KURTOSIS-BASED BLIND BEAMFORMING: AN ADAPTIVE, SUBBAND IMPLEMENTATION WITH A CONVERGENCE IMPROVEMENT

BY

DANIEL C. KLINGLER

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Professor Douglas L. Jones

ABSTRACT

In many speech applications, a single talker is captured in the presence of background noise using a multi-microphone array. Without knowledge of the array geometry, talker location, or the room response, many traditional beamforming techniques cannot be used effectively. An adaptive, maximum-kurtosis objective is used in the frequency domain to blindly enhance the speech signal. The algorithm provides SNR gains of 3.5 - 7.5 dB with just two microphones in low-SNR, real-world scenarios. An improvement is presented that allows for faster and more stable convergence of the algorithm in real-time implementations. Finally, an alternative formulation to the problem is given, framing it in a way that might inspire new discussion or alternative solutions.

“I remember loving sound before I ever took a music lesson. And so we make our lives by what we love.” - John Cage

ACKNOWLEDGMENTS

I would like to thank my parents for their constant love and support. I wouldn't be here without their encouragement and reassurance in pursuing my interests.

To my brother, who shares my passion for music, sound, and learning. Having a friend with similar interests that I can always talk to is instrumental (no pun intended) in keeping me motivated and excited about what I do.

And to Amanda, who has supported me since before I even knew what engineering was. Trying to fit my thanks in a few sentences would be a huge understatement no matter what I wrote.

Thanks to Prof. Doug Jones for his guidance and insights throughout this project. I've learned a ton in the few years I've worked with him, and many of his attitudes and ideas will stick with me throughout my career. I'd also like to thank all the guys in the Mic Array Lab, who were always willing to answer my questions and engage in discussion.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

From the origins of speech signal processing to the present day, the problem of noise and interference is a primary concern in the design of audio systems. It corrupts the desired signal in some way, and requires that extreme care is taken from the desired signal's initial capture or generation to its end use. A variety of techniques have been developed to reduce noise and interference, with applications in hearing aids, cellular phones, teleconferencing, and automatic speech recognition. In the past, improving the quality and intelligibility of speech for only human listeners was the goal of many noise reduction systems. However, with automatic speech recognition becoming a common feature in automobiles, phones, computers, video games, and many other devices, continuing research in noise reduction is important for both the human listener and the automatic speech recognizer.

1.1 Speech Processing

Speech signal processing deals with signals in the audio range (20Hz to 20kHz), that are generated by the human vocal tract. Since speech is inherently acoustic in nature, and signals are usually processed as electrical signals, or samples of electrical signals, the acoustic signal is converted via a microphone to an electrical signal for processing. Ideally, a single microphone can capture frequencies over the entire range of human hearing. However, frequency content is not the only important feature of sound present in a natural acoustic environment. Since a microphone responds to the acoustic signal at approximately a single location in space, the microphone is not capturing much spatial information about the sound. In other words, information about the direction of the sound's origin is not known.

This problem can be addressed by using multiple microphones. Just as a human being's ability to discriminate sounds in a noisy environment is enhanced by using two ears to capture sound, spatial audio processing is made possible with the use of multiple microphones in an array.

1.2 Noise Reduction

There are two main families of noise reduction techniques: single channel and multi-channel. Each of these techniques has its benefits, and often a combination of the two is used in real speech enhancement systems.

1.2.1 Single-channel Noise-reduction Techniques

In single-channel noise-reduction, only one channel of audio is used as input to the algorithm, often along with other information about the signal, such as its statistical properties or the acoustic space through which it was transmitted. This information is used to modify the sound in some way, with the intent of reducing the amount of noise present in the signal. Due to the nature of single-channel techniques, only temporal, statistical, and spectral information can be exploited, while spatial information is not taken into consideration. While single-channel techniques have been found to improve audio quality, they have not been shown to be effective in improving recognition [1].

Common single-channel techniques include spectral subtraction [2], Wiener filtering [3], minimum statistical methods [4], and subspace methods [5]. Since this thesis is concerned with array processing and multi-channel techniques, these noise-reduction methods will not be presented. The referenced papers provide a good introduction to these topics.

1.2.2 Multi-channel Noise-reduction Techniques

In multi-channel noise-reduction techniques, multiple microphones are used as input to the algorithm, allowing the use of spatial information in the audio processing. Beamforming, a form of spatial filtering, introduces a directional dependence on the captured sound. When the signal and noise do not originate from the same direction relative to the microphone array, it is possible to amplify sound from directions that contribute the desired signal, and attenuate sound from directions that may contribute noise.

One way to classify beamformers is to split them into two classes: fixed and adaptive. Fixed beamformers do not change their pickup pattern over time, and are set beforehand based on some previous information, such as the desired signal's direction of arrival, or the acoustic properties of the room. Adaptive beamformers can change their pickup pattern over time, adapting to some parameter being examined by the system. In some cases, the beamformer is completely blind, and has no prior information regarding the desired signal's direction of arrival, the acoustic properties of the room, or even the geometry of the array itself.

A few common beamformers include filter-and-sum (including delay-and-sum), maximum signal-to-noise ratio (SNR), and minimum variance distortionless response (MVDR). These beamformers will be discussed in more detail in the next chapter, to give some context to the subband maximum-kurtosis beamformer we derived and implemented.

1.3 Motivation

Many beamforming methods require prior knowledge of the number, type, or placement of microphones, as well as the acoustical properties of the environment, speech, or noise sources. Flexible methods are valuable, since commercial technologies often impose constraints in these areas. In addition to real-world use of noise-reduction algorithms, it is important to continue to study the human auditory system's response to various noise-reduction methods. The auditory system and the brain are incredibly complex, and intelligibility does not map to a single simple mathematical function that can be maximized or minimized. Therefore, it is worthwhile to continue to explore many methods and heuristics that may result in a greater understanding of what types of algorithms are most useful in noise reduction. We implement a subband maximum kurtosis beamformer, and present improvements and insights for using the maximum kurtosis objective in real-time noise-reduction.

This thesis is organized as follows: In Chapter 2, we introduce common beamforming terminology and review some standard beamformimg techniques. In Chapter 3, we derive the adaptive maximum-kurtosis beamformer, and extend it for use in the frequency domain. The convergence of the algorithm is improved by finding a factorization that removes the updated beamformer weights' dependence on old, less precise values. In Chapter 4, we present the results of the beamformer's performance on both simulated and real-world signals. In Chapter 5, we introduce an alternative formulation of the kurtosis maximization problem, and we discuss how other mathematical methods might be used to approach this problem. We conclude in Chapter 6 with a discussion of the potential problems with this algorithm, and of future work that is relevant to this topic.

CHAPTER 2

BACKGROUND

In this chapter, we present some background information on beamforming and independent component analysis, two techniques relevant to the problem of multi-channel noise reduction.

2.1 Beamforming

Simply put, beamforming is a technique used for directional signal transmission or reception [6]. In the reception of acoustical signals, beamforming amplifies signals radiating from certain directions in space, and attenuates signals from other directions. Using a continuous aperture, or multiple discrete sensors, one can apply spatial filtering to attenuate noise. Beamforming has applications in audio, RADAR, SONAR, communications, imaging, geophysical and astrophysical exploration, and biomedical fields. We only consider beamforming with discrete arrays for audio processing, since microphones are the most commonly used acoustic sensors, and can be easily modeled using discrete arrays.

Since beamforming involves the use of samples from M microphones at each time instant, a vector notation is used to simplify the mathematics. At sample index n, the input to the microphone array is written as

$$\boldsymbol{x}_n = \begin{bmatrix} x_1(n) \\ x_2(n) \\ x_3(n) \\ \vdots \\ x_M(n) \end{bmatrix}$$

The current beamforming weights on each channel are written as

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_M \end{bmatrix}$$

In the simplest case, each beamforming weight is a scale factor applied to its corresponding input channel. The weights are applied to each channel, and the results are summed together, giving a single output. Using vector notation is convenient, as it allows the output at time n to be compactly written as the inner product $\boldsymbol{w}^H \boldsymbol{x}$.

Another concept that arises in beamforming is the notion of a sound's steering vector. A steering vector contains the amplitude and phase that, when applied to each microphone signal, aligns all microphone signals to the reference microphone. If the first microphone is taken as a reference, the steering vector will be of the form

$$\boldsymbol{d} = \begin{bmatrix} 1 \\ d_2 e^{-j\phi_2} \\ d_3 e^{-j\phi_3} \\ \vdots \\ d_M e^{-j\phi_M} \end{bmatrix}$$

A sound's steering vector is usually a function of ω, meaning that a different steering vector exists for each frequency in the sound to be steered. Steering vectors arise in some beamformers, since they can convey the directional information of a desired sound.

2.1.1 Filter-and-Sum Beamformer

Perhaps the most common beamformer is the filter-and-sum beamformer. Often used when the signal of interest is broadband, it allows for both spatial and temporal filtering. In a digital implementation, sound is captured at M microphones, passed through M Q-tap FIR filters, and summed to obtain the output signal y. The filter-and-sum beamformer structure is shown in Figure 2.1, and the output signal $y(n)$ is obtained according to Equation (2.1) [6].

$$y(n) = \sum_{m=1}^{M} \sum_{q=0}^{Q-1} w_m(q) x_m(n-q) \tag{2.1}$$

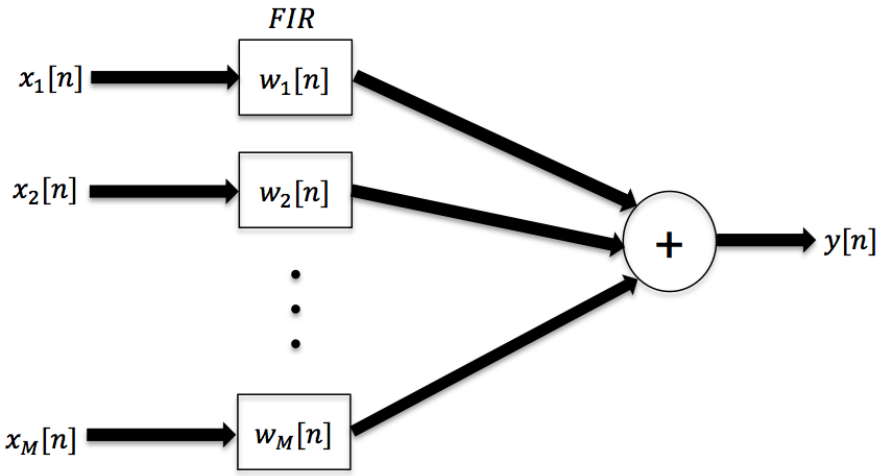

Figure 2.1: Structure of the time-domain filter-and-sum beamformer.

Since the filter-and-sum beamformer is simply a summation of FIR filter outputs, it can be alternatively formulated as multiplications in the frequency domain. Filtering in the Fourier domain can save computation when the filters become long, and convolutions become computationally expensive. Provided that the beamformer is designed to avoid the circular effects of frequency-domain multiplication, the outputs of an FFT can be treated as narrowband signals where a complex weight is applied in each bin. This kind of frequency-domain structure is shown in Figure 2.2.

The popular delay-and-sum beamformer is a subset of the filter-and-sum beamformer, where the filters used are of the form $w_m(q) = \delta(q - d_m)$, a Kronecker delta delayed by d_m samples for each channel.

2.1.2 Maximum SNR Beamformer

While the filter-and-sum beamformer defines a structure, it does not suggest how to choose the filters for some desired performance. This is where statis-

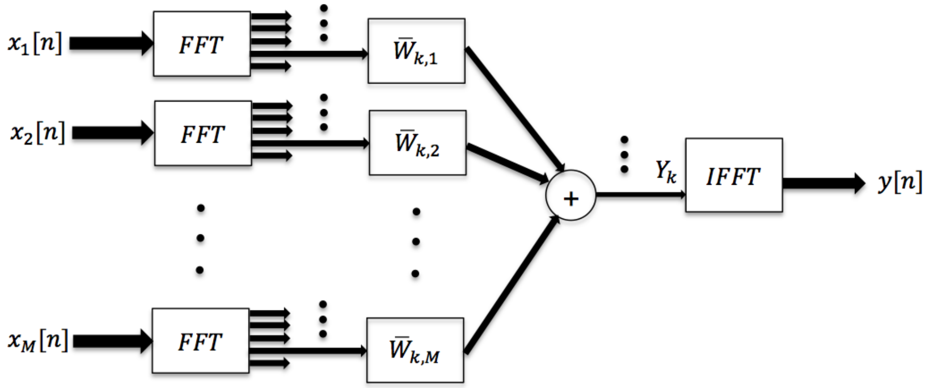

Figure 2.2: Structure of the frequency-domain filter-and-sum beamformer.

tically optimum beamformers enter array processing. The maximum signal-to-noise-ratio (SNR) beamformer is an example of a statistically optimum beamformer that is derived based on a maximization of the signal-to-noise ratio of the output signal. It seeks to find the weights $\boldsymbol{w}$ that provide a combination of the inputs that gives the highest possible signal-to-noise ratio of the output. The maximum SNR beamformer is defined as:

$$\boldsymbol{w} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \frac{\boldsymbol{w}^H \boldsymbol{R}_s \boldsymbol{w}}{\boldsymbol{w}^H \boldsymbol{R}_n \boldsymbol{w}} \tag{2.2}$$

where

$$\boldsymbol{R}_s = E[\boldsymbol{s}\boldsymbol{s}^H] \tag{2.3}$$

and

$$\boldsymbol{R}_n = E[\boldsymbol{n}\boldsymbol{n}^H] \tag{2.4}$$

The signal-to-noise ratio is defined as a ratio of quadratic forms, and can be maximized by using the method of Lagrange multipliers.

$$L = \boldsymbol{w}^H \boldsymbol{R}_s \boldsymbol{w} - \lambda(\boldsymbol{w}^H \boldsymbol{R}_n \boldsymbol{w} - k) \tag{2.5}$$

$$\nabla_w L = 2\boldsymbol{R}_s \boldsymbol{w} - 2\lambda \boldsymbol{R}_n \boldsymbol{w} \tag{2.6}$$

$$0 = 2\boldsymbol{R}_s \boldsymbol{w} - 2\lambda \boldsymbol{R}_n \boldsymbol{w} \tag{2.7}$$

$$\boldsymbol{R}_s \boldsymbol{w} = \lambda \boldsymbol{R}_n \boldsymbol{w} \tag{2.8}$$

$$\boldsymbol{R}_n^{-1} \boldsymbol{R}_s \boldsymbol{w} = \lambda \boldsymbol{w} \tag{2.9}$$

This is a generalized eigenvalue problem, and the optimal $\boldsymbol{w}$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{R}_n^{-1}\boldsymbol{R}_s$ [6].

While the maximum SNR beamformer is a useful tool, the signal and noise covariance matrices must be known. These matrices are often hard to obtain in blind problems, since the incoming microphone signals often contain a mixture of the speech and noise. They can be estimated by attempting to detect speech-only and noise-only segments, but this is difficult in our application.

2.1.3 MVDR Beamformer

The minimum-variance distortionless-response beamformer (also known as the Capon beamformer) is another commonly used statistically optimum beamformer. The MVDR beamformer seeks to preserve the signal in the look direction $\boldsymbol{d}$, and minimize the output signal power from all other directions. This scheme works well in single-source noise reduction if the look direction is chosen as the steering vector of the speech source. Since the output power from other directions is minimized, any noise incident on the array from other directions will be optimally reduced, constrained on a distortionless response in the look direction [7].

Formally, the MVDR beamformer is defined as

$$\boldsymbol{w} = \min_{\boldsymbol{w}} \boldsymbol{w}^H \boldsymbol{R}_x \boldsymbol{w} \tag{2.10}$$

subject to

$$\boldsymbol{w}^H \boldsymbol{d} = 1 \tag{2.11}$$

where

$$\boldsymbol{R}_x = E[\boldsymbol{x}\boldsymbol{x}^H] \tag{2.12}$$

This problem is solved using Lagrange multipliers, giving the standard MVDR beamformer, with the optimal $\boldsymbol{w}$ being [7]

$$\boldsymbol{w} = \frac{\boldsymbol{R}_x^{-1}\boldsymbol{d}}{\boldsymbol{d}^H \boldsymbol{R}_x \boldsymbol{d}} \tag{2.13}$$

Intuitively, the MVDR is a great solution to reducing noise and interference present in a single speech source, since it can maintain the integrity of the signal in the look direction, while optimally reducing the noise from other directions. However, the solution depends on $\boldsymbol{d}$, the steering vector, which is unknown in blind applications. The steering vector can be estimated, but the effectiveness of the beamformer heavily relies on the accuracy of the estimate of $\boldsymbol{d}$. Estimating $\boldsymbol{d}$ using a kurtosis-based method (similar to the method in this thesis) is discussed in [8].

2.2 Independent Component Analysis

While traditional beamformimg methods are commonly used to spatially filter a signal to reduce the presence of noise, other statistical methods exist that can be applied to this problem. Independent component analysis (ICA) is a relatively modern method that can be used to separate independent signals mixed together and received in multiple channels [9]. A standard example in which ICA is useful is source separation of audio signals captured with multiple microphones. The mathematical setup to this problem is shown in Equation (2.14).

$$\begin{bmatrix} channel_1 \\ channel_2 \end{bmatrix} = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix} \begin{bmatrix} talker1 \\ talker2 \end{bmatrix} \tag{2.14}$$

Assuming that the signals from the two talkers are independent, ICA provides a framework from which the two talker signals can be recovered from the two mixed channels when the mixing matrix is unknown.

Since ICA deals with instantaneous mixtures, it cannot separate convolutional mixtures when it is used conventionally. However, a convolutional mixture can be treated as an instantaneous mixture in each frequency bin. This allows for the possibility of employing multiple instances of ICA on every bin in the frequency domain, using complex instead of real linear combinations [10]. When ICA is used in this manner, an issue arises known as the "permutation problem". ICA forces the output signals to be as independent as possible, but it does not label the signals once they are separated. Even if each frequency bin were to be successfully separated, it is not obvious how to correctly label the signals in each bin. Some methods to deal with this problem are addressed in [11], [12], and [13].

Some ICA implementations have the option to extract one source at a time, iterating until all sources have been separated. This method of using ICA is more relevant to the noise-reduction problem, since recovery of only one source (the talker) is desired. Using one-unit ICA with a kurtosis contrast function will recover the highest-kurtosis source in each bin, and the permutation problem becomes relevant only if multiple talkers are present, or the background noise is high-kurtosis.

FastICA is one of the most commonly used software packages to perform ICA, but its goal and implementation are not suited for this application. For example, running FastICA on two mixed input channels gives two channels of unmixed output. However, the unmixing process is performed in a "batch" fashion, using all of input samples at each iteration, with the assumption that the mixing weights are unchanged during the duration of the input. In real-time noise reduction, neither one of these assumptions is true. In our application, only a small window of the input samples can be examined during each iteration, and the environment may be non-stationary, allowing for the optimal unmixing weights to change over time.

CHAPTER 3

APPROACH

In this chapter, we present a beamforming algorithm for speech via kurtosis maximization. The algorithm is first described for a single frequency bin and operates on instantaneous mixtures. Then, a convergence improvement is presented for use with real-time implementations. Finally, the algorithm is extended to all frequency bins, for use with convolutive mixtures.

Similar methods have been developed in [14], [15], [16], [17], and [18]. While all based on the same principle of kurtosis maximization, they provide a variety of algorithms for achieving this goal. Our approach is designed for online, real-time use cases where the speech and noise sources may be non-stationary.

3.1 Problem Statement

Speech recorded in real acoustic environments can be modeled as the desired speech source $s(n)$ convolutively mixed with interference $v_m(n), m = \{1, ..., M\}$, recorded at M microphones.

The signals appear at the m^{th} microphone in the array as

$$x_m(n) = \sum_{p=0}^{P-1} h_m(p) s(n-p) + v_m(n)$$

To attempt to recover the speech signal, a Q-tap FIR filter is applied to each microphone channel, and all channels are summed to form the output $y(n)$:

$$y(n) = \sum_{m=1}^{M} \sum_{q=0}^{Q-1} w_m(q) x_m(n-q)$$

Since w_m is an FIR filter, the problem can be restated in the frequency domain as:

$$Y_k[r] = \boldsymbol{W}_k^H \boldsymbol{X}_k[r]$$

where $k = \{0...K-1\}$ is the frequency bin index, $r = \{0...R-1\}$ is the frame index, $\boldsymbol{X}_k[r] = [X_{1,k}[r], ..., X_{M,k}[r]]^T$, and $\boldsymbol{W}_k = [W_{1,k}, ..., W_{M,k}]^T$

3.2 Maximum-Kurtosis Objective

It has been shown that higher-order signal statistics can be used as a basis for beamforming and source-separation algorithms [14]. In fact, the maximum-kurtosis subband beamformer presented here is a special case of single-source independent component analysis, with a kurtosis contrast function. Improvements to this class of algorithms would have applications to problems using a maximum or minimum kurtosis objective.

Kurtosis is a measure of the peakedness of a distribution, with respect to the Gaussian distribution (which is defined to have zero kurtosis). Distributions that have positive kurtosis are considered super-Gaussian, and tend to have a narrow peak with heavy tails. Distributions with negative kurtosis are sub-Gaussian, and have a much wider main lobe for a given variance.

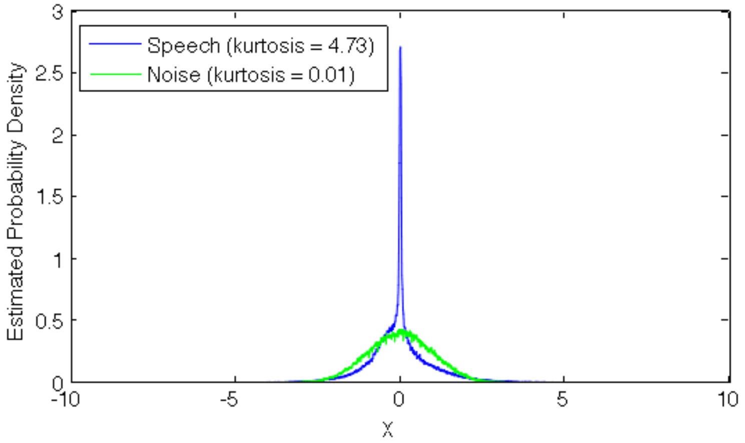

Figure 3.1: Estimated distributions of real speech and noise recorded in a car (10 seconds).

Knowing that the probability distribution function of human speech is high-kurtosis, and noise and interference distributions are typically lower-kurtosis, a maximum-kurtosis objective can be used in source-separation and noise-reduction algorithms [14]. When a high-kurtosis speech source is instantaneously mixed with many lower-kurtosis noise sources, a maximum-kurtosis objective can blindly provide the reconstruction weights to maximize the kurtosis of the output signal. Since the probability density function of speech has a higher kurtosis than noise, the beamformer steers toward a solution that rejects noise while maintaining the speech signal. In the special case when there are M microphones with $M-1$ interferers, a complete removal of the interference sources is possible if the speech is the highest-kurtosis source.

Figure 3.2 shows the kurtosis surface of a signal as a function of the beamformer weights. The goal is to find a linear combination of the input signals that maximizes the kurtosis of the output. As we will see later, we constrain $\boldsymbol{w}$ to lie on the unit circle. Figure 3.3 shows the kurtosis with the unit-norm constraint.

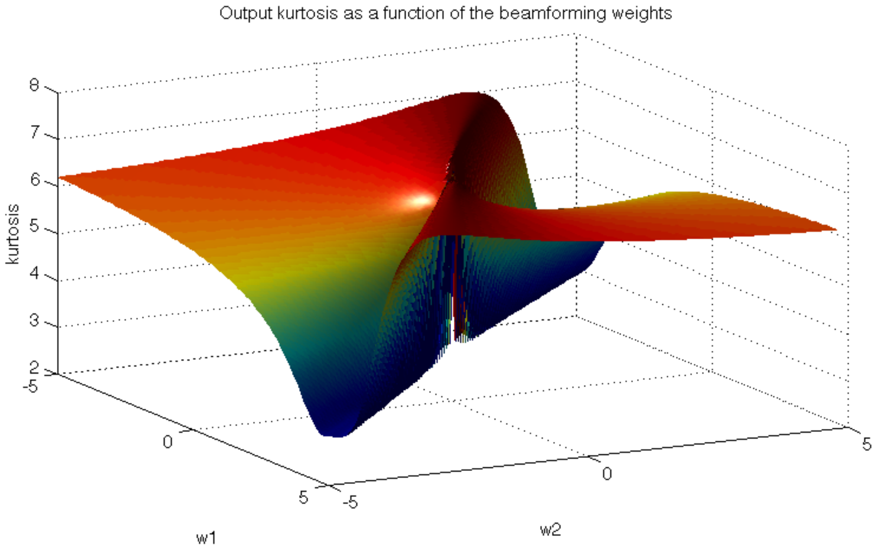

Figure 3.2: Kurtosis surface of an instantaneous mixture of high-kurtosis speech and low-kurtosis noise.

When the problem is extended to convolutive mixtures, a maximum-kurtosis objective can be applied to every bin in the frequency domain. This will provide the reconstruction filters to maximize the kurtosis in each bin, since each bin is a complex instantaneous mixture. Again, an optimal solution should yield a result with a reduction in the presence of noise relative to the speech.

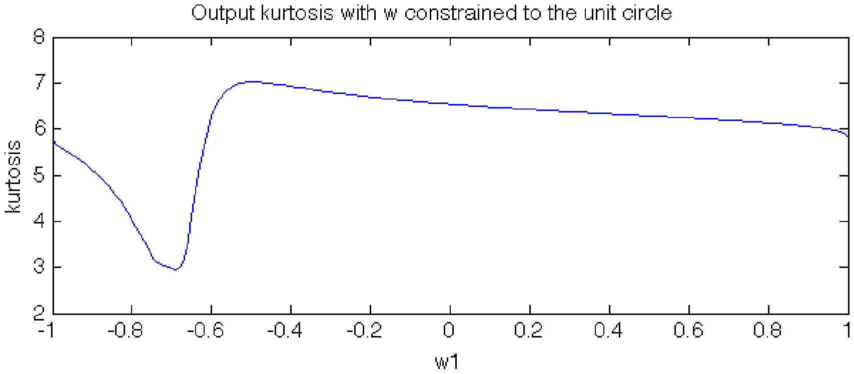

Figure 3.3: Kurtosis of an instantaneous mixture of high-kurtosis speech and low-kurtosis noise (constrained to the unit circle).

To begin, the kurtosis of a complex random variable y is defined as

$$\kappa(y) = \gamma(y) - |\rho(y)|^2 - 2$$

where

$$\gamma(y) = \frac{E\left[|y|^4\right]}{\left(E\left[|y|^2\right]\right)^2}$$

is the normalized fourth-order moment of y, and

$$\rho(y) = \frac{E[y^2]}{E\left[|y|^2\right]}$$

is the circularity quotient [19]. There are many definitions of complex kurtosis [20], but the above definition is the most common in the literature (used in [16], [21], [22],[23]). The circularity quotient goes to zero when y is zero-mean and circular, or in other words, when y has the same distribution as $e^{j\theta}y$ for $\forall\theta \in \mathbb{R}$ [19]. This property is satisfied in our application, since there is no preferred phase of a single frequency bin of the input signal from frame to frame.

3.3 Problem Geometry

To simplify notation, the vector $\boldsymbol{x}$ will represent a single frequency bin's Fourier coefficient for each of the microphones in the array. The beamformer weights in a single frequency bin will be notated as $\boldsymbol{w}$.

The objective is to find the reconstruction weights $\boldsymbol{w}_{opt}$ that maximize $\kappa(\boldsymbol{w}^H\boldsymbol{x})$, the narrowband kurtosis. Since the kurtosis surface is circularly symmetric, $\beta\boldsymbol{w}_{opt}$ is also a maximizer, for any $\beta \neq 0$. Therefore, we constrain $||\boldsymbol{w}||_2^2 = 1$ to ensure $\boldsymbol{w}^H\boldsymbol{x}$ does not grow without bound. Since the problem is non-convex, a gradient ascent method can be employed to numerically find a local maxima. An LMS algorithm is employed, with modifications to project the normalized gradient back onto the unit sphere. The algorithm is described below.

First, the gradient of the kurtosis of the output signal with respect to the reconstruction weights is estimated and normalized. The kurtosis surface is circularly symmetric, and the normalized gradient lies tangent to the unit sphere. The purpose of normalizing the gradient is to ensure a fixed step size in all frequency bins. While this may not be the optimal update strategy, it provides a simple way to ensure all bins converge at a similar rate, with stable convergence properties. The normalized gradient is scaled and added to the current $\boldsymbol{w}$, as in a standard LMS update. This gives the intermediate vector $\boldsymbol{a}$.

$$\boldsymbol{a} = \boldsymbol{w}_{n-1} + \mu\frac{\nabla_{\boldsymbol{w}}\kappa}{||\nabla_{\boldsymbol{w}}\kappa||_2}$$

Since $\boldsymbol{a}$ no longer lies on the unit-norm constraint space, it is projected back onto the unit-sphere.

$$\boldsymbol{w}_n = \frac{\boldsymbol{a}}{||\boldsymbol{a}||_2}$$

Figure 3.4 illustrates the geometry of the LMS update.

3.4 Kurtosis Gradient

The previous section described how to update $\boldsymbol{w}$, but it did not show how to calculate the gradient used in the LMS update. This section describes how that gradient is estimated, and Subsection 3.4.1 presents an improvement to the gradient estimate that is very useful for real-time applications of maximum-kurtosis beamforming.

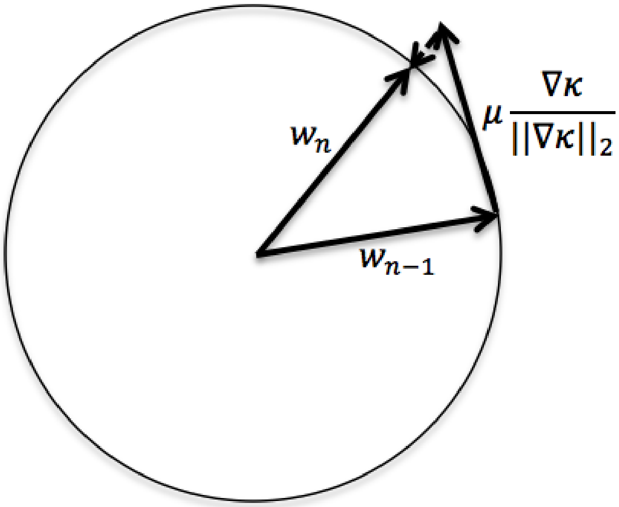

Figure 3.4: Algorithm update geometry.

Recalling the definition of complex kurtosis,

$$\kappa(y) = \frac{E\left[|y|^4\right]}{\left(E\left[|y|^2\right]\right)^2} - 2$$

a single frequency bin $y = \boldsymbol{w}^H\boldsymbol{x}$ can be computed as a linear combination of a single bin of the microphone array $\boldsymbol{x}$ with the reconstruction weight vector $\boldsymbol{w}$:

$$\kappa(y) = \frac{E\left[\left|\boldsymbol{w}^H\boldsymbol{x}\right|^4\right]}{\left(E\left[|\boldsymbol{w}^H\boldsymbol{x}|^2\right]\right)^2} - 2$$

Then, the numerator and denominator are expanded so the absolute value operation is no longer required. Since the matrix $\boldsymbol{xx^H}$ is Hermitian symmetric, the product $\boldsymbol{w^Hxx^Hw}$ is non-negative, giving

$$\kappa(y) = \frac{E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]}{\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^2} - 2$$

The gradient of kurtosis with respect to the reconstruction weights is

$$\frac{\partial\kappa(y)}{\partial\boldsymbol{w}} = \frac{\partial}{\partial\boldsymbol{w}}\frac{E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]}{\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^2}$$

The gradient is expanded using the product rule,

$$\begin{aligned}\frac{\partial\kappa(y)}{\partial\boldsymbol{w}} &= \frac{\partial}{\partial\boldsymbol{w}}\left(E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]\right)\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^{-2} \\ &+ E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]\frac{\partial}{\partial\boldsymbol{w}}\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^{-2}\end{aligned}$$

$$\begin{aligned}\frac{\partial\kappa(y)}{\partial\boldsymbol{w}} &= E\left[2\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)\left(2\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right)\right]\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^{-2} \\ &+ E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right](-2)\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^{-3}E\left[2\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial\kappa(y)}{\partial\boldsymbol{w}} &= \frac{4E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right]}{\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^2} \\ &- \frac{4E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right]}{\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^3}\end{aligned} \tag{3.1}$$

3.4.1 Convergence Improvement for Real-time Implementations

Before Equation (3.1) is used in an adaptive algorithm, the four expectations appearing in the equation must be estimated in some manner. One common estimate of the expected value of a function $f(\boldsymbol{w},\boldsymbol{x})$ at time n is simply the sample mean of length N, shown in Equation (3.2).

$$E_n[f(\boldsymbol{w},\boldsymbol{x})] \approx \frac{1}{N}\sum_{k=n-N+1}^{n} f(\boldsymbol{w}_k,\boldsymbol{x}_k) \tag{3.2}$$

Since $\boldsymbol{w}$ is being updated over time, it may be desired that $\boldsymbol{w}_k$ in Equation (3.2) be replaced with $\boldsymbol{w}_n$, the most recent estimate of $\boldsymbol{w}$. This gives an estimate that assumes the most recent, and presumably more accurate, value of $\boldsymbol{w}$ is used over the entire window of the sum. Old values of $\boldsymbol{w}$ would not be used for current estimates of the expectation, essentially “flushing” the old, less-accurate values of $\boldsymbol{w}$ from the computation. This modified version of Equation (3.2) is shown in Equation (3.3).

$$E_n[f(\boldsymbol{w},\boldsymbol{x})] \approx \frac{1}{N}\sum_{k=n-N+1}^{n} f(\boldsymbol{w}_n,\boldsymbol{x}_k) \tag{3.3}$$

A problem with using the sample mean is that the entire sum must be recomputed often, since $\boldsymbol{w}$ changes each step. This is usually expensive for real-time applications, so an autoregressive moving average can be used instead, shown in Equation (3.4). This update requires little computation, making it suitable for use in real-time systems.

$$E_n[f(\boldsymbol{w},\boldsymbol{x})] \approx \alpha E_{n-1}[f(\boldsymbol{w},\boldsymbol{x})] + (1-\alpha) f(\boldsymbol{w_n},\boldsymbol{x_n}) \tag{3.4}$$

At this point in the derivation, it is tractable to use the Equation (3.1) in an online adaptive algorithm, with expectations estimated using Equation (3.4). However, Equation (3.4) suffers from the same issue as Equation (3.2): the current expectation estimates rely on old values of $\boldsymbol{w}$. This slows the convergence of the algorithm, and can cause the value of $\boldsymbol{w}$ to overshoot the desired value, due to the gradient's dependence on past values of $\boldsymbol{w}$.

Ridding Equation (3.4) of its dependence on old values of $\boldsymbol{w}$ is not as simple as the modification made to Equation (3.2), since Equation (3.4) is defined to be autoregressive. The computational savings of the autoregressive average comes from the fact that we use old estimates of the expected value to update the new estimate. However, there is a solution to the problem if $E_n(f(\boldsymbol{w},\boldsymbol{x}))$ can be factored into $a(\boldsymbol{w}_n)E_n(b(\boldsymbol{x}))c(\boldsymbol{w}_n)$, for some functions a, b, and c. This way, the most recent value $\boldsymbol{w}_n$ can be used, making the expectation depend only on $\boldsymbol{x}$. Such a factorization can indeed be found, retaining the computational savings of an autoregressive average without using old values of $\boldsymbol{w}$ in the computation.

Using the matrix vectorization operator *vec*, the $\boldsymbol{w}$ vectors were factored out of the four expectations. The *vec* operator stacks the columns of an m x n matrix to form a mn x 1 column vector. The following relationships were used, and are expanded on in Appendix A:

$$E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right] = vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H E\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)$$

$$E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right] = vec\Big(\boldsymbol{w}\left(vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\right)^H\Big)^H E\Big[vec\Big(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\Big)\boldsymbol{x}^H\Big]$$

$$E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right] = \boldsymbol{w}^H E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\boldsymbol{w}$$

$$E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right] = \boldsymbol{w}^H E\left[\boldsymbol{x}\boldsymbol{x}^H\right]$$

The final gradient expression then becomes:

$$\frac{\partial\kappa(y)}{\partial\boldsymbol{w}} = \frac{4\,vec\Big(\boldsymbol{w}\left(vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\right)^H\Big)^H E\Big[vec\Big(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\Big)\boldsymbol{x}^H\Big]}{\left(\boldsymbol{w}^H E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\boldsymbol{w}\right)^2} \tag{3.5}$$

$$-\frac{4\,vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H E\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\boldsymbol{w}^H E\left[\boldsymbol{x}\boldsymbol{x}^H\right]}{\left(\boldsymbol{w}^H E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\boldsymbol{w}\right)^3}$$

To update the gradient according to the autoregressive averager in Equation (3.4), the following three values must be updated at every time step n. The parameter α can be used to control the time constant over which the expectation is estimated.

$$\begin{aligned} E_n\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] = &\alpha E_{n-1}\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] \\ &+\ (1-\alpha)\,vec\left(\boldsymbol{x}_n\boldsymbol{x}_n^H\right)vec\left(\boldsymbol{x}_n\boldsymbol{x}_n^H\right)^H \end{aligned}$$

$$\begin{aligned} E_n\left[vec\left(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\right)\boldsymbol{x}^H\right] = &\alpha E_{n-1}\left[vec\left(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\right)\boldsymbol{x}^H\right] \\ &+\ (1-\alpha)\,vec\left(\boldsymbol{x}_n\left(vec\left(\boldsymbol{x}_n\boldsymbol{x}_n^H\right)\right)^H\right)\boldsymbol{x}_n^H \end{aligned}$$

$$E_n\left[\boldsymbol{x}\boldsymbol{x}^H\right] = \alpha E_{n-1}\left[\boldsymbol{x}\boldsymbol{x}^H\right] + (1-\alpha)\,\boldsymbol{x}_n\boldsymbol{x}_n^H$$

The intuition behind this convergence improvement is the following: Suppose the current $\boldsymbol{w}$ is $\boldsymbol{w}_{opt}$. If the optimal $\boldsymbol{w}$ has already been found, the kurtosis gradient should be zero, because further adaptation of $\boldsymbol{w}$ is not required. When using the factored gradient from Equation (3.5), this is the case in the mean, since only the current $\boldsymbol{w}$ is used in the gradient calculation. However, using Equation (3.4) in Equation (3.1) may produce a nonzero gradient even if $\boldsymbol{w} = \boldsymbol{w}_{opt}$, due to its dependence on values of $\boldsymbol{w}$ before convergence. A nonzero gradient produces an update for $\boldsymbol{w}_{opt}$, causing a deviation from the desired $\boldsymbol{w}$. This process repeats each time $\boldsymbol{w}_{opt}$ is reached, causing some amount of oscillation, as evidenced by Figure 4.1, presented in the next chapter.

3.5 Frequency-Domain Extension

Using the above algorithm to find the beamforming weights maximizing the kurtosis for a single frequency bin, we examine the extension to all bins in parallel.

The first problem to consider is that the kurtosis-maximizing beamformer is invariant to a complex scale factor. Without careful treatment of this issue, the phase in each frequency bin would be essentially random, and the output signal would have poor phase coherence and unnecessary distortion. The solution is to constrain the phase of w_1, the first element of $\boldsymbol{w}$, to be zero across all bins. This ensures consistency among bins, and forces the phase of the rest of $\boldsymbol{w}$ to converge in reference to w_1. This is accomplished by multiplying each element of $\boldsymbol{w}$ by $e^{-j\theta_1}$, during each LMS update, where θ_1 is the phase of w_1.

A second challenge is ensuring the algorithm's convergence in all frequency bins. If there is little energy in a particular bin for an extended period of time, the kurtosis gradient is very noisy, and it will not provide an update that increases the output kurtosis in that bin. To try to force all frequency bins to converge at approximately the same rate, a normalized gradient was used in the LMS update. For robustness in a real-world application, a more sophisticated update scheme is probably necessary to ensure convergence over all frequency bins.

CHAPTER 4

IMPLEMENTATION AND TESTING

To verify the algorithm's performance, four tests were run using the maximum-kurtosis algorithm developed in Chapter 3. The tests ranged from a simulated source separation to real-world noise reduction using a dashboard microphone array in a moving car.

4.1 Implementation Details

All signals used in testing were 16-bit WAV files with a sample rate of 11.025 kHz. Before processing, the signals were high-pass filtered with a cutoff of 150 Hz, in order to remove the bias caused by a lack of speech content in the low frequencies. A 1024-point Hann window was used in a mixed overlap save/add implementation. The beamformer weights were recomputed every 128 samples, the step size of the algorithm. This implies the reconstruction filters used are 1024-tap FIR filters, giving around 5 Hz resolution in the frequency domain.

Another important parameter is the kurtosis time-constant α, which determines the window over which the kurtosis gradient is estimated. A primary reason why the pdf of speech becomes very large near zero is due to the amplitude fluctuations between syllables or words, so choosing a window length that captures more than one syllable is vital in measuring a consistently high kurtosis value over time. Choosing the time constant between 0.25 and 3 seconds provides the best results, and intuitively agrees with why speech is higher kurtosis than noise.

In each of the tests, the standard measure of SNR gain is given, quantifying the reduction of noise present in the speech signal.

4.2 Testing and Results

4.2.1 Test 1: Source Separation of Simulated Instantaneous Mixtures

The first test was to separate two discrete sources mixed in two channels. This problem can be modeled as passing a speech and noise source through the mixing matrix $\boldsymbol{M}$ to give two mixed channels.

$$\begin{bmatrix} channel_1 \\ channel_2 \end{bmatrix} = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix} \begin{bmatrix} speech \\ noise \end{bmatrix}$$

The speech can be recovered by finding the weights $\boldsymbol{w}$ to recover *speech*, where M is assumed to be unknown to the algorithm:

$$speech = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} channel_1 \\ channel_2 \end{bmatrix}$$

This is the blind source separation problem, where the recovery of only one source is desired. Since this is an instantaneous mixture, a single instance of the algorithm was employed to adaptively find $\boldsymbol{w}$. The speech source was a male talker, captured in a quiet room, and the noise source was a relatively stationary recording of street noise. Both sound files were obtained from the MIT ICA Synthetic Benchmarks website [24].

The mixing matrix was arbitrarily chosen as

$$\boldsymbol{M} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.3 \end{bmatrix}$$

making the true recovery weights

$$\boldsymbol{w} = \pm \begin{bmatrix} -0.5145 \\ 0.8575 \end{bmatrix}$$

The convergence of $\boldsymbol{w}$ over time is shown in Figure 4.1.

To demonstrate the convergence benefits of factoring $\boldsymbol{w}$ out of the expected-value estimates (developed in Section 3.4.1), two plots are shown. The upper plot shows the convergence of $\boldsymbol{w}$ using the unfactored gradient from Equation (3.1), and the lower plot shows the convergence when using the improved

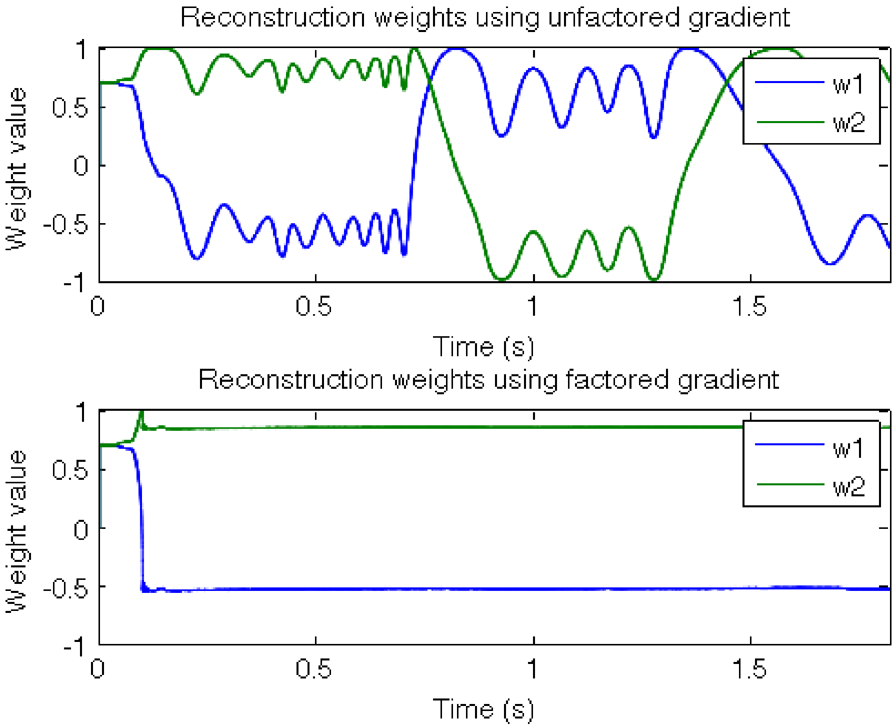

Figure 4.1: Convergence of reconstruction weights for an instantaneous mixture.

factored gradient expression from Equation (3.5). When the unfactored gradient is used, the algorithm has memory of old values of $\boldsymbol{w}$, and overshoots the true value. When the factored gradient is used instead, there is no memory of old, less accurate values of $\boldsymbol{w}$, and it is obvious that the algorithm converges more smoothly, with little ringing or oscillation.

The maximum-kurtosis algorithm consistently provided an SNR increase of more than 50 dB, signaling a successful extraction of the speech source in a simple two-channel instantaneous mixture.

4.2.2 Test 2: Two-Channel Source Separation of Simulated Convolutive Mixtures

The second test was an extension of the first test to convolutive mixtures. In this example, the speech and noise sources were passed through FIR channels, instead of being combined using an instantaneous mixing matrix. The 50-tap FIR filters were randomly generated, and the two simulated microphone

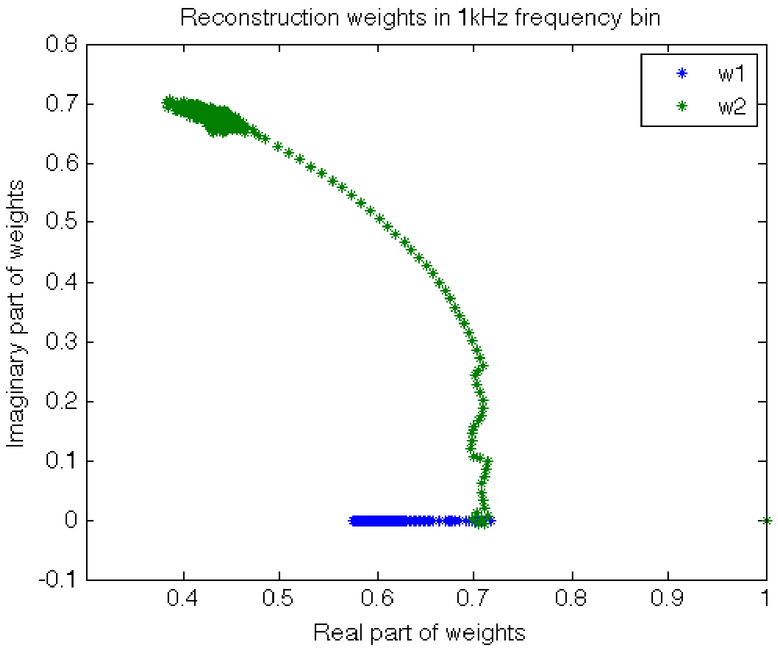

Figure 4.2: Convergence of reconstruction weights (1 kHz frequency bin) for convolutive mixture.

recordings were obtained using the equations in (4.1), where (*) denotes convolution and $C_{i,j}$ is a 50-tap FIR filter.

$$\begin{aligned} channel_1 &= C_{1,1} * speech + C_{1,2} * noise \\ channel_2 &= C_{2,1} * speech + C_{2,2} * noise \end{aligned} \tag{4.1}$$

To deal with convolutive mixtures, an instance of the maximum kurtosis algorithm was employed in each frequency bin.

Figure 4.2 shows the convergence of the reconstruction weights for the 1 kHz frequency bin. Notice that w_1 is constrained to be real, as discussed in Section 3.5.

The subband maximum-kurtosis algorithm consistently provided an SNR increase of more than 25 dB, indicating significant extraction of the speech source in convolutive mixtures. The SNR gain is not as large as the instantaneous mixture case due to the fact that the algorithm must converge in all frequency bins independently. Finding parameters that ensure convergence in all bins is more difficult, especially since some bins contain almost entirely noise, and hardly any speech.

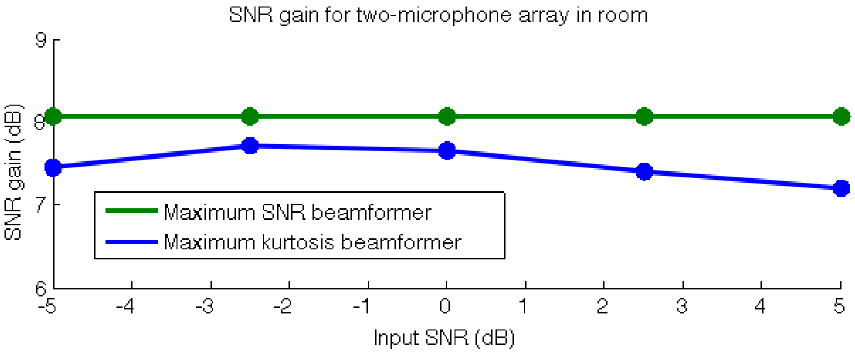

Figure 4.3: SNR gains for speech and noise recorded in a real room.

4.2.3 Test 3: Two-Channel Noise Reduction in a Real Room

The third test was a real-world noise-reduction problem using a two-microphone array in a medium sized, reverberant room. The microphone array comprised two Shure KSM137 cardioid microphones spaced 11 cm apart as a linear array, with broadside defined as 0 degrees. The room was 5 x 5 m, with a carpeted floor and hard walls. A pair of speakers were set up 1 m from the array. The first speaker was placed at 0 degrees, and played a recording of a male talker. The second speaker was placed at 90 degrees to the array, and played a recording of street noise. The speech and noise were recorded separately so that they could be mixed at various signal-to-noise ratios for testing.

To qualify the results obtained with the maximum kurtosis beamformer, we compared its results to the maximum SNR beamformer. Since the speech and noise were recorded separately and mixed at various SNRs, the signal and noise covariances could be directly computed to find the maximum SNR solution. These statistics could be unknown in practice, but provide an upper bound to the performance of the maximum kurtosis beamformer, which estimates these unknown parameters with no prior information.

The subband maximum-kurtosis algorithm was run using input SNRs of -5 dB, -2.5 dB, 0 dB, 2.5 dB, and 5 dB. The algorithm consistently produced SNR gains of around 7 dB over the range of input SNRs, coming rather close to the maximum SNR beamformer's optimal SNR gains of 8 dB. Figure 4.3 shows the SNR gain as a function of input SNR for this experiment.

The reconstruction filters' impulse responses are shown in Figure 4.4. The first filter appears to approximate an impulse, while the second filter appears

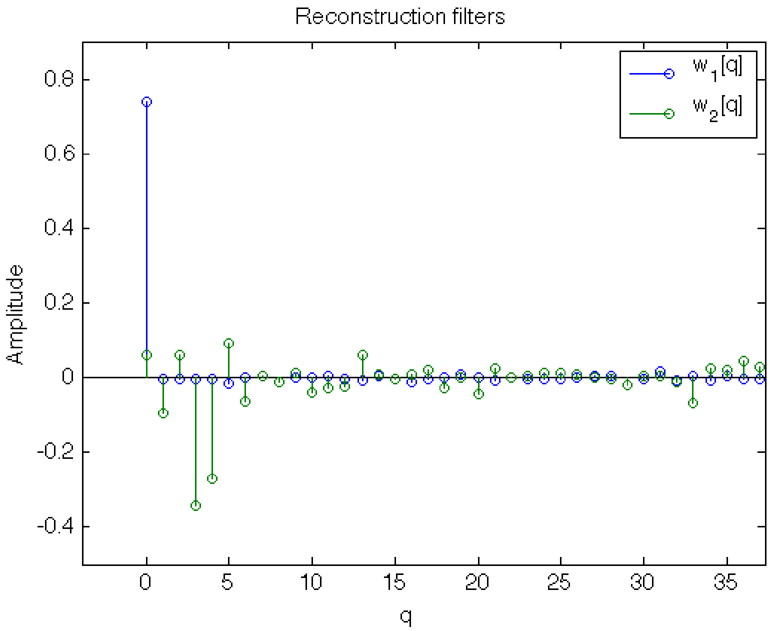

Figure 4.4: Reconstruction filters for speech and noise recorded in a real room.

similar to a scaled and shifted sinc function. This agrees with the intuitive solution for removing the signal present at 90 degrees. Applying a 3.5 sample delay and polarity inversion to one microphone, and summing it with the second microphone, produces a null at 90 degrees for the current array geometry and sampling rate.

Both beamformers' beam-patterns are shown in Figure 4.5. Notice the null produced at 90 degrees, in an effort to remove the noise source present from that direction. The downside is a comb filtering of the speech source present at 0 degrees. Visually, the beam-patters appear quite similar, verifying the maximum-kurtosis beamformer's ability to estimate the unknown statistics used in calculating the maximum-SNR beamformer.

4.2.4 Test 4: Two-Channel Noise Reduction in a Car

The fourth test was a very challenging real-world noise-reduction problem using a two-microphone array in a car. The linear array was mounted in the center of the dashboard of a 2005 Nissan Altima, pointed directly toward the back of the car. The talker was seated in the driver's seat and was recorded while the car was stationary in a very quiet environment. The noise signals

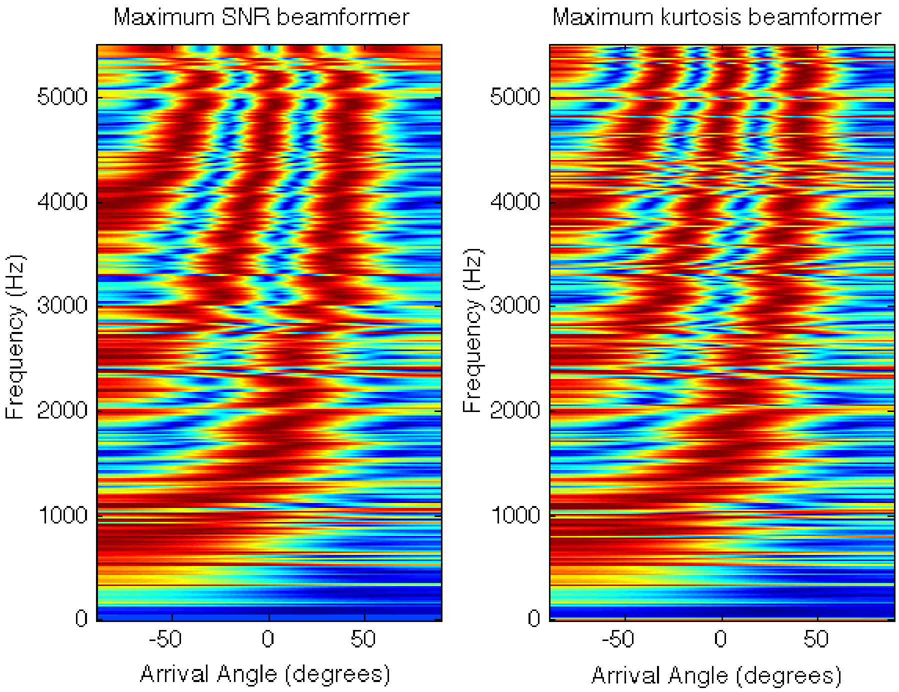

Figure 4.5: Beampattern comparison between the subband maximum SNR beamformer and the subband maximum kurtosis beamformer for speech and noise recorded in a real room.

were recorded while the car was traveling at 100 kph on the highway with the windows closed.

Like the previous experiment, the speech and noise signals were recorded separately, and were mixed at a variety of SNRs for testing. The subband maximum-kurtosis algorithm was run using input SNRs of -5 dB, -2.5 dB, 0 dB, 2.5 dB, and 5 dB, and again compared to subband maximum SNR beamformer. The results are presented in Figure 4.6 and show improvements of close to 5 dB at low SNRs. The maximum SNR beamformer provided an SNR gain of 5.1 dB, slightly better than the blind maximum-kurtosis method.

Both beam patterns are shown in Figure 4.7. While the overall structure is the same, the maximum-kurtosis beamformer's pattern looks noisier and less defined when compared to the maximum SNR beamformer. This is due to the adaptive nature of the maximum-kurtosis algorithm, and the difficulty in ensuring convergence in each bin.

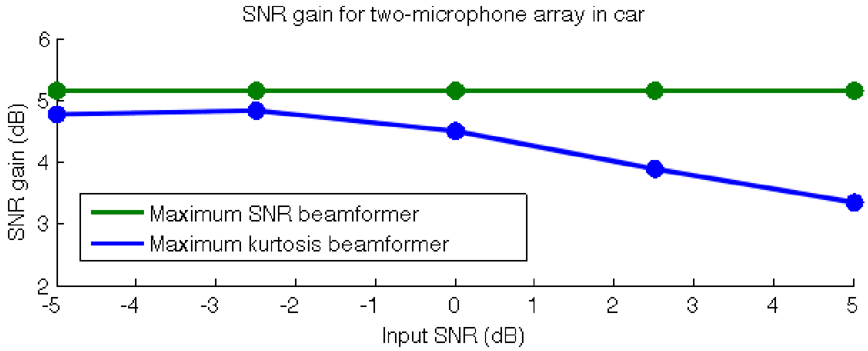

Figure 4.6: SNR gains for speech recorded in a car with highway noise.

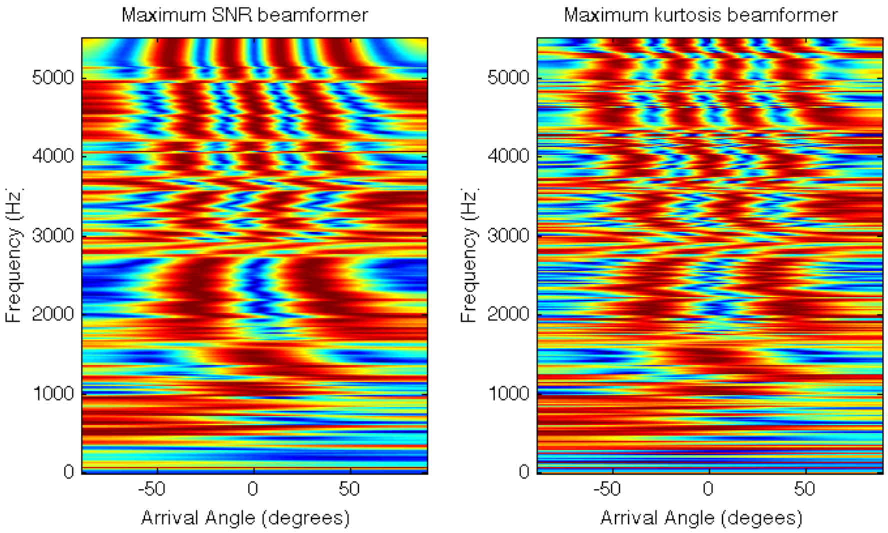

Figure 4.7: Beampattern comparison between the subband maximum-SNR beamformer and the subband maximum-kurtosis beamformer for speech recorded in a car with highway noise (Test 4).

4.3 Discussion

In Tests 1 and 2, the algorithm performed very well. This is no surprise, as kurtosis is often used with ICA in blind source-separation problems. Since both of these experiments tested a two-source, two-channel mixture, complete separation was to be expected. While the subband beamformer in Test 2 provided at least a 25 dB SNR increase, it did not entirely separate the signals, due to difficulties in the convergence of all frequency bins. For frequencies where speech power is low (due to the spectrum of the speech itself, or the channel response), the gradient contains little information about which reconstruction weights can maximize output kurtosis. This test exposes one difficulty in using adaptive frequency-domain algorithms: extreme care must be taken in ensuring the convergence of each bin.

Tests 3 and 4 are much more interesting, since they deal with real-world signals in very challenging environments. Test 3 exhibits the difficulty in extending a theoretically simple problem to the real world. A two-channel source separation should be possible if the sources are discrete, and two microphones are used to capture the sound. However, the reverberant room can be thought of as smearing the discrete sources into a nearly infinite number of directions. While the algorithm provided up to 7.6 dB SNR gain, a larger microphone array would allow more degrees of freedom, and better handle reverberant environments. The similar performance of the maximum-SNR beamformer illustrates the maximum-kurtosis beamformer's ability to estimate the optimal beam pattern.

In Test 4, the algorithm performed fairly well in reducing highway noise in an in-car speech recording. Gains of around 5 dB at -5dB input SNR are respectable when using a two-microphone array in diffuse noise. Again, the performance of the maximum-SNR beamformer highlights the maximum-kurtosis beamformer's ability to achieve similar results.

CHAPTER 5

ALTERNATIVE PROBLEM FORMULATION

In the previous sections, the gradient of kurtosis with respect to the reconstruction weights is calculated for use with a gradient-ascent algorithm. It is also possible to state the optimization problem in another form, resulting in a non-convex, homogeneous, quadratically constrained quadratic program (QCQP), a problem that can be approached with other mathematical tools. Although no closed-form solution has been found, other types of adaptive algorithms might be more easily developed after examining the problem in this alternative manner. A conversion of the problem to this standard mathematical form is presented in this section.

To simplify the expressions in this section, a constant value of two is added to the definition of complex kurtosis.

$$\kappa = \frac{E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right]}{\left(E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]\right)^2}$$

This equation can be expanded further, giving:

$$\kappa = \frac{E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]}{E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right]}$$

Using the factorizations presented in the last chapter, the kurtosis becomes:

$$\kappa = \frac{vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H E\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)}{vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H vec\left(E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\right) vec\left(E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\right)^H vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)} \tag{5.1}$$

Equation (5.1) is a ratio of quadratic forms in terms of $\boldsymbol{v} = vec(\boldsymbol{w}\boldsymbol{w}^H)$, and is commonly known as a Rayleigh quotient. Rayleigh quotients are of the form:

$$\kappa = \frac{\boldsymbol{v}^H\boldsymbol{A}\boldsymbol{v}}{\boldsymbol{v}^H\boldsymbol{B}\boldsymbol{v}}$$

where in this instance

$$\boldsymbol{A} = E\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right]$$

$$\boldsymbol{B} = vec\left(E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\right) vec\left(E\left[\boldsymbol{x}\boldsymbol{x}^H\right]\right)^H$$

$$\boldsymbol{v} = vec\big(\boldsymbol{w}\boldsymbol{w}^H\big)$$

For clarity, only the two-microphone case ($\boldsymbol{w}$ is a 2x1 vector) will be examined. This can be extended to larger dimensions if desired.

Generally, $\boldsymbol{A}$ is a 4x4 rank-three matrix, and $\boldsymbol{B}$ is a 4x4 rank-one matrix. Due to the conjugate symmetry of the outer products used in the construction of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{v}$, the second and third rows of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{v}$ are identical, as well as the second and third columns of $\boldsymbol{A}$ and $\boldsymbol{B}$. This allows the 4-dimensional problem to be converted to a 3-dimensional problem. Since the two middle dimensions always contribute equally to the kurtosis, the ratio can be redefined as

$$\kappa = \frac{\tilde{\boldsymbol{v}}^H \tilde{\boldsymbol{A}} \tilde{\boldsymbol{v}}}{\tilde{\boldsymbol{v}}^H \tilde{\boldsymbol{B}} \tilde{\boldsymbol{v}}}$$

where the middle two dimensions of $\boldsymbol{v}$ were summed to give

$$\tilde{\boldsymbol{v}} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \tilde{v}_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 + v_3 \\ v_4 \end{bmatrix}$$

and the redundant third dimension of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be removed to form $\tilde{\boldsymbol{A}}$ (now full rank) and $\tilde{\boldsymbol{B}}$ (still rank one).

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} A_{11} & A_{12} & A_{14} \\ A_{21} & A_{22} & A_{24} \\ A_{41} & A_{42} & A_{44} \end{bmatrix}$$

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} B_{11} & B_{12} & B_{14} \\ B_{21} & B_{22} & B_{24} \\ B_{41} & B_{42} & B_{44} \end{bmatrix}$$

If one wished to maximize κ over $\tilde{\boldsymbol{v}}$, choosing $\tilde{\boldsymbol{v}} \in Null(\tilde{\boldsymbol{B}})$ would cause κ to go to infinity. However, the maximizing $\boldsymbol{w}$ is desired, not the maximizing $\tilde{\boldsymbol{v}}$. This can be thought of as a requirement that $\boldsymbol{v}$ must always be representable as the vectorized outer product $vec(\boldsymbol{w}\boldsymbol{w}^H)$. In other words, applying the vec-transpose operator to re-order $\boldsymbol{v}$ as a 2x2 matrix should result in a rank one matrix, from which the optimal $\boldsymbol{w}$ can be obtained.

To formalize this requirement as a constraint, it can be noted that if $\boldsymbol{v}$ is to equal $vec(\boldsymbol{w}\boldsymbol{w}^H)$, then

$$\tilde{\boldsymbol{v}} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \tilde{v}_3 \end{bmatrix} = \begin{bmatrix} w_1^2 \\ 2w_1w_2 \\ w_2^2 \end{bmatrix}$$

and

$$\tilde{v}_1\tilde{v}_3 = \frac{1}{4}\tilde{v_2}^2$$

This constraint can be expressed in matrix form as

$$\tilde{\boldsymbol{v}}^H\boldsymbol{C}\tilde{\boldsymbol{v}} = 0$$

where

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & -\frac{1}{2} \\ 0 & \frac{1}{4} & 0 \\ -\frac{1}{2} & 0 & 0 \end{bmatrix}$$

In this way, the maximum-kurtosis beamformer can be found by solving the following optimization problem: maximize $\tilde{\boldsymbol{v}}^H\tilde{\boldsymbol{A}}\tilde{\boldsymbol{v}}$ subject to $\tilde{\boldsymbol{v}}^H\tilde{\boldsymbol{B}}\tilde{\boldsymbol{v}} = k$ and $\tilde{\boldsymbol{v}}^H\boldsymbol{C}\tilde{\boldsymbol{v}} = 0$. $\tilde{\boldsymbol{A}}$ is positive definite, $\tilde{\boldsymbol{B}}$ is positive semidefinite (of rank one), $\boldsymbol{C}$ is full rank but indefinite, and k is an arbitrary constant. Matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are known. This belongs to a family of problems known as homogeneous, quadratically constrained quadratic programs (QCQP).

Like the gradient-ascent algorithm derived earlier, the optimal $\boldsymbol{w}$ is optimal up to a scale factor, since the kurtosis surface is scale invariant (spherically symmetric). This ambiguity could technically be resolved with the additional constraint $||\boldsymbol{w}||_2^2 = 1$, but it is not a true constraint, since any $\boldsymbol{w}$ on the solution line can be projected back onto the unit sphere and remains optimal.

Methods for solving this equivalent problem were not explored and are open to further research.

CHAPTER 6

DISCUSSION

6.1 Issues and Future Work

One important consideration in using the maximum-kurtosis objective is that this method does not impose any distortionless constraints on the processed signal. Unlike the MVDR beamformer, which attempts to minimize the noise power constrained on a distortionless look direction, this algorithm allows distortion of the speech source in an attempt to maximize the kurtosis. It is unclear how the distortion affects intelligibility or perceived sound quality. In some instances, the distortion might be distracting to a human listener, or detrimental to the performance of a speech recognizer. In other cases, however, the additional noise reduction may provide additional intelligibility improvements at the expense of distortion. For example, applying a Wiener postfilter can improve SNR at the expense of distortion, suggesting that distortionless beamformers may not always be desired. This question is open to further research.

Another issue with using a maximum-kurtosis objective is convergence when multiple talkers are present. While the algorithm is designed under the assumption that one speech source is present in lower-kurtosis noise, real-world use surely yields situations where this is not the case. Depending on the environment and the attributes of each talker (distance from the array, signal amplitude, and frequency content), the behavior of the algorithm is unclear.

In a simulation using two talkers at equal amplitude, a local maximum was observed that would surely cause convergence issues for the algorithm. Figure 6.1 shows the output kurtosis with the weights constrained to the unit circle. Two maxima can be seen, potentially preventing the proposed algorithm from reaching the global maximum. The occurrence of situations like these in common applications is open to further research.

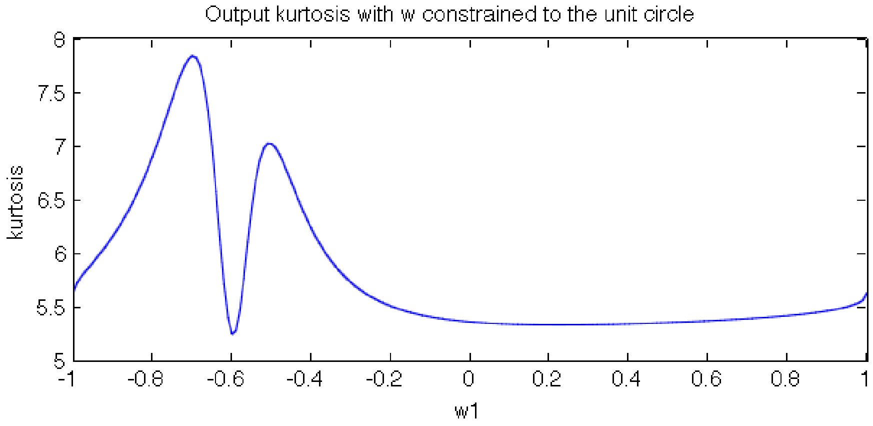

Figure 6.1: Kurtosis of an instantaneous mixture with two speech sources present.

One last consideration deals with the unit-norm constraint of the reconstruction weights. Suppose there are frequency bins containing no speech, only noise. These bins will probably not converge to any meaningful solution, and will contain only noise even after they have been weighted to maximize kurtosis. The unit-norm constraint forces noise-only bins to contribute to the output just as heavily as bins containing the desired speech signal. A postfilter can be used to attenuate bins with low SNR, but this attenuation could also be designed into the calculation of the beamformer weights themselves.

6.2 Contribution and Conclusion

In this thesis, we derive and implement a subband maximum kurtosis beamformer, showing SNR gains around 3.5 - 7.5 dB in real-world situations when used with a two-microphone linear array. Factoring the beamformer weights from the expected-value estimations has shown to greatly improve the convergence properties of the algorithm. Finally, an alternative formulation of the problem was developed, from which new approaches to the problem might be formulated.

APPENDIX A

EXPECTED-VALUE FACTORIZATIONS

$$
\begin{aligned}
&E\left[\left(\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right)^2\right] \\
&= E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\right] \\
&= E\left[\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right] \\
&= E\left[\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}\right] \\
&= E\left[\begin{bmatrix} w_1^*w_1 & w_1^*w_2 & w_2^*w_1 & w_2^*w_2 \end{bmatrix}\begin{bmatrix} x_1x_1^* \\ x_1x_2^* \\ x_2x_1^* \\ x_2x_2^* \end{bmatrix}\begin{bmatrix} w_1w_1^* & w_1w_2^* & w_2w_1^* & w_2w_2^* \end{bmatrix}\begin{bmatrix} x_1^*x_1 \\ x_1^*x_2 \\ x_2^*x_1 \\ x_2^*x_2 \end{bmatrix}\right] \\
&= E\left[\begin{bmatrix} w_1^*w_1 & w_1^*w_2 & w_2^*w_1 & w_2^*w_2 \end{bmatrix}\begin{bmatrix} x_1x_1^* \\ x_1x_2^* \\ x_2x_1^* \\ x_2x_2^* \end{bmatrix}\begin{bmatrix} x_1^*x_1 & x_1^*x_2 & x_2^*x_1 & x_2^*x_2 \end{bmatrix}\begin{bmatrix} w_1w_1^* \\ w_1w_2^* \\ w_2w_1^* \\ w_2w_2^* \end{bmatrix}\right] \\
&= E\left[vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\right] \\
&= vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)^H E\left[vec\left(\boldsymbol{x}\boldsymbol{x}^H\right) vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)^H\right] vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)
\end{aligned}
$$

$$E\left[\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\boldsymbol{w}\boldsymbol{w}^H\boldsymbol{x}\boldsymbol{x}^H\right]$$

$$= E\left[\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\right]$$

$$= E\left[\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}\begin{bmatrix} w_1^* & w_2^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\right]$$

$$= E\left[\begin{bmatrix} w_1^*w_1w_1^* & w_1^*w_1w_2^* & \cdots & w_2^*w_2w_2^* \end{bmatrix}\begin{bmatrix} x_1x_1^*x_1 \\ x_1x_1^*x_2 \\ \vdots \\ x_2x_2^*x_2 \end{bmatrix}\begin{bmatrix} x_1^* & x_2^* \end{bmatrix}\right]$$

$$= E\left[vec\left(\boldsymbol{w}\left(vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\right)^H\right)^H vec\left(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\right)\boldsymbol{x}^H\right]$$

$$= vec\left(\boldsymbol{w}\left(vec\left(\boldsymbol{w}\boldsymbol{w}^H\right)\right)^H\right)^H E\left[vec\left(\boldsymbol{x}\left(vec\left(\boldsymbol{x}\boldsymbol{x}^H\right)\right)^H\right)\boldsymbol{x}^H\right]$$

REFERENCES

[1] Y. Hu and P. C. Loizou, "A comparative intelligibility study of single-microphone noise reduction algorithms," *Journal of the Acoustical Society of America*, vol. 122, pp. 1777–1786, September 2007.

[2] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, 1979.

[3] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction Wiener filter," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1218–1234, 2006.

[4] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 504–512, 2001.

[5] Y. Hu and P. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 4, pp. 334–341, 2003.

[6] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *ASSP Magazine, IEEE*, vol. 5, no. 2, pp. 4–24, 1988.

[7] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[8] M. D. Kleffner and D. L. Jones, "Blind recovery of a speech source in noisy, reverberant environments," *Journal of the Acoustical Society of America*, submitted for publication.

[9] A. Hyvrinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, 2000.

[10] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1, pp. 21–34, 1998.

[11] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind

source separation," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 530–538, 2004.

[12] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 530–538, 2004.

[13] V. Reju, S. N. Koh, and I. Soon, "A robust correlation method for solving permutation problem in frequency domain blind source separation of speech signals," in *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, 2006, pp. 1891–1894.

[14] P. De Leon, "Short-time kurtosis of speech signals with application to co-channel speech separation," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 2, 2000, pp. 831–833 vol.2.

[15] Z. Ding and T. Nguyen, "Stationary points of a kurtosis maximization algorithm for blind signal separation and antenna beamforming," *Signal Processing, IEEE Transactions on*, vol. 48, no. 6, pp. 1587–1596, 2000.

[16] H. Li and T. Adali, "A class of complex ICA algorithms based on the kurtosis cost function," *Neural Networks, IEEE Transactions on*, vol. 19, no. 3, pp. 408–420, 2008.

[17] B. Sallberg, N. Grbic, and I. Claesson, "Online maximization of subband kurtosis for blind adaptive beamforming in realtime speech extraction," in *Digital Signal Processing, 2007 15th International Conference on*, 2007, pp. 603–606.

[18] Z. Yermeche, *Soft-Constrained Subband Beamforming for Speech Enhancement.* Department of Signal Processing, School of Engineering, Blekinge Institute of Technology, 2007.

[19] E. Ollila, "On the circularity of a complex random variable," *Signal Processing Letters, IEEE*, vol. 15, pp. 841–844, 2008.

[20] C. L. Nikias and A. P. Petropulu, *Higher Order Spectra Analysis. A Nonlinear Signal Processing Framework.* Prentice-Hall, 1993.

[21] E. Ollila, V. Koivunen, and H. Poor, "Complex-valued signal processing — essential models, tools and statistics," in *Information Theory and Applications Workshop (ITA), 2011*, 2011, pp. 1–10.

[22] S. C. Douglas, "Fixed-point algorithms for the blind separation of arbitrary complex-valued non-Gaussian signal mixtures," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 036525, 2007. [Online]. Available: http://asp.eurasipjournals.com/content/2007/1/036525

[23] E. Ollila and V. Koivunen, *Robust Estimation Techniques for Complex-Valued Random Vectors*. John Wiley & Sons, Inc., 2010, pp. 87–141. [Online]. Available: http://dx.doi.org/10.1002/9780470575758.ch2

[24] "ICA '99 synthetic benchmarks," 1999. [Online]. Available: http://sound.media.mit.edu/ica-bench/