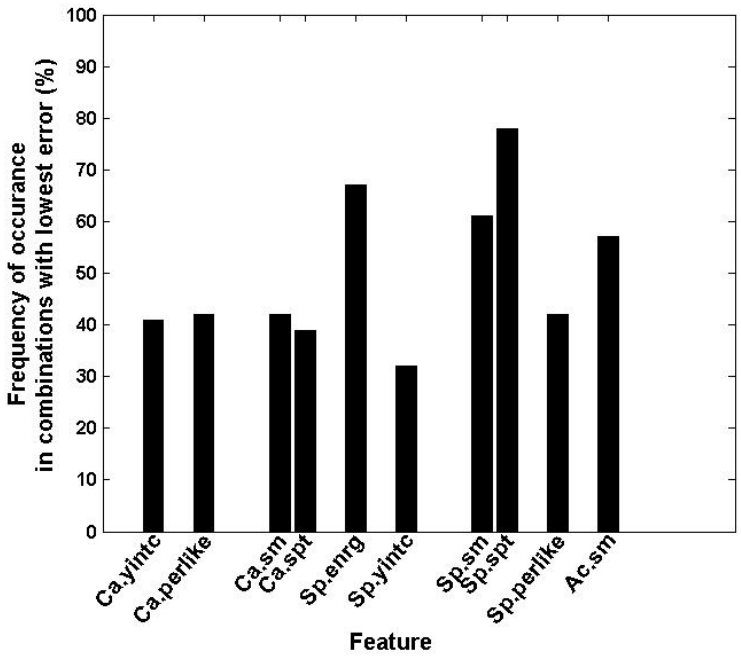


Figure 6.8 - Features that perform best in combinations for Data set #2 using the linear classifier

The features that performed best for this data set were slightly different. But just as in the previous test features Sp.enrg and Sp.sm performed well with each of them occurring in more than 60% of the combinations. In addition features Sp.spt and Ac.sm also performed well with Sp.spt occurring in 80% of the combinations and Ac.sm occurring in 55% of the combinations. The rest of the features did not stand out and seemed to perform about the same.

6.5.3 Summary of feature combination tests

From these two tests it is clear that features **10 and 15 (Sp.enrg and Sp.sm)** performed the best and seems to give the best information to the classifier. In addition to this there were three other features **2, 16 and 20 (Ca.yintc, Sp.spt and Ac.sm)** that seemed to perform well. All these 5 features measured different aspects of the stochastic data and thus added extra information without adding redundancy and noise to the classifier and thus worked the best for classification between sleep and wake. But if these five features fail to properly differentiate between sleep and wake then any of the other features can be added to the classification process. But these five features seem to suffice and the addition of an extra feature unnecessary.

6.6 Calculation of feature weights based on Fisher Discriminant

The *classify* function that was used in the previous section to select the best features does not provide the weights that must be applied to a test data set so that the data can be weighted as either sleep or wake. Instead the Fisher Discriminant was used to calculate the weights (the *classify* function uses the same method for its linear classification even though it does not actually give these weights). The Fisher Discriminant weights were calculated by using two sets of sleep and wake features already classified as sleep and wake using the human scoring. These features are used to decide how the features ought to be weighted in order to separate the sleep and wake classes. If the two training feature sets are D1 (sleep) and D2 (wake) then the definition of the weights (W) from the Fisher Discriminant is as follows:

$$W = \Sigma^{-1}(\mu_1 - \mu_2) \qquad (6.1)$$

where Σ is the sum of the covariance matrices of the D1 and D2 feature sets (covariance(D1) + covariance(D2)), μ_1 is the mean of the D1 feature set and μ_2 is the mean of the D2 feature set.

The Fisher Discriminant weights should give a larger weighting to the features that give a substantial contribution in the discrimination process. To test which features are weighted larger and which features are weighted lower, the Fisher Discriminant weights were calculated for the ten best features listed in table 6.6. They were calculated

for both data sets, Data set #1 and Data set #2. The weights for Data set #1 are shown below in Figure 6.9.

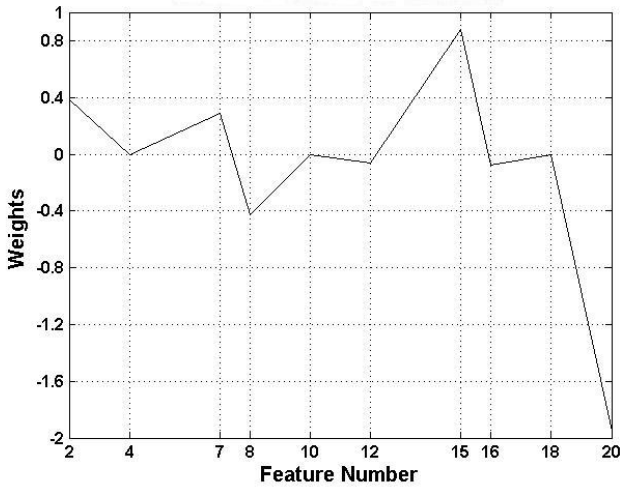


Figure 6.9 Weight for features in Table 6.5 for data set #1

For this data set it is clear that the largest weights were for features 2, 7, 8, 15 and 20 (Ca.yintc, Ca.sm, Ca.spt, Sp.sm and Ac.sm). It should also be noted that all of these five features performed well when tested individually (see Table 6.3), with each feature giving less than 15 % error. Features such as 4, 12, 16, and 18 gave errors greater than 15% in the individual testing and thus were weighted smaller (see Table 6.3).

Figure 6.10 below shows the weights for Data set #2. For this data set features 15, 16 and 20 (Sp.sm, Sp.spt and Ac.sm) seem to be weighted more than the others. This is again consistent with the individual feature tests that were done for this data set (See Table 6.5). In fact these are the only three features that gave less than 15 % error for the individual feature test. So just as for Data set #1 the Fisher Discriminant weights are consistent in that they weigh the best performing features more than the rest. This also confirms that the tests done to select the best features using the *classify* function were consistent with other results using these features.

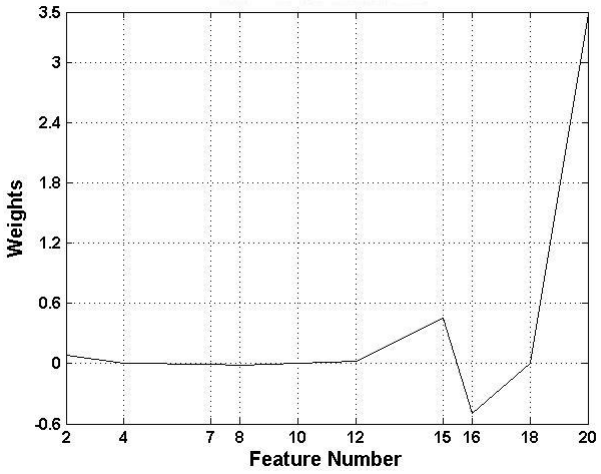


Figure 6.10 Weight for features in Table 6.5 for data set #2

The weights for the five best performing features (Ca.yintc, Sp.enrg, Sp.sm, Sp.spt and Ac.sm) were also calculated and they are given in Table 6.7 below. These Fisher Discriminant weights were consistent for both Data set #1 and Data set #2 and thus can be used for weighting the features of any data set.

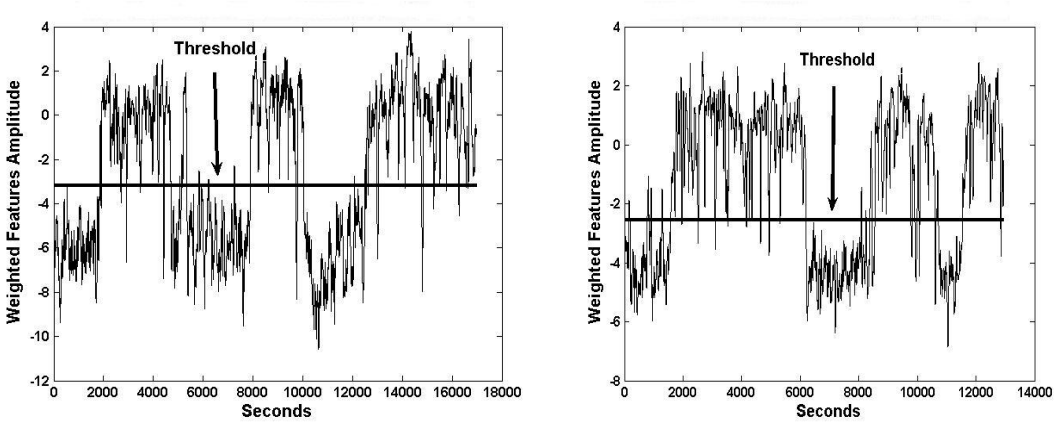
Table 6.7 – Five best features and their associated weights.

Feature Number	**Feature**	**Fisher Discriminant Weights**
2	Ca.yintc	0.3177
10	Sp.enrg	-0.0026
15	Sp.sm	1.0341
16	Sp.spt	-0.0638
20	Ac.sm	-1.4242

6.7 Computation of threshold for differentiating sleep and wake

Once the weights were calculated the same training sleep and wake features are then used again to obtain a threshold that will efficiently separate sleep from wake. First the sleep and wake features are weighted by the five weights shown above in table 6.7. Then 512 thresholds are computed between the maximum and minimum of these weighted

sleep and wake features. After which the probability of detection P_d and the probability of false alarm P_{fa} are calculated from the weighted sleep and wake features for each of these 512 thresholds. The optimum threshold was then chosen such that there was an equal probability of (P_d) and (P_{fa}). The optimum threshold was calculated as around -3.1 for both data sets. Figure 6.11 below shows the plots for Data set #1 and Data set #2 with the weighted features and the optimum threshold of -3.1. Sleep states will be above the threshold and wake states below the threshold.



(a) (b)

Figure 6.11 – (a) Weighted features and threshold for data set #1 (b) Weighted features and threshold for data set #2.

6.8 Misclassification error rates versus amount of sleep and wake in data

The error that we see from the classifier is due to a misclassification, the classifier is presented a case and it classifies the case incorrectly. In this application the misclassification can be caused by either sleep being misclassified as wake or by wake being misclassified as sleep. To see which case gave the higher errors a test was done. Dataset #1 and Dataset #2 were each divided into three 1 hour segments and then each segment at a time was run through the linear classifier and the error rates recorded. The percentage of sleep and wake in each segment was also recorded. The features used for the classification were the 5 best performing features described in section 6.5.3. They were also median filtered before they were run through the classifier. Table 6.8 and Table

6.9 below show the amount sleep and wake for each segment and also the error rates for each segment.

Table 6.8 - Percentage of sleep and wake for each hour and respective error rates for Data set #1.

Hour	Amount of sleep (%)	Amount of wake (%)	Error (%)
Hour 1	47 %	53 %	2.4 %
Hour 2	35 %	65 %	10.6 %
Hour 3	57 %	43 %	4.7 %

Table 6.9- Percentage of sleep and wake for each hour and respective error rates for Data set #2.

Hour	Amount of sleep (%)	Amount of wake (%)	Error (%)
Hour 1	50 %	50 %	9.8 %
Hour 2	72 %	28 %	1.8 %
Hour 3	56 %	44 %	10.5 %

From the results above what stands out is hour 2 of data from Data set #1 and Data set #2. Hour 2 in Data set #1 has around 2/3 of wake and about 1/3 of sleep resulting in a higher error rate than the other two hours. Hour 2 in Data set #2 has around 1/3 wake and around 2/3 of sleep and resulting in a very low error rate. So it seems like the more wake states there is in the data the more misclassification and errors from the classifier. Hence it can be deduced from this that for this application the misclassification occurs mainly due to wake being misclassified as sleep and not vice versa.

6.9 Comparison of Linear Classifier with Neural Network Classifier

A comparison was done between the performance of the Linear Classifier and a Neural Network classifier. The Neural Network classifier was implemented with a two layer feedforward network with 20 neurons in the first layer and 1 neuron in the second layer. All neurons use a hyperbolic tangent sigmoid transfer function and a bias weight. The desired output is set to +1 if the input features correspond to sleep and -1 if it

corresponds to wake. The neural network was trained and tested in the same way as the linear classifier (described in section 6.2) with 25% of the features used for training and 25% used for testing with no overlap of the features.

Just as for the linear classifier the individual feature tests were done for the Neural Network classifier as well. It turned out that the best performing individual features were the same as for the linear classifier. Hence the 10 best features were the same as the ones chosen with the linear classifier (see Table 6.6).

Tests for the performance of multiple features were also done for the Neural Network. The tests were the same as the combinations of features tests done for the linear classifier described in sections 6.5.1 and 6.5.2 (the 10 best features were taken in different combinations of 5).

Figure 6.12 below gives the features that performed the best in combinations for the Neural Network classifier for Data set #1. Just as for the linear classifier the Sp.enrg and the Sp.sm features performed the best in combinations for this data set. In addition the Ca.sm feature also performed well.

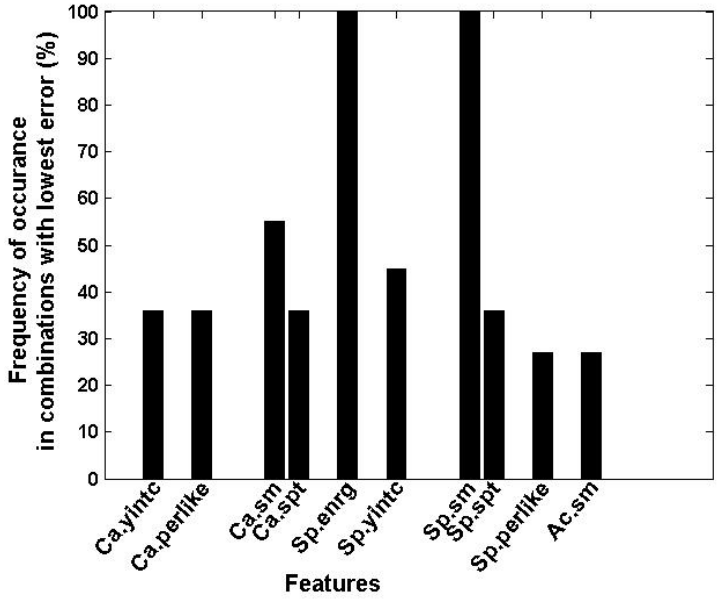


Figure 6.12 - Features that perform best in combinations for Data set #1 using the Neural Network classifier.

Figure 6.13 below gives the features that performed the best in combinations for the Neural Network classifier for Data set #2. Just as for the linear classifier the Sp.sm, Sp.spt and Ac.sm features performed the best in combinations for this data set. In addition the Ca.sm and Ca.spt features also performed well.

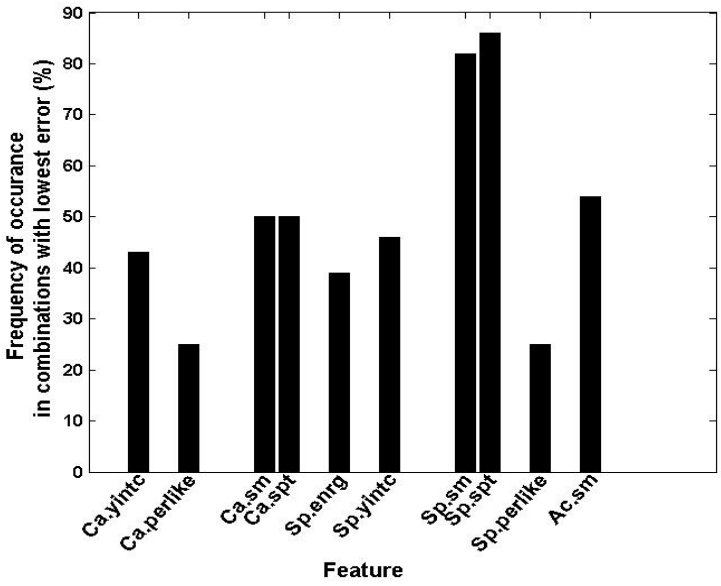


Figure 6.13 - Features that perform best in combinations for Data set #2 using the Neural Network classifier.

From these two tests for the Neural Network classifier, it is clear that the best performing feature for both data sets was the **Sp.sm** feature. In addition to this the **Ca.sm**, **Sp.enrg, Sp.spt** and the **Ac.sm** features also performed well. These 5 features can be used to do any future classification between sleep and wake states when a Neural Network classifier is used.

Table 6.10 below shows the minimum error rates achievable with a combination of the 5 best features for both the linear and Neural Network classifier for Data set #1 and Data set #2. The 5 best features used for the linear classifier was the **Ca.yintc, Sp.enrg, Sp.sm, Sp.spt and Ac.sm** features and the 5 best features used for the Neural Network were **Ca.sm, Sp.enrg, Sp.sm, Sp.spt and Ac.sm**.

Table 6.10 – Error rates for Data set #1 and Data set #2 for linear and Neural Network classifiers

Data set	Minimum error rate from linear classifier	Minimum error rate from Neural Network classifier
Data set #1	4%	6%
Data set #2	9%	12%

Chapter 7
Conclusions and future work

7.1 Conclusions and future work for sensor system

The designing of the amplifier for reading the piezoelectric signals was successful. Even though the ideal response could not be achieved the response in the frequencies of interest was adequate for generating data for this first study in testing sleep and wake state classification. But there are a few problems that exist with this amplifier that can be corrected for future versions of the amplifier.

1) The signal from the piezo film was very sensitive to how much contact area the mouse has on the sensor.
2) In some instances the amplifier tended to saturate for 5-10 seconds. Especially when the mouse was in an active state.

For a future version of the circuit a parallel capacitance can be placed between the piezofilm and the R1 resistances of the differential op amp. This will increase the capacitance seen by the differential op amp and also reduce the cutoff of the highpass filter created by the piezofilm and the R1 resistances of the differential op amp. The extra capacitance will also decrease the sensitivity of the signal to changes in the capacitance of the piezofilm sensor, since this capacitance changes depending on how much contact area the mouse has with the sensor.

Another addition to this amplifier can be the implementing of a voltage trimmer in the differential op amp using the voltage trimming connections available in the op amp. This can be used to control the offset or drift that sometimes occurs in the data. The current system has no such control, and in some severe cases caused the amplifier to saturate for long periods of time.

7.2 Conclusions and future work for classifier.

The extracted features and classifier also worked well for the classification of the data into sleep and wake states. Of the features the PS features seemed to perform the

best, even though at first it was assumed that the CA would perform better due to its ability to characterize non-stationarities, such as transients (from gross movements or transitions between states), and cyclostationarities from the envelope of the respiration motion. The reasons why the PS outperformed the CA are likely to be due to the detrending of the data and the removal of the linear trend from the DFT values which would have taken out any non-stationarities due to ramp-like transitions or level shift in the data that would have obscured the low-frequency features, thus making it more stable and stationary. Another reason for this is likely that the CA is more sensitive to detecting transients or spike-like signals in the data, and in the case of weak harmonics, the transients dominated the CA.

It should be noted that this work concentrated only on a distinction between sleep and wake. But in the future further analysis can be done to try and differentiate sleep into REM and non-REM sleep or other stages in sleep. During this study the breathing rates during sleep seem to vary between 2.5-3.5 Hz. These breathing rates may be related to the different sleep stages the mice are in and will need to be researched further if distinctions between different sleep stages are to be found.

The linear classifier gave error rates as low as 4% and 9% for the two data sets, while the Neural Network classifier gave error rates around 6% and 12% for these two data sets. But a limitation of this study is that the "truth" that the classifiers performance was scored against was based on human observation. The human observation would not account for very short segments of sleep and wake as it is likely that the human would not record sleep and wake states unless they lasted for more then 4 or 5 seconds. Thus even though the features extracted from the data using the algorithms will record sleep and wake in these short segments, the human scoring likely will not. This can lead to higher misclassification error rates as was observed with the individual feature tests without the median filtering. But the median filtering of the features in a way helped in smoothing of these very short sleep and wake recordings in the extracted features and made the comparison of the features and the human scoring more even.

The results that were discussed in this thesis were for around 4 hours of data or more. Even for larger time durations such as 24 hours or more the error rates will be determined by the amount of wake there was in the data (as was discussed in section 6.8). If there

was more wake in the 24 hours that the data was recorded, a higher error rate can be expected. The same holds true for shorter time durations such as 10-15 minutes.

All in all this system was successful in classifying sleep and wake states in mice and will allow the analysis of a large number of mice. It will also take less time and be more cost efficient than methods such as EEG, EMG and EOG.

Reference

[1] Oppenheim Alan V. & Schafer Ronald W. (1999). *Discrete-time signal processing* (2nd ed.). Prentice-Hall.

[2] Shanmugan K. Sam. & Breipohl Arthur M. (1988). *Random signals: detection, estimation and data analysis* (1st ed.). John Wiley & sons.

[3] Chatfield C. (1980). *The Analysis of Time Series: An Introduction* (2nd ed.) Chapman and Hall.

[4] National Center on Sleep Disorders Research. (2001 September 10th). *Neurobiology of sleep and waking: Workshop report.* Retrieved 2005 May. http://www.nhlbi.nih.gov/about/ncsdr/research/research-a.htm

[5] Lo Chung-Chuan, Chou Thomas, Penzel Thomas, Stanley Eugene H.(2004) *Common scale-invariant patterns of sleep-wake transitions across mammalian species.* Retrieved 2005 May. www.pnas.org_cgi_doi_10.1073_pnas.0408242101

[6] Megens A.A.H.P, Voeten J. , Rombouts J. (1987) “Behavioral activity of rats measured by a new method based on the piezo-electric principle”. *Psychopharmacology (1987) 93:382-388*

[7] Sethares William A. , Staley Thomas W. (1999). “Periodicity Transforms” . *IEEE Transactions on Signal Processing vol.47 No.11 November 1999.*

[8] Kanjilal P.P., Palit Sarbani, (1994). “Extraction of Multiple Periodic Waveforms from Noisy Data”. *1994 IEEE International conference on Acoustics, Speech and Signal Processing, vol.2, pp 361-364, April 1994.*

[9] Gerr Neil L., Allen Jeffery C., “The Generalized Spectrum and Spectral Coherence of a Harmonizable Time Series”, *Digital Signal Processing 4, pp 222-238 (1994).*

[10] *Piezo Film Sensors Technical Manual*. (1999) : Measurement Specialties Inc., Retrieved 2005 May.
http://www.msiusa.com/piezo/documentation.asp

[11] Franco Sergio (2002), *Design with operational amplifiers and analog integrated circuits.* New York : McGraw-Hill, 2002.

[12] Huang Lexun (2003), *Classification of Breast Tumor Masses Through Ultrasonic Signal Characterization.* Published Doctoral Thesis, University of Kentucky, Lexington, Kentucky, USA.

[13] Venkatachalam Vidya, Aravena Jorge L. , " Detecting periodic behavior in non-stationary signals" , *Proceedings of the IEEE-SP International Symposium on Time-frequency and Time-scale Analysis" , pp 497-500, Oct. 1998*

[14] Donohue Kevin D., Huang Lexun , "Ultrasonic scatterer structure classification with the generalized spectrum", *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3401 – 3404, May 2001.*

[15] Donohue Kevin D., Black Tim R., " Pitch Determination of Music Signals Using the Generalized Spectrum", *IEEE SoutheastCon 2000 'Preparing for the New Millennium', Apr 7-Apr 9 2000, Nashville, TN, USA*

[16] Burden Richard, & Faires Douglas J. (1997). *Numerical Analysis* (6th ed.). Brooks/Cole.

VITA

Dharshan Charith Medonza was born on the 11th of April 1975 in Colombo, Sri Lanka. He received his Bachelor of Science degree in Electrical Engineering from the University of Kentucky in May of 2000, after which he worked as an Application Support Engineer for the MathWorks Inc. in Natick, Massachusetts. He returned to the University of Kentucky and completed his Master of Science degree in Electrical Engineering in November of 2005. His research interests is in the area of Digital Signal Processing. He is also an IEEE member.