THE APPLICATION OF BLIND SOURCE SEPARATION TO FEATURE DECORRELATION AND NORMALIZATION

by

Manuel Laura

BS, University of Kansas, 2003

Submitted to the Graduate Faculty of

the School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2005

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Manuel Laura

It was defended on

April 11th 2005

and approved by

Amro A. El-Jaroudi, Associate Professor, Department of Electrical and Computer Engineering

Luis F. Chaparro, Associate Professor, Department of Electrical and Computer Engineering

Patrick Loughlin, Associate Professor, Department of Electrical and Computer Engineering

Thesis Advisor: Amro A. El-Jaroudi, Associate Professor, Department of Electrical and Computer Engineering

ABSTRACT

THE APPLICATION OF BLIND SOURCE SEPARATION TO FEATURE DECORRELATION AND NORMALIZATION

Manuel Laura, M.S.

University of Pittsburgh, 2005

We apply a Blind Source Separation (BSS) algorithm to the decorrelation of Mel-warped cepstra. The observed cepstra are modeled as a convolutive mixture of independent "source" cepstra. The algorithm aims to minimize a cross-spectral correlation at different lags to reconstruct the source cepstra. Results show that using "independent" cepstra as features leads to a reduction in the WER.

Finally, we present three different enhancements to the BSS algorithm. We also present some results of these deviations of the original algorithm.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Preface

I this section I would like to thank all the people who guided and encouraged me through this journey in life.

I necessarily have to start by thanking my advisor, Dr Amro A. El-Jaroudi. I am very grateful for the knowledge and wisdom shared with me. I know that without his help and continuous motivation this work would not have had the quality of the final product we achieved.

I would like to give thanks to Zuochang Zheng (Abel), the other graduate student working in this project. Without his assistance I would not have been able to deal with some implementation problems.

Finally, I would also like to give thanks to my family for their continuous moral support from a distance and to Patti for her tireless day-to-day listening of the current problem and moral support through the thesis writing.

1.0 INTRODUCTION

In recent years, a better understanding of speech recognition has been achieved. The rapid increase in processing power has given speech recognition systems the means to engineer models and tools that better extract the features of speech. Speech engineers have gathered knowledge from different fields to better identify and reproduce the human speech. Two of these fields are digital signal processing DSP and probability and statistics. For example, speech engineers use digital filters or discrete transforms to manipulate or to study characteristics of the signal that are not obvious in the time domain. We also use statistical models and concepts to describe signal's behavior.

Even though we have multiple tools to process speech signals, there are still many problems to be solved. One of these problems is the large variability and redundancy in speech features used in speech recognition. Most speech recognition systems model the speech features as independent random variables using Gaussian distributions with diagonal covariance matrices. In reality, the speech features are correlated creating a mismatch between the data and the model. Approaches such as covariance normalization have been attempted to correct this mismatch with moderate success.

The solution we propose for this problem is the application of a blind source separation (BSS) algorithm to decorrelate the speech features. The use of this algorithm to decorrelate and normalize the speech features decreases the word error rate (WER) of a speech recognition system, as we show in Chapter 4.

Originally, BSS gave a solution to the the *cocktail party problem.* This problem is described briefly to understand the similarity with our problem. When a person is surrounded by many "sources" of speech and noise is hard for him or her to focus on a single emitter. However, a person can establish a conversation with another person or listen to a particular noise or sound in the environment. The nature of how the human brain separates the different sounds and noises is complex and unknown. Blind Source Separation (BSS) is a solution to this problem. The approach is called "blind" because it involves no training or apriori information about the sources. The only data available is the observed mixture of sounds. The main assumption of this approach is that the unobserved signals are mutually independent. In this work we explain how the BSS algorithm manipulates a mixture of signals (the speech features) to obtain the desired property on them: mutual independence.

BSS have been used to decompose other types of signals besides speech signals; i.e., it is not limited to audio signals as it was originally used. Consequently, the approach has been used in a variety of applications; for example: data analysis and compression, Bayesian detection, source localization. It has also been used in different fields such as bioimaging, speech recognition, spectral estimation, etc. In our application, we use the BSS approach to decompose mel-warped cepstra into a set of independent features for use in speech recognition.

The Speech Recognition system used to test the BSS algorithms consists of two phases: the training and the decoding stage. We introduce the BSS algorithm in each stage to decorrelate speech features. We input the mel-warped cepstra into the BSS algorithm and use the algorithm as a filter. Including the BSS algorithm in our speech recognition system has considerably improved the Word Error Rate (WER) of the overall system.

This manuscript is divided into six chapters. In Chapter 1 we give an overview of most of the tools we use to analyze the BSS algorithm. We divide it into three sections: stochastic processes, digital signal processing, and speech recognition. In Chapter 2 we formulate the speech feature decorrelation problem. In Chapter 3 we develop the BSS algorithm. We subdivide the decorrelation algorithm chapter into five sections. In the first three sections we explain how to calculate the optimal filter to decorrelate the features. In the fourth section we show the computational cost of the algorithm. In the last section we give a summary of the whole algorithm emphasizing its constraints and parameters. In Chapter 4 we show and discuss our results graphically and numerically. In Chapter 5 we provide some possible enhancement to the BSS algorithm. Finally in Chapter 6 we show our conclusions.

Before we go in depth into the BSS algorithm and show its mel-warped cepstrum analysis application, we briefly review some basic concepts and how they relate to our work.

1.1 STOCHASTIC PROCESSES

Signals can be divided into two groups: those that have a fixed behavior and those that change randomly. Unfortunately, most real world signals of interest are random or stochastic like speech. Stochastic signals cannot be modeled by a simple, well-behave mathematical equation and their future values cannot be completely predicted. Instead, we have to use tools from probability and statistics to analyze their behavior. Also, knowing an individual signal is not useful because signals change randomly. Instead average values from a collection of signals give us insight on the usual behavior of the signals. These collections of signals are called random processes.

In speech, one can have an idea what word will follow the current one based on grammar or syntax. However, one cannot be certain of what word will be the next word in a sentence. Thus, it is a stochastic process. One can use stochastic methods to model speech in an effective way.

In this section we discuss some basic stochastic concepts: random variables and processes, independence between random variables, covariance, cross-correlation and cross power spectrum.

1.1.1 Random Variables and Processes

A random variable is the mapping from a probability space to a measurable space. This measurable space contents all the possible outcomes yield by the probability space. This mapping is non-deterministic or random. For example, picking cards from a deck and recording the suits yields a random variable with range {spades, clubs, hearts, diamonds}.

A sequence or collection of random variables constitutes a random process. There are two types of random processes: the countable or discrete and the uncountable or continuous. The notion of random process is of extreme importance to us because speech is a random process.

1.1.2 Independence

Two random variables or processes X and Y are statistically independent if and only if the conditional probability of X given Y, $P(X|Y)$, is equal to the probability of X. In other words, this means that Y does not contain any information about X. Thus, knowing Y does not chance the conditional probability of X.

$$\begin{aligned} P(X|Y) &= \frac{P(X,\ Y)}{P(Y)} \\ &= \frac{P(X)P(Y)}{P(Y)} \\ &= P(X) \end{aligned} \tag{1.1}$$

Equation 1.1 implies that the joint probability of X and Y is equal to the product of the probability of X times the probability of Y:

$$P(X,Y) = P(X)P(Y) \tag{1.2}$$

The two vectors X and Y can also be considered to be orthogonal to each other. In our application, we want to decompose mel-warped cepstra into a set of mutually independent features. Thus

$$P(X_1, X_2, X_3, ..., X_N) = P(X_1)P(X_2)P(X_3)...P(X_N)$$

where N is the number of features used by the mel-cepstrum analysis. We explain how we perform this task using the BSS algorithm in Chapter 3.

1.1.3 Covariance and Cross-Correlation

The covariance between two processes $x(t)$ and $y(t)$, measures how similar the processes are. This is analogous to the variance of a single variable. More precisely, the covariance between random processes measures how much the deviation between them match. This is important to us because signals are random processes, as it is mentioned in the previous section, and this tool measures how closely related the signals are.

Mathematically, the covariance between $x(t)$and $y(t)$ is defined as

$$\mathbf{C}_{xy}(t) = E\{(\mathbf{x}(t) - \mu)\ (\mathbf{y}(t) - \nu)\} \tag{1.3}$$

where E{.} represents expected value and μ is equal to the mean of $x(t)$ and ν is equal to the mean of $y(t)$.

One important property of the covariance is that a positive value indicates that the two random process tend to increase together. On the other hand a negative value of the covariance represents that an increase in one random process is accompanied by a decrease in the other random process.

If two random processes are independent or uncorrelated or if one of them is identical to its mean at all times, then

$$\mathbf{C}_{xy}(t) = 0 \tag{1.4}$$

However, a covariance equal to zero does not mean that the random processes are independent. It only means that the two random processes are "linearly independent." It does not say anything about their higher moments.

If the means of the random process are equal to zero equation 1.3 simplifies to

$$\mathbf{C}_{xy}(t) = E\{\mathbf{x}(t)\ \mathbf{y}(t)\} \tag{1.5}$$

Another measurement tools, similar to the covariance is the cross-correlation. The cross-correlation is another measurement of dependency between random processes. It is defined as

$$\mathbf{R}_{xy}(t) = E\{\mathbf{x}(t)\ \mathbf{y}(t)\} \tag{1.6}$$

The cross-correlation between two random processes is the same as their covariance if the mean of the random processes is equal to zero.

These concepts are of importance to us because the BSS algorithm is based on the concepts of covariance and cross-correlation as we will see in Chapter 3

1.1.4 Cross-Power Spectrum

The Cross-Power Spectrum density of two random processes $x(t)$ and $y(t)$ is defined as the Fourier Transform of the cross-correlation function $R_{xy}(t)$ as

$$\mathbf{R}_{xy}(\omega) = \int_{-\infty}^{\infty} \mathbf{R}_{xy}(t) e^{-j2\pi ft} dt \tag{1.7}$$

1.2 DIGITAL SIGNAL PROCESSING

In this section we cover some basic DSP concepts that help us develop the BSS algorithm we use to decorrelate speech features. We discuss some basic theory on digital filters and explain their importance to us.

1.2.1 Digital Filters

Any “Black Box” that relates an input $x[n]$ to an output $y[n]$ can be considered a digital filter. Thus, a filter can be a window, an amplifier, a modulator, etc. The output of a linear time-invariant digital filter is related to the input by an N^{th} order difference equation of the form:

$$y[n] = \sum_{k=1}^{N} \alpha_k y[n-k] + \sum_{k=0}^{M} \beta_k x[n-k] \tag{1.8}$$

where x is the input to the system and y is the output. In other words, Equation 1.8 says that the current output depends on the current and past values of the input and past values of the output. Figure 1.1 shows the block diagram representation of Equation 1.8.

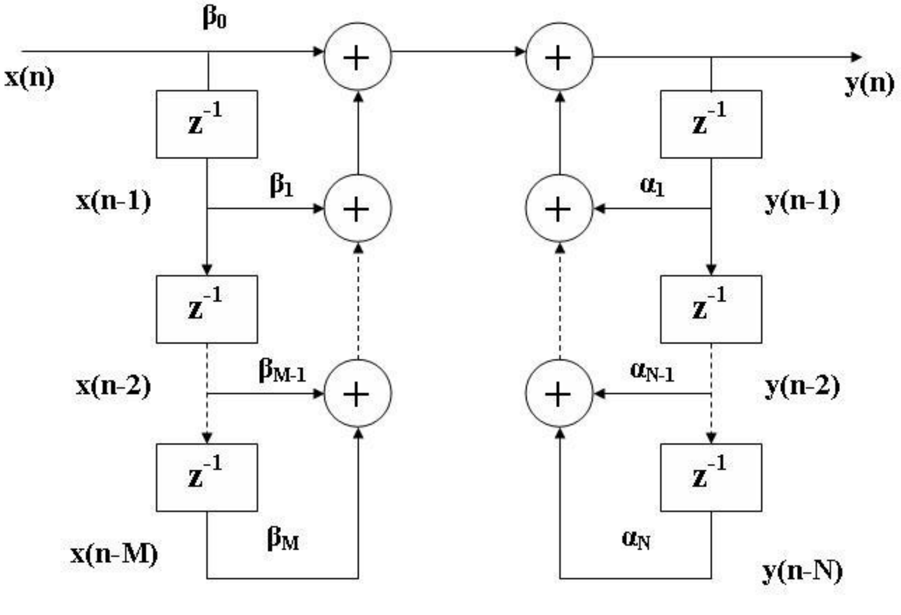


Figure 1.1: Block diagram representation for a general N^{th} order difference equation

We can describe the filter by finding its frequency response. We can do this by transforming Equation 1.8 to the z domain, factoring $Y(z)$ and finally dividing $Y(z)$ by $X(z)$; i.e.

$$\begin{aligned}
Y(z) &= \sum_{k=1}^{N} \alpha_k Y(z) z^{-k} + \sum_{k=0}^{M} \beta_k X(z) z^{-k} \\
Y(z)(1 - \sum_{k=1}^{N} \alpha_k z^{-k}) &= \sum_{k=0}^{M} \beta_k X(z) z^{-k} \\
Y(z) &= \frac{\sum_{k=0}^{M} \beta_k X(z) z^{-k}}{(1 - \sum_{k=1}^{N} \alpha_k z^{-k})}
\end{aligned}$$

Then,

$$\begin{aligned}
H(z) &= \frac{Y(z)}{X(z)} \\
&= \frac{\sum_{k=0}^{M} \beta_k z^{-k}}{1 - \sum_{k=1}^{N} \alpha_k z^{-k}} \qquad (1.9)
\end{aligned}$$

Filters can be classified into two categories: Finite impulse response (FIR) and infinite impulse response (IIR) filters. The next two subsections briefly discuss these categories.

1.2.1.1 FIR Filters As the name implies it, this type of filter has a finite duration in time. The filter has no feedback, so equation 1.8 simplifies to

$$y[n] = \sum_{k=1}^{N} \alpha_k y[n-k] \quad N < \infty \qquad (1.10)$$

This means that the current output only depends on the current and past values of input. Figure 1.2 shows the block diagram of a FIR filter.

The frequency response of the filter can be found in the same manner as for the regular filters. Its frequency response is given by

$$H(z) = \sum_{k=0}^{M} \beta_k z^{-k} \qquad (1.11)$$

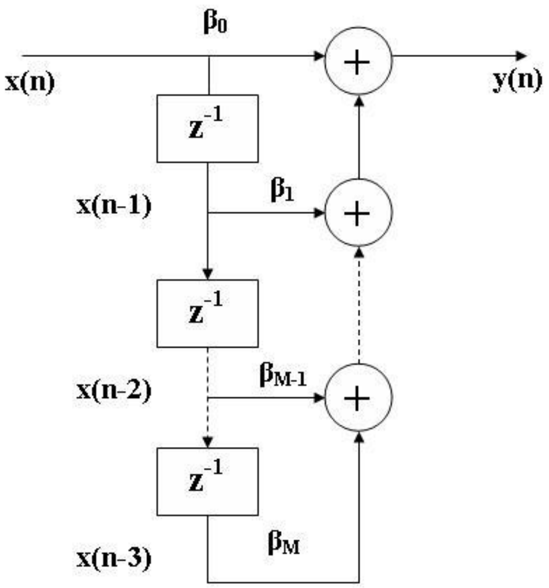


Figure 1.2: Block diagram representation for a FIR filter

Equation 1.11 is similar to Equation 1.9 but without the denominator. This is due to the lack of feedback of the filter. The only operators we need to implement this filter are: delays, multipliers and adders.

Finally, FIR filters are stable due to the finite duration constraint and they are causal since they only depend on the current and past value of the input.

We will use a FIR filter implementation to find the optimal filter for our BSS algorithm.

1.2.1.2 IIR Filters The complement of FIR filters are IIR filters. This type of filters has infinite response in time. Equation 1.8 describes them. Thus, the implementation of the filter includes a feedback path and the frequency response has a denominator. This also implies that these filters have poles and zeros.

There is more flexibility in the implementation of this type of filter than in the implementation of FIR filters [2]. However, we will not cover this topic.

Currently we are implementing an IIR filter version of our BSS algorithm. In Chapter 5 we explain the reason for this approach.

1.3 SPEECH RECOGNITION

In this section we cover Word Error Rate (WER), cepstrum analysis, speech features, and a basic introduction to blind source separation.

1.3.1 Word Error Rate

The Word Error Rate (WER) of a Speech Recognition (SR) system is a measure of the accuracy of the system. It calculates a percentage of how many words are recognized correctly by the system. There are three types of errors in an SR system:

- **Substitution.-** When the correct word is replaced by an incorrect word in the recognition sequence.
- **Deletion.-** When the correct word is replaced by an empty string in the recognition sequence.
- **Insertion.-** When an incorrect word is inserted into the recognition sequence.

We can see that from these three types of errors, the WER of a SR system can be greater than 100% due to the insertion error. We can have a system that not only fails to recognize every word but also inserts words that were not in the original recognition sequence.

Finally, the WER of a system should be taken as a reference measurement and not as an absolute measurement of performance; i.e. a system with a greater WER can perform better in certain cases than another system with a lesser WER. This is because the second system might miss all the key words in a sentence while the first one only captures the keywords in the same sentence.

To test the performance of our BSS process, we compare the WER of the SR system with the BSS algorithm against the WER of the SR system without the BSS algorithm (baseline).

1.3.2 Cepstrum Analysis

According to some models, speech is composed of two signals convolved together: an excitation sequence and an impulse response of the vocal tract model. The former one has fast variations while the latter one has slow variations over time and contains the phonetic information necessary for speech recognition. In speech analysis one wants to eliminate the excitation sequence in order to clearly capture the vocal tract information. However, these two signals are not combined in a linear fashion. One needs to use mathematical tools to analyze only the impulse response of the vocal tract without the interference of the excitation sequence. Cepstrum analysis is widely used in speech recognition to perform this task. It allows the user to separate these two signals. The mixture of signals can be represented as

$$x(n) = e(n) \star \theta(n) \tag{1.12}$$

where $x(n)$ is the speech signal, $e(n)$ is the excitation signal and $\theta(n)$ is the impulse response of the vocal tract.

The ultimate goal of cepstrum analysis is to separate these two components in a linear manner. One could do this operation preserving the phase of the complex numbers using complex cepstrum analysis. However, the real version for the analysis is usually exercised because the phase is not worth the computational complexity [1].

It can be shown that the Discrete Time Fourier Transform (DTFT) of Equation 1.12 is given by

$$X(\omega) = E(\omega)\Theta(\omega) \tag{1.13}$$

where ω stands for frequency[1].

By taking the natural logarithm one turn the multiplication in Equation 1.13 into a summation of the two components.

$$\begin{aligned} C_x(\omega) &= \ln|X(\omega)| \\ &= \ln|E(\omega)\Theta(\omega)| \\ &= \ln|E(\omega)| + \ln|\Theta(\omega)| \\ &= C_e(\omega) + C_\theta(\omega) \end{aligned} \tag{1.14}$$

Finally, one can take the Inverse Discrete Time Fourier Transform (IDTFT) to finally get the cepstrum signal in the "quefrequency" domain.

$$\begin{aligned} c_x(n) &= \mathcal{F}^{-1}\{C_e(\omega)\} + \mathcal{F}^{-1}\{C_\theta(\omega)\} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} C_e(\omega)e^{jwn}d\omega + \frac{1}{2\pi}\int_{-\pi}^{\pi} C_\theta(\omega)e^{jwn}d\omega \\ c_x(n) &= c_e(n) + c_\theta(n) \end{aligned} \tag{1.15}$$

Figure 1.3 shows the block diagram representation of how to compute the Real Cepstrum.

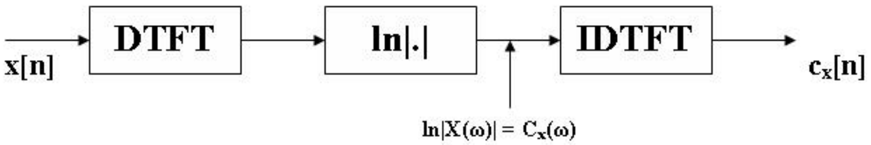


Figure 1.3: Block Diagram that represents the calculation of the Real Cepstrum

From the derivation of equation 1.14, one can see that the cepstra function in the frequency domain (ω) is real and even. Thus, one can use a cosine function instead of an exponential to find the Real Cepstrum in the *quefrequency* domain.

The *quefrequency* domain is a scaled version of the time domain. We are introduced into this domain after taking the natural logarithm of the frequency domain and then the inverse Fourier Transform as equations 1.14 and 1.15, show. This domain is called *quefrequency* because of its similarity with the frequency domain. Moreover, the term *cepstrum* comes as an analogy to spectrum.

A graphical representation of the analysis of the mixed signal, the excitation signal and the impulse response of the vocal system model is shown in Figure 1.4. This Figure also shows how the excitation sequence and the impulse response of the vocal system model are separated in the *quefrequency.* This characteristic is what we try to exploit with the BSS algorithm. This is further explained in Chapter 3.

However, this is not the only cepstrum implementation. Stevens and Volkman conducted experiments in 1937 in which they introduced the mel-scale. Mel is a unit that measures perceived pitch or frequency of a tone. It does not correspond linearly to a physical frequency as it is usually measured, in Hz. These two researchers arbitrarily chose a frequency equal to 1000 Hz and designated it to be the reference point: 1000 mels. Then, some listeners were asked to change the pitch frequency until it felt as twice and as ten times the reference frequency. Then they were asked to change the frequency of the pitch to half and $\frac{1}{10}$ as of the reference frequency. Stevens and Volkman labeled these measurements as 2,000 and 10,000 mels and 500 and 100 mels respectively. Then, these measurements were mapped into the real frequency (Hz) scale. The mapping is shown in Figure 1.5. We can observed in this figure that the mapping is almost linear below 1 kHz and logarithmic above.

The mel scale exploits auditory principles and decorrelating property of the cepstrum because of its logarithmic behavior. Moreover, the mel scale is a natural measurement of speech; i.e. it is a scale that the human brain uses to measure sounds.

This change of scale has proven to provide better results than when using the conventional cepstrum for many phonetically similar monosyllabic words [9].

The calculation of the mel-cepstrum is not too different from the calculation of the regular cepstrum. Figure 1.6 shows the block diagram implementation.

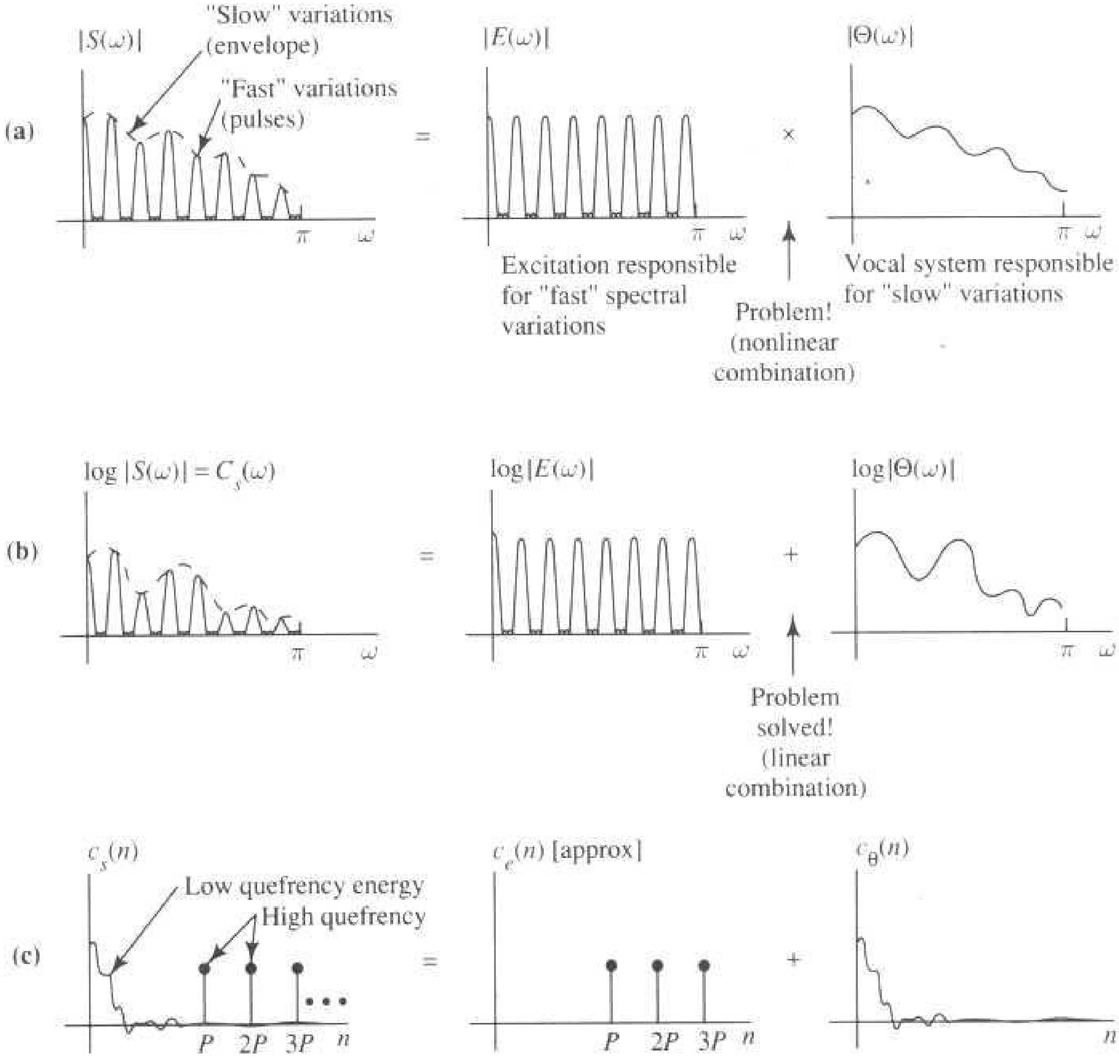


Figure 1.4: Graphical Interpretation of the Cepstral Analysis [1]

The first step that Quatieri suggests is to window the speech signal shown in Equation 1.12 and then take the DTFT of the resultant.

$$X(n, \omega_k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_k m} \tag{1.16}$$

where $\omega_k = \frac{2\pi}{N}k$ and N is the length of the DTFT. This is the Short Time Fourier Transform (STFT).

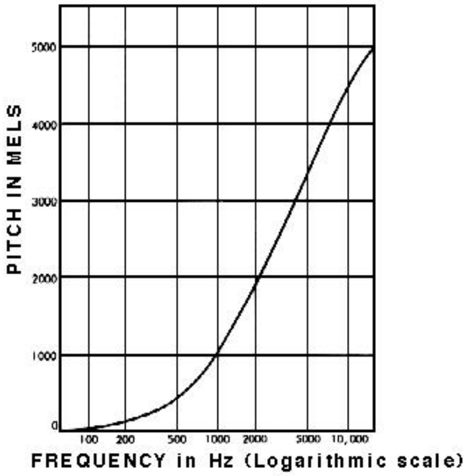


Figure 1.5: The mel scale as a function of frequency [6]

Then, Quatieri determines the energy in the STFT weighted by each mel-scale frequency response. $V_l(\omega)$ represents the frequency response of the l^{th} mel-scale filter in Figure 1.6.

$$E_{mel}(n,l) = \frac{1}{A_l} \sum_{k=L_l}^{U_l} |V_l(\omega_k) X(n,\omega_k)|^2 \tag{1.17}$$

where L_l and U_l represent the lower and upper frequency indices over which each filter is different than zero.

A_l is equal to $\sum_{k=L_l}^{U_l} |V_l(\omega_k)|^2$ in order to normalize the filters according to their varying bandwidths. This is done to give equal energy for a flat impulse response [9].

Finally, we can use the fact that the cepstrum is real and even and find real mel-cepstrum using only a cosine, in the same way we suggested in the real cepstrum case.

$$C_{mel}[n,m] = \frac{1}{R} \sum_{l=0}^{R-1} \ln\{E_{mel}(n,l)\} \cos\left(\frac{2\pi}{R} lm\right) \tag{1.18}$$

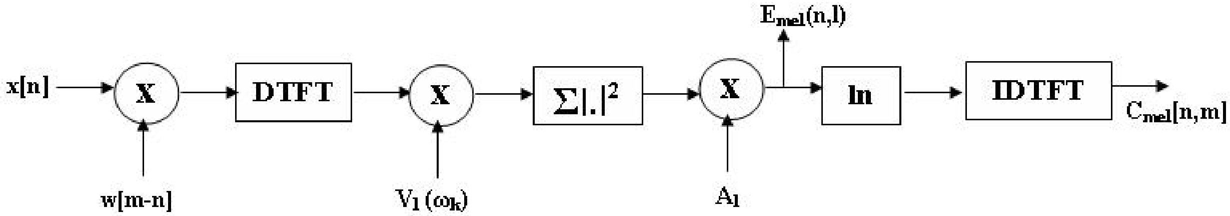


Figure 1.6: Block diagram of the mel cepstrum calculation

1.3.3 Speech Features

We can divide the cepstrum analysis into two types of results: “low time” and “high time” results. The “low time” corresponds to small values in the time axis and the ”high time” corresponds to large values in the time axis. This is similar to low and high frequency. The “low time” coefficients correspond to the impulse results of the vocal tract and the “high time” coefficients correspond to the excitation sequence as shown in Figure 1.4. From these two results we only take the one that corresponds to the “low time” because this section contains the phonetic information and the shape of the vocal tract. This information is used to train and decode speech recognition systems. In our speech recognition system, only the first fifteen coefficients of the cepstrum analysis are considered. The rest of the coefficients that correspond to the “high time” are neglected. Each of these coefficients can also be considered as a dimension of the cepstum result.

The first dimension of the cepstra $\mathbf{c}_x(0)$ usually describes the overall energy contained in the spectrum. The second dimension, $\mathbf{c}_x(1)$, measures the balance between the upper and lower halves of the spectrum. Usually, higher order coefficients are concerned with increasingly finer features in the spectrum [1].

Most speech recognition systems assume that the cepstrum or mel-warped cepstrum features are orthogonal or statistically independent of each other [1]. The cepstrum features carry redundant information between them. In Chapter 3 we show how the BSS algorithm enforces the independence of the speech features.

1.3.4 Blind Source Separation

As mentioned before, the Blind Source Separation (BSS) algorithm task is to reconstruct statistically independent signals. This task is is briefly discussed. Assume a vector of dimension d_s that is composed of statistically independent sources $\mathbf{s}(t) = [s_1(t), s_2(t)..., \ s_{d_s}(t)]^T$. These sources are convolved with an unknown channel response and mixed to produce the observation vector of dimension d_x, $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_{d_x}(t)]^T$. In other words,

$$\mathbf{x}(t) = \sum_{\tau=0}^{P} \mathbf{A}(\tau)\mathbf{s}(t-\tau) \tag{1.19}$$

where $\mathbf{A}(\tau)$ is the channel impulse response with $d_s d_x P$ coefficients.

Given $\mathbf{x}(t)$, the goal is to recover $\mathbf{s}(\mathbf{t})$. If we attempt to use (1.19) to recover $\mathbf{s}(t)$ (*the Forward Model* approach), we would need to calculate the response matrix $\mathbf{A}(\tau)$ (not an easy task) and then hope the matrix is invertible so we could recover the vector of sources.

Another approach is to estimate the sources by passing the observations through a FIR filter to undo the convolutive and mixing effect of the channel, $\mathbf{A}(\tau)$, i.e.,

$$\bar{\mathbf{s}}(t) = \sum_{\tau=0}^{Q} \mathbf{W}(\tau)\mathbf{x}(t-\tau) \tag{1.20}$$

where $\mathbf{W}(\tau)$ is an FIR inverse model with $d_s d_x Q$ coefficients and $\bar{\mathbf{s}}(t)$ is the estimate of the sources signals. This approach is called the *Backward Model* approach [10].

To solve the problem proposed in Equation 1.19 or 5.1 numerous approaches have been presented (e.g., by Cardoso, Herault and Jutten, Pham) for the instantaneous mixture case ($P = 1, \ and \ Q = 1$) [11], [12], [13]. Also, various methods have been proposed to address the convolutive mixture case by Yellin and Weinstein, Thi and Jutten, Shamsunder and Giannaki, [14], [15], [16].

Initially, it was thought that only decorrelating the measured signals was enough to make the signals independent [18], [19]. This approach was sufficient to recover the sources for the instantaneous case of nonwhite signals. However, this solution is not unique when applied to a convolutive mixture ($P > 1$ or $Q > 1$). One has to arbitrarily select a part of the filter and then, based on this selection, finish calculating the filter. Moreover, the solutions might be decorrelated but not statistically independent since uncorrelated signals do not imply independent signals as it was mentioned in Section 1.1.3. In 1993, Weinstein *et al* clearly indicated that additional conditions are required to achieve independent signals.

In 2003, Parra and Spence [10] suggested an algorithm that utilizes additional information provided by non-stationary signals. It incorporates statistics at multiple frequencies and the effect of additive noise while estimating the independent sources, thereby increasing its robustness. The algorithm takes into account second order statistic squared of non-stationary signals. This gives enough conditions to completely describe the FIR filter and reconstruct independent signals.

We adapt the algorithm by Parra and Spence and apply it to deconvolve multidimensional mel-warped cepstrum. In our application, we restrict the number of independent sources to equal the number of observed signals so as not to lose any dimensionality in our acoustic features.

The BSS algorithm is included in two sections of the speech recognition system: the training and the decoding stages. Multiple combinations of parameters have been tested to improve the WER of the system. These combinations are shown in Chapter 4.

2.0 PROBLEM FORMULATION

Mel-warped cepstrum features are extracted from the same source: the speech signal as shown in Equation 1.12. Thus, each speech feature is expected to contain unique information about the speech signal. The features are assumed to be statistically independent; i.e., orthogonal to each other. However, this is not the case. The features have redundant information.

As we mention in Chapter 1, BSS is used in speech processing to deconvolve an incoming mixture of speech signals into their sources; i.e., produce output signals statistically independent of each other. The features of a speech signal pose a similar problem: they are a mixture of signals coming from the same source: the speech data. We want the features to be independent of each other. Thus, we have to impose an independence condition on the features.

We have implemented the BSS algorithm suggested by Parra and Spence [10] into the EARS system. The EARS system is a speech recognition system specifically focus on Conversational Telephone Speech CTS. It uses 40 hours of Swicthboard training data, DEV01 test set and performed unadapted training and decoding. The baseline's WER of the system experiment is 44.99%

We will use the (*Backward Model*) approach in our application of BSS to speech recognition. In our model, $\mathbf{x}(t)$ represents the observed mel-warped cepstrum features corresponding to a speaker, with t representing the frame index. Each speaker has multiple utterances. $\bar{\mathbf{s}}(t)$ represents the new mutually independent features (or bss_cepstrum as we call it in the algorithm) to be used in building the acoustic models.

Initially, we tested the algorithm in Matlab and to improve the computational time, we implemented the algorithm in C. Then we worked on the interface issues with the EARS system and placed the BSS program to capture the mel-cepstrum features, process them, and output independent mel-cepstrum features. Finally, we optimized the BSS algorithm to improve the results of the Speech Recognition system.

3.0 ALGORITHM DEVELOPMENT

The key aspect of the algorithm presented by Parra and Clay [10] is the estimation of the optimal filter $\mathbf{W}$ in the frequency domain by solving simultaneous separation problems for every frequency. This transforms the time-domain convolutive mixture into an instantaneous mixture at each frequency. We then define a cost function in the frequency domain that, when minimized, produces an optimal filter that separates the input mel-cepstrum into independent mel-cepstra features.

As mentioned in Subsection 1.3.4, the algorithm is based on the FIR *backward model* described in Equation (5.1). From this equation, we can see that we need to calculate $d_s d_x Q$ coefficients to completely describe the FIR filter.

Figure 3.1 shows a graphical interpretation of the input/output relationship of the BSS filter for each output. A similar setup is used to estimate each of the d_s sources.

The BSS algorithm has three steps:

1. Estimate the cross power spectrum of the observed (CPSO) signals (mel-warped cepstrum) as a function of blocks of time, $\overline{\mathbf{R}}_x(\omega, t_k)$.
2. Estimate the optimal filter $\mathbf{W}$ that minimizes a Least Squared Cost function, J, based on the Cross Power Spectrum of the estimated source signals as a function of of blocks time, $\Lambda_{\overline{s}}(\omega, t_k)$.
3. Convolve the optimal FIR filter $\mathbf{W}$ with the input mel-warped cepstrum to produce the final estimate of the independent source signals.

In this Chapter, we discuss in detail the three steps of the BSS algorithm. Moreover, we show its input parameters and constraints, and the computational cost of the BSS algorithm [10].

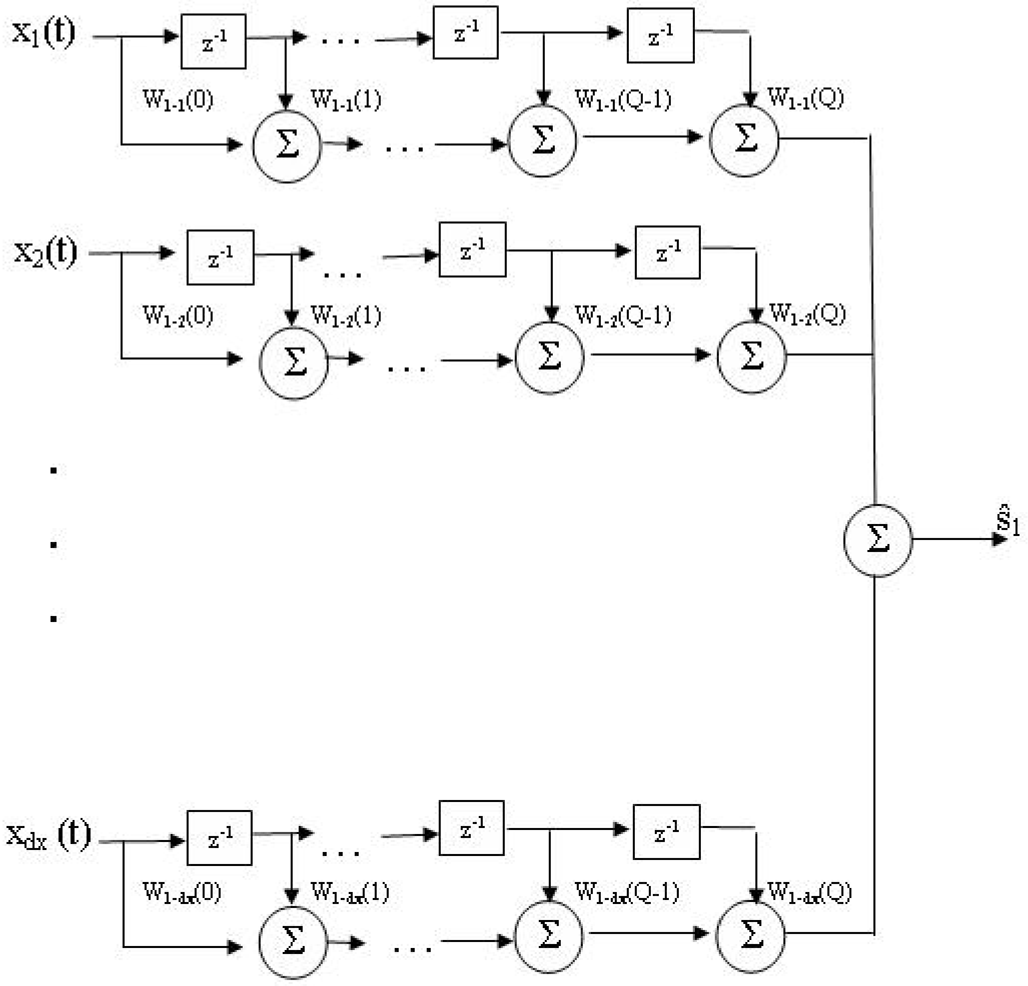


Figure 3.1: Input/output relation for the first source estimate

3.1 CROSS POWER SPECTRUM OF THE OBSERVED AND ESTIMATED SIGNAL

The first step of the algorithm is to find the Cross Power Spectrum of the observed signal (CPSO). Our goal, as stated before, is to find a CPSO function in terms of frequency at different lags. However, if we want the CPSO functions to keep the linearly independent condition at the different frequencies, the mel-cepstra are required to be nonstationary.

Before calculating the CPSO, we analyze the cross-correlation of the instantaneous case ($P = 1$ *or* $Q = 1$). For this case Equation 1.19 simplifies to

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{3.1}$$

and $\mathbf{A}$ only has $d_x d_s$ coefficients. Then the cross correlation of the observation can be found by

$$\begin{aligned} \mathbf{R}_x(t) &= < \mathbf{x}(t)\ \mathbf{x}(t)^T > \\ &= \mathbf{A} < \mathbf{s}(t)\ \mathbf{s}(t)^T > \mathbf{A}^T \\ &= \mathbf{A}\boldsymbol{\Lambda}_s(t)\mathbf{A}^T \end{aligned} \tag{3.2}$$

Since we know the sources are independent, $< \mathbf{s}(\tau)\ \mathbf{s}(\tau)^T >$ simplifies to a diagonal matrix, $\boldsymbol{\Lambda}_s(\tau)$. Also, we can see that any permutation of scaling corresponding to $\boldsymbol{\Lambda}_s(t)$ can be absorbed by the matrix $\mathbf{A}$. Thus, we can choose the scaling and permutation coefficients of the coordinates in $\mathbf{s}$. Thus, we can choose $A_{ii} = 1, i = 1, ..., d_s$. This places d_s conditions on our solution.

If we consider the nonstationary case, a set of K equations of the form 3.2 for different blocks of time: $t_1, t_2, ..., t_K$ and the d_s scaling condition gives us a total of $Kd_x(\frac{d_x+1}{2}) + d_s$ constraints. We also know we have $d_s d_x + d_s K$ unknown parameters corresponding to $\mathbf{A}$ and $\Lambda_s(t_1), \Lambda_s(t_2), ..., \Lambda_s(t_K)$. In our case, we have the same number of sources as sensors so $d_s = d_x$. Equating the number of constraints to the number of unknowns, we conclude that K has to be greater or equal to 2.

On the other hand, the backward model equation (5.1) for the instantaneous case simplifies to

$$\bar{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t) \tag{3.3}$$

Moreover, if we combine this result with the result from Equation 3.1, we get

$$\bar{\mathbf{s}}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t) \tag{3.4}$$

Using Equation 3.4, we can find the cross correlation of this approximation using the same procedure as for the forward model case.

$$\begin{aligned}
\mathbf{\Lambda}_{\overline{s}}(t) &= < \overline{\mathbf{s}}(t)\ \overline{\mathbf{s}}^T(t) > \\
&= < \mathbf{WAs}(t)\ [\mathbf{WAs}(t)]^T > \\
&= < \mathbf{WAs}(t)\ \mathbf{s}(t)^T \mathbf{A}^T \mathbf{W}^T > \\
&= \mathbf{WA} < \mathbf{s}(t)\ \mathbf{s}(t)^T > \mathbf{A}^T \mathbf{W}^T \\
&= \mathbf{WR}_x(t)\mathbf{W}^T \qquad (3.5)
\end{aligned}$$

Likewise to the filter in Equation 3.2, we can choose the scaling and permutation of the coordinates in $\mathbf{s}$ so: $W_{ii} = 1, i = 1, ..., d_x$. This places d_x conditions on our solution. In the nonstationary case, a set of K equations of the form 3.5 and the d_x scaling conditions gives us a total of $Kd_s(\frac{d_s+1}{2}) + d_x$ constraints. At the same time we have $d_x d_s + d_x K$ unknown parameters corresponding to $\mathbf{W}$ and $R_x(t_1),\ R_x(t_2),\ ...,\ R_x(t_K),$.

If we equate the constrains with the unknowns,

$$Kd_s\frac{d_s+1}{2} + d_x \geq d_s d_x + Kd_x \qquad (3.6)$$

we can conclude that in order to have enough constraints we need at least $K = 2$. This is one of the key constraints for the parameters of the algorithm.

For the convolutive mixture we want to have equations similar to 3.2 and 3.5. However, we transform the problem in the frequency domain to change the convolutive mixture into multiple instantaneous mixtures. Consequently, we end up with equations that are functions of frequency and time. This means that we have to calculate cross power spectrum of the mel-warped cepstrum at different lags. As we mention in Subsection 1.1.4, cross-power spectrum is the DTFT of the cross correlation. Thus,

$$\begin{aligned}
\overline{\mathbf{R}}_x(\omega, t) &= \mathbf{A}(\omega)\mathbf{\Lambda}_s(\omega, t)\mathbf{A}^H(\omega) \qquad (3.7) \\
\mathbf{\Lambda}_{\overline{s}}(\omega, t) &= \mathbf{W}(\omega)\overline{\mathbf{R}}_x(\omega, t)\mathbf{W}^H(\omega) \qquad (3.8)
\end{aligned}$$

However, in these equations we are assuming circular convolution instead of linear convolution as Equations 1.19 and 5.1 propose. The solution for this problem is to approximate linear convolution with circular convolution. We can do this by choosing the size of the DTFT, T, much larger than the size of the channel, $\mathbf{A}$ or $\mathbf{W}$, according to the model used. Thus,

$$T \gg P \quad \vee \quad T \gg Q \tag{3.9}$$

for the forward and backward model respectively. This is another of the key constraints of the algorithm. The size of the DTFT has to be much larger than the size of the filter.

Now, we can calculate the cross power spectrum of the nonstationary mel-warped cepstrum at different lags. We assume that the mel-warped cepstrum is stationary within the lags. Moreover, in the stationary cases, the cross correlation does not depend on absolute time but in the relative time between the signals. Thus,

$$\begin{aligned} \mathbf{R}_x(t, t+\tau) &= E\{\mathbf{x}(t)\,\mathbf{x}(t+\tau)\} \\ &= \mathbf{R}_x(\tau) \end{aligned} \tag{3.10}$$

Calculating the cross power spectrum is not an easy task because the resolution of $\frac{1}{T}$ is difficult if the stationary time of the signal is on the order of magnitude of the T or smaller.

The cross-power spectrum that we use is the average of a cross-power spectrum that diagonalizes for the source signals.

$$\begin{aligned} \overline{R}_x(\omega, t_k) &= \frac{1}{NT}\sum_{n=0}^{N-1}\mathbf{x}(\omega, t_k + nT)\mathbf{x}^H(\omega, t_k + nT) \\ k &= 1,\ 2,\ \ldots,\ K \end{aligned} \tag{3.11}$$

where N represents the number of intervals used to calculate each cross power spectrum matrix. N has to be sufficiently large so the result of Equation 3.8 can be modeled as a diagonal matrix.

Finally, to get independent conditions each time we calculate the CPSO, we choose the time intervals so they do not overlap: $t_k = kTN$. We can enforce this by choosing N such that it is equal to the total time over the size of the DTFT, T, and total number of of matrices to diagonalize, K. Thus

$$N = \frac{Total\ Number\ of\ Frames}{KT} \tag{3.12}$$

where the $Total\ Number\ of\ Frames$ accounts for $the\ Total\ Time$

Since we want N to be large we have to choose K and T to be as small as possible. However, we already know that K has to be greater than 2 and that T has to be as large as possible in order to estimate the linear convolution with a circular convolution. This is a trade off between the constraints.

If the observations were independent and we have a large enough N, the CPS matrix would be diagonal for each frequency ω and time t_k. The goal is then to estimate source signals whose CPS matrices are diagonal.

3.2 LEAST SQUARED COST FUNCTION

One way to measure independence is by calculating the Cross Power Spectrum of the signals. As mention in Subsection 1.1.3 the cross correlation or covariance does not assure independence in the stationary case.

Thus, we can use the difference between the cross power spectrum of the estimated signals and that of the independent sources as an error measurement.

$$\begin{aligned}
\mathbf{E}(\omega, t_k) &\equiv \mathbf{\Lambda}_{\bar{s}}(\omega, t_k) - \mathbf{\Lambda}_s(\omega, t_k) \\
\mathbf{E}(\omega, t_k) &= \mathbf{W}(\omega)\overline{\mathbf{R}}_x(\omega, t_k)\mathbf{W}^T(\omega) - \mathbf{\Lambda}_s(\omega, t_k) \qquad (3.13)
\end{aligned}$$

Since we do not know the sources, we cannot calculate $\mathbf{\Lambda}_s(\omega, t_k)$. However, we know that it is a diagonal matrix due to the independence assumption. Thus, $\mathbf{E}(\omega, t_k)$ is set to zero along the main diagonal.

Then we can define the cost function for the optimization problem as the sum of the squared error measurement, $\mathbf{E}(\omega, t_k)$ over all times t_k.

$$J(\omega) \quad = \quad \sum_{k=1}^{K} ||\mathbf{E}(\omega, t_k)||^2 \tag{3.14}$$

Our ultimate goal is to find the optimal filter $\mathbf{W}(\omega)$. Thus we compute the gradient of Equation 3.14 with respect of the filter,

$$\frac{\delta J(\omega)}{\delta \mathbf{W}^*(\omega)} \quad = \quad 2\sum_{k=1}^{K} \mathbf{E}(\omega, t_k)\mathbf{W}(\omega)\overline{\mathbf{R}}_x(\omega, t_k) \tag{3.15}$$

We optimize $J(\omega)$ using a Gradient Descent Algorithm (GDA). We choose an identity filter as the initial guess for the first lag of $\mathbf{W}(\omega)$.

Then , we calculate the new filter as

$$\mathbf{W}_{i+1}(\omega) \quad = \quad \mathbf{W}_i(\omega) + (\eta)\frac{\delta J}{\delta \mathbf{W}^*(\omega)} \tag{3.16}$$

where η is the learning rate of the algorithm.

The algorithm iterates using Equations 3.15 and 3.16 until convergence is achieved. The optimal filter is the one that diagonalizes $\mathbf{\Lambda}_{\overline{s}}(\omega, t_k)$ for every ω and t_k.

3.3 POWER NORMALIZATION

The mel-cepstrum power varies considerably across frequency. Thus, to improve the convergence of the BSS algorithm, we apply a power normalization to the partial derivative of J with respect of $\mathbf{W}$. Equation 3.17 shows this normalization.

$$\mathbf{m} = \left(\sum_{k=1}^{K} ||\overline{\mathbf{R}_x}(\omega, t_k)||^2 \right)^{-1} \tag{3.17}$$

$\mathbf{m}$ improves the performance of the BSS algorithm. Thus, the final equation for the partial derivative of J with respect of $\mathbf{W}$ is

$$\frac{\delta J}{\delta \mathbf{W}^*(\omega)} = 2\mathbf{m} \sum_{k=1}^{K} \mathbf{E}(\omega, t_k)\mathbf{W}(\omega)\mathbf{R}_x(\omega, t_k) \tag{3.18}$$

3.4 COMPUTATIONAL COST

The presented algorithm is relatively fast. We are able to process 40 hours of data in approximately 2 hours in machine time. The BSS algorithm is dominated by the computational cost of estimating the cross power spectrum of the observed data ($\mathrm{O}[KNd_xT(\log T + d_x)]$), the computation of the Least Squared Error Cost function($\mathrm{O}[KTd_xd_x(2d_s + d_x)]$) and the optimization of the filter($\mathrm{O}(d_xd_sKT\log T)$) [10].

3.5 SUMMARY

In summary, we have demixed the mel-warped cepstrum features with an FIR filter. We have done it by transforming the problem in Equation 5.1 into the frequency domain and solving simultaneously a separation problem for every frequency. This approach is similar to the instantaneous case in the time domain. We approximate this by using circular convolution instead of linear convolution.

The BSS algorithm has five parameters that constrain its performance:

1. T: Size of the Discrete Time Fourier Transform.
2. Q: Size of the filter.
3. K: Number of matrices to diagonalize
4. *Number of iterations - #*: in the GDA.
5. η: learning rate of the GDA.

The constraints on the algorithm based on the parameters are:

- The **Length of the Discrete Time Fourier Transform**, T, has to be much larger than the **Length of the Filter**, Q, in order to properly approximate linear convolution with circular convolution as shown in Equations 3.7 and 3.8 ($T \gg Q$).
- The **Number of Matrices to Diagonalize**, K, has to be equal or greater than 2 in order to have enough constraints to solve the equations as Equation 3.6 states. ($K \geq 2$).
- The **Number of Intervals** used to estimate each cross power spectrum of the observed data, N, must be sufficiently large to produce a robust estimate of the cross power spectrum of the estimate. Thus, according to Equation 3.12, T and K has to be as small. However, T has to be relatively greater than Q and K has to be greater than 2 as we just explained.

The FIR filter $W(\omega)$ also constrains the algorithm.

- The initial guess for the filter has to be an identity matrix for the first lag and zero matrices for the other lags.

- For a given mix source signal, the filter coefficient for the corresponding input signal has to be unity for the first lag and zero for the other lags, in order to maintain proper ordering of the estimated sources. This is similar to the previous constraint. This constraint guarantees that “independent” features from different speakers will have the same meaning and can be combined to generate the acoustic model.

4.0 TESTING OF ALGORITHM AND DISCUSSION

To test the effect of the proposed BSS algorithm on decorrelating mel-warped cepstrum features, we used 40 hours of Swicthboard training data, DEV01 test set and performed unadapted training and decoding. The baseline WER for this experiment is 44.99%. Table 4.1 below shows the WER and the final value of the cost function for various combinations of the BSS parameters. The best result we got was a WER equal to 43.80%.

Additional insight into the BSS algorithm may be obtained by examining the coefficients of the optimal filter. The FIR filter can be represented as a combination of 15 sub-filters (one for each estimated source). We use the 1^{st}, 3^{rd}, 6^{th}, 9^{th}, 12^{th} and 15^{th} sub-filters to illustrate our results. All the sub-filter graphs can be found in Appendix .

We present surf plots to get a general idea in 3 dimensions on how the sub-filters behave. Next we show waterfall plots of the magnitude of the sub-filters against time and mel-cepstrum features. Finally we show contour maps to get another view of how the sub-filters behave against time. However, before we analysis these graphs, we discuss the Cost Error Function **J** as a function of its parameters.

4.1 COST ERROR FUNCTION AS A FUNCTION OF THE BSS PARAMETERS

Some characteristics on the Cost Error Function, J were observed. These characteristics are organized according to parameters.

- **Length of the Filter.-** as the length of the filter Q, increases, J decreases. This behavior is not surprising. By increasing the number of free parameters, the cost function is expected to decrease.
- **Length of the DTFT.-** as the length of the DTFT T, increases the final result improves. This is because the circular convolution used in Equation 3.7 and 3.8 approximates better the linear convolution of Equation 5.1.
- **Number of Matrices to Diagonalize.-** as the number of matrices to diagonalize K increases, the error increases as well. This is because as K increases, the number of intervals used to estimate the cross power spectrum of the observation N decreases. As N decreases, the independence assumption gets weaker and the cross power spectrum of the sources is harder to diagonalize.
- **Learning Rate.-** as the learning rate η, increases, the error decreases. This is because the gradient descent method is able to take bigger steps between iterations. However if the learning rate is too large, the algorithm may wonder around the optimal value as it tries to converge.
- **Number of Iterations.-** as the number of iterations increases, the error decreases. This is because the algorithm has more trials to achieve the optimal filter. However, the computational time also increases. After a certain number of iterations, the improvement in the results is not worth the computational time.

Table 4.1 shows all the results we have obtained from the algorithm. We can see how J changes as a function of its parameters as mentioned above.

4.2 GENERAL FIR FILTER

For all the graphs we present, we use $T = 256$, $Q = 8$, $K = 2$, $\eta = 1.0$ and 8 iterations as the input parameters of the BSS algorithm. These set of parameters produce the lowest WER to our knowledge: 43.80% in our system. This is because a small K produces a large N so the cross power spectrum of the observation can be diagonalized as it is desired. Moreover, T

Table 4.1: BSS Parameters, WER and J

Experiment	T	Q	K	#	η	WER	J
Expt 1	128	32	5	8	1	44.58	109
Expt 2	128	8	5	20	1	44.47	96
Expt 3	128	8	5	8	1	44.32	125
Expt 4	512	8	5	8	1	44.48	287
Expt 5	64	8	5	8	1	44.11	88
Expt 6	64	2	5	8	1	44.24	116
Expt 7	64	4	5	8	1	44.41	101
Expt 8	64	8	5	12	1	44.48	76
Expt 9	64	8	5	16	0.7	44.64	78
Expt 10	64	16	5	8	1	44.75	80
Expt 11	64	8	5	8	2	44.29	67
Expt 12	64	8	5	16	2	44.37	52
Expt 13	64	8	6	8	1	44.44	102
Expt 14	64	8	6	8	1.4	44.28	90
Expt 15	64	8	5	8	1.045	44.47	87
Expt 16	64	8	3	8	1	44.22	56
Expt 17	512	8	3	8	1	44.13	177
Expt 18	512	128	3	8	1	44.23	131
Expt 19	512	8	4	8	1	44.27	232
Expt 20	512	8	2	8	1	44.32	119
Expt 21	256	8	2	8	1	43.80	80
Expt 22	256	8	4	8	1	44.43	158
Expt 23	1024	16	2	8	1	44.47	181

much greater than Q is a good constraint as well because it lets us approximate the circular convolution as it is mentioned in Chapter 3.

Figure 4.1 displays six sub-filters. Each sub-filter has dimensions d_x by Q. Thus, it gets convolves with all the mel-warped features as Figure 3.1 shows. We can see how each sub-filter emphasizes only one mel-cepstrum feature: the corresponding one. For example, the first sub-filter emphasizes the first feature of the mel-cepstra. The emphasis vector is identically to one at time t_1 and identically to zero for the other times.

However, the other $d_x - 1$ dimensions of the sub-filter are not equal to zero, even though Figure 4.1 might suggest it. To be able to zoom into the other dimensions, we have zero out $W_{i-i}(0)$ [1] which is identically to one as mentioned before.

Figure 4.2 shows the same as Figure 4.1 with $W_{i-i}(0) = 0$. This Figure gives us a better perspective of how all the features are convolved with each sub-filter.

From Figure 4.2 it is easy to see that the energy in the sub-filters is concentrated in a small area. We can also see how the significant sub-filter coefficients appear around the index of the estimated sources (1, 3, 6, 9, 12, and 15).

4.3 FIR FILTER ANALYZED VS LAG AND FEATURES

Even though we have been able to describe some of the characteristics of the sub-filters, the angle of the figures is not the most appropriate to analyze the behavior of the sub-filters against mel-cepstrum features and lag. We can gain additional insight by rotating Figure 4.2 to get a better view of how the sub-filter affects the different features of the mel-cepstra. Figure 4.3 shows more clearly how the sub-filters surrounding the corresponding feature have a greater magnitude than the ones further from it. This shows that neighboring features are more correlated than further features. Thus, the filter removes more redundancy from the neighboring features than from the ones that are further away.

[1] i corresponds to the number of the sub-filter being displayed

We also rotate Figure 4.2 in the opposite direction to see how the sub-filter coefficients appear against the filter lag. Figure 4.4 shows that the sub-filter coefficients get attenuated as the lag increases.

4.4 CONTOUR MAPS OF THE FIR FILTER

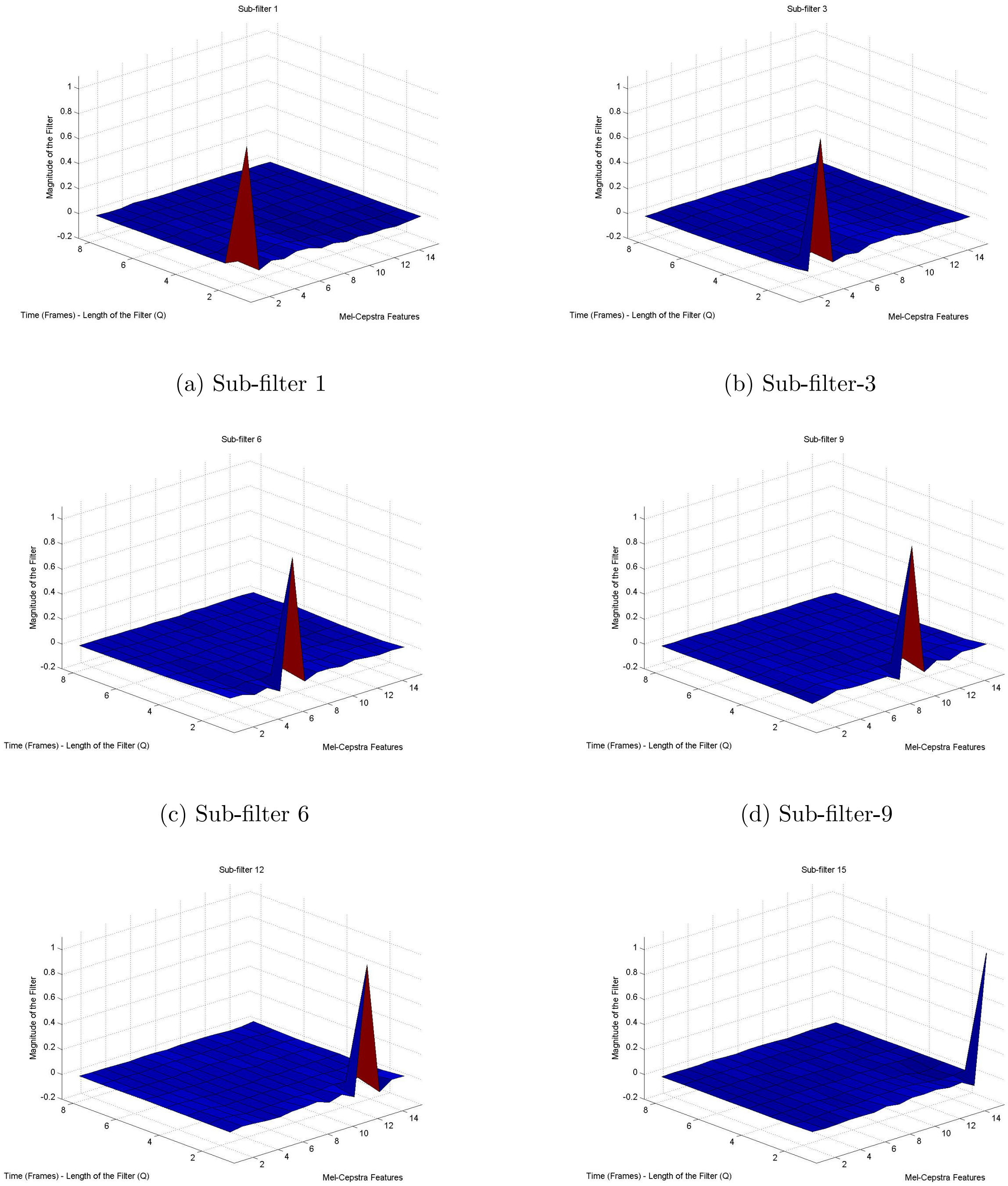
Figure 4.5 is a display of the contour maps of the sub-filters. It confirms again that the significant sub-filter coefficients appear around the index of the estimated features. We can also see in these pictures that most of the energy is at low times.

4.5 WORD ERROR RATE VS COST FUNCTION

Even though we want the Cost Function J to go zero, we have found that the WER does not necessarily decrease when J decreases. We can see this behavior in Figure 4.6. This figure plots the WER as a function of J. We can see that the lowest value of the WER does not correspond to the lowest value of the Cost Function. Moreover, we can see how the largest point of the Cost Function does not correspond to the Highest WER value. This can be due to the fact that the WER is not a measurement of independence but accuracy.

4.6 MISCELANEOUS

Figure 4.7 shows the results of summing the sub-filter coefficients along the time dimension and graphing them against the cepstral dimension for the first and second sub-filters. From the figure, it is easy to see the emphasis placed on neighboring cepstra as well as the surprising emphasis on the twelfth cepstrum.



(a) Sub-filter 1

(b) Sub-filter-3

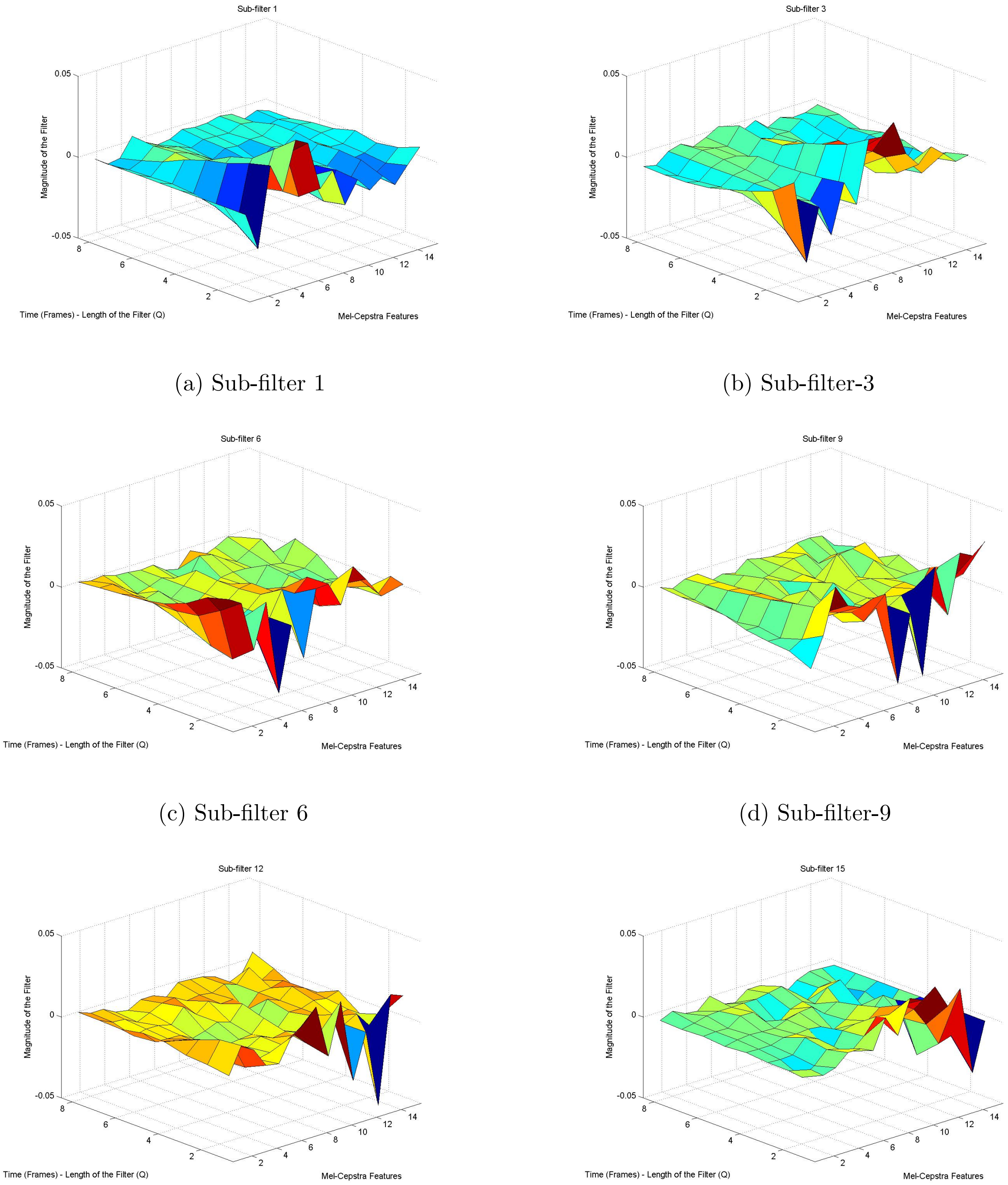
(c) Sub-filter 6

(d) Sub-filter-9

(e) Sub-filter 12

(f) Sub-filter-15

Figure 4.1: Surf Plot corresponding to 6 sub-filters. We can see an emphasis on the mel-cepstrum dimension corresponding to the sub-filter.



(a) Sub-filter 1

(b) Sub-filter-3

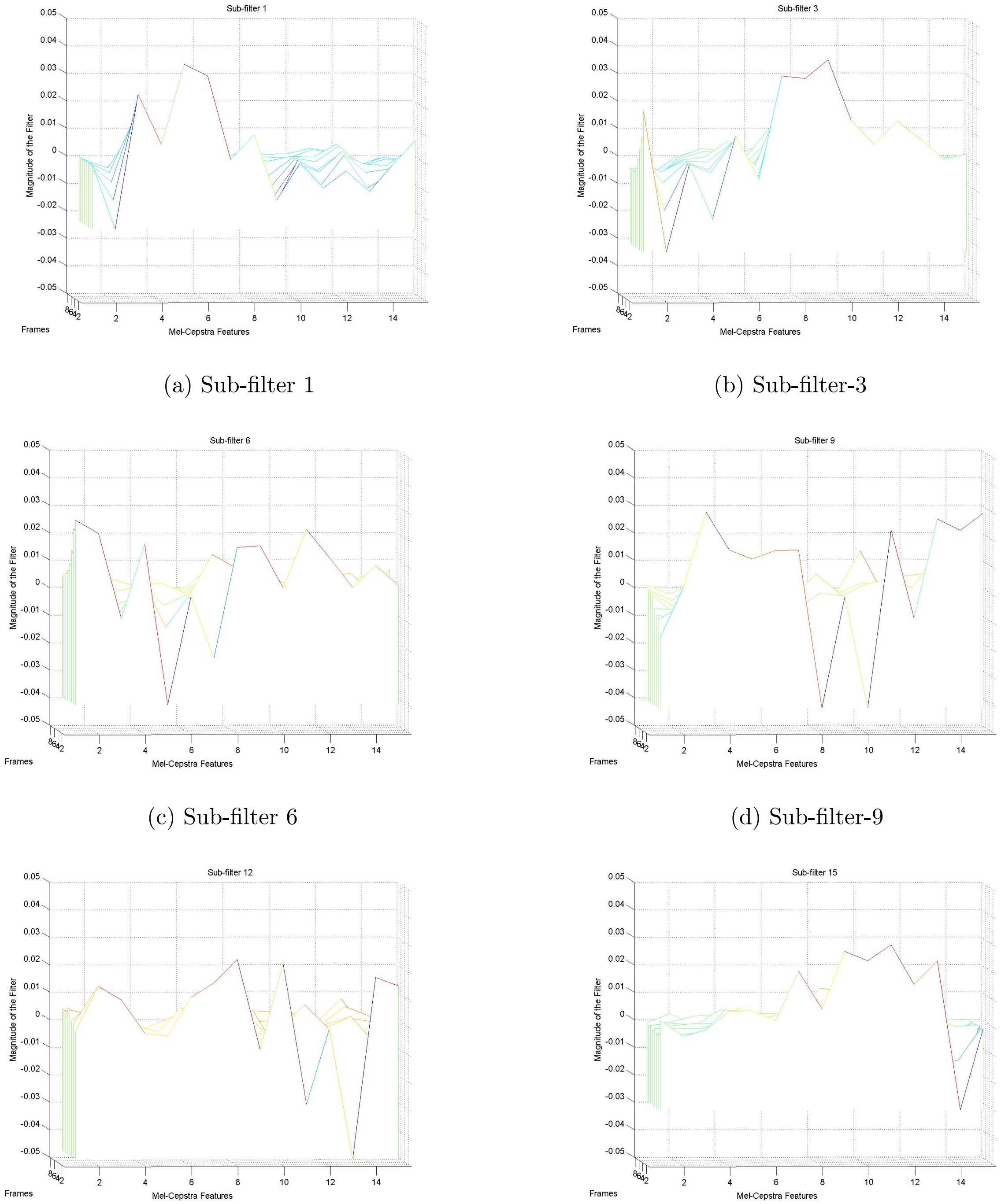
(c) Sub-filter 6

(d) Sub-filter-9

(e) Sub-filter 12

(f) Sub-filter-15

Figure 4.2: Surf Plot corresponding to 6 sub-filters without the emphasis. We have deleted the vector corresponding to the corresponding to the mel-cepstrum feature to get a better perspective of the other vectors.



(a) Sub-filter 1

(b) Sub-filter-3

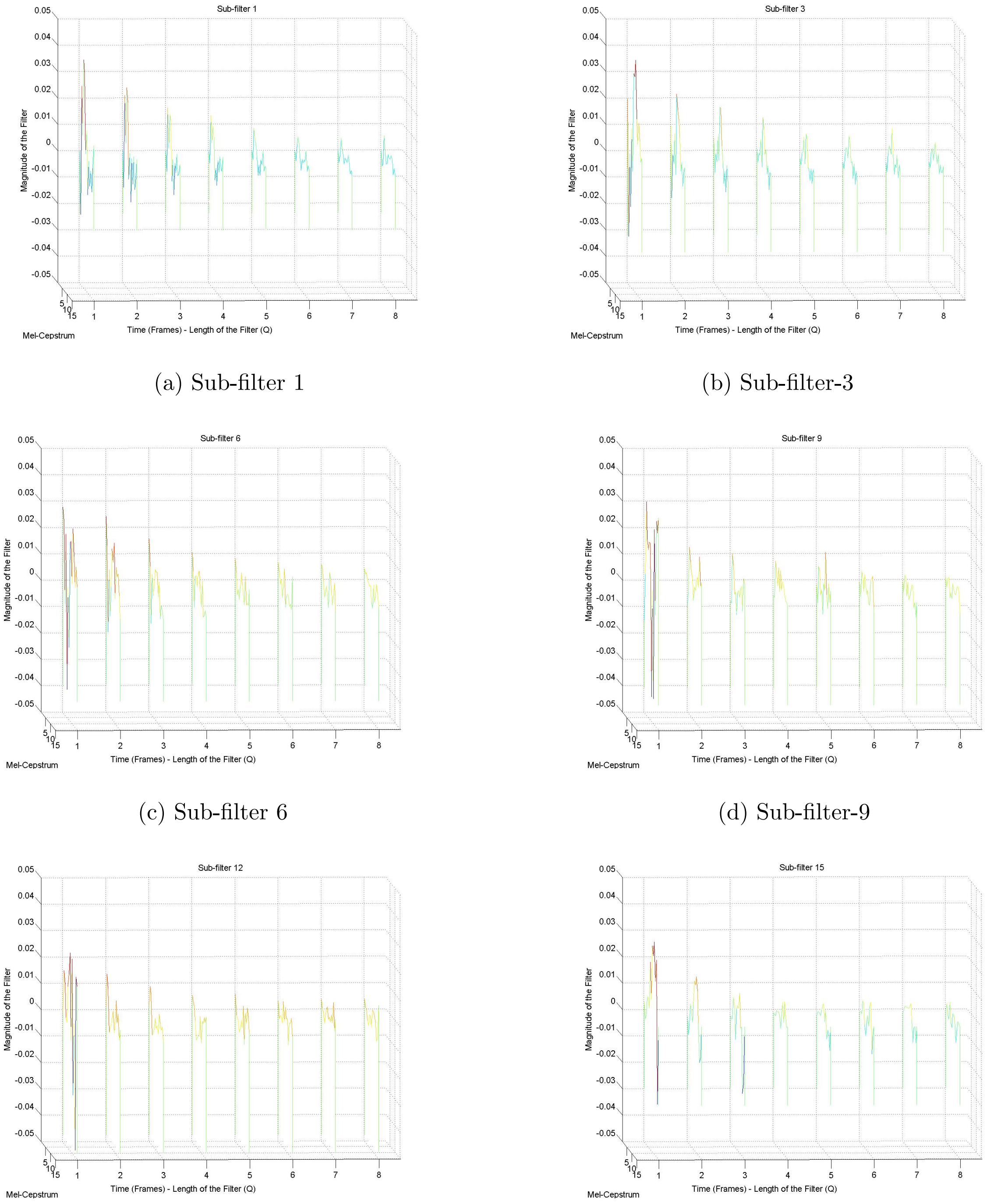
(c) Sub-filter 6

(d) Sub-filter-9

(e) Sub-filter 12

(f) Sub-filter-15

Figure 4.3: Waterfall plot of the sub-filters against mel-cepstra features.



(a) Sub-filter 1

(b) Sub-filter-3

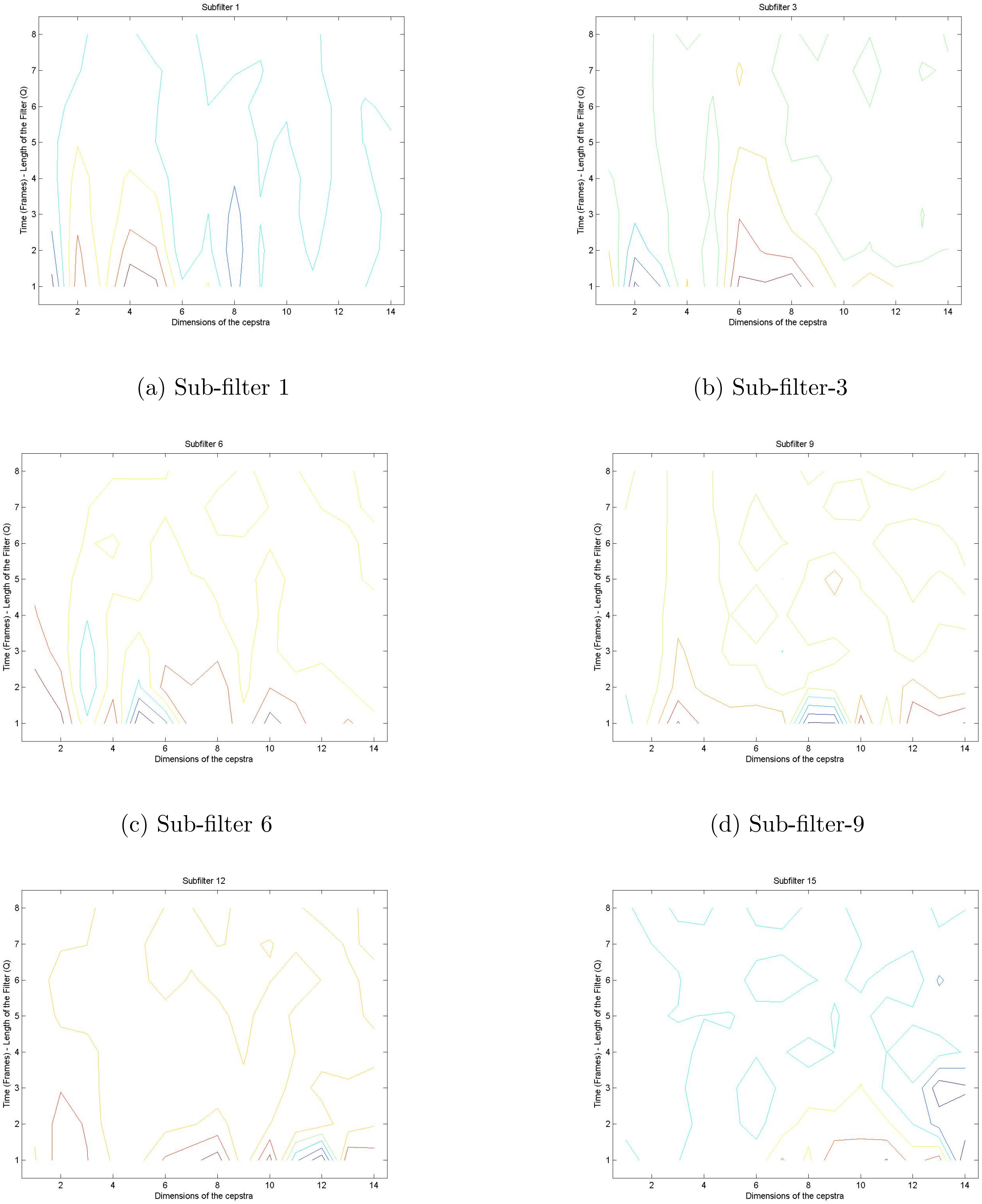
(c) Sub-filter 6

(d) Sub-filter-9

(e) Sub-filter 12

(f) Sub-filter-15

Figure 4.4: Waterfall plot of the sub-filters against time.



(a) Sub-filter 1

(b) Sub-filter-3

(c) Sub-filter 6

(d) Sub-filter-9

(e) Sub-filter 12

(f) Sub-filter-15

Figure 4.5: Contour plot of the sub-filters.

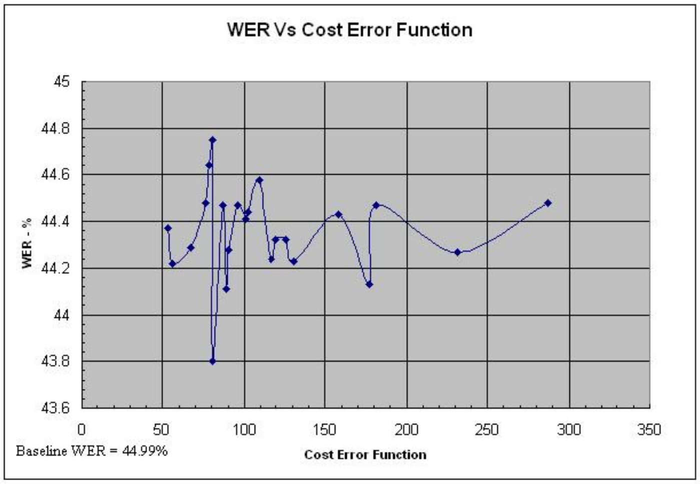


Figure 4.6: Word Error Rate VS Cost error function, J

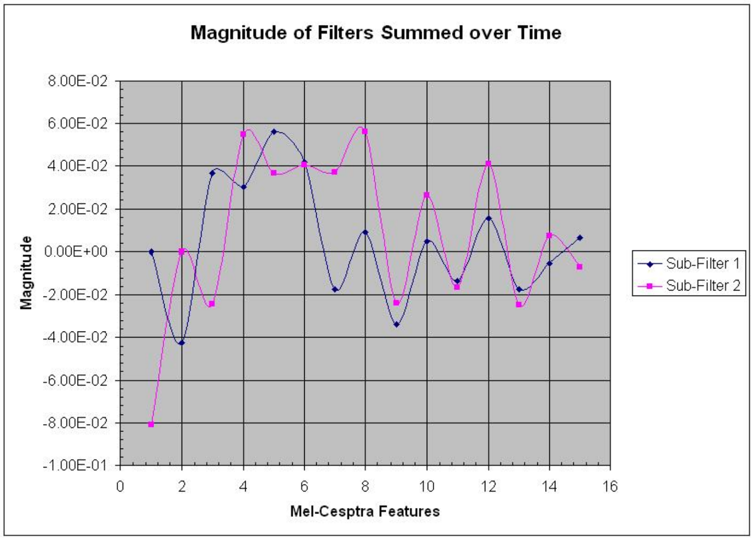


Figure 4.7: Summed Magnitude of the sub-Filter 1 and Sub-Filter 2 Coefficients over Time.

5.0 ENHANCEMENT OF THE ALGORITHM

In order to improve the performance of the FIR filter we have tried some modifications of the original algorithm. We made three changes:

- Separating the mel-warped features coming from each speaker into multiple lag sections and analyzing each section with a different filter; i.e., having multiple filters per speaker instead of 1.
- Using an anticausal filter instead of a causal filter.
- Using an IIR filter design instead of an FIR filter.

In this chapter we discuss these approaches and present the results we have obtained. The IIR filter approach minimizes the Cost Error function. The results obtained with the other two approaches are not as encouraging as the IIR results.

5.1 MULTIPLE FILTERS

We separated the speaker features coming into the BSS algorithm from each speaker into equally distributed blocks. This seems reasonable because the data is nonstationary and can vary substantially from the first to the last block. Thus, we calculate an FIR filter for each block. The WER's obtained were better than the baseline WER but not as good as when we keep all the information together.

Some possible reasons for this results are:

- The constraints that we summarize in Section 3.5 are not met. For example, since we are breaking the features into different blocks, we have fewer frames to estimate each

Table 5.1: BSS Parameters and WER for using multiple filters

Experiment	T	Q	K	#	η	m	WER
Expt 1	64	8	5	8	1	3	44.58
Expt 2	256	8	2	8	1	2	44.39

cross power spectrum of the observation. Thus, the number of intervals used to estimate each cross power spectrum matrix, N, might not be big enough to model the cross power spectrum of the estimation as diagonal

- Another reason is that the data should not be equally distributed into the lags. Probably we are separating the data at points that it should not be separated.

The results we have obtained are shown in Table 5.1. The number of filters used is represented as m.

These results reaffirm what we said before. The smaller K is and the bigger the difference between T and Q is, the better WER we obtained. However, if we compare these results with the original case, we can see that the WER is better in both cases: 44.11% and 43.80%.

5.2 FILTER WITH CAUSAL AND ANTICAUSAL COMPONENTS

The filter we used to decorrelate the mel cepstrum features is causal. This means that it only uses current and past values of the input. However, in speech we can predict some of the future inputs. Thus, it is possible to implement the FIR filter using the current, past and future values of the observation. Thus, the backward model equation changes to

$$\overline{\mathbf{s}}(t) = \sum_{\tau=-P}^{Q} \mathbf{W}(\tau)\mathbf{x}(t-\tau) \tag{5.1}$$

where P is the length of the anticausal part of the filter. Thus, the total length of the filter is

$$Length\ of\ the\ filter = P + 1 + Q \tag{5.2}$$

The 1 in the equation represents the 0 position.

To implement this new filter we can derive a new set of equations and write a new algorithm. However, we can also reuse the causal algorithm we used to decorrelate the mel cepstrum features. We can just advance the filter P positions when calculating it. However, we have to enforce that the P^{th} position of the new filter corresponds to the 0 position of the anticausal filter. This means that in this position we should have a diagonal fill with ones. Moreover, we have to keep enforcing that we have zeros in all the other diagonals. These conditions enforce the independence requirement. Finally, when convolving the new filter with the features, we have to discard the first P results because they correspond to noise

We expect this algorithm to perform better than the original BSS algorithm because the length of the filter is longer. However, we might need to use a longer FFT so we do not violate algorithm constraints.

This idea is still in a developing stage, thus we have not implemented it.

5.3 IIR FILTER

The BSS algorithm that we have implemented corresponds to an FIR filter, $\mathbf{W}(\tau)$. Moreover, the channel effect $\mathbf{A}(\tau)$ that we are trying to undo is also assumed to be an FIR filter. Thus, we are trying to undo an FIR filter with another FIR filter. This cancellation gives us our estimate $\overline{\mathbf{s}}(t)$ to equal the original signal $\mathbf{s(t)}$ as Parra and Spence propose [10]:

$$\bar{\mathbf{s}}(t) \quad = \quad \sum_{\tau=0}^{Q} \mathbf{W}(\tau)\mathbf{x}(t-\tau) \tag{5.3}$$

where $\mathbf{W}(\tau)$ is the FIR filter that decorrelates the sources and

$$\mathbf{x}(t) = \sum_{\tau=0}^{P} \mathbf{A}(\tau)\mathbf{s}(t-\tau) \tag{5.4}$$

Even though this approach has demonstrated good performance, it is not intuitive in the sense that we use an FIR filter to undo another FIR filter. It would be more intuitive to eliminate the mixing of the channel with an IIR filter approach as:

$$\bar{\mathbf{s}}(t) \quad = \quad \sum_{\tau=0}^{Q} \mathbf{W}(\tau)\mathbf{x}[t-\tau] + \sum_{\tau=1}^{P} \mathbf{V}(\tau)\bar{\mathbf{s}}[t-\tau] \tag{5.5}$$

Thus, we will keep our FIR channel assumption and we will apply the BSS algorithm using an IIR filter. To approximate the filter, we follow a similar approach as Parra and Spence suggest. We start with the simplest IIR filter we can use to invert the FIR filter and then generalize the results to the convolutive case at multiple frequencies.

The general IIR filter takes into consideration the current and past values of the input and the previous values of the output in order to calculate the current output. Initially, the IIR filter that we use to undo the effect of the instantaneous channel only considers the current input and the previous output. Thus,

$$\bar{\mathbf{s}}(t) \quad = \quad \mathbf{W}\mathbf{x}(t) + \mathbf{V}\bar{\mathbf{s}}[t-1] \tag{5.6}$$

where

$$\mathbf{x}(t) \quad = \quad \mathbf{A}\mathbf{s}(t) \tag{5.7}$$

To find the cross-correlation of the observation we follow a similar approach as the one discuss in Chapter 3. Then, the cross-correlation of the observation is given by

$$\mathbf{R}_x(t) \quad = \quad \mathbf{A}\mathbf{\Lambda}_{\mathbf{s}}(t)\mathbf{A}^T \tag{5.8}$$

On the other hand, the cross-correlation of the estimated signal can be calculated as

$$\begin{aligned}
\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t) &= <\overline{\mathbf{s}}(t)\ \overline{\mathbf{s}}^T(t)> \\
&= <\mathbf{W}\mathbf{x}(t)+\mathbf{V}\overline{\mathbf{s}}(t-1)\ [\mathbf{W}\mathbf{x}(t)+\mathbf{V}\overline{\mathbf{s}}(t-1)]^T> \\
&= \mathbf{W}\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{W}^T+\mathbf{W}\mathbf{x}(t)\overline{\mathbf{s}}^T(t-1)\mathbf{V}^T+\mathbf{V}\overline{\mathbf{s}}(t-1)\mathbf{x}^T(t)\mathbf{W}^T+\mathbf{V}\overline{\mathbf{s}}(t-1)\overline{\mathbf{s}}^T(t-1)\mathbf{V}^T \\
&= \mathbf{W}\mathbf{R}_{\mathbf{x}}(t)\mathbf{W}^T+\mathbf{W}\mathbf{x}(t)\overline{\mathbf{s}}^T(t-1)\mathbf{V}^T+\mathbf{V}\overline{\mathbf{s}}(t-1)\mathbf{x}^T(t)\mathbf{W}^T+\mathbf{V}\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t-1)\mathbf{V}^T \\
&= \mathbf{W}\mathbf{R}_{\mathbf{x}}(t)\mathbf{W}^T+\mathbf{W}\mathbf{A}\mathbf{s}(t)\overline{\mathbf{s}}^T(t-1)\mathbf{V}^T+\mathbf{V}\overline{\mathbf{s}}(t-1)\mathbf{s}^T(t)\mathbf{A}^T\mathbf{W}^T+\mathbf{V}\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t)\mathbf{V}^T
\end{aligned}$$

There is no reason to believe that the estimation of the signal and the actual signal are related at different lags. Thus

$$\begin{aligned}
\mathbf{s}(t)\overline{\mathbf{s}}^T(t-1) &= 0 \\
\overline{\mathbf{s}}(t-1)\mathbf{s}^T(t) &= 0
\end{aligned}$$

Finally, the cross-correlation of the estimated signal simplifies to

$$\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t) \quad = \quad \mathbf{W}\mathbf{R}_{\mathbf{x}}(t)\mathbf{W}^T+\mathbf{V}\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t)\mathbf{V}^T$$

This can be rewritten in the form

$$\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t)-\mathbf{V}\mathbf{\Lambda}_{\overline{\mathbf{s}}}(t)\mathbf{V}^T \quad = \quad \mathbf{W}\mathbf{R}_{\mathbf{x}}(t)\mathbf{W}^T \tag{5.9}$$

Equation 5.9 looks similar to the Lyapunov equation:

$$\mathbf{M}-\mathbf{A}\mathbf{M}\mathbf{A}^T \quad = \quad \mathbf{C} \tag{5.10}$$

The solution for the Lyapunov equation is given by [7]:

$$\mathbf{M} = \sum_{m=0}^{\infty} (\mathbf{A}^T)^m \mathbf{N} \mathbf{A}^m \tag{5.11}$$

The only assumption of this result is that the magnitudes of all the eigenvalues of $\mathbf{A}$ are less than one. This is a reasonable assumption in our case because the deviations that the backward path add to the system are minimum.

Thus, the cross-correlation of our observation is:

$$\mathbf{\Lambda}_{\bar{\mathbf{s}}}(t) = \sum_{m=0}^{\infty} \mathbf{V}^m \mathbf{W} \mathbf{R}_{\mathbf{x}}(t) \mathbf{W}^T (\mathbf{V}^T)^m \tag{5.12}$$

Using the same justification as when deriving the FIR BSS algorithm, we can take our result one step further into the convolutive case assuming a FFT much longer than the size of the filters $\mathbf{W}$ and $\mathbf{V}$ and solve the problem in the frequency domain at multiple lags. Our error measurement is now

$$\begin{aligned} \mathbf{E}(\omega, t_k) &\equiv \mathbf{\Lambda}_{\bar{s}}(\omega, t_k) - \mathbf{\Lambda}_s(\omega, t_k) \\ \mathbf{E}(\omega, t_k) &= \mathbf{W}(\omega)\mathbf{R}_{\mathbf{x}}(\omega, t_k)\mathbf{W}^T(\omega) + \mathbf{V}(\omega)\mathbf{\Lambda}_{\bar{\mathbf{s}}}(\omega, t_k)\mathbf{V}^T(\omega) - \mathbf{\Lambda}_s(\omega, t_k) \end{aligned} \tag{5.13}$$

The cost function remains as

$$J(\omega) = \sum_{k=1}^{K} ||\mathbf{E}(\omega, t_k)||^2 \tag{5.14}$$

Now to calculate both optimal filters $\mathbf{W}$ and $\mathbf{V}$, we have to compute the gradient of the cost function with respect of the filters,

$$\frac{\delta J(\omega)}{\delta \mathbf{W}^*(\omega)} = 2\sum_{k=1}^{K} \mathbf{E}(\omega, t_k)\mathbf{W}(\omega)\overline{\mathbf{R}}_x(\omega, t_k) \tag{5.15}$$

$$\frac{\delta J(\omega)}{\delta \mathbf{V}^*(\omega)} = 2\sum_{k=1}^{K} \mathbf{E}(\omega, t_k)\mathbf{V}(\omega)\mathbf{\Lambda}_{\bar{\mathbf{s}}}(\omega, t_k) \tag{5.16}$$

where $\overline{\mathbf{R}}_x(\omega, t_k)$ is the same cross power spectrum of the observed signal as we use to calculate the FIR model:

$$\begin{aligned} \overline{R}_x(\omega, t_k) &= \frac{1}{NT}\sum_{n=0}^{N-1} \mathbf{x}(\omega, t_k + nT)\mathbf{x}^H(\omega, t_k + nT) \\ k &= 1, \ ,2, \ ..., \ ,K \end{aligned} \tag{5.17}$$

We optimize the cost function in the same manner as before: using a Gradient Descent Method.

Finally, we have to normalize the backward filter $\mathbf{V}$ using the cross power spectrum of the estimated signal in the same way we normalize the forward filter $\mathbf{W}$. Otherwise, the estimation is not accurate.

We have not implemented this function in C so we do not have the WER of the speech recognition system after using it. However, we have implemented it in Matlab and obtained some encouraging results. The Cost Error function of the algorithm gets minimized. This is an indication that a further level of independence is obtained by the mel-warped features.

6.0 CONCLUSIONS

In this work, we applied a BSS algorithm to the decorrelation of mel-warped cepstra for use as acoustic features in speech recognition. The BSS algorithm is based on a backward model approach suggested by Parra and Spence [10] in 2000. The algorithm produces an optimal FIR filter in the frequency that decorrelates the mel-warped cepstrum features at different lags.

The algorithm generates a set of independent features that minimize a cross spectral distance measure. The algorithm produces an improvement in WER for many combinations of the algorithm parameters. The optimal filter coefficients demonstrate the interdependence among the cepstra.

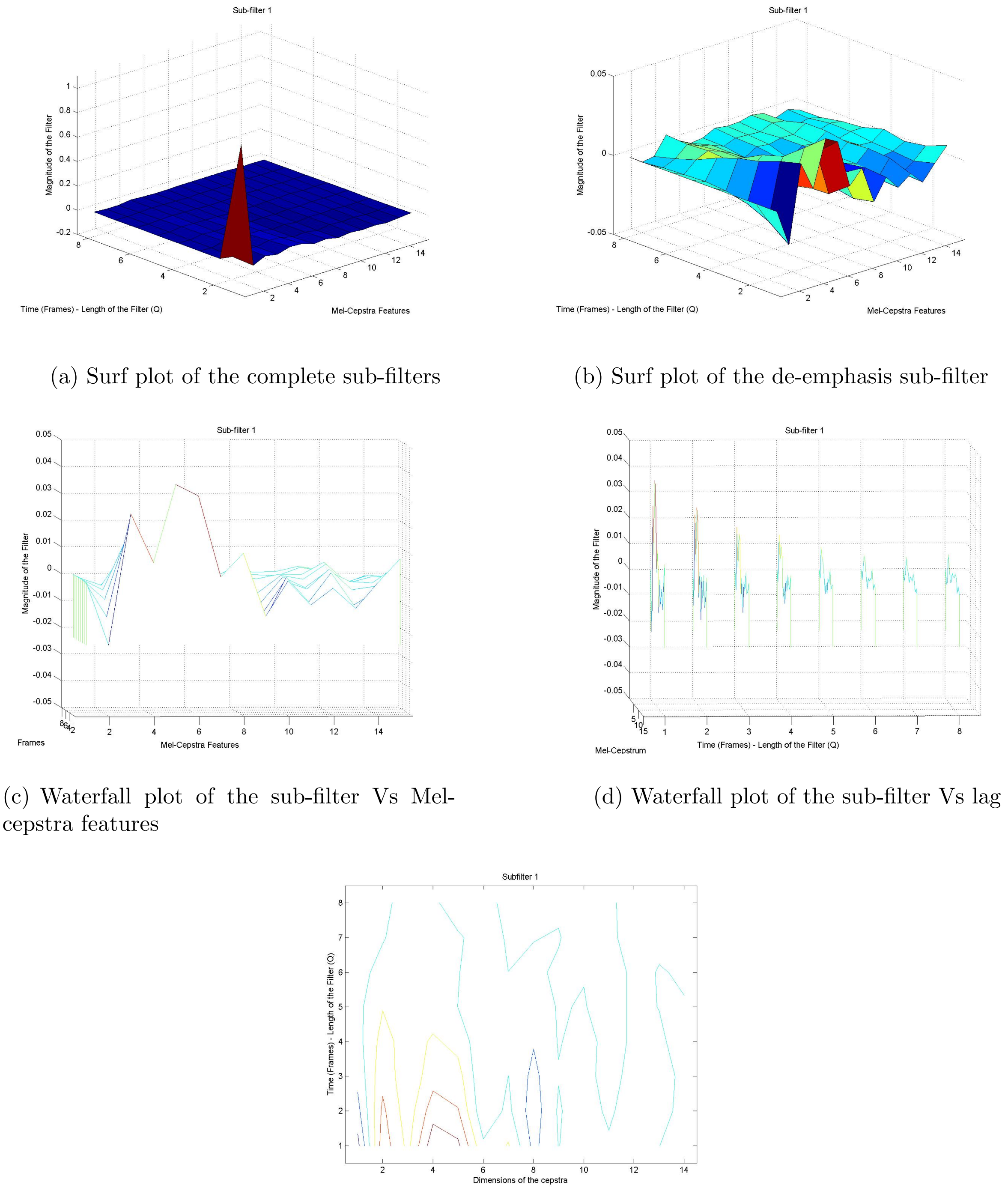
The best WER we obtained in the EARS system was 43.80%. This is 1.2% better than the baseline WER. This is a significant improvement based on the computational time added to the system. A lower WER is expected if more hours of training data were available.

Three different types of enhancements have been tried on the algorithm. None of the enhancements produced any further improvement on the system. However, the results obtained were always better than the baseline results.

APPENDIX

GRAPHS

In this appendix we collect the graphs for every sub-filter. The parameters chosen for the filter are the ones corresponding to our best result: WER equal to 43.80%.



(a) Surf plot of the complete sub-filters

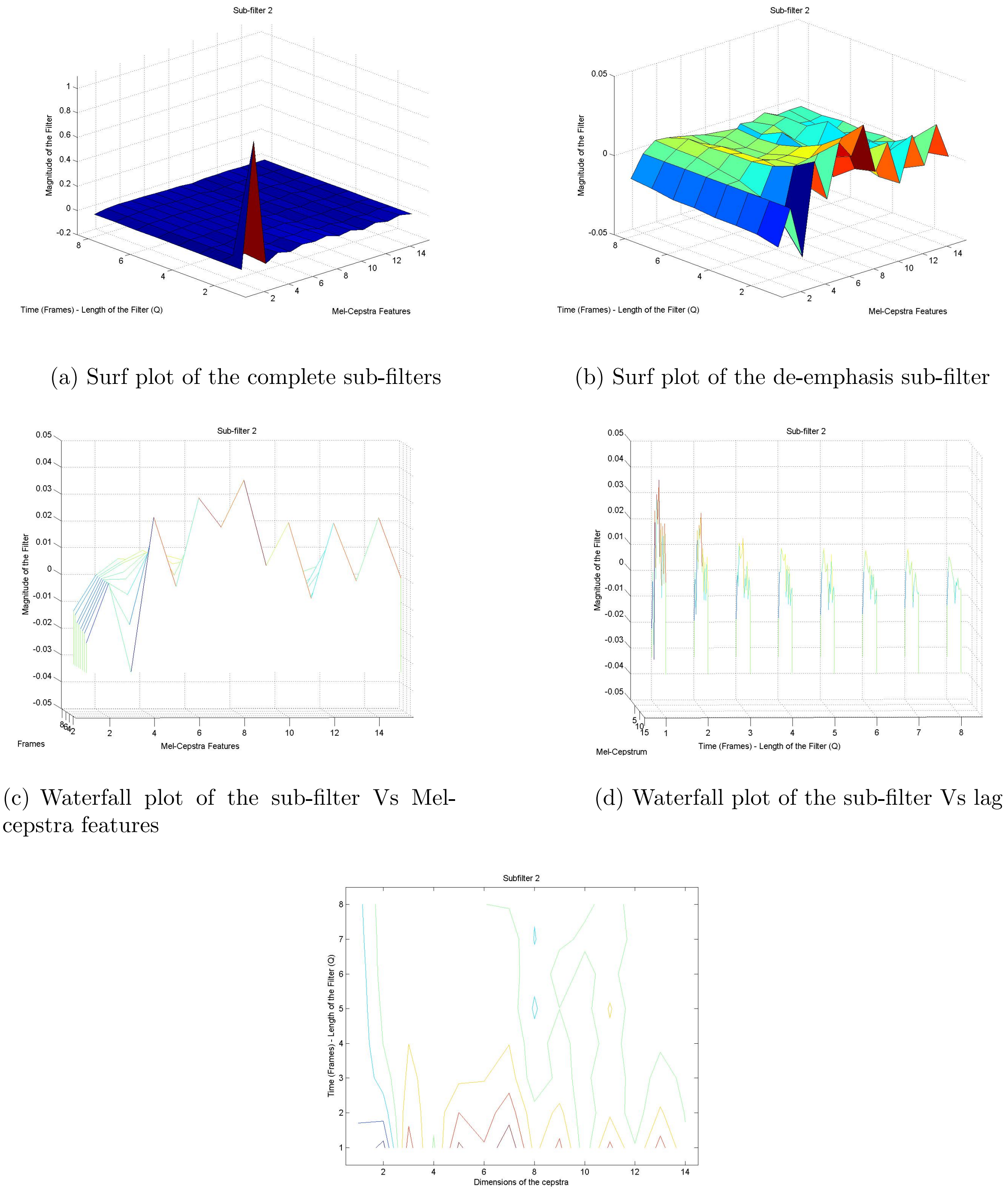
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A1: Sub-filter 1



(a) Surf plot of the complete sub-filters

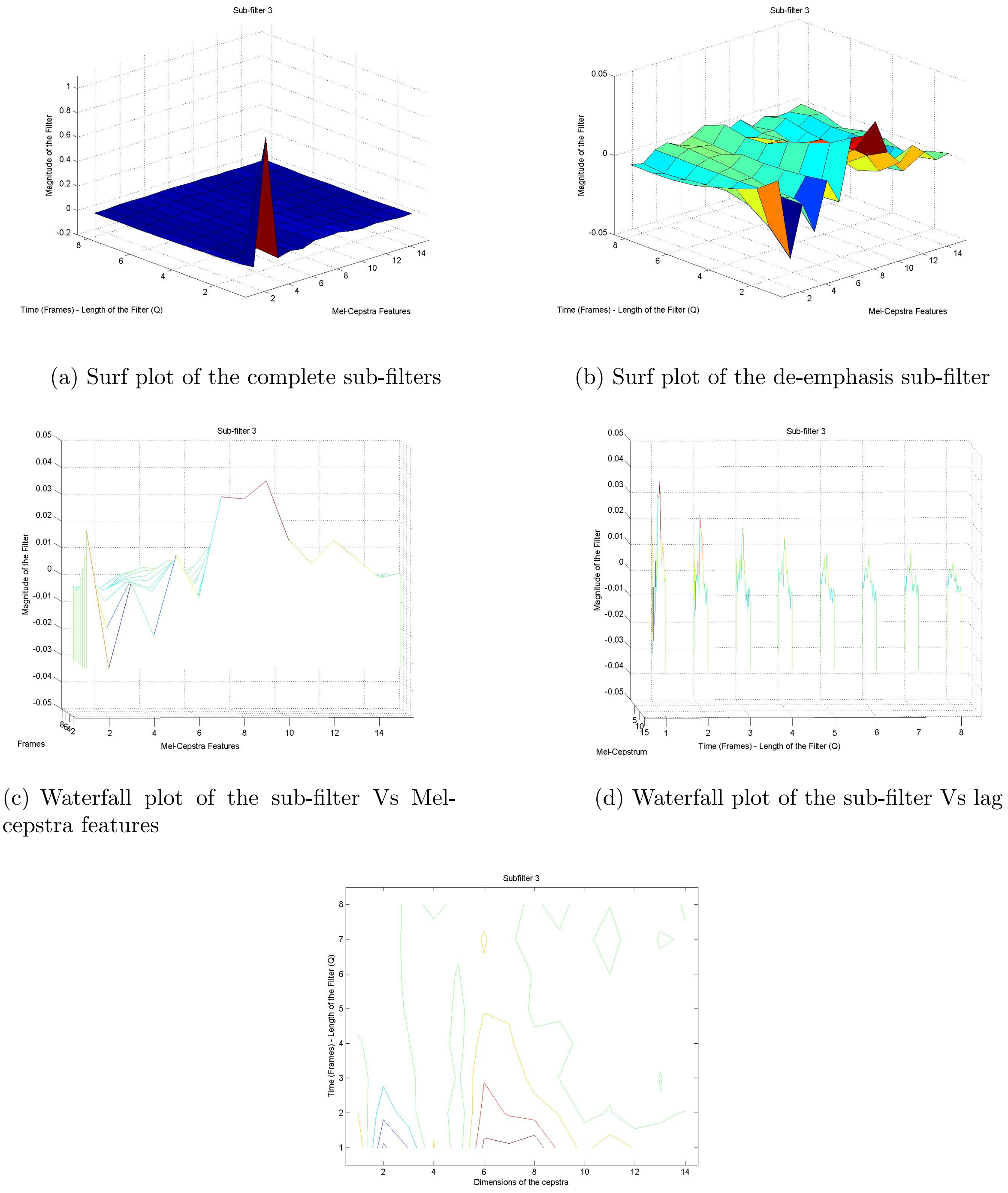
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A2: Sub-filter 2



(a) Surf plot of the complete sub-filters

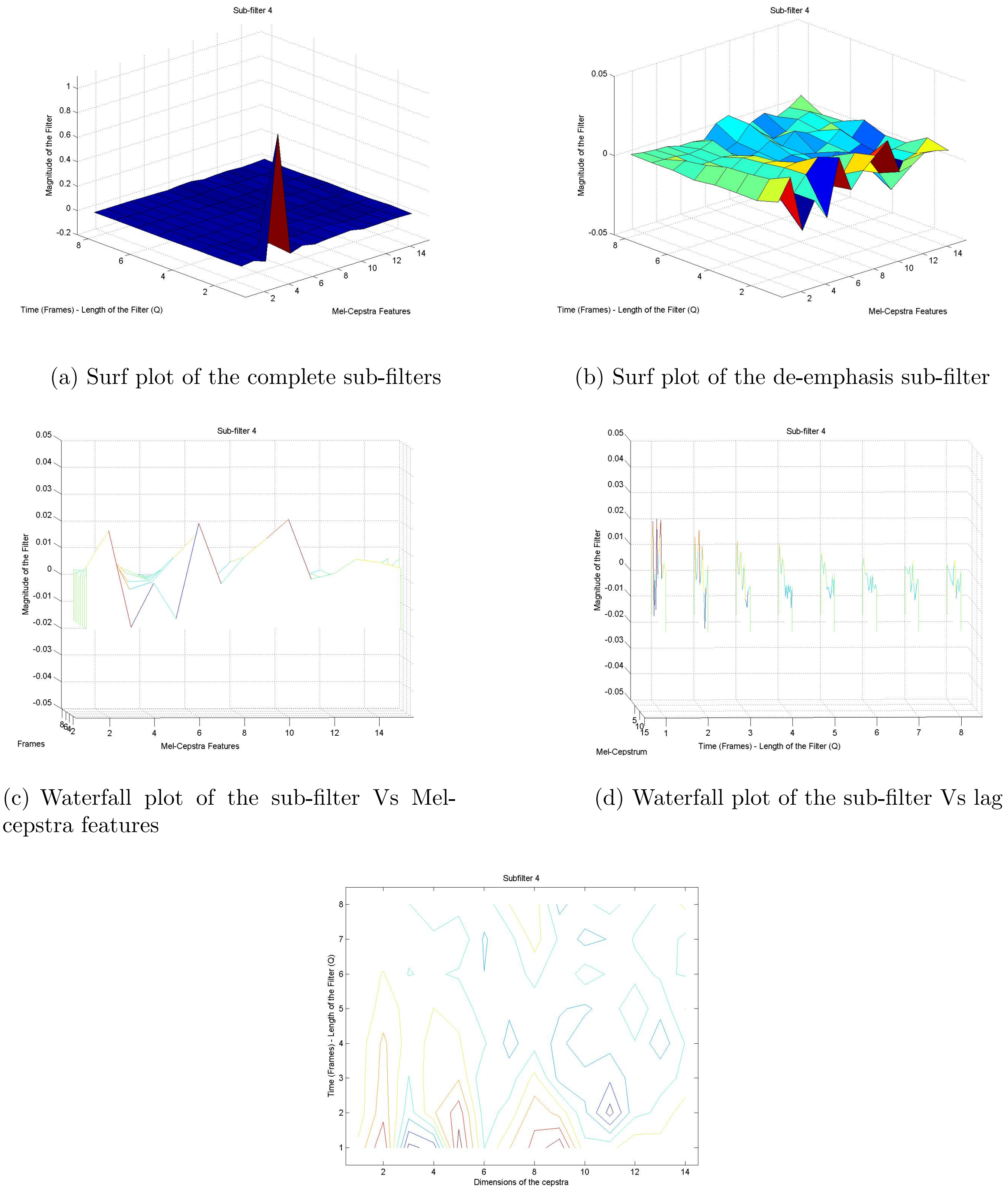
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A3: Sub-filter 3



(a) Surf plot of the complete sub-filters

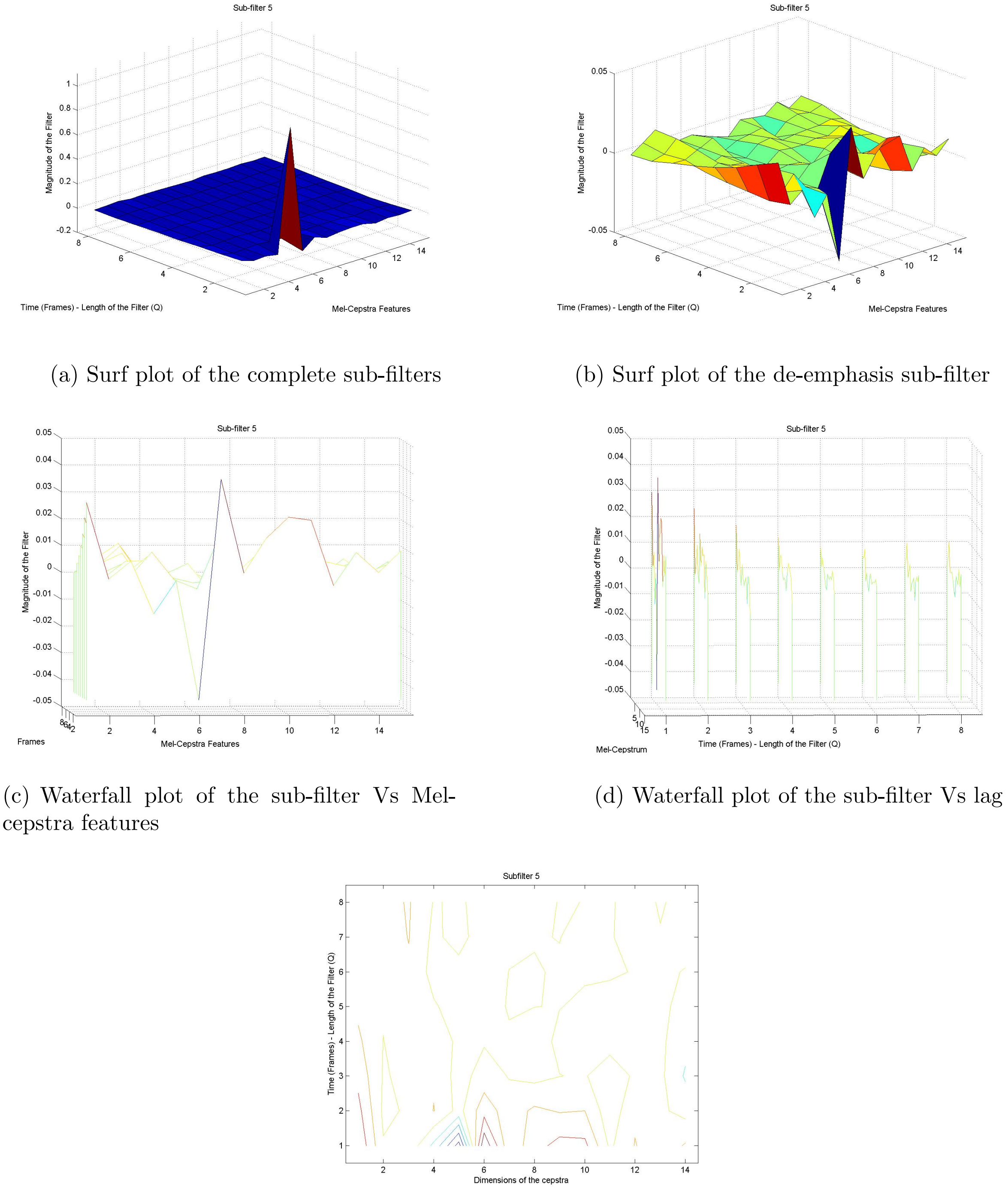
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A4: Sub-filter 4



(a) Surf plot of the complete sub-filters

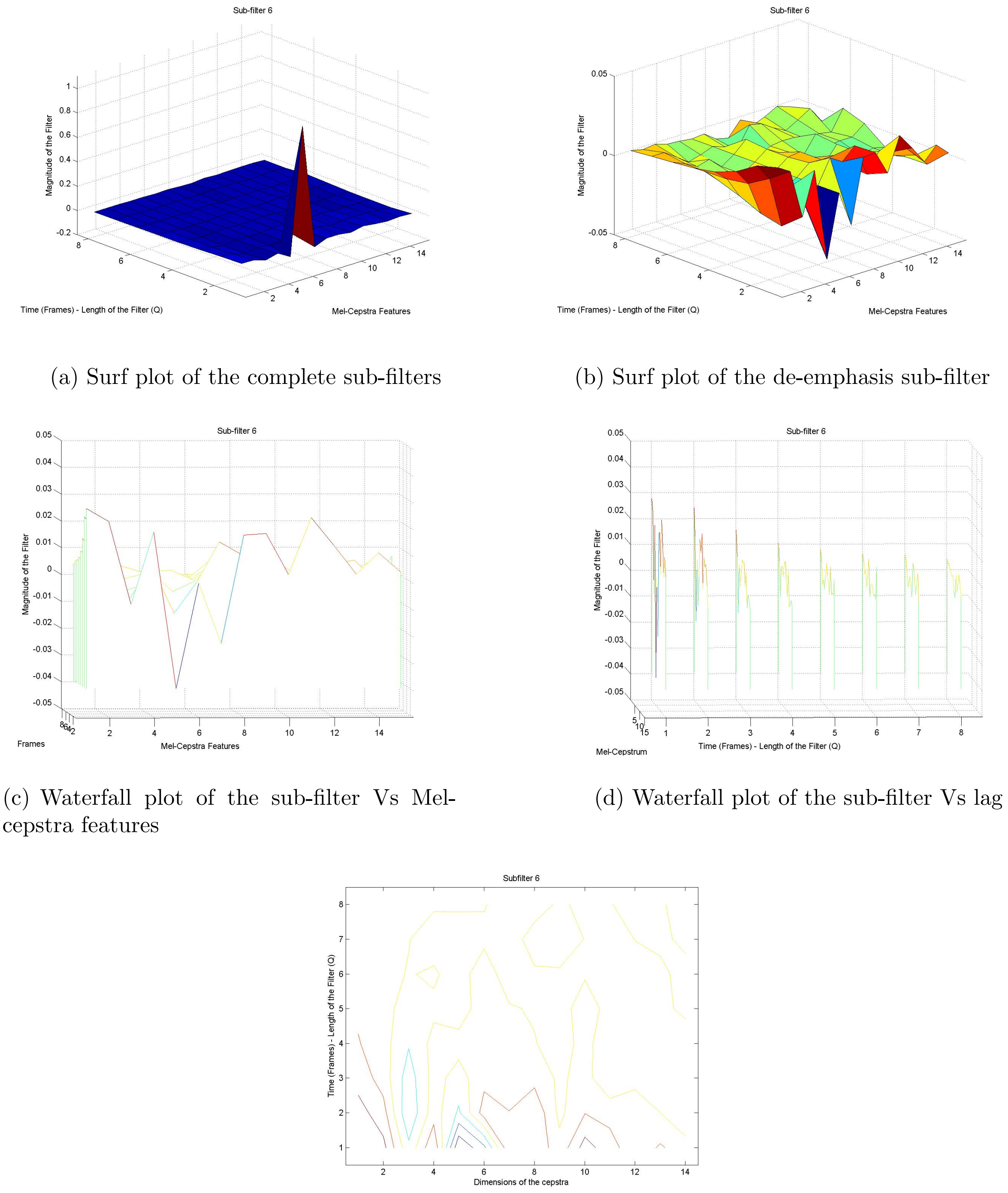
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A5: Sub-filter 5



(a) Surf plot of the complete sub-filters

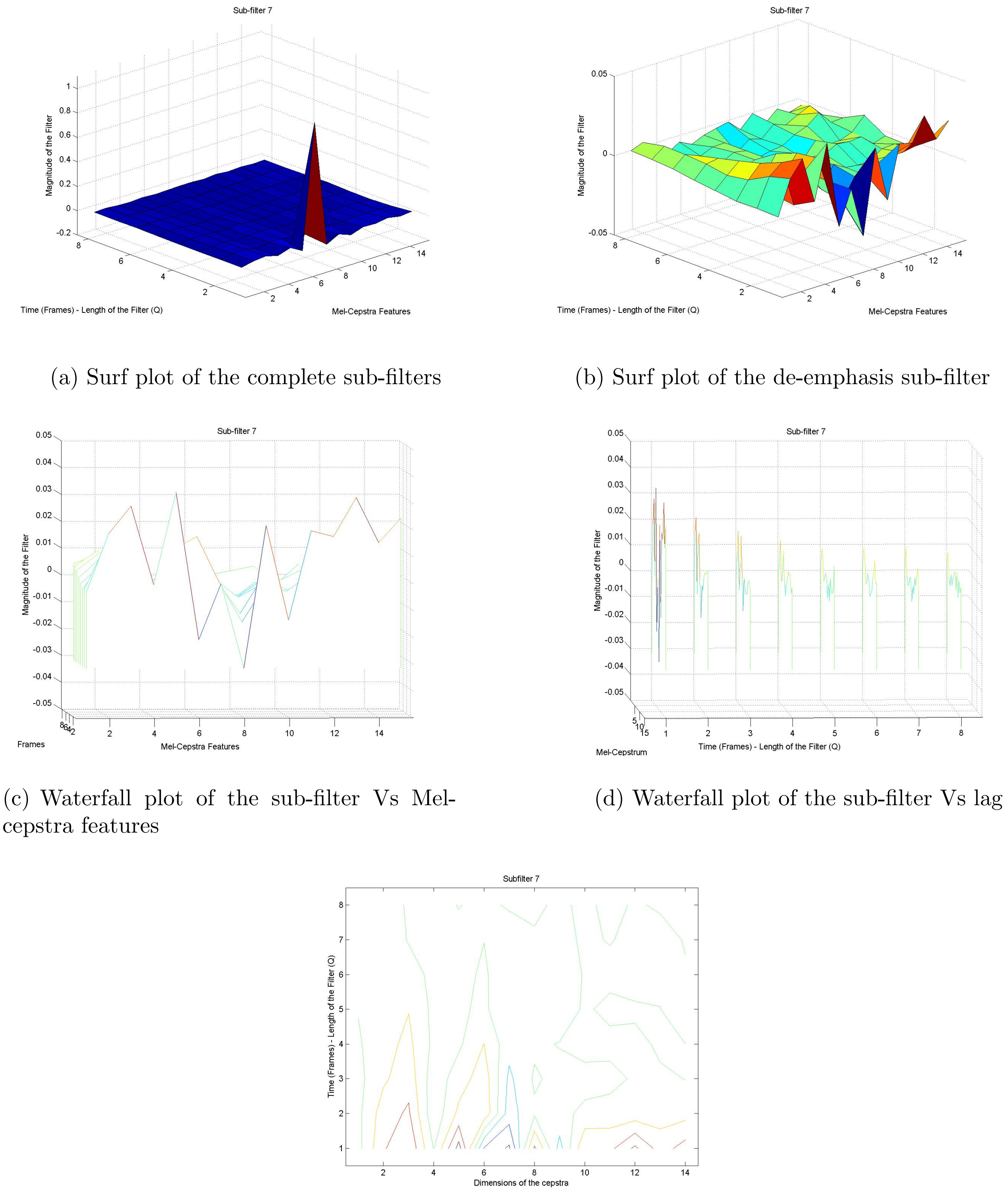
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A6: Sub-filter 6



(a) Surf plot of the complete sub-filters

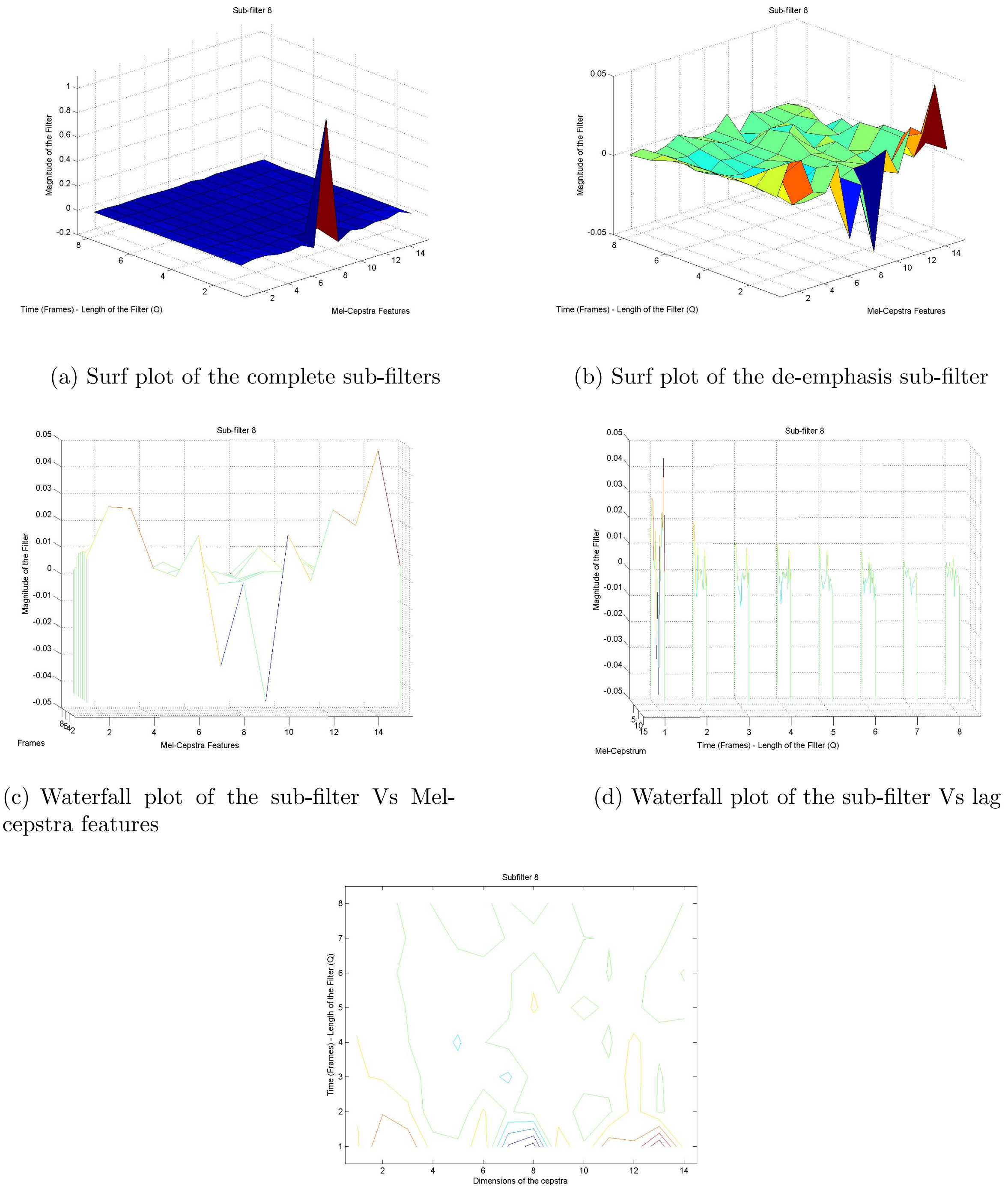
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A7: Sub-filter 7



(a) Surf plot of the complete sub-filters

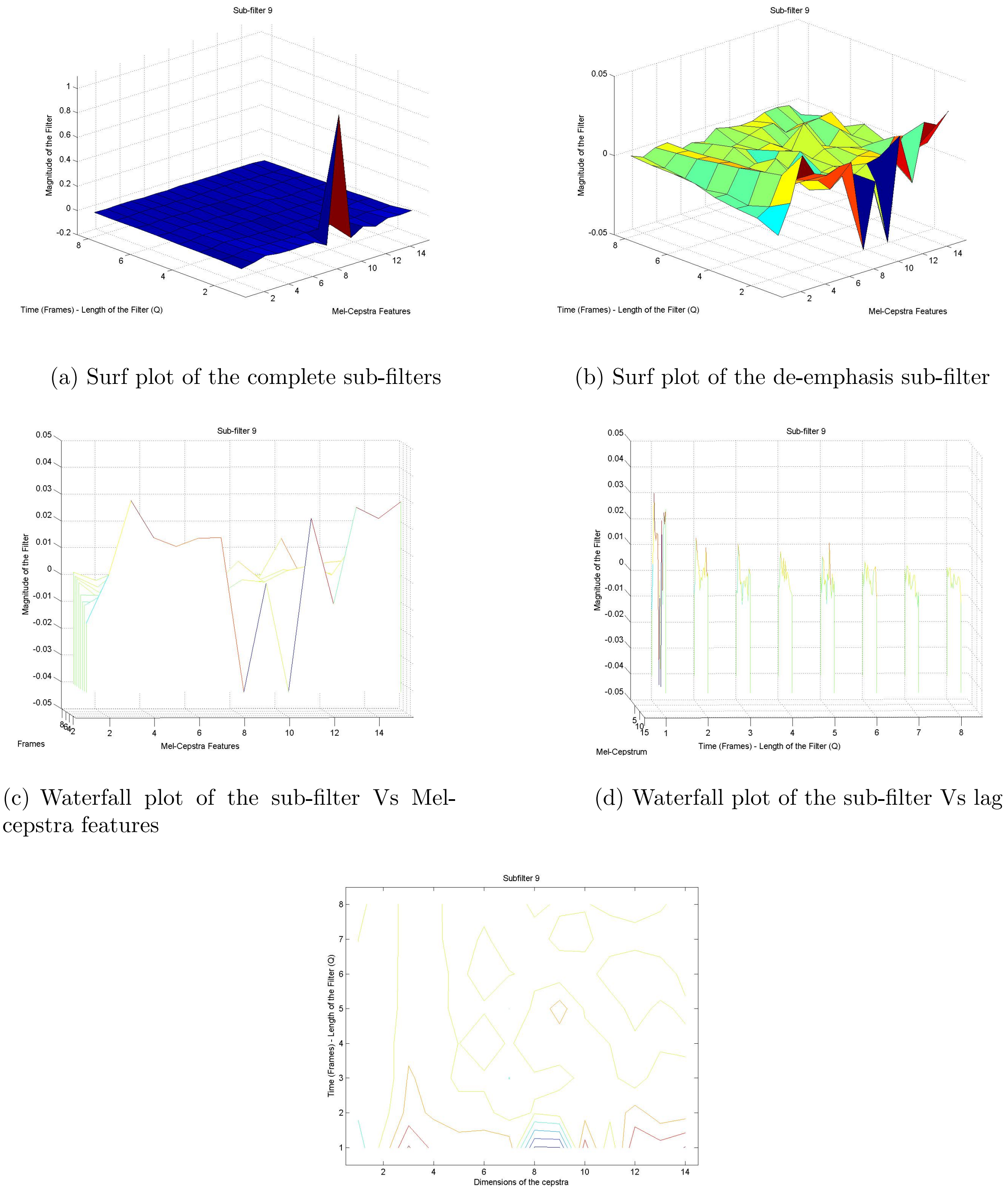
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A8: Sub-filter 8



(a) Surf plot of the complete sub-filters

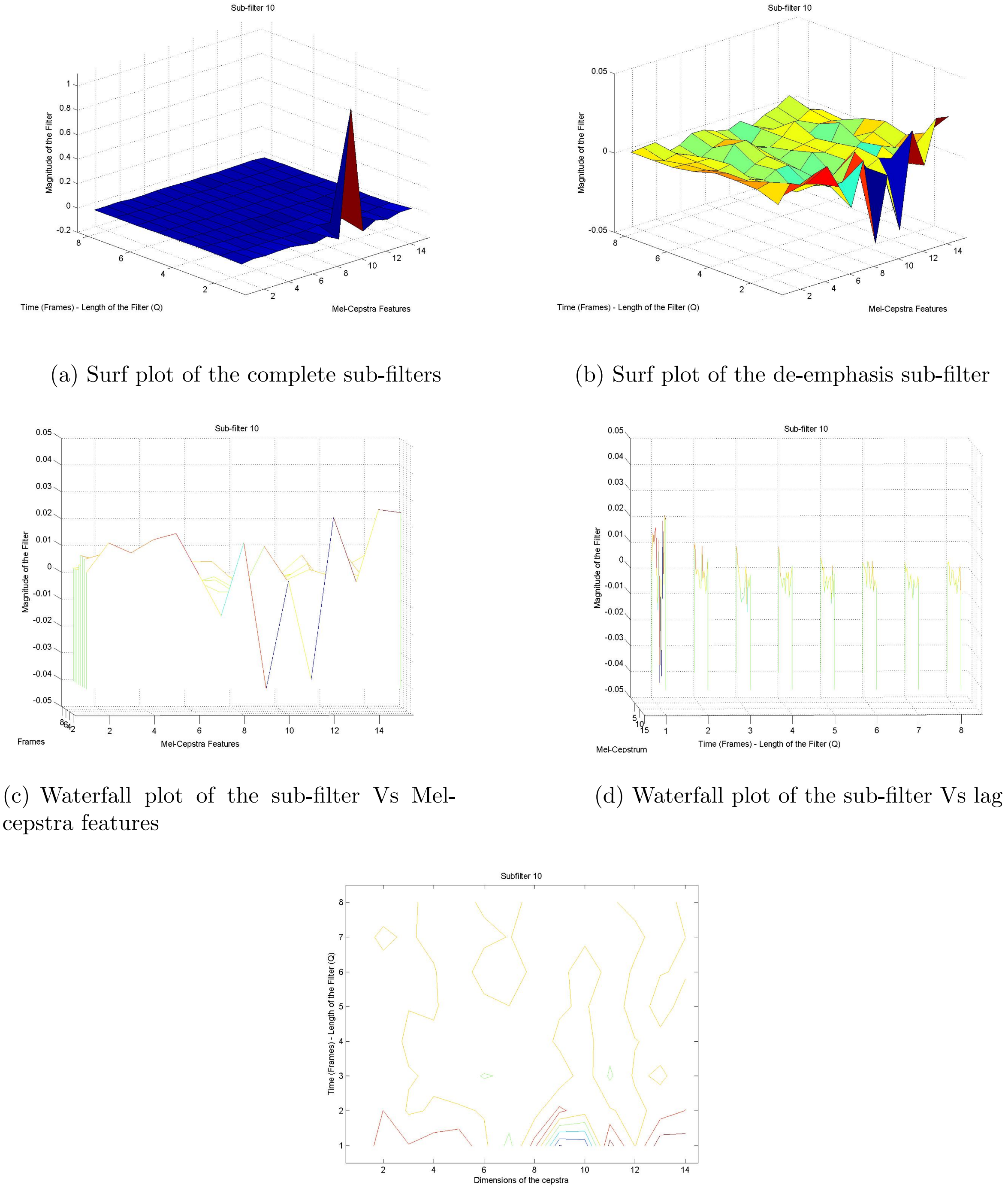
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A9: Sub-filter 9



(a) Surf plot of the complete sub-filters

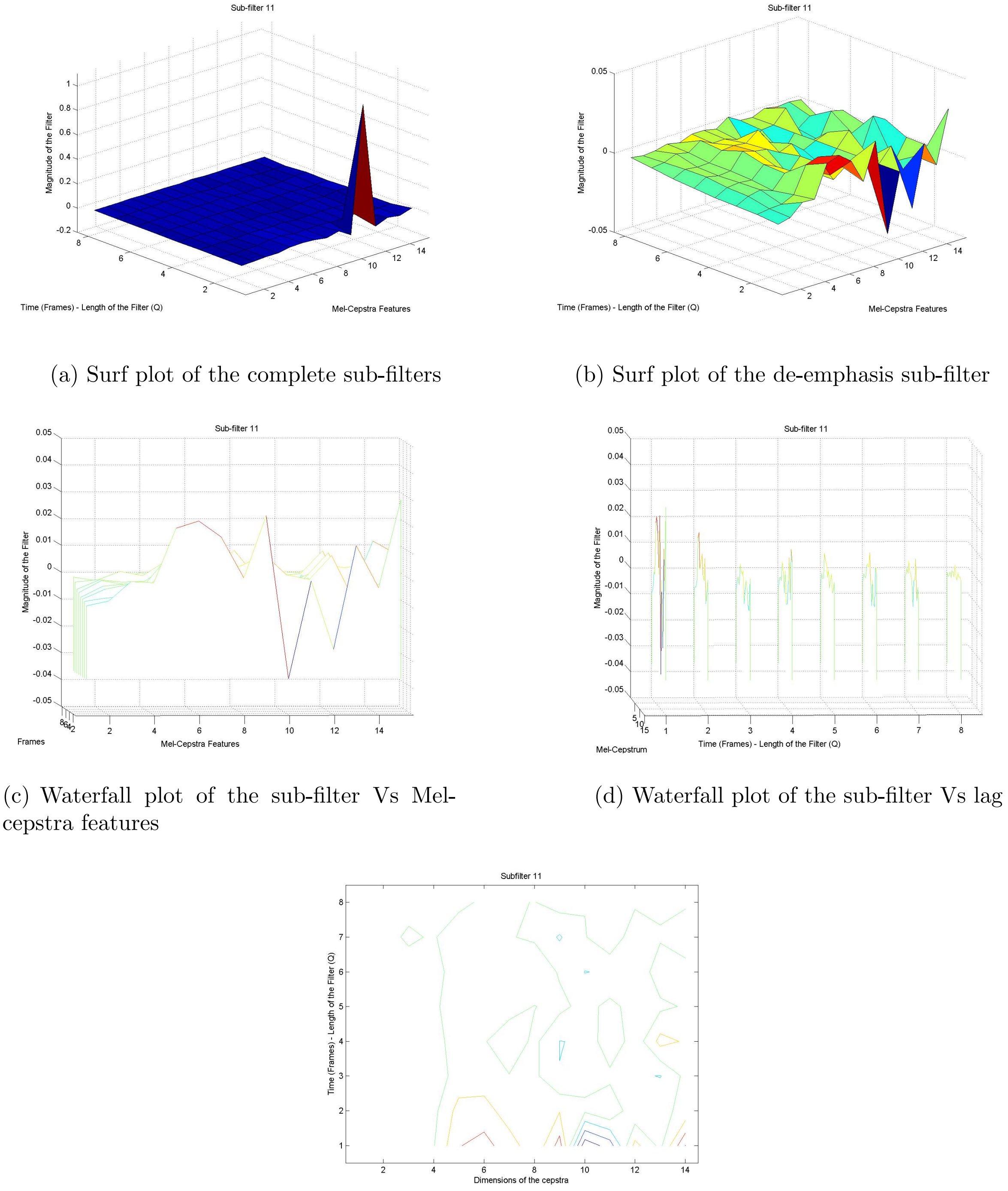
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A10: Sub-filter 10



(a) Surf plot of the complete sub-filters

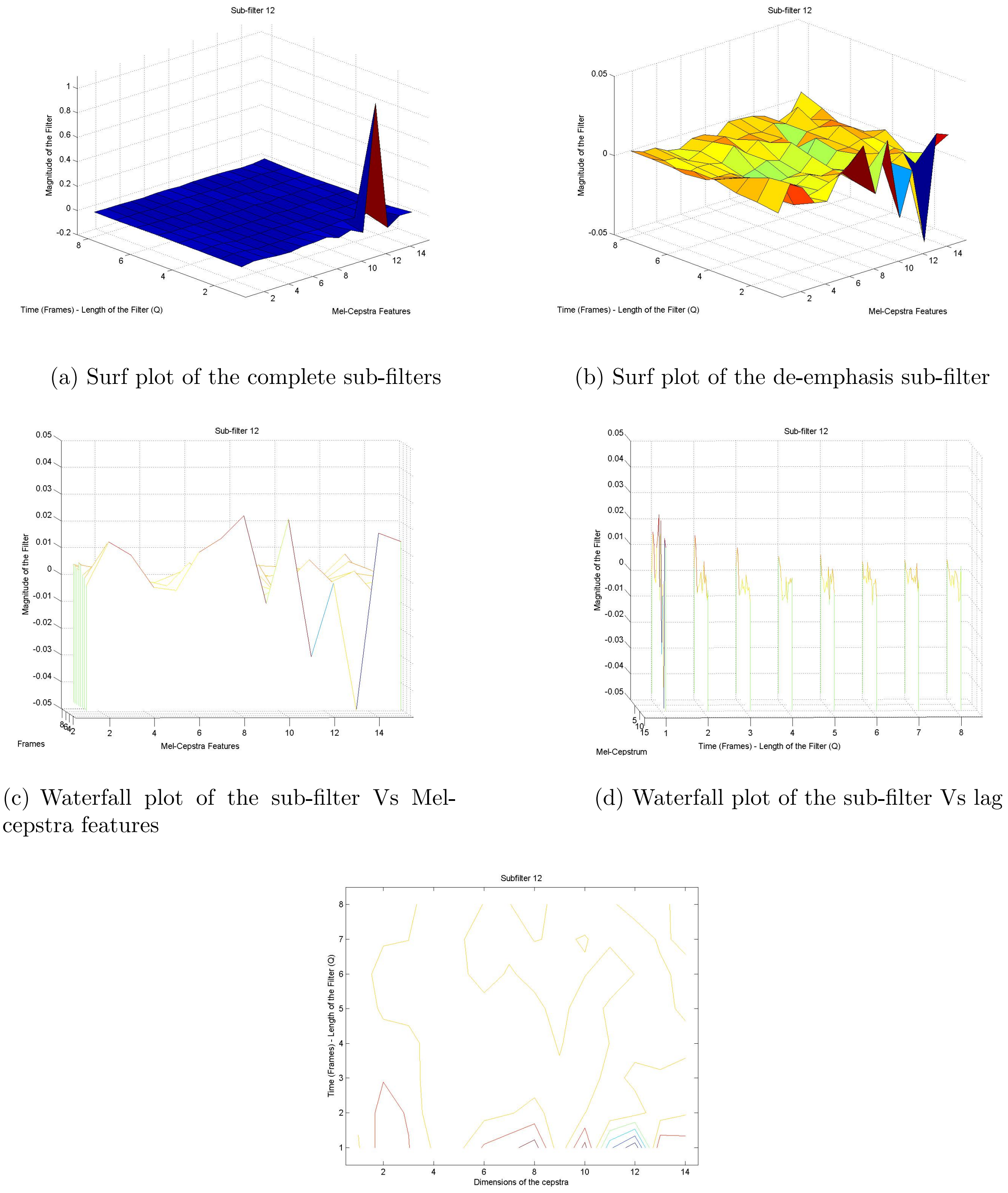
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A11: Sub-filter 11



(a) Surf plot of the complete sub-filters

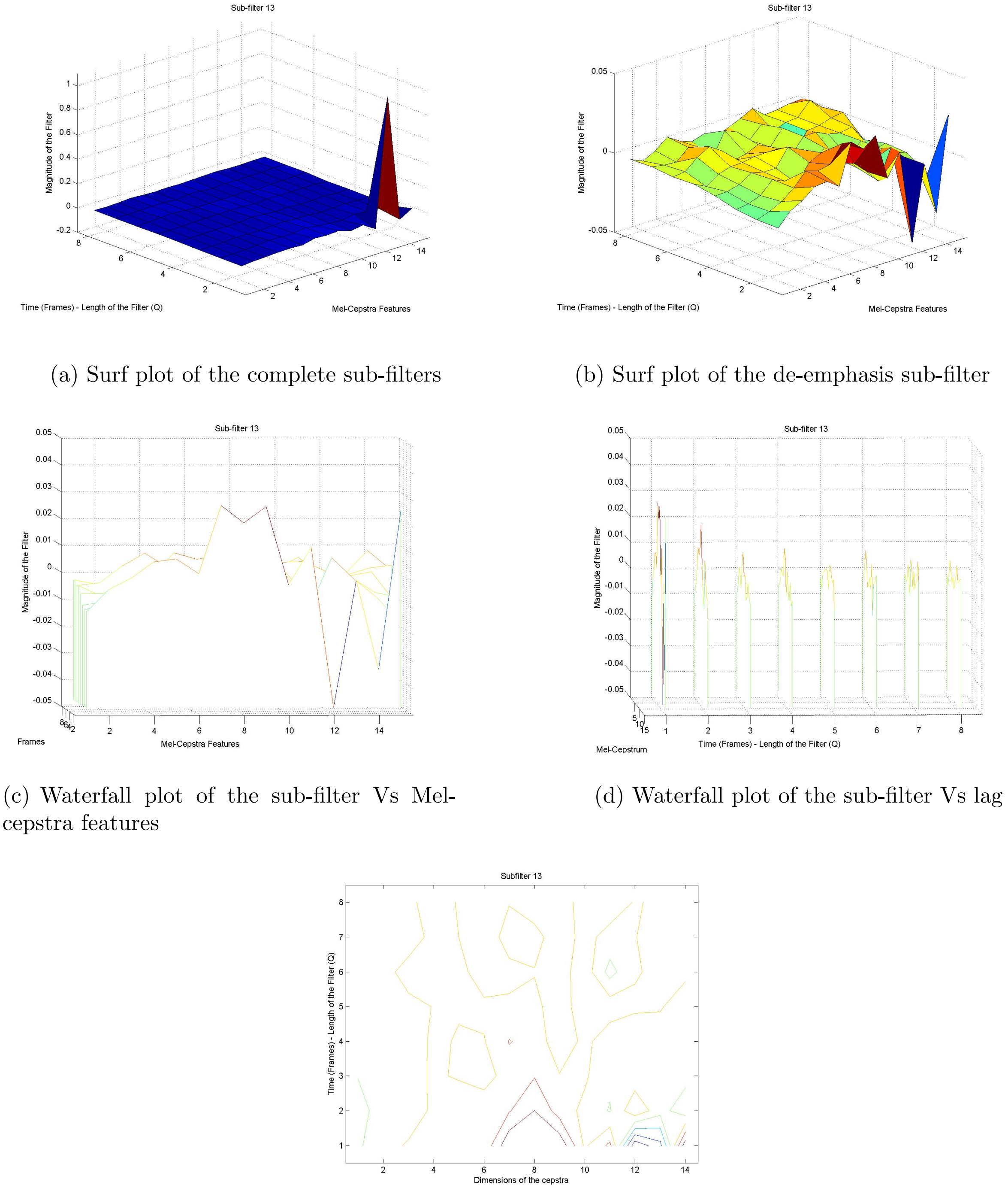
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A12: Sub-filter 12



(a) Surf plot of the complete sub-filters

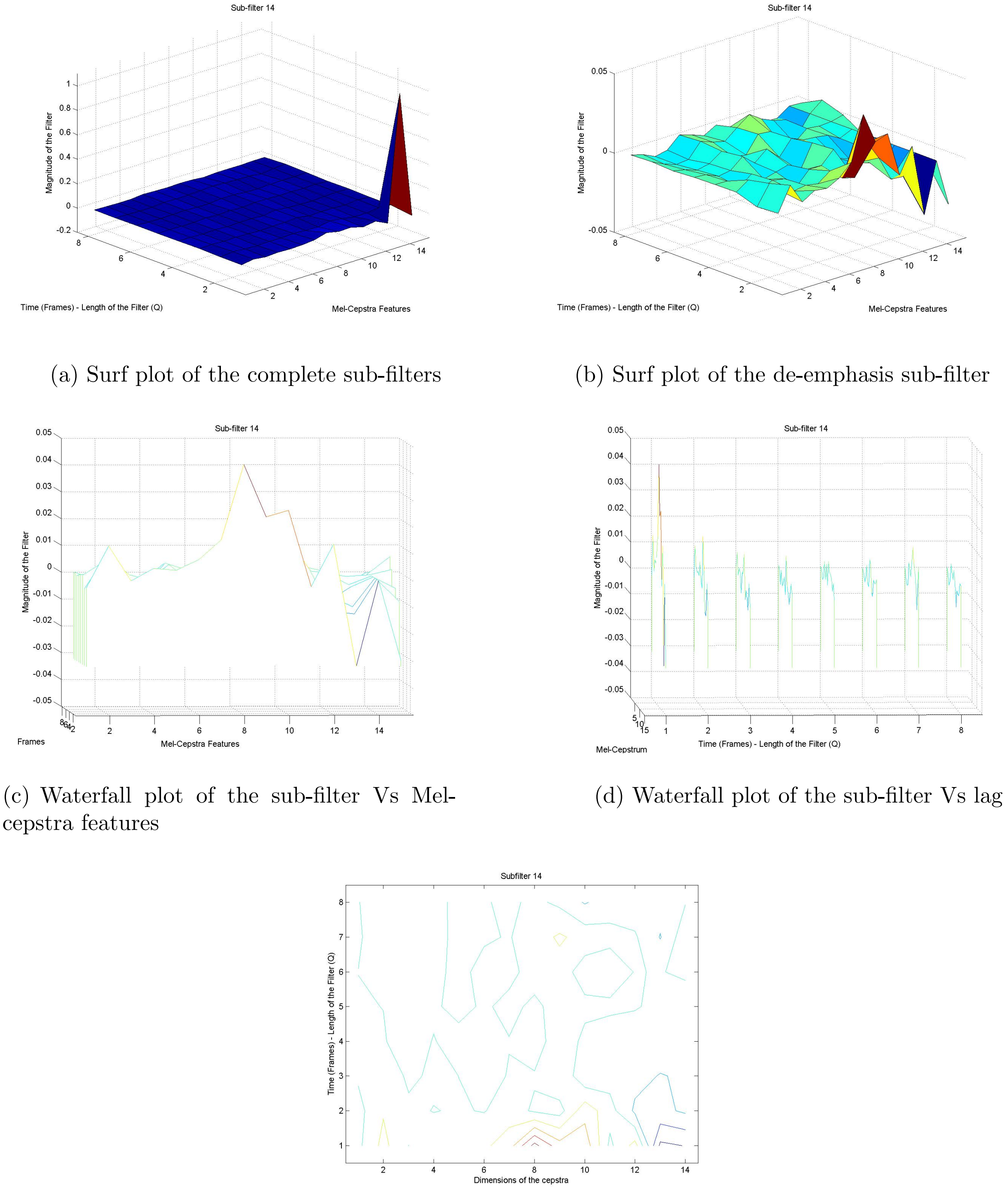
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A13: Sub-filter 13



(a) Surf plot of the complete sub-filters

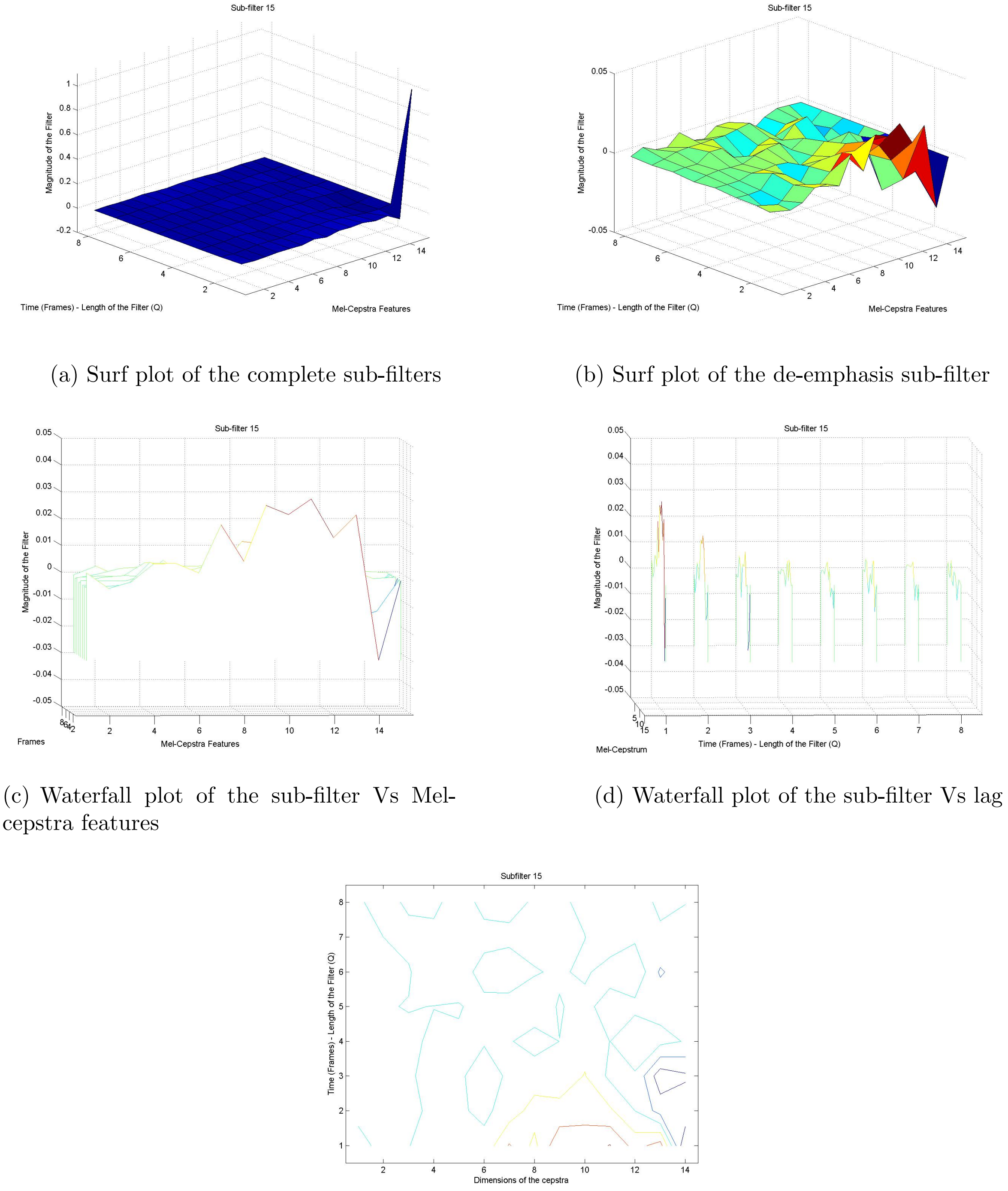
(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A14: Sub-filter 14



(a) Surf plot of the complete sub-filters

(b) Surf plot of the de-emphasis sub-filter

(c) Waterfall plot of the sub-filter Vs Mel-cepstra features

(d) Waterfall plot of the sub-filter Vs lag

(e) Contour plot of the sub-filter

Figure A15: Sub-filter 15

BIBLIOGRAPHY

[1] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan Publishing Co., New York, NY, 1993.

[2] T.F. Quatieri, *Discrete-Time Speech Signals Processing Principles and Practice*, Prentice Hall PTR, Upper Saddle River, NJ 2002.

[3] Y. Viniotis, *Probability and Random Processes for Electrical Engineers*, WCB McGraw-Hill, New York, NY 1998.

[4] A. V. Oppenheim, and R. W. Schafer, with J. R. Buck, *Discrete-Time Signal Processing* Second Edition, Prentice Hall PTR, Upper Saddle River, NJ 1999.

[5] A. V. Oppenheim, and R. W. Schafer, *Digital Signal Processing*, Prentice Hall International, Inc.1975.

[6] Appleton and Perera, *The Development and Practice of Electronic Music*, Prentice-Hall, 1975

[7] Chi-Tsong Chen, *Linear System Theory and Design*, Third Edition Oxford University Press, 1999

[8] E.Weinstein, M. Feder, and A. V. Oppenheim, “Multi-channel signal separation by decorrelation, ” *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 405-413, Apr. 1993.

[9] S.B. Davies and P. Mermelstein, ”Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuous Spoken Sentences,” *IEEE Trans. Acoustic, Speech, and Signal Processing*, vol. ASSP-28, no. 4, pp. 357-366, Aug. 1980.

[10] L. Parra and C. Spence, “Convolutive blind separation of non-stationary sources,” *IEEE Trans. Speech Audio Processing*, vol. 8, no. 3, May 2000.

[11] J.-F. Cardoso, “Eigenstructure of the 4th -order cumulant tensor with aplplication to the blind source separation problem,” *in Proc. ICASSP 89*, 1989, pp. 2109-2112.

[12] C.Jutten and J. Herault, “Blind separation sources part I: An adaptive algorithm based on neuromimetic architecture, ” *Signal Process.*, vol 24, no 1, pp. 1-10 1991.

[13] D. T. Pham, P. Garrat, and C. Jutten, "Separation of a mixture of independent sources through a maximum likelihood approach," *in Proc. EUSIPCO*, 1992, pp. 771-774.

[14] D. Yellin and E. Weinstein, "Multichannel signal separation: Methods and analysis," *IEEE Trans. Signal Processing*, vol. 44, pp. 106-118, Jan. 1996

[15] H.-L. N. Thi and C. Jutten, "Blind source separation for convolutive mixtures," *Signal Process.*, vol 45, no. 2, pp. 209-229, 1995.

[16] S. Shamsunder and G. Giannakis, "Multichannel blind signal separation and reconstruction," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 515-528, Nov. 1997.

[17] L. Parra and C. Spence, and B. De Vries, "Convolutive blind source separation based on multiple decorrelation," *IEEE Workshop Neural Networks Signal Processing*, Cambridge, U.K., Sept 1998.

[18] D. Bradwood, "Cros-coupled cancellation systems for improving cross-polarization discrimination," *Proc. IEEE Int. Conf. Antennas Propagation*, vol. 1, Now. 1978, pp. 41-45.

[19] Y. Bar-Ness, J. Carlin, and M. Steinberger, "Bootstrapping adaptive cross-pol canceller for satellite communications," *Proc. IEEE Int. Conf. Communications*, 1982, pp. 4F5.1-4F5.5.

[20] http://www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi

[21] http://en.wikipedia.org/wiki/Main_Page

[22] http://cnx.rice.edu/content/m10673/latest/

[23] http://www2.sfu.ca/sonic-studio/handbook/Mel.html