Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE **Array-based Spectro-temporal Masking For Automatic Speech Recognition**

PRESENTED BY **Amir Reza Moghimi**

ACCEPTED BY THE DEPARTMENT OF

Electrical and Computer Engineering

__Richard Stern_____ _5/1/14_____
ADVISOR, MAJOR PROFESSOR DATE

__Jelena Kovacevic_____ ____5/1/14_____
DEPARTMENT HEAD DATE

APPROVED BY THE COLLEGE COUNCIL

_____Vijayakumar Bhagavatula_____ _____5/1/14_____
DEAN DATE

Array-based Spectro-temporal Masking for Automatic Speech Recognition

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Amir R. Moghimi
B.S., Electrical Engineering, Sharif University of Technology
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May 2014

Supervisor:	Dr. Richard Stern Department of Electrical and Computer Engineering
Committee Members:	Dr. Vijayakumar Bhagavatula Department of Electrical and Computer Engineering
	Dr. Bhiksha Raj Language Technologies Institute
	Dr. Mike Seltzer Microsoft Research

Notice of Copyright

Acknowledgements

I'd like to begin by thanking my advisor, Dr. Richard Stern. His support – financial, intellectual and moral – has carried me through graduate school. His belief in me has bolstered my own resolve and sustained my efforts. Most importantly, though, he has been a mentor and role model, one worthy of my respect and admiration; truly, that is all anyone can ask. I'd also like to thank Dr. Bhiksha Raj and Dr. Rita Singh for numerous discussions, ideas, pointers, lessons and general pieces of advice given throughout my time in graduate school. Their academic guidance and moral support have been essential and are greatly appreciated. If for no other reason, I value my time at Carnegie Mellon for the opportunity it provided to befriend and learn from these individuals.

There are many others who have helped me get to this point – teachers, professors, relatives and friends who have enlightened, inspired, encouraged, helped or otherwise touched me over the years. Even if I have failed to adequately express my gratitude to them, it is felt wholeheartedly. There are too many to mention, but I would like to thank a few by name. I thank my committee members, Dr. Bhagavatula and Dr. Seltzer, for the time, patience and expertise they have contributed to evaluating and improving my work. I am grateful to my professors at Sharif University – particularly to Dr. Shamsollahi and Dr. Salehi, for believing in me and giving me opportunities to learn and grow as an undergraduate.

To my own family, of course, I can hardly do justice in this limited space. I'd like to thank my brother, Omid, for being an everlasting source of friendship, camaraderie, antagonism, entertainment, challenge, self-awareness and discussions both inane and profound. Throughout our lives, his presence has enriched, balanced and seasoned me. To my parents, Reza and Susan, I am indebted for many things. They have shaped my character, giving me a sense of self-worth to sustain me and a moral and ethical compass to guide me. They instilled in me a love of knowledge, of learning, of curiosity, of reflection. Perhaps most importantly, though, through hard work and sacrifice they provided me with the tools, the experiences and the position in life to do what I have

done and to become who I am; they placed me on the high branch from which I first jumped. This is a privilege few people enjoy in this world, and for it I will forever be grateful.

Finally, I dedicate this thesis to my wife, Akram. Without your encouragement and unwavering support, this work would never have happened and this thesis never been written – nor, in all likelihood, would most things I do ever bear fruit. You have given my life meaning, passion, excitement and direction; you inspire me to do more, aim higher and be better. You have changed my life immeasurably, in ways of many of which even you are probably unaware. You are my motivation, my inspiration, the beginning and end of all my journeys. I love you.

Acknowledgement of Financial Support

The work appearing in this thesis has been funded by grants from:

- The National Science Foundation (Grant IIS-I0916918)
- Cisco Systems, Inc. (Grant 570877)

Abstract

Over the years, a variety of array processing techniques have been applied to the problem of enhancing degraded speech to improve automatic speech recognition. In this context, linear beamforming has long been the approach of choice, for reasons including good performance, robustness and analytical simplicity. While various nonlinear techniques – typically based to some extent on the study of auditory scene analysis – have also been of interest, they tend to lag behind their linear counterparts in terms of simplicity, scalability and flexibility. Nonlinear techniques are also more difficult to analyze and lack the systematic descriptions available in the study of linear beamformers.

This work focuses on a class of nonlinear processing, known as *time-frequency (T-F) masking* – a.k.a. spectro-temporal masking – whose variants comprise a significant portion of the existing techniques. T-F masking is based on accepting or rejecting individual time-frequency cells based on some estimate of local signal quality. Analyses are developed that attempt to mirror the beam patterns used to describe linear processing, leading to a view of T-F masking as "nonlinear beamforming". Two distinct formulations of these "nonlinear beam patterns" are developed, based on different metrics of the algorithms behavior; these formulations are modeled in a variety of scenarios to demonstrate the flexibility of the idea. While these patterns are not quite as simple or all-encompassing as traditional beam patterns in microphone-array processing, they do accurately represent the behavior of masking algorithms in analogous and intuitive ways.

In addition to analyzing this class of nonlinear masking algorithm, we also attempt to improve its performance in a variety of ways. Improvements are proposed to the baseline two-channel version of masking, by addressing both the mask estimation and the signal reconstruction stages; the latter more successfully than the former. Furthermore, while these approaches have been shown to outperform linear beamforming in two-sensor arrays, extensions to larger arrays have been few and unsuccessful. We find that combining beamforming and masking is a viable method of bringing the benefits

of masking to larger arrays. As a result, a hybrid beamforming-masking approach, called "post-masking", is developed that improves upon the performance of MMSE beamforming (and can be used with any beamforming technique), with the potential for even greater improvement in the future.

Contents

List of Figures

Chapter 1

Introduction

There is a long history of using array processing techniques to improve the robustness of automatic speech recognition systems in adverse enviromental conditions. In this case, the array elements will be sound sensors (*i.e.,* microphones) and the adverse conditions of interest are some combination of the following:

- Additive Noise - Usually modeled as Gaussian and independent of the signal(s)
- Interfering Signals - Frequently other speech, sometimes music or other sounds
- Reverberation - Modeled as a set of linear filters applied to the target and interferering signals

Of the three, additive noise is usually the least problematic; it also lends itself more easily to single-sensor processing solutions [1]. Therefore, recent array processing research tends to focus on the latter two; this work is no exception. The problem of interfering signals is of particular interest, as speech-on-speech interference is one of the most damaging forms of degradation, while being relatively common in real-world scenarios. The so-called "cocktail-party problem" has, in fact, long been of interest to researchers of the human auditory system [2, 3] and to those who attempt to mimic its functionality articificially [4]. In the domain of automatic speech recognition (ASR), it remains an open problem as cutting-edge systems

still fail when presented with interfering sources that are too numerous or too powerful.

The various array-based approaches to these problems can be broadly categorized into two groups: linear and nonlinear. The linear algorithms are based for the most part on classical linear beamforming ideas [5], with many modified to exploit specific properties of speech (*e.g.,* [6]). These approaches tend to have solid theoretical bases and lend themselves well to analyses, comparisons and secondary metrics; some of these will be discussed in Section 2.2.1 and referenced in Chapter 3. They also degrade relatively gracefully in the face of small errors or changes in the environment. The nonlinear approaches, on the other hand, are more of a mixed bag; most are based to some extent on various models of human auditory processing, itself a highly nonlinear process. They are more difficult to conceptualize, and thus, to analyze without resorting to experimental performance metrics such as word error rate (WER) [1] (see Sections 2.3 and 3.1). As will be discussed in Chapter 4, there are also significant performance gaps between linear and nonlinear array processing. Perhaps the most important gap is scalability; the performance of linear processing techniques can be improved simply by using larger and larger arrays, while nonlinear processing techniques typically do not scale nearly as well.

This work focuses on an important class of the nonlinear approaches, those based on time-frequency (T-F) masking [4]; its many variations and derivatives make up a considerable portion of the overall nonlinear array processing literature. Observations based on a review of existing literature on these methods will be presented in Sections 3.1 and 4.1; these, in turn, motivate the overall goals of this work:

1. To achieve a better understanding of these methods and their strengths and weaknesses, especially when compared to linear beamforming.

2. To develop analyses that will facilitate a more intuitive discussion of the methods, preferably using metrics with counterparts in the area of linear beamforming.

3. To improve the performance and scalability of these methods.

The remainder of this work is organized as follows: Chapter 2 will outline the problem and provide some background. Chapter 3 describes efforts to develop an analysis framework for masking algorithms – cultivating a view of these techniques as "nonlinear beamformers" – in effect addressing Objectives 1 and 2 above. Chapter 4 focuses on potential improvements to masking (Objective 3), comparing masking to beamforming in the process (Objective 1). Chapter 5 will summarize the contributions presented in Chapters 3 and 4 and conclude the discussion.

Chapter 2

Background

As mentioned in Chapter 1, the main goals of this work are to develop a better understanding of array-based T-F masking algorithms and to improve their performance. In both of these efforts, linear beamformers are the standard against which the various nonlinear array processing methods are measured.

Therefore, this chapter will begin with a brief overview of the array processing paradigm, touch upon linear processing and then move on to masking. A particular variation of masking, known as PDCW, will then be introduced; the remainder of this work will use variations of PDCW as stand-ins for masking in general. Care will be taken to develop analyses of these methods that are easily extensible to other variations of masking algorithms and potentially to other nonlinear methods, enabling more straightforward comparisons between linear and nonlinear array processing.

2.1 Setup of problem

In the realm of automatic speech recognition, array processing techniques have long been applied to the problem of signal separation. Consider Figure 2.1, wherein a target signal and multiple interfering signals originating from speakers (*i.e.,* sources) at different locations arrive at the various sensors of an array. The signal received at the p-th microphone (*i.e.,*

sensor) is

$$x_p(t) = a_{0p}s\left(t - \tau_{0p}\right) + \sum_l a_{lp}i_{lp}\left(t - \tau_{lp}\right) \tag{2.1}$$

where a_{lp} and τ_{lp} are the attenuation and delay of the path from the l-th interfering speaker to the p-th microphone (and $l = 0$ indicates the target speaker).

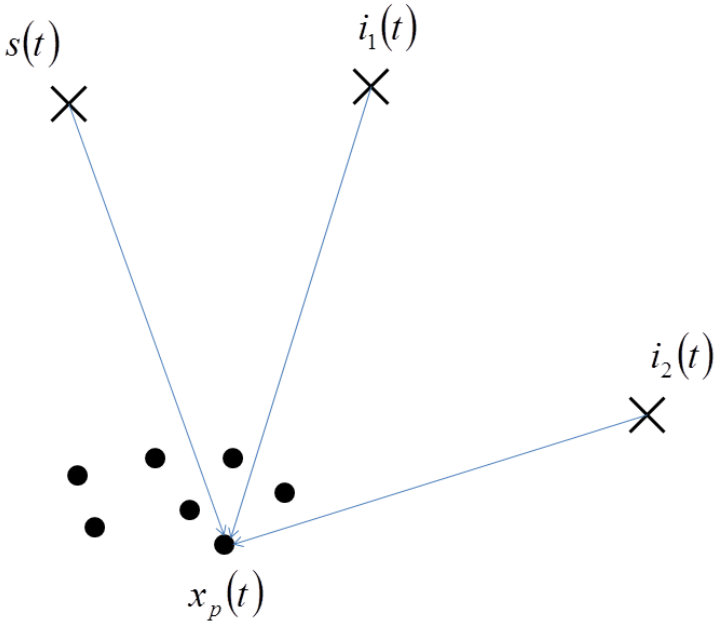


Figure 2.1: A generic array processing environment with multiple sources and multiple sensors – $s(t)$ represents the target signal and the $i_l(t)$ represent interfering signals

If the environment is reverberant, the room impulse responses [7] must be included:

$$x_p(t) = a_{0p}s\left(t - \tau_{0p}\right) * h_{0p}(t) + \sum_l \left[a_{lp}i_{lp}\left(t - \tau_{lp}\right) * h_{lp}(t)\right] \tag{2.2}$$

where the $h_{lp}(t)$ are causal, decaying impulse responses.

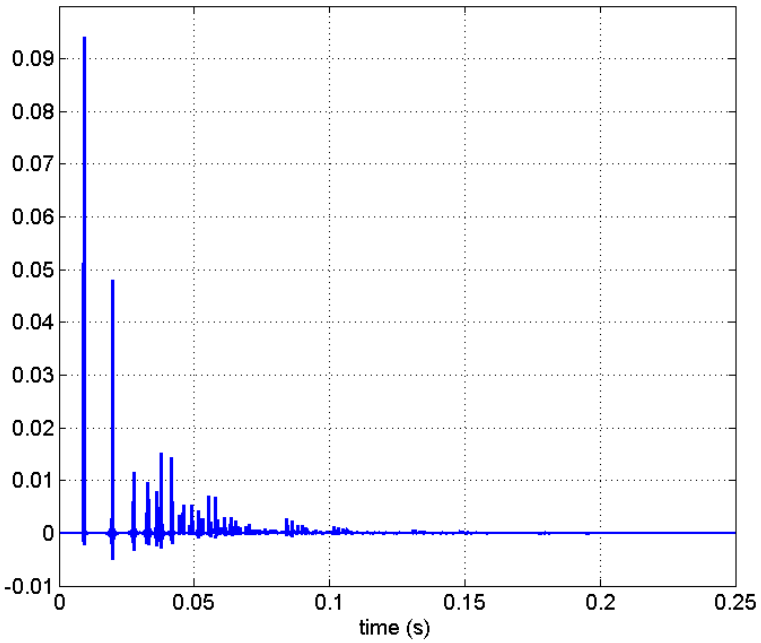
One of the most popular methods for the simulation of reverberant path impulse responses, known as the *image method* [8], can be used to model reverberant components. In short, each speaker is mirrored across the surfaces of the room to create a virtual speaker, whose signal is attenuated and delayed by amounts corresponding to the reflection charac-

teristics of the surface and the geometry of the room. Thus, each arriving component of the reverberation can be thought of as free-path propagation from one of these virtual sources. Equation (2.2) can be rewritten accordingly:

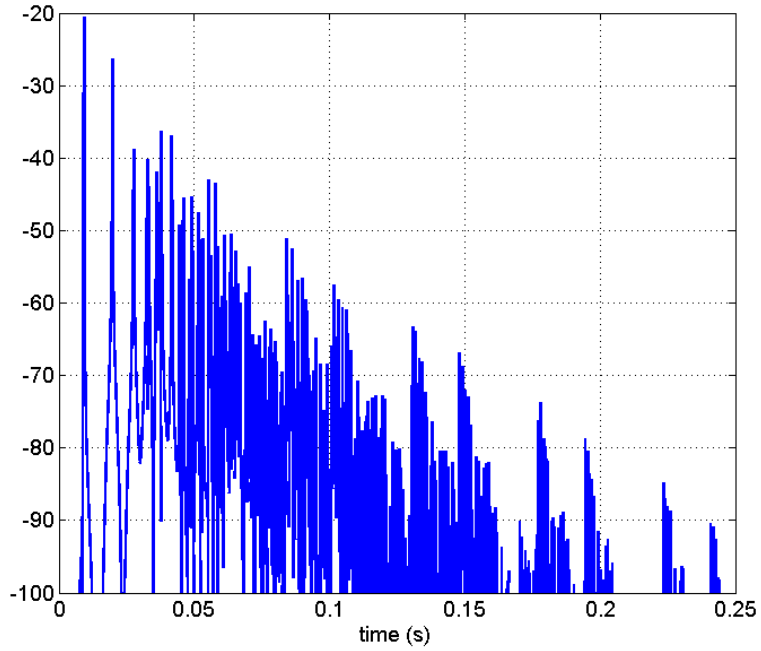
$$x_p(t) = \overbrace{\underbrace{a_{0p}s\left(t-\tau_{0p}\right)}_{\text{direct}} + \underbrace{\sum_r a'_{0pr}s\left(t-\tau'_{0pr}\right)}_{\text{reverberant}}}^{\text{signal}} + \overbrace{\sum_l \left[\underbrace{a_{lp}i_{lp}\left(t-\tau_{lp}\right)}_{\text{direct}} + \underbrace{\sum_r a'_{lpr}i_{lp}\left(t-\tau'_{lpr}\right)}_{\text{reverberant}}\right]}^{\text{interference}} \tag{2.3}$$

where the a' and τ' represent the attenuation and delay of the virtual sources. Note that the signal from each of these virtual sources is equal to either the target signal or one of the interferers; thus, each term in (2.3) representing a reverberated component will be correlated with a corresponding direct component. As an example, Figure 2.2a illustrates the impulse response of a path from a source to a destination in a featureless room with dimensions of $8 \times 5 \times 3$ meters, simulated using Habets' implementation of the image method [9].

Reverberant impulse responses typically consist of a few discrete impulses, representing the first few arriving copies of the signal (*i.e.,* the signal paths with few reflections and shorter delays), followed by a denser "tail". As an example, consider the first roughly 40 ms vs. the rest of the signal in Figure 2.2. The amplitude of the early reflections are usually much larger than that of the tail, whose amplitude decays over time in a roughly exponential fashion. This gives rise to a typical measure of reverberation, the *reverberation time* (RT) [7], which is defined as the time it takes for the envelope of the impulse response to drop to a certain level. A typical choice of level is 60 dB below the peak amplitude, with the corresponding reverb time labeled RT_{60}. To demonstrate, Figure 2.2b is a plot of the impulse response from Figure 2.2a but with the amplitude in dB; as can be seen, the RT_{60} of this particular impulse response is roughly 190 ms. It is also worth noting that



(a) Linear scale



(b) dB scale

Figure 2.2: Example of a reverberant impulse response, simulated using the image method

while the reverberation time depends heavily on the shape and physical properties of the room and objects in the environment, in most scenarios small changes in the locations of source and receiver do not affect the reverberation time significantly; thus, the concept of reverb time is often generalized to describe an acoustical environment, not just particular source/destination pairs. The impulse response from Figure 2.2, for example, comes from a room with a nominal $RT_{60} = 200$ ms.

2.1.1 A special case: The two-element array

Thus far, only a very generalized framework has been presented with the most basic of analyses. It will be useful to consider the simplification of this framework to a simple, yet widely used type of array: one with only two sensors. This configuration is illustrated in Figure 2.3, with a target and a single interferer. In the absence of a discussion on localization and steering errors, which are beyond the scope of this work, it can be assumed without loss of generality that the target signal is directly on the bisecting plane (*i.e.,* the array's broadside); this direction will be chosen as the origin for azimuth angles, and from it the interferer's position measured.

To simplify Equation (2.1), some assumptions will be made. The first is that the distance from the sources to the array is significantly greater than the size of the array; in this case, the free-path attenuations from each speaker will be equal for all microphones (*i.e.,* $a_{l1} = a_{l2} = a_l$). Furthermore, from the array's perspective, these attenuations can be folded into the original signal and interference powers; any difference in path loss will only be reflected as a shift in the received SIR (signal-to-interference ratio):

$$x_p(t) = s\left(t - \tau_{0p}\right) + i\left(t - \tau_{1p}\right)$$

This can be further simplified by choosing the origin of time for each signal as being the

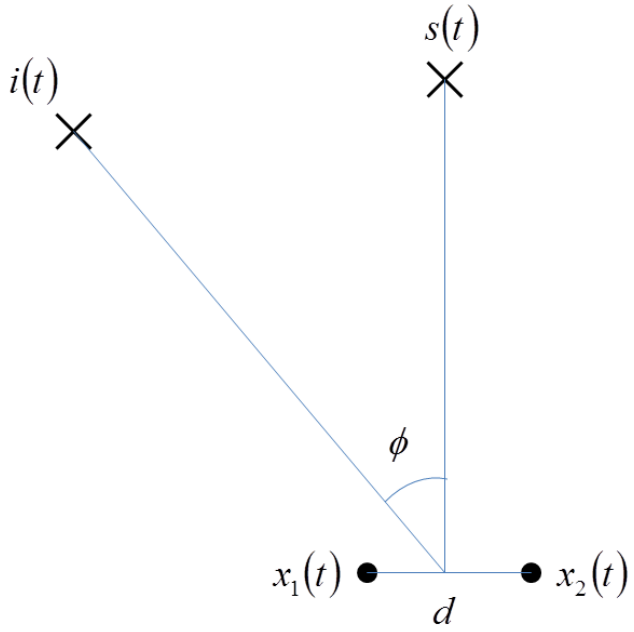


Figure 2.3: Two-sensor array with single interferer – d is the distance between the two sensors (microphones) and ϕ is the azimuth angle of the interferer

instant when the signal first impinges on a certain microphone, say the left[1]. The target will arrive at the two microphones simultaneously, so there is no relative delay. The interfering signal wavefront must travel a distance beyond the left microphone to arrive at the right; assuming a speed of sound c, the relative time delay will be $\tau = \frac{d}{c}\sin\phi$, leading to the following simplification of (2.1) for this configuration:

$$\begin{cases} x_1(t) = s(t) + i(t) \\ x_2(t) = s(t) + i\left(t - \dfrac{d}{c}\sin\phi\right) \end{cases} \tag{2.4}$$

or, in the frequency domain:

$$\begin{cases} X_1(f) = S(f) + I(f) \\ X_2(f) = S(f) + I(f)\, e^{-j2\pi f \frac{d}{c}\sin\phi} \end{cases} \tag{2.5}$$

[1]In the two-microphone setting, this text will refer to the left microphone and microphone 1 interchangeably. Similarly, the right microphone is microphone 2.

So the target signal is in phase at the two microphones, while the interferer is out of phase by an amount dependent on the geometry of the array (and which is a linear function of frequency). Keep in mind that in these equations, $s(t)$ and $i(t)$ are the signal and interference as received by the left microphone; they are attenuated and delayed, but not otherwise degraded, versions of the original target and interfering signals. Now, for the remainder of this text, signals and systems will be expressed in discrete time to accurately represent actual implementations. The sampling process is assumed to be ideal and free of aliasing, with sampling frequency f_S. Remember that discrete-time angular frequency ω and continuous-time frequency f are linearly related, as are discrete time index n and the corresponding continuous time points t_n at which the samples are taken (as in, $x[n] = x(t_n)$):

$$t_n = \frac{n}{f_S} \qquad \wedge \qquad f = \frac{\omega}{2\pi} f_S \tag{2.6}$$

Converting (2.5) to discrete time under these assumptions results in

$$\begin{cases} X_1\left(e^{j\omega}\right) = S\left(e^{j\omega}\right) + I\left(e^{j\omega}\right) \\ X_2\left(e^{j\omega}\right) = S\left(e^{j\omega}\right) + I\left(e^{j\omega}\right) e^{-j\omega f_S \frac{d}{c} \sin\phi} \end{cases} \tag{2.7}$$

The goal in these configurations is to separate the signal from the interferer(s); *i.e.,* to supress the interfering signal(s) as much as possible while causing minimal distortion of the target signal. This is, by and large, accomplished by creating a distinction based on the direction of arrival of a particular signal component, accepting components from the direction known to be close to the target and rejecting others. Incidentally, the "virtual sources" view of reverberation expressed in (2.3) suggests that successful spatial separation will also suppress much of the contribution of the reverberant path components.

2.2 Existing approaches

The various array-based approaches to spatial separation of incoming signals can be broadly categorized as either linear or nonlinear; these will be discussed in the following sections.

2.2.1 Linear beamforming

Linear array processing (a.k.a. beamforming) is defined as applying a linear filter to the signal received at each microphone. In short:

$$y[n] = \sum_p x_p[n] * g_p[n] \quad \overset{\mathcal{F}}{\longleftrightarrow} \quad Y\left(e^{j\omega}\right) = \sum_p X_p\left(e^{j\omega}\right) G_p\left(e^{j\omega}\right) \tag{2.8}$$

There are large bodies of literature on the basic theory of linear beamforming [5] and the numerous methods of applying it to speech processing [4]. Almost all variations have one point in common: the array filters $g_p[n]$ are chosen so that the target signal is not distorted. The methods vary in areas such as array geometry [10], formulation of the filters (*e.g.,* time domain vs. frequency domain or various system-level expressions of the array filters in terms of combinations of other filters) [11] and adaptations to fit a particular type of signal or environment. Another important categorization is fixed vs. adaptive beamformers. Fixed beamformers are designed beforehand and the filters are LTI, while adaptive beamformers have time-varying filters that adapt in real time to optimize some criterion (itself a source of many variations [12, 13, 14]). Of course, most adaptive beamformers are designed to converge to a steady-state LTI system in the presence of relatively stationary signals and environments.

In addition to the fact that state-of-the-art linear beamformers have proven to be robust in the face of interference and reverberation [4, 14], linear beamformers are simple to understand and to implement. In fact, the filtering operation itself is relatively computationally efficient and can be easily included in real-time systems (although, to be fair, some of the adaptive variations are much more complex). Furthermore, a number of what we will call "secondary"

metrics exist to quantify their behavior; some of these include:

1. *Beam Pattern*: Frequency response of the beamformer in a given direction. If the direction is expressed in spherical coordinates (as is customary), the beam pattern will be a function of frequency, azimuth and elevation: $B(f;\phi,\theta)$. In two dimensions, this will simplify to just frequency and azimuth: $B(f;\phi)$. As an example, Figure 2.4 illustrates the two-dimensional beam pattern of a simple four-element array where $\forall p:\ G_p(e^{j\omega}) = 1$.

2. *Beam Width*: Angular width of the main lobe of the beam pattern. Measured at the points where $|B(f;\phi)|$ drops to some specified fraction of its maximum.

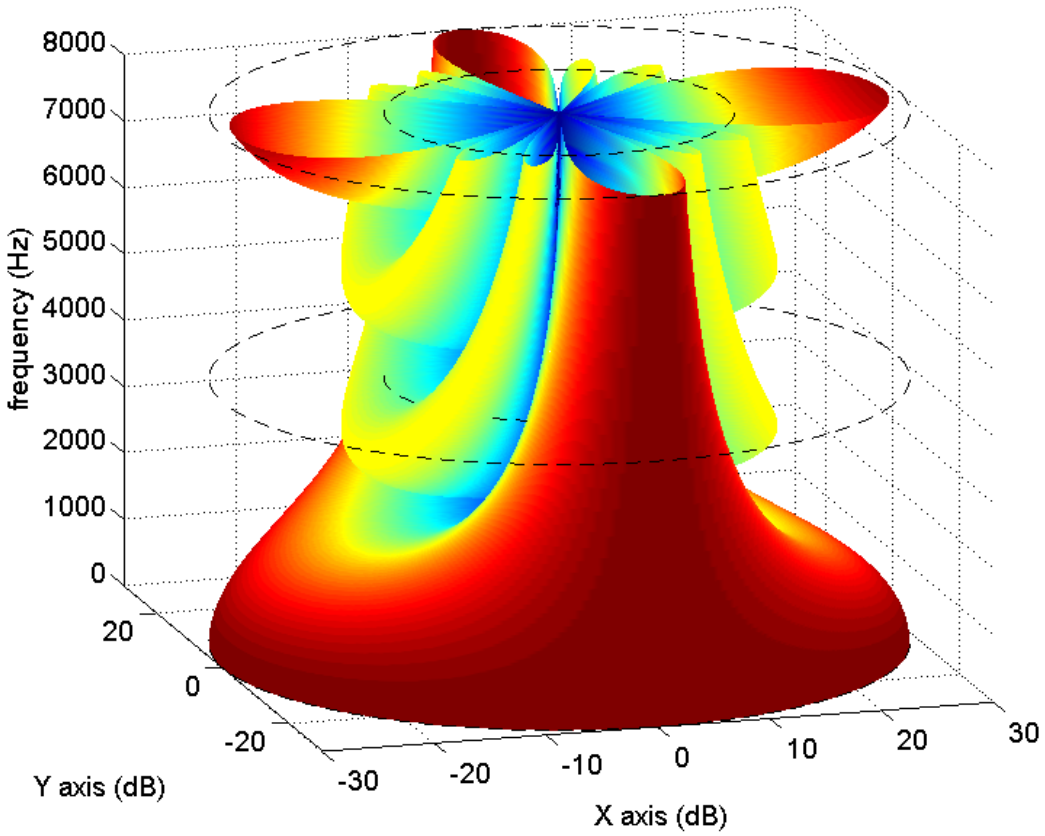


Figure 2.4: Beam pattern, in dB, of a four-element linear array with elements spaced at 4 cm

3. *Gain*: Ratio of maximum of the beam pattern to its average ($\langle\cdot\rangle_\phi$ denotes averaging over ϕ):

$$\text{Gain}\,(f) = \frac{\max_\phi |B\,(f;\phi)|}{\langle |B\,(f;\phi)| \rangle_\phi}$$

4. *Output SIR*: Signal-to-interference ratio on the output, after processing. Assuming a signal with power $|S|^2$ at $\phi = 0$ and interferers with powers $|I_l|^2$ at angles ϕ_l, this is:

$$SIR_{out} = \frac{|B\,(f;0)|^2\,|S|^2}{\sum_l |B\,(f;\phi_l)|^2\,|I_l|^2}$$

Chapter 3 will detail attempts to develop similar analysis metrics for nonlinear array processing methods.

2.2.2 Nonlinear techniques

The many array-based approaches to this problem that do not fit in the mold described in Section 2.2.1 can be loosely grouped together under the heading "nonlinear array processing". They can differ wildly in motivation, methodology and the type, number and order of nonlinear stages of processing. The ideas behind many are rooted to some extent in some model of human auditory perception [4] (itself a highly nonlinear process). For example, Stern, *et al.* [15] have developed a system based on correlating subband signals across time and frequency, which borrows heavily from models of human sound localization as described in [16].

Other variations are less directly motivated by physiology and based on other intuitive ideas. For example, a large class of algorithms are based on the idea of time-frequency (T-F) masking, which involves running the incoming signal(s) through some fashion of time-frequency analysis (typically the short-time Fourier transform), suppressing certain T-F cells based on some criterion and then reconstructing the signal from the residual. This class of algorithms is the focus of this work and will thus be discussed in more detail in Section 2.3.

2.3 Time-frequency masking

As mentioned, T-F masking algorithms share the following general pattern:

1. Run the microphone signals through some form of time-frequency analysis, typically the short-time Fourier transform (STFT).

2. Determine which T-F cells are worth keeping; set the others to zero. Mathematically, this is equivalent to multiplying the T-F-domain signal $X[n,k]$ by a "mask" $M[n,k]$, where the mask is always 0 or 1. Of course, this 0 or 1 constraint is far from universal. When it is in effect, the masking is called binary, but many variations of T-F masking relax this constraint in different ways.

3. Reconstruct the signal from this masked version.

As described, none of this is inherently nonlinear: The STFT operation is linear (as, indeed, are almost all analysis methods used in this first step), the masking operation is equivalent to a time-varying linear filter and the reconstruction can be linear or nonlinear, depending on the specific algorithm in question. In addition, there is nothing array-based in the procedure as presented; in fact, T-F masking is also widely applied to mono audio to improve signal quality for ASR [17, 18, 19] and for human intelligibility [20, 21].

The nonlinearity and the array dynamics come into play in determining the mask itself; this decision is also a major source of variety among the various flavors of masking algorithms. Numerous, sometimes incompatible, assumptions and objectives have been considered when developing decision mechanisms (*e.g.,* [22, 23, 24, 25, 26, 27, 28]). In the signal separation paradigm, one common and intuitive objective is to select those T-F cells where the signal power exceeds the interference power and reject the others. Going back to the two-microphone configuration of Figure 2.3 and (2.7), this can be expressed as

$$M[n,k] = \begin{cases} 1 & |S[n,k]| > |I[n,k]| \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

Of course, since $S[n,k]$ and $I[n,k]$ are not directly known to the array, this "oracle mask" must be estimated on a cell-by-cell basis from the input signals $X_1[n,k]$ and $X_2[n,k]$. Even aiming for this particular oracle mask, the method of estimation varies considerably. In a two-microphone setting, one approach is to estimate the inter-microphone time difference (ITD)[2] on a cell-by-cell basis and from there estimate a direction of arrival (DoA) for the signal in that cell. Assuming the interferer direction (*i.e.*, ϕ) is known, if the estimated DoA is closer to the target, $M[n,k] = 1$; otherwise $M[n,k] = 0$. ITD estimation itself can take many forms [27, 28], one of which will be detailed in Setion 2.3.1 as part of PDCW, the algorithm to be used as the baseline version of T-F masking throughout this work.

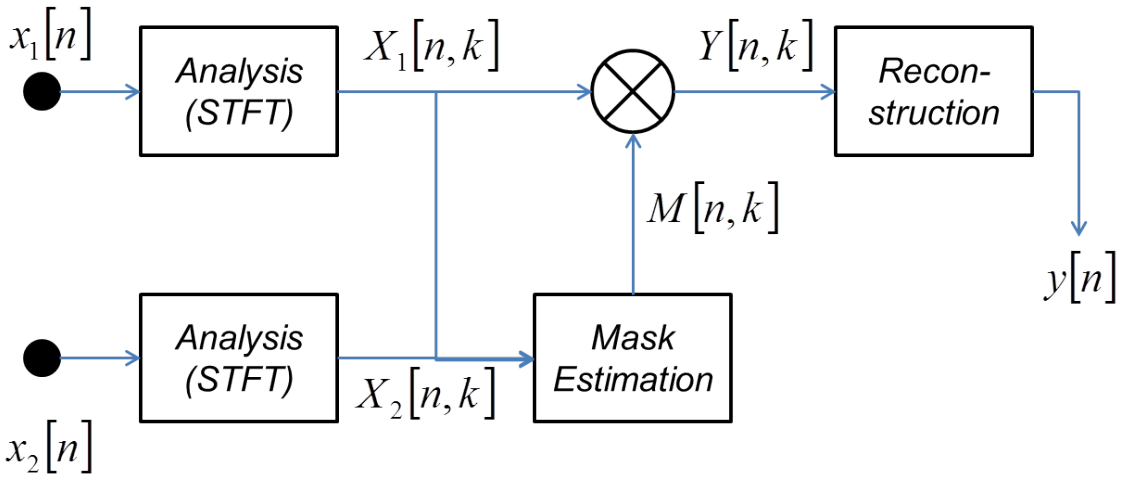


Figure 2.5: Block diagram of a generic two-sensor T-F masking algorithm

2.3.1 Phase-difference channel weighting (PDCW)

In [28], Kim and Stern introduced a version of two-sensor T-F masking which they termed Phase-Difference Channel Weighting (PDCW). In terms of the framework for masking presented in Section 2.3, the defining features of PDCW are outlined below:

The T-F analysis method is a regular STFT. The window length is set longer than the 20 ms typical in speech recognition; this is referred to as *medium-time processing* and

[2]Occasionally, the inter-microphone intensity difference (IID) is used as well; however, in small arrays in the absence of objects between the sensors, IID is usually negligibly small.

explained in [28]. In the experiments described in this work, 80-ms windows have been used.

The mask estimation stage aims to estimate the oracle mask of (2.9) via ITD estimation, as discussed. ITD, in turn, is estimated via the phase difference between the left and right signals:

$$\theta\left[n,k\right]=\angle X_1\left[n,k\right]-\angle X_2\left[n,k\right]=\angle\left(X_1\left[n,k\right]X_2^*\left[n,k\right]\right) \tag{2.10}$$

Combined with (2.7), it is easy to deduce that in the absence of interference or noise, $\theta\left[n,k\right]=0$. With no target signal and an interferer at azimuth ϕ, $\theta\left[n,k\right]=\omega_k f_S\frac{d}{c}\sin\phi$, where ω_k is the center frequency of subband k; *e.g.,* with an N-point DFT, $\omega_k=\frac{2\pi}{N}k$. A target and interferer of equal power will combine for a phase difference halfway between the two; this is the decision threshold:

$$M\left[n,k\right]=\begin{cases}1 & \left|\theta\left[n,k\right]\right|<\left|\frac{1}{2}\omega_k\frac{f_S d}{c}\sin\phi\right| \\ 0 & \text{otherwise}\end{cases} \tag{2.11}$$

Note the symmetry created by using absolute values; this ensures that interferers at both positive and negative azimuths (*i.e.,* off to the left and off to the right) are rejected, creating a "cone of acceptance" around the target speaker. In fact, this decision threshold is one of the more important tunable parameters of PDCW; by decreasing or increasing it, one can tighten or widen the cone of acceptance around the target. For example, in reverberant environments, a tighter bound is beneficial as it excludes more of the virtual sources, and therefore a higher amount of reverberant signal energy, resulting in a cleaner signal. To set a cone of acceptance at ϕ_T around the receiver, the phase decision threshold should be set as below:

$$M\left[n,k\right]=\begin{cases}1 & \left|\theta\left[n,k\right]\right|<\left|\gamma\left(\omega_k;\phi_T\right)\right| \\ 0 & \text{otherwise}\end{cases},\qquad \gamma\left(\omega;\phi\right)=\omega\frac{f_S d}{c}\sin\phi \tag{2.12}$$

where the definition of $\gamma\left(\omega;\phi\right)$ corresponds to the left-right phase difference that would be

observed in response to a hypothetical single source at azimuth ϕ and angular frequency ω. This is the threshold definition that will be used throughout this work.

For reconstruction, PDCW uses a simple overlap-add (OLA) synthesis, with one additional detail: Before masking, the binary masks are smoothed according the shape of gammatone filters distributed according to the Equivalent Rectangular Bandwidth (ERB) scale [29, 30]. This process is termed *channel weighting* and described in detail in [28]. This smoothing is done across frequency only, as the medium-time windowing at the first stage already has a smoothing effect across time. Specifically, given the gammatone filter frequency responses $H_i\left(e^{j\omega}\right)$ a set of weighting factors are calculated for the corresponding filters at each frame (n) from the estimated binary mask $M\left[n,k\right]$ and the input signal $X\left[n,k\right]$ for that frame:

$$w_i\left[n\right] = \frac{\sum_k M\left[n,k\right]\left|X\left[n,k\right]H_i\left(e^{j\omega_k}\right)\right|}{\sum_k \left|X\left[n,k\right]H_i\left(e^{j\omega_k}\right)\right|} \tag{2.13}$$

These weights are then used to smooth the mask:

$$\tilde{M}\left[n,k\right] = \frac{\sum_i w_i\left[n\right]\left|H_i\left(e^{j\omega_k}\right)\right|}{\sum_i \left|H_i\left(e^{j\omega_k}\right)\right|} \tag{2.14}$$

The smoothed mask $\tilde{M}\left[n,k\right]$ is then applied to the signal. Channel weighting has been shown to improve output signal quality, both subjectively and for ASR experiments, by reducing the distortion caused by the sudden level changes a binary mask introduces to the spectrogram.

Chapter 3

Analysis of Masking Techniques

3.1 Introduction and motivation

A review of the literature on nonlinear array processing techniques (*e.g.,* [15, 22, 23, 25, 26, 27, 28, 31, 32]) results in the following observation (among others): While many of these techniques are well motivated and based on intuitive reasoning, there is a conspicuous absence of thorough analysis methodology. The vast majority of papers discuss the reasoning behind and the implementation of their specific variation, then move directly into speech recognition experiments – or equivalent end-result experimentation, if targeting a domain other than ASR – to demonstrate the algorithm's performance. This is especially noticeable when compared to the linear beamforming literature. Although it is true that the definitive test of an algorithm designed for improving ASR performance is none other than ASR performance, there are a few reasons that an additional analysis technique would be beneficial, including:

- Speech recognition experiments tend to be complicated and time consuming.
- ASR results depend on a a number of factors, including the speech database, language model, recognition method and many tunable parameters. Comparisons between front-end processes do not necessarily hold across different scenarios.
- Due to the complexity of the speech recognition, cause-and-effect relationships are

frequently difficult to predict. In other words, it may be simple to observe that a certain change in the processing results in a performance improvement (or deterioration), but it is often very difficult to produce a satisfying explanation as to the reason why.

Furthermore, as T-F masking competes with linear beamforming, any metric that allows a more intuitive and direct comparison between a beamformer and a masking technique would give insight into the workings of both. This is no straightforward task, but this chapter details an attempt to do so, culminating in the model and analyses developed in Sections 3.3 and 3.4.

3.2 Sources of distortion and error

For an ASR system trained on clean speech, the system output $y[n]$ of a masker represents a distorted version of the clean target signal $s[n]$ expected by the recognizer. To quantify this distortion, we first assume a lossless T-F analysis-synthesis pair, such as the STFT and OLA used in PDCW; this allows us to calculate the distortion at the pre-reconstruction stage (*i.e.,* in $Y[n,k]$). For simplicity, the two-microphone configuration of Figure 2.3 will be considered here; extensions are trivial. In this case, from Figure 2.5 and (2.7), we obtain:

$$Y[n,k] = X_1[n,k] \cdot M[n,k] = S[n,k]\,M[n,k] + I[n,k]\,M[n,k] \tag{3.1}$$

Hence, the distortion relative to the clean signal can be expressed as:

$$D[n,k] = S[n,k] - Y[n,k] = \underbrace{S[n,k]\,(1 - M[n,k])}_{\text{signal suppression}} + \underbrace{I[n,k]\,M[n,k]}_{\text{interference leakthrough}} \tag{3.2}$$

If the masking is binary, the terms $1 - M[n,k]$ and $M[n,k]$ represent a simple T-F cell selection. In this light, masking algorithm design can be viewed as an optimization problem whose cost function is a nondecreasing function of these two competing distortions.

It should be noted that (3.1) does not accurately represent the actual implementation of

PDCW. In practice, both left and right signals are masked, after channel weighting, and the results averaged; *i.e.,* $Y[n,k] = \frac{1}{2}\left(X_1[n,k] \cdot \tilde{M}[n,k] + X_2[n,k] \cdot \tilde{M}[n,k]\right)$. This functions as a delay-and-sum beamforming step that further improves the performance of the masker. This averaging step has been omitted from the equations above in the interest of simplicity, but its inclusion does not change the basic point.

3.2.1 Feature distances and statistical prediction

In (3.2), the distortion is measured as the difference between the spectrograms of the clean and processed signal. Let us define the average distortion of a particular processing algorithm as the average Euclidean distance between the clean and processed signals, when expressed as spectrograms:

$$\bar{D} = E\left[\sqrt{\sum_k \left|D[n,k]\right|^2}\right] = E\left[\sqrt{\sum_k \left|S[n,k] - Y[n,k]\right|^2}\right] \tag{3.3}$$

where the expected value is over the independent, random signals $S[n,k]$ and $I[n,k]$; in practice, this will be estimated via time-averaging over long durations and many input signals.

Many similar distortion metrics can be defined, based on different transformations of the signals (*i.e.,* features) and different distance metrics. For example, power spectrograms, log-spectrograms, mel-cepstra (MFCCs) [4], gammatone filterbank outputs [29] or PNCC coefficients [33] can be used instead of spectrograms. As for distance metrics, instead of the L^2 norm (*i.e.,* Euclidean distance), any of the L^p norms can be used, in addition to statistical metrics such as K-L divergence and mutual information [34][1]. The value in defining such an average distortion metric for an algorithm is that, if it proves to be descriptive of the algorithm's performance, it is relatively simple to calculate computationally; simply run a large number of signals, degraded by various environments, through the array processing

[1]These statistical metrics rely on an assumption of some underlying probability distribution of the features; this can be a source of both complexity and inaccuracy.

and measure the average distortion.

To test the various metrics, we have done exactly that: run many speech recognition experiments, using basic PDCW, in a variety of scenarios, involving a target speaker at 0 and interferer at 30, 45 or 60 degrees with an signal-to-interference ratio (SIR) of 0, 10 or 20 dB or infinity (*i.e.,* no interfering signal) in an enclosure with a reverberation time of 0, 200 or 500 ms, for a total of 33 scenarios, with multiple experiments within each scenario. To create each (degraded) test utterance, the target and interfering signals are drawn randomly from the DARPA Resource Management (RM1) database. For each experiment, an average distortion metric is caluclated based on different combinations of the features and distance metrics mentioned above. Additionally, the processed signals are passed to the Sphinx-3 recognizer, with its acoustic models trained on clean speech drawn from the RM1 training set, and a word error rate (WER) is calculated. To determine which of the distance metrics is a better predictor of the algorithm's performance, the correlation coefficient of each of these distortion metrics with the experimental WER is calculated; the result can be seen in Figure 3.1. Notice that the L^2 norm of MFCC features has a roughly 0.8 correlation coefficient with WER across these various scenarios, indicating that it should be relatively predictive of WER behavior.

Thus far, we have obtained a distortion metric that is descriptive of a processing algorithm in that its value is highly correlated with the algorithm's performance. In fact, using feature combination methods such as PCA and LDA on the feature-distance combinations above, we are able to construct combined features with correlation coefficients of close to 0.90, across scenarios and even different parameterizations of PDCW.

The next step is to construct a predictive function $f(\cdot)$ such that: $WER \approx f\left(\bar{D}\right)$. One approach to this problem is to choose a parameterized family of functions and attempt to fit the function to the data. We attempted this approach with polynomials of various orders, using linear regression to fit the parameters of a function of the form $WER \approx \sum_{p=0}^{P} a_p \bar{D}^p$. Unfortunately, when considering all the scenarios mentioned above (*i.e.,* different SIRs and/or

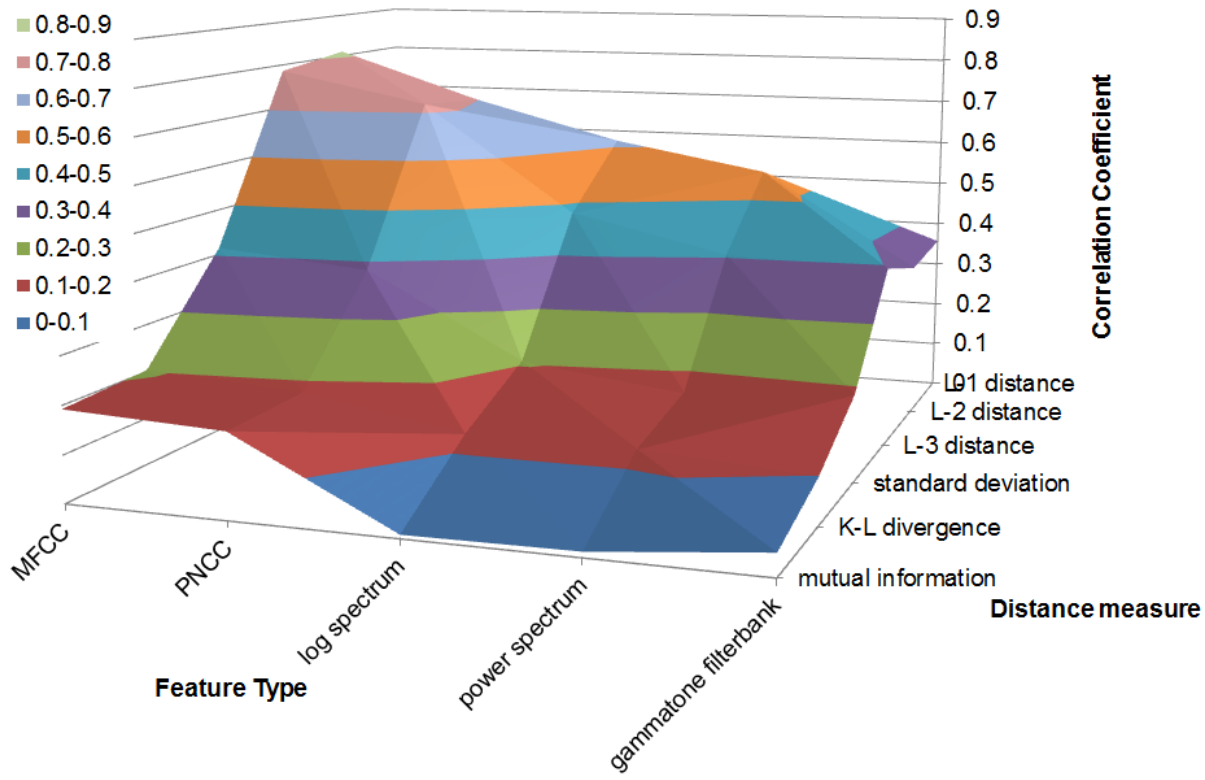


Figure 3.1: Correlation coefficient of various distortion metrics with WER

reverberation times), none of the fits are particularly close. By limiting to data from a particular scenario, the fits can be improved, but the parameters vary considerably from one scenario to the next.

3.3 Analysis: Nonlinear beam pattern

The elegance and intuitiveness of the secondary metrics used with linear beamforming leads to the question, "Can a set of equivalent metrics be developed for masking algorithms?" In particular, the behavior of beamformers are characterized relatively completely by their beam patterns; if a similar metric were to be developed for masking, it would allow us to view masking algorithms as "nonlinear beamformers".

The difficulty, of course, lies in the nonlinearity. The power of the beam pattern as a descriptor is that since all the combinations are linear, any signal from any direction is

processed independently of the existence, position or power of any other sources. This is not the case for nonlinear techniques; the effect of any T-F masking algorithm on the target signal is heavily affected by the position and power of the interferer(s). However, given a simple paradigm such as the one from Figure 2.3, the free variables can be limited to a manageable few:

1. *Frequency (ω)*: Of course, since the array geometry manifests itself differently at different frequencies (*e.g.,* the phase difference between the copies of the interferer at the left and right microphones is a function of frequency), any model will have to take that into account.

2. *Interferer azimuth (ϕ)*: A linear beam pattern describes the reponse to a signal from a source in a particular direction, regardless of how many sources there happen to be and whether each one is desirable (*i.e.,* target) or undesirable (*i.e.,* interference). In the masking paradigm, it is assumed the target is straight ahead, but the interferer can be at any location. The behavior of the algorithm will depend on this location; *e.g.,* more spatial separation leads to better signal separation by the algorithm. This should be noted as a difference between the nonlinear beam patterns described herein and convential beam patterns; the azimuth in nonlinear beam patterns corresponds to *interferer* azimuth only.

3. *Signal-to-interference ratio (SIR)*: This is the major difference between the linear and nonlinear cases. A more powerful interferer will cause more signal masking than a weaker one. In fact, this comes into play at the level of individual T-F cells; however, for simplicity's sake, we will attempt to develop an average characterization dependent only on the overall SIR.

Thus, the nonlinear beam pattern will be a function of the form $B(\omega; \phi; SIR)$. The following sections will describe options for the quantity $B(\cdot)$ to be thus mapped and the models used to calculate them. The material in this section forms the basis for [35].

3.3.1 Analytical and computational modeling of mask presence

Ignoring for a while the smoothing step in PDCW and looking at the initial binary mask, one option is to calculate the probability that a cell in a given band will be accepted by the mask; *i.e.,* $\Pr\{M[n,k]=1\}$. Since the mask on each cell is a Bernoulli variable, this is equivalent to $E\big[M[n,k]\big]$. Thus, we have defined a form of the nonlinear beam pattern:

$$B_M(\omega_k;\phi;SIR) = E\big[M[n,k]\,|\omega_k,\phi,SIR\big] \tag{3.4}$$

when calculated in the presence of an interferer at azimuth ϕ at the given nominal SIR (note that ω_k is the angular frequency corresponding to subband k). The statistical averaging is over various conditions of the signal and interference; the exact quantities will be discussed below. The result is dependent on frequency, and thus will approximate time averages of the masks in the various subbannds. We will call this quantity *mask presence*, as it shows the fraction of time (or circumstances) where the masking allows the incoming signals to pass through; *e.g.,* if $E\big[M[n,k]\big]=0.6$, it means that in a given experiment, roughly 60% of the T-F cells in subband k will be accepted, with the other 40% being masked out. This definition is also directly applicable to non-binary masks.

To calculate this, we must first isolate the sources of randomness in the mask generation. The signals $S[n,k]$ and $I[n,k]$ are random; when looking at a single cell, each will have a random amplitude and phase. Since phase references are arbitrary, we can collapse the relative phases of the signal and interference into one random variable: $\alpha = \angle S[n,k] - \angle I[n,k]$; since the signal and interference are assumed to be independent, we can reasonably assume that the phase difference is uniform: $\alpha \sim U(-\pi,\pi]$. The random amplitudes can be named S and I:

$$S[n,k] = S \quad \wedge \quad I[n,k] = Ie^{-j\alpha} \tag{3.5}$$

Since the signals are assumed to be of similar type (*e.g.,* both speech), their long-term spectral profiles will be similar; thus, the nominal SIR will also be the nominal SIR for each

subband. Specifically, this means

$$\frac{E\left[\,|S|^2\,\right]}{E\left[\,|I|^2\,\right]} = SIR$$

Now, (2.7) can be expressed in terms of the random variables of the model – remember that the phase shift will be dependent on the frequency under analysis:

$$X_1 = S + Ie^{-j\alpha} \quad \wedge \quad X_2 = S + Ie^{-j\alpha}e^{-j\omega f_S \frac{d}{c}\sin\phi} \tag{3.6}$$

Combining (3.4), (2.12), (2.10) and (3.6):

$$B_M\left(\omega_k;\phi;SIR\right) = E\big[M\left[n,k\right]\big] = \Pr\big\{M\left[n,k\right]=1\big\} = \Pr\big\{\,|\theta| < |\gamma\left(\omega_k;\phi_T\right)|\,\big\} \tag{3.7}$$

$$\begin{aligned}
\text{where:} \qquad \theta &= \angle\left(X_1X_2^*\right) = \angle\left[\left(S+Ie^{-j\alpha}\right)\left(S+Ie^{-j\alpha}e^{-j\gamma(\omega_k;\phi)}\right)^*\right] \\
&= \angle\left[\left(S+Ie^{-j\alpha}\right)\left(S+Ie^{j(\alpha+\gamma(\omega_k;\phi))}\right)\right] \\
&= \angle\left[S^2 + SIe^{j(\alpha+\gamma(\omega_k;\phi))} + SIe^{-j\alpha} + I^2e^{j\gamma(\omega_k;\phi)}\right]
\end{aligned}$$

$$\Rightarrow \quad \theta = \angle\left[S^2 + 2SI\cos\left(\alpha+\frac{1}{2}\gamma\left(\omega_k;\phi\right)\right)e^{\frac{j}{2}\gamma(\omega_k;\phi)} + I^2e^{j\gamma(\omega_k;\phi)}\right] \tag{3.8}$$

Unfortunately, we are not aware of an analytical solution to (3.8). There are three independent random variables, α, S and I, the latter two of which have irregular distributions. Solving for the distribution of θ and then for $B_M\left(\omega_k;\phi;SIR\right)$ is impossible without gross approximations.

However, this problem can be solved computationally. The first step is to build probability distributions for S and I; in this scenario, where both are assumed to be speech sources, this can be done by building signal level histograms of actual speech spectrograms. Since speech spectra are far from flat, these distributions will be frequency-dependent. Furthermore, since the signal and interference are both speech signals, the distributions will be

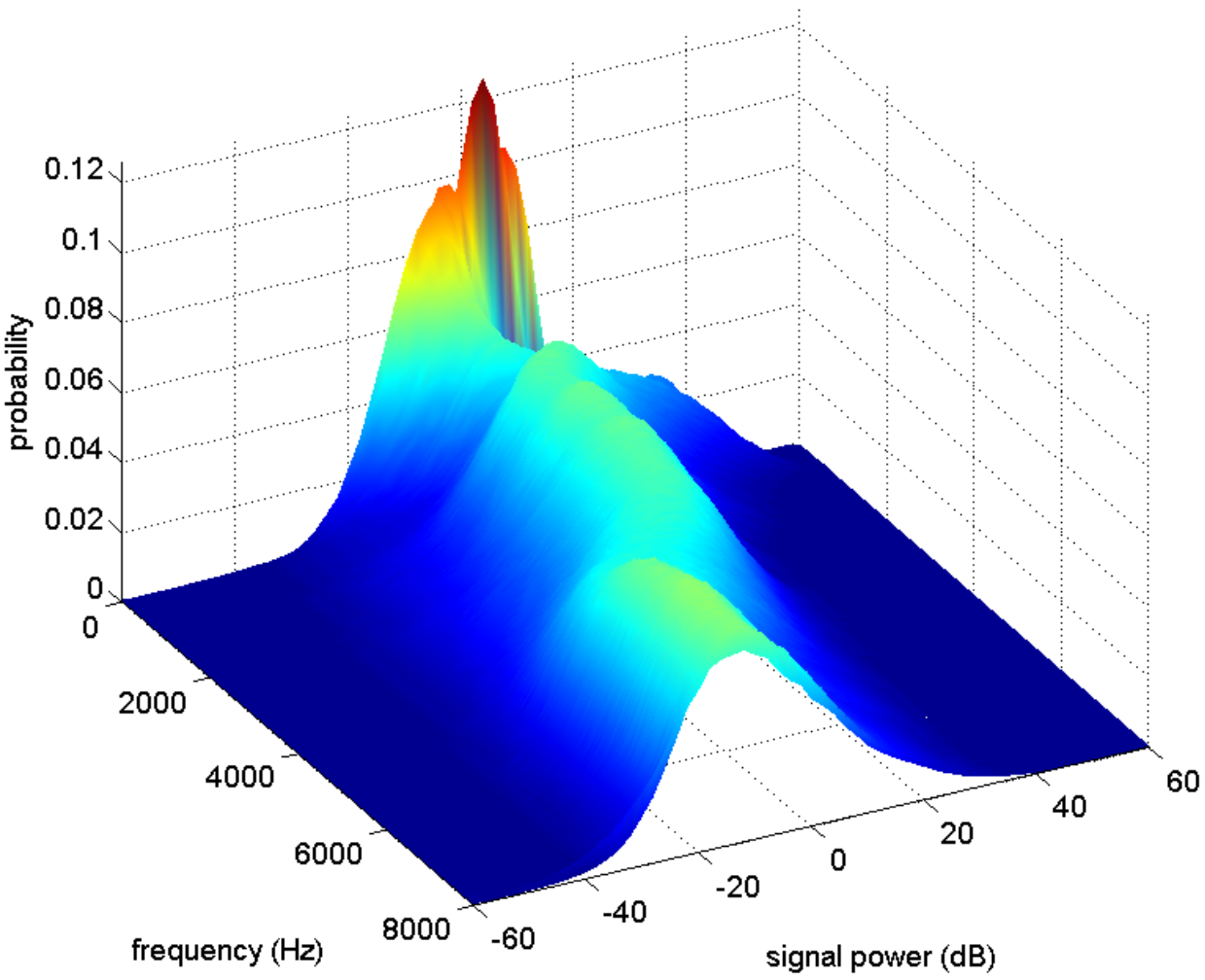


Figure 3.2: Distributions of speech subband signal levels in dB

identical, with one caveat: the interference distribution must be attenuated by the amount of the nominal SIR. To simplify this operation, we collect histograms of log-spectrograms; this way, the attentuation of the interference amounts to a simple shift, without changing the shape of the distribution. The frequency-dependent histogram of log-spectral speech is shown in Figure 3.2.

The only remaining step is to calculate the value of the mask for each possible combination of the three random variables, then average according to their probabilities; in effect, averaging the value of the mask over the joint p.d.f. of these variables. Doing this for specific values of the three free paramters (frequency, interferer azimuth and nominal SIR) will

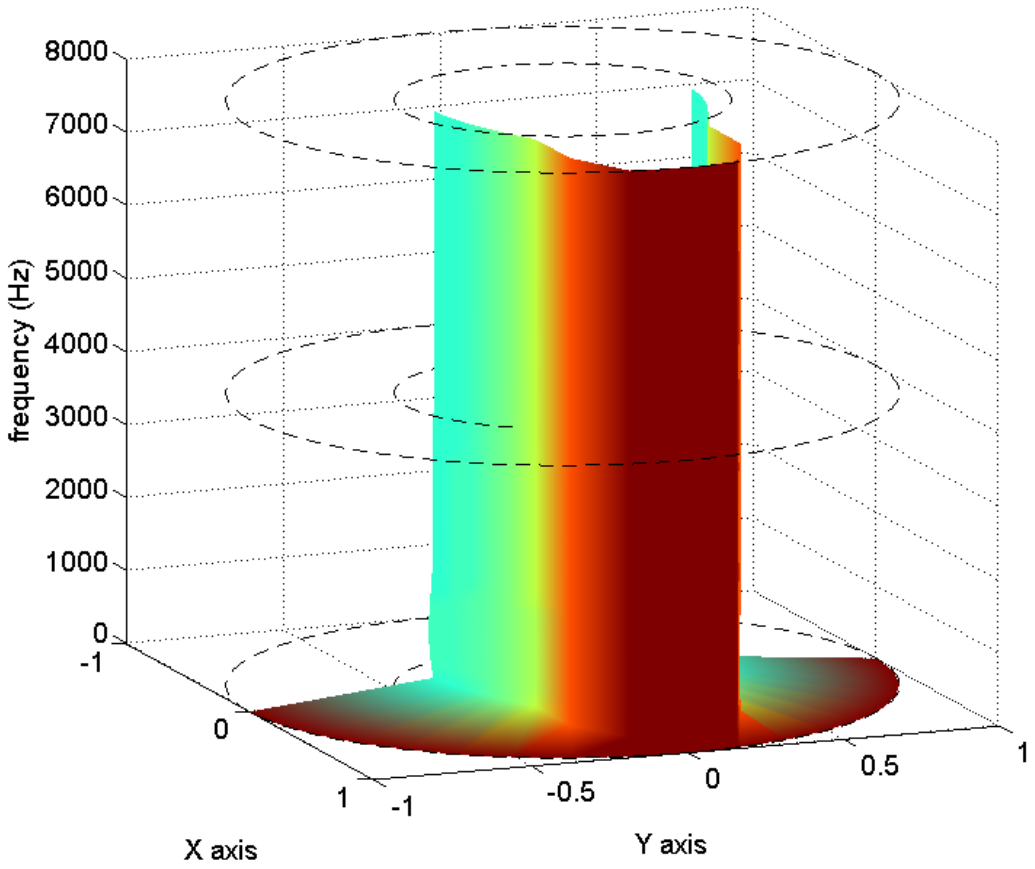


Figure 3.3: Mask presence pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR = 0 dB (angle in the X-Y plane corresponds to interferer azimuth, with the array's look direction being along positive X)

yield one value of $B_M(f;\phi;SIR)$, from which a complete "nonlinear beam pattern" can be constructed.

Figures 3.3 through 3.5 show examples of this, for a two-microphone array with elements 4 cm apart and the phase threshold set for a cone of acceptance 20° wide around the target direction. Note that in both cases, when the interferer is at $\phi = 0°$, the mask is always 1; this makes sense because when the interferer and target are in the same direction, the masker will simply accept everything and reduce to a simple receiver. As the interferer moves off to the side, when SIR = 0 dB the mask presence drops to about 0.5; this is also expected as, according to (2.9), with equal signal and interference powers about half the cells will be

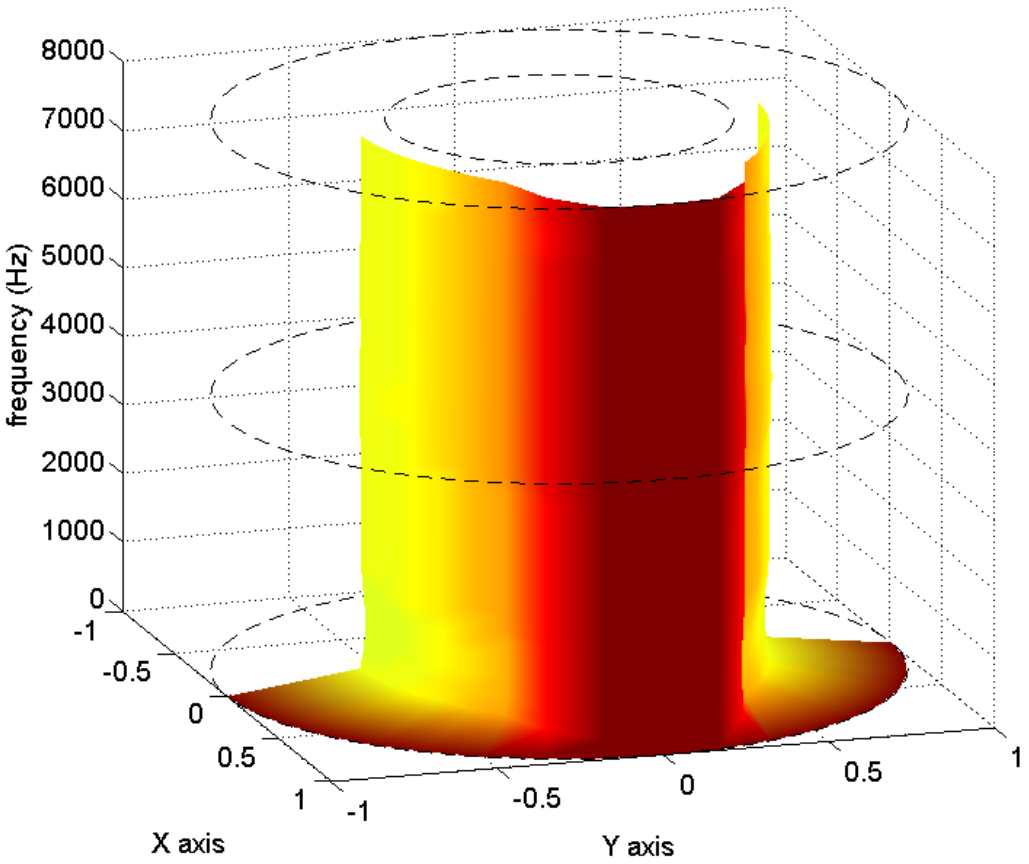


Figure 3.4: Mask presence pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR = 10 dB (angle in the X-Y plane corresponds to interferer azimuth)

accepted and half rejected. When the SIR increases to 20 dB, a higher percentage of cells are accepted as the signal overpowers the interference more frequently.

Thus far, we have developed a "beam pattern" for T-F masking algorithms. This pattern emulates a linear beam pattern in that it shows, on average, how frequency and direction affect the incoming signals. However, it falls short of describing the quality of the masker. In a masking algorithm, if a cell is masked it affects both target and interferer; the probability of this occurring is plotted in Figures 3.3 through 3.5. However, the amount of interference power masked is usually higher than the target power; indeed this is the goal of masking in the first place. The more adept the algorithm is at identifying the appropriate cells, the better it will perform. Unfortunately, the beam pattern developed above does not capture

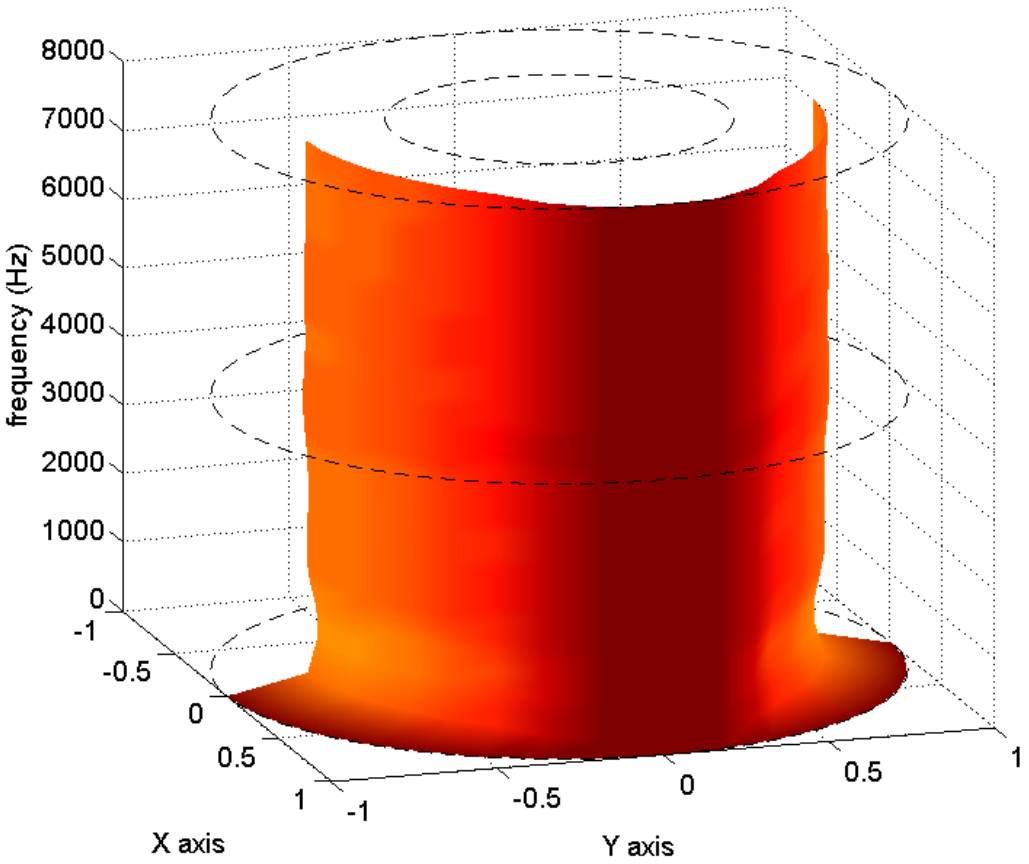


Figure 3.5: Mask presence pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR = 20 dB (angle in the X-Y plane corresponds to interferer azimuth)

this behavior. In the following section we turn our attention to an alternative metric, based on output noise, that does.

3.3.2 Output noise and SNR

In (3.2), a distinction was drawn between the two sources of distortion in the output of a masking algorithm. This can lead to a characterization of noise on the output of the processing, at a given frequency, as (substituting the signal and interference phasors introduced in (3.5))

$$N^2 = \overbrace{\left(S\left(1-M\right)\right)^2}^{\text{signal suppression}} + \overbrace{\left(IM\right)^2}^{\text{interference leakthrough}} \tag{3.9}$$

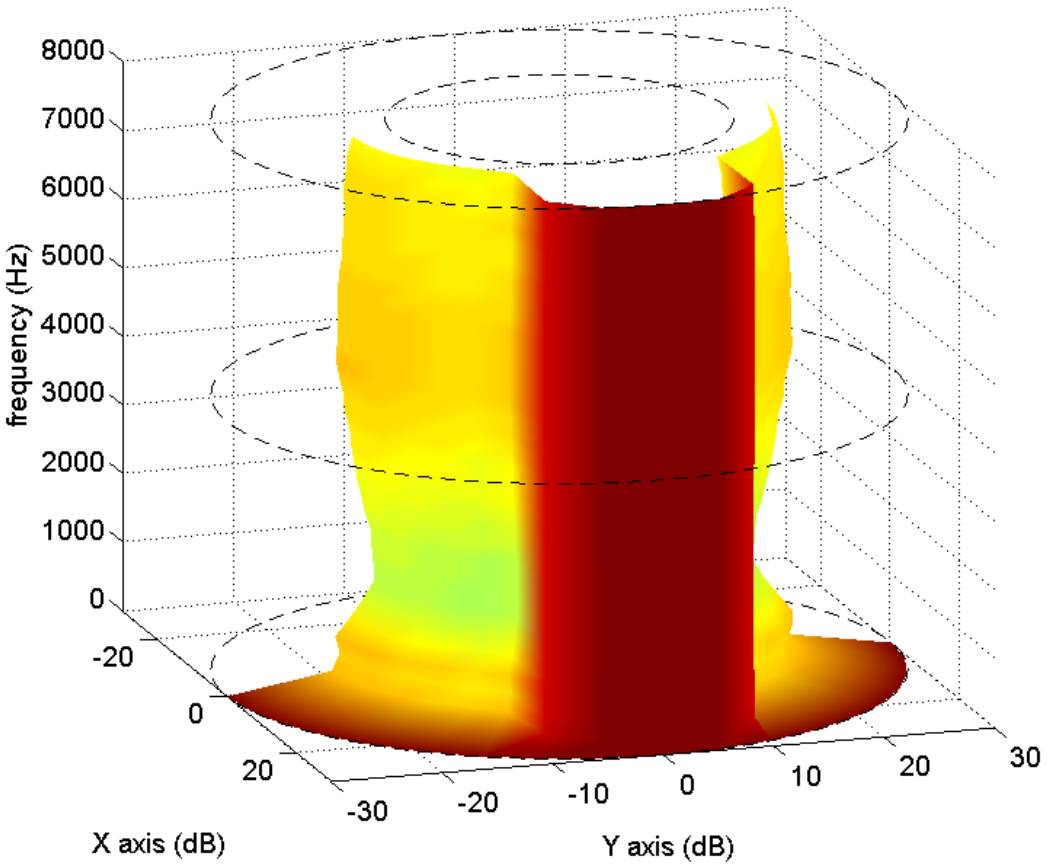


Figure 3.6: Output noise pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR $= 0$ dB (angle in the X-Y plane corresponds to interferer azimuth, with the array's look direction being along positive X)

In other words, the noise power is the sum of the signal power in rejected cells and the interference power in accepted cells. Similarly, the output signal-to-noise ratio can be expressed as

$$SNR_{out} = \frac{E\left[S^2\right]}{E\left[N^2\right]} = \frac{E\left[S^2\right]}{E\left[\left(S\left(1-M\right)\right)^2\right] + E\left[\left(IM\right)^2\right]} \tag{3.10}$$

Of course, this is not technically correct; when defining SNR metrics in this fashion, the signal and noise components should be independent of each other. In this case, not only is one of the noise terms a direct function of the signal, but the mask in both terms is a function of both signal and interference, which makes the definition somewhat messy. However, it is useful as a rough estimate of how corrupted the output signal is compared to the input

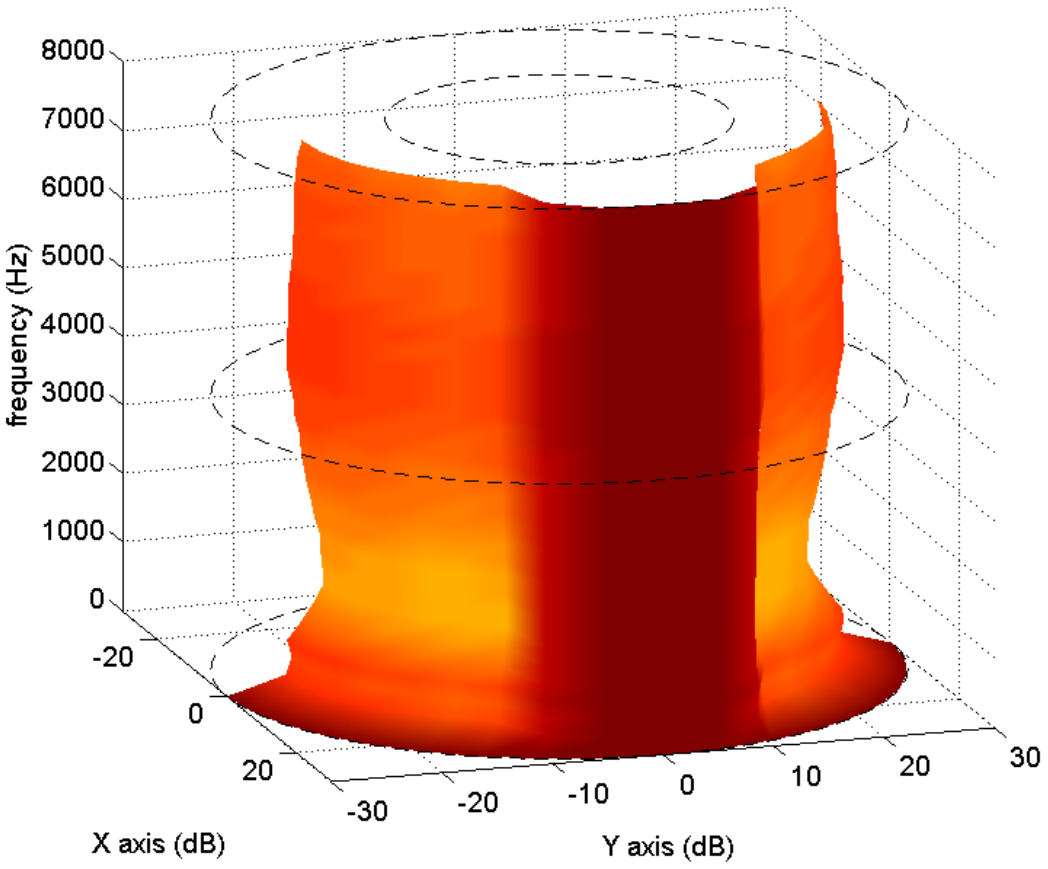


Figure 3.7: Output noise pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR = 10 dB (angle in the X-Y plane corresponds to interferer azimuth)

signal.

Now, this output noise can be used to construct a variant of the nonlinear beam pattern: $B_N(\omega_k; \phi; SIR) = E\left[N^2 | \omega_k, \phi, SIR\right]$. In the computational model described in Section 3.3.1, the mask for each instance of the random triplet (α, S, I) has already been calculated; the calculation of the average noise from that information is trivial. This can then be averaged over the joint distribution to construct this new type of beam pattern.

Figures 3.6 through 3.8 show the output noise pattern of the array whose mask presence pattern was shown in Figures 3.3 through 3.5. The noise levels in these plots are normalized; the values at $\phi = 0°$, where all the interference is passed through, can be considered the reference. When SIR = 0 dB, as the interferer moves off to the side, the output noise level

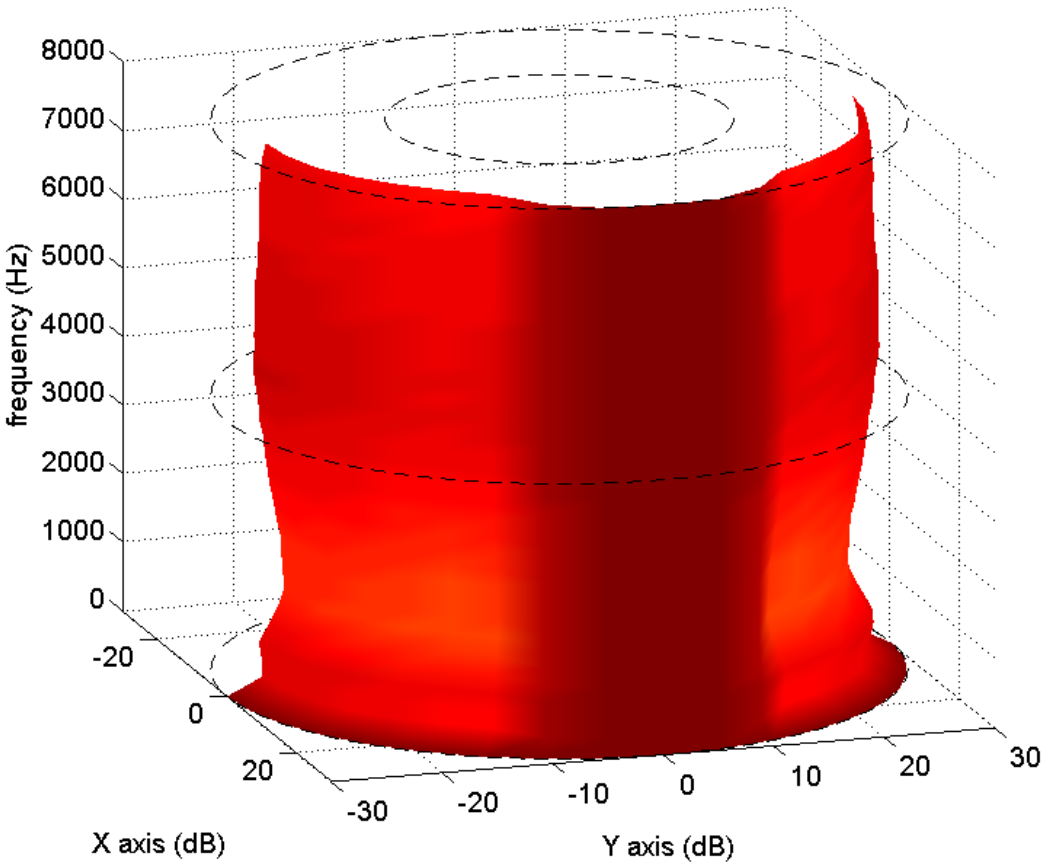


Figure 3.8: Output noise pattern for two-microphone array, $d = 4$ cm, $\phi_T = 20°$, SIR = 20 dB (angle in the X-Y plane corresponds to interferer azimuth)

drops by about 10 dB (depending on the frequency in question); this is equivalent to a processing gain of 10 dB achieved by the algorithm. This gain is lower when the SIR is higher, as there is less interference to suppress in the first place. Also note the consistency across frequency and azimuth. This would help explain why, with two sensors, masking outperforms beamforming (see Figure 4.10); with only two sensors, beamforming is unable to achieve the necessary consistency.

3.3.3 Comparisons with linear beam patterns

Figure 3.9 depicts the beam pattern of a delay-and-sum beamformer with the same array used to obtain Figures 3.6 through 3.8. With a target signal in the direction of the main

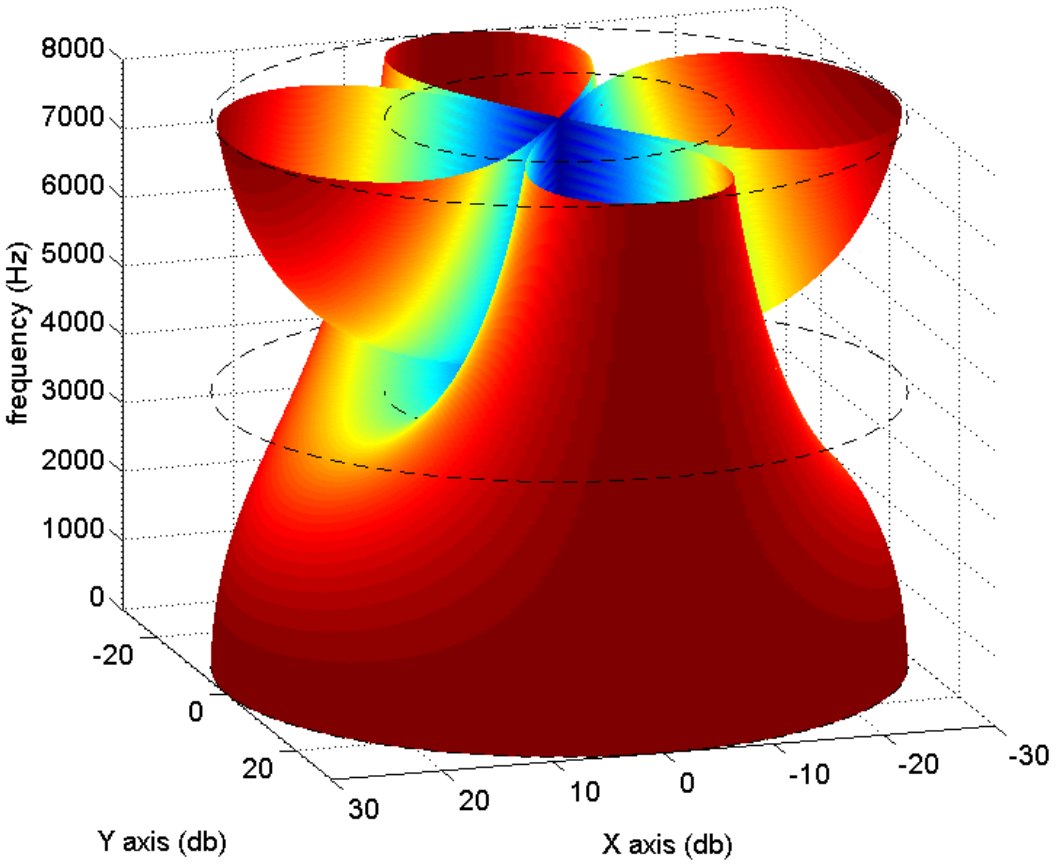


Figure 3.9: Beam pattern of two-element delay-and-sum beamformer

lobe and an interferer at various azimuth angles, the beam pattern at any angle will be the amount of interferer power if the interferer is at that same angle, normalized by the signal power. This is equivalent to the nonlinear beam patterns of Figures 3.6 through 3.8, except:

1. The nonlinear beam patterns are based on the power of the output noise metric rather than the power of the interference signal at the output.

2. The nonlinear beam patterns are a function of input SIR.

Even given these differences, it is clear that interference suppression of the nonlinear T-F masker is much more consistent than the two-element linear beamformer, across both azimuth and frequency.

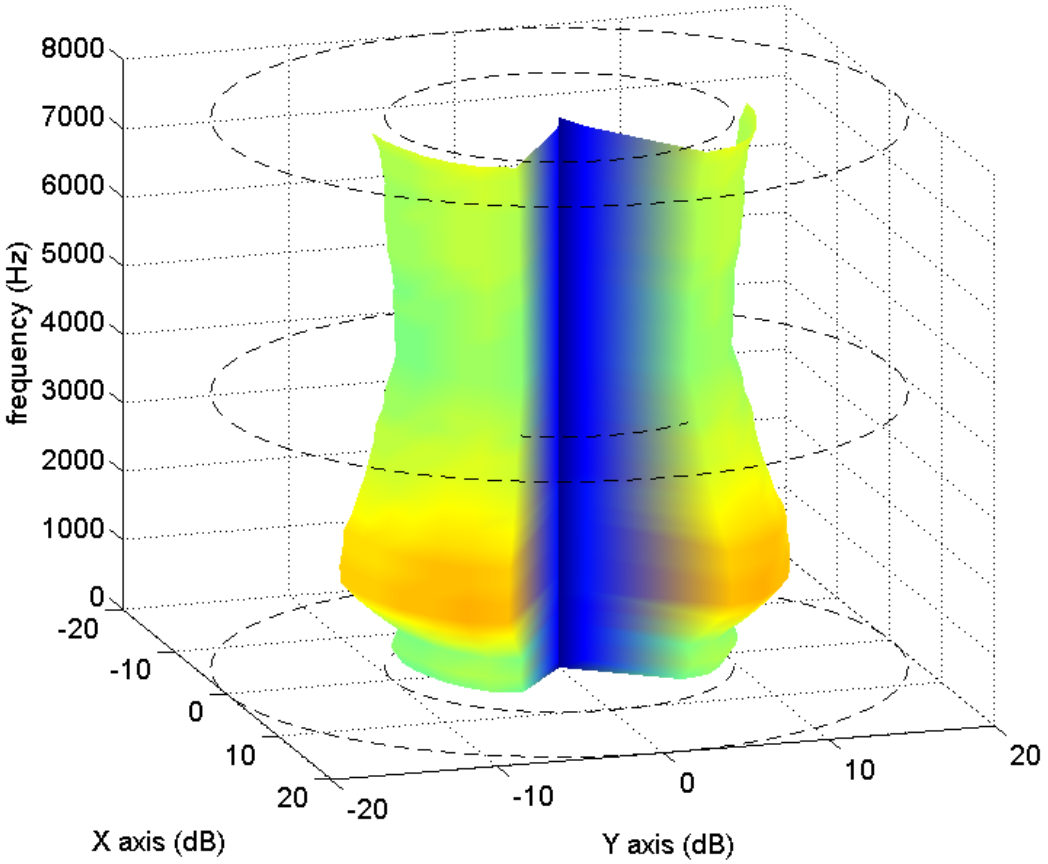


Figure 3.10: Equivalent processing gain pattern of T-F masking scenario depicted in Figure 3.6 (angle in the X-Y plane corresponds to interferer azimuth)

To observe the same phenomenon from an alternative viewpoint, we can compare the processing gain of the masker and the beamformer. Figure 3.10 illustrates the processing gain of the masker whose output noise pattern is depicted in Figure 3.6; Figure 3.11 illustrates the processing gain of the beamformer depicted in Figure 3.9. Here, processing gain is defined as the improvement in SNR from the input to the output of the system. For a given interferer direction, the linear beamformer can improve quality greatly over a small range of frequencies, while the nonlinear T-F masker suppresses the interferer to a lesser degree but consistently across all frequencies.

Incidentally, the same observation can be used to visualize the scaling of linear beam patterns. Compare Figure 3.9, the beam pattern of a two-element beamformer, with Fig-

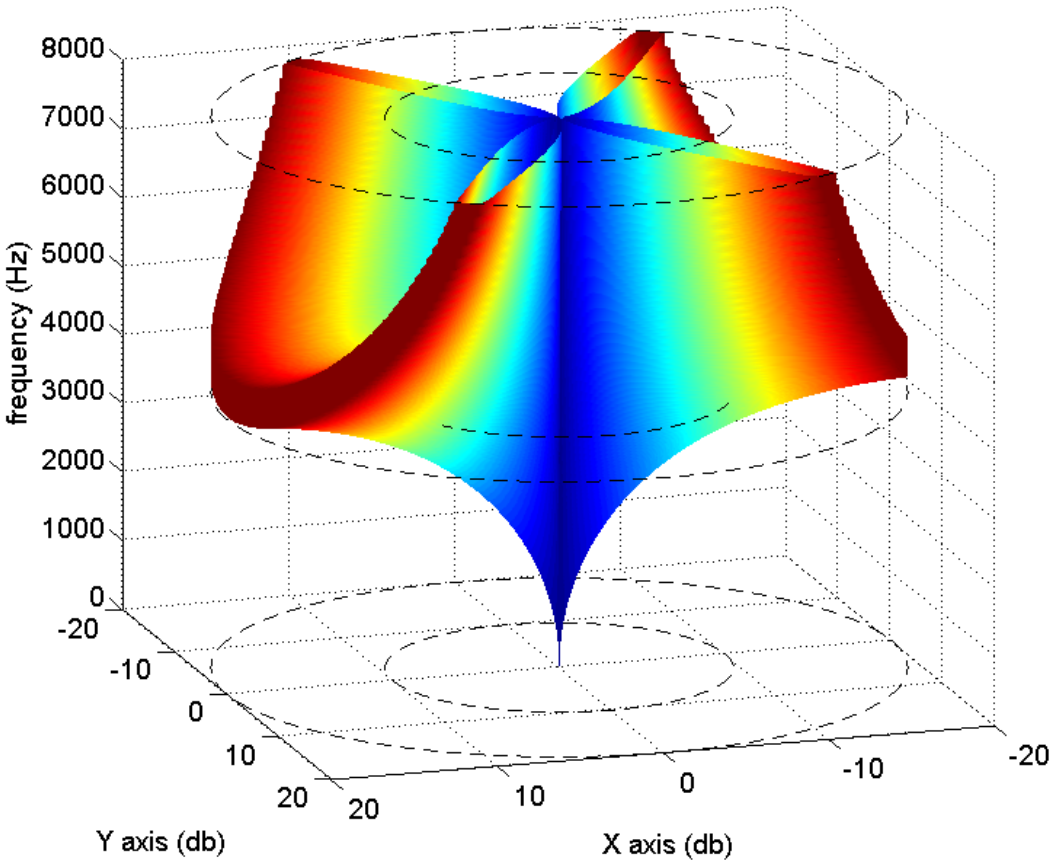


Figure 3.11: Equivalent processing gain pattern of beam pattern depicted in Figure 3.9 (angle in the X-Y plane corresponds to interferer azimuth)

ure 2.4, the beam pattern of a four-element beamformer with the same element spacing. An interfering source off at an angle will be suppressed far more consistently by the pattern of Figure 2.4 than by that of Figure 3.9; in other words, a four-element delay-and-sum outperforms a two-element delay-and-sum beamformer. This behavior of linear beamformers will be discussed in Chapter 4, forming the goal of the work detailed in Section 4.3.

3.3.4 Verification of the model

Interestingly, even though this definition of SNR is flawed, the values calculated seem to correspond reasonably well to what one would expect from such a metric. To support this claim, a series of parallel experiments was run to confirm that the general concept of SNR

and processing gain for nonlinear T-F masks is valid.

In one set of experiments, the physical configuration of Figure 2.3 is simulated with the interferer at $\phi = 60°$ at various SIR levels. Two variations of PDCW are used to process the received signals: one with the binary masks, with no frequency smoothing, and one with smoothed masks (for convenience, we will refer to the former as PD and the latter as PDCW). The masker's output is then passed to a speech recognizer.

In the second set of experiments, the output noise patterns $B_N(\omega_k; \phi; SIR)$ from the model (such as those of Figures 3.6 through 3.8) for $\phi = 60°$ and those same SIR levels are calculated; at each SIR, this will just be a function of frequency. This function is applied like a filter's frequency response to the interfering signal, which is then just added to the target signal. This "artificially degraded" signal is then passed to the same recognizer with no further processing. Mathematically, the synthesis procedure of the second set of experiments can be described as

$$Y[n,k] = S[n,k] + I[n,k] \cdot \frac{1}{SIR} \cdot B(\omega_k; \phi = 60°; SIR) \tag{3.11}$$

where $S[n,k]$ and $I[n,k]$ are the STFTs of the (clean) target and interfering utterances, respectively, and SIR and the pattern B_N are assumed to be in amplitude scale rather than decibels. $Y[n,k]$ is passed directly to the recognizer. Thus, the SIR of the signals used in the second set of experiments has the same spectral profile as our model's prediction of the output SNR of the masking algorithm used in the first set.

In both sets of experiments, the target and interfering signals are drawn from the DARPA Resource Management (RM1) 2.0 test database, consisting of 600 utterances. For each target utterance, a different utterance is randomly selected from the test set to function as the interfering signal. The same target/interfering utterance pairing is maintained across all experiments; *i.e.,* a given target utterance is paired with the same interfering utterance in all scenarios across all the sets of experiments, but (as stated) that pairing is randomly

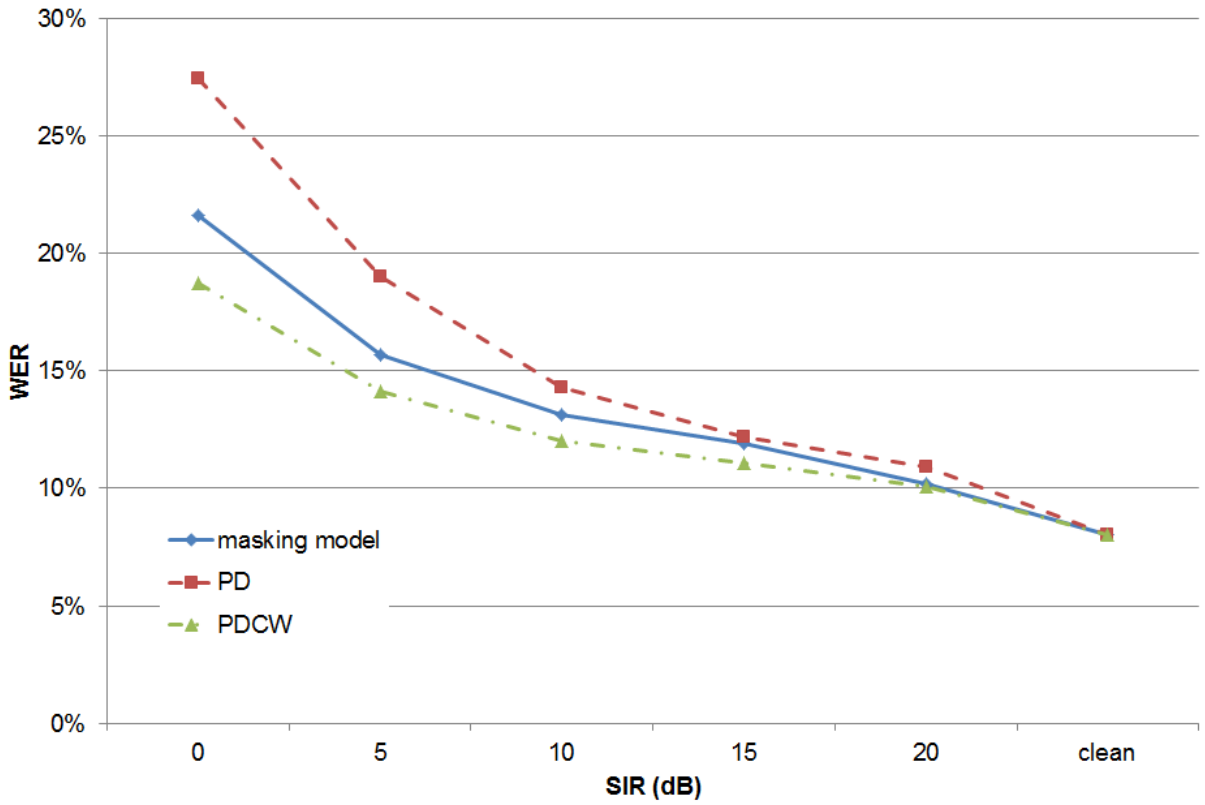


Figure 3.12: Word error rates (WER) of masked speech vs. speech with interferer with spectral profile predicted by output noise model

determined.

All the speech recognition experiments appearing in this thesis also share the following details, unless explicitly noted otherwise: The speech recognizer is CMU Sphinx-3, with its acoustic models trained using CMU SphinxTrain on clean speech from the DARPA RM1 speaker-independent training set, consisting of 1600 utterances. The acoustic models are 3-state hidden Markov models (HMMs), with output probabilities modeled by Gaussian mixture models (GMMs) with 8 Gaussians per mixture. State tying is used to reduce the training parameter space, resulting in 1000 total tied states (senones). The feature extraction is conventional MFCC processing as implemented in CMU SphinxBase, using the standard 13 MFCCs and the delta and delta-delta coefficients, resulting in 39-dimensional feature vectors.

The results of these experiments are shown in Figure 3.12; as can be seen, the model-

predicted WER tracks actual performance quite well. Although the model currently only considers binary masks, it is a bit optimistic in predicting the behavior of a binary masker. This is probably due to the fact that the model is predicting the average values for output noise, which – as can be seen in Figures 3.6 through 3.8 – tend to be quite smooth across frequency, while the actual masks produced by PD, being binary, have sharp local transitions that hurt recognition accuracy. On the other hand, it seems that smoothing the masks makes up for much of that degradation, to the point that the model predicts the performance of PDCW very closely.

3.4 Extensions of the nonlinear beam pattern

While the nonlinear beam patterns developed in Section 3.3 are both intuitive and fairly predictive, they are based on a fairly limited model of the behavior of T-F masking algorithms. Specifically, the analysis leading to the patterns presented above is based on the assumption of a two-element array receiving signals from a single target source and single interferer, both producing speech signals, in a non-reverberant environment with no other sources of noise (*e.g.,* additive noise at the sensors). The following sections demonstrate how some of these assumptions can be modified or relaxed, leading to extensions of the model to a wider variety of scenarios.

3.4.1 Different signal types

The first, and perhaps simplest, extension is to allow the model to work with signals other than speech. To achieve this, it is enough to modify the distributions of the random variables representing signal levels in the model, namely, S and I. For example, to model a single speech source in the presence of a single music source, we use the subband distributions illustrated in Figure 3.2 for S and the corresponding distributions of music for I, with the remainder of the model unchanged. It is worth noting that with mixed signal types, the

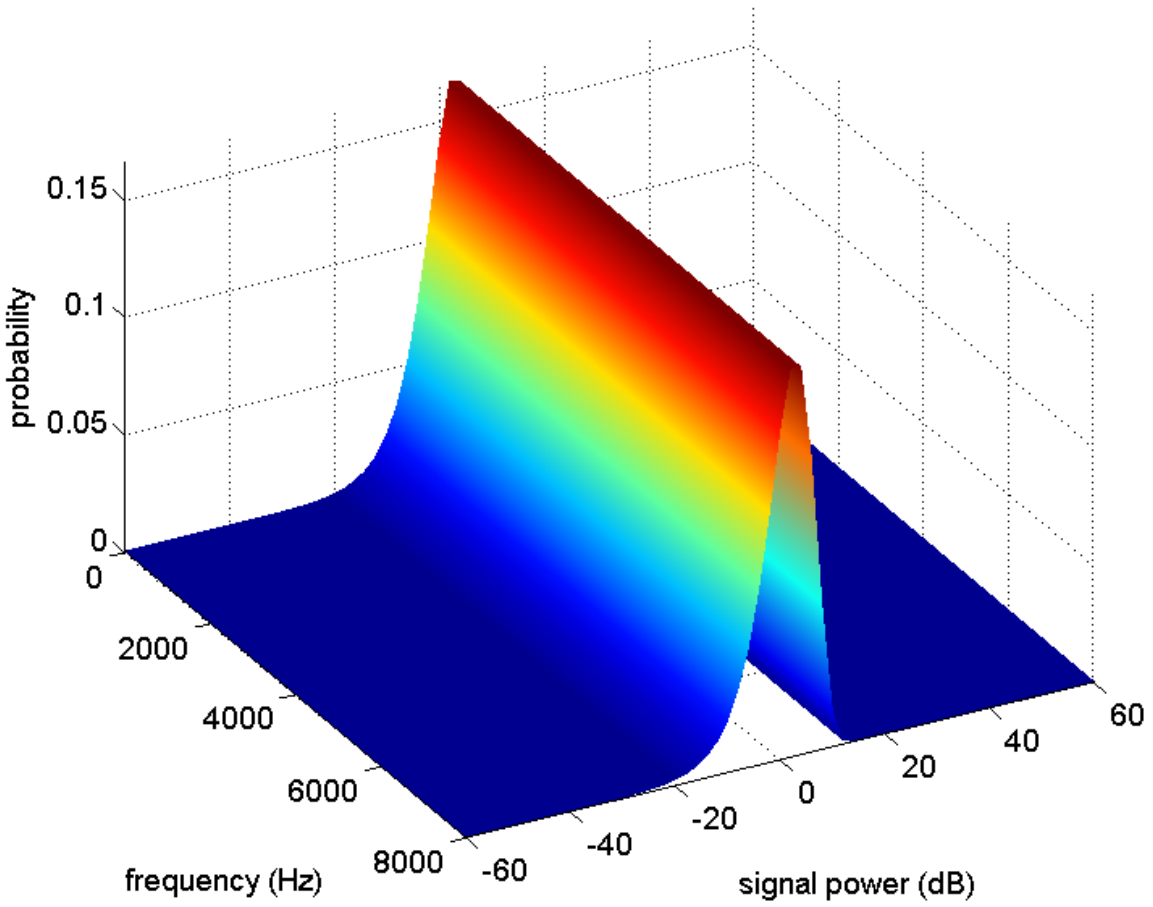


Figure 3.13: Distributions of white noise subband signal levels in dB

nominal SIR per subband will not necessarily be the same as the overall nominal SIR, as the signals will have different spectral shapes. This will not be an issue as long as the normalization of the distributions is done so that their overall average is kept equal (*i.e.,* at unity, for simplicity). Mathematically, if $f_{|X|}(x;\omega_k)$ denotes the probability density function of the signal magnitude in subband k, the functions must be normalized, using the same factor across all frequencies, so that

$$\left\langle \int x^2 f_{|X|}(x;\omega_k)\,\mathrm{d}x \right\rangle_k = 1 \tag{3.12}$$

where $\langle\cdot\rangle_k$ denotes averaging over k.

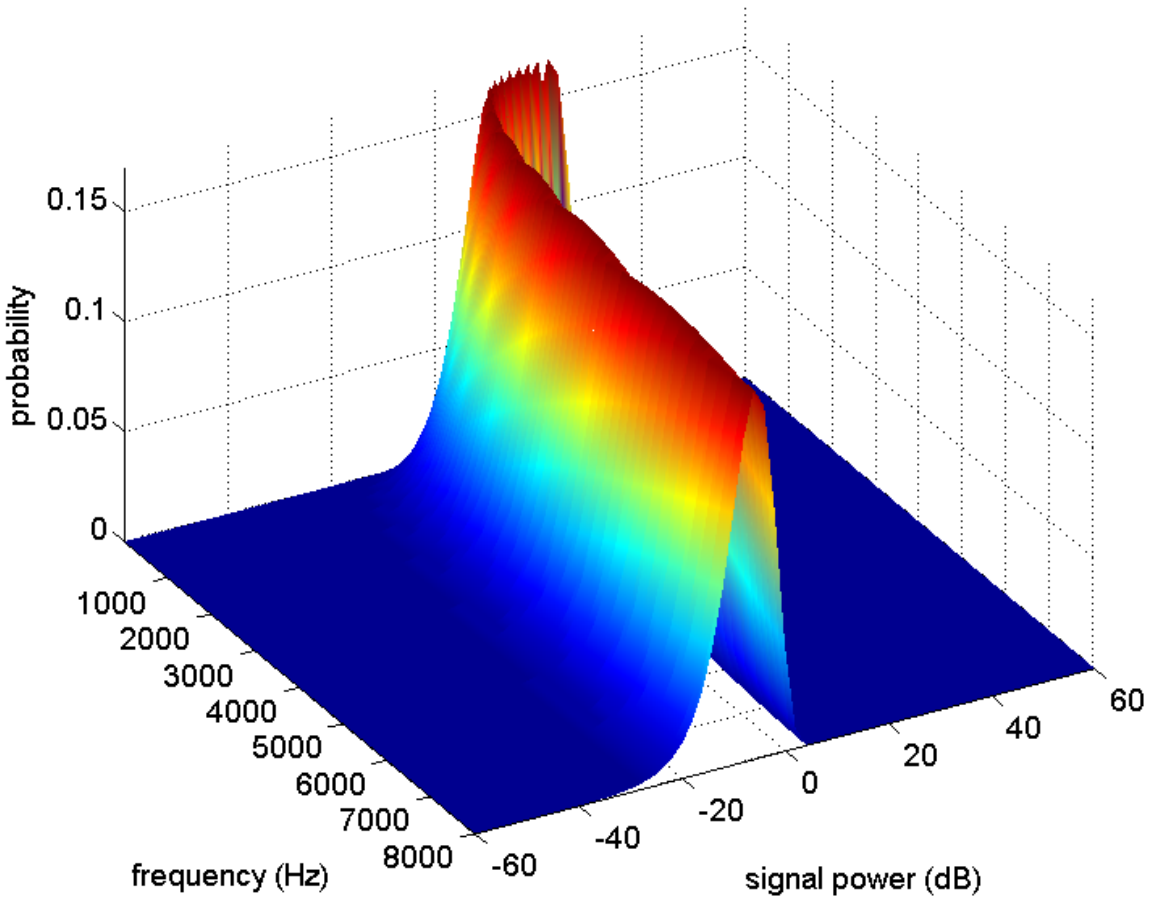


Figure 3.14: Distributions of pink noise subband signal levels in dB

To calculate the distributions for other noise sources, we can follow the same procedure outlined in Section 3.3.1 with a large dataset of that type of signal (*e.g.,* music). For simpler signal types, however, the distributions are analytically obtainable. For example, if a Gaussian white noise signal $x[n]$ is processed by STFT using a window $w[n]$:

$$\begin{aligned} X[n,k] &= \left(x[n]\, e^{-j\omega_k n}\right) * w[n] \\ &= \underbrace{\left(x[n]\cos(\omega_k n)\right) * w[n]}_{X_I[n,k]} + j\underbrace{\left(x[n]\sin(-\omega_k n)\right) * w[n]}_{X_Q[n,k]} \end{aligned} \tag{3.13}$$

where $*$ denotes convolution. Thus, the value of each cell of the STFT will be a complex

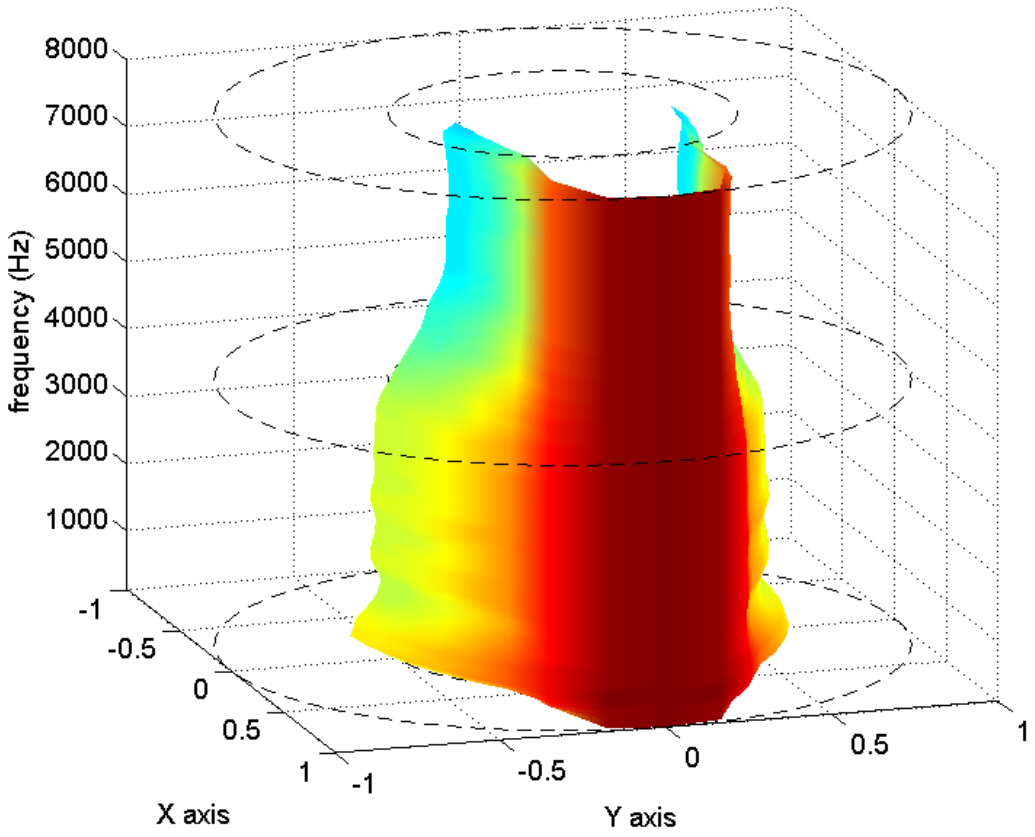


Figure 3.15: Mask presence pattern for two-microphone array with an interfering pink noise signal, $d = 4$ cm, $\phi_T = 20°$, SIR = 0 dB (angle in the X-Y plane corresponds to interferer azimuth)

random variable, whose real and imaginary parts (X_I and X_Q) are each Gaussian random variables, as each is a linear combination of Gaussians (*i.e.,* the samples of $x[n]$), and uncorrelated with each other, due to the fact that the in-phase and quadrature components of the complex exponential (*i.e.,* the $\cos(\cdot)$ and $\sin(\cdot)$) are orthogonal. It is also easy to show that they are both zero-mean and of equal variance; let us name this variance σ^2. We are interested in the distribution of the magnitude of X:

$$|X| = \sqrt{{X_I}^2 + {X_Q}^2}$$

Fortunately, it is well known that the magnitude of such a complex random variable follows

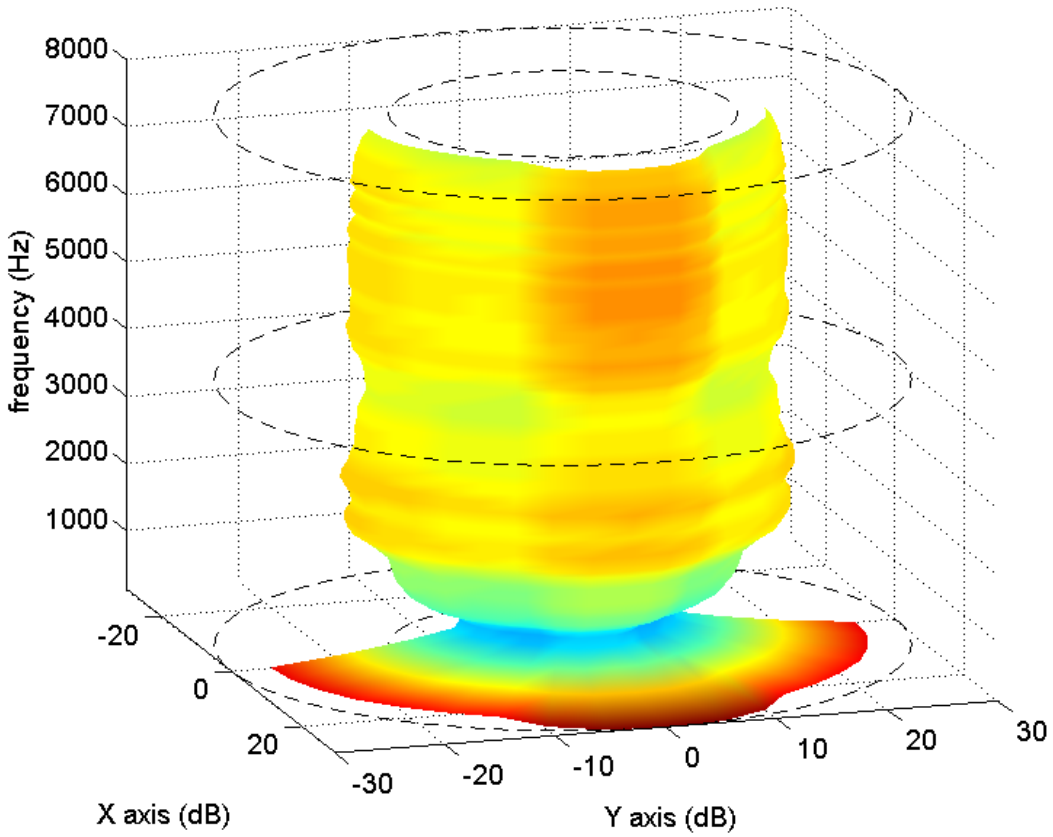


Figure 3.16: Output noise pattern for two-microphone array with an interfering pink noise signal, $d = 4$ cm, $\phi_T = 20°$, SIR = 0 dB (angle in the X-Y plane corresponds to interferer azimuth)

the Rayleigh distribution with parameter σ [36]:

$$f_{|X|}(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \tag{3.14}$$

For reasons discussed in Section 3.3, for this model it is preferable to express the signal level in dB; the distribution can be modified accordingly:

$$\begin{aligned} Y = 20\log_{10}|X| \quad \Rightarrow \quad f_Y(y) &= \frac{\ln 10}{20\sigma^2} 10^{y/10} e^{-\frac{10^{y/10}}{2\sigma^2}} \\ &= \frac{\ln 10}{20\sigma^2} x^2 e^{-\frac{x^2}{2\sigma^2}} \quad , \quad \text{where } x = 10^{y/20} \end{aligned} \tag{3.15}$$

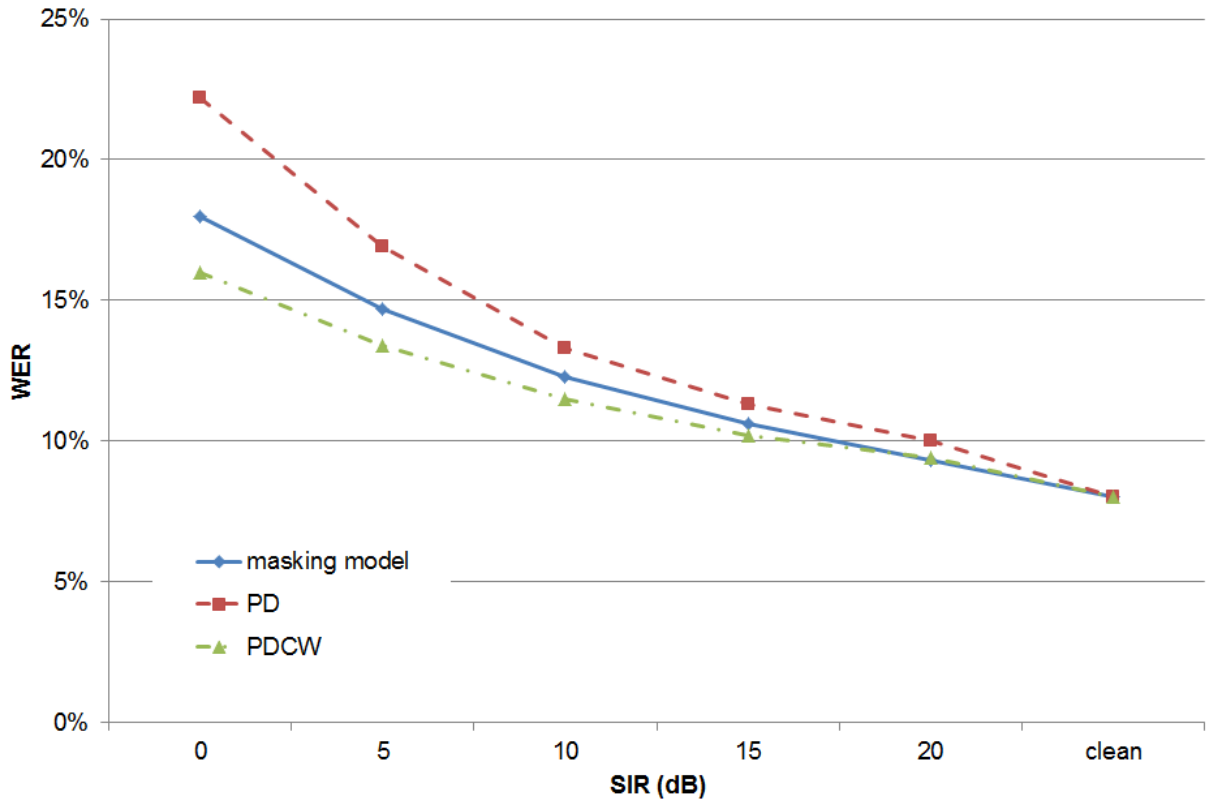


Figure 3.17: Word error rates (WER) of masked speech in the presence of pink noise vs. speech with Gaussian noise interferer with spectral profile predicted by output noise model

The average power of X, $E\left[|X|^2\right]$, is equal to σ^2. Since the model will set the input SIR to a specific value, the distributions of any signal type can be normalized to have an average power of 1 (as is the case with Figure 3.2); setting $\sigma = 1$ in (3.15) achieves this. This is true for all subbands, as white noise is, by definition, spectrally flat. Figure 3.13 illustrates the distribution of subband signal levels for white noise obtained in this fashion.

Similarly, pink noise signal level distributions can be analytically obtained. The power of pink noise is inversely proportional to frequency; thus, the σ paramter must now be a function of frequency such that

$$\sigma^2(\omega) \propto \frac{1}{\omega} \tag{3.16}$$

During normalization, care must be taken that the average power across all subbands be kept at unity, according to (3.12). The resulting distributions are illustrated in Figure 3.14.

Figures 3.15 and 3.16 illustrate the result of using the pink noise distributions obtained above to model the behavior of PDCW when faced with a target speech signal and interfering pink noise signal at 0 dB SIR. The setup is otherwise identical to that of Figures 3.3 and 3.6. The figures show that the masking algorithm loses some of its consistency across frequency when dealing with signals with different spectral profiles.

The noise model can be verified using a procedure identical to that of Section 3.3.4, except that the interfering signal $I[n, k]$ is now a unit-variance pink noise signal, rather than a speech utterance. As before, a given target utterance is paired with the same randomly-generated pink noise sample function across all experiments. The results, illustrated in Figure 3.17, demonstrate that the predictive accuracy of the model is not harmed by the use of the new signal type.

3.4.2 Independent sensor noise

Input noise at the array's sensors have, thus far, not been considered in the nonlinear beam pattern model. This noise is often modeled as additive, Gaussian and independent of the incoming signals. Spectrally, they can be of many shapes but white and pink noise are very common models. Another frequent assumption is that the noise present at each sensor is independent of the others; this is the basis for many beamforming, filtering and other techniques in the field of noise reduction. To model the behavior of masking in such scenarios, we can modify (3.6) to accommodate the sensor noises. Each sensor noise will be a complex random variable with its own amplitude and phase:

$$X_1 = S + N_1 e^{-j\alpha_1} \quad \wedge \quad X_2 = S + N_2 e^{-j\alpha_2} \tag{3.17}$$

The random variables N_1 and N_2 are given distributions based on their particular noise type; *e.g.,* that of Figure 3.13 or 3.14 for white or pink Gaussian noise. Since independent sensor noise is also frequently assumed to be equally powerful across all sensors, the same

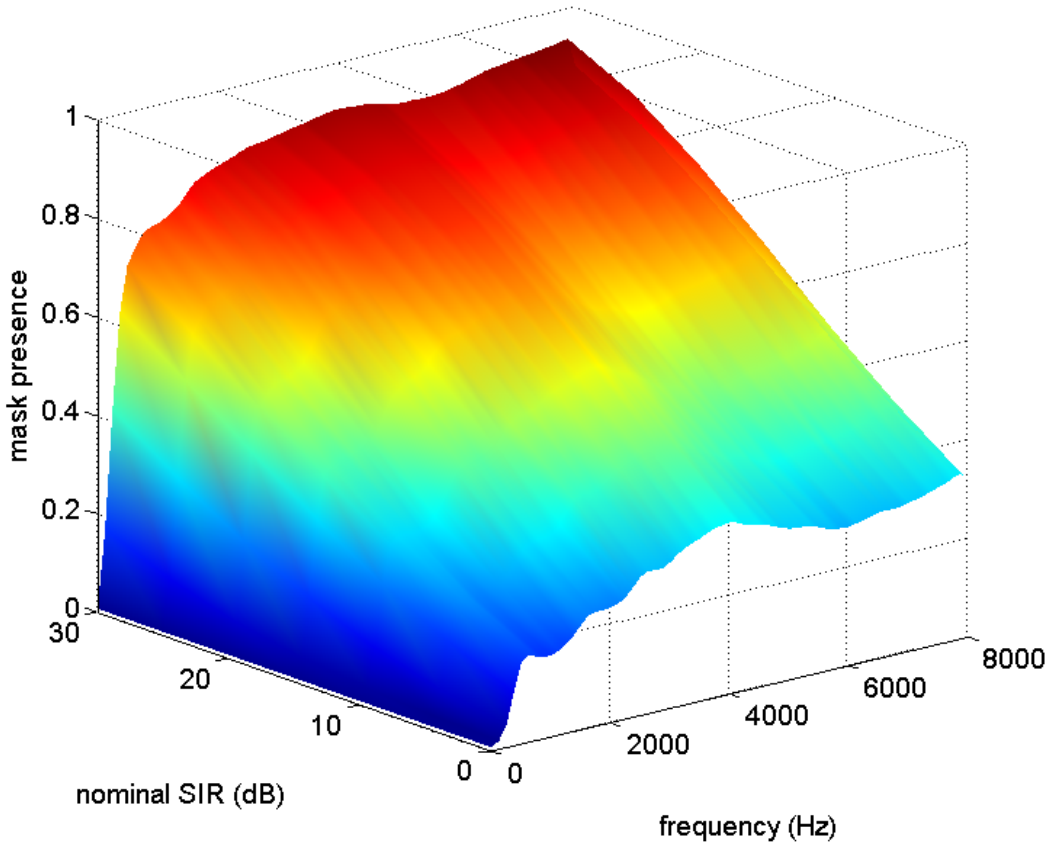


Figure 3.18: Mask presence pattern for two-microphone array with additive pink Gaussian noise at sensors, $d = 4$ cm, $\phi_T = 20°$

distribution can be used for both variables (but not the same value, of course – they must be drawn independently).

The resulting beam patterns (either mask presence or noise) will have one fewer free parameter, as the concept of interferer location does not apply to this setup, resulting in functions of the form $B(\omega_k; SIR)$, where SIR is the overall nominal input SIR for the target signal compared to either of the noises. Figures 3.18 and 3.19 illustrate the mask presence and output noise patterns corresponding to a target speech signal and additive pink Gaussian sensor noise at various nominal input SIRs. The consistency of masking across frequency appears again in these graphs: The mask presence patterns are not flat across frequency, as the input signal and noise have different spectral shapes, but the resulting output noise

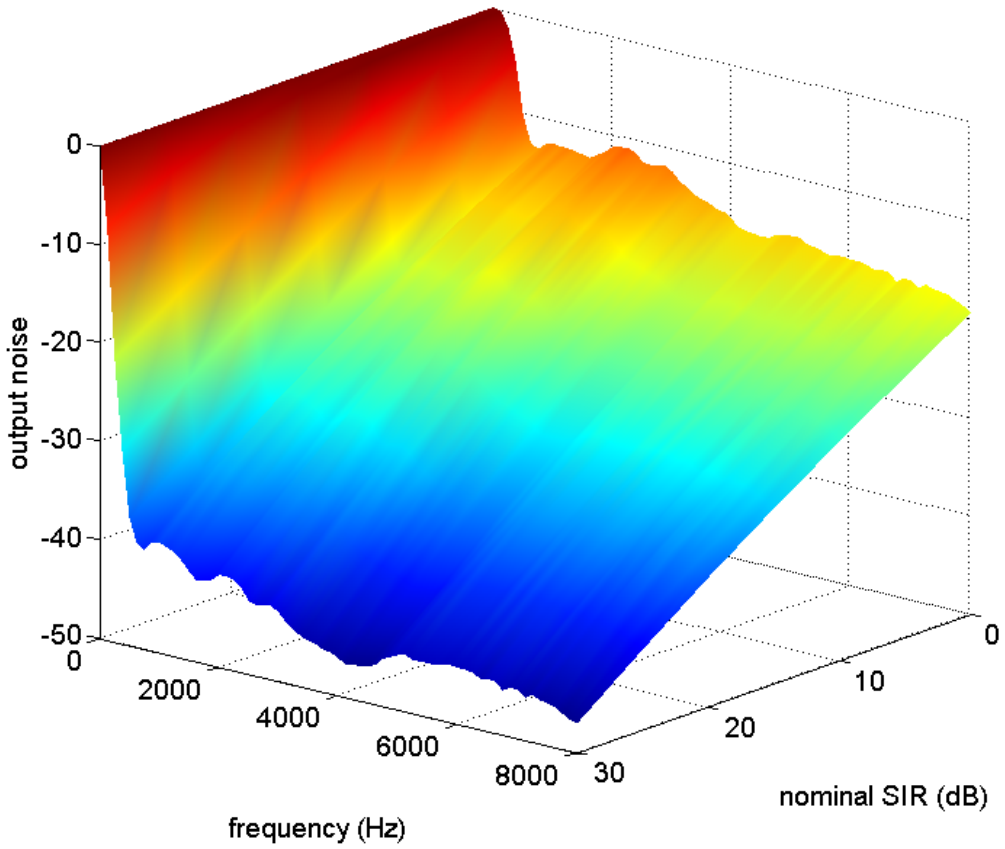


Figure 3.19: Output noise pattern for two-microphone array with additive pink Gaussian noise at sensors, $d = 4$ cm, $\phi_T = 20°$

is quite consistently low (except at $f = 0$, where there is no phase information anyway and array processing is useless). The algorithm exhibits a 15-20 dB gain across different frequencies and input SIRs, far better than a two-element linear beamformer (compare to Figure 3.9 or Figure 3.11).

As before, the noise model is verified using a procedure identical to that of Section 3.3.4, using pink noise signals as the "interfering utterances". The results, illustrated in Figure 3.20, demonstrate that the predictive accuracy of the model of modeling sensor noise is simliar to that of modeling a spatially discrete interfering source.

Of course, the sensor noise model and discrete interferer can be used in tandem. Com-

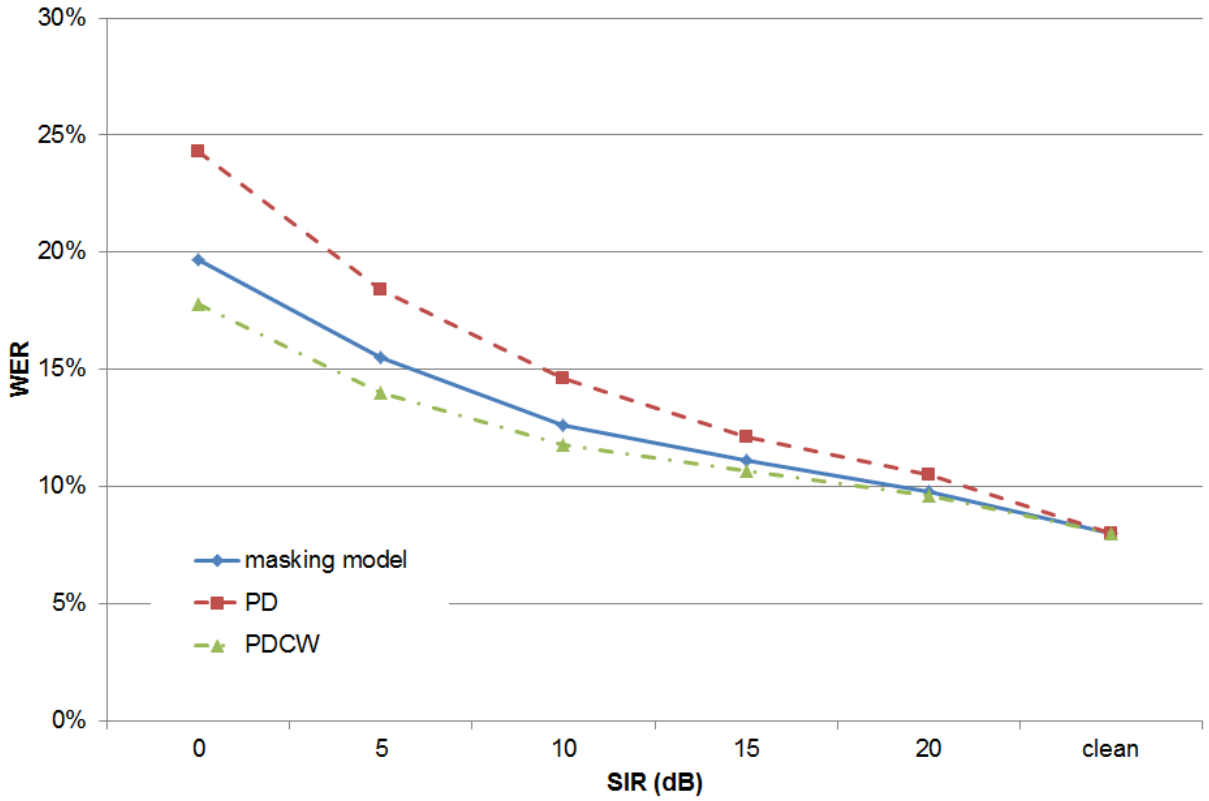


Figure 3.20: Word error rates (WER) of masked speech in the presence of pink noise at sensors vs. speech with Gaussian noise interferer with spectral profile predicted by output noise model

bining (3.6) and (3.17) produces

$$\begin{cases} X_1 = S + N_1 e^{-j\alpha_1} + I e^{-j\alpha} \\ X_2 = S + N_2 e^{-j\alpha_2} + I e^{-j\alpha} e^{-j\omega f_S \frac{d}{c} \sin\phi} \end{cases} \tag{3.18}$$

which contains seven independent random variables (S, I, α, N_1, α_1, N_2 and α_2) over which the pattern quantities (*i.e.,* mask presence or output noise) must be averaged. The resulting beam pattern will have four free variables: $B(\omega_k; \phi; SIR_{\text{interferer}}, SIR_{\text{sensor noise}})$

3.4.3 Multiple interferering sources

Another limitation of the model as presented in Section 3.3 is its assumption of a single target speaker and single interfering source. Many applications will call for T-F masking in

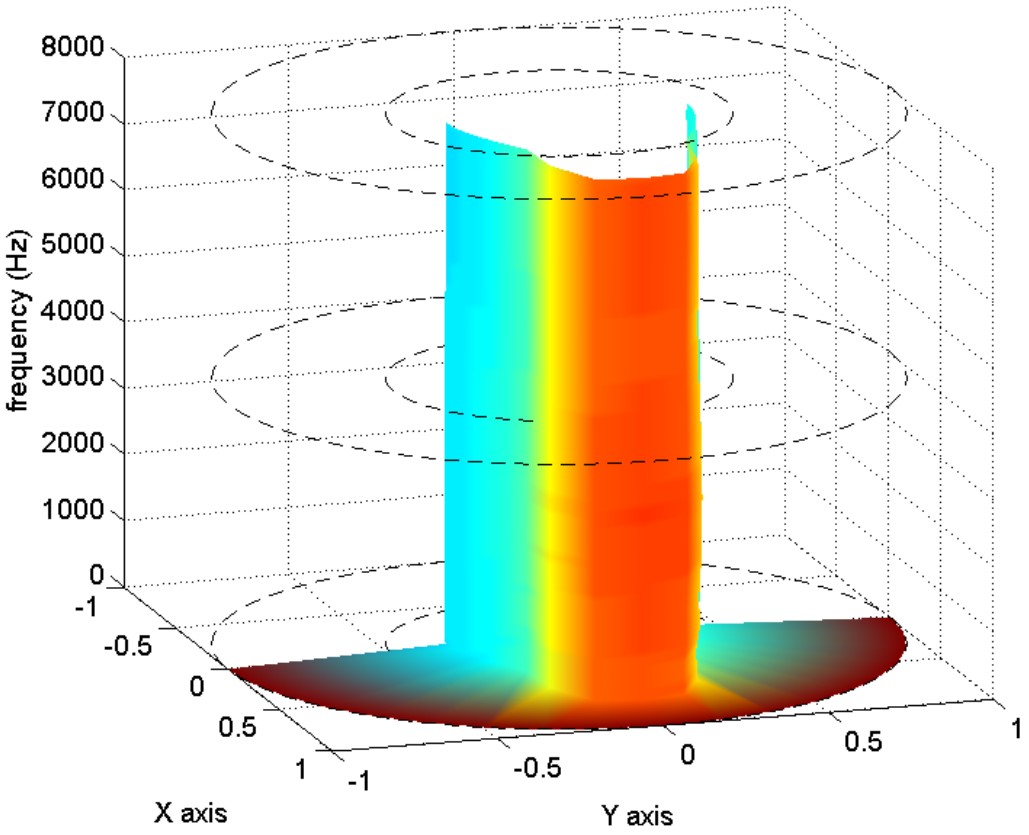


Figure 3.21: Mask presence pattern for two-microphone array with two interfering speech signals, as a function of one interferer's position (ϕ_1, represented as angle on the X-Y plane), $d = 4$ cm, $\phi_T = 20°$, $SIR_1 = 0$ dB, $\phi_2 = -45°$, $SIR_2 = 10$ dB

the presence of multiple interferers. Fortunately, extending the model to this type of scenario simply requires adding magnitude and phase random variables for each interferer. Working this into an equation similar to (3.6) produces the received signal random phasors

$$\begin{cases} X_1 = S + \sum_l I_l e^{-j\alpha_l} \\ X_2 = S + \sum_l I_l e^{-j\alpha_l} e^{-j\omega f_S \frac{d}{c} \sin\phi_l} \end{cases} \tag{3.19}$$

where the I_l and α_l variables represent the magnitudes and phases of each interfering source and ϕ_l is the source location's azimuth. Each I_l and α_l is drawn independently from the others; I_l from the corresponding sources signal type's distribution (at the appropriate fre-

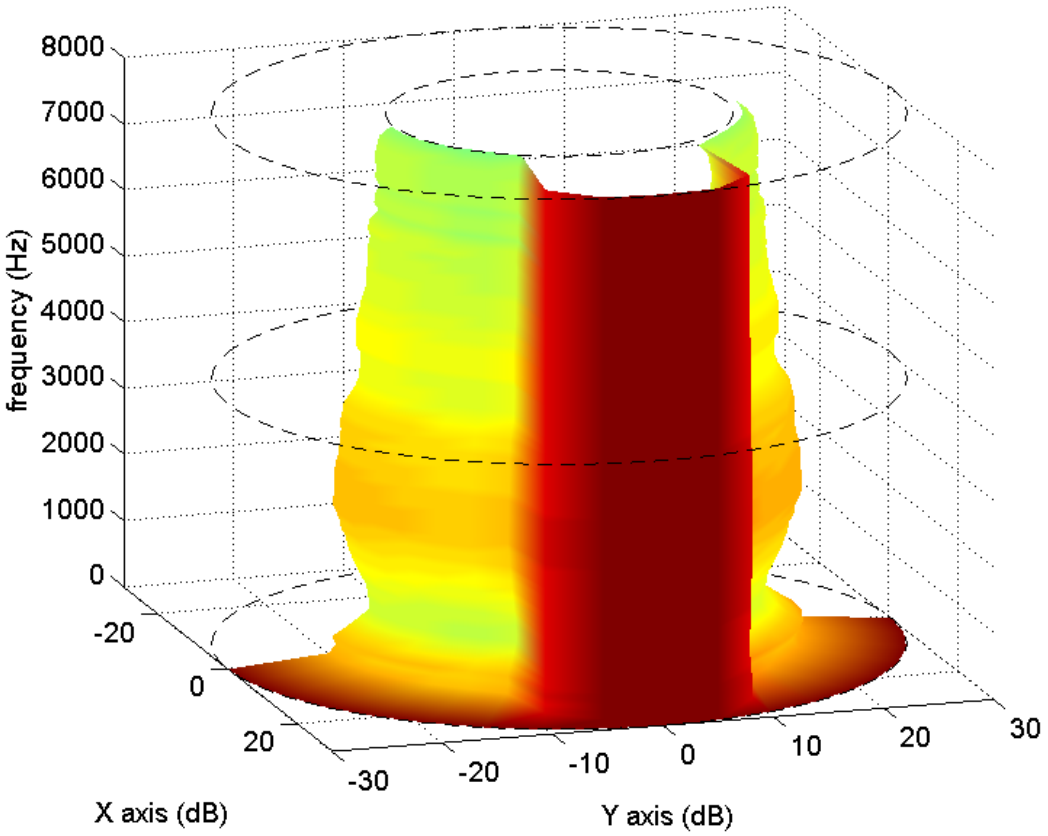


Figure 3.22: Output noise pattern for two-microphone array with two interfering speech signals, as a function of one interferer's position (ϕ_1, represented as angle on the X-Y plane), $d = 4$ cm, $\phi_T = 20°$, $SIR_1 = 0$ dB, $\phi_2 = -45°$, $SIR_2 = 10$ dB

quency) and α_l from a uniform distribution.

While this is conceptually simple, two practical issues arise:

1. The number of random variables grows linearly with the number of interferers. Since the averaging of the beam pattern quantities is being done over the joint distributions of all the random variables, the complexity of modeling a setup with L interferers will be $\Theta\left((R_M R_P)^L\right)$, where R_M and R_P are the resolutions of the magnitude and phase distributions, respectively. In practice, this limits the number of interferers based on the implementation of the model and hardware in use.

2. The number of free variables in the beam pattern also grows linearly with the number

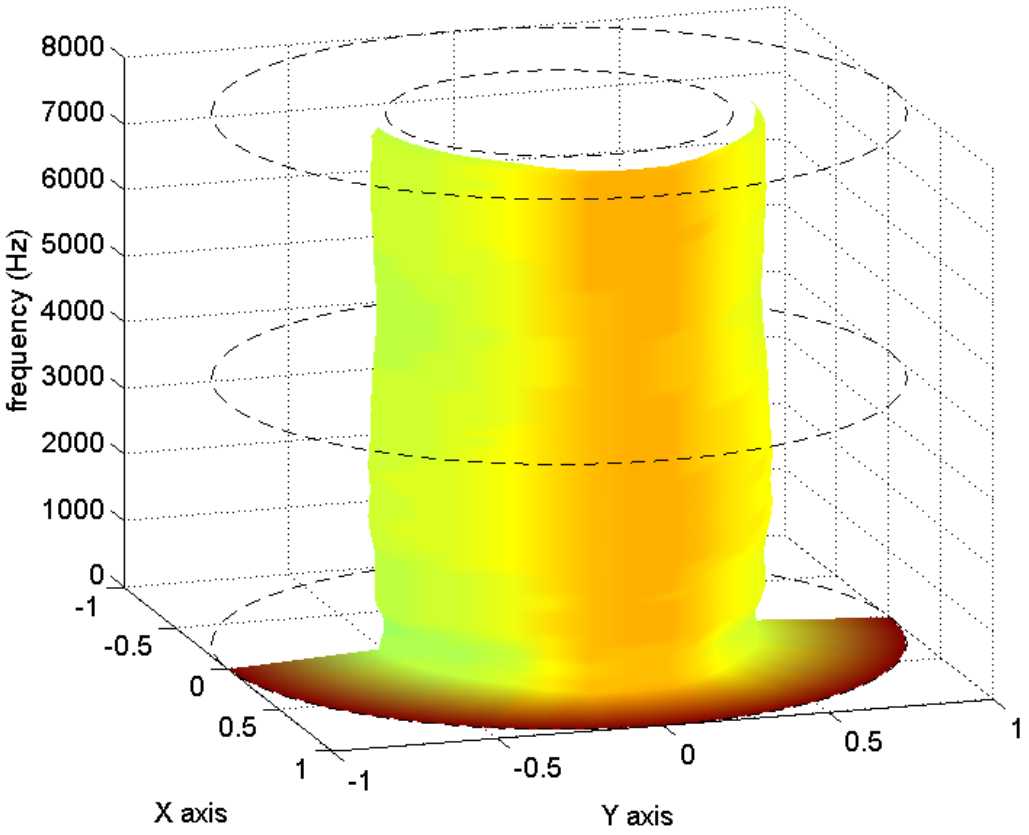


Figure 3.23: Mask presence pattern for two-microphone array with two interfering speech signals, as a function of one interferer's position (ϕ_1), $d = 4$ cm, $\phi_T = 20°$, $SIR_1 = 20$ dB, $\phi_2 = -45°$, $SIR_2 = 10$ dB

of interferers. Specifically, each interfering source will have its own location (ϕ_l) and power (SIR_l); thus, the beam pattern will now be of the form

$$B\left(f; \phi_1, \phi_2, \ldots, \phi_L; SIR_1, SIR_2, \ldots, SIR_L\right)$$

With three-dimensional graphs, it is impossible to visualize the beam pattern with more than two free variables (*e.g.,* frequency and azimuth in Figure 3.3 or frequency and SIR in Figure 3.18). Thus, the best that can be done is to set values for all but two of the free variables and plot the resulting slice of the pattern.

With these limitations in mind, Figures 3.21 through 3.24 illustrate the mask presence

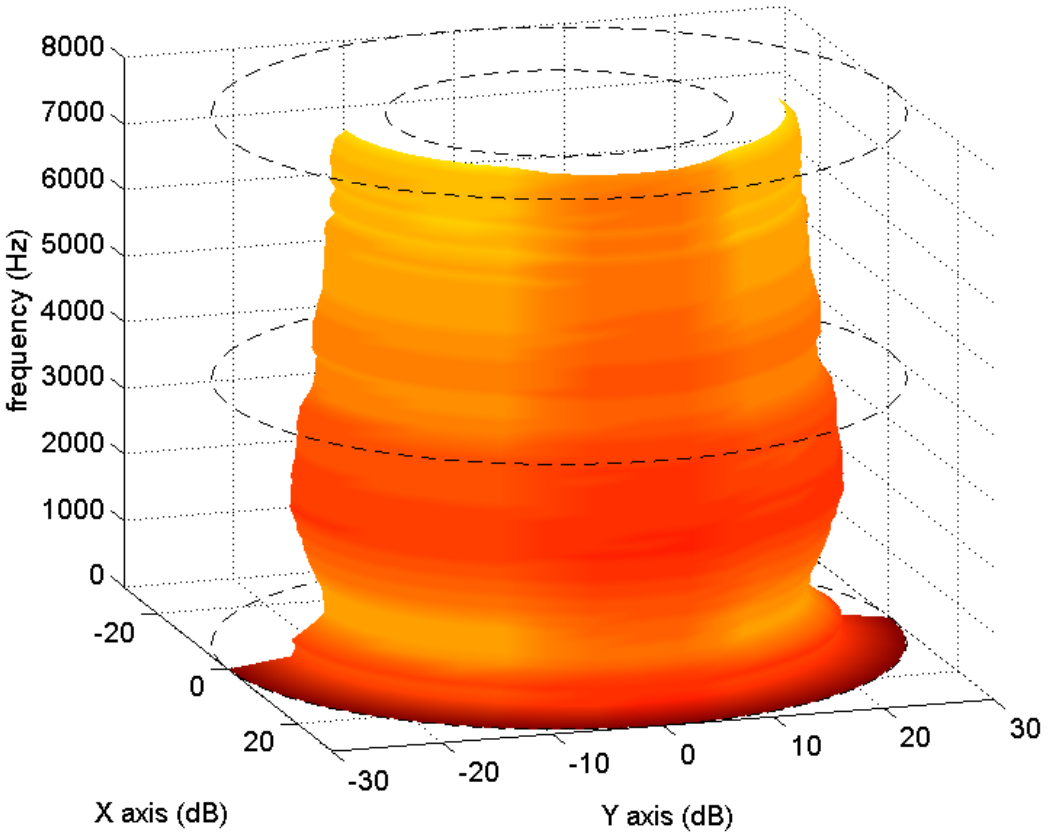


Figure 3.24: Output noise pattern for two-microphone array with two interfering speech signals, as a function of one interferer's position (ϕ_1), $d = 4$ cm, $\phi_T = 20°$, $SIR_1 = 20$ dB, $\phi_2 = -45°$, $SIR_2 = 10$ dB

and output noise patterns corresponding to a target speech signal in the presence of two interfering speech signals. Interferer 2 is kept at $\phi_2 = -45°$ with $SIR_2 = 10$ dB. Interferer 1 is moved across space, tracing the patterns. Figures 3.21 and 3.22 are at $SIR_1 = 0$ dB, while Figures 3.23 and 3.24 are at $SIR_1 = 20$ dB. It is interesting to note that at $SIR_1 = 0$ dB, the patterns are not heavily affected by the presence of the second interferer (compare Figures 3.21 and 3.22 to Figures 3.3 and 3.6), while at $SIR_1 = 20$ dB, the discriminative power of the azimuth threshold is almost invisible, as the interference (and therefore, the masking operation) is dominated by the second interferer.

As before, the noise model is verified using a procedure identical to that of Section 3.3.4, this time using a pair of interfering speech utterances for each target utterance. Each in-

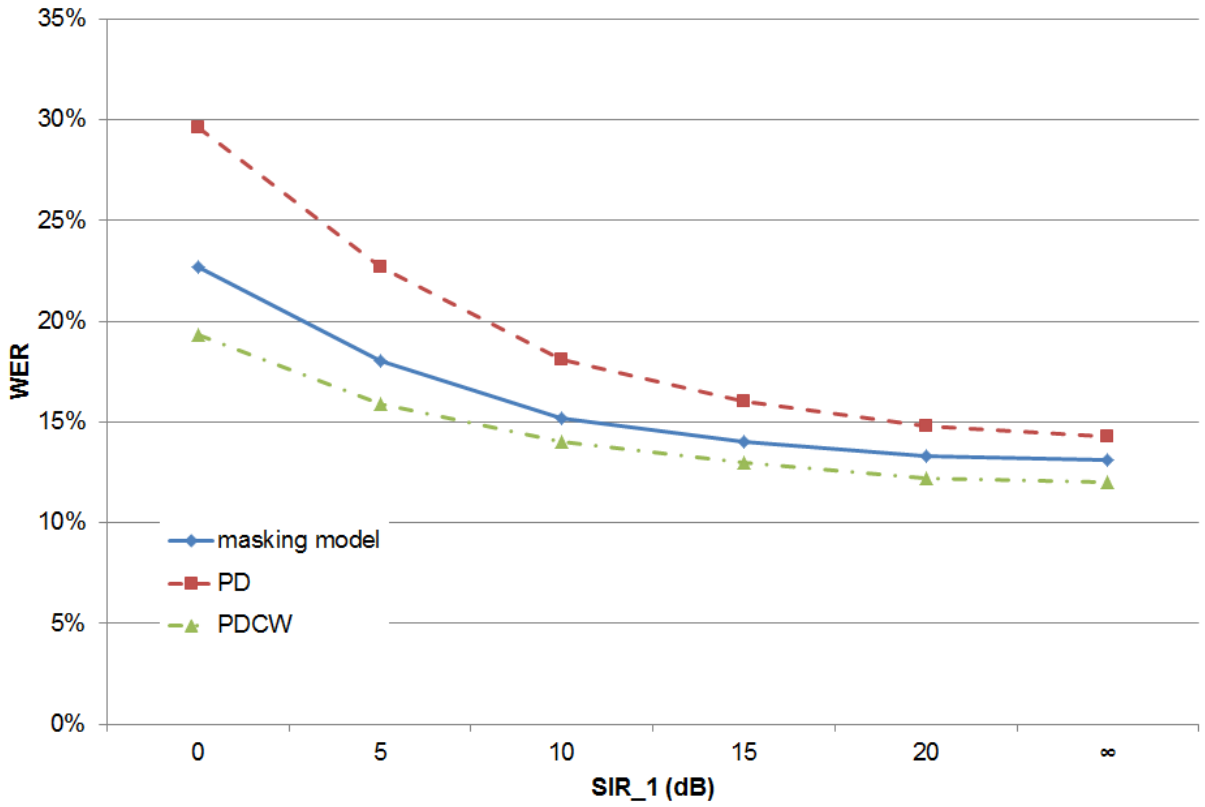


Figure 3.25: Word error rates (WER) of masked speech in the presence of two speech interferers vs. speech with Gaussian noise interferer with spectral profile predicted by output noise model, SIR_1 varies across the graph, $SIR_2 = 10$ dB

terfering utterance is shaped spectrally by the its corresponding contribution to the model-predicted output noise. Interferer 1 is kept at $\phi_1 = 60°$, similar to the results from Section 3.3.4, while the second interferer is located at $\phi_2 = -45°$ and kept at a power of 10 dB below the target signal (*i.e.,* $SIR_2 = 10$ dB). The results, illustrated in Figure 3.25, demonstrate that the predictive accuracy of the model is not significantly affected by the introduction of the second interferer. Of interest in Figure 3.25 is the flattening of the WER lines at higher SIRs; because the SIR values on the graph correspond to one interferer, once its power drops beyond a certain point the WER is dominated by the second interferer.

3.4.4 Reverberant environments

Thus far, the nonlinear beamforming model has only been applied to scenarios in which the target signal, as received by the sensors, is distorted by nothing other than additive interference or noise. This does not take into account the effect of the acoustical environment, namely, reverberation. As discussed in Section 2.1, the effect of reverberation is to filter the signal with an impulse response similar in form to that of Figure 2.2. In order to model the behavior of a masking algorithm in the presense of such distortion, one option is to turn to one of many models of the behavior of reverberant fields at the sensors of an array (*e.g.,* [7, 37, 38]) and build it into our cell-level phasor calculations. However, in the interest of utility and flexibility, a different route has been taken: The model will receive the reverberant impulse response signals in time domain (*i.e.,* $h_{lp}(t)$ as defined in (2.2)). This way, the model can work from actual measured impulse responses or simulated ones. Given these impulse responses, the target and any interferers as received at each sensor can be calculated.

To work this into the random phasor model, recall that these phasors are intended to model the behavior of an individual time-frequency cell of the signal spectrograms. Looking at the lowpass filter formulation of the STFT operation [39] on a sensor input signal:

$$X_1[n,k] = \left(x_1[n]\, e^{-j\omega_k n}\right) * w[n]$$

where $w[n]$ is the window function used in the STFT and $*$ represents convolution. To include the reverberant impulse responses, let us assume for now that there is no interferer present and analyze the target signal only. In this section, the discrete-time versions of the impulse responses between the target source and the two sensors are denoted $h_1[n]$ and $h_2[n]$:

$$
\begin{aligned}
X_1[n,k] &= \left[\left(s[n] * h_1[n]\right) e^{-j\omega_k n}\right] * w[n] \\
&= \sum_m w[n-m]\, e^{-j\omega_k m} \left(s[m] * h_1[m]\right) \\
&= \sum_m w[n-m]\, e^{-j\omega_k m} \left(\sum_r s[m-r]\, h_1[r]\right) \\
&= \sum_m \sum_r w[n-m]\, e^{-j\omega_k m} s[m-r]\, h_1[r] \\
&= \sum_r h_1[r]\, e^{-j\omega_k r} \left(\sum_m w[n-m]\, s[m-r]\, e^{-j\omega_k (m-r)}\right) \\
(m' = m-r) \quad &= \sum_r h_1[r]\, e^{-j\omega_k r} \left(\sum_{m'} w[n-r-m']\, s[m']\, e^{-j\omega_k m'}\right) \\
&= \sum_r h_1[r]\, e^{-j\omega_k r} S[n-r,k] \\
&= \underbrace{\left(h_1[n]\, e^{-j\omega_k n}\right)}_{\text{wideband}} * \underbrace{S[n,k]}_{\text{lowpass}}
\end{aligned}
\tag{3.20}
$$

where the convolution is across the time variable n. As noted, of the two signals being convolved to produce the subband values of the STFT, one is lowpass (*i.e.,* band-limited at low frequencies), by definition of the STFT – this, incidentally, is why STFTs can be downsampled – producing a resulting STFT subband signal that is also lowpass. STFTs created with Hamming windows – the most commonly used window type – are typically downsampled by half the window length, creating frames with 50% overlap [40]. For example, in the analyses producing the results presented earlier we used 80-ms Hamming windows, which at a sampling frequency of 16 KHz produces a window length of $N = 80$ ms$\times$16 KHz $=$ 1280 samples, leading to downsampling factor of 640.

The problem with incorporating this into the current form of the random phasor model is that the model analyzes each T-F cell in isolation, whereas from (3.20) it is clear that each T-F cell value of $X_1[n,k]$ will be a function of the current and many past values of $S[n,k]$ – keep in mind that $h_1[n]$ is the impulse repsonse of a physical system and therefore causal, so

future values of $S[n,k]$ do not matter. Our solution to this problem begins with replacing $S[n,k]$ with a piecewise-constant approximation of itself, $\hat{S}[n,k]$, whose each sample is equal to the the nearest downsampled value of $S[n,k]$ from the same subband; *i.e.,*

$$\hat{S}[n,k] = S\left[\frac{N}{2}\left\|\frac{2}{N}n\right\|, k\right] \tag{3.21}$$

where $\|\cdot\|$ denotes rounding to the nearest integer. This is equivalent to a rectangular interpolation of the downsampled version of $S[n,k]$ (as opposed to the correct sinc-based interpolation). The dependence on past values remains, but there are now fewer of them; combining (3.21) back into (3.20) results in

$$\begin{aligned} X_1[n,k] &\approx \sum_r h_1[r]\, e^{-j\omega_k r} \hat{S}[n-r,k] \\ &\approx \sum_{r=0}^{N_{60}} h_1[r]\, e^{-j\omega_k r} S\left[\frac{N}{2}\left\|\frac{2}{N}(n-r)\right\|, k\right] \end{aligned}$$

where $N_{60} = RT_{60} \cdot f_S$ is the 60-dB reverb time of the impulse response, in samples (roughly, the number of samples over which the impulse response has significant nonzero values). Let us look at the calculation at a time n that is a multiple of the downsampling factor (we are only interested in calculating the downsampled version of the output STFT in any case). Since the ultimate goal is to remove the time factor completely and replace it with random phasor averaging, and since the time reference on the signals is arbitrary, we can without loss of generality look at the calculation at $n = 0$:

$$X_1[0,k] \approx \sum_{r=0}^{N_{60}} h_1[r]\, e^{-j\omega_k r} S\left[\frac{N}{2}\left\|-\frac{2}{N}r\right\|, k\right]$$

Now, using the variable transformation $r = \frac{N}{2}m - r'$, we can break the sum into frames over which $\hat{S}[n,k]$ is constant; *i.e.,* $\frac{N}{2}m$ will be the frame centers and r' can run from $-\frac{N}{4}$ to

$\frac{N}{4} - 1$. This means $-\frac{1}{2} \leq \frac{2}{N} r' < \frac{1}{2}$, which in turn simplifies the argument of $S[\cdot, k]$ above:

$$\frac{N}{2} \left\| -\frac{2}{N} r \right\| = \frac{N}{2} \left\| -m + \frac{2}{N} r' \right\| = -\frac{N}{2} m$$

The convolution sum can now be broken in two:

$$\begin{aligned} X_1[0,k] &\approx \sum_{m=0}^{2\frac{N_{60}}{N}} \sum_{r'=-\frac{N}{4}}^{\frac{N}{4}-1} h_1\left[\frac{N}{2}m - r'\right] e^{-j\omega_k\left(\frac{N}{2}m - r'\right)} S\left[-\frac{N}{2}m, k\right] \\ &\approx \sum_{m=0}^{2\frac{N_{60}}{N}} S\left[-\frac{N}{2}m, k\right] \left(\sum_{r'=-\frac{N}{4}}^{\frac{N}{4}-1} h_1\left[\frac{N}{2}m - r'\right] e^{-j\omega_k\left(\frac{N}{2}m - r'\right)} \right) \end{aligned} \tag{3.22}$$

The inner sum in (3.22) is independent of any signal behavior and can be calculated beforehand. The result will be one value per frame m in subband k:

$$\begin{cases} h_1^{(k)}[m] = \displaystyle\sum_{r'=-\frac{N}{4}}^{\frac{N}{4}-1} h_1\left[\frac{N}{2}m - r'\right] e^{-j\omega_k\left(\frac{N}{2}m - r'\right)} \\ h_2^{(k)}[m] = \displaystyle\sum_{r'=-\frac{N}{4}}^{\frac{N}{4}-1} h_2\left[\frac{N}{2}m - r'\right] e^{-j\omega_k\left(\frac{N}{2}m - r'\right)} \end{cases} \tag{3.23}$$

where the corresponding set for the right sensor is also included. (3.22) can be rewritten with this simplified notation:

$$X_1[n,k] \approx \sum_{m=0}^{2\frac{N_{60}}{N}} S\left[-\frac{N}{2}m, k\right] h_1^{(k)}[m] \tag{3.24}$$

In (3.24), given that the upper limit on m is

$$2\frac{N_{60}}{N} = 2 \times \frac{RT_{60} \cdot f_S}{\text{window length} \cdot f_S} = 2 \times \frac{RT_{60}}{\text{window length}}$$

the sum will consist of a relatively small number of terms; *e.g.,* with a reverberation time of

400 ms and 80-ms windows, there will be only 11 terms. Thus, only 11 values (1 present and 10 past) of the signal STFT are necessary to approximate the received STFTs at the left and right sensors. As per the nonlinear beam pattern approach from the previous sections, each of these values can now be modeled with a random phasor whose magnitude is distributed according to speech subband levels and whose phase is random:

$$\begin{cases} X_1 = \sum_{m=0}^{2\frac{N_{60}}{N}} S_m e^{j\alpha_m} \cdot h_1^{(k)}[m] \\ X_2 = \sum_{m=0}^{2\frac{N_{60}}{N}} S_m e^{j\alpha_m} \cdot h_2^{(k)}[m] \end{cases} \tag{3.25}$$

In this model, each $S_m e^{j\alpha_m}$ represents the T-F cell value of the signal at m half-frames in the past (with $m = 0$ being the current frame). The main flaw of this model is that the S_m variables are drawn independently from the subband signal level distribution. While the assumption of identical distributions is realistic, their independence is not; most real signals, including speech, are correlated with themselves to varying degrees over time. However, the use of longer windows validates this assumption to some extent; the farther apart the samples of the STFT are taken, the less correlated they are with each other.

Now that the left and right sensor phasors' distributions are calculable from signal distributions, the remainder of the process is similiar to that of Section 3.3, with a few exceptions discussed here. The first is that the output noise metric must be modified. For reverberation, it seems reasonable to consider the present signal level, S_0 in (3.25) to be the desirable signal portion, and the rest of the S_ms to be interfering. In other words, we assume that the early arriving copies in a reverberant field are desirable and the later arrivials, belonging to past values of the signal, interfere with the early arrivals, which is a reasonable description of the

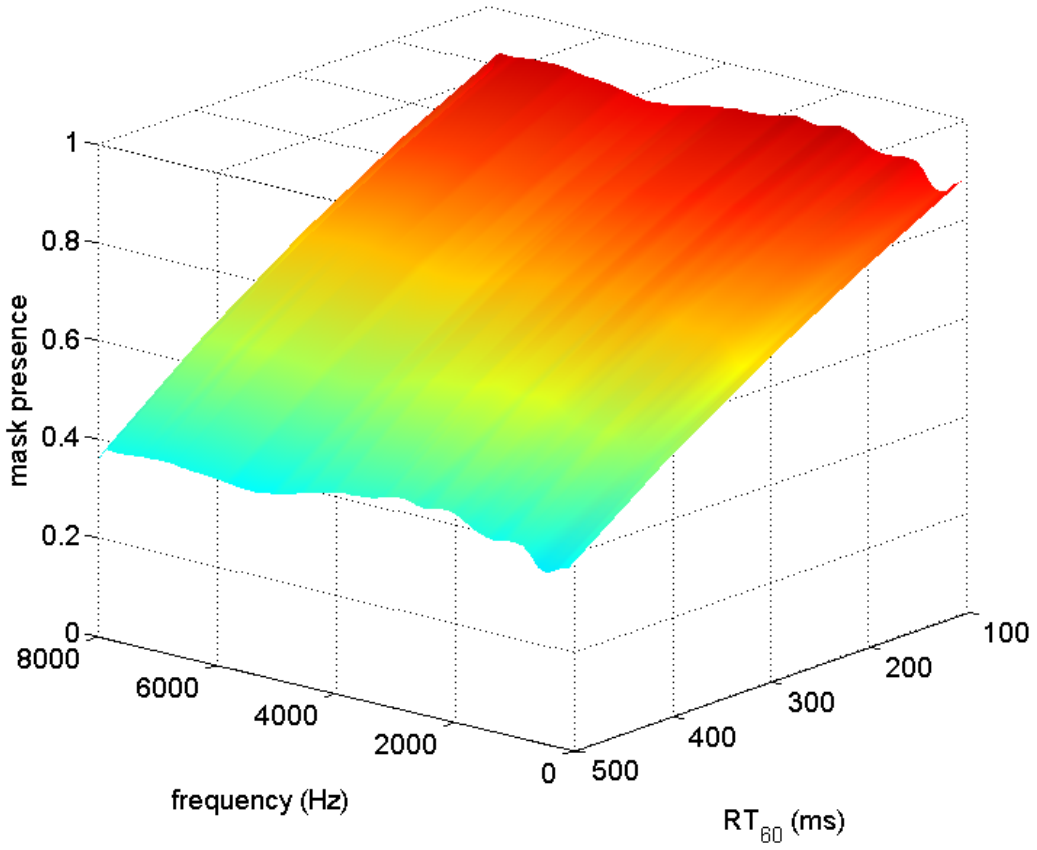


Figure 3.26: Mask presence pattern for two-microphone array with target source in reverberant environment, $d = 4$ cm, $\phi_T = 20°$

adverse effects of reverberation in general. (3.9) can be modified accordingly:

$$N^2 = \overbrace{\left|S_0 h_1^{(k)}[0](1-M)\right|^2}^{\text{signal suppression}} + \overbrace{\sum_{m=1}^{2\frac{N_{60}}{N}} \left|S_m h_1^{(k)}[m] M\right|^2}^{\text{reverberation leakthrough}} \tag{3.26}$$

Another important distiction is that, thus far, only a single target signal has been considered in reverberant fields; therefore, the definitions of interferer azimuth and SIR are no longer relevant. However, a new free variable does appear: the reverberation time. Thus, the beam patterns will be of the form $B(\omega; RT_{60})$. Figures 3.26 and 3.27 demonstrate the mask presence and output noise patterns obtained using the procedure described in this section. To generate these figures, the model was given impulse responses simluated using

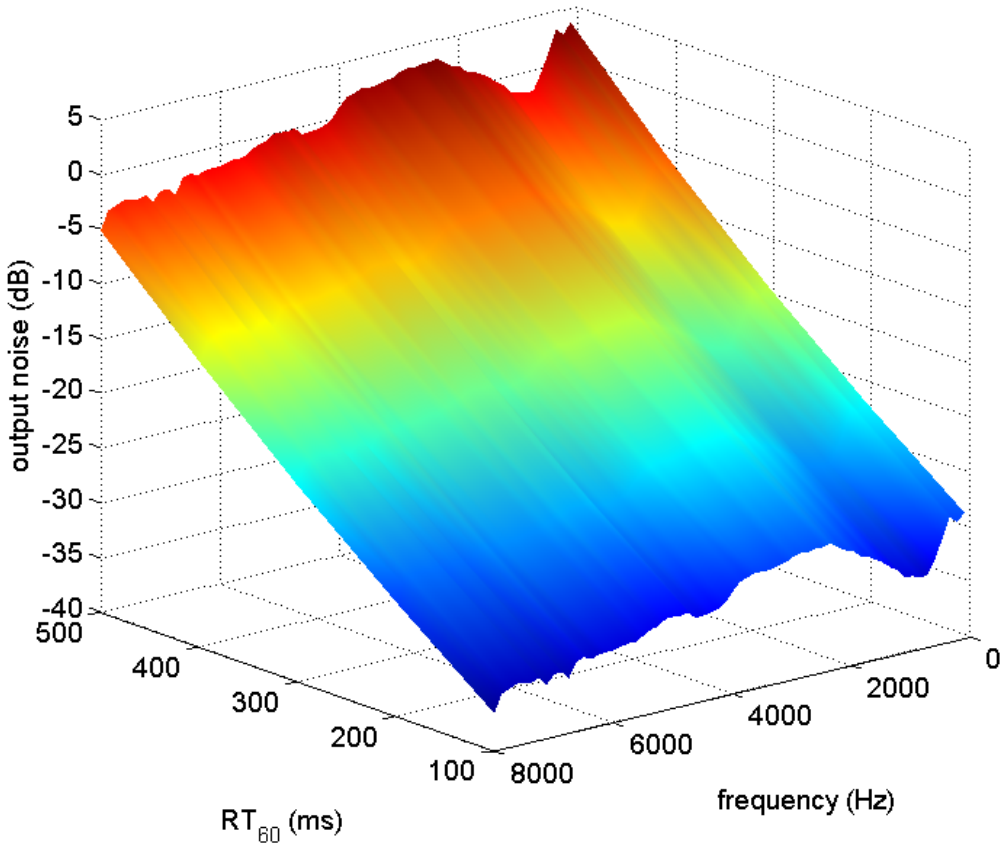


Figure 3.27: Output noise pattern for two-microphone array with target source in reverberant environment, $d = 4$ cm, $\phi_T = 20°$

Habets' implementation of the image method [9] in a featureless room with dimensions of $8 \times 5 \times 3$ meters (Figure 2.2 is an example). Different reverberation times are obtained by varying the absorption coefficients of the room's surfaces. The two sensors of the array are spaced at 4 cm, with the source at a distance of 2 m along the array's broadside axis. The source and sensors are all positioned on a plane halfway between the ceiling and floor (*i.e.*, 1.5 m away from each).

Of course, including one or more interferers would be conceptually simple. Terms identical to those of (3.23) can be constructed for each impulse response from each interferer to each sensor, which would then need to be added into (3.24) and (3.25). The resulting beam

patterns would have multiple free variables:

$$B\left(f;\phi_1,\phi_2,\ldots,\phi_L;SIR_1,SIR_2,\ldots,SIR_L;RT_{60}\right)$$

The difficulty, however, lies in the implementation. Because each interferer would now require its own set of $2\times\frac{RT_{60}}{\text{window length}}+1$ random phasors (each having a random amplitude and phase), the computational complexity of the problem would increase greatly. For this reason, only results with a single target are included here.

The remaining task is to perform verification experiments for the reverberant model. As in the non-reverberant cases, the goal is to synthesize test signals using a procedure that reflects the prediction of the model in some fashion and perform speech recognition on them. The results are then compared to parallel experiments, wherein test signals with the actual degradation in question (in this case, reverberation with the same impulse responses) are processed using the masking algorithm. The latter experiments are simple to run; the key lies in the synthesis procedure for the first set. In this case, the model is predicting the contribution of each S_m – *i.e.,* each of the past few frames of the signal – to the overall output noise (see (3.26)). It would be reasonable, therefore, to synthesize a signal corrupted by its own past frames in a manner consistent with the predicted value of these contributions. Since the model calculates these quantities at each frequency, the synthesis can operate on each subband separately. Mathematically, if $X\left[n,k\right]$ is the STFT of a clean utterance, downsampled by a factor of half the window length (which corresponds to the separation of the model's S_m values in time), the STFT of the model-predicted output signal for the reverberated input signal will be

$$Y\left[n,k\right]=\sum_{m=0}^{2\frac{N_{60}}{N}}\overbrace{E\left[\left|S_m h_L^{(k)}\left[m\right]M\right|^2\right]}^{\text{from model}}X\left[n-m,k\right] \tag{3.27}$$

Figure 3.28 illustrates the results of these experiments. The array, clean data, masking

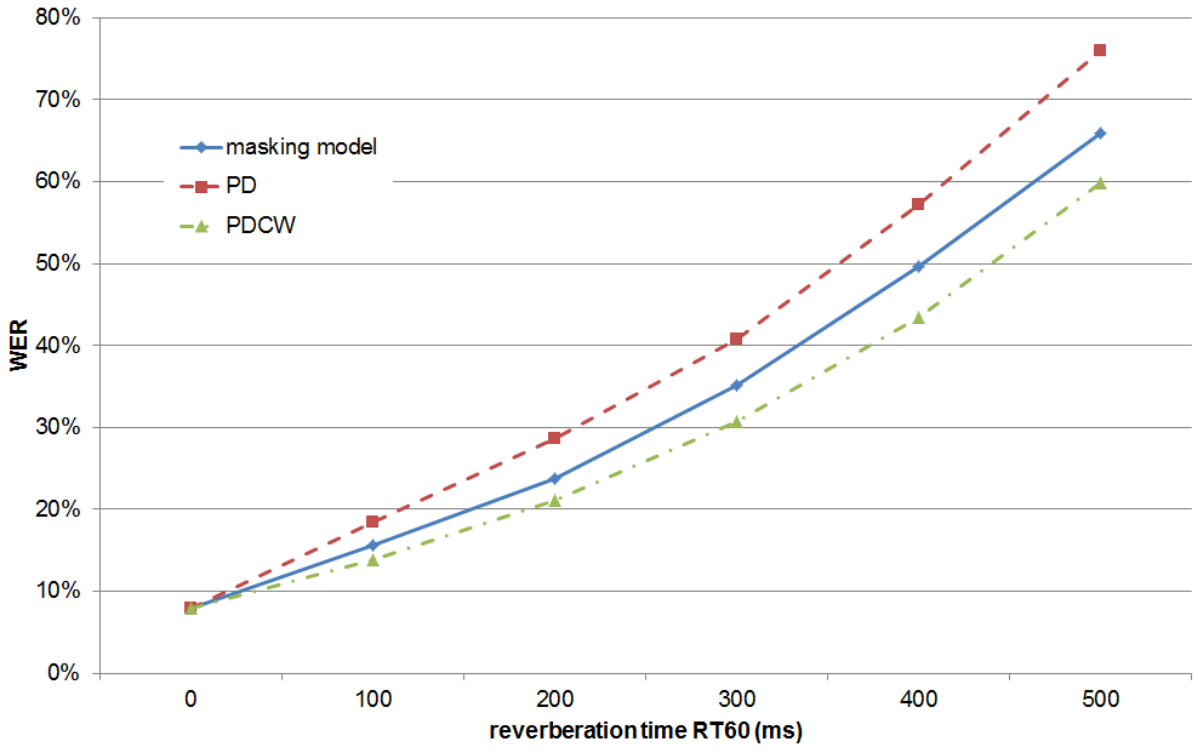


Figure 3.28: Word error rates (WER) of masked reverberated speech vs. speech self-corrupted according to output noise model

algorithms and recognition engine are identical to those of Section 3.3.4. The impulse responses used in these experiments are identical to those used in generating Figures 3.26 and 3.27.

3.4.5 Comparison of different scenarios

Having adapted our analysis method to various scenarios, we return to a familiar question: How descriptive (or predictive) is the metric in use? Within a particular environment type, the question has been answered by the verification experiements performed; the performance predicted by the model tracks actual performance relatively well in each of these scenarios. However, as the definition of output noise (and SNR) differs somewhat from one scenario to the next, the consistency of these definitions comes into question.

To examine this, we look at the output noise-to-signal ratio (*i.e.*, output noise normalized by signal level) in a number of different scenarios. These scenarios are chosen so that PDCW

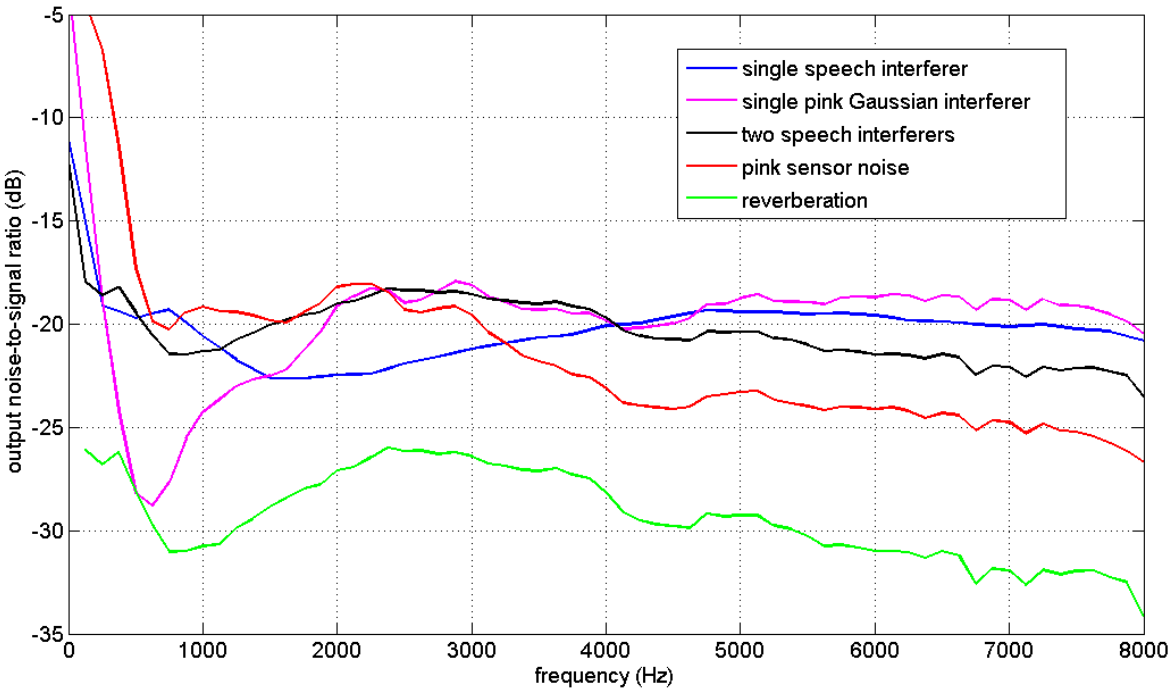


Figure 3.29: Model-predicted output noise levels in various scenarios, each with $WER \approx 15\%$

processing produces a word error rate of roughly 15% in each case – look at the $WER = 15\%$ intercepts on the green lines in Figures 3.12, 3.17, 3.20, 3.25 and 3.28). Ideally, a well-defined output noise metric would look similar (although not necessarily identical) for these cases, regardless of the underlying adaptations.

Figure 3.29 illustrates this output noise, as a function of frequency, for the five cases in question. As can be seen, all but one of the models produce output noise values that are roughly similar, in level if not in shape. The one exception is when modeling reverberant environments; as was explained in Section 3.4.4, the output noise in that model is defined using variables representing past signal values, which is qualitatively somewhat different than the other cases. What this means is that our nonlinear beam patterns are not useful for comparing the performance of an algorithm in reverberation with its performance in non-reverberant environments. While this is unfortunate, it is also true with conventional beam patterns; the shape of the pattern itself does not provide much detail about the performance of the beamformer in reverberant vs. non-reverberant environments.

3.5 Conclusions

This chapter has detailed the development of a model for the analysis of the behavior of masking algorithms. The model produces intuitive, descriptive metrics – based loosely on the concept of the beam patterns of linear beamformers – that can be presented as *nonlinear beam patterns*. The model is based on simple, straightforward theory and not particularly computationally costly. Furthermore, the resulting metrics prove to be relatively predictive of the performance of the algorithm under analysis. Although the initial model is designed for a limited use case, extensions are not difficult to develop and implement. To demonstrate this, several other scenarios have been considered and the model adapted to analyze the behavior of the masking algorithm in each one.

The main benefit of such a model is that it provides a standalone tool for examining the behavior of a masking technique (and, by extension, the comparison of techniques to each other). Since it is based solely on probabilistic calculations, it alleviates the need to process real signals. In the context of speech recognition (and similar pattern recognition tasks) it produces descriptors independent of database, recognizer, *etc.* – factors that introduce a great deal of variability into the current analyses and comparisons of masking techniques.

There is more to be done, of course. Perhaps the most pressing area of inquiry is the definition of the output noise metrics (*e.g.,* (3.9) or (3.26)). Currently, these are chosen to be simple and intuitive. It could well be, however, that simple addition of the distortion terms in these equations is not the optimal combination when considering predictive accuracy. Other linear combinations (more generally, all nondecreasing functions $C\left(\left(S\left(1-M\right)\right)^2, \left(IM\right)^2\right)$ of the distortion terms) can be examined. Once such a metric has been established, it can be calculated and plotted similarly to the noise metric from Section 3.3.2 to produce a more predictive, if slightly less intuitive, form of the nonlinear beam pattern itself. Furthermore, either the current noise measure or a more optimal one can be applied as a cost function to blindly optimize the parameters of a masking algorithm.

Another potential area of inquiry is more practical in nature: an examination of the

tradeoffs between complexity and accuracy. Although the model is, in most cases, computationally simple, the dependence on averaging over joint distributions does create the potential for great complexity when multiple variables are involved (as was discussed in Section 3.4.4). It could well be that the number of necessary calculations can be reduced in an intelligent way that minimizes the impact on the descriptive power and predictive accuracy of the resulting nonlinear beam patterns.

Chapter 4

Can Masking Be Improved?

This chapter focuses on improving the performance of T-F masking algorithms. Section 4.1 begins by examining some of the current shortcomings of masking algorithms and identifying a few potential areas of improvement. Section 4.2 focuses on improving the familiar two-channel version of masking, while Section 4.3 details attempts to develop a method to extend masking to arrays of more than two elements.

4.1 Strengths and weaknesses of masking

Two-channel masking algorithms have been shown to substantially improve the quality of speech in the presence of reverberation and interference [27, 28]; we have also confirmed this via experimentation. With a two-microphone configuration, we have also found masking to be superior to linear beamforming, especially in the presence of reverberation. With only two elements, linear beamforming techniques produce sidelobes that are too large to properly suppress off-target interference components across the wide range of frequencies that comprise speech signals (see Figure 3.9), whereas masking is able to achieve consistent, if not stellar, suppression outside the "cone of acceptance" (see Figures 3.3 and 3.6). In addition, our implementation of two-channel masking includes a step where both input signals are masked and then averaged before reconstruction; this is itself a beamforming

operation. The computational cost of masking, while higher than linear beamforming, is not a significant issue; case in point, even when implemented in MATLAB, PDCW runs at a small fraction of real time and its computational cost is typically dwarfed by the cost of the speech recognition (or other pattern recognition task).

However, there are many issues with masking that limit its usefulness. One obvious area is the significant gap between the quality of the ouptut of a masker and the original, clean signal; *e.g.,* compare the recognition accuracy of PDCW-processed speech in the presence of even a single interferer to that of clean, unprocessed speech in Figure 3.12. Closing this gap is, of course, the overarching goal of all array processing research. Section 4.2 details attempts to improve the performance of T-F masking. There are also more immediate concerns, especially when masking is compared to linear beamforming. Results of previous studies using these techniques (*e.g.,* [22, 23, 25, 26, 27, 28, 32]) suggest that while T-F masking techniques typically perform well in their intended target scenarios, they do not generalize as easily or degrade as gracefully as linear beamforming techniques. One example is sensitivity to steering errors. While, in theory, modifying PDCW to target sources off the array's broadside axis should be as simple as shifting the received signals, in practice this degrades performance significantly, especially when presented with real-world signals as opposed to simulated data. Furthermore, the adaptive processing techniques used in linear beamforming have yet to find competitive counterparts in masking; although attempts have been made with some success (*e.g.,* [43]), the flexibility and robustness of linear beamforming when faced with unknown, and even moving, signal source locations is unmatched.

Perhaps the most glaring problem with masking is scalability; the performance of linear processing techniques can be improved simply by using larger arrays, while nonlinear processing techniques typically do not scale well, if at all. Indeed, while there are large bodies of literature on single- and dual-channel masking, multi-channel masking seems to have been comparatively neglected. This is mainly because there are very few intuitive approaches to scaling these algorithms; this issue will be discussed in more detail in Section 4.3.

4.2 Improving two-channel masking

As stated in Section 4.1, even in a two-microphone configuration with no steering errors or other such issues, there remains room for improvement. Of the three main building blocks of a T-F masking algorithm – namely, time-frequency analysis, mask estimation and reconstruction (see Figure 2.5) – we will focus on the latter two and attempt to devise modifications that will lead to the desired performance improvements. As always, we are using PDCW as a launching point, as it is a relatively simple and intuitive, yet high-performing, version of T-F masking.

4.2.1 Mask estimation

Recall that in PDCW the left-right phase difference thresholding of (2.11) and (2.12) are used to approximate the oracle mask introduced in (2.9). This leads to two potential causes of mask suboptimality: the oracle mask as defined could be suboptimal and/or the phase-difference-based estimation could be error prone.

Fortunately, the latter is easily quantifiable. We experimented with the same configuration used to produce Figure 3.12, with a single interferer at various SIR levels. Since the degradation (*i.e.,* speech-on-speech interference) is simulated, the actual clean target and interfering signals are known. The oracle mask can therefore be constructed from their spectrograms as per (2.9); for the remainder of this section, this will be referred to as $M[n,k]$. An actual mask is also obtained using PD, as per (2.11); this will be named $\hat{M}[n,k]$. These two are then compared to calculate a mask estimation error for that particular utterance (this is pre-smoothing, so the masks are binary):

$$e_M = \left\langle \left\langle M[n,k] \oplus \hat{M}[n,k] \right\rangle_k \right\rangle_n \tag{4.1}$$

where $\oplus$ denotes exclusive-or and $\langle\cdot\rangle_n$ denotes averaging over n. These per-utterance errors can then be averaged across utterances created using similar SIR values. The result is that,

consistently across SIR, the binary PD mask $\hat{M}[n,k]$ estimates the oracle mask $M[n,k]$ with less than 5% error.

The two masks (oracle and PD-based) can also be used to calculate an error power metric akin to that of (3.3) and (3.9). Given

$$\begin{cases} Y[n,k] = \frac{1}{2}\left(X_1[n,k] + X_2[n,k]\right) M[n,k] \\ \hat{Y}[n,k] = \frac{1}{2}\left(X_1[n,k] + X_2[n,k]\right) \hat{M}[n,k] \end{cases} \tag{4.2}$$

(*i.e.,* the output of binary masking using the oracle and PD-based masks, respectively), the error power metric would be

$$\epsilon = \frac{\sqrt{\left\langle \sum_k \left| Y[n,k] - \hat{Y}[n,k] \right|^2 \right\rangle_n}}{\sqrt{\left\langle \sum_k \left| Y[n,k] \right|^2 \right\rangle_n}} \tag{4.3}$$

where the error is scaled by the rms error value of the output signal spectrogram itself (the denominator of (4.3)). This error turns out to consistently lie around 1-2%, across various input SIRs. These results indicate that mask estimation itself is not a significant source of performance degradation.

As for suboptimality of the oracle mask itself, it is clear that (2.9) as defined is the mask that minimizes total output noise as defined in (3.9). However, it can still be experimented with; the oracle mask definition can be modified to accomodate an optimal SIR threshold other than 0 dB:

$$M[n,k] = \begin{cases} 1 & \frac{|S[n,k]|^2}{|I[n,k]|^2} > SIR_T \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

It turns out, however, that the optimal threshold is indeed 0 dB. We have run speech recognition experiments using PDCW to process simulated data, replacing the mask estimation stage of PDCW with (4.4) with various threshold SIRs. The target and interfering

utterances are drawn randomly from the DARPA Resource Management (RM1) database, with the target/interfering utterance pairing maintained across experimental scenarios. The recognizer is CMU Sphinx-3, with its acoustic models trained on clean speech from the RM1 training set. Experiments were performed at nominal input SIRs of 0, 10 and 20 dB and threshold SIRs ranging from -9 to 30 dB, in increments of 3 dB, for each scenario. We found that, across all these scenarios, $SIR_T = 0$ dB is consistently the best choice – *i.e.,* produces the lowest word error rate[1].

In closing, we note that while current mask estimation techniques seem nearly optimal, these observations and conclusions do not hold in reverberant environments. On one hand, the oracle mask is not so easy to define in reverberation. Ideally, reverberant signal components that arrive early enough to reinforce the signal would be kept and later reverberations would be rejected; however, the exact cutoff point is both difficult to determine and frequency dependent. On the other hand, a phase-difference-based estimator for such a mask would not be as simple as (2.11). This can also be observed experimentally; in the absence of reverberation, as long as the algorithm's threshold azimuth is less than the interferer azimuth, changes in the threshold do not noticeably impact performance. With reverberation, however, the dependence is significant; the optimum ϕ_T in reverberation is noticeably smaller than halfway between the signal and interferer, which is consistent with rejecting a number of the reverberant signal components as well. Thus, optimal mask generation for reverberation remains an open question, which will be discussed further in Section 4.4.

4.2.2 Reconstruction and feature extraction

As mentioned in Section 2.3.1 and discussed in [28], the smoothing of the binary masks before application (*i.e.,* the "channel weighting" step) provides a noticeable improvement to recognition accuracy, not to mention subjective signal quality and human intelligibility. This performance gap is also on display in Figure 3.12; note the difference between the lines

[1] This, incidentally, also supports the use of the particular noise metric used in Chapter 3, as it is minimized using this particular oracle mask.

marked "PD" and "PDCW". This is due to the sharpness of the spectrograms produced by binary masking; that type of rapid signal level increase and decrease across time and frequency is unnatural and distortive. This suggests that other reconstruction methods, perhaps also aimed at producing "natural"-looking spectra, might improve signal quality even more. The following sections will describe attempts to improve the performance of PDCW by modifying or replacing the channel weighting and/or the overlap-add reconstruction step.

Missing feature reconstruction

One approach is to mask the signal using the binary mask (*i.e.,* remove the smoothing step), then apply a missing feature reconstruction method to the resulting spectrogram. There are numerous such methods in the open literature, all based on reconstructing features that conform to what one would expect of realistic speech features; they differ in the metrics used to measure said realism, and of course in the chosen method of filling in the absent features. To experiment with this approach, we have chosen a method developed by Raj, *et al.* [44] that is based on clustering spectral vectors. We found that doing so consistently improves performance over PD (*i.e.,* binary masking). When compared to PDCW (*i.e.,* smoothed masking), however, the improvement only comes at higher SIR levels. As an example, Figure 4.1 shows the results of this experiment run on the same configuration as that of Figure 3.12: A single target and single interferer, located at $\phi = 60°$, at various nominal input SIRs, with the masker using an azimuth threshold of $\phi_T = 20°$. The database and recognition details are as described in Section 3.3.4. Looking at Figure 4.1, the benefit of using cluster-based missing feature reconstruction is not negligible at 20 or 15 dB SIR (about 10% relative WER reduction in this scenario at 20 dB), but the missing feature reconstruction falls behind the existing reconstruction method at lower SIRs.

This behavior can probably be explained as follows: At higher SIR levels, there is more of the original spectrogram on which to base the reconstruction, whereas at lower SIRs, large swaths of the signal are masked out (and therefore, not available to the algorithm). The

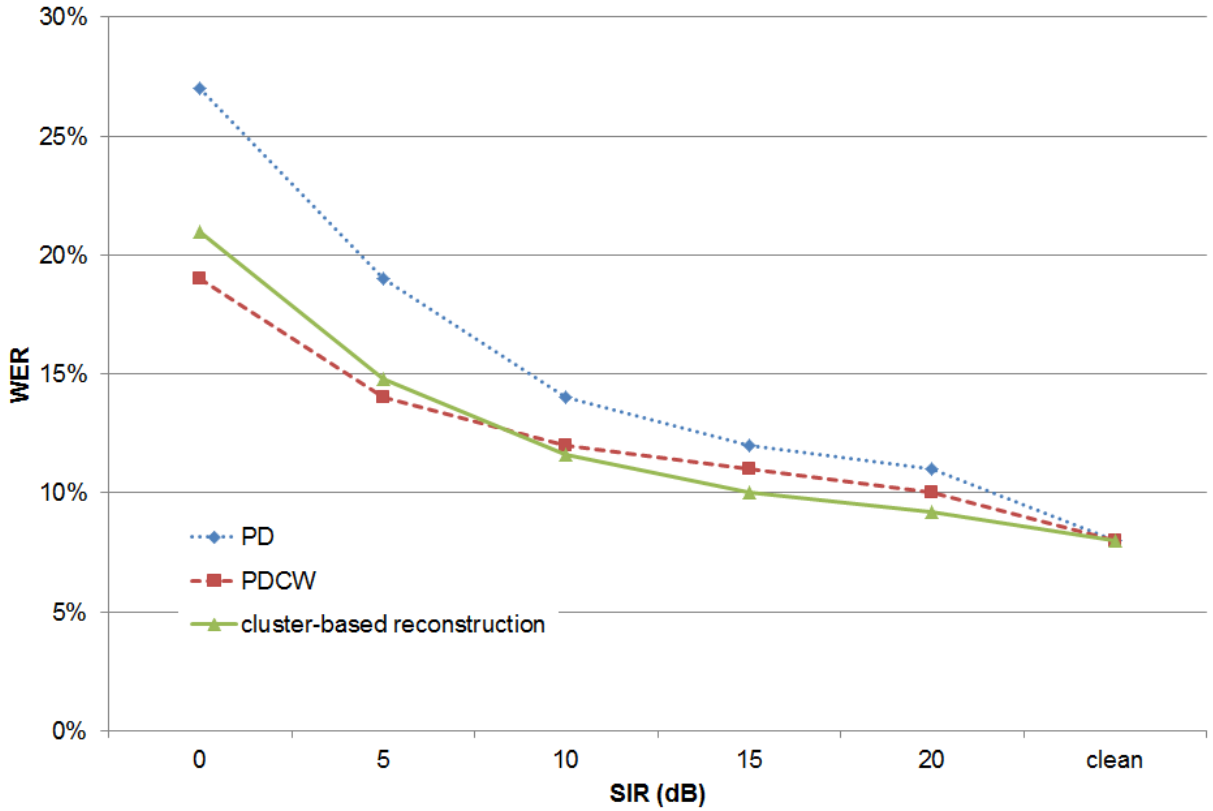


Figure 4.1: Word error rates (WER) of PD, PDCW and PD-masking with cluster-based reconstruction

cluster-based reconstruction algorithm, like most of its peers, tends to perform poorly when confronted with large areas of missing data; it is much better suited to filling in gaps that are spread uniformly throughout time and frequency [44, 45].

There are other issues with using this type of reconstruction method: not only does it greatly increase the computational complexity of the algorithm, it also includes a training phase. While it is true that a speech recognition application will require heavy processing power and a training database of clean speech anyway, there are scenarios where it can be an encumbrance for the front end to require training as well; *e.g.,* mobile applications where the front end is on the device but feature vectors are sent to a server for recognition.

Direct feature extraction

An alternative approach is to bypass reconstruction altogether. In a speech recognition application, the reconstructed time-domain speech signal is just passed to a feature extraction algorithm (usually MFCC processing [1]) anyway, so aside from demonstrations to human listeners, there is no real value in reconstructing a time-domain waveform. Figure 4.2 illustrates the reconstruction phase of masking, followed by the MFCC feature extraction phase of a typical speech recognizer.

Replacing the reconstruction and the following feature extraction steps with a feature extraction directly off $Y[n,k]$ would, at the very least, save the computational cycles devoted to the OLA reconstruction of the masker and the initial STFT of the feature extraction. The procedure here involves passing the masked signal $Y[n,k]$ to the post-STFT stages of MFCC processing; namely, power calculation (*i.e.,* magnitude-squaring), mel-frequency-scale power

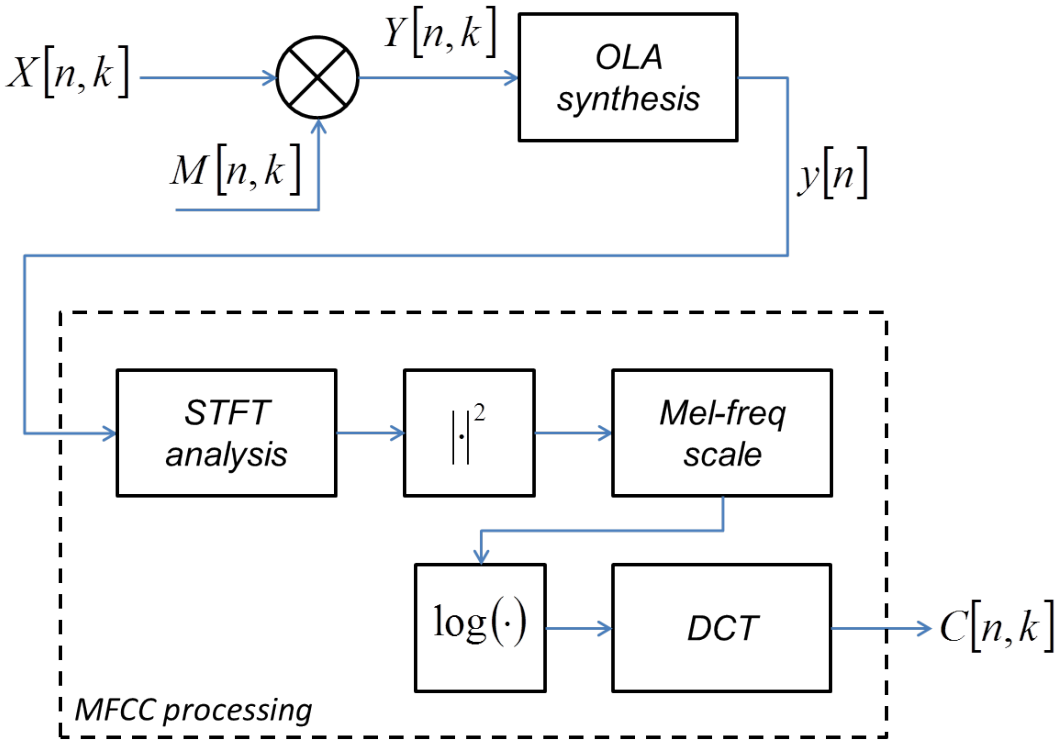


Figure 4.2: Reconstruction of T-F masking, followed by MFCC feature extraction

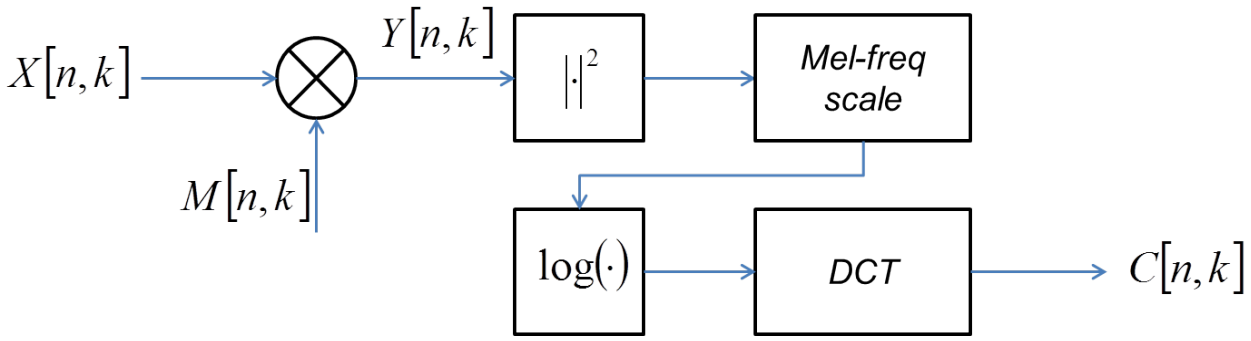


Figure 4.3: Direct MFCC-style feature extraction from the masked signal

mapping, a logarithm and finally the discrete cosine transform (DCT). Figure 4.3 illustrates this approach.

As Figure 4.4 shows, this approach also proves to be mildly beneficial in terms of performance. The results are from experiments in a scenario identical to that of Figure 4.1, with one exception: the standard MFCC processing is now replaced with the direct feature

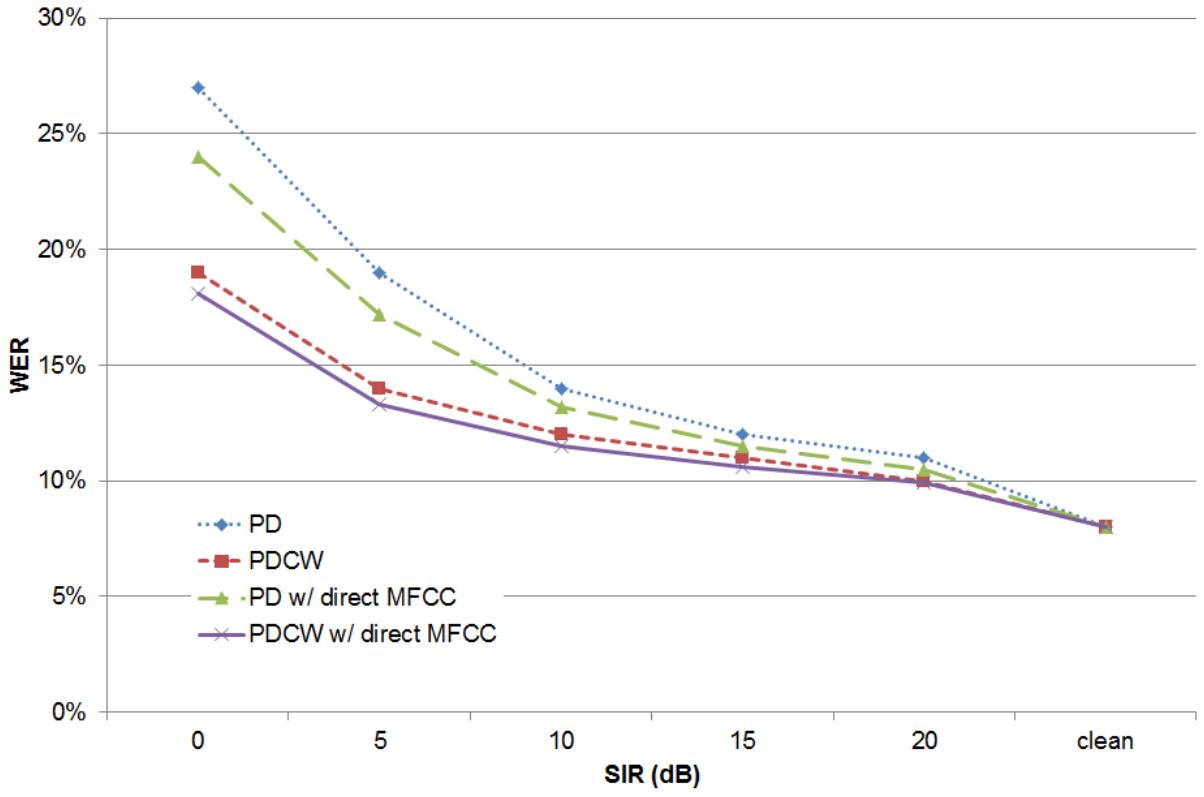


Figure 4.4: Word error rates (WER) of PD and PDCW with reconstruction vs. direct MFCC-style feature extraction

extraction implementation detailed above. We experimented with both the binary mask (PD) and the smoothed mask (PDCW); in each case, direct MFCC reconstruction provided a slight advantage over the corresponding algorithm with its orginial OLA synthesis. This is likely due to the small discontinuities created across time during the masking operation; if the windowing operation of the STFTs of the masker and subsequent feature extraction are not perfectly aligned, this will create small spectral differences between $Y[n,k]$ and the output of the feature extraction's STFT operation, which in turn will harm performance.

Comparing the results from Figures 4.1 and 4.4, we can see that direct feature extraction provides a more consistent performance boost than missing feature reconstruction, if a slightly lower one at high SIRs. As explained, direct feature extraction is also far less computationally costly. Given these advantages, focusing on alternative direct feature extraction methods seems to be the better investment of time and resources in improving the performance of maskers. Although MFCC features, on which we have based Figure 4.3, are widely used in speech recongition front-ends, there are a number of different features that have proven more robust to various adverse environmental conditions (*e.g.,* [33, 46, 47]). Of these, PNCC features are a prime candidate for direct feature extraction from masked speech. They have been shown to be high performing in noisy environments; they are also one of the feature types based on the gammatone filterbanks that have shown to improve performance when used as kernel functions for smoothing the binary masks (*i.e.,* channel weighting – see Section 2.3.1).

PNCC processing, described in [33] and detailed in [48], follows the same general flow of MFCC processing, but differs in the following ways:

- The frequency response of the ERB-based gammatone filters [29, 30] are used instead of MFCC's triangular mel-scale filters.
- A power-law nonlinearity is used instead of MFCC's logarithmic nonlinearity. In [33], the exponent is chosen to be $1/15$ after manual tuning; we have used the same.

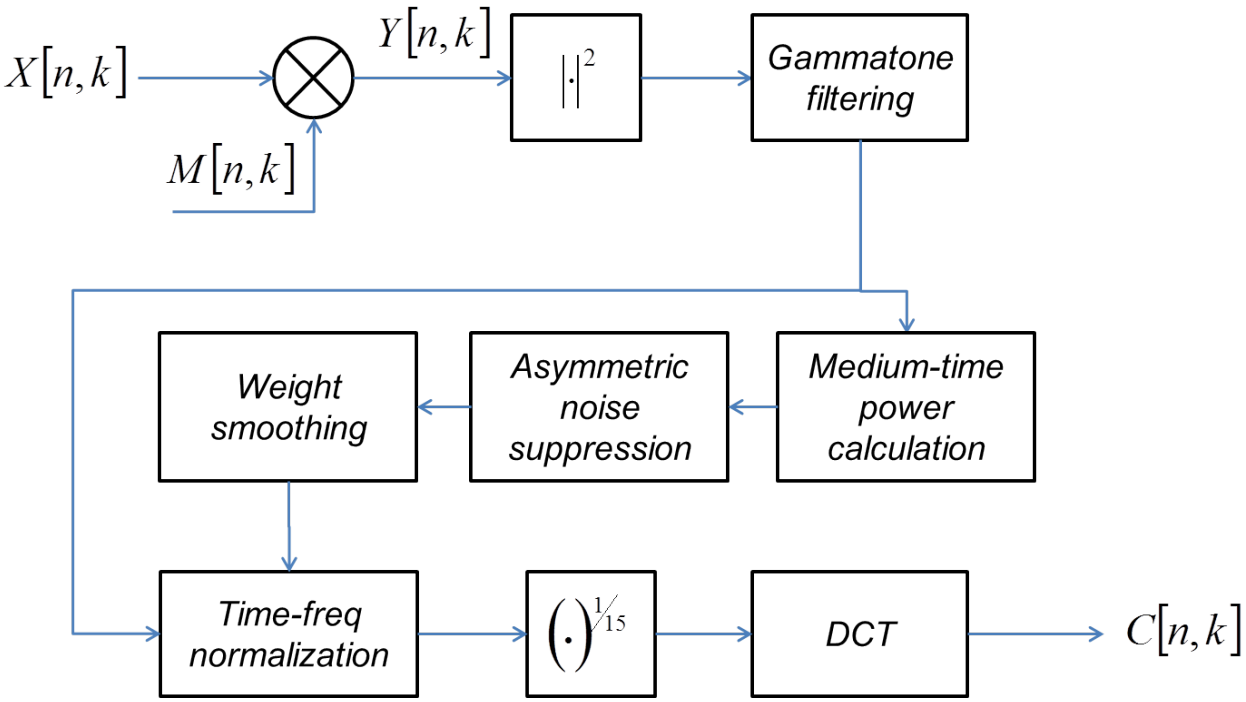


Figure 4.5: Direct PNCC-style feature extraction from the masked signal

- The introduction of intermediate subsystems between the two steps mentioned above. These subsystems include:
  - A calculation of average power over longer frame durations; we have used the same 80-ms duration also used in mask estimation.
  - "Asymmetric Noise Suppression", which uses an "asymmetric nonlinear filter" to estimate acoustical background noise levels over these longer durations, and a form of temporal masking that tracks signal envelope onsets, crudely mimicing the auditory system.
  - Mean power normalization, necessitated by the modification of the nonlinearity.

Building these subsystems into our masking algorithm as a direct feature extractor results in the system illustrated in Figure 4.5.

Figure 4.6 presents the results of using this technique in speech recognition experiments in the same scenario as Figure 4.1. The results of MFCC-style feature extraction (from

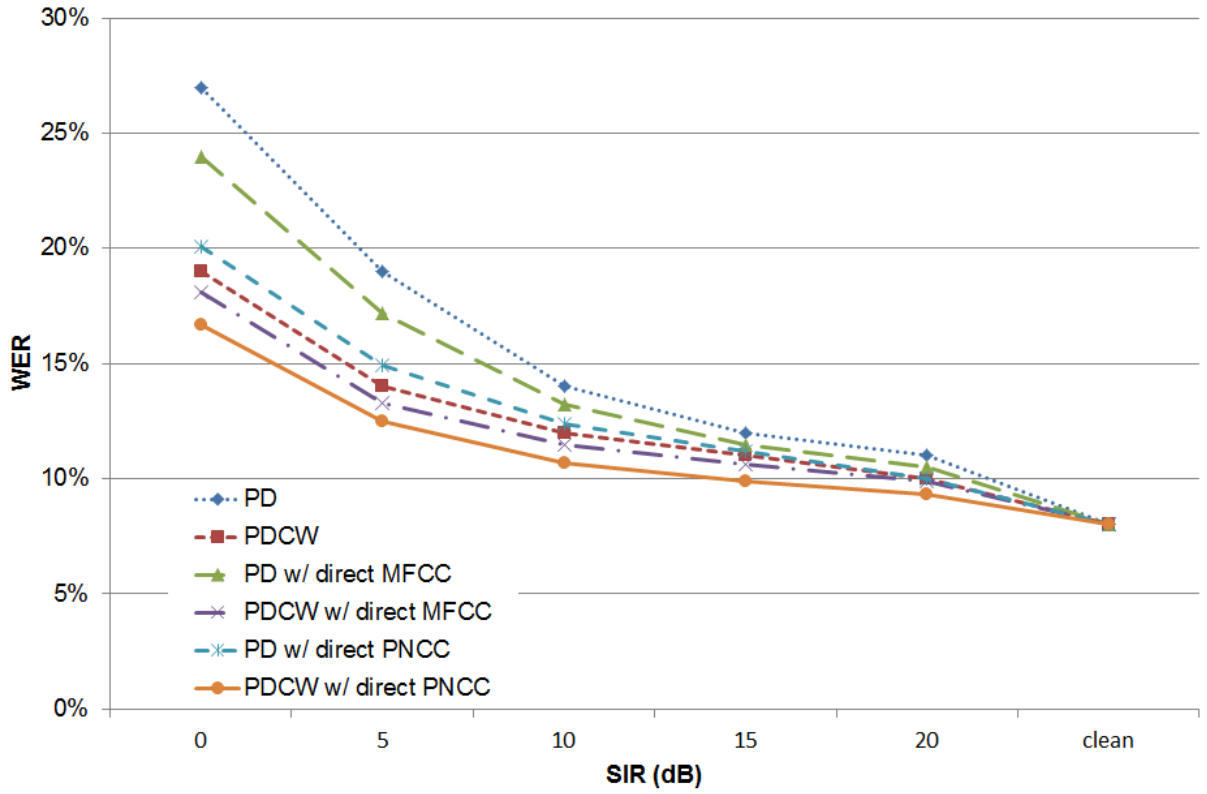


Figure 4.6: Word error rates (WER) of PD and PDCW with reconstruction vs. direct MFCC-style and PNCC-style feature extraction

Figure 4.4) are also included for ease of comparison.

As Figure 4.6 shows, PNCC-style feature extraction improves upon the performance of MFCC-style feature extraction. Note that at higher SIRs (such as 20 dB) the performance of PDCW masking with PNCC-style feature extraction approaches that of clean speech; *i.e.*, the gap is being closed. Lower SIRs, of course, still have a way to go. This improvement in performance is not unexpected; the motivation for using PNCC-style feature extraction in the first place was that PNCC features have been painstakingly designed to be more robust to noise and degradation than MFCC features. In fact, despite its relative complexity, PNCC is gaining wide recognition and acceptance as a preferred feature extraction method in situations where noise robustness is a factor.

4.3 Multi-channel masking

Linear beamforming techniques are generally well formulated and easily adaptable to various array geometries, including different numbers of microphones. Of course, array geometry does affect the characteristics and behavior of the array processing. In particular, increasing the array size (*i.e.,* number of sensors) increases the number of free parameters, which in turn allows for narrower beams, better sidelobe suppression and overall better performance – compare the sidelobes in Figure 2.4 (a four-element array) to Figure 3.9 (a two-element array). Masking algorithms, on the other hand, derive no such benefit from increasing the array size, in large part because the formulation is not as robust and there is not an obvious extension from two microphones to many. The remainder of this section explores options for scaling masking algorithms; this material forms the basis for [49].

4.3.1 Mask combination

In two-channel masking algorithms like PDCW, phase difference information from a pair of microphones is used to estimate the mask, which is then applied to the signal. Perhaps the most intuitive generalization of masking to a larger array would then be to apply the same procedure to each pair of microphones and combine the masks. In an array with P elements, there will be $\binom{P}{2}$ pairs. One option for mask combination is simple averaging:

$$M[n,k] = \frac{1}{\binom{P}{2}} \sum_{p=1}^{\binom{P}{2}} M_p[n,k] \tag{4.5}$$

where $M_p[n,k]$ is the mask estimated by the p-th pair. Note that now, with pairs at different locations, the target signal will not be on the broadside axis for each pair, which means that the cone of acceptance will have to be around some nonzero azimuth. Assuming that the target direction and array geometry are known, this target azimuth can be calculated for

each pair. Naming this quantity ϕ_p, (2.12) can be modified for this scenario as below:

$$M_p[n,k] = \begin{cases} 1 & \gamma(\omega_k; \phi_p - \phi_T) < \theta_p[n,k] < \gamma(\omega_k; \phi_p + \phi_T) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where the $\gamma(\omega;\phi)$ function is defined in (2.12). Conceptually (4.6), adjusts the phase thresholding of (2.12) to accept signals with a perceived DoA of at most ϕ_T away from the correct target direction ϕ_p, rather than zero. The averaged mask from (4.5) is then smoothed (see the discussion on channel weighting in Section 2.3.1) and applied to one of the input signals.

Unfortunately, it turns out that this approach is not particularly beneficial. For example, Figure 4.7 (red squares) illustrates the performance, in terms of WER, of the procedure outlined above when used in uniformly-spaced line arrays of different sizes. For comparison, we have also used adaptive beamforming (green triangles) on the same arrays; the beamformers are designed to have a response of unity in the target direction with adaptive sidelobe cancellation based on the MMSE criterion [5]. In all cases the element separation is 4 cm and the interferer is at $\phi = 60^\circ$ with an SIR of 10 dB. The threshold azimuth of the mask estimation is set at $\phi_T = 15^\circ$. To keep the comparison with linear beamformers fair, the environment is chosen to be reverberant, with a reverberation time of 200 ms. This is because adaptive beamforming can easily suppress a single interferer, at the expense of creating large sidelobes in other directions; the existence of reverberation precludes this type of solution, as large sidelobes in any direction are detrimental. The reverberation is simulated, using Habets' implementation of the image method [9], in a featureless room with dimensions of $8 \times 5 \times 3$ meters with a uniform surface absorption coefficient selected to result in the desired reverberation time. The beamformers are first allowed to converge and then the coefficients are used for the testing runs. The database and speech recognition details are as described in Section 3.3.4.

Figure 4.7 demonstrates the superiority of the scaling of linear beamforming. The reason is that the masks generated by the different microphone pairs are highly correlated with each

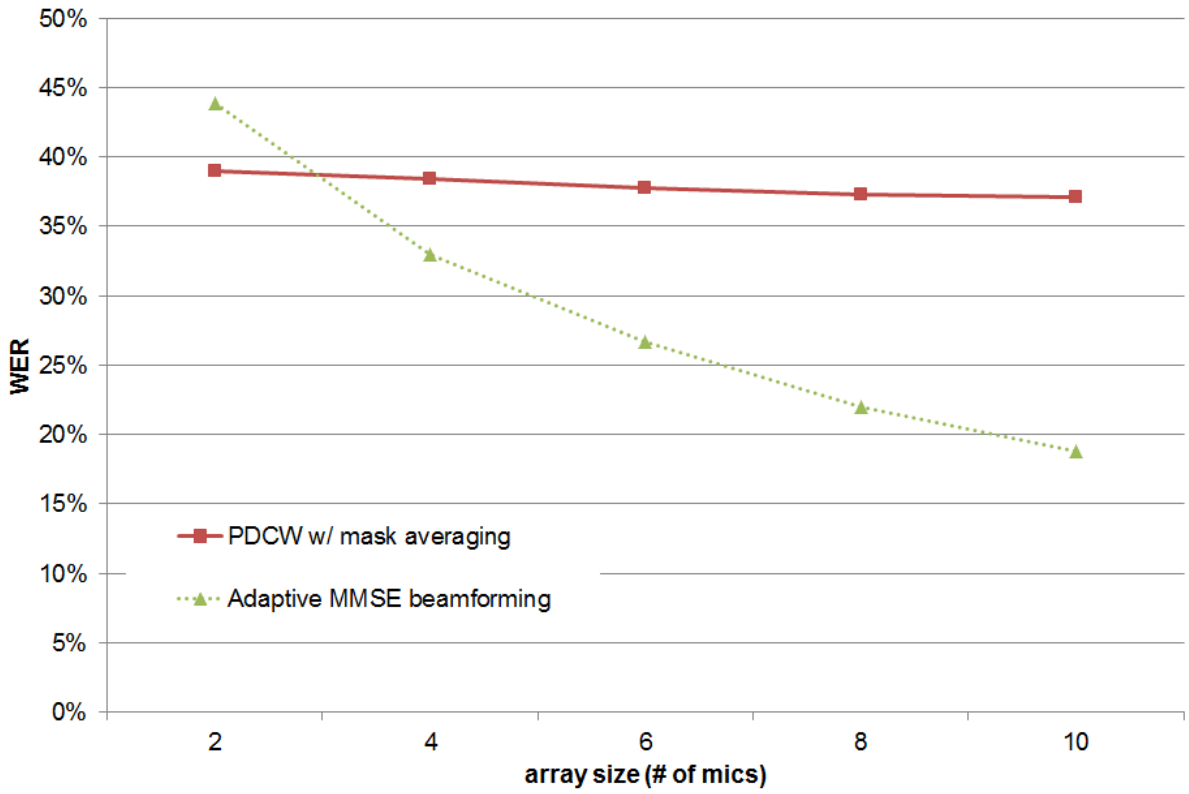


Figure 4.7: Word error rates (WER) of multi-channel PDCW with mask averaging vs. linear beamforming

other. To examine this, the correlation coefficient between the masks produced by any two microphone pairs can be calculated:

$$\rho_{pq} = \frac{\left\langle \left\langle \left(M_p\left[n,k\right]-\mu_p\right)\left(M_q\left[n,k\right]-\mu_q\right)\right\rangle_k \right\rangle_n}{\sqrt{\left\langle \left\langle \left(M_p\left[n,k\right]-\mu_p\right)^2\right\rangle_k \right\rangle_n \left\langle \left\langle \left(M_q\left[n,k\right]-\mu_q\right)^2\right\rangle_k \right\rangle_n}} \quad , \quad \mu_p = \left\langle \left\langle M_p\left[n,k\right]\right\rangle_k \right\rangle_n$$

For example, in the experiments described above, even when using the ten-microphone linear array, the correlation coefficients were all greater than 0.92. The implication is that the addition of extra pairs does little to change the masks generated by a single pair, which in turn leaves performance largely unaffected. This is hardly surprising; as discussed in Section 4.2.1, the mask estimation method is quite accurate to begin with. In arrays with different geometries (*e.g.,* with elements arranged around a circle), the situation does improve slightly but masking is still greatly eclipsed by beamforming.

4.3.2 Two-channel masking with sub-array beamformers

With the failure of mask combination, other methods must be sought to extend masking to multiple channels. One idea is to combine linear beamforming and two-channel masking by performing two-channel masking on signals other than sensor inputs. In an array with P elements, we divide the array into two (symmetric, if possible) segments, called "sub-arrays". A linear beamformer is designed and applied to each of these sub-arrays; for simplicity, if the sub-arrays can be chosen symmetrically the same set of beamforming filters is used for both. The outputs of the two arrays are then combined using basic two-channel masking. Figure 4.8 illustrates the general idea of this approach, on an array with six sensors.

Mathematically, the outputs of the sub-array beamformers can be expressed as (see (2.8))

$$\begin{cases} V_1\left(e^{j\omega}\right) = \sum\limits_{p\in A_1} X_p\left(e^{j\omega}\right) G_{p'}^{(1)}\left(e^{j\omega}\right) \\ V_2\left(e^{j\omega}\right) = \sum\limits_{p\in A_2} X_p\left(e^{j\omega}\right) G_{p'}^{(2)}\left(e^{j\omega}\right) \end{cases} \tag{4.7}$$

where $X_p\left(e^{j\omega}\right)$ is the received signal at the p-th sensor, p' represents the corresponding index

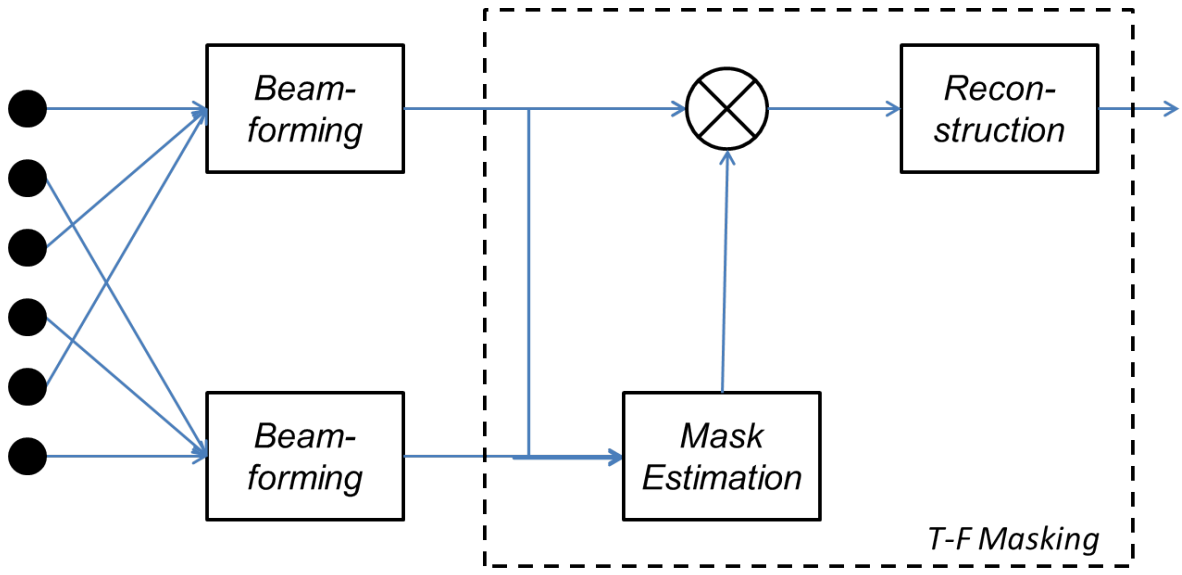


Figure 4.8: Masking with sub-array beamforming system

on the sub-array set and A_1 and A_2 represent the sub-array index sets:

$$A_1 \cap A_2 = \varnothing \quad \wedge \quad A_1 \cup A_2 = \{1, 2, \ldots, P\}$$

where P is the number of elements in the array. Mask estimation and application are then performed on $V_1[n,k]$ and $V_2[n,k]$, the STFTs of $V_1(e^{j\omega})$ and $V_2(e^{j\omega})$, as per Section 2.3.1.

There are a number of details that must be considered when implementing this idea. One is the geometry of the array and the selection of sub-array elements. We have not yet developed a systematic, optimal method of division, but have instead operated on a case-by-case basis. For example, for line arrays with even number of sensors, the sub-arrays are designated as per Figure 4.9. This way, the geometric separation between the two sub-arrays is equal to the separation between adjacent sensors. In (4.7), this means

$$A_1 = \{1, 3, 5\} \quad \wedge \quad A_2 = \{2, 4, 6\}$$

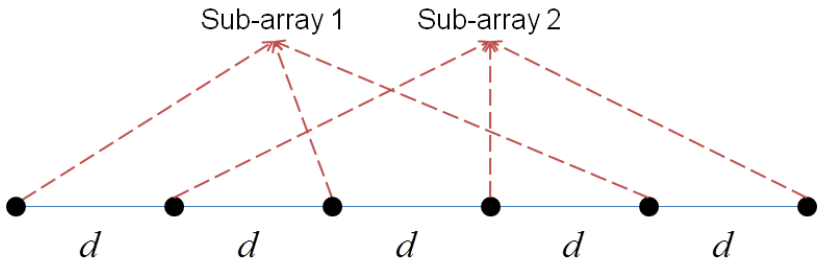


Figure 4.9: Staggered division of a six-element line array into sub-arrays

The next issue is sub-array beamformer design. The use of adaptive beamforming becomes difficult here, as adaptation in the presence of the masker is not straightforward and requires further study. For this reason, and due to the necessity of phase compensation mentioned below, we have elected to use fixed sub-array beamformers that have all been designed via adaptive beamforming in a stand-alone scenario and then applied to our test configurations. For example, in the configuration from Figure 4.9, a three-element array

with elements spaced at $2d$ is placed in the target configuration and an adaptive beamformer is allowed to converge. Because the subarray selection of Figure 4.9 is symmetric, and we are assuming the signal sources are in the far field, this beamformer is then used as the sub-array beamformer on both sub-arrays (*i.e.,* $G_{p'}^{(1)}\left(e^{j\omega}\right) = G_{p'}^{(2)}\left(e^{j\omega}\right) = G_{p'}\left(e^{j\omega}\right)$). Incorporating these specifications allows us to rewrite (4.7) as below:

$$\begin{cases} V_1\left(e^{j\omega}\right) = \sum_{p'=1}^{\frac{P}{2}} X_{2p'-1}\left(e^{j\omega}\right) G_{p'}\left(e^{j\omega}\right) \\ V_2\left(e^{j\omega}\right) = \sum_{p'=1}^{\frac{P}{2}} X_{2p'}\left(e^{j\omega}\right) G_{p'}\left(e^{j\omega}\right) \end{cases} \tag{4.8}$$

The remaining issue is phase compensation. In regular PDCW, the phase difference caused by a source at a given aziumth is simple to calculate (see (2.12)). Now, the signals will first be passed through a pair of beamformers; the phases at the various elements will be different and produce a phase difference between the outputs of sub-arrays 1 and 2 that will not necessarily follow (2.12). However, in the case of equally-spaced line arrays, since the corresponding sensors in the two sub-arrays are always exactly d apart, if the two beamformers are identical, the difference will still boil down to $\gamma\left(\omega;\phi\right)$. To demonstrate, we calculate $\gamma_A\left(\omega;\phi\right)$, the phase difference observed between $V_1\left(e^{j\omega}\right)$ and $V_2\left(e^{j\omega}\right)$ in the presence of a single hypothetical source at azimuth ϕ, under the assumptions made so far:

First, looking at Figure 4.9, we choose the midpoint of the array to be the origin of geometric coordinates and the phase reference for the received signal; *i.e.,* the incoming signal as observed at the origin will be named $S\left(e^{j\omega}\right)$. Assuming the sensors are numbered 1 through P from left to right, the position of sensor p will be $\left(p - \frac{P+1}{2}\right) d$. Thus, following a logic similar to that of Section 2.1.1, the difference in travel time between the path from source to midpoint and the path from source to sensor p is

$$\tau_p = \left(p - \frac{P+1}{2}\right) \frac{d}{c} \sin\phi$$

Hence, the signal received at each sensor will be a phase-shifted version of the signal as received at the midpoint:

$$x_p(t) = s(t-\tau_p) = s\left(t - \left(p - \frac{P+1}{2}\right)\frac{d}{c}\sin\phi\right)$$
$$\Rightarrow \quad X_p\left(e^{j\omega}\right) = S\left(e^{j\omega}\right) e^{j\omega f_S\left(\frac{P+1}{2}-p\right)\frac{d}{c}\sin\phi} \tag{4.9}$$

This expression can be compared to the phase shift experienced by the interferer at ϕ in (2.7). Substituting (4.9) back into (4.8), the outputs of the two sub-arrays become

$$\begin{cases} V_1\left(e^{j\omega}\right) = \sum_{p'=1}^{\frac{P}{2}} S\left(e^{j\omega}\right) e^{j\omega f_S\left(\frac{P+1}{2}-2p'+1\right)\frac{d}{c}\sin\phi} G_{p'}\left(e^{j\omega}\right) \\ V_2\left(e^{j\omega}\right) = \sum_{p'=1}^{\frac{P}{2}} S\left(e^{j\omega}\right) e^{j\omega f_S\left(\frac{P+1}{2}-2p'\right)\frac{d}{c}\sin\phi} G_{p'}\left(e^{j\omega}\right) \end{cases}$$

$$\Rightarrow \quad V_1\left(e^{j\omega}\right) = \sum_{p'=1}^{\frac{P}{2}} S\left(e^{j\omega}\right) e^{j\omega f_S\left(\frac{P+1}{2}-2p'\right)\frac{d}{c}\sin\phi} e^{j\omega f_S\frac{d}{c}\sin\phi} G_{p'}\left(e^{j\omega}\right)$$
$$= e^{j\omega f_S\frac{d}{c}\sin\phi} \sum_{p'=1}^{\frac{P}{2}} S\left(e^{j\omega}\right) e^{j\omega f_S\left(\frac{P+1}{2}-2p'\right)\frac{d}{c}\sin\phi} G_{p'}\left(e^{j\omega}\right)$$
$$= V_2\left(e^{j\omega}\right) e^{j\omega f_S\frac{d}{c}\sin\phi}$$

$$\Rightarrow \quad \gamma_A(\omega;\phi) = \omega f_S \frac{d}{c}\sin\phi = \gamma(\omega;\phi)$$

As a result, the same formulation can be used to obtain the threshold azimuth when processing the sub-array outputs with PDCW. In other geometries, this difference must be calculated on a case-by-case basis with knowledge of the beamforming filters and array geometry.

Perhaps a more intuitive view of this system, which corroborates the result above, is the following: The sub-array system can be viewed as identical to that of Figure 2.3, except that now each of the two elements has its own directivity pattern (*i.e.,* the beam pattern of the respective sub-array). We have further assumed that the sub-arrays are chosen symmetrically

and the beamformers assigned to the two are identical. Let us name this beam pattern $B_A(\omega;\phi)$ and assume it is normalized so that $B_A(\omega;0) = 1$ (*i.e.,* signals coming from the look direction are unaltered). Now we can determine the response of the sub-arrays to a single source at azimuth ϕ:

$$\begin{cases} V_1\left(e^{j\omega}\right) = S\left(e^{j\omega}\right) e^{j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} B_A(\omega;\phi) \\ V_2\left(e^{j\omega}\right) = S\left(e^{j\omega}\right) e^{-j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} B_A(\omega;\phi) \end{cases} \tag{4.10}$$

where signal phase is shifted at the input to each sub-array due to the spatial separation of the two sub-arrays by d (as above, the midpoint of the array is assumed to be the phase reference). Thus:

$$\begin{aligned} \gamma_A(\omega;\phi) &= \angle V_1\left(e^{j\omega}\right) - \angle V_2\left(e^{j\omega}\right) \\ &= \angle\left(S\left(e^{j\omega}\right) e^{j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} B_A(\omega;\phi)\right) - \angle\left(S\left(e^{j\omega}\right) e^{-j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} B_A(\omega;\phi)\right) \\ &= \angle S\left(e^{j\omega}\right) + \angle e^{j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} + \angle B_A(\omega;\phi) - \angle S\left(e^{j\omega}\right) - \angle e^{-j\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi} - \angle B_A(\omega;\phi) \\ &= \omega f_S \frac{1}{2}\frac{d}{c}\sin\phi - \left(-\omega f_S \frac{1}{2}\frac{d}{c}\sin\phi\right) = \omega f_S \frac{d}{c}\sin\phi = \gamma(\omega;\phi) \end{aligned}$$

Figure 4.10 (blue diamonds) shows the performance of this approach, compared to the mask combination method of Section 4.3.1 with the same array, simulated environment, database and recognizer. In these experiments, the beamformer used in both sub-arrays is designed beforehand in a standalone scenario via adaptive sidelobe cancellation based on the MMSE criterion; once the beamformer converges, the resulting coefficients are used in the sub-array system. As Figure 4.10 demonstrates, the use of sub-array beamformers greatly improves the scalability of masking, but it still falls somewhat short of linear beamforming. However, the "crossover point" where linear beamforming starts out-performing masking has now been moved up to about 4 sensors.

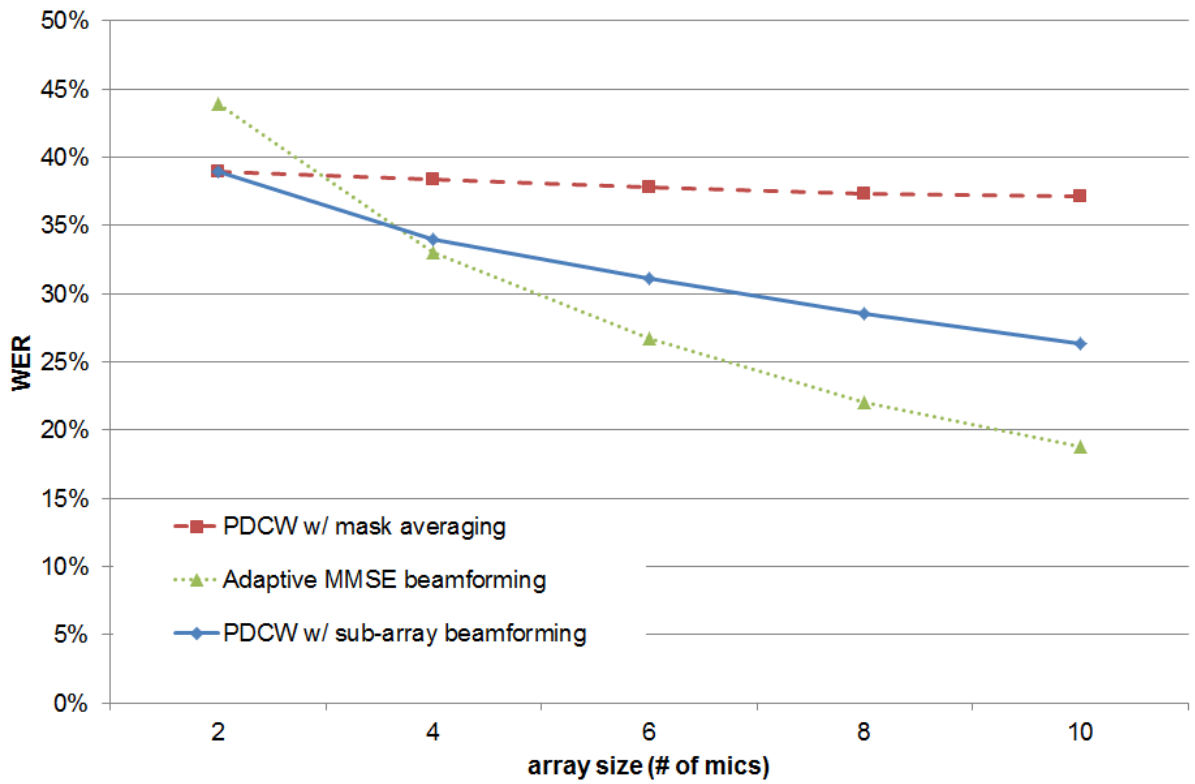


Figure 4.10: Word error rates (WER) of multi-channel PDCW with mask averaging and PDCW with sub-array beamforming vs. linear beamforming

4.3.3 Post-masking

The idea of a masking/beamforming hybrid introduced in Section 4.3.2 holds promise. The difference with linear beamforming, however, is still significant; especially so if we take into account the fact that there are many beamforming techniques that outperform the one used for comparison in Figure 4.10 [4, 5]. The truth is that the sub-array division approach suffers from two major weaknesses. The first is that the beamforming, operating at the sub-array level, does not make use of the full array size. The second is that the mask estimation is based on the outputs of the sub-arrays. Since the phase difference information has been distorted by the beamforming stage, the mask estimation will be using degraded data.

A different approach to the masking/beamforming hybrid potentially solves both these problems. The mask is estimated directly from the sensor inputs using the pairwise mask combination method of Sec. 4.3.1; *i.e.*, each possible pair of sensors produces a mask $M_p[n, k]$,

according to (4.6), which are then combined using (4.5) to produce a single mask $M[n,k]$. This mask is put aside, while all the signals are passed to a linear beamformer operating on the full array:

$$V\left(e^{j\omega}\right) = \sum_{p=1}^{P} X_p\left(e^{j\omega}\right) G_p\left(e^{j\omega}\right)$$

The mask is then smoothed according to the channel weighting discussed in [28] and mentioned in Section 2.3.1 (replace "X" in those equations with "V" as defined above) and applied to the output of the linear beamformer (a single channel):

$$Y[n,k] = V[n,k]\,\tilde{M}[n,k]$$

Figure 4.11 illustrates this approach, which will be named "post-masking" for the obvious parallels to the post-filtering techniques (*e.g.,* [50, 51, 52]) that inspired it. In post-filtering, the array inputs are used, pre-combination, to design an LTI filter which filters the output of a beamformer; in post-masking, the array inputs are used to estimate a T-F mask which is then applied to a beamformer's output.

Figure 4.12 (orange circles) shows the performance of this approach, compared to the

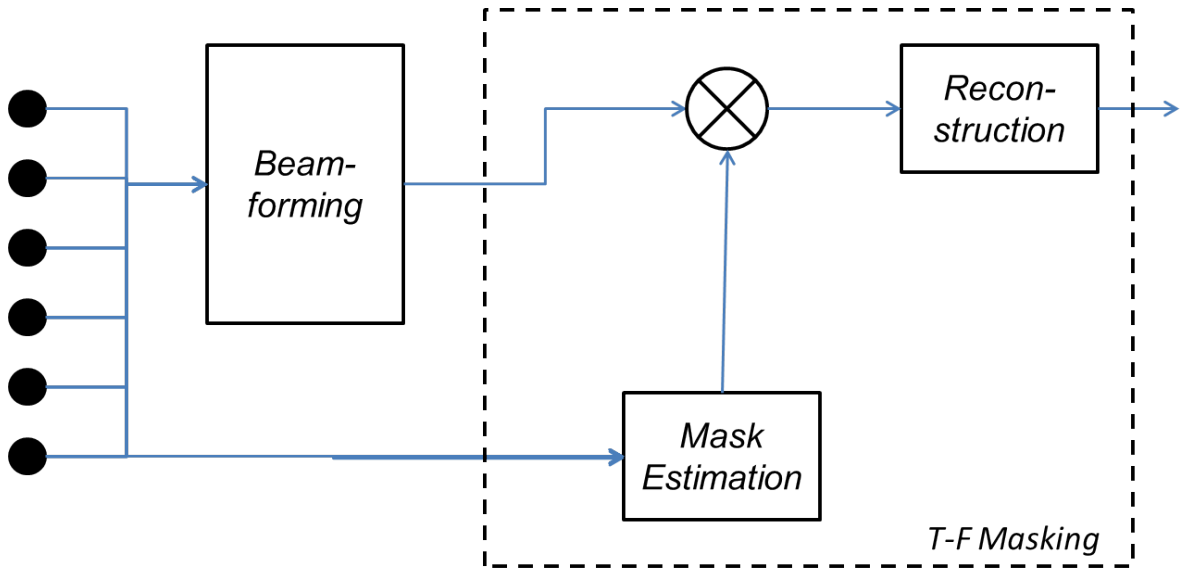


Figure 4.11: Beamforming with post-masking system

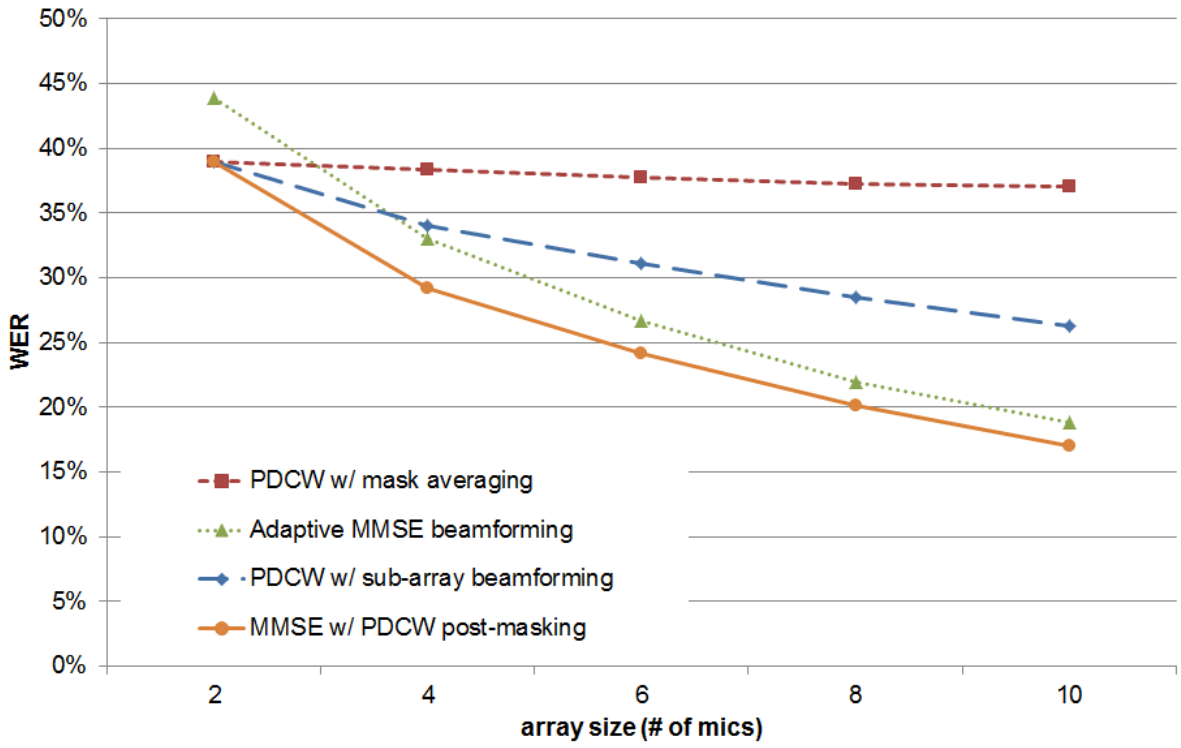


Figure 4.12: Word error rates (WER) of PDCW post-masking vs. sub-array beamforming and mask combination

methods of Sections 4.3.1 and 4.3.2. The array, simulated environment, database and recognizer are identical to the setup of those experiments. As was the case with the sub-array beamforming system, due to the complexity of the interplay between the adaptive convergence of the beamformer and the (time-varying) masking operation, the beamformers are first allowed to converge in a training run and then used in the post-masking system as constant beamformers. In fact, the beamformer used for the post-masker and for the straight beamformer (green triangles) are identical. As can be seen, the post-masking system outperforms the straight MMSE beamformer, although the gap closes as the number of sensors increases. Because the beamformers are identical, the difference between the green and orange lines can be interpreted as the contribution of the post-masking system. It seems that this contribution diminishes as the performance of the beamformer itself improves; this will be discussed in slightly more detail in Section 4.4.

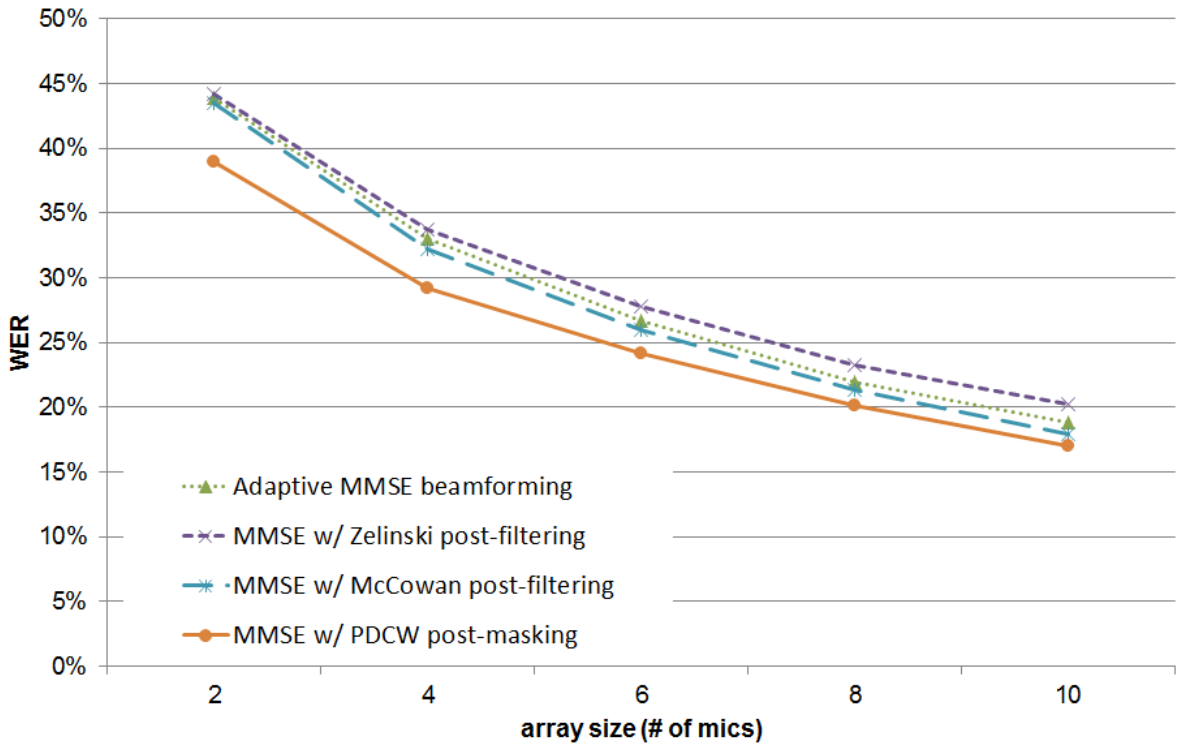


Figure 4.13: Word error rates (WER) of PDCW post-masking vs. Zelinski and McCowan post-filtering

For a more fair comparison, Figure 4.13 compares the post-masking system to the Zelinski [50] and McCowan [52] post-filters, operating with the same beamformers on the same data sets. The post-masker outperforms even the McCowan post-filter, albeit slightly, while the Zelinski post-filter lags behind the other systems – this is not unexpected, as the Zelinksi post-filter is designed for noise fields with characteristics not entirely descriptive of simulated reverberation.

As seen in Figures 4.12 and 4.13, the post-masking idea significantly improves on the performance both of simple mask combination with no beamforming and of sub-array beamforming (the first iteration of the masking-beamforming hybrid approach). The advantage over mask combination is clear; post-masking uses the same mask estimation and application steps, with the addition of a powerful linear beamformer in between. This linear beamformer accounts for most of the gain of post-masking over mask combination. The advantage over

sub-array beamforming, on the other hand, is slightly more subtle. As explained at the beginning of this section, in sub-array beamforming the beamformers each utilize only half of the available elements; the beamformer in post-masking, on the other hand, operates on the full array. Given the scaling of beamforming, this is a serious advantage in and of itself – compare the performance of the beamformers with two, four and eight elements in Figure 4.12. Furthermore, the mask estimation in sub-array beamforming is done after the signals have already been processed by the beamformers. Since the beamformers themselves make use of phase difference in their processing, the phase differences on the outputs of the beamformers is a distorted version of the input phase difference; some of the accuracy of the estimation is lost in the transition. In post-masking, on the other hand, the input signals are used directly to estimate the mask before any phase distortion the beamforming may introduce.

4.4 Conclusions

In this chapter, we approached the problem of improving the performance of a typical T-F masking algorithm from a number of different angles. One area of focus was mask estimation; Section 4.2.1 showed that the current mask estimation is quite accurate, the caveat being that this statement is limited to non-reverberant environments. Reverberant environments, however, are less straightforward to model; there is also more room for improvement in terms of ASR performance. Recent studies [21] have shown that, in reverberant environments, oracle binary masking using an SIR threshold (see (4.4)) greater than 0 dB significantly improves human intelligibility: The higher the nominal input SIR, the higher the optimal threshold SIR. If this result carries over to ASR, it could be used to improve the performance of masking in reverberation. Using simulated data, where the clean signal and interferer are known and the nominal SIR is tunable, a brute-force search can be attempted to determine the optimal SIR threshold for a given input SIR in various levels of reverberation, similar to

the experiments outlined toward the end of Section 4.2.1.

An alternative approach to improving mask estimation in reverberation is to look at criteria other than phase difference. For example, the precedence effect [4] allows humans to localize signals even in the presence of reverberation, and many algorithms, such as [53], have emulated that functionality to enhance single-channel reverberated speech for ASR. Such a criterion could also be used to emphasize mask estimation; for example, the phase difference calculated in T-F cells that are determined to contain early echoes could be stored and combined with the phase difference calculated in the following cells to temper the masks estimated therein. This is, however, potentially quite a complicated problem.

Another area of focus was reconstruction of masked spectrograms. We discussed two alternate approaches to OLA-style reconstruction in Section 4.2.2: resconstruction using a missing-feature reconstruction technique (specifically, cluster-based reconstruction) and the avoidance of reconstruction altogether in favor of direct feature extraction. Given the results of cluster-based reconstruction vs. direct MFCC-style feature extraction, it seems more reasonable to focus further improvement efforts on direct feature extraction. With the introduction of PNCC-style feature extraction, we have improved the performance of masking even further, at the cost of increased computational complexity. This increase in complexity, however, is still dwarfed by the ensuing speech recognition operation.

As for the competition with linear beamforming, with two microphones, masking already outperforms beamforming; the combination of improvements in feature extraction and mask estimation mentioned above can widen that gap and establish masking as the go-to method when processing the inputs from a microphone pair. This is, in itself, significant; the abundance of stereo microphones and recordings and the expense of constructing larger arrays make two-channel masking a prevalent use case. However, the performance of two-channel masking in the more adverse environments (*e.g.,* low SIR and/or heavy reverberation) is still unsatisfactory; while linear beamforming can compensate for this by using larger arrays, masking does not scale properly with array size.

Section 4.3 explored approaches to improve the scalability of masking algorithms. Using PDCW as a representative case of two-channel time-frequency masking algorithms, we have demonstrated that this type of algorithm does not easily generalize to arrays of more than two elements. However, masking can be combined with linear beamforming, which does scale well to large arrays, to reap the benefits of T-F masking in these scenarios. Specifically, using the novel post-masking system, we have successfully used T-F masking to enchance the performance of a linear beamformer in arrays of up to ten elements. This post-masking system is also shown to be competitive with the post-filtering techniques that partially inspired it.

Now that these intial results have revealed the potential of post-masking, there is much room for improvement. The question of optimal mask estimation for post-masking, for one, is far from settled. While the method described in (4.6) does indeed estimate (2.9) relatively accurately, it is not certain that (2.9) itself is a good target when using post-masking. The linear beamformer in post-masking changes the SIR, so that on the beamformer's output the mask is likely far too conservative; *i.e.,* too many cells are rejected. This, in turn, could be the reason that the added benefit of this post-masking technique diminishes in larger arrays; the better the beamformer, the less realistic the oracle mask. Given knowledge of the beamformer's beam pattern, the mask estimation stage of post-masking can be modified to estimate a mask of the form discussed in Section 4.2.1 (and presented in (4.4)), where the SIR threshold is set differently at different frequencies, in order to estimate the optimal mask at the *output* of the beamformer, rather than the input. Moving forward, this seems to be a reasonable first avenue of investigation.

Chapter 5

Summary of Contributions

Chapters 3 and 4 described the two main thrusts of this work. The first is to develop analysis methods that will help explain the behavior of masking algorithms more intuitively and ease comparison and contrast between nonlinear and linear array processing techniques. The second is to improve the performance of masking algorithms in order to make them more competitive with linear beamforming. This chapter will summarize the main points, results and conclusions of the previous two.

5.1 Analysis

Chapter 3 detailed the development of a model for the analysis of the behavior of masking algorithms. The model produces intuitive, descriptive metrics – based loosely on the concept of the beam patterns of linear beamformers – that can be viewed as *nonlinear beam patterns*. Beginning with the single target, single interferer scenario (see Section 2.1.1), these beam patterns are set to be functions of frequency, interferer location and input SIR. Two different quantities are mapped as functions of these free variables, leading to two different versions of the nonlinear beam pattern: the *mask presence pattern* $B_M(f;\phi;SIR)$ and the *output noise pattern* $B_N(f;\phi;SIR)$. The patterns are calculated by examining the sources of randomness in the mask estimation process at the level of individual time-frequency cells. Each source is

modeled as a random variable with an appropriate distribution, which are then appropraitely transformed and averaged to produce the patterns above. It turns out that, in addition to being intuitive, these metrics are relatively predictive of the performance of the algorithm under analysis.

Although the initial model is designed for a limited use case, extensions can also be developed. To demonstrate this, several other scenarios are considered and the model is adapted to analyze the behavior of the masking algorithm in each one. These scenarios include:

- Different signal types (Section 3.4.1)
- Independent noise at the sensors (Section 3.4.2), which produces patterns of the form $B\left(f; SNR\right)$
- Multiple interfering sources (Section 3.4.3); with one target signal and L interferers, this produces patterns of the form $B\left(f; \phi_1, \phi_2, \ldots, \phi_L; SIR_1, SIR_2, \ldots, SIR_L\right)$
- Reverberant environments (Section 3.4.4), producing patterns of the form $B\left(f; RT_{60}\right)$

The *nonlinear beam patterns* developed herein represent a novel method for the analysis and visualization of the behavior of masking algorithms. Since the analysis is based solely on probabilistic models, it alleviates the need for costly and time-consuming large-scale data processing. It allows masking systems to be studied independently of signal database and other processing steps, including the pattern recognition task (*e.g.,* speech recognition) for which these algorithms typically form the front end – factors that introduce a great deal of variability into the current analyses and comparisons of masking techniques. These models and the patterns they produce can be used to compare different variations of masking – *e.g.,* they can be used to tune the parameter(s) of an algorithm before any actual data is processed. They can also be used to compare the behvavior of a specific algorithm in different circumstances – compare Figures 3.6 through 3.8 or the various scenarios depicted in Figure 3.29.

5.2 Improvements

Chapter 4 detailed attempts to improve the performance of masking algorithms, in an effort to make them more competitive with linear beamforming approaches. The problem is approached from a number of different angles. One area of focus is mask estimation; it is established (Section 4.2.1) that the current mask estimation method is quite accurate, the caveat being that this statement is limited to non-reverberant environments.

Another area of focus is the reconstruction of masked spectrograms. Two alternate approaches to OLA-style reconstruction are discussed in Section 4.2.2: resconstruction using a missing-feature reconstruction technique (specifically, cluster-based reconstruction) and the avoidance of reconstruction altogether in favor of direct feature extraction. It is argued that while cluster-based reconstruction does provide a boost at higher input SIR values (when the masking is less intrusive), this boost is inconsistent and comes at the cost of a greatly increased complexity. On the other hand, direct MFCC-style feature extraction is both computationally simpler and more consistent in its gain over the baseline. Given these points, it seems more reasonable to focus further improvement efforts on direct feature extraction. With the introduction of PNCC-style feature extraction, the performance of masking is consistently improved even further, at the cost of somewhat increased computational complexity. This increase in complexity, however, is still dwarfed by the speech recognition operation.

As for the competition with linear beamforming, with two microphones, masking already outperforms beamforming; the combination of improvements in feature extraction and mask estimation mentioned above can widen that gap and establish masking as the go-to method when processing the inputs from a pair of microphones. This is, in itself, significant; the abundance of stereo mics and recordings and the expense of larger arrays make two-channel masking a prevalent use case. It would not be surprising to see spectro-temporal masking become the dominant approach in two-channel scenarios in the near future. However, the performance of two-channel masking in the more adverse environments (*e.g.,* low SIR and/or heavy reverberation) is still unsatisfactory; while linear beamforming can compensate for this

by using larger arrays, masking does not scale properly with array size.

Improving the scalability of masking algorithms is studied in Section 4.3. Using PDCW as a representative case of two-channel time-frequency masking algorithms, it is demonstrated that the this type of algorithm does not easily generalize to arrays of more than two elements. However, masking can be combined with linear beamforming, which does scale well to large arrays, to reap the benefits of T-F masking in these scenarios. Specficially, using the novel *post-masking* system, we successfully use T-F masking to enhance the performance of a linear beamformer in arrays of up to ten elements, even if the margin of performance gain diminishes as the array size grows. To our knowledge, this is the first masking-based technique that outperforms conventional adaptive beamforming techniques in larger arrays. While there are other beamforming algorithms that outperform the post-masking system as described in Section 4.3.3, we are hopeful that, with improvements to the mask estimation of post-masking (see Section 5.3), the idea can be used in conjunction with any beamforming technique, no matter how powerful its standalone behavior. This is a major practical benefit of using a hybrid masking-beamforming approach; it benefits from advances in beamforming technology as well.

5.3 Potential directions for future work

Both thrusts (the analysis and the performance improvement) provide opportunities for continued investigation. On the analysis side, perhaps the most pressing area of inquiry is the definition of the output noise metrics (*e.g.,* (3.9) or (3.26)). Currently, these are chosen to be simple and intuitive. It could well be, however, that simple addition of the distortion terms in these equations is not the optimal combination when considering predictive accuracy. Furthermore, either the current noise measure or a more optimal one can be applied as a cost function to blindly optimize the parameters of a masking algorithm.

Another potential area of inquiry is more practical in nature: an examination of the

tradeoffs between complexity and accuracy. Although the model is, in most cases, computationally simple, the dependence on averaging over joint distributions does create the potential for great complexity when multiple variables are involved (as was discussed in Section 3.4.4). It could well be that the number of necessary calculations can be reduced in an intelligent way that minimizes the impact on the descriptive power and predictive accuracy of the resulting nonlinear beam patterns.

On the improvement side, beginning with mask estimation, there does seem to be room for improvement in reverberant environments. Recent studies [21] have shown that, in reverberant environments, oracle binary masking using an SIR threshold (see (4.4)) greater than 0 dB significantly improves human intelligibility: the higher the nominal input SIR, the higher the optimal threshold SIR. If this result carries over to ASR, it could be used to improve the performance of masking in reverberation. Experiments similar to those outlined toward the end of Section 4.2.1 can be devised to examine this question. An alternative approach to improving mask estimation in reverberation is to look at criteria other than phase difference. For example, the precedence effect [4] allows humans to localize signals even in the presence of reverberation, and many algorithms, such as [53], have emulated that functionality to enhance single-channel reverberated speech for ASR. Such a criterion could also be used to emphasize mask estimation.

As for the question of masking in larger arrays, the results from Section 4.3 have revealed the potential of combining masking and beamforming into a hybrid system. Even though this *post-masking* approach is already quite useful, there is much room for improvement. The question of optimal mask estimation for post-masking, for one, is far from settled. While the mask estimation method in use accurately estimates the oracle mask chosen as its target, it is not certain that this target, itself, is optimal. The linear beamformer in post-masking changes the SIR, so that on the beamformer's output the mask is likely far too conservative; *i.e.,* too many cells are rejected. This, in turn, could be the reason that the added benefit of this post-masking technique diminishes in larger arrays; the better the beamformer, the

less realistic the oracle mask. Moving forward, this seems to be a reasonable first avenue of investigation.

A potential solution to this problem is to allow for different mask estimation decisions at different frequencies. Given knowledge of the beamformer's beam pattern, the mask estimation stage of post-masking can be modified to estimate a mask of the form discussed in Section 4.2.1 (and presented in (4.4)), where the SIR threshold is set differently at different frequencies, in order to estimate the optimal mask at the *output* of the beamformer, rather than the input. In the context of PDCW, this might take the form of replacing the scalar azimuth threshold ϕ_T in (2.12) with a function of frequency $\phi_T(\omega)$; this would create a different cone of acceptance at each frequency. A new problem will then be the design of this frequency-dependent threshold. One intuitive starting point is to set the azimuth threshold at each frequency according to the beam width of the underlying beamformer. The narrower the beam width of the beamformer, the greater the suppression of the interference is likely to be. Thus, a more permissive decision threshold at the beamformer's input – *i.e.,* a larger ϕ_T – will likely produce a more realistic mask, pointing to an inverse relationship between the beam width and the ideal width of the mask estimator's cone of acceptance. Of course, this is a very qualitative argument, but one that might provide the basic idea behind a greatly improved version of post-masking.

Bibliography

[1] X. Huang, A. Acero, H.-W. Hon *et al.*, *Spoken language processing*. Prentice Hall PTR New Jersey, 2001.

[2] W. A. Yost, "The cocktail party problem: Forty years later," in *Binaural and spatial hearing in real and virtual environments*, R. H. Gilkey and T. R. Anderson, Eds. Lawrence Erlbaum Associates, Inc, 1997, pp. 329–347.

[3] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[4] G. Brown and D. Wang, *Computational Auditory Scene Analysis*, G. Brown and D. Wang, Eds. Hoboken, NJ: IEEE Press/Wiley-Interscience, 2006.

[5] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Optimum Array Processing*. John Wiley & Sons, 2004.

[6] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[7] L. L. Beranek, *Acoustics*. The Acoustical Society of America, 1954.

[8] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, p. 943, 1979.

[9] E. Habets, "Room impulse response generator," 2006.

[10] J. Flanagan, J. Johnston, R. Zahn, and G. Elko, "Computer-steered microphone arrays for sound transduction in large rooms," *The Journal of the Acoustical Society of America*, vol. 78, p. 1508, 1985.

[11] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, 1996.

[12] W. Herbordt and W. Kellermann, "Adaptive beamforming for audio signal acquisition," in *Adaptive Signal Processing*. Springer, 2003.

[13] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 489–498, 2004.

[14] K. Kumatani, J. McDonough, B. Rauch, D. Klakow, P. N. Garner, and W. Li, "Beamforming with a maximum negentropy criterion," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 5, pp. 994–1008, 2009.

[15] R. M. Stern, E. Gouvêa, and G. Thattai, "'Polyaural' array processing for automatic speech recognition in degraded environments," in *Interspeech 2007*, Antwerp, Belgium, August 2007.

[16] R. M. Stern and C. Trahiotis, "The role of consistency of interaural timing over frequency in binaural lateralization," *Auditory physiology and perception*, pp. 547–554, 1992.

[17] K. J. Palomäki, G. J. Brown, and J. Barker, "Missing data speech recognition in reverberant conditions," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. I–65.

[18] M. L. Seltzer, B. Raj, and R. M. Stern, "A bayesian classifier for spectrographic mask estimation for missing feature speech recognition," *Speech Communication*, vol. 43, no. 4, pp. 379–393, 2004.

[19] A. Narayanan and D. Wang, "Robust speech recognition from binary masks," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. EL217–EL222, November 2010.

[20] O. Hazrati, J. Lee, and P. C. Loizou, "Blind binary masking for reverberation suppression in cochlear implants," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1607–1614, March 2013.

[21] N. Roman and J. Woodruff, "Speech intelligibility in reverberation with ideal binary masking: Effects of early reflections and signal-to-noise ratio threshold," *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1707–1717, March 2013.

[22] G. Shi and P. Aarabi, "Robust digit recognition using phase-dependent time-frequency masking," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–684.

[23] K. J. Palomäki, G. J. Brown, and D. Wang, "A binaural processor for missing data speech recognition in the presence of noise and small-room reverberation," *Speech Communication*, vol. 43, no. 4, pp. 361–378, 2004.

[24] D. Wang, "On ideal binary mask as the computational goal of auditory scene analysis," *Speech separation by humans and machines*, vol. 60, pp. 63–64, 2005.

[25] S. Harding, J. Barker, and G. J. Brown, "Mask estimation for missing data speech recognition based on statistics of binaural interaction," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 58–67, 2006.

[26] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006.

[27] H.-M. Park and R. M. Stern, "Spatial separation of speech signals using amplitude estimation based on interaural comparisons of zero crossings," *Speech Communication*, vol. 51, pp. 15–25, January 2009.

[28] C. Kim, K. Kumar, B. Raj, and R. M. Stern, "Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain," in *Interspeech 2009*, Brighton, UK, September 2009.

[29] M. Slaney, "An efficient implementation of the Patterson-Holdsworth auditory filter bank," *Apple Computer, Perception Group, Tech. Rep*, 1993.

[30] B. C. J. Moore and B. R. Glasberg, "A revision of Zwicker's loudness model," *Acta Acustica united with Acustica*, vol. 82, no. 2, pp. 335–345, 1996.

[31] W. Kim and R. M. Stern, "Band-independent mask estimation for missing-feature reconstruction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 2006.

[32] R. M. Stern, E. Gouvêa, C. Kim, K. Kumar, and H.-M. Park, "Binaural and multiple-microphone signal processing motivated by auditory perception," in *HSCMA Joint Workshop on Hands-free Speech Communication and Microphone Arrays*, Trento, Italy, May 2008.

[33] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012.

[34] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[35] A. R. Moghimi and R. M. Stern, "An analysis of binaural spectro-temporal masking as nonlinear beamforming," in *ICASSP2014 - Audio and Acoustic Signal Processing (ICASSP2014 - AASP)*, Florence, Italy, May 2014.

[36] L. L. Scharf, *Statistical signal processing*. Addison-Wesley Reading, MA, 1991, vol. 98.

[37] H. Nélisse and J. Nicolas, "Characterization of a diffuse field in a reverberant room," *The Journal of the Acoustical Society of America*, vol. 101, no. 6, pp. 3517–3524, 1997.

[38] B. Rafaely, "Spatial-temporal correlation of a diffuse sound field," *The Journal of the Acoustical Society of America*, vol. 107, no. 6, pp. 3254–3258, 2000.

[39] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*. Prentice-hall Englewood Cliffs, 1978, vol. 100.

[40] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*, 3rd ed. Upper Saddle River, NJ: Pearson Education, Inc., 2010.

[41] P. Aarabi and G. Shi, "Phase-based dual-microphone robust speech enhancement," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 4, pp. 1763–1773, August 2004.

[42] T. May, S. van de Par, and A. Kohlrausch, "A binaural scene analyzer for joint localization and recognition of speakers in the presence of interfering noise sources and reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2016–2030, September 2012.

[43] C. Kim, R. M. Stern, K. Eom, and J. Lee, "Automatic selection of thresholds for signal separation algorithms based on interaural delay," in *Interspeech 2010*, Makuhari, Japan, September 2010.

[44] B. Raj, M. L. Seltzer, and R. M. Stern, "Reconstruction of missing features for robust speech recognition," *Speech Communication Journal*, vol. 43, no. 4, pp. 275–296, September 2004.

[45] B. Raj, "Reconstruction of incomplete spectrograms for robust speech recognition," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, April 2000.

[46] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[47] H. Hermansky and N. Morgan, "RASTA processing of speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 578–589, 1994.

[48] C. Kim, "Signal processing for robust speech recognition motivated by auditory processing," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, December 2010.

[49] A. R. Moghimi, B. Raj, and R. M. Stern, "Post-masking: A hybrid approach to array processing for speech recognition," in *Interspeech 2014*, Singapore, September 2014.

[50] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 2578–2581.

[51] I. A. McCowan and H. Bourlard, "Microphone array post-filter for diffuse noise field," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 905–908.

[52] ——, "Microphone array post-filter based on noise field coherence," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, no. 6, pp. 709–716, 2003.

[53] C. Kim and R. M. Stern, "Nonlinear enhancement of onset for robust speech recognition," in *Interspeech 2010*, Makuhari, Japan, September 2010.