TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING - SYSTEMS

Array Processing of Nonstationary Signals with Application to Speech

Thesis submitted for the degree "Doctor of Philosophy"

by

SHARON GANNOT

Submitted to the Senate of Tel-Aviv University

July, 2000

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING - SYSTEMS

Array Processing of Nonstationary Signals with Application to Speech

Thesis submitted for the degree "Doctor of Philosophy"

by

SHARON GANNOT

Submitted to the Senate of Tel-Aviv University

This research work was carried out at the
Ddepartment of Electrical Engineering - Systems,
Tel-Aviv University
Under the supervision of
Prof. Ehud Weinstein and Dr. David Burshtein

July, 2000

This work[1] was carried out under the supervision of

Prof. Ehud Weinstein and **Dr. David Burshtein**

[1] Thesis submitted in July, 2000. Revised in November 2000.

Acknowledgment

First, I wish to express my gratitude to my supervisors.
Many thanks to Dr. David Burshtein for his devoted guidance, his patience and his inspiring ideas.
Thanks to Prof. Ehud Weinstein who helped me enter the fascinating world of *Signal Processing* and for the privilege to be guided by him towards both my M.Sc and Ph.D degrees.

Warm thanks to Yona Eyal for making the computer network so simple and understandable and, especially, for her affability.

Thanks to Ziva Katz Bliberg, the department's administrator, for her warm hospitality since my first day in the department and for her assistance.

Thanks to prof. Meir Feder for understanding that nice environment is essential for fruitful research and for his helpful advice.

To the academic and administrative staff of the Electrical Engineering - Systems department, and particularly, the members of the Signal Processing lab along the years - thanks for creating such a pleasant working environment.

Thanks to my friends and colleagues - Dr. Arie Yeredor, Dr. Dani Seidner, Nadav Shulman and Uri Erez - with whom it was a privilege to work, for our fruitful discussions.

Warm thanks to my good friends - Dr. Agnès Cohen, Ofra Golani,
Nir Friedman and Ruth Grossmann - for their support and encouragement during the days the ending line of this work seemed to be beyond the horizon.

Warm thanks to my brother Dr. Israel Gannot for sharing his experience with me and for his wise advice.

Words can not express my warmest gratitude to my wonderful, supporting and understanding parents Yosef and Frieda Gannot. Deepest thanks for the help, the support and for planting the eager for knowledge.

Abstract

In this study we consider a sensor array located in a an enclosure. The array is used for enhancing a desired signal contaminated by interference.

Constrained minimum power adaptive beamforming, suggested by Frost, and specifically, the generalized sidelobe canceler (GSC) version, developed by Griffiths and Jim are the most widely used beamforming techniques. These methods rely on modeling the received signals at each sensor as differently delayed versions of the source signal. The high-quality interference suppression attained under this assumption is severely degraded in complicated environments, where more general transfer functions (TFs) are encountered.

In this work we begin by redeveloping Frost's algorithm (namely, a constrained minimum power adaptive beamformer) in the frequency domain. This formulation enables us to deal with a complicated TF in the same simple manner as Frost deals with delay-only arrays and to prove the optimality of the algorithm. We proceed, following the footsteps of Griffiths and Jim, by deriving a GSC version of the suggested algorithm, which requires knowledge of the TFs, or uses their estimated values. We then show that a practical algorithm can be implemented by substituting estimation of the entire TFs with estimates of the ratios between the different TFs. Three blocks constitute the suggested algorithm. A fixed beamformer, a reference noise constructor and a multi-channel noise canceller. The first two depend on the ratio of the TFs.

The suggested algorithm can be regarded as an extension of the original Griffiths and Jim algorithm for the case of arbitrary TFs.

An unbiased estimate of the TFs' ratio is attained by exploiting the desired signals' nonstationarity.

An alternative approach can be taken by observing that the desired signal and the reference noise signals are uncorrelated. This property can be exploited in estimating the TFs' ratio. As a byproduct of this approach, we also obtain

a reduction in the self-cancellation phenomena, due to reduction of the desired signal leakage into the reference noise signals.

The derivation of the algorithm is followed by analytical evaluation of the expected performance as a function of the TFs involved and the noise field.

The algorithm is applied to the problem of speech enhancement in a noisy and reverberating room with several noise fields (such as point, diffused or uncorrelated noise fields) in a large variety of noise levels.

The discussion is supported by an experimental study using speech and noise signals recorded in an actual room acoustics environment. The performance evaluation consists of the assessment of sound sonograms, signal-to-noise ratio (SNR) enhancement, noise reduction level a well as informal subjective listening tests.

Comparison with the conventional Griffiths and Jim algorithm demonstrates a clear advantage of the suggested algorithm in all aspects and in the entire input SNR range.

The computational burden and memory requirements of both algorithms, on the other hand, are approximately equivalent.

It has shown, that further improvement can be attained by applying a single microphone speech enhancement algorithm as a post-processor at the output of the multi-microphone algorithm. As a supplement to this work, we derived two such single microphone algorithms, one in the frequency domain and the other in the time domain. Overall SNR improvement of up to 25dB might be achieved by applying the multi-microphone algorithm followed by the single microphone post-processor (frequency domain version) for point source noise field.

Contents

List of Figures

List of Tables

Chapter 1

Preface

1.1 Introduction and Background

In this study we consider a sensor array located in an enclosure, where general transfer functions (TFs) relate the source signals and the sensors. The array is used either to enhance a signal contaminated by interference or to separate several signals from their received mixture. The general problem may be stated in the following manner,

$$z_m(t) = \sum_{k=1}^{K} a_{mk}(t) * s_k(t) + \sum_{l=1}^{L} b_{ml}(t) * v_l(t) + w_m(t) \; ; \; m = 1, \ldots, M \qquad (1.1)$$

where, $z_m(t);\ m = 1, \ldots, M$ are the received signals at the sensors, $s_k(t);\ k = 1, \ldots, K$ are the desired signals, $v_l(t);\ l = 1, \ldots, L$ are some point source interference signals, and $w_m(t);\ m = 1, \ldots, M$ are sensor (or ambient) noise signals. It is assumed that $M \geq K$, i.e., the number of sensors is larger than the number of desired signals. We assume that we can rank the signals in terms their level of stationarity. The desired signals $s_k(t)$ are highly non-stationary. The interference signals' ($v_l(t)$) statistics change more slowly. The ambient noise sources $w_m(t)$ are assumed to be stationary. The involved TFs, $a_{mk}(t)$, $b_{ml}(t)$ are assumed to be slow changing linear systems, i.e. their change rate is slower than the rate of change of the interference statistics. Note, that by this general model we can handle several important problems. When $K > 1$, the problem becomes the separation problem or the echo cancellation problem, if only one signal is desired.

The problem of signal enhancement is concerned when a single signal ($K = 1$) is desired. The noise signal is determined to be a point source if $L \geq 1$ and $w_m(t) = 0;\ m = 1, \ldots, M$. The noise source is assumed to be spatially extended if $L = 0$ and $w_m(t)$ exist for $m = 1, \ldots, M$. Of course, any noise field can be viewed as a combination of these two possibilities. For $M = 1$ we have the single sensor signal enhancement problem.

In this study we will primarily address the problem of a single desired signal in any noise field, received by multiple sensors ($M > 1$). The problem of single sensor enhancement will be mentioned in the context of application of the suggested methods to the speech signal.

This problem was extensively addressed in the literature and in the industry in numerous papers.

An important series of works is referred to as *blind signal separation* (BSS). BSS consists of recovering unobserved signals from several observed mixtures. The simplest model assumes the existence of an equal number of sensors and sources, with the mixtures being linear and instantaneous. Cardoso [1] gives a thorough review of the approaches developed to address this problem. The basic model can be extended. Considering the problem of more sensors than sources brings us to the framework of *beamforming*. Another extension is the problem of convolutive mixtures.

In this paper we will concentrate on the *beamformer*-based approaches. We will primarily deal with the problem of convolutive mixtures, where the number of sensors is larger than the number of desired signals.

A beamformer is a processor constructed of an array of sensors which performs spatial filtering. The beamformer can be applied to a variety signals, e.g. sonar, radar, communication, geophysical, biomedical and speech.

A simple beamformer is implemented by the *delay & sum* approach, where the signals at the sensors are aligned and summed coherently. This fixed beamformer structure is quite limited.

The most versatile beamformers are the *adaptive arrays*. Van Veen and Buck-

ley [2], in their tutorial review on beamforming, and Cox *et al.* [3], in their thorough overview of optimization criteria for adaptive beamformers summarize a large number of these works and applications.

Constrained minimum power adaptive beamforming suggested by Frost [4] deals with the problem of a broadband signal received by an array, where only a delay exists between sensors. Each sensor signal is processed by a tap delay line after applying a proper time delay compensation. The algorithm is able to maintain a chosen frequency response in the look direction, while minimizing output noise power by a constrained minimization of the total output power. This minimization is achieved by adjusting the filters' taps under the desired constraint. Frost developed a constrained LMS-type algorithm. Griffiths and Jim [5] considered Frost's algorithm and introduced the generalized sidelobe canceller (GSC) approach. The constrained beamformer is divided into three branches. The first is a fixed beamformer, which maintains the desired constraint. The second is a blocking matrix, which blocks the desired signal (for instance, by subtracting pairs of aligned signals), thus constituting a noise-only reference signal. The third is an unconstrained LMS-type branch, which cancels the noise in the fixed beamformer output. In [5] it is shown that Frost's algorithm can be viewed as a special case of the suggested GSC. The main drawback of the algorithm is its delay-only propagation assumption, rather than general transfer function (TF) propagation. Van Veen and Buckley [2] introduce a wider range of constraints on the beam pattern than suggested by Frost and Griffiths and Jim. Cox *et al.* [3] also suggest constraining the norm of the adaptive canceller coefficients to solve the *super-directivity* problem, i.e., its sensitivity to steering errors. Cox *et al.* suggest updating Frost's (or Griffiths and Jim's) algorithm by applying a quadratic constraint on the norm of the noise canceller coefficients. This constraint, which can limit the super-directivity, is added to the usual linear constrains.

The GSC concept, was adopted to the field of speech enhancement in a reverberating environment, and generally, with modifications to the different blocks.

Hoshuyama *et al.* [6],[7],[8] use a three-block structure structure to the GSC,

but modify the blocking matrix to an adaptive structure. Desired signal leakage, which distorts the output signal, is reduced by introducing a quadratic constraint on the norm of the noise canceller coefficients, or by using the equivalent leaky LMS procedure (for the equivalence see [9]). A real-time implementation of the norm coefficient-constrained noise canceller is introduced by Hoshuyama and Sugiyama [10].

Nordholm *et al.* [11] use the GSC structure, but replace the blocking matrix by a more complicated spatial high-pass filtering, yielding more accurate noise-only reference signals. Meyer and Sydow [12] construct the noise-only reference signals by steering the lobes of a multi-beam beamformer towards the noise and and desired signal directions separately.

Widrow and Stearns [13] suggest a dual structure beamformer. The *master* beamformer adapts its coefficients to minimize its output power while maintaining the beam-pattern towards a pre-determined *pilot* signal from the desired direction. Those coefficients are continuously copied to a *slave* beamformer, whose inputs are only the real-time inputs (without pilot). Based on this concept Dahl *et al.* [14] suggest a practical dual beamformer for use in a car environment with both noise and jammer signals (e.g., loudspeaker). The pilot signal is constructed by offline recording of the jammer signal and the desired signal in the actual car environment, during a calibration phase. Thus, both echo cancellation and car noise suppression are achieved simultaneously.

Analysis of the GSC structure is introduced by Bitzer *et al.* [15]. The dependency on the noise field is addressed. They showed that the noise reduction might be infinitely large, when the noise source is directional, but that in the more practical situation such as a reverberant enclosure, when the noise field can be regarded as diffused, performance is severely degraded.

A family of algorithms use a beamformer approach followed by a post-processor. The post-processor works in conjunction with the beamformer by using its outcome. Zelinski [16] suggested the use of a Wiener filter, implemented in the time domain, which performs an optimal *minimum mean square error* (MMSE) fil-

ter between the received signals and the D&S beamformer output. Meyer and Simmer [17] show that when high coherence between microphone signals is encountered (Dal-Degan and Prati [18] indicate that this is the situation in low-frequencies), performance of both the D&S beamformer and the Wiener post-filter is severely degraded. To overcome this problem, they suggest the use of a spectral subtraction algorithm in the low frequency band and a Wiener filter (similar to Zelinski's suggestion, but implemented in the frequency domain) in the high frequency band. Fischer and Kammeyer [19] suggested further splitting the microphone array into differentially equi-spaced sub-arrays, instead of using the conventional D&S array, thus reducing the coherence between sensors at the lower frequencies. A thorough and comprehensive study of the structure is given by Marro *et al.* [20].

It is suggested by Bitzer *et al.* [21], to use the GSC structure but with fixed Wiener filters in the noise cancelling branch and further post-filters at the GSC output. An improved performance in the lower frequency range is achieved. It is shown in [22] that the noise canceller, implemented by the Wiener filters, can be computed in advance, by using prior knowledge of the noise field. The dependency of this structure on the noise field is also analyzed in a series of papers, e.g. [23],[24] (for the two channel case), [15],[25] and [26].

We note that the post-processing approach can be adopted, but by treating the beamformer and the single channel post-processor separately. In this manner we can apply any applicable single channel enhancer to the output of the multi-sensor algorithm to further improve the noise reduction. When applied to speech signal, a variety of single microphone speech enhancers exists. Two novel single microphone speech enhancers are suggested in this paper, and presented in Chapter 7. One is a time domain method [27],[28], and the other is a frequency-domain method [29],[30]. The latter is described in Appendix D.

Doclo and Moonen [31] suggest the use of an SVD-based optimal Wiener filter. They apply their algorithm to the single-microphone and multi-microphone speech enhancement problems.

Jan and Flanagan [32] address the application of a fixed beamformer structure to speech signals. They suggest the use of matched filter beamforming (MFBF) instead of conventional D&S beamforming (DSBF). In the MFBF structure, signal alignment is achieved by convolving the microphone signals with the (reversed) acoustic transfer functions (ATF) filters. If the ATFs are unknown, an estimate should be used. Rabinkin *et al.* [33] prove that the performance of this structure is superior to simpler DSBF in a simple room environment, when several arrivals of the desired signal are encountered due to reverberation. Nevertheless, they suggest truncating the ATFs' length, to avoid the use of erroneous estimates. The MFBF approach is adopted in several works in the fixed beamformer branch of GSC structured arrays.

A series of works by Grenier *et. al.* [34], [35], [36], [37], [38], [39], [40], [41] deal with the GSC structure. First a constant position talker in a delay only environment is addressed [35]. Eigenvector constraints are used for widening the beam pattern to cope with the uncertainty of the talker and microphones positions. Movement of the talker is dealt via LMS-like target tracking [36],[38], which is based on the subspace tracking procedure suggested by Yang [42]. The resulting ATFs are then structure fitted to become a legal steering vector. Generalization to the wide-band case is given in [37]. The problem of multi-source beamforming and multi-target tracking is addressed in [40]. The most applicable work for our problem, when general ATFs relate the source and the sensors, rather than a simple time delay, is given in [41]. Again, a subspace tracking is performed. The resulting transfer function is constrained to the array manifold, by assuming FIR model and small movements of the talker. The fixed beamformer branch of the GSC structure is implemented by MFBF.

The more general problem of a microphone array receiving desired speech, noise signal and acoustic echo is treated in several papers. The existence of an echo signal for double talk situations is treated in [39], where a joint identification of the acoustic paths from the desired speech and echo signal is implemented via a dual subspace tracking procedure. Kellermann [43] suggests incorporating an

acoustic echo canceller (AEC) into a time-invariant beamformer, while activating some control mechanism. Both noise reduction and echo cancellation in car environment is achieved by the dual phase beamformer, already introduced [14].

1.2 Thesis Outline

In this work we first derive Frost's [4] algorithm (namely, a constrained minimum power beamformer) in the frequency domain. We proceed by deriving the GSC structure of the suggested algorithm, which can be regarded as an extension to Griffiths and Jim [5] algorithm for the general TF case, rather than the delay-only case. This frequency domain formulation serves as a proof of optimality of the intuitive structure suggested by other authors to deal with the beamforming problem in reverberant environments. We then show, that a sub-optimal algorithm can be implemented, by estimating the ratios between the TFs, instead of estimating the actual TFs. The nonstationarity of the desired signal is exploited in this work to improve the estimation accuracy of the TFs ratio. We do so by extending the nonstationarity principle, suggested by Shalvi and Weinstein [44]. The *decorrelation criterion*, suggested by Weinstein *et al.* [45], is used as an alternative approach for estimating the TFs ratio and to deal with the problem of desired signal leakage to the reference noise signals.

The derivation of the suggested algorithm is followed by an analysis of the expected performance and its dependence on the TFs involved, on the noise field and the estimation accuracy. The algorithm is applied to the problem of speech enhancement in a reverberating room. The discussion is supported by an experimental study using speech and noise signals recorded in an actual room acoustics environment. The performance evaluation consists of the assessment of sound sonograms, signal to noise ratio (SNR) enhancement and informal subjective listening tests. The suggested method can also be found in [46].

The paper is organized as follows. In Chapter 2 we reformulate the problem in the frequency domain. The frequency domain derivation of the constrained power

minimization is presented in Chapter 3. The TFs' ratio identification problem is addressed in Chapter 4, using the signal nonstationarity, and in Chapter 5, using the decorrelation criterion. Chapter 6 is devoted to analytical evaluation of the expected performance. Chapter 8 presents simulation results under real conditions. The single microphone speech enhancement algorithm (MIXMAX), which serves as a post-processor to the multi-microphone algorithm is presented in Appendix D.

Chapter 2

Problem Formulation

In this chapter we will address the problem of a single desired signal, received by a sensor array and contaminated by some noise signals. This important problem is a special case of the general problem introduced in Eq. 1.1.

Consider an array of sensors in a noisy environment, as shown in Figure 2.1. The received signal is comprised of two components. The first is some nonstationary (e.g., speech) signal. The second is some stationary interference signal. Our goal is to reconstruct the nonstationary signal component from the received signals. We use the following notation. $z_m(t)$ is the m-th sensor signal. $s(t)$ is the desired signal source. $n_m(t)$ is the interference signal of the m-th sensor. $n_m(t)$ is comprised of some point source noise component and some ambient noise component (combination of the two types of noise sources introduced in Eq. 1.1). $a_m(t)$ are time-varying transfer functions (TFs) from the desired speech source to the m-th sensor. We have,

$$z_m(t) = a_m(t) * s(t) + n_m(t) \; ; \; m = 1, \ldots, M \tag{2.1}$$

where $*$ denotes convolution. Suppose that the analysis frame duration T is chosen such that the signal may be considered stationary over the analysis frame. Typically, the TFs change slowly over time, so that they may also be considered time invariant over the analysis frame. Multiplying both sides of Equation (2.1) by a rectangular window function $w(t)$ ($w(t) = 1$ over the analysis frame, $w(t) = 0$

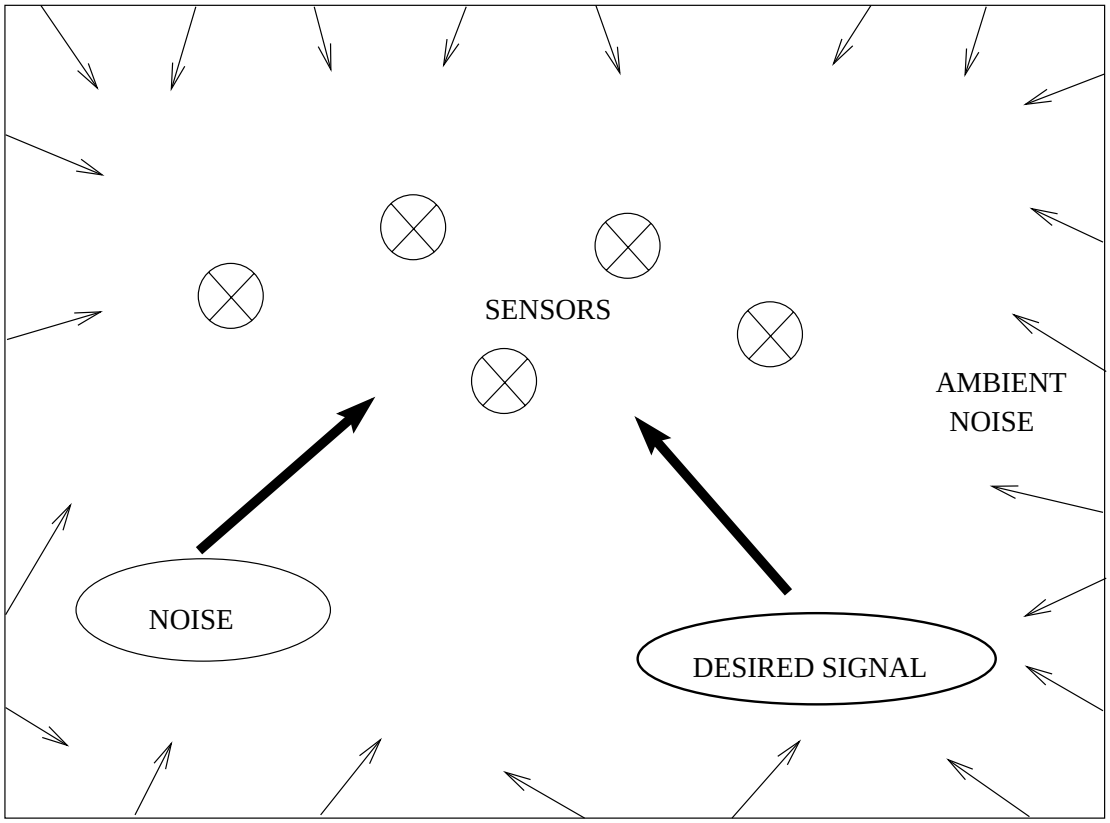

Figure 2.1: An array of sensors in a noisy environment.

otherwise) and applying the DTFT operator, yields,

$$Z_m(t, e^{j\omega}) \approx A_m(e^{j\omega})S(t, e^{j\omega}) + N_m(t, e^{j\omega}) \; ; \; m = 1, \ldots, M. \tag{2.2}$$

The approximation is justified for T sufficiently large. $Z_m(t, e^{j\omega})$, $S(t, e^{j\omega})$ and $N_m(t, e^{j\omega})$ are the the short time Fourier transforms (STFT) of the respective signals. $A_m(e^{j\omega})$ is the transfer function of the m-th sensor. Note that we have assumed that the TFs are time invariant. The vector formulation of the equation set (2.2) is

$$\mathbf{Z}(t, e^{j\omega}) = \mathbf{A}(e^{j\omega})S(t, e^{j\omega}) + \mathbf{N}(t, e^{j\omega}), \tag{2.3}$$

where

$$\mathbf{Z}^T(t, e^{j\omega}) \;\; = \;\; \left[\; Z_1(t, e^{j\omega}) \quad Z_2(t, e^{j\omega}) \quad \cdots \quad Z_M(t, e^{j\omega}) \; \right]$$

$$\begin{aligned} \mathbf{A}^T(e^{j\omega}) &= \left[\begin{array}{cccc} A_1(e^{j\omega}) & A_2(e^{j\omega}) & \cdots & A_M(e^{j\omega}) \end{array} \right] \\ \mathbf{N}^T(t, e^{j\omega}) &= \left[\begin{array}{cccc} N_1(t, e^{j\omega}) & N_2(t, e^{j\omega}) & \cdots & N_M(t, e^{j\omega}) \end{array} \right]. \end{aligned}$$

Based on this problem formulation, in the next chapter we will derive the constrained output power minimization, which serves as an extension of Frost [4] and GSC [5] algorithms.

Chapter 3

Algorithm Derivation

In [4] a beamforming algorithm was proposed under the assumption that the transfer function from the desired signal source to each sensor includes only gain and delay values. In this chapter we consider the general case of arbitrary transfer functions. By following the derivation of [4] in the frequency domain, we derive a beamforming algorithm for the general TF case. In Section 3.1 we derive a frequency domain Frost algorithm. First we obtain a closed form, linearly constrained, minimum variance beamformer. Then we derive an adaptive solution. The outcome will be a constrained LMS-type algorithm. In Section 3.2 we proceed, following the footsteps of Griffiths and Jim [5], and formulate an unconstrained adaptive solution, which will give a GSC interpretation to the frequency domain Frost algorithm. During the algorithm derivation, we will assume, that the TFs are known. Later, in Chapter 4 and Chapter 5, we address the problem of estimating the TFs.

3.1 Frequency Domain Frost Algorithm

3.1.1 Optimal Solution

Let $W^*(t, e^{j\omega})\ ; m = 1, \ldots, M$ be a set of M filters.

$$\mathbf{W}^{\dagger}(t, e^{j\omega}) = \left[\begin{array}{cccc} W_1^*(t, e^{j\omega}) & W_2^*(t, e^{j\omega}) & \cdots & W_M^*(t, e^{j\omega}) \end{array} \right].$$

A beamformer is realized by filtering each sensor output by $W^*(t, e^{j\omega})$; $m = 1, \ldots, M$ and summing the outputs (as also shown in Figure 3.1):

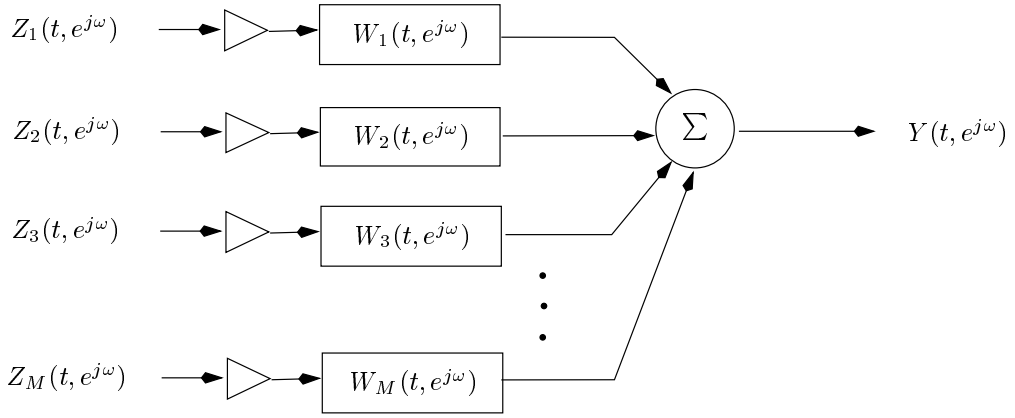

Figure 3.1: An M sensors wide-band beamformer.

$$\begin{aligned} Y(t, e^{j\omega}) &= \mathbf{W}^\dagger(t, e^{j\omega})\mathbf{Z}(t, e^{j\omega}) \\ &= \mathbf{W}^\dagger(t, e^{j\omega})\mathbf{A}(e^{j\omega})S(t, e^{j\omega}) + \mathbf{W}^\dagger(t, e^{j\omega})\mathbf{N}(t, e^{j\omega}) \\ &\triangleq Y_s(t, e^{j\omega}) + Y_n(t, e^{j\omega}), \end{aligned} \tag{3.1}$$

where $Y_s(t, e^{j\omega})$ is the desired signal part and $Y_n(t, e^{j\omega})$ is the noise part. The output power of the beamformer is

$$\begin{aligned} E\{Y(t, e^{j\omega})Y^*(t, e^{j\omega})\} &= E\{\mathbf{W}^\dagger(t, e^{j\omega})\mathbf{Z}(t, e^{j\omega})\mathbf{Z}^\dagger(t, e^{j\omega})\mathbf{W}(t, e^{j\omega})\} \\ &= \mathbf{W}^\dagger(t, e^{j\omega})S_{zz}(t, e^{j\omega})\mathbf{W}(t, e^{j\omega}), \end{aligned}$$

where $S_{zz}(t, e^{j\omega}) \triangleq E\{\mathbf{Z}(t, e^{j\omega})\mathbf{Z}^\dagger(t, e^{j\omega})\}$. We want to minimize the output power subject to the following constraint on $Y_s(t, e^{j\omega})$:

$$Y_s(t, e^{j\omega}) = \mathbf{W}^\dagger(t, e^{j\omega})\mathbf{A}(e^{j\omega})S(t, e^{j\omega}) = \mathcal{F}^*(t, e^{j\omega})S(t, e^{j\omega}),$$

where $\mathcal{F}^*(t, e^{j\omega})$ is some pre-specified filter (usually a simple delay). We thus have the following minimization problem,

$$\min_{\mathbf{W}} \left\{ \mathbf{W}^\dagger(t, e^{j\omega})S_{zz}(t, e^{j\omega})\mathbf{W}(t, e^{j\omega}) \right\} \text{ subject to } \mathbf{W}^\dagger(t, e^{j\omega})\mathbf{A}(e^{j\omega}) = \mathcal{F}^*(t, e^{j\omega}). \tag{3.2}$$

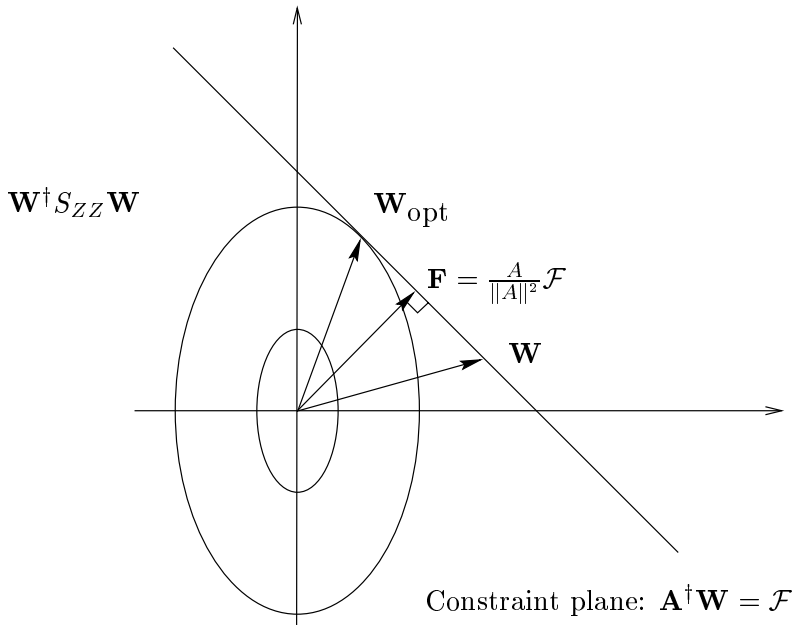

Figure 3.2: Constrained minimization of output power.

The minimization (3.2) is demonstrated in Figure 3.2. The point, where the equi-power contours are tangent to the constraint plane, is the optimum vector of beamforming filters. The distance, $\mathbf{F}$, of the origin from the constraint plane will be calculated in Section 3.1.2.

To solve (3.2) we first define the following complex Lagrange multiplier,

$$\begin{aligned}\mathcal{L}(\mathbf{W}) &= \mathbf{W}^\dagger(t,e^{j\omega})S_{zz}(t,e^{j\omega})\mathbf{W}(t,e^{j\omega}) + \\ &\quad \lambda\left[\mathbf{W}^\dagger(t,e^{j\omega})\mathbf{A}(e^{j\omega}) - \mathcal{F}^*(t,e^{j\omega})\right] + \lambda^*\left[\mathbf{A}^\dagger(t,e^{j\omega})\mathbf{W}(e^{j\omega}) - \mathcal{F}(t,e^{j\omega})\right].\end{aligned}$$

Setting the derivative with respect to $\mathbf{W}^*$ to 0 (e.g. [47] and Appendix B) yields,

$$\nabla_{\mathbf{W}^*}\mathcal{L}(W) = S_{zz}(t,e^{j\omega})\mathbf{W}(t,e^{j\omega}) + \lambda\mathbf{A}(e^{j\omega}) = 0.$$

Now, recalling the constraint in (3.2), we obtain the following set of optimal filters:

$$\mathbf{W}^{\text{opt}}(t,e^{j\omega}) = [\mathbf{A}^\dagger(e^{j\omega})S_{zz}^{-1}(t,e^{j\omega})\mathbf{A}(e^{j\omega})]^{-1}S_{zz}^{-1}(t,e^{j\omega})\mathbf{A}(e^{j\omega})\mathcal{F}(e^{j\omega}).$$

This closed form solution is difficult to implement, and does not have the ability to track changes in the environment. Therefore an adaptive solution should be more useful.

3.1.2 Adaptive Solution

Consider the following steepest descent, adaptive algorithm:

$$\begin{aligned}\mathbf{W}(t+1,e^{j\omega}) &= \mathbf{W}(t,e^{j\omega}) - \mu\nabla_{\mathbf{W}^*}\mathcal{L}(e^{j\omega}) \\ &= \mathbf{W}(t,e^{j\omega}) - \mu\left[S_{zz}(t,e^{j\omega})\mathbf{W}(t,e^{j\omega}) + \lambda\mathbf{A}(e^{j\omega})\right].\end{aligned}$$

Imposing our constraint on $\mathbf{W}(t+1,e^{j\omega})$ yields,

$$\begin{aligned}\mathcal{F}(e^{j\omega}) = \mathbf{A}^\dagger(e^{j\omega})\mathbf{W}(t+1,e^{j\omega}) = \mathbf{A}^\dagger(e^{j\omega})\mathbf{W}(t,e^{j\omega}) - \\ \mu\mathbf{A}^\dagger(e^{j\omega})S_{zz}(t,e^{j\omega})\mathbf{W}(t,e^{j\omega}) - \mu\mathbf{A}^\dagger(e^{j\omega})\mathbf{A}(e^{j\omega})\lambda.\end{aligned}$$

Solving for the Lagrange multiplier and applying further rearrangement of terms yields:

$$\mathbf{W}(t+1,e^{j\omega}) = P(e^{j\omega})\mathbf{W}(t,e^{j\omega}) - \mu P(e^{j\omega})S_{zz}(t,e^{j\omega})\mathbf{W}(t,e^{j\omega}) + \mathbf{F}(e^{j\omega}),$$

where

$$P(e^{j\omega}) = I - \frac{\mathbf{A}(e^{j\omega})\mathbf{A}^\dagger(e^{j\omega})}{\|\mathbf{A}(e^{j\omega})\|^2} \tag{3.3}$$

and

$$\mathbf{F}(e^{j\omega}) = \frac{\mathbf{A}(e^{j\omega})}{\|\mathbf{A}(e^{j\omega})\|^2}\mathcal{F}(e^{j\omega}). \tag{3.4}$$

Further simplification can be achieved by replacing $S_{zz}(t,e^{j\omega})$ by its instantaneous estimator, $\mathbf{Z}(t,e^{j\omega})\mathbf{Z}^\dagger(t,e^{j\omega})$, and recalling (3.1). We thus obtain,

$$\mathbf{W}(t+1,e^{j\omega}) = P(e^{j\omega})\left[\mathbf{W}(t,e^{j\omega}) - \mu\mathbf{Z}(t,e^{j\omega})Y^*(t,e^{j\omega})\right] + \mathbf{F}(e^{j\omega}).$$

3.1.3 Geometrical Interpretation

Consider the null space of $\mathbf{A}(e^{j\omega})$, defined by

$$\mathcal{N}(e^{j\omega}) = \left\{\mathbf{W} \mid \mathbf{A}^\dagger(e^{j\omega})\mathbf{W} = 0\right\}.$$

The constraint hyper-plane,

$$\Lambda(e^{j\omega}) = \left\{ \mathbf{W} \mid \mathbf{A}^{\dagger}(e^{j\omega})\mathbf{W} = \mathcal{F}(e^{j\omega}) \right\}$$

is parallel to $\mathcal{N}(e^{j\omega})$. In addition, let

$$\mathcal{R}(e^{j\omega}) = \left\{ \kappa \mathbf{A}(e^{j\omega}) \mid \text{ for any real } \kappa \right\}$$

be the column space. By the Fundamental theorem of Linear Algebra (e.g., [48]), $\mathcal{R}(e^{j\omega}) \perp \mathcal{N}(e^{j\omega})$. In particular, $\mathbf{F}(e^{j\omega})$ is perpendicular to $\mathcal{N}(e^{j\omega})$, since $\mathbf{F}(e^{j\omega}) = \frac{\mathcal{F}(e^{j\omega})}{||\mathbf{A}(e^{j\omega})||^2}\mathbf{A}(e^{j\omega}) \in \mathcal{R}(e^{j\omega})$. Furthermore,

$$\mathbf{A}^{\dagger}(e^{j\omega})\mathbf{F}(e^{j\omega}) = \mathbf{A}^{\dagger}(e^{j\omega})\mathbf{A}(e^{j\omega}) \left(\mathbf{A}^{\dagger}(e^{j\omega})\mathbf{A}(e^{j\omega}) \right)^{-1} \mathcal{F}(e^{j\omega}) = \mathcal{F}(e^{j\omega}).$$

Thus, $\mathbf{F}(e^{j\omega}) \in \Lambda(e^{j\omega})$, and $\mathbf{F}(e^{j\omega}) \perp \Lambda(e^{j\omega})$. Hence, $\mathbf{F}(e^{j\omega})$ is the perpendicular from the origin to the constraint hyper-plane, $\Lambda(e^{j\omega})$. The matrix $P(e^{j\omega})$, defined in (3.3), is the "projection matrix" to the null space of $\mathbf{A}(e^{j\omega})$, $\mathcal{N}(e^{j\omega})$.

Under this terminology, the sequential update formula can be interpreted in the following manner. We first form an initial LMS update to the filters,

$$\tilde{\mathbf{W}}(t+1, e^{j\omega}) = \mathbf{W}(t, e^{j\omega}) - \mu \mathbf{Z}(t, e^{j\omega}) Y^{*}(t, e^{j\omega}).$$

Next, we project $\tilde{\mathbf{W}}(t+1, e^{j\omega})$ into the null space of $\mathbf{A}(e^{j\omega})$, and add the result with $\mathbf{F}(e^{j\omega})$:

$$\mathbf{W}(t+1, e^{j\omega}) = P(e^{j\omega})\tilde{\mathbf{W}}(t+1, e^{j\omega}) + \mathbf{F}(e^{j\omega}).$$

The resulting vector of updated filters, $\mathbf{W}(t+1, e^{j\omega})$ lies in the constraint hyper-plane, $\Lambda(e^{j\omega})$. The constrained LMS algorithm is summarized in Figure 3.3 and displayed graphically in Figure 3.4.

3.2 Generalized Sidelobe Canceller (GSC) Interpretation

In [5], Griffiths and Jim considered the case where each TF is a delay element (with some gain). Griffiths and Jim obtained an unconstrained adaptive enhancement algorithm, using the same constrained, minimum output power criterion

$$\mathbf{W}(t=0, e^{j\omega}) = \mathbf{F}(e^{j\omega})$$

$$\mathbf{W}(t+1, e^{j\omega}) = P(e^{j\omega})\left[\mathbf{W}(t, e^{j\omega}) - \mu \mathbf{Z}(t, e^{j\omega}) Y^*(t, e^{j\omega})\right] + \mathbf{F}(e^{j\omega})$$

$$t = 0, 1, \ldots$$

($P(e^{j\omega})$ and $\mathbf{F}(e^{j\omega})$ are defined by (3.3) and (3.4)).

Figure 3.3: Frequency domain Frost algorithm.

used by Frost [4]. The unconstrained algorithm is computationally more efficient than the constrained algorithm. Furthermore, the unconstrained algorithm is based on the well-behaved NLMS scheme. In Section 3.1.2 we obtained an adaptive algorithm for the case where each TF is represented by an arbitrary linear time invariant system, by tracing the derivation of Frost in the frequency domain. We now repeat the arguments of Griffiths and Jim for our case (arbitrary TFs), and derive an unconstrained adaptive enhancement algorithm.

Consider some vector in linear space. This vector can be split uniquely into a sum of two vectors in mutually orthogonal sub-spaces (e.g., [48]). Hence,

$$\mathbf{W}(t, e^{j\omega}) = \mathbf{W}_0(t, e^{j\omega}) - \mathbf{V}(t, e^{j\omega}), \tag{3.5}$$

where $\mathbf{W}_0(t, e^{j\omega}) \in \mathcal{R}(e^{j\omega})$ and $-\mathbf{V}(t, e^{j\omega}) \in \mathcal{N}(e^{j\omega})$. By the definition of $\mathcal{N}(e^{j\omega})$,

$$\mathbf{V}(t, e^{j\omega}) = \mathcal{H}(e^{j\omega})\mathbf{G}(t, e^{j\omega}) \tag{3.6}$$

where $\mathcal{H}(e^{j\omega})$ is some $M \times (M-1)$ matrix, such that the columns of $\mathcal{H}(e^{j\omega})$ span the null space of $\mathbf{A}(e^{j\omega})$, i.e.,

$$\mathbf{A}^\dagger(e^{j\omega})\mathcal{H}(e^{j\omega}) = 0 \qquad \text{rank}\, \mathcal{H}(e^{j\omega}) = M - 1. \tag{3.7}$$

The vector $\mathbf{G}(t, e^{j\omega})$ is an $(M-1) \times 1$ vector of adjustable filters. By the geometrical interpretation of Frost's algorithm,

$$\mathbf{W}_0(t, e^{j\omega}) = \mathbf{F}(e^{j\omega}) = \frac{\mathbf{A}(e^{j\omega})}{\|\mathbf{A}(e^{j\omega})\|^2}\mathcal{F}(e^{j\omega}). \tag{3.8}$$

(Recall that $\mathbf{F}(e^{j\omega})$ is the perpendicular from the origin to the constraint hyperplane, $\Lambda(e^{j\omega})$. The second equality is (3.4)). Now, using (3.1), (3.5) and (3.6) we

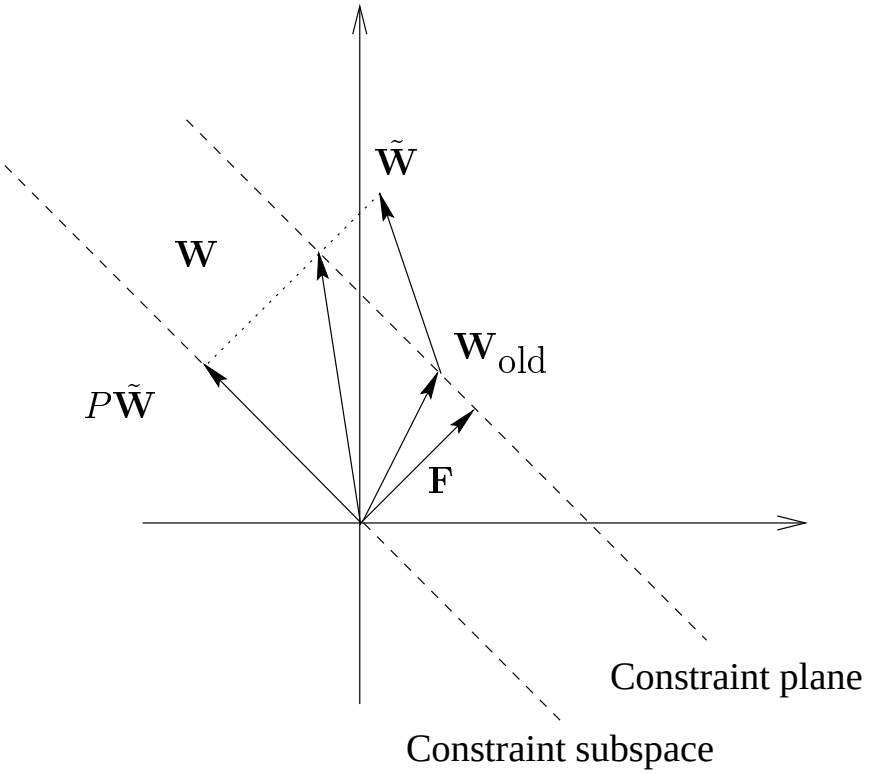

Figure 3.4: Frequency domain Frost algorithm: constrained LMS.

get

$$Y(t,e^{j\omega}) = Y_{\text{FBF}}(t,e^{j\omega}) - Y_{\text{NC}}(t,e^{j\omega}), \tag{3.9}$$

where

$$\begin{aligned} Y_{\text{FBF}}(t,e^{j\omega}) &= \mathbf{W}_0^{\dagger}(t,e^{j\omega})\mathbf{Z}(t,e^{j\omega}) \\ Y_{\text{NC}}(t,e^{j\omega}) &= \mathbf{G}^{\dagger}(t,e^{j\omega})\mathcal{H}^{\dagger}(e^{j\omega})\mathbf{Z}(t,e^{j\omega}). \end{aligned} \tag{3.10}$$

The output of the constrained beamformer is a difference of two terms, both operating on the input signal $\mathbf{Z}(t,e^{j\omega})$. The first term, $Y_{\text{FBF}}(t,e^{j\omega})$, utilizes only fixed components (which depend on the TFs), so it can be viewed as a fixed beamformer (FBF). We now examine the second term, $Y_{\text{NC}}(t,e^{j\omega})$. Note that

$$\begin{aligned} \mathbf{U}(t,e^{j\omega}) &= \mathcal{H}^{\dagger}(e^{j\omega})\mathbf{Z}(t,e^{j\omega}) \\ &= \mathcal{H}^{\dagger}(e^{j\omega})[\mathbf{A}(e^{j\omega})S(t,e^{j\omega}) + \mathbf{N}(t,e^{j\omega})] = \mathcal{H}^{\dagger}(e^{j\omega})\mathbf{N}(t,e^{j\omega}). \end{aligned} \tag{3.11}$$

The last transition is due to (3.7). $\mathbf{U}(t, e^{j\omega})$ are *reference noise* signals. Hence, the signal-dependent component of $Y_{\mathrm{NC}}(t, e^{j\omega})$ is completely eliminated (blocked) by $\mathcal{H}^{\dagger}(e^{j\omega})$, so that $Y_{\mathrm{NC}}(t, e^{j\omega})$ is a pure noise term. The noise term of $Y_{\mathrm{FBF}}(t, e^{j\omega})$ can be reduced by properly adjusting the filters $\mathbf{G}(t, e^{j\omega})$, using the minimum output power criterion. This adjustment problem is in fact the classical multi-channel noise cancellation problem. An adaptive LMS solution to the problem was proposed by Widrow [49].

The GSC solution is comprised of three components: a fixed beamformer (FBF), a blocking matrix (BM) that constructs the noise reference signals and a multi-channel noise canceller (NC). We now discuss each of these components in depth.

3.2.1 Fixed Beamformer (FBF)

By (3.10), (3.8) and (2.3) we have

$$Y_{\mathrm{FBF}}(t, e^{j\omega}) = \mathcal{F}^{*}(e^{j\omega}) S(t, e^{j\omega}) + \frac{\mathcal{F}^{*}(e^{j\omega})}{\|\mathbf{A}(e^{j\omega})\|^{2}} \mathbf{A}^{\dagger}(e^{j\omega}) \mathbf{N}(t, e^{j\omega}).$$

The first term on the right hand side is the signal term. The second is the noise term. Note that by setting $\mathcal{F}^{*}(e^{j\omega}) = e^{-j\omega\tau}$ (i.e., a delay), the signal component of $Y_{\mathrm{FBF}}(t, e^{j\omega})$ is an undistorted, delayed version of the desired signal.

Unfortunely, we usually do not have access to the actual TFs $(A_m(e^{j\omega})\,; m = 1, \ldots, M)$. Later we show how we can estimate the ratio of the TFs:

$$H_m(e^{j\omega}) = \frac{A_m(e^{j\omega})}{A_1(e^{j\omega})}\,; m = 1, \ldots, M.$$

Let

$$\mathbf{H}^{T}(e^{j\omega}) = \left[\begin{array}{cccc} 1 & \frac{A_2(e^{j\omega})}{A_1(e^{j\omega})} & \cdots & \frac{A_M(e^{j\omega})}{A_1(e^{j\omega})}. \end{array}\right] = \frac{\mathbf{A}^{T}(e^{j\omega})}{A_1(e^{j\omega})}$$

If in (3.8), the actual TFs are replaced by the TFs ratios, then

$$\mathbf{W}_0(t, e^{j\omega}) = \frac{\mathbf{H}(e^{j\omega})}{\|\mathbf{H}(e^{j\omega})\|^{2}} \mathcal{F}(e^{j\omega}). \tag{3.12}$$

By (3.10) and (2.3) we have

$$Y_{\mathrm{FBF}}(t, e^{j\omega}) = A_1(e^{j\omega}) \mathcal{F}^{*}(e^{j\omega}) S(t, e^{j\omega}) + \frac{\mathcal{F}^{*}(e^{j\omega})}{\|\mathbf{H}(e^{j\omega})\|^{2}} \mathbf{H}^{\dagger}(e^{j\omega}) \mathbf{N}(t, e^{j\omega}).$$

Thus, when $\mathbf{W}_0(t, e^{j\omega})$ is given by (3.12), the signal term of $Y_{\text{FBF}}(t, e^{j\omega})$ is the desired signal distorted only by the first TF, $A_1(e^{j\omega})$. Now suppose that

$$\mathbf{W}_0(t, e^{j\omega}) = \mathbf{H}(e^{j\omega})\mathcal{F}(e^{j\omega}). \tag{3.13}$$

In this case $\mathbf{W}_0(t, e^{j\omega})$ is comprised of the cascade of $\mathbf{H}(e^{j\omega})$, which is a filter matched to the ratio of the TFs, and $\mathcal{F}(e^{j\omega})$. The new $\mathbf{W}_0(t, e^{j\omega})$ can be derived from (3.12) under the assumption that $\|\mathbf{H}(e^{j\omega})\|^2$ is constant. In fact, Grenier *et al.* [41] argue that this assumption can be verified empirically. The FBF term of the output is now given by

$$Y_{\text{FBF}}(t, e^{j\omega}) = \frac{\|\mathbf{A}(e^{j\omega})\|^2}{A_1^*(e^{j\omega})}\mathcal{F}^*(e^{j\omega})S(t, e^{j\omega}) + \mathcal{F}^*(e^{j\omega})\mathbf{H}^\dagger(e^{j\omega})\mathbf{N}(t, e^{j\omega}). \tag{3.14}$$

The signal component of $Y_{\text{FBF}}(t, e^{j\omega})$ is now distorted. Hence, only a suboptimal solution is achieved. Note, however, that all the sensor outputs are added together coherently (as can be seen from the term $\|\mathbf{A}(e^{j\omega})\|^2$).

3.2.2 Blocking Matrix (BM)

Consider the following $M \times (M-1)$ matrix $\mathcal{H}(e^{j\omega})$,

$$\mathcal{H}(e^{j\omega}) = \begin{bmatrix} -\frac{A_2^*(e^{j\omega})}{A_1^*(e^{j\omega})} & -\frac{A_3^*(e^{j\omega})}{A_1^*(e^{j\omega})} & \dots & -\frac{A_M^*(e^{j\omega})}{A_1^*(e^{j\omega})} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \ddots & \\ & \dots & & \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{3.15}$$

It is easily verified that this matrix satisfies (3.7), and is hence a proper blocking matrix that may be used for generating the reference noise signals, $\mathbf{U}(t, e^{j\omega})$. By (3.11) we have,

$$U_m(e^{j\omega}) = Z_m(t, e^{j\omega}) - \frac{A_m(e^{j\omega})}{A_1(e^{j\omega})}Z_1(t, e^{j\omega})\ ; m = 2, \dots, M. \tag{3.16}$$

Thus, the knowledge of the TFs' ratios $H_m(e^{j\omega}) = \frac{A_m(e^{j\omega})}{A_1(e^{j\omega})}$ is sufficient to implement the sidelobe canceller.

3.2.3 Noise Canceller (NC)

By the GSC derivation we have constructed two signals. The first is $Y_{\rm FBF}(t,e^{j\omega})$, which contains both a desired signal term and a residual noise term. The second signal is $Y_{\rm NC}(t,e^{j\omega})$. $Y_{\rm NC}(t,e^{j\omega})$ consists of an adaptive set of filters $\mathbf{G}(t,e^{j\omega})$ which are applied to the noise only signals $\mathbf{U}(t,e^{j\omega})$.

Recall that our goal is to minimize the output power under a constraint on the response at the desired direction. By setting $\mathbf{W}_0(t,e^{j\omega})$ according to (3.8), the constraint is satisfied. Hence, minimization of the output power is achieved by adjusting the filters $\mathbf{G}(t,e^{j\omega})$. This is an unconstrained minimization, exactly as in Widrow's classical problem [49]. We can implement it by using the multi-channel Wiener filter. Recall (3.9), our goal is to set $\mathbf{G}(t,e^{j\omega})$ so as to minimize

$$\mathrm{E}\left\{||Y_{\rm FBF}(t,e^{j\omega})-\mathbf{G}^{\dagger}(t,e^{j\omega})\mathbf{U}(t,e^{j\omega})||^2\right\}.$$

Let

$$\begin{aligned} S_{\mathbf{U}Y}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})Y^{*}_{\rm FBF}(t,e^{j\omega})\} \\ S_{\mathbf{UU}}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})\mathbf{U}^{\dagger}(t,e^{j\omega})\}. \end{aligned}$$

Then the multi-channel Wiener filter is given by [50], [22]

$$\mathbf{G}(t,e^{j\omega})=S^{-1}_{\mathbf{UU}}(t,e^{j\omega})S_{\mathbf{U}Y}(t,e^{j\omega}). \tag{3.17}$$

In order to be able to track changes, we process the signals by segments. The following frequency domain LMS algorithm is used. Let the residual signal be

$$Y(t,e^{j\omega})=Y_{\rm FBF}(t,e^{j\omega})-\mathbf{G}^{\dagger}(t,e^{j\omega})\mathbf{U}(t,e^{j\omega}).$$

Note that the residual signal is also the output of the enhancement algorithm. By the orthogonality principle, the error is orthogonal to the measurements. Thus,

$$E\{\mathbf{U}(t,e^{j\omega})Y^{*}(t,e^{j\omega})\}=0. \tag{3.18}$$

Following the standard Widrow procedure, the solution is

$$\mathbf{G}(t+1,e^{j\omega})=\mathbf{G}(t,e^{j\omega})+\mu\mathbf{U}(t,e^{j\omega})Y^{*}(t,e^{j\omega}).$$

Usually, a more stable solution will be the normalized LMS (NLMS) procedure, in which each frequency will be normalized separately, yielding:

$$G_m(t+1,e^{j\omega}) = G_m(t,e^{j\omega}) + \mu\frac{U_m(t,e^{j\omega})Y^*(t,e^{j\omega})}{P_{\text{est}}(t,e^{j\omega})}, \quad m=2,\ldots,M$$

where,

$$P_{\text{est}}(t,e^{j\omega}) = \lambda P_{\text{est}}(t-1,e^{j\omega}) + (1-\lambda)\sum_m |Z_m(t,e^{j\omega})|^2 \tag{3.19}$$

λ is a forgetting factor (typically $0.8<\lambda<1$). Another possibility is to calculate P_{est} using the power of the noise reference signals. However, in that case an energy detector is required, so that $\mathbf{G}(t,e^{j\omega})$ is updated only when there is no active signal. If on the other hand, we calculate $P_{\text{est}}(t,e^{j\omega})$ using the input sensor signals, as indicated in (3.19), then an energy detector may be avoided. This is due to the fact that the adaptation term becomes relatively small during periods of active input signal.

A common procedure in echo cancellation systems is to constrain the adapted filters to an FIR structure (e.g. [51]). Assume a noncasual FIR model both for the TFs' ratios, $\mathbf{h}_m$, and for the noise cancelling filters, $\mathbf{g}_m$, $m=1,\ldots,M$:

$$\begin{aligned}\mathbf{h}_m^T &= [h_m(-q_L),\cdots,h_m(q_R)]\\ \mathbf{g}_m^T &= [g_m(-K_L),\cdots,g_m(K_R)]\end{aligned} \tag{3.20}$$

(note that both $\mathbf{h}_m$ and $\mathbf{g}_m$ are functions of time; however, for notational simplicity we omit this dependence). Note that the TFs might have zeros outside the unit circle. Thus to ensure stability of the TFs ratios we do not impose them to be causal.

In order to fulfill the FIR structure constraint (3.20), the filters update is now given by

$$\begin{aligned}\tilde{G}_m(t+1,e^{j\omega}) &= G_m(t,e^{j\omega}) + \mu\frac{U_m(t,e^{j\omega})Y^*(t,e^{j\omega})}{P_{\text{est}}(t,e^{j\omega})}\\ G_m(t+1,e^{j\omega}) &\xleftarrow{\text{FIR}} \tilde{G}_m(t+1,e^{j\omega})\end{aligned} \tag{3.21}$$

for $m=2,\ldots,M$. The operator $\xleftarrow{\text{FIR}}$ includes the following three stages. First, we transform $\tilde{G}_m(t+1,e^{j\omega})$ to the time domain. Second, we truncate the resulting

impulse response to the interval $[-K_L, K_R]$ (i.e. we impose an FIR constraint). Third, we transform back to the frequency domain.

Note that the various filtering operations (multiplications in the transform domain) are realized using the *overlap & save* method [52]. Thus, aliasing effect due to cyclic convolution is eliminated.

3.2.4 Time Domain Implementation

We derived a minimum variance beamformer in the frequency domain. The GSC structure can also be implemented in the time domain. For that purpose we assume again the noncasual FIR model both for the TFs' ratios, $\mathbf{h}_m$, and for the noise cancelling filters, $\mathbf{g}_m$.

Fixed Beamformer

Using Eq. (3.13) and assuming delay only constraint, we have

$$y_{\mathrm{FBF}}(t) = \sum_{m=1}^{M} \sum_{k=-q_L}^{q_R} h_m(k) z_m(t+k).$$

Reference Noise Signals

The desired signal Using Eq. (3.16) we have

$$u_m(t) = z_m(t) - \sum_{k=-q_L}^{q_R} h_m(k) z_1(t-k).$$

Noise Cancellation

The filters $\mathbf{g}_m$ are updated in the time domain, using the multi-channel Wiener-Hopf equations, on a sample by sample basis. The error signal (which is also the output of the enhancement algorithm) is given by,

$$y(t) = y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \sum_{k=-K_L}^{K_R} g_m(k) u_m(t-k). \tag{3.22}$$

By the orthogonality principle,

$$E\{y(t)\mathbf{u}_m(t)\} = 0 \; ; \; m = 2, 3, \ldots, M \tag{3.23}$$

where,

$$\mathbf{u}_m^T(t) = \left[\begin{array}{ccccccc} u_m(t+K_L) & \cdots & u_m(t) & \cdots & u_m(t-K_R) \end{array} \right].$$

The adaptive LMS solution is given by

$$\mathbf{g}_m(t) = \mathbf{g}_m(t-1) + \mu \mathbf{u}_m(t) y(t) \; ; m = 2, \ldots, M.$$

Thus, $M-1$ Widrow-LMS algorithms are activated simultaneously. Again, the NLMS version is more stable:

$$\mathbf{g}_m(t) = \mathbf{g}_m(t-1) + \frac{\mu}{p_{\text{est}}(t)} \mathbf{u}_m(t) y(t) \; ; m = 2, \ldots, M.$$

where,

$$p_{\text{est}}(t) = \sum_{m=1}^{M} \|\mathbf{z}_m(t)\|^2 \tag{3.24}$$

and,

$$\mathbf{z}_m^T(t) = \left[\begin{array}{ccccccc} z_m(t+K_L) & \cdots & z_m(t) & \cdots & z_m(t-K_R) \end{array} \right].$$

As in the frequency domain, another possibility for the normalizing factor is to calculate $p_{\text{est}}(t)$ using the power of the noise reference signals. However, in that case an energy detector is required, so that $\mathbf{g}_m$ are updated only when there is no active signal. If on the other hand, we calculate $p_{\text{est}}(t)$ using the input sensor signals, as indicated in (3.24), then an energy detector may be avoided.

3.3 Algorithm Summary

The algorithm is constructed from three main blocks. The first is the fixed beamformer implemented as a matched filter (alignment block). The second is the blocking matrix which blocks the desired signal. The third is the unconstrained noise canceller which cancels the signal components that have no correlation with the desired signal, using the reference signals produced by the blocking matrix. The new algorithm can be regarded as an extension of the Griffiths and Jim algorithm for the general TF case. The algorithm is summarized in Figure 3.5 (frequency domain) and in Fig 3.6 (time domain).

1) TFs ratios: $\mathbf{H}(e^{j\omega}) = \frac{\mathbf{A}(e^{j\omega})}{A_1(e^{j\omega})}$
2) Fixed beamformer:
$$Y_{\text{FBF}}(t, e^{j\omega}) = \mathbf{F}^{\dagger}(e^{j\omega})\mathbf{Z}(t, e^{j\omega})$$
3) Noise reference signals:
$$\mathbf{U}(t, e^{j\omega}) = \mathcal{H}^{\dagger}(e^{j\omega})\mathbf{Z}(t, e^{j\omega})$$
4) Output signal:
$$Y(t, e^{j\omega}) = Y_{\text{FBF}}(t, e^{j\omega}) - \mathbf{G}^{\dagger}(t, e^{j\omega})\mathbf{U}(t, e^{j\omega})$$
5) Filters update, for $m = 2, \ldots, M$:
$$\tilde{G}_m(t+1, e^{j\omega}) = G_m(t, e^{j\omega}) + \mu \frac{U_m(t, e^{j\omega}) Y^*(t, e^{j\omega})}{P_{\text{est}}(t, e^{j\omega})}$$
$$G_m(t+1, e^{j\omega}) \xleftarrow{\text{FIR}} \tilde{G}_m(t+1, e^{j\omega})$$
where, $P_{\text{est}}(t, e^{j\omega}) = \lambda P_{\text{est}}(t-1, e^{j\omega}) + (1-\lambda) \sum_m |Z_m(t, e^{j\omega})|^2$
6) keep only non-aliased samples.
(note: $\mathbf{F}(e^{j\omega})$ is defined in (3.4). $\mathcal{H}(e^{j\omega})$ is defined in (3.15)).

Figure 3.5: Suggested Algorithm (frequency domain)

Figure 3.7 depicts our suggested solution.

The ratios of the TFs are assumed to be known at this stage.

1) TFs' ratios, for $m = 2, \ldots, M$:
$h_m(e^{j\omega}) = \mathrm{IDTFT}\{\mathbf{H}_m(e^{j\omega})\}$

2) Fixed beamformer
$y_{\mathrm{FBF}}(t) = \sum_{m=1}^{M} \sum_{k=-q_L}^{q_R} h_m(k) z_m(t+k)$

3) Noise reference signals, for $m = 2, \ldots, M$:
$u_m(t) = z_m(t) - \sum_{k=-q_L}^{q_R} h_m(k) z_1(t-k)$

4) Output signal
$y(t) = y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \sum_{k=-K_L}^{K_R} g_m(k) u_m(t-k)$

5) Filters' update, for $m = 2, \ldots, M$:
$\mathbf{g}_m(t) = \mathbf{g}_m(t-1) + \frac{\mu}{p_{\mathrm{est}}(t)} \mathbf{u}_m(t) y(t)$

where, $p_{\mathrm{est}}(t) = \sum_{m=1}^{M} \|\mathbf{z}_m(t)\|^2$

Figure 3.6: Suggested algorithm (time domain).

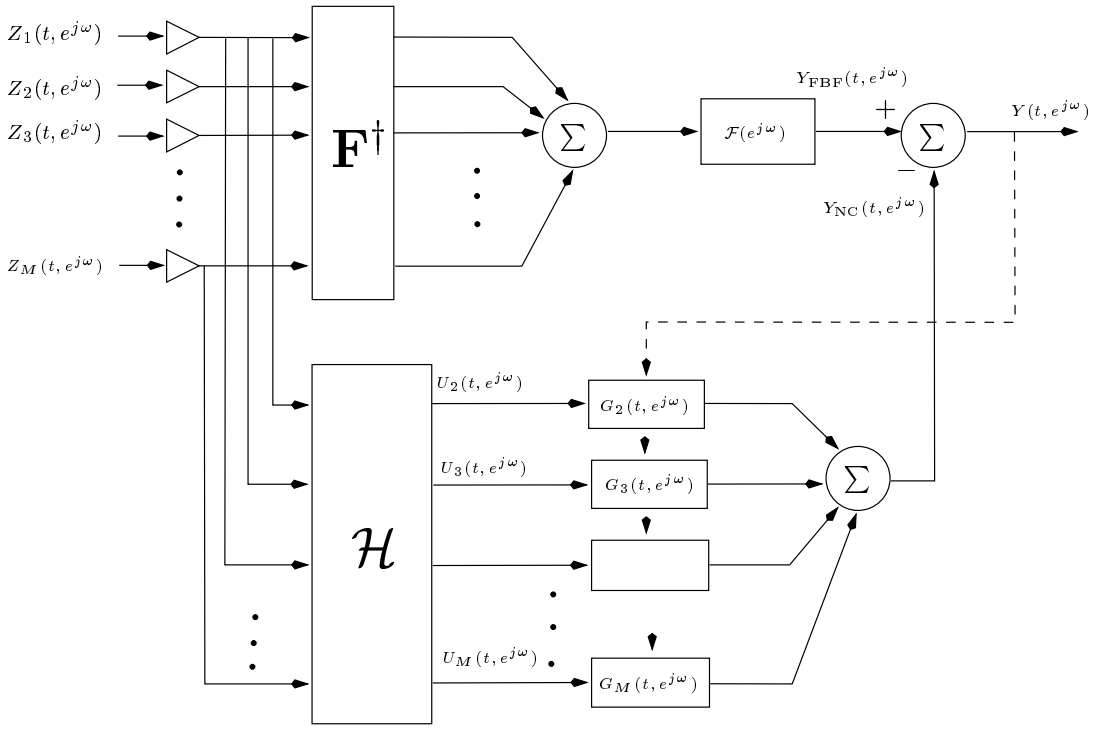

Figure 3.7: Linearly constrained adaptive beamformer.

Chapter 4

Identification Using Nonstationarity

Thus far we have assumed that the vector of TFs ratio, $\mathbf{H}(e^{j\omega})$, is known. In practice, however, $\mathbf{H}(e^{j\omega})$ is not known and should be estimated. Reformulating Eq. 3.16 in the frequency domain

$$Z_m(t, e^{j\omega}) = H_m(e^{j\omega}) Z_1(t, e^{j\omega}) + U_m(t, e^{j\omega}) \;\; ; m = 2, \ldots, M$$

and in the time domain,

$$z_m(t) = h_m(t) * z_1(t) + u_m(t) \;\; ; m = 2, \ldots, M.$$

We note that the TFs' ratios relate the first sensor signal, $z_1(t)$, and each of the other sensors, $z_m(t),\ m = 2, \ldots, M$. Since the noise terms $u_m(t),\ m = 2, \ldots, M$ are correlated with the input signal $z_1(t)$, a standard least squares system identification procedure would result in a biased estimate. Instead, we propose using the system identification criterion suggested in [44], which is based on signal nonstationarity.

The estimation can be conducted either in the frequency domain (Section 4.1) or in the time domain (Section 4.2).

4.1 System Identification in the Frequency Domain

We assume that the TFs' ratios change slowly over time compared to the time variations of the desired signal, and that the noise components are more stationary than the desired signal components.

Consider some time period T during which the noise signal is assumed to be stationary and the TFs fixed. We divide that period into frames of length T_k, $T = \sum_{k=1}^{K} T_k$. During each frame the input signal correlation (or spectrum) and the input-output cross-correlation (or cross-spectrum) are estimated, yielding a set of several equations: a separate set of equations is used for each channel. For each frame we obtain,

$$\begin{aligned} \hat{S}^{(k)}_{z_m z_1}(e^{j\omega}) &= H_m(e^{j\omega})\hat{S}^{(k)}_{z_1 z_1}(e^{j\omega}) + \hat{S}_{u_m z_1}(e^{j\omega}) \\ &= H_m(e^{j\omega})\hat{S}^{(k)}_{z_1 z_1}(e^{j\omega}) + S_{u_m z_1}(e^{j\omega}) + \varepsilon^{(k)}_m(e^{j\omega});\ k = 1, \ldots, K \end{aligned}$$

where, $\hat{S}^{(k)}_{z_1 z_1}(e^{j\omega})$ is the input signal auto-spectrum in the $k-th$ frame, $\hat{S}^{(k)}_{z_m z_1}(e^{j\omega})$ is the input-output cross-spectrum in the $k-th$ frame and $\varepsilon^{(k)}_m(e^{j\omega}) = \hat{S}^{(k)}_{u_m z_1}(e^{j\omega}) - S_{u_m z_1}(e^{j\omega})$ is the estimation error of the cross-spectrum function between $z_1(t)$ and $u_m(t)$ in the $k-th$ frame. Note, that $S_{u_m z_1}(e^{j\omega})$ is not dependent on the frame index, since it depends only on the noise statistics (desired signal and noise signal are assumed to be uncorrelated). K is the number of frames used to exploit the nonstationarity characteristics of the signal. Note, that using only one frame, as in the conventional system identification methods, one obtains a biased estimate of the desired filter. By using K separate segments for each channel, $m = 2, \ldots, M$, we obtain K equations in the two unknowns, $H_m(t, e^{j\omega})$ and $S_{u_m z_1}(e^{j\omega})$. The LS technique can be used to obtain an unbiased TFs' ratio estimate. In matrix form:

$$\begin{bmatrix} \hat{S}^{(1)}_{z_m z_1}(e^{j\omega}) \\ \hat{S}^{(2)}_{z_m z_1}(e^{j\omega}) \\ \vdots \\ \hat{S}^{(K)}_{z_m z_1}(e^{j\omega}) \end{bmatrix} = \begin{bmatrix} \hat{S}^{(1)}_{z_1 z_1}(e^{j\omega}) & 1 \\ \hat{S}^{(2)}_{z_1 z_1}(e^{j\omega}) & 1 \\ \vdots & \\ \hat{S}^{(K)}_{z_1 z_1}(e^{j\omega}) & 1 \end{bmatrix} \begin{bmatrix} H_m(e^{j\omega}) \\ S_{u_m z_1}(e^{j\omega}) \end{bmatrix} + \begin{bmatrix} \varepsilon^{(1)}_m(e^{j\omega}) \\ \varepsilon^{(2)}_m(e^{j\omega}) \\ \vdots \\ \varepsilon^{(K)}_m(e^{j\omega}) \end{bmatrix}. \qquad (4.1)$$

Analysis of the suggested method is given in [44]. The Blackman-Tukey approach for spectrum estimation is used. Spectrum estimation is obtained by multiplying the empirical correlation function (in each subframe) by a window function. Subframe errors are assumed to be mutually uncorrelated. If far segments are used, this assumption can be verified. The resulting TFs' ratio estimates, using the nonstationarity equations, are bias free, and the variance of the estimation is given by,

$$\begin{aligned} var\{E^m(e^{j\omega})\} &= \frac{S_{u_m u_m}(t,e^{j\omega})}{BT}\frac{< 1/S_{z_1 z_1}(t,e^{j\omega}) >}{< S_{z_1 z_1}(t,e^{j\omega}) >< 1/S_{z_1 z_1}(t,e^{j\omega}) > -1} \\ & \frac{1}{BT}\frac{S_{u_m u_m}(t,e^{j\omega})}{< S_{z_1 z_1}(t,e^{j\omega}) >}\frac{< S_{z_1 z_1}(t,e^{j\omega}) >< 1/S_{z_1 z_1}(t,e^{j\omega}) >}{< S_{z_1 z_1}(t,e^{j\omega}) >< 1/S_{z_1 z_1}(t,e^{j\omega}) > -1} \end{aligned}$$

where, the $< >$ operation denotes frame averaging, T is the total time used for estimation, $B = \frac{1}{\sum_\tau w^2(\tau)}$ is related to the window bandwidth. The conventional estimation procedure is biased and its variance is given by the well-known expression (see, for instance, [44]),

$$var\{E^m(e^{j\omega})\} = \frac{1}{BT}\frac{S_{u_m u_m}}{< S_{z_1 z_1}(t,e^{j\omega}) >}.$$

Define *relative efficiency* (the ratio between nonstationarity method and conventional method variances),

$$\Xi(e^{j\omega}) = \frac{< S_{z_1 z_1}(t,e^{j\omega}) >< 1/S_{z_1 z_1}(t,e^{j\omega}) >}{< S_{z_1 z_1}(t,e^{j\omega}) >< 1/S_{z_1 z_1}(t,e^{j\omega}) > -1}.$$

Note, that as $S_{z_1 z_1}(t,e^{j\omega}) \geq 0$, the relative efficiency always fulfills $\Xi(e^{j\omega}) > 1$. For a stationary signal $\Xi(e^{j\omega}) \to \infty$. Thus, $1 < \Xi(e^{j\omega}) < \infty$. The more the signal is nonstationary $\Xi(e^{j\omega})$ tends to 1, and the more it is stationary $\Xi(e^{j\omega})$ tends to infinity. Define, averaged signal to noise ratio,

$$SNR^m_{ave}(e^{j\omega}) = \frac{< S_{z_1 z_1}(t,e^{j\omega}) >}{S_{u_m u_m}(t,e^{j\omega})}$$

As,

$$Z_m(t,e^{j\omega}) = H_m(e^{j\omega})Z_1(t,e^{j\omega}) + U_m(t,e^{j\omega})$$

the average SNR depends on the power ratio between the input signal and the related disturbance. Collecting all definitions we obtain,

$$var\{E^m(e^{j\omega})\} = \frac{1}{BT}\frac{1}{SNR^m_{ave}(e^{j\omega})}\Xi(e^{j\omega}). \tag{4.2}$$

The estimation error variance is increasing as the input signal ($z_1(t)$) becomes more stationary, and as the averaged SNR decreases. Note, that $z_1(t)$ and $u_m(t)$ have related terms. If the first sensor noise signal, $n_1(t)$ is very strong, both $z_1(t)$ and $u_m(t)$ become large and the averaged SNR tends to 1. The degradation in the TFs' estimate is then due to the term $\Xi(e^{j\omega})$. When the (presumably stationary) noise signal becomes dominant, the signal $z_1(t)$ becomes more stationary, causing $\Xi(e^{j\omega}) \to \infty$. This phenomenon may be interpreted in the following manner. As the signals are stationary, all the equations in the over-determined set given in Eq. 4.1 become similar, yielding an ill-conditioned problem.

Special attention should be paid with regard to the frame length. On one hand, it should be longer than the correlation length of $z_m(t)$, which itself must be longer than the length of the filter $a_m(t)$, which might very high. On the other hand, it should be short enough for the quasi-stationarity assumption to hold.

4.2 System Identification in the Time Domain

Under the FIR model for the TFs' ratio, Eq. (3.16) can be rewritten in the time domain:

$$z_m(t) = \mathbf{z}_1^T(t)\mathbf{h}_m(t) + u_m(t) \quad m = 2, \ldots, M$$

where,

$$\mathbf{z}_1^T(t) = [z_1(t + q_L), \ldots, z_1(t - q_R)].$$

The signals' non-stationarity can be exploited in the time domain as follows:

$$\begin{aligned} \hat{r}^{(k)}_{\mathbf{z}_1 z_m} &\triangleq \frac{1}{T_k}\sum_{t=1}^{T_k}\mathbf{z}_1(t)z_m(t) \\ &= \frac{1}{T_k}\sum_{t=1}^{T_k}\mathbf{z}_1(t)\left[\mathbf{z}_1^T(t)\mathbf{h}_m + u_m(t)\right] \end{aligned} \tag{4.3}$$

$$
\begin{aligned}
&= \left(\frac{1}{T_k}\sum_{t=1}^{T_k}\mathbf{z}_1(t)\mathbf{z}_1^T(t)\right)\mathbf{h}_m + \frac{1}{T_k}\sum_{t=1}^{T_k}\mathbf{z}_1(t)u_m(t) \\
&\triangleq \hat{R}^{(k)}_{\mathbf{z}_1\mathbf{z}_1}\mathbf{h}_m + \hat{r}^{(k)}_{\mathbf{z}_1 u_m} \qquad (4.4)\\
&= \hat{R}^{(k)}_{\mathbf{z}_1\mathbf{z}_1}\mathbf{h}_m + r_{\mathbf{z}_1 u_m} + \varepsilon_m^{(k)},\ k = 1,\ldots,K
\end{aligned}
$$

where, $\hat{r}^{(k)}_{\mathbf{z}_1 z_m}$,$\hat{r}^{(k)}_{\mathbf{z}_1 u_m}$ are cross-correlation vector estimates, and $\hat{R}^{(k)}_{\mathbf{z}_1\mathbf{z}_1}$ is auto-correlation matrix estimate, and $\varepsilon_m^{(k)} = \hat{r}^{(k)}_{\mathcal{Z}_1 u_m} - r_{\mathbf{z}_1 u_m}$ is the estimation error of the cross-correlation vector. All estimates are in the $k-th$ frame. T_k is the frame length. In matrix form:

$$
\begin{pmatrix} \hat{r}^{(1)}_{\mathbf{z}_1 z_m} \\ \hat{r}^{(2)}_{\mathbf{z}_1 z_m} \\ \vdots \\ \hat{r}^{(K)}_{\mathbf{z}_1 z_m} \end{pmatrix} = \begin{bmatrix} \hat{R}^{(1)}_{\mathbf{z}_1\mathbf{z}_1} & I \\ \hat{R}^{(2)}_{\mathbf{z}_1\mathbf{z}_1} & I \\ \vdots & \\ \hat{R}^{(K)}_{\mathbf{z}_1\mathbf{z}_1} & I \end{bmatrix} \begin{pmatrix} \mathbf{h}_m \\ r_{\mathbf{z}_1 u_m} \end{pmatrix} + \begin{pmatrix} \varepsilon_m^{(1)} \\ \varepsilon_m^{(2)} \\ \vdots \\ \varepsilon_m^{(K)} \end{pmatrix}. \qquad (4.5)
$$

The number of correlation lags used by this time domain version is the length of the TFs' ratios $\mathbf{h}_m(t)$ estimated. Experimently, we verified that only a moderate number of coefficients are required in speech enhancement application in a large room.

4.3 Chapter Summary

We have derived both frequency and time domain expressions for estimating the TFs' ratio, $\mathbf{h}_m(t)$ or $H_m(t, e^{j\omega})$. The estimation procedure exploits the desired signal nonstationarity to construct an over-determined set of equations, which yield a bias-free estimation of the TFs' ratio.

On one hand, the time domain equations have an advantage over the frequency domain equations, since fewer number of correlations lags should be estimated. On the other hand, in the frequency domain formulation each frequency band can be processed separately, yielding a more efficient matrix inversion (only 2×2), and enabling discrimination of unimportant frequency bands.

Both time domain and frequency domain estimates might be used in the suggested algorithms summarized in Figures 3.5, 3.6.

Chapter 5

Identification Using Decorrelation

In Chapter 4 we introduced a system identification method exploiting the non-stationarity of the signals involved. This method is bias-free, but has a slight degradation in estimation variance, in comparison to standard LS estimation methods. Performance analysis can be found in Chapter 4 and in [44]. Estimation error in the TFs $\mathbf{h}_m(t)$ (or, in frequency domain, $\mathbf{H}(e^{j\omega})$) would manifest itself as a leakage of the desired signal into the the reference noise signal, $u_m(t)$, due to non-perfect blocking of the matrix $\mathcal{H}(e^{j\omega})$. This leakage would cause self-cancellation of the desired signal, i.e., would impose a distortion on the output signal. This phenomena was noticed by Widrow in his early works on noise cancellation [49]. Weinstein *et al.* [45] suggested coping with the leakage problem of Widrow's noise cancelling algorithm by imposing the desired signal and the noise signal to be mutually uncorrelated. Bar-Ness *et al.* [53] suggest the use of multiuser decorrelating detector for asynchronous CDMA.

In this chapter we suggest an extension for the *decorrelation criterion* to deal with the multi-channel case, and incorporate it into the suggested GSC structure. The suggested method might improve the performance of the noise canceller branch of our sensor array. As a byproduct, we also obtain estimation algorithm for tracking changes in the TFs' ratio, $\mathbf{h}_m(t)$, together with pre-derived updating procedure for the noise cancelling filters, $\mathbf{g}_m(t)$.

The decorrelation criterion is introduced in Section 5.1. We derive an iterative solution for the filters $\mathbf{h}_m(t)$, $\mathbf{g}_m(t)$ in Section 5.2. We suggest a sequential LMS-like algorithm for the filters estimation in Section 5.3.

5.1 The General Decorrelation Problem

Recall, the (time domain) definition of the signals $y(t)$ and $u_m(t)$:

$$\begin{aligned} u_m(t) &= z_m(t) - \sum_{k=-q_L}^{q_R} h_m(k) z_1(t-k) \\ y(t) &= y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \sum_{k=-K_L}^{K_R} g_m(k) u_m(t-k). \end{aligned}$$

We want to adjust the filters $\mathbf{g}_m(t)$ to achieve elimination of the noise component in the output signal $y(t)$, and to adjust the filters $\mathbf{h}_m(t)$ to achieve elimination of the desired signal component (caused by leakage) in the reference signals $u_m(t)$.

Using the previously defined FIR constraints on the filters,

$$\begin{aligned} \mathbf{h}_m^T &= [h_m(-q_L), \cdots, h_m(q_R)] \\ \mathbf{g}_m^T &= [g_m(-K_L), \cdots, g_m(K_R)] \end{aligned}$$

and the state-space presentation of the signals,

$$\begin{aligned} \mathbf{u}_m^T(t) &= \left[\begin{array}{ccccc} u_m(t+K_L) & \cdots & u_m(t) & \cdots & u_m(t-K_R) \end{array} \right] \\ \mathbf{y}^T(t) &= \left[\begin{array}{ccccc} y(t+q_L) & \cdots & y(t) & \cdots & y(t-q_R) \end{array} \right] \\ \mathbf{z}_1^T(t) &= \left[\begin{array}{ccccc} z_1(t+q_L) & \cdots & z_1(t) & \cdots & z_(t-q_R) \end{array} \right] \end{aligned}$$

we can reformulate the signals' definitions:

$$\begin{aligned} u_m(t) &= z_m(t) - \mathbf{z}_1^T(t)\mathbf{h}_m(t) \\ y(t) &= y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \mathbf{g}_m^T(t)\mathbf{u}_m(t). \end{aligned}$$

Observing that the signals $y(t)$ and $u_m(t)$ should be uncorrelated (as the desired signal and noise signals are uncorrelated), we have the following equations:

$$E\{y(t)u_m(t+\tau)\} = 0, \quad m = 2, \ldots, M, \ \forall \tau.$$

Since the filters are constrained to be FIR structured, we have enough equations to solve the problem. In order for the decorrelation criterion to be sufficient to separate the signal and the noise signals from their mixed observations, we need to have only pair of source signals in each equation. For the case of a point source noise signal, this condition is fulfilled.

5.2 Iterative Solution

We can state the *decorrelation criterion* [45], and find an iterative solution for the filters, by using the following dual set of equations,

$$E\{u_m(t)\mathbf{y}(t)\} = 0 \quad \Rightarrow \quad \mathbf{h}_m(t) \; ; \; m = 2, \ldots, M \tag{5.1}$$

$$E\{y(t)\mathbf{u}_m(t)\} = 0 \quad \Rightarrow \quad \mathbf{g}_m(t) \; ; \; m = 2, \ldots, M. \tag{5.2}$$

Note, that Eq. 5.1 is exactly the same as the orthogonality principle applied in forming Eq. 3.23.

Starting from Eq. 5.2

$$E\left\{\mathbf{u}_p(t)\left[y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \mathbf{u}_m^T(t)\mathbf{g}_m(t)\right]\right\} = 0 \; ; p = 2, \ldots, M.$$

Rearranging terms,

$$E\left\{\mathbf{u}_p(t)y_{\mathrm{FBF}}(t)\right\} = \sum_{m=2}^{M} E\left\{\mathbf{u}_p(t)\mathbf{u}_m^T(t)\right\}\mathbf{g}_m(t); \; p = 2, \ldots, M. \tag{5.3}$$

Define, assuming that the signals involved are stationary,

$$\begin{aligned} R_{\mathbf{u}_p\mathbf{u}_m} &= E\left\{\mathbf{u}_p(t)\mathbf{u}_m^T(t)\right\} \\ \mathbf{r}_{\mathbf{u}_p y_{\mathrm{FBF}}} &= E\left\{y_{\mathrm{FBF}}(t)\mathbf{u}_p(t)\right\}. \end{aligned}$$

Using Eq. 5.3 and concatenating the filters $\mathbf{g}_m(t)$ yields,

$$\begin{bmatrix} R_{\mathbf{u}_2\mathbf{u}_2} & R_{\mathbf{u}_2\mathbf{u}_3} & \ldots & R_{\mathbf{u}_2\mathbf{u}_M} \\ R_{\mathbf{u}_3\mathbf{u}_2} & R_{\mathbf{u}_3\mathbf{u}_3} & \ldots & R_{\mathbf{u}_3\mathbf{u}_M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{\mathbf{u}_M\mathbf{u}_2} & R_{\mathbf{u}_M\mathbf{u}_3} & \ldots & R_{\mathbf{u}_M\mathbf{u}_M} \end{bmatrix} \begin{bmatrix} \mathbf{g}_2(t) \\ \mathbf{g}_3(t) \\ \vdots \\ \mathbf{g}_M(t) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\mathbf{u}_2 y_{\mathrm{FBF}}} \\ \mathbf{r}_{\mathbf{u}_3 y_{\mathrm{FBF}}} \\ \vdots \\ \mathbf{r}_{\mathbf{u}_M y_{\mathrm{FBF}}} \end{bmatrix}. \tag{5.4}$$

This matrix equation is the multi-channel Wiener-Hopf set of equation. Note, that existence of correlation between the measurements, as observed in the case of one point source noise signal, would cause an ill-conditioned set of equations. This problem can be overcomed by introducing some small stabilizing error signal to each measurement.

Now, starting from Eq. 5.1 we obtain:

$$\begin{aligned} 0 &= E\{\mathbf{y}(t)u_m(t)\} \\ &= E\left\{\mathbf{y}(t)\left[z_m(t) - \mathbf{z}_1^T(t)\mathbf{h}_m(t)\right]\right\} \;; m = 2, \ldots, M. \end{aligned} \tag{5.5}$$

Rearranging terms,

$$E\left\{\mathbf{y}(t)z_m(t)\right\} = E\left\{\mathbf{y}(t)\mathbf{z}_1^T(t)\right\}\mathbf{h}_m(t);\; m = 2, \ldots, M. \tag{5.6}$$

Define the correlation matrices:

$$\begin{aligned} \mathbf{r}_{\mathbf{y}z_m} &= E\left\{\mathbf{y}(t)z_m(t)\right\} \\ R_{\mathbf{y}\mathbf{z}_1} &= \left\{\mathbf{y}(t)\mathbf{z}_1^T(t)\right\} \end{aligned}$$

yielding,

$$R_{\mathbf{y}\mathbf{z}_1}\mathbf{h}_m(t) = \mathbf{r}_{\mathbf{y}z_m};\; m = 2, \ldots, M. \tag{5.7}$$

Note, that in order to solve Eq. 5.4 for the filter $\mathbf{g}_m(t)$, we need the FBF signal $y_{\mathrm{FBF}}(t)$, which depends on the filters $\mathbf{h}_m(t)$, and to solve Eq. 5.7 we need the signal $y(t)$, which depends on the noise cancelling filters $\mathbf{g}_m(t)$. Thus, both sets of equations must be solved simultaneously. By alternating between Eqs. 5.4,5.7, we obtain an iterative algorithm for adjusting both filter coefficients.

5.3 Sequential LMS-like Solution

We suggest a sequential solution for the decorrelation criterion. Using again Eqs. 5.1,5.2 and applying the Robbins-Monro first order approximation method [54] give rise to the following sequential algorithm,

$$\mathbf{h}_m(t) = \mathbf{h}_m(t-1) + \eta\mathbf{y}(t)u_m(t)\;; m = 2, \ldots, M \tag{5.8}$$

$$\mathbf{g}_m(t) = \mathbf{g}_m(t-1) + \mu\mathbf{u}_m(t)y(t)\;; m = 2, \ldots, M. \tag{5.9}$$

Thus, dual $M-1$ Widrow-LMS algorithms are activated simultaneously. Again, the NLMS version is more stable:

$$\begin{aligned}\mathbf{h}_m(t) &= \mathbf{h}_m(t-1) + \frac{\eta}{\|\mathbf{y}(t)\|^2}\mathbf{y}(t)u_m(t)\ ; m = 2, \ldots, M \\ \mathbf{g}_m(t) &= \mathbf{g}_m(t-1) + \frac{\mu}{\|\mathbf{u}_m(t)\|^2}\mathbf{u}_m(t)y(t)\ ; m = 2, \ldots, M.\end{aligned}$$

Note, the the updating equations for the cancelling filters $\mathbf{g}_m(t)$ are left unchanged as there was no leakage of desired signal. A good practice might be to stop adaptation of $\mathbf{h}_m(t)$ during non-signal periods, and to stop adaptation of $\mathbf{g}_m(t)$ during signal existence periods. Initialization difficulties of the suggested method might be overcome by using the filters' estimates, achieved by exploiting the nonstationarity equations, as derived in Chapter 4.

5.4 Chapter Summary

Observing that the desired signal and the reference noise signals are uncorrelated, a new criterion can be used. This property can be exploited to give rise to a twofold advantage. The leakage problem of the desired signal into the noise reference signals might be eliminated. And, as a byproduct, we gain a tracking procedure for the filters $\mathbf{h}_m(t)$.

Chapter 6

Performance Analysis

In this chapter we analyze the expected performance of the suggested algorithm.

In [23],[22][24] (for the two channel case), [15],[25] and [26], the limits of the GSC structure, when closed form Wiener filters replaces the adaptive solutions, are analyzed. The performance penalty for constraining the Wiener filters to an FIR structure is demonstrated by Nordholm *et al.*in [50]. The resulting performance limits depend on the cross-correlation between the sensors' signals induced by the noise field, as shown in the above references and by Cox [55].

In this chapter, we also use the closed form frequency domain solution (instead of the practical adaptive solution) for analyzing the suggested algorithm. This solution bounds the expected performance of the adaptive solution.

In Section 6.1 we formulate a general expression for the algorithm's output power. The desired signal *distortion* is addressed in Section 6.2, and the achievable *Noise reduction* and its dependency on the noise field is addressed in Section 6.3. Evaluation of the suggested algorithm for special cases of noise fields and TFs will be presented in Section 6.4 (distortion) and Section 6.5 (noise reduction).

6.1 Output Signal Power Spectrum

Using Eqs. 3.9,3.10 and 3.12 the algorithm's output is given by,

$$Y(t,e^{j\omega}) \;=\; Y_{\mathrm{FBF}}(t,e^{j\omega}) - \mathbf{G}^{\dagger}(t,e^{j\omega})\mathbf{U}(t,e^{j\omega})$$

$$= \frac{\mathcal{F}^*(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathbf{H}}^\dagger(e^{j\omega})\mathbf{Z}(t,e^{j\omega}) - \mathbf{G}^\dagger(t,e^{j\omega})\widehat{\mathcal{H}}^\dagger(e^{j\omega})\mathbf{Z}(t,e^{j\omega}),$$

where we assume that only estimates of the TFs' ratio, $\widehat{\mathbf{H}}(e^{j\omega})$ are given. Using this expression, we can calculate the power spectrum of the output signal,

$$\begin{aligned}
S_{oo}(t,e^{j\omega}) &= E\left\{Y(t,e^{j\omega})Y^*(t,e^{j\omega})\right\} \\
&= E\left\{(Y_{\mathrm{FBF}}(t,e^{j\omega}) - \mathbf{G}^\dagger(t,e^{j\omega})\mathbf{U}(t,e^{j\omega}))(Y_{\mathrm{FBF}}(t,e^{j\omega}) - \mathbf{G}^\dagger(t,e^{j\omega})\mathbf{U}(t,e^{j\omega}))^\dagger\right\} \\
&= E\left\{|Y_{\mathrm{FBF}}(t,e^{j\omega})|^2 - \mathbf{G}^\dagger(t,e^{j\omega})\mathbf{U}(t,e^{j\omega})Y^*_{\mathrm{FBF}}(t,e^{j\omega})\right. \\
&\left. -Y^*_{\mathrm{FBF}}(t,e^{j\omega})\mathbf{U}^\dagger(t,e^{j\omega})\mathbf{G}(t,e^{j\omega}) + \mathbf{G}^\dagger(t,e^{j\omega})\mathbf{U}(t,e^{j\omega})\mathbf{U}^\dagger(t,e^{j\omega})\mathbf{G}(t,e^{j\omega})\right\}.
\end{aligned}$$

The output spectrum depends on the input signal $\mathbf{Z}(t,e^{j\omega})$ and the optimal filter $\mathbf{G}(t,e^{j\omega})$. This optimal filter implements the multi-channel Wiener filter given by,

$$\mathbf{G}(t,e^{j\omega}) = S^{-1}_{\mathbf{UU}}(t,e^{j\omega})S_{\mathbf{U}Y}(t,e^{j\omega})$$

where,

$$\begin{aligned}
S_{\mathbf{U}Y}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})Y^*_{\mathrm{FBF}}(t,e^{j\omega})\} \\
S_{\mathbf{UU}}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})\mathbf{U}^\dagger(t,e^{j\omega})\}.
\end{aligned}$$

The filter is usually calculated during nonactive periods of the desired signal, i.e. $\mathbf{Z}(t,e^{j\omega}) = \mathbf{N}(t,e^{j\omega})$. Thus,

$$\mathbf{U}(t,e^{j\omega}) = \widehat{\mathcal{H}}^\dagger(e^{j\omega})\mathbf{N}(t,e^{j\omega}).$$

Calculating the involved spectra

$$\begin{aligned}
S_{\mathbf{U}Y}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})Y^*_{\mathrm{FBF}}(t,e^{j\omega})\} \\
&= E\left\{\widehat{\mathcal{H}}^\dagger(e^{j\omega})\mathbf{N}(t,e^{j\omega})\left(\frac{\mathcal{F}^*(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathbf{H}}^\dagger(e^{j\omega})\mathbf{N}(t,e^{j\omega})\right)^\dagger\right\} \\
&= \frac{\mathcal{F}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathcal{H}}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega})
\end{aligned}$$

and

$$\begin{aligned} S_{\mathbf{UU}}(t,e^{j\omega}) &= E\{\mathbf{U}(t,e^{j\omega})\mathbf{U}^{\dagger}(t,e^{j\omega})\} \\ &= E\left\{\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})\mathbf{N}(t,e^{j\omega})\mathbf{N}^{\dagger}(t,e^{j\omega})\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})\right\} \\ &= \widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega}) \end{aligned}$$

yielding the optimal Wiener filter,

$$\mathbf{G}(t,e^{j\omega}) = S^{-1}_{\mathbf{UU}}(t,e^{j\omega})S_{\mathbf{U}Y}(t,e^{j\omega}) = \frac{\mathcal{F}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\left(\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\right)^{-1}\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega}). \tag{6.1}$$

Now, we can calculate the output spectrum for any arbitrary input $\mathbf{Z}(t,e^{j\omega}) = \mathbf{S}(t,e^{j\omega})$. Later, using the independence of the desired signal and the noise signal, we will calculate the noise reduction and the distortion imposed by the algorithm. The spectrum of the desired signal part of the output is given by,

$$\begin{aligned} &S^{s}_{oo}(t,e^{j\omega}) = E\{Y^{s}(t,e^{j\omega})(Y^{s}(t,e^{j\omega}))^{*}\} = \\ &E\left\{\left(\frac{\mathcal{F}^{*}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})\mathbf{S}(t,e^{j\omega}) - \mathbf{G}^{\dagger}(t,e^{j\omega})\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})\mathbf{S}(t,e^{j\omega})\right) \times \right. \\ &\left.\left(\frac{\mathcal{F}^{*}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})\mathbf{S}(t,e^{j\omega}) - \mathbf{G}^{\dagger}(t,e^{j\omega})\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})\mathbf{S}(t,e^{j\omega})\right)^{\dagger}\right\} \end{aligned}$$

opening brackets and using the spectrum definition $S_{\mathbf{SS}}(t,e^{j\omega}) = E\{S(t,e^{j\omega})S(t,e^{j\omega})^{\dagger}\}$ yields,

$$\begin{aligned} S^{s}_{oo}(t,e^{j\omega}) &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^4}\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega}) \\ &- \frac{\mathcal{F}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\mathbf{G}^{\dagger}(t,e^{j\omega})\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega}) \\ &- \frac{\mathcal{F}^{*}(e^{j\omega})}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^2}\widehat{\mathbf{H}}^{\dagger}(t,e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\mathbf{G}(e^{j\omega}) \\ &+ \mathbf{G}^{\dagger}(t,e^{j\omega})\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\mathbf{G}(e^{j\omega}). \end{aligned}$$

Using the optimal filters $\mathbf{G}(t,e^{j\omega})$ (6.1) yields,

$$S^{s}_{oo}(t,e^{j\omega}) = \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\widehat{\mathbf{H}}(e^{j\omega})\|^4}\times \tag{6.2}$$

$$\Big\{\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega})-$$
$$\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\left(\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\right)^{-1}\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega})-$$
$$\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\left(\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\right)^{-1}\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega})+$$
$$\widehat{\mathbf{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\left(\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\right)^{-1}\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})$$
$$\widehat{\mathcal{H}}(e^{j\omega})\left(\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathcal{H}}(e^{j\omega})\right)^{-1}\widehat{\mathcal{H}}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\widehat{\mathbf{H}}(e^{j\omega})\Big\}.$$

This complicated expression depends on various parameters: the input signal spectrum, $S_{\mathbf{SS}}(t,e^{j\omega})$, the noise spectrum used for calculating the optimal filters, $S_{\mathbf{NN}}(t,e^{j\omega})$, and the TFs' ratio estimate $\widehat{\mathbf{H}}(e^{j\omega})$ (which are also used for the blocking matrix $\widehat{\mathcal{H}}(e^{j\omega})$).

6.2 Desired Signal Distortion

The distortion imposed by the algorithm can be calculated by using Eq. 6.2 for a signal with a vector of TFs $\mathbf{A}(e^{j\omega})$. Assume we have exact knowledge of the TFs' ratio $\mathbf{H}(e^{j\omega})$, i.e., $\widehat{\mathbf{H}}(e^{j\omega}) = \mathbf{H}(e^{j\omega}) = \frac{\mathbf{A}(e^{j\omega})}{A_1(e^{j\omega})}$. Thus, using the signal spectrum expression $S_{\mathbf{SS}}(t,e^{j\omega}) = S_{ss}(t,e^{j\omega})\mathbf{A}(e^{j\omega})\mathbf{A}^{\dagger}(e^{j\omega})$ and the identity $\mathcal{H}^{\dagger}(e^{j\omega})\mathbf{A}(e^{j\omega}) = 0$, we have

$$\begin{aligned} S^{s}_{oo}(t,e^{j\omega}) &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^{\dagger}(e^{j\omega})S_{\mathbf{SS}}(t,e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= S_{ss}(t,e^{j\omega})\frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^{\dagger}(e^{j\omega})\mathbf{A}(e^{j\omega})\mathbf{A}^{\dagger}(e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= S_{ss}(t,e^{j\omega})|\mathcal{F}(e^{j\omega})|^2|A_1(e^{j\omega})|^2. \end{aligned}$$

The filter $\mathcal{F}(e^{j\omega})$ is an arbitrary predetermined filter, so it should not be regarded as a distortion. The filter $A_1(e^{j\omega})$ is the TF from the signal source to the first (reference) sensor. It can not be eliminated by the algorithm. Actually, it imposes on the output signal the same amount of distortion imposed on the arbitrary reference sensor. Thus, we will define the total distortion of the algorithm by normalizing the output,

$$DIS(t,e^{j\omega}) = \frac{S^{s}_{oo}(t,e^{j\omega})}{|\mathcal{F}(e^{j\omega})|^2|A_1(e^{j\omega})|^2}.$$

In actual scenarios the filters $\mathbf{H}(e^{j\omega})$ are not known in advance, so estimation error might occur.

$$\widehat{\mathbf{H}}(e^{j\omega}) = \mathbf{H}(e^{j\omega}) + \mathbf{E}(e^{j\omega})$$

This estimation error has twofold influence. First, the FBF is not accurate, so it could degrade the alignment of the signal, causing noncoherent addition. Second, the blocking matrix, whose terms depend on $\mathbf{H}(e^{j\omega})$ estimate, would not block the desired signal completely, causing self-cancellation. In Chapter 4 we introduced the variance of the estimation error, and it is given by,

$$var\{E^m(e^{j\omega})\} = \frac{1}{BT}\frac{1}{SNR^m_{ave}(e^{j\omega})}\Xi(e^{j\omega}).$$

6.3 Noise Reduction

To calculate the noise reduction of the algorithm, we will use the general expression for the output signal given in Eq. 6.2 with a noise signal (the same noise signal used to calculate the optimal Wiener filter) as the input signal. Throughout this analysis we will assume perfect knowledge of the TFs' ratio, i.e., $\widehat{\mathbf{H}}(e^{j\omega}) = \mathbf{H}(e^{j\omega})$.

$$S^n_{oo}(t, e^{j\omega}) = S^n_{fbf}(t, e^{j\omega}) - S^{\dagger}_{\mathbf{UY}}(t, e^{j\omega}) S^{-1}_{\mathbf{UU}}(t, e^{j\omega}) S_{\mathbf{UY}}(t, e^{j\omega})$$

where, $S^n_{fbf}(t, e^{j\omega})$ is given by,

$$\begin{aligned} S^n_{fbf}(t, e^{j\omega}) &= E\{Y^n_{\mathrm{FBF}}(t, e^{j\omega}) Y^{n*}_{\mathrm{FBF}}(t, e^{j\omega})\} \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^{\dagger}(e^{j\omega}) S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}). \end{aligned}$$

An alternative and useful form for the noise component of the output signal is given by,

$$S^n_{oo}(t, e^{j\omega}) = S^n_{fbf}(t, e^{j\omega}) - \mathbf{G}^{\dagger}(t, e^{j\omega}) S_{\mathbf{UU}}(t, e^{j\omega})\mathbf{G}(t, e^{j\omega}).$$

Using all the precalculated terms we have,

$$\begin{aligned} S^n_{oo}(t, e^{j\omega}) = S^n_{fbf}(t, e^{j\omega}) - \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^{\dagger}(e^{j\omega}) S_{\mathbf{NN}}(t, e^{j\omega})\mathcal{H}(e^{j\omega}) \times \\ \left(\mathcal{H}^{\dagger}(e^{j\omega}) S_{\mathbf{NN}}(t, e^{j\omega})\mathcal{H}(e^{j\omega})\right)^{-1} \mathcal{H}^{\dagger}(e^{j\omega}) S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}). \end{aligned} \tag{6.3}$$

An interesting figure of merit is the extra noise reduction achieved by the noise cancelling branch (see also [15]),

$$NR_{nc}(t, e^{j\omega}) = \frac{S^n_{fbf}(t, e^{j\omega})}{S^n_{oo}(t, e^{j\omega})}. \tag{6.4}$$

6.3.1 Dependency on Noise Field

The resulting expression for the noise cancellation depends on the noise spectrum at the sensors. We will now calculate the expected noise reduction of the algorithm for three important noise fields: coherent (point source), diffused (spatially extended) and incoherent (noise signals generated at the sensors, e.g., amplifier noise, are assumed to uncorrelated).

Coherent Noise Field

Assume a single point source noise signal with power spectrum $S_{nn}(t, e^{j\omega})$ and assume that $b_m(t)$ are time varying transfer functions relating the noise source and the m-th sensor. Define,

$$\mathbf{N}(t, e^{j\omega}) = \mathbf{B}(e^{j\omega})N(t, e^{j\omega})$$

where,

$$\mathbf{B}^T(e^{j\omega}) = \left[\begin{array}{cccc} B_1(e^{j\omega}) & B_2(e^{j\omega}) & \cdots & B_M(e^{j\omega}) \end{array} \right].$$

So, the spectral matrix of the noise signal on the sensors is given by,

$$S_{\mathbf{NN}}(t, e^{j\omega}) = Snn(t, e^{j\omega})\mathbf{B}(e^{j\omega})\mathbf{B}^{\dagger}(e^{j\omega}) + \varepsilon I$$

where, I is an $M \times M$ identity matrix, and $\varepsilon \to 0$. The last term is added for stability reasons. For $\mathbf{B}(e^{j\omega}) = \mathbf{A}(e^{j\omega})$, the achievable noise reduction is infinite, i.e.,

$$S^n_{oo}(t, e^{j\omega}) = 0 \ \text{ for } \ \mathbf{B}(e^{j\omega}) = \mathbf{A}(e^{j\omega}).$$

Thus, perfect noise cancellation is achieved. The exact derivation is given in Appendix C. Note, that this is not a surprising result, as for $M \geq 2$ the Wiener filter can completely eliminate the noise component.

This result is valid for all TFs $\mathbf{B}(e^{j\omega})$ except for the signal direction TFs $\mathbf{B}(e^{j\omega}) = \mathbf{A}(e^{j\omega})$, that was analyzed in Section 6.2, yielding an output signal with spectrum:

$$S_{ss}(t, e^{j\omega})|\mathcal{F}(e^{j\omega})|^2|A_1(e^{j\omega})|^2.$$

This result shows that the suggested algorithm can eliminate any point source noise signal as well as the simple GSC, which assumes delay-only propagation, and can eliminate a directional noise signal (see, for instance [15]). It is also interesting to evaluate the noise part of the FBF branch,

$$\begin{aligned} S^n_{fbf}(t, e^{j\omega}) &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega})S_{nn}(t, e^{j\omega})\mathbf{B}(e^{j\omega})\mathbf{B}^\dagger(e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}S_{nn}(t, e^{j\omega})\mathbf{H}^\dagger(e^{j\omega})\mathbf{B}(e^{j\omega})\left(\mathbf{H}^\dagger(e^{j\omega})\mathbf{B}(e^{j\omega})\right)^\dagger. \end{aligned}$$

The noise level is not necessarily reduced, but signal transmitting from the desired signal source is added coherently by the array. The infinite noise reduction is achieved by the noise canceller branch.

Diffused Noise Field

In highly reverberant acoustical environment, such as a car enclosure, the noise field tends to be diffused (see for instance [18] and [22]). A diffused noise source is assumed to be equidistributed on a sphere in the far field of the array. The cross-coherence function between signals received by two sensors (i,j) with distance d_{ij} is derived in Appendix A and given in the following expression:

$$\Gamma_{Z_iZ_j}(e^{j\omega}) = \frac{S_{Z_iZ_j}(e^{j\omega})}{\sqrt{S_{Z_iZ_i}(e^{j\omega})S_{Z_jZ_j}(e^{j\omega})}} = \frac{\sin(\omega d_{ij}/c)}{\omega d_{ij}/c},$$

where c is the speed of sound. Thus, the coherence matrix is given by,

$$\Gamma(e^{j\omega}) = \begin{bmatrix} 1 & \Gamma_{Z_1Z_2}(e^{j\omega}) & \cdots & \Gamma_{Z_1Z_M}(e^{j\omega}) \\ \Gamma_{Z_2Z_1}(e^{j\omega}) & 1 & \cdots & \\ & & \ddots & \\ \Gamma_{Z_MZ_1}(e^{j\omega}) & & & 1 \end{bmatrix}.$$

The noise spectrum at the sensors input is thus,

$$S_{\mathbf{NN}}(t, e^{j\omega}) = S_{nn}(t, e^{j\omega})\Gamma(e^{j\omega}).$$

The noise spectrum at the FBF output is:

$$\begin{aligned} S^n_{fbf}(t, e^{j\omega}) &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega})S_{nn}(t, e^{j\omega})\Gamma(e^{j\omega})\mathbf{H}(e^{j\omega}). \end{aligned}$$

The extra noise reduction achieved by the noise canceller is given by,

$$\begin{aligned} NR_{nc}(t, e^{j\omega}) &= \frac{S^n_{fbf}(t, e^{j\omega})}{S^n_{oo}(t, e^{j\omega})} \\ &= \frac{1}{1 - \frac{\mathbf{H}^\dagger(e^{j\omega})\Gamma(e^{j\omega})\mathcal{H}(e^{j\omega})\left(\mathcal{H}^\dagger(e^{j\omega})\Gamma(e^{j\omega})\mathcal{H}(e^{j\omega})\right)^{-1}\mathcal{H}^\dagger(e^{j\omega})\Gamma(e^{j\omega})\mathbf{H}(e^{j\omega})}{\mathbf{H}^\dagger(e^{j\omega})\Gamma(e^{j\omega})\mathbf{H}(e^{j\omega})}}. \end{aligned}$$

This expression depends on the ratio of TFs $\mathbf{H}(e^{j\omega})$.

Incoherent Noise Field

We now assume that the noise at the sensors has no spatial correlation.

$$S_{\mathbf{NN}}(t, e^{j\omega}) = S_{nn}(t, e^{j\omega})I$$

$$\begin{aligned} S^n_{oo}(t, e^{j\omega}) &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}) - \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\times \\ &\quad \mathbf{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathcal{H}(e^{j\omega})(\mathcal{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathcal{H}(e^{j\omega}))^{-1}\mathcal{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t, e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^\dagger(e^{j\omega}) \times \\ &\quad \left\{S_{nn}(t, e^{j\omega})I - S_{nn}(t, e^{j\omega})\mathcal{H}(e^{j\omega})(S_{nn}(t, e^{j\omega})\mathcal{H}^\dagger(e^{j\omega})\mathcal{H}(e^{j\omega}))^{-1}\mathcal{H}^\dagger(e^{j\omega})S_{nn}(t, e^{j\omega})\right\} \\ &\quad \times\mathbf{H}(e^{j\omega}) \\ &= \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}S_{nn}(t, e^{j\omega})\mathbf{H}^\dagger(e^{j\omega})\left\{I - \mathcal{H}(e^{j\omega})(\mathcal{H}^\dagger(e^{j\omega})\mathcal{H}(e^{j\omega}))^{-1}\mathcal{H}^\dagger(e^{j\omega})\right\}\mathbf{H}(e^{j\omega}). \end{aligned}$$

The term $\mathbf{H}^\dagger(e^{j\omega})\mathcal{H}(e^{j\omega})$ can be calculated explicitly. Recall that,

$$\mathcal{H}(e^{j\omega}) = \begin{bmatrix} -H^*_2(e^{j\omega}) & -H^*_3(e^{j\omega}) & \dots & -H^*_3(e^{j\omega}) \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & & \dots & \ddots \\ 0 & 0 & \dots & 1 \end{bmatrix},$$

and

$$\mathbf{H}^{\dagger}(e^{j\omega}) = \left[\begin{array}{cccc} 1 & H_2^*(e^{j\omega}) & \cdots & H_M^*(e^{j\omega}) \end{array} \right].$$

Thus,

$$\mathbf{H}^{\dagger}(e^{j\omega})\mathcal{H}(e^{j\omega}) = \mathbf{0}_{1\times(M-1)}.$$

Furthermore, $\mathcal{H}^{\dagger}(e^{j\omega})\mathcal{H}(e^{j\omega})$ is a positive matrix, so its inverse always exists. Thus, the contribution of the noise cancelling branch is zero, and the noise reduction is due only to the fixed beamformer. The noise power at the output is thus:

$$S_{oo}^{n}(t, e^{j\omega}) = S_{nn}(t, e^{j\omega})\frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\mathbf{H}^{\dagger}(e^{j\omega})\mathbf{H}(e^{j\omega}) = S_{nn}(t, e^{j\omega})\frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^2}.$$

Again, no noise reduction is guaranteed by this structure, and the result depends on the TFs' ratio involved.

6.4 Performance Evaluation: Distortion

We saw that knowledge of the exact TFs yields a distortionless output. In this section, the influence of errors in estimating the TFs will be determined. We will evaluate the expected distortion imposed by the algorithm. The results depend on the desired signal's TFs and the noise field.

We will begin with the simple steering array, where only delay relates the sources and the sensors. Thus, the *direction of arrival* (DoA) of the sources completely determines the TFs. We will determine the performance degradation as a function of the steering error. Then, we will deal with the more general TFs' case. We will use the estimation error variance (derived in [44]) to predict the amount of distortion.

6.4.1 Signal's TFs: Pure Delay

Assume, free space propagation, i.e., only pure delay relates the desired signal source and the sensors. Assume, also, that the inter-element distance is 6 cm.

Directional Noise Signal

We optimize the array to cancel noise source from $\theta = 40^o$ (by optimization of the array we refer to designing the optimal Wiener filter in the noise cancellation branch). Assume that the desired signal impinges the array from $\theta = 90^o$. We will determine the distortion imposed as a function of the steering error, regardless of the TFs' identification method. In Figure 6.1 we present the spectrum of the output signal of the array as a function of the frequency and the steering angle. It is clearly shown that the output signal is distorted as we move the steering angle of the array away from the desired signal direction $\theta = 90^o$ in the range $\theta \in [80^o, 100^o]$. Note, that at the lower frequencies the influence of the steering errors is less notable, as there is no phase difference between the signals at the sensors. As shown, the imposed distortion can be as large as 10 dB. Note that distortion can also cause an increase in the output power, since unwanted components leak into the algorithm's output.

Diffused Noise Field

If the array is designed to work with a diffused noise field, the distortion imposed on a desired signal with $DoA = 90^o$ is given in Figure 6.2 Again, the distortion imposed is clearly seen. Note the symmetric behavior of the output spectrum as a function of the steering error. This is due to the inherent symmetry in the noise field.

Incoherent Noise Field

The same trend can be observed for an incoherent noise field, as seen in Figure 6.3.

6.4.2 Signal's TFs: Arbitrary

Although we have a closed form formula for estimating the TFs' ratio (see Chapter 4), calculating the predicted distortion of the algorithm for the case of arbitrary TFs is a complex task. First, the performance depends on the actual

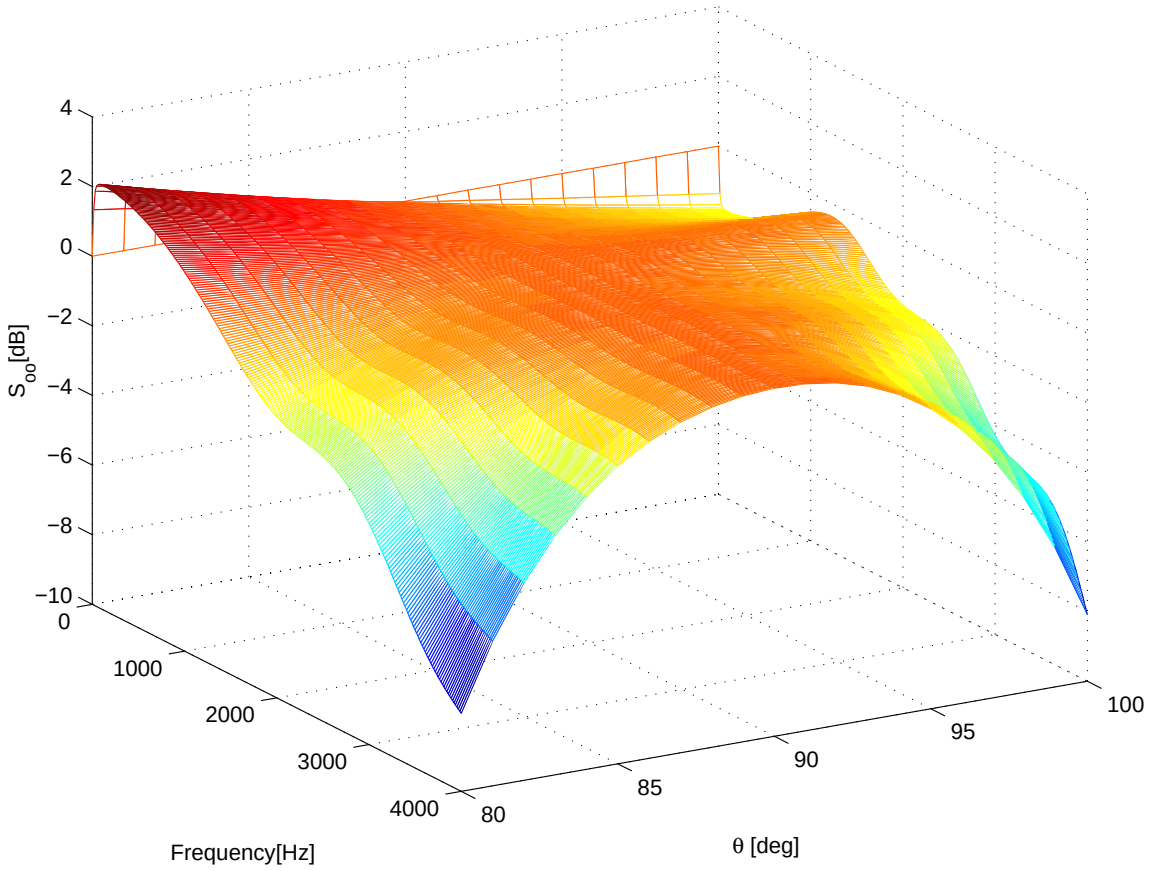

Figure 6.1: Distortion as a function of the frequency and direction of arrival. Desired signal direction $\theta = 90^o$. Nulled direction $\theta = 40^o$. $M = 5$ sensors. Inter-element spacing 6 cm.

TFs used. For that purpose we will use the acoustical TFs (ATFs), which will be presented in Chapter 7. The estimation error also depends on the estimation procedure used. We will check the suggested estimation procedure, based on signal nonstationarity. The estimated TFs are bias-free, but with increased variance in comparison with the standard LS method. The error variance is given by Eq. 4.2. Note, that to calculate the error variance we need the reference noise signals $u_m(t);\ m = 1, \ldots, M$. These signals themselves depend on the blocking matrix, which terms depend on the estimated TFs. Thus, the problem of predicting the distortion becomes nonlinear. Furthermore, each TF estimation depends on the

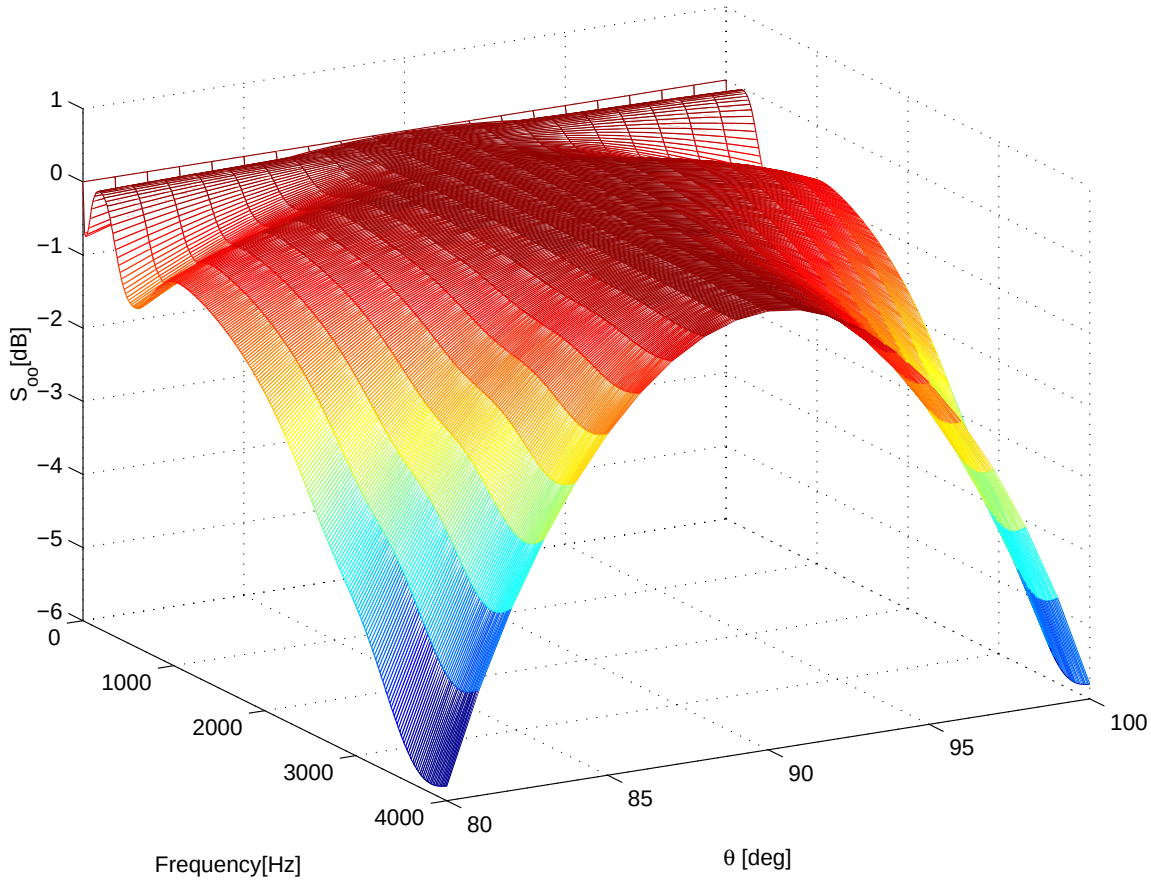

Figure 6.2: Distortion as a function of the frequency and direction of arrival. Desired signal direction $\theta = 90^o$. Diffused noise field. $M = 5$ sensors. Inter-element spacing 6 cm.

signal $z_1(t)$, resulting in correlation of the estimation errors.

Several simplifications and assumptions will enable a rough estimation of the expected distortion.

- The errors in estimating each TF are uncorrelated.
- The signals $u_m(t)$ are only slightly changed (due to TFs' estimation errors), so reestimating the predicted error with the new noise signals is unnecessary. This assumption states that we will use the exact TFs $H(e^{j\omega})$ for calculating the reference signals $u_m(t)$ spectrum, and use the resulting spec-

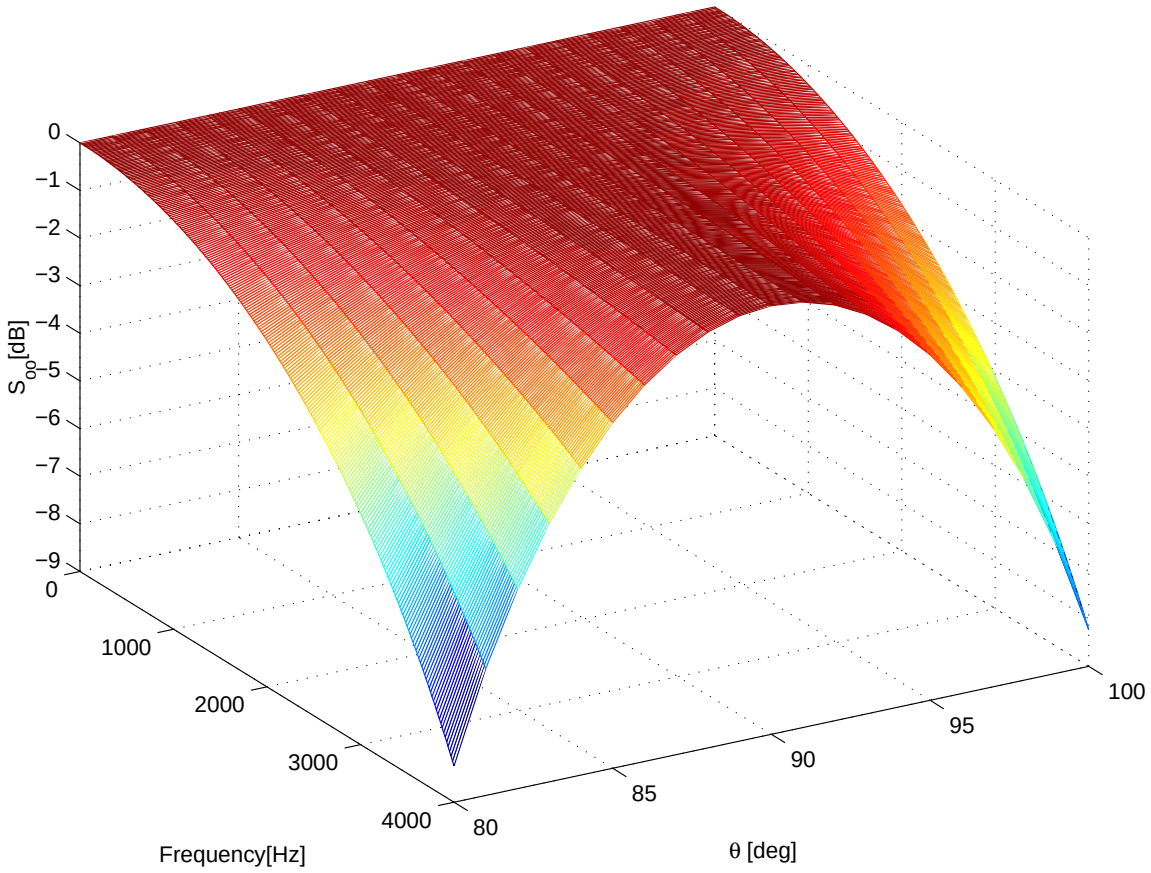

Figure 6.3: Distortion as a function of the frequency and direction of arrival. Desired signal direction $\theta = 90^o$. Incoherent noise field. $M = 5$ sensors. Inter-element spacing 6 cm.

tra to calculate the imposed error in estimating $H(e^{j\omega})$. This assumption may be referred to as *small error analysis*.

- The estimation error is Gaussian distributed with zero mean (unbiased estimate) and its variance is given by Eq. 4.2.

The estimation error spectrum depends on the signal nonstationarity. A typical nonstationary function for clean and noisy speech signals is given in Figure 6.4 (left side). It is clearly shown, that except for outliers in several frequencies, the clean signal is quite nonstationary (values range between 1 and 1.5). The noisy

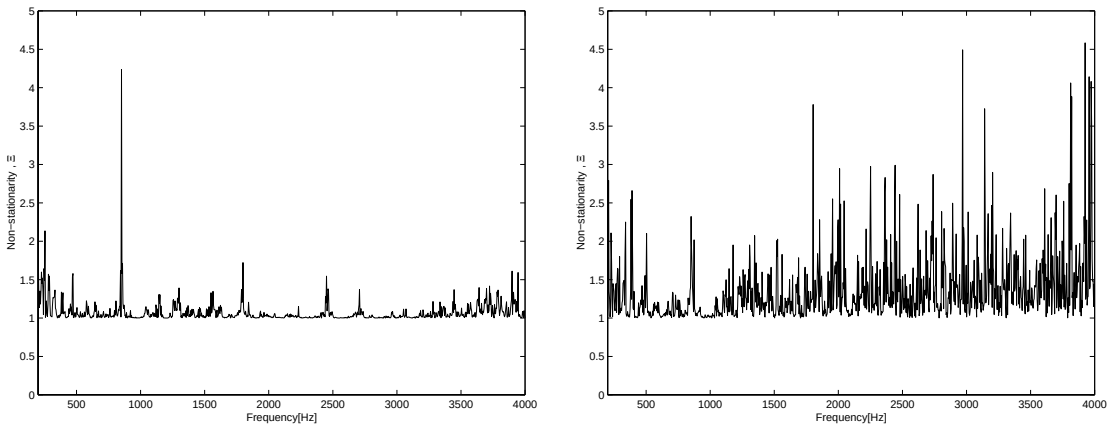

Figure 6.4: Typical nonstationarity function for speech signal. Clean (left). Noisy SNR=-5 dB (right).

signal becomes more stationary, yielding an increased value of $\Xi(e^{j\omega})$, as shown in Figure 6.4 (right side). The error variance is likewise dependent on the noise field. Our experiments show weak dependence on the noise field. Thus, we give only the results for noise field caused by ATF. The desired signal uses one set of ATFs (originating from left side of the array) and the noise signal uses another set (originating from right side of the array). Results are shown in Figure 6.5. Even for low input SNR of -5 dB, the predicted distortion is no more than 6 dB in the interesting frequency band. Although we made some restrictive assumptions, the low distortion predicted by this analysis may be noticed while applying our algorithm to a speech signal, as shown in Chapter 8.

6.5 Performance Evaluation: Noise Reduction

To evaluate the expected noise reduction of the algorithm, as we evaluated the predicted distortion, we should determine the TFs involved and the noise field. We will begin with the simple steering array, where only delay relates the sources and the sensors. Thus, the DoA of the sources determines completely the TFs. Then, we will deal with the more general TFs case. Throughout the noise reduction evaluation we will assume that the TFs' ratio are exactly known.

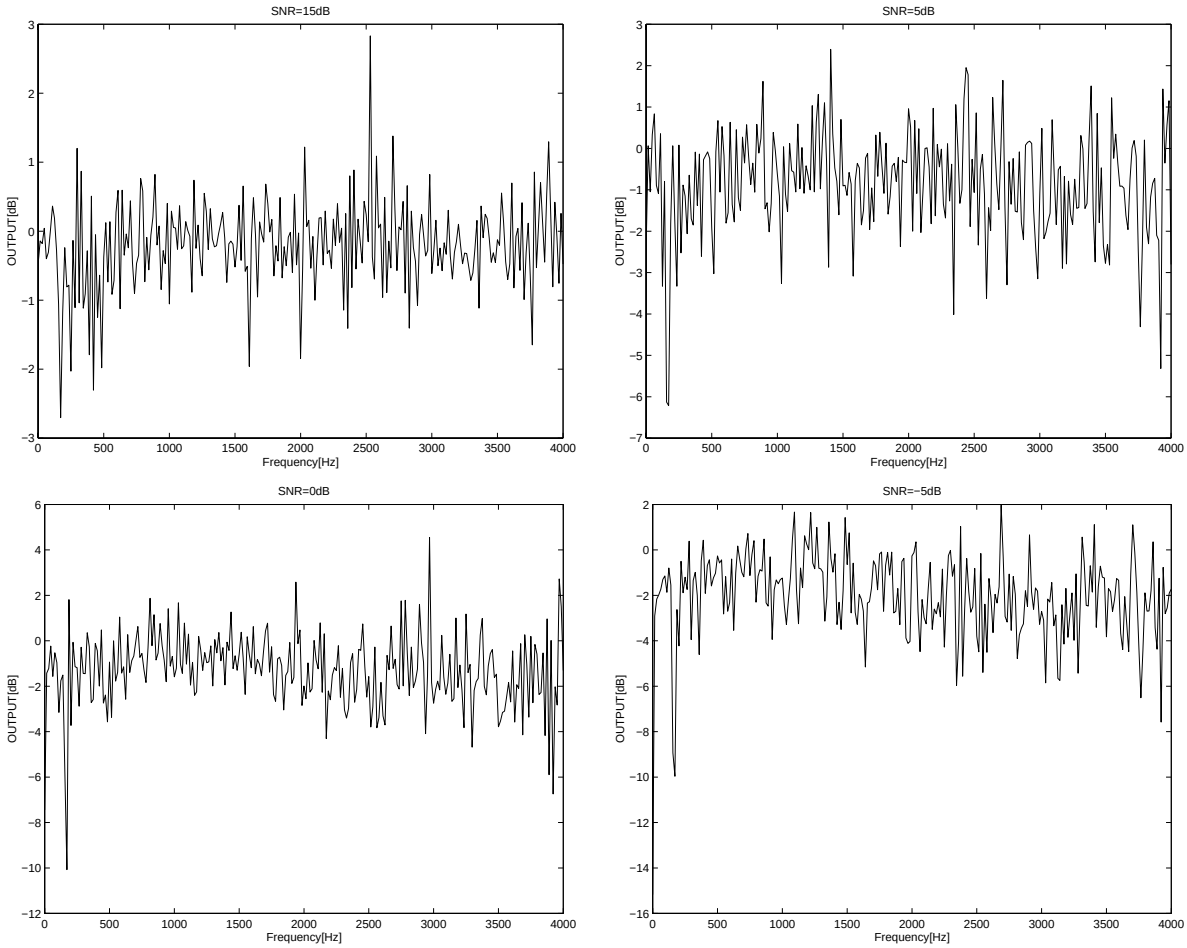

Figure 6.5: Distortion for arbitrary ATF.

6.5.1 Signal's TFs: Pure Delay

Assume, free space propagation, i.e., only pure delay relates the desired signal source and the sensors. Assume, also, that the inter-element distance is 6 cm.

Directional Noise Signal

We optimize the array to receive a signal from $\theta = 90^o$ and to cancel noise source from $\theta = 40^o$. In Figure 6.6 we present the output signal of the array as a function of the frequency and DoA. It is clearly shown that the main lobe is maintained, while a null is constructed in all frequencies at the noise angle. The main lobe is wider in the lower frequency band. This is with good agreement with the theory, since at $\omega = 0$ [rad/sec] there is no phase difference between the signals at the sensors.

Diffused Noise Field

In Figure 6.7 we evaluate the extra noise reduction of the noise canceller branch for four different steering angles as a function of the frequency and the number of sensors. Compare the results with [22] and [23]. The case of $M = 5$ sensors is shown in Figure 6.8 for various steering angles and the entire frequency band.

Incoherent Noise Field

No extra noise reduction is achieved by the noise canceller branch. The FBF branch becomes a simple *Delay & Sum* array, thus the expected noise reduction is M, the number of sensors.

6.5.2 Signal's TFs: Arbitrary

The more general case where arbitrary TFs relate the sources and the sensors is more complicated to introduce, as the expected performance depends on the actual TFs used.

For the evaluation of the general TFs case, we used acoustical TFs (ATFs) shown in Chapter 7. Three signal sources were used and received by 6 cm inter-

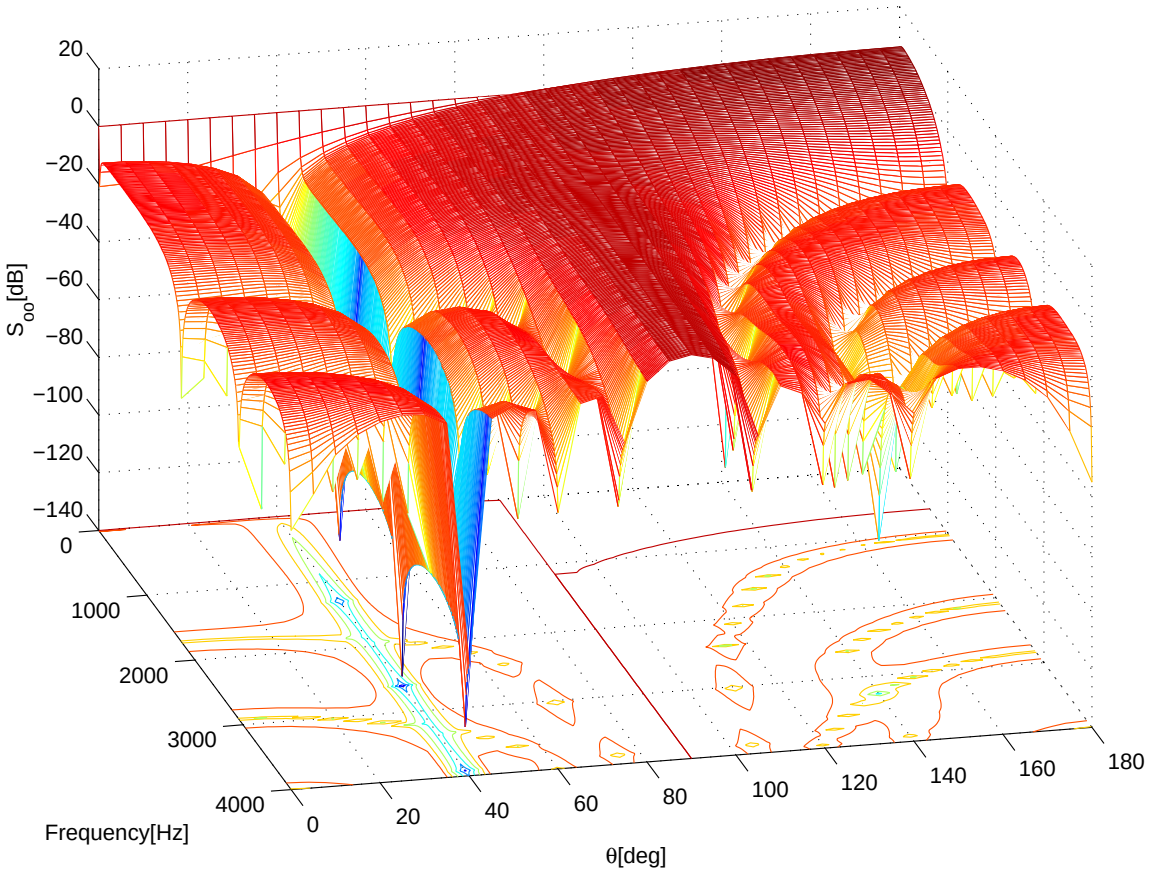

Figure 6.6: Output spectrum for white noise input as a function of the frequency and direction of arrival. Desired direction $\theta = 90^o$. Nulled direction $\theta = 40^o$. $M = 5$ sensors. Inter-element spacing 6 cm.

microphone distance array. The first source is located to the left of the array. The second source is located at the same source position, but with barrier placed between the source and the array. The third source is located to the right of the array. We will now analyze the dependence on the noise field.

Noise Field: Arbitrary TFs

The algorithm was designed (i.e., $\mathbf{H}(e^{j\omega})$ was determined) to receive signals from the left direction. We will present results for the three figures of merit introduced before: normalized FBF output, normalized total output and extra noise

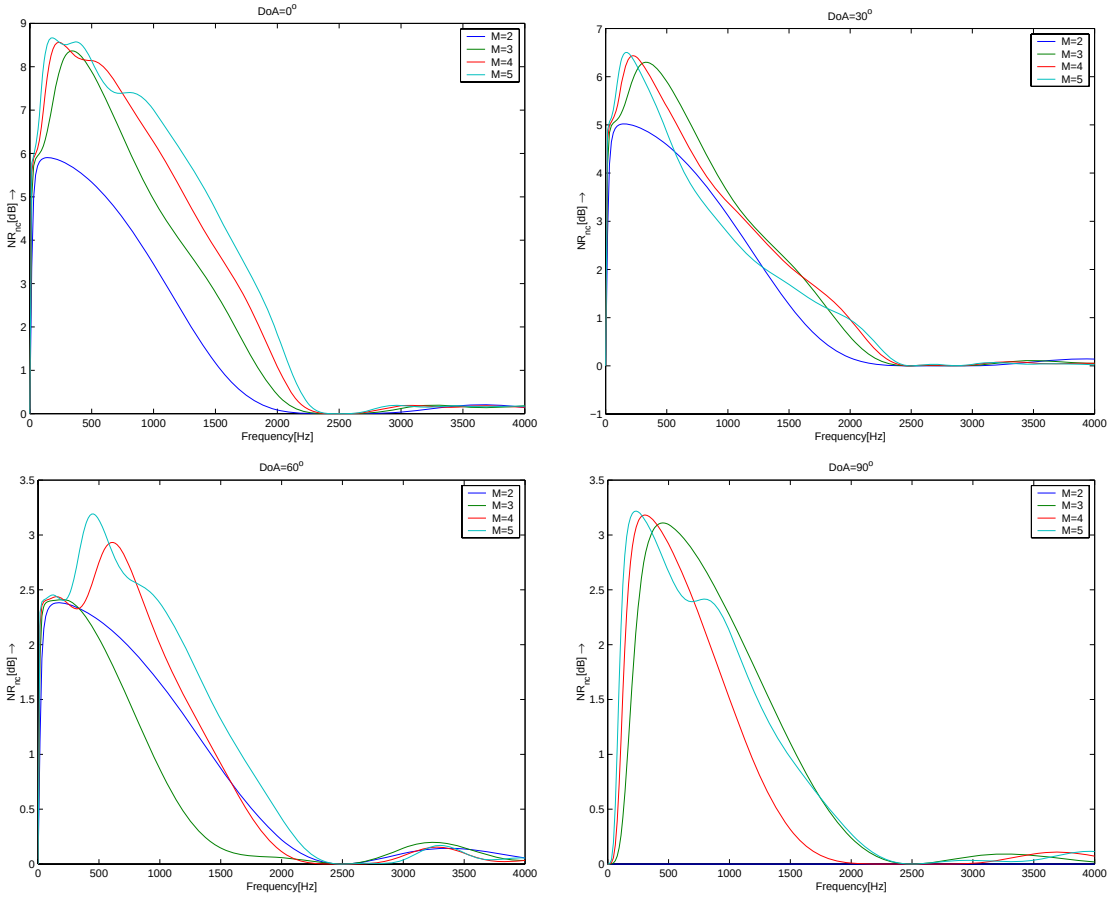

Figure 6.7: Noise reduction of noise cancelling branch for diffused noise field as a function of the steering angle and the number of sensors.

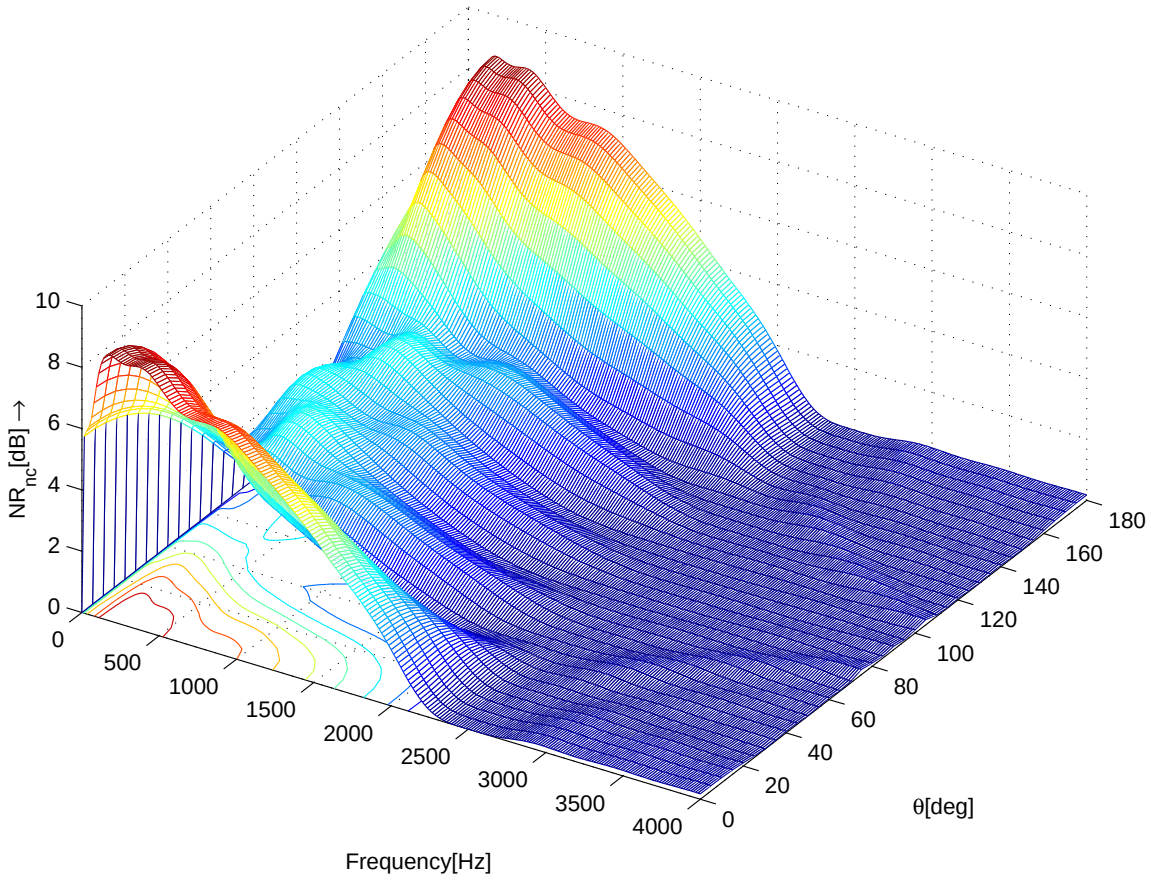

Figure 6.8: Extra noise reduction of noise cancelling branch for $M = 5$ sensors and for various steering directions.

reduction of the noise cancelling branch. If the TFs' ratio is known exactly, no attenuation is imposed on signals transmitted from the left direction. In Figure 6.9 the array response for a noise source transmitted from the right direction is introduced. Noise reduction of more than 70 dB is demonstrated.

If the signal is transmitted from the barriered left direction the amount of cancellation is far inferior, as can be seen from Figure 6.10. This fact demonstrates the sensitivity to errors in estimating the TFs' ratio.

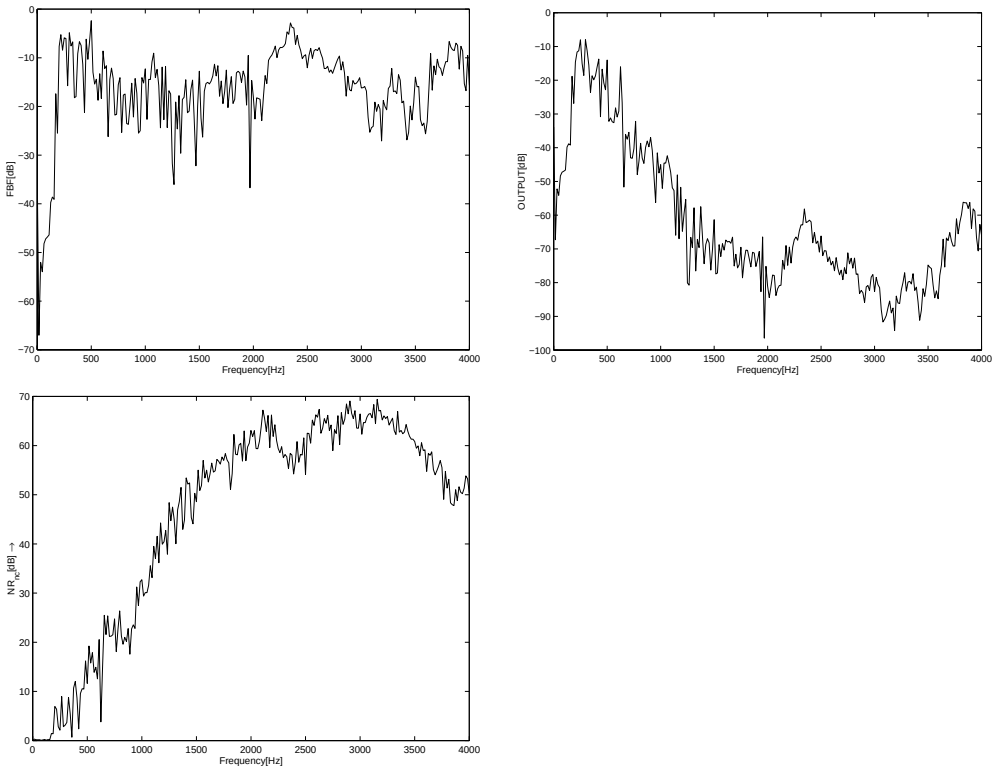

Figure 6.9: Expected performance - array designed for "left" originating signals. Signal received from "right" direction.

Diffused Noise Field

If the noise field is diffused, the performance of the array while designed to receive signals from the three directions is given in Figures 6.11, 6.12, 6.13 for the "left", "left with barrier" and "right" directions, respectively.

It is clearly shown, that the expected performance of the algorithm is degraded in comparison to the point source noise field. Although, a considerable noise reduction is achieved in the low frequencies, we should state that, there is almost no signal in this band, as can be shown from the typical ATF introduced in Figure 7.2

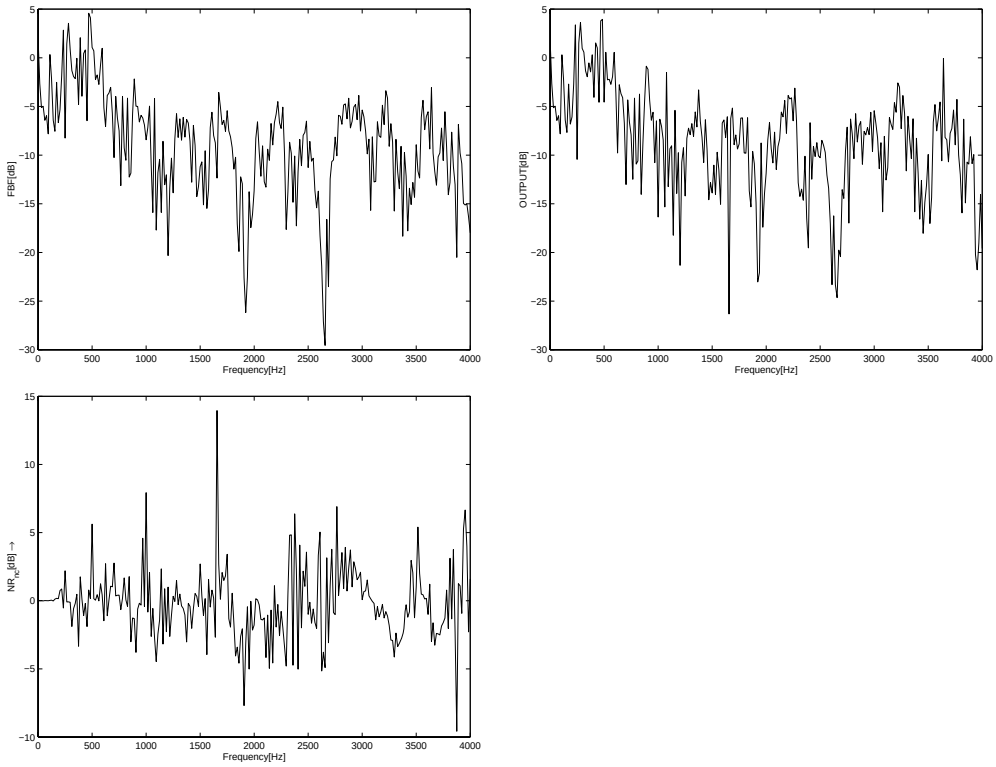

Figure 6.10: Expected performance - array designed for "left" originating signals. Signal received from "left with barrier" direction.

Incoherent noise field

In an incoherent noise field, there is no extra noise reduction achieved by the noise cancelling branch. The only noise reduction is caused by the FBF branch. The normalized FBF output for the array designed to receive signals from the three directions is shown in Figure 6.14. Almost no noise reduction is achieved.

6.6 Chapter Summary

We have derived a general expression for the output signal spectrum. From this expression we derived expressions for the expected distortion imposed on the desired signal and the amount of achievable noise reduction. The distortion depends on the quality of estimating the ratio of TFs. The noise reduction

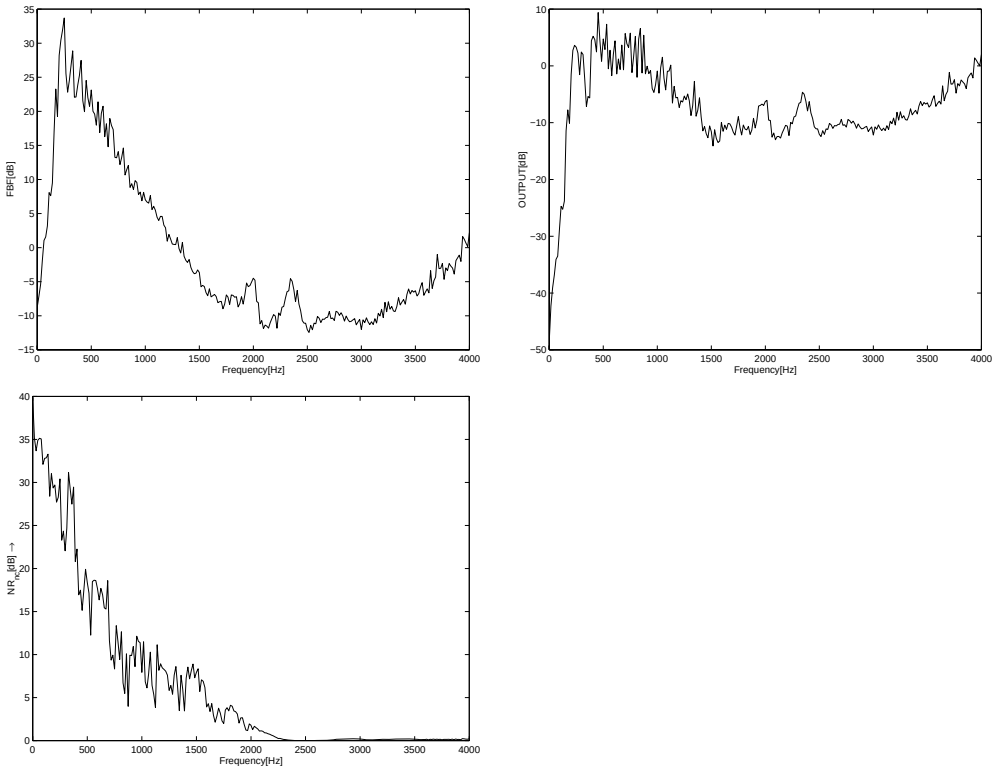

Figure 6.11: Expected performance Array designed for "left" originating signals. Diffused noise field.

depends on the TFs' ratio involved and the noise field. High noise reduction is achieved in a coherent noise field. In an incoherent noise field only the FBF branch is responsible for noise reduction. The amount of noise reduction depends on the TFs' ratio involved, and it might be negligible. In a diffused noise field we expect a significant noise reduction only in the lower frequency band, but recalling that there is almost no desired signal in that band, the amount of noise reduction may also be almost negligible. We note that in actual scenarios the noise field is a mixture between point sources and diffused sources, thus the expected result should fall between these two extremes. Recall that in activating the suggested algorithm we use sequential update for the filters involved rather than the closed form optimal solution. This might degrade the performance of the algorithm, yielding inferior results than predicted in this chapter.

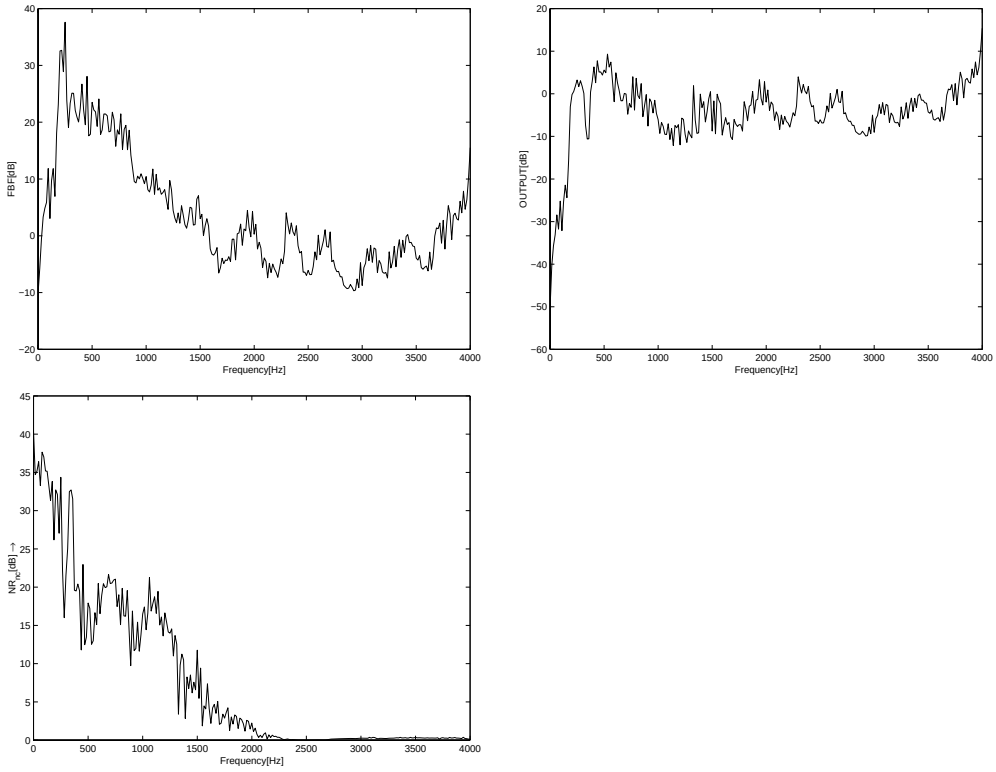

Figure 6.12: Expected performance array designed for "left with barrier" originating signals. Diffused noise field.

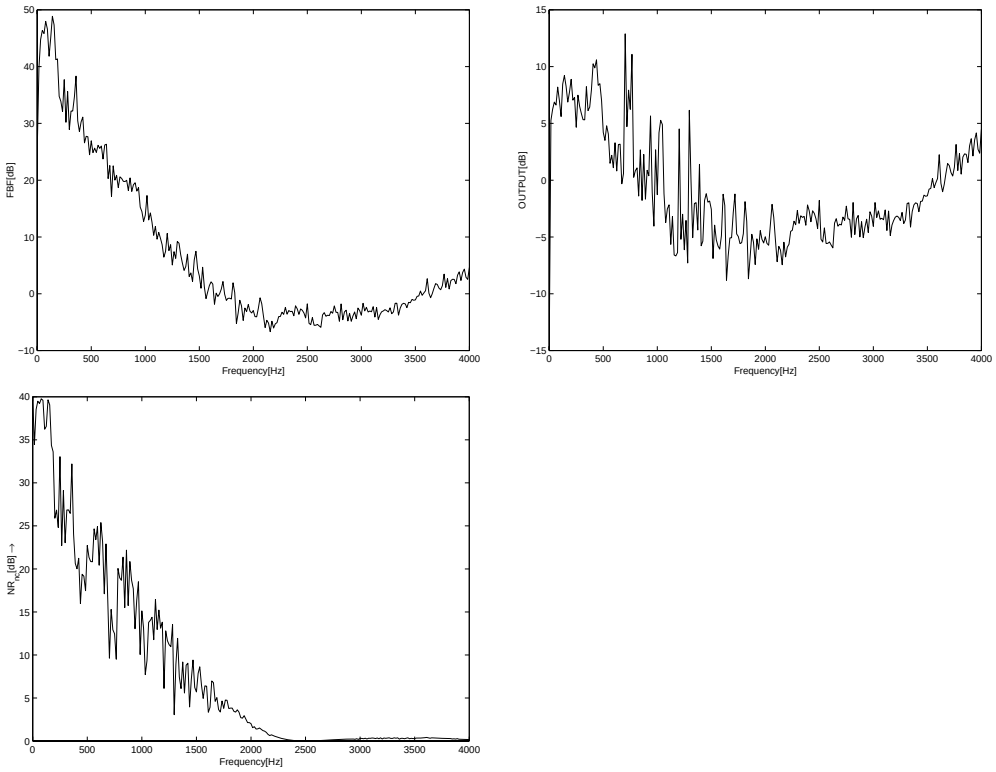

Figure 6.13: Expected performance - array designed for "right" originating signals. Diffused noise field

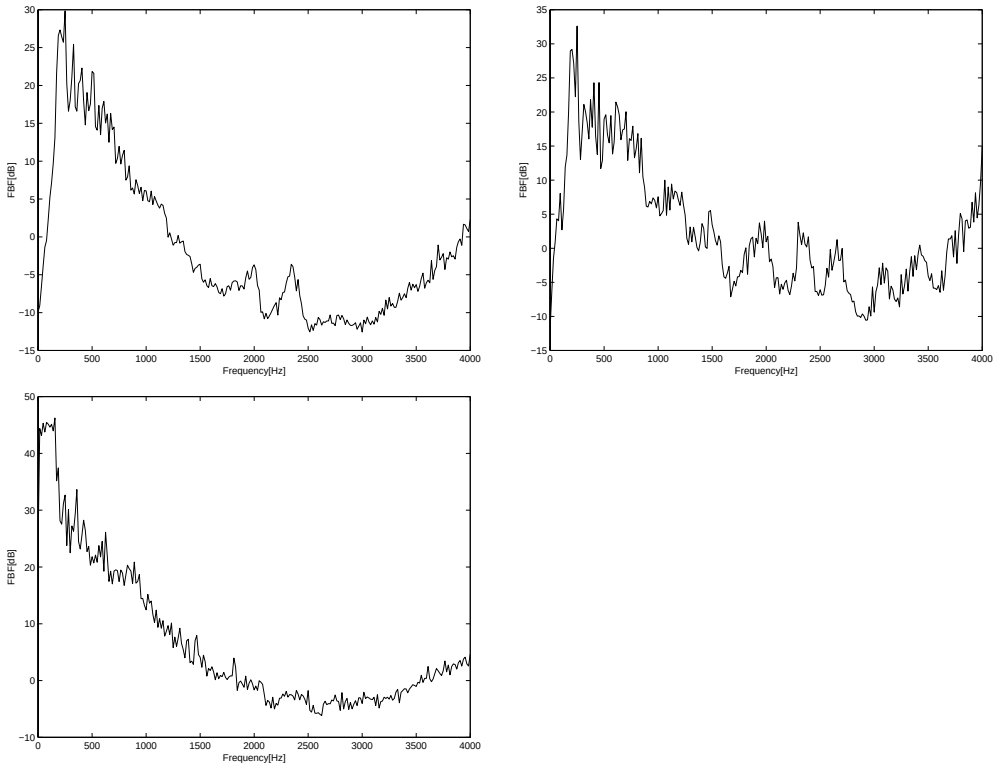

Figure 6.14: Expected performance - array designed for "left" originating signals. Incoherent noise field.

Chapter 7

Application to Speech in Acoustical Environment

The suggested beamformer deals with a general TF relating the signal and noise sources and the sensor array. The primary desired signal characteristics exploited by the algorithm is its nonstationarity. For this reason, a natural application of the algorithm is the speech signal in reverberant enclosure. In this chapter we present some details related to the speech signal in an acoustical environment contaminated by noise signals, showing the reason why the suggested algorithm for signal enhancement can be applied to the problem of speech enhancement.

In Section 7.1 we introduce the speech signal. The acoustical environment is presented in Section 7.2. Various noise fields are presented in Section 7.3. It is observed that the output of the beamformer can be further processed to yield much better results. Therefore, we summarize some important single microphone speech enhancers and present two novel approaches in Section 7.4.

7.1 Speech Signal

As speech signal is generally referred to as quasi-stationary signal. The speech production model is responsible for these nonstationary characteristics. A common speech production model divides the operation into two functions, *excitation* and *modulation*. Excitation takes place mostly at the glottis, and modulation is

imposed by various organs of the vocal tract. Excitation can be performed by *phonation*, i.e., oscillation of the vocal cords, or by forcing airflow from the lungs. Vocal cord excitation is modeled by an impulse train. Airflow is represented by a white noise source. The vocal tract also changes rapidly as the mouth and tongue move. The vocal tract is modeled by an all-pole filter and represented by the LPC parameters. It is generally assumed that the waveform of the speech is stationary for approximately 20 mSec and then changes to the next syllable. The speech signal also includes silent periods, where no sound is heard. These periods can be short, such as between sentences, or relatively long, as in conversation between several talkers.

7.2 Acoustical Environment

When a speech signal propagates in an enclosed area, such as a room or a car cabinet, it becomes reverberated. Reverberation is imposed by reflection of the signal from objects, furniture, walls or windows. The reverberation can be interpreted as an acoustical transfer function (ATF), which is usually modeled as an FIR filter, although very long ($1000-2000$ coefficients in a medium-sized room, $200-300$ coefficients in a car, for 8KHz sampling rate). Allen and Berkley [56] suggest simulating room acoustics by using the image method. By this model, the talker is modeled as a point source in a rectangular cavity. The reflecting boundary may be replaced by placing an image symmetrically on its far side. In the general case of six walls, each image is itself imaged. In each reflection an attenuation exists due to absorption by the wall. Thus an infinite decaying series of reflections constitutes the entire response. Peterson [57] extended the model to cope with noninteger delays, by using band-limited pulses. Although, this model provides a good understanding of the complexity and length of the ATFs, proving that the free-space propagation is a too naive assumption for the source-array ATFs, we preferred using more realistic ATFs. Actual ATFs can be recorded from real data. The scenario shown in Figure 7.1 was used. To estimate

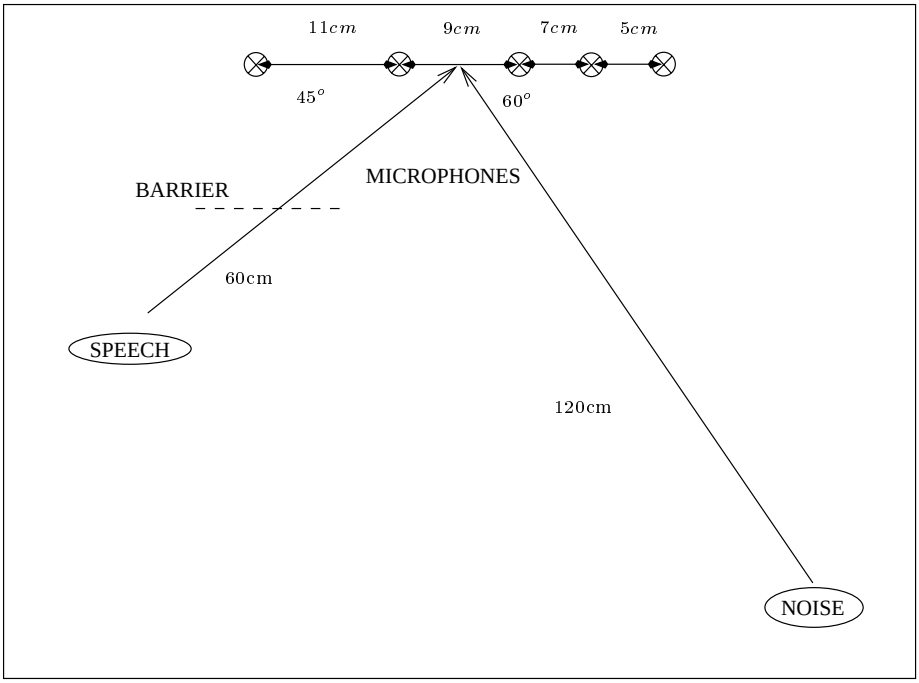

Figure 7.1: Test scenario: a five-microphone array in a noisy conference room.

realistic ATFs we positioned two loudspeakers in a small conference room (with dimensions $5 \times 4 \times 2.8m^3$). One of loudspeakers transmitted (clean) speech and the other a noise signal. The distance between the speech loudspeaker (left) and the center of the array is 60cm and its angle is 45^o, the distance between the noise loudspeaker (right) and the center of the array is 120cm and its angle is 60^o, as depicted in Figure 7.1. The signals were received (separately) by five simple omnidirectional carbon microphones. The microphone signals were sampled and recorded synchronously by the computer. Two other recording channels were dedicated to the original signals (speech and noise).

A conventional LS fit between each original signal and each of the received signals was used to estimate the ATFs (altogether we estimated five ATFs from the left position to the five microphones and another set of five ATFs from the right position to the microphones). These estimated ATFs include the microphone transfer function but exclude the loudspeaker transfer function. Another

set of five ATFs was estimated in the same manner for the situation in which a barrier (a simple screen) was mounted approximately half the way between the left loudspeaker and the microphone array. The exact geometry is shown in Figure 7.1.

The estimated ATF between the left source and one of the microphones (both time and frequency domain) is presented in Figure 7.2. This ATF is typical for a medium-sized room ATF.

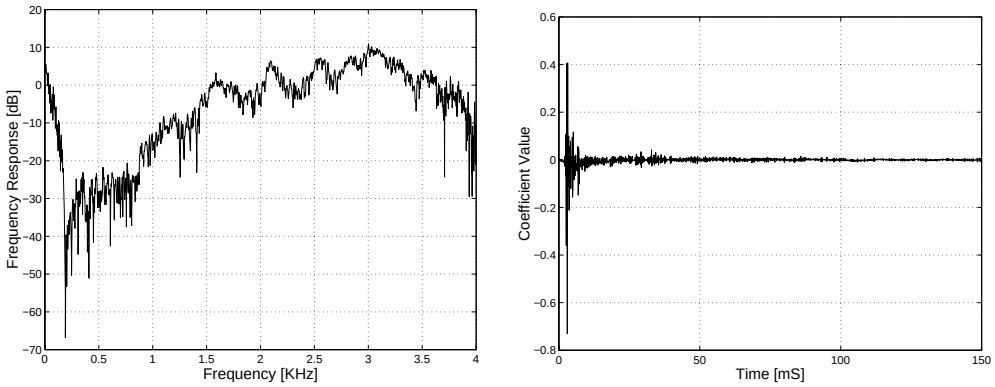

Figure 7.2: Typical Acoustic transfer function.

A common measure for the reverberation time of an enclosure is the *energy decay curve*. Let the *energy decay curve* (EDC) corresponding to some impulse response, $a(t)$, be defined by [41]

$$\mathrm{EDC}(t) \triangleq \sum_{\tau=t}^{\infty} a^2(\tau)$$

The point where the EDC slope changes abruptly is called *total duration* (TD). The *clarity index* is defined by

$$C(a) \triangleq \frac{\mathrm{EDC}(t=0)}{\mathrm{EDC}(t=\mathrm{TD})}$$

In Figure 7.3 we show the EDC of the impulse response between the speech source and the first microphone. The corresponding clarity index is 6.7dB which indicates a reverberated environment [41].

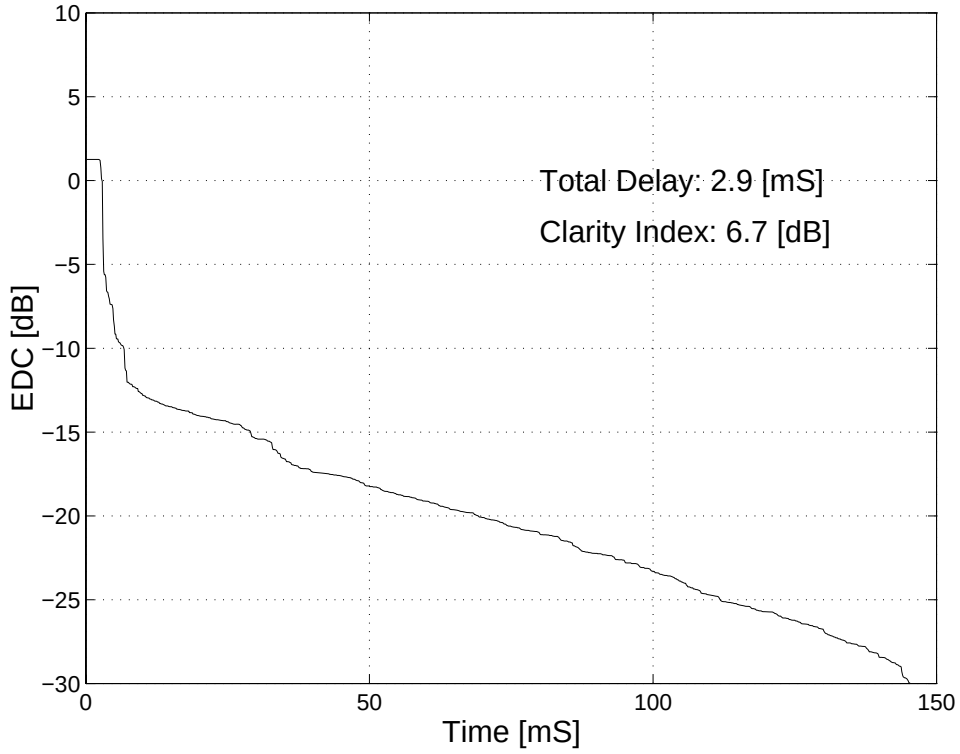

Figure 7.3: EDC of typical Acoustic transfer function.

To show the relative ATF between the different microphone we draw in Figure 7.4 the first 100 coefficients (12.5mSec) of the five ATFs in the three positions. The relative delay of the strongest arrival between the different microphones can be easily encountered for the left and right positions. The situation is less clear when the barrier was mounted, since the direct path is attenuated.

Although the ATFs change over time, the rate of change is slower than the rate of change of the speech signal. Thus, it enables us to exploit the nonstationarity of the speech signal to improve the estimate of the ATFs.

Thus, the speech enhancement problem in a reverberating enclosure is a good candidate for application of the suggested algorithm.

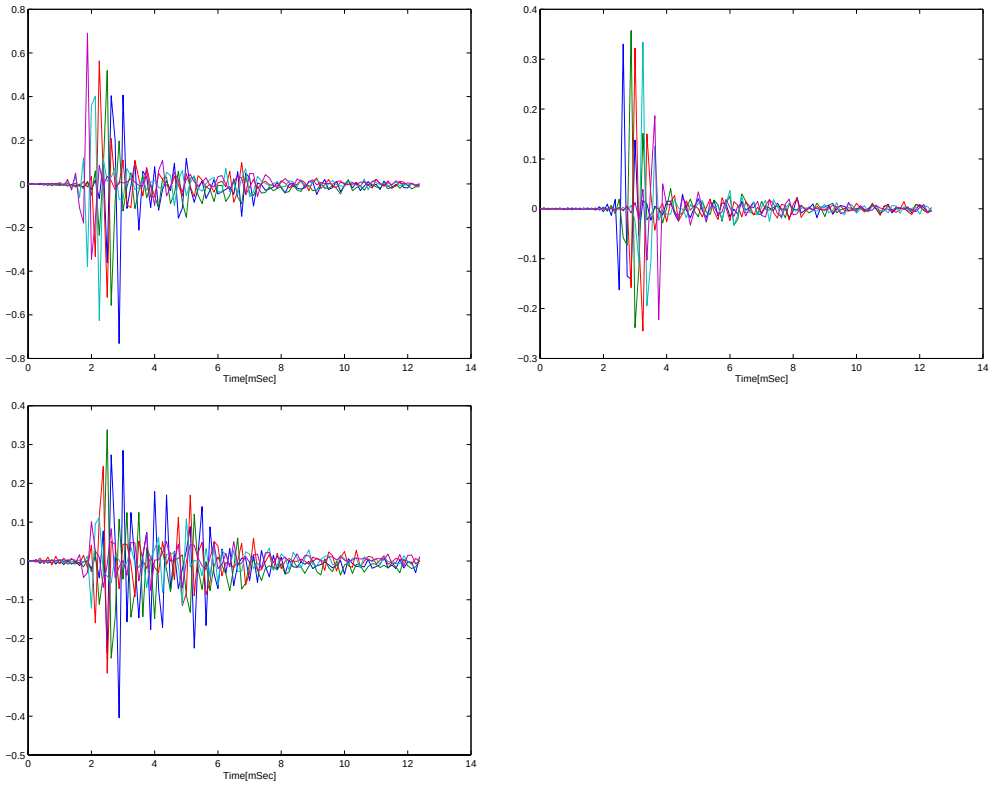

Figure 7.4: Relative acoustic transfer function for three positions: "left" (upper left), "right" (upper right) and "left barriered" (lower left).

7.3 Noise Field

In a reverberating enclosure, the noise sound propagation also experiences reflections from rigid objects. For applying the system identification method presented in Chapter 4 the noise signal should be more stationary than the speech signal. Thus, the identification accuracy degrades severely, when the noise signal is a competing speech, but improves when the interfering signal is a mixture of several speakers (*Cocktail party*). The best type of noise signal is an almost stationary one.

Typically, three sorts of stationary noise signal are considered. The first is a simple *incoherent* noise source, which normally originates from thermal noise produced by the amplifiers, and is assumed be low level and have no spatial

correlation between microphones.

The second is a point source noise, which similar to the speech signal, has ATFs relating the source and each microphone.

The third is a diffused noise source. In a highly reverberant enclosure, or in situation where large amount of uncorrelated noise sources from different directions impinge the array, the noise field is modeled as spherically distributed set of uncorrelated wave fronts. The coherence function between microphones is derived in Appendix A, based on Dal-Degan and Prati [18] and Goulding and Bird [58] derivations, and is given by,

$$\Gamma(\omega) = \frac{\sin(\omega d/c)}{\omega d/c}.$$

7.4 Single Microphone Speech Enhancer

We note that the output of the algorithm is one channel with, hopefully, improved SNR. Thus, we can activate, as a post-processor, any single microphone speech enhancement algorithm. Applying post-processors to the multimicrophone speech enhancer was suggested in several works (e.g. [16],[19],[20]), generally in conjunction with the beamformer. We suggest the use of a separate single microphone speech enhancer in the post-processing stage. We give a brief summary of some important single microphone algorithms. We then introduce two novel approaches: one in the time domain and the other in the frequency domain.

7.4.1 Introduction

Speech enhancement algorithms have attracted a great deal of interest in the past two decades [59], [60], [61], [62], [63], [28], [64], [65], [66], [67], [68], [69].

Speech enhancement algorithms may be broadly classified as belonging to one of the following two categories. The first is the class of time domain, parametric, model-based methods. Here, the general approach is to fit time domain models for the speech and noise signals, and then utilize these models for speech enhancement. Lim and Oppenheim [66] have suggested modeling the speech signal

as a stochastic auto-regressive (AR) model embedded in additive white Gaussian noise. This model has been extended by several authors, including [67], [64], [65] and [70] by generalizing it, using state-space formulations. These methods consist of two stages. The first is a parameter estimation stage, in which we fit model parameters. Parameter estimation is followed by the speech enhancement stage, which is usually implemented by utilizing Wiener or Kalman filtering. It should be noted though, that some of the algorithms produce the enhanced speech signal as a by-product of the parameter estimation stage.

The second class of speech enhancement algorithms is the class of spectral domain algorithms. A subset of this class is the popular spectral subtraction-based algorithms, e.g., [59], [69]. Essentially, spectral subtraction may be viewed as a non-parametric approach, in which the estimated noise spectrum is subtracted from the noisy spectrum of the speech, to produce the spectrum of the enhanced speech. Other spectral domain algorithms include the short time spectral amplitude (STSA) estimator, and the log spectral amplitude estimator (LSAE), both proposed by Ephraim and Malah [60], [61], and the hidden Markov model (HMM)-based filtering algorithms proposed by Ephraim *et al.* [62], [63].

In general, the computational requirements of the spectral domain algorithms are lower than the computational requirements of the time domain algorithms. This property makes spectral domain algorithms attractive candidates, especially for low cost and/or low power (e.g., battery operated) applications, such as cellular telephony.

7.4.2 A Novel Time Domain Algorithm

In [70],[27],[28] we presented iterative-batch and sequential speech enhancement algorithms in the presence of colored background noise. The iterative-batch algorithm employs the EM method to estimate the spectral parameters of the speech signal and noise process. Each iteration of the algorithm is composed of an estimation (E) step and a maximization (M) step. The E-step is implemented by using the Kalman filtering equations. The M-step is implemented by using a

non-standard YW equation set, in which correlations are replaced by their a-posteriori values, that are calculated by using the Kalman filtering equations. The enhanced speech is obtained as a byproduct of the E-step. A distinct advantage of the proposed algorithm, compared to alternative algorithms is that it enhances the quality and SNR of the speech, while preserving its intelligibility and natural sound. Another advantage of the algorithm is that a *voice activity detector* (VAD) is not required.

Forth order cumulant-based equations are shown to provide a reliable initialization to the EM algorithm. Alternative initialization methods that we tried, such as third order statistics-based equations, were not as effective.

The iterative-batch EM algorithm requires the use of an analysis window over which the signal and noise statistics are assumed to be stationary. To avoid this assumption, we suggested a sequential speech enhancement algorithm which is no longer an EM algorithm. The resulting sequential algorithm is more computationally efficient than the iterative-batch algorithm. Another benefit of the sequential algorithm is that it is delay-less, unlike the iterative-batch algorithm that has an inherent delay of one processing window frame. Although in general, the performance of the iterative-batch algorithm is superior, at low SNRs the differences in performance are small.

7.4.3 A Novel Frequency Domain Algorithm

In [29], [30] we presented a spectral domain algorithm, which produces high quality enhanced speech on the one hand, and has low computational requirements on the other.

The algorithm assumes that the log-spectral speech (frame) vector, X, can be modeled by a mixture of diagonal covariance Gaussians. It also assumes that the log-spectral noise vector, Y, can be modeled by a single diagonal covariance vector Gaussian. We assume an additive colored noise model. Hence the spectrum of the noisy speech is equal to the sum of the spectra of the clean speech and noise signals. Consequently, the log-spectral noisy speech vector, Z, can be approxi-

mated by max(X,Y) (the maximum assumption). The maximum assumption has been used in the past to devise noise-robust speech recognition algorithms. We use the same maximum assumption to derive a speech enhancement algorithm which is both powerful and computationally efficient.

Suppose that a mixture model for X has been obtained using some clean speech database. Suppose also that based on the voice inactivity periods of the input noisy speech vector (or the noise only reference signals obtained from the multi-microphone speech enhancer), we obtain an adaptive estimator to the density of the log-spectral noise, Y. Using the mixture modeling and the maximum assumption we derive a simple, closed form estimator to X given Z.

Several modifications and simplifications were made to this algorithm. It was found that a common variance component may be used for the k-th spectral component in the mixture model, regardless of the mixture, without any noticeable degradation in the performance of the algorithm. Consequently, assuming M mixtures and K spectral components, only K parameters are required to represent the set of variances (instead of KM). It was also found that non-linear post processing (in fact, simple limiting of the suppression at each spectral component) at the output of the algorithm was very effective in terms of speech quality. In addition, it was found useful to use a dual codebook mixture model for the speech signal, in order to reduce the amount of model parameters required for a reliable representation. The first codebook is related to the gain normalized log-spectral vector. The other codebook is related to the gain (energy) of the speech frame.

The performance of the algorithm was compared to alternative speech enhancement algorithms, such as non-linear spectral subtraction, minimum mean square error (MMSE) algorithms that utilize hidden Markov modeling (HMM), and Kalman filter-based speech enhancement algorithms. The last two algorithms are significantly more complicated than the new proposed algorithm. Both objective (SNR and other measures) and subjective listening tests were employed. It was found that with as few as 5 mixtures the algorithm outperforms all these alternative algorithms in terms of both the objective and subjective measures. The

algorithm was tested as a stand-alone speech enhancer (see Appendix D), as well as a post-processor for multichannel speech enhancer. The improvement in SNR in the post-processor application was as high as 13dB, for fan noise recorded in a reverberating room. For complete results, see Chapter 8. For a detailed description of the proposed algorithm (nicknamed MIXMAX), see Appendix D.

7.5 Chapter Summary

In this chapter we showed that a speech signal in a reverberant environment is a natural candidate for our suggested algorithm. We introduced the speech signal, the acoustical transfer function (ATF) and various noise fields. In the context of speech enhancement a single microphone post-processor can be applied. We have introduced two novel single microphone approaches, one in the time domain and the other in the frequency domain.

Chapter 8

Experimental Results

In this chapter we apply the suggested algorithm to the speech enhancement problem and evaluate its performance. The test scenario is introduced in Section 8.1. The results for various noise fields and ATFs are presented in Sections 8.2.1, 8.2.2 and 8.2.3. Results for the suggested time domain methods (including the use of the decorrelation criterion) are introduced in Section 8.3. Comparison with the conventional, time-domain, delay-only Griffiths & Jim algorithm [5] are presented in Section 8.4. We conclude this chapter in Section 8.5 by presenting the computational burden and memory requirements imposed by the suggested algorithm and compare them to the conventional algorithm's figures.

8.1 Test Scenario

The scenario shown in Figure 7.1 was studied. The enclosure is a small sized conference room (with dimensions $5 \times 4 \times 2.8m^3$). A linear array was mounted on a table. Two loudspeakers were used: one for the speech source and the other for a point noise source. The distances and angles are marked in Figure 7.1. The speech source constitutes of four TIMIT sentences with various gain levels. We used several noise types. The first is a computer-generated white Gaussian noise signal transmitted by the loudspeaker. The second was a fan noise transmitted by the loudspeaker. The third is computer-generated diffused noise source, high-pass filtered above 500Hz (see Appendix A). The fourth was a computer generated

spatially uncorrelated noise source. The speech signal and first two types of noise signals were transmitted *separately* and received by the five-microphone array, as explained in Section 7.2. These real recordings were mixed at various SNR levels during the evaluation. The ATFs presented in Section 7.2 were used for demonstrating the reverberant environment, as well as for analyzing the expected performance (introduced in Chapter 6), but not during the experimental performance evaluation introduced in this chapter..

To further improve the performance, we applied a single microphone speech enhancement algorithm (we chose to use the frequency domain algorithm suggested in [29] and Appendix D, because it yields good results and has very low computational requirements) on the output of the multi-microphone speech enhancement algorithm, i.e., the single microphone algorithm was used in a post-processing stage.

The parameter setting of the suggested algorithm (hereby designated TF-GSC) was as follows. The blocking filters $\mathbf{h}_m$ were modeled by non-causal FIRs with 181 coefficients in the interval $[-90, 90]$. The cancelling filters $\mathbf{g}_m$ were modeled by non-causal FIRs with 251 coefficients in the interval $[-125, 125]$. In order to implement the *overlap & save* procedure, segments with 512 samples were used. Overlap length was set to the number of filters' coefficients, to avoid cyclic convolution affects. By this choice, the segments are long enough for the approximation in 2.2 to hold, while still maintaining the filters' time-invariance. The system identification procedure utilized 13 segments. The length of each segment was 1000-samples (sampling rate was 8KHz). We note that system identification was applied only during active speech periods. However an accurate VAD is not necessary for this purpose. In our experiments, the noise canceller (NC) block was always active. As was noted earlier, this is due to the fact that in (3.19) we used the input signals, and not the noise reference signals, in order to calculate $P_{\text{est}}(t, e^{j\omega})$. Hence a voice activity detector (VAD) was not necessary.

We have also implemented the standard (delay only) GSC algorithm (hereby designated D-GSC). This algorithm was implemented in the time domain. In

order to estimate the delays we used a cross correlation criterion that was also applied only during active speech periods. The noise canceller filters were realized using the same length as in our implementation of the TF-GSC algorithm.

8.2 Performance Evaluation

In this section we evaluate the performance of the TF-GSC algorithm (with and without post-processor) in several noise fields. In order to evaluate the performance of the algorithms we used the following objective quality measures, denoted averaged SNR ($\mathrm{SNR}_{\mathrm{avg}}$). Given some signal, $x(t)$, the averaged SNR is define by

$$\mathrm{SNR}_{\mathrm{avg}} \triangleq \frac{\sum_{t\in T_s} x^2(t) - \sum_{t\in T_n} x^2(t)}{\sum_{t\in T_n} x^2(t)} \tag{8.1}$$

where, T_s denotes periods in time where the speech signal is active, and T_n denotes periods in time where the speech signal is inactive. This quality measure compares the signal energy in $x(t)$ to the noise energy. $x(t)$ is replaced by $z_1(t)$ for measuring the quality of the (arbitrarily chosen) input signal and by $y(t)$ for the quality of the outcome signal (both with- and without- the application of the post-processing algorithm).

8.2.1 $\mathrm{SNR}_{\mathrm{avg}}$ for Coherent Noise Field

The results demonstrated in Tables 8.1, 8.2 are the outcome of the frequency domain algorithm, including single microphone post-processing, for the cases of fan noise and white noise. In Table 8.1 results for fan noise are demonstrated. In Table 8.2 results for white noise are demonstrated. Figure 8.1 shows the waveforms of the recorded speech, noisy speech and enhanced speech (with and without post-processing). The noise signal used was fan noise at an averaged SNR level of 0 dB for $M = 5$ microphones. Figure 8.2 shows sonograms for a signal portion with input $\mathrm{SNR}_{\mathrm{avg}}$ of 3 dB.

To further assess the output speech quality, have conducted informal listening tests. All our listeners clearly indicated impressive noise reduction (almost noise

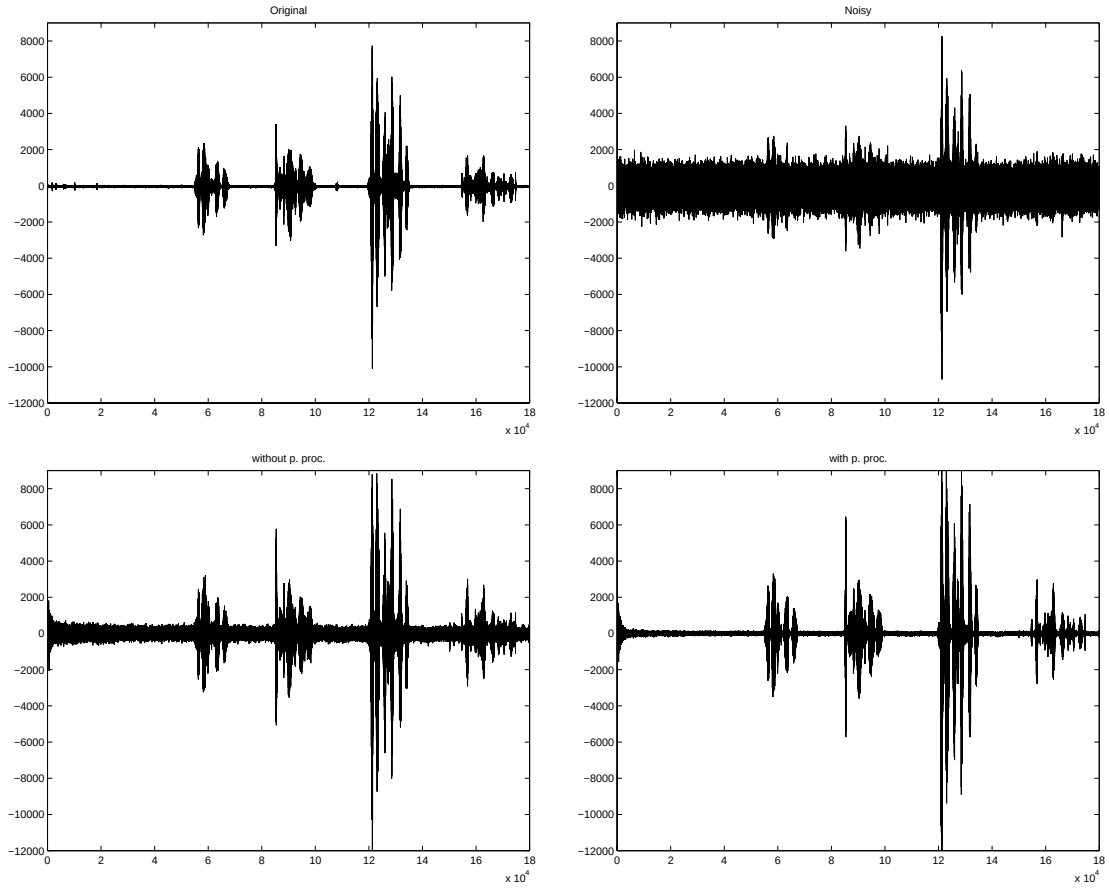

Figure 8.1: Speech waveforms for the TF-GSC algorithm: original and enhanced (with and without post-processing) for point source fan noise. $\mathrm{SNR}_{\mathrm{avg}} \approx 0$ dB. $M = 5$ microphones.

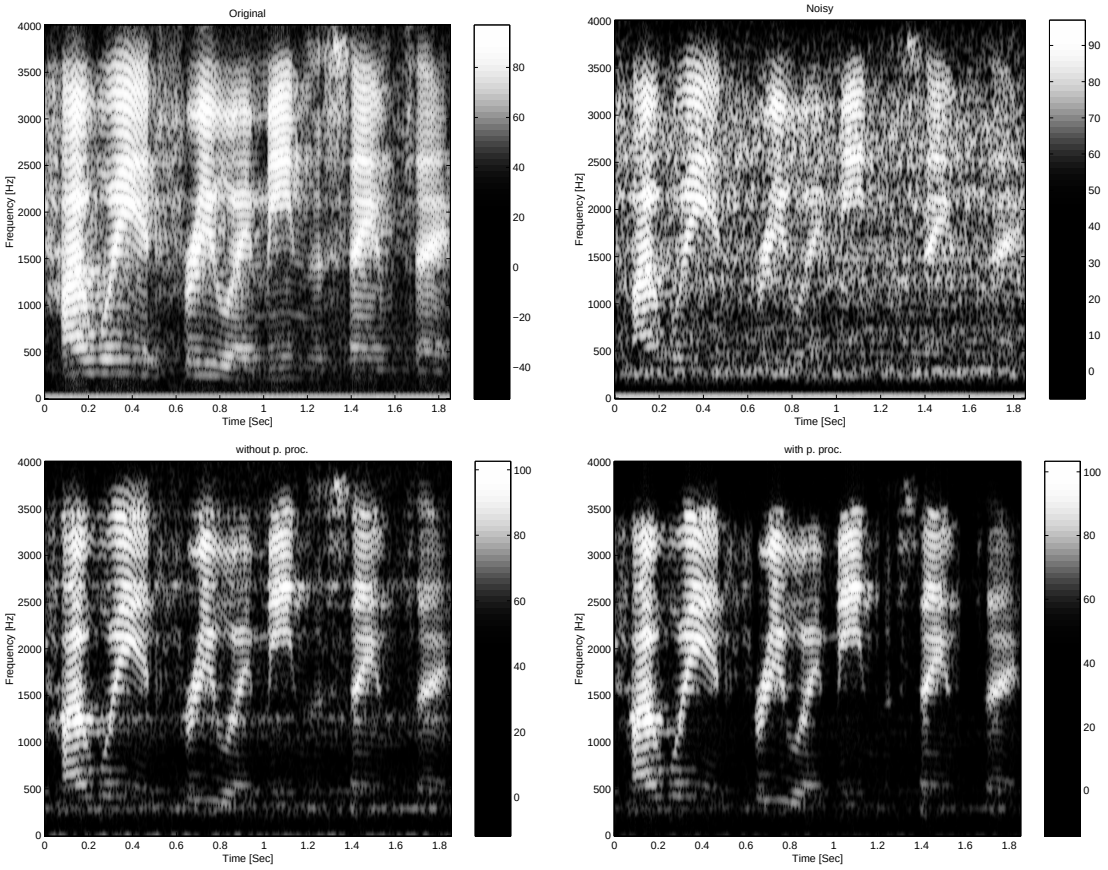

Figure 8.2: Sonograms for the TF-GSC algorithm: original, noisy and enhanced (with and without post-processing) for point source fan noise. $\mathrm{SNR}_{\mathrm{avg}} \approx 0$ dB. $M = 5$ microphones.

Input SNR_{avg}	SNR_{avg} w.o. p. proc.[dB]	SNR_{avg} w. p. proc.[dB]
−5.6	6.1	16.5
−2.7	9.1	20.8
−0.7	11.3	23.1
1.2	13.4	25.6
4.3	18.2	30.1
7.2	21.2	33.0
10.2	23.8	35.6
14.1	26.8	39.0

Table 8.1: SNR_{avg} improvement (TF-GSC algorithm with and without single microphone post-processing) for point source fan noise. $M = 5$ microphones. Frequency domain version.

elimination), without any noticeable distortion at SNR_{avg} levels as low as -5 dB (excluding the experiment with white noise at -5 dB input SNR_{avg}, for the lower gain sentence, which produced a muffled speech signal). This low-level distortion was predicted qualitatively by our analysis in Chapter 6.

We also examined the effect of the number of microphones used on the noise reduction. The results for noisy signal SNR_{avg} of -0.7 dB are given in Table 8.3. Fig 8.3 shows the waveforms of the recorded speech, noisy speech and enhanced speech (with and without post-processing). The noise signal used was fan noise at an averaged SNR level of 0 dB, with $M = 2$ microphones. Although, the performance with $M = 2$ microphones is inferior to the results obtained for $M = 5$ microphones (presented in Figure 8.1), the achievable noise reduction is still significant.

8.2.2 SNR_{avg} for Diffused Noise Field

Now we will evaluate the performance of the algorithm with a diffused noise field. In this experiment the array was determined to look towards the left source. Results for various SNR_{avg} levels are shown in Table 8.4. Figure 8.4 shows the waveforms of the recorded speech, noisy speech and enhanced speech (with and without post-processing). The noise field used is diffused at an averaged

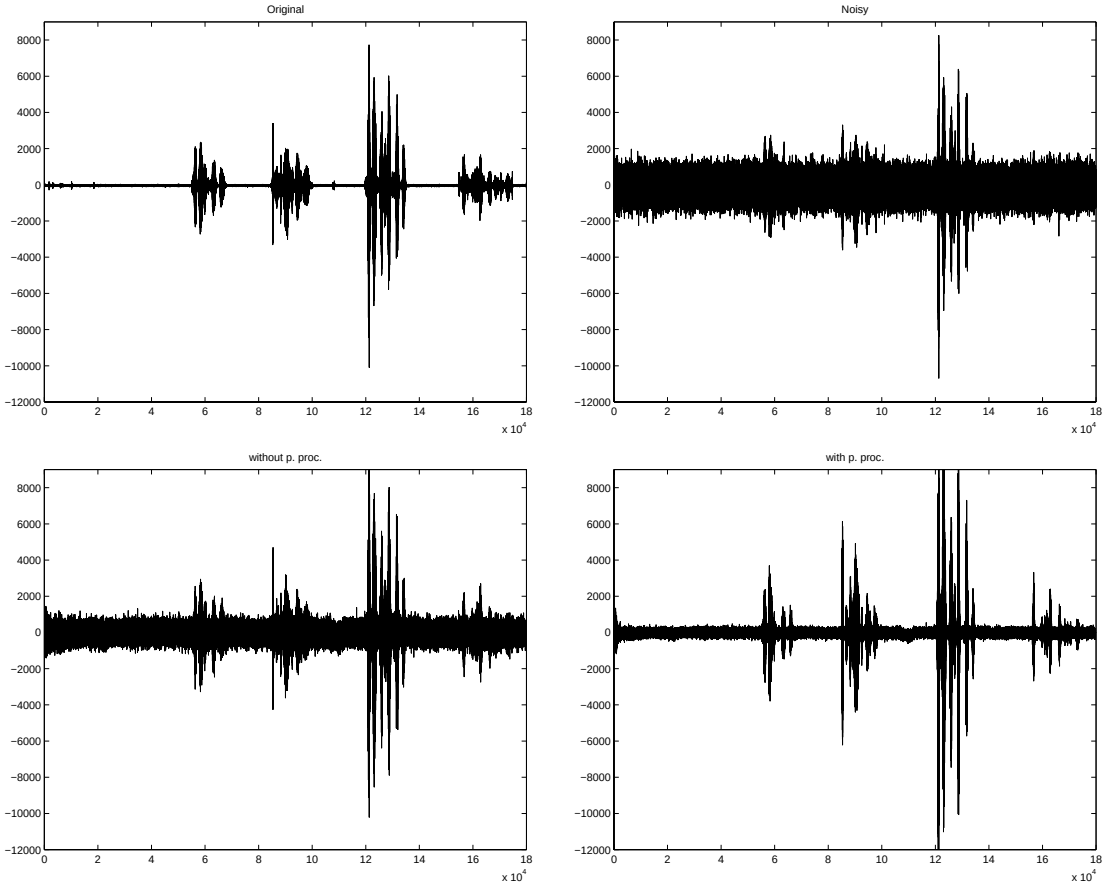

Figure 8.3: Speech waveforms: original and enhanced (TF-GSC algorithm with and without post-processing). $\mathrm{SNR}_{\mathrm{avg}}$ level of 0 dB. $M = 2$ microphones.

Input SNR_{avg}	SNR_{avg} w.o. p. proc.[dB]	SNR_{avg} w. p. proc.[dB]
−5.9	7.3	16.2
−3.0	10.3	23.8
−1.0	12.3	26.0
3.9	17.4	32.0
7.0	19.1	33.6
9.0	16.1	30.6
11.0	17.8	32.3
13.9	26.7	40.4

Table 8.2: SNR_{avg} improvement (TF-GSC algorithm with and without single microphone post-processing) for point source white noise. $M = 5$ microphones. Frequency domain version.

No. of Mic.	SNR_{avg} w.o. p. proc.[dB]	SNR_{avg} w. p. proc.[dB]
2	4.8	15.1
3	8.6	20.5
4	9.6	21.3
5	11.3	23.1

Table 8.3: The effect of the number of microphones on noise reduction. Noisy signal $SNR_{avg} - 0.7$ dB. TF-GSC algorithm. Frequency domain version.

SNR level of 0 dB for $M = 5$ microphones. It is noted that the results are far inferior when compared with the results for directional noise field. The achievable noise reduction of the array is non-significant, and the resulting signal after post-processing seems distorted. This observation is verified by unofficial listening tests. The reason for the poor performance is the coherence structure of the diffused noise, as shown in Appendix A. There is virtually no correlation between sensors in high frequencies, and in low frequencies there is almost no signal, due to the microphone characteristics. Thus, the noise canceller branch is almost useless. Most of the resulting noise reduction is due to the post-processor. Thus, in the low-SNR range, when the multi-channel algorithm does not yield significant enhancement, the algorithm collapses.

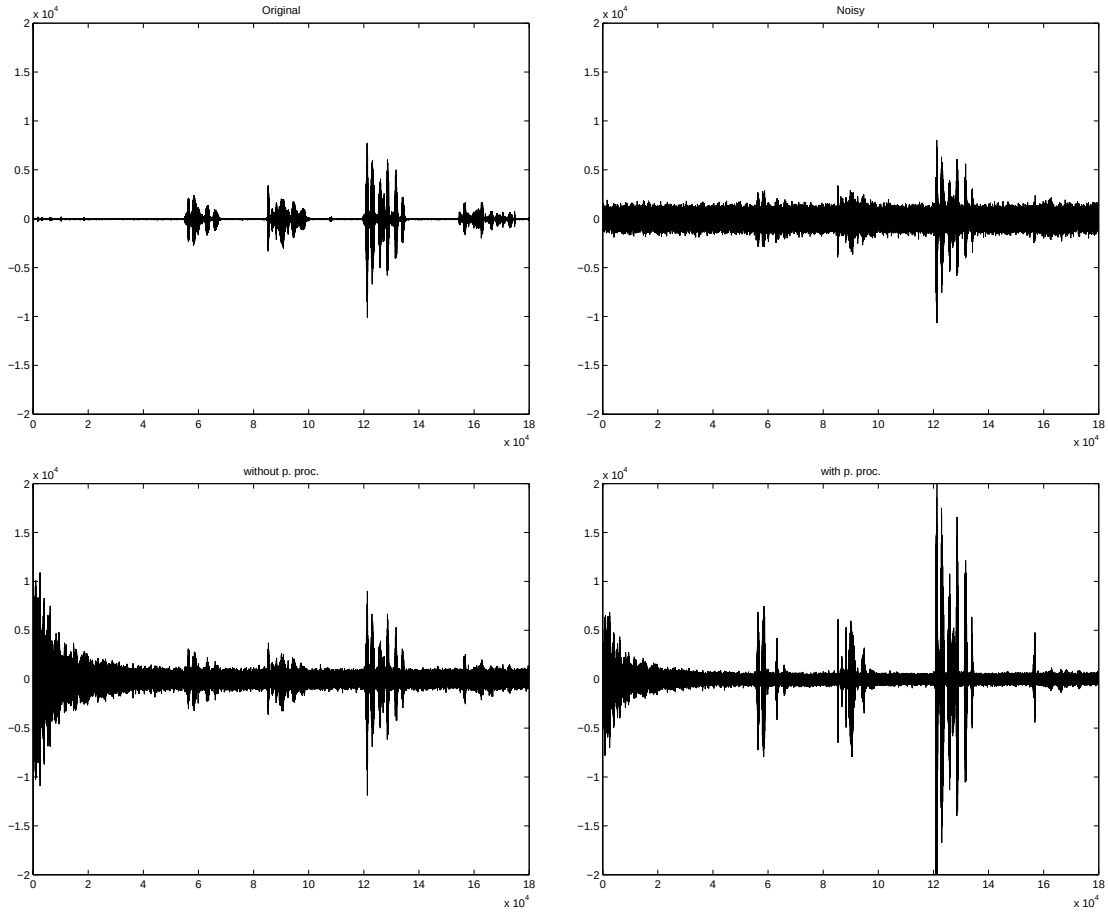

Figure 8.4: Speech waveforms for TF-GSC algorithm: original and enhanced (with and without post-processing) in diffused noise field. $M = 5$ microphones.

Input SNR_{avg}	SNR_{avg} w.o. p. proc.[dB]	SNR_{avg} w. p. proc.[dB]
−5.0	−2.0	−1.2
−2.3	0.8	6.3
−0.5	2.5	12.6
1.4	2.4	14.2
4.3	6.9	18.5
7.3	9.9	22.4
10.2	13.2	26.4
14.1	17.5	31.6

Table 8.4: SNR_{avg} improvement with and without single microphone post-processing for diffused noise field. $M = 5$ microphones. TF-GSC algorithm. Frequency domain version.

8.2.3 SNR_{avg} for Incoherent Noise Field

The same trend as in a diffused noise field can be shown in Table 8.5 for incoherent noise source. No extra noise reduction is achieved by the noise cancelling branch, thus making the array processor almost useless.

Input SNR_{avg}	SNR_{avg} w.o. p. proc.[dB]	SNR_{avg} w. p. proc.[dB]
−5.7	−2.3	−1.6
−2.7	0.6	4.7
−0.8	2.7	11.8
1.2	4.8	15.8
4.2	7.9	21.0
7.7	10.9	24.8
10.2	13.8	28.1
14.1	17.8	32.2

Table 8.5: SNR_{avg} improvement (with and without single microphone post-processing) for incoherent noise field. $M = 5$ microphones. TF-GSC algorithm. Frequency domain version.

8.3 Time Domain Algorithm

Results for the time domain version of the algorithm are shown in Table 8.6. It is obviously seen, that the performance of the time domain version is inferior to that of the frequency domain version. The problem is emphasized in the higher $\mathrm{SNR}_{\mathrm{avg}}$ levels, where very low noise levels are encountered causing nonstable behavior. This is most likely due to the more stable nature of the frequency domain algorithm. Time domain adaptation is performed at each sample, but frequency domain adaptation is performed frame by frame. Furthermore, the time domain version is more time consuming, especially the system identification task. See also the discussion in Chapter 4. The decorrelation approach for the TFs

Input $\mathrm{SNR}_{\mathrm{avg}}$	$\mathrm{SNR}_{\mathrm{avg}}$ w.o. p. proc.[dB]	$\mathrm{SNR}_{\mathrm{avg}}$ w. p. proc.[dB]
−5.6	3.8	14.5
−2.7	7.6	18.3
−0.7	10.0	20.5
1.2	12.0	22.2
4.3	14.1	23.8
7.2	13.6	23.1
9.2	15.2	24.7
14.1	14.1	23.6

Table 8.6: $\mathrm{SNR}_{\mathrm{avg}}$ improvement (with and without single microphone post-processing) for point source fan noise. $M = 5$ microphones. TF-GSC algorithm. Time domain version.

ratio identification can be applied only in the time domain (a frequency domain counterpart is a topic for further research). The decorrelation approach was used only for tracking slight changes in the TFs' ratio, as the recordings were with fixed loudspeakers. The TFs' ratio identification was initialized by the regular nonstationarity approach. Adaptation of the TFs' ratio, $h_m(t);\ m = 1, \ldots, M$, is conducted only during active speech periods, while the adaptation of the noise canceler filters, $h_m(t);\ m = 1, \ldots, M$, is conducted during silence periods. When applied to the case of time invariant TFs, only a slight improvement over the

regular time domain version is encountered. An average improvement of 1 dB is encountered. In the low average SNR level (input SNR_{avg} of -5.6 dB), the algorithm demonstrated unstable behavior. This is probably due to difficulties in tracking highly disturbed speech portions.

8.4 Comparison between TF-GSC and D-GSC

In this section we focus on comparison between the TF-GSC and the D-GSC algorithms (for this purpose we omit the post-processing algorithm, which can be applied to both algorithms).

In Table 8.7 we assess the ability of the blocking matrix (BM) to generate noise-only reference signals. For each input SNR value we evaluated the SNR of the reference signals both for the D-GSC algorithm and for the TF-GSC algorithm. A high SNR value indicates that there is a high leakage of speech to the noise reference, and hence the resulting output is expected to be reverberated due to self cancellation. As can be seen, the quality of the noise reference produced by the TF-GSC algorithm is better than that produced by the D-GSC algorithm. This holds both for the point noise source and for the diffused noise source.

In order to evaluate and compare the performance of the algorithms we used three objective quality measures. The first is signal to noise ratio (SNR) defined by,

$$\text{SNR} \triangleq \frac{\sum_{t\in T_s} z_{1,s}^2(t)}{\sum_{t\in T_s} \left(z_{1,s}(t) - Ky(t)\right)^2}$$

where $z_{1,s}(t)$ is the signal component recorded by the first microphone. $y(t)$ is the algorithm output (reconstructed speech signal). T_s denotes periods in time where the speech signal is active. K is a gain factor that compensates for possible gain level variations of the signals. In addition to that $z_{1,s}(t)$ and $y(t)$ are time-aligned.

The second quality measure is noise reduction (NR), defined by

$$\text{NR} \triangleq \frac{\sum_{t\in T_n} (Ky(t))^2}{\sum_{t\in T_n} z_1^2(t)}$$

where T_n denotes periods in time where the speech signal is inactive. The quality

Input SNR	TF-GSC SNR	D-GSC SNR
-4.5	-9.1	0.5
-1.5	-8.9	3.5
1.5	-7.8	5.8
4.5	-5.8	-2
7.5	-3.4	2.5
10.5	-0.6	5.0
13.5	2.3	8.1
16.5	5.3	10.8
20.4	8.2	14.0

Input SNR	TF-GSC SNR	D-GSC SNR
-4.6	-12.0	-9.2
-1.9	-10.7	-6.2
1.8	-8.7	-3.2
4.4	-6.2	0.1
7.4	-3.4	2.7
10.4	-0.5	5.8
13.4	2.3	8.2
16.5	5.3	11.2
19.4	8.2	14.2

Table 8.7: Blocking ability for point source (top) and diffused noise (bottom) in decibels. Five microphones.

measure, NR, compares the noise level in the reconstructed speech to the noise level recorded by the first microphone. Table 8.8 summarizes the SNR and NR values in dB when using the D-GSC and TF-GSC algorithms. While the noise reduction ability of both algorithms is comparable, the SNR level achieved by the TF-GSC is much higher. These observations indicate that TF-GSC is characterized by a significantly lower speech distortion compared to D-GSC, while keeping the same level of noise reduction. In the high input SNR region, although the algorithm degrades the SNR measure, it results in an overall enhanced output. This is due to the fact that it reduces the noise level. Finally comparing our results for the two noise sources, it can be seen that the SNR and NR values of both algorithms are higher for the point noise source case (except for the SNR

measure of D-GSC). This is due to the low coherence function in the diffused noise case, which degrades the performance of the noise cancelling block of the algorithm [18].

In SNR	D-GSC SNR	TF-GSC SNR	D-GSC NR	TF-GSC NR
-6.4	-3.7	-0.8	6.5	6.8
-3.4	-2.4	3.3	6.3	8.4
-0.4	0.6	6.8	11.8	10.0
2.6	2.4	9.3	10.0	11.4
5.6	2.6	11.1	9.5	10.6
8.6	3.2	11.7	9.0	9.1
11.6	3.2	12.0	8.2	7.8
14.6	3.3	12.3	7.2	6.7
17.6	3.3	12.5	6.0	5.7

In SNR	D-GSC SNR	TF-GSC SNR	D-GSC NR	TF-GSC NR
-6.5	-3.6	-3.7	3.7	3.0
-3.5	-1.4	-0.6	3.7	3.2
-0.5	0.3	2.2	3.6	3.2
2.5	1.6	4.9	3.5	3.2
5.5	2.4	7.3	3.3	3.2
8.5	2.8	9.5	3.2	3.2
11.5	3.1	11.1	2.3	3.2
14.5	3.3	12.1	2.1	3.0
17.5	3.3	12.7	1.9	2.8

Table 8.8: SNR improvement and noise reduction (NR) for point source (top) and for diffused source (bottom) in decibels. Five microphones.

The third quality measure is the averaged SNR, $\mathrm{SNR}_{\mathrm{avg}}$ introduced before in Eq. 8.1. This quality measure compares the signal energy in $x(t)$ to the noise energy. Figure 8.5 shows $\mathrm{SNR}_{\mathrm{avg}}$ in dB both for D-GSC and TF-GSC for both noise types (point and diffused). As can be seen, TF-GSC yields higher values of $\mathrm{SNR}_{\mathrm{avg}}$. The $\mathrm{SNR}_{\mathrm{avg}}$ values of both algorithms are higher for the point noise source case.

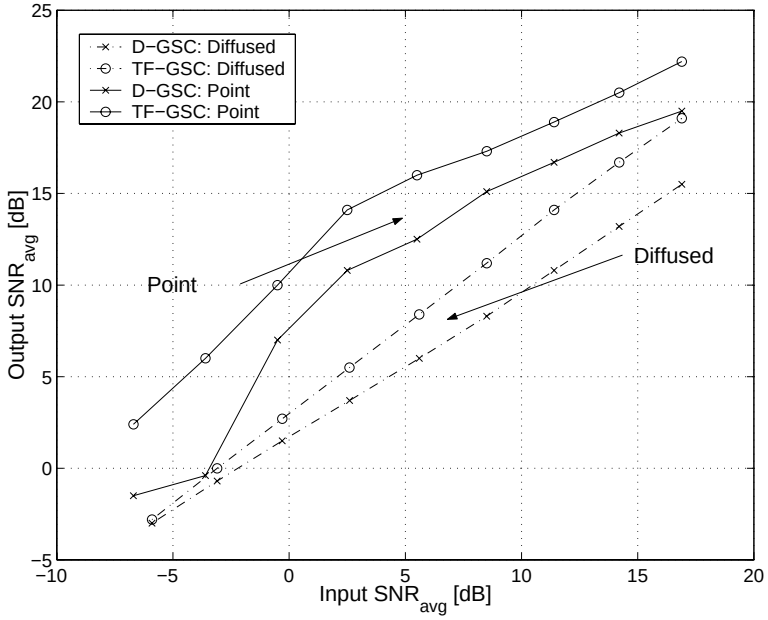

Figure 8.5: Averaged SNR improvement for point noise source and diffused noise source.

Figure 8.6 shows the waveforms of the speech component recorded by the first microphone, the noisy speech at the first microphone and the enhanced speech for both D-GSC and TF-GSC algorithms. The noise signal used was a point source at an SNR level of 0dB. Figure 8.7 shows sonograms of the same data. It can be seen that the TF-GSC algorithm produces an enhanced speech signal, with higher noise reduction and lower distortion.

To further assess the output speech quality we have conducted informal listening evaluations. All our listeners clearly indicated impressive noise reduction without any noticeable distortion for the TF-GSC algorithm. On the other hand, the D-GSC algorithm was classified as reverberated. This is due to self-cancellation which is caused by leakage of the desired signal into the noise reference.

All our algorithms were implemented without a VAD in the noise cancelling block. When a VAD is incorporated, there is no significant change in the performance of the TF-GSC algorithm. However, there is an improvement in the

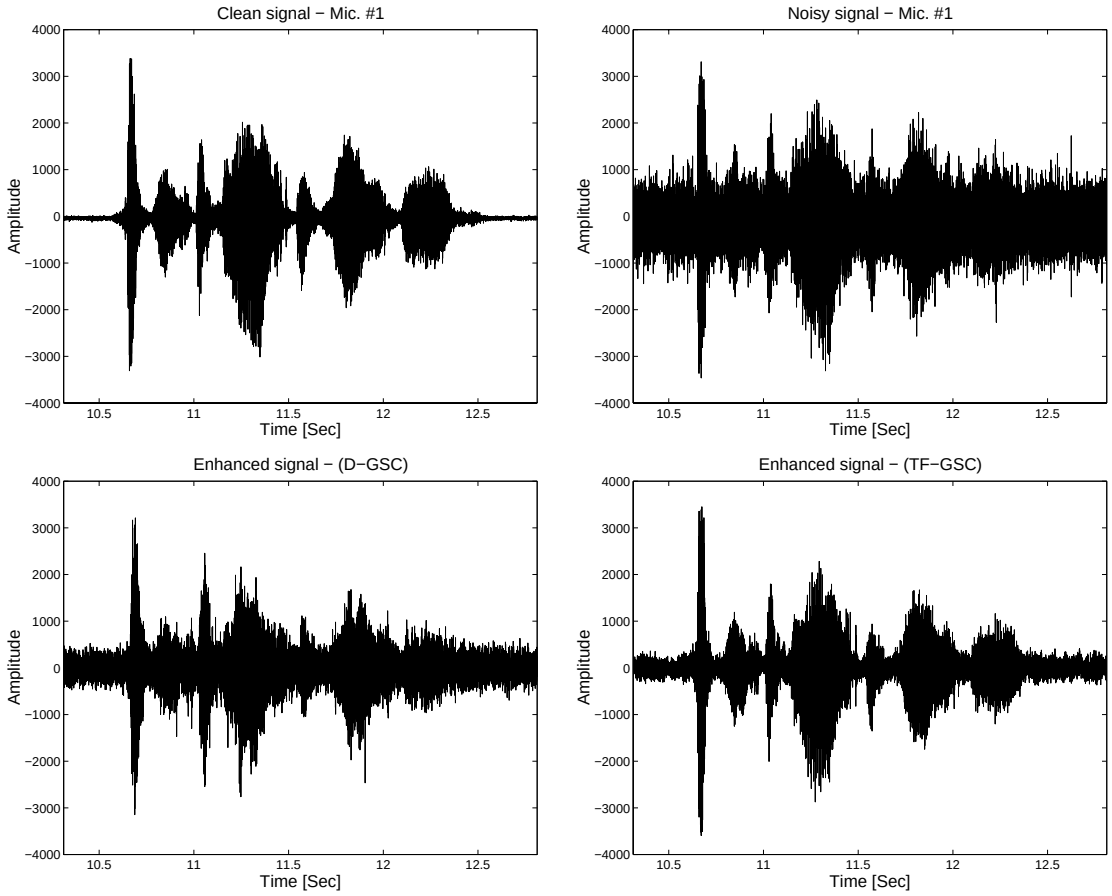

Figure 8.6: Speech waveforms: Clean Microphone #1, Noisy and enhanced (D-GSC, TF-GSC)

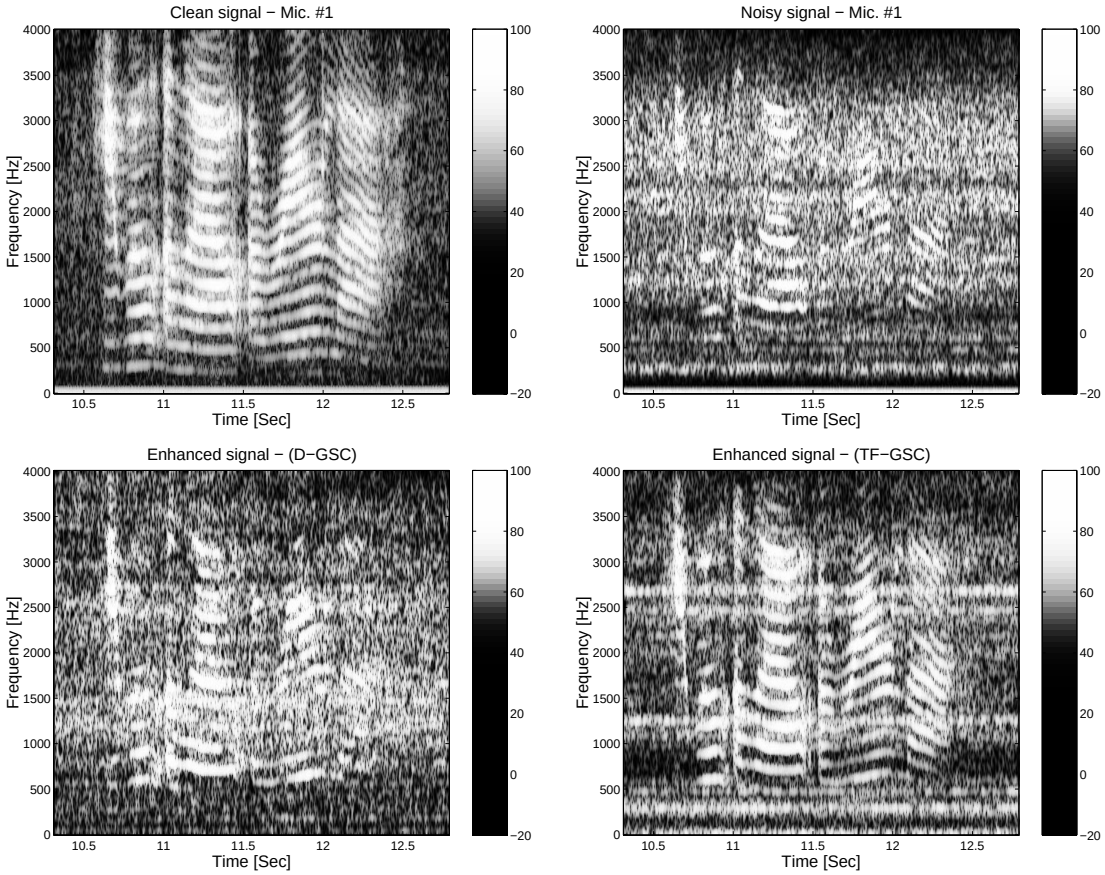

Figure 8.7: Sonograms : Clean Microphone #1, Noisy and enhanced (D-GSC, TF-GSC)

performance of the D-GSC as noted in [71]. Even so, the quality of the enhanced speech produced by the TF-GSC algorithm is significantly higher than that produced by D-GSC.

8.5 Computational Complexity and Memory Requirements

In this section we discuss the computational complexity of the TF-GSC algorithm (implemented in Frequency domain) and its memory requirements. We then compare them to the D-GSC algorithm (implemented in time domain) figures . We will examine the algorithms with the following parameter settings. Number of microphones is M. The blocking filters' length is $2L_h + 1$ and the cancelling filters' length is $2L_g + 1$. Segments' length (for frequency analysis) is K ($K > \max(2L_h + 1, 2L_g + 1)$) and overlap length is set to $\max(2L_h + 1, 2L_g + 1)$.

The TF-GSC algorithm process the data frame-wise, $K - \max(2L_h+1, 2L_g+1)$ new samples are created in each K length frame. We state now the number of operations required for implementing each stage of the algorithm introduced in Figure 3.5.

Spectral analysis $M \cdot K \log_2 K$ multiplications and additions.

Fixed beamformer $M \cdot \frac{K}{2}$ multiplications and additions.

Noise reference signals construction $(M-1) \cdot \frac{K}{2}$ multiplications and additions.

Output signal construction $(M-1) \cdot \frac{K}{2}$ multiplications and $(M-1) \cdot \frac{K}{2} + \frac{K}{2}$ additions.

Filters update $M \cdot K$ multiplications and $M \cdot \frac{K}{2}$ additions for constructing the normalization factor $P_{\text{est}}(t, e^{j\omega})$. $(M-1) \cdot K$ multiplications and $(M-1) \cdot \frac{K}{2}$ additions and $(M-1) \cdot \frac{K}{2}$ divisions for constructing the intermediate filters

and $2(M-1)\cdot K\log_2 K$ multiplications and additions for imposing the FIR structure.

Spectral synthesis $M\cdot K\log_2 K$ multiplications and additions.

Summary of the operations can be found in Table 8.9.

Multiplications	$K\log_2 K\cdot(4M-2)+K\cdot(4.5M-2)$
Additions	$K\log_2 K\cdot(4M-2)+\frac{K}{2}\cdot(5M-2)$
Divisions	$(M-1)\cdot\frac{K}{2}$

Table 8.9: Total number of operations required for the TF-GSC algorithm (for $K-\max(2L_h+1,2L_g+1)$ samples).

The D-GSC algorithm process the data sample by sample. We state now the number of operations required for implementing each stage of the algorithm introduced in [5]

Fixed beamformer M additions and multiplications (multiplying by the corresponding gain).

Noise reference signals construction $M-1$ multiplications and additions.

Output signal construction $(M-1)\cdot(2L_g+1)$ multiplications and $(M-1)\cdot(2L_g+1)+1$ additions.

Filters update $M\cdot(2L_g+1)$ multiplications and $M\cdot(2L_g+1)+M$ additions for constructing the normalization factor $p_{\text{est}}(t)$. $2(M-1)\cdot(2L_g+1)$ multiplications, $(M-1)$ additions and $(M-1)\cdot(2L_g+1)$ divisions for constructing the updated filters.

Summary of the operations can be found in Table 8.10. As $K\approx 2\max(2L_h+1,2L_g+1)$, it is clear from Tables 8.9,8.10 that the computational burden is approximately equivalent (recall that the number of operations depicted in Table 8.9 refers to calculating approximately $\frac{K}{2}$ samples).

Multiplications	$(2L_g+1)\cdot(4M-3)+2M-1$
Additions	$(2L_g+1)\cdot(2M-1)+4M-2$
Divisions	$(2L_g+1)\cdot(M-1)$

Table 8.10: Total number of operations required for the D-GSC algorithm (for one sample).

Parameter estimation ($\mathbf{H}(e^{j\omega})$ for TF-GSC and delay for D-GSC) is conducted only once in a while. TFs ratio estimation involves 2×2 matrix inversion and multiplication for each frequency bin (for the LS estimator), as well estimating the involved spectra. Time delay estimation is conducted using cross-correlation techniques, which involves multiplications and maximum finding. Although, the TF-GSC algorithm imposes heavier computational burden, the overall penalty is not severe, since the estimation is not conducted frequently.

The TF-GSC algorithm requires dynamic memory of about $2K\cdot M$ for saving the current data frame and updated filters. The D-GSC algorithm requires about $2K\cdot(2*L_g+1)$ for the same purpose. Both algorithms do not need fixed memory for saving parameters. Again, both algorithm are almost equivalent.

8.6 Chapter Summary

We evaluated the performance of the algorithm for various noise fields. For an almost real-life situation, when using speech and noise signals recorded in a reverberating room, an impressive noise reduction is encountered almost without any noticeable distortion of the desired signal. In diffused and incoherent noise fields (artificially generated by the computer) the array processor is almost useless, and the significant noise reduction is almost entirely due to the single microphone speech enhancer.

Using single microphone algorithm as a post-processor yields even more impressive results. Up to 25dB increase in SNR value is encountered for the point source noise. Although significant portion of the increase in the SNR is due to the post-processor, activating both multi-microphone and single microphone al-

gorithms is necessary for the enhancement task, since the MixMax algorithm used needs the SNR to be above a threshold level to perform adequately.

The time domain version of the TF-GSC algorithm is inferior to the frequency domain version regarding performance and computational burden, although it exhibits significant noise reduction in mid-level SNR values. The decorrelation criterion seems to improve the tracking ability, but further assessment should be conducted to prove this point.

Comparison between the suggested algorithm and the conventional Griffiths & Jim algorithm demonstrates impressive advantage of the new algorithm in respect to both objective and subjective quality measures, almost without increasing the computational burden or memory requirements.

Chapter 9

Summary

9.1 Discussion

In this work we considered a sensor array located in a an enclosure. The array is used to enhance a desired signal contaminated by interference.

We have began by redeveloping Frost's algorithm (namely, a constrained minimum power adaptive beamformer) in the frequency domain. This formulation enabled us to deal with a complicated TF in the same simple manner as Frost dealt with delay-only arrays and to prove the optimality of the algorithm. A GSC version of the suggested algorithm, which requires knowledge of the TFs, was then developed. A practical algorithm was implemented by substituting estimation of the entire TFs with estimates of the ratios between the different TFs. Three blocks constitute the suggested algorithm. A fixed match filter beamformer (MFBF), reference noise constructor and multi-channel noise canceller. The first two depend on the TFs' ratio.

An unbiased estimate of the TFs' ratio was achieved by exploiting the desired signal nonstationarity.

An alternative approach for estimating the TFs' ratio was achieved by using the fact that the desired signal and the reference noise signals are uncorrelated. As a byproduct of this approach, the desired signal leakage to the reference noise signals might be eliminated, thus reducing the self-cancellation phenomena.

The suggested algorithm can be applied for enhancing an arbitrary nonsta-

tionary signal corrupted by stationary noise. An arbitrary transfer function and array geometry can be used. The use of TFs' ratios rather than the TFs themselves (which is the counterpart of relative delay in delay-only arrays) improves the efficiency and robustness of the algorithm, since shorter filters can be used. This observation, which is justified empirically, might be due to pole-zero cancellation in the TFs' ratios.

The computational burden (and memory requirements) of the suggested TF-GSC frequency domain algorithm is significantly smaller than that of its time domain counterpart and comparable to the conventional Griffiths and Jim algorithm (D-GSC). In our (probably inefficient) MATLAB© implementation only three times real time was required (on our ultra 5 SUN workstation).

This is due to two reasons. First, the system identification in the frequency domain involves only a 2×2 matrix inversion for each frequency bin examined (in the time domain it is required to invert a matrix whose order is the dimension of the desired filter). In addition, the frequency domain system identification need not be implemented for frequency bands with lower-level speech signal components. Second, the LMS algorithm in the frequency domain is more efficient (and stable) since it utilizes segmentation.

A natural application of the suggested algorithm is speech signal received by several microphones in a reverberating environment. In this application it proved beneficial to use single channel speech enhancement post-processing stage. We suggested the use of a frequency domain algorithm, nicknamed "MixMax", which uses a Gaussian mixture model for the speech log-spectrum.

The performance of the multi-channel algorithm depends strongly on the noise field. In point noise field the noise reduction is significant, and the resulting signal can be further processed by the single channel algorithm (Up to 25dB increase in SNR value is encountered for the point source noise). On the other hand, if the noise field is diffused or incoherent, the noise reduction of the array is poor, thus almost no advantage is gained by its use. The single microphone enhancer performance depends on the input SNR, and is not very good at the low SNR range.

The poor performance of the array is due to the lack of correlation between noise signals at the microphones. We will note that in actual environments the noise field is a combination of point sources and diffused signals, thus, the expected performance should be better than predicted for a completely diffused noise, but worse than the performance for a completely point source.

Comparison between the suggested algorithm and the conventional Griffiths & Jim algorithm demonstrates impressive advantage of the new algorithm in respect to both objective and subjective quality measures.

9.2 Topics for Further Research

- In Chapter 3 we suggest the use of an $(M-1)\times M$ blocking matrix, $\mathcal{H}(e^{j\omega})$. As suggested by Nordholm [11] for the delay-only case, we can search for a more stable algorithm, which constructs fewer than $M-1$ noise reference signals.

- Note that the suggested algorithm is completely independent of the TFs ratio estimation procedure. In this study, we suggest two estimation procedures: exploiting signal nonstationarity and imposing the decorrelation criterion. We can use other unbiased system identification methods, which exploit other characteristics of the signals involved. One such method might be the use of *higher order statistics* (HOS), by exploiting the different divergence from Gaussianity of the desired signal and the noise signal.

- In Chapter 5 we suggest a time domain estimation procedure for the TFs' ratio, based on the decorrelation criterion. A frequency domain method should be also derived, as the frequency domain version has better performance. The frequency domain tracking procedure might be also more stable.

- Note that the general problem introduced in Chapter 1 deals with more than one desired signal. Throughout this study we have dealt with only

one desired signal. An extension to the suggested algorithm is thus required for dealing, for instance, with the problem of echo signals, which are nonstationary as the desired signal.

Appendix A

Diffused Noise Field

When a sound is created in the interior of a closed enclosure (e.g., automobile), the hard surfaces (windows, walls, etc.) reflect the sound back into the interior, resulting in reverberation. When a noise signal is reflected from such a hard surface, the reflected signal may be considered as an another noise source on the opposite side of the reflecting boundary. Allen and Berkley [56] and Peterson [57] calculated the reverberant response of a cube-shaped room, by applying the image model. By this model an infinite number of images of each noise source are summed to create the noise field. When a sufficient number of noise sources and images occur, the noise field approaches that of a spatially uniform or *diffuse* noise field.

Dal-Degan and Prati [18] and Goulding and Bird [58] considered the problem of calculating the noise power spectral density and the coherence between two separated microphones.

The noise spectrum received in each microphone should be measured empirically and depends on the actual noise source (e.g., wind, car engine, fan, etc.). We will concentrate on calculating the *coherence* between noise signals received in two microphones. Assume the structure shown in Figure A.1. Assume that the distance between microphones A and B is d, c is the speed of sound, and ω is the frequency in rad/sec. The diffuse noise model asserts that uncorrelated noise sources radiate from a spherical shell around the microphones and that a far field

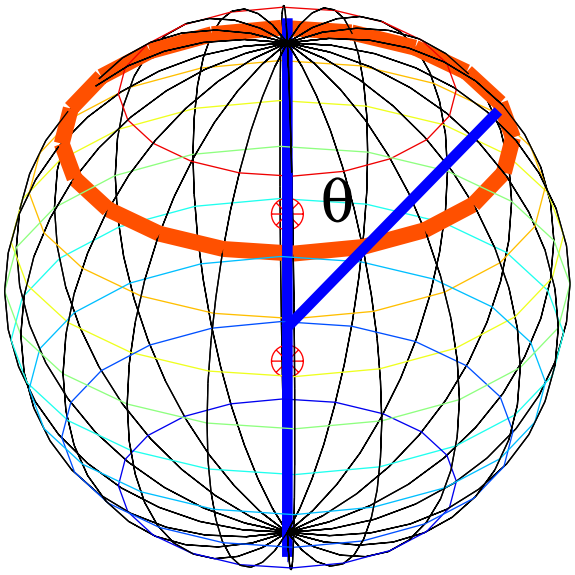

Figure A.1: Two microphones in diffused noise field.

assumption holds, i.e., the sphere radius fulfills $R \gg d$.

We will begin our derivation by assuming that two noise sources impinge the array from two different angles, θ_1, θ_2:

$$\begin{aligned} s_A(t) &= s_1(t) + s_2(t) \\ s_B(t) &= s_1(t - \frac{d\cos\theta_1}{c}) + s_2(t - \frac{d\cos\theta_2}{c}). \end{aligned} \tag{A.1}$$

Calculate the auto- and cross-correlation between signals:

$$\begin{aligned} R_A(\tau) &= E\{s_A(t)s_A(t+\tau)\} = R_1(\tau) + R_2(\tau) \\ R_B(\tau) &= E\{s_B(t)s_B(t+\tau)\} = R_1(\tau) + R_2(\tau) \\ R_{AB}(\tau) &= E\{s_A(t)s_B(t+\tau)\} = R_1(\tau - \frac{d\cos\theta_1}{c}) + R_2(\tau - \frac{d\cos\theta_2}{c}). \end{aligned} \tag{A.2}$$

Calculate the auto- and cross-spectra of the signals by applying the Fourier trans-

form

$$\begin{aligned} S_{AA}(e^{j\omega}) &= S_1(e^{j\omega}) + S_2(e^{j\omega}) \\ S_{BB}(e^{j\omega}) &= S_1(e^{j\omega}) + S_2(e^{j\omega}) \\ S_{AB}(e^{j\omega}) &= S_1(e^{j\omega})e^{-j\omega\frac{d\cos\theta_1}{c}} + S_2(e^{j\omega})e^{-j\omega\frac{d\cos\theta_2}{c}}. \end{aligned} \tag{A.3}$$

The coherence function is defined:

$$\gamma(e^{j\omega}) \triangleq \frac{S_{AB}(e^{j\omega})}{\sqrt{S_{AA}(e^{j\omega})S_{BB}(e^{j\omega})}} \tag{A.4}$$

Assuming equal spectrum for both sources, $S_1(e^{j\omega}) = S_2(e^{j\omega})$, we obtain

$$\gamma(e^{j\omega}) = \frac{1}{2}\left(e^{-j\omega\frac{d\cos\theta_1}{c}} + e^{-j\omega\frac{d\cos\theta_2}{c}}\right). \tag{A.5}$$

In the same manner, if N uncorrelated sources were transmitting noise, the coherence function would be:

$$\gamma(e^{j\omega}) = \frac{1}{N}\sum_{j=1}^{N} e^{-j\omega\frac{d\cos\theta_j}{c}}. \tag{A.6}$$

Now, assume continuous and uniform distribution of noise sources on the sphere, which is the case in diffuse noise. Note, that the problem is symmetric in the azimuth angle. Thus, we can calculate the contribution of the sphere by integrating over the rings at different elevation angles, θ, as depicted in Figure A.1. The area of the ring is $2\pi R\sin\theta R d\theta$. The total area of the sphere is $4\pi R^2$.

$$\begin{aligned} \gamma(e^{j\omega}) = \frac{1}{S}\oint_S e^{j\omega\frac{d\cos\theta}{c}}\mathrm{d}s = \\ \frac{1}{4\pi R^2}\int_0^{\pi} e^{j\omega\frac{d\cos\theta}{c}} 2\pi R^2\sin\theta\mathrm{d}\theta = \frac{\sin(\omega d/c)}{\omega d/c}. \end{aligned} \tag{A.7}$$

We can see that at a fixed distance between microphones, the coherence drops as the frequency increases, or that reduced distance between microphones is needed for higher frequencies to maintain the same coherence. Thus, the use of a noise cancelling branch in microphone arrays is applicable only in the lower frequency band, where high coherence is demonstrated.

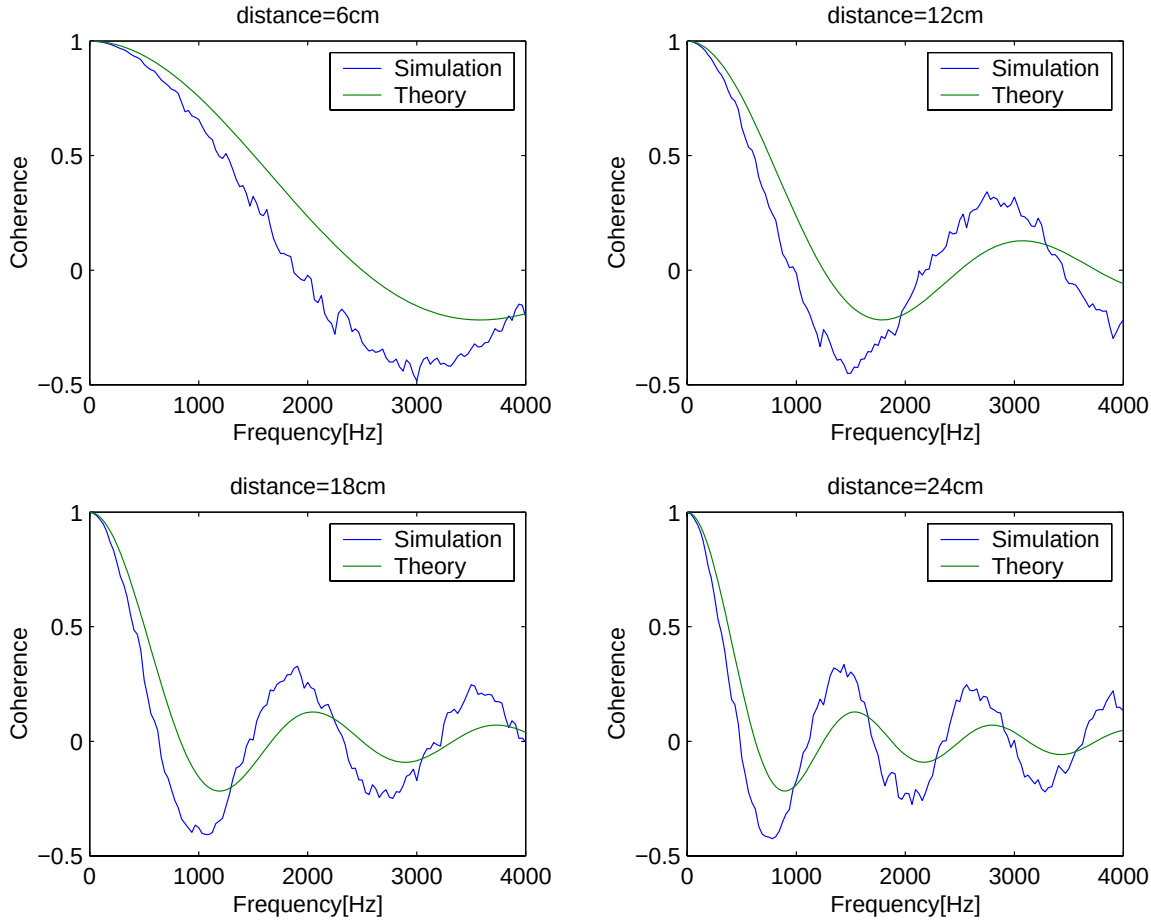

Figure A.2: Cross-coherence between two microphones with various distances in diffused noise field.

To simulate a diffused noise we summed uncorrelated white Gaussian noise sources from 1000 elevation angles equally spaced in the range $\theta \in [0, \pi]$. The microphones were placed in a linear equi-spaced array with inter-element distance of 6 cm. It can be shown from Figure A.2 , that there is quite a good agreement between the simulation and the theory for each microphone pair.

Appendix B

Minimizing Real Function with Respect to a Complex Parameter

Let $z = x + jy$ be a complex variable, where x and y are real variables. We want to minimize the real function $f(x, y)$ with respect to its real parameters. This may be achieved, of course, by the following equations:

$$\frac{\partial f(x,y)}{\partial x} = 0 \quad \frac{\partial f(x,y)}{\partial y} = 0. \tag{B.1}$$

Define, a variable transformation:

$$g(u, v) = g(u(x, y), v(x, y)) = f(x, y). \tag{B.2}$$

Using the chain rule for differentiation gives:

$$\frac{\partial f}{\partial x} = \frac{\partial g}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial x} \tag{B.3}$$

$$\frac{\partial f}{\partial y} = \frac{\partial g}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial g}{\partial v}\frac{\partial v}{\partial y}. \tag{B.4}$$

Now assume $u = z = x + jy$ and $v = z^* = x - jy$ and that they can be treated as independent variables, yielding:

$$\frac{\partial f}{\partial x} = \frac{\partial g}{\partial z} - j\frac{\partial g}{\partial z^*} \tag{B.5}$$

$$\frac{\partial f}{\partial y} = j\frac{\partial g}{\partial z} + \frac{\partial g}{\partial z^*}. \tag{B.6}$$

Thus,

$$\frac{\partial g}{\partial z} = \frac{1}{2}\left[\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right] \tag{B.7}$$

$$\frac{\partial g}{\partial z^*} = \frac{1}{2}\left[\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right]. \tag{B.8}$$

Now, a necessary condition for obtaining the minimum of $f(x, y)$ is either $\frac{\partial g}{\partial z} = 0$ or $\frac{\partial g}{\partial z^*} = 0$. These results can be generalized to the case of a scalar quantity that depends on a vector parameter. Define,

$$\nabla_z f \triangleq \frac{1}{2}(\nabla_x f - j\nabla_y f) \tag{B.9}$$

$$\nabla_{z^*} f \triangleq \frac{1}{2}(\nabla_x f + j\nabla_y f). \tag{B.10}$$

Following are some examples for gradient relations:

Quantity Q	$\mathbf{a}^\dagger\mathbf{b}$	$\mathbf{b}^\dagger\mathbf{a}$	$\mathbf{a}^\dagger B\mathbf{a}$
Gradient $\nabla_{\mathbf{a}} Q$	0	$\mathbf{b}^*$	$(B\mathbf{a})^*$
Gradient $\nabla_{\mathbf{a}^*} Q$	$\mathbf{b}$	0	$B\mathbf{a}$

Table B.1: Gradient relations for complex vectors.

Quantity Q	a^*b	ab	$\lvert a\rvert^2 = aa^*$
Gradient $\nabla_{\mathbf{a}} Q$	0	b	a^*
Gradient $\nabla_{\mathbf{a}^*} Q$	b	0	a

Table B.2: Gradient relations for complex scalars.

If constrained minimization should be conducted, the following steps should be taken. We will concentrate on the minimization in respect to a complex vector. We want to minimize $Q(\mathbf{a})$ subject to the (possibly complex) constraint $\mathcal{C}(\mathbf{a}) = 0$. We can divide the constraint to its real and imaginary parts:

$$\mathcal{C}_r(\mathbf{a}) = 0 \tag{B.11}$$

$$\mathcal{C}_i(\mathbf{a}) = 0. \tag{B.12}$$

We can use now the Lagrange multipliers:

$$\mathcal{L}(\mathbf{a}) = Q(\mathbf{a}) + \mu_1 \mathcal{C}_r(\mathbf{a}) + \mu_2 \mathcal{C}_i(\mathbf{a}). \tag{B.13}$$

Defining $\lambda = \lambda_r + j\lambda_i$ and using the obvious relationship:

$$\lambda \mathcal{C}(\mathbf{a}) + \lambda^* \mathcal{C}^*(\mathbf{a}) = 2\Re\lambda \mathcal{C}(\mathbf{a}) = 2\lambda_r \mathcal{C}_r(\mathbf{a}) - 2\lambda_i \mathcal{C}_i(\mathbf{a}). \tag{B.14}$$

We can identify $\mu_1 = 2\lambda_r$ and $\mu_2 = -2\lambda_i$, thus giving

$$\mathcal{L}(\mathbf{a}) = Q(\mathbf{a}) + \lambda \mathcal{C}(\mathbf{a}) + \lambda^* \mathcal{C}^*(\mathbf{a}). \tag{B.15}$$

Now the optimization can be performed by using the gradient in respect to either $\mathbf{a}$ or $\mathbf{a}^\dagger$. If C($\mathbf{a}$)is purely real, the last term is redundant and can be omitted.

Appendix C

Derivation of Expected Noise Reduction for Coherent Noise Field

Recall 6.3 and define

$$\begin{aligned} NC(t,e^{j\omega}) &= \mathbf{H}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\mathcal{H}(e^{j\omega})\times \\ &\left(\mathcal{H}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\mathcal{H}(e^{j\omega})\right)^{-1}\mathcal{H}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\mathbf{H}(e^{j\omega}) \\ &= \mathbf{H}^{\dagger}(e^{j\omega})\mathcal{X}(t,e^{j\omega})\mathbf{H}(e^{j\omega}) \end{aligned}$$

thus,

$$S^{n}_{oo}(t,e^{j\omega}) = S^{n}_{fbf}(t,e^{j\omega}) - \frac{|\mathcal{F}(e^{j\omega})|^2}{\|\mathbf{H}(e^{j\omega})\|^4}\times NC(t,e^{j\omega})$$

and further define,

$$\mathcal{X}(t,e^{j\omega}) = S_{\mathbf{NN}}(t,e^{j\omega})\mathcal{H}(e^{j\omega})\left(\mathcal{H}(e^{j\omega})^{\dagger}S_{\mathbf{NN}}(t,e^{j\omega})\mathcal{H}(e^{j\omega})\right)^{-1}\mathcal{H}^{\dagger}(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})$$

For clarity and simplicity we will omit the time and frequency dependency during the next derivation. Thus,

$$\mathcal{X}(t,e^{j\omega}) = \mathcal{X} = S_{\mathbf{NN}}\mathcal{H}\left(\mathcal{H}^{\dagger}S_{\mathbf{NN}}\mathcal{H}\right)^{-1}\mathcal{H}^{\dagger}S_{\mathbf{NN}}.$$

We further denote,

$$\mathcal{X} = \mathcal{K}\times\mathcal{L}\times\mathcal{M}$$

where, $\mathcal{K} = S_{\mathbf{NN}}\mathcal{H}$, $\mathcal{L} = \left(\mathcal{H}^\dagger S_{\mathbf{NN}}\mathcal{H}\right)^{-1}$ and $\mathcal{M} = \mathcal{H}^\dagger S_{\mathbf{NN}}$ Thus, $\mathcal{X}$ is a multiplication of three terms. Starting from $\mathcal{L}$ and using the detailed noise structure,

$$\begin{aligned}\mathcal{L} &= \left(\mathcal{H}^\dagger(S_{nn}\mathbf{B}\mathbf{B}^\dagger + \varepsilon I)\mathcal{H}\right)^{-1} \\ &= \left(S_{nn}(\mathcal{H}^\dagger\mathbf{B})(\mathcal{H}^\dagger\mathbf{B})^\dagger + \varepsilon\mathcal{H}^\dagger\mathcal{H}\right)^{-1}.\end{aligned}$$

If we use the filters $\mathbf{B}(e^{j\omega}) = \mathbf{A}(e^{j\omega})$, i.e., the noise source is located exactly at the desired signal location, we have $\mathcal{H}^\dagger(e^{j\omega})\mathbf{A}(e^{j\omega}) = 0$, and the calculation of the inverse is straight forward, yielding $\mathcal{L} = (\varepsilon\mathcal{H}^\dagger\mathcal{H})^{-1}$, and $\mathcal{K} = \mathcal{M} = 0$. Thus, no extra noise reduction is produced by the noise canceller branch, as expected. The total noise part of the output is given by,

$$S^n_{oo}(t, e^{j\omega}) = S^n_{fbf}(t, e^{j\omega}) = |\mathcal{F}(e^{j\omega})|^2|A_1(e^{j\omega})|^2$$

which is exactly the signal part derived in Section 6.2.

For the general case, $\mathbf{B}(e^{j\omega}) \neq \mathbf{A}(e^{j\omega})$, we should use the *Matrix Inversion Lemma*, yielding:

$$\mathcal{L} = \left(\frac{1}{\varepsilon}(\mathcal{H}^\dagger\mathcal{H})^{-1} - \frac{\frac{1}{\varepsilon^2}S_{nn}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}}{1 + \frac{1}{\varepsilon}S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}}\right).$$

Now, using the approximation $\frac{1}{1+\mu} \approx 1 - \mu$, for $\mu \to 0$ (μ properly defined), yields,

$$\begin{aligned}\mathcal{L} = \Bigg(&\frac{1}{\varepsilon}(\mathcal{H}^\dagger\mathcal{H})^{-1} - \frac{\frac{1}{\varepsilon}S_{nn}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}}{S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}} + \\ &\frac{S_{nn}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}}{\left(S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\right)^2}\Bigg)\end{aligned}$$

We can further denote,

$$\mathcal{L} = \mathcal{L}_1\mathcal{L}_2\mathcal{L}_3.$$

Now, calculating $\mathcal{X}$,

$$\mathcal{X} = \mathcal{K}\mathcal{L}\mathcal{M} = S_{\mathbf{NN}}\mathcal{H}\mathcal{L}\mathcal{H}^\dagger S_{\mathbf{NN}} = (Snn\mathbf{B}\mathbf{B}^\dagger + \varepsilon I)\mathcal{H}\mathcal{L}\mathcal{H}^\dagger(Snn\mathbf{B}\mathbf{B}^\dagger + \varepsilon I).$$

Opening the brackets we have twelve terms:

I. $\mathcal{K}_1\mathcal{L}_1\mathcal{M}_1 = \frac{1}{\varepsilon}S^2nn\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger$

II. $\mathcal{K}_1\mathcal{L}_2\mathcal{M}_1 = -\frac{1}{\varepsilon}S^3_{nn} \times \frac{\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathbf{B}\mathbf{B}^\dagger}{S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}}$
$= -\frac{1}{\varepsilon}S^2nn\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger$

III. $\mathcal{K}_1\mathcal{L}_3\mathcal{M}_1 = \frac{S^3_{nn}\mathbf{B}(\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B})^2\mathbf{B}^\dagger}{(S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B})^2} = S_{nn}\mathbf{B}\mathbf{B}^\dagger$

IV. $\mathcal{K}_2\mathcal{L}_1\mathcal{M}_1 = S_{nn}\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger$

V. $\mathcal{K}_2\mathcal{L}_2\mathcal{M}_1 = -S^2_{nn}\frac{\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathbf{B}\mathbf{B}^\dagger}{S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}} = -S_{nn}\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger$

VI. $\mathcal{K}_2\mathcal{L}_3\mathcal{M}_1 = \varepsilon\frac{S^2_{nn}\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger}{(S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B})^2} = \varepsilon\frac{\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger}{\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}}$

VII. $\mathcal{K}_1\mathcal{L}_1\mathcal{M}_2 = S_{nn}\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger$

VIII. $\mathcal{K}_1\mathcal{L}_2\mathcal{M}_2 = -S^2_{nn}\frac{\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger}{S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}} = -S_{nn}\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger$

IX. $\mathcal{K}_1\mathcal{L}_3\mathcal{M}_2 = \varepsilon\frac{S^2_{nn}\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger}{(S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B})^2} = \varepsilon\frac{\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger}{\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}}$

X. $\mathcal{K}_2\mathcal{L}_1\mathcal{M}_2 = \varepsilon\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger$

XI. $\mathcal{K}_2\mathcal{L}_2\mathcal{M}_2 = -\varepsilon\frac{\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger}{\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}}$

XII. $\mathcal{K}_2\mathcal{L}_3\mathcal{M}_2 = \varepsilon^2\frac{S_{nn}\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger}{(S_{nn}\mathbf{B}^\dagger\mathcal{H}(\mathcal{H}^\dagger\mathcal{H})^{-1}\mathcal{H}^\dagger\mathbf{B})^2}$

Note, that terms I,II terms IV,V and terms VII,VIII eliminate each other, and that terms VI,IX,X,XI,XII tend to zero as ε tends to zero. Only term III is left. Collecting all terms, we have for the noise part of the output,

$$S^n_{oo}(t,e^{j\omega}) = \frac{|\mathcal{F}(e^{j\omega})|^2}{||\mathbf{H}(e^{j\omega})||^4}\times$$
$$\Big\{\mathbf{H}^\dagger(e^{j\omega})S_{\mathbf{NN}}(t,e^{j\omega})\mathbf{H}(e^{j\omega}) - \mathbf{H}^\dagger(e^{j\omega})S_{nn}(t,e^{j\omega})\mathbf{B}(e^{j\omega})\mathbf{B}^\dagger(e^{j\omega})\mathbf{H}(e^{j\omega})\Big\} = 0$$

Appendix D

Speech Enhancement Using a Mixture-Maximum Model

We present a new spectral domain, speech enhancement algorithm [29],[30]. The new algorithm is based on a mixture model for the short time spectrum of the clean speech signal, and on a maximum assumption in the production of the noisy speech spectrum. The new algorithm is shown to be effective in improving the quality of speech signals corrupted by additive noise. The computational requirements of the algorithm can be significantly reduced, essentially without paying performance penalties, by incorporating a dual codebook scheme with tied variances. Experiments, using recorded speech signals and actual noise sources, show that in spite of its low computational requirements, the algorithm shows improved performance compared to alternative speech enhancement algorithms.

It is also noted that the suggested algorithm proved to be an efficient post-processor stage for multi-microphone speech enhancers.

The organization of the chapter is as follows. In Section D.2 we review the MixMax model of Nadas *et al.* [72] and show how it can be used for speech enhancement. In Section D.3 we discuss the speech model training issue. Various improvements and simplifications to the model that were found useful are discussed in Section D.4. Experimental results are provided in Section D.5. Section D.6 concludes the paper.

D.1 Introduction

The purpose of this work is to present a spectral domain algorithm, which produces high-quality enhanced speech on the one hand, and has low computational requirements on the other hand.

The algorithm is similar to the HMM-based, minimum mean square error (MMSE) filtering algorithm proposed by Ephraim *et al.* [62], [63], in the sense that it also uses a mixture of Gaussians HMM to model the speech signal. We note that for the purpose of speech enhancement, our experience is that a degenerated model with a single HMM state is as effective as the original model, provided that a sufficient number of mixtures is used. That is, the information provided by temporal acoustic transitions is far less important than the current acoustic information. Consequently, it is sufficient to use a mixture of Gaussians model which assumes independence from one frame to the other. Although the removal of the HMM assumption results in a significant simplification of the enhancement algorithm, it is still necessary to design and activate a series of Wiener filters (one for each mixture), whose outputs are properly combined to form the enhanced speech signal. Hence the simplified HMM MMSE algorithm might still be too complicated to implement in low-cost or low-power applications.

In the present paper we follow the simple MixMax model, which was originally suggested by Nadas *et al.* [72] to design a noise adaptive, speech recognition algorithm. In [72] a discrete-density HMM speech recognition system is considered, where the standard distance-measure approach of vector quantization has been replaced by a probabilistic quantization rule. A Gaussian mixture model has been trained, associating a prototype (codeword) with a single mixture component, and the quantization rule of minimum distance has been replaced by a rule of most likely prototype. In the presence of noise, the probability distributions (PDs) were modified in accordance with the noise level, using a simple MAXIMUM model (see Section D.2). The quantization rule has thus been adapted to the noise. The algorithm has been nicknamed the MixMax labeler, after the mix-

ture and maximum models. In [73] we proposed alternative noise robust speech recognition algorithms based on the MixMax model.

In this study we present an effective speech enhancement algorithm based on the MixMax model, with some modifications and further simplifications. Our discussion is supported by an experimental study using recorded speech signals and actual noise sources. The outcomes consist of the assessment of subjective distortion measures, including total output signal to noise ratio (SNR) and segmental SNR, and informal subjective quality evaluations.

D.2 The MixMax Model

Let $s[l] \quad l = 0, 1, \ldots, L-1$ denote the speech signal samples of the current frame (possibly weighted by some window function), and let $S(e^{j2\pi k/L})$ denote the Discrete Fourier Transform (DFT) of $s[l]$,

$$S(e^{j2\pi k/L}) = \sum_{l=0}^{L-1} s[l] e^{-j2\pi lk/L} \qquad k = 0, 1, \ldots, L-1. \tag{D.1}$$

Let $\mathbf{X}$ denote the $L/2+1$ dimensional, log-spectral vector of the speech signal, as shown in Figure D.1.

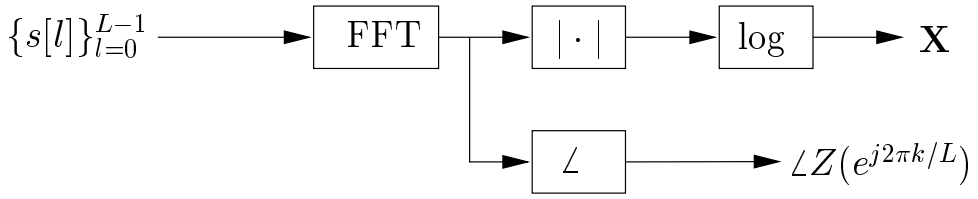

Figure D.1: Front-end signal processing.

More precisely, the k-th component of $\mathbf{X}$ is defined by,

$$X_k = X(e^{j2\pi k/L}) = \log ||S(e^{j2\pi k/L})|| \qquad k = 0, 1, \ldots, K-1.$$

where $K = L/2+1$ ($X_k \quad k = L/2+1, \ldots, L-1$ may be obtained using symmetry, i.e., $X_k = X_{L-k}$). We assume an additive colored noise model, which

is statistically independent of the speech signal. Similarly, let $\mathbf{Y}$ and $\mathbf{Z}$ denote the log-spectral vectors of the noise and noisy speech signals, respectively. We assume the availability of a voice activity detector (VAD). Based on the VAD indications of voice inactivity periods, we collect noise statistics, continuously and adaptively. Hence, we may assume that the (time varying) probability density of the noise, $\mathbf{Y}$, is known. For each frame we obtain an estimate $\hat{\mathbf{X}}$ to $\mathbf{X}$, based on $\mathbf{Z}$ and on the current density of the noise. The reconstructed speech signal, $\hat{s}[l]$, is given by,

$$\hat{s}[l] = \frac{1}{L}\sum_{l=0}^{L-1}\hat{S}(e^{j2\pi k/L})e^{j2\pi lk/L} \tag{D.2}$$

$$\hat{S}(e^{j2\pi k/L}) = \exp\left\{\hat{X}_k\right\}\angle Z(e^{j2\pi k/L}). \tag{D.3}$$

Note that the reconstructed phase angle is the original phase angle of the noisy speech, as is generally the case when using spectral-domain enhancement methods [60]. The rest of the derivation is concerned with models and methods to obtain $\hat{\mathbf{X}}$.

Let $f(\mathbf{x})$ denote the probability density function of $\mathbf{X}$ (for simplicity we avoid the more accurate notation, $f_{\mathbf{X}}(\mathbf{x})$). We assume that $f(\mathbf{x})$ can be modeled by a mixture of diagonal covariance Gaussians, i.e.:

$$f(\mathbf{x}) = \sum_i c_i f_i(\mathbf{x}) = \sum_i c_i \prod_k f_{i,k}(x_k), \tag{D.4}$$

where

$$f_{i,k}(x) = \mathcal{N}(x, \mu_{i,k}, \sigma_{i,k}).$$

$\mathcal{N}(x, \mu, \sigma)$ is the Gaussian density,

$$\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}.$$

Let $g(\mathbf{y})$ denote the probability density function of the spectral noise, $\mathbf{Y}$. We assume that $g(\mathbf{y})$ can be modeled by a single diagonal covariance Gaussian, i.e.:

$$g(\mathbf{y}) = \prod_k g_k(y_k)$$

where

$$g_k(y) = \mathcal{N}(y, \mu_{Y,k}, \sigma_{Y,k}).$$

Let Z denote the log-energy frame vector of the noisy speech signal. Note that

$$Z_k = \log\left(\exp(X_k) + \exp(Y_k)\right).$$

The last equation is accurate under the statistically independent, additive noise model, provided that X_k and Y_k are the log-spectral components corresponding to the speech and noise signals, respectively. Note however, that X_k and Y_k are defined by the corresponding short-term spectra, based on signal frames of N samples, so that the last equation involves some approximation. The assumption in the MixMax model, suggested by Nadas *et al.* [72], is that we can approximate Z_k by $\max(X_k, Y_k)$, that is

$$\mathbf{Z} \approx \max(\mathbf{X}, \mathbf{Y}),$$

where the maximum is carried out component-wise.

Let $F_{i,k}(x)$, $G_k(y)$ denote the cumulative distribution functions of $f_{i,k}(x)$ and $g_k(y)$, respectively. Note that,

$$G_k(y) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}\sigma_{Y,k}} e^{-\frac{1}{2\sigma_{Y,k}^2}(u-\mu_{Y,k})^2} du = \frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{y-\mu_{Y,k}}{\sqrt{2}\sigma_{Y,k}}) \tag{D.5}$$

where

$$\mathrm{erf}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-t^2} dt$$

is the error function. Similarly,

$$F_{i,k}(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}(\frac{x-\mu_{i,k}}{\sqrt{2}\sigma_{i,k}}). \tag{D.6}$$

The cumulative distribution function of Z_k given the i-th mixture, $H_{i,k}(z)$ is obtained by invoking the statistical independence of $\mathbf{X}$ and $\mathbf{Y}$ as follows,

$$\begin{aligned} H_{i,k}(z) &= \Pr\{Z_k < z | I = i\} \\ &= \Pr\{X_k < z, Y_k < z | I = i\} \\ &= F_{i,k}(z) G_k(z). \end{aligned} \tag{D.7}$$

Here I is the class (mixture) random variable. The density of Z_k given the i-th mixture, $h_{i,k}(z)$, is obtained by differentiating (D.7), [72],

$$h_{i,k}(z) = f_{i,k}(z)G_k(z) + F_{i,k}(z)g_k(z).$$

The probability density of Z is hence given by,

$$h(\mathbf{z}) = \sum_i c_i h_i(\mathbf{z}) = \sum_i c_i \prod_k h_{i,k}(z_k) = \sum_i c_i \prod_k \left[f_{i,k}(z_k)G_k(z_k) + F_{i,k}(z_k)g_k(z_k) \right]. \tag{D.8}$$

Nadas *et al.* used (D.8) to adapt a discrete density HMM-based speech recognition system in the presence of additive noise, by quantizing the feature vector using a probabilistic rule based on (D.8). In [73] the MixMax model is used to design other noise-adaptive speech recognition systems. In the present paper we apply the MixMax model to the related problem of speech enhancement.

Now, the estimated speech vector is given by

$$\hat{\mathbf{X}} = \mathrm{E}(X|Z) = \sum_i \hat{\mathbf{X}}_i \, q(i \mid \mathbf{Z} = \mathbf{z}) \tag{D.9}$$

where $q(i \mid \mathbf{Z} = \mathbf{z})$, the class conditioned probability is given by

$$q(i \mid \mathbf{Z} = \mathbf{z}) = \frac{c_i h_i(\mathbf{z})}{h(\mathbf{z})} = \frac{c_i h_i(\mathbf{z})}{\sum_j c_j h_j(\mathbf{z})}. \tag{D.10}$$

$\hat{X}_{i,k}$, the k-th component of $\hat{\mathbf{X}}_i$ is the expected value of X_k given the class i and the noisy observation z_k,

$$\hat{X}_{i,k} = \mathrm{E}\left\{ \mathrm{X_k} \mid \mathrm{Z_k} = \mathrm{z_k}, \mathrm{I} = \mathrm{i} \right\} = \int \mathrm{x_k} \frac{\mathrm{f_{i,k}(x_k) h_{i,k}(z_k \mid X_k = x_k)}}{\mathrm{h_{i,k}(z_k)}} \, \mathrm{dx_k}. \tag{D.11}$$

where $h_{i,k}(\cdot \mid X_k = x_k)$ is the conditional density of Z_k given $I = i$ and $X_k = x_k$. Note that

$$\Pr\left\{ Z_k < z_k \mid X_k = x_k \right\} = \Pr\left\{ Y_k < z_k \right\} u(z_k - x_k)$$

where $u()$ is the unit step function. Differentiating the last expression with respect to z_k, $h_{i,k}(z_k \mid X_k = x)$ is obtained. Now, recalling the Gaussian assumption for $f_{i,k}$, and invoking the integration required by (D.11), we obtain

$$\hat{X}_{i,k} = z_k \rho_{i,k} + (\mu_{i,k} - \sigma_{i,k}^2 R_{i,k})(1 - \rho_{i,k})$$

where

$$R_{i,k} = f_{i,k}(z_k)/F_{i,k}(z_k); \quad R_{Y,k} = g_k(z_k)/G_k(z_k); \quad \rho_{i,k} = \frac{1}{1 + R_{Y,k}/R_{i,k}}.$$

D.3 Model Training

To apply the method of Section D.2, a mixture model of the type of equation (D.4) needs to be trained, using the maximum likelihood approach outlined in [72]. Let the training data consist of N frames, $x = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N)$. The objective is to set $c_i, \mu_{i,k}, \sigma_{i,k}$ so as to maximize the log-likelihood,

$$\log f(\mathbf{x}) = \sum_n \log f(\mathbf{x}^n).$$

The maximization may be carried out by using the expectation-maximization (EM) algorithm [74], as in [72]. Let $\gamma_{n,i}$, and $\alpha_{n,i}$ be defined by,

$$\begin{aligned} \gamma_{n,i} &= f(\mathbf{x}^n, I_n = i) = c_i f(\mathbf{x}^n | I_n = i) \\ &= c_i \prod_{k=0}^{K-1} \frac{1}{\sqrt{2\pi}\sigma_{i,k}} \exp\left\{ -\frac{(x_k^n - \mu_{i,k})^2}{2\sigma_{i,k}^2} \right\} \\ \alpha_{n,i} &= \Pr(I_n = i | \mathbf{x}^n) = \frac{\gamma_{n,i}}{\sum_{i'=0}^{M-1} \gamma_{n,i'}} \end{aligned} \tag{D.12}$$

M is the total number of mixtures. Note that $\alpha_{n,i}$ are the class-conditioned probabilities. Let c_i, $\mu_{i,k}$ and $\sigma_{i,k}^2$ denote the current values of the model parameters, and let $\hat{c}_i$, $\hat{\mu}_{i,k}$ and $\hat{\sigma}_{i,k}^2$ denote the values of the model parameters after the iteration. The EM iteration is given by,

$$\hat{c}_i = \frac{\sum_{n=0}^{N-1} \alpha_{n,i}}{N} \qquad i = 0, \ldots, M-1 \tag{D.13}$$

$$\hat{\mu}_{i,k} = \frac{\sum_{n=0}^{N-1} \alpha_{n,i} x_k^n}{\sum_{n=0}^{N-1} \alpha_{n,i}} \qquad i = 0, \ldots, M-1 \quad k = 0, \ldots, K-1 \tag{D.14}$$

$$\begin{aligned} \hat{\sigma}_{i,k}^2 &= \frac{\sum_{n=0}^{N-1} \alpha_{n,i} (x_k^n - \hat{\mu}_{i,k})^2}{\sum_{n=0}^{N-1} \alpha_{n,i}} \\ &= \frac{\sum_{n=0}^{N-1} \alpha_{n,i} (x_k^n)^2}{\sum_{n=0}^{N-1} \alpha_{n,i}} - \hat{\mu}_{i,k}^2 \; i = 0, \ldots, M-1 \; k = 0, \ldots, K-1 \end{aligned} \tag{D.15}$$

where $\alpha_{n,i}$ are computed using the current values of the parameters, c_i, $\mu_{i,k}$ and $\sigma^2_{i,k}$.

A simplified version of the EM algorithm is obtained by replacing equation (D.12) with the following,

$$\alpha_{n,i} = \begin{cases} 1 & \text{if } \gamma_{n,i} > \gamma_{n,j} \quad \forall j \\ 0 & \text{otherwise} \end{cases} .$$

In this case, for each data sample, n, we make a hard decision on the identity of the class, i. In our experience this modification reduces the computational cost, without any performance penalties.

To initialize the algorithm, we set $\boldsymbol{\mu}_i = (\mu_{i,0}, \mu_{i,1}, \ldots, \mu_{i,K-1})^T \quad i = 0, \ldots, M-1$ equal to M randomly chosen spectral vectors. We also set $\sigma_{i,k} = 1 \quad k = 0, \ldots, K-1 \quad i = 0, \ldots, M-1$, and $c_i = \frac{1}{M} \quad i = 0, \ldots, M-1$ (uniformly distributed mixtures).

D.4 Implementation

The following improvements and simplifications were found useful.

D.4.1 Tied Variances

We use the same mixture model (D.4), except that the variance of the k-th spectral component is now independent of the mixture.

$$\sigma_{i,k} = \sigma_k \qquad \forall \quad i = 0, \ldots, M-1 \quad k = 0, \ldots, K-1. \tag{D.16}$$

That is, the variances, $\{\sigma_{i,k}\}_{i=0}^{M-1}$ are tied together. The EM iteration is now described by equations (D.13), (D.14) and by the following equation that replaces equation (D.15),

$$\hat{\sigma}^2_k = \frac{1}{N} \sum_{i=0}^{M-1} \sum_{n=0}^{N-1} \alpha_{n,i} \left(x^n_k - \hat{\mu}_{i,k}\right)^2 \qquad k = 0, \ldots, K-1.$$

Tied variances enable a more compact representation, that is, when tying is applied, only K variance parameters are required (instead of $K \cdot M$). Our exper-

iments indicate that this saving is usually possible without a significant loss in the performance.

D.4.2 Dual Codebook Scheme

Given the speech signal samples of the current frame $s[l] \quad l = 0, \ldots, L-1$ (possibly weighted by some window function), we define,

$$\begin{aligned} \Gamma &= \log \sqrt{\sum_{l=0}^{L-1} s^2[l]} \\ \tilde{X}_k &= \log ||S(e^{j2\pi k/L})|| - \Gamma = X_k - \Gamma \qquad k = 0, \ldots, K-1 \\ \tilde{\mathbf{X}} &= \left[\tilde{X}_0, \tilde{X}_1, \ldots, \tilde{X}_{K-1}\right]^T \end{aligned}$$

where $S(e^{j2\pi k/L})$ is defined by equation (D.1). Hence,

$$X_k = \tilde{X}_k + \Gamma.$$

Γ and $\tilde{\mathbf{X}}$ are the (logarithmic) gain and gain normalized spectrum of the frame. We assume separate mixture models to $\tilde{X}_k$ and Γ. Let i denote the mixture index that corresponds to $\tilde{\mathbf{X}}$, and let j denote the mixture index that corresponds to Γ. The class conditioned density of X_k is

$$f_{i,j,k}(x_k) = \mathcal{N}(x_k, \mu_{i,k} + \mu_j^g, \sigma_k).$$

$\mu_{i,k}$ is the mean value that corresponds to the k-th component of the i-th mixture of $\tilde{\mathbf{X}}$. Similarly, μ_j^g is the mean value that corresponds to the j-th mixture of Γ. Note that we assume a tied variances model. Denote by M_1, the total number of mixtures that correspond to $\tilde{\mathbf{X}}$. Similarly, denote by M_2, the total number of mixtures that correspond to Γ. The density of $\mathbf{X}$ is,

$$f(\mathbf{x}) = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} c_i c_j^g f_{i,j}(\mathbf{x}) = \sum_i \sum_j c_i c_j^g \prod_k f_{i,j,k}(x_k)$$

where c_i, c_j^g are the mixture components that correspond to $\tilde{\mathbf{X}}$ and Γ respectively.

We estimate $\boldsymbol{\mu}_i = (\mu_{i,0}, \mu_{i,1}, \ldots, \mu_{i,K-1})^T \quad i = 0, \ldots, M_1 - 1$ by clustering the gain normalized spectrum $\tilde{\mathbf{X}}$, using a K-means algorithm. We then estimate

$\mu_j^g \quad j = 0, \ldots, M_2 - 1$ by clustering the gains, Γ. c_i is obtained as a by-product of the K-means algorithm, by calculating the relative frequency of gain normalized spectrum vectors, classified as belonging to the i-th mixture. c_j^g is obtained similarly, by calculating the relative frequency of gains classified as belonging to the j-th mixture. Finally, the variances, σ_k are obtained using,

$$\hat{\sigma}_k^2 = \frac{1}{N} \sum_{n=0}^{N-1} \left(x_k^n - \mu_{i_n,k} - \mu_{j_n}^g \right)^2$$

where i_n is the index of the mixture mean which is closest to $\tilde{x}^n$, i.e.,

$$i_n = \arg\min_i ||\boldsymbol{\mu}_i - \tilde{\mathbf{x}}^n||^2.$$

Similarly,

$$j_n = \arg\min_j ||\mu_j^g - \Gamma^n||^2$$

i_n and j_n are obtained as a by-product of the K-means procedure.

D.4.3 Replacing Weighted Mixtures by the Most Probable Mixture Element

A simplification that was found useful was construction of the enhanced speech based only on the most probable mixture. That is, equation (D.9) is now replaced by

$$\hat{\mathbf{X}} = \hat{\mathbf{X}}_l$$

where

$$l = \arg\max_i q(i \mid \mathbf{Z} = \mathbf{z}) = \arg\max_i c_i h_i(\mathbf{z}).$$

D.4.4 Logarithmic Arithmetic

To avoid numerical problems in the calculations, it is recommended to use logarithmic arithmetic [72]. Let $\{v_i\}$ be some given set of real numbers. Then, to evaluate $\log \sum_i e^{v_i}$, we use the following relation,

$$\log \sum_i e^{v_i} = v_{\max} + \log \sum_i e^{v_i - v_{\max}} \tag{D.17}$$

where $v_{\max} = \max_{1 \le i \le N} v_i$. Equation (D.17) may be used to evaluate equations (D.8) and (D.10) by noting the relations,

$$\begin{aligned}
\log h(\mathbf{z}) &= \log \sum_i e^{\log c_i + \log h_i(\mathbf{z})} \\
\log h_i(\mathbf{z}) &= \sum_k \log h_{i,k}(z_k) \\
\log h_{i,k}(z) &= e^{\log f_{i,k}(z) + \log G_k(z)} + e^{\log F_{i,k}(z) + \log g_k(z)} \\
q(i|Z=z) &= \exp\left\{\log c_i + \log h_i(\mathbf{z}) - \log h(\mathbf{z})\right\}.
\end{aligned}$$

Note that $\log f_{i,k}(\mathbf{z})$ and $\log g_k(\mathbf{z})$ are quadratic forms in z. Recall (D.5) and (D.6), $\log G_k(z)$ and $\log F_{i,k}(z)$ may be computed efficiently using (possibly a tabulated form of) the function, $\log(.5 + .5\mathrm{erf}(u))$.

D.4.5 Nonlinear Post-processing

Non-linear post-processing was applied in the past in spectral subtraction methods [59], [69]. We found nonlinear post-processing to be very effective in improving the quality of the enhanced speech. Let $\gamma_k = \exp\left\{\hat{X}_k - Z_k\right\}$. γ_k is the spectral gain (in fact, suppression, since $\gamma_k < 1$) of the k-th channel. A simple and useful post-processing is obtained by constraining γ_k to be above some frequency dependent threshold, δ_k. That is, the reconstructed speech signal is given by

$$\begin{aligned}
\tilde{s}[l] &= \frac{1}{L} \sum_{l=0}^{L-1} \tilde{S}(e^{j2\pi k/L}) e^{j2\pi lk/L} \\
\tilde{S}(e^{j2\pi k/L}) &= \exp\left\{\tilde{X}_k\right\} \angle Z(e^{j2\pi k/L}) \\
\tilde{X}_k &= \max(Z_k + \log \delta_k, \hat{X}_k).
\end{aligned}$$

D.5 Experiments

To test the performance of the new MixMax algorithm, we used 8 sentences from the TIMIT and Resource Management databases (3 females, 5 males). All sentences were initially down-sampled from 16KHz to 8KHz. Clean speech model

training was performed using 10 other TIMIT sentences. Frame length was $L = 256$. Hence, $K = 129$. The threshold δ_k in Subsection D.4.5 was:

$$\delta_k = \begin{cases} 0.35 & \text{if } 0 \leq k \leq 36 \\ 0.18 & \text{if } 37 \leq k \leq 128 \end{cases} . \tag{D.18}$$

Hence δ_k is higher for frequencies lower than 1125Hz ($k = 36$). Frame overlapping of 50% was used.

The sentences were corrupted by additive noise, using two types of noise signals. The first was a synthetic white Gaussian noise source. The second was a computer fan signal that was recorded in our laboratory. Various signal to noise ratios (SNRs) were used in the experiments. We assumed the existence of a reliable voice activity detector. Hence, in all our experiments, prior to the speech enhancement stage, the noise parameters $\mu_{Y,k}, \sigma_{Y,k}$ were estimated using some independent segment from the noise source. The duration of that noise segment was set to 250 mSec.

To test the performance of the new algorithm, both objective and subjective listening tests were employed. The performance was compared to the performance of the HMM-based minimum mean square error (MMSE) speech enhancement algorithm that was proposed by Ephraim *et al.* [62], [63], with the post-processing modification that was outlined in Subsection D.4.5. We note that this post-processing operation significantly improved the quality of the HMM MMSE algorithm. We used the same set of values for the thresholds δ_k as indicated in equation (D.18) above.

We also compared the performance to the simple nonlinear spectral subtraction algorithm proposed by Boll [59]. Generally speaking, the spectral subtraction algorithm significantly reduces the noise level, but generates an annoying *musical noise* effect, i.e., tones with fast shifting frequencies, especially at low SNR values. Overall, the performance of the spectral subtraction algorithm was significantly inferior, both to the HMM MMSE algorithm and to the new proposed MixMax algorithm. This result is consistent with [68]. Hence in the rest of this section we report only on the performance of the HMM MMSE and MixMax algorithms.

Our objective set of experiments consisted of total output and segmental SNR measurements. These distortion measures are known to be correlated with the subjective perception of speech quality [75]. The total output SNR is defined by

$$\text{SNR} = \frac{\sum_t s^2[t]}{\sum_t \left(s[t] - \hat{s}[t]\right)^2} \tag{D.19}$$

where $s[t]$ and $\hat{s}[t]$ are the reference (e.g., clean) and test (e.g., enhanced) speech signals, and where the time summations are over the entire duration of the signals. Prior to the application of (D.19), $s[t]$ and $\hat{s}[t]$ are scaled to have unit energy over the entire sentence. Segmental SNR is defined by the mean value of the individual SNR measurements (using (D.19)) over the frames of the sentence. Segmental SNR is known to have better correlation with subjective quality, and is similar in that sense to the performance of the Itakura-Saito distance measure [75]. However, total output SNR is more robust to the presence of low energy regions (frames), or to frames for which the energy of $s[t] - \hat{s}[t]$ is small. To increase the robustness of the segmental SNR measure, we also used the median value of the individual SNR measurements (using (D.19)) over the frames of the sentence. Median averaging eliminates outliers (which are due to the reasons outlined above), and is therefore superior to the common definition of segmental SNR that involves simple averaging.

Figure D.2 illustrates the performances of the HMM MMSE and MixMax algorithms, for the case where 20 Gaussian mixtures are used, both for white Gaussian noise (left) and for computer fan noise (right). The following objective distance measures were used: total SNR (TOTSNR), mean value segmental SNR (MEANSEG), and median value segmental SNR (MEDSEG). Figure D.2 presents the SNR (TOTSNR, MEANSEG and MEDSEG) enhancement, defined by the difference, in dB scale, between the output SNR (SNR of the enhanced speech, using the reference clean speech), and input SNR (SNR of the noisy speech, using the reference clean speech) as a function of the input SNR. All three distance measures show clear advantage to the simpler MixMax algorithm. The same trend can be seen in Figure D.3, where 5 Gaussian mixtures are used (for simplicity

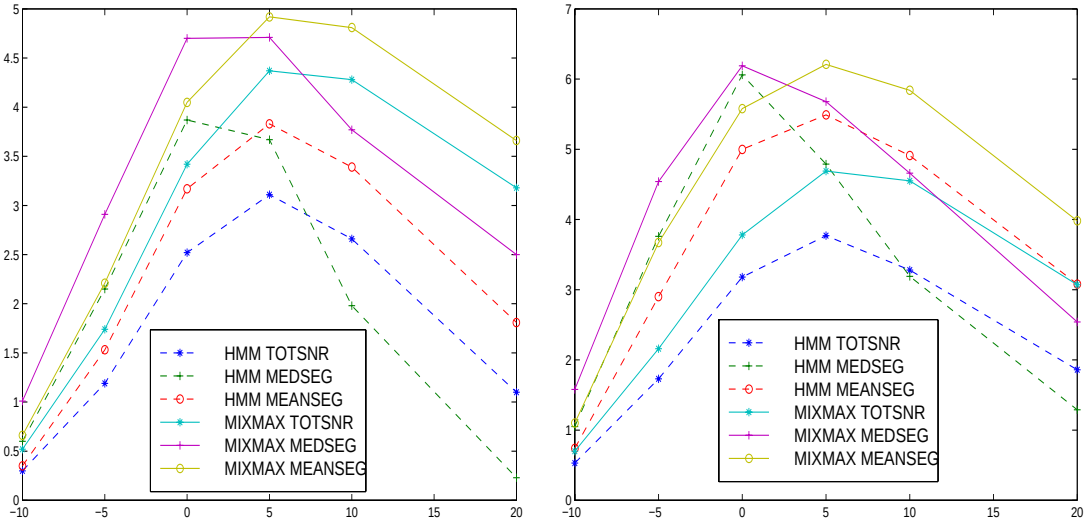

Figure D.2: HMM vs. MixMax, 20 mixtures (left: white noise, right: computer fan noise)

of presentation only TOTSNR and MEDSEG are shown, however MEANSEG demonstrates the same trend). These results were verified by informal listening tests, using several listeners. The quality of the enhanced MixMax speech signal was clearly improved compared to the HMM MMSE enhanced speech, over the entire SNR range examined. It should be noted that post-processing reduces objective quality measures such as SNR enhancement. However, at the same time it results in a significant improvement in the quality of the enhanced speech, especially for low SNR inputs.

Figure D.4 compares the performance of the MIXMAX algorithm when 20 mixtures are used, as in Figure D.2 (denoted by 1CB) to the performance of a reduced parameters, simplified variant of the algorithm (denoted by 2CB), in which the following changes were made:

1. We used the two codebook scheme, outlined in Subsection D.4.2, with $M_2 = 6$ mixtures for the gain and only $M_1 = 1$ mixture for the gain-normalized spectrum.

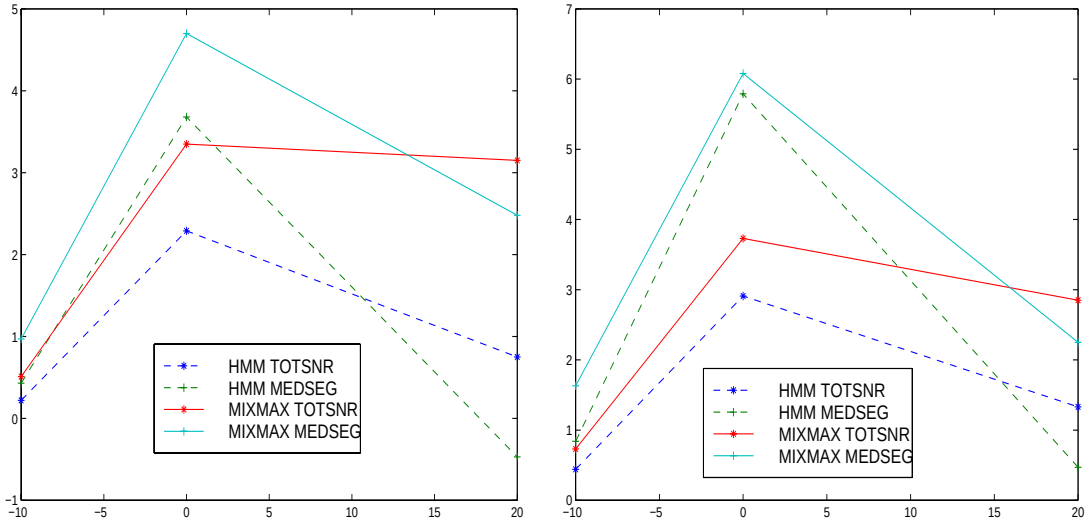

Figure D.3: HMM vs. MixMax, 5 mixtures (left: white noise, right: computer fan noise).

2. We used the most probable mixture variant of the algorithm, indicated in Subsection D.4.3.

3. We used uniform mixture components ($c_i = 1/M_1$ and $c_j^g = 1/M_2$).

In spite of these simplifications, and the significant reduction in the number of parameters used, there is almost no loss in terms of objective SNR. There is virtually no loss in the performance in terms of subjective speech quality.

D.6 Conclusions

We presented a new speech enhancement algorithm which was shown to be effective for improving the quality of the reconstructed speech, as well as a post-processor for multi-microphone speech enhancement algorithm. The new algorithm has very low computational requirements. The derivation is based on the MixMax model which was originally proposed for designing a noise adaptive speech recognition algorithm. Several modifications and simplifications were found useful. In particular, by using a dual codebook scheme, that also incor-

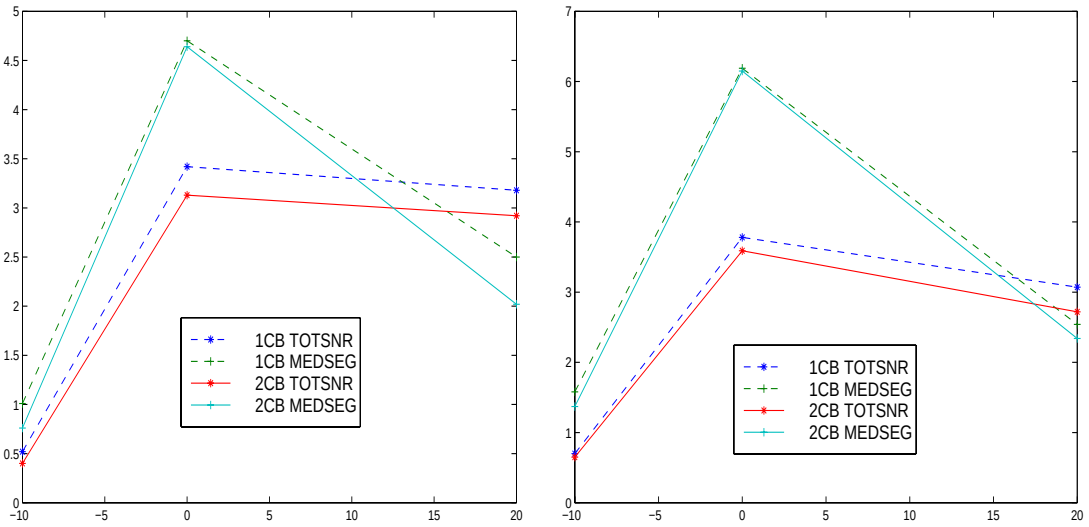

Figure D.4: One vs. two codebooks (left: white noise, right: computer fan noise).

porates tied variances, it is possible to significantly reduce the amount of model parameters (thus minimizing the memory and computational requirements of the algorithm), essentially without paying performance penalties. Post-processing was essential to produce high quality speech, although objective distortion measures, such as total or segmental SNR, indicate the opposite.

Bibliography

[1] J.F. Cardoso, "Blind Signal Separation: Statistical Principles," *Proc. of the IEEE*, vol. 86, no. 10, pp. 2009–2025, Oct. 1998.

[2] B.D. Van Veen and K.M. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering," *IEEE Acoustics, Speech and Signal Proc. magazine*, Apr. 1988.

[3] H. Cox, R.M. Zeskind M.M Owen, "Robust Adaptive Beamforming," *IEEE trans. on Acoustics, Speech and Signal Proc.*, vol. 35, no. 10, pp. 1365–1376, Oct. 1987.

[4] O.L. Frost III, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, Jan. 1972.

[5] L. J. Griffiths and C. W. Jim, "An Alternative Approach to Linearly Constrained Adaptive Beamforming," *IEEE Trans. on Antennas and Propagation*, vol. AP-30, no. 1, pp. 27–34, Jan. 1982.

[6] O. Hoshuyama and A. Sugiyama, "A Robust Adaptive Beamformer for Microphone Arrays with a Blocking Matrix Using Constrained Adaptive Filters," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Atlanta, Georgia, USA, May 1996, pp. 925–928.

[7] O. Hoshuyama A. Sugiyama and A. Hirano, "A Robust Adaptive Microphone Array with Improved Spatial Selectivity and its Evaluation in a Real

Environment," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, Apr. 1997, pp. 367–370.

[8] O. Hoshuyama A. Sugiyama and A. Hirano, "A Robust Adaptive Beamformer for Microphone Arrays with a Blocking Matrix Using Constrained Adaptive Filters," *IEEE trans. on Signal Proc.*, vol. 47, no. 10, pp. 2677–2684, Oct. 1999.

[9] S. Haykin, *Adaptive Filter Theory*, chapter 9, p. 363, Information and System Sciences Series. Prentice Hall, 2nd edition, 1991.

[10] O. Hoshuyama and A. Sugiyama, "A Realtime Microphone Array on a Single DSP System," in *Int. Workshop on Acoustic Echo and Noise Control*, Pocono Manor, Pennsylvania, USA, Sep. 1999, pp. 92–95.

[11] S. Nordholm, I. Claesson and B. Bengtsson, "Adaptive Array Noise Suppression of Handsfree Speaker Input in cars," *IEEE trans. on Vehicular tech.*, vol. 42, no. 4, pp. 514–518, Nov. 1993.

[12] J. Meyer and C. Sydow, "Noise Cancelling for Microphone Array," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, Apr. 1997, pp. 211–214.

[13] B. Widrow S.D. Stearns, *Adaptive Signal Processing*, chapter 13, p. 393, Prentice-Hall Signal Processing Series. Prentice-Hall, 1985.

[14] M. Dahl I. Claesson and S. Nordebo, "Simultaneous Echo Cancellation and Car Noise Suppression Employing a Microphone Array," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, Apr. 1997, pp. 239–242.

[15] J. Bitzer, K.U. Simmer and K.D. Kammeyer, "Theoretical Noise Reduction Limits of the Generalized Sidelobe Canceller (GSC) for Speech Enhancement," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Phoenix, Arizona, USA, May 1999.

[16] R. Zelinski, "A Microphone Array With Adaptive Post-Filtering For Noise Reduction In Reverberant Rooms," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, 1988, pp. 2578–2581.

[17] J. Meyer and K.U. Simmer, "Multichannel Speech Enhancement in a Car Environment using Wiener Filtering and Spectral Subtraction," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, Apr. 1997.

[18] N. Dal-Degan and C. Prati, "Acoustic noise analysis and speech enhancement techniques for mobile radio application," *Signal Processing*, vol. 15, no. 4, pp. 43–56, Jul. 1988.

[19] S. Fischer and K.D. Kammeyer, "Broadband Beamforming with Adaptive Postfiltering for Speech Acquisition In Noisy Environment," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, 1997, pp. 359–362.

[20] C. Marro, Y. Mahieux and K.U. Simmer, "Analysis of Noise Reduction and Dereverberation Techniques Based on Microphone Arrays with Postfiltering," *IEEE trans. on Speech and Audio Proc.*, vol. 6, no. 3, pp. 240–259, May 1998.

[21] J. Bitzer, K.U. Simmer and K-D. Kammeyer, "Multi-Microphone Noise Reduction by Post-Filter and Superdirective Beamformer," in *Int. Workshop on Acoustic Echo and Noise Control*, Pocono Manor, Pennsylvania, USA, Sep. 1999, pp. 100–103.

[22] J. Bitzer K-D. Kammeyer and K.U. Simmer, "An Alternative Implementation of the Superdirective Beamformer," in *IEEE Workshop on Application of Signal Processing to Audio and Acoustics*, New Paltz, New York, USA, Oct. 1999.

[23] S.E. Nordholm and Y. H. Leung, "Performance Limits of the Broadband Generalized sidelobe cancelling Structure in an Isotropic Noise Field," *J. Acoust. Soc. Am.*, vol. 107, no. 2, pp. 1057–1060, Feb. 2000.

[24] J. Meyer, K.U. Simmer and K.-D. Kammeyer, "Comparison of One- and two Channel Noise-Estimation Techniques," in *Int. Workshop on Acoustic Echo and Noise Control*, Imperial College, London, UK, Sep. 1999, pp. 17–19.

[25] J. Bitzer, K.U. Simmer and K.D. Kammeyer, "Multichannel Noise Reduction - Algorithms and Theoretical Limits," in *EUSIPCO*, Rhodes, Greece, Sep. 1998, vol. I, pp. 105–108.

[26] S. Fischer, K.-D. Kammeyer and K.U. Simmer, "Adaptive Microphone Arrays for Speech Enhancement in Coherent and Incoherent Noise Fields," in *3rd joint meeting of the Acoustical Society of America and the Acoustical Society of Japan*, Honolulu, Hawaii, USA, Dec. 1996, Slides are available in the WWW site: http://www.comm.uni-bremen.de.

[27] Sharon Gannot, David Burshtein and Ehud Weinstein, "Iterative-Batch and Sequential Algorithms for Single Microphone Speech Enhancement," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1997, vol. 2, pp. 1215–1218.

[28] S. Gannot, D. Burshtein and E. Weinstein, "Iterative and Sequential Kalman Filter-Based Speech Enhancement Algorithms," *IEEE Trans. on Speech and Audio Proc.*, vol. 6, no. 4, pp. 373–385, Jul. 1998.

[29] D. Burshtein and S. Gannot, "Speech Enhancement Using a Mixture-Maximum Model," in *6th European Conf. on Speech Communication and Tech. - EUROSPEECH*, Budapest, Hungary, Sep. 1999, vol. 6, pp. 2591–2594.

[30] D. Burshtein and S. Gannot, "Speech Enhancement Using a Mixture-Maximum Model," Submitted to IEEE Trans. on Speech and Audio Proc., Feb. 1999.

[31] S. Doclo and M. Moonen, "Robustness of SVD-Based Optimal Filtering for Noise Reduction in Multi-Microphone Speech Signals," in *Int. Workshop on*

Acoustic Echo and Noise Control, Pocono Manor, Pennsylvania, USA, Sep. 1999, pp. 80–83.

[32] E.E. Jan and J. Flanagan, "Sound Capture from Spatial Volumes: Matched-Filter Processing of Microphone Arrays Having Randomly-Distributed Sensors," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, Georgia, USA, 1996, pp. 917–920.

[33] D. Rabinkin, R. Renomeron, J. Flanagan and D.F. Macomber, "Optimal truncation time for matched filter array processing," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Seattle, Washington, USA, May 1998, pp. 3269–3272.

[34] Sofiène Affes, *Formation de voie adaptive en milieux réverbérants*, Ph.D. thesis, Ecole Nationale Supérieure des Télécommunications, Octobre 1995.

[35] Y. Grenier, "A Microphone Array for Car Environments," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, San Francisco, California, USA, 1992, pp. 305–308.

[36] S. Affes, S. Gazor and Y. Grenier, "Robust Adaptive Beamforming via LMS-like Target Tracking," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Adelaide, Australia, Apr. 1994, pp. 269–272.

[37] S. Gazor, S. Affes and Y. Grenier, "Wideband Multisource Beamforming with Adaptive Array Location Calibration and Direction Finding," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, 1995, pp. 1904–1907.

[38] S. Gazor, S. Affes and Y. Grenier, "Robust Adaptive Beamforming via Target Tracking," *IEEE trans. on Signal Proc.*, vol. 44, no. 6, pp. 1589–1593, Jun. 1996.

[39] S. Affes and Y. Grenier, "A Source Subspace Tracking Array of Microphones for Double Talk Situations," in *Int. Conf. on Acoustics, Speech and Signal Proc.*, Munich, Germany, Apr. 1996, pp. 269–272.

[40] S. Affes, S. Gazor and Y. Grenier, "An Algorithm for Multisource Beamforming and Multitarget Tracking," *IEEE trans. on Signal Proc.*, vol. 44, no. 6, pp. 1512–1522, Jun. 1996.

[41] S. Affes and Y. Grenier, "A Signal Subspace Tracking Algorithm for Microphone Array Processing of Speech," *IEEE trans. on Speech and Audio Proc.*, vol. 5, no. 5, pp. 425–437, Sep. 1997.

[42] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Trans. on Sig. proc.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[43] W. Kellerman, "Joint Design of Acoustic Echo Cancellation and Adaptive Beamforming for Microphone Arrays," in *Int. Workshop on Acoustic Echo and Noise Control*, Imperial College, London, UK, 1997, pp. 81–84.

[44] O. Shalvi and E. Weinstein, "System Identification Using Nonstationary Signals," *IEEE Trans. Signal Proc.*, vol. 44, no. 8, pp. 2055–2063, Aug. 1996.

[45] E. Weinstein M. Feder and A.V. Oppenheim, "Multichannel signal separation by decorrelation," *IEEE Trans. Speech Audio Processing*, vol. Vol. 1, no. No. 4, pp. 405–413, Oct. 1993.

[46] S. Gannot, D. Burshtein and E. Weinstein, "Signal Enhancement Using Beamforming and Non-Stationarity with application to Speech," Accepted for publication in IEEE Trans. on Sig. Proc. pending minor revisions, Nov. 2000.

[47] D.H. Brandwood, "A Complex Gradient Operator and its Application in Adaptive Array Theory," *IEE Proceedings*, vol. 130, no. 1, Parts F and H, pp. 11–16, Feb. 1983.

[48] G. Strang, *Linear Algebra and its Application*, Academic press, 2nd edition edition, 1980.

[49] B. Widrow, J.R. Glover Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeider, E. Dong Jr. and R.C. Goodlin, "Adaptive Noise Cancelling: Principals and Applications," *Proceeding of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[50] S. Nordholm, I. Claesson and P. Eriksson, "The Broadband Wiener solution for Griffiths-Jim Beamformers," *IEEE trans. on Signal Proc.*, vol. 40, no. 2, pp. 474–478, Feb. 1992.

[51] G.A. Clark, S.R. Parker and S.K. Mitra, "A Unified Approach to Time- and Frequency-Domain Realization of FIR Adaptive Digital Filters," *IEEE trans. on Acoustics, Speech and Signal Proc.*, vol. 31, no. 5, pp. 1073–1083, Oct. 1983.

[52] R.E Crochiere, "A weighted overlap-add method of short-time fourier analysis/synthesis," *IEEE Trans. on ASSP*, vol. Vol 28, no. No 1, pp. 99–102, Feb. 1980.

[53] Y. Bar-Ness, D.W. Chen and Z. Siveski, "Adaptive Multiuser Bootstrapped Decorrelating CDMA Detector in Asynchronous Unknown Channels," in *Personal, Indoor and Mobile Radio Communications (PIMRC'94)*, J.H. Weber, J.C. Arnbak and R. Prasad, Ed., The Hague, Netherlands, 1994, vol. 3, pp. 533–537.

[54] H. Robbins and S. Monro, "A Stochastic Approximation Method," *Ann. Math. Stat.*, vol. 22, pp. 400–407, 1951.

[55] H. Cox, "Spatial correlation in arbitrary noise fields with application to ambient sea noise," *J. Acoust. Soc. Am.*, vol. 54, no. 5, pp. 1289–1301, 1973.

[56] J.B. Allen and D.A. Berkley, "Image Method for Efficiently Simulating Small-Room Acoustics," *J. Acoust. Soc. Am*, vol. 65, no. 4, pp. 943–950, Apr. 1979.

[57] P.M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," *J. Acoust. Soc. Am*, vol. 76, no. 5, pp. 1527–1529, Nov. 1986.

[58] M.M. Goulding and J.S. Bird, "Speech Enhancement for Mobile Telephony," *IEEE trans. on Vehicular Tech.*, vol. 39, no. 4, pp. 43–56, Nov. 1990.

[59] Steven F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," in *Speech Enhancement*, Jae S. Lim, Ed., Alan V. Oppenheim series, pp. 61–68. Prentice-hall, 1983.

[60] Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 1109–1121, 1984.

[61] Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean Square Error Log-Spectral Amplitude Estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 443–445, 1985.

[62] Y. Ephraim, D. Malah and B. H. Juang, "On the application of hidden markov models for enhancing noisy speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1846–1856, 1989.

[63] Y. Ephraim, "A Bayesian Estimation Approach for Speech Enhancement Using Hidden Markov Models," *IEEE Trans. Signal Processing*, vol. 40, pp. 725–735, 1992.

[64] J. D. Gibson, B. Koo and S. D. Gray, "Filtering of colored noise for speech enhancement and coding," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 39, pp. 1732–1742, 1991.

[65] B. G. Lee, K. Y. Lee and S. Ann, "An em-based approach for parameter enhancement with an application to speech signals," *Signal Processing*, vol. 46, pp. 1–14, 1995.

[66] Jae S. Lim, *Speech Enhancement*, Prentice-Hall, 1983.

[67] K.K. Paliwal and A. Basu, "A Speech Enhancement Method Based on Kalman Filtering," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, 1987, pp. 177–180.

[68] H. Sameti, H. Sheikhzadeh, L. Deng and R. Brennan, "Comparative Performance of Spectral Subtraction and HMM-Based Speech Enhancement Strategies with Application to Hearing Aid Design," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Adelaide, Australia, Apr. 1994, vol. 1, pp. 13–16.

[69] R.J. Vilmur, J.J. Barlo, I.A. Gerson and B.L. Lindsley, *Noise Suppression System*, U.S. patent no. 4,811,404, 1989.

[70] Sharon Gannot, "Algorithms for Single Microphone Speech Enhancement," M.S. thesis, Tel-Aviv Univ., Apr. 1995.

[71] D. van Compernolle, W. Ma and F. Xie, "Speech Recognition in Noisy Environments with the Aid of Microphone Arrays," *Elsevier Speech Communication*, vol. 9, pp. 433–442, 1990.

[72] A. Nádas,D. Nahamoo and M.A. Picheny, "Speech Recognition Using Noise-Adaptive Prototype," *IEEE Trans. on Speech and Audio Proc.*, vol. 37, no. 10, pp. 1495–1505, Oct. 1989.

[73] A. Erell and D. Burshtein, "Noise Adaptation of HMM Speech Recognition Systems using Tied-Mixtures in Spectral Domain," *IEEE trans. on Speech and Audio Processing*, vol. 5, pp. 72–74, Jan. 1997.

[74] A.P. Dempster N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. Ser. 3g, pp. 1–38, 1977.

[75] A. H. Gray R. M. Gray, A. Buzo and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. vol. 28, pp. pp. 367–376, 1980.