Subband Beamforming with Higher Order Statistics for Distant Speech Recognition

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät II
- Physik und Mechatronik -
der Universität des Saarlandes

von

Kenichi Kumatani

Saarbrücken

2010

Tag des Kolloquiums: 21.10.2010

Dekanin/Dekan: Univ.-Prof. Dr. rer. nat. H. Seidel

Mitglieder des

Prüfungsausschusses: Univ.-Prof. Dr.-Ing. C. Xu

: Univ.-Prof. Dr. rer. nat. D. Klakow

: Univ.-Prof. Dr. tech. R. Dyczij-Edlinger

: Dr.-Ing. F. Felgner

Acknowledgements

I would like to thank my supervisor Dr. John McDonough for his guidance and support throughout my doctoral studies. His overall view on research and his high-quality work standards have deeply influenced my work, not to mention his positive attitude. Thanks to him, my English skills including some slang expressions have been much improved. It is an absolute pleasure to work with him. I always admire his expertise.

I would also like to thank Prof. Dietrich Klakow for giving me the opportunity of joining Spoken Language Systems at Saarland University and for encouraging me to finish this doctoral thesis. I am deeply grateful to him for giving me the possibility to pursue my interest in the distant speech recognition.

I am much obliged to Dr. Matthias Wölfel for giving me important advice about my work as well as private life many times. His great work has inspired me a lot. Thanks is due also to Matthias's wife Irina Wölfel.

I would also like to thank great colleagues Barbara Rauch and Friedrich Faubel for improving my work and inspiring me at Saarland University. I am much obliged to Tobias Gehrig, Uwe Mayer and Emilian Stoimenov who helped me at University of Karlsruhe.

I am grateful to Prof. Hervé Bourlard for giving me the opportunity to study distant speech recognition at Idiap. Thanks is also due to Philip N. Garner, Dr. Weifeng Li and Dr. John Dines for supporting me at Idiap. I would furthermore like to thank Prof. Satoshi Nakamura for his recommendation to work with the Interactive Systems Laboratories at University of Karlsruhe. He also guided me

to the field of automatic speech recognition.

Finally, there is no way I would be where I am today without my parents. I am much obliged to my mother Tamiko Kumatani and my father Takeo Kumatani who have raised me. I do hope my mother is proud of me in heaven. Thanks is also due to my sister Naoko Kumatani and my brother Hayato Kumatani.

Summary

This dissertation presents novel beamforming methods for distant speech recognition (DSR). Such techniques can relieve users from the necessity of putting on close talking microphones. DSR systems are useful in many applications such as humanoid robots, voice control systems for automobiles, automatic meeting transcription systems and so on.

A main problem in DSR is that recognition performance is seriously degraded when a speaker is far from the microphones. In order to avoid the degradation, noise and reverberation should be removed from signals received with the microphones.

Acoustic beamforming techniques have a potential to enhance speech from the far field with little distortion since they can maintain a distortionless constraint for a look direction. In beamforming, multiple signals propagating from a position are captured with multiple microphones. Typical conventional beamformers then adjust their weights so as to minimize the variance of their own outputs subject to a distortionless constraint in a look direction. The variance is the average of the second power (square) of the beamformer's outputs. Accordingly, it is considered that the conventional beamformer uses *second order statistics* (SOS) of the beamformer's outputs.

The conventional beamforming techniques can effectively place a null on any source of interference. However, the desired signal is also canceled in reverberant environments, which is known as the *signal cancellation* problem. To avoid that problem, many algorithms have been developed. However, none of the

algorithms can essentially solve the signal cancellation problem in reverberant environments.

While many efforts have been made in order to overcome the signal cancellation problem in the field of acoustic beamforming, researchers have addressed another research issue with the microphone array, that is, blind source separation (BSS) [1]. The BSS techniques aim at separating sources from the mixture of signals without information about the geometry of the microphone array and positions of sources. It is achieved by multiplying an un-mixing matrix with input signals. The un-mixing matrix is constructed so that the outputs are stochastically independent. Measuring the stochastic independence of the signals is based on the theory of the independent component analysis (ICA) [1].

The field of ICA is based on the fact that distributions of information-bearing signals are not Gaussian and distributions of sums of various signals are close to Gaussian. There are two popular criteria for measuring the degree of the *non-Gaussianity*, namely, kurtosis and negentropy. As described in detail in this thesis, both criteria use more than the second moment. Accordingly, it is referred to as *higher order statistics* (HOS) in contrast to SOS.

HOS is not considered in the field of acoustic beamforming well although Arai et al. showed the similarity between acoustic beamforming and BSS [2]. This thesis investigates new beamforming algorithms which take into consideration higher-order statistics (HOS). The new beamforming methods adjust the beamformer's weights based on one of the following criteria:

- minimum mutual information of the two beamformer's outputs,
- maximum negentropy of the beamformer's outputs and
- maximum kurtosis of the beamformer's outputs.

Those algorithms do not suffer from the signal cancellation, which is shown in this thesis. Notice that the new beamforming techniques can keep the distortionless constraint for the direction of interest in contrast to the BSS algorithms.

The effectiveness of the new techniques is finally demonstrated through a series of distant automatic speech recognition experiments on *real* data recorded with *real* sensors unlike other work where signals artificially convolved with measured impulse responses are considered. Significant improvements are achieved by the beamforming algorithms proposed here.

Zusammenfassung

Diese Dissertation präsentiert neue Methoden zur Spracherkennung auf Entfernung. Mit diesen Methoden ist es möglich auf Nahbesprechungsmikrofone zu verzichten. Spracherkennungssysteme, die auf Nahbesprechungsmikrofone verzichten, sind in vielen Anwendungen nützlich, wie zum Beispiel bei Humanoiden-Robotern, in Voice Control Systemen für Autos oder bei automatischen Transcriptionssystemen von Meetings.

Ein Hauptproblem in der Spracherkennung auf Entfernung ist, dass mit zunehmendem Abstand zwischen Sprecher und Mikrofon, die Genauigkeit der Spracherkennung stark abnimmt. Aus diesem Grund ist es elementar die Störungen, nämlich Hintergrundgeräusche, Hall und Echo, aus den Mikrofonsignalen herauszurechnen.

Durch den Einsatz von mehreren Mikrofonen ist eine räumliche Trennung des Nutzsignals von den Störungen möglich. Diese Methode wird als *akustisches Beamformen* bezeichnet.

Konventionelle akustische Beamformer passen ihre Gewichte so an, dass die Varianz des Ausgangssignals minimiert wird, wobei das Signal in "Blickrichtung" die Bedingung der Verzerrungsfreiheit erfüllen muss.

Die Varianz ist definiert als das quadratische Mittel des Ausgangssignals. Somit werden bei konventionellen Beamformingmethoden Second-Order Statistics (SOS) des Ausgangssignals verwendet.

Konventionelle Beamformer können Störquellen effizient unterdrücken, aber leider auch das Nutzsignal. Diese unerwünschte Unterdrückung des Nutzsignals

wird im Englischen *signal cancellation* genannt und es wurden bereits viele Algorithmen entwickelt um dies zu vermeiden. Keiner dieser Algorithmen, jedoch, funktioniert effektiv in verhallter Umgebung.

Eine weitere Methode das Nutzsignal von den Störungen zu trennen, diesesmal jedoch ohne die geometrische Information zu nutzen, wird Blind Source Separation (BSS) [1] genannt. Hierbei wird eine Matrixmultiplikation mit dem Eingangssignal durchgeführt. Die Matrix muss so konstruiert werden, dass die Ausgangssignale statistisch unabhängig voneinander sind. Die statistische Unabhängigkeit wird mit der Theorie der Independent Component Analysis (ICA) gemessen [1].

Die ICA nimmt an, dass informationstragende Signale, wie z.B. Sprache, nicht gaußverteilt sind, wohingegen die Summe der Signale, z.B. das Hintergrundrauschen, gaußverteilt sind. Es gibt zwei gängige Arten um den Grad der Nichtgaußverteilung zu bestimmen, *Kurtosis* und *Negentropy*. Wie in dieser Arbeit beschrieben, werden hierbei höhere Momente als das zweite verwendet und somit werden diese Methoden als Higher-Order Statistics (HOS) bezeichnet.

Obwohl Arai et al. zeigten, dass sich Beamforming und BSS ähnlich sind, werden HOS beim akustischen Beamforming bisher nicht verwendet [2] und beruhen weiterhin auf SOS. In der hier vorliegenden Dissertation werden neue Beamformingalgorithmen entwickelt und evaluiert, die auf HOS basieren. Die neuen Beamformingmethoden passen ihre Gewichte anhand eines der folgenden Kriterien an:

- Minimum Mutual Information zweier Beamformer Ausgangssignale
- Maximum Negentropy der Beamformer Ausgangssignale und
- Maximum Kurtosis der Beamformer Ausgangssignale.

Es wird anhand von Spracherkennerexperimenten (gemessen in Wortfehlerrate) gezeigt, dass die hier entwickelten Beamformingtechniken auch erfolgreich Störquellen in verhallten Umgebungen unterdrücken, was ein klarer Vorteil gegenüber den herkömmlichen Methoden ist.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

There has been great and growing interest in microphone array processing for distant speech recognition (DSR) [3, 4, 5, 6]. Such techniques have the potential to relieve users from the necessity of donning close talking microphones (CTMs) before dictating or otherwise interacting with automatic speech recognition (ASR) systems. DSR techniques can be used in many applications such as intelligent room environments, humanoid robots, voice control systems for automobiles, automatic speech annotation systems in meetings and speech-to-speech translation systems.

A main problem in DSR is that a speech signal is corruped in realistic environments. In the case that a speaker is far from microphones, the sensors capture noise signals as well as a speech signal from the speaker. Because of the noise signals, the performance of ASR systems is seriously degraded. In addition to the noise, there are reverberation effects which also deteriorate the performance of ASR systems. The reverberation effects occur when hard surfaces such as tables and walls reflect a sound wave signal which conveys speech information.

In order to remove those unwanted noise or reverberation effects, many techniques have been developed [6]. Table 1.1 shows traditional speech enhancement techniques. As shown in Table 1.1, those techniques could be grouped

Table 1.1: Speech enhancement techniques.

Input type	Method	Additive Noise	Reverberation
Single Channel	Spectral subtraction [6, §6]	Yes	No
	Wiener filtering [8, §2]	Yes	No
	Bayesian filtering [6, §4]	Yes	No
	Blind de-convolution [8, §16]	Yes	No
	Joint particle filter approach [7]	Yes	Yes
Multi-channel	Beamforming [6, §13]	Yes	Yes

into two categories, single channel and multi-channel processing techniqes. Although most of the speech enhancement techniques with the single sensor have addressed either noise or reverberation, Wölfel proposed a method which can remove both the effects [7]. However, those signle channel processing techniques rely on noise spectral estimation which may fail in realistic environments. The errors of the noise estimation cause unexpected distortion of the speech signal, which also leads to the deterioration of the performance of ASR systems.

Acoustic beamforming is a promising technique for the DSR systems. Unlike single-channel speech enhancement techniques, it can take account of spatial information from sound sources to a microphone array and enhance speech coming from a target speaker while suppressing interference signals propagating from the other positions.

Beamforming algorithms are normally implemented in the frequency or subband domain for computational efficiency. In such a system, multi-channel input signals are first transformed into the subband domain with an analysis filter bank. The beamforming algorithm is then used to process these subband components and produce a single-channel output for each subband. After that, the processed signals are transformed back into the time domain, where a synthesis filter bank is used to obtain the final time-domain signal.

However, the filter bank design for beamforming poses problems not seen in traditional applications such as speech coding [9][10]. It was shown in [10] that the perfect reconstruction (PR) filter banks were not suitable for beamforming applications because PR is achieved through *aliasing cancellation* [11, §5], which can reconstruct an input signal correctly only if the outputs of the individual subbands are *not* subject to arbitrary magnitude scaling and phase shifts.

In this dissertation, filter bank design methods for beamforming are investigated. This thesis also describes a new filter bank design method which minimizes the magnitude of an individual aliasing term instead of canceling it. New filter banks are thoroughly analyzed. The effectiveness of beamforming with the filter banks is demonstrated through speech recognition experiments.

Beamformers are usually adapted to a specific acoustic environment in order to improve the performance of speech enhancement further. This is, for example, achieved by estimating the weights of the beamformer so as to minimize the variance of its outputs subject to a distortionless constraint in a look direction [6, §13.3.1].

We can efficiently implement such beamformers in *generalized sidelobe canceller* (GSC) configuration [6, §13.3.7]. Typical GSC beamformers consist of three blocks, a *quiescent vector*, a *blocking matrix* and an *active weight vector*. The quiescent vector is calculated to provide unity gain for the direction of interest. The blocking matrix is usually constructed in order to maintain the distortionless constraint for the signal filtered with the quiescent vector. Subject to this constraint, the total output power of the beamformer is minimized through the adjustment of the active weight vector, which effectively places a null on any source of interference, but can also lead to undesirable *signal cancellation* [12]. To avoid the latter, many algorithms have been developed. These approaches fall into one of the following categories:

- updating the active weight vector only when noise signals are dominant [13, 14, 15];
- constraining the update formula for the active weight vector with the leaky

least mean square (LMS) algorithm [16, 17] or with power of outputs of the blocking matrix [18];

- using multi-channel target signals received by the microphone array and correlation matrices of the clean and noise corrupted target signals in a calibration phase, [19];
- blocking the leakage of desired signal components into the sidelobe canceller by appropriately designing the blocking matrix [18, 20, 21, 22];
- taking speech distortion due to the leakage of a target signal into account using a multi-channel Wiener filter which aims at minimizing a weighted sum of residual noise and speech distortion terms [23]; and
- using acoustic transfer functions from a desired source to microphones instead of merely compensating for time delays of arrival of a signal [15, 22, 24, 25].

Blind source separation (BSS) might be considered as another approach to DSR [26]. The general goal of BSS is to separate each source signal from mixtures based solely on statistical independence of each signal. It is assumed in BSS that a priori knowledge such as the geometry of a microphone array and source positions are not given except for the number of active multiple sources.

BSS techniques have two well-known problems, that is, the scaling ambiguity and permutation problems [26]. The scaling ambiguity is eliminated by forcing the determinant of the unmixing matrix to unity [27]. The permutation problem is typically alleviated through use of the geometry of the microphone array [28]. However, many BSS techniques correct possible permutation by using the layout of the microphones once a solution to the BSS problem is obtained. It might be straightforward if the solution is sought with the distortionless constraint based on the geometry of the microphone array in order to prevent permutation from happening. Moreover, the BSS algorithms only provide a local solution which is highly dependent on the initial values. Furthermore, a lower bound on the performance of the speech separation is unpredictable. The unmixing matrix

obtained with this technique may fail to extract the target signal in some situations. Such uncertain behavior would be unacceptable for many applications.

Low et al. [29] proposed a method that combines a BSS technique and an adaptive noise canceller with the modified leaky LMS algorithm. Their algorithm first estimates the unmixing matrix with the information maximization technique [26], followed by solving the permutation problem by using the geometry information of the microphone array [28] and removing the scaling ambiguity by keeping the determinant of the unmixing matrix unity [27]. The output channel with the highest kurtosis value is then taken as the target speech and the others are labeled as reference signals. The adaptive noise canceller finally removes any components that are correlated to the reference signals, which also leads to the signal cancellation problem. To prevent it, Low et al. proposed the modified leaky LMS algorithm, which adjusts a step-size used for the weight update with a non-linear function. In their algorithm, the weights of the unmixing matrix for extracting the desired signal can be regarded as the block of the upper branch in the GSC structure and the other weights can be associated with the blocking matrix. Then, the active noise canceller corresponds to the active weight vector. Therefore, the method proposed by Low et al. could be viewed as a GSC beamforming algorithm without the distortionless constraint. However, their algorithm has the same problems as the BSS techniques.

One of the different points between acoustic beamforming and BSS approaches would be the criterion for adjusting the parameters. The traditional acoustic beamforming techniques have employed criteria based on the variance of the beamformer's outputs. Since the variance is the second moment of the outputs, it is referred to as *second order statistics* (SOS) in this thesis. On the other hand, most of the BSS techniques consider not only SOS but also *higher order statistics* (HOS) such as kurtosis and negentropy. The BSS algorithms use the fact that the distribution of information-bearing signals such as speech is not Gaussian but non-Gaussian. In contrast to the Gaussian probability density function, non-Gaussian probability density functions are not fully characterized by the first and second moment of their random variables only. This is why

HOS is employed in the field of BSS.

In this dissertation, HOS is considered for adjusting the active weight vectors of the GSC beamformers. More specifically, this thesis presents new HOS-based beamforming methods that optimize the active weight vectors of the GSC so as to make the distribution of the beamformer's outputs as non-Gaussian as possible.

There are three popular ways of measuring the distance between different distributions. First, the *mutual information* criterion is used for a speech separation task. Second, the *negentropy* and *kurtosis* criteria are taken into consideration for the beamformers in GSC configuration for a situation where only a single sound source is active. In particular, the latter two criteria measure the distance between Gaussian and non-Gaussian distributions, how far a distribution of a particular signal is from Gaussian. Computing the mutual information value requires the existence of multiple sound sources. In other words, it cannot be applied to the speech enhancement problem in the case that only a single speaker is speaking. In contrast to it, the negentropy and kurtosis criteria can be used in the single speaker scenario. Notice that all the criteria can take HOS into account.

As the author will demonstrate, these new beamforming algorithms with HOS can remove or suppress noise and reverberation without the signal cancellation problem encountered in conventional beamforming algorithms [12]. Moreover, these techniques can avoid the permutation and scaling ambiguity problems by maintaining the distortionless constraint in the look direction.

The effectiveness of the new techniques is demonstrated through a series of distant automatic speech recognition experiments on the *Multi-Channel Wall Street Journal Audio Visual Corpus* (MC-WSJ-AV) collected under the European Union integrated project *Augmented Multi-party Interaction* (AMI) [3]. The data was recorded with real sensors in a real meeting room, and hence contains noise from computers, fans, and other apparatus in the room. Moreover, some recordings include noise coming from outside the meeting room, such as that produced by passing cars or speakers in an adjacent room. The test data

is neither artificially convolved with measured impulse responses nor unrealistically mixed with separately recorded noise.

The balance of this thesis is organized as follows. Chapter 2 reviews the properties of propagating waves which convey human speech. Such signals can be expressed as functions of time and space. Knowledge of these properties is crucial for understanding array signal processing since an essential difference between single-channel signal processing and array processing is the incorporation of geometrical information. The most fundamental beamforming technique in array signal processing, *delay-and-sum* beamforming, is then described in Chapter 3. Beamforming in time and frequency domains is also discussed in Chapter 3. Chapter 4 describes the filter bank implementation for subband beamforming. In Chapter 5, conventional data-dependent beamforming algorithms are described. These beamformers adaptively update their weights based on the covariance matrices of the subband signals captured by the individual sensors, which are, of course, second order statistics. Chapter 6 reviews the theory of independent component analysis (ICA). The BSS technique is well-known as one of the applications of the ICA. Chapter 7 presents the novel beamforming algorithms which take HOS into consideration. In contrast to the conventional beamforming methods described in Chapter 5, the new beamforming algorithms do not suffer from the signal cancellation problem as demonstrated in simulations and experiments. Chapter 8 describes modern ASR systems which are used to evaluate the beamforming algorithms presented here. Chapter 9 shows the results of distant automatic speech recognition (ASR) experiments of two kinds of tasks, a speech separation challenge and single speaker scenario. In Chapter 10, the conclusions of this work are presented.

Contributions

Contributions of this thesis are summarized as follows:

- Filter bank design for beamforming. The undesired aliasing effects can

be alleviated in the case that the property of the perfect reconstruction is destroyed by arbitrary scaling of magnitude and phase shift [30, 31].

- Minimum mutual information (MMI) beamforming. It can separate sound sources without the signal cancellation problem encountered in the conventional beamforming techniques. Moreover, it is free from any problem seen in the BSS techniques [4].

- Maximum negentropy (MN) beamforming. Distant speech can be enhanced by this technique without the signal cancellation problem [32].

- Maximum kurtosis (MK) beamforming. This beamforming algorithm has the same advantage as MN beamforming. Furthermore, it can be simply implemented since the prior speech model is not required. However, the MK beamforming algorithm is influenced by outliers [33].

The list of publications related to this dissertation is presented in Chapter 11.

Chapter 2

Signals in Space and Time

Array processing techniques deal with propagating waves which convey signals from a source to the array. Such signals are functions of not only time but also space and referred to as *spatio-temporal signals* in this thesis.

The properties of the spatio-temporal signals are governed by the *wave equation*. In other words, we can analyze the signals by solving the wave equation. A well-known plane wave is derived by solving the wave equation in the Cartesian coordinate system. A spherical wave, which is also sometimes assumed in acoustic beamforming, can be derived from the wave equation represented in the spherical coordinates.

The balance of this chapter is organized as follows. Section 2.1 describes coordinate systems which are essential for representing the spatio-temporal signals. In Section 2.2, the wave equation is described. Section 2.3 and 2.4 describe the solutions of the wave equations in Cartesian coordinates and spherical coordinates, respectively.

2.1 Coordinate Systems

Figure 2.1 shows two coordinate systems, the Cartesian coordinate system and spherical coordinate system.

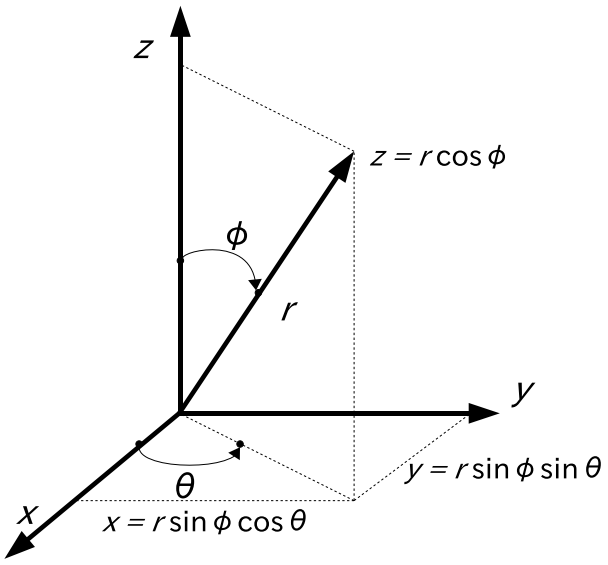


Figure 2.1: The Cartesian coordinate system and spherical coordinate system.

As illustrated in Figure 2.1, spatio-temporal signals are expressed as $s(x, y, z, t)$ with the spatial variables, x, y and z, and temporal variable, t, in the Cartesian coordinate system. We represent the unit vectors in the three spatial directions as $\boldsymbol{l}_x$, $\boldsymbol{l}_y$ and $\boldsymbol{l}_z$. Relationships between these directions are written as

$$\begin{aligned} \boldsymbol{l}_x \cdot \boldsymbol{l}_x &= \boldsymbol{l}_y \cdot \boldsymbol{l}_y = \boldsymbol{l}_z \cdot \boldsymbol{l}_z = 1 \\ \boldsymbol{l}_x \cdot \boldsymbol{l}_y &= \boldsymbol{l}_y \cdot \boldsymbol{l}_z = \boldsymbol{l}_z \cdot \boldsymbol{l}_x = 0 \\ \boldsymbol{l}_x \times \boldsymbol{l}_y &= \boldsymbol{l}_z. \end{aligned}$$

Let the *position vector* $\boldsymbol{p}_c$ denote the triple of the spatial variables (x, y, z) in the Cartesian coordinate system. We can then rewrite the spatio-temporal signal as $s(\boldsymbol{p}_c, t)$.

The spherical coordinate system is also used in array processing. As shown in Figure 2.1, a point in the spherical coordinate system is indicated with the

distance from the origin r, the azimuth on the xy-plane θ and polar angle down from the vertical axis ϕ. The spherical coordinate system is generally convenient for representing waves with spherical symmetry. For example, an isotropically spreading spherical wave can be represented by $s(r,t)$ without the angular coordinates.

The relations between the Cartesian and spherical coordinates are described as

$$\begin{aligned} r &= \sqrt{x^2+y^2+z^2} \\ \theta &= \cos^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right) = \sin^{-1}\left(\frac{y}{\sqrt{x^2+y^2}}\right) \\ \phi &= \cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \\ x &= r\sin\phi\cos\theta \\ y &= r\sin\phi\sin\theta \\ z &= r\cos\phi. \end{aligned}$$

2.2 Wave Equation

Information about a distant source signal is carried to the sensors through propagating waves which are governed by the wave equation. For a sound wave in a general field, the acoustic pressure $s(\boldsymbol{p}_c, t)$ satisfies the wave equation

$$\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2}, \tag{2.1}$$

where the parameter of the wave equations c can be interpreted as the speed of propagation [34]. The wave equation (2.1) has information about how signals propagate from a source radiating energy to an array.

Table 2.1 shows the sound speed and parameters in a gas and fluid [35]. As shown in Table 2.1, the sound signal depends on the medium which a wave passes through.

Table 2.1: Sound speed and parameters in different mediums.

Medium	Formula	Values
Air	$c = \sqrt{\gamma RT_0/M}$	330.7 m/s
	γ : the specific heat ratio	1.4
	R : the gas constant per mole	8.3×10^7 erg/^oK
	T_0 : the ambient temperature	273 ^oK
	M : the molar mass	29 g
Sea water	$c = \sqrt{\gamma B/\rho}$	$1,498$ m/s^b
	γ : the specific heat ratio	1.01
	B : the isothermal bulk modulus	2.28×10^9 N/m^2
	ρ : the density	1.026×10^3 kg/m^3

2.3 Solutions of the Wave Equation in Cartesian Coordinates

There are several well-known solutions to the wave equation. In this section, we derive the solution corresponding to the *plane wave*.

Let us first assume that the acoustic pressure $s(\boldsymbol{p}_c, t)$ has a complex form

$$s(\boldsymbol{p}_c, t) = A \exp\left\{j(\omega t - \boldsymbol{k} \cdot \boldsymbol{p}_c)\right\}, \tag{2.2}$$

where A is a complex constant, ω is a real constant and $\boldsymbol{k} = (k_x, k_y, k_z)$ is a constant vector with real values called the *wave number vector*. Substituting (2.2) into the wave equation (2.1) and expanding the components of the wave number vector, we have

$$k_x^2 s(\boldsymbol{p}_c, t) + k_y^2 s(\boldsymbol{p}_c, t) + k_z^2 s(\boldsymbol{p}_c, t) = \frac{\omega^2}{c^2} s(\boldsymbol{p}_c, t). \tag{2.3}$$

Canceling $s(\boldsymbol{p}_c, t)$, we obtain the constraint

$$k_x^2 + k_y^2 + k_z^2 = |\boldsymbol{k}|^2 = \frac{\omega^2}{c^2}. \tag{2.4}$$

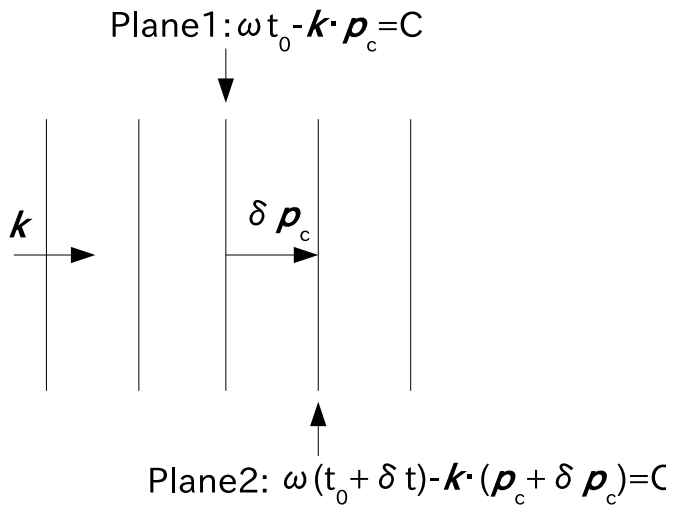


Figure 2.2: Visualization of a propagating plane wave with a constant phase.

As long as (2.4) is satisfied, the signal expressed with (2.2) satisfies the wave equation (2.1).

The solution to the wave equation given by (2.2) may be interpreted as a *monochromatic* plane wave. The term *plane wave* arises because the values of the signal at any instant of time t_0 are the same at all the points on the plane given by $k_x x + k_y y + k_z z = C$, where C is a constant. The surface on which the values of the signal are the same is referred to as the wavefront.

Let us consider a case where a plane with a constant phase moves by a distance $\delta \boldsymbol{p}_c$ during time δt as shown in Figure 2.2. In this case, we have

$$s(\boldsymbol{p}_c + \delta \boldsymbol{p}_c, t + \delta t) = s(\boldsymbol{p}_c, t). \tag{2.5}$$

By expressing (2.5) as in (2.2), we have

$$A \exp\left\{j(\omega(t + \delta t) - \boldsymbol{k} \cdot (\boldsymbol{p}_c + \delta \boldsymbol{p}_c))\right\} = A \exp\left\{j(\omega t - \boldsymbol{k} \cdot \boldsymbol{p}_c)\right\}. \tag{2.6}$$

Upon dividing both sides of (2.6) by $A \exp\{j(\omega t - \boldsymbol{k} \cdot \boldsymbol{p}_c)\}$, we obtain

$$\exp\left\{j(\omega \delta t - \boldsymbol{k} \cdot \delta \boldsymbol{p}_c)\right\} = 1, \tag{2.7}$$

which indicates

$$\omega\delta t - \boldsymbol{k} \cdot \delta \boldsymbol{p}_c = 0. \tag{2.8}$$

We may take the direction of $\delta \boldsymbol{p}_c$ to be the same as that of $\boldsymbol{k}$. In the case that $\delta \boldsymbol{p}_c$ and $\boldsymbol{k}$ have the same direction, the inner product $\delta \boldsymbol{p}_c \cdot \boldsymbol{k}$ is $|\delta \boldsymbol{p}_c||\boldsymbol{k}|$. Then, upon modifying (2.8), we have

$$\frac{|\delta \boldsymbol{p}_c|}{\delta t} = \frac{\omega}{|\boldsymbol{k}|}. \tag{2.9}$$

The ratio $|\delta \boldsymbol{p}_c|/\delta t$ can be interpreted as the speed of propagation of the plane wave.

Based on (2.4) and (2.9), we have

$$\frac{|\delta \boldsymbol{p}_c|}{\delta t} = c, \tag{2.10}$$

where c is the speed of propagation.

The distance propagated during one temporal period $T = 2\pi/\omega$ is called the wavelength λ. Substituting $\delta t = 2\pi/\omega$ into (2.9), we obtain

$$|\delta \boldsymbol{p}_c| = \lambda = \frac{2\pi}{|\boldsymbol{k}|}. \tag{2.11}$$

The magnitude of the wave number vector $|\boldsymbol{k}|$ expresses the number of cycles in radians per meter.

For the sake of simplicity, we rewrite (2.2) as

$$s(\boldsymbol{p}_c, t) = A \exp\left\{\omega(t - \boldsymbol{\alpha} \cdot \boldsymbol{p}_c)\right\}, \tag{2.12}$$

where $\boldsymbol{\alpha} = \boldsymbol{k}/\omega$. Then, $s(\boldsymbol{p}_c, t)$ can be expressed as a function of a single argument

$$\acute{s}(t - \boldsymbol{\alpha} \cdot \boldsymbol{p}_c), \tag{2.13}$$

where $\acute{s}(u) = A\exp(j\omega u)$. The vector $\boldsymbol{\alpha}$ has the magnitude which is equal to the reciprocal of the propagation speed. For this reason, it is often called a *slowness vector*.

The wave equation is linear: If $s_1(\boldsymbol{p}_c, t)$ and $s_2(\boldsymbol{p}_c, t)$ are solutions of the wave equation, then the linear combination $A_1 s_1(\boldsymbol{p}_c, t) + A_2 s_2(\boldsymbol{p}_c, t)$ is also a

solution. By using the linear property, we can obtain a more detailed solution

$$s(\boldsymbol{p}_c, t) = \acute{s}(t - \boldsymbol{\alpha} \cdot \boldsymbol{p}_c) = \sum_{n=-\infty}^{\infty} S_n \exp\left\{jn\omega_0(t - \boldsymbol{\alpha} \cdot \boldsymbol{p}_c)\right\}. \tag{2.14}$$

Equation (2.14) represents the harmonic series with a fundamental frequency ω_0. With Fourier's Theorem, the coefficient S_n can be written as

$$S_n = \frac{1}{T}\int_0^T \acute{s}(u)\exp(-jn\omega_0 u)du, \tag{2.15}$$

where $T = 2\pi/\omega_0$.

2.4 Solutions of the Wave Equation in Spherical Coordinates

The wave equation (2.1) can be also expressed in the spherical coordinates (r, θ, ϕ). The calculations result in the general spherical wave equation

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial s}{\partial r}\right) + \frac{1}{r^2\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial s}{\partial\phi}\right) + \frac{1}{r^2\sin^2\phi}\frac{\partial^2 s}{\partial\phi^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2}. \tag{2.16}$$

This equation can be solved by the method of separation of variables [35]. General solutions involve Bessel functions and associated Legendre polynomials.

In this section, we derive a simple solution in the case that the signal has spherical symmetry. By removing the dependencies on ϕ and θ, the spherical wave equation can be simplified to

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial s}{\partial r}\right) = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2}. \tag{2.17}$$

By expanding the left side of (2.17) and multiplying both sides with r, we have

$$2\frac{\partial s}{\partial r} + r\frac{\partial^2 s}{\partial r^2} = \frac{r}{c^2}\frac{\partial^2 s}{\partial t^2}. \tag{2.18}$$

Then, by manipulating

$$\frac{\partial^2(rs)}{\partial r^2} = 2\frac{\partial s}{\partial r} + r\frac{\partial^2 s}{\partial r^2}$$

and

$$\frac{\partial^2(rs)}{\partial t^2} = r\frac{\partial^2 s}{\partial t^2},$$

equation (2.17) can be modified as

$$\frac{\partial^2(rs)}{\partial r^2} = \frac{1}{c^2}\frac{\partial^2(rs)}{\partial t^2}. \tag{2.19}$$

One solution of (2.19) is a monochromatic wave

$$s(r,t) = \frac{A}{r}\exp\left\{j(\omega t - kr)\right\}. \tag{2.20}$$

This can be interpreted as a spherical wave propagating outward from the origin. With $k^2 = \omega^2/c^2$, we can rewrite (2.20) as

$$s(r,t) = \frac{A}{r}\exp\left\{j\omega\left(t - \frac{r}{c}\right)\right\}. \tag{2.21}$$

The propagating spherical wave has another form

$$s(r,t) = \frac{B}{r}\exp\left\{j\omega\left(t + \frac{r}{c}\right)\right\}. \tag{2.22}$$

Notice that the wave expressed with (2.22) propagates toward the origin.

In the similar way with the Cartesian case, we can build up more complicated inwardly propagating waves by superimposing complex exponentials of this form. Because of the linearity of the equation, it is also possible to obtain solutions that consist of the superposition of both inwardly and outwardly propagating waves.

Chapter 3

Beamforming

Beamforming can be generally described as *spatial filtering*. Due to the geometry of the array, the sensors in fact sample the propagating wave both in time and space. This enables the subsequent signal processing on the output of each sensor to make the array more sensitive in a desired direction, and to suppress undesired signals arriving from other directions. Beamforming techniques have been applied to many areas such as radar, sonar, seismology and speech enhancement.

This chapter reviews beamforming techniques in the context of signal enhancement. Section 3.1 describes the most fundamental delay-and-sum beamforming algorithm. In Section 3.2, discrete-time beamforming is described. Section 3.3 discusses discrete-time beamformers operating in the frequency domain. Section 3.4 depicts null-steering beamforming techniques which are able to null interference signals.

3.1 Delay-and-Sum Beamforming

Delay-and-sum beamforming is the oldest and simplest array signal processing algorithm and still used in many applications today.

Let $x_i(t)$ denote a wave signal measured at the i-th sensor at a time instant

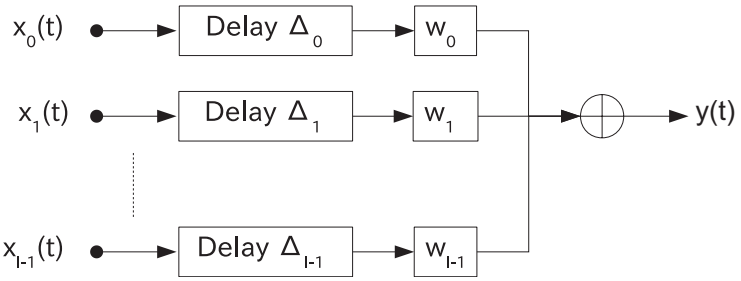


Figure 3.1: The delay-and-sum beamformer.

t. With a sensor weight for the i-th sensor w_i, the delay-and-sum beamformer's output can be written as

$$y(t) = \sum_{i=0}^{I-1} w_i \, x_i(t + \Delta_i), \tag{3.1}$$

where I is the number of microphones and Δ_i is the time delay for the i-th microphone. The amplitude weighting w_i enhances the beam's shape and reduces sidelobe levels. The delays are compensated to strengthen the signal coming from a particular point or direction in space.

Figure 3.1 shows a block diagram of the delay-and-sum beamformer. The delay-and-sum beamforming algorithm applies the delay and weight to the received signal and sums the resulting signals.

3.1.1 Far-field and Near-field Assumptions

In delay-and-sum beamforming, the choices of methods to compensate delays would depend on whether the sources are located in the *far-field* or *near-field*.

A majority of literature assume that the sound source is at an infinite distance from the array - in the far-field. Under the far-field assumption, the sound wave is assumed to be the plane wave described in Chapter 2, which indicates that the surface of the wave (wavefront) received from a single point source is plane. Figure 3.2 illustrates the planar wavefront under the far-field assumption.

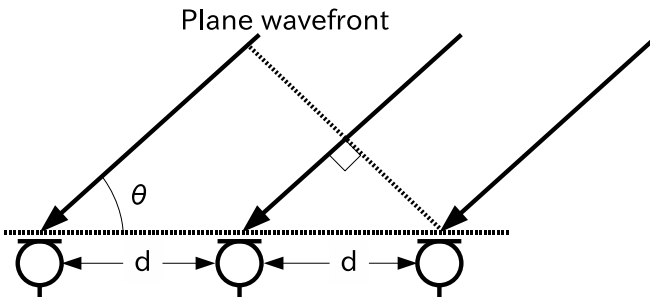


Figure 3.2: Illustration of the plane wave arriving at each sensor in the far-field case.

Under the near-field assumption, the wavefront is assumed to be spherical. Figure 3.3 shows the spherical wave propagating in the near-field. Several authors have shown that beamforming under the near-field assumption provides superior enhancement performance as compared to beamforming under the far-field assumption [36, 37].

It is clear from Figure 3.2 and Figure 3.3 that the measurement of the delay to each sensor in the far-field is different from that in the near-field. The common rule for the approximate distance at which the far-field approximation begins to be valid is $r = 2d_L^2/\lambda$, where d_L is the *aperture* of the array; i.e., the distance between the two most distantly spaced elements of the array [38]. However, many studies employ the far-field assumption regardless of the inaccurate approximation since it significantly simplifies problems in beamforming.

3.1.2 Beam Patterns

An analysis of the frequency response of a linear time-variant system to a sinusoidal input provides the relationship between system's input and output in single-channel signal processing. We can analyze the delay-and-sum beamformer's output in the similar way. It is only necessary to examine the delay-and-sum beamformer's response to a monochromatic plane wave propagating

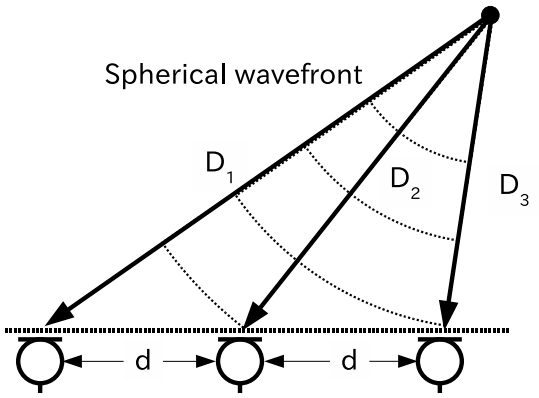


Figure 3.3: Illustration of the spherical wave arriving at each sensor in the near-field case.

with the slowness vector described in Section 2.3. The delay-and-sum beamformer's response to a monochromatic wave is often called a *beam pattern*. The beam pattern is sometimes called directivity pattern or spatial pattern.

Linear Array

Let us first denote a monochromatic signal propagating with a frequency ω^o and slowness vector $\boldsymbol{\alpha}^o$ at a position $\boldsymbol{p}_c$ as

$$s(t - \boldsymbol{\alpha}^o \cdot \boldsymbol{p}_c) = \exp\left\{j\omega^o(t - \boldsymbol{\alpha}^o \cdot \boldsymbol{p}_c)\right\}. \tag{3.2}$$

In the case that the i-th sensor is located at a position $\boldsymbol{p}_{c,i}$, the output of the delay-and-sum beamformer with the sensor weight for the i-th sensor w_i can be expressed as

$$\begin{aligned} z(t) &= \sum_{i=0}^{I-1} w_i s\left(t + (\boldsymbol{\alpha} - \boldsymbol{\alpha}^o) \cdot \boldsymbol{p}_{c,i}\right) \\ &= \left[\sum_{i=0}^{I-1} w_i \exp\left\{j\omega^o(\boldsymbol{\alpha} - \boldsymbol{\alpha}^o) \cdot \boldsymbol{p}_{c,i}\right\}\right] \exp(j\omega^o t). \end{aligned} \tag{3.3}$$

It is clear from (3.3) that if we set the wrong look direction $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}^o$, we obtain the degraded signal.

In order to investigate the response of the delay-and-sum beamformer with the sensors which spatially sample signals propagating from specific directions, let us introduce the Fourier transform of the sensor weight w_i as

$$W(\omega\boldsymbol{\alpha}) = \sum_{i=0}^{I-1} w_i \exp(j\omega\boldsymbol{\alpha} \cdot \boldsymbol{p}_{c,i}). \tag{3.4}$$

Based on (3.4), we can re-write (3.3) as

$$z(t) = W\left(\omega^o(\boldsymbol{\alpha} - \boldsymbol{\alpha}^o)\right) \exp(j\omega^o t). \tag{3.5}$$

It is clear from (3.5) that the quantity $W\left(\omega^o(\boldsymbol{\alpha} - \boldsymbol{\alpha}^o)\right)$ determines the amplitude and phase of the beamformer's output.

Consider a situation where a plane wave is arriving at a linear array of $I = 2I_{1/2} + 1$ equally spaced microphones separated by d m, as shown in Figure 3.2. In such a situation, when all the sensor weights are equal to 1, the response of the delay-and-sum beamformer can be expressed [35, §4] as

$$W\left(\omega^o(\boldsymbol{\alpha} - \boldsymbol{\alpha}^o)\right) = \frac{1}{I} \frac{\sin \frac{I}{2}(\omega^o(\alpha_x - \alpha_x^o))d}{\sin \frac{1}{2}(\omega^o(\alpha_x - \alpha_x^o))d}. \tag{3.6}$$

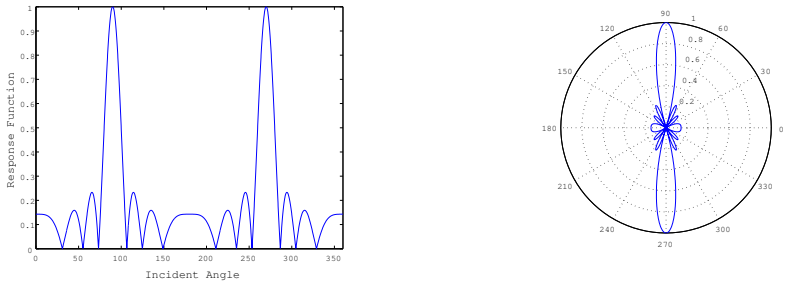
In applications, the beam pattern as a function of the incident angle is useful. For the linear array, the argument of the beam pattern can be written as

$$\begin{aligned} \omega^o\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^o\right) &= \omega^o(\alpha_x - \alpha_x^o) \\ &= \frac{\omega^o}{c}(\cos\theta - \cos\theta^o) \\ &= \frac{2\pi}{\lambda}(\cos\theta - \cos\theta^o) \end{aligned} \tag{3.7}$$

Notice that a relationship $\omega^o/c = 2\pi/\lambda$ can be easily obtained based on substituting (2.10) and (2.11) into (2.9).

Upon substituting (3.7) into (3.6), we have the beam pattern as a function of the incident angle [35, §4]

$$W(\theta) = \frac{1}{I} \frac{\sin \frac{I}{2}\left(\frac{2\pi}{\lambda}(\cos\theta - \cos\theta^o)d\right)}{\sin \frac{1}{2}\left(\frac{2\pi}{\lambda}(\cos\theta - \cos\theta^o)d\right)}. \tag{3.8}$$



(a) (b)

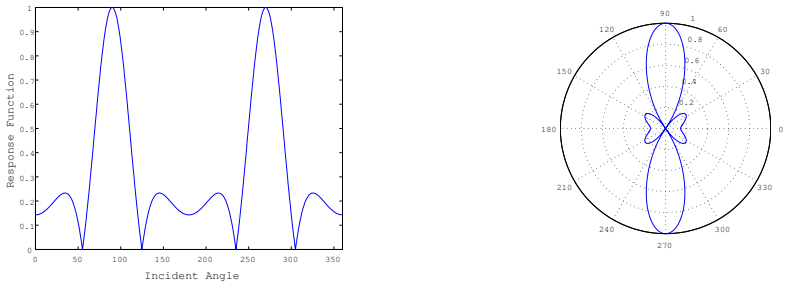
Figure 3.4: Beam patterns of the linear array with 7 sensors with $\lambda = 2d$: (a) in Cartesian coordinates and (b) in polar coordinates.

Figure 3.4 shows the beam patterns of the equi-spaced linear array as a function of the incident angle θ with $I = 7$ and $\lambda = 2d$ in the case that the look direction θ^o is $\pi/2$ $(= 90^o)$. In Figure 3.4, the region with the highest amplitude is called a *mainlobe* and the others are called *sidelobe*s. One important parameter regarding the mainlobe is a beamwidth which is defined as the region between the first zero-crossings on either side of the mainlobe. The height of the sidelobes represents suppression performance for interference signals arriving from the directions other than the desired look direction.

Figure 3.5 and Figure 3.6 also plot the beam patterns with 7 sensors with $\lambda = 4d$ and $\lambda = 8d$, respectively. By comparing Figure 3.5 with Figure 3.6, we can see that the larger the wavelength of the propagating signal is, the broader the beamwidth is. We can also find that there is no deep null in Figure 3.6, which implies that interference signals cannot be suppressed very well.

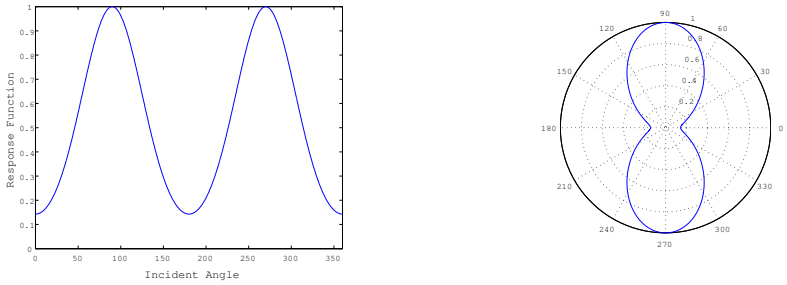
The performance of the array can be improved by increasing the number of sensors. Figure 3.8 shows the beam patterns with 15 sensors, where the other parameters are the same as those in Figure 3.6. Comparing Figure 3.6 with Figure 3.8, it is obvious that the larger number of sensors leads to a sharper beam.

Figure 3.8 illustrates the beam patterns of the linear array with 7 sensors



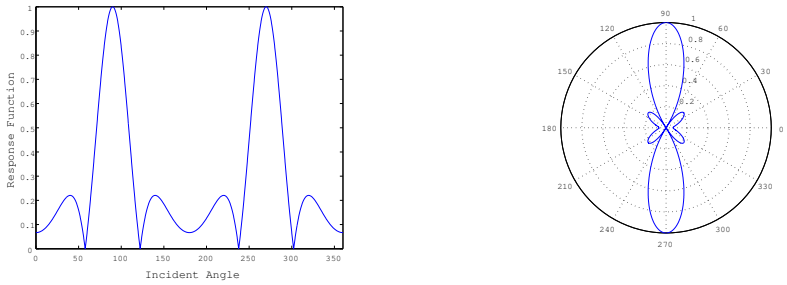
(a) (b)

Figure 3.5: Beam patterns of the linear array with 7 sensors with $\lambda = 4d$: (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 3.6: Beam patterns of the linear array with 7 sensors with $\lambda = 8d$: (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

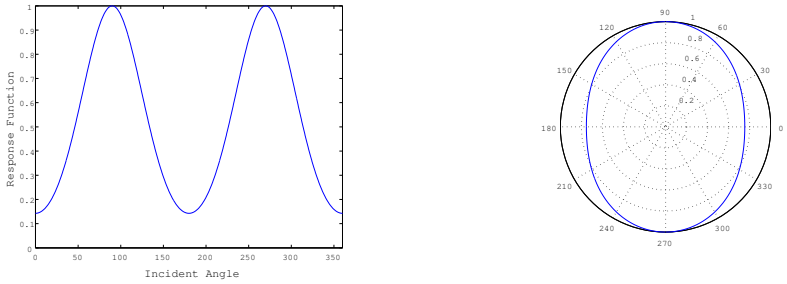
Figure 3.7: Beam patterns of the linear array with 15 sensors with $\lambda = 8d$: (a) in Cartesian coordinates and (b) in polar coordinates.

in the case that the frequency value f is 400 Hz and each distance between sensors d is 0.05 m (λ/d=17.2). On the other hand, Figure 3.9 shows the beam patterns with the frequency value $f = 400$ Hz and the distance between sensors $d = 0.2$ m (λ/d=4.3). It is clear by comparing Figure 3.8 to Figure 3.9 that the suppression performance for interference signals is poor in the case that the distance between sensors is small.

It could be expected that the performance of the array is improved by placing sensors further apart. However, a problem called the *spatial aliasing* arises when d is larger than $\lambda/2 = c/(2/f)$ [39, §2.5]. Figure 3.10 shows the beam patterns in the case of $f = 3200$ Hz and $d = 0.2$ m (λ/d=0.54). Observe that there are five additional lobes that are as large as the mainlobe. These undesired lobes are known *gratinglobes* and represent strong sensitivities of the array in undesired directions. Their appearance is known as spatial aliasing because it implies that the array is incapable of distinguishing between the directions of arrival of plane waves that are not arriving from the look direction.

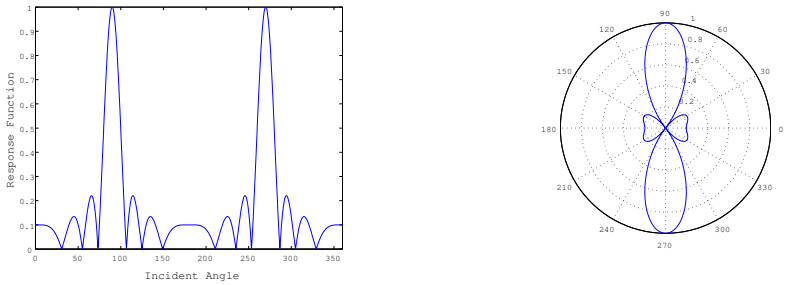
Circular Array

In order to illustrate the fundamental properties of beamforming, we considered the beam patterns of linear microphone arrays. Even if the geometry of the array



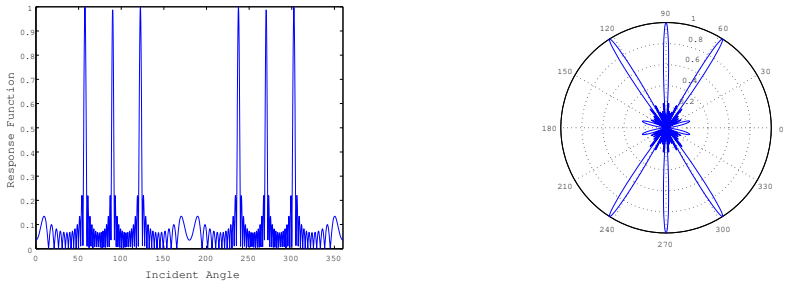
(a) (b)

Figure 3.8: Beam patterns of the linear array with 7 sensors with $f = 400$ Hz, $d = 0.05\ m$ ($\lambda/d = 17.19$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 3.9: Beam patterns of the linear array with 7 sensors with $f = 400$ Hz, $d = 0.2\ m$ ($\lambda/d = 4.297$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 3.10: Beam patterns of the linear array with 7 sensors with $f = 3200$ Hz, $d = 0.2\ m$ ($\lambda/d = 0.5371$): (a) in Cartesian coordinates and (b) in polar coordinates.

is not linear, the fundamental properties remain the same. Let us consider a circular array with uniformly spaced microphones as shown in Figure 3.11. In such a case, with a radius of the circular array R and azimuth from the x-axis to the i-th sensor on the xy-plane θ_i, the position vector for the i-th microphone in the Cartesian coordinate can be expressed by

$$\boldsymbol{r}_i = [\ R\cos\theta_i,\ R\sin\theta_i,\ 0\ ]^T. \tag{3.9}$$

Then, in the case that the delay-and-sum beamformer is constructed to steer the beam for the look direction $[\theta^o, \phi^o]$, the beam pattern of the circular array for the direction $[\theta, \phi]$ can be expressed as

$$W_c(\theta,\phi) = \frac{1}{I}\sum_{i=0}^{I-1} \exp\left[j\frac{2\pi}{\lambda} R\sin\phi\cos(\theta-\theta_i) + j\beta_i \right], \tag{3.10}$$

where β_i is a phase factor with respect to the origin [39, §4.2] and

$$\beta_i = -\frac{2\pi}{\lambda} R\sin\phi^o\cos(\theta^o-\theta_i). \tag{3.11}$$

Figure 3.12, 3.13 and 3.14 show the beam patterns of a uniform circular array with eight microphones as a function of the azimuth θ, where the polar angle ϕ is $\pi/2$. In these figures, the look direction of $[\theta = \pi/2,\ \phi = \pi/2]$ is

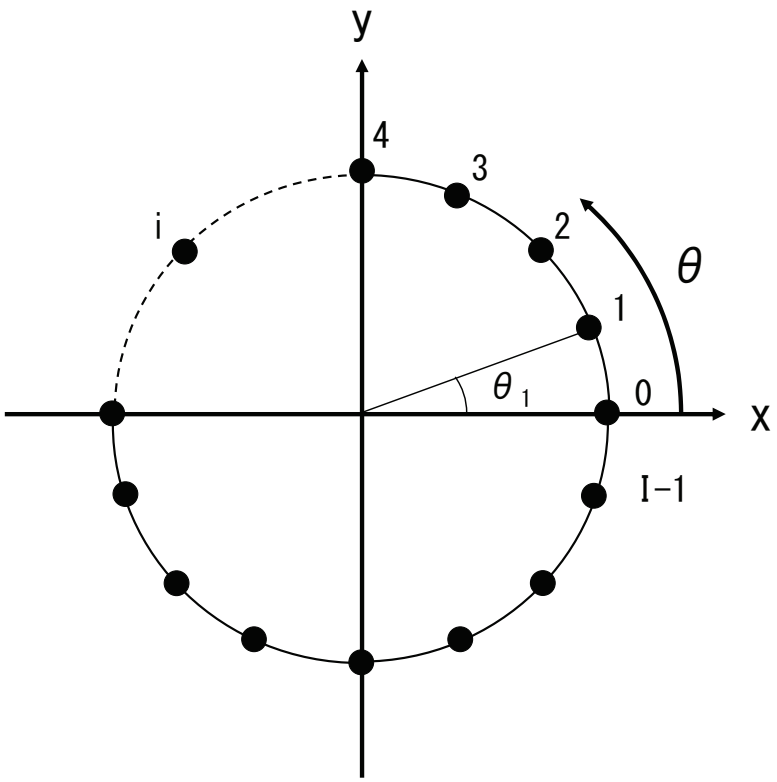
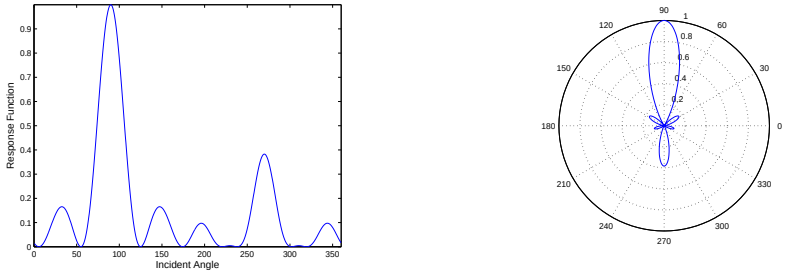
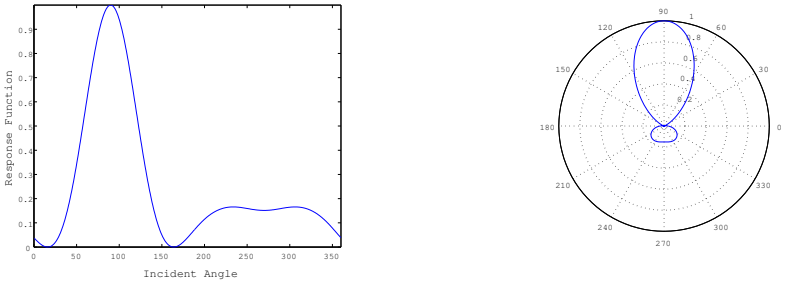


Figure 3.11: The geometry of the circular array with equally spaced microphones.



(a) (b)

Figure 3.12: Beam patterns of the circular array with 8 sensors with $\lambda = 2d_{\mathrm{arc}}$: (a) in Cartesian coordinates and (b) in polar coordinates.

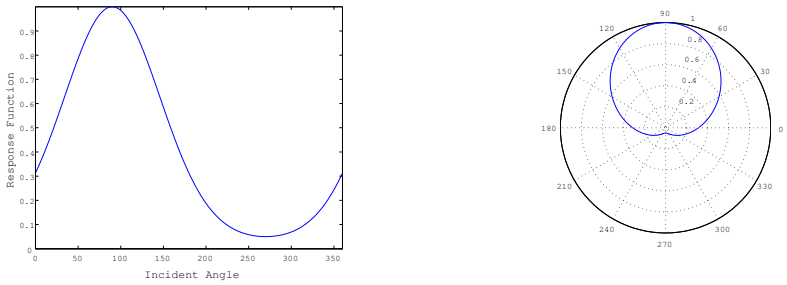


(a) (b)

Figure 3.13: Beam patterns of the circular array with 8 sensors with $\lambda = 4d_{\mathrm{arc}}$: (a) in Cartesian coordinates and (b) in polar coordinates.

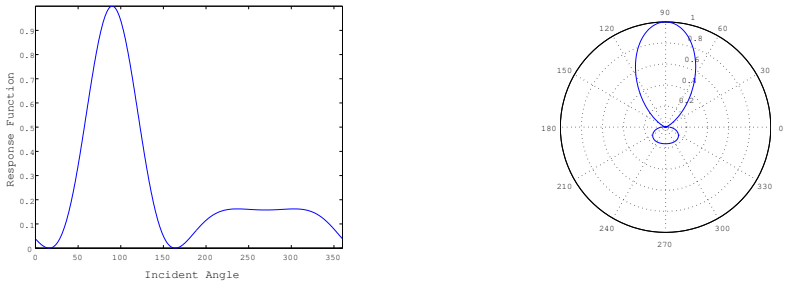
maintained. The beam patterns shown in Figure 3.12 are computed in the case of $\lambda = 2d_{\mathrm{arc}}$, where d_{arc} indicates the distance on the arc of the circular array between two adjacent microphones and hence $d_{\mathrm{arc}} = 2\pi R/I$. In Figure 3.13 and Figure 3.14, $\lambda = 4d_{\mathrm{arc}}$ and $\lambda = 8d_{\mathrm{arc}}$ are set. We can see the same trend as the case of the linear microphone array from Figure 3.12, Figure 3.13 and Figure 3.14 that the beamwidth becomes large when the wavelength is large.

Figure 3.15 shows the beam patterns of the uniform circular array with 16 microphones as a function of the incident angle θ. The other conditions are the same as those for Figure 3.14. By comparing Figure 3.14 with Figure 3.15, we



(a) (b)

Figure 3.14: Beam patterns of the circular array with 8 sensors with $\lambda = 8d_{\text{arc}}$: (a) in Cartesian coordinates and (b) in polar coordinates.



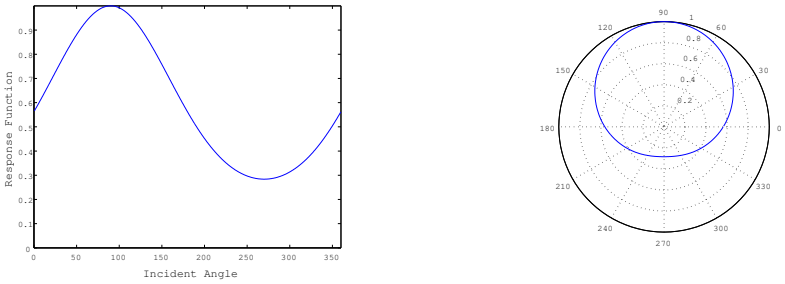
(a) (b)

Figure 3.15: Beam patterns of the circular array with 16 sensors with $\lambda = 8d_{\text{arc}}$: (a) in Cartesian coordinates and (b) in polar coordinates.

can confirm that we can make the shape of the beam sharper by increasing the number of the microphones, which is also the case in the linear array.

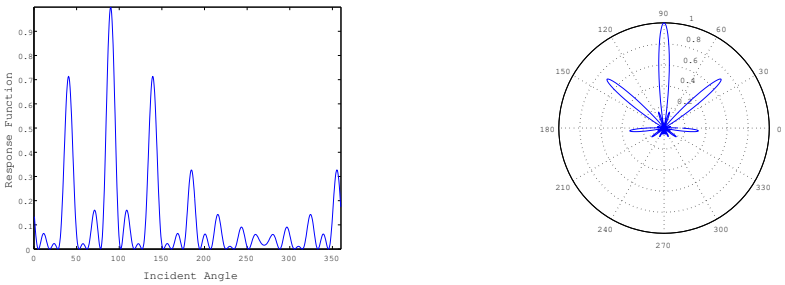
Figure 3.16 and 3.17 show beam patterns of the circular array with 8 microphones for frequencies 400 Hz and 6400 Hz, respectively. In order to calculate the beam patterns in Figure 3.16 and 3.17, the diameter of the circular array is set to 0.2 m and the plane wave is assumed to propagate parallel to the plane of the circular array.

It is clear from Figure 3.16 that the performance of the circular array at low frequencies is also poor since the wavelength is much longer than the aperture



(a) (b)

Figure 3.16: Beam patterns of the circular array with 8 sensors with $f = 400$ Hz, $\lambda = 0.8594$ ($\lambda/d_{\text{arc}} = 10.94$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 3.17: Beam patterns of the circular array with 8 sensors with $f = 6400$ Hz, $\lambda = 0.0537$ ($\lambda/d_{\text{arc}} = 0.6838$): (a) in Cartesian coordinates and (b) in polar coordinates.

of the array at low frequencies. From Figure 3.17, it is also clear that at high frequencies, the beam patterns are characterized by very large sidelobes.

3.2 Discrete-Time Beamforming

We have considered beamforming for analog signals. Here we consider the digital version of the delay-and-sum beamformer. The output of the discrete-time delay-and-sum beamformer can be expressed as

$$y[n] = \sum_{i=0}^{I-1} w_i \, x_i[n + \tau_i], \tag{3.12}$$

where τ_i is an integer which represents the delay corresponding to the i-th microphone and w_i is a weight which amplifies a signal attenuated by the propagation effect.

If the i-th microphone is located at a position $\boldsymbol{p}_{c,i}$ and the monochromatic signal is sampled with a period T, the discrete-time signal is expressed as

$$x_i[n] = \exp\left\{ j\omega^o \left(nT - \boldsymbol{\alpha}^o \cdot \boldsymbol{p}_{c,i} \right) \right\}. \tag{3.13}$$

The beamformer's output for the discrete-time signal with the frequency ω^o is then written as

$$y[n] \quad = \quad \left[\sum_{i=0}^{I-1} w_i \exp\left\{ j\omega^o (\tau_i T - \boldsymbol{\alpha}^o \cdot \boldsymbol{p}_{c,i}) \right\} \right] \exp\left(j\omega^o nT \right). \tag{3.14}$$

We ideally adjust the discrete-time delay $\tau_i T$ to equal $-\boldsymbol{\alpha}^o \cdot \boldsymbol{p}_{c,i}$ in order to steer a beam to the propagation direction. However, these ideal delays are generally not integer multiples of the sampling period T. We can no longer steer beams to arbitrary directions precisely. In order to form more accurate beams with integer delays, we have to adjust d and T so that $d \gg cT$. The reduction of T leads to the higher resolution for the original continuous signal. Increasing d makes the beamwidth shorter but might cause the spatial aliasing.

3.3 Discrete-Time Frequency-Domain Beamforming

Frequency-domain beamforming systems have advantages compared to the time-domain implementation [40]. These advantages are summarized as follows:

1. saving a great deal of computation time by using the Fast Fourier Transform (FFT) and

2. obtaining approximately uncorrelated signals by orthogonal transformation, which leads to the fast convergence of the estimation of beamformer's weights.

The short-time Fourier transform (STFT) of a sampled signal $x_i[n]$ is given by

$$X_i(k,\omega) = \sum_{n=k}^{k+L_h-1} h[n-k]\, x_i[n]\, \exp(-jn\omega T), \tag{3.15}$$

where $h[n]$ is a window function with duration L_h which assumes nonzero values only in the interval $[0,\ L_h - 1]$. The Hanning and Hamming window functions [41] are popular in acoustic and speech processing. Figure 3.18 shows the conceptual procedure of the STFT for a sound signal. The samples within the analysis window are first transformed by the Fourier transform and then the analysis window is shifted to the next incoming samples. This process is repeated until the analysis window reaches the end of the signal.

The beamformer's output can be expressed as

$$Y(k,\omega) = \sum_{i=0}^{I-1} w_i\, X_i(k,\omega)\, \exp\left\{j\omega(k+\Delta_i)\right\}. \tag{3.16}$$

With a little modification of (3.15), we have

$$X_i(k,\omega)\, \exp(jk\omega T) = \sum_{n=0}^{L_h-1} h[n]\, x_i[k+n]\, \exp(-jn\omega T). \tag{3.17}$$

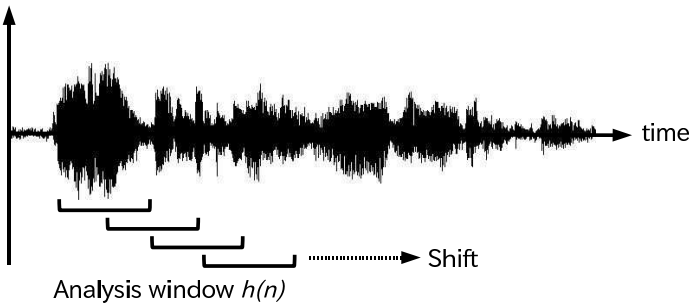


Figure 3.18: The procedure of the STFT.

We now re-write (3.17) with discrete frequency values $\omega T = 2\pi\, m/L_h$, where $m = 0, \cdots, L_h - 1$. The simplified form can be written as

$$X_i(k,m)\, \exp\left(j\frac{2\pi mk}{L_h}\right) = \sum_{n=0}^{L_h-1} h[n]\, x_i[k+n]\, \exp\left(-j\frac{2\pi mn}{L_h}\right). \qquad (3.18)$$

If you closely look at the right side of (3.18), you will find that it represents the discrete Fourier transform (DFT) of the signal $z[n] = h[n]x_i[k+n]$. The window function $h[n]$ can be considered the unit-sample response of a narrowband lowpass filter. The term $h[n]\exp(-j2\pi mn/L_h)$ thus represents the unit-sample response of a bandpass filter centered at the frequency of the index m. The equation (3.18) represents the response of the bandpass filter to the signal $x_i[n]$.

With the notations of the discrete-time and discrete-frequency, the frequency-domain beamformer's output can be expressed as

$$Y(k,m) = \sum_{i=0}^{I-1} w_i\, X_i(k,m) \exp\left\{j\frac{2\pi m}{L_h}\left(k + \frac{\Delta_i}{T}\right)\right\}, \qquad (3.19)$$

where Δ_i indicates a time delay.

Upon substituting (3.18) into (3.19), we have

$$Y(k,m) = \sum_{i=0}^{I-1}\sum_{n=0}^{L_h-1} w_i\, h[n]\, x_i[k+n] \exp\left\{-j\frac{2\pi mn}{L_h}\right\} \exp\left\{j\frac{2\pi m}{L_h}\frac{\Delta_i}{T}\right\}. \qquad (3.20)$$

After beamforming in the frequency domain, we have to transform the frequency components back into time-sampled signals. However, multiplication of the DFT by a phase factor would perform the *circular convolution* which results

in the circular shift of the corresponding time-domain signal [41]. This effect is called the *circular aliasing* wherein values trailing over a frame, once circular shifted to the right, appear as leading samples. In more detail, the inverse DFT (IDFT) of the beamformer's output is expressed as

$$\begin{aligned} y[k,l] &= \text{IDFT}\left[Y(k,m)\right] \\ &= \sum_{i=0}^{I-1} \text{IDFT}\left[w_i X_i(k,m)\, \exp\left\{j\frac{2\pi mk}{L_h}\right\} \exp\left\{j\frac{2\pi m}{L_h}\frac{\Delta_i}{T}\right\}\right] \\ &= \sum_{i=0}^{I-1} w_i\, h[l+\Delta_i/T]_{L_h}\, x_m[k+(l+\Delta_i/T)]_{L_h}, \end{aligned} \tag{3.21}$$

where $(n)_{L_h}$ denotes that the index n is evaluated modulo L_h. We can see from (3.21) that the circular convolution is different from the linear convolution as described in [40, 42]. The time-domain output of the frequency-domain beamformer might contain the circular aliasing.

There are two techniques which can perform the linear convolution using the DFT, that is, the overlap-save and overlap-add methods [40]. By overlapping elements of the data sequence and discarding the components of the circular shift from the output of the DFT product, the linear convolution of a finite length sequence and an infinite-length sequence is obtained. However, in order to avoid the circular aliasing, both methods require two additional DFT for the beamformer's weights. It is computationally expensive.

The circular aliasing can be also avoided by replacing the STFT with one of the filter banks discussed in Chapter 4. It would be unnecessary in the filter bank system to perform the DFT additionally.

3.4 Null-steering Beamformer

The delay-and-sum beamformer can pick up a signal propagating from a direction of interest. However, it cannot suppress interference signals coming from other directions explicitly. How can we remove those undesired signals while emphasizing the desired signal? It could be achieved by a null-steering beamformer which is obtained by solving a multiple linear constraint problem.

Let us first define the steering vector for a source n at a (subband) frequency bin m as $\mathbf{d}_n(m)$. The weight vector of the delay-and-sum beamformer $\mathbf{w}_{\rm ds}(m)$ then satisfies a linear constraint

$$\mathbf{d}_n(m)^H \mathbf{w}_{\rm ds}(m) = c_g, \tag{3.22}$$

where c_g is a constant. Note that $\mathbf{d}_n(m)$ and $\mathbf{w}_{\rm ds}(m)$ are M-by-1 vectors.

We may generalize the notion of the linear constraint by introducing multiple liner constraints defined as

$$\mathbf{C}_n(m)^H \mathbf{w}_{\rm null}(m) = \mathbf{c}_g, \tag{3.23}$$

where the columns of the *constraint matrix* $\mathbf{C}_n(m)$ consist of the steering vectors and the *gain vector* $\mathbf{c}_g$ determines which source is emphasized or suppressed. The desired weight vector can be obtained by solving the linear equation (3.23).

For example, in the case that we would like to emphasize the source 0 and eliminate the other, (3.23) can be simplified as

$$[\,\mathbf{d}_0(m)\;\mathbf{d}_1(m)\,]^H\,\mathbf{w}_{\rm null}(m) = [\,1\;0\,]^T. \tag{3.24}$$

It follows the two constraints

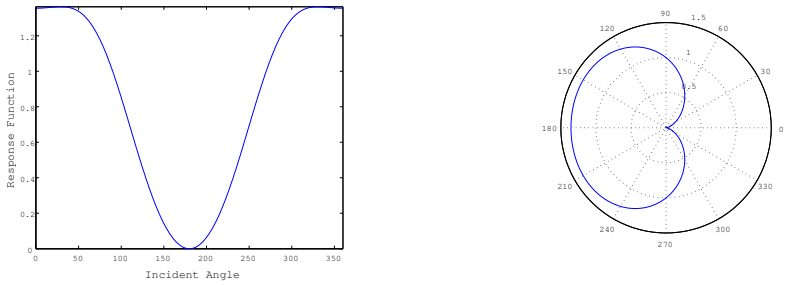
$$\mathbf{d}_0(m)^H \mathbf{w}_{\rm null}(m) = 1 \tag{3.25}$$

$$\mathbf{d}_1(m)^H \mathbf{w}_{\rm null}(m) = 0 \tag{3.26}$$

(3.25) implies that any signal coming from the direction associated with the steering vector $\mathbf{d}_0^H(m)$ is kept unity. We can also see from (3.26) that the interference signal propagating along the direction indicated with the vector $\mathbf{d}_1(m)$ is nulled.

As mentioned above, we can null out interference signals while maintaining the unity constraint to the look direction if the positions of all the sources are known. Such a beamformer is called the null-steering beamformer.

Figure 3.19, 3.20 and 3.21 illustrate beam patterns of the null-steering beamformers of the equally spaced linear array in the case that the nulls are placed on the waves propagating with incident angle 0 and the distortionless constraints

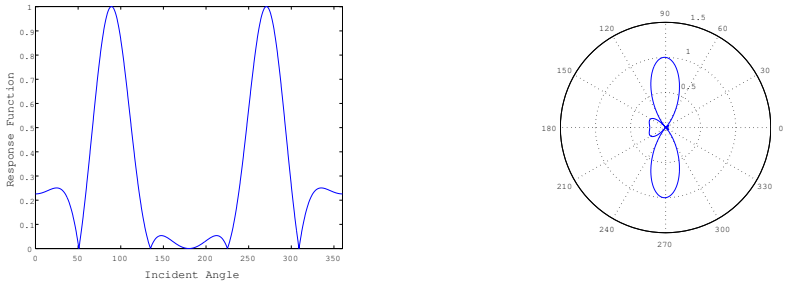


(a) (b)

Figure 3.19: Beam patterns of the null-steering beamformer with 2 linear constraints and 7 sensors at $f = 400$ Hz and $d = 0.05\ m$ ($\lambda/d = 17.19$): (a) in Cartesian coordinates and (b) in polar coordinates.

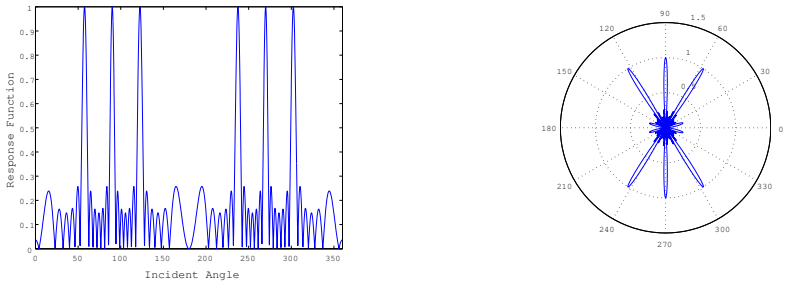
are put on arrival angle $\pi/2$. Figure 3.19 and 3.20 show the plots at frequency $f = 400$ Hz with distances between the sensors $0.05\ m$ and $0.2\ m$, respectively. The beam patterns at frequency $f = 3200$ Hz are shown in Figure 3.21 where the distance between the sensors is $0.2\ m$. It is clear from these figures that the responses to the incident angle 0 are zero while they are kept unity against the angle $\pi/2$.

The beam patterns of the null-steering beamformers have the same characteristics as those of the delay-and-sum beamformers, that is, the larger distance between the sensors makes the beamwidth smaller while it can cause spatial aliasing as seen in Figure 3.21. However, we have to pay more careful attention to the fact that interference signals could be emphasized by the null-steering beamformer. We can see from 3.19 that the responses for the signals coming from the angle $\theta < -\pi/2$ and $\theta > \pi/2$ exceed unity.



(a) (b)

Figure 3.20: Beam patterns of the null-steering beamformer with 2 linear constraints and 7 sensors at $f = 400$ Hz and $d = 0.2\ m$ ($\lambda/d = 4.297$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 3.21: Beam patterns of the null-steering beamformer with 2 linear constraints and 7 sensors at $f = 3200$ Hz and $d = 0.2\ m$ ($\lambda/d = 0.5371$): (a) in Cartesian coordinates and (b) in polar coordinates.

Chapter 4

Filter Bank Systems

In this chapter, we discuss a more general form of filter bank implementation which includes the STFT implementation. In the filter bank framework, band-pass filters with high stop-band suppression are generally used to obtain subband components of an input signal. Accordingly, such subband components are more separated from each other than the frequency components obtained by STFT. The better separation of each frequency component could lead to the better convergence performance of estimation of beamformer's weights at each frequency bin. In addition, computational amounts are significantly reduced by down-sampling the subband signals to a lower rate (decimation). Moreover, a longer analysis duration can be taken into account without too high frequency resolution while the analysis length in the simple SFFT analysis is equal to the number of the frequency bins.

Various filter bank design approaches have been proposed for speech coding [11]. However, the filter bank design for subband adaptive filtering poses problems not encountered in speech coding [9, 10, 30]. De Haan showed in [10] that the perfect reconstruction (PR) filter banks were not suitable for beamforming applications because PR is achieved through aliasing cancellation [11, §5], which can reconstruct an input signal correctly only if the outputs of the individual subbands are *not* subject to arbitrary magnitude scaling and phase

shifts. In [9], de Haan et al. proposed a method to design analysis and synthesis prototypes for modulated filter banks so as to minimize the weighted combination of the *response error* and *aliasing distortion*. The filter banks proposed in [9] are referred to as de Haan filter banks here.

This chapter shows that the response error defined in [9] can be driven to null by constraining the analysis and synthesis prototypes to be *Nyquist(M)* filters [11, §4.6.1]. Thereafter, the minimization of the aliasing distortion is shown to reduce to the solution of an eigenvalue problem in the case of the analysis prototype, and to the solution of a set of linear equations in the case of the synthesis prototype. We also discuss the performance limitations of the filter banks due to numerical problems, and propose an alternate solution for a special case for which we can eliminate not only the total response error but also residual aliasing distortion completely.

The rest of this chapter is organized as follows. In Section 4.1, the definition of the uniform DFT filter bank is reviewed and the notation to be used throughout this chapter is introduced. Most importantly, Section 4.1 describes expressions for the total response error and residual aliasing distortion that will subsequently be minimized. In Section 4.2, we consider the design of suitable analysis and synthesis prototypes for the modulated filter banks discussed in Section 4.1. In particular, Sections 4.2.1 and 4.2.2 briefly present the prototypes design methods of [9], and then show how slight modifications of those techniques can produce prototypes with zero response error and the minimal aliasing distortion. Section 4.3 presents an alternate method which provides null residual aliasing distortion as well as zero total response error in a special case. In Section 4.5, we analyze new filter banks and compare the total response errors and aliasing distortions obtained with them to those obtained with the de Haan filter bank design.

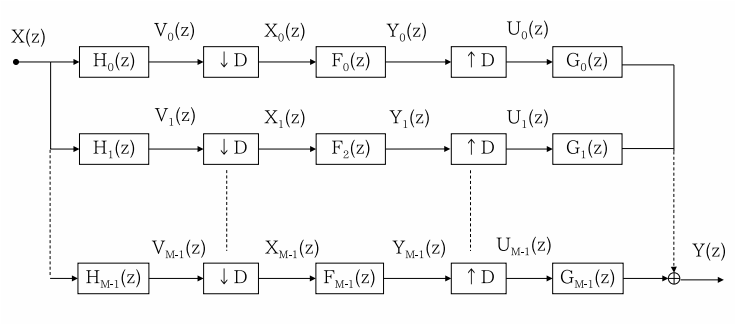


Figure 4.1: Schematic of a modulated subband analysis-synthesis filter bank.

4.1 Modulated Filter Bank

Figure 4.1 illustrates the filter bank implementation in the Z-transform in the case that single channel data is processed [11]. The filter bank analysis splits the speech signal into frequency *subbands*. In Figure 4.1, a *uniform DFT filter bank* with M subbands and a *decimation factor* of D is described. The impulse responses $h_m[n]$ of the analysis filters are obtained by modulating a single prototype $h[n]$ according to

$$h_m[n] = h[n]\, W_M^{-mn} \leftrightarrow H_m(z) = H(zW_M^m), \tag{4.1}$$

where $W_M = e^{-j2\pi/M}$ denotes the M-th root of unity. As noted in [9][10], however, PR is achieved through the aliasing cancellation, which functions properly only when the outputs of the individual subbands are *not* subject to arbitrary magnitude scalings and phase shifts. Hence, the PR design is not suitable for beamforming and adaptive filtering.

Following [9], we will define a separate prototype $g[n]$ for the synthesis bank, and stipulate that the individual prototypes $g_m[n]$ are related to $g[n]$ according to

$$g_m[n] = g[n]\, W_M^{-mn} \leftrightarrow G_m(z) = G(zW_M^m). \tag{4.2}$$

The outputs $V_m(z)$ of the analysis filters can be expressed as

$$V_m(z) = H_m(z)X(z) = H(zW_M^m)\, X(z). \tag{4.3}$$

Then the decimators expand the spectrums [11, §4.2] according to

$$\begin{aligned} X_m(z) &= \frac{1}{D}\sum_{d=0}^{D-1} V_m(z^{1/D}W_D^d) \\ &= \frac{1}{D}\sum_{d=0}^{D-1} H(z^{1/D}W_M^m W_D^d)X(z^{1/D}W_D^d). \end{aligned} \tag{4.4}$$

The last equation indicates that $X_m(z)$ consists of the sum of a stretched output of the mth filter bank and $D-1$ aliasing terms.

At this point, the fixed subband weights F_m can be applied to the decimated signals to achieve the desired adaptive filtering effect

$$Y_m(z) = F_m X_m(z). \tag{4.5}$$

The expanders then compress the signals $Y_m(z)$ according to

$$U_m(z) = Y_m(z^D) = \frac{1}{D}F_m \sum_{d=0}^{D-1} H(zW_M^m W_D^d)X(zW_D^d). \tag{4.6}$$

In the last step, the signals $U_m(z)$ are processed by the synthesis filters $G_m(z)$ in order to suppress the spectral images created by the expanders, and the outputs of the synthesis filters are summed together according to

$$Y(z) = \sum_{m=0}^{M-1} U_m(z)\, G_m(z). \tag{4.7}$$

The final relation between the input and output signals can be expressed as

$$Y(z) = \frac{1}{D}\sum_{d=0}^{D-1} X(zW_D^d) \sum_{m=0}^{M-1} F_m\, H(zW_M^m W_D^d)\, G(zW_M^m). \tag{4.8}$$

Upon defining

$$A_{m,d}(z) = \frac{1}{D}\, F_m\, H(zW_M^m W_D^d)\, G(zW_M^m), \tag{4.9}$$

the output relation (4.8) can be written more conveniently as

$$Y(z) = \sum_{d=0}^{D-1} A_d(z)\, X(zW_D^d), \tag{4.10}$$

where

$$A_d(z) = \sum_{m=0}^{M-1} A_{m,d}(z). \tag{4.11}$$

The transfer function $A_0(z)$ produces the desired signal, while the remaining transfer functions $A_d(z)$ for $d = 1, \ldots, D-1$ give rise to the residual aliasing distortion in the output signal.

Subband beamforming deals with multi-channel inputs. As you might easily imagine, it will be realized by adding the blocks of the analysis filter banks for the multiple channels to Figure 4.1 and replacing $F_m X$ with beamformer's weights so that filtering can be performed on every channel data.

4.2 Prototype Design

4.2.1 Analysis Prototype Design

In order to design the analysis prototype $h[n]$, de Haan *et al.* [9] define the objective function

$$\epsilon_{\mathbf{h}} = \alpha_{\mathbf{h}} + \beta_{\mathbf{h}}, \tag{4.12}$$

where the *passband response error* is

$$\alpha_{\mathbf{h}} = \frac{1}{2\omega_p} \int_{-\omega_p}^{\omega_p} \left| H(e^{j\omega}) - H_{\mathrm{d}}(e^{j\omega}) \right|^2 d\omega, \tag{4.13}$$

and the *inband-aliasing distortion* is given by

$$\beta_{\mathbf{h}} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{d=1}^{D-1} \left| H(e^{j\omega/D} W_D^d) \right|^2 d\omega. \tag{4.14}$$

In (4.13) the *desired filter bank response* $H_{\mathrm{d}}(e^{j\omega})$ is assumed to correspond to a pure delay of τ_H samples, such that

$$H_{\mathrm{d}}(e^{j\omega}) = e^{-j\omega\tau_H}. \tag{4.15}$$

Notice that the time delay corresponds to the phase shift in the frequency domain. Defining

$$\mathbf{h} = \begin{bmatrix} h[0] & h[1] & \cdots h[L_{\mathbf{h}} - 1] \end{bmatrix}^T, \tag{4.16}$$

$$\phi_{\mathbf{h}}(z) = \begin{bmatrix} 1 & z^{-1} & \cdots z^{-(L_{\mathbf{h}}-1)} \end{bmatrix}^T, \tag{4.17}$$

they then demonstrate that the passband response error can be expressed as

$$\alpha_{\mathbf{h}} = \mathbf{h}^T \mathbf{A} \mathbf{h} - 2\mathbf{h}^T \mathbf{b} + 1, \tag{4.18}$$

where

$$\mathbf{A} = \frac{1}{2\omega_p} \int_{-\omega_p}^{\omega_p} \phi_{\mathbf{h}}(e^{j\omega}) \phi_{\mathbf{h}}^H(e^{j\omega}) d\omega, \tag{4.19}$$

$$\mathbf{b} = \frac{1}{2\omega_p} \int_{-\omega_p}^{\omega_p} Re\left\{ e^{j\omega\tau_H} \phi_{\mathbf{h}}(e^{j\omega}) \right\} d\omega. \tag{4.20}$$

Based on (4.19–4.20), the components of $\mathbf{A}$ and $\mathbf{b}$ can be expressed as

$$A_{i,j} = \frac{\sin(\omega_p(j-i))}{\omega_p(j-i)}, \tag{4.21}$$

$$b_i = \frac{\sin(\omega_p(\tau_H - i))}{\omega_p(\tau_H - i)}. \tag{4.22}$$

The inband-aliasing term (4.14) can be expressed as

$$\beta_{\mathbf{h}} = \frac{1}{2\pi} \sum_{d=1}^{D-1} \mathbf{h}^T \left[\int_{-\pi}^{\pi} \phi_{\mathbf{h}} \left(e^{j\frac{\omega}{D}} W_D^d \right) \phi_{\mathbf{h}}^H \left(e^{j\frac{\omega}{D}} W_D^d \right) d\omega \right] \mathbf{h}. \tag{4.23}$$

The last equation can be rewritten as

$$\beta_{\mathbf{h}} = \mathbf{h}^T \mathbf{C} \mathbf{h}, \tag{4.24}$$

where

$$\mathbf{C} = \frac{1}{2\pi} \sum_{d=1}^{D-1} \int_{-\pi}^{\pi} \phi_{\mathbf{h}} \left(e^{j\frac{\omega}{D}} W_D^d \right) \phi_{\mathbf{h}}{}^H \left(e^{j\frac{\omega}{D}} W_D^d \right) d\omega. \tag{4.25}$$

The components of $\mathbf{C}$ can then be expressed as

$$C_{i,j} = \frac{\varphi[j-i] \sin\left(\frac{\pi(j-i)}{D}\right)}{\pi(j-i)} \tag{4.26}$$

where

$$\varphi[n] = D \sum_{k=-\infty}^{\infty} \delta[n - kD] - 1.$$

Combining all terms above, de Haan *et al.* then seek to minimize the objective function

$$\begin{aligned} \epsilon_{\mathbf{h}} &= \alpha_{\mathbf{h}} + \beta_{\mathbf{h}} \\ &= \mathbf{h}^T (\mathbf{A} + \mathbf{C}) \mathbf{h} - 2\mathbf{h}^T \mathbf{b} + 1. \end{aligned} \tag{4.27}$$

Thus, the prototype $\mathbf{h}$ proposed in [9] must satisfy

$$(\mathbf{A} + \mathbf{C}) \mathbf{h} = \mathbf{b}. \tag{4.28}$$

Polyphase Components

Any given filter function $H(z)$ can be decomposed as

$$H(z) = \sum_{l=0}^{M-1} z^{-l} \, E_l(z^M), \tag{4.29}$$

where

$$E_l(z) \triangleq \sum_{n=-\infty}^{\infty} e_l(n) \, z^{-n} \tag{4.30}$$

and

$$e_l[n] \triangleq h(Mn + l), \text{ for all } 0 \leq l \leq M-1. \tag{4.31}$$

Equation (4.29) is known as the Type 1 polyphase representation of $H(z)$ and the set $\{E_l(z)\}$ is, by definition, composed of the Type 1 polyphase components of $H(z)$; see [11, §4.3]. The Type 1 polyphase components are very useful for the efficient implementation of a modulated *analysis* filter bank. The implementation of a modulated *synthesis* bank typically relies on the Type 2 polyphase representation:

$$H(z) = \sum_{l=0}^{M-1} z^{-(M-1-l)} \, R_l(z^M), \tag{4.32}$$

where the set of Type 2 polyphase components $\{R_l(z)\}$ are obtained from permutation of the Type 1 polyphase components,

$$R_l(z) = E_{M-1-l}(z). \tag{4.33}$$

Nyquist(M) Filters

Suppose that a filter function $H(z)$ has been represented in Type 1 polyphase form, and the 0-th polyphase component is constant, such that

$$H(z) = c + z^{-1} \, E_1(z^M) + \cdots + z^{-(M-1)} \, E_{M-1}(z^M). \tag{4.34}$$

A filter with this property is said to be a *Nyquist(M)* or *M-th band filter* [11, §4.6.1], and its impulse response clearly satisfies

$$h[Mn] = \begin{cases} c, & n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.35}$$

The definition in (4.34) can be generalized by assuming that

$$\begin{aligned} H(z) = cz^{-m_{\mathrm{d}}M} + z^{-1}\,E_1(z^M) + \cdots \\ +z^{-(M-1)}\,E_{M-1}(z^M). \end{aligned} \tag{4.36}$$

The impulse response of $H(z)$ must then satisfy

$$h[Mn] = \begin{cases} c, & n = m_{\mathrm{d}} \\ 0, & \text{otherwise} \end{cases} \tag{4.37}$$

If $H(z)$ satisfies (4.34) with $c = 1/M$, then

$$\sum_{k=0}^{M-1} H(zW^k) = Mc = 1, \tag{4.38}$$

where $W = e^{-j2\pi/M}$. Hence, all M uniformly shifted versions of $H(e^{j\omega})$ add up to a constant. Similarly, if $H(z)$ satisfies (4.36), then

$$\sum_{k=0}^{M-1} H(zW^k) = z^{-m_{\mathrm{d}}M}, \tag{4.39}$$

in which case, in the absence of decimation, the output of the analysis filter bank would be equivalent to the input delayed by $m_{\mathrm{d}}M$ samples.

Notice that (4.39) represents a much stronger condition than that aimed at by the minimization of (4.13), in that (4.39) implies the response error will vanish, not just for the pass band of a single filter, but for the entire working spectrum, including the transition bands between the passbands of adjacent filters. Hence, we can replace the term $\alpha_{\mathbf{h}}$ in the optimization criterion (4.12) with a constraint of the form (4.37), then minimize the inband-aliasing distortion (4.23) subject to this constraint. The inband-aliasing distortion reduces to (4.24), the optimization of which clearly admits the trivial solution $\mathbf{h} = \mathbf{0}$. To exclude this solution, we impose the additional constraint

$$\mathbf{h}^T\mathbf{h} = 1, \tag{4.40}$$

which is readily achieved through the method of *undetermined Lagrange multipliers*. We posit the modified objective function

$$f(\mathbf{h}) = \mathbf{h}^T\mathbf{C}\mathbf{h} + \lambda(\mathbf{h}^T\mathbf{h} - 1) \tag{4.41}$$

where λ is a *Lagrange multiplier*. Upon setting

$$\nabla f(\mathbf{h}) = \mathbf{0},$$

we find

$$\mathbf{C}\mathbf{h} + \lambda\mathbf{h} = \mathbf{0},$$

which implies

$$\mathbf{C}\mathbf{h} = -\lambda\mathbf{h}. \tag{4.42}$$

Hence, $\mathbf{h}$ is clearly an eigenvector of $\mathbf{C}$. Moreover, in order to ensure $\mathbf{h}$ minimizes (4.24), it must be that eigenvector associated with the *smallest* eigenvalue of $\mathbf{C}$. Note that, in order to ensure that $\mathbf{h}$ satisfies either (4.35) or (4.37), we must delete those rows and columns of $\mathbf{C}$ corresponding to the components of $\mathbf{h}$ that are identically zero. We then solve the eigenvalue problem (4.42) for the remaining components of $\mathbf{h}$, and finally reassemble the complete prototype by appropriately concatenating the zero and non-zero components. This is similar to the construction of the *eigenfilter* described in [11, §4.6.1].

4.2.2 Synthesis Prototype Design

In order to design the synthesis prototype, de Haan *et al.* [9] take as an objective function

$$\epsilon_{\mathbf{g}}(\mathbf{h}) = \gamma_{\mathbf{g}}(\mathbf{h}) + \delta_{\mathbf{g}}(\mathbf{h}) \tag{4.43}$$

where the *total response error* is defined as

$$\gamma_{\mathbf{g}}(\mathbf{h}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|A_0(e^{j\omega}) - e^{-j\omega\tau_T}\right|^2 d\omega, \tag{4.44}$$

the total analysis-synthesis filter bank delay is denoted as τ_T, and the *residual aliasing distortion* is

$$\delta_{\mathbf{g}}(\mathbf{h}) = \frac{1}{2\pi}\sum_{d=1}^{D-1}\sum_{m=0}^{M-1}\int_{-\pi}^{\pi} \left|A_{m,d}(e^{j\omega})\right|^2 d\omega. \tag{4.45}$$

Through manipulations similar to those used in deriving the quadratic objective criterion for the analysis filter bank, it can be shown that

$$\gamma_{\mathbf{g}}(\mathbf{h}) = \mathbf{g}^T\mathbf{E}\mathbf{g} - 2\mathbf{g}^T\mathbf{f} + 1. \tag{4.46}$$

The components of $\mathbf{E}$ and $\mathbf{f}$ are given by

$$E_{i,j} = \frac{M^2}{D^2} \sum_{k=-\infty}^{\infty} h^*[kM - i]\, h[kM - j] \tag{4.47}$$

$$f_i = \frac{M}{\pi D}\, h[\tau_T - i]. \tag{4.48}$$

Similarly, the quadratic form for the residual aliasing distortion is

$$\delta_{\mathbf{g}}(\mathbf{h}) = \mathbf{g}^T \mathbf{P} \mathbf{g}, \tag{4.49}$$

where the components of $\mathbf{P}$ are given by

$$P_{i,j} = \frac{M}{D^2} \sum_{l=-\infty}^{\infty} h^*[l + j]\, h[l + i]\, \varphi[i - j],$$

$$\varphi[n] = D \sum_{k=-\infty}^{\infty} \delta[n - kD] - 1.$$

De Haan *et al.* [9] introduced a weighting factor v to emphasize either the total response error (for $0 < v < 1$) or residual aliasing distortion (for $v > 1$):

$$\epsilon_{\mathbf{g}}(\mathbf{h}) = \gamma_{\mathbf{g}}(\mathbf{h}) + v\delta_{\mathbf{g}}(\mathbf{h}) \tag{4.50}$$

$$= \mathbf{g}^T(\mathbf{E} + v\mathbf{P})\mathbf{g} - 2\mathbf{g}^T\mathbf{f} + 1. \tag{4.51}$$

Hence, their synthesis prototype $\mathbf{g}$ must satisfy

$$(\mathbf{E} + v\mathbf{P})\mathbf{g} = \mathbf{f}. \tag{4.52}$$

Nyquist(M) Constraint

As with the analysis prototype, we can now impose the Nyquist(M) constraint on the *complete analysis-synthesis prototype* $(h * g)[n]$ such that

$$(h * g)[Mn] = \begin{cases} c, & n = m_{\mathrm{d}}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.53}$$

in which case the total response error (4.44) must be identically zero. Subject to this constraint, we minimize the residual aliasing distortion (4.51). Satisfaction

of (4.53) clearly reduces to a set of linear constraints of the form

$$\begin{aligned} \mathbf{g}^T\mathbf{h}_{-m+1} &= 0, \\ &\vdots \\ \mathbf{g}^T\mathbf{h}_0 &= c, \\ &\vdots \\ \mathbf{g}^T\mathbf{h}_{m-1} &= 0, \end{aligned} \tag{4.54}$$

where $\mathbf{h}_k$ is obtained by shifting a time-reversed version of $\mathbf{h}$ by kM samples and padding with zeros as needed. Equation (4.54) can be rewritten as

$$\mathbf{g}^T\mathbf{H} = \mathbf{c}^T, \tag{4.55}$$

where

$$\mathbf{H} = [\mathbf{h}_{-m+1}, \ldots, \mathbf{h}_0, \ldots, \mathbf{h}_{m-1}], \tag{4.56}$$

$$\mathbf{c}^T = [0, \ldots, c, \ldots, 0]. \tag{4.57}$$

For the constrained minimization problem at hand, we again draw upon the method of undetermined Lagrange multipliers and formulate the objective function

$$f(\mathbf{g}) = \mathbf{g}^T\mathbf{P}\mathbf{g} + (\mathbf{g}^T\mathbf{H} - \mathbf{c}^T)\lambda, \tag{4.58}$$

where $\lambda = [\lambda_{-m+1}, \ldots, \lambda_0, \ldots, \lambda_{m+1}]^T$. Setting

$$\nabla f(\mathbf{g}) = 2\mathbf{P}\mathbf{g} + \mathbf{H}\lambda = \mathbf{0}, \tag{4.59}$$

we find

$$\mathbf{g} = -\frac{1}{2}\mathbf{P}^{-1}\mathbf{H}\lambda. \tag{4.60}$$

The values of the multipliers $\{\lambda_k\}$ can be determined by substituting (4.60) into (4.55) and solving

$$\lambda = -2\left(\mathbf{H}^T\mathbf{P}^{-1}\mathbf{H}\right)^{-1}\mathbf{c}. \tag{4.61}$$

By substituting (4.61) into (4.60), we finally obtain a synthesis prototype

$$\mathbf{g} \quad = \quad \mathbf{P}^{-1}\mathbf{H}\left(\mathbf{H}^T\mathbf{P}^{-1}\mathbf{H}\right)^{-1}\mathbf{c}. \tag{4.62}$$

4.3 Alternative Method for Singular C and P

The optimal prototypes can be obtained by solving (4.42) and (4.62) if the matrices $\mathbf{C}$ and $\mathbf{P}$ are not singular. When those matrices are ill-conditioned, however, a different solution is required.

Theoretically speaking, the matrices $\mathbf{C}$ and $\mathbf{P}$ should not be singular because they are positive definite. From (4.14) and (4.24), the matrix $\mathbf{C}$ clearly satisfies

$$\mathbf{h}^T\mathbf{C}\mathbf{h} \geq 0. \tag{4.63}$$

With (4.45) and (4.49), we also find

$$\mathbf{g}^T\mathbf{P}\mathbf{g} \geq 0. \tag{4.64}$$

It is obvious from (4.63) and (4.64) that the matrices $\mathbf{C}$ and $\mathbf{P}$ are positive definite unless the frequency responses of the analysis and synthesis prototypes are identically zero in the stopbands. In our cases, those matrices should be positive definite and accordingly invertible. We have observed, however, that as energy in the stopbands of the analysis and synthesis prototypes approaches zero, the matrices $\mathbf{C}$ and $\mathbf{P}$ became *computationally* singular due to the limitations of floating point accuracy. This typically occurs when the decimation factor D is small compared to the length of the prototype filter $L_{\mathbf{h}}$. In those cases when $\mathbf{C}$ singular, we can denote its nullspace as $\mathbf{C}_{\text{null}}$, which consists of those column vectors $\mathbf{q} \in \mathbf{R}^n : \mathbf{C}\mathbf{q} = \mathbf{0}$. The *singular value decomposition* (SVD) [43] can be used in order to obtain a basis for the nullspace of $\mathbf{C}$. Under the SVD, $\mathbf{C}$ is decomposed into

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{4.65}$$

The bases of the null space of $\mathbf{C}$ can be obtained from the columns of $\mathbf{V}$, which correspond to singular values below a threshold. In this work, the threshold σ is chosen such that

$$\sigma = \max(m, n) \times \max(\sigma_i) \times \epsilon_f$$

where m and n are respectively the number of rows and columns in $\mathbf{C}$, the ith singular value is denoted by σ_i, and ϵ_f is the floating point accuracy of the machine used for SVD computations.

Obviously, the inband-aliasing distortion can be driven to null by an analysis prototype which is represented as a linear combination of the basis vectors of the nullspace $\mathbf{h} = \mathbf{C}_{\text{null}}\,\mathbf{x}$. The free parameters $\mathbf{x}$ are determined so as to minimize the passband response error (4.18), the solution of which can be expressed as

$$\mathbf{h} = \mathbf{C}_{\text{null}}(\mathbf{C}_{\text{null}}^T\mathbf{A}\mathbf{C}_{\text{null}})^{-1}\mathbf{C}_{\text{null}}^T\mathbf{b}, \tag{4.66}$$

where rows and columns of $\mathbf{C}_{\text{null}}$, $\mathbf{A}$ and $\mathbf{b}$, corresponding to the components of $\mathbf{h}$ that are identically zero, are deleted, and $\mathbf{h}$ is reassembled so as to keep the Nyquist(M) constraint.

For the synthesis prototype design, we can also eliminate residual aliasing distortion (4.49) in a similar manner. Denoting the nullspace of $\mathbf{P}$ as $\mathbf{P}_{\text{null}}$, we can express the synthesis prototype as $\mathbf{g} = \mathbf{P}_{\text{null}}\mathbf{y}$. Then by substituting it into (4.55), we have

$$\mathbf{y} = (\mathbf{H}^T\mathbf{P}_{\text{null}})^{+}\mathbf{c} \tag{4.67}$$

where $(\cdot)^{+}$ indicates the peseudoinverse of $(\cdot)$. If the number of column vectors of $\mathbf{P}_{\text{null}}$ is greater than or equal to $2m-1$, we can find a synthesis prototype $\mathbf{g} = \mathbf{P}_{\text{null}}\mathbf{y}$ with zero total response error and residual aliasing distortion. We finally express the synthesis prototype with basis of the nullspace as

$$\mathbf{g} = \mathbf{P}_{\text{null}}(\mathbf{H}^T\mathbf{P}_{\text{null}})^{+}\mathbf{c}. \tag{4.68}$$

In practice, as the inband-aliasing distortion is very small, $\mathbf{P}$ becomes practically singular. In that case, with the method described here, we can achieve zero inband-aliasing and residual aliasing distortions.

4.4 Design Examples

Energy in the stopband of the filters results in aliasing. The stopband attenuation is one of the important factors to indicate how good a prototype is. Figures 4.2, 4.3 and 4.4 show the frequency responses of the analysis, synthesis and composite analysis-synthesis prototypes respectively. Each figure presents the frequency responses of a uniform DFT filter bank using the PR prototype

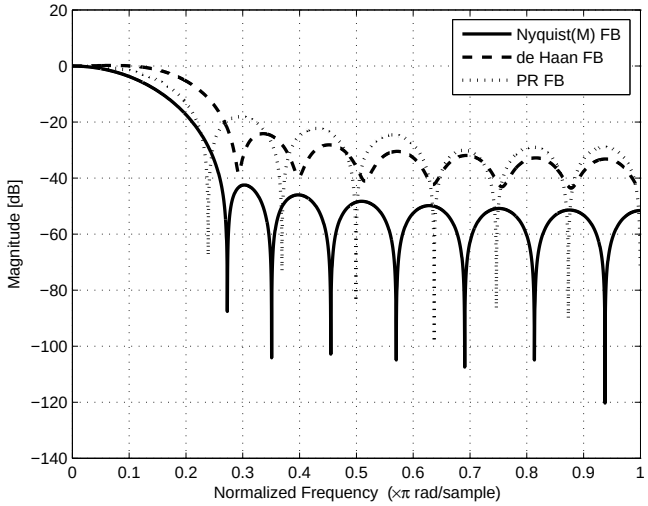


Figure 4.2: Frequency response of analysis filter bank prototypes with $M = 8$ subbands, decimation factor $D = 4$, and filter length $L_{\mathbf{h}} = 16$.

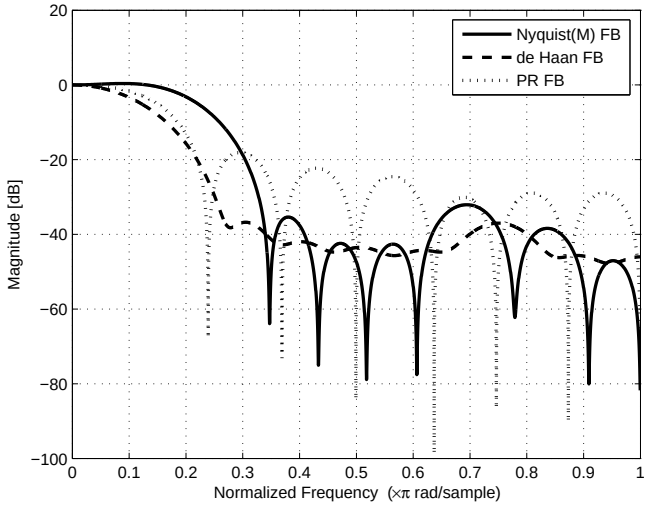


Figure 4.3: Frequency response of synthesis filter bank prototypes with $M = 8$ subbands , decimation factor $D = 4$, and filter length $L_{\mathbf{h}} = 16$.

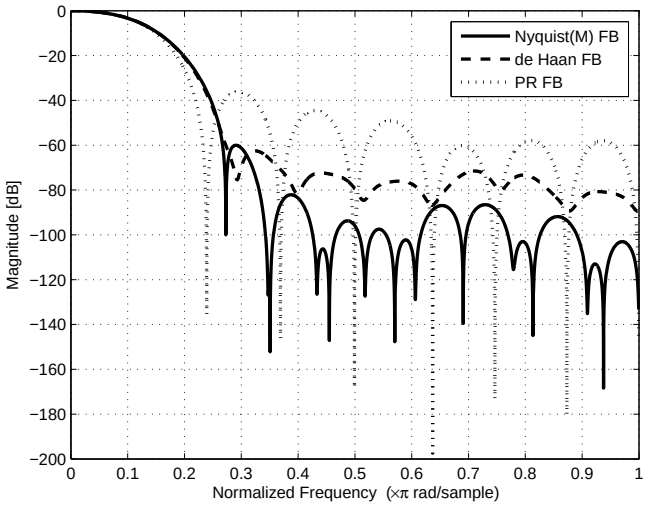


Figure 4.4: Frequency response of proposed composite analysis-synthesis filter bank prototypes with $M = 8$ subbands, decimation factor $D = 4$ and filter length $L_{\mathbf{h}} = 16$.

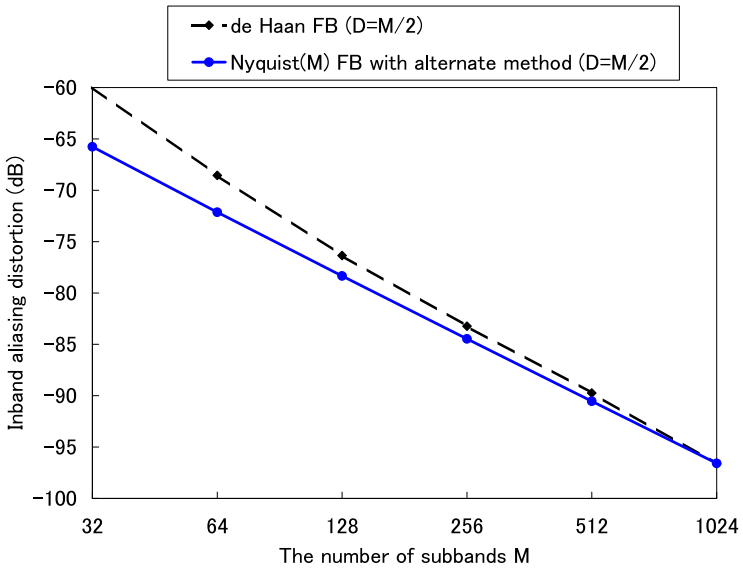


Figure 4.5: Inband aliasing distortion $\beta_{\mathbf{h}}$ for the number of subbands M. The filter length is set to $L_{\mathbf{h}} = 2M$.

design (PR FB), de Haan prototype design (de Haan FB), and the proposed prototype design (Nyquist(M) FB), where the number of subbands is $M = 8$ and the decimation factor is $D = 4$. From those figures, we can readily see that the filter banks designed by the proposed algorithm provide the highest suppression in the stopband, followed by de Haan prototype and then by the PR filter prototype. Again, in the case that arbitrary magnitude scalings and phase shifts are applied to the subband samples, the PR property is not retained. Hence, it is important to minimize the stopband energy of each filter individually.

4.5 Evaluation of Errors in Filter Prototypes

From the inband and residual aliasing distortions, we can predict the robustness of filter banks for aliasing caused by arbitrary magnitude scaling and phase shifts [10].

A relationship between the aliasing distortions and the number of subbands

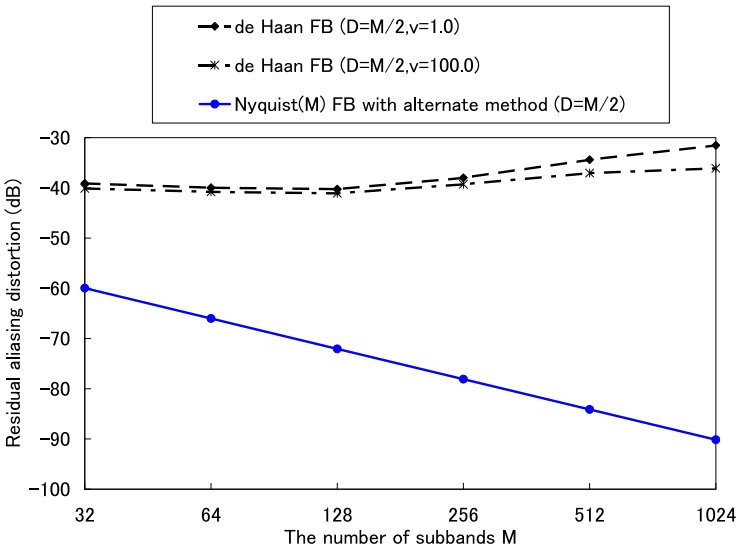


Figure 4.6: Residual aliasing distortion $\epsilon_{\mathbf{g}}(\mathbf{h})$ for the number of subbands M. The filter length is set to $L_{\mathbf{h}} = 2M$.

might be helpful for designing a beamforming system. Figures 4.5 and 4.6 show the inband and residual aliasing distortions against the number of subbands, where the decimation factor is set to $D = M/2$.

Inasmuch as decreasing the inband-aliasing distortion leads to smaller residual aliasing distortion, it is important to minimize the inband-aliasing distortion. As shown in Figure 4.5, the proposed Nyquist(M) filter prototype provides a smaller inband-aliasing distortion than the prototype designed by de Haan's algorithm, because the proposed design algorithm minimizes the inband-aliasing distortion directly while de Haan's minimize a linear combination of the passband response error and inband-aliasing distortion.

We can see from Figure 4.6 that the proposed algorithm can keep the residual aliasing distortion much lower than the conventional method. It is also clear from Figure 4.6 that the residual aliasing distortion of the Nyquist(M) filter banks monotonically decreases with respect to the number of subbands M while those of de Haan filter banks are rather invariant to it. De Haan's algorithm

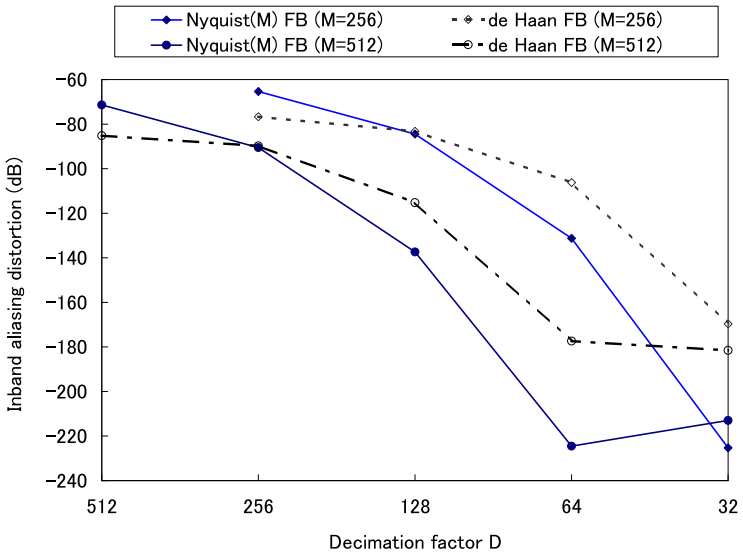


Figure 4.7: Inband-aliasing distortion $\beta_{\mathbf{h}}$ for decimation factor D. The number of subbands is $M = 256$ or $M = 512$ and the filter length is set to $L_{\mathbf{h}} = 2M$.

minimizes the linear combination of the total response error and residual aliasing distortion, equation (4.50). Hence, the additional term of the total response error $\gamma_{\mathbf{g}}(\mathbf{h})$ prevents the residual aliasing error $\delta_{\mathbf{g}}(\mathbf{h})$ from being suppressed. In contrast, the new design technique minimizes the residual aliasing distortion only while keeping zero total response error. As a result, the residual aliasing distortion of the Nyquist(M) filter simply decreases as M increases, due mostly to the increase of the number of free parameters with respect to the number of constraints.

The aliasing errors can be also reduced by decreasing the decimation factor D although it increases the computational cost associated with adaptive processing. Figure 4.7 presents the inband-aliasing distortions for decimation factor D with de Haan's and proposed filter banks, where each line corresponds to a number of subbands, $M = 256$ or 512, and the filter length is set to $2M$. It is clear from Figure 4.7 that the proposed method suppresses the inband-aliasing distortion more than de Haan's algorithm in most cases.

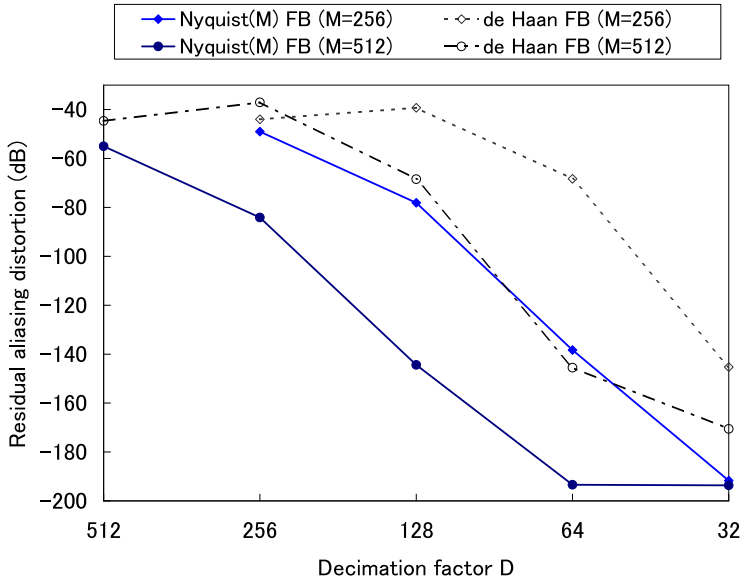


Figure 4.8: Residual aliasing distortion $\epsilon_{\mathbf{g}}(\mathbf{h})$ for decimation factor D. The number of subbands is M= 256 or M = 512 and the filter length is set to $L_{\mathbf{h}} = 2M$.

In calculating the inband-aliasing distortion for the decimation factor, it was observed that the matrix $\mathbf{C}$ was singular when the number of subbands and the decimation factor were set to $M = 256$ and $D \leq 32$ or $M = 512$ and $D \leq 64$. In such cases, we could find the nullspace and then use the alternate solution for the design of the analysis prototype instead of the eigen decomposition solution.

Figure 4.8 shows the residual aliasing distortion calculated with (4.49) in the same conditions as Figure 4.7. In Figure 4.8, de Haan filter banks are calculated with weighting factor $v = 100.0$. It is clear from Figure 4.8 that the smallest residual aliasing distortions are achieved with the proposed Nyquist(M) filter banks. It is worth noting that when $\mathbf{C}$ was singular, $\mathbf{P}$ was also singular and the synthesis prototypes were calculated with the bases of the nullspace of the matrix $\mathbf{P}$.

Figure 4.9 shows the residual aliasing distortions of the Nyquist(M) filter banks with the alternate method and without it, where the number of sub-

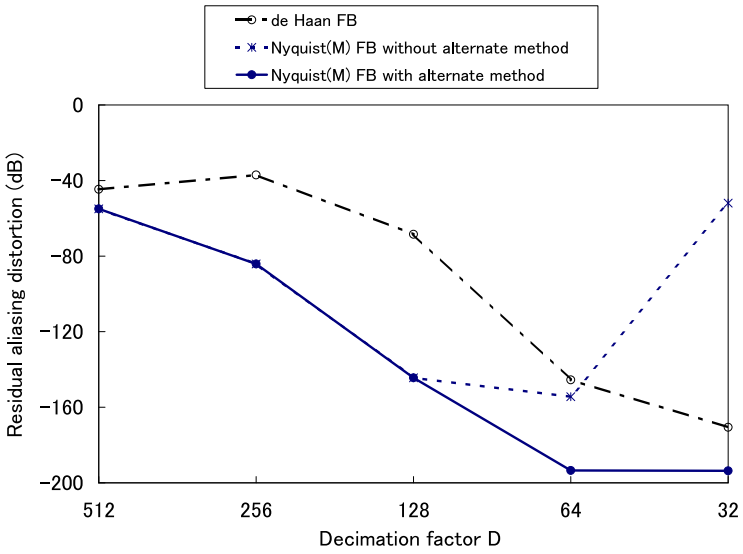


Figure 4.9: Comparison of the Nyquist(M) filter banks designed with the alternate method and without it. The number of subbands is M = 512 and the filter length is set to $L_{\mathbf{h}} = 2M$.

bands is set to 512. As a reference, the residual aliasing distortion of the de Haan filter bank is also shown in Figure 4.9. In the case that the filterbanks whose decimation factor D is set to less than 128 are entirely designed based on (4.42) and (4.62), the obtained solutions are instable due to the singular matrices. It is clear from Figure 4.9 that the residual aliasing distortion does not decrease monotonically but increases when the matrices $\mathbf{C}$ and $\mathbf{P}$ are singular. The nullspace based method can suppress the aliasing distortion even if these matrices are singular.

It could be important to know when the matrices $\mathbf{C}$ and $\mathbf{P}$ are ill-conditioned and computationally singular. We show common logarithms of *condition numbers* of those matrices in Figure 4.10. It is generally considered that a matrix is ill-conditioned when the condition number is too big, i.e. close to a reciprocal of floating point accuracy which is described as the threshold in Figure 4.10. As indicated in Figure 4.10, the smaller the decimation factor is set, the larger the

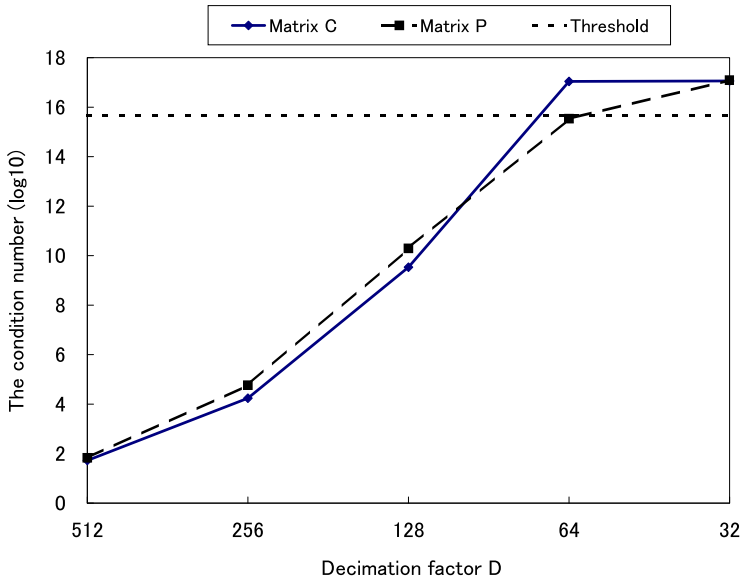


Figure 4.10: The common logarithm of the condition number of $\mathbf{C}$ and $\mathbf{P}$ for decimation factor D. The number of subbands is M = 512 and the filter length is set to $L_{\mathbf{h}} = 2M$.

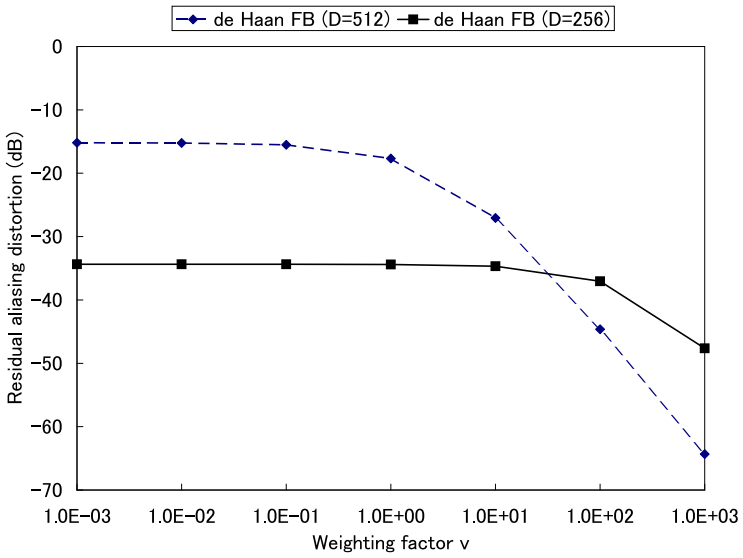


Figure 4.11: Residual aliasing distortion $\epsilon_{\mathbf{g}}(\mathbf{h})$ for weighting factor v. The number of subbands is M = 512 and the filter length is $L_{\mathbf{h}} = 2M$.

condition number becomes. The condition numbers reach the threshold in the case of decimation factor $D \leq 64$ in Figure 4.10.

One might intuitively consider that the residual aliasing distortion would decrease for the decimation factor monotonically. However, Figure 4.8 shows that each curve of the residual aliasing distortion of de Haan's filter bank has a peak at $D = M/2$.

In order to look further into the reason, we calculated the residual aliasing distortions with $D = 256$ and $D = 512$. Figure 4.11 shows the residual aliasing distortions for weighting factor v. From Figure 4.11 it is seen that the residual aliasing distortion of $D = 512$ is smaller than that of $D = 256$ in the case of $v \geq 100.0$.

We also show the total response errors for weighting factor v in Figure 4.12. It is clear from Figure 4.11 and Figure 4.12 that the residual aliasing distortion can be reduced by setting a large weighting factor v at the expense of the total response error. Notice that the total response error is zero in the Nyquist(M)

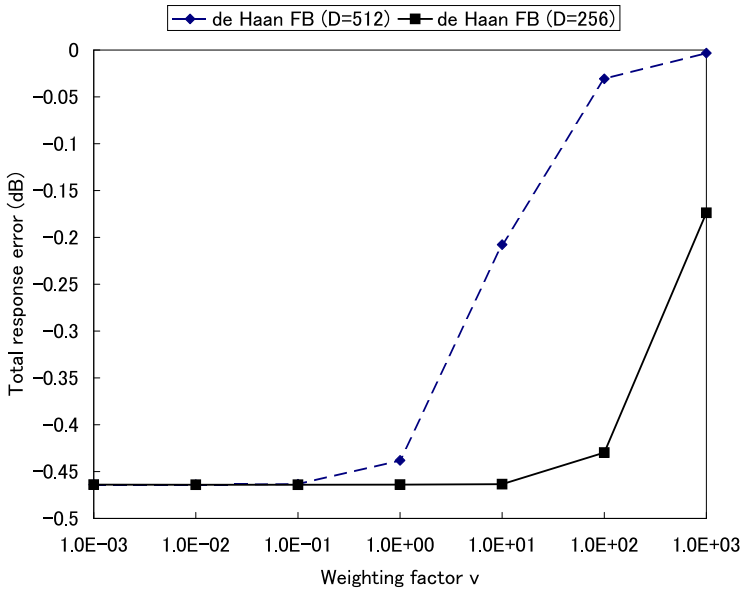


Figure 4.12: Total response error $\gamma_{\mathbf{g}}(\mathbf{h})$ for weighting factor v. The number of subbands is M = 512 and the filter length is $L_{\mathbf{h}} = 2M$.

filter bank.

Since the Nyquist(M) filter banks achieve zero total response error and their residual aliasing distortion can be driven down below machine precision through a suitable selection of the decimation factor, it could be that such filter banks provide reconstruction that is "perfect" up to machine precision . In order to investigate this possibility, we calculated the mean square (MS) error ϵ_{MS} between the input $x[n]$ and output $y[n]$ of the filter bank normalized by the MS amplitude of the input, which can be expressed as

$$\epsilon_{\mathrm{MS}} = 10 \log_{10} \frac{\sum_{n=0}^{N-1} (x[n] - y[n])^2}{\sum_{n=0}^{N-1} x^2[n]}.$$

Figure 4.13 shows the MS errors of the PR, Nyquist(M), de Haan filter banks for the decimation factor. Of course, the PR filter bank can reconstruct an exact input signal through the aliasing cancellation. Therefore, as shown in Figure 4.13, the PR filter bank provides the smallest MS error. The error of the PR filter bank is mainly because of the round-off error. We can also see from

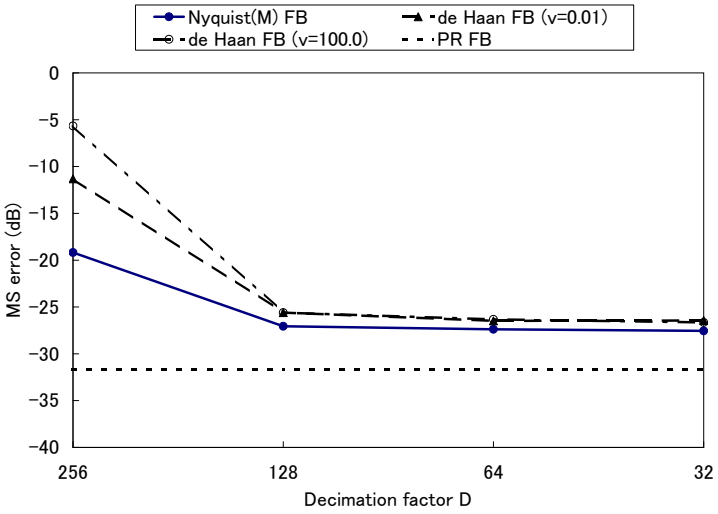


Figure 4.13: Mean square error (dB) for the decimation factor D, where M=512

Figure 4.13 that the MS errors of the Nyquist(M) filter banks with $D \leq 128$ are negligibly small. The total response error of de Haan's filter banks can be decreased by setting the small weighting factor v. However, even if v is set to 0.01, as indicated in Figure 4.13, its MS error is the highest of the three filter banks.

Amplification of a signal would make no difference for automatic speech recognition (ASR), given that ASR front-ends all apply gain control. Therefore we also consider the normalized MS error which is invariant to such a scaling can be expressed as

$$\epsilon_{\text{Norm MS}} = 10 \log_{10} \frac{\sum_{n=0}^{N-1} \left(x[n] - y[n] \sqrt{\frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1} y^2[n]}} \right)^2}{\sum_{n=0}^{N-1} x^2[n]}.$$

In this measure, the output signal is scaled so that its MS amplitude is equivalent to that of the input. Figure 4.14 shows the normalized MS errors of the PR, Nyquist(M), de Haan filter banks for the decimation factor. Just as in Figure 4.13, the de Haan filter bank provides the worst MS error, followed by the Nyquist(M) filter bank, then by the PR filter bank. It can also seen from

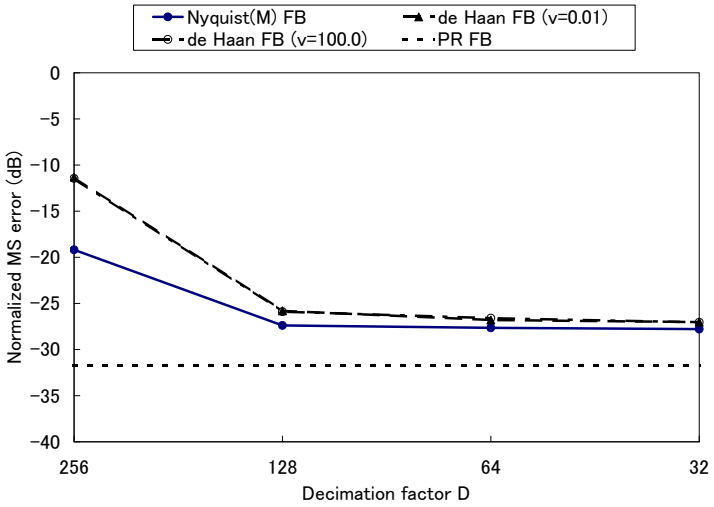


Figure 4.14: Normalized mean square error (dB) for the decimation factor D, where M=512

Figure 4.14 that the weighting factor v of de Haan's filter bank has no impact on the normalized MS error. This would seem to suggest that it is better to set the weighting factor v to a large value for ASR.

Chapter 5

Beamforming with Second Order Statistics

The delay-and-sum beamforming methods discussed in Chapter 3 do not take into consideration characteristics of a desired signal and noise which are available in many cases. By incorporating the characteristics of the signals into the beamforming algorithm, we might improve performance of the speech enhancement.

Several beamforming techniques for adaptively adjusting the sensor weights according to the characteristics of the observed signals have been developed. Most of the conventional adaptive beamforming algorithms use covariance matrices of noise and observed signals which are referred to as *second order statistics*(SOS) in this dissertation. The adaptive beamforming algorithms with SOS can generally suppress noise better than delay-and-sum beamforming. They have the sharper directivity pattern at low frequency.

The literature uses the term *adaptive* inconsistently. The term can mean any algorithm whose characteristics depend on the observed data or refer to only those algorithms that update the weights as each observation is received. This thesis refers to the former as *adaptive* algorithms while denoting the latter

as *dynamic adaptive* algorithms. In this thesis, unless the dynamic adaptive beamforming algorithms are described, the time index is omitted for the sake of simplicity.

The balance of this chapter is organized as follows. Section 5.1 reviews the most basic adaptive beamforming algorithm which minimizes the variance of beamformer's outputs subject to a distortionless constraint. In 5.2, an alternative implementation in generalized sidelobe canceller (GSC) configuration is presented. Section 5.3 describes a different GSC beamformer which takes into account a transfer function (TF) from a desired signal to microphones. That beamformer is referred to as the TF-GSC beamformer. A beamforming algorithm viewed as an extension of the TF-GSC beamformer is described in Section 5.4.

5.1 Minimum Variance Distortionless Response Beamformer

Interference signals can be suppressed by minimizing the variance of beamformer's output while maintaining the distortionless constraint in the direction of the desired signal.

Let us first define the steering vector $\mathbf{d}(m)$ at subband m as

$$\mathbf{d}(m) = \left[\exp\left\{-j\frac{2\pi m}{M}\frac{\Delta_1}{T}\right\}, \cdots, \exp\left\{-j\frac{2\pi m}{M}\frac{\Delta_i}{T}\right\}, \cdots, \exp\left\{-j\frac{2\pi m}{M}\frac{\Delta_{I-1}}{T}\right\}\right]^T. \tag{5.1}$$

We then determine the optimum weight vector that minimizes the variance of the beamformer's output

$$\mathbf{w}^H(m)\Sigma_{\mathbf{VV}}(m)\mathbf{w}(m), \tag{5.2}$$

subject to the distortionless constraint in the look direction

$$\mathbf{w}^H(m)\mathbf{d}(m) = 1, \tag{5.3}$$

where $\Sigma_{\mathbf{VV}}(m)$ is the covariance matrix of noise.

The well-known solution is called a minimum variance distortionless response (MVDR) beamformer [6, §13.3.1]. The weight vector of the MVDR beamformer at subband m can be expressed as

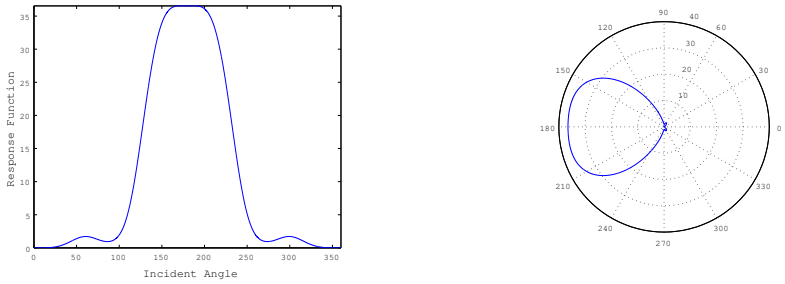
$$\mathbf{w}_{\mathrm{MVDR}}(m) = \frac{\mathbf{\Sigma}_{\mathbf{VV}}^{-1}(m)\mathbf{d}(m)}{\mathbf{d}^H(m){\Sigma_{\mathbf{VV}}}^{-1}(m)\mathbf{d}(m)}. \tag{5.4}$$

In order to avoid excessively large sidelobes in the beam pattern and the attendant non-robustness , small values are typically added to the main diagonal of $\mathbf{\Sigma}_{\mathbf{VV}}(m)$, which is called the *diagonal loading* [6, §13.3.7]. With the amount of diagonal loading σ_d, the weight vector of the MVDR beamformer can be written as

$$\mathbf{w}_{\mathrm{MVDR}}(m) = \frac{\left(\mathbf{\Sigma}_{\mathbf{VV}} + \sigma_d\mathbf{I}\right)^{-1}(m)\mathbf{d}(m)}{\mathbf{d}^H(m)\left(\mathbf{\Sigma}_{\mathbf{VV}} + \sigma_d\mathbf{I}\right)^{-1}(m)\mathbf{d}(m)}. \tag{5.5}$$

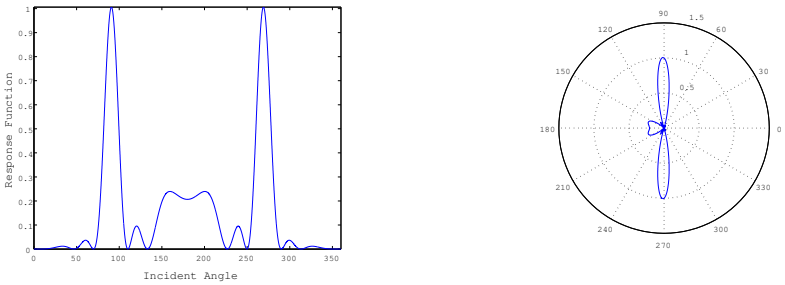
Figure 5.1, 5.2 and 5.3 show the beam patterns of the MVDR beamformer at frequencies 400 Hz, 800 Hz and 3200 Hz, respectively. In the figures, the MVDR beamformer is constructed for a linear array with seven equally-spaced sensors, that have an intersensor spacing of $d = 0.2\ m$. The diagonal loading of the MVDR beamformer is 0.001, and the look direction is set to $\pi/2$ $(= 90^o)$; the interference signal is assumed to come from angle 0. It is clear from Figure 5.1, 5.2 and 5.3 that the MVDR beamformer can maintain the unity gain for the look direction $\pi/2$. It is also clear that the MVDR beamformer can place a null on the arrival direction of the interference signal. It is also clear from Figure 5.2 that the gain for other than the look direction can exceed unity since there is no constraint except for the directions 0 in this case. This implies that performance of the noise suppression of the MVDR beamformer can be seriously degraded in the event of a steering error, which by definition occurs when the direction of arrival of the desired source is not precisely known.

The MVDR beamformer can suppress $I - 1$ directional interference signals, where I is the number of sensors. In the case that there are two interference signals at directions 0 and π $(= 180^o)$, the beam patterns at 400 Hz, 800 Hz and 3200 Hz are shown in Figure 5.4, 5.5 and 5.6, respectively. The plots in Figure 5.4, 5.5 and 5.6 are computed under the same conditions as in Figure 5.1, 5.2 and 5.3 except for the number of the interference signals. It is clear



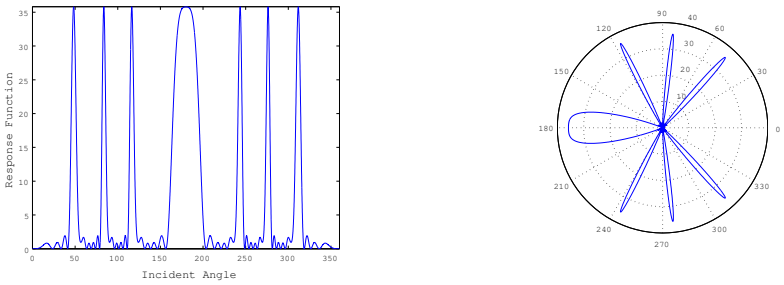
(a) (b)

Figure 5.1: Beam patterns of the MVDR beamformer with 7 sensors at $f = 400$ Hz and $d = 0.2\ m$ ($\lambda/d = 4.297$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 5.2: Beam patterns of the MVDR beamformer with 7 sensors at $f = 800$ Hz and $d = 0.2\ m$ ($\lambda/d = 2.148$): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 5.3: Beam patterns of the MVDR beamformer with 7 sensors at $f = 3200$ Hz and $d = 0.2\ m$ ($\lambda/d = 0.5371$): (a) in Cartesian coordinates and (b) in polar coordinates.

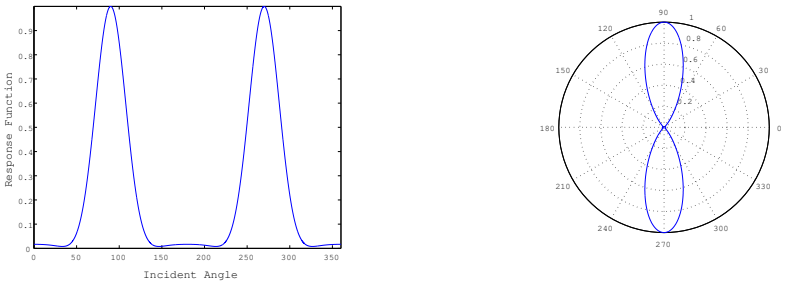
from Figure 5.4, 5.5 and 5.6 that the MVDR beamformer can suppress two interference signals by putting nulls on the directions 0 and π while maintaining the distortionless constraint for the look direction $\pi/2$.

The MVDR beamfomers would attempt to null out any interfering signal, but are prone to the signal cancellation problem [12] whenever there is an interfering signal that is correlated with the desired signal. In realistic environments, interference signals are highly correlated with a target signal since the target signal is reflected from hard surfaces such as walls and tables. Therefore, the adaptation of the weight vector is usually halted whenever the desired source is active.

5.1.1 Model of Noise Field

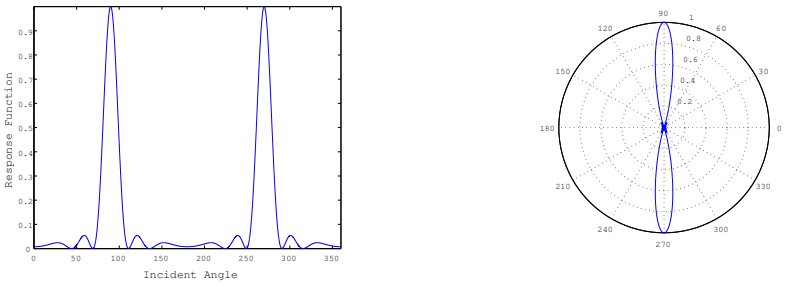
It is often better to use a noise filed model than to calculate the covariance matrix from actual noise observations directly. Two models that appear frequently in the literature are the incoherent and diffuse noise models [44].

In the case that a noise field is spatially uncorrelated (incoherent), the correlation of noise signals received at microphones at any given spatial location is zero. The covariance matrix in (5.2) then becomes an identity matrix, that



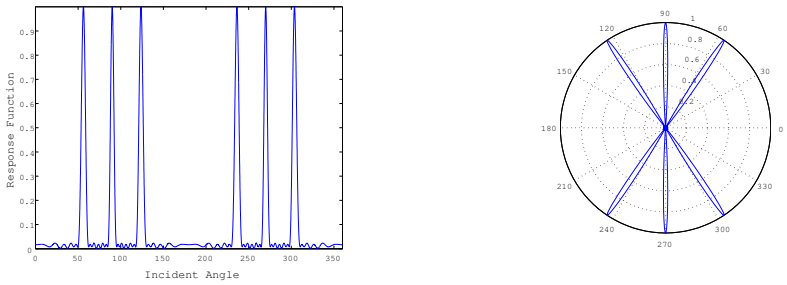
(a) (b)

Figure 5.4: Beam patterns of the MVDR beamformer with 7 sensors at $f = 400$ Hz and $d = 0.2\ m$ ($\lambda/d = 4.297$) in the case that there are two interference signals arriving from 0 and π (=180°): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 5.5: Beam patterns of the MVDR beamformer with 7 sensors at $f = 800$ Hz and $d = 0.2\ m$ ($\lambda/d = 2.148$) in the case that there are two interference signals arriving from 0 and π (=180°): (a) in Cartesian coordinates and (b) in polar coordinates.



(a) (b)

Figure 5.6: Beam patterns of the MVDR beamformer with 7 sensors at $f = 3200$ Hz and $d = 0.2\ m$ ($\lambda/d = 0.5371$) in the case that there are two interference signals arriving from 0 and π (=180°): (a) in Cartesian coordinates and (b) in polar coordinates.

is, $\mathbf{\Sigma_{VV}}(m) = \mathbf{I}$. Under these conditions, the MVDR solution for the sensor weights becomes equivalent to those of the delay-and-sum beamformer. The incoherent noise model is often appropriate when the distance between microphones is large and there are no coherent noise sources.

If spatially separated microphones receive equal energy and random phase noise signals from all directions simultaneously, it is called a spatially isotropic (diffuse) noise field. In the case of the diffuse noise field, each component of the matrix $\mathbf{\Sigma_{VV}}(m)$ can be expressed as

$$\Sigma_{v_i v_j}(m) = \mathrm{sinc}\left(2\pi \frac{m}{M} \frac{d_{ij}}{c}\right), \tag{5.6}$$

where d_{ij} is the distance between the i-th and j-th microphones [44].

5.2 Generalized Sidelobe Canceller

The MVDR beamformer could be implemented in a more computationally efficient way where the beamformer is constructed in *generalized sidelobe canceller* (GSC) configuration. McCowan et al. [36] reported that the GSC beamformer can improve performance of speech enhancement as well as speech recognition.

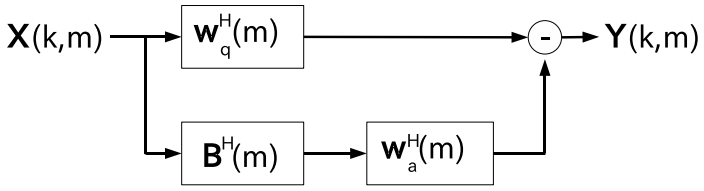


Figure 5.7: The Generalized Sidelobe Canceller (GSC) beamformer.

Figure 5.7 illustrates a beamformer in GSC configuration. The weights of the GSC beamformer at frequency bin m consists of three components, the quiescent vector $\mathbf{w}_{\mathrm{q}}(m)$, the blocking matrix $\mathbf{B}(m)$ and the active weight vector $\mathbf{w}_{\mathrm{a}}(m)$.

The output of the beamformer at frame k for a given subband can be expressed as

$$Y(k,m) = \left(\mathbf{w}_{\mathrm{q}}(m) - \mathbf{B}(m)\mathbf{w}_{\mathrm{a}}(m)\right)^{H} \mathbf{X}(k,m), \tag{5.7}$$

where $\mathbf{X}(k,m)$ is the input subband *snapshot vector*.

In keeping with the GSC formalism, $\mathbf{w}_{\mathrm{q}}(m)$ is chosen to give unity gain in the desired *look direction* [6, §13.3.7]; i.e., to satisfy the *distortionless constraint*. The blocking matrix $\mathbf{B}(m)$ is chosen to be orthogonal to $\mathbf{w}_{\mathrm{q}}(m)$ such that $\mathbf{B}^{H}(m)\,\mathbf{w}_{\mathrm{q}}(m) = \mathbf{0}$. The blocking matrix can be calculated with an orthogonalization technique such as the modified Gram-Schmidt method, QR decomposition or singular value decomposition (SVD) [45]. The orthogonality implies that the distortionless constraint will be satisfied for any choice of $\mathbf{w}_{\mathrm{a}}(m)$. In the case that the position of a sound source is static, the active weight vector is typically adjusted so that the variance of the beamformer's outputs is minimized as

$$\min_{\mathbf{w}_{\mathrm{a}}(m)} \mathcal{E}\{|Y(k,m)|^2\}. \tag{5.8}$$

By using (5.7), we can rewrite the variance as

$$\mathcal{E}\{|Y(k,m)|^2\} = \left(\mathbf{w}_{\mathrm{q}}(m) - \mathbf{B}(m)\mathbf{w}_{\mathrm{a}}(m)\right)^H \Sigma_{\mathbf{XX}}(m) \left(\mathbf{w}_{\mathrm{q}}(m) - \mathbf{B}(m)\mathbf{w}_{\mathrm{a}}(m)\right), \tag{5.9}$$

where $\Sigma_{\mathbf{XX}}(m)$ is the covariance matrix of the inputs. Without the diagonal loading, the solution of the active weight vector can be expressed as

$$\mathbf{w}_{\mathrm{a}}(m) = \left(\mathbf{B}^H(m)\Sigma_{\mathbf{XX}}(m)\mathbf{B}(m)\right)^{-1} \mathbf{B}^H(m)\Sigma_{\mathbf{XX}}(m)\mathbf{w}_{\mathrm{q}}(m). \tag{5.10}$$

The diagonal loading is often applied in order to penalize large active weight vectors, and thereby improve robustness by inhibiting the formation of excessively large sidelobes [6, §13.3.8]. Such a regularization term can be applied in the present instance by defining the modified optimization criterion

$$\mathcal{J}(Y(m);\alpha) = J(Y(m)) + \alpha\|\mathbf{w}_{\mathrm{a}}(m)\|^2, \tag{5.11}$$

where $\alpha > 0$ and $J(Y(m)) = \mathcal{E}\{|Y(k,m)|^2\}$.

Like the MVDR beamformer, the GSC beamformer can suppress interference signals effectively. However, it also leads to the signal cancellation in the case that there are signals which are correlated with the desired signal. In order to avoid the signal cancellation problem, the blocking matrix has to be carefully designed [18, 20].

5.3 Transfer Function GSC Beamformer

The quiescent vectors of the conventional GSC beamformer described in the previous section compensate each delay of arrival of the desired signal. Although the desired signal should be eliminated by the blocking matrix, the outputs of the blocking matrix usually contain components of the desired signal in practice due to steering errors or reverberation effects. The leakage of the desired signal to the adaptive noise canceller causes the signal cancellation problem. One of the solutions to that problem is to build a blocking matrix which is orthogonal to the acoustic transfer function (TF) from the desired source to the microphones.

As discussed in more detail below, such a blocking matrix can remove the desired signal from the output of the blocking matrix.

This section first formulates the description of the problem and then describes that the signal cancellation problem can be solved by using the ratios of the TFs instead of just compensating the delays. After that, a method for estimating the TF ratio is briefly explained.

5.3.1 Problem formulation

In realistic environments, sounds reflect from hard surfaces such as tables, walls, cellings and so on. Accordingly, sensors receive echoes of a desired signal. Consider that a source signal $S(k,m)$ is captured with I microphones in a reverberant enclosure. Let us define the time invariant TF from the source to the i-th microphone $A_i(m)$ and an additive noise signal at the i-th microphone $V_i(k,m)$.

Subband components of the received signals at frame k and frequency bin m can be expressed as

$$\mathbf{X}(k,m) = \mathbf{A}(m)S(k,m) + \mathbf{V}(k,m), \tag{5.12}$$

where

$$\begin{aligned} \mathbf{A}(m) &= [A_0(m), A_1(m), \cdots, A_{I-1}(m)]^T \\ \mathbf{V}(k,m) &= [V_0(k,m), V_1(k,m), \cdots, V_{I-1}(k,m)]\,.^T \end{aligned}$$

The final goal of the beamforming algorithms is to extract the source signal from the received noisy signal.

5.3.2 GSC Beamformer with TF Ratio

In the case that we know the actual acoustic TF exactly, we can extract the source signal. It is, however, difficult to estimate it in practice. Instead, Gannot et al. [24] estimated the ratios of the TFs.

That ratio can be written as

$$\tilde{\mathbf{H}}(m) = \frac{\mathbf{A}^T(m)}{A_0(m)} = [1, \frac{A_1(m)}{A_0(m)}, \cdots, \frac{A_{I-1}(m)}{A_0(m)}]^T. \tag{5.13}$$

In [24], the quiescent vector is then replaced with the TF ratio. The output of the fixed beamformer at the upper branch can be expressed as

$$Y_{\mathrm{up}}(k,m) = \frac{\tilde{\mathbf{H}}^H(m)}{||\tilde{\mathbf{H}}(m)||^2}\mathbf{X}(k,m) \tag{5.14}$$

$$= A_0(m)S(k,m) + \frac{\tilde{\mathbf{H}}^H(m)}{||\tilde{\mathbf{H}}(m)||^2}\mathbf{V}(k,m). \tag{5.15}$$

It is clear from (5.15) that the output at the upper branch has components of the desired signal distorted by the first TF.

The blocking matrix should prevent the components of the desired signal from entering into the following noise canceller in order to avoid the signal cancellation. It will be achieved if we design the blocking matrix so that $\mathbf{A}^H\mathbf{B} = \mathbf{0}$ is satisfied. In order to satisfy that orthogonality, Gannot et al. in [24] considered the following blocking matrix with the TF ratio

$$\mathbf{B}(m) = \begin{bmatrix} -\frac{A_1^*(m)}{A_0^*(m)} & -\frac{A_2^*(m)}{A_0^*(m)} & \cdots & -\frac{A_{I-1}^*(m)}{A_0^*(m)} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{5.16}$$

The output of the blocking matrix can be then expressed as

$$Y_{b,i}(k,m) = X_i(k,m) - \frac{A_i(m)}{A_0(m)}X_0(k,m) \;\; i = 1, \cdots, I-1. \tag{5.17}$$

This matrix can null out the acoustic paths from the desired signal to the microphones, which implies that any component of the desired signal is blocked. Accordingly, we can expect to obtain the ideal reference noise signal from the output of the blocking matrix.

Recall that the goal is to minimize the total output power of the beamformer under the constraint on the response at the desired direction. It is achieved by adjusting the active weight vectors in the same way as the conventional GSC beamformer. If the blocking matrix can remove the desired signal component perfectly, the adaptive canceller will suppress noise only without the signal

cancellation. Gannot et al. used the leaky least mean square (LMS) algorithm for this purpose [24].

5.3.3 Methods for Estimating the TF Ratios

In practice, the TF ratio is not known and must be estimated. From (5.17), we have

$$X_i(k,m) = H_i(k)X_0(k,m) + Y_{b,i}(k,m). \tag{5.18}$$

Gannot et al. assumed that the TF ratios were slowly changing in time compared to the time variations of the desired signal. Furthermore, they assumed that the statistics of the noise signal were also slowly changing compared with those of the desired signal. An analysis interval is then divided into frames such that the desired signal might be considered stationary during each frame. Let $\phi_{X_iX_j}(k,m)$ denote the cross power spectral density (PSD) between X_i and X_j, i-th and j-th noisy signal observations, during the k-th frame for all $k = 0, \cdots, K-1$. Further define $\phi_{Y_{b,i}X_0}(k,m)$ to be the cross-PSD between the m-th reference noise signal and X_0. Finally let $\hat{\phi}_{X_iX_j}(k,m)$ and $\hat{\phi}_{Y_{b,i}X_0}(k,m)$ represent the corresponding estimates. An unbiased estimate for $H_i(m)$ is obtained by applying the least squares fit to the following set of over-determined equations

$$\begin{bmatrix} \hat{\phi}_{X_iX_0}(0,m) \\ \hat{\phi}_{X_iX_0}(1,m) \\ \vdots \\ \hat{\phi}_{X_iX_0}(K-1,m) \end{bmatrix} = \begin{bmatrix} \hat{\phi}_{X_0X_0}(0,m) & 1 \\ \hat{\phi}_{X_0X_0}(1,m) & 1 \\ \vdots & \vdots \\ \hat{\phi}_{X_0X_0}(K-1,m) & 1 \end{bmatrix} \begin{bmatrix} H_i(m) \\ \phi_{Y_{b,i}X_0}(m) \end{bmatrix} + \begin{bmatrix} \epsilon_i(0,m) \\ \epsilon_i(1,m) \\ \vdots \\ \epsilon_i(K-1,m) \end{bmatrix} \quad i = 1, \cdots, I-1 \tag{5.19}$$

where K is the number of frames within the analysis interval. The solution to (5.19) is given by

$$H_i(k) = \frac{\mathcal{E}\{\hat{\phi}_{X_0X_0}(m)\hat{\phi}_{X_iX_0}(m)\} - \mathcal{E}\{\hat{\phi}_{X_0X_0}(m)\}\mathcal{E}\{\hat{\phi}_{X_iX_0}(m)\}}{\mathcal{E}\{\hat{\phi}^2_{X_0X_0}(m)\} - \mathcal{E}\{\hat{\phi}_{X_0X_0}(m)\}^2}, \qquad (5.20)$$

where $\mathcal{E}\{\}$ is the expectation operator. The ratios of the TF are estimated with the least squares method in periods when the speech signal is present.

5.4 Generalized Eigenvector Beamformer

Warsitz et al. [22] also proposed a new design method for the blocking matrix. They first find the weight vector which maximizes the signal-to-noise ratio (SNR) criterion. It was shown in [22] that the weight vector which provides the maximum SNR has the component of the TF from the source signal to each microphone. Thus, in the similar manner as described in Section 5.3, we can construct the blocking matrix which stops the desired signal from leaking into the noise canceler.

5.4.1 Maximum SNR Criterion

For the blocking matrix design, Warsitz et al. indirectly used information about the transfer functions from the signal source to the microphones. This information can be obtained from the weight vector of a generalized eigenvector beamformer [46]. The weight vector which provides the maximum SNR of the outputs at the frequency bin m can be expressed as

$$\mathbf{w}_{SNR}(m) = \underset{\mathbf{w}(m)}{\text{argmax}} \frac{\mathbf{w}^H(m)\mathbf{\Sigma}_{SS}(m)\mathbf{w}(m)}{\mathbf{w}^H(m)\mathbf{\Sigma}_{\mathbf{VV}}(m)\mathbf{w}(m)}, \qquad (5.21)$$

where

$$\begin{aligned} \mathbf{\Sigma}_{SS}(m) &= \mathcal{E}\{S^2(k,m)\mathbf{A}(m)\mathbf{A}^H(m)\}, \\ \mathbf{\Sigma}_{\mathbf{VV}}(m) &= \mathcal{E}\{\mathbf{V}(k,m)\mathbf{V}^H(k,m)\}, \end{aligned} \qquad (5.22)$$

and k indicates a frame index.

We here consider an analysis interval where the source and noise signals are assumed to be stationary. We can thus omit the frame index for the corresponding PSD. We further assume that the speech and noise are uncorrelated and that each of the signals has zero mean. This allows the PSD of the microphone signals to be split into two parts

$$\mathbf{\Sigma_{XX}}(m) = \mathbf{\Sigma}_{SS}(m) + \mathbf{\Sigma_{VV}}(m). \tag{5.23}$$

In that case, the solution of (5.21) is equivalent to the eigenvector belonging to the largest eigenvalue of a generalized eigenvalue problem (GEVP) [46]

$$\mathbf{\Sigma_{XX}}(m)\mathbf{w}(m) = \lambda(m)\mathbf{\Sigma_{VV}}(m)\mathbf{w}(m). \tag{5.24}$$

With the assumption that $\mathbf{\Sigma_{VV}}(m)$ is not singular, the GEVP can be transformed to the special eigenvalue problem (SEVP)

$$\mathbf{\Sigma}_{\mathbf{VV}}^{-1}(m)\mathbf{\Sigma_{XX}}(m)\mathbf{w}(m) = \lambda(m)\mathbf{w}(m). \tag{5.25}$$

The TFs from the desired source to the sensors are assumed to change slowly in time. Then for a large window length, we could approximate

$$\mathbf{\Sigma_{XX}}(m) \approx \phi_{S_0S_0}(k,m)\mathbf{A}(m)\mathbf{A}^H(m) + \mathbf{\Sigma_{VV}}(m). \tag{5.26}$$

In that case, the SEVP (5.25) can be reformulated as follows

$$\mathbf{\Sigma}_{\mathbf{VV}}^{-1}(m)\mathbf{A}(m)\mathbf{A}^H(m)\mathbf{w}(m) = \frac{\lambda(m)-1}{\phi_{S_0S_0}(k,m)}\mathbf{w}(m). \tag{5.27}$$

As the rank of the positive semidefinite matrix $\mathbf{\Sigma}_{\mathbf{VV}}^{-1}(m)\mathbf{A}(m)\mathbf{A}^H(m)$ is one, there is obviously only one eigenvector belonging to an eigenvalue greater than zero. This eigenvector can be expressed as

$$\mathbf{w}(m) = \zeta(m)\mathbf{\Sigma}_{\mathbf{VV}}^{-1}(m)\mathbf{A}(m), \tag{5.28}$$

where $\zeta(m)$ is an arbitrary complex constant. This can be easily verified by substituting (5.28) into (5.27). We can now see that the weight of the GEV beamformer contains the TF such as the TF-GSC beamformer.

5.4.2 Blocking Matrix Design for the GEV Beamformer

The optimal transfer function $\mathbf{w}_{SNR}(m)$ is used to construct a blocking matrix which projects any space into the orthogonal complement of $\mathbf{A}(m)$. That blocking matrix would produce noise reference signals which are orthogonal to a speech reference.

The speech reference can be written as

$$Y_{SNR}(k,m) = \mathbf{w}_{SNR}^H(m)\mathbf{X}(k,m). \tag{5.29}$$

We suppose that the the noise reference signal can be expressed with a projection vector $\mathbf{P}(m)$ as

$$\mathbf{U}(k,m) = \mathbf{X}(k,m) - \mathbf{P}(m)Y_{SNR}(k,m). \tag{5.30}$$

$\mathbf{P}(m)$ should be solved in order to satisfy the following orthogonality condition

$$\mathcal{E}\{\mathbf{U}(k,m)Y_{SNR}^*(k,m)\} = \mathbf{0}. \tag{5.31}$$

Upon substituting (5.30) into (5.31), we have

$$\mathbf{P}(k,m) = \frac{\mathcal{E}\{\mathbf{X}(k,m)Y_{SNR}^*(k,m)\}}{\mathcal{E}\{Y_{SNR}^2(k,m)\}}. \tag{5.32}$$

By substituting (5.29) into (5.32), we then obtain

$$\mathbf{P}(m) = \frac{\Sigma_{\mathbf{XX}}(m)\mathbf{w}_{SNR}(m)}{\mathbf{w}_{SNR}^H(m)\Sigma_{\mathbf{XX}}(m)\mathbf{w}_{SNR}(m)}. \tag{5.33}$$

By using (5.24), (5.33) can be further modified as

$$\mathbf{P}(m) = \frac{\Sigma_{\mathbf{VV}}(m)\mathbf{w}_{SNR}(m)}{\mathbf{w}_{SNR}^H(m)\Sigma_{\mathbf{VV}}(m)\mathbf{w}_{SNR}(m)}. \tag{5.34}$$

Upon taking $\mathbf{X}(k,m)$ out of (5.30), we can extract the blocking matrix

$$\mathbf{B}^H(m) = \mathbf{I} - \mathbf{P}(m)\mathbf{w}_{SNR}^H(m), \tag{5.35}$$

where $\mathbf{I}$ is the identity matrix. By substituting (5.34) into (5.35) and replacing $\mathbf{w}_{SNR}$ in the resultant equation based on (5.28), we have

$$\mathbf{B}^H(m) = \mathbf{I} - \frac{\mathbf{A}(m)\mathbf{A}^H(m)\Sigma_{\mathbf{VV}}{}^{-1}(m)}{\mathbf{A}^H(m)\Sigma_{\mathbf{VV}}{}^{-1}(m)\mathbf{A}(m)}. \tag{5.36}$$

Finally, we have the output of the blocking matrix

$$\begin{aligned} \mathbf{U}(k,m) &= \mathbf{B}^H(m)\mathbf{X}(k,m) && (5.37) \\ &= \left(\mathbf{I} - \frac{\mathbf{A}(m)\mathbf{A}^H(m){\Sigma_{\mathbf{VV}}}^{-1}(m)}{\mathbf{A}^H(m){\Sigma_{\mathbf{VV}}}^{-1}(m)\mathbf{A}(m)}\right)\mathbf{V}(m). && (5.38) \end{aligned}$$

We can easily see that the output of the blocking matrix does not contain any component of the desired signal. Hence, the signal cancellation will be avoided whenever all the assumptions mentioned above are satisfied. From those formulae, the blocking matrix in the TF-GSC beamformer can be viewed as a special case of that of the GEV beamformer.

Chapter 6

Independent Component Analysis (ICA)

Independent component analysis (ICA) is a method for finding underlying factors or components from multi-variate statistical data. ICA looks for components that are statically independent.

One of the popular applications based on the ICA theory is blind source separation (BSS). Consider a situation where multiple speakers are talking simultaneously and mixed signals are captured with a microphone array. BSS algorithms attempt to separate each source signal from the mixture of speech captured with the multiple microphones. These algorithms do not use prior knowledge such as the geometry of the microphones and source positions although the number of the sources is usually assumed to be known. Blind source separation is achieved by multiplying an un-mixing matrix with an input vector of a multi-channel signal. The un-mixing matrix is constructed so that components of the output vector obtained by that multiplication are statically independent. Each element of the output vector would correspond to each source signal.

However, those blind assumptions lead to the well-known permutation and

scaling ambiguity problems [1]. The components of the output vector might be permutated and criteria for measuring the statistical independence are unable to determine the scale uniquely. The scaling ambiguity problem can be avoided by calculating the pseudo-inverse of the un-mixing matrix [27]. The permutation problem could be alleviated by interchanging the components based on the property of continuity of speech spectral envelopes [47]. The geometry of the array is sometimes used for solving the permutation problem [28]. However, in that case, it is not blind anymore since the prior knowledge is used.

Another problem of the BSS techniques is that they rely on numerical optimization algorithms which only provide a local solution. Consequently, the performance of the BSS algorithms always depends on the initial values. The un-mixing matrix obtained with those techniques may fail to extract the target signal in some situations. One of the solutions to the problem is to repeatedly initialize the un-mixing matrix with the weights of beamformers which use information about the geometry of the array [48]. After all, the beamforming techniques are used for solving the problems caused by the blind assumption. It is worth mentioning that the relationship between beamforming and BSS techniques has been thoroughly analyzed in [2].

In contrast to those BSS approaches, the current author directly applies the ICA method to the GSC beamformer, which will be described in Chapter 7.

The rest of this chapter is organized as follows. Basic concepts of ICA is described in Section 6.1. Section 6.1 also shows that the distribution of clean speech is in fact non-Gaussian. It is then illustrated that the distribution gets closer to Gaussian in the case that speech signals are corrupted with noise or reverberation. These facts lead to the conclusion that noise can be removed by a beamformer by making the distribution of its outputs as super-Gaussian as possible. Section 6.2 introduces several super Gaussian probability functions (pdfs). Section 6.3 describes criteria for measuring degree of *super-Gaussianity*, which is essential for developing the new beamforming algorithms. An actual speech distribution modeled with one of the super-Gaussian pdfs is investigated in Section 6.4.

6.1 ICA and its Application to Speech

The entire field of ICA is founded on the assumption that all signals of real interest are *not* Gaussian-distributed [1]. Briefly, the reasoning is grounded on two points:

1. The *central limit theorem* states that the pdf of the sum of independent random variables (r.v.s) will approach Gaussian in the limit as more and more components are added, *regardless* of the pdfs of the individual components. This implies that the sum of several r.v.s will be closer to Gaussian than any of the individual components. Thus, if the original independent components comprising the sum are sought, one must look for components with pdfs that are the *least* Gaussian.

2. The *entropy* for a continuous complex-valued r.v. Y is defined as

$$H(Y) \triangleq -\int p_Y(v) \log p_Y(v) dv = -\mathcal{E}\left\{\log p_Y(v)\right\}, \tag{6.1}$$

   where $p_Y(.)$ is the pdf of Y. Entropy is the basic measure of information in *information theory* [49]. It is well known that a Gaussian r.v. has the highest entropy of all r.v.s with a given variance [49, Thm. 7.4.1], which also holds for complex Gaussian r.v.s [50, Thm. 2]. Hence, a Gaussian r.v. is, in some sense, the least *predictable* of all r.v.s. Information-bearing signals, on the other hand, are redundant and thus contain structure that makes them more predictable than Gaussian r.v.s. Hence, if an information-bearing signal is sought, one must once more look for a signal that is *not* Gaussian.

The fact that the pdf of speech is super-Gaussian has often been reported in the literature [4, 51, 52]. Noise, on the other hand, is more nearly Gaussian-distributed. In fact, the pdf of the sum of several super-Gaussian r.v.s. becomes closer to Gaussian. Thus, a mixture consisting of a desired signal and several interfering signals can be expected to be nearly Gaussian-distributed.

The Gaussian and four super-Gaussian univariate pdfs are plotted in Fig. 6.1. From the figure, it is clear that the Laplace, K_0, Γ, and generalized Gaussian

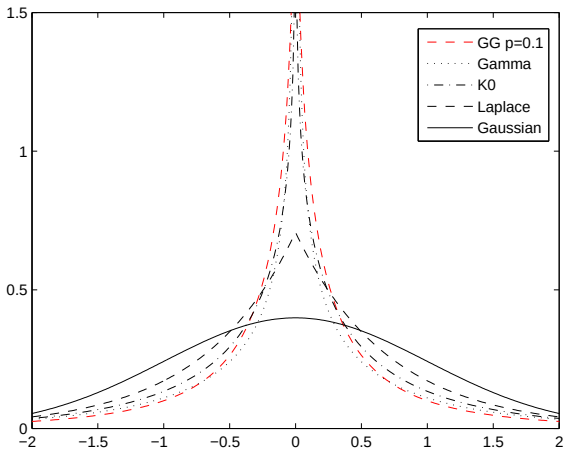


Figure 6.1: Gaussian and super-Gaussian pdfs.

(GG) densities exhibit the "spikey" and "heavy-tailed" characteristics that are typical of super-Gaussian pdfs. This implies that they have a sharp concentration of probability mass at the mean, relatively little probability mass as compared with the Gaussian at intermediate values of the argument, and a relatively large amount of probability mass in the tail; i.e., far from the mean.

Fig. 6.2 shows a histogram of the real parts of subband samples of speech at $f_s = 800$ Hz. To generate the histograms, the author used 43.9 minutes of clean speech recorded with a close-talking microphone (CTM) from the development set of the Speech Separation Challenge, Part 2 (SSC2) [3]. The Gaussian, Laplace, K_0, Γ, and GG pdfs are also shown in Fig. 6.2. For this plot, parameters of the GG pdf was estimated from training data. It is clear from Fig. 6.2 that the distribution of clean speech is not Gaussian but super-Gaussian. Fig. 6.2 also suggests that the GG pdf can be suitable for modeling subband samples of speech.

Fig. 6.3 shows a histogram of magnitude in the subband domain[1]. We can see from Fig. 6.3 that the GG pdf can model the distribution of magnitude in the subband domain very well.

Fig. 6.4 shows histograms of real parts of subband components calculated from clean speech and noise-corrupted speech. It is clear from this figure that the pdf of the noise-corrupted speech has less probability mass around the center spike, and less probability mass in the tail than the clean speech, but more probability mass in the intermediate regions. This indicates that the pdf of the noise-corrupted signal, which is in fact the sum of the speech and noise signals, is closer to Gaussian than that of clean speech.

Fig. 6.5 shows histograms of clean speech and reverberant speech in the subband domain. In order to produce the reverberant speech, a clean speech signal was convolved with an impulse response measured in a room; see Section 9.1 for the configuration of the room. We can observe from Fig. 6.5 that the pdf of reverberated speech is also closer to Gaussian than the original clean speech.

We also present a histogram of magnitude of noise corrupted speech in Fig. 6.6 and that of reverberant speech in Fig. 6.7. We can again see from Fig. 6.6 and Fig. 6.7 that the pdfs of corrupted speech have less probability mass around the mean and less probability mass in the tail, but once more more probability mass in the intermediate regions. Interestingly, Fig. 6.7 shows that the peak of the histogram of the speech is shifted from zero to the right by the reverberation effect.

These facts would indeed support the hypothesis that seeking an enhanced speech signal that is maximally non-Gaussian is an effective way to suppress the distorting effects of noise and reverberation.

[1]The pdfs in Fig. 6.3 are generally defined over the interval (-∞, +∞). Precisely speaking, the double-sided pdfs should be modified in order to model magnitude whose value is always positive. This is easily done by multiplying both sides by a factor of two and redefining the interval as [0, +∞). Such modifications, however, are not necessary in our algorithm in that the factor of two in the normalization is constant in the log-likelihood domain and has no effect on the gradient algorithm.

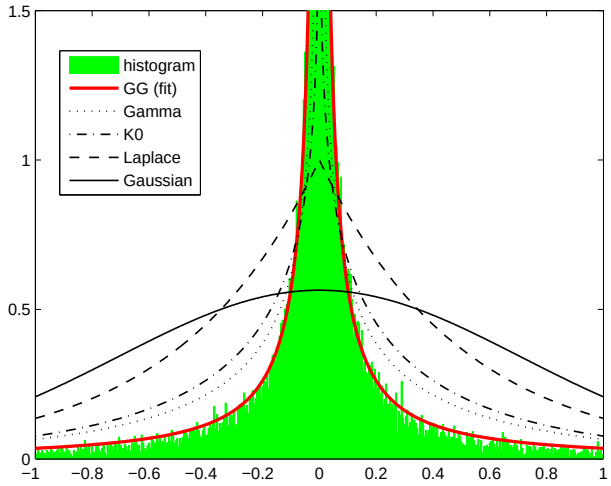


Figure 6.2: Histogram of real parts of subband components and pdfs.

6.2 Super-Gaussian pdf

In this section, two classes of representative super-Gaussian pdfs are described. One is the pdfs derived from the *Meijer G-function*. The other is the generalized Gaussian (GG) pdf.

The author starts with the explanation of the pdfs derived from Meijer G-function and then describe the GG pdf.

6.2.1 Super-Gaussian pdf derived from the Meijer G-function

As explained in Brehm and Stammler [53], it is useful to assume that the Laplace, K_0, and Γ pdfs belong to the class of SIRPs for two principal reasons. Firstly, this implies that multivariate pdfs of all orders can be readily derived from the univariate pdf using the theory of *Meijer G-functions* based solely on the knowledge of the covariance matrix of the random vectors. Secondly, such variates can be extended to the case of complex r.v.s, which is essential for our

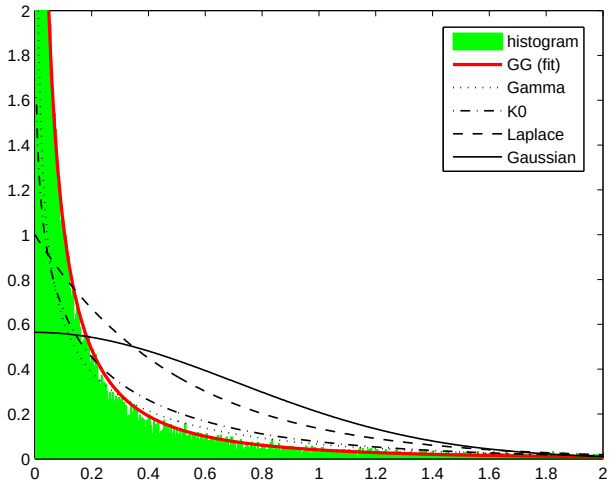


Figure 6.3: Histogram of magnitude in the subband domain and pdfs.

current development.

For complex Laplace r.v.s $Y_i \in \mathbf{C}$, the univariate pdf can be expressed as

$$p_{\text{Lap}}(Y_i) = \frac{4}{\sqrt{\pi}\sigma_Y^2} K_0\left(\frac{2\sqrt{2}|Y_i|}{\sigma_Y}\right) \tag{6.2}$$

where $K_0(z)$ is an irregular modified Bessel function and $\sigma_Y^2 = \mathcal{E}\{|Y_i|^2\}$. For $\mathbf{Y} \in \mathbf{C}^2$, the bivariate Laplace pdf is given by

$$p_{\text{Lap}}(\mathbf{Y}) = \frac{16}{\pi^{3/2}|\Sigma_{\mathbf{Y}}|\sqrt{s}} K_1\left(4\sqrt{s}\right) \tag{6.3}$$

where $\Sigma_{\mathbf{Y}} = \mathcal{E}\{\mathbf{Y}\mathbf{Y}^H\}$ and

$$s = \mathbf{Y}^H \Sigma_{\mathbf{Y}}^{-1} \mathbf{Y}.$$

Similarly, we can write the univarite K_0 pdf for complex r.v.s $Y_i \in \mathbf{C}$ as

$$p_{K_0}(Y_i) = \frac{1}{\sqrt{\pi}\sigma_Y|Y_i|} \exp\left(-2\,|Y_i|/\sigma_Y\right). \tag{6.4}$$

The bivariate K_0 pdf for $\mathbf{Y} \in \mathbf{C}^2$ can be expressed as

$$p_{K_0}(\mathbf{Y}) = \frac{\sqrt{2}+4\sqrt{s}}{2\pi^{3/2}\,|\Sigma_{\mathbf{Y}}|\,s^{3/2}} \exp\left(-2\sqrt{2\,s}\right). \tag{6.5}$$

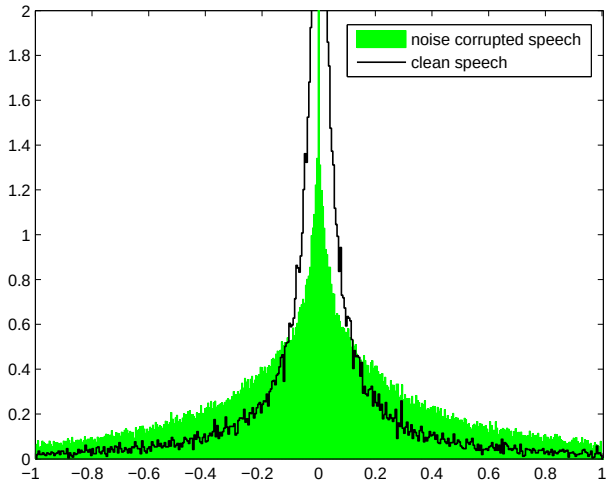


Figure 6.4: Histograms of clean speech and noise corrupted speech in the subband domain.

Derivations of (6.2)–(6.5) are provided in Appendix A. For the Γ pdf, the complex univariate and bivariate pdfs *cannot* be expressed in closed form in terms of elementary or even special functions. As explained in Appendix A, however, it is possible to derive Taylor series expansions that enable the required variates to be calculated to arbitrary accuracy. These developments are also described in [4].

Table 6.1 shows the average log-likelihood of subband samples of speech recorded with the close-talking microphone as calculated with the Gaussian and three super-Gaussian pdfs, namely, the Laplace, K_0, and Γ pdfs averaged over $K = 1000$ time instants and $M = 512$ subbands. It is clear from these log-likelihood values that the complex subband samples of speech are in fact better modeled by the super-Gaussian pdfs considered here than the Gaussian. Hence, the abstract arguments on which the field of ICA are founded correspond well to the actual characteristics of speech. It is worth noting that the use of *spherically-invariant random processes* (SIRPs) in the context of BSS is

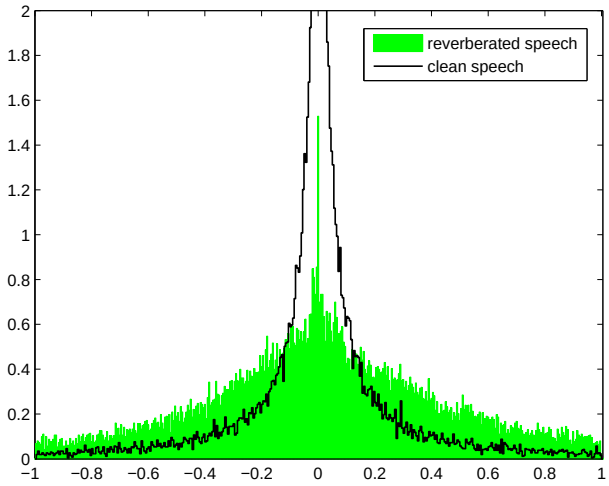


Figure 6.5: Histograms of clean speech and reverberant speech in the subband domain.

discussed by Buchner *et al.* [26].

6.2.2 Generalized Gaussian pdf

Due to its definition as a contour integral, finding maximum likelihood estimates for the parameters of the Meijer G-function must necessarily devolve to a grid search over the relevant parameter space [53]. Instead, it might be better to use a simple super-Gaussian pdf whose parameters can easily be adjusted so as to match the actual subband samples. The generalized Gaussian (GG) pdf is well-known and finds frequent application in the BSS and ICA fields. Moreover, it subsumes the Gaussian and Laplace pdfs as special cases. The GG pdf with zero mean for a real-valued r.v. y can be expressed as

$$p_{\mathrm{GG}}(y) = \frac{1}{2\Gamma(1+1/p)A(p,\hat{\sigma})} \exp\left[-\left|\frac{y}{A(p,\hat{\sigma})}\right|^{p}\right], \tag{6.6}$$

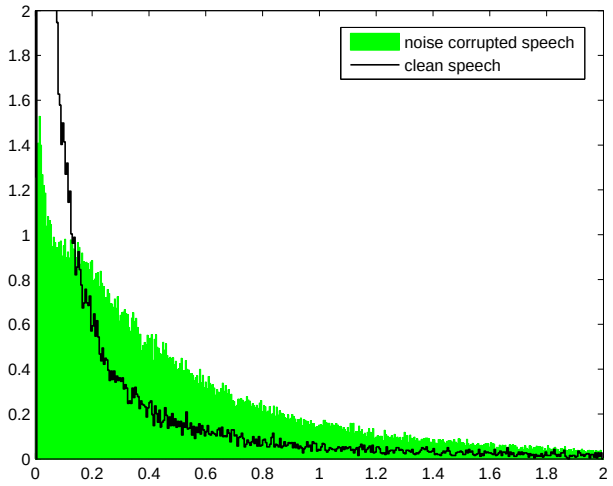


Figure 6.6: Histograms of the magnitude of clean speech and noise corrupted speech in the subband domain.

where p is the *shape parameter*, $\hat{\sigma}$ is the *scale parameter* which controls how fast the tail of the pdf decays, and

$$A(p,\hat{\sigma}) \quad = \quad \hat{\sigma}\left[\frac{\Gamma(1/p)}{\Gamma(3/p)}\right]^{1/2}. \tag{6.7}$$

In (6.7), $\Gamma(.)$ is the gamma function. Note that the GG with $p=1$ corresponds to the Laplace pdf, and that setting $p=2$ yields the Gaussian pdf, whereas in the case of $p \to +\infty$ the GG pdf converges to a uniform distribution.

Fig. 6.8 shows the GG pdf with the same scale parameter $\hat{\sigma}^2 = 1$ and different shape parameters, $p = 0.5,\ 1,\ 2,\ 4$. From the figure, it is clear that a smaller shape parameter yields a spikier pdf with a heavier tail.

The differential entropy of the GG pdf for the real-valued r.v. y is obtained with the help of *Mathematica* [54] as

$$\begin{aligned} H_{\mathrm{GG}}(y) \quad &= \quad -\int_{-\infty}^{+\infty} p_{\mathrm{gg}}(\xi)\log p_{\mathrm{gg}}(\xi)d\xi \\ &= \quad \frac{1}{p} + \log\left[2\Gamma(1+1/p)A(p,\hat{\sigma})\right]. \end{aligned} \tag{6.8}$$

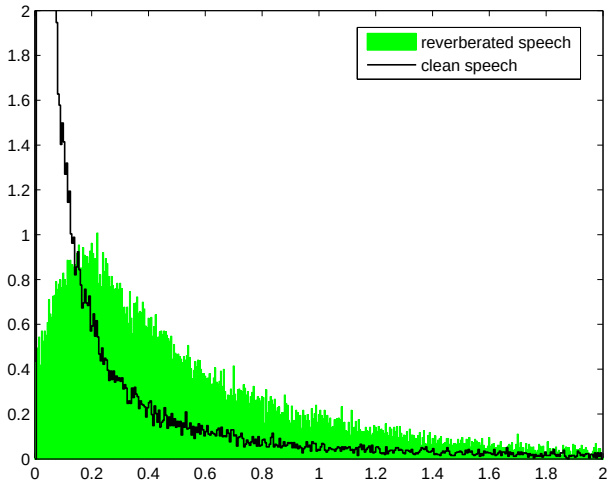


Figure 6.7: Histograms of magnitude of clean speech and reverberated speech in the subband domain.

Maximum likelihood (ML) estimates of the shape and scale parameters can be determined from a set of training data, as described in Section 6.4.

6.3 Criteria for Super-Gaussianity

There are two popular criteria for measuring non-Gaussianity, namely, kurtosis and negentropy. In addition to these criteria, mutual information is also frequently used in the field of ICA [1].

Mutual information can be viewed as a direct measure of representing how r.v.s are independent of each other. However, we can calculate the mutual information measure only if multiple signal sources are active. In other words, we cannot apply the mutual information criterion to a situation where a single source is active.

Kurtosis and negentropy criteria indicate how the distribution of r.v.s is far from the Gaussian distribution. These criteria can be used for speech enhance-

Table 6.1: Average log-likelihoods of subband speech samples for various pdfs.

pdf	$\frac{1}{KM}\sum_{k=0}^{K-1}\sum_{m=0}^{M-1}\log p(X(k,m);\text{pdf})$
Γ	–0.779
K_0	–1.11
Laplace	–2.48
Gaussian	–9.93

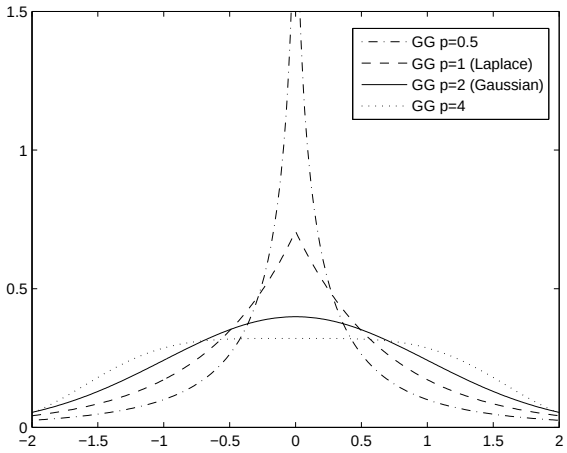


Figure 6.8: The generalized Gaussian (GG) pdfs.

ment of a single speaker.

In this section, mutual information under Gaussian and super-Gaussian assumptions is first reviewed. The definitions of kurtosis and negentropy are then described.

6.3.1 ICA by Minimization of Mutual Information

Mutual information indicates how useful a given random variable is for explaining one or more other random variables. By definition, mutual information of

two random variables, Y_1 and Y_2, is given by

$$I(\mathrm{Y}) \triangleq \mathcal{E}\left\{\log \frac{p(\mathrm{Y})}{p(Y_1)p(Y_2)}\right\} \tag{6.9}$$

where $\mathcal{E}\{\}$ denotes ensemble expectation and $\mathrm{Y} = [Y_1, Y_2]^T$. Two random variables are statistically independent whenever the mutual information between them is zero.

Mutual information is always non-negative, and zero if and only if the variables are statistically independent. Mutual information takes into account the whole dependence structure of the r.v.s, and includes the covariance. On the other hand, the SOS-based methods ignore it.

Most of the work consider a situation where two sound sources are active and the extension to the case of the multiple sources is rather easy. Accordingly, this thesis explains the case of two r.v.s which correspond to the active sound sources. From (6.9), we have the mutual information of two variables

$$\begin{aligned} I(Y_1, Y_2) &= \mathcal{E}\left\{\log \frac{p(Y_1, Y_2)}{p(Y_1)p(Y_2)}\right\} \\ &= \mathcal{E}\{\log p(Y_1, Y_2)\} - \mathcal{E}\{\log p(Y_1)\} - \mathcal{E}\{\log p(Y_2)\}. \end{aligned} \tag{6.10}$$

MMI under a Gaussian Assumption

The basic concept of ICA is that every signal of interest is not Gaussian but super-Gaussian. Nevertheless, the Gaussian assumption is useful in many situations and employed in various applications.

Here, mutual information of two r.v.s under the Gaussian assumption is derived. The resultant formula indicates that it is a simple function of their cross-correlation coefficient.

For Gaussian r.v.s, we have

$$p(y_n) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{y_n^2}{2\sigma_n^2}\right).$$

Hence, we can solve the latter two expectations in (6.10) as

$$\begin{aligned}\mathcal{E}\{\log p(Y_n)\} &= \mathcal{E}\left\{-\frac{1}{2}\log 2\pi\sigma_n^2 - \frac{1}{2}\frac{Y_n^2}{\sigma_n^2}\right\} \\ &= -\frac{1}{2}\log 2\pi\sigma_n^2 - \frac{1}{2}\int_{-\infty}^{\infty}\frac{y_n^2}{\sigma_n^2}\,p(y_n)\,dy_n. \end{aligned} \tag{6.11}$$

For jointly Gaussian r.v.s,

$$p(Y_1, Y_2) = \frac{1}{\sqrt{|2\pi\Sigma|}}\exp\left[-\frac{1}{2}\mathbf{Y}^T\Sigma^{-1}\mathbf{Y}\right]$$

where $\mathbf{Y} = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^T$ and the *covariance matrix* of $\mathbf{Y}$ is given by [55, §2.3]

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{12} \\ \sigma_1\sigma_2\rho_{12} & \sigma_2^2 \end{bmatrix} \tag{6.12}$$

with

$$\rho_{12} = \frac{\epsilon_{12}}{\sigma_1\,\sigma_2}$$

where $\epsilon_{12} = \mathcal{E}\{Y_1\,Y_2^*\}$. Hence, the first expectation in (6.9) can be rewritten as

$$\begin{aligned}\mathcal{E}\{\log p(Y_1, Y_2)\} &= \mathcal{E}\left\{-\frac{1}{2}\log|2\pi\Sigma| - \frac{1}{2}\mathbf{Y}^T\Sigma^{-1}\mathbf{Y}\right\} \\ &= -\frac{1}{2}\log|2\pi\Sigma| - \frac{1}{2}\int_{\mathbf{Y}}\mathbf{Y}^T\Sigma^{-1}\mathbf{Y}\,p(\mathbf{Y})\,d\mathbf{Y}. \end{aligned} \tag{6.13}$$

Due to the *whitening* [56, §2.3] provided by the term Σ^{-1}, the integral in (6.13) decouples into two integrals of the form of the integral in (6.11). Hence, when (6.11) and (6.13) are substituted back into (6.9), the integral terms cancel out, and what remains is

$$\begin{aligned} I(Y_1, Y_2) = &-\tfrac{1}{2}\log\left[4\pi^2\sigma_1^2\sigma_2^2(1-\rho_{12}^2)\right] \\ &+\tfrac{1}{2}\log 2\pi\sigma_1^2 + \tfrac{1}{2}\log 2\pi\sigma_2^2 \end{aligned}$$

or, upon cancelling common terms,

$$I(Y_1, Y_2) = -\tfrac{1}{2}\log\left(1-\rho_{12}^2\right).$$

For the complex r.v.s which will be considered in Section 7.1.1, it is straightforward to show that

$$I(Y_1, Y_2) = -\log\left(1-|\rho_{12}|^2\right). \tag{6.14}$$

From (6.14) it is clear that minimizing the mutual information between two zero-mean Gaussian r.v.s is equivalent to minimizing the squared magnitude of their cross correlation coefficient and that

$$I(Y_1, Y_2) = 0 \leftrightarrow |\rho_{12}| = 0.$$

MMI under a super-Gaussian Assumption

The mutual information for the Laplace, K_0 and Γ pdfs can no longer be expressed in closed form as in (6.14) for the super-Gaussian pdfs. We can, however, replace the exact mutual information with the *empirical mutual information*

$$I(Y_1, Y_2) \approx \frac{1}{K} \sum_{k=0}^{K-1} \log p(\mathbf{Y}(k)) - \sum_{n=1}^{2} \left[\frac{1}{K} \sum_{k=0}^{K-1} \log p(Y_n(k)) \right]. \tag{6.15}$$

The details for calculating the probabilities under each super-Gaussian assumption are described in Appendix A.

6.3.2 ICA by Maximization of Kurtosis

The *excess kurtosis* or simply kurtosis of a complex-valued r.v. Y with zero mean is defined as

$$\text{kurt}(Y) \triangleq \mathcal{E}\{|Y|^4\} - 3(\mathcal{E}\{|Y|^2\})^2. \tag{6.16}$$

The Gaussian pdf has zero kurtosis, pdfs with positive kurtosis are super-Gaussian, those with negative kurtosis are *sub-Gaussian*.

As shown in (6.16), the kurtosis measure considers not only the variance but also the fourth moment of HOS. Notice that the Gaussian pdf can be specified with the mean and variance up to SOS only. Accordingly, it could be viewed that the Gaussian assumption would ignore HOS.

Of the three super-Gaussian pdfs in Fig. 6.1, the Γ pdf has the highest kurtosis, followed by the K_0, then by the Laplace pdf. As is clear from Fig. 6.1, as the kurtosis increases, the pdf becomes more spikey and heavy-tailed. Note that the kurtosis of the GG pdf can be controlled by adjusting the shape parameter p, as explained in Section 6.4.

In practice, kurtosis can be calculated by simply averaging samples according to

$$\text{kurt}(Y) = \frac{1}{K}\sum_{k=0}^{K-1}|Y(k)|^4 - 3\left(\frac{1}{K}\sum_{k=0}^{K-1}|Y(k)|^2\right)^2. \tag{6.17}$$

The kurtosis criterion does not require any explicit assumption as to the exact form of the pdf. Due to its simplicity, it is widely used as a measure of non-Gaussianity. The value calculated for kurtosis, however, can be strongly influenced by a few samples with a low observation probability. Hyvärinen and Oja [1] noted that negentropy is generally more robust in the presence of outliers than kurtosis.

6.3.3 ICA by Maximization of Negentropy

The negentropy of a complex-valued r.v. Y is defined as

$$J(Y) \triangleq H(Y_{\text{gauss}}) - H(Y) \tag{6.18}$$

where Y_{gauss} is a Gaussian variable which has the same variance σ_Y^2 as Y. The entropy of Y_{gauss} can be expressed as

$$H(Y_{\text{gauss}}) = \log\left|\sigma_Y^2\right| + (1 + \log\pi)\,. \tag{6.19}$$

Note that negentropy is non-negative, and zero if and only if Y has a Gaussian distribution.

Computing entropy of the super-Gaussian variables $H(Y)$ normally requires for a specific pdf assumption. It is, thus, important to find the pdf which closely matches to the distributions of actual speech signals.

Negentropy also takes HOS into account by using super-Gaussian pdfs. Super-Gaussian pdfs are not represented with SOS only and their higher moments are normally specified with parameters other than the mean and variance. For example, as described in Appendix B, the GG pdf can be specified with the shape and scale parameters which are not SOS. On the other hand, each moment of the Gaussian pdf is determined with the variance and the order of the moment only.

6.4 Speech Modeling with the GG pdf

We can model the clean speech signals by estimating the parameters of the GG pdf. In this section, the estimation method is first described and the properties of the GG pdf estimated with actual speech samples are then investigated.

6.4.1 Estimating Scale and Shape Parameters

Among several methods for estimating the shape parameter p of the GG pdf [57][58], the moment and ML methods are arguably the most straightforward. In this work, we used the moment method in order to initialize the parameters of the GG pdf and then updated them with the ML estimate [58]. The shape parameters are estimated from training samples offline and are then held fixed during beamforming. The shape parameters are estimated independently for each subband, as the optimal pdf is frequency-dependent.

For a set $\mathcal{Y} = \{y_0, y_1, \ldots, y_{K-1}\}$ of K real-valued training samples, the log-likelihood function under the GG pdf can be expressed as

$$\begin{aligned} l(\mathcal{Y}; \hat{\sigma}, p) = &-K \log \{2\Gamma(1+1/p) A(p, \hat{\sigma})\} \\ &- \frac{1}{A(p, \hat{\sigma})^p} \sum_{k=0}^{K-1} |y(k)|^p. \end{aligned} \tag{6.20}$$

In this work, we considered three kinds of training sample $y(k)$, namely, the magnitude as well as the real and imaginary parts of the subband samples of speech.

The parameters $\hat{\sigma}$ and p can be obtained by solving the following equations

$$\frac{\partial l(\mathcal{Y}; \hat{\sigma}, p)}{\partial \hat{\sigma}} = -\frac{K}{\hat{\sigma}} + \frac{p}{\hat{\sigma}^{p+1}} \left[\frac{\Gamma(1/p)}{\Gamma(3/p)}\right]^{-\frac{p}{2}} \sum_{k=0}^{K-1} |y(k)|^p = 0, \tag{6.21}$$

$$\begin{aligned} \frac{\partial l(\mathcal{Y}; \hat{\sigma}, p)}{\partial p} =& K a(p) - \sum_{k=0}^{K-1} \left(\frac{|y(k)|}{A(p, \hat{\sigma})}\right)^p \\ &\times \left[\log\left\{\frac{|y(k)|}{A(p, \hat{\sigma})}\right\} + b(p)\right] = 0, \end{aligned} \tag{6.22}$$

where

$$\begin{aligned} a(p) &= (p^{-2}/2)[2\Psi(1+1/p) + \Psi(1/p) - 3\Psi(3/p)], \\ b(p) &= (p^{-1}/2)[\Psi(1/p) - 3\Psi(3/p)], \end{aligned}$$

and $\Psi(.)$ is the digamma function. By solving (6.21) for $\hat{\sigma}$, we obtain

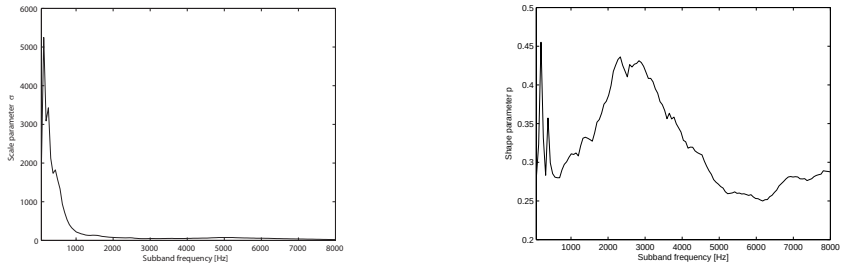
$$\hat{\sigma} = \left[\frac{\Gamma(3/p)}{\Gamma(1/p)}\right]^{1/2} \left(\frac{p}{K}\sum_{k=0}^{K-1}|y(k)|^p\right)^{1/p}. \tag{6.23}$$

Due to the presence of the special functions, it is impossible to solve (6.22) for p explicitly. Varanasi [59] showed, however, that (6.22) has a unique root given the scale parameter. Hence, the gradient descent algorithm [60] can be used to find the unique solution which maximizes the likelihood. The solution of (6.22) can be also obtained with the secant algorithm [54, 59]. The estimation of the parameters is repeated until the log-likelihood function (6.20) converges.

6.4.2 Analysis of the Estimated Parameters

Subband components of speech can be precisely modeled by estimating the parameters of the GG pdf from training samples. From the trained parameters, insight can be gained into statistical properties of human speech. Fig. 6.9 shows the scale parameter $\hat{\sigma}_{|Y|}$ and the shape parameter p calculated from the magnitude of subband components plotted as functions of frequency, where the number of the subbands is 256. The training samples used for estimating the GG pdf here were also taken from clean speech data recorded with a close-talking microphone.

It is clear from Fig. 6.9 that the scale parameter $\hat{\sigma}_{|Y|}$ becomes smaller at higher frequencies. The scale parameter $\hat{\sigma}_{|Y|}$ is related to the variance of $|Y|$, although not identical to it in the case that the ML method is used in its estimation. Fig. 6.9 indicates that the magnitude at lower frequencies varies more than that at higher frequencies. Moreover, the GG pdfs trained with actual speech data are super-Gaussian with $p < 2$ in all subbands; they are in fact *super-Laplacian* with $p < 1$ in all subbands. As mentioned previously, kurtosis



(a) (b)

Figure 6.9: The parameters of the GG pdf for frequency; (a) scale parameter $\hat{\sigma}_{|Y|}$ and (b) shape parameter p, where the sampling frequency is 16 kHz.

is a measure of super-Gaussianity of a pdf. It is therefore of interest to examine the behavior of kurtosis of the GG pdf. As demonstrated in Appendix B, the latter can be expressed as

$$\mathrm{kurt}(Y_{gg}) = \hat{\sigma}^4 \left\{ \frac{\Gamma(1/p)\,\Gamma(5/p)}{\Gamma^2(3/p)} - 3 \right\}. \tag{6.24}$$

Fig. 6.10 shows a plot of kurtosis values as a function of frequency. In Fig. 6.10, a solid line indicates the kurtosis of the GG pdf calculated with (6.24) and a broken line presents the empirical kurtosis computed with (6.17). It is clear from Fig. 6.10 that the GG pdf can also model the kurtosis of speech, which would make the negentropy criterion more robust for outliers than the empirical kurtosis. It is also clear from Fig. 6.10 that kurtosis becomes smaller at higher frequencies, which indicates that the pdf of lower frequency components are more super-Gaussian than those of higher frequency components.

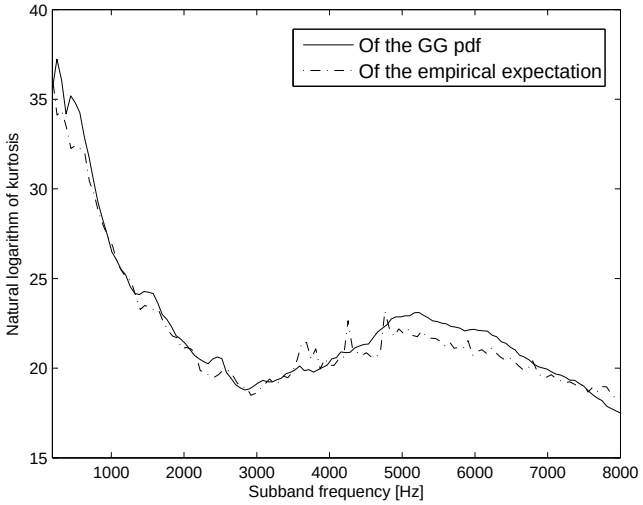


Figure 6.10: Kurtosis vs. frequency, where the sampling rate is 16 kHz.

Chapter 7

Beamforming with Higher-Order Statistics

This chapter presents novel GSC beamforming algorithms which take into consideration higher order statistics (HOS). As described in Chapter 6, HOS can be obtained by measuring the degree of super-Gaussianity such as the kurtosis and negentropy. The new HOS-based beamforming methods adjust the active weight vectors of the GSC so as to make the distribution of the output signals as much as super-Gaussian.

Chapter 5 described the GSC beamformers with second order statistics (SOS) which is typically associated with the covariance matrix. The conventional GSC beamforming algorithms are based on utilization of SOS in quasi-linear discrete-time systems. These algorithms effectively place a null on any source of interference. Although they are useful in many applications and their theory is well-developed, their performance is limited due to the assumptions of the Gaussianity and linearity.

Not only the interference signal but also the desired signal can be removed by those SOS-based beamforming algorithms in the case that there are signals correlated with the target signal. That problem is referred to as the *signal*

cancellation [12].

The new algorithms can suppress noise and reverberation without the signal cancellation problems encountered in the conventional beamforming algorithms. This will be demonstrated through simulations and experiments.

The balance of this chapter is organized as follows. Section 7.1 describes new GSC beamforming algorithms which minimize mutual information of the beamformer's outputs [4]. These algorithms have been fundamentally developed for a speech separation task where multiple coherent sound sources are active. As described in Section 6.3.1, mutual information measures a distance between two distributions. Therefore, outputs of more than two beamformers are required in order to adjust the active weight vectors based on the minimum mutual information criterion. Such beamforming algorithms are not suitable for the situation where there is only one active source. Section 7.2 presents another new beamforming algorithm which estimates the active weight vectors so as to maximize negentropy of beamformer's outputs subject to the distortionless constraint [32]. Furthermore, the GSC beamformer which provides maximum kurtosis [33] is depicted in Section 7.3. The beamforming techniques described in Section 7.2 and Section 7.3 have been developed for a situation where a single active sound source exists (single-speaker scenario).

7.1 Minimum Mutual Information Beamformer

Assuming there are two GSC beamformers aimed at different sources, as shown in Fig. 7.1, the output of the nth beamformer at frame k for subband m can be expressed as,

$$Y_n(k,m) = \left(\mathbf{w}_{\mathrm{q},n}(m) - \mathbf{B}_n(m)\mathbf{w}_{\mathrm{a},n}(m)\right)^H \mathbf{X}(k,m), \tag{7.1}$$

where $\mathbf{w}_{\mathrm{q},n}(m)$ is the quiescent weight vector for the nth source, $\mathbf{B}_n(m)$ is the blocking matrix, $\mathbf{w}_{\mathrm{a},n}(m)$ is the active weight vector, and $\mathbf{X}(k,m)$ is the input subband snapshot vector, which is common to both sources.

In the same manner as described in Section 5.2, the blocking matrix $\mathbf{B}_n(m)$ is

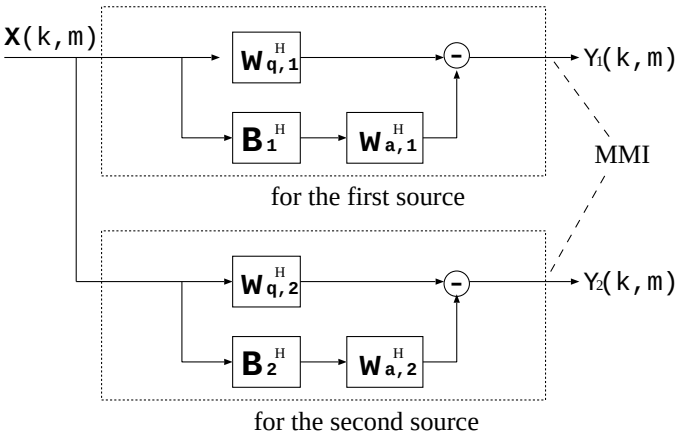


Figure 7.1: Schematic of generalized sidelobe cancelling (GSC) beamformers for each active source.

chosen to be orthogonal to $\mathbf{w}_{\mathrm{q},n}(m)$. While the active weight vector $\mathbf{w}_{\mathrm{a},n}(m)$ has been chosen to maximize the total power of beamformer's outputs or the SNR in the conventional beamformers, we have developed an optimization procedure to find the $\mathbf{w}_{\mathrm{a},n}(m)$ that minimizes the mutual information $I(Y_1(m), Y_2(m))$; based on the development of Section 6.3.1.

7.1.1 MMI Beamforming with the Gaussian Assumption

As shown in Section 6.3.1, minimizing mutual information under the assumption of the Gaussian r.v.s is equivalent to minimizing the magnitude $|\rho_{12}|$ of the cross-correlation coefficient. Let the variance of $Y_n(k,m)$ be denoted by $\sigma_n^2 = \mathcal{E}\{|Y_n(k,m)|^2\}$ for the sake of simplicity, where $\mathcal{E}\{|Y_n(k,m)|^2\}$ can be calculated in the same way as (5.9).

The cross-correlation coefficient ρ_{12} between $Y_1(k,m)$ and $Y_2(k,m)$ can be expressed as [55, §2.3]

$$\rho_{12} = \frac{\epsilon_{12}}{\sigma_1\,\sigma_2} \tag{7.2}$$

where

$$\begin{aligned}\epsilon_{12} &= \mathcal{E}\{Y_1(k,m)\, Y_2^*(k,m)\} \\ &= \left(\mathbf{w}_{\mathrm{q},1}(m) - \mathbf{B}_1(m)\mathbf{w}_{\mathrm{a},1}(m)\right)^H \Sigma_{\mathbf{XX}}(m)\left(\mathbf{w}_{\mathrm{q},2}(m) - \mathbf{B}_2(m)\mathbf{w}_{\mathrm{a},2}(m)\right).\end{aligned} \tag{7.3}$$

Hence,

$$|\rho_{12}|^2 = \frac{|\epsilon_{12}|^2}{\sigma_1^2\,\sigma_2^2}. \tag{7.4}$$

Minimizing the mutual information criterion yields a weight vector $\mathbf{w}_{\mathrm{a},n}(m)$ capable of canceling interference that leaks through the sidelobes without the signal cancellation problems encountered in conventional beamforming.

Parameter Optimization

In the absence of a closed-form solution for those $\mathbf{w}_{\mathrm{a},n}(m)$ minimizing $|\rho_{12}|^2$, we must use a numerical optimization algorithm. Such an optimization algorithm typically requires gradient information. We here omit the frequency index m for convenience sake.

Let us apply the chain rule [39, §A.7.4] to (7.4), and write

$$\begin{aligned}\frac{\partial|\rho_{12}|^2}{\partial\mathbf{w}_{\mathrm{a},1}^*} &= \frac{1}{\sigma_1^4\,\sigma_2^4}\left(\frac{\partial\epsilon_{12}}{\partial\mathbf{w}_{\mathrm{a},1}^*}\epsilon_{12}^*\sigma_1^2\sigma_2^2 - \frac{\partial\sigma_1^2}{\partial\mathbf{w}_{\mathrm{a},1}^*}|\epsilon_{12}|^2\sigma_2^2\right) \\ &= \frac{1}{\sigma_1^4\,\sigma_2^4}\Big[-\mathbf{B}_1^H\Sigma_{\mathbf{XX}}(\mathbf{w}_{\mathrm{q},2} - \mathbf{B}_2\mathbf{w}_{\mathrm{a},2})\epsilon_{12}^*\sigma_1^2\sigma_2^2 \\ &\qquad +\mathbf{B}_1^H\Sigma_{\mathbf{XX}}(\mathbf{w}_{\mathrm{q},1} - \mathbf{B}_1\mathbf{w}_{\mathrm{a},1})|\epsilon_{12}|^2\sigma_2^2\Big].\end{aligned}$$

The last equation can be simplified to

$$\begin{aligned}\frac{\partial|\rho_{12}|^2}{\partial\mathbf{w}_{\mathrm{a},1}^*} = \frac{1}{\sigma_1^4\,\sigma_2^4}\mathbf{B}_1^H\Sigma_{\mathbf{XX}}\big[&|\epsilon_{12}|^2\sigma_2^2(\mathbf{w}_{\mathrm{q},1} - \mathbf{B}_1\mathbf{w}_{\mathrm{a},1}) \\ &- \epsilon_{12}^*\sigma_1^2\sigma_2^2(\mathbf{w}_{\mathrm{q},2} - \mathbf{B}_2\mathbf{w}_{\mathrm{a},2})\big].\end{aligned} \tag{7.5}$$

From symmetry it then follows

$$\begin{aligned}\frac{\partial|\rho_{12}|^2}{\partial\mathbf{w}_{\mathrm{a},2}^*} = \frac{1}{\sigma_1^4\,\sigma_2^4}\mathbf{B}_2^H\Sigma_{\mathbf{XX}}\big[&|\epsilon_{12}|^2\sigma_1^2(\mathbf{w}_{\mathrm{q},2} - \mathbf{B}_2\mathbf{w}_{\mathrm{a},2}) \\ &-\epsilon_{12}\sigma_1^2\sigma_2^2(\mathbf{w}_{\mathrm{q},1} - \mathbf{B}_1\mathbf{w}_{\mathrm{a},1})\big].\end{aligned} \tag{7.6}$$

Equations (7.5) and (7.6) are sufficient to implement a numerical optimization algorithm based, for example, on the method of *conjugate gradients* which is described in Appendix C.

Regularization

The regularization term can be applied in the present instance by defining the modified optimization criterion

$$\mathcal{I}(Y_1(m), Y_2(m); \alpha) = I(Y_1(m), Y_2(m)) + \alpha\|\mathbf{w}_{\mathrm{a},1}(m)\|^2 + \alpha\|\mathbf{w}_{\mathrm{a},2}(m)\|^2 \quad (7.7)$$

for some real $\alpha > 0$. Taking the partial derivative on both sides of (7.7) yields

$$\frac{\partial\mathcal{I}(Y_1(m), Y_2(m); \alpha)}{\partial\mathbf{w}^*_{\mathrm{a},n}(m)} = \frac{1}{2\,(1 - |\rho_{12}|^2)} \cdot \frac{\partial|\rho_{12}|^2}{\partial\mathbf{w}^*_{\mathrm{a},n}(m)} + \alpha\mathbf{w}_{\mathrm{a},n}(m). \quad (7.8)$$

7.1.2 MMI Beamforming with the Super-Gaussian Assumption

The mutual information can no longer be expressed in closed form as in (7.7) for the super-Gaussian pdfs. We can, however, replace the exact mutual information with the *empirical mutual information*

$$I(Y_1(m), Y_2(m)) \approx \frac{1}{K}\sum_{k=0}^{K-1} \log p(\mathbf{Y}(k,m)) - \sum_{i=1}^{2}\left[\frac{1}{K}\sum_{k=0}^{K-1} \log p(Y_i(k,m))\right]. \quad (7.9)$$

The relations necessary to evaluate the partial derivative of (7.9) with respect to $\mathbf{w}_{\mathrm{a},i}(m)$ for the super-Gaussian pdfs considered here are given in Appendix A. Notice that calculating mutual information under the super-Gaussian assumptions requires HOS, which is not the case in the Gaussian assumption.

7.1.3 Geometric Source Separation (GSS)

Parra and Alvino [61] proposed a *geometric source separation* (GSS) algorithm with many similarities to the MMI beamforming algorithm under the Gaussian assumption. Their work was based on two beamformers with geometric

constraints that made them functionally equivalent to GSC beamformers. The principal difference between GSS and the algorithm proposed here is that GSS seeks to minimize $|\epsilon_{12}|^2$ instead of $|\rho_{12}|^2$. Although the difference between minimizing $|\epsilon_{12}|^2$ instead of $|\rho_{12}|^2$ may seem very slight, it can in fact lead to radically different behavior.

To achieve the desired optimum, both criteria will seek to place deep nulls on the unwanted source; this characteristic is associated with $|\epsilon_{12}|^2$, which also comprises the *numerator* of $|\rho_{12}|^2$. Such null steering is also observed in conventional adaptive beamformers [39, §6.3].

The difference between the two optimization criteria is due to the presence of the terms σ_i^2 in the denomimnator of $|\rho_{12}|^2$, which indicate that, in addition to nulling out the unwanted signal, an improvement of the objective function is also possible by *increasing* the strength of the desired signal. For acoustic beamforming in realistic environments, there are typically strong reflections from hard surfaces such as tables and walls. A conventional beamformer would attempt to null out strong reflections of an interfering signal, but strong reflections of the desired signal can lead to signal cancellation. The GSS algorithm would attempt to null out those reflections from the unwanted signal. But in addition to nulling out reflections from the unwanted signal, the MMI beamforming algorithm would attempt to *strengthen* those reflections from the desired source; assuming statistically independent sources, strengthening a reflection from the desired source would have little or no effect on the numerator of $|\rho_{12}|^2$, but would increase the denominator, thereby leading to an overall reduction of the optimization criterion.

Of course, any reflected signal would be delayed with respect to the direct path signal. Such a delay would, however, manifest itself as a phase shift in the subband domain, and could thus be removed through a suitable choice of $\mathbf{w}_\mathrm{a}$ as far as the number of delayed samples is less than the length of the analysis filter. Hence, the MMI beamformer offers the possibility of steering both nulls *and* sidelobes; the former towards the undesired signal and its reflections, the latter towards reflections of the desired signal.

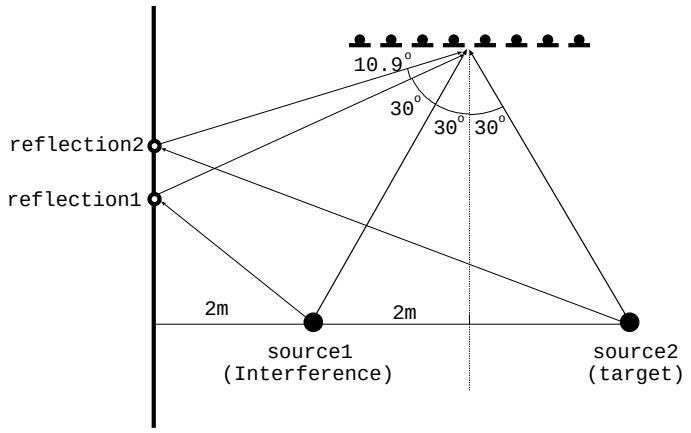


Figure 7.2: Configuration of sources, sensors, and reflective surface for a simulation comparing GSS and MMI beamformer.

In order to verify that the MMI beamforming algorithm forms sidelobes directed towards the reflections of a desired signal, we conducted experiments with a simulated acoustic environment. As shown in Fig. 7.2, we considered a simple configuration where there are two sound sources, a reflective surface, and an eight-channel linear microphone array that captures both the direct and reflected waves from each source. Actual speech data were used as sound sources in this simulation, which was based on the *image method* [62].

Fig. 7.3 shows beam patterns at $f_s = 1500$ Hz and $f_s = 3000$ Hz obtained with the MMI beamformer and the GSS algorithm. In order to make the techniques directly comparable, the implementation of the GSS algorithm used for the simulation, as well as the ASR experiments described in Chapter 9, was based on two GSCs, each aimed at one target. Both MMI beamformer and GSS algorithm formed the beam patterns so that the signal from Source 2 in Fig. 7.2 was enhanced while the other from Source 1 was suppressed. It is clear that both algorithms have unity gain in the look direction, and place deep nulls on the direct path of the unwanted source. The suppression of Reflection 1, the undesired interference, by the MMI beamformer is equivalent to or better than that provided by the GSS algorithm for both frequencies. Moreover, the

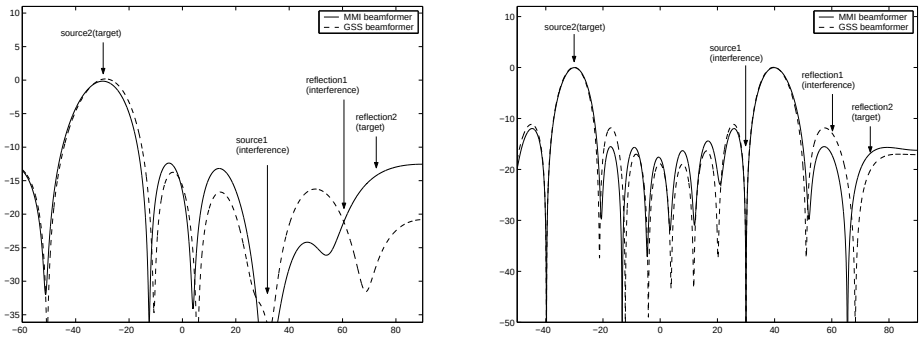


Figure 7.3: Beam patterns produced by the MMI beamformer and GSS algorithm using a spherical wave assumption for (a) $f_{\mathrm{s}} = 1500$ Hz and (b) $f_{\mathrm{s}} = 3000$ Hz.

enhancement of Reflection 2, the desired signal, by the MMI beamformer is stronger than that of the GSS algorithm.

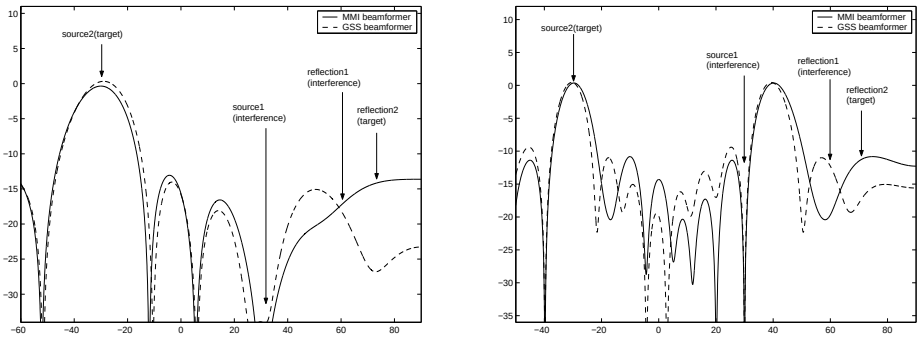
Given that a beam pattern shows the sensitivity of an array to plane waves, but the beam patterns in Fig. 7.3 were made with near-field sources and reflections, we also ran a second set of simulations in which all sources and reflections were assumed to produce plane waves. The results of this second simulation are shown in Fig. 7.4. Once more, it is apparent that the MMI beamformer emphasizes Reflection 2 from the desired source.

If a regularization term is added as before, we obtain the GSS optimization criterion

$$\mathcal{I}'(Y_1(m), Y_2(m); \alpha) = |\epsilon_{12}|^2 + \alpha \|\mathbf{w}_{\mathrm{a},1}(m)\|^2 + \alpha \|\mathbf{w}_{\mathrm{a},2}(m)\|^2. \tag{7.10}$$

Then taking partial derivatives of (7.10) gives

$$\begin{aligned} \frac{\mathcal{I}'(Y_1(m), Y_2(m); \alpha)}{\partial \mathbf{w}^*_{\mathrm{a},1}(m)} &= -\mathbf{B}_1^H(m)\Sigma_{\mathbf{XX}}(m)(\mathbf{w}_{\mathrm{q},2}(m) - \mathbf{B}_2(m)\mathbf{w}_{\mathrm{a},2}(m))\epsilon^*_{12} \\ &\quad + \alpha \mathbf{w}_{\mathrm{a},1}(m) \end{aligned} \tag{7.11}$$



(a) (b)

Figure 7.4: Beam patterns produced by the MMI beamformer and GSS algorithm using a plane wave assumption for (a) $f_{\mathrm{s}} = 1500$ Hz and (b) $f_{\mathrm{s}} = 3000$ Hz.

$$\frac{\mathcal{I}'(Y_1(m), Y_2(m); \alpha)}{\partial \mathbf{w}_{\mathrm{a},2}^*(m)} = -\mathbf{B}_2^H(m)\Sigma_{\mathbf{XX}}(m)(\mathbf{w}_{\mathrm{q},1}(m) - \mathbf{B}_1(m)\mathbf{w}_{\mathrm{a},1}(m))\,\epsilon_{12} + \alpha \mathbf{w}_{\mathrm{a},2}(m). \tag{7.12}$$

Although at first blush it may seem that a closed-form solution for $\mathbf{w}_{\mathrm{a},1}(m)$ and $\mathbf{w}_{\mathrm{a},2}(m)$ could be derived, the presence of ϵ_{12}^* and ϵ_{12} in (7.11) and (7.12) respectively actually makes this impossible. Hence, a numerical optimization algorithm is needed, as before.

7.2 Maximum Negentropy Beamformer

In some cases, we might not have multiple active sources. The MMI beamforming algorithm cannot be applied to a situation where there is only one coherent signal since the calculation of mutual information requires more than two kinds of r.v.s. In contrast, negentropy and kurtosis can be computed with r.v.s generated from a single source. The GSC beamformer with the maximum negentropy criterion would be expected to have the same advantage as MMI beamforming.

This section describes a method of estimating the active weight vectors of

the GSC beamformer so as to obtain the maximum negentropy of the outputs. Section 7.2.1 describes the formulae necessary for estimating the active weight vectors under the assumption of the pdf derived from the Meijer G-function. Section 7.2.2 depicts the beamforming algorithm in the case that the generalized Gaussian is used as the speech model. Section 7.2.1 discusses differences between the MN and SOS-based beamformers.

7.2.1 Estimation of Active Weights under the Γ pdf

For the empirical studies reported here, the Γ pdf was used, as it achieved a higher likelihood than the other two named pdfs, namely, Laplace, and K_0

The differential entropy for the Γ pdf cannot be expressed in closed form. Hence, in order to use the Γ pdf, it is necessary to replace the exact differential entropy with the *empirical entropy*

$$H(Y) = -\mathcal{E}\left\{\log p_Y(v)\right\} \approx -\frac{1}{K}\sum_{k=0}^{K-1} \log p_Y(Y(k,m)), \tag{7.13}$$

where $Y(k,m)$ is an observed subband sample.

Substituting (7.13) and (6.19) into (6.18), we can express the negentropy as

$$J(Y(m)) = \log\left|\sigma_Y^2\right| + 2\left(1+\log 2\pi\right) + \frac{1}{K}\sum_{k=0}^{K-1} \log p_Y(Y(k,m)), \tag{7.14}$$

where K is the number of frames used for weight vector adaptation and $\sigma_Y^2 = \mathcal{E}\{Y(k,m)Y^*(k,m)\}$.

By applying the regularization term, we have the modified object function

$$\mathcal{J}(Y(m);\alpha) = J(Y(m)) - \alpha\|\mathbf{w}_{\mathrm{a}}(m)\|^2 \tag{7.15}$$

for some real $\alpha > 0$. We maximize the objective function (7.15).

In the absence of a closed-form solution for the $\mathbf{w}_{\mathrm{a}}(m)$ maximizing the negentropy (7.15), we must resort to the *conjugate gradients* method [63, §1.6]. By substituting (7.14) into (7.15) and taking the partial derivative on both sides,

we obtain the gradient function

$$\begin{aligned}\frac{\partial \mathcal{J}(Y(m);\alpha)}{\partial \mathbf{w_a}^*(m)} &= \frac{\partial J(Y(m);\alpha)}{\partial \mathbf{w_a}^*(m)} - \alpha \mathbf{w_a}(m) \\ &= \frac{1}{|\sigma_Y^2|}\frac{\partial |\sigma_Y^2|}{\partial \mathbf{w_a}^*(m)} + \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{p_Y(Y(k,m))}\frac{\partial p_Y(Y(k,m))}{\partial \mathbf{w_a}^*(m)} - \alpha \mathbf{w_a}(m),\end{aligned} \tag{7.16}$$

where

$$\frac{\partial |\sigma_Y^2|}{\partial \mathbf{w_a}^*(m)} = \frac{1}{K}\sum_{k=0}^{K-1}\left\{-\mathbf{B}^H(m)\mathbf{X}(k,m)Y^*(k,m)\right\}. \tag{7.17}$$

Equations (7.16) and (7.17) are sufficient to implement a numerical optimization algorithm, whereby the negentropy $J(Y(m))$ can be maximized. The details of the conjugate gradient algorithm are described in Appendix C.

7.2.2 Parameter Optimization under the Generalized Gaussian Assumption

Parameter optimization with magnitude of outputs

Unlike the pdfs that can be expressed as Meijer G-functions, the GG pdf cannot be readily extended from the univariate to the multi-variate. Hence, we use the magnitude of the beamformer's output as the r.v. for calculating the entropy. By substituting (6.8) and (6.19) into (6.18) and applying the regularization term, we arrive at the following expression for negentropy

$$\mathcal{J}(Y(m);\alpha) = \log\left|\sigma_Y^2\right| + 2\left(1+\log 2\pi\right) - H_{\mathrm{GG}}(|Y|) - \alpha\|\mathbf{w_a}(m)\|^2, \tag{7.18}$$

where $\alpha > 0$.

In order to apply the conjugate gradients algorithm, we must once more derive an expression for the gradient. By taking the partial derivative on both sides of (7.18) while holding the shape parameter fixed, we obtain

$$\frac{\partial \mathcal{J}(Y(m);\alpha)}{\partial \mathbf{w_a}^*(m)} = \frac{1}{\sigma_Y^2}\frac{\partial \sigma_Y^2}{\partial \mathbf{w_a}^*(m)} - \frac{\partial H_{\mathrm{GG}}(|Y|)}{\partial \mathbf{w_a}^*(m)} - \alpha \mathbf{w_a}(m), \tag{7.19}$$

where

$$\frac{\partial H_{\mathrm{GG}}(|Y|)}{\partial \mathbf{w_a}^*(m)} = \frac{1}{\hat{\sigma}_{|Y|}}\frac{\partial \hat{\sigma}_{|Y|}}{\partial \mathbf{w_a}^*(m)}. \tag{7.20}$$

Taking the derivative on both sides of (6.23), we find

$$\frac{\partial \hat{\sigma}_{|Y|}}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} = \frac{p}{K}\left[\frac{\Gamma(3/p)}{\Gamma(1/p)}\right]^{\frac{1}{2}} \times \left[\frac{p}{K}\sum_{k=0}^{K-1}|Y(k,m)|^{p}\right]^{\frac{1}{p}-1} \times \left[\sum_{k=0}^{K-1}|Y(k,m)|^{p-1}\frac{\partial |Y(k,m)|}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)}\right], \tag{7.21}$$

where the gradient of the magnitude at each frame is

$$\frac{\partial |Y(k,m)|}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} = -\frac{1}{2|Y(k,m)|}\mathbf{B}^{H}(m)\mathbf{X}(k,m)Y^{*}(k,m). \tag{7.22}$$

Based on (7.19) through (7.22), a numerical algorithm for optimizing the active weight vector can be implemented; See Appendix C for the details.

Parameter optimization with each component of a complex value

It is conceivable that the entropy of the GG pdf for the complex valued r.v. could be approximated by assuming that the real and imaginary parts are independent. Under such an assumption, the differential entropy of the GG pdf can be expressed as

$$H(Y) \approx H_{\mathrm{r}}(Y_r) + H_{\mathrm{i}}(Y_i), \tag{7.23}$$

where Y_{r} is the real part of Y and Y_{i} is its imaginary part. Notice that the shape parameters for the real and imaginary parts must be trained individually.

Then, upon substituting (6.19) and (7.23) into (6.18) and adding the regularization term, we obtain the objective function

$$\begin{aligned}\mathcal{J}(Y(m);\alpha) =& \log\left|\sigma_Y^2\right| + 2\left(1+\log 2\pi\right) \\ &- H_{\mathrm{r}}(Y_{\mathrm{r}}) - H_{\mathrm{i}}(Y_{\mathrm{i}}) - \alpha\|\mathbf{w}_{\mathrm{a}}(m)\|^2.\end{aligned} \tag{7.24}$$

In order to employ the gradient algorithm, we take the partial derivative of (7.24)

$$\begin{aligned}\frac{\partial \mathcal{J}(Y(m);\alpha)}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} &= \frac{1}{|\sigma_Y^2|}\frac{\partial |\sigma_Y^2|}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \frac{\partial H_{\mathrm{r}}(Y_{\mathrm{r}})}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \frac{\partial H_{\mathrm{i}}(Y_{\mathrm{i}})}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \alpha\mathbf{w}_{\mathrm{a}}(m) \\ &= \frac{1}{|\sigma_Y^2|}\frac{\partial |\sigma_Y^2|}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \frac{1}{\hat{\sigma}_{|Y_{\mathrm{r}}|}}\frac{\partial \hat{\sigma}_{|Y_{\mathrm{r}}|}}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \frac{1}{\hat{\sigma}_{|Y_{\mathrm{i}}|}}\frac{\partial \hat{\sigma}_{|Y_{\mathrm{i}}|}}{\partial \mathbf{w}_{\mathrm{a}}{}^{*}(m)} - \alpha\mathbf{w}_{\mathrm{a}}(m).\end{aligned} \tag{7.25}$$

We can readily calculate $\hat{\sigma}_{|Y_\mathrm{r}|}$ and $\hat{\sigma}_{|Y_\mathrm{i}|}$ in (7.25) based on (6.23). Each derivative can be obtained by replacing the magnitude $|Y(k,m)|$ with an absolute value of the real part $|Y_\mathrm{r}(k,m)|$ or that of the imaginary part $|Y_\mathrm{i}(k,m)|$ in (7.21). The derivatives of the absolute values of the real and imaginary parts can be expressed, respectively, as

$$\frac{\partial |Y_\mathrm{r}(k,m)|}{\partial \mathbf{w}_\mathrm{a}{}^*(m)} = -\frac{1}{2}\mathbf{B}^H(m)\mathbf{X}(k,m)\cdot \mathrm{sign}(Y_\mathrm{r}(k,m)) \tag{7.26}$$

and

$$\frac{\partial |Y_\mathrm{i}(k,m)|}{\partial \mathbf{w}_\mathrm{a}{}^*(m)} = j\frac{1}{2}\mathbf{B}^H(m)\mathbf{X}(k,m)\cdot \mathrm{sign}(Y_\mathrm{i}(k,m)). \tag{7.27}$$

Equations (7.25) through (7.27) are used for the gradient algorithm.

7.2.3 Simulations and Discussions

The conventional beamforming algorithms would attempt to null out any interfering signal, but are prone to the signal cancellation problem [12] whenever there is an interfering signal that is correlated with the desired signal. In realistic environments, interference signals are highly correlated with a target signal since the target signal is reflected from hard surfaces such as walls and tables. Therefore, the adaptation of the weight vector is usually halted whenever the desired source is active.

Many techniques have been proposed in the literature to avoid signal cancellation. Perhaps the best-known of such algorithms is the robust beamformer in GSC configuration proposed by Hoshuyama *et al.* [18]. In the lower branch, their algorithm adaptively estimates a blocking matrix which cancels the signal correlated with the output from the upper branch. Accordingly, the reflections of a desired signal can be eliminated from the lower branch by the adaptive blocking matrix (ABM). The coefficient of the ABM has upper and lower limits in order to specify the maximum allowable target-direction error. Then, the active weight vectors are estimated so as to minimize the output of the beamformer. Since the ABM can remove the reflections from the lower branch, the signal cancellation problem is alleviated. However, the ABM cancels not only

the reflections but also interference signals in the case that the output of the upper branch contains the interference components. In this case, their algorithm is unable to suppress the leaked interference signals. In reality, the interference signals are often present in the upper branch due to steering errors and *spatial aliasing* [6, §13.1.4]. Therefore, Hoshuyama's algorithm requires in some sense a trade-off between the avoidance of signal cancellation and suppression of the interference signals. This problem can be solved by simply halting the adaptation of the ABM and only updating the active weight vectors in the case of a high signal-to-noise ratio (SNR) [20]. Such a switching algorithm is based on SNR, however, and requires complicated rules which must generally be determined empirically.

The TF-GSC beamformer described in Section 5.3 takes into account the transfer functions from the desired source to the microphones into the upper branch of the GSC. The quiescent vectors are calculated with the estimated ratios as indicated in (5.13). The blocking matrices are then computed so as to satisfy the orthogonality condition with those quiescent weight vectors. Thus, it can avoid the leakage of the desired signal into the lower branch. It, however, needs to estimate the ratios of the transfer function without source positions in acoustically stationary environments. It is difficult to obtain stable solutions under non-stationary conditions. Although the algorithm proposed by Gannot et al. can be used in moderately reverberant environments, it does not reduce the amount of reverberation in the final signal [64].

As explained in Section 5.4, the generalized eigenvector (GEV) beamforming algorithm also incorporates the transfer function from the source to the microphones indirectly. It was demonstrated that their method could reduce signal distortion and noise more than the TF-GSC without post-filtering. It was also shown in [22] that their GEV beamforming algorithm can achieve almost the same noise suppression performance of the theoretical upper bound obtained by Hoshuyama's beamformer.

Based on the solutions mentioned above that have appeared in the literature, it could be argued that conventional robust beamforming algorithms with SOS

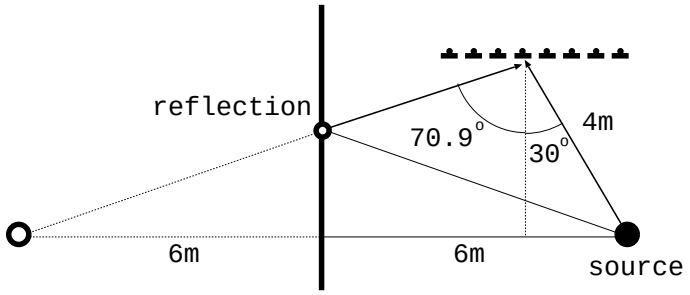
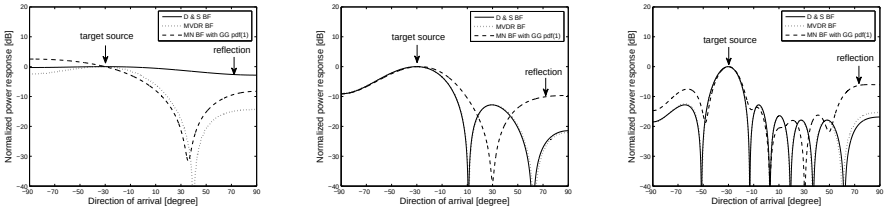


Figure 7.5: Configuration of a source, sensors, and reflective surface for simulation.

have essentially addressed the problem of removing reflections that are highly correlated with the target signal in order to circumvent the signal cancellation problem.

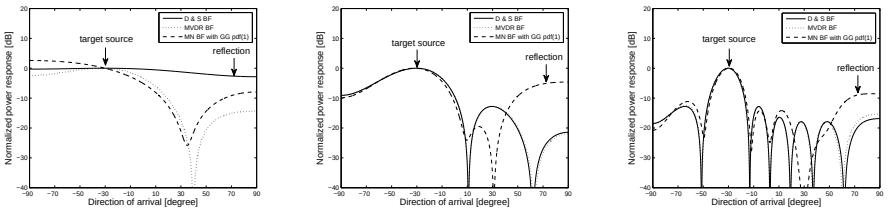
In contrast, the MN beamforming algorithm uses the reflections from the desired source in order to enhance the target signal in addition to eliminating the interference signals. Although the reflected signals could be delayed with respect to the direct path, such delays would be compensated through appropriate adjustments of the active weight vectors. Notice that the length of the analysis filter should be long enough to take in the delayed reflections. The MN beamformer can steer both nulls and sidelobes, assuming the desired sound source is statistically independent of the other sources.

In order to verify that the MN beamforming algorithm forms sidelobes directed towards the reflection of a desired signal, we conducted experiments with a simulated acoustic environment. As shown in Fig. 7.5, we considered a simple configuration with a sound source, a reflective surface, and a linear array of eight microphones positioned with 10 cm inter-sensor spacing. Actual speech data were used as a source in this simulation, which was based on the *image method* [62]. White Gaussian noise was added to the output of each microphone to achieve a SNR of 0 dB. It was assumed that the speed of sound is 343.74 meter per second and used a reflection coefficient of 0.7 for the wall. Fig. 7.6 shows beam patterns at $f_s = 150$ Hz, $f_s = 650$ Hz and $f_s = 1600$ Hz obtained



(a) (b) (c)

Figure 7.6: Beam patterns produced by a delay-and-sum beamformer, the MVDR beamformer and the MN beamforming algorithm using a spherical wave assumption for (a) $f_s = 150$ Hz, (b) $f_s = 650$ Hz and (c) $f_s = 1600$ Hz.



(a) (b) (c)

Figure 7.7: Beam patterns produced by a delay-and-sum beamformer, the MVDR beamformer and the MN beamforming algorithm using a plane wave assumption for (a) $f_s = 150$ Hz, (b) $f_s = 650$ Hz and (c) $f_s = 1600$ Hz.

with a delay-and-sum (D&S) beamformer, the MVDR beamformer and the MN beamforming algorithm with the GG pdf of the magnitude. The weights of the MVDR beamformer were optimized for isotropic (diffuse) noise in the simulation; See Section 5.1.1.

The beam patterns in Fig. 7.6 were produced with a near-field source and reflection although a beam pattern shows the sensitivity of an array to plane waves. The author also ran a second set of simulations in which the source and reflection were assumed to produce plane waves. The results of this second simulation are shown in Fig. 7.7. It is clear from these figures that the MN beamformer emphasizes reflections from the desired source. The MVDR

beamformer optimized for the diffuse noise, on the other hand, tends to suppress such reflections. It is also apparent from Fig. 7.6 (a) and Fig. 7.7 (a) that MVDR and MN beamformers can suppress interference at low frequencies, while the suppression performance of the delay-and-sum beamformer is poor at low frequencies.

7.3 Maximum Empirical Kurtosis Beamformer

This section considers kurtosis as a criterion for estimating the active weight vectors in a GSC and describes the beamforming algorithm which optimizes the active weight vectors so as to achieve the output with the *maximum kurtosis* (MK).

Much like the MN beamformer, the MK beamformer can suppress noise and reverberation without the signal cancellation problem. In contrast to negentropy, kurtosis does not require knowledge of the actual pdf of subband samples of speech. Rather, kurtosis can be simply calculated in the non-parametric manner. However, the kurtosis measure is influenced by samples with a low observation probability [1].

It is worth mentioning that Gillespie et al. [65] used the MK criterion to build a multi-microphone speech enhancement system without the GSC implementation and demonstrated speech enhancement with relatively little enrollment data. Applying the MK criterion to a beamformer in GSC configuration enables the beam to be steered as desired.

7.3.1 Estimation of the Active Weight Vectors

With the variance of the outputs $Y(k,m)$, σ_Y^2, the kurtosis of beamformer's output can be expressed as

$$J(Y(m)) = \left(\frac{1}{K} \sum_{k=0}^{K-1} Y^4(k,m) \right) - 3 \left(\sigma_Y^2 \right)^2 . \tag{7.28}$$

By applying the regularization term, we obtain the objective function

$$\mathcal{J}(Y(m);\alpha) = J(Y(m)) - \alpha\|\mathbf{w}_{\mathrm{a}}(m)\|^2. \tag{7.29}$$

We want to find the active weight vectors which maximize the objective function (7.29)

In the absence of a closed-form solution, we must resort to one of the numerical optimization algorithms again. By taking the partial derivative (7.29), we obtain

$$\begin{aligned}\frac{\partial\mathcal{J}(Y(m);\alpha)}{\partial{\mathbf{w}_{\mathrm{a}}}^*(m)} = &\left(\frac{1}{K}\sum_{k=0}^{K-1} -2Y^2(k,m)\mathbf{B}^H(m)\mathbf{X}(k,m)Y^*(k,m)\right)\\ &- 6\sigma_Y^2\left(\frac{1}{K}\sum_{k=0}^{K-1} -\mathbf{B}^H(m)\mathbf{X}(k,m)Y^*(k,m)\right) + \alpha\mathbf{w}_{\mathrm{a}}(m),\end{aligned} \tag{7.30}$$

Equation (7.30) is sufficient to implement a numerical optimization algorithm based on the method of *conjugate gradients* [60, §1.6], whereby the kurtosis of the beamformer's output can be maximized.

Chapter 8

Automatic Speech Recognition (ASR)

The final goal of this thesis is to construct an automatic speech recognition (ASR) system capable of robustly recognizing speech captured with far-field sensors. Therefore, the beamforming algorithms proposed here are all evaluated in terms of recognition performance.

The key point in all ASR systems is to model various sounds of a language to be recognized. In other words, signal components which are unnecessary for speech recognition are disregarded. For example, a minimum variance distortionless response (MVDR) feature extraction technique for the ASR smooths valleys which are readily corrupted by noise while estimating spectrum peaks more accurately [6]. Recognition accuracy, as measured by *word error rate* (WER), is our metric of choice because improvements in simpler metrics such as *signal-to-noise ratio* (SNR) correlate poorly with reductions in WER. Hence, evaluating WER directly is necessary to establish effectiveness of new acoustic beamforming algorithms. This is what is meant by the often repeated adage, "You improve what you *measure*."

This chapter introduces one of the state-of-the-art ASR systems. The rest

of this chapter is organized as follows. Section 8.1 overviews configuration of a modern ASR system. Section 8.2 describes a metric for measuring the recognition performance, a word error rate (WER). The front-end of the ASR system is then reviewed in Section 8.3. Section 8.4 describes how to stochastically model phones with the *hidden Markov model* (HMM). Section 8.5 reviews methods that adapt parameters of models to an unseen speaker and new environment with little data.

8.1 Framework of a Modern ASR System

Figure 8.1 illustrates a block chart of a typical large vocabulary continuous speech recognizer (LVCSR) with a beamformer for recognizing speech from the far-field. In that system, the speaker's position is first determined by a speaker tracking system [66]. Multi-channel signals are then processed with one of the beamforming techniques described in the previous chapters. After that, the LVCSR converts the enhanced speech signal into a sequence of vectors which represents the *speech feature* for discriminating phonemes. Finally, a decoder finds a sequence of words which is most likely to have generated the sequence of the feature vectors. This problem in the decoding process can be formulated as

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{w}|\boldsymbol{O}), \tag{8.1}$$

where $\boldsymbol{O} = [\boldsymbol{o}_0, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_{T_o-1}]$ is the sequence of the feature vectors and $\boldsymbol{w} = [w_0, w_1, \ldots, w_{N_w-1}]$ is the sequence of words. Since it is difficult to model $P(\boldsymbol{w}|\boldsymbol{O})$ directly, Bayes' Rule is usually applied to (8.1) and the problem can be re-rewritten

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\boldsymbol{w})P(\boldsymbol{w}). \tag{8.2}$$

The probablity $P(\boldsymbol{O}|\boldsymbol{w})$ is calculated with an *acoustic model* and P($\boldsymbol{w}$) is referred to as a *language model*. Those probabilities are normally computed in the log domain in order to avoid the floating point underflow error.

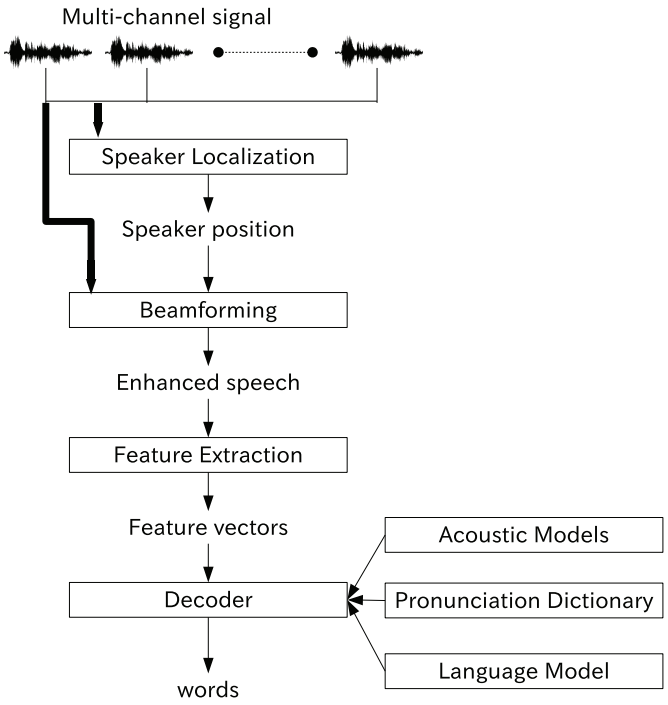


Figure 8.1: Basic block chart of ASR with the beamforming front-end.

8.2 Word Error Rate

The word error rate (WER) is often used for evaluating the performance of the ASR system. In the case that mechanisms of human perception are incorporated into the ASR system, the WER would be related to human perception.

Recognition errors are typically grouped into three types:

- an insertion which occurs when an extra word which is not spoken is recognized
- a substitution which occurs when a correct word is replaced by an incorrect word and
- a deletion which happens when a recognizer fails to hypothesize a word which is spoken.

The minimum error rate can be determined by aligning the hypothesized word string with the correct reference string. This problem is know as maximum substring matching and can be solved by dynamic programming [67]. After the alignment, the WER is calculated as

$$\text{WER} = \frac{\text{substitutions} + \text{deletions} + \text{insertions}}{\text{total number of word tokens in the reference}} \times 100. \tag{8.3}$$

8.3 Feature Extraction

Human speech contains information and characteristics such as pitch, prosody and accent, that, while lending individuality and charm to the voice of a given speaker, are irrelevant to distinguishing between different phones. To obtain optimal performance, such irrelevant characteristics must be eliminated from the features used for recognition. Thus, the feature extraction techniques play an important role in the ASR systems.

Many speech feature extraction algorithms have been proposed and studied over the years. The most widely used methods are based on *Mel-frequency cepstral coefficients* (MFCCs) [6] and *perceptual linear prediction* (PLP) [68].

MFCCs are obtained by truncating discrete cosine transformation (DCT) coefficients of log power spectra smoothed with Mel-filter banks, which mimic the frequency resolution of the human ear. PLP computes linear prediction (LP) coefficients from a perceptually weighted non-linearly compressed power spectrum and then transforms LP coefficients to cepstral coefficients. PLP features can give small improvements over the MFCCs in moderately noisy environments.

The LP methods minimize the squared prediction error, which leads to the elimination of the harmonics present in the original spectrum [69, §3.4] [70]. The effect of nulling out the harmonics is emphasized by increasing the model order of LP. In this process, contours in the spectral envelope become sharper and the harmonics are overestimated [70]. Such effects are problematic for estimating the power spectrums at the harmonic frequencies in voiced speech.

In order to overcome the problems associated with LP, Murthi et al. proposed the minimum variance distortionless response (MVDR) spectral estimation method in [71]. The detailed discussions are found in [70]. Wölfel and McDonough recently proposed a new feature extraction algorithm based on the MVDR spectral estimation [72]. They estimate the MVDR spectral envelope in the frequency domain warped by the bi-linear transformation (BLT) [73, 74], which takes into account the human perception. In order to improve robustness against additive noise, the warped MVDR spectral envelope is then scaled to the highest peak of the logarithmic power spectrum. They demonstrated that their MVDR feature extraction algorithm can provide the better recognition performance than the MFCC and PLP features. Three different front-ends are thoroughly analyzed in [6]

This section briefly reviews the feature extraction method based on the MVDR spectral estimation.

8.3.1 MVDR-envelope

The MVDR also known as Capon's method or the maximum-likelihood method [75] was originally introduced in [76]. It has been demonstrated in [77]

that this method provides an unbiased minimum variance estimate of the spectral components. Detailed discussions of the speech spectral estimation using the MVDR can be found in [70].

MVDR spectral estimation can be posed as a problem in filter bank design, wherein the final filter bank is subject to the *distortionless constraint* [78]:

> The signal at the frequency of interest ω_{foi} must pass undistorted with unity gain.

This can be expressed as

$$H(e^{j\omega_{\text{foi}}}) = \sum_{m=0}^{M} h[m]\, e^{-jm\omega_{\text{foi}}} = 1,$$

where $h[m]$ is the mth sample in the time signal associated with $H(e^{j\omega_{\text{foi}}})$. This constraint can be rewritten in vector form as

$$\mathbf{v}^H(e^{j\omega_{\text{foi}}})\mathbf{h} = 1,$$

where $\mathbf{v}(e^{j\omega_{\text{foi}}})$ is the *fixed frequency vector*

$$\mathbf{v}(e^{j\omega}) = [1, e^{-j\omega}, \ldots, e^{-jM\omega}]^T,$$

and

$$\mathbf{h} = [h[0], h[1], \ldots, h[M]]^T.$$

The distortionless filter $\mathbf{h}$ can now be obtained by solving the constrained minimization problem

$$\min_{\mathbf{h}} \mathbf{h}^H \mathbf{\Phi} \mathbf{h} \text{ subject to } \mathbf{v}^H(e^{j\omega_{\text{foi}}})\mathbf{h} = 1, \tag{8.4}$$

where $\mathbf{\Phi}$ is the $(M+1)\cdot(M+1)$ Toeplitz autocorrelation matrix with (m,n)th element $\phi_{m,n} = R[m-n]$ of the input signal

$$R[n] = \sum_{m=0}^{M} x[m]x[m-n].$$

The solution to the constrained minimization problem can be found, for example, in [78] as

$$\mathbf{h} = \frac{\mathbf{\Phi}^{-1}\mathbf{v}(e^{j\omega_{\text{foi}}})}{\mathbf{v}^H(e^{j\omega_{\text{foi}}})\mathbf{\Phi}^{-1}\mathbf{v}(e^{j\omega_{\text{foi}}})}.$$

This implies that $\mathbf{h}$ is the impulse response of the distortionless filter for the frequency ω_{foi}. The MVDR-envelope of the spectrum $S(e^{-j\omega})$ at frequency ω_{foi} is then obtained as the output of the optimized constrained filter

$$S_{\text{MVDR}}(e^{j\omega_{\text{foi}}}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|H(e^{j\omega_{\text{foi}}})\right|^2 S(e^{-j\omega})d\omega. \tag{8.5}$$

Although MVDR spectral estimation was posed as a problem of designing a distortionless filter for a given frequency ω_{foi}, this was only a conceptual device. The MVDR spectral envelope can in fact be represented in parametric form for all frequencies and computed as

$$S_{\text{MVDR}}(e^{j\omega_{\text{foi}}}) = \frac{1}{\mathbf{v}^H(e^{j\omega_{\text{foi}}})\mathbf{\Phi}^{-1}\mathbf{v}(e^{j\omega_{\text{foi}}})}. \tag{8.6}$$

Under the assumption that the $(M+1)\cdot(M+1)$ Hermitian Toeplitz correlation matrix $\mathbf{\Phi}$ is positive definite and thus invertible, [75] derived a fast algorithm to calculate the MVDR spectral envelope from a set of LP coefficients.

The MVDR-envelope copes well with the problem of overestimation of the spectral power at the harmonics of voiced speech [6]. Hence, the MVDR-envelope models the perceptually important speech harmonics very well. Unlike warped-envelopes, however, it neither mimics the human auditory system nor model the different frequency bands with varying accuracy.

8.3.2 Warped MVDR-envelope

To overcome the problems inherent in LP while emphasizing the perceptually relevant portions of the spectrum, the bi-linear transformation must be applied prior to MVDR spectral envelope estimation [72]. The derivation of the so-called *warped MVDR* will be presented in this section.

To obtain the solution of the distortionless filter $\tilde{\mathbf{h}}$ in the warped domain, we must once more solve the constrained minimization problem, wherein the constraint is applied in the warped frequency domain

$$\min_{\tilde{\mathbf{h}}} \tilde{\mathbf{h}}^H\tilde{\mathbf{\Phi}}\tilde{\mathbf{h}} \text{ subject to } \tilde{\mathbf{v}}^H(e^{j\omega_{\text{foi}}})\tilde{\mathbf{h}} = 1 \tag{8.7}$$

where $\tilde{\mathbf{v}}$ is defined as the *warped frequency vector*

$$\tilde{\mathbf{v}}(e^{j\omega}) = \left[1, \frac{e^{-j\omega}-\alpha}{1-\alpha\cdot e^{-j\omega}}, \cdots, \frac{e^{-jM\omega}-\alpha}{1-\alpha\cdot e^{-jM\omega}}\right]^T.$$

and $\tilde{\mathbf{\Phi}}$ is the Toeplitz autocorrelation matrix which can be, for example, calculated by Matsumoto's method [79]; see also [6, §5.3.6] for details.

The solution to the warped constrained minimization problem is very similar to its unwarped counterpart. The warped MVDR-envelope of the spectrum $S(e^{-j\omega})$ at frequency ω_{foi} can be obtained as the output of the optimal filter,

$$S_{\text{warpedMVDR}}(e^{j\omega_{\text{foi}}}) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\tilde{H}(e^{j\omega_{\text{foi}}})\right|^2 S(e^{-j\omega})d\omega, \tag{8.8}$$

under the constraint

$$\tilde{H}(e^{j\omega_{\text{foi}}}) = \sum_{m=0}^{M}\tilde{h}(m)\frac{e^{-jm\omega_{\text{foi}}}-\alpha}{1-\alpha\cdot e^{-jm\omega_{\text{foi}}}} = 1.$$

Assuming that the Hermitian Toeplitz correlation matrix $\tilde{\mathbf{\Phi}}$ is positive definite and thus invertible, Musicus's algorithm [75] can be readily applied to compute the warped MVDR spectral envelope with a little modification [6, §5.3.6].

A warped envelope estimate on the linear frequency scale can be expressed as

$$\tilde{S}_{\text{MVDR}}(e^{j\omega}) = \frac{1}{\sum_{m=-M}^{M}\tilde{\mu}_m\frac{e^{-jm\omega}-\alpha}{1-\alpha\cdot e^{-jm\omega}}}. \tag{8.9}$$

where with the prediction error in the warped domain $\tilde{e}_M$ and the LPC $a_{0\cdots M}^{(M)}$ of order M

$$\mu_m = \begin{cases} \dfrac{1}{\tilde{e}_M}\displaystyle\sum_{i=0}^{M-m}(M+1-m-2i)a_i^{(M)}a_{i+m}^{*(M)} & ,\ m=0,\cdots,M \\ \\ \mu_{-m}^* & ,\ m=-M,\cdots,-1. \end{cases}$$

8.3.3 Scaled MVDR-envelope

The spectral peaks in the non-logarithmic domain can be influenced by additive noise. In contrast, the peaks in the logarithmic domain are known to be particularly robust to additive noise, as $\log(a+b)\approx\log(\max\{a,b\})$ [80]. It was

shown in [72] that the spectral peaks of the logarithmic warped MVDR envelope are also not as robust to noise as the spectral peaks of the logarithmic power spectrum. Accordingly, the MVDR spectrum is matched to the highest spectral peak of the logarithmic power spectrum [72].

8.3.4 Feature Projection

Feature projection is used for reducing the number of dimensions of a feature vector. It can improve robustness of estimation of classifier's parameters especially when the amount of training data is scarce.

Although the feature extraction algorithms described in the previous sections can reduce the dimensionality of the feature vector at each frame, the feature projection method is generally used for capturing dynamics of speech characteristics from the sequence of the feature vectors effectively.

The most straightforward approach would be the linear transformation

$$\boldsymbol{o}_t = \boldsymbol{A}_{[p]}\bar{\boldsymbol{o}}_t, \tag{8.10}$$

where $\boldsymbol{A}_{[p]}$ is a $p \times d$ matrix, d is the dimension of the source vector $\bar{\boldsymbol{o}}_t$ formed by concatenating the static feature vectors of several frames.

Different criteria for estimating $\boldsymbol{A}_{[p]}$ have been used in the ASR systems and are summarized as:

- the minimum reconstruction error achieved by principal component analysis (PCA),
- the maximum class separability obtained by linear discriminant analysis (LDA) [81] or heteroscedastic discriminant analysis (HDA) [82], and
- the best class separability whilst ensuring that distributions for all dimensions to be removed are the same, e.g., heteroscedastic LDA (HLDA) [83].

Although the HLDA-based method outperforms the other algorithms, it is more computationally expensive and requires more memory. Full covariance

matrix statistics for each component are required to estimate an HLDA transform, whereas only the average within and between class covariance matrices are required for LDA. This makes the HLDA projections from large dimensional features spaces with large numbers of components impractical. Accordingly, LDA is used in experiments described in Chapter 9.

8.4 HMM Parameter Estimation

The hidden Markov model (HMM) can represent a variety of speech characteristics. Modern ASR systems are based on the HMM trained from a huge amount of speech data associated with correct transcriptions [84, 85, 86, 87, 88, 89, 90].

Training algorithms of the HMMs could be grouped into two approaches : maximum likelihood (ML) estimation and discriminative estimation. The parameter estimation based on the ML criterion is perhaps simpler and more popular. Therefore, the ML estimation algorithm will be explained here. Readers who are interested in the discriminative training algorithms can find the details in [6].

The rest of this section is organized as follows. Section 8.4.1 describes a structure of a HMM. Initialization and re-estimation algorithms are then described in Section 8.4.2 and 8.4.3, respectively. A transformation method of HMM's covariance matrix is depicted in Section 8.4.4.

8.4.1 Structure of HMM

In the similar manner with the decomposition of the words as mentioned in Section 8.1, each spoken word w can be also decomposed into a sequence of K_w phones. This sequence is called its pronunciation $\boldsymbol{q}_{1:K_w}^{(w)} = q_1, \cdots, q_{K_w}$. To allow for the possibility of multiple pronunciations, the likelihood $P(\boldsymbol{O}|\boldsymbol{w})$ can be expressed

$$P(\boldsymbol{O}|\boldsymbol{w}) = \sum_{\boldsymbol{Q}} P(\boldsymbol{O}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{w}), \tag{8.11}$$

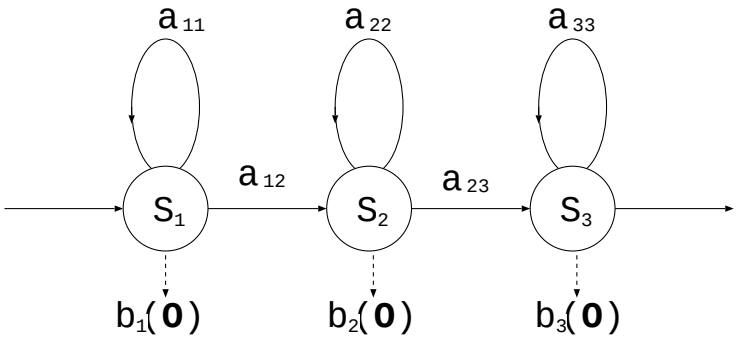


Figure 8.2: An example of a HMM structure.

where the summation is over all the valid pronunciation sequences for $\boldsymbol{w}$, $\boldsymbol{Q}$ is a particular sequence of pronunciations,

$$P(\boldsymbol{Q}|\boldsymbol{w}) = \prod_{l=1}^{L} P(\boldsymbol{q}^{(w_l)}|w_l), \tag{8.12}$$

and where each $\boldsymbol{q}^{(w_l)}$ is a valid pronunciation for word w_l.

Figure 8.2illustrates a phone model represented by a continuous density HMM with states $\{s_j\}$ which are associated with output distributions $\{b_j(\cdot)\}$ and their transition probabilities $\{a_{ij}\}$. At the state s_j, a feature vector is emitted with the probability $b_j(\boldsymbol{o})$ and the state transition from the state s_i to s_j occurs with the probability a_{ij}. In particular, the HMM shown in Figure 8.2is called the left-to-right HMM which allows states to transit in temporal order.

For the output distribution, a Gaussian mixture model (GMM) is typically used as

$$b_j(\boldsymbol{o}) \quad = \quad \sum_{m=1}^{M} c_{jm}\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}), \tag{8.13}$$

where c_{jm} is the mixture weight which satisfies $\sum_{m=1}^{M} c_{jm} = 1$, $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

In the case that the HMM uses the GMMs, we have to estimate the mixture weights, the mean vectors and covariance matrices as well as the transition probabilities $\{a_{ij}\}$ for each phone.

The observation vectors might consist of different modal features such as speech (audio) and lip image (visual) features. A concept of the *stream* has been introduced to the HMM in order to represent such multi-modal features. The output probability of the HMM with the multiple streams can be expressed as

$$\begin{aligned} b_j(\boldsymbol{o}) &= \prod_{s=1}^{N_r} b_j(\boldsymbol{o}_s) \\ &= \prod_{s=1}^{N_r} \sum_{m=1}^{M_s} c_{jsm} \mathcal{N}(\boldsymbol{o}_s; \boldsymbol{\mu}_{jsm}, \boldsymbol{\Sigma}_{jsm}) \end{aligned} \tag{8.14}$$

where $\boldsymbol{o}_s$ represents the observation vector at stream s and N_r is the number of streams.

In the multi-stream HMM, events of a stream are assumed to be stochastically independent of each other at a state while the transition probabilities can be tied together, which implies that the events changes synchronously [91, 92].

The multi-stream HMM framework is often applied to the audio-visual ASR which can improve the recognition performance by using visual information such as images around mouths.

Although the single-stream HMM is used in experiments which will described in Chapter 9, the following sections present update formulae for the multi-stream HMM which are more general expressions. We can easily obtain the formulae for the normal HMM by setting the number of streams to 1.

8.4.2 Viteribi Training (Initialization)

After the basic structure of the HMM is defined, the HMM's parameters such as the mean vectors, covariance matrices and mixture weights have to be initialized. For the initialization, each training observation, $\boldsymbol{O}^r$, $1 \leq r \leq R$, is uniformly divided into N equal segments so that each uniform segment is associated to each state. The parameters of the HMM are then computed with the uniformly segmented features.

After the first estimation with the uniform segmentation, the Viterbi algorithm finds the most likely state sequence corresponding to the feature vectors and then assign a set of the observation vectors to each state. These processes are repeated until the total likelihood converges. The parameters of the state are then estimated with the associated feature vectors.

Apart from the first iteration on the new model, each training sequence $\boldsymbol{O}$ is segmented using the Viterbi algorithm which results from maximizing

$$\phi_{N_s}(T-1) = \max_i \phi_i(T-1)a_{iN_s}$$

where

$$\phi_j(t) = \left[\max_i \phi_i(t-1)a_{ij}\right] b_j(\boldsymbol{o}_t)$$

with initial conditions given by

$$\phi_1(0) = 1$$

$$\phi_j(0) = a_{1j}b_j(\boldsymbol{o}_0).$$

In this and all subsequent cases, i indicates the previous state, j is the index of the state at the time instance t, and the output probability $b_j(\cdot)$ is defined as (8.14).

If A_{ij} represents the total number of transitions from state i to state j in performing the above maximizations, then the transition probabilities can be estimated by counting the relative frequencies

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_{k=1}^{N_s} A_{ik}}$$

The sequence of states which maximizes $\phi_N(T-1)$ implies an alignment of training samples with states. Within each state, a further alignment of observations to the mixture components is made. For each state and each stream

1. use clustering to allocate each observation $\boldsymbol{o}_{st}$ to one of M_s clusters, or

2. associate each observation $\boldsymbol{o}_{st}$ with the mixture component with the highest probability

In either case, the result is that every observation is associated with a single unique mixture component. This association can be represented by the indicator function $\psi^r_{jsm}(t)$ which is 1 if $\boldsymbol{o}^r_{st}$ is associated with mixture component m of stream s of state j and is zero otherwise.

The means and variances are then estimated via simple averages

$$\hat{\boldsymbol{\mu}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\psi^r_{jsm}(t)\boldsymbol{o}^r_{st}}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\psi^r_{jsm}(t)}$$

$$\hat{\boldsymbol{\Sigma}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\psi^r_{jsm}(t)(\boldsymbol{o}^r_{st}-\hat{\boldsymbol{\mu}}_{jsm})(\boldsymbol{o}^r_{st}-\hat{\boldsymbol{\mu}}_{jsm})'}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\psi^r_{jsm}(t)}$$

Finally, the mixture weights are based on the number of observations allocated to each component

$$\boldsymbol{c}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\psi^r_{jsm}(t)}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1}\sum_{l=1}^{M_s}\psi^r_{jsl}(t)}$$

8.4.3 Baum-Welch Training (Re-estimation)

Baum-Welch training is similar to the Viterbi training method described in the previous section except that the *hard* boundary determined by the ψ function is replaced by a *soft* boundary function L which represents the probability of an observation associated with a Gaussian mixture component. This *occupation* probability is computed from the *forward* and *backward* probabilities. Baum-Welch training has two styles, an isolated training style where each phone model is individually updated and embedded training one where a word model concatenated from several phone models without phone labels is re-estimated.

Isolated training

For the isolated-unit style of training, the forward probability $\alpha_j(t)$ for $1 \leq j \leq N_s$ and $0 < t < T$ is calculated by the forward recursion

$$\alpha_j(t) = \left[\sum_{i=1}^{N_s}\alpha_i(t-1)a_{ij}\right] b_j(\boldsymbol{o}_t)$$

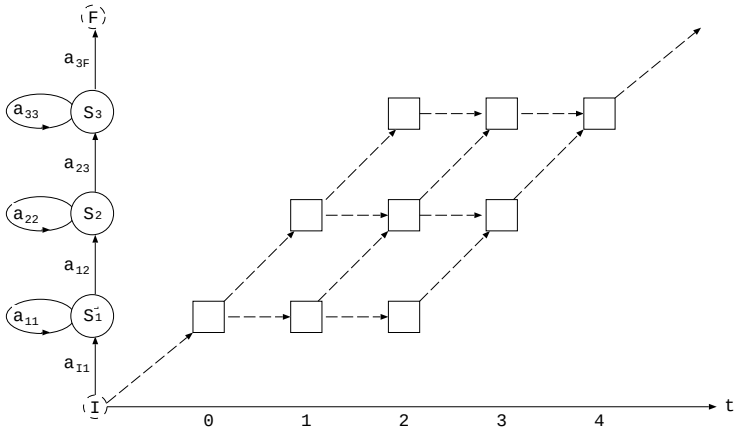


Figure 8.3: Visualization of computing the forward probabilities.

with initial conditions given by

$$\alpha_j(0) = a_{Ij} b_j(\boldsymbol{o}_0)$$

for $1 \leq j \leq N_s$ and final condition given by

$$\alpha_F(T-1) = \sum_{i=1}^{N_s} \alpha_i(T-1) a_{iF}$$

where a_{Ij} indicates the initial transition path to the state j and a_{iF} is the transition probability from state i to the last state with no emission.

Figure 8.3 visualizes the procedure of the calculation of the forward probabilities in the case of $T = 5$ and $N_s = 3$. In the figure, the horizontal axis indicates the frame t, the vertical axis corresponds to the state s and each rectangle box in the trellis is associated with the forward probability of each state at each frame. The forward probability can be obtained by recursively multiplying the transition probabilities with the emission probabilities along the paths indicated by the broken arrows in Figure 8.3.

The backward probability $\beta_i(t)$ for $1 \leq i \leq N_s$ and $T-1 > t \geq 0$ is calculated by the backward recursion

$$\beta_i(t) = \sum_{j=1}^{N_s} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_j(t+1)$$

with initial conditions given by

$$\beta_i(T-1) = a_{iF}$$

for $1 \leq i \leq N_s$ and final condition given by

$$\beta_I(0) = \sum_{j=1}^{N_s} a_{Ij} b_j(\boldsymbol{o}_0)\beta_j(0).$$

Notice that isolated training requires phone labels.

In this style of model training, a set of training observations $\boldsymbol{O}^r$, $1 \leq r \leq R$, is used to estimate the parameters of a single HMM. The basic formula for the re-estimation of the transition probabilities is

$$\hat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-2} \alpha_i^r(t) a_{ij} b_j(\boldsymbol{o}_{t+1}^r)\beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^r(t)\beta_i^r(t)}$$

where $1 \leq i \leq N_s$ and $1 \leq j \leq N_s$ and P_r is the total probability $P = \text{prob}(\boldsymbol{O}^r|\lambda)$ of the r-th observation. The transitions from the non-emitting entry state are re-estimated by

$$\hat{a}_{Ij} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P_r} \alpha_j^r(0)\beta_j^r(0)$$

where $1 \leq j \leq N_s$ and the transitions from the emitting states to the final non-emitting exit state are re-estimated by

$$\hat{a}_{iF} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \alpha_i^r(T-1)\beta_i^r(T-1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^r(t)\beta_i^r(t)}$$

where $1 \leq i \leq N_s$.

For the HMM with M_s mixture components in stream s, the means, covariances and mixture weights are re-estimated as follows. Firstly, the probability of occupying the m-th mixture component in stream s at time t for the r-th observation is

$$L_{jsm}^r(t) = \frac{1}{P_r} U_j^r(t) c_{jsm} b_{jsm}(\boldsymbol{o}_{st}^r)\beta_j^r(t) b_{js}^*(\boldsymbol{o}_t^r)$$

where

$$U_j^r(t) = \begin{cases} a_{Ij} & \text{if } t = 0 \\ \sum_{i=1}^{N_s} \alpha_i^r(t-1) a_{ij} & \text{otherwise} \end{cases} \tag{8.15}$$

and

$$b^*_{js}(\boldsymbol{o}^r_t) = \prod_{k \neq s} b_{jk}(\boldsymbol{o}^r_{kt})$$

For a single Gaussian stream, the probability of mixture component occupancy is equal to the probability of state occupancy and hence it is more efficient in this case to use

$$L^r_{jsm}(t) = L^r_j(t) = \frac{1}{P_r}\alpha_j(t)\beta_j(t)$$

Given the above definitions, the re-estimation formulae may now be expressed in terms of $L^r_{jsm}(t)$ as follows.

$$\hat{\boldsymbol{\mu}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_{jsm}(t)\boldsymbol{o}^r_{st}}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_{jsm}(t)}$$

$$\hat{\boldsymbol{\Sigma}}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_{jsm}(t)(\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})(\boldsymbol{o}^r_{st} - \hat{\boldsymbol{\mu}}_{jsm})'}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_{jsm}(t)} \tag{8.16}$$

$$\boldsymbol{c}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_{jsm}(t)}{\sum_{r=1}^{R}\sum_{t=0}^{T_r-1} L^r_j(t)}$$

Embedded training

In the case of embedded training, the HMM spanning the observations is a composite constructed by concatenating Q subword models. An advantage of embedded training is that the phone labels which require labor of expert labelers are not needed.

For the forward probability, the initial conditions are established at time $t = 0$ as follows

$$\alpha^{(q)}_j(0) = \begin{cases} \alpha^{(q)}_j(0) = a^{(q)}_{Ij} b^{(q)}_j(\boldsymbol{o}_0) & \text{if } q = 1 \\ 0 & \text{otherwise} \end{cases}$$

where the superscript q in parentheses refers to the index of the model in the sequence of concatenated models. All unspecified values of α are zero. For time $t > 0$,

$$\alpha^{(q)}_j(t) = \begin{cases} \left[a^{(q)}_{Ij} + \sum_{i=1}^{N_s^{(q-1)}} \alpha^{(q)}_i(t-1)a^{(q)}_{ij}\right] b^{(q)}_j(\boldsymbol{o}_t) & \text{if } q = 1 \\ \left[\sum_{i=1}^{N_s^{(q-1)}} \alpha^{(q-1)}_i(t-1)a^{(q-1,q)}_{ij} + \sum_{i=1}^{N_s^{(q)}} \alpha^{(q)}_i(t-1)a^{(q)}_{ij}\right] b^{(q)}_j(\boldsymbol{o}_t) & \text{otherwise} \end{cases}$$

$$\alpha_F^{(q)}(t) = \sum_{i=1}^{N_s^{(q)}} \alpha_i^{(q)}(t) a_{iF}^{(q)}$$

For the backward probability, the initial conditions are set at time $t = T-1$ as follows

$$\beta_i^{(q)}(T-1) = \begin{cases} a_{iF}^{(q)} & \text{if } q = Q \\ 0 & \text{otherwise} \end{cases}$$

where once again, all unspecified β values are zero. For time $t < T-1$,

$$\beta_i^{(q)}(t) = \begin{cases} \sum_{j=1}^{N_s^{(q)}} a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}) \beta_j^{(q)}(t+1) + \sum_{j=1}^{N_s^{(q)}} a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}) a_{jF}^{(q)} & \text{if } q = Q \\ \sum_{j=1}^{N_s^{(q)}} a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}) \beta_j^{(q)}(t+1) + \sum_{j=1}^{N_s^{(q+1)}} a_{ij}^{(q,q+1)} b_j^{(q+1)}(\boldsymbol{o}_{t+1}) \beta_j^{(q+1)}(t+1) & \text{otherwise} \end{cases}$$

$$\beta_I^{(q)}(t) = \sum_{j=1}^{N_s^{(q)}} a_{Ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_t) \beta_j^{(q)}(t)$$

The total probability $P_r = \text{prob}(\boldsymbol{O}|\lambda)$ can be computed from either the forward or backward probabilities

$$P_r = \alpha_F(T-1) = \beta_I(0)$$

The basic formula for the re-estimation of the transition probabilities within a phone model is

$$\hat{a}_{ij}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-2} \alpha_i^{(q)r}(t) a_{ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_{t+1}^r) \beta_j^{(q)r}(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}.$$

The transitions between two phone HMMs are re-estimated by

$$\hat{a}_{ij}^{(q,q+1)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-2} \alpha_i^{(q)r}(t) a_{ij}^{(q,q+1)} b_j^{(q+1)}(\boldsymbol{o}_{t+1}^r) \beta_j^{(q+1)r}(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t) + \alpha_i^{(q)r}(t) a_{ij}^{(q,q+1)} b_j^{(q+1)}(\boldsymbol{o}_{t+1}^r) \beta_j^{(q+1)r}(t+1)}$$

and the transitions from the non-emitting entry to the first emission states are re-estimated by

$$\hat{a}_{Ij}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} a_{Ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_t^r) \beta_j^{(q)r}(t)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \beta_I^{(q)r}(t)}$$

Finally, the direct transitions to the non-emitting exit state are re-estimated by

$$\hat{a}_{iF}^{(q)} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^{(q)r}(t) a_{iF}^{(q)}}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=0}^{T_r-1} \alpha_i^{(q)r}(t) \beta_i^{(q)r}(t)}$$

The re-estimation formulae for the output distributions are the same as those for the single model case except for the obvious additional subscript for q. However, the probability calculations must now allow for transitions from the entry states by changing $U_j^r(t)$ in (8.15) to

$$U_j^{(q)r}(t) = \begin{cases} a_{Ij}^{(q)} b_j^{(q)}(\boldsymbol{o}_t^r) & \text{if } t = 0 \\ \sum_{i=1}^{N_s^{(q-1)}} \alpha_i^{(q-1)}(t-1) a_{ij}^{(q-1,q)} + \sum_{i=1}^{N_q} \alpha_i^{(q)r}(t-1) a_{ij}^{(q)} & \text{otherwise} \end{cases}$$

8.4.4 Semi-Tied Covariance

Semi-tied transforms [93] and the updates for HLDA are very similar.

Semi-tied covariance matrices have the form

$$\boldsymbol{\mu}_{m_r} = \boldsymbol{\mu}_{m_r}, \quad \boldsymbol{\Sigma}_{m_r} = \boldsymbol{H}_r \boldsymbol{\Sigma}_{m_r}^{\texttt{diag}} \boldsymbol{H}_r^T. \tag{8.17}$$

For efficiency reasons the transforms are stored and likelihoods are calculated using

$$\begin{aligned} \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{m_r}, \boldsymbol{H}_r \boldsymbol{\Sigma}_{m_r}^{\texttt{diag}} \boldsymbol{H}_r^T) &= \frac{1}{|\boldsymbol{H}_r|} \mathcal{N}(\boldsymbol{H}_r^{-1}\boldsymbol{o}; \boldsymbol{H}_r^{-1}\boldsymbol{\mu}_{m_r}, \boldsymbol{\Sigma}_{m_r}^{\texttt{diag}}) \\ &= |\boldsymbol{A}_r| \mathcal{N}(\boldsymbol{A}_r\boldsymbol{o}; \boldsymbol{A}_r\boldsymbol{\mu}_{m_r}, \boldsymbol{\Sigma}_{m_r}^{\texttt{diag}}), \end{aligned} \tag{8.18}$$

where $\boldsymbol{A}_r = \boldsymbol{H}_r^{-1}$. The transformed mean, $\boldsymbol{A}_r\boldsymbol{\mu}_{m_r}$, is stored in advance for efficiency.

The estimation of the semi-tied transforms is a doubly iterative process. Given a current set of covariance matrix estimates, the semi-tied transforms are estimated in a similar fashion to the full variance transforms.

$$\mathbf{a}_{ri} = \mathbf{c}_{ri} \mathbf{G}_r^{(i)-1} \sqrt{\left(\frac{\beta_r}{\mathbf{c}_{ri} \mathbf{G}_r^{(i)-1} \mathbf{c}_{ri}^T} \right)}, \tag{8.19}$$

where $\mathbf{a}_{ri}$ is ith row of $\boldsymbol{A}_r$, the $1 \times n$ row vector $\mathbf{c}_{ri}$ is the vector of cofactors of $\boldsymbol{A}_r$, $c_{rij} = \text{cof}(\boldsymbol{A}_{rij})$, and $\mathbf{G}_r^{(i)}$ is defined as

$$\mathbf{G}_r^{(i)} = \sum_{m_r=1}^{M_r} \frac{1}{\sigma_{m_r i}^{\texttt{diag2}}} \sum_{t=0}^{T-1} L_{m_r}(t) (\boldsymbol{o}(t) - \boldsymbol{\mu}_{m_r})(\boldsymbol{o}(t) - \boldsymbol{\mu}_{m_r})^T \boldsymbol{\Sigma}_{m_r}. \tag{8.20}$$

This iteratively estimates one row of the transform at a time.

Having estimated the transform, the diagonal covariance matrix is updated as

$$\boldsymbol{\Sigma}_{m_r}^{\mathtt{diag}} = \mathrm{diag}\left(\frac{\boldsymbol{A}_r \sum_{t=0}^{T-1} L_{m_r}(t)(\boldsymbol{o}(t) - \boldsymbol{\mu}_{m_r})(\boldsymbol{o}(t) - \boldsymbol{\mu}_{m_r})^T \boldsymbol{A}_r^T}{\sum_{t=0}^{T-1} L_{m_r}(t)} \right). \quad (8.21)$$

8.5 Adaptation and Normalization

In reality, we cannot collect speech data of every speaker in all the acoustic environments. It is inevitable for ASR systems to encounter a new speaker and noise which acoustic models do not represent well. The recognition performance will degrade seriously because of the mismatch between training and test data.

One of the best solutions to that problem is *adaptation*. Adaptation techniques transform the acoustic models with a small amount of data so as to match them to a target speaker. The adaptation algorithms are performed in either *supervised* manner or *unsupervised* way. The supervised adaptation techniques require correct transcriptions for all of the adaptation data and update the acoustic models to the target speaker and environment. In unsupervised training, those transcriptions must be hypothesized.

The rest of this section is organized as follows. First, Section 8.5.1 discusses about the feature transformation techniques which remove unnecessary characteristics for speech recognition from feature vectors. Second, model transformation techniques, where parameters of HMMs are adapted, are described in Section 8.5.2.

8.5.1 Feature Transformation Techniques

The most common feature space adaptation techniques transforms intermediate or the final features used for recognition. These algorithms are used for removing speaker and/or environment-dependent components which have no information to discriminate different phones.

Vocal Tract Length Normalization

A vocal tract length of an individual would shift formant frequencies. Much like a longer pipe in an organ produces a lower tone than a short pipe, the resonances or formants produced by the vocal tract of a taller speaker will generally be lower than those of a shorter speaker, simply because the former will, on average, have a longer vocal tract. Although the vocal tract length is useful information for identifying a speaker, it will not help discriminating different phonemes. Variations in the vocal tract length degrade the performance of ASR.

Although a lot of vocal tract length normalization (VLTN) techniques have been proposed in ASR, one basic idea of the normalization methods is to capture features that appear to have been generated by some average speaker so that differences of the vocal tract lengths are compensated. It is, for example, achieved by linearly scaling the center frequencies of the Mel-filter bank [94].

In order to develop the VTLN algorithm, two issues first need to be addressed:

- definition of the scaling function and
- estimation of the appropriate *warping parameters* of the scaling function for each speaker

In an early stage of the VTLN development, the linear transformation was used as the scaling function. In contrast, Acero [95] and McDonough et al. [73, 74] have proposed applying the bilinear transform (BLT) as a means of achieving a frequency warping effect. The BLT has a useful property that such warping can be achieved through a linear transformation of the cepstral coefficients.

In the VTLN algorithms, the different warping parameters which control how much the formants are mapped up or down are usually estimated for each speaker so that the likelihood of the resulting features with respect to a GMM or HMM is maximized. Parameter estimation is performed using a grid search plotting likelihoods against the parameter values. Once the optimal values for

all training speakers have been computed, the training data is normalized and the acoustic models are re-estimated. This is repeated until the parameters have converged. In the case of very large systems, the amount of computation becomes problematic. An alternative is to approximate the effect of VTLN by a linear transform. The advantage of this approach is that the optimal transformation parameters can be determined from the auxiliary function in a single pass over the data [96].

Most investigators have reported that the performance of recognition can be improved by the VTLN algorithms and further enhanced using the other forms of normalization and speaker adaptation.

8.5.2 Model Transformation Techniques

Model transformation techniques [97, §5.2] update the parameters of the acoustic HMMs with a limited amount of adaptation data.

These model transformation techniques can adapt the acoustic models to not only new speakers but also unknown acoustic environments. In that sense, the model-based adaptation techniques are more powerful than the feature transformation techniques.

Transcriptions for the adaptation data have to be given. In the case that they are not available, outputs from an ASR system with high confidence measures are typically used as the transcriptions.

Maximum Likelihood Linear Regression

Maximum likelihood linear regression (MLLR) algorithms estimate a set of linear transformations for the mean and variance parameters of Gaussian mixtures so as to match the acoustic models to new adaptation data. The parameters are normally estimated based on the maximum likelihood criterion.

MLLR is generally very robust and well suited to unsupervised incremental adaptation. This section presents MLLR in the form of a single global linear transform for all the Gaussian components. The multiple transform case,

where different transforms are used depending on the Gaussian component to be adapted, is discussed later.

There are two main variants of MLLR: unconstrained and constrained [98, 99]. In unconstrained MLLR, separate transforms are trained for the means and variances

$$\begin{aligned} \hat{\boldsymbol{\mu}}^{(sm)} &= \boldsymbol{A}^{(s)}\boldsymbol{\mu}^{(m)} + \boldsymbol{b}^{(s)} \\ \hat{\boldsymbol{\Sigma}}^{(sm)} &= \boldsymbol{H}^{(s)}\boldsymbol{\Sigma}^{(m)}\boldsymbol{H}^{(s)T} \end{aligned} \tag{8.22}$$

where s indicates the speaker. Although (8.22) suggests that the likelihood calculation is expensive to compute, unless $\boldsymbol{H}$ is constrained to be diagonal, it can in fact be made efficient using the following equality

$$\begin{gathered} \mathcal{N}(\boldsymbol{o}; \hat{\boldsymbol{\mu}}^{(sm)}, \hat{\boldsymbol{\Sigma}}^{(sm)}) = \\ \frac{1}{|\boldsymbol{H}^{(s)}|}\mathcal{N}(\boldsymbol{H}^{(s)-1}\boldsymbol{o}; \boldsymbol{H}^{(s)-1}(\boldsymbol{A}^{(s)}\boldsymbol{\mu}^{(m)} + \boldsymbol{b}^{(s)}), \boldsymbol{\Sigma}^{(m)}). \end{gathered} \tag{8.23}$$

If the original covariances are diagonal, then by appropriately caching the transformed observations and means, the likelihood can be calculated at the same cost as when using the original diagonal covariance matrices. For MLLR there are no constraints between the adaptation applied to the means and the covariances. If the two matrix transforms are constrained to be the same, then a linear transform related to the feature-space transforms described earlier may be obtained. This is constrained MLLR (CMLLR)

$$\begin{aligned} \hat{\boldsymbol{\mu}}^{(sm)} &= \tilde{\boldsymbol{A}}^{(s)}\boldsymbol{\mu}^{(m)} + \tilde{\boldsymbol{b}}^{(s)} \\ \hat{\boldsymbol{\Sigma}}^{(sm)} &= \tilde{\boldsymbol{A}}^{(s)}\boldsymbol{\Sigma}^{(m)}\tilde{\boldsymbol{A}}^{(s)T}. \end{aligned} \tag{8.24}$$

In this case, the likelihood can be expressed as

$$\mathcal{N}(\boldsymbol{o}; \hat{\boldsymbol{\mu}}^{(sm)}, \hat{\boldsymbol{\Sigma}}^{(sm)}) = |\boldsymbol{A}^{(s)}|\mathcal{N}(\boldsymbol{A}^{(s)}\boldsymbol{o} + \boldsymbol{b}^{(s)}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}), \tag{8.25}$$

where $\boldsymbol{A}^{(s)} = \tilde{\boldsymbol{A}}^{(s)-1}$ and $\boldsymbol{b}^{(s)} = -\tilde{\boldsymbol{A}}^{(s)-1}\tilde{\boldsymbol{b}}^{(s)}$.

Thus, the actual model parameters are not transformed with this constraint. CMLLR is the form of linear transform most often used for adaptive training.

For both forms of linear transform, the matrix transformation may be full, block-diagonal, or diagonal. For a given amount of adaptation data, more diagonal transforms may be reliably estimated than full ones. However, in practice, full transforms normally outperform larger numbers of diagonal transforms [100]. Hierarchies of transforms of different complexities may also be used [101].

Parameter Estimation The maximum likelihood estimation formulae for the various forms of linear transform are given in [98, 99]. Whereas there are closed-form solutions for the unconstrained mean MLLR, the constrained and unconstrained variance cases are similar to the semi-tied covariance transform discussed in Section 8.4.4 and they require an iterative solution.

Both forms of linear transforms require transcriptions of the adaptation data in order to estimate the model parameters. For supervised adaptation, the transcription is known and may be directly used without further consideration. When used in the unsupervised mode, the transcription must be derived from the recognizer output and in this case, MLLR is normally applied iteratively [102] to ensure that the best hypothesis for estimating the transform parameters is used. First, unknown speech is recognized, then the hypothesized transcription is used to estimate MLLR transforms. The unknown speech is then re-recognized using the adapted models. This is repeated until convergence is achieved. Using this approach, all words within the hypothesis are treated as equally probable.

A refinement is to use recognition lattices in place of the 1-best hypothesis to accumulate the adaptation statistics. This approach is more robust to recognition errors and avoids the need to re-recognize the data since the lattice can be simply re-scored [133]. An alternative use of lattices is to obtain confidence scores, which may then be used for confidence-based MLLR [103].

An initial development of transform-based adaptation methods used the ML criterion and it was then extended to include maximum a posteriori estimation [104]. Linear transforms can also be estimated using discriminative criteria [105]. For supervised adaptation, any of the standard approaches may be

used. However, if unsupervised adaptation is used, for example in BN transcription systems, then there is an additional concern. As discriminative training schemes attempt to modify the parameters so that the posterior of the transcription (or a function thereof) is improved, it is more sensitive to errors in the transcription hypotheses than ML estimation. This is the same issue as was observed for unsupervised discriminative training and, in practice, discriminative unsupervised adaptation is not commonly used.

Regression Class Trees A powerful feature of model transformation techniques is that it can control the number of transforms based on the amount of adaptation data. When the amount of adaptation data is limited, a global transform can be shared across all the Gaussians in the system. As the amount of data increases, the HMM state components can be grouped into regression classes with each class having its own transform, for example A(r) for regression class r. As the amount of data increases further, the number of classes and therefore transforms can be increased correspondingly to give better and better adaptation [99].

In order to automatically determine which tranforms are shared, a regression tree is usually used [84]. Figure 8.4 illustrates the regression tree whose node is typically associated with the class of Gaussian components. A set of transforms which belongs to each node in the tree is shared and treated as a single transform. The total occupation count associated with any node in the tree can easily be computed since the counts are known at the leaf nodes. Then, for a given set of adaptation data, the tree is descended so that the most specific set of nodes is associated with sufficient data. Regression class trees may either be specified using expert knowledge, or more commonly by automatically training the tree by assuming that Gaussian components that are close to one another are transformed using the same linear transform [99].

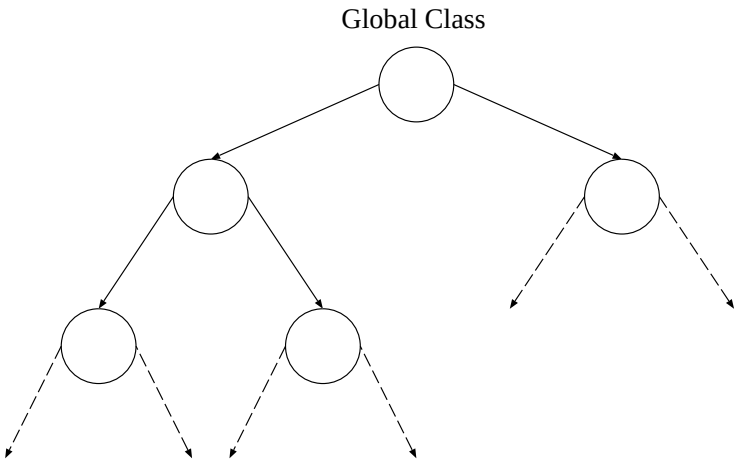


Figure 8.4: An illustration of a regression tree.

Speaker Adaptived Training

In order to build a speaker-independent (SI) speech recognition system, we have to train acoustic models with a large number of speakers. That SI acoustic models will have characteristics of speakers which are irrelevant to information to discriminate phones. One approach to handling this problem is speaker adaptive training (SAT) [106].

Figure 8.5 illustrates schematic of SAT. As shown in Figure 8.5, SAT proceeds along much the same lines as conventional HMM training, with a forward-backward step followed by a parameter update designed to maximize an appropriate auxiliary function. Before training, all utterances in the training set are partitioned by speakers, and the adaptation parameters for the given speaker are used to transform the means of the SI model before the forward-backward pass over each partition. With the completion of the forward-backward step, the speaker-dependent (SD) transformation parameters for the relevant speaker are re-estimated, just as in normal speaker adaptation. The maximization step in SAT includes an iterative parameter update wherein the SI means and variances are each updated in turn while holding all other HMM parameters fixed at their current values. The advantage of the iterative approach lies in the fact that

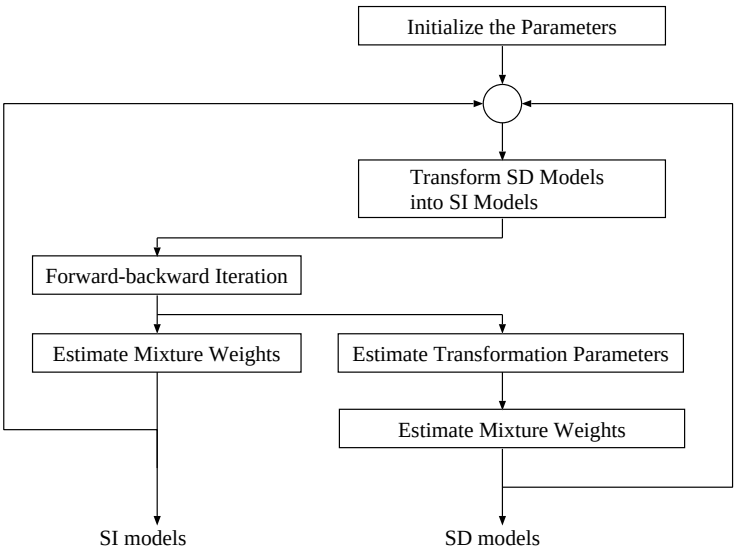


Figure 8.5: Flow charts of SAT training.

a closed-form solution exists for the optimal values for each set of parameters when the other sets are held constant.

Chapter 9

Distant Speech Recognition Experiments

Beamforming algorithms have been traditionally evaluated from the viewpoint of the signal-to-noise ratio (SNR) or speech distance measures such as Itakura-Saito distance [107, §5.3.5]. Subjective tests have been also employed. Not surprisingly, such evaluation results are often based on a few utterance data whose length is less than 10 minutes. The experimental results highly depend on which test data is used or which experimental condition is chosen.

Unlike those skeptical experiments, this work uses *real* speech data for evaluations of beamforming algorithms. Here, the *real* data does not mean the data artificially convoluted with measured impulse responses. The author evaluates beamforming algorithms through a set of speech recognition experiments on data captured with real sensors in a real meeting room.

The rest of this chapter is organized as follows. Section 9.1 describes the specification of multi-channel speech data used in recognition experiments. In Section 9.3, recognition performance achieved by MMI beamforming described in Section 7.1 is investigated in the speech separation task where two speakers are simultaneously speaking. In particular, Section 9.3.1 shows how much

separation performance can be improved by MMI beamforming and examines relationships between recognition performance and pdf assumptions used in MMI beamforming. In Section 9.3.2, we see how important filter bank design methods are in terms of acoustic beamforming. The Nyquist(M) filter bank is also compared to other conventional methods, that is, the perfect reconstruction and de Haan's filter banks. Furthermore, relationships between filter parameters and beamforming performance are thoroughly studied. Section 9.4 shows recognition experiments in the case that a single speaker is talking. In Section 9.4.1, the beamforming methods proposed in Section 7.2 and Section 7.3 are compared to the conventional SOS-based beamformers. Effects of the regularization term on beamforming performance are analyzed in Section 9.4.2. Section 9.4.3 investigates which numerical optimization algorithm is suitable for estimating the active weight vectors in MN and MK beamforming.

9.1 Database Specification

Distant automatic speech recognition (ASR) experiments are performed on the *Multi-Channel Wall Street Journal Audio Visual Corpus* (MC-WSJ-AV) collected by the *Augmented Multi-party Interaction* (AMI) project.

Fig. 9.1 illustrates a configuration of a meeting room where speech data was recorded. The details of the data collection apparatus are also described in [3]. The room size was 650 cm $\times$ 490 cm $\times$ 325 cm and reverberation time T_{60} was approximately 380 milliseconds. In addition to being reverberant, the meeting room data collected includes background noise from computers and the building ventilation. Some recordings also contain audible noise from outside the meeting room, such as that generated by passing cars and speakers in an adjacent room.

The MC-WSJ-AV database contains multi-channel speech data recorded under two conditions:

- Overlapping Speakers Stationary. Here, two speakers are asked to simultaneously read their sentences from different positions within the room. The

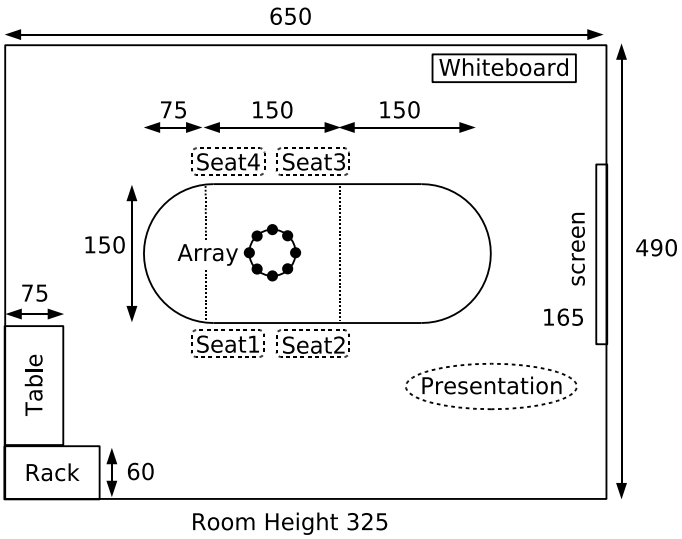


Figure 9.1: A configuration of a meeting room (measurements in cm).

speakers remain in the same positions for the entirety of these recordings and separate recordings are made from each of the 15 pairs of positions.

- Single Speaker Stationary. For this condition the speaker is asked to read sentences from six positions within the meeting room four seated around the table, one standing at the whiteboard and one standing at the presentation screen. One sixth of each speaker's sentences are read from each position.

The far-field speech data was captured with a circular, eight-channel microphone array with a diameter of 20 cm. Additionally, a close-talking microphone was used for each speaker to capture the best possible signal as a reference. The sampling rate of the recordings was 16 kHz.

As the data was recorded with real speakers in a realistic acoustic environment, the positions of the speakers' heads as well as the speaking volume varied even though the speakers were largely stationary. Indeed, it is exactly this behavior of real speakers that makes working with data from corpora such as MC-WSJ-AV so much more challenging than working with data that was played

through a loudspeaker into a room, not to mention data that was *artificially convolved* with previously-measured impulse responses.

9.2 Specification of the ASR system

The distant ASR experiments reported here were conducted with the *Millenium* automatic speech recognition system. Millenium is based on the *Enigma* weighted finite-state transducer (WFST) library, which contains implementations of all standard WFST algorithms, including weighted composition, weighted determinization, weight pushing, and minimization [108]. The *word trace decoder* in Millenium is implemented along the lines suggested by Saon *et al.* [109], and is capable of generating word lattices, which can then be optimized with WFST operations as in [110]; i.e., the raw lattice from the decoder is projected onto the output side to discard all arc information save for the word identities, and then compacted through epsilon removal, determinization, and minimization. In addition to the word trace decoder, Millenium also contains a *state trace decoder*, which maintains the full alignment of acoustic features to states during decoding and lattice generation. This state trace decoder is useful for both speaker adaptation and *hidden Markov model* (HMM) parameter estimation.

The feature extraction of the ASR system was based on cepstral features estimated with the warped MVDR spectral envelope of model order 30. Due to the properties of the warped MVDR, neither the Mel-filterbank nor any other filterbank was needed. The warped MVDR provides an increased resolution in low–frequency regions relative to the conventional Mel-filterbank. The MVDR also models spectral peaks more accurately than spectral valleys, which leads to improved robustness in the presence of noise. Front-end analysis involved extracting 20 cepstral coefficients per frame of speech and performing global cepstral mean subtraction (CMS) with variance normalization. The final features were obtained by concatenating 15 consecutive frames of cepstral features together, then performing a linear discriminant analysis (LDA) to obtain a fea-

ture of length 42. The LDA transformation was followed by a second global CMS, then the global STC transform.

Acoustic models estimated with two different HMM training schemes were used for several decoding passes: conventional maximum likelihood (ML) HMM training described in Section 8.4, and speaker-adapted training under a ML criterion (ML-SAT) [106]; See also Section 8.5.2.

The four decoding passes are performed on the beamformed waveforms. Each pass of decoding used a different acoustic model, language model, or speaker adaptation scheme. For all passes save the first unadapted pass, speaker adaptation parameters were estimated using the word lattices generated during the prior pass, as in [103]. A description of the four decoding passes follows:

1. Decode with the unadapted, conventional ML acoustic model and bigram language model (LM).

2. Estimate vocal tract length normalization (VTLN) [111] parameters and constrained maximum likelihood linear regression parameters (CMLLR) [98] for each speaker, then redecode with the conventional ML acoustic model and bigram LM.

3. Estimate VTLN, CMLLR, and maximum likelihood linear regression (MLLR) [99] parameters for each speaker, then redecode with the conventional model and bigram LM.

4. Estimate VTLN, CMLLR, MLLR parameters for each speaker, then redecode with the ML-SAT model and bigram LM.

9.3 ASR Experiments in the Speech Separation Task

9.3.1 Evaluation of MMI Beamforming Algorithms

Prior to beamforming, the speaker's positions are estimated by the *Orion* source tracking system [66]. In addition to the speaker's position, Orion is also capable of determining when each speaker is active. This information proved very useful segmenting the utterances of each speaker, given that an utterance spoken by one speaker was often much longer than that spoken by the other. In the absence of perfect separation, which we could *not* achieve with the algorithms described here, running the speech recognizer over the entire waveform produced by the beamformer instead of only that portion where a given speaker was actually active would have resulted in significant insertion errors. These insertions would also have proven disastrous for speaker adaptation, as the adaptation data from one speaker would have been contaminated with speech of the other speaker.

Based on the average speaker position estimated for each utterance, utterance-dependent active weight vectors $\mathbf{w}_{a,1}(m)$ and $\mathbf{w}_{a,2}(m)$ at each frequency bin m were estimated. The active weights for each subband were initialized to zero for estimation with the Gaussian pdf. The snapshot covariance matrix $\Sigma_{\mathbf{XX}}(m)$ was estimated for an entire utterance. This matrix was all that was required to estimate $\{\mathbf{w}_{\mathrm{a},i}\}$ for the Gaussian case. For estimation with the super-Gaussian pdfs, the active weights were initialized to their optimal values under the Gaussian assumption. Thereafter iterations of the conjugate gradients algorithm were run on the entire utterance until convergence was achieved.

The training data used for the experiments was taken from the ICSI, NIST, and CMU *meeting corpora*, as well as the *Transenglish Database* (TED) corpus, for a total of 100 hours of training material. In addition to these corpora, approximately 12 hours of speech from the WSJCAM0 corpus [112] were used for HMM training in order to provide coverage of the British accents for the speakers in the MC-WSJ-AV database [3]. The baseline system was fully continuous with

Table 9.1: WERs for every beamforming algorithm after every decoding passes, as well as the close-talking microphone (CTM).

Beamforming Algorithm	Pass (%WER)			
	1	2	3	4
Delay & Sum	85.1	77.6	72.5	70.4
GSS	80.1	65.5	60.1	56.3
MMI: Gaussian	79.7	65.6	57.9	55.2
MMI: Laplace	81.1	67.9	59.3	53.8
MMI: K_0	78.0	62.6	54.1	52.0
MMI: Γ	80.3	63.0	56.2	53.8
CTM	37.1	24.8	23.0	21.6

3,500 codebooks and a total of 180,656 Gaussian components.

Table 9.1 shows the word error rate (WER) for each beamforming algorithm after every decoding pass on the overlapping speech data. After the fourth pass, the delay-and-sum beamformer has the worst recognition performance of 70.4% WER. This is not surprising given that the mixed speech was not well separated by the delay-and-sum beamformer for the reasons mentioned above. The WER achieved by the MMI beamformer with a Gaussian pdf of 55.2% was somewhat better than the 56.3% WER from GSS algorithm, which is what should be expected given the reasoning in Section 7.1.3. The best performance of 52.0% WER was achieved with the MMI beamformer by assuming the subband samples are distributed according to the K_0 pdf.

The WER of 52.0% achieved with the best beamforming algorithm is still more than double the WER of 21.6% achieved with the close-talking microphone (CTM). Hence, there is still a great need for further research to reduce the WER obtained with the separated speech to that obtained with the CTM. A WER of 15–20% is sufficient for a variety of applications including audio indexing; a WER of over 50%, on the other hand, would lead to greatly degraded

performance.

Although the Γ pdf assumption gave the highest log-likelihood, as reported in Table 6.1, the K_0 pdf achieved the best recognition performance. This is because data recorded in real environments contains background noise as well as speech. If the pdf of the noise signal is super-Gaussian, it could conceivably be emphasized by the MMI beamformer with a super-Gaussian pdf assumption. Feature and model adaptation algorithms such as CMLLR and MLLR can, however, robustly estimate parameters to compensate for the background noise. As a result, such an effect is mitigated by the speaker adaptation. From Table 9.1, this is evident from the significant improvement after the second pass when the Γ pdf is used; to wit, the results obtained with the Γ pdf go from being somewhat worse than the Gaussian results after the first unadapted pass to significantly better after the second pass with VTLN and CMLLR adaptation, and remain significantly better after all subsequent adapted passes.

9.3.2 Evaluation of Filter Bank Design Methods

In order to see if separation performance is influenced by filter banks, recognition experiments on speech separated are conducted with MMI beamforming only and investigated four methods :

1. Conventional frequency domain processing based on the FFT [47],
2. Cosine modulated filter bank described by [11, §6], which yields PR under optimal conditions,
3. de Haan filter bank [9], and
4. Nyquist(M) filter banks designed by the proposed algorithms.

Table 9.2 shows the word error rates (WERs) for every filter bank when we set parameters for each filter bank to obtain the best recognition performance. As a baseline, WERs for speech recorded with close-talking microphones are shown in Table 9.2

Table 9.2: WERs without post-filtering for every filter bank design algorithm after every decoding passes.

Filter bank	Pass (%WER)			
	1	2	3	4
FFT	88.5	71.1	58.8	55.5
PR	87.7	65.2	54.0	50.7
de Haan	88.7	68.2	56.1	53.5
Nyquist(M)	88.5	67.0	55.6	52.5
CTM	37.1	24.8	23.0	21.6

MMI beamforming with the PR filter banks provided the best recognition performance when post-filtering was not applied. Although it certainly scaled magnitudes and shifted phases of input subband components, strong aliasing noise was not observed. Hence, we were led to conclude that MMI beamforming could estimate active weight vectors while retaining aliasing cancellation. On the other hand, de Haan filter banks have a total response error which deteriorates the recognition performance. FFT analysis achieved the worst performance of all the subband processing methods.

Table 9.3 depicts the WERs when different parameters for filter banks were set. In all the experiments, the filter lengths are set to twice the number of subbands, $2\times M$. It is clear from Table 9.3 that the proposed filter banks can provide smaller WERs than those of de Haan filter banks. These improvements are mainly because the proposed Nyquist(M) filter banks can have zero total response error. From Table 9.3, one can also see that as the number of subbands M increases, the WER decreases. the MMI beamforming algorithm can strengthen a target wave by using its echoes which are caused by a reflection on a hard surface such as a table. Thus, the larger number of subbands generally leads to the better performance of speech enhancement of the MMI beamformer. In order to enhance this advantageous effect, we need to make the

Table 9.3: WERs without post-filtering for 2 filter bank design algorithms after every decoding passes.

Filter bank	Parameters		Pass (%WER)			
	M	D	1	2	3	4
de Haan	64	32	88.1	69.5	57.9	55.3
	256	128	87.3	69.9	58.2	54.4
	512	256	88.1	68.8	57.5	53.8
	512	128	87.8	68.9	56.6	53.7
	512	64	88.7	68.2	56.1	53.5
Nyquist(M)	64	32	88.6	69.5	57.3	55.2
	256	128	88.0	70.0	57.1	54.5
	512	256	88.0	67.1	55.7	53.4
	512	128	88.5	67.0	55.6	52.5
	512	64	88.1	68.5	57.1	53.9

length of the analysis filter enough long to include such reflected waves in the analysis window. This can be done by increasing the number of subbands.

Contrary to our expectations, Table 9.3 shows that the WER of the Nyquist(M) filter bank does not monotonically decrease with a decreasing decimation factor although the residual aliasing distortion does. We suppose that it is because of the numerical instability discussed in Section 4.5.

Finally, speech recognition experiments are performed on speech enhanced with Zelinski post-filtering [113] and binary masking after MMI beamforming. Table 9.4 shows WERs in those experiments. In this case, the PR property was not kept because of the rapid change of filter weights. The aliasing distortions were not observed when the PR filter banks were used. In contrast, de Haan and the proposed filter banks could suppress such aliasing noise because those filter banks are designed so as to minimize aliasing terms individually.

In Table 9.4, unlike the trend seen in Table 9.3, the larger number of subbands, which leads a higher frequency resolution, does not necessarily provide the better recognition performance. That is perhaps due to inaccurate estimation of noise spectrums in Zelinski post-filtering. In the case of a high frequency resolution, many filter coefficients must be estimated for Zelinski post-filtering, which could lead to robustness problems.

One could find correlation between the WER and the residual aliasing distortion by paying attention to the relationship between Table 9.4 and Figure 4.8. Generally, as the residual aliasing distortion is reduced, the WER becomes smaller. However, this is not always true because there are many other factors impacting recognition performance. For example, although the residual aliasing distortion of the Nyquist(M) filter bank vanishes with an increase in the number of subbands, increasing the number of subbands can lead to robustness problems in estimating the post-filter coefficients; hence, the WER does not monotonically decrease for an increasing number of subbands.

Table 9.4 also shows that the systems with de Haan and Nyquist(M) filter banks can reduce the absolute WER by about 5.0 % compared to those with the PR filter banks. This suggests that the PR filter bank is less suitable for adaptive processing. It is also clear from Table 9.4 that the proposed method achieved a bigger WER reduction than de Haan's algorithm. In particular, the improvement of the recognition performance is significant with $M = 256$. The proposed filter banks achieved the best recognition performance, WER 39.6 % with the number of subbands $M = 512$ and decimation factor $D = 128$. On the other hand, de Haan filter banks provided the same number with $M = 512$ and $D = 64$. Therefore, our method can be thought of as halving the computational cost of that of de Haan.

Table 9.4: WERs with post-filtering for every filter bank design algorithm after every decoding passes.

Filter bank	Parameters		Pass (%WER)			
	M	D	1	2	3	4
PR	64	-	83.7	61.5	47.5	44.7
	512	-	84.6	60.5	47.6	44.4
de Haan	64	32	82.4	59.2	46.2	43.3
	256	128	82.0	60.5	44.7	42.0
	512	256	83.9	59.1	43.2	41.3
	512	128	81.6	58.9	43.2	40.3
	512	64	82.7	57.7	42.7	39.6
Nyquist(M)	64	32	80.7	57.0	44.3	42.0
	256	128	81.0	56.2	41.8	39.8
	512	256	84.1	58.6	43.4	40.6
	512	128	81.8	54.9	42.2	39.6
	512	64	81.4	56.5	42.6	40.3

9.4 ASR Experiments in the Single-Speaker Scenario

9.4.1 Evaluation of Beamforming Algorithms

In addition to the speech separation task, speech recognition experiments in the single-speaker scenario are also conducted in order to investigate performance of MN and MK beamforming.

Thirty hours of American WSJ and the 12 hours of Cambridge WSJ data are used in order to train triphone acoustic models for the experiments in this task. The latter was found to be necessary in order to provide coverage of the British accents. The ASR system used here is fully continuous with 1,743 codebooks and

a total of 67,860 Gaussian components. The four decoding passes are performed on the waveforms obtained with each of the beamforming algorithms described in prior sections. The decoding algorithm in each pass is the same as that described in the previous section except that all passes used the full trigram LM for the 5,000 word WSJ task, which was made possible through the fast-on-the-fly composition algorithm described in [114].

The parameters of the GG pdf for MN beamforming were trained with 43.9 minutes of speech data recorded with the CTM in the SSC development set. The training data set for the GG pdf contains recordings of 5 speakers.

The speaker's position is first estimated with a speaker tracking system [66]. Based on the average speaker position estimated for each utterance, utterance-dependent active weight vectors $\mathbf{w}_a$ were estimated for a source. The active weight vector for each subband was initialized to zero for estimation. Iterations of the conjugate gradients algorithm were run on the entire utterance until the convergence was obtained. After beamforming, Zelinski post-filtering [113] was performed.

Table 9.5 shows the word error rates (WERs) for every beamforming algorithm. As references, WERs in recognition experiments on speech data recorded with the single distant microphone (SDM) and CTM are also given.

It is clear from Table 9.5 that every MN beamforming algorithm can provide better recognition performance than the simple delay-and-sum beamformer (D&S BF) which can be improved by Zelinski post-filtering (D&S BF with PF). It is also clear from Table 9.5 that MN beamforming with the GG pdf assumption which uses the magnitude in calculating the negentropy (MN BF with GG pdf (1)) achieves the best recognition performance. This is due to the fact that the GG pdf models the magnitudes of the subband samples of speech better than the other pdfs in that the shape parameter for each subband is estimated individually from training data.

The recognition performance, however, did not improve for MN beamforming with the GG pdf when the real and imaginary parts of the subband components were assumed to be independent (MN BF with GG pdf (2)). These results

imply that it is better to treat the subband components as spherically-invariant random processes (SIRPs) as in [4, 53] and we are led to conclude that the real and imaginary parts are dependent as mentioned in [52].

Table 9.5 suggests that the Γ pdf assumption (MN BF with Γ pdf) can lead to better noise suppression performance to some extent. The reduction over the D&S BF with the PF case, however, is limited because the Γ pdf cannot model the subband components of speech as precisely as the GG pdf which takes the magnitude as the r.v.

Table 9.5 also shows that MK beamforming (MK BF) can achieve almost the same recognition performance as MN beamforming where one utterance speech data was used for calculating the active weight vectors.

Recognition experiments are also performed on speech enhanced by the MVDR beamformer with Zelinski post-filtering, which is equivalent to the minimum mean-squared error beamformer (MMSE BF) [6, §13.3.5]. Table 9.5 demonstrates that the MVDR beamformer with post-filtering (MMSE BF) provides better recognition performance than D&S BF with PF. The MMSE beamformer would suppress the reflections of the desired signal. On the other hand, as demonstrated in Section 7.2.3, the MN beamforming algorithms can strengthen the target signal by using the reflections solely based on the maximum negentropy criterion. The same thing is applied to MK beamforming. Note that the MVDR beamforming algorithms require speech activity detection in order to avoid signal cancellation. For the adaptation of the MVDR beamformer, we used the first 0.1 and last 0.1 seconds in each utterance, which contain only background noise.

Table 9.5 also shows the recognition results obtained with the generalized eigenvector beamformer (GEV BF) proposed by E. Warsitz et al. [22]. It achieved slightly better recognition performance than the MMSE beamformer. In this task, the transfer function from the sound source to the microphone array changes in time due to movements of the speaker's head. Moreover, it is difficult to determine whether or not the signal observed at any given time contains both speech and noise components in each frequency bin, which is required

to estimate the transfer function. Due to these difficulties, the performance of the GEV beamformer is limited in realistic environments. Once more, in contrast to conventional beamforming methods, the new beamforming algorithms with HOS do not need to detect the start and end points of target speech since the proposed method can suppress noise and reverberation without the signal cancellation problem.

Super-directive beamforming is data-independent and thus does not suffer the signal cancellation. As shown in Table 9.5, the super-directive beamformer with Zelinski post-filtering (SD BF) provides the better recognition performance, 14.1 % WER, than the GEV beamformer. However, since it cannot adapt the beamformer's weights to the specific environment, the recognition performance is limited and worse than those obtained with maximum negentropy and maximum kurtosis beamforming in the fourth pass.

It is worth noting that the best result of 13.2% in Table 9.5 is significantly less than half the word error rate reported elsewhere in the literature on this distant ASR task [3].

We implemented each beamforming algorithm in C/C++ and python. The computational cost of the MN beamforming algorithm (MN BF with GG pdf (1)) is approximately 2.6 times as much as that of the MMSE beamformer per frame on a machine with an Intel Core 2 DUO E6750/2.66GHz processor and 3.36 GB RAM.

9.4.2 Dependence of WER on Regularization Term

We also examined the effect of the regularization term in equation (7.18). Table 9.6 shows WERs as a function of the regularization parameter α, where we used the MN beamforming algorithm with the GG pdf of the magnitude r.v. We can see from the table that the regularization parameter $\alpha = 10^{-2}$ provided the lowest word error rate, although the impact of different values of α on recognition performance was slight. The regularization parameter α could be interpreted as an indicator of the sufficiency of the input data in estimating

Table 9.5: WERs for each beamforming algorithm after every decoding pass.

Beamforming Algorithm	Pass (%WER)			
	1	2	3	4
D&S BF	80.1	39.9	21.5	17.8
D&S BF with PF	79.0	38.1	20.2	16.5
MMSE BF	78.6	35.4	18.8	14.8
GEV BF	78.7	35.5	18.6	14.5
SD BF	71.4	31.9	16.6	14.1
MN BF with Gamma pdf	75.6	34.9	19.8	15.8
MN BF with GG pdf (1)	75.1	32.7	16.5	13.2
MN BF with GG pdf (2)	79.0	37.2	20.0	16.7
MK BF	76.6	33.5	17.2	13.6
SDM	87.0	57.1	32.8	28.0
CTM	52.9	21.5	9.8	6.7

Note that WERs of 12.3% for CTM and 66.5% for SDM were achieved with the adaption techniques described by Lincoln *et al* [3], who also reported that their beamforming algorithm achieved a WER of 28.1%. To the best of our knowledge, no other error rates at present have been reported in the literature on this ASR task.

the active weight vector. Thus, the requirement of a small α may imply that the input data are not sufficiently reliable to completely determine the active weight vector due to, for example, steering errors.

Fig. 9.4.2 shows the waveforms of (a) noisy speech recorded with the far-field microphone, (b) the speech signal enhanced with MMSE BF, (c) the speech signal processed with the proposed method (MN BF with GG pdf (1)), and speech recorded with the CTM. Informal listening tests confirmed that the annoying distortions were not particularly observed in speech enhanced by MN beamforming algorithm.

Table 9.6: WERs against the regularization parameter α.

Regularization parameter α	Pass (%WER)			
	1	2	3	4
$\alpha = 0.0$	72.7	31.9	16.4	13.7
$\alpha = 10^{-3}$	73.9	32.2	16.6	13.6
$\alpha = 10^{-2}$	75.1	32.7	16.5	13.2
$\alpha = 10^{-1}$	76.2	32.5	17.5	13.5

9.4.3 Influence of Gradient Algorithm in MK Beamforming

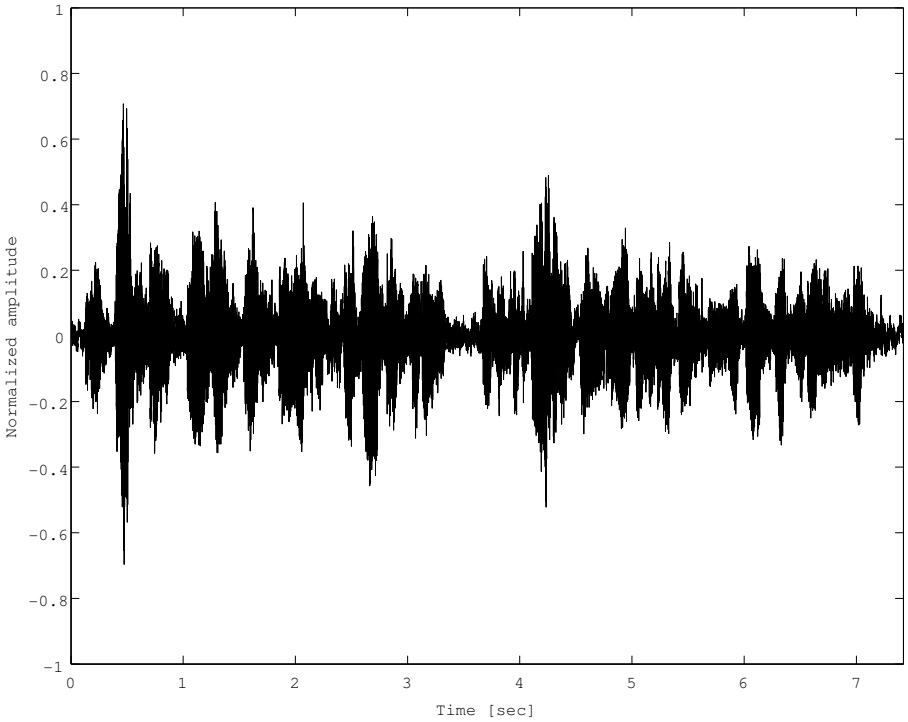
In MK beamforming, the estimation of the active weight vectors is greatly influenced by outliers. We observed that the active weight vectors became extremely large in the case that the amount of data for the adaptation was insufficient. It could not be avoided even if the regularization weight α was increased. The author, therefore, put a constraint on the active weight vector: $\|\mathbf{w}_{\mathrm{a}}\| = 1$ if $\|\mathbf{w}_{\mathrm{a}}\| \geq 1$. The active weight vector is projected on the unit circle after every step if the vector norm exceeds unity. Such a projection procedure could destroy the convergence property of the Polak-Ribiere conjugate gradient algorithm because it uses the sequence of search directions in order to approximate the curvature of the objective function around an evaluation point. Hence, we implemented the projection procedure in the steepest descent algorithm [1].

Table 9.7 shows the WERs for the amount of data for each beamforming algorithm. It is clear from Table 9.7 that MN beamforming can provide good recognition performance even if very little adaptation data are available. That is mainly because the speech models trained with sufficient data are used for the calculation of negentropy. Such prior speech models make MN-beamforming robust for outliners. It is also clear from Table 9.7 that good recognition performance is not obtained by MK beamforming with the Polak-Ribiere conjugate gradient algorithm because the active weight vector $\mathbf{w}_{\mathrm{a}}$ grows excessively large.

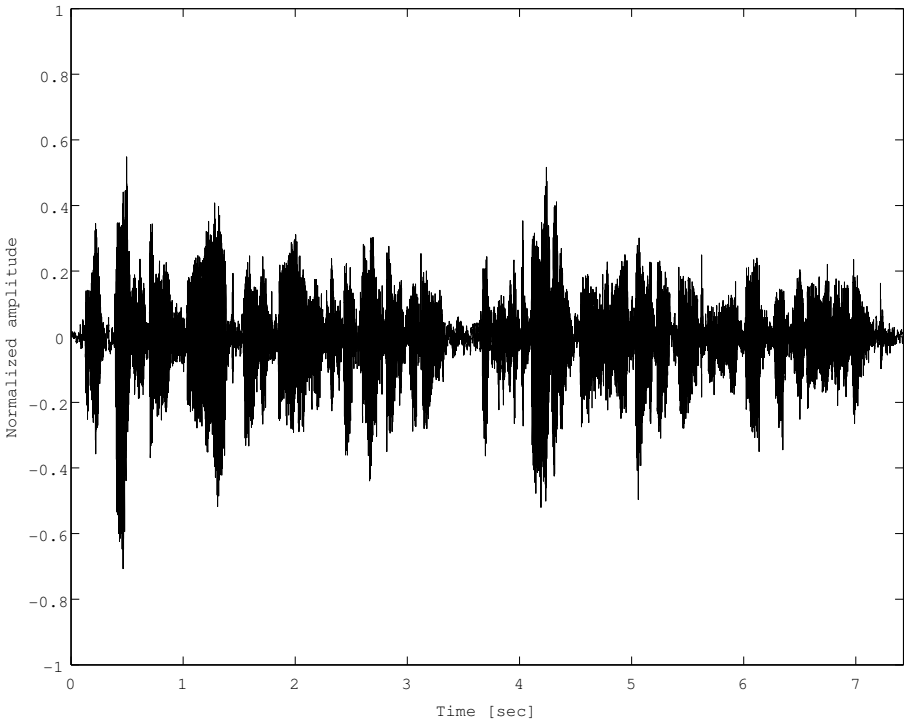
Table 9.7: WERs for the number of frames used in adaptation for each beamforming algorithm .

Beamforming Algorithm	milli-second	Pass (%WER)			
		1	2	3	4
MN BF with Polak-Ribiere conjugate gradient	192	73.2	38.2	19.2	15.3
	384	75.7	35.0	18.9	15.4
	576	75.8	33.5	17.8	14.5
	1 utt.	75.1	32.7	16.5	13.2
MK BF with the Polak-Ribiere conjugate gradient	192	94.1	90.1	81.3	-
	384	93.3	87.2	77.0	74.7
	576	87.3	79.3	52.9	50.0
	1 utt.	76.6	33.5	17.2	13.6
MK BF with the steepest descent with the unit NC	192	80.2	41.7	21.9	18.6
	384	82.0	44.0	21.5	18.5
	576	80.1	41.1	20.5	17.5
	1 utt.	75.7	32.8	17.3	13.7

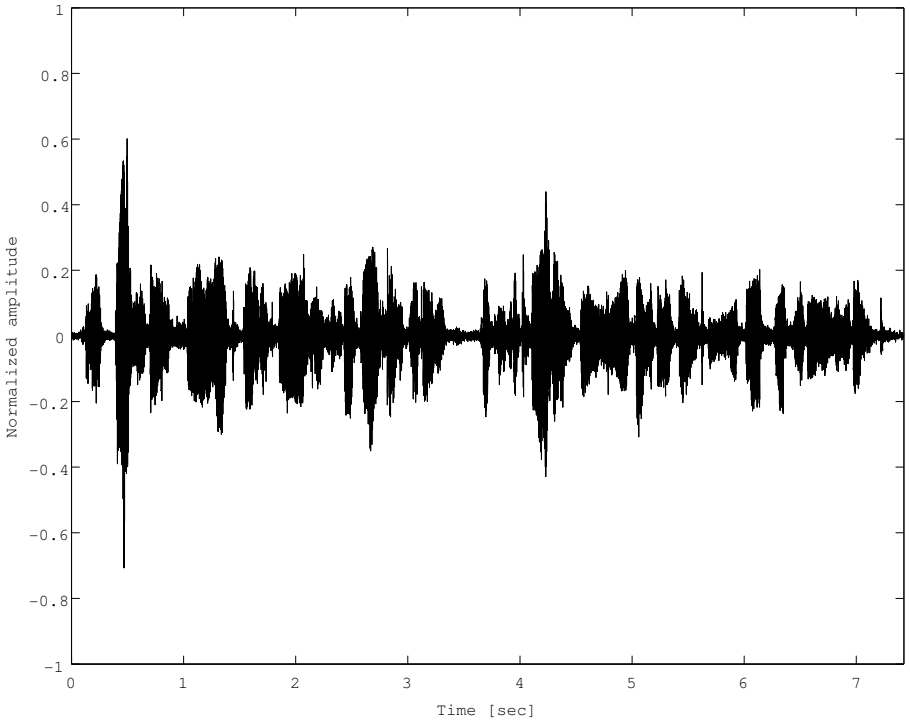
Table 9.7 suggests that such a problem can be alleviated by projecting the active weight vector into the unit circle.



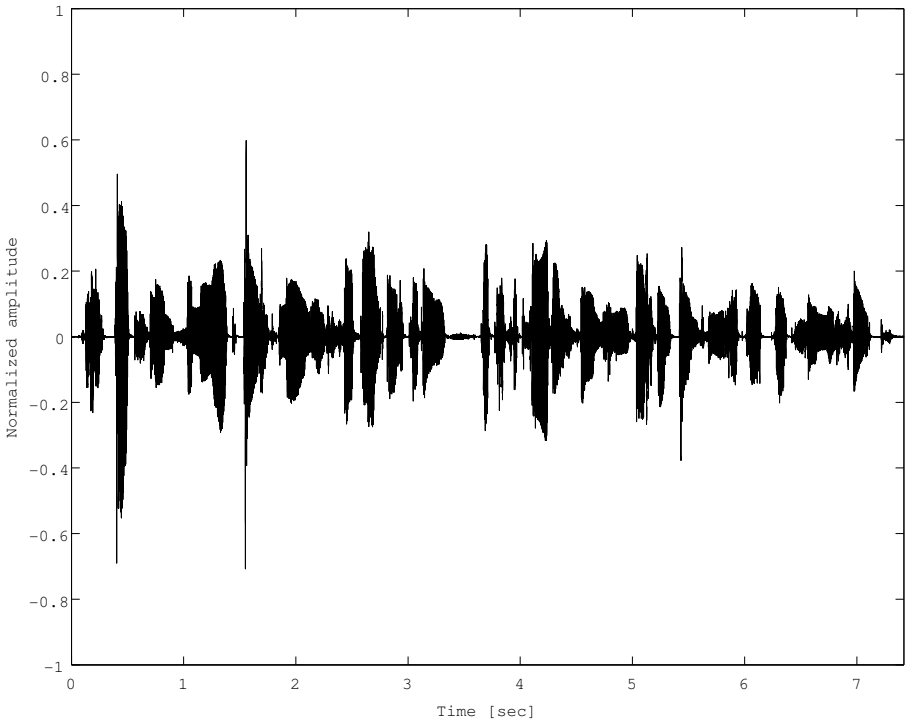
(a) speech recorded with the microphone array.



(b) speech processed with MMSE BF.



(c) speech processed with MN BF with GG pdf (1).



(d) speech recorded with the CTM.

Figure 9.2: normalized speech wave forms

Chapter 10

Conclusions

The distant automatic speech recognition in the *real* meeting room has been studied in this work. In particular, the author has addressed two main issues:

- separating speech of a target speaker from overlapping speech, and
- enhancing speech in the single speaker scenario.

In contrast to the majority of literature, the *real* speech data captured with the *real* sensors is used in the distant speech recognition experiments. The speech data also contains variations in characteristics of speakers as well as various phonemes.

Beamforming is one of the most important techniques for the distant automatic speech recognition. This thesis has proposed three novel beamforming algorithms with generalized sidelobe canceller (GSC) configuration, minimum mutual information (MMI) beamforming, maximum negentropy (MN) beamforming and maximum kurtosis (MK) beamforming. The new beamformers proposed here estimate the active weight vectors with the criteria based on higher order statistics (HOS) subject to the distortionless constraint for the look direction.

MMI beamforming estimates the active weight vectors so as to minimize mutual information of outputs of two GSC beamformers. This technique has

been applied to the speech separation task and its effectiveness has been demonstrated through the speech recognition experiments. However, we can use the MMI beamforming algorithm only for the situation where there are multiple directional sound sources. In other words, it cannot be applied to the case that a single speaker is only active.

In contrast, MN and MK beamforming can be available for the single speaker scenario. The basic idea of both algorithms is to maximize the degree of super-Gaussianity of the distributions of the beamformer's outputs. The difference between the algorithms proposed here is just the criterion of measuring the degree of super-Gaussianity. It has been shown in the experiments that the negentropy criterion is more robust than the empirical kurtosis measure although prior knowledge for the distribution of clean speech is required for calculating negentropy.

It has been shown in the simulations that the new beamforming algorithms can strengthen the target signal by manipulating the reflected wave. It has been also observed in the speech recognition experiments that all the beamforming algorithms proposed here can continue updating active weight vectors without degrading the recognition performance while the target signals are active, which suggests that the new beamforming algorithms are free from signal cancellation problems encountered in the conventional beamforming with second order statistics (SOS).

The permutation and scaling ambiguity problems seen in the blind source separation (BSS) can be also avoided by the proposed methods owing to the distortionless constraint for the look direction. Moreover, the optimization of the active weight vectors with the distortionless constraint enables the performance of speech enhancement to be higher than that of the delay-and-sum beamformer at least. In contrast, any BSS algorithm depends on initial weight values and has instability of finding the weights of un-mixing matrices. As a consequence, the separation performance could be worse than delay-and-sum beamformer. Notice that the BSS techniques cannot applied to the single speaker stationary condition.

It could be considered that beamforming algorithms with HOS take the best parts of the conventional beamformers and BSS techniques. Finally, it is clear from the experimental results that the new beamforming techniques can provide the better recognition performance than that of the SOS-based beamformers.

Contributions of this thesis are summarized as follows:

- Filter bank design for beamforming. The undesired aliasing effects can be alleviated in the case that the property of the perfect reconstruction is destroyed by arbitrary scaling of magnitude and phase shift [30, 31].

- Minimum mutual information (MMI) beamforming. It can separate sound sources without the signal cancellation problem encountered in the conventional beamforming techniques. Moreover, it is free from any problem seen in the BSS techniques [4].

- Maximum negentropy (MN) beamforming. Distant speech can be enhanced by this techniques without the signal cancellation problem [32].

- Maximum kurtosis (MK) beamforming. This beamforming algorithm has the same advantage as MN beamforming. Furthermore, it can be simply implemented since the prior speech model is not required. However, the MK beamforming algorithm is influenced by outliers [33].

Chapter 11

My publications related to this work

Journal Papers

- Kenichi Kumatani, John McDonough, Barbara Rauch, Dietrich Klakow, Philip N. Garner and Weifeng Li, "Adaptive Beamforming with a Maximum Negentropy Criterion," IEEE Trans. Audio, Speech and Language Processing, Vol. 17, pp. 994-1008, July 2009.

- Kenichi Kumatani, Tobias Gehrig, Uwe Mayer, Emilian Stoimenov, John McDonough and Matthias Wölfel, "Adaptive Beamforming with a Minimum Mutual Information Criterion," IEEE Trans. Audio, Speech and Language Processing, Vol. 15, pp. 2527-2541, November, 2007.

International Conference Papers

- Kenichi Kumatani, John McDonough, Barbara Rauch, Philip N. Garner, Weifeng Li and John Dines, "Maximum kurtosis beamforming with the generalized sidelobe canceller," in Proc. Interspeech-2008, Brisbane, Aus-

tralia, Septempber 2008.

- Weifeng Li, Kenichi Kumatani, John Dines, MMathew Magimai Doss and Herve Bourlard, "A Neural Network based Regression Approach for Recogninizing Simultaneous Speech," in Proc. the Joint Workshop on Machine Learning and Multi-modal Interaction (MLMI2008), Utrecht, Netherlands, September 2008.

- Kenichi Kumatani, John McDonough, Dietrich Klakow, Philip N. Garner and Weifeng Li "Adaptive Beamforming with a Maximum Negentropy Criterion," for the Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA), Trento, Italy, May 2008.

- Kenichi Kumatani, John McDonough, Stefan Schacht, Dietrich Klakow, Philip Garner and Weifeng Li, "Filter Bank Design Based on Minimization of Individual Aliasing Terms for Minimum Mutual Information Subband Adaptive Beamforming," in Proc. ICASSP, Las Vegas, Nevada, U.S.A, March - April, 2008.

- Kenichi Kumatani, Uwe Mayer, Tobias Gehrig, Emilian Stoimenov, John McDonough and Matthias Wölfel, "Minimum Mutual Information Beamforming for Simultaneous Active Speakers," in Proc. ASRU, Kyoto, Japan, December, 2007.

- Kenichi Kumatani and Rainer Stiefelhagen, "State Synchronous Modeling on Phone Boundary for Audio Visual Speech Recognition and Application to Multi-view face images," in Proc. ICASSP ,Honolulu, Hawaii, U.S.A, April 2007.

- John McDonough, Kenichi Kumatani, Tobias Gehrig, Emilian Stoimenov, UweMayer, Stefan Schacht, Matthias Wölfel, and Dietrich Klakow, "To separate speech! a system for recognizing simultaneous speech," in Proc. 4th Joint Workshop on Machine Learning and Multimodal Interaction (MLMI), Brno, Czech Republic, June 2007.

Bibliography

[1] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, 2000.

[2] S. Araki, S. Makino, Y. Hinamoto, R. Mukai, T. Nishikawa, and H. Saruwatari, "Equivalence between frequency-domain blind source separation and frequency-domain adaptive beamforming for convolutive mixtures," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 1157–1166, 2003.

[3] M. Lincoln, I. McCowan, I. Vepa, and H. K. Maganti, "The multi-channel Wall Street Journal audio visual corpus (mc-wsj-av): Specification and initial experiments," in *Proc. IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2005, pp. 357–362.

[4] K. Kumatani, T. Gehrig, U. Mayer, E. Stoimenov, J. McDonough, and M. Wölfel, "Adaptive beamforming with a minimum mutual information criterion," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 2527–2541, 2007.

[5] H. K. Maganti, D. Gatica-Perez, and I. McCowan, "Speech enhancement and recognition in meetings with an audio-visual sensor array," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 2257–2269, 2007.

[6] M. Wölfel and J. McDonough, *Distant Speech Recognition*. Wiley, 2009.

[7] M. Wölfel, "Enhanced speech features by single-channel joint compensation of noise and reverberation," *IEEE Transactions Audio, Speech and Language Processing*, vol. 17, pp. 312–323, 2009.

[8] S. Haykin, *Adaptive filter theory.* Upper Saddle River, New Jersey: Prentice Hall, 2002.

[9] J. M. de Haan, N. Grbic, I. Claesson, and S. E. Nordholm, "Filter bank design for subband adaptive microphone arrays," *IEEE Transactions Speech Audio Proc.*, vol. 11, no. 1, pp. 14–23, Jan. 2003.

[10] J. M. de Haan, "Filter bank design for subband adaptive filtering," Ph.D. dissertation, Karlskrona. Blekinge Institute of Technology, 2001.

[11] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs: Prentice Hall, 1993.

[12] B. Widrow, K. M. Duvall, R. P. Gooch, and W. C. Newman, "Signal cancellation phenomena in adaptive antennas: Causes and cures," *IEEE Transactions on Antennas and Propagation*, vol. AP-30, pp. 469–478, 1982.

[13] S. Nordholm, I. Claesson, and B. Bengtsson, "Adaptive array noise suppression of handsfree speaker input in cars," *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 514–518, 1993.

[14] W. Herbordt and W. Kellermann, "Adaptive beamforming for audio signal acquisition," in *Adaptive Signal Processing: Applications to Real-World Problems*, J. Benesty and Y. Huang, Eds. Berlin, Germany: Springer, 2003, pp. 155–194.

[15] I. Cohen, S. Gannot, and B. Berdugo, "An integrated real-time beamforming and postfiltering system for nonstationary noise environments," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 1064–1073, 2003.

[16] I. Claesson and S. Nordholm, "A spatial filtering approach to robust adaptive beaming," *IEEE Transactions on Antennas and Propagation*, vol. 19, pp. 1093–1096, 1992.

[17] S. Nordebo, I. Claesson, and S. Nordholm, "Adaptive beamforming: spatial filter designed blocking matrix," *IEEE Journal of Oceanic Engineering*, vol. 19, pp. 583–590, 1994.

[18] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Transactions on Signal Processing*, vol. 47, pp. 2677–2684, 1999.

[19] N. Grbić, "Optimal and adaptive subband beamforming," Ph.D. dissertation, Blekinge Institute of Technology, 2001.

[20] W. Herbordt and W. Kellermann, "Frequency-domain integration of acoustic echo cancellation and a generalized sidelobe canceller with improved robustness," *European Transactions on Telecommunications (ETT)*, vol. 13, pp. 123–132, 2002.

[21] W. Herbordt, H. Buchner, S. Nakamura, and W. Kellermann, "Multichannel bin-wise robust frequency-domain adaptive filtering and its application to adaptive beamforming," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1340–1351, 2007.

[22] E. Warsitz, A. Krueger, and R. Häb-Umbach, "Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, U.S.A, 2008.

[23] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel wiener filter for

robust noise reduction," *Speech Communication, special issue on Speech Enhancement*, vol. 49, pp. 636–656, 2007.

[24] S. Gannot, David, Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, pp. 1614–1626, 2001.

[25] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Transactions Speech and Audio Processing*, vol. 12, pp. 561–571, 2004.

[26] H. Buchner, R. Aichner, and W. Kellermann, "Blind source seperation for convolutive mixtures: A unified treatment," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y. Huang and J. Benesty, Eds. Boston: Kluwer Academic, 2004, pp. 255–289.

[27] P. Smaragdis, "Efficient blind separation of convolved sound mixtures," in *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, U.S.A, 1997.

[28] H. Saruwatari, T. Kawamura, and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ica and beamforming," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 666–678, 2006.

[29] S. Y. Low, S. Nordholm, and R. Togneri, "Convolutive blind signal separation with post-processing," *IEEE Transactions on Speech and Audio Processing*, vol. 12, pp. 539–548, 2004.

[30] K. Kumatani, J. McDonough, S. Schacht, D. Klakow, P. N. Garner, and W. Li, "Filter bank design based on minimization of individual aliasing terms for minimum mutual information subband adaptive beamforming," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, Nevada, U.S.A, 2008.

[31] J. McDonough, K. Kumatani, T. Gehrig, E. Stoimenov, U. Mayer, S. Schacht, M. Wölfel, and D. Klakow, "To separate speech! a system for recognizing simultaneous speech," in *Proc. MLMI*, 2007.

[32] K. Kumatani, J. McDonough, B. Rauch, D. Klakow, P. N. Garner, and W. Li, "Beamforming with a maximum negentropy criterion," *IEEE Transactions Audio, Speech and Language Processing*, vol. 17, pp. 994–1008, 2009.

[33] K. Kumatani, J. McDonough, B. Rauch, P. N. Garner, W. Li, and J. Dines, "Maximum kurtosis beamforming with the generalized sidelobe canceller," in *Proc. Interspeech*, Brisbane, Australia, 2008.

[34] L. Brillouin, *Wave Propagation and Group Velocity*. New York and London: Academic Press, 1960.

[35] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Simon & Schuster, 1992.

[36] I. McCowan, D. Moore, and S. Sridharan, "Speech enhancement using near-field superdirectivity with an adaptive sidelobe canceler and post-filter," in *In Proceedings of the 2000 Australian International Conference on Speech Science and Technology*, Canberra, Australia, 2000, pp. 268–273.

[37] I. McCowan, C. Marro, and L. Mauuary, "Robust speech recognition using near-field superdirective beamforming with post-filtering," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 2000, pp. 1723–17 263.

[38] J. G. Ryan, "Criterion for the minimum source distance at which plane-wave beamforming can be applied," *Journal of the Acoustical Society of America*, vol. 108, pp. 595–598, 1998.

[39] H. L. Van Trees, *Optimum Array Processing*. New York: Wiley-Interscience, 2002.

[40] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, pp. 14–37, 1992.

[41] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, New Jersey: Prentice-Hall, 1989.

[42] L. Pelkowitz, "Frequency domain analysis of wrap-around error in fast convolution algorithm," *IEEE Transactions Audio, Speech and Language Processing*, vol. ASSP-29, pp. 413–422, 1981.

[43] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. am T. Vetterling, *Numerical Recipes in C.* Cambridge University Press, 1992.

[44] M. Brandstein and D. Ward, Eds., *Microphone Arrays.* Heidelberg, Germany: Springer Verlag, 2001.

[45] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: The Johns Hopkins University Press, 1996.

[46] E. Warsitz and R. Häb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1529–1539, 2007.

[47] F. Asano, S. Ikeda, M. Ogawa, H. Aso, and N. Kitawaki, "Combined approach of array processing and independent component analysis for blind separation of acoustic signals," *IEEE Transactions Speech and Audio Processing*, vol. 11, no. 3, pp. 204–215, May 2003.

[48] H. Saruwatari, T. KAWAMURA, and K. SHIKANO, "Blind source separation based on fast-convergence algorithm using ica and array signal processing," *IEEE Transactions Audio, Speech, Lang. Proc*, vol. 14, pp. 666–678, 2001.

[49] R. G. Gallager, *Information Theory and Reliable Communication.* New York: John Wiley & Sons, 1968.

[50] F. D. Neeser and J. L. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions Info. Theory*, vol. 39, no. 4, pp. 1293–1302, July 1993.

[51] R. Martin, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors," *IEEE Transactions Speech and Audio Processing*, vol. 13, no. 5, pp. 845–856, Sept. 2005.

[52] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1741–1752, 2007.

[53] H. Brehm and W. Stammler, "Description and generation of spherically invariant speech-model signals," *Signal Processing*, vol. 12, pp. 119–141, 1987.

[54] S. Wolfram, *The Mathematica Book*, 3rd ed. Cambridge: Cambridge University Press, 1996.

[55] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. New York: John Wiley & Sons, 1984.

[56] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1990.

[57] K. Kokkinakis and A. K. Nandi, "Exponent parameter estimation for generalized gaussian probability density functions with application to speech modeling," *Signal Processing*, vol. 85, pp. 1852–1858, 2005.

[58] M. K. Varanasi and B. Aazhang, "Parametric generalized gaussian density estimation," *J. Acoust. Soc. Am.*, vol. 86, pp. 1404–1415, 1989.

[59] M. K. Varanasi, "Parameter estimation for the generalized gaussian noise model," Ph.D. dissertation, Rice University, 1987.

[60] D. P. Bertsekas, *Nonlinear Programming*. Belmont, Massachusetts: Athena Scientific, 1995.

[61] L. C. Parra and C. V. Alvino, "Geometric source separation: Merging convolutive source separation with geometric beamforming," *IEEE Transactions Speech Audio Processing*, vol. 10, no. 6, pp. 352–362, September 2002.

[62] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acou stics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, April 1979.

[63] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1995.

[64] E. A. P. Habets, "Single- and multi microphone speech dereverberation using spectral enhancement," Ph.D. dissertation, Eindhoven University of Technology, 2007.

[65] B. W. Gillespie, H. S. Malvar, and D. A. F. Floêncio, "Speech dereverberation via maximum-kurtosis subband adaptive filtering," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, U.S.A, 2001.

[66] T. Gehrig, U. Klee, J. McDonough, S. Ikbal, M. Wölfel, and C. Fügen, "Tracking and beamforming for multiple simultaneous speakers with probabilistic data association filters," in *Proc. Interspeech*, 2006, pp. 2594–2597.

[67] R. Bellman, *Dynamic Programming*. Dover Publications, 2003.

[68] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustical Society of America*, vol. 87, pp. 1738–1752, 1990.

[69] M. Wölfel, "Robust automatic transcription of lectures," Ph.D. dissertation, Universität Karlsruhe, 2009.

[70] M. N. Murthiand and B. D. Rao, "All–pole modeling of voiced speech base on the minimum variance distortionless response spectrum," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 221–239, 2000.

[71] M. N. Murthi and B. D. Rao, "Minimum variance distortionless response (mvdr) modeling of voiced speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, Germany, 1997, pp. 1687–1690.

[72] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation: Review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.

[73] J. McDonough, W. Byrne, and X. Luo, "Speaker normalization with all-pass transforms," in *Proc. ICSLP*, Sydney, Australia, 1998.

[74] J. McDonough, "Speaker compensation with all-pass transforms," Ph.D. dissertation, The Johns Hopkins University, 2000.

[75] B. Musicus, "Fast mlm power spectrum estimation from uniformly spaced correlations," *IEEE Transactions on ASSP*, vol. 33, pp. 1333–1335, 1985.

[76] J. Capon, "High–resolution frequency–wavenumber spectrum analysis," *Proc. of the IEEE*, vol. 57, pp. 1408–1418, August 1969.

[77] R. Lacoss, "Data adaptive spectral analysis methods," *Geophysics*, vol. 36, no. 4, pp. 661–675, 1971.

[78] S. Haykin, *Adaptive Filter Theory*, 4th ed. New York: Prentice Hall, 2002.

[79] M. Matsumoto, Y. Nakatoh, and Y. Furuhata, "An efficient mel-lpc analysis method for speech recognition," *Proc. of ICSLP*, pp. 1051–1054, 1998.

[80] J. Barker and M. Cooke, "Modelling the recognition of spectrally reduced speech," in *Proc.Eurospeech1997*, 1997, pp. 2127–2130.

[81] R. Häb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, San Francisco, U.S.A, 1992, pp. 13–16.

[82] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen, "Maximum likelihood discriminant feature spaces," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP*, Istanbul, Turkey, 2000, pp. 1129–1132.

[83] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–297, 1998.

[84] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*, http://htk.eng.cam.ac.uk/docs/docs.shtml, 2006.

[85] J. G. Fiscus, J. Ajot, M. Michel, and J. S. Garofolo, "The rich transcription 2006 spring meeting recognition evaluation," in *Proc. MLMI*, Washington DC, 2006, pp. 309–322.

[86] C. Fügen, S. Ikbal, F. Kraft, K. Kumatani, K. Laskowski, J. W. McDonough, M. Ostendorf, S. Stüker, and M. Wölfel, "The ISL RT-06S speech-to-text system," in *Proc. MLMI*, Washington DC, 2006, pp. 407–418.

[87] J. Huang, M. Westphal, S. Chen, O. Siohan, D. Povey, V. Libal, A. Soneiro, H. Schulz, T. Ross, and G. Potamianos, "The IBM rich transcription 2006 speech-to-text systems for lecture meetings," in *Proc. MLMI*, Washington DC, 2006, pp. 432–443.

[88] A. Janin, A. Stolcke, X. Anguera, K. Boakye, O. Çetin, J. Frankel, and J. Zheng, "The ICSI-SRI spring 2006 meeting recognition system," in *Proc. MLMI*, Washington DC, 2006, pp. 444–456.

[89] L. Lamel, E. Bilinski, G. Adda, J.-L. Gauvain, and H. Schwenk, "The LIMSI RT06s lecture transcription system," in *Proc. MLMI*, Washington DC, 2006, pp. 457–468.

[90] J. Huang, E. Marcheret, K. Visweswariah, V. Libal, and G. Potamianos, "The IBM rich transcription 2007 speech-to-text systems for lecture meetings," in *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007*, Baltimore, MD, 2007, pp. 429–441.

[91] S. Nakamura, K. Kumatani, and S. Tamura, "Robust bi-modal speech recognition based on state synchronous modeling and stream weight optimization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando, Florida, U.S.A, 2002, pp. 309–312.

[92] K. Kumatani and S. Nakamura, "Audio-visual speech recognition based on optimized product hmms and gmm based-mce-gpd stream weight estimation," *IEICE Transactions on Information and Systems*, vol. E86-D, pp. 454–463, 2003.

[93] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[94] D. Pye and P. Woodland, "Experiments in speaker normalisation and adaptation for large vocabulary speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Munich, Germany, 1997, pp. 1047–1050.

[95] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," Ph.D. dissertation, Carnegie Mellon University, 1990.

[96] D. Y. Kim, S. Umesh, M. J. F. Gales, T. Hain, and P. Woodland, "Using vtln for broadcast news transcription," in *Proc. ICSLP*, Jeju, Korea, 2004, pp. 1953–1956.

[97] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, pp. 195–304, 2008.

[98] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, 1998.

[99] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, April 1995.

[100] L. R. Neumeyer, A.Sankar, and V. V. Digalakis, "comparative study of speaker adaptation techniques," in *Proc. Eurospeech*, Madrid, Spain, pp. 1127–1130.

[101] V. Digalakis, H. Collier, S. Berkowitz, A. Corduneanu, E. Bocchieri, A. Kannan, C. Boulis, S. Khudanpur, W. Byrne, and A. Sankar, "Rapid speech recognizer adaptation to new speakers," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Washington, DC, USA, 1999, pp. 765–768.

[102] P. Woodland, D. Pye, and M. J. F. Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *Proc. ICSLP1996*, Philadelphia, pp. 1133–1136.

[103] L. Uebel and P. Woodland, "Improvements in linear transform based speaker adaptation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[104] W. Chou, "Maximum a posteriori linear regression with elliptically symmetric matrix priors," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 1–4.

[105] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalisation and speaker adaptation in hmm estimation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 367–376, 2005.

[106] T. Anastasakos, J. McDonough, R. Schwarz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP*, 1996, pp. 1137–1140.

[107] J. Deller, J. Hansen, and J. Proakis, *Discrete-Time Processing of Speech Signals*. New York: Macmillan Publishing, 1993.

[108] M. Mohri and M. Riley, "Network optimizations for large vocabulary speech recognition," *Speech Communication*, vol. 28, no. 1, pp. 1–12, 1999.

[109] G. Saon, D. Povey, and G. Zweig, "Anatomy of an extremely fast LVCSR decoder," in *Proc. Interspeech*, Lisbon, Portugal, 2005, pp. 549–552.

[110] A. Ljolje, F. Pereira, and M. Riley, "Efficient general lattice generation and rescoring," in *Proc. Eurospeech*, Budapest, Hungary, 1999, pp. 1251–1254.

[111] L. Welling, H. Ney, and S. Kanthak, "Speaker adaptive modeling by vocal tract normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 415–426, 2002.

[112] J. Fransen, D. Pye, T. Robinson, P. Woodland, and S. Y. g, "WSJCAM0 corpus and recording description," Cambridge University Engineering Department (CUED), Speech Group, Trumpington Street, Cambridge CB2 1PZ, UK, Tech. Rep. CUED/F-INFENG/TR.192, Sept. 1994.

[113] C. Marro, Y. Mahieux, and K. U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 240–259, 1998.

[114] J. McDonough, E. Stoimenov, and D. Klakow, "An algorithm for fast composition of weighted finite-state transducers," in *Proc. IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2007.

[115] Y. L. Luke, *The Special Functions and their Approximations*. New York: Academic Press, 1969.

[116] W. Rudin, *Real and Complex Analysis*. Boston, Massachusetts: McGraw-Hill, 1987.

[117] R. E. Greene and S. G. Krantz, *Function Theory of One Complex Variable*. New York: Wiley-Interscience, 1997.

[118] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980.

[119] J. McDonough and K. Kumatani, "Minimum mutual information beamforming," Interactive Systems Lab, Universität Karlsruhe, Tech. Rep. 107, August 2006.

[120] Welcome to swig. [Online]. Available: http://www.swig.org/

[121] Python documentation. [Online]. Available: http://docs.python.org/

Appendix A

Super-Gaussian Distributions

As explained in Brehm and Stammler [53], it is useful to assume that the Laplace, K_0, and Γ pdfs belong to the class of *spherically invariant random processes* (SIRPs) for two principal reasons. Firstly, this implies that multivariates of all orders can be derived from the univariate pdf as soon as the covariance matrix is known; this is most readily accomplished using the formalism of the Meijer G-function. Secondly, such variants can be extended to the case of complex r.v.s, which is essential for our current development. In this appendix, we provide a brief exposition of the Meijer G-function and its use in deriving multivariate super-Gaussian pdfs for complex r.v.s.

A.1 Meijer G-functions

In this section, we very briefly introduce the notation of the Meijer G-function, along with the most important relations required to use G-functions to model super-Gaussian pdfs.

To denote the Meijer G-function, we will use one of the following equivalent

forms

$$
\begin{aligned}
&G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_p\\ b_q\end{matrix}\right.\right)\\
&\quad= G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_1,\ldots,a_p\\ b_1,\ldots,b_q\end{matrix}\right.\right)\\
&\quad= G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_1,\ldots,a_n & | & a_{n+1},\ldots,a_p\\ b_1,\ldots,b_m & | & b_{m+1},\ldots,b_q\end{matrix}\right.\right).
\end{aligned}
$$

The G-function is defined by the contour integral

$$
\begin{aligned}
G_{p\,q}^{m\,n}\left(x\left|\begin{matrix}a_1,\ldots,a_p\\ b_1,\ldots,b_q\end{matrix}\right.\right) &= \frac{1}{2\pi i}\oint_{\Gamma_L} x^s ds \\
&\times \frac{\prod_{j=1}^{m}\Gamma(b_j-s)\prod_{j=1}^{n}\Gamma(1-a_j+s)}{\prod_{j=n+1}^{p}\Gamma(a_j-s)\prod_{j=m+1}^{q}\Gamma(1-b_j+s)}
\end{aligned}
\tag{A.1}
$$

where $\Gamma(z)$ is the Gamma function and Γ_L is a contour of integration defined as in [53]. The definition (A.1) implies

$$
G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_p\\ b_q\end{matrix}\right.\right) = z^{-u} G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_p+u\\ b_q+u\end{matrix}\right.\right) \tag{A.2}
$$

where $a_p + u$ and $b_q + u$ indicate that u is to be added to all $a_1, \ldots, a_p$ and all $b_1, \ldots, b_q$, respectively. To determine the normalizing constants of the several pdfs generated from the Meijer G-function, it will be useful to apply the Mellin transform

$$
M\{f(x); z\} = \int_0^\infty dx\, x^{z-1} f(x). \tag{A.3}
$$

Under suitable conditions [53], the Mellin transform of a Meijer G-function can

be expressed as

$$M\left\{G_{p\,q}^{m\,n}\left(z\left|\begin{matrix}a_p\\ b_q\end{matrix}\right.\right);z\right\} = \frac{\prod_{i=1}^{m}\Gamma(b_i+z)\prod_{i=1}^{n}\Gamma(1-a_i-z)}{\prod_{i=1}^{m}\Gamma(1-b_i-z)\prod_{i=1}^{n}\Gamma(a_i+z)}. \tag{A.4}$$

A.2 Spherically Invariant Random Processes

We now show how G-functions can be used to represent SIRPs. To begin, we can express a univariate pdf of a SIRP as

$$p_1(x) = AG_{p\,q}^{m\,n}\left(\lambda\, x^2\left|\begin{matrix}a_p\\ b_q\end{matrix}\right.\right) \tag{A.5}$$

for all $-\infty < x < \infty$. As can be verified by the Mellin transform relations (A.3)–(A.4), the normalization factor A and the constant λ, which assures unity variance, must be chosen according to

$$A = \lambda^{1/2}\frac{\prod_{i=m+1}^{q}\Gamma(\frac{1}{2}-b_i)\prod_{i=n+1}^{p}\Gamma(\frac{1}{2}+a_i)}{\prod_{i=i}^{m}\Gamma(\frac{1}{2}+b_i)\prod_{i=1}^{n}\Gamma(\frac{1}{2}-a_i)} \tag{A.6}$$

$$\lambda = (-1)^{\epsilon}\frac{\prod_{i=1}^{q}(\frac{1}{2}+b_i)}{\prod_{i=1}^{p}(\frac{1}{2}+a_i)}, \qquad \epsilon = n-(q-m). \tag{A.7}$$

Brehm and Stammler [53] note that the subclass of SIRPs that are useful for modeling the statistics of speech can be expressed as

$$p_1(y) = A\, G_{0\,2}^{2\,0}(\lambda x^2|b_1,b_2) \tag{A.8}$$

for the real parameters b_1 and b_2, where (A.6–A.7) are specialized as

$$A = \frac{\lambda^{1/2}}{\Gamma(\frac{1}{2}+b_1)\Gamma(\frac{1}{2}+b_2)} \tag{A.9}$$

Table A.1: Meijer G-function parameter values for the Laplace, K_0, and Γ pdfs.

pdf	$p(x)$	b_1	b_2	A	λ
Laplace	$\frac{1}{\sqrt{2}}e^{-\sqrt{2}\|x\|}$	0	$\frac{1}{2}$	$(2\pi)^{-1/2}$	$\frac{1}{2}$
K_0	$\frac{1}{\pi}K_0(\|x\|)$	0	0	$(2\pi)^{-1}$	$\frac{1}{4}$
Γ	$\frac{\sqrt{3}}{4\sqrt{\pi}}\left(\frac{\sqrt{3}\|x\|}{2}\right)^{-1/2}e^{-\sqrt{3}\|x\|/2}$	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{\sqrt{3/2}}{4\pi}$	$\frac{3}{16}$

and

$$\lambda = (\tfrac{1}{2} + b_1)(\tfrac{1}{2} + b_2). \tag{A.10}$$

Table A.1, taken from Brehm and Stammler [53], lists the values of these parameters for the Laplace, K_0, and Γ pdfs. In many cases of interest, a Meijer G-function with a given set of parameters can be represented in closed-form in terms of elementary or special functions. These special cases are tabulated in reference books such as Luke [115]. Alternatively, they have been programmed into computer algebra systems, such as *Mathematica* [54, §3.2.10]. In particular, we can write

$$G_{0\,2}^{2\,0}(z|0, \tfrac{1}{2}) = \sqrt{\pi}e^{-2\sqrt{z}} \tag{A.11}$$

$$G_{0\,2}^{2\,0}(z|0, 0) = 2\,K_0(2\sqrt{z}). \tag{A.12}$$

These equations can be used to verify the correctness of the Laplace and K_0 pdfs. To verify the correctness of the Γ density, we write

$$G_{0\,2}^{2\,0}(z|-\tfrac{1}{4}, \tfrac{1}{4}) = z^{-1/4}G_{0\,2}^{2\,0}(z|0, \tfrac{1}{2}) \tag{A.13}$$

$$= \sqrt{\pi}\,z^{-1/4}\,e^{-2\sqrt{z}} \tag{A.14}$$

where (A.13) follows from (A.2), and (A.14) follows from (A.11).

In general, the multivariate density of order ν can also be expressed in terms of Meijer's G-functions according to [53]

$$p_\nu(\mathbf{x}) = \pi^{-\nu/2}f_\nu(s) \tag{A.15}$$

where

$$f_\nu = \pi^{1/2}\, A_\nu\, s^{(1-\nu)/2} \times G^{3\ 0}_{1\ 3}\left(\lambda_\nu s \,\middle|\, \begin{matrix} 0 \\ \frac{1}{2}(\nu-1), b_1, b_2 \end{matrix}\right) \tag{A.16}$$

and $s = \mathbf{x}^T\mathbf{x}$. In this case (A.6) and (A.7) can be specialized as

$$\epsilon = 0$$

$$A_\nu = \lambda_\nu^{1/2} \frac{\Gamma(\frac{1}{2})}{\Gamma(\frac{1}{2}\nu)\Gamma(\frac{1}{2}+b_1)\,\Gamma(\frac{1}{2}+b_2)} \tag{A.17}$$

$$\lambda_\nu = \nu\,(\tfrac{1}{2}+b_1)\,(\tfrac{1}{2}+b_2). \tag{A.18}$$

The bivariate pdf is obtained by specializing (A.15) and (A.16) as,

$$p_2(\mathbf{x}) = \frac{A_2}{\sqrt{\pi s}}\, G^{3\ 0}_{1\ 3}\left(\lambda_2 s \,\middle|\, \begin{matrix} 0 \\ \frac{1}{2}, b_1, b_2 \end{matrix}\right). \tag{A.19}$$

For the moment, assume $\mathbf{x}$ is real-valued; this analysis will be extended to the case of complex $\mathbf{x}$ in Section A.6. If the components of $\mathbf{x}$ are correlated, we must set

$$s = \mathbf{x}^T {\Sigma_{\mathbf{X}}}^{-1}\mathbf{x}$$

and modify (A.19) according to

$$p_2(\mathbf{x}) = \frac{A_2}{\sqrt{\pi s|\Sigma_{\mathbf{X}}|}}\, G^{3\ 0}_{1\ 3}\left(\lambda_2 s \,\middle|\, \begin{matrix} 0 \\ \frac{1}{2}, b_1, b_2 \end{matrix}\right) \tag{A.20}$$

where $\Sigma_{\mathbf{X}} = \mathcal{E}\{\mathbf{X}\mathbf{X}^T\}$ is the covariance matrix of $\mathbf{X}$.

For the four-variate case, we have

$$p_4(\mathbf{x}) = \frac{A_4}{(\pi s)^{3/2}\,|\Sigma_{\mathbf{X}}|^{1/2}}\, G^{3\ 0}_{1\ 3}\left(\lambda_4 s \,\middle|\, \begin{matrix} 0 \\ \frac{3}{2}, b_1, b_2 \end{matrix}\right). \tag{A.21}$$

A.3 Laplace Density

The Laplace density is perhaps the simplest and best known super-Gaussian distribution. In Table A.1, the univariate form of the Laplace density is given,

along with the parameter values required to represent it with Meijer's G-function as in (A.8–A.10). With the help of Mathematica, we learn

$$G_{1\,3}^{3\,0}\left(z\,\middle|\,\begin{matrix}0\\ \frac{1}{2},0,\frac{1}{2}\end{matrix}\right)=2\sqrt{z}\,K_0(2\sqrt{z}).$$

Hence, specializing (A.20) with $b_1 = 0$ and $b_2 = \frac{1}{2}$, then simplifying provides the bivariate pdf

$$p_2(\mathbf{x})=\frac{2A_2\sqrt{\lambda_2}}{\sqrt{\pi|\Sigma_{\mathbf{X}}|}}\,K_0\left(2\sqrt{\lambda_2 s}\right) \tag{A.22}$$

where from (A.17–A.18) we have

$$\lambda_2 = 2\,(\tfrac{1}{2}+0)\,(\tfrac{1}{2}+\tfrac{1}{2}) = 1 \tag{A.23}$$

$$A_2=\frac{\Gamma(\frac{1}{2})}{\Gamma(1)\,\Gamma(\frac{1}{2})\,\Gamma(1)}=\frac{1}{\Gamma^2(1)}=1. \tag{A.24}$$

Substituting (A.23–A.24) into (A.22), we have

$$p_2(\mathbf{x})=\frac{2}{\sqrt{\pi|\Sigma_{\mathbf{X}}|}}\,K_0(2\sqrt{s}). \tag{A.25}$$

Once more resorting to Mathematica, we find

$$G_{1\,3}^{3\,0}\left(z\,\middle|\,\begin{matrix}0\\ \frac{3}{2},0,\frac{1}{2}\end{matrix}\right)=2z\,K_1(2\sqrt{z}).$$

Hence, specializing (A.21) provides the four-variate pdf

$$p_4(\mathbf{x})=\frac{2A_4\lambda_4}{\pi^{3/2}\,s^{1/2}\,|\Sigma_{\mathbf{X}}|^{1/2}}K_1\left(2\sqrt{\lambda_4 s}\right) \tag{A.26}$$

where

$$\lambda_4 = 2\cdot 1 = 2 \tag{A.27}$$

$$A_4=\sqrt{2}\cdot\frac{\Gamma(\frac{1}{2})}{\Gamma(2)\,\Gamma(\frac{1}{2})\,\Gamma(1)}=\frac{\sqrt{2}}{\Gamma(2)\Gamma(1)}=\sqrt{2}. \tag{A.28}$$

Substituting (A.27–A.28) back into (A.26) provides

$$p_4(\mathbf{x})=\frac{4\sqrt{2}}{\pi^{3/2}\,s^{1/2}\,|\Sigma_{\mathbf{X}}|^{1/2}}K_1(2\sqrt{2s}). \tag{A.29}$$

A.4 K_0 Density

From Mathematica

$$G_{1\,3}^{3\,0}\left(z\,\middle|\,\begin{matrix}0\\ \frac{1}{2},0,0\end{matrix}\right)=\sqrt{\pi}e^{-2\sqrt{z}}$$

so that the bivariate K_0 pdf can be obtained by substituting $b_1 = b_2 = 0$ into (A.20), whereupon we find

$$p_2(\mathbf{x}) = \frac{A_2}{\sqrt{s|\Sigma_{\mathbf{X}}|}}\, e^{-2\sqrt{\lambda_2 s}} \tag{A.30}$$

where

$$\lambda_2 = \tfrac{1}{2} \tag{A.31}$$

$$A_2 = \frac{\sqrt{2}}{2}\cdot\frac{\Gamma(\frac{1}{2})}{\Gamma(1)\,\Gamma(\frac{1}{2})\,\Gamma(\frac{1}{2})} = \frac{\sqrt{2}}{2\,\Gamma(1)\,\Gamma(\frac{1}{2})} = \frac{1}{\sqrt{2\pi}}. \tag{A.32}$$

Substituting (A.31–A.32) into (A.30), we find

$$p_2(\mathbf{x}) = \frac{1}{\sqrt{2\pi\, s|\Sigma_{\mathbf{X}}|}}\, e^{-\sqrt{2s}}. \tag{A.33}$$

From Mathematica

$$G_{1\,3}^{3\,0}\left(z\,\middle|\,\begin{matrix}0\\ \frac{3}{2},0,0\end{matrix}\right)=\frac{\sqrt{\pi}(1+2\sqrt{z})}{2}e^{-2\sqrt{z}},$$

so the four-variate K_0 pdf can be obtained from (A.21),

$$p_4(\mathbf{x}) = \frac{A_4(1+2\sqrt{\lambda_4 s})}{2\pi s^{3/2}|\Sigma_{\mathbf{X}}|^{1/2}}\, e^{-2\sqrt{\lambda_4\, s}} \tag{A.34}$$

where

$$\lambda_4 = 2\cdot\tfrac{1}{2} = 1 \tag{A.35}$$

$$A_4 = \frac{\Gamma(\frac{1}{2})}{\Gamma(2)\,\Gamma(\frac{1}{2})\,\Gamma(\frac{1}{2})} = \frac{1}{\Gamma(2)\,\Gamma(\frac{1}{2})} = \frac{1}{\sqrt{\pi}}. \tag{A.36}$$

Substituting (A.35–A.36) into (A.34), we have

$$p_4(\mathbf{x}) = \frac{(1+2\sqrt{s})}{2(\pi s)^{3/2}|\Sigma_{\mathbf{X}}|^{1/2}}\, e^{-2\sqrt{s}}. \tag{A.37}$$

A.5 Γ Density

For the Γ pdf, $b_1 = -\frac{1}{4}$, $b_2 = \frac{1}{4}$. Subsituting these values into the G-functions appearing in (A.20–A.21) and applying (A.2), we find

$$\begin{aligned} G^{3\,0}_{1\,3}\left(z\left|\begin{matrix}0\\ \frac{1}{2},-\frac{1}{4},\frac{1}{4}\end{matrix}\right.\right) = \\ z^{-1/4}G^{3\,0}_{1\,3}\left(z\left|\begin{matrix}\frac{1}{4}\\ \frac{3}{4},0,\frac{1}{2}\end{matrix}\right.\right) \end{aligned} \tag{A.38}$$

and

$$\begin{aligned} G^{3\,0}_{1\,3}\left(z\left|\begin{matrix}0\\ \frac{3}{2},-\frac{1}{4},\frac{1}{4}\end{matrix}\right.\right) = \\ z^{-1/4}G^{3\,0}_{1\,3}\left(z\left|\begin{matrix}\frac{1}{4}\\ \frac{7}{4},0,\frac{1}{2}\end{matrix}\right.\right). \end{aligned} \tag{A.39}$$

Then, the bi-variate Γ pdf can be expressed as

$$p_2(\mathbf{x}) = \frac{A_2}{\sqrt{\pi s|\Sigma_{\mathbf{X}}|}}(\lambda_2 s)^{-1/4} g_2(\lambda_2 s) \tag{A.40}$$

where

$$g_2(z) = G^{3\,0}_{1\,3}\left(z\left|\begin{matrix}\frac{1}{4}\\ \frac{3}{4},0,\frac{1}{2}\end{matrix}\right.\right) \tag{A.41}$$

$$\lambda_2 = \frac{(\frac{1}{2}+\frac{1}{2})(\frac{1}{2}-\frac{1}{4})(\frac{1}{2}+\frac{1}{4})}{(\frac{1}{2}+0)} = \frac{3}{8} \tag{A.42}$$

$$A_2 = \sqrt{\frac{3}{8}} \cdot \frac{\Gamma(\frac{1}{2}+0)}{\Gamma(\frac{1}{2}+\frac{1}{2})\,\Gamma(\frac{1}{2}-\frac{1}{4})\,\Gamma(\frac{1}{2}+\frac{1}{4})} \approxeq 0.2443. \tag{A.43}$$

For the four-variate Γ pdf, we can write

$$p_4(\mathbf{x}) = \frac{A_4}{(\pi s)^{3/2}\sqrt{|\Sigma_{\mathbf{X}}|}}(\lambda_4 s)^{-1/4} g_4(\lambda_4 s) \tag{A.44}$$

where

$$g_4(z) = G^{3\,0}_{1\,3}\left(z \middle| \begin{matrix} \frac{1}{4} \\ \frac{7}{4}, 0, \frac{1}{2} \end{matrix}\right) \tag{A.45}$$

$$\lambda_4 = \frac{(\frac{1}{2}+\frac{3}{2})(\frac{1}{2}-\frac{1}{4})(\frac{1}{2}+\frac{1}{4})}{(\frac{1}{2}+0)} = \frac{3}{4} \tag{A.46}$$

$$A_4 = \frac{\sqrt{3}}{2} \cdot \frac{\Gamma(\frac{1}{2}+0)}{\Gamma(\frac{1}{2}+\frac{3}{2})\,\Gamma(\frac{1}{2}-\frac{1}{4})\,\Gamma(\frac{1}{2}+\frac{1}{4})} \approxeq 0.1949. \tag{A.47}$$

Unfortunately, the G-functions appearing on the R.H.S. of (A.38–A.39) cannot be expressed in closed-form in terms of elementary or special functions. Hence, it is necessary to use a series expansion to calculate them. The Taylor series [116, §19] of any function $f(z)$ about $z = z_0$ can be expressed as

$$f(z) = \sum_{n=0}^{\infty} \frac{(z-z_0)^n}{n!} f^{(n)}(z_0)$$

where $f^{(n)}(z)$ indicates the nth derivative of $f(z)$ evaluated at $z = z_0$. For series expansions of G-functions, the relation [115, §5.4]

$$z^k \frac{d^k}{dz^k}\left\{ G^{m\,n}_{p\,q}\left(z^{-1}\middle| \begin{matrix} a_p \\ b_q \end{matrix}\right)\right\} = (-)^k G^{m\,n+1}_{p+q\,q+1}\left(z^{-1}\middle| \begin{matrix} 1-k, a_p \\ b_q, 1 \end{matrix}\right)$$

can be used to evaluate the required derivatives. Note that it is not possible to expand the G-function about the origin $z = 0$, as the G-function has a *branch point singularity* at the origin [117, §10.2]. The G-function can, however, be expanded about any point on the positive real axis, which is sufficient for our purposes here.

In practice, a log-likelihood of the Γ pdf is required. Accordingly we need to calculate the logarithm of the G-function. In order to calculate it precisely, the series expansion is performed about 74 points, and we use the series expanded about the point closest to the given argument up to the 12th order. In the case of $s \geqq 70$ in (A.40) or (A.44), we use the derivative to the first order, that is, we used a linear approximation in the log domain. This is because the G-function for those values effectively vanishes leading to floating point errors. Table A.2 shows the series coefficients when $\log g_2(z)$ and $\log g_4(z)$ are expanded about $z_0 = 1$.

Table A.2: Series coefficients of $\log g_2(z)$ and $\log g_4(z)$.

n	$(\log g_2)^{(n)}(z=1)$	$(\log g_4)^{(n)}(z=1)$
0	0.254766	0.389422
1	-0.198347	-0.17901
2	0.228596	0.0967777
3	-0.382523	-0.0266552
4	0.887333	-0.179479
5	-2.70435	1.17531
6	10.3182	-6.79936
7	-47.3711	42.6283
8	253.441	-299.361
9	-1538.09	2358.89
10	10330.3	-20730.1
11	-74825.6	201601.8
12	565360.5	-2.15304 $\times 10^6$

A.6 Complex Densities

The multivariate pdfs derived thus far have been for real-valued random vectors. In order to extend this development for complex-valued subband samples, we will adapt a theorem proven by Neeser and Massey [50, Appendix].

The following definition is due to Neeser and Massey [50, Appendix].

Definition 1 *The random vector* $\mathbf{Y} \in \mathbf{C}^N$ *is a* proper random vector *if*

$$\mathcal{E}\{\mathbf{Y}\mathbf{Y}^T\} = \mathbf{0}. \tag{A.48}$$

Neeser and Massey [50] call the matrix on the L.H.S. of (A.48), the *pseudo-covariance matrix*. Hence, a proper complex random vector is one for which the psuedo-covariance matrix vanishes.

Lemma 1 *Let* $\mathbf{C}^N \ni \mathbf{Y} = \mathbf{X}_c + i\mathbf{X}_s$ *be a proper random vector with pseudo-*

covariance matrix.

$$\Sigma_{\mathbf{Y}} = \mathcal{E}\{\mathbf{Y}\mathbf{Y}^T\} = \mathbf{\Sigma}_{cc} - \mathbf{\Sigma}_{ss} + i(\mathbf{\Sigma}_{sc} + \mathbf{\Sigma}_{sc}^T) \tag{A.49}$$

where

$$\mathbf{\Sigma}_{cc} = \mathcal{E}\{\mathbf{X}_c {\mathbf{X}_c}^T\} \tag{A.50}$$

$$\mathbf{\Sigma}_{ss} = \mathcal{E}\{\mathbf{X}_s {\mathbf{X}_s}^T\} \tag{A.51}$$

$$\mathbf{\Sigma}_{sc} = \mathcal{E}\{\mathbf{X}_s {\mathbf{X}_c}^T\}. \tag{A.52}$$

Then

$$\mathbf{\Sigma}_{cc} = \mathbf{\Sigma}_{ss} \tag{A.53}$$

$$\mathbf{\Sigma}_{sc} = -\mathbf{\Sigma}_{sc}^T. \tag{A.54}$$

Proof: The definition of properness requires that the R.H.S. of (A.49) vanishes, which implies (A.53) and (A.54). □

Note that a matrix satisfying (A.54) is said to be *skew symmetric*. Hence, the conditions (A.53) and (A.54) state that the covariance matrices of the real and imaginary parts of a proper complex random vector must be equal, and the cross-covariance matrices must be skew symmetric.

We now state another intermediate result.

Lemma 2 *Let* $\mathbf{M}_{cc}$, $\mathbf{M}_{ss}$, $\mathbf{M}_{sc}$, *and* $\mathbf{M}_{sc}$, *be real* $N \times N$ *matrices, where* $\mathbf{M}_{cc}$ *and* $\mathbf{M}_{ss}$ *are symmetric and* $\mathbf{M}_{cs}^T = \mathbf{M}_{sc}$. *Define the* $N \times N$ *Hermitian matrix*

$$\mathbf{M} = \mathbf{M}_c + i\mathbf{M}_s \triangleq \mathbf{M}_{cc} + \mathbf{M}_{ss} + i(\mathbf{M}_{sc} - {\mathbf{M}_{sc}}^T) \tag{A.55}$$

and the symmetric $2N \times 2N$ *matrix*

$$\Upsilon \triangleq 2 \begin{bmatrix} \mathbf{M}_{cc} & \mathbf{M}_{cs} \\ \mathbf{M}_{sc} & \mathbf{M}_{ss} \end{bmatrix}. \tag{A.56}$$

Then the quadratic forms

$$\mathcal{E} \triangleq \mathbf{z}^H \mathbf{M} \mathbf{z} \tag{A.57}$$

and

$$\mathcal{E}' \triangleq \begin{bmatrix} \mathbf{z}_c^T & \mathbf{z}_s^T \end{bmatrix} \Upsilon \begin{bmatrix} \mathbf{z}_c \\ \mathbf{z}_s \end{bmatrix} \tag{A.58}$$

are equal for all $\mathbf{z} \triangleq \mathbf{z}_c + i\mathbf{z}_s$*, if and only if*

$$\mathbf{M}_{cc} = \mathbf{M}_{ss} \text{ and } \mathbf{M}_{sc} = -\mathbf{M}_{sc}^T. \tag{A.59}$$

Moreover, under conditions (A.59) $\mathbf{M}$ *is positive (semi-)definite if and only if* Υ *is positive (semi-)definite.*

Proof: See Neeser and Massey [50, Appendix]. □

We now state and prove the main result of this section based on [50, Appendix].

Theorem 1 *Consider a proper complex random vector*

$$\mathbf{C}^N \ni \mathbf{Y} = \mathbf{X}_c + i\mathbf{X}_s$$

with the covariance matrix

$$\Sigma_{\mathbf{Y}} = 2(\mathbf{\Sigma}_{cc} + i\mathbf{\Sigma}_{sc}) \tag{A.60}$$

where $\mathbf{\Sigma}_{cc}$ *and* $\mathbf{\Sigma}_{sc}$ *are defined in (A.50) and (A.52), respectively. Define the stacked random vector*

$$\mathbf{R}^{2N} \ni \mathbf{X} = \begin{bmatrix} \mathbf{X}_c \\ \mathbf{X}_s \end{bmatrix}$$

with covariance matrix

$$\Sigma_{\mathbf{X}} = \mathcal{E}\{\mathbf{X}\,\mathbf{X}^T\} = \begin{bmatrix} \mathbf{\Sigma}_{cc} & \mathbf{\Sigma}_{cs} \\ \mathbf{\Sigma}_{sc} & \mathbf{\Sigma}_{ss} \end{bmatrix}.$$

Then,

$$\mathbf{x}^T \Sigma_{\mathbf{X}}^{-1} \mathbf{x} = 2\,\mathbf{y}^H \Sigma_{\mathbf{Y}}^{-1} \mathbf{y} \tag{A.61}$$

for all

$$\mathbf{y} = \mathbf{x}_c + i\mathbf{x}_s \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_s \end{bmatrix}.$$

Moreover,

$$\sqrt{|\Sigma_{\mathbf{X}}|} = 2^{-N} |\Sigma_{\mathbf{Y}}|. \tag{A.62}$$

Proof: Based on a well-known result for the inverse of block matrices [118, pg. 656], we can write

$$\mathbf{\Sigma}_{\mathbf{X}}^{-1} = \begin{bmatrix} \Delta^{-1} & {\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}}\Delta^{-1} \\ -\Delta^{-1}\mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1} & \Delta^{-1} \end{bmatrix} \tag{A.63}$$

where

$$\Delta \triangleq \mathbf{\Sigma}_{\mathrm{cc}} + \mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}} \tag{A.64}$$

is symmetric. We must now show that the upper-right block of $\mathbf{\Sigma}_{\mathbf{X}}^{-1}$ is skew symmetric. Observe that

$$\Delta{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}} = \mathbf{\Sigma}_{\mathrm{sc}} + \mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}} = \mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\Delta$$

which implies

$$\begin{aligned} {\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}}\Delta^{-1} = \Delta^{-1}\mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1} &= \left({\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}{\mathbf{\Sigma}_{\mathrm{sc}}}^{T}\Delta^{-1}\right)^{T} \\ &= -\left({\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}\mathbf{\Sigma}_{\mathrm{sc}}\Delta^{-1}\right)^{T}. \end{aligned}$$

Hence, the upper and lower blocks are skew symmetric. Therefore, $\mathbf{\Sigma}_{\mathbf{X}}^{-1}$ satisfies (A.59) and Lemma 2 applies for $\Upsilon \triangleq \frac{1}{2}\mathbf{\Sigma}_{\mathbf{X}}^{-1}$ and

$$\mathbf{M} \triangleq \Delta^{-1}(\mathbf{I} - i\mathbf{\Sigma}_{\mathrm{sc}}{\mathbf{\Sigma}_{\mathrm{cc}}}^{-1}) \tag{A.65}$$

where (A.65) follows from associating the block components in (A.56) with their counterparts in (A.63), then applying (A.55). Multiplying $\mathbf{M}$ in (A.65) with (A.60) yields the identity matrix, which implies $\mathbf{M} = \mathbf{\Sigma}_{\mathbf{Y}}^{-1}$. Therefore (A.61) follows from Lemma 1.

Using a well-known result on the determinant of block matrices [118, pg. 650] and the skew symmetry of $\mathbf{\Sigma}_{cs}$, we find

$$|\Sigma_{\mathbf{X}}| = |\mathbf{\Sigma}_{\mathrm{cc}}|\,|\Delta|. \tag{A.66}$$

Observe that

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{Y}}^{T} &= 2(\boldsymbol{\Sigma}_{\mathrm{cc}} - i\boldsymbol{\Sigma}_{\mathrm{sc}}) \\ &= 2(\mathbf{I} - i\boldsymbol{\Sigma}_{\mathrm{sc}}{\boldsymbol{\Sigma}_{\mathrm{cc}}}^{-1})\boldsymbol{\Sigma}_{\mathrm{cc}}. \end{aligned} \tag{A.67}$$

Hence, from (A.65) and (A.67), along with $\mathbf{M} = \boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}$, it follows that

$$\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1} = \tfrac{1}{4}\Delta^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}}^{T}{\boldsymbol{\Sigma}_{\mathrm{cc}}}^{-1} \tag{A.68}$$

Now

$$\left|\Sigma_{\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}\right| = \left|\tfrac{1}{4}\Sigma_{\mathbf{Y}}\Delta^{-1}\boldsymbol{\Sigma}_{\mathbf{Y}}^{T}{\boldsymbol{\Sigma}_{\mathrm{cc}}}^{-1}\right| \tag{A.69}$$

$$= \frac{|\Sigma_{\mathbf{Y}}|^2}{2^{2N}\,|\Delta|\,|\boldsymbol{\Sigma}_{\mathrm{cc}}|} = 1 \tag{A.70}$$

where (A.69) follows from (A.68), and (A.70) follows from a basic property of determinants of matrices. Substituting as in (A.66) for $|\Delta|\,|\boldsymbol{\Sigma}_{\mathrm{cc}}|$ in (A.70) and rearranging is sufficient to prove (A.62). □

Based on Theorem 1, we can rewrite (A.25) for proper $y \in \mathbf{C}$ as

$$p_{\mathrm{Laplace}}(y) = \frac{4}{\sqrt{\pi}\sigma_Y^2}\,K_0\left(\frac{2\sqrt{2}|y|}{\sigma_Y}\right) \tag{A.71}$$

where $\sigma_Y^2 = \mathcal{E}\{|Y|^2\}$. For proper $\mathbf{y} \in \mathbf{C}^2$, we can rewrite (A.29) as

$$p_{\mathrm{Laplace}}(\mathbf{y}) = \frac{16}{\pi^{3/2}s^{1/2}|\Sigma_{\mathbf{Y}}|}\,K_1\left(4\sqrt{s}\right) \tag{A.72}$$

where $\Sigma_{\mathbf{Y}} = \mathcal{E}\{\mathbf{Y}\mathbf{Y}^H\}$ and

$$s = \mathbf{y}^H\Sigma_{\mathbf{Y}}^{-1}\mathbf{y}.$$

Similarly, for the K_0 density, we can rewrite (A.33) and (A.37) respectively as

$$p_{K_0}(y) = \frac{1}{\sqrt{\pi}|y|\sigma_Y}e^{-2\,|y|/\sigma_Y} \tag{A.73}$$

$$p_{K_0}(\mathbf{y}) = \frac{\sqrt{2}+4\sqrt{s}}{2\,(\pi s)^{3/2}\,|\Sigma_{\mathbf{Y}}|}\,e^{-2\sqrt{2\,s}}. \tag{A.74}$$

For the Γ pdf, it is necessary to calculate the bi– and four–variates with a series expansion, as mentioned previously. It is clear from (A.40) and (A.44),

however, that the functional dependence of the Γ pdf on the subband samples and their statistics enters exclusively through the terms $|\Sigma_{\mathbf{X}}|$ and $s = \mathbf{x}^T \mathbf{\Sigma}_{\mathbf{X}}^{-1} \mathbf{x}$. Hence, variates of the Γ pdf can also be specialized for complex data using the results of Theorem 1.

A.7 Partial Derivate Calculation

In order to estimate beamforming parameters with an MMI criterion using non-Gaussian pdfs, we first approximate

$$\begin{aligned} I(Y_1, Y_2) \approx \frac{1}{N} \sum_{t=0}^{N-1} \Big[& \log p \left(y_1^{(t)}, y_2^{(t)} \right) \\ & - \log p \left(y_1^{(t)} \right) - \log p \left(y_2^{(t)} \right) \Big] \end{aligned} \tag{A.75}$$

where

$$y_i^{(t)} = (\mathbf{w}_{\mathrm{q},i} - \mathbf{B}_i \mathbf{w}_{\mathrm{a},i})^H \mathbf{x}^{(t)}$$

for each $\mathbf{x}^{(t)}$ drawn from a *training set* $\mathcal{X} = \left\{ \mathbf{x}^{(t)} \right\}_{t=0}^{N-1}$. From (A.75), it follows that

$$\begin{aligned} \frac{\partial I(Y_1, Y_2)}{\partial {\mathbf{w}_{\mathrm{a},i}}^*} \approx \frac{1}{N} \sum_{t=0}^{N-1} \Bigg[& \frac{\partial \log p \left(y_1^{(t)}, y_2^{(t)} \right)}{\partial {\mathbf{w}_{\mathrm{a},i}}^*} \\ & - \frac{\partial \log p \left(y_1^{(t)} \right)}{\partial {\mathbf{w}_{\mathrm{a},i}}^*} - \frac{\log p \left(y_2^{(t)} \right)}{\partial {\mathbf{w}_{\mathrm{a},i}}^*} \Bigg]. \end{aligned} \tag{A.76}$$

The partial derivative (A.76) is specialized for the Laplace, K_0 and, Γ pdfs in [119].

Appendix B

The r-th moment and kurtosis of the GG pdf

Here, we derive two useful statistics of the GG pdf, the r-th moment and kurtosis.

The rth moment of the GG pdf can be expressed as

$$\mathcal{E}\{y^r\} = \frac{1}{2\Gamma(1+1/p)A(p,\hat{\sigma})}\int_{-\infty}^{\infty} y^r \exp\left[-\frac{|y|^p}{A(p,\hat{\sigma})}\right] dy. \tag{B.1}$$

Since the GG pdf is an even function about the mean, we can rewrite (B.1) as

$$\mathcal{E}\{y^r\} = \frac{1}{\Gamma(1+1/p)A(p,\hat{\sigma})}\int_{0}^{\infty} y^r \exp\left[-\frac{y^p}{A^p(p,\hat{\sigma})}\right] dy. \tag{B.2}$$

Upon defining

$$v = \frac{y^p}{A^p(p,\hat{\sigma})},$$

from which it follows

$$\frac{dv}{dy} = \frac{py^{p-1}}{A^p(p,\hat{\sigma})},$$

then (B.2) can be solved as

$$\begin{aligned}\mathcal{E}\{y^r\} &= \frac{A^r(p,\hat{\sigma})}{p\,\Gamma(1+1/p)}\int_0^{\infty} v^{\frac{r+1}{p}-1}\, e^{-v}\, dv \\ &= \frac{A^r(p,\hat{\sigma})}{p\,\Gamma(1+1/p)}\,\Gamma\left(\frac{r+1}{p}\right).\end{aligned} \tag{B.3}$$

By substituting the second and fourth moments obtained from Equation (B.3), the kurtosis of the GG pdf can now be expressed as

$$\text{kurt}\left(Y_{gg}\right) = \frac{A(p,\hat{\sigma})^4}{p\Gamma(1+1/p)}\Gamma\left(5/p\right) - 3\left\{\frac{A(p,\hat{\sigma})^2}{p\Gamma(1+1/p)}\Gamma\left(3/p\right)\right\}^2. \tag{B.4}$$

As $p\Gamma(1+1/p) = \Gamma(1/p)$, Eqn. (B.4) can be simplified to

$$\text{kurt}\left(Y_{gg}\right) = \hat{\sigma}^4\left\{\frac{\Gamma(1/p)\,\Gamma(5/p)}{\Gamma^2(3/p)} - 3\right\}. \tag{B.5}$$

Appendix C

The implementation of the optimization algorithm

Here we describe a nonlinear conjugate gradient method for our beamforming algorithm. Gradient algorithms are generally used to find the local minimum of a function [60, §1.6]. However, we have to maximize the objective function in the case that either negentropy or kurtosis is used. Accordingly, the author explain how to find the local minimum of the negative of the corresponding objective function with a conjugate gradient algorithm, which is equivalent to seeking the local maximum. Notice that we do not need it when the minimum mutual information criterion is used.

The conjugate algorithms proceed as a succession of line minimizations. The sequence of *conjugate directions* is used to approximate the curvature of a cost function in the neighborhood of the minimum.

Expressing the objective function as $\mathcal{I}(\mathbf{w}_{\mathrm{a}}{}^{*}) = -\mathcal{J}(Y;\alpha)$, we can calculate the initial search direction as that opposite to the gradient according to

$$\Delta\mathbf{w}_{\mathrm{a}(0)}^{*} = -\frac{\partial\mathcal{I}(\mathbf{w}_{\mathrm{a}(0)}^{*})}{\partial\mathbf{w}_{\mathrm{a}}{}^{*}},$$

where the required partial derivative is specified by one of (7.16), (7.19) or (7.25).

A line search is performed in that direction and a step size is optimized as follows:

$$\begin{aligned} \beta_{(0)} &:= \text{argmin}_\beta \, \mathcal{I}(\mathbf{w}_\mathrm{a}{}^* + \beta \Delta \mathbf{w}^*_{\mathrm{a}(0)}) \text{ and} \\ \mathbf{w}^*_{\mathrm{a}(1)} &= \mathbf{w}^*_{\mathrm{a}(0)} + \beta_{(0)} \Delta \mathbf{w}^*_{\mathrm{a}(0)}, \end{aligned}$$

where the initial active weight vector is set to zero in this work.

After the first iteration, the following steps constitute one iteration of searching the minimum along a subsequent conjugate direction $\Lambda \mathbf{w}^*_{\mathrm{a}(n)}$, where $\Lambda \mathbf{w}^*_{\mathrm{a}(0)} = \Delta \mathbf{w}^*_{\mathrm{a}(0)}$:

1. Calculate the gradient of the objective function

$$\Delta \mathbf{w}^*_{\mathrm{a}(n)} = -\frac{\partial \mathcal{I}(\mathbf{w}^*_{\mathrm{a}(n)})}{\partial \mathbf{w}_\mathrm{a}{}^*}.$$

2. Compute the modified Polak-Ribière formula

$$\gamma_{(n)} = Re \left\{ \frac{\Delta \mathbf{w}^T_{\mathrm{a}(n)} \left(\Delta \mathbf{w}^*_{\mathrm{a}(n)} - \Delta \mathbf{w}^*_{\mathrm{a}(n-1)} \right)}{\Delta \mathbf{w}^T_{\mathrm{a}(n-1)} \Delta \mathbf{w}^*_{\mathrm{a}(n-1)}} \right\},$$

   where $(\cdot)^T$ denotes the transpose operation.

3. Update the conjugate direction

$$\Lambda \mathbf{w}^*_{\mathrm{a}(n)} = \Delta \mathbf{w}^*_{\mathrm{a}(n)} + \gamma_{(n)} \Lambda \mathbf{w}^*_{\mathrm{a}(n-1)}.$$

4. Perform the line search and optimize the step size

$$\beta_{(n)} = \text{argmin}_\beta \, \mathcal{I}(\mathbf{w}^*_{\mathrm{a}(n)} + \beta \Lambda \mathbf{w}^*_{\mathrm{a}(n)}). \tag{C.1}$$

5. Update the estimate of the active weight vector

$$\mathbf{w}^*_{\mathrm{a}(n+1)} = \mathbf{w}^*_{\mathrm{a}(n)} + \beta_{(n)} \Lambda \mathbf{w}^*_{\mathrm{a}(n)}.$$

In each step, the line search is repeated until

$$\mathrm{Re} \left\{ \Delta \mathbf{w}_{\mathrm{a}(n)} \cdot \Lambda \mathbf{w}^*_{\mathrm{a}(n)} \right\} < \mathrm{tol} \, |\Delta \mathbf{w}_{\mathrm{a}(n)}| \, |\Lambda \mathbf{w}_{\mathrm{a}(n)}|. \tag{C.2}$$

where tol indicates the accuracy of the line search. We set tol = 0.001 in our experiments. The convergence properties of the numerical search were not significantly altered by changing the method used to calculate $\gamma_{(n)}$, nor by adjusting the accuracy of the line search. Applying a more accurate model for the pdf of the subband samples of speech had a larger effect on the speed of convergence than any adjustment of the parameters of the conjugate gradients search.

Appendix D

Beamforming Toolkit

D.1 Introduction

This appendix describes how to construct beamforming applications with speech feature extraction (sfe) and beamforming toolkit (btk). The fundamental components are written in C++. Users can build the entire system in C++. In addition to the C++ program interface, sfe and btk provide the Python interface produced by SWIG [120]. Programming in Python is much easier and quicker just at the expense of computational time; See [121] for the detail. The toolkits are developed on the Linux platform and we confirmed that they worked on Suse and Debian Linux.

D.2 Installation and Configuration

You need to compile sfe and btk. Before you do it, you might need to set the environmental variables, PKG_CONFIG PYTHONPATH LD_LIBRARY_PATH and LIBRARY_PATH.

Those can be specified in .cshrc by writing:

setenv PYTHONPATH $DIST/lib/python2.4/site-packages

setenv PKG_CONFIG_PATH $DIST/lib/pkgconfig

setenv LD_LIBRARY_PATH $DIST/lib:.:$LD_LIBRARY_PATH
setenv LIBRARY_PATH $LD_LIBRARY_PATH
where we assume that you want to install the toolkit in $DIST.

You can download the latest version from the Subversion (SVN) repository, http://distant-automatic-speech-recognition.org/repos. Those can be done by typing the commands,
svn co http://distant-automatic-speech-recognition.org/repos/sfe/trunk and svn co http://distant-automatic-speech-recognition.org/repos/btk/branches/kenichi.

In order to obtain a user account for the access, you have to ask an administrator. The contact is JohnDOTMcDonoughATlsvDOTuni-saarlandDOTde.

After you download them, you have to first install the sfe. You change the directory which contains autogen.sh, and execute ./autogen.sh, ./configure –prefix=$DIST. If you are lucky, Makefile will be produced in the current directly. Otherwise, you will have errors in the case that your system does not have required software. Typically you have to install libsndfile, libsamplerate, FFTW, GSL, SWIG, pkg-config, Python, Numerical python (24.2) and pygsl. We do not yet incorporate numpy and scipy. Therefore, you have to download the old version of the numpy and scipy, Numerical python 24.2.

If you successfully obtain the Makefile, you just type make;make install. The sfe will be installed in $DIST.

The btk can be installed in the same way as you did sfe.

D.3 How to use the Toolkits in Python

After you installed sfe and btk and configured the environmental variables properly, you should be able to use the libraries from Python.

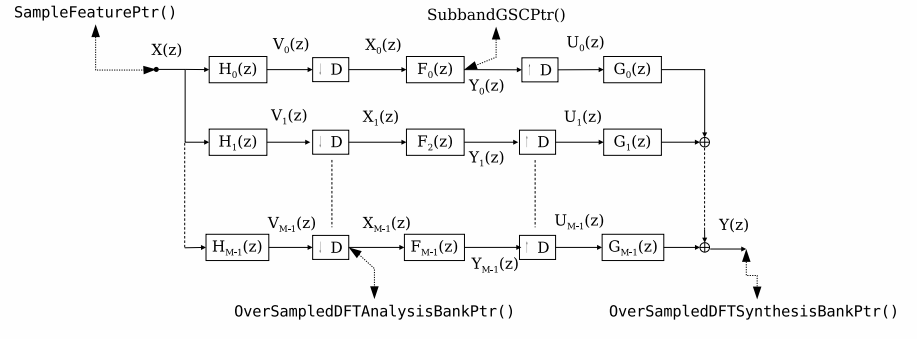


Figure D.1: Schematic of a modulated subband analysis-synthesis filter bank.

D.3.1 Subband Processing

Here we describe a sample script for reconstructing a speech signal with over-sampled uniform DFT filter banks.

Figure D.2 shows a schematic of a *modulated filter bank* with M *subbands* and a *decimation factor* of D.

First you have to design prototypes for the filter banks. The btk provides the several scripts for the filter bank design. You can find them in tools/filterbank. The Nyquist(M) filter bank is most suitable for adaptive beamforming[30]. The path of the script is tools/filterbank/DesignNyquistFilterBank.m. You may have to convert an output ASCII file produced with DesignNyquistFilterBank.m to a different file format so that Python scripts can load those filter coefficients. That can be done by the script, tools/filterbank/convertPythonBin.py. Once you have the binary file of the coefficients of the analysis and synthesis filter prototype, you can enjoy subband processing.

Here is the sample script for that:

```
import os.path
import pickle
import wave
import copy
```

```
import getopt, sys
from Numeric import *
from types import FloatType

from sfe.common import *
from sfe.stream import *
from sfe.feature import *
from sfe.utils import *

from btk.modulated import *

M = 256 # the number of subbands
m = 2   # filter length factor (filter length = m * M)
r = 1   # decimation factor
R = 2**r
D = M / R # corresponds to the frame shift

# Read analysis prototype 'h'
protoFile  = '%s/h-M=%d-m=%d-r=%d.txt' %(protoPath, M, m,
    r)
print 'Loading analysis prototype from \'%s\'' %protoFile
fp = open(protoFile, 'r')
h_fb = pickle.load(fp)
fp.close()

# Read synthesis prototype 'g'
protoFile = '%s/g-M=%d-m=%d-r=%d.txt' %(protoPath, M, m,
    r)
print 'Loading synthesis prototype from \'%s\'' %
    protoFile
```

```
fp = open(protoFile, 'r')
g_fb = pickle.load(fp)
fp.close()

# A Python iterative object for reading wave data.
sampleFeature = SampleFeaturePtr(blockLen = D, shiftLen =
    D, padZeros = True)
# A Python iterative object which returns the subband
    components.
analysisFB  = OverSampledDFTAnalysisBankPtr(sampleFeature
    , prototype =
 h_fb, M = M, m = m, r = r, delayCompensationType=2 )
# A Python iterative object which reconstructs the wave
    data.
synthesisFB =
 OverSampledDFTSynthesisBankPtr(
     PyVectorComplexFeatureStreamPtr(analysisFB),
     prototype = g_fb, M = M, m = m, r = r,
     delayCompensationType=2 )

# read wave data of sampling rate 16k from file 'sample.
    wav'.
sampleFeature.read( 'sample.wav', 16000 )

wavebuffer  = []
# reconstruct input wave data
# The methods next() of the C++ objects are actually
    called at each step.
# This script executes SampleFeature.next(),
    OverSampledDFTAnalysisBank.next() and
```

```
# OverSampledDFTSynthesisBank.next() in the C++ module.
for b in synthesisFB:
    # The values returned by OverSampledDFTSynthesisBank.
        next() are
    # accumulated into wavebuffer.
    wavebuffer.extend(copy.deepcopy(b))

# write the wave data into a Microsoft wave file.
storewave = array(wavebuffer, Float)
wavefile = wave.open('output.wav', 'w')
wavefile.setnchannels(1)
wavefile.setsampwidth(2)
wavefile.setframerate(int(sampleRate))
storewave *= float(D)
wavefile.setnframes(len(storewave))
wavefile.writeframes(storewave.astype('s').tostring())
wavefile.close()
```

Figure D.2 also shows the relationship between the Python object and process in adaptive subband processing. As shown in Figure D.2, Python object SampleFeaturePtr() corresponds to the input signal to the subband processing system. OverSampledDFTAnalysisBankPtr() is also connected to processing with the analysis filter banks. The outputs of the analysis filter banks are obtained with OverSampledDFTAnalysisBankPtr().next(). Those outputs will be processed with a beamforming technique which is described in the next section. The process with the synthesis filter banks is associated with OverSampledDFTSynthesisBankPtr() whose member next() returns the reconstructed signal in the time domain. The dependency between two objects is usually created by feeding an object to the other. In the above script, the object for analysis filterbank processing, analysisFB, is given to the object for the synthesis filter bank, synthesisFB. Notice that analysisFB has the object, sampleFeatureis. In such a

relationship, the execution of the iterator of synthesisFB calls every iterator implemented in all the dependent objects, analysisFB and sampleFeatureis. The iterator is normally implemented as a method next() in Python or C++ module. The scripts for the experiment are in /idiap/kkumata/project/ssc1/bf/ssc[1-2]/bf/5814.

D.3.2 Subband Beamforming

Delay-and-sum Beamforming

Now you are ready to implement beamforming algorithm. Let us begin with the delay-and-sum beamformer. In addition to files for subband processing, the delay-and-sum beamforming algorithm generally requires source positions and geometry information of a microphone array. You can estimate source position with the source localization toolkit *sltk* (although this document does not describe how to use sltk). You can see the scripts in /idiap/kkumata/project/ssc1/bf/ssc1/bf/000001 for the actual experiments.

```
import sys
import os
import os.path
import shutil
import pickle
import glob
import wave
from Numeric import *
from types import FloatType
import getopt, sys
from copy import *


import string
import re
```

```
from sfe.common import *
from sfe.stream import *
from sfe.feature import *
from sfe.utils import *

from btk import dbase
from btk.modulated import *
from btk.subbandBeamforming import *
from btk.beamformer import *

def calcDelaysPolar2(phi, theta, mpos):
    # @brief Calculate the delays.

    sspeed = 343740.0
    chanN  = len(mpos)

    delays = []
    c_x = - Numeric.sin(theta) * Numeric.cos(phi)
    c_y = - Numeric.sin(theta) * Numeric.sin(phi)
    c_z = - Numeric.cos(theta)
    for i in range(chanN):
        t = (c_x * mpos[i, 0] + c_y * mpos[i, 1] + c_z *
            mpos[i, 2]) / sspeed
        delays.append( t )

    delays = Numeric.array(delays, Numeric.Float)

    return delays
```

```
def beamform( inputFilePrefix, azimuth, elevation, M, m,
    r, postfilterType ):

    # Geometry of the microphone array (circular
        microphone array)
    chanN = 8
    arrgeom = []
    arrgeom.append([100.0, 0.0, 0.0])
    arrgeom.append([70.7106781, 70.7106781, 0.0])
    arrgeom.append([0.0, 100.0, 0.0])
    arrgeom.append([-70.7106781, 70.7106781, 0.0])
    arrgeom.append([-100.0, 0.0, 0.0])
    arrgeom.append([-70.7106781, -70.7106781, 0.0])
    arrgeom.append([0.0, -100.0, 0.0])
    arrgeom.append([70.7106781, -70.7106781, 0.0])

    # Filter bank parameters
    R     = 2**r
    D     = M / R # frame shift

    sampleRate          = 16000
    outSampleRate       = 16000
    alpha       = 0.2 # for post-filtering

    # Load filter bank prototype
    # Read analysis prototype 'h'
    protoFile = './h-M=%d-m=%d-r=%d.txt' %(protoPath, M,
        m, r)
    fp = open(protoFile, 'r')
    h_fb = pickle.load(fp)
```

```
fp.close()

# Read synthesis prototype 'g'
protoFile = '%s/g-M=%d-m=%d-r=%d.txt' %(protoPath, M,
    m, r)
fp = open(protoFile, 'r')
g_fb = pickle.load(fp)
fp.close()

# output file name
filename  = './wav.PF%d_a%0.2f_M=%d-m=%d-r=%d/output.
   wav' %(postfilterType, alpha, M, m, r)

# Init the beamformer object
pBeamformer = SubbandGSCPtr( fftLen=M, halfBandShift=
   False )
output = ZelinskiPostFilterPtr(
   PyVectorComplexFeatureStreamPtr(pBeamformer),
   fftLen, alpha, postfilterType )

# Build the analysis chain
sampleFeats = []
analysisFBs = []
for chX in range(chanN):
    sampleFeature = SampleFeaturePtr( blockLen=D,
        shiftLen=D, padZeros=True )
    sampleFeats.append(sampleFeature)
    analysisFB = OverSampledDFTAnalysisBankPtr(
        sampleFeature, prototype=h_fb, M=M, m=m, r=r )
    analysisFBs.append(analysisFB)
```

```
        pBeamformer.setChannel(analysisFB)

    # Init synthesisFB
    synthesisFB =  OverSampledDFTSynthesisBankPtr(
        PyVectorComplexFeatureStreamPtr(output), prototype
        =g_fb, M=M, m=m, r=r )

    # read multi-channel data.
    for chanX in range(chanN):
        nextFile = '%s%02d.wav' %(inputFilePrefix, chanX)
        if not os.path.exists(nextFile):
            print 'Could not find file %s' %nextFile
        print 'Loading file %s' %nextFile
        sampleFeats[chanX].read(nextFile, samplerate =
            sampleRate)

    # calculate time delays and beamformer's weights.
    delays1 = calcDelaysPolar2( azimuth, elevation, array
        (arrgeom) )
    pBeamformer.calcGSCWeights( sampleRate, delays1 )

    # Here we go....
    wavebuffer  = []
    # The methods next() of the C++ modules are called at
        each step.
    # In this case, SampleFeature.next(),
    # OverSampledDFTAnalysisBank.next(),
    # SubbandGSC.next() and OverSampledDFTSynthesisBank.
        next()
    # are called.
```

```
    for b in synthesisFB:
        output.setBeamformer( pBeamformer )
        wavebuffer.extend(deepcopy(b))

    # Write WAV file to disk
    storewave = array(wavebuffer, Float)
    if not os.path.exists(os.path.dirname(filename)):
        os.makedirs(os.path.dirname(filename))
    wavefile = wave.open(filename, 'w')
    wavefile.setnchannels(1)
    wavefile.setsampwidth(2)
    wavefile.setframerate(int(outSampleRate))
    wavefile.setnframes(len(storewave))
    wavefile.writeframes(storewave.astype('s').tostring()
        )

    wavefile.close()
    pBeamformer.reset()

try:
    opts, args = getopt.getopt(sys.argv[1:], "hi:s:p:", [
        "help", "input=", "pf="])
except getopt.GetoptError:
    # print help information and exit:
    sys.exit(2)


# parameters for filter banks
M = 256 # the number of subbands
m = 2   # filter length factor
```

```
r = 1    # decimation factor
# the direction of arrival of a sound source
azimuth   = 0.0
elevation = 0.0
# path for multiple wave files
inputFilePrefix   = 'inputdir/test-ch'
# which type of post-filtering is used
postfilterType    = 0 # 0 (no post-filter), 2 (Zelinski
    with abs() real operator), 8 ( use beamformer output )

for o, a in opts:
    if o in ("-h", "--help"):
        sys.exit()
    elif o in ("-i", "--input"):
        inputFilePrefix = a
    elif o in ("-p", "--pf"):
        postfilterType = int(a)

beamform( inputFilePrefix, M, m, r, postfilterType )
```

D.4 How to use the Toolkits in Python

After you installed sfe and btk and configured the environmental variables properly, you should be able to use the libraries from Python.

D.4.1 Subband Processing

Here we describe a sample script for reconstructing a speech signal with oversampled uniform DFT filter banks.

Figure D.2 shows a schematic of a *modulated filter bank* with M *subbands* and a *decimation factor* of D.

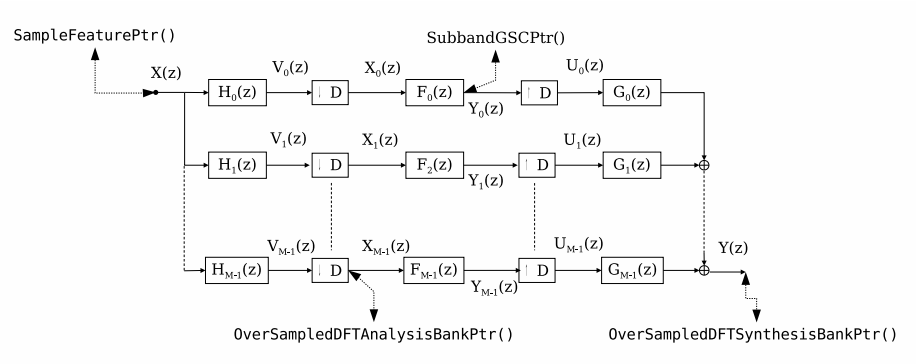


Figure D.2: Schematic of a modulated subband analysis-synthesis filter bank.

First you have to design prototypes for the filter banks. The btk provides the several scripts for the filter bank design. You can find them in tools/filterbank. The Nyquist(M) filter bank is most suitable for adaptive beamforming[30]. The path of the script is tools/filterbank/DesignNyquistFilterBank.m. You may have to convert an output ASCII file produced with DesignNyquistFilterBank.m to a different file format so that Python scripts can load those filter coefficients. That can be done by the script, tools/filterbank/convertPythonBin.py. Once you have the binary file of the coefficients of the analysis and synthesis filter prototype, you can enjoy subband processing.

Here is the sample script for that:

```
import os.path
import pickle
import wave
import copy
import getopt, sys
from Numeric import *
from types import FloatType


from sfe.common import *
from sfe.stream import *
```

```
from sfe.feature import *
from sfe.utils import *

from btk.modulated import *

M = 256 # the number of subbands
m = 2   # filter length factor (filter length = m * M)
r = 1   # decimation factor
R = 2**r
D = M / R # corresponds to the frame shift

# Read analysis prototype 'h'
protoFile  = '%s/h-M=%d-m=%d-r=%d.txt' %(protoPath, M, m,
    r)
print 'Loading analysis prototype from \'%s\'' %protoFile
fp = open(protoFile, 'r')
h_fb = pickle.load(fp)
fp.close()

# Read synthesis prototype 'g'
protoFile = '%s/g-M=%d-m=%d-r=%d.txt' %(protoPath, M, m,
    r)
print 'Loading synthesis prototype from \'%s\'' %
    protoFile
fp = open(protoFile, 'r')
g_fb = pickle.load(fp)
fp.close()

# A Python iterative object for reading wave data.
sampleFeature = SampleFeaturePtr(blockLen = D, shiftLen =
```

```
    D, padZeros = True)
# A Python iterative object which returns the subband
    components.
analysisFB  = OverSampledDFTAnalysisBankPtr(sampleFeature
    , prototype =
 h_fb, M = M, m = m, r = r, delayCompensationType=2 )
# A Python iterative object which reconstructs the wave
    data.
synthesisFB =
 OverSampledDFTSynthesisBankPtr(
     PyVectorComplexFeatureStreamPtr(analysisFB),
     prototype = g_fb, M = M, m = m, r = r,
     delayCompensationType=2 )

# read wave data of sampling rate 16k from file 'sample.
    wav'.
sampleFeature.read( 'sample.wav', 16000 )

wavebuffer  = []
# reconstruct input wave data
# The methods next() of the C++ objects are actually
    called at each step.
# This script executes SampleFeature.next(),
    OverSampledDFTAnalysisBank.next() and
# OverSampledDFTSynthesisBank.next() in the C++ module.
for b in synthesisFB:
    # The values returned by OverSampledDFTSynthesisBank.
        next() are
    # accumulated into wavebuffer.
    wavebuffer.extend(copy.deepcopy(b))
```

```
# write the wave data into a Microsoft wave file.
storewave = array(wavebuffer, Float)
wavefile = wave.open('output.wav', 'w')
wavefile.setnchannels(1)
wavefile.setsampwidth(2)
wavefile.setframerate(int(sampleRate))
storewave *= float(D)
wavefile.setnframes(len(storewave))
wavefile.writeframes(storewave.astype('s').tostring())
wavefile.close()
```

Figure D.2 also shows the relationship between the Python object and process in adaptive subband processing. As shown in Figure D.2, Python object SampleFeaturePtr() corresponds to the input signal to the subband processing system. OverSampledDFTAnalysisBankPtr() is also connected to processing with the analysis filter banks. The outputs of the analysis filter banks are obtained with OverSampledDFTAnalysisBankPtr().next(). Those outputs will be processed with a beamforming technique which is described in the next section. The process with the synthesis filter banks is associated with OverSampledDFTSynthesisBankPtr() whose member next() returns the reconstructed signal in the time domain. The dependency between two objects is usually created by feeding an object to the other. In the above script, the object for analysis filterbank processing, analysisFB, is given to the object for the synthesis filter bank, synthesisFB. Notice that analysisFB has the object, sampleFeatureis. In such a relationship, the execution of the iterator of synthesisFB calls every iterator implemented in all the dependent objects, analysisFB and sampleFeatureis. The iterator is normally implemented as a method next() in Python or C++ module. The scripts for the experiment are in /idiap/kkumata/project/ssc1/bf/ssc[1-2]/bf/5814.

D.4.2 Subband Beamforming

Delay-and-sum Beamforming

Now you are ready to implement beamforming algorithm. Let us begin with the delay-and-sum beamformer. In addition to files for subband processing, the delay-and-sum beamforming algorithm generally requires source positions and geometry information of a microphone array. You can estimate source position with the source localization toolkit *sltk* (although this document does not describe how to use sltk). You can see the scripts in /idiap/kkumata/project/ssc1/bf/ssc1/bf/000001 for the actual experiments.

```
import sys
import os
import os.path
import shutil
import pickle
import glob
import wave
from Numeric import *
from types import FloatType
import getopt, sys
from copy import *

import string
import re

from sfe.common import *
from sfe.stream import *
from sfe.feature import *
from sfe.utils import *

from btk import dbase
```

```
from btk.modulated import *
from btk.subbandBeamforming import *
from btk.beamformer import *

def calcDelaysPolar2(phi, theta, mpos):
    # @brief Calculate the delays.

    sspeed = 343740.0
    chanN  = len(mpos)

    delays = []
    c_x = - Numeric.sin(theta) * Numeric.cos(phi)
    c_y = - Numeric.sin(theta) * Numeric.sin(phi)
    c_z = - Numeric.cos(theta)
    for i in range(chanN):
        t = (c_x * mpos[i, 0] + c_y * mpos[i, 1] + c_z *
            mpos[i, 2]) / sspeed
        delays.append( t )

    delays = Numeric.array(delays, Numeric.Float)

    return delays

def beamform( inputFilePrefix, azimuth, elevation, M, m,
    r, postfilterType ):

    # Geometry of the microphone array (circular
        microphone array)
    chanN = 8
    arrgeom = []
```

```
arrgeom.append([100.0, 0.0, 0.0])
arrgeom.append([70.7106781, 70.7106781, 0.0])
arrgeom.append([0.0, 100.0, 0.0])
arrgeom.append([-70.7106781, 70.7106781, 0.0])
arrgeom.append([-100.0, 0.0, 0.0])
arrgeom.append([-70.7106781, -70.7106781, 0.0])
arrgeom.append([0.0, -100.0, 0.0])
arrgeom.append([70.7106781, -70.7106781, 0.0])

# Filter bank parameters
R     = 2**r
D     = M / R # frame shift

sampleRate          = 16000
outSampleRate       = 16000
alpha       = 0.2 # for post-filtering

# Load filter bank prototype
# Read analysis prototype 'h'
protoFile = './h-M=%d-m=%d-r=%d.txt' %(protoPath, M,
    m, r)
fp = open(protoFile, 'r')
h_fb = pickle.load(fp)
fp.close()

# Read synthesis prototype 'g'
protoFile = '%s/g-M=%d-m=%d-r=%d.txt' %(protoPath, M,
     m, r)
fp = open(protoFile, 'r')
g_fb = pickle.load(fp)
```

```
fp.close()

# output file name
filename  = './wav.PF%d_a%0.2f_M=%d-m=%d-r=%d/output.
    wav' %(postfilterType, alpha, M, m, r)

# Init the beamformer object
pBeamformer = SubbandGSCPtr( fftLen=M, halfBandShift=
    False )
output = ZelinskiPostFilterPtr(
    PyVectorComplexFeatureStreamPtr(pBeamformer),
    fftLen, alpha, postfilterType )

# Build the analysis chain
sampleFeats = []
analysisFBs = []
for chX in range(chanN):
    sampleFeature = SampleFeaturePtr( blockLen=D,
        shiftLen=D, padZeros=True )
    sampleFeats.append(sampleFeature)
    analysisFB = OverSampledDFTAnalysisBankPtr(
        sampleFeature, prototype=h_fb, M=M, m=m, r=r )
    analysisFBs.append(analysisFB)
    pBeamformer.setChannel(analysisFB)

# Init synthesisFB
synthesisFB =  OverSampledDFTSynthesisBankPtr(
    PyVectorComplexFeatureStreamPtr(output), prototype
    =g_fb, M=M, m=m, r=r )
```

```
# read multi-channel data.
for chanX in range(chanN):
    nextFile = '%s%02d.wav' %(inputFilePrefix, chanX)
    if not os.path.exists(nextFile):
        print 'Could not find file %s' %nextFile
    print 'Loading file %s' %nextFile
    sampleFeats[chanX].read(nextFile, samplerate =
        sampleRate)

# calculate time delays and beamformer's weights.
delays1 = calcDelaysPolar2( azimuth, elevation, array
    (arrgeom) )
pBeamformer.calcGSCWeights( sampleRate, delays1 )

# Here we go....
wavebuffer = []
# The methods next() of the C++ modules are called at
    each step.
# In this case, SampleFeature.next(),
# OverSampledDFTAnalysisBank.next(),
# SubbandGSC.next() and OverSampledDFTSynthesisBank.
    next()
# are called.
for b in synthesisFB:
    output.setBeamformer( pBeamformer )
    wavebuffer.extend(deepcopy(b))

# Write WAV file to disk
storewave = array(wavebuffer, Float)
if not os.path.exists(os.path.dirname(filename)):
```

```
            os.makedirs(os.path.dirname(filename))
        wavefile = wave.open(filename, 'w')
        wavefile.setnchannels(1)
        wavefile.setsampwidth(2)
        wavefile.setframerate(int(outSampleRate))
        wavefile.setnframes(len(storewave))
        wavefile.writeframes(storewave.astype('s').tostring()
            )

        wavefile.close()
        pBeamformer.reset()

try:
    opts, args = getopt.getopt(sys.argv[1:], "hi:s:p:", [
        "help", "input=", "pf="])
except getopt.GetoptError:
    # print help information and exit:
    sys.exit(2)



# parameters for filter banks
M = 256 # the number of subbands
m = 2   # filter length factor
r = 1   # decimation factor
# the direction of arrival of a sound source
azimuth   = 0.0
elevation = 0.0
# path for multiple wave files
inputFilePrefix   = 'inputdir/test-ch'
# which type of post-filtering is used
```

```
postfilterType     = 0 # 0 (no post-filter), 2 (Zelinski
    with abs() real operator), 8 ( use beamformer output )

for o, a in opts:
    if o in ("-h", "--help"):
        sys.exit()
    elif o in ("-i", "--input"):
        inputFilePrefix = a
    elif o in ("-p", "--pf"):
        postfilterType = int(a)

beamform( inputFilePrefix, M, m, r, postfilterType )
```