ENHANCEMENT OF ADAPTIVE DE-CORRELATION FILTERING SEPARATION MODEL FOR ROBUST SPEECH RECOGNITION

A Dissertation

Presented to

the Faculty of the Graduate School

University of Missouri - Columbia

In Partial Fulfillment for the Degree

Doctor of Philosophy

by

RONG HU

Dr. Yunxin Zhao, Dissertation Supervisor

May 2007

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled

ENHANCEMENT OF ADAPTIVE DE-CORRELATION FILTERING SEPARATION MODEL FOR ROBUST SPEECH RECOGNITION

Presented by Rong Hu,

a candidate for the degree of Doctor of Philosophy,

and hereby certify that in their opinion it is worth acceptance.

____

Dr. Yunxin Zhao

____

Dr. Xinhua Zhuang

____

Dr. Hongchi Shi

____

Dr. Wenjun Zeng

____

Dr. Dominc Ho

DEDICATION

This dissertation is dedicated to my wife, Hui, and my daughter, Angie (Yuewen).

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Yunxin Zhao. Her scientific insight has significantly influenced this work. Her willingness to support my work and her guidance throughout my Ph.D. studies have helped me develop research skills in a cooperative team environment. My appreciation goes to my committee members, Profs. Xinhua Zhuang, Hongchi Shi, Wenjun Zeng, and Dominic K.C. Ho, for their suggestions and supervision. I greatly appreciate their time and input to this dissertation.

I would like to thank my fellow lab mates, Dr. Xiaodong He, Mrs. Chuanhong Ma, Dr. Xiaolong Li, Dr. Rusheng Hu, Dr. Jian Xue, Mrs. Lili Che, Mrs. Xiaojia Zhang, and Mr. Yi Zhang for their discussions, collaboration, exchange of ideas, and their helps both in my Ph.D. study and in the daily-life. My thank also goes to Dr. Kuan-Chieh Yen, for his information and suggestions during our discussions at several conferences.

I am deeply indebted to my family, especially my wife Hui Zhao and my parents, Bogui Hu and Yufang Wang. Their understanding, encouragement, and love supported my work and study greatly.

Part of the experimental data comes from ATR Japan and special thanks to Dr. Nakamura for providing the data.

Contents

List of Tables

List of Figures

ABSTRACT

The development of automatic speech recognition (ASR) technology has enabled an increasing number of applications, such as dictation, call routing, broadcast news and medical transcriptions, and voice control, etc. However, the robustness of ASR under real acoustic environments still remains to be a challenge for practical applications. Interfering speech and background noise have severe degrading effects on ASR. Speech source separation separates target speech from interfering speech but its performance is affected by adverse environmental conditions of acoustical reverberation and background noise. This dissertation works on the enhancement of a speech source separation technique, namely adaptive de-correlation filtering (ADF), for robust ASR applications.

To overcome these difficulties and develop practical ADF speech separation algorithms for robust ASR, enhancement and improvement are introduced in several aspects. From the perspectives of speech spectral characteristics, the procedures of pre-whitening are applied to flatten the long-term spectral tilt for the improvement of adaptation robustness and decrease of convergence ADF estimation error. To speedup convergence rate, block-iterative implementation and variable step-size (VSS) methods are proposed. To exploit scenarios where multiple pairs of sensors are available, multi-ADF post-processing is developed. To overcome the limitations of ADF separation model under background noise, procedures of noise-compensation

(NC) and adaptive speech enhancement are proposed for the achievement of improved robustness in diffuse noise.

Speech separation simulations and speech recognition experiments are carried out based on TIMIT database and ATR acoustic measurement database. Evaluations of the methods presented in this dissertation demonstrate significant improvement of performances over baseline ADF algorithm in speech separation and recognition.

Chapter 1

INTRODUCTION

In this dissertation, we study the enhancement techniques for the adaptive decorrelation filtering (ADF) sound source separation model with applications in robust speech recognition.

In this chapter, we present the background for the basics of ADF model and for the development of related algorithms. The problem of speech separation will be described in the scenario of automatic speech recognition (ASR) and the state of the art for speech separation will be discussed. After the introduction of the basic ADF model and the baseline ADF algorithm, the difficulties that affect the ADF model will be discussed.

1.1 Robust ASR and Speech Separation

Over the years, efforts have been made to improve the accuracies of automatic speech recognition [1]. However, the robustness of ASR system still remains to be a difficult and important problem, despite the emerging ASR applications and products for specific scenarios in the industry. This is because in real environments, especially for hands-free applications of ASR, speech signals captured by microphones will have distortions relative to their statistical parametric representations in the acoustic models used by speech recognizers that were estimated from clean training data.

Among many factors affecting ASR system performances, interfering speech from speakers who speak simultaneously with the target speaker, background noises, and room reverberation cause serious degradation. This drastic drop of performance is one of the major obstacles in deploying a commercial recognition system in normal environments. Therefore, for successful application of ASR in real scenarios, it is desirable to pick up a speech signal of interest by separating it from interfering signals. Speech separation and enhancement procedures are crucial to improvement of speech quality and ASR performances.

The topics of making ASR system more robust have been intensively studied within the framework of hidden Markov model (HMM) [1–3] and statistical pattern recognition as well as from the perspectives of removing the effects of background noises or reverberations on recognition [4, 5]. Most of the researches focus on each or both of these effects and attempt to overcome their influence on ASR, either in the feature domain at the frontend or in the acoustic modeling domain at the backend.

- Effect of background noise. The interference of noise causes feature vectors and acoustic model parameters to deviate from their training expectations. Among many types of noises, diffuse noise and sensor noises are very common ones. Current methods dealing with noise effects for ASR include speech enhancement and noise reduction [6, 7], feature domain compensation techniques, acoustic model adaptation, etc.

- Effect of reverberations. Unlike close-talk or free-space recordings, speech signals acquisition by far field recording microphones in an enclosed acoustic environment exhibit reverberation due to increased speaker-microphone distances.

> The effect of reverberation causes both spectral shaping (coloration) and temporal smearing for the time-frequency distribution of speech signals [8].

Compared to studies on noisy and reverberant environments, the problem of removing the effects of interference speech on the recognition of target speech still needs more attention and efforts. To accomplish the task of speech separation, several types of methods were proposed utilizing different information, such as single channel speech segregation which does sound scene analysis by utilizing the knowledge on human abilities to sort auditory components into individual sources and/or by relying on information of speech structures [9], and blind source separation (BSS) which uses information provided by multiple channels of speech signals. The former topic is more challenging.

Many BSS algorithms [10–34] were developed to separate simultaneous co-channel speech sources with multiple microphone recordings of speech mixtures. Many of the earlier research efforts in the broader field of BSS and independent component analysis [10] (ICA) were mainly on the much simpler case of instantaneous signal mixture [35]. The focus soon shifted to convolutive mixture separations afterwards. Because of its potential abilities of improving speech quality and ASR system performance under convolutive mixing environment, blind separation of co-channel speech signals has become an active research topic in recent years.

Figure 1.1 shows an example of multiple-input-multiple-output (MIMO, with M=2) convolutive speech mixture and blind signal separation system. In the mixture model, the acoustic paths between speech sources S_j and the i-th microphone is denoted as H_{ij}, the Y_i' s are speech mixtures. Figure 1.2 gives the example waveforms of the source and mixture speeches. In BSS model, the separation filters W_{ij} should be estimated, either online or in offline batch processing, from the mixture signals only, without observation of speech sources.

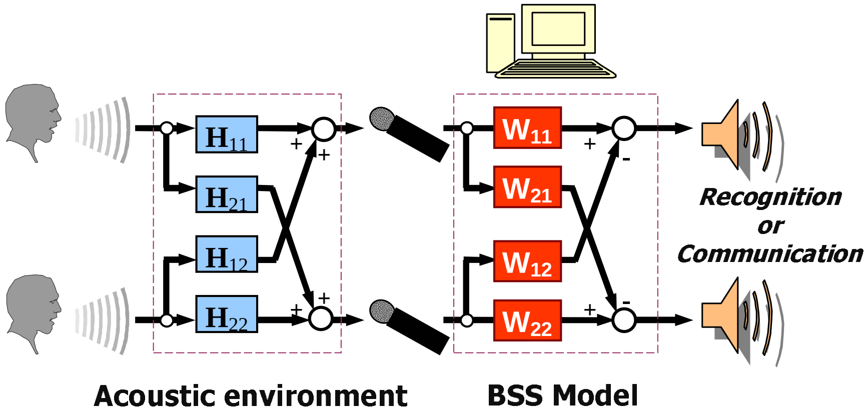


Figure 1.1: TITO speech mixing and BSS separation model

By utilizing assumptions on the properties of source signals, BSS methods can be used to suppress interfering speech sources and enhance target sources for both ASR and speech communication. The criteria used for BSS algorithms include: independence, non-Gaussianality, mutual information, contrast function, time-frequency sparseness [15] of speech sources, and decorrelation between outputs, etc [10,11]. The mutual information based methods seek a minimization of the amount of shared information between separation outputs [11]. The methods of maximization of some statistical contrast function, e.g., normalized kurtosis, indicate when an output of the separation system contains only one source signal. The decorrelation-based methods [17–34] perform a diagonalization of the spatial correlation matrix of BSS output signals. One typical example of BSS method is the frequency domain ICA algorithm. It transforms time-domain convolutive mixture signals into Fourier domain to perform instantaneous ICA in each frequency bin [16]. This method may suffer from the problem of permutation, and post-processing are necessary to group together frequency components of the same sources.

Time-domain adaptive de-correlation filtering (ADF) [17,20–23] is another promising approach for BSS. Compared with other existing BSS and ICA methods, the ADF model has the advantage of simplicity in both structure and implementation. However, in the presence of background noise and under acoustic conditions of long

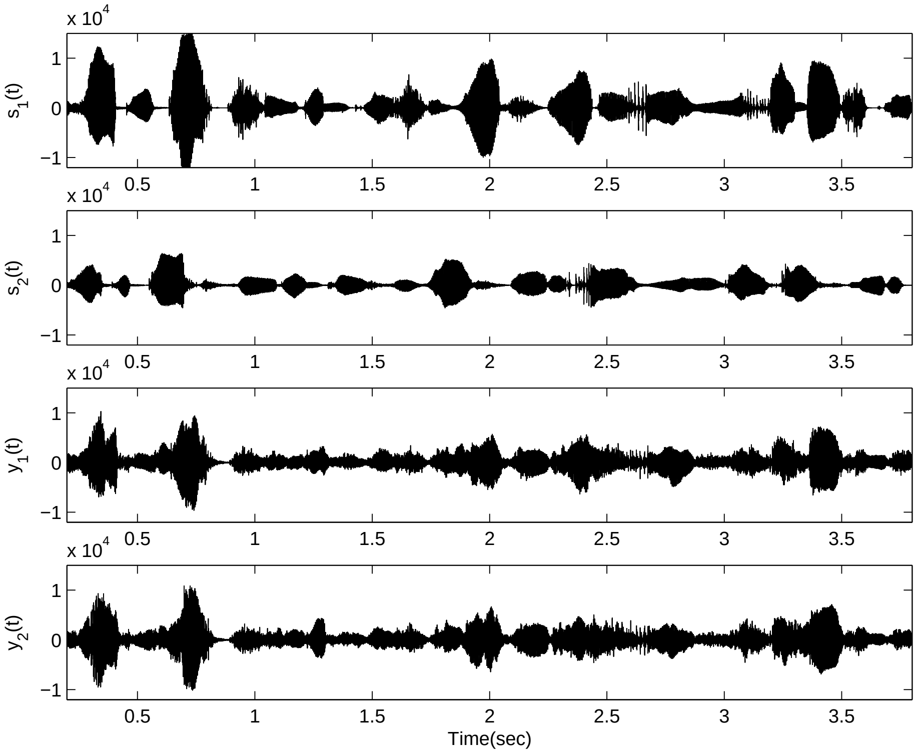


Figure 1.2: Waveform examples of source and mixture speeches, where sources are $s_1(t)$ and $s_2(t)$, and mixtures are $y_1(t)$ and $y_2(t)$.

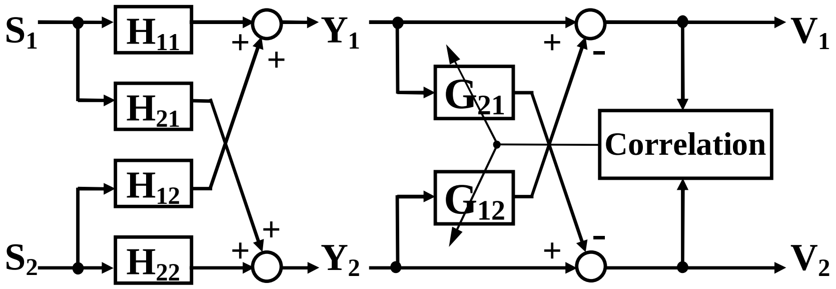


Figure 1.3: Speech mixing and ADF separation model

reverberation, the separation performances of basic ADF algorithm are unsatisfactory and convergence speed still needs to be improved for online applications.

In this dissertation work, several techniques are developed to enhance the performances of ADF separation model for robust ASR applications, and to improve speech recognition accuracies under adverse conditions of diffuse noise.

1.2 ADF Speech Separation Model and Algorithms

Fig. 1.3 shows the block diagram of the two-input-two-output (TITO) ADF separation model together with the speech mixing model. Under the strict causal FIR constraint for cross-coupling filters, the degeneracy problem discussed in [30] can be avoided [24]. The adaptations of separation filters G_{ij}'s are based on output decorrelation. In the following discussions, we will base our analysis on the models with two inputs and two outputs only and assume that the strict causal FIR constraint in our system can be ensured by appropriate configurations of speaker sources and microphones.

1.2.1 Basic ADF algorithm

The speech convolutive mixture model can be described by

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} 1 & \frac{H_{12}(z)}{H_{22}(z)} \\ \frac{H_{21}(z)}{H_{11}(z)} & 1 \end{bmatrix} \cdot \begin{bmatrix} H_{11}(z)S_1(z) \\ H_{22}(z)S_2(z) \end{bmatrix} \tag{1.1}$$

and the ADF separation model

$$\begin{bmatrix} V_1(z) \\ V_2(z) \end{bmatrix} = \begin{bmatrix} 1 & -G_{12}(z) \\ -G_{21}(z) & 1 \end{bmatrix} \cdot \begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} \tag{1.2}$$

achieves the goal of speech separation and obtains filtered versions of source speech signals when the filters reach the values of

$$G^o_{ij}(z) = H_{ij}(z)/H_{jj}(z), \tag{1.3}$$

where $G^o_{ij}(z)$ is the ideal separation filter.

The separation filtering procedure is represented in time domain by

$$v_i(t) = y_i(t) - \sum_{n=0}^{N-1} g^{(t)}_{ij}(n) y_j(t-n), i, j = 1, 2_{,i \neq j}\ , \tag{1.4}$$

which corresponds to the forward model for separation derived in [17]. In addition to the time-domain online processing ADF algorithms discussed in this thesis, several other types of methods exist for the solution of separation filter parameters $g_{ij}(n)$'s. In [24], a two step batch processing algorithm was proposed that used frequency domain eigen-decompositions for an initial estimate of separation filter transfer functions and a Monte Carlo algorithm for a refined estimation of separation filters. The computation complexities required by both eigen-decomposition and Monte Carlo optimization, as well as their form of processing in long batches made such method impractical for real application. Lindgren and Broman [28] applied Newton method to the same separation model with another decorrelation criterion function. Their simulations were performed on simple mixing models.

Based on the Robbins-Monro stochastic approximation [36, 37] method, the basic filter adaptation procedure of ADF algorithm was given in [17] as

$$g_{ij}^{(t+1)}(n) = g_{ij}^{(t)}(n) + \mu v_i(t) v_j(t-n), n = 0, \cdots, N-1,_{i \neq j}, \tag{1.5}$$

where N is filter length and μ the adaptation step-size.

The basic ADF algorithm was enhanced and its application extended by Yen and Zhao [20–22] and applied to asssistive listening [23]. A stabilizing input normalized step size was proposed in [20] as

$$\mu(t) = 2\gamma / N \left(\sigma_{y_1}^2(t) + \sigma_{y_2}^2(t)\right), \tag{1.6}$$

based on the stability analysis on the convergence process. The constant gain factor $\gamma(0 < \gamma < 1)$ controls the convergence speed and the denominator of (1.6) is the total short-term power estimate of ADF input speech mixtures. Similar to this technique, an earlier technique of output normalized step-size was used by Thi and Jutten [31]. Yen and Zhao [20] also extended ADF algorithm to MIMO scenarios with number of sources greater than 2 [21] and tested other implementation structures [22].

The performances of speech separation algorithms are evaluated in the current work by two objective measures: gains in target-to-interference-ratios (TIR) before and after ADF separation and the phone recognition accuracy measured in speech recognition tests. The TIR measure is defined by

$$\Delta TIR = TIR_{out} - TIR_{in}, \tag{1.7}$$

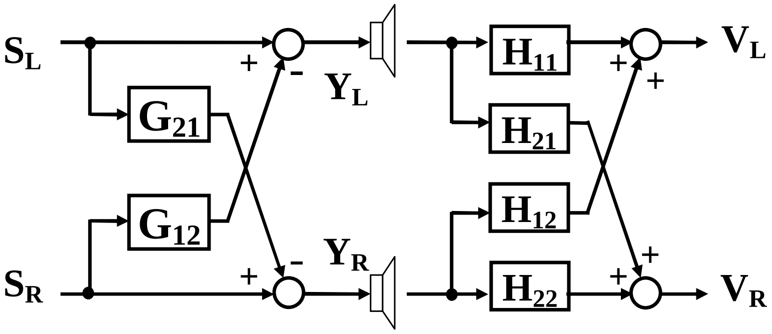


Figure 1.4: The cross-talk cancellation system for loudspeaker reproduction of 3D sound: S_L and S_R are left and right sources; Y_L and Y_R are left and right loudspeaker signals; V_L and V_R are signals perceived by left and right ears.

where TIRs are defined by $10log_{10}P_T/P_I$ in dB values, with P_T and P_I the power of target and interference speech signals, respectively.

1.2.2 Other related topics

The ADF model for speech separation is similar or related to several other research topics in audio and/or speech processing, such as null beam former, adaptive noise cancelation (ANC), transaural cross-talk cancelation [38–41] for 3D sound reproduction with loudspeakers, and speech segregation [9].

The similarities of ADF source separation algorithm to ANC was compared in the early development and analysis of ADF in the name of symmetric adaptive decorrelation (SAD) algorithms by Gervern and Compernolle [25]. In fact, for the TITO system in Figure 1.3, each decoupling subsystem that obtains one specific source speech estimate has the same structure as ANC model with leakages of the target signal in the reference input [25]. Their main difference lies in adaptation algorithms, although still bear similarities. Yen and Zhao [20] analyzed such a similarity and proposed an algorithm switching method to change between ADF and LMS adaptations for better estimation of separation filters.

When we consider the modeling topology, the loudspeaker crosstalk canceler [39, 42] also shares the same model structure except that it is performed prior to acoustic mixing through stereo loudspeakers, as illustrated in Figure 1.4. When viewed from the perspective of physical essence, both the ADF separation system and crosstalk canceler could be regarded as multiple null beam-formers of doing filter-and-subtract processing. In a very special case where each coupling filter only contains one non-zero coefficient at a certain time-delay, the ADF model in Figure 1.3 could be viewed as containing two simplest fixed null beam-formers that perform delay-and-subtraction. However, the differences between these techniques and ADF processing are also obvious because they could not be classified as blind processing techniques. The transaural crosstalk cancellation usually requires accurate information of the mixing system, such as the listener-loudspeaker position provided by a head-tracking system [43]. Microphone array beam-forming methods usually need the estimation of look direction and array errors such as sensor mismatch and mis-steering [11] will affect the beam-forming performance, especially for adaptive beamforming. As a comparison, ADF does not require knowledge of geometrical configurations of the mixing system. In fact, ADF model can tolerate certain levels of microphone mismatches. If we absorb the speech acquisition channel distortions and sensor mismatches into transfer functions $M_i(z)$'s prior to ADF processing, the mixing system takes the form

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} M_1(z) & 0 \\ 0 & M_2(z) \end{bmatrix} \cdot \begin{bmatrix} 1 & \frac{H_{12}(z)}{H_{22}(z)} \\ \frac{H_{21}(z)}{H_{11}(z)} & 1 \end{bmatrix} \cdot \begin{bmatrix} H_{11}(z)S_1(z) \\ H_{22}(z)S_2(z) \end{bmatrix}, \qquad (1.8)$$

which is equivalent to

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} 1 & \frac{H_{12}(z)M_1(z)}{H_{22}(z)M_2(z)} \\ \frac{H_{21}(z)M_2(z)}{H_{11}(z)M_1(z)} & 1 \end{bmatrix} \cdot \begin{bmatrix} H_{11}(z)M_1(z)S_1(z) \\ H_{22}(z)M_2(z)S_2(z) \end{bmatrix}. \qquad (1.9)$$

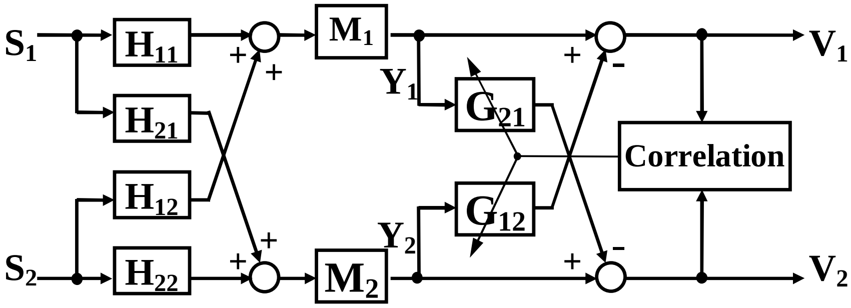


Figure 1.5: ADF model considering sensor mismatch.

If we assume that the distortions $M_i(z)$ will not introduce total loss of information in individual frequency bands, e.g., zeroing out some frequency components, we could still use ADF to obtain separated speech, as long as the mismatches between $M_1(z)$ and $M_2(z)$, especially phase differences, are within certain limits. Following the same analysis as (1.1)-(1.3), the overall ADF separation model can be obtained with the ideal separation filters

$$\mathbf{G}_{ij}^{o}(z) = \frac{H_{ij}(z)M_i(z)}{H_{jj}(z)M_j(z)}, i, j = 1, 2, i \neq j, \tag{1.10}$$

which absorbs the effects of channel distortion and mismatch. This property obviates the need for array calibration in ADF model. When ADF filters take the value of (1.10), the outputs $V_i(z) = H_{ii}(z)M_i(z)S_i(z)$'s are transformed versions of the original speech sources.

As far as the task of separating speech signal is concerned, there is another perspective, i.e., utilizing the properties and structures of the speech signal itself instead of relying merely on the uncorrelated assumption for speech sources in ADF. Such methods include speech segregation [9], computational auditory scene analysis (CASA) [9, 44], or methods based on speech modeling [45], and ICA/BSS algorithms that integrate extra information about the sources [15]. These algorithms usually take additional assumptions on the properties of speech signals. In [15], the spectral non-overlapping structure was assumed in addition to source independence assumption for ICA. The techniques of CASA are based on within-signal cues of speech such as harmonicity. A comparison between CASA and BSS methods was given in [46]. Rules derived from the studies of speech structures are used to group mixture components into individual sources. Although some of these algorithms could separate speech in single mixture they usually make a very strong assumption and are computationally complicated. Some of them even requires off-line training from data for the separation system. For example, the speech separation system of [45] achieved a high separation performance with the help of multiple acoustical models and required the training data from specific speakers prior to separation. Generally speaking, the blindness nature of the separation problem can be helped by introducing additional information about the sources. The above mentioned several types of speech separation methods are based on much stronger assumptions, or equivalently, utilized more information about sources. In fact, the enhancement of ADF separation model for speech recognition applications can also be helped from the perspective of partially reducing the blindness to the source signals or the mixing system, with a slight increase of complexity. Since one of the advantages of ADF separation model is low cost in implementation, we will keep this merit by introducing effective and computational efficient methods for its enhancement in the rest work of this thesis.

1.3 Difficulties and Problems in Real Applications

The applications of ADF model for speech separation in practical scenarios is complicated by the acoustic environment. Among the acoustic interferences, reverberation and background noise are two major problems that deteriorate separation and speech recognition performances. The effects of environment on speech recognition has been discussed in 1.1. In addition to such effects, the degradation of separation performance itself will cause degradation to the subsequent recognition processing. It is a challenging task to achieve a satisfactory performance for the overall separation and recognition system in these adverse conditions. To make the ADF model practical for ASR applications, the environmental effects on ADF separation model and algorithms should also be considered.

1.3.1 Separation in noisy environment

Real acoustic environments have many kinds of interferences and noises, and interfering speech and background noise often appear together, making the separation and recognition problems more difficult. To improve noise robustness of ASR system, speech model compensation for noisy environment at the back-end, or techniques of noise-robust features and speech enhancement algorithms at the front-end can be used [4]. Although speech enhancement techniques are intended to recover the waveform of clean speech embedded in noise and not usually directly aim at improving speech recognition performance, we will investigate the possibilities of combining ADF separation algorithm with speech enhancement techniques because ADF processing is also performed directly on waveform and not directly related to any speech acoustic models. Many speech enhancement techniques [47] already exist for single or multichannel of speeches, e.g., spectral subtraction [6], etc. However, these techniques of speech separation and speech enhancement may interact with each other. Therefore, we still need to look into appropriate integration methods by considering the properties of both ADF and speech enhancement techniques.

1.3.2 Separation in reverberant conditions

Room reverberation affects many speech and audio signal processing procedures. One measure that describes the reverberation characteristics of a room is the reverberation time, T_{60}, defined as the length of time it takes for the energy in the sound field in a room to decay by 60dB after the sound excitation source is suddenly turned off [48]. Due to the complexities caused by both reflection and defraction, the actual sound propagations in the room is complicated. The interactions between sound propagation and room environmental factors, such as room geometries, absorbance and shapes of reflector materials (walls, furniture, etc.), air temperature and density distributions, etc., make the effects of reverberation more complicated.

The reverberant room effect on speech can be characterized as convolution of the speech signal with the acoustic path room impulse response, and a microphone far away from the speakers receives a filtered version of the source signal. Fig. 1.6 shows an example of room impulse response. The room impulse response is invertible only when the response is minimum phase [49]. So it is difficult to blindly (i.e., without knowledge of impulse responses) and totally reconstruct the speech source signals by doing channel inversion [8]. Many ICA or BSS algorithms are based on independence assumptions between sources. However, the reverberated versions of sources are also independent. As pointed out by [12] and [13], some blind deconvlution based methods over-whitened sources in their efforts to achieve channel inversion. Therefore, it is difficult to achieve blind reconstruction, and for the discussions in this dissertation, we will make no attempts to do total reconstruction of sources by ADF model iteself.

1.4 Contributions

The algorithms that we present in this dissertation extend the state of the art and address the difficulties and problems for the application of ADF model in robust speech recognition. The methods developed include pre-processing for the improvement of ADF stability and convergence, the methods of integrating and post-processing multiple ADF signals for multiple microphones, the algorithms that speedup convergence rate, such as block-iterative and variable step-size methods, and noise compensation methods and corresponding fast algorithms for the enhancement of separation performances in noisy scenarios. In addition, adaptive speech enhancement post-processing module and their fast algorithms are also proposed for the integration with noise-compensated ADF, so that both improved speech separation and noise reduction in separated speech are achieved.

1.5 Document Organization

The remainder of this dissertation documents the details of our algorithms and the above contributions. Chapter 2 introduces the mathematical representations for ADF model. Based on such derivations, the enhancement algorithms in this thesis are developed. Chapter 3 describes enhancement of ADF model for robust speech recognition from the perspective of convergence speedup, post-processing, and multi-ADF model integration. In Chapter 4, the methods of variable step-size (VSS) are discussed. Chapters 5 and 6 provide noise-compensated ADF (NCADF) and adaptive enhancement algorithms for dealing with the application of ADF in noisy scenario. Comparative speech recognition experiments and selected results are presented in Chapter 7 for evaluating the techniques proposed. Finally Chapter 8 concludes the thesis work and discusses potential future work directions.

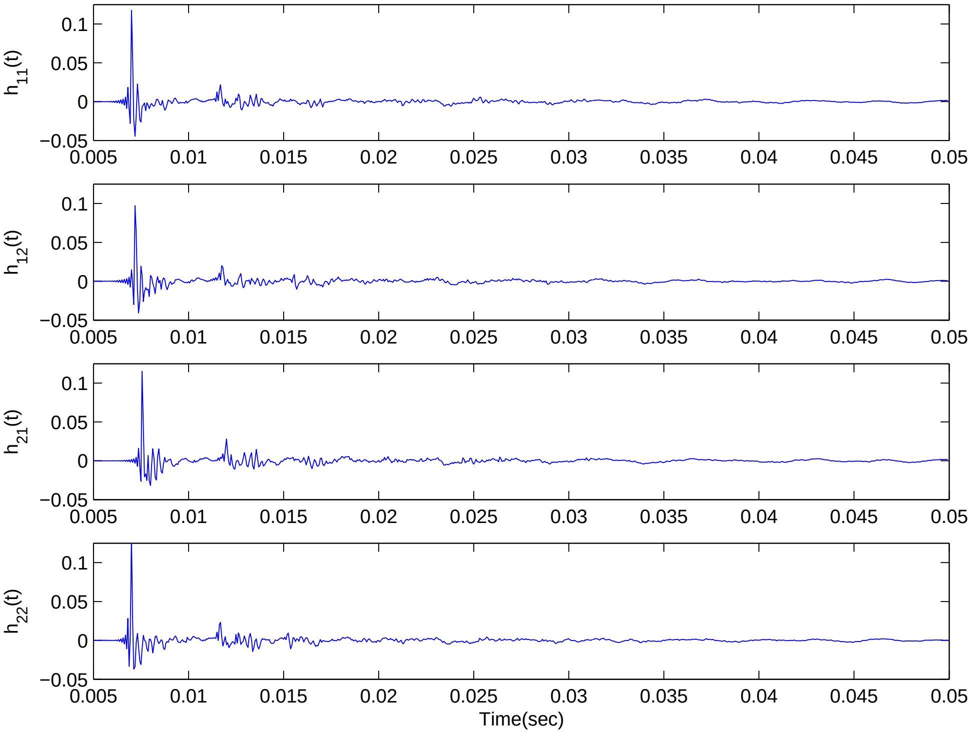


Figure 1.6: An example of room impulse responses (truncated for the beginning $30ms$).

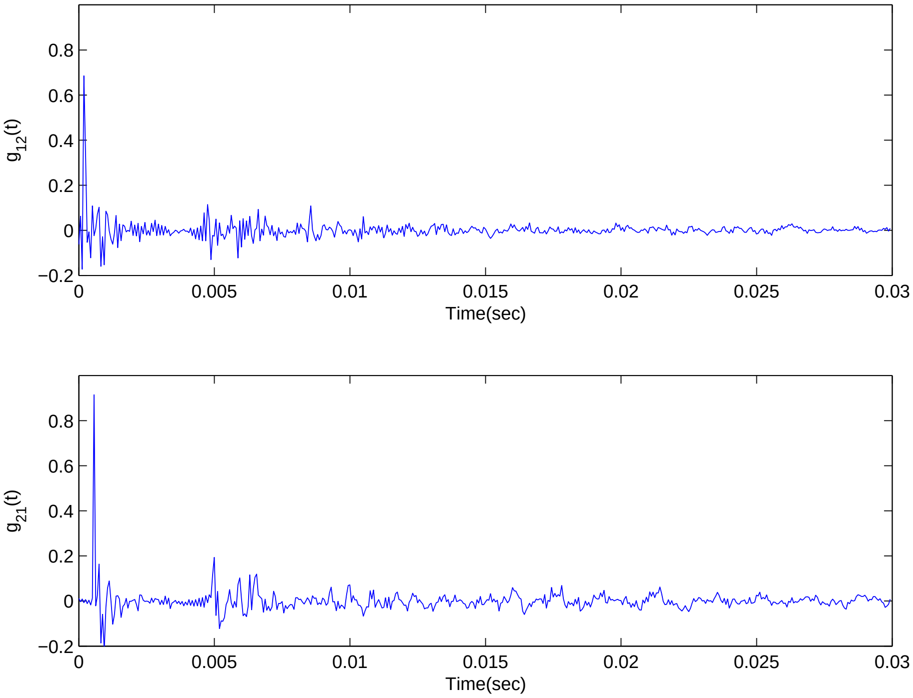


Figure 1.7: An example of ideal cross-coupling ADF filters.

Chapter 2

VECTOR FORMULATIONS AND ANALYSIS OF ADF SYSTEM

In this chapter, we will present the vector formulation and analysis of ADF system. Based on the mathematical framework established here, further improvement on ADF separation model could be derived. Generally, vector formulations and analyses of a discrete adaptive signal processing system use either the Hankel data matrix or the Toeplitz system matrix representation.

2.1 Vector Formulations

For the ADF separation model shown in Figure 1.3, the N-point cross-coupling filters

$$\mathbf{g}_{ij} = [g_{ij}(0), g_{ij}(1), \cdots, g_{ij}(N-1)]^T, i, j = 1, 2, i \neq j, \tag{2.1}$$

are to be adaptively identified, with superscript $(\cdot)^T$ denoting matrix and vector transposition. The following notations will be used in the rest parts of the dissertation: vector variables are in bold lower case, matrices are in bold upper case, $\mathbf{I}$ is the identity matrix, $E\{\}$ is for expectation, and "*" for convolution. Speech and noise signal vectors contain N consecutive samples up to the current time t; their $(2N-1)$-point counterparts are marked with tilde. The cross-correlation vector between a signal scalar $a(t)$ and a signal vector $\mathbf{b}(t)$ is denoted as $\mathbf{r}_{a\mathbf{b}} = E\{a(t)\mathbf{b}(t)\}$, and the correlation matrix formed by signal vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{R}_{\mathbf{ab}} = E\{\mathbf{a}(t)\mathbf{b}^T(t)\}$.

With the above notations, the basic ADF separation filtering and adaptation procedures (1.4)-(1.5) can be written in vector forms.

2.1.1 Toeplitz matrix representation and I/O relations

With the Toeplitz system representation, the input-output (I/O) relation of ADF system ((1.4) for clean (noise-free) speech mixtures can be put in a vector form

$$\mathbf{v} = \mathbf{G} \cdot \tilde{\mathbf{y}}, \tag{2.2}$$

where $\tilde{\mathbf{y}} = \left[\tilde{\mathbf{y}}_1^T(t), \tilde{\mathbf{y}}_2^T(t)\right]^T$ and $\mathbf{v} = \left[\mathbf{v}_1^T(t), \mathbf{v}_2^T(t)\right]^T$ are $(4N-2)\times 1$ input vector and $2N \times 1$ output vector, respectively, and

$$\mathbf{G} = \begin{bmatrix} [\mathbf{I}_N \ \mathbf{0}_{N\times(N-1)}] & -\mathbf{G}_{12} \\ -\mathbf{G}_{21} & [\mathbf{I}_N \ \mathbf{0}_{N\times(N-1)}] \end{bmatrix} \tag{2.3}$$

is the $2N \times (4N-2)$ system matrix. Specifically,

$$\mathbf{G}_{ij} = \begin{bmatrix} g_{ij}(0) & g_{ij}(1) & \cdots & g_{ij}(N{-}1) & 0 & \cdots & 0 \\ 0 & g_{ij}(0) & g_{ij}(1) & \cdots & g_{ij}(N{-}1) & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & g_{ij}(0) & g_{ij}(1) & \cdots & g_{ij}(N{-}1) \end{bmatrix}_{N\times(2N-1)}, \tag{2.4}$$

$\mathbf{v}_i(t) = [v_i(t), \cdots, v_i(t-N+1)]^T$, and $\tilde{\mathbf{y}}_i(t) = [y_i(t), \cdots, y_i(t-2N+2)]^T$.

The ADF I/O relation in terms of input and output correlation matrices is

$$\mathbf{R_{vv}} = E\{\mathbf{v}(t) \cdot \mathbf{v}^T(t)\} = \mathbf{G}\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}\mathbf{G}^T, \tag{2.5}$$

with

$$\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} = E\{\tilde{\mathbf{y}}(t) \cdot \tilde{\mathbf{y}}^T(t)\}, \tag{2.6}$$

By partitioning the matrices $\mathbf{R_{vv}}$ and $\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}$ into blocks,

$$\mathbf{R_{vv}} = \begin{bmatrix} \mathbf{R}_{\mathbf{v}_1\mathbf{v}_1} & \mathbf{R}_{\mathbf{v}_1\mathbf{v}_2} \\ \mathbf{R}_{\mathbf{v}_2\mathbf{v}_1} & \mathbf{R}_{\mathbf{v}_2\mathbf{v}_2} \end{bmatrix}, \tag{2.7}$$

$$\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} = \begin{bmatrix} \mathbf{R}_{\tilde{\mathbf{y}}_1\tilde{\mathbf{y}}_1} & \mathbf{R}_{\tilde{\mathbf{y}}_1\tilde{\mathbf{y}}_2} \\ \mathbf{R}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_1} & \mathbf{R}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_2} \end{bmatrix}, \tag{2.8}$$

and following basic matrix algebra, the off-diagonal and diagonal component blocks are derived as

$$\mathbf{R}_{\mathbf{v}_i\mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_i\mathbf{y}_j} - \mathbf{R}_{\mathbf{y}_i\tilde{\mathbf{y}}_i}\mathbf{G}_{ji}^T - \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{y}_j} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_i}\mathbf{G}_{ij}^T, \tag{2.9}$$

$$\mathbf{R}_{\mathbf{v}_i\mathbf{v}_i} = \mathbf{R}_{\mathbf{y}_i\mathbf{y}_i} - \mathbf{R}_{\mathbf{y}_i\tilde{\mathbf{y}}_i}\mathbf{G}_{ij}^T - \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{y}_i} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j}\mathbf{G}_{ij}^T. \tag{2.10}$$

Another useful relation of signal second order statistics is

$$\mathbf{R_{yv}} = \mathbf{R}_{\mathbf{y}\tilde{\mathbf{y}}}\mathbf{G}^T, \tag{2.11}$$

with diagonal and off-diagonal components

$$\mathbf{R}_{\mathbf{y}_i\mathbf{v}_i} = \mathbf{R}_{\mathbf{y}_i\mathbf{y}_i} - \mathbf{R}_{\mathbf{y}_i\tilde{\mathbf{y}}_j}\mathbf{G}_{ij}^T, \tag{2.12}$$

$$\mathbf{R}_{\mathbf{y}_i\mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_i\mathbf{y}_j} - \mathbf{R}_{\mathbf{y}_i\tilde{\mathbf{y}}_i}\mathbf{G}_{ji}^T, \tag{2.13}$$

where $\mathbf{R_{yv}} = E\{\mathbf{yv}^T\}$, with $\mathbf{y} = \left[\mathbf{y}_1^T(t), \mathbf{y}_2^T(t)\right]^T$ and $\mathbf{y}_i(t) = [y_i(t), \cdots, y_i(t-N+1)]^T$.

The system output correlation matrix is also related to system input-output cross-correlation matrix by

$$\mathbf{R_{vv}} = \mathbf{G}\mathbf{R}_{\tilde{\mathbf{y}}\mathbf{v}} = \mathbf{R_{yv}} - \begin{bmatrix} \mathbf{0}_{N\times(2N-1)} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{0}_{N\times(2N-1)} \end{bmatrix} \cdot \mathbf{R}_{\tilde{\mathbf{y}}\mathbf{v}}. \tag{2.14}$$

with components

$$\mathbf{R}_{\mathbf{v}_i\mathbf{v}_i} = \mathbf{R}_{\mathbf{y}_i\mathbf{v}_i} - \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{v}_i}, \tag{2.15}$$

$$\mathbf{R}_{\mathbf{v}_i\mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_i\mathbf{v}_j} - \mathbf{G}_{ji}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{v}_j}. \tag{2.16}$$

2.1.2 Data matrix representation

Defining the Hankel data matrix of the input signals as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix}_{2N\times N}, \tag{2.17}$$

where,

$$\mathbf{Y}_i = \left[\mathbf{y}_i(t), \mathbf{y}_i(t-1), \cdots, \mathbf{y}_i(t-N+1)\right], \tag{2.18}$$

we can represent the ADF system I/O relations in the following vector form

$$\begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}_{2\times N} = \begin{bmatrix} \mathbf{e}_1^T & -\mathbf{g}_{12}^T \\ -\mathbf{g}_{21}^T & \mathbf{e}_1^T \end{bmatrix}_{2\times 2N} \cdot \mathbf{Y}, \tag{2.19}$$

where the $N \times 1$ vector $\mathbf{e}_1 = [1, 0, \cdots, 0]^T$.

The second-order statistics analysis of the I/O relations based on (2.19) will lead to equivalent results as those based on (2.2), following the relationship between Hankel and Toeplitz matrices. In fact, both representations can be described as special forms of the more general matrix representation

$$\mathbf{V} = \mathbf{G} \cdot \tilde{\mathbf{Y}}, \tag{2.20}$$

where the $N \times N$ output data matrix

$$\mathbf{V} = [\mathbf{v}(t), \mathbf{v}(t-1), \cdots, \mathbf{v}(t-N+1)] \tag{2.21}$$

is also Hankel, and

$$\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}(t), \tilde{\mathbf{y}}(t-1), \cdots, \tilde{\mathbf{y}}(t-N+1)] \tag{2.22}$$

is the $(4N-2) \times N$ matrix augmented from (2.17). Both (2.2) and (2.19) are reduced forms of the matrix representation (2.20): (2.2) is the first column of (2.20) and (2.19) can be extracted from the first and the $(N+1)$-th rows of (2.20). It should be noted that all these forms are based on the assumption of short-term stationarity of the acoustic mixing system, which simplifies the representation of the de-coupling filter matrix/vector by keeping them constant within short-terms. The following discussions will be based on the Toeplitz representation (2.2) for simplicity.

2.2 Solutions and Analysis

2.2.1 Solutions to ADF parameters

Imposing decorrelation conditions on (2.14) to force the off-diagonal blocks of $\mathbf{R}_{\mathbf{vv}}$ to zero, we obtain

$$\begin{bmatrix} \mathbf{0}_{N\times(2N-1)} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{0}_{N\times(2N-1)} \end{bmatrix} \cdot \mathbf{R}_{\tilde{\mathbf{y}}\mathbf{v}} = \mathbf{R}_{\mathbf{yv}} - \begin{bmatrix} \mathbf{R}_{\mathbf{v}_1\mathbf{v}_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathbf{v}_2\mathbf{v}_2} \end{bmatrix} \tag{2.23}$$

which is an over-determined system of equations since the solution of ADF coefficients need $2N$ constraints only. Instead of solving filter coefficients g_{ij}'s by the block-diagonalization of (2.14) directly (as actually did by minimizing a criterion in [50]), we can derive solutions by appropriately selecting a subset of constraints from the off-diagonal blocks in (2.23). Choosing N constraints from the 1st row and N constraints from the $(N+1)$-th rows of (2.23) respectively, the equations for ADF system solutions are obtained as

$$\begin{bmatrix} \mathbf{0}_{N\times N} & \mathbf{R}^T_{\mathbf{y_1v1}} \\ \mathbf{R}^T_{\mathbf{y}_2\mathbf{v}_2} & \mathbf{0}_{N\times N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{g}_{12} \\ \mathbf{g}_{21} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{y_2\mathbf{v}_1} \\ \mathbf{r}_{y_1\mathbf{v}_2} \end{bmatrix} \tag{2.24}$$

The least-square solution of Eq. (2.24)

$$\arg\min \|\mathbf{R}^T_{\mathbf{y}_j\mathbf{v}_j}\mathbf{g}_{ij} - \mathbf{r}_{y_i\mathbf{v}_j}\|^2, \tag{2.25}$$

coincides with the least-cross-correlation of ADF outputs because the error vector of (2.24) actually coincides with the cross-correlation vector $\mathbf{r}_{v_i\mathbf{v}_j}$, i.e.,

$$\mathbf{r}_{v_i\mathbf{v}_j} = \mathbf{r}_{y_i\mathbf{v}_j} - \mathbf{R}^T_{\mathbf{y}_j\mathbf{v}_j}\mathbf{g}_{ij}, \tag{2.26}$$

and the least-cross-correlation solution is

$$\arg\min \|\mathbf{r}_{v_i\mathbf{v}_j}\|^2. \tag{2.27}$$

By alternating between the following two cross-correlation minimization steps

$$\begin{aligned} \mathbf{g}_{12}^{opt} &= \arg\min\nolimits_{\mathbf{g}_{12}} J_{12} = \arg\min\nolimits_{\mathbf{g}_{12}} \left(\tfrac{1}{2}\mathbf{r}^T_{v_1\mathbf{v}_2}\mathbf{r}_{v_1\mathbf{v}_2}\right) \\ \mathbf{g}_{21}^{opt} &= \arg\min\nolimits_{\mathbf{g}_{21}} J_{21} = \arg\min\nolimits_{\mathbf{g}_{21}} \left(\tfrac{1}{2}\mathbf{r}^T_{v_2\mathbf{v}_1}\mathbf{r}_{v_2\mathbf{v}_1}\right) \end{aligned} \tag{2.28}$$

the filter parameters can be searched by the gradient descent procedure

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) - \mu(t)\nabla_{\mathbf{g}_{ij}} J_{ij}. \tag{2.29}$$

Assuming independence between the pair of de-coupling filters $\mathbf{g}_{ij}$'s, the gradient vectors $\nabla_{\mathbf{g}_{ij}} J_{ij}$ are derived from the first row of (2.9) as

$$\nabla_{\mathbf{g}_{ij}} J_{ij} = \frac{\partial}{\partial \mathbf{g}_{ij}} \frac{1}{2}\left(\mathbf{r}_{v_i\mathbf{v}_j}^T \mathbf{r}_{v_i\mathbf{v}_j}\right) = \left(\frac{\partial}{\partial \mathbf{g}_{ij}} \mathbf{r}_{v_i\mathbf{v}_j}^T\right) \mathbf{r}_{v_i\mathbf{v}_j} = -\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}. \tag{2.30}$$

Therefore, the gradient based adaptation for the solution of ADF filters is obtained as

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}. \tag{2.31}$$

Alternative methods could also be used to solve (2.30) with varying performances. In fact, the RLS-like algorithm proposed in [17] can be derived from the alternating solution of (2.24) with the following Newton's method

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) - \mu(t)\left(\mathbf{H}\left(J_{ij}\left(\mathbf{g}_{ij}(t)\right)\right)\right)^{-1} \cdot \frac{\partial}{\partial \mathbf{g}_{ij}} J_{ij}, \tag{2.32}$$

using the Hessian matrices implied by (2.30), where the definition of Hessian is

$$\begin{aligned} \mathbf{H}\left(J_{ij}\left(\mathbf{g}_{ij}(t)\right)\right) &= \tfrac{\partial}{\partial \mathbf{g}_{ij}}\left(\nabla_{\mathbf{g}_{ij}} J_{ij}\right)^T = \tfrac{\partial}{\partial \mathbf{g}_{ij}}\left(\tfrac{\partial}{\partial \mathbf{g}_{ij}} J_{ij}\right)^T \\ &= \begin{bmatrix} \frac{\partial^2 J_{ij}}{\partial^2 g_{ij}(0)} & \frac{\partial^2 J_{ij}}{\partial g_{ij}(0)\partial g_{ij}(1)} & \cdots & \frac{\partial^2 J_{ij}}{\partial g_{ij}(0)\partial g_{ij}(N-1)} \\ \frac{\partial^2 J_{ij}}{\partial g_{ij}(1)\partial g_{ij}(0)} & \frac{\partial^2 J_{ij}}{\partial^2 g_{ij}(1)} & \cdots & \frac{\partial^2 J_{ij}}{\partial g_{ij}(1)\partial g_{ij}(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J_{ij}}{\partial g_{ij}(N-1)\partial g_{ij}(0)} & \frac{\partial^2 J_{ij}}{\partial g_{ij}(1)\partial g_{ij}(N-1)} & \cdots & \frac{\partial^2 J_{ij}}{\partial^2 g_{ij}(N-1)} \end{bmatrix}. \end{aligned} \tag{2.33}$$

Substituting (2.30) into (2.33), we have

$$\mathbf{H}\left(J_{ij}\left(\mathbf{g}_{ij}(t)\right)\right) = \mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}^T, \tag{2.34}$$

so that, by (2.30)-(2.34), the adaptation using Newton's method becomes

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t) \left(\mathbf{R}^T_{\mathbf{y}_j\mathbf{v}_j} \right)^{-1} \mathbf{r}_{v_i\mathbf{v}_j}. \tag{2.35}$$

The RLS-like adaptation of [17] could be obtained by replacing $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$'s with their time average estimates, and by estimating their inverse matrices recursively (see Eqs.(63-66) of [17] and the Appendix therein). Usually, adaptations based on Newton's method can be expected to have faster convergence rate compared with stochastic gradient based algorithms. However, our tests also show that they suffer from instability in addition to its disadvantage of high computation complexity. In practice, the Newton-Raphson adaptations often use some modified estimate of Hessian to improve robustness of algorithm, such as the "modified Newton" solution to the decorrelation problem in [28].

Assuming that the correlation matrix $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$ remains positive definite in all its quadratic forms for any real vectors, i.e.,

$$\langle \mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{a}, \mathbf{a} \rangle = \mathbf{a}^T \mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{a} > 0, \forall \mathbf{a} \neq \mathbf{0}, and\ \mathbf{a} \in \mathbf{R}^N, \tag{2.36}$$

which means that the angle between the direction of vector $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}$ and that of vector $\mathbf{r}_{v_i\mathbf{v}_j}$ is less than $\frac{\pi}{2}$, or,

$$\cos\left(\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}, \mathbf{r}_{v_i\mathbf{v}_j}\right) = \frac{\langle \mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}, \mathbf{r}_{v_i\mathbf{v}_j} \rangle}{\|\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}\mathbf{r}_{v_i\mathbf{v}_j}\| \cdot \|\mathbf{r}_{v_i\mathbf{v}_j}\|} > 0, \tag{2.37}$$

where $\langle \cdot, \cdot \rangle$ denotes vector inner-product and $\| \cdot \|$ stands for vector norm. Therefore, we can replace the gradient vectors $\mathbf{R}_{ij}\mathbf{r}_{v_i\mathbf{v}_j}(t)$ in (2.30) with $\mathbf{r}_{v_i\mathbf{v}_j}(t)$, and obtain

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t)\mathbf{r}_{v_i\mathbf{v}_j}(t)_{,i\neq j}\,, \tag{2.38}$$

whose instantaneous approximation coincides with the adaptation direction of basic ADF algorithm (1.5), derived in [17] from a zero-searching formulation using the method of Robbins-Monro stochastic approximation [36]. According to the Theorem.1 in [37], such a stochastic approximation procedure will converge to the zero-correlation solution. With the vector formulation, the basic ADF adaptation (1.5) can be rewritten as

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t) v_i(t) \mathbf{v}_j(t)_{,i \neq j} \, , \tag{2.39}$$

where $\mu(t)$ is the adaptation step-size that controls convergence rate.

It is worth noting that, although the positive-definite assumption is usually not guaranteed for arbitrary cross-correlation matrices, it is held in an approximate sense for $\mathbf{R}_{\mathbf{y}_j \mathbf{v}_j}$ in the application of ADF model for the current scenario. Under practical working conditions of sound source separation and following the Eq. 2.12 in Section 2.1.1, i.e.,

$$\mathbf{R}_{\mathbf{y}_j \mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_j \mathbf{y}_j} - \mathbf{R}_{\mathbf{y}_j \tilde{\mathbf{y}}_i} \mathbf{G}_{ji}^T, \tag{2.40}$$

where the elements of $\mathbf{G}_{ji}$'s are less than 1 because direct-path components are usually stronger than cross-path components. As such, $\mathbf{R}_{\mathbf{y}_j \mathbf{v}_j}$ is dominated by $\mathbf{R}_{\mathbf{y}_j \mathbf{y}_j}$ and hence its positive definite property. Actual measurements in experimental conditions on the eigen-values of $\mathbf{R}_{\mathbf{y}_j \mathbf{v}_j}$ during ADF adaptation also verified this property. Figures 2.1 and 2.2 show the probability density distributions of an variable that reflects the signs of the real part of the eigen-values (λ) of the matrix $\mathbf{R}_{\mathbf{y}_j \mathbf{v}_j}$. Such distribution properties during ADF adaptations almost surely guarantee the positive definitiveness of the matrix.

2.2.2 Convergence and error analysis

The actual convergence behavior of ADF adaptation procedure 2.39 is very complicated. As was pointed out in [25] that whether the procedure converged and whether ADF converged to the desired separation filters as a unique solution depended on the initial values chosen for ADF filters, the mixing condition, and step sizes. The analyses were based on white source assumptions and low dimensional mixing filter vector. Although examples of undesired "phantom" separation filters were given, [25] also confirmed the validness of using ADF in practical conditions that prevented instability and convergence to false solutions. In fact, the physical realities in our application are in agreement with the mathematical constraints of [25]. For example, the initialization of ADF filters takes zero values, and the absolute causality of mixing system is guaranteed for applications based on circular microphone arrays adopted in this dissertation.

ADF algorithm is an adaptive signal processing method in essence. As is known from the theory of adaptive signal processing, the convergence behavior of the adaptive FIR filter is determined by the eigen-value spreadness of input correlation matrix [51]. Similar results hold for the convergence of ADF system by analyzing converging behaviors of the expectation of ADF filter error vectors. In [20], it was shown that the convergence of the expected ADF error vector is dependent on eigen-value spread properties determined by input correlations, similar to adaptive FIR filter. The following analysis from [20] and [52] are listed below for completeness.

Define the ADF separation filter error vector by

$$\phi(t) = E\{\left[\mathbf{g}_{12}^T(t), \mathbf{g}_{21}^T(t)\right]^T - \mathbf{g}^*\}, \tag{2.41}$$

where $\mathbf{g}_{ij}(t) = [\mathbf{g}_{ij}(t), \cdots, \mathbf{g}_{ij}(t-N+1)]^T$'s are the separation filter vectors, and

$$\mathbf{g}^* = E\{\left[\mathbf{g}_{12}^T(t), \mathbf{g}_{21}^T(t)\right]^T\} \tag{2.42}$$

is the expectation of ADF filter vector. The first-order approximation [20] of the error vector adaptation can be described by

$$\phi(t+1) = (\mathbf{I} - \mu\Psi)\,\phi(t) \tag{2.43}$$

where the input correlation matrix was define as

$$\Psi = \begin{bmatrix} E\{\mathbf{y}_2(t)\mathbf{y}_2^T(t)\} & E\{y_1(t)\mathbf{Y}_1(t)\} \\ E\{y_2(t)\mathbf{Y}_2(t)\} & E\{\mathbf{y}_1(t)\mathbf{y}_1^T(t)\} \end{bmatrix}, \tag{2.44}$$

with the Hankel input data matrices $\mathbf{Y}_i = [\mathbf{y}_i(t), \cdots, \mathbf{y}_i(t-N+1)]^T, i = 1, 2$. Applying the eigen-value decomposition to the matrix Ψ,

$$\Psi = U\Lambda U^{-1}, \tag{2.45}$$

with U the eigen-vector matrix and $\Lambda = diag(\lambda_1, \cdots, \lambda_{2N})$ the eigen-value matrix, the update of the i^{th} tap of the filter error can be expressed as [20]

$$\phi_i(t+1) = \sum_{k=1}^{2N} (1 - \mu\lambda_k)^t \mu_{ik} \phi(0) \tag{2.46}$$

where $\phi(0) = U^{-1}\phi(0)$.

From (2.46), we know that the convergence time of each filter tap is dominated by the smallest eigen-value λ_{min} of the input correlation matrix Ψ. The larger the value of λ_{min}, the smaller the time it will take for the separation filters to converge, and therefore the larger the convergence rate. Therefore, the convergence behavior of ADF adaptation depends heavily on properties of input signals. According to [20], λ_{min} attains its maximum value when all the eigen-values are identical, i.e., when the input signals are white. This is similar to other adaptive signal processing methods such as LMS algorithm. It is based on the above analysis that the input-normalized step size (1.6) was proposed. In fact, the normalization of the adaptation step size by the energy of the tap-input data vector is equivalent to dividing the it by the sum of all eigenvalues [20].

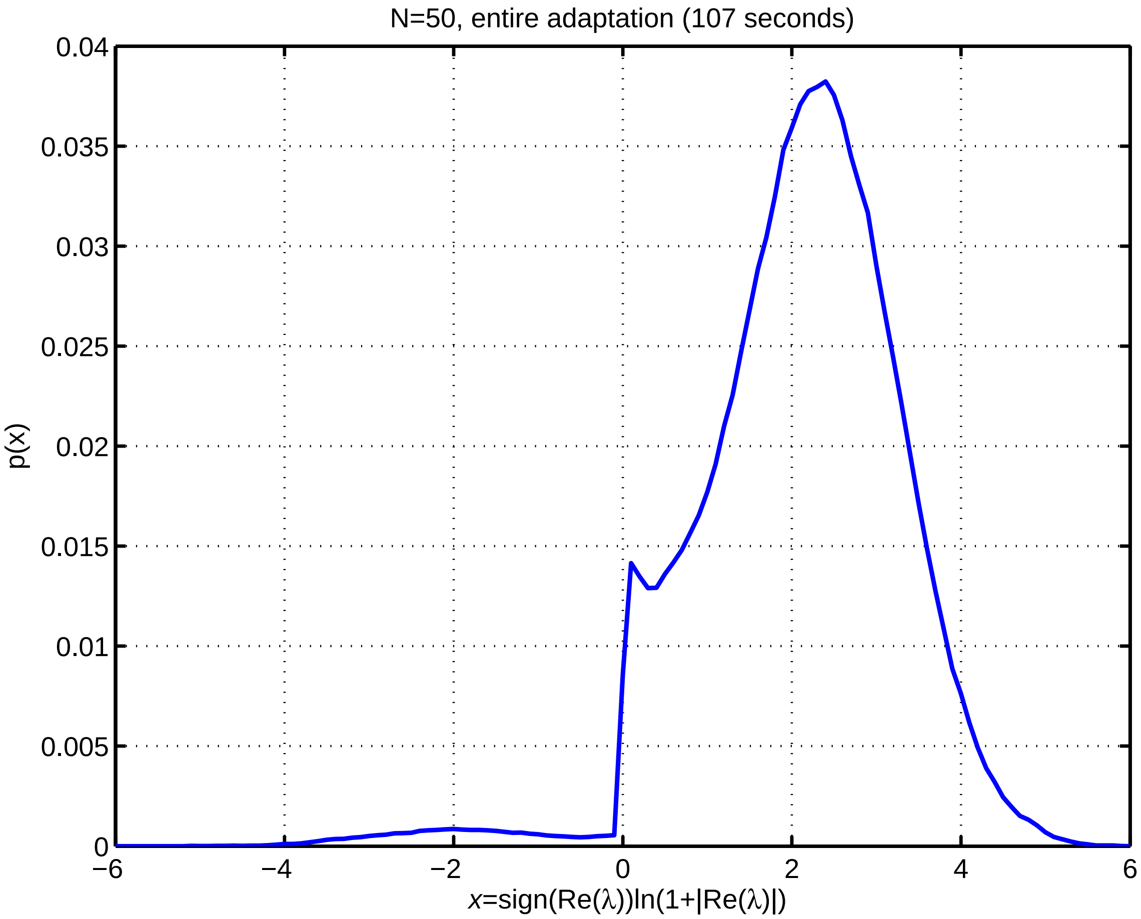


Figure 2.1: The probability density of the variable $x{=}sign\,(Re(\lambda))\,log_{10}\,(1+|Re(\lambda)|)$, measured for ADF adaptation with preemphasis and $N=50$, $\gamma=0.005$, where λ is the eigen-values of matrix $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$; its positive-definitiveness is almost surely held.

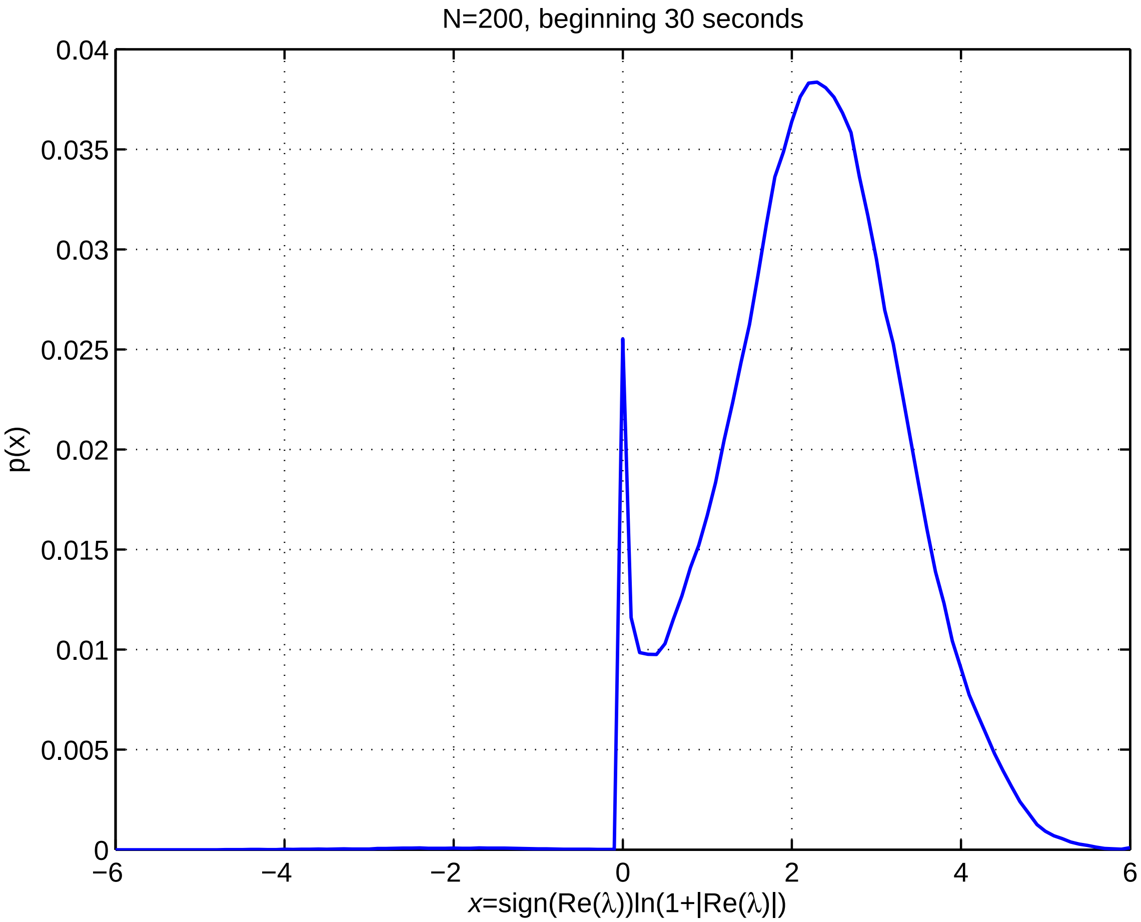


Figure 2.2: The probability density of the variable $x{=}sign\,(Re(\lambda))\,log_{10}\,(1+|Re(\lambda)|)$, , measured for ADF adaptation with preemphasis and $N = 200$, $\gamma = 0.005$,where λ is the eigen-values of matrix $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$; its positive-definitiveness is almost surely held.

Chapter 3

CONVERGENCE SPEEDUP PROCESSING AND MULTI-ADF INTEGRATION

In this chapter, we discuss the processing techniques that speedup convergence rate of ADF and multi-ADF post-filtering that reduces reverberation noise after ADF processing. The convergence behavior of an ADF system depends on the properties of speech signals. An analysis on ADF convergence properties is first introduced, and then techniques for speeding up convergence are presented based on such analysis. When multiple pairs of recording sensors are available, methods of multi-ADF integration are proposed to improve the separation performances.

3.1 Prewhitening Processing

From the convergence analysis of ADF adaptation in Section 2.2.2, we know that the convergence of filter estimation achieves best conditioning when the input signals are white. It is well known that speech signal is colored, especially for voiced sound, and there is a tilt of $6dB$ per octave [53] in its spectrum. To obtain fast convergence rate, we can do whitening processing to input data before doing ADF filter adaptation so that the spread of the eigen-values could be minimized. Due to the non-stationarity of speech signals, complete whitening of an input speech source may require adaptive processing such as adaptive linear prediction. However, for the problem of blind convolutive mixture separation, real sources are unobservable. It is also infeasible to completely whiten the input speech mixtures since it will cause undesirable speech distortions. In practice, we could choose to partially whiten the speech mixtures without loosing phonetic contents of speech sounds to help reduce the eigen-spread that result from the long-term speech spectral characteristics [52]. This can be achieved by linear time-invariant pre-whitening filtering. Since we are not interested in totally whiten the speech signals by removing their detailed structures in spectrum, finite impulse response (FIR) filters with very low orders should suffice.

Denote the prewhitening filter by $W(z)$ and apply this filter to speech mixture inputs, we have

$$\begin{bmatrix} W(z)Y_1(z) \\ W(z)Y_2(z) \end{bmatrix} = \begin{bmatrix} 1 & G_{12}^o(z) \\ G_{21}^o(z) & 1 \end{bmatrix} \begin{bmatrix} H_{11}(z)W(z)S_1(z) \\ H_{22}(z)W(z)S_2(z) \end{bmatrix}. \tag{3.1}$$

Eq. 3.1 indicates that the whitening filtering applied to speech mixtures are equivalent to $W(z)$ applied to speech sources. Therefore, the whitening filters can be designed for speech sources and then applied to speech mixtures. Based on the long-term spectral properties of clean speech signals, we can derive several types of whitening filters, including inverse-PSD FIR filters and preemphasis filters. When the shape of the speech spectrum curve is unknown, we may require the estimation of signal spectral characteristics from training data using such methods as linear predictive coding (LPC) or auto-regressive (AR) modeling. Our testing of whitening filters designed with LPC methods also show that such low order LPC whitening filters trained from speech data has similar spectral properties and comparable performances for ADF speech separation model. However, speech spectral properties have been well-studied and it is much easier to design both inverse-PSD and preemphasis types of filters to flatten the long-term speech spectrum, based on such knowledge.

3.1.1 Inverse-PSD

Prewhitening techniques by filters that implement the reciprocal magnitude response of long-term speech spectrum was previously utilized in hearing aid arrays [54]. The inverse-PSD whitening processing follows the same line of thoughts. The design of inverse-PSD prewhitening filter was based on the average curve of power spectral density of human speech described in [55]. After obtaining sample points of average speech PSD, we obtain their inverse and use a frequency sampling method presented in [54] to compute the FIR coefficients of the inverse-PSD type of prewhitening filter. The procedures for the computation of filter coefficients are listed in Appendix A. Figure 3.1 shows the designing sample points taken from the long-term spectrum of speech. Figure 3.2 depicts the desired and actual magnitude responses of the whitening filter designed from the inverse of long-term speech PSD. A 4-th order FIR filter was obtained by the frequency-sampling filter design method [54]. The transfer function of the inverse-PSD prewhitening filter is

$$W(z) = 0.026448 - 0.223982z^{-1} + 0.423960z^{-2} - 0.223982zz^{-3} + 0.026448zz^{-4}. \quad (3.2)$$

An 80-th order FIR dewhitening filter to be applied to ADF output signals was also computed as the inverse of the whitening filter. Figure 3.3 gives its magnitude response. The coefficients of both filters are shown in Figure 3.4 .

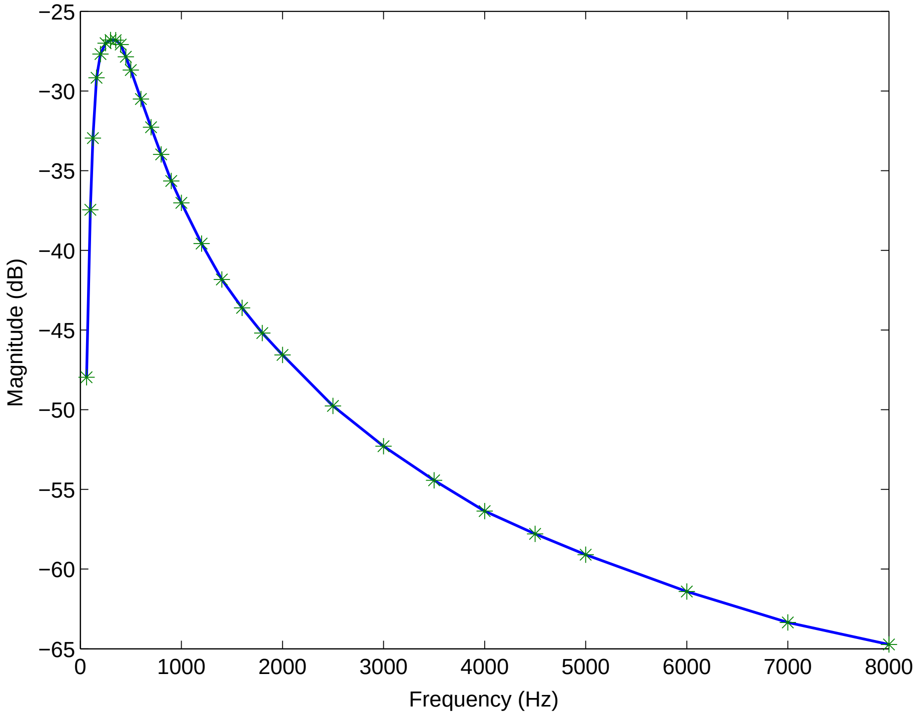


Figure 3.1: Data points sampled from the long-term spectrum of speech.

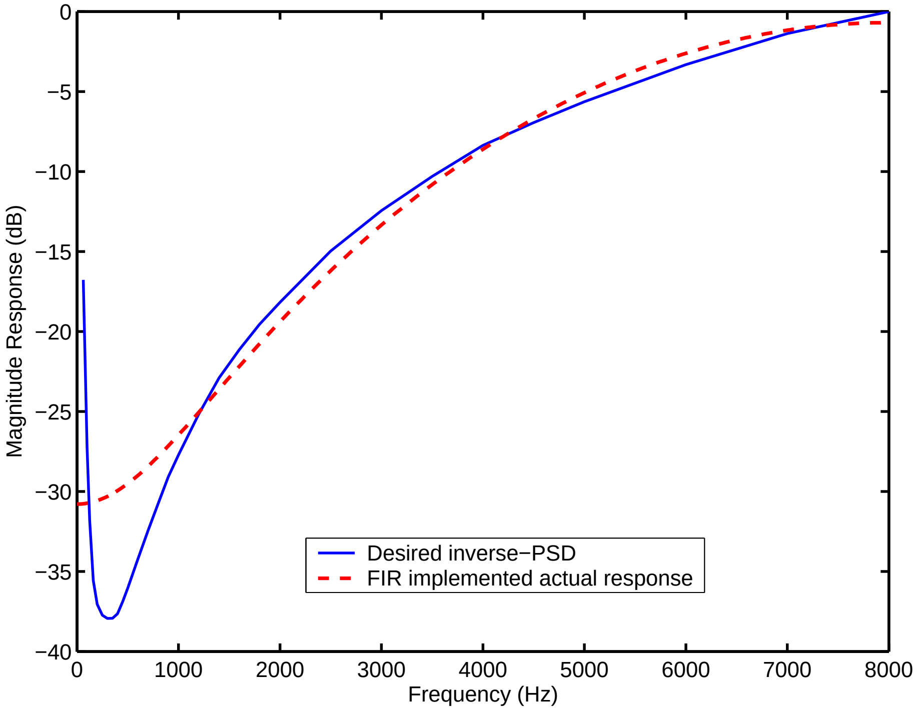


Figure 3.2: Desired whitening filter response by inverse-PSD and actual response implemented by FIR filter of length 5.

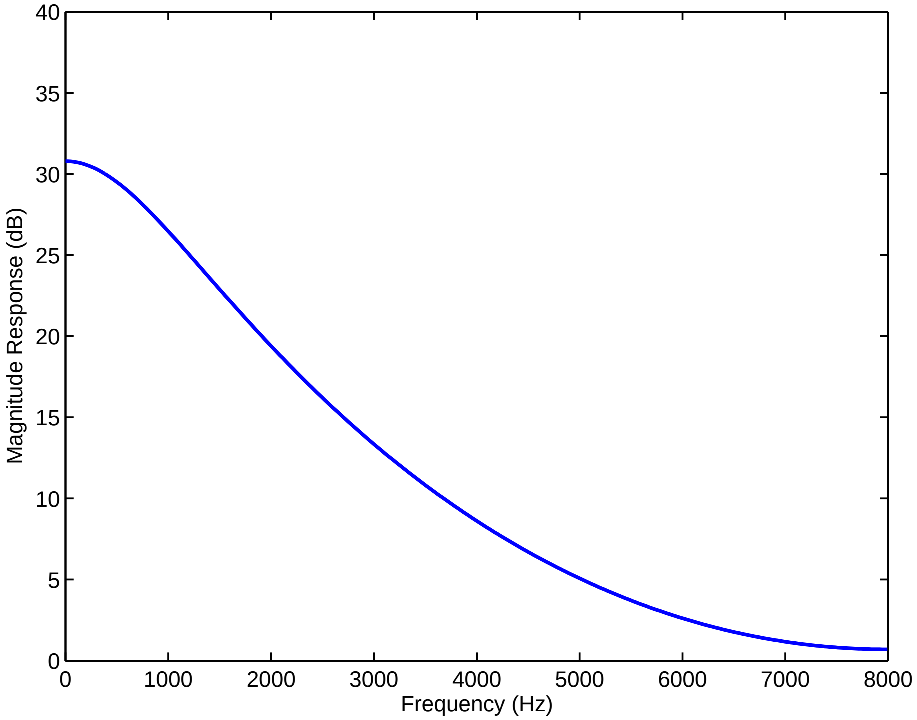


Figure 3.3: Magnitude response of the de-whitening filter.

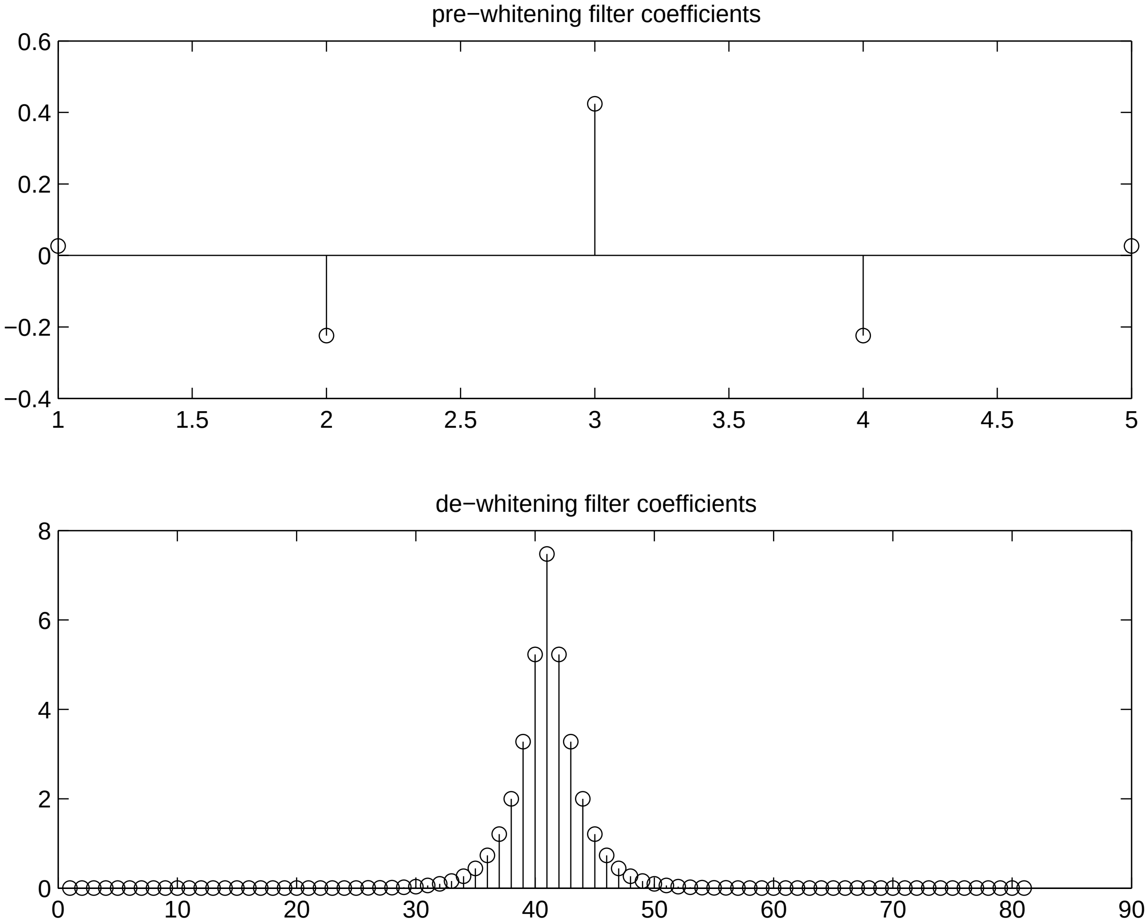


Figure 3.4: Filter coefficients of pre-whitening and de-whitening filters.

3.1.2 Preemphasis

Preemphasis is a first-order high-pass filter to compensate for the 6dB per octave spectral tilt of voiced speech. It is a commonly used technique in linear prediction coding of speech. The whitening filter of preemphasis has the form

$$W(z) = 1 - \mu z^{-1}, \tag{3.3}$$

where the constant coefficient $\mu (0 < \mu \leq 1)$ controls the level of suppression for low frequency bands. The deemphasis processing to ADF output signals does the inverse of the preemphasis filter $(1 - \mu z^{-1})^{-1}$. It can be easily implemented with auto-recursion of output signals $v_i(t)$'s.

Figure 3.5 compares the magnitude responses of the both types of prewhitening filters. Both types of prewhitening filters have similar high-pass characteristics in the frequency range of 1KHz to 5KHz. It is obvious that the value of μ for preemphasis filtering mainly controls the suppression of low-frequency part of speech. Experiments show that the preemphasis filters with $\mu = 1$ achieves better enhancement performances to ADF. Therefore, we use $\mu = 1$ for preemphasis processing in subsequent discussions.

Both inverse-PSD and preemphasis improve the convergence of ADF adaptations by flattening speech long-term spectrum. As analyzed in Section 2.2.2, the whitening processing reduces eigen-spreadness of the matrix Ψ in (2.44) and provides better conditioning for the adaptation. As an verification, we compared the averaged condition numbers of Ψ with and without whitening filtering of inverse-PSD. The result shows that whitening filtering reduces the condition number by two orders of magnitude. The averaging was performed over the beginning five seconds of speech mixtures and the averaged condition numbers were computed to be 2.16×10^7 and 4.53×10^5, with and without inverse-PSD, respectively.

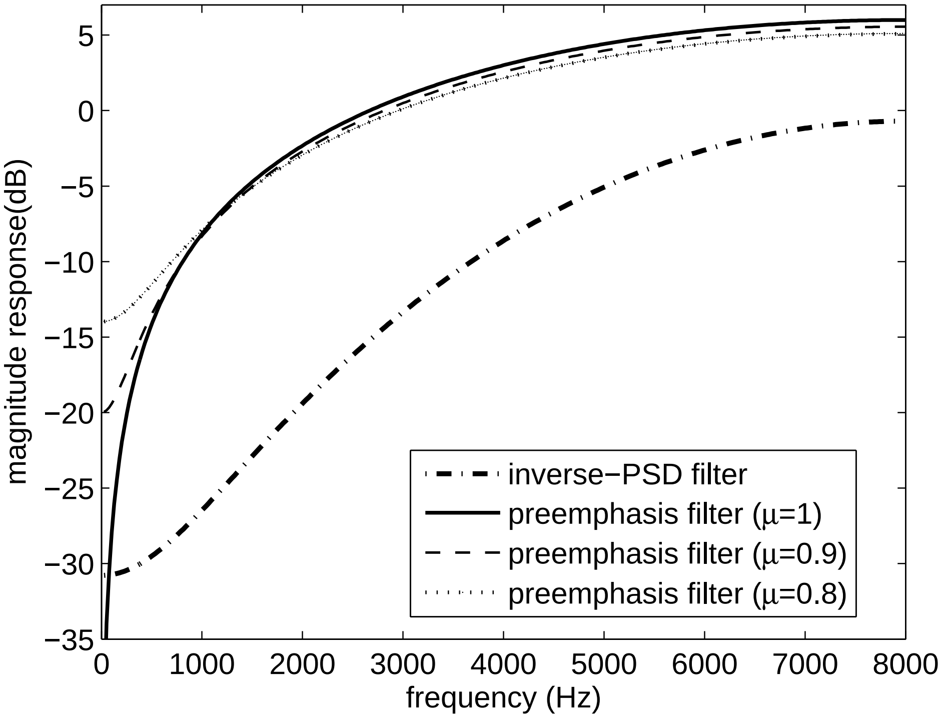


Figure 3.5: Magnitude responses of prewhitening filters

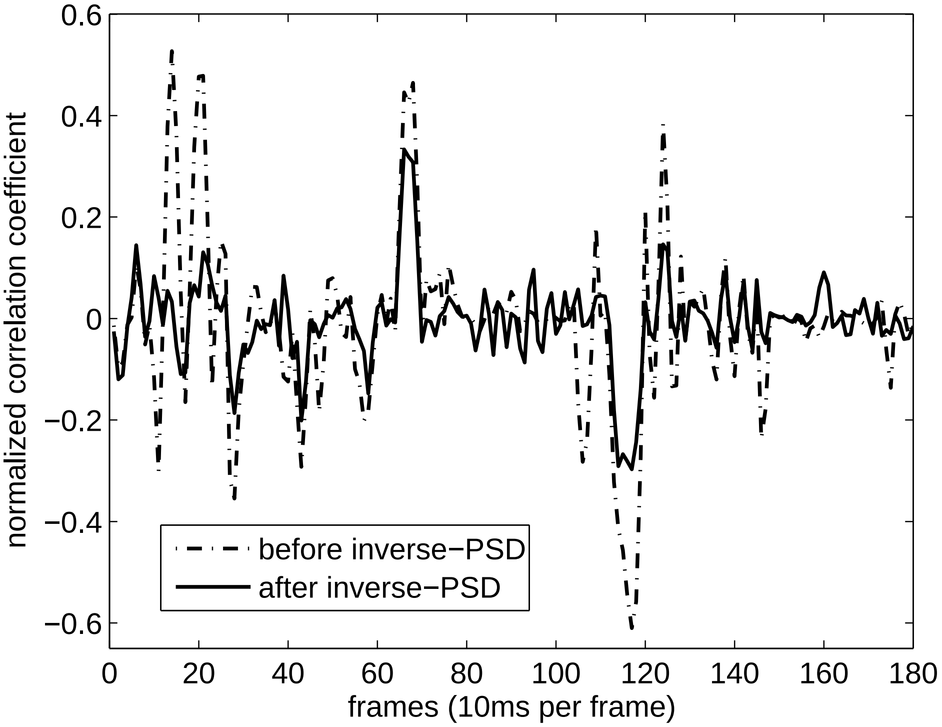


Figure 3.6: Evolution of correlation coefficients at lag zero between two speech sources, before and after pre-whitening

In addition to the effect of improving adaptation stabilities, prewhitening also has the effect of reducing cross-correlation between the source components and thus effectively avoids the utilization of the correlated information between the sources. Figure 3.6 shows the cross-correlation coefficients at lag zero between two clean source speeches measured over successive frames of two source speech signals, before and after the inverse-PSD pre-whitening processing. It is observed that the theoretical assumption of statical uncorrelation only holds in an approximate sense in practice, and strong cross correlation occurred from time-to-time. In fact, pre-whitened speech source signals are significantly reduced in cross-correlations at all time lags, and this makes the subsequent ADF work at a condition more close to its basic assumption.

3.2 Block-Iteration

The baseline ADF algorithm is a sample-by-sample sequential procedure for adaptive filter estimation and source separation. Consider a two-input ADF with the input of each pair of speech mixture samples, the ADF filters are adapted and one pair of new output samples are computed every $1/F_s$ seconds, where F_s is the sampling frequency. This form of processing has the advantage of small delay in time. Such a procedure is feasible when speech data sequences are long and acoustic paths are stationary, but it is ineffective in utilizing data for fast convergence. For ASR scenarios, speech recognition processings are usually performed after the end-point of speech sentences are reached. ASR processings, such as decoding, require longer time and has delays much larger than one time sample. Therefore, it is reasonable in ASR applications to extract more information from large data blocks iteratively, at the cost of large delays that may not be tolerable in human speech communication scenarios, but is acceptable for ASR applications.

In the proposed block iterative ADF, input speech data are divided into blocks, and within each block, adaptive estimation and separation are iteratively performed by (1.4) and (1.5). Assume that each block consists of B samples, $Y_{i,k} = [y_{i,kB+t}, t = 0, 1, \cdots, B-1], i = 1, 2$, with k indexing blocks. For block k, the filters estimated at the end of the block in the r^{th} iteration , i.e., $\mathbf{g}_{ij}^{(B-1)}(k,r)$'s are used as the initial filter estimates at the $(r+1)^{th}$ iteration, i.e.,

$$\mathbf{g}_{ij}^{(0)}(k, r+1) = \mathbf{g}_{ij}^{(B-1)}(k, r). \tag{3.4}$$

The initial estimates of filters for the block $k+1$ are set to be the final filter estimates obtained in the previous block k. The actual number of ADF iterations for each block is determined by a measure of convergence. One example of such a measure is the relative change of filter estimates between two consecutive iterations, defined as

$$C_{k,r+1} = \frac{1}{2}\left(\frac{\|\mathbf{g}_{12}^{(B-1)}(k,r+1) - \mathbf{g}_{12}^{(B-1)}(k,r)\|}{\|\mathbf{g}_{12}^{(B-1)}(k,r+1)\|} + \frac{\|\mathbf{g}_{21}^{(B-1)}(k,r+1) - \mathbf{g}_{21}^{(B-1)}(k,r)\|}{\|\mathbf{g}_{21}^{(B-1)}(k,r+1)\|}\right). \tag{3.5}$$

If $C_{k,r+1} \geq \varepsilon$, where ε is an empirically chosen threshold, then the procedure of ADF adaptation will continue for the next round of iteration on the same block of input data; otherwise, the ADF estimation terminates and move on to the next block $k+1$.

Two important parameters in the block-iterative ADF are the block length B and the iteration-stop threshold ε. A block should be sufficiently long so that multiple phonetic sounds are included within each block, since second-order statistic methods of blind source separation do not guarantee correct solutions for stationary signal sources. Obviously the block iterative method also introduces a buffering delay that is determined by the block length. Therefore, the choice of B controls the tradeoff between accuracy and delay. For online applications or time-varying acoustic conditions, block length needs to be kept short to minimize buffering delay; for offline and stationary acoustic conditions, block length can be extended for higher estimation accuracy. Iterative batch processing is resulted when block length is set to be the length of the data sequence. The threshold ε is basically a tradeoff between convergence rate and computation load. A small threshold calls for more iterations which improves convergence rate but incurs more computation, and a large threshold calls for fewer iterations and therefore slower convergence and less computation. Since a block of data can provide only limited information for adaptation, it is in general inefficient to use a very small threshold for excessive iterations within each data block.

Figure 3.7 illustrates the convergence curve of ADF estimation error for the block-iterative ADF implementation with inverse-PSD prewhitening processing. Significant speedup was introduced by block-iteration compared with the curves recorded from batch ADFs. For batch ADFs, those utilizing preemphasis and inverse-PSD processings demonstrated significantly smaller steady state filter estimation errors than the baseline ADF. In all these simulations, the filter length of $N = 400$ was used and the normalizing step size in (1.6) was added small number β in the denominator to prevent divide-by-zeros.

$$\mu(t) = 2\gamma/N\left(\beta + \sigma_{y_1}^2(t) + \sigma_{y_2}^2(t)\right), \tag{3.6}$$

where the gain factor $\gamma = 0.005$ was employed and $\beta = 0.8$.

3.3 Post-Filtering and Multi-ADF Integration

Due to estimation errors in the adaptation of ADF filters, the ADF output speech signals still contain certain levels of residual interference speech. On the other hand, as discussed in Section 1.3, the reverberation effects of acoustic impulse responses could not be removed by ADF separation system itself. In fact, the reverberation effects in ADF output speech actually worsens the reverberant condition of the separated

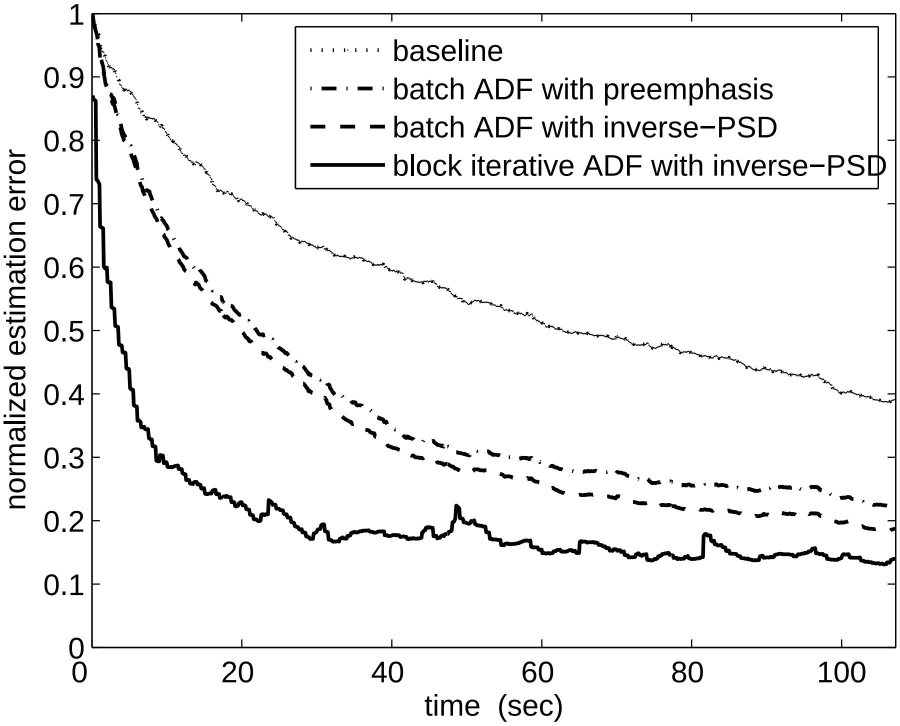


Figure 3.7: Comparison of convergence rates for ADF enhancement techniques of prewhitening and block-iteration.

speech at ADF output. Based on the analysis in Section 1.2, we could only get linearly transformed version of the source speech even at ideal source separation. From (1.1) and (1.2), the ideally separated signal $v_i(t)$ is described in Z-domain by

$$V_i^o(z) = \left(1 - G_{ij}^o(z)G_{ji}^o(z)\right) H_{ii}(z)S_i(z)_{,i\neq j}\,. \tag{3.7}$$

In (3.7), the linear distortion term $\left(1 - G_{ij}^o(z)G_{ji}^o(z)\right)$ in fact adds further reverberation effects to the natural reverberation of the direct acoustic path $H_{ii}(z)$. The actual situation will be worse than this. Due to filter estimation errors, residual interference speech signals will never be zeros and such residual signals are also reverberant in nature. We have to rely on additional information for the removal of those kinds of reverberant effects. Multiple outputs from multiple pairs of ADF processing provide such kind of information. Acquired with multiple microphones from different points in space, these ADF output signals preserve the spatial diversity useful for the removal of reverberation noise and residual interference speech.

The basic principle of the proposed post filtering is to align multiple target speech signals that are produced by different ADF modules so as to further enhance target speech. Assume that Q pairs of microphone signals are available and multiple estimates of the two source speech signals are generated by these Q pairs of ADF processing. Denoting these pairs of ADF outputs as $\left(v_1^q(t), v_2^q(t)\right), q = 1, 2, \cdots, Q$, we will apply post-filtering to combine individual ADF pairs to obtain a single pair of enhanced estimate of output speech $(\hat{v}_1(t), \hat{v}_2(t))$.

The post-filtering is performed in frequency domain in a segment by segment fashion. The derivation of the post filters are based on the minimum-mean-squared-error (MMSE) criterion. According to [56], the filter $\hat{f}(t)$ that aligns and filters a signal $x(t)$ optimally, in the sense of MMSE, relative to a reference signal $s(t)$ can be obtained from the minimization of the mean-square-error (MSE) as

$$\hat{f}(t) = argmin_{f(t)} E\left[s(t) - f(t) * x(t)\right]^2 , \tag{3.8}$$

where “*” denotes convolution. The solution to (3.8) in frequency domain results in the optimal filter $F^o(f)$ that takes the value of coherence function [56], i.e.,

$$F^o(f) = P_{xs}(f)/P_{xx}(f), \tag{3.9}$$

where $P_{xs}(f)$ and $P_{xx}(f)$ are the cross and auto PSD respectively.

Without loss of generality, assume that the target speech to be enhanced is source 1 and the corresponding ADF output $v_1^{(1)}(t)$ is designated as the reference signal in the multiple ADF post-filtering procedure. Then the optimal filters of the current segment for all other ADF outputs $v_1^{(q)}(t), q = 2, \cdots, Q$ are determined by

$$\hat{F}_1^{(q)}(f) = \frac{P_{v_1^{(q)} v_1^{(1)}}(f)}{P_{v_1^{(q)} v_1^{(q)}}(f)}. \tag{3.10}$$

The optimally aligned and filtered target signal for the q^{th} ADF pair is

$$\bar{V}_1^{(q)}(f) = \hat{F}_1^{(q)}(f) V_1^{(q)}(f). \tag{3.11}$$

The enhanced target signal is taken as an average of the post-filtered target signals

$$\hat{V}_1^{(q)}(f) = \frac{1}{Q}\left(V_1^{(1)}(f) + \sum_{q=2}^{Q} \bar{V}_1^{(q)}(f)\right). \tag{3.12}$$

For online applications and nonstationary acoustic paths, segmental processing is necessary. In the multi-ADF integration procedure of (3.11) and (3.12), the estimation of optimal filters $\hat{F}_1^{(q)}(f)$ in (3.10) are important for the reliability of post-filtering. To improve the robustness of the estimates of $\hat{F}_1^{(q)}(f)$, the segment-wise recursive smoothing are applied to both the post-filters and the cross and auto PSDs, with the forgetting factors α_1 and α_2, respectively, similar to the processing employed in [57]. Let m index segments and M denote the segment length, the smoothing procedures are defined as follows:

$$P_{v_1^{(q)}v_1^{(p)}}(f, mM) = \alpha_1 P_{v_1^{(q)}v_1^{(p)}}(f, (m-1)M) + (1-\alpha_1) V_1^{(q)}(f, mM) V_1^{(p)}(f, mM)^H, \tag{3.13}$$

$$F_1^{(q)}(f, mM) = \alpha_2 F_1^{(q)}(f, (m-1)M) + (1-\alpha_2) \frac{P_{v_1^{(q)}v_1^{(1)}}(f, mM)}{P_{v_1^{(q)}v_1^{(q)}}(f, mM)}, \tag{3.14}$$

where, the superscript ^{H} denotes complex conjugation and the index $p = q$ is for auto PSD and $p \neq q$ for cross PSD.

The smoothed filter estimate $F_1^{(q)}(f, mM)$ is used in (3.11) and (3.12) to compute signal spectral estimate, and the time-domain target speech signal is then obtained by the overlap-and-add method [58, 59]. FFTs of length-4096 were performed by setting M to be 2048 samples and padding 2048 zeros. The forgetting factors α_1 and α_2 were empirically chosen as 0.999 and 0.95 respectively. It should be noted that, since the alignment delays between multi-ADF outputs and reference signals may be negative, the optimal filters are not guaranteed to be causal. The overlap-and-add procedures are implemented with the ability to deal with this problem so that discontinuity in the resulting combined speech could be avoided.

The system diagram for multi-ADF post-filtering is illustrated in Figure 3.8, where the signals are acquired from a circular microphone array and the proposed enhancement techniques are also integrated into the speech source separation and ASR system. The circular array is specific to the RWCP measurement data [60]. Input mixture signals are subject to whitening filtering prior to speech separation. ADF modules are implemented by the block iterative method. ADF outputs are de-whitened and the post-filtering module combines the target signals to generate an enhanced target signal which is then recognized by the ASR system.

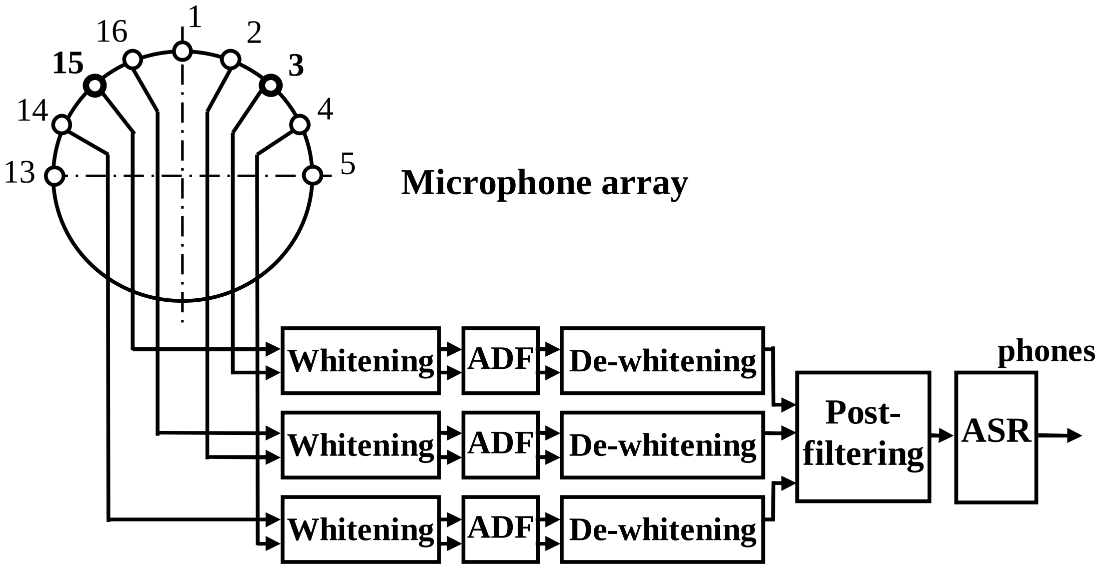


Figure 3.8: Integration of multiple ADF separation models

3.4 Speech Separation Experiments

3.4.1 Experimental data and setup

Cochannel speech data were generated by convolving the impulse responses of acoustic paths measured in RWCP [60] with the source speech materials of TIMIT database at the sampling rate of 16 KHz. As shown in Figure. 3.9, a circular microphone array with a radius of $15cm$ was used to capture speech signals of two sources located at 130 and 50 degrees, respectively. The speaker-to-microphone distances were approximately 2 meters. Different numbers of microphone pairs on the circular array 3.8 were used in experiments: 16 and 2, 15 and 3, 14 and 4, 13 and 5, as illustrated in Figure. The pair 15 and 3 was also used in the condition of single microphone pair. The recording room had a reverberation time $T_{60} = 300ms$. At the target speaker location , speech data of four speakers (faks0, felc0, mdab0, mreb0) from TIMIT database were used, with each speaker contributing ten sentences. At the jammer speaker location , speech data were randomly taken from the entire set of TIMIT sentences excluding those of the target speakers.

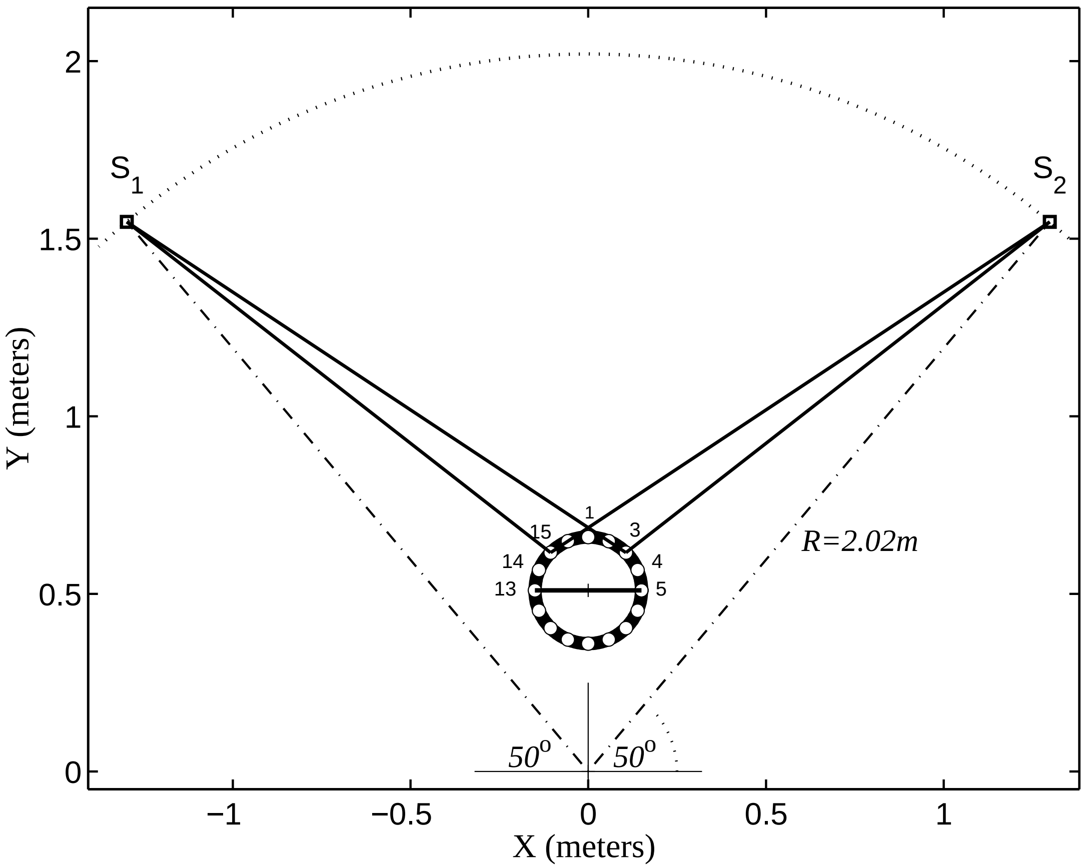


Figure 3.9: Microphone array configuration

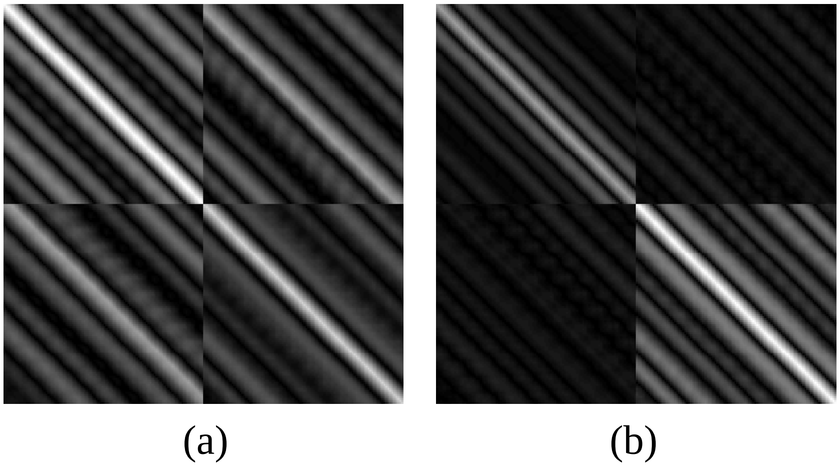


Figure 3.10: Correlation matrices of ADF system (a) input $\mathbf{R_{yy}}$; (b) output $\mathbf{R_{vv}}$, computed from the input and output of block-iterative ADF with inverse-PSD whitening filtering.

Assume that the microphones at the locations 15 and 3 acquire speech mixtures y_1 and y_2, respectively, which are convolutive mixtures of speech signals s_i's and h_{ij}'s . The input target-to-interference ratio in for the input speech mixture y_i, i.e., TIR_{y_1}, is defined as the energy ratio (dB) of the target component s_i in y_i to all the interference components s_j's in y_i. The ADF outputs are defined accordingly. The TIR gains for muti-ADF post-filtering is defined by

$$\Delta TIR = TIR_{\hat{v}_1} - TIR_{average}, \tag{3.15}$$

where

$$TIR_{average} = \frac{1}{Q}\sum_{q=1}^{Q} TIR_q. \tag{3.16}$$

The initial conditions were $TIR_{y_1} = 0.53dB$ and $TIR_{y_2} = -0.56dB$. For multiple pairs of microphones, the combinations of multiple ADF modules were defined as the following: $15-3$ and $14-4$ for two pairs, $16-2$, $15-3$, and $14-4$ for three pairs, and $16-2$, $15-3$, $14-4$, and $13-5$ for four pairs.

The decorrelation property of ADF processing was verified by the system input-output data correlation structure for block-iterative ADF with inverse-PSD prefiltering. The estimates of the system input and output correlation matrices, defined as $\mathbf{R}_{\mathbf{yy}} = E\{\mathbf{y}(t)\mathbf{y}^T(t)\}$ and $\mathbf{R}_{\mathbf{vv}}$ defined in (2.5), respectively, are computed from the time average of input and output signal vectors $\mathbf{y}(t)$ and $\mathbf{v}(t)$. Their 2D images are shown in Figure 3.10, where the pixel brightness corresponds to magnitude of each correlation coefficient. The structures of the correlation matrices clearly displayed four blocks, where the diagonal blocks are auto-correlations and the off-diagonal blocks are cross-correlations. After ADF processing, the cross-correlation blocks are significantly reduced by the decorrelation effect of ADF.

3.4.2 Convergence performances

The convergence speedup resulted from the introduction of prewhitening processing and block-iterative implementation are evaluated by the averaged normalized filter estimation error $\epsilon(t)$, defined on filter estimates $\mathbf{g}_{ij}(t)$'s, relative to true separation filters $\mathbf{g}_{ij}^o$'s, by

$$\epsilon(t) = \frac{1}{2}\left(\frac{\mathbf{g}_{12}(t) - \|\mathbf{g}_{12}^o\|}{\|\mathbf{g}_{12}^o\|} + \frac{\mathbf{g}_{21}(t) - \|\mathbf{g}_{21}^o\|}{\|\mathbf{g}_{21}^o\|}\right), \tag{3.17}$$

where $\mathbf{g}_{ij}(t)$'s are initialized with zeros at $t = 0$ and true filters $\mathbf{g}_{ij}$'s are computed directly from the known raw impulse response data by transforming the frequency representation (FFT) of (1.3) into time domain using IFFT's.

For block-iterative ADF, the choice of the adaptation stop threshold of ADF error ε can be determined pragmatically based on experiments. In the separation simulations, the threshold ε was set to be 0.0005. The within-block iteration number was further hard limited to be between 3 and 8, to avoid potential divergences caused by excessive number of iterations. Figure 3.11 shows the within-block iteration count. It is obvious that at the beginning of adaptation, maximum number of iterations were

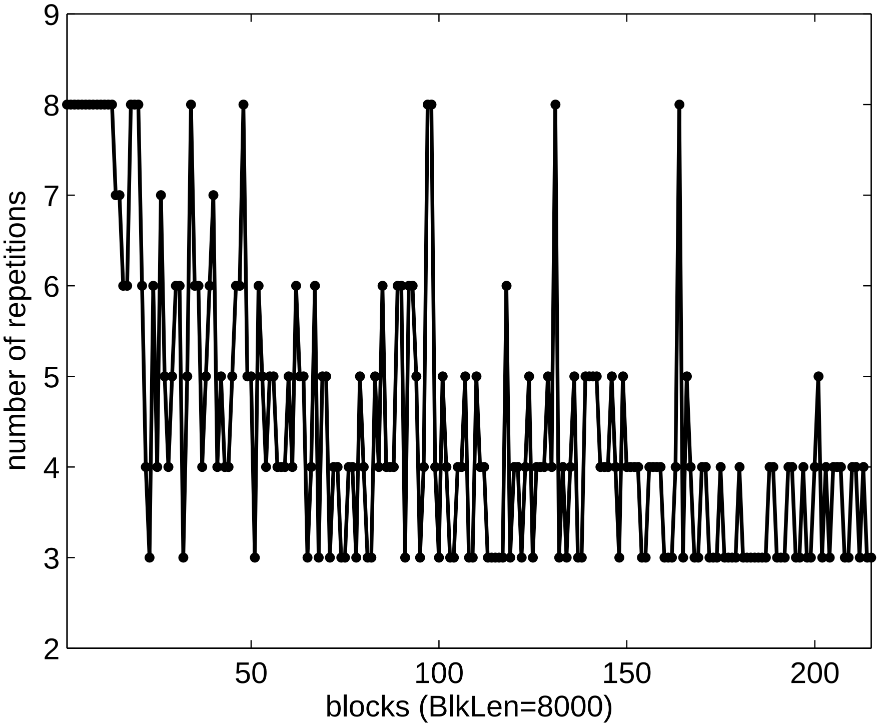


Figure 3.11: Within-block iteration counts for block-iterative ADF

reached at each block. As ADF system starts to converge, the general trend is that the actual number of iterations begin to decrease. However, for blocks that contain abundant information for separation, the iteration counts can still reach maximum, even if near convergence.

The effects of block length B on block-iterative ADF are analyzed by comparison of adaptation stability and convergence rate under various choices of B. The convergence rate with four different block durations of $750ms$, $500ms$, $250ms$ and $62.5ms$ are shown in Figure 3.12. For very short block length, the block-iteration of ADF was unstable. Large block length made the adaptation converge faster and more robust. After trading off between the convergence and time-delay performances, the block-length of $500ms$ was chosen in the rest experiments.

3.4.3 Comparison of separation performances

The proposed algorithms were tested in the speech separation experiments. Speech separation performances are compared in the measurements of TIR gains for the target signal. For single ADF processing, the TIR gain is the same as (1.7). For multiple-ADF post-filtering, ΔTIR is computed from (3.15).

The following cases were tested: baseline batch ADF, batch ADF with inverse-PSD, block-iterative ADF with inverse-PSD, and with and without multi-ADF post-filtering for each cases. For post-filtering, the number of microphone pairs ranged from one to four. To enable meaningful comparison across the cases of with and without inverse-PSD, the computation of TIRs for both input and output speech signals are based on the signal components after the whitening filtering of inverse-PSD. The TIR for mixtures are $0.53dB$ and $-0.56dB$ at two microphones. After prewhitening, the initial conditions are $TIR_{y_1} = 4.22dB$ and $TIR_{y_2} = -3.01dB$, for target speech and interference, respectively.

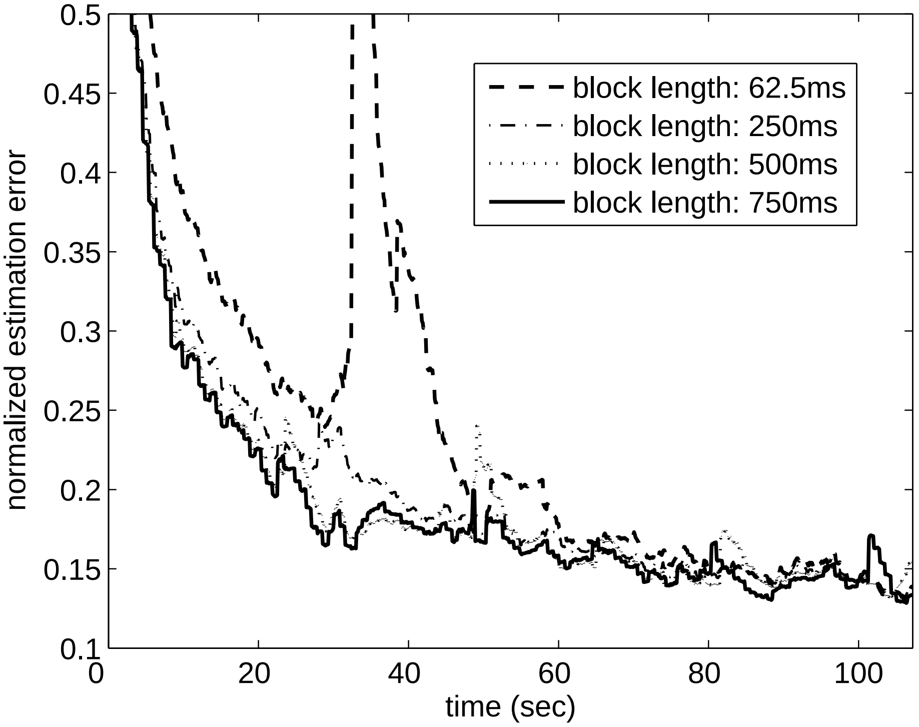


Figure 3.12: Convergence behavior of block-iterative ADF vs. block-size

Number of microphones	baseline	inverse-PSD	block-iterative+ inverse-PSD
One pairs	3.22	8.01	11.53
Two pairs	5.51	10.22	14.26
Three pairs	6.71	11.04	14.19
Four pairs	8.03	11.83	14.66

Table 3.1: Target-to-interference ratio (TIR) gains (in dB) for the target speech, from the first iteration of ADF algorithms with and without multi-ADF post-filtering.

The separation results in TIR gains listed in Table 3.1 demonstrate significant enhancement over baseline ADF by the proposed algorithms of prewhitening filtering, block-iterative implementation, and multi-ADF post-filtering. The combination of these three techniques achieved the highest separation performance.

Chapter 4

VARIABLE STEP-SIZE ADF TECHNIQUES

In this chapter, we introduce the variable step size (VSS) algorithms for ADF adaptation. Two VSS methods [61] are integrated into the ADF algorithm to improve the performance of competing speech separation. The first VSS method applies gradient adaptive step-size (GAS) to increase ADF convergence rate. Under some simplifying assumptions, the GAS technique is generalized to allow its combination with additional VSS techniques for ADF algorithm. The second VSS method is based on numerical error analysis of ADF estimates under a simplified signal model to decrease steady state filter error. An integration of both techniques into ADF successfully improved convergence rate and reduced steady-state error when tested with clean TIMIT speech convolutively mixed by reverberant room impulse responses.

4.1 Introduction

From the perspective of adaptive signal processing, many BSS algorithms are controlled by some step-size. It is well known from the analysis of the LMS family of adaptive algorithms [62, 63] that better trade-off between convergence rate and mean steady-state error (MSE) could be achieved by suitable choices of step-sizes. The basic idea behind the method of variable or adaptive step-size is to use a large step-size at the beginning of adaptation to achieve fast convergence and to reduce the step-size when the system approaches convergence so as to decrease the steady state error. For the ADF speech separation model, Yen and Zhao [20] applied decreasing gain factors [64] to accelerate ADF convergence. The step-sizes for each filter coefficient decreases whenever the most recent two consecutive adaptations change signs. Here, we attempt to provide a more general framework for VSS-ADF algorithms that incorporates several gain factors of various roles into the variable step-size scheme.

4.2 Gradient adaptive gain factor

The gradient adaptive step-size (GAS) algorithm [62] was proposed as one of the many VSS LMS algorithms for the purpose described above. Douglas and Cichocki [65] introduced GAS into a natural gradient based blind source separation (BSS) algorithm. It is fortunate that ADF algorithm is very similar to basic LMS algorithm in the form of adaptation, and it is possible to apply to ADF similar techniques used in the derivations of VSS LMS algorithms.

Based on the vector formulations in Chapter 2, the vector representations of basic ADF (2.2) and (2.39) are obtained. To facilitate the derivations in this chapter, the baseline ADF algorithm in vector form is summarized below

$$v_i(t) = y_i(t) - \mathbf{g}_{ij}(t) y_j(t), \tag{4.1}$$

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t) v_i(t) \mathbf{v}_j(t). \tag{4.2}$$

To introduce GAS into ADF, we can adapt step-size in the negative direction of the instantaneous gradient of ADF output cross-correlation. In the meantime, we introduce additional gain factors into the adaptation equation. The VSS ADF filter update modified from (4.2) becomes

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \gamma_{ij}(t)\mu_{ij}(t)v_i(t)\mathbf{v}_j(t), \tag{4.3}$$

where the product of two separate variable gains, $\gamma_{ij}(t)$ and $\mu_{ij}(t)$ forms the overall step size. The gain factor $\gamma_{ij}(t)$ represents the component of VSS that is adjustable through gradient adaptive procedures, and the gain factor $\mu_{ij}(t)$ absorbs the rest "non-gradient adaptive" procedures (e.g., normalization as in (1.6)) that are independent of GAS.

The update of the GAS gain-factor γ_{ij} at time $t+1$ aims at the minimization of the instantaneous criterion function

$$J_{\gamma_{ij}}(t+1) = \frac{1}{2}\|v_i(t+1)\mathbf{v}_j(t+1)\|^2, \tag{4.4}$$

which measures the norm of instantaneous cross-correlation vector of ADF at time $t+1$. The minimization of (4.4) with respect to $\gamma_{ij}(t+1)$ is performed iteratively by the gradient-descent adaptation

$$\gamma_{ij}(t+1) = \gamma_{ij}(t) - \varepsilon \frac{\partial}{\partial \gamma_{ij}(t)} J_{\gamma_{ij}}(t+1). \tag{4.5}$$

From (4.1) and (4.3), it is observed that $\mathbf{v}_j(t+1)$ is determined by the filters $\mathbf{g}_{ji}$ that connect all the ADF inputs to the j-th output, not by the filters $\mathbf{g}_{ij}$ that link inputs to the i-th output. Therefore, we can assume that $\mathbf{v}_j(t+1)$ is independent of $\mathbf{g}_{ij}$ and then obtain

$$\frac{\partial}{\partial \gamma_{ij}(t)} J_{\gamma_{ij}}(t+1) = v_i(t+1)\mathbf{v}_j^T(t+1)\mathbf{v}_j(t+1) \cdot \frac{\partial}{\partial \gamma_{ij}(t)} v_i(t+1), \tag{4.6}$$

where, by using (4.1) for time $t+1$ and (4.3), the gradient in the RHS of (4.6) can be expressed by

$$\frac{\partial}{\partial \gamma_{ij}(t)} v_i(t+1) = -\mu_{ij}(t) v_i(t) \mathbf{y}_j^T(t+1) \mathbf{v}_j(t). \tag{4.7}$$

The derivation of 4.7 also utilizes the simplifying assumption that the non-gradient-adaptive gain factor $\mu_{ij}(t)$ is independent of the GAS gain factor $\gamma_{ij}(t)$.

Finally, substituting (4.7) into (4.6) and then substituting (4.6) back to (4.5), the adaptation of GAS for ADF is obtained as

$$\gamma_{ij}(t+1) = \gamma_{ij}(t) + \varepsilon \mu_{ij}(t) v_i(t) v_i(t+1) \mathbf{v}_j^T(t+1) \mathbf{v}_j(t+1) \mathbf{v}_j^T(t) \mathbf{y}_j(t+1). \tag{4.8}$$

This GAS adaptation can be re-arranged and expressed in the form

$$\gamma_{ij}(t+1) = \gamma_{ij}(t) + \varepsilon \mu_{ij}(t) \|\mathbf{v}_j(t+1)\|^2 \langle \hat{\mathbf{r}}_{v_i \mathbf{v}_j}(t), \hat{\mathbf{r}}_{v_i \mathbf{y}_j}(t+1) \rangle, \tag{4.9}$$

where $\langle \cdot, \cdot \rangle$ denotes vector inner product and $\| \cdot \|$ is vector norm.

4.3 Gain factor based on source energy

For the non-gradient-adaptive gain factor $\mu_{ij}(t)$, an effective choice can be made from the error analysis of ADF. We consider the two-speaker-two-microphone signal mixing and ADF system models as a whole, as illustrated in Fig. 4.1, and do numerical analysis on ADF estimate errors. Under simplified conditions and assuming parameters of the whole system are known, we could extract information useful for the selection of non-gradient-adaptive gain factors.

Under a finite-length assumption for the convolutive-mixing filters and adopting the vector representation similar to Section 4.2, when filter length is long enough, the

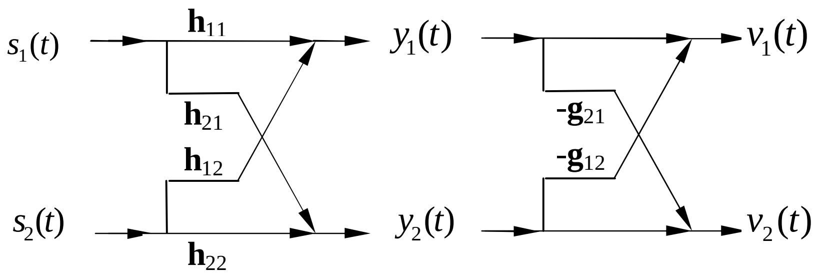


Figure 4.1: Speech mixing and ADF separation system for error analysis.

time domain speech mixing model can be approximately described by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}} \cdot \bar{\mathbf{s}}, \tag{4.10}$$

where $\bar{\mathbf{s}} = \left[\bar{\mathbf{s}}_1^T(t)\ \bar{\mathbf{s}}_2^T(t)\right]^T$ is the $(8N-6)\times 1$ source signal vector, and the mixing filter matrix is in the form of

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{H}}_{11} & \tilde{\mathbf{H}}_{12} \\ \tilde{\mathbf{H}}_{21} & \tilde{\mathbf{H}}_{22} \end{bmatrix} \tag{4.11}$$

with the $(2N-1)\times(4N-3)$ matrix block components

$$\tilde{\mathbf{H}}_{ij} = \begin{bmatrix} h_{ij}(0) & h_{ij}(1) & \cdots & h_{ij}(2N-2) & 0 & \cdots & 0 \\ 0 & h_{ij}(0) & \cdots & h_{ij}(2N-3) & h_{ij}(2N-2) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{ij}(0) & h_{ij}(1) & \cdots & h_{ij}(2N-2) \end{bmatrix}. \tag{4.12}$$

Since speech signals are complex and a direct analysis based on (4.10) and (2.2) is difficult, we consider the simplified case that sources are white with zero mean, i.e.,

$$\mathbf{R}_{\bar{\mathbf{s}}\bar{\mathbf{s}}} = blkdiag\left(p_1\mathbf{I}_{4N-3}, p_2\mathbf{I}_{4N-3}\right), \tag{4.13}$$

where p_1 and p_2 are variances of sources s_1 and s_2 respectively.

It is obvious from (2.24) that the accuracy and stability of new ADF estimates $\mathbf{g}_{ij}^{est}$'s, based on current $\mathbf{g}_{ij}$'s, are determined by the following equation,

$$\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}^T \mathbf{g}_{ij}^{est} = \mathbf{r}_{y_i\mathbf{v}_j}. \tag{4.14}$$

Following the development in the Appendix B under white-source assumption, we obtain

$$\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j} = p_i\mathbf{T}_i^j + p_j\mathbf{T}_j^j = p_i\left(\mathbf{T}_i^j + \frac{p_j}{p_i}\mathbf{T}_j^j\right), \tag{4.15}$$

with

$$\mathbf{T}_i^j = \left[\mathbf{I}_N\ \mathbf{0}_{N\times(N-1)}\right]\tilde{\mathbf{H}}_{ji}\left(\left[\mathbf{I}_N\ \mathbf{0}_{N\times(N-1)}\right]\tilde{\mathbf{H}}_{ji} - \mathbf{G}_{ji}\tilde{\mathbf{H}}_{ii}\right)^T, \tag{4.16}$$

$$\mathbf{T}_j^j = \left[\mathbf{I}_N\ \mathbf{0}_{N\times(N-1)}\right]\tilde{\mathbf{H}}_{jj}\left(\left[\mathbf{I}_N\ \mathbf{0}_{N\times(N-1)}\right]\tilde{\mathbf{H}}_{jj} - \mathbf{G}_{jj}\tilde{\mathbf{H}}_{ij}\right)^T, \tag{4.17}$$

and

$$\mathbf{r}_{\mathbf{y}_i\mathbf{v}_j} = p_i\boldsymbol{\xi}_i^j + p_j\boldsymbol{\xi}_j^j = p_i\left(\boldsymbol{\xi}_i^j + \frac{p_j}{p_i}\boldsymbol{\xi}_j^j\right), \tag{4.18}$$

with

$$\boldsymbol{\xi}_i^j = \left(\mathbf{H}_{ji} - \mathbf{G}_{ji}\tilde{\mathbf{H}}_{ii}\begin{bmatrix}\mathbf{I}_{2N-1}\\ \mathbf{0}_{(2N-2)\times(2N-1)}\end{bmatrix}\right)\tilde{\mathbf{h}}_{ii}, \tag{4.19}$$

$$\boldsymbol{\xi}_j^j = \left(\mathbf{H}_{jj} - \mathbf{G}_{ji}\tilde{\mathbf{H}}_{ij}\begin{bmatrix}\mathbf{I}_{2N-1}\\ \mathbf{0}_{(2N-2)\times(2N-1)}\end{bmatrix}\right)\tilde{\mathbf{h}}_{ij}, \tag{4.20}$$

where $\mathbf{H}_{ji}$ and $\mathbf{H}_{jj}$ are the $N\times(2N-1)$ upper-left sub-matrices of $\tilde{\mathbf{H}}_{ji}$ and $\tilde{\mathbf{H}}_{jj}$, respectively, and $\tilde{\mathbf{h}}_{ii}$ and $\tilde{\mathbf{h}}_{ij}$ the $(2N-1)\times 1$ are the direct-path and cross-coupling impulse response vectors, respectively, of the acoustic mixing system, defined as

$$\tilde{\mathbf{h}}_{ij} = \left[h_{ij}(0),\cdots,h_{ij}(2N-1)\right]^T. \tag{4.21}$$

Both (4.15) and (4.18) indicate that the contributions of the i^{th} and j^{th} sources to the input-output cross correlations are functions of source powers p_i and p_j. Based on (4.15)-(4.20), numerical analyses can be performed on the condition of $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$ and on the error of $\mathbf{g}_{ij}^{est}$. As filter length N grows, the condition worsens. To alleviate the negative effect on filter estimates, a regularized solution of (4.14) is used,

$$\mathbf{g}_{ij}^{est} = \left(\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}^T\right)^{-1} \mathbf{r}_{y_i\mathbf{v}_j} = \Phi^T\Theta\Psi\mathbf{r}_{y_i\mathbf{v}_j}, \tag{4.22}$$

where $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}^T = \Psi^T\Sigma\Phi$ is the singular value decomposition (SVD) of $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}^T$. The singular values are $\Sigma = diag(\sigma_1, \cdots, \sigma_N)$, and the diagonal elements of the matrix $\Theta = diag(\theta_1, \cdots, \theta_N)$ are determined by a regularized inverse of singular values as follows.

$$\theta_n = \begin{cases} 1/(\sigma_n + \delta_0\sigma_{max}) & if\sigma_n > \delta_1\sigma_{max} \\ 0 & if\sigma_n \le \delta_1\sigma_{max} \end{cases} \tag{4.23}$$

Numerical analyses are performed based on known impulse response data measured in a room with reverberation time $T_{60} = 0.3sec$ [60], with the same microphone speaker configurations as shown in Figure 3.9. For filter length $N = 400$ ($25ms$), the error of new ADF estimate $\mathbf{g}_{12}^{est}$ based on current value $\mathbf{g}_{12}$, as a function of power ratio p_2/p_1, is shown in Figure 4.2, where the normalized ADF errors for individual filter $\mathbf{g}_{ij}$'s are defined as

$$\epsilon_{ij}^2 = \|\mathbf{g}_{ij}^{est} - \mathbf{g}_{ij}^o\|^2/\|\mathbf{g}_{ij}^o\|^2, \tag{4.24}$$

similar to the definition of averaged ADF errors (3.17). The thresholds in (4.23) are chosen to be $\delta_1 = 0.05$ and $\delta_0 = 0.01$. The values of current filters are set to be 0.6 times the ideal values, i.e., $\mathbf{g}_{ij} = 0.6\mathbf{g}_{ij}^o$, to simulate near-convergence cases.

The result of the above error analysis under white-source assumption shows that the lower the power of the j^{th} source is, the higher the error will be for the estimate of filter $\mathbf{g}_{ij}$. The filter error reaches its minimum when two sources are balanced in

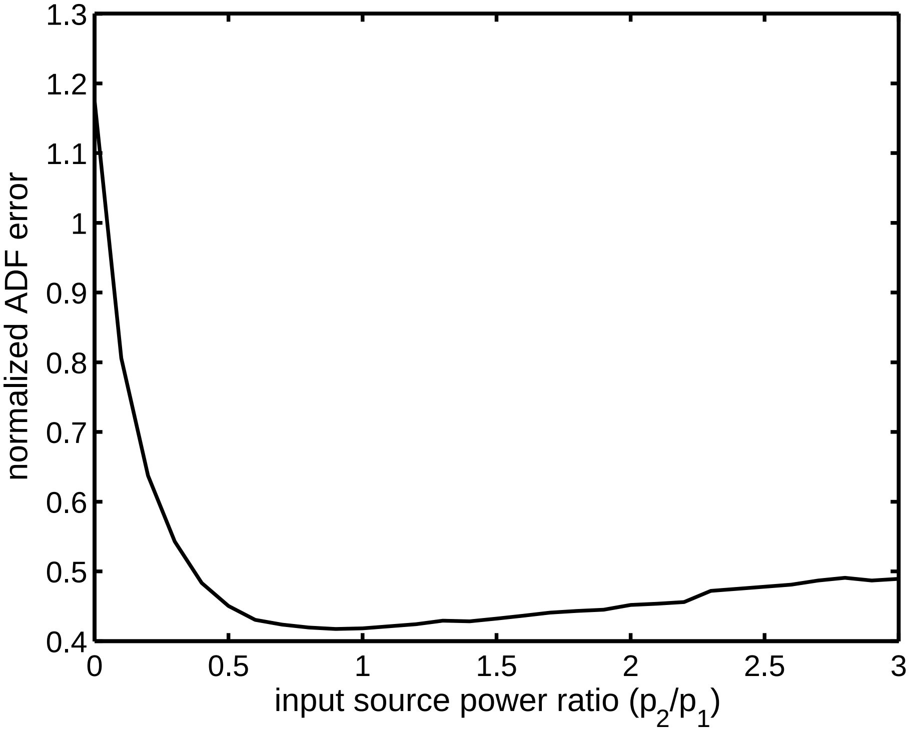


Figure 4.2: Numerical analysis of ADF errors as a function of input source power ratio p_2/p_1

power, and unequal source strengths will cause a larger error. Based on this a priori knowledge, a heuristic VSS gain factor is proposed to discount the step-size for filter $\mathbf{g}_{ij}$ when source j is weak. By decreasing the step-sizes during the time-intervals that correspond to weak energy periods, we attempt to reduce the ADF filter estimation error by counting more on the "source-strong" input data that have more information about the j^{th} source.

Since source powers are unavailable, ADF output powers are used as approximations. The non gradient-adaptive gain factor $\mu_{ij}(t)$ can now be modified as a scalar that discounts the adaptation of filters by

$$\mu_{ij}(t) = \frac{\sigma_{v_j}^2(t)}{\sigma_{v_1}^2(t) + \sigma_{v_2}^2(t)} \cdot \mu(t). \tag{4.25}$$

where $\mu(t)$ can be other choices of variable step-sizes, and $\sigma_{v_j}^2(t)$'s are the powers of ADF output signals estimated from short-term segments.

Another observation can also provide a useful cue for a heuristic gain factor. In practice, filter coefficients in our algorithm are initialized with zeros, producing ADF output powers that are equal to input powers; as ADF algorithm converges, filter values will deviate from zero and in the mean time reduce ADF output powers. Therefore, we can also utilize this property and introduce another discounting scalar

$$\frac{\sigma_{v_1}^2(t) + \sigma_{v_2}^2(t)}{\sigma_{y_1}^2(t) + \sigma_{y_2}^2(t)} \tag{4.26}$$

to reduce the step-size in (4.25) when the system is in convergence. If choosing to incorporate this heuristic gain into (4.25), we obtain

$$\mu_{ij}(t) = \frac{\sigma_{v_j}^2(t)}{\sigma_{y_1}^2(t) + \sigma_{y_2}^2(t)} \cdot \mu(t). \tag{4.27}$$

However, when noise is present, this discounting scalar is decreased even if the ADF filters estimated are not doing separation, as long as they are doing noise-cancelation to some extent. Since the convergence of filter estimates to the ones that separate speech sources and to those that cancels background noises are both reflected by the decreasing values of (4.26), it will no longer be suitable to measure the level of convergence for separation filters. Therefore, for noisy conditions discussed in subsequent chapters, (4.25) will be used, instead of (4.27).

4.4 VSS-ADF Implementation and Experiments

4.4.1 Implementation of gain factors

Although white source assumption is a simplification, the results of the above ADF error analysis could be carried over to speech signals in an approximate sense. In practice, pre-whitening processing is performed, which makes ADF source signals closer to the white-assumption. Therefore, the applications of non-GAS adaptation gains in (4.25) are also valid for pre-whitened real speech signals. In (4.25), the short-term powers of separated speech signals ($\sigma_{v_i}^2$'s) are estimated from the most recent 37.5ms samples from ADF outputs.

The GAS update term in (4.9) is a function of filter length N. As the filter length N increases, its magnitude increases with the scale of N^2. This makes it difficult for the setting of the step size ε for GAS adaptation. To eliminate this effect, (4.9) is normalized by N^2. As in LMS, the adaptation of GAS gain factor can be improved by introducing forgetting factor into step size update [66]. It is actually a first order recursive processing and it helps to improve the stability of step-size adaptation by smoothing out noises in the process. This technique is incorporated into VSS-ADF to improve performance. Therefore, the GAS update equation (4.9) implemented with forgetting factor ρ, $0 << \rho < 1$, is

$$\gamma_{ij}(t+1) = \rho\gamma_{ij}(t) + \varepsilon\mu_{ij}(t)\|\mathbf{v}_j(t+1)\|^2\langle\hat{\mathbf{r}}_{v_i\mathbf{v}_j}(t), \hat{\mathbf{r}}_{v_i\mathbf{y}_j}(t+1)\rangle/N^2, \tag{4.28}$$

where $\mu_{ij}(t)$ is the non-GAS gain factor from (4.25). Since (4.28) requires the norm of the newest ADF output vectors $\|\mathbf{v}_j(t+1)\|$'s, the GAS gain factor update is performed after each ADF filter adaptation.

4.4.2 Separation experiments and convergence performances

Speech separation with a single microphone-pair ADF model was carried out for noise-free speech mixtures to evaluate the convergence and separation performances of the proposed VSS-ADF algorithm. The speech mixtures were generated from the TIMIT speech database and ATR acoustic database in the same way as described in Section 3.4. The single microphone pair used was 3 and 15.

The performance of the integrated VSS-ADF algorithm was evaluated by normalized filter errors, shown in Fig. 4.3, and it is compared with block-iterative ADF and baseline ADF with and without prewhitening. To analyze the effects of GAS and non-GAS gain factors on system convergence properties, the VSS-ADF algorithm with only GAS gain γ_{ij} and with only non-adaptive gain $\gamma_c\mu_{ij}(t)$'s were also tested, where $\gamma_c = 2.4$ was the average value of the GAS gain γ_{ij} shown in Fig. 4.4. The following conditions were used in all experiments of this section: filter lengths $N = 400$, $\alpha = 0.005$ (0.0035 for GAS-only case due to the problem of instability), the inverse-PSD type of prewhitening processing was the same as in Section 3.1. For VSS-ADF algorithms with GAS gain adaptations, the initial value of GAS gain was set to zero, i.e., $\gamma_{ij}(0) = 0$, the gain for step-size update was $\varepsilon = 4 \times 10^{-4}$, the forgetting factor was $\rho = 0.999994$.

VSS-ADF Methods	VSS Gains	Computation Equations
Non-GAS	$\gamma_c \mu_{ij}(t)$	(4.27),(1.6)
GAS-only	$\gamma_{ij}(t)\mu(t)$	(4.28), (1.6)
Combined	$\gamma_{ij}(t)\mu_{ij}(t)$	(4.28), (4.27), (1.6)

Table 4.1: Three adaptive gain factor options for VSS-ADF algorithms.

	Mixture	Baseline	Block-Iterative	VSS-ADF
TIR (dB)	0.53	7.29	8.57	12.66

Table 4.2: Target-to-interference ratio (TIR) in dB for VSS-ADF separation and comparison with baseline and block-iterative ADF.

The VSS schemes evaluated are summarized in Table 4.1. With the absence of the gain factor (4.27), the GAS-only adaptation is less accurate in its steady state estimation of ADF filters, as shown in Figure 4.3. The increased ADF estimation error of GAS-only VSS also caused divergence when using the adaptation constant $\gamma = 0.005$, and requires to work under a smaller value of $\gamma = 0.0035$, compared with the other two options.

The adaptation histories of the gradient adaptive step-sizes are illustrated in Figure 4.4. The convergence rate of the proposed VSS-ADF algorithm that combines both types of step-sizes is shown, in ADF estimation errors, to yield a convergence speed significantly faster than baseline algorithms (Figure 4.3), where the speed is comparable to block-iterative implementation but with much less delay than a block. The combined method also has a lower steady-state error which comes from the error-reducing gain factor (4.27) in the total step size.

As demonstrated by the TIR results in Table 4.2, the separation performance was significantly improved by the integration of both types of VSS gain factors.

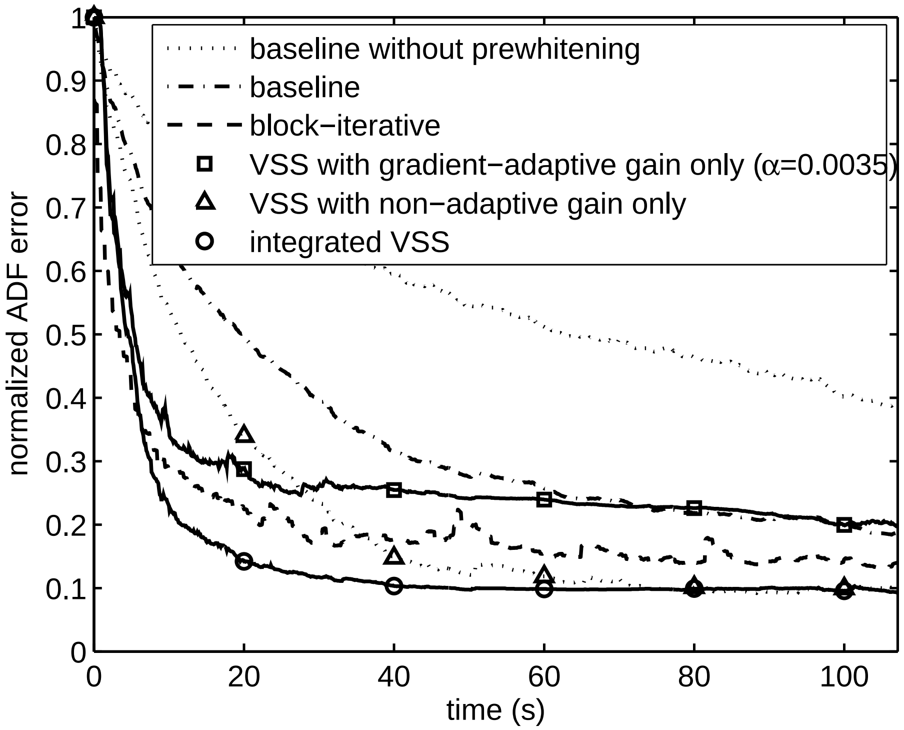


Figure 4.3: Normalized ADF error for VSS-ADF

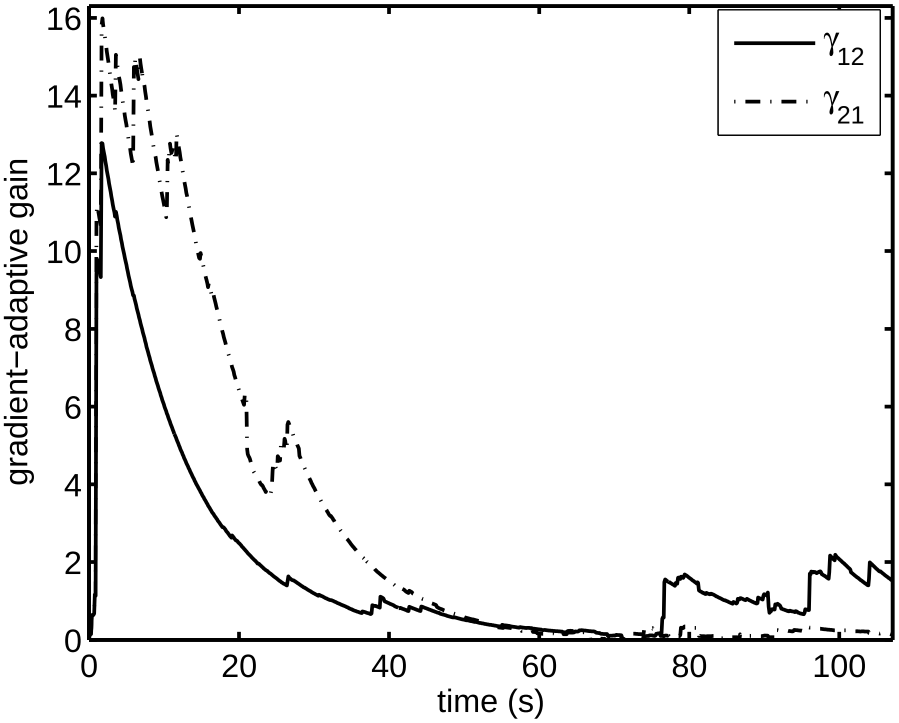


Figure 4.4: Evaluation of gradient-adaptive gain factors.

Chapter 5

NOISE COMPENSATED ADF

When speech interference occurs with simultaneous background noise, the performances of ADF model will degrade. We need to model ADF separation system by taking noise effects into account. In this chapter, we analyze the noise effects on ADF system and present the noise-compensated ADF (NC-ADF) [67, 68] algorithm for the application in real diffuse noise environments. Fast algorithms are developed for NC-ADF and simplified NC-ADF is also provided for uncorrelated noises.

5.1 Noisy ADF model and analysis of noise effects

For practical applications of ADF in speech recognition, it is important to investigate into the noise effects in speech separation and provide potential solutions for the degradations caused by noise. From the more general perspective of noisy BSS models, the difficulties caused by noise to separation systems are two folds:

1. the presence of noise may affect the working conditions of BSS algorithms and, therefore, degrade the separation performances;

2. a BSS algorithm by itself, aiming mainly at source separation, has limited abilities in suppressing diffuse noise.

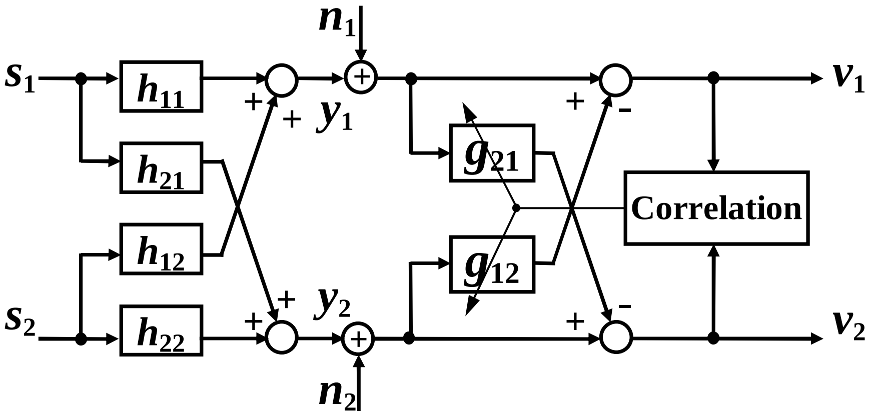


Figure 5.1: Speech mixing and ADF separation system in noise.

For the first problem, the general approach to improve separation performance in noisy BSS is to do "bias removal" [10]. How the separation performances are affected by noises depends on specific algorithms, and some noise compensation (NC) algorithms, e.g., [13], were proposed for a natural gradient based convolutive separation model. For the second problem about output noise suppression, the limitations of BSS in noise suppression has been studied. Araki *et. al.*, [69, 70], established the mechanism similarities between BSS and adaptive null beamformer. In fact, BSS could be viewed as a filter and subtractive adaptive beam former. Asano *et. al.* [71] grouped both approaches into "spatial inverse" type of processing and pointed out that they are only able to suppress directional interferences but not ambient noises which are omni-directional. Therefore, efforts should also be devoted to the reduction of output noise in addition to ADF speech separation. This problem will be discussed later as the topic of adaptive speech enhancement in Chapter 6. We will solve the first problem by analyzing noisy ADF model in this section.

In the following, we mainly consider stationary or quasi-stationary noise, especially the real diffuse noise, in room environments. Although theories to deal with stationary noises are well established, they should be applied properly with considerations of the properties of ADF separation model.

With the vector formulation of Section 2.1, the I/O relations for the noisy ADF system in Figure 5.1 can be described in terms of signal vectors as

$$\mathbf{v}_n = \mathbf{G}(\tilde{\mathbf{y}} + \tilde{\mathbf{n}}), \tag{5.1}$$

and in terms of second order statistics as

$$\mathbf{R}_{\mathbf{v}_n\mathbf{v}_n} = \mathbf{R}_{vv} + \mathbf{R}_{\boldsymbol{\eta}\boldsymbol{\eta}}, \tag{5.2}$$

assuming that noise is uncorrelated with speech. In the above, $\tilde{\mathbf{y}}(t) = \left[\tilde{\mathbf{y}}_1^T(t), \tilde{\mathbf{y}}_2^T(t)\right]^T$ and $\tilde{\mathbf{n}} = \left[\tilde{\mathbf{n}}_1^T(t), \tilde{\mathbf{n}}_2^T(t)\right]^T$ are $(4N-2) \times 1$ vectors of clean speech mixture and noise, respectively. Clean speech at ADF output is $\mathbf{v}(t) = \left[\mathbf{v}_1^T(t), \mathbf{v}_2^T(t)\right]^T$, and the output noise component $\boldsymbol{\eta}(t) = \left[\boldsymbol{\eta}_1^T(t), \boldsymbol{\eta}_2^T(t)\right]^T$. From the I/O relations in correlation matrix (2.9) and (2.10), we have the following correlation vector counterparts of I/O relations

$$\mathbf{r}_{v_i\mathbf{v}_j} = \mathbf{r}_{y_i\mathbf{y}_j} - \mathbf{G}_{ji}\mathbf{r}_{y_i\tilde{\mathbf{y}}_i} - \mathbf{R}_{\mathbf{y}_j\mathbf{y}_j}\mathbf{g}_{ij} + \mathbf{G}_{ji}\mathbf{R}_{\tilde{\mathbf{y}}_i\mathbf{y}_j}\mathbf{g}_{ij}, \tag{5.3}$$

$$\mathbf{r}_{v_i\mathbf{v}_i} = \mathbf{r}_{y_i\mathbf{y}_i} - \mathbf{G}_{ij}\mathbf{r}_{y_i\tilde{\mathbf{y}}_j} - \mathbf{R}_{\mathbf{y}_i\mathbf{y}_j}\mathbf{g}_{ij} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{y}_j}\mathbf{g}_{ij}. \tag{5.4}$$

It is obvious that noise component of ADF output $\boldsymbol{\eta}$ also satisfy the same correlation vector I/O equations as

$$\mathbf{r}_{\eta_i\boldsymbol{\eta}_j} = \mathbf{r}_{n_i\mathbf{n}_j} - \mathbf{G}_{ji}\mathbf{r}_{n_i\tilde{\mathbf{n}}_i} - \mathbf{R}_{\mathbf{n}_j\mathbf{n}_j}\mathbf{g}_{ij} + \mathbf{G}_{ji}\mathbf{R}_{\tilde{\mathbf{n}}_i\mathbf{n}_j}\mathbf{g}_{ij}, \tag{5.5}$$

$$\mathbf{r}_{\eta_i\boldsymbol{\eta}_i} = \mathbf{r}_{n_i\mathbf{n}_i} - \mathbf{G}_{ij}\mathbf{r}_{n_i\tilde{\mathbf{n}}_j} - \mathbf{R}_{\mathbf{n}_i\mathbf{n}_j}\mathbf{g}_{ij} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{n}}_j\mathbf{n}_j}\mathbf{g}_{ij}. \tag{5.6}$$

It can be seen that as filters $\mathbf{g}_{ij}$ evolve during adaptation, the noise properties at ADF output vary over time. The cross-correlation term (5.5) causes a bias in the filter adaptation procedure (2.38) and it should be compensated for. The auto-correlation in (5.6) represents ADF output noise statistics and it needs to be removed to enhance the separated speech.

5.2 Noise Compensated ADF

The presence of noise deteriorates the separation performance of baseline ADF system adapted by (2.39), because the objective function in the form of (2.28) becomes

$$J_{n_{ij}} = \frac{1}{2}\mathbf{r}^T_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}}\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}}, \tag{5.7}$$

which contains bias caused by output noise cross-correlations. As shown in (5.5), the noise component in ADF output cross-correlation varies as filters $\mathbf{g}_{ij}$ evolve from their initial states to convergence, or as the filters adapt to track changing acoustic paths. The time-varying noise effect on ADF can be reduced by using a noise-compensated objective function, which is based on the estimate of speech cross correlation $\mathbf{r}_{v_i\mathbf{v}_j} = \mathbf{r}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}$, i.e.,

$$J'_{ij} = \frac{1}{2}\left(\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}\right)^T\left(\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}\right). \tag{5.8}$$

Based on (5.8) and following the same derivation as clean speech ADF, the noise-compensated ADF (NC-ADF) [68] is obtained as

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t)(v_{n_i}(t)\mathbf{v}_{n_j}(t) - \hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}(t)), \tag{5.9}$$

where $\mu(t)$ is the step-size, and the compensation term $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}(t)$ is the estimate of output noise cross-correlation $\mathbf{r}_{\eta_i\boldsymbol{\eta}_j}(t) = E\{\eta_i(t)\boldsymbol{\eta}_j(t)\}$. By (5.5), the estimate $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}(t)$ can be directly computed from the estimates of input noise statistics

$$\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j} = \hat{\mathbf{r}}_{n_i\mathbf{n}_j} - \mathbf{G}_{ji}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i} - \hat{\mathbf{R}}_{\mathbf{n}_j\mathbf{n}_j}\mathbf{g}_{ij} + \mathbf{G}_{ji}\hat{\mathbf{R}}_{\tilde{\mathbf{n}}_i\mathbf{n}_j}\mathbf{g}_{ij}. \tag{5.10}$$

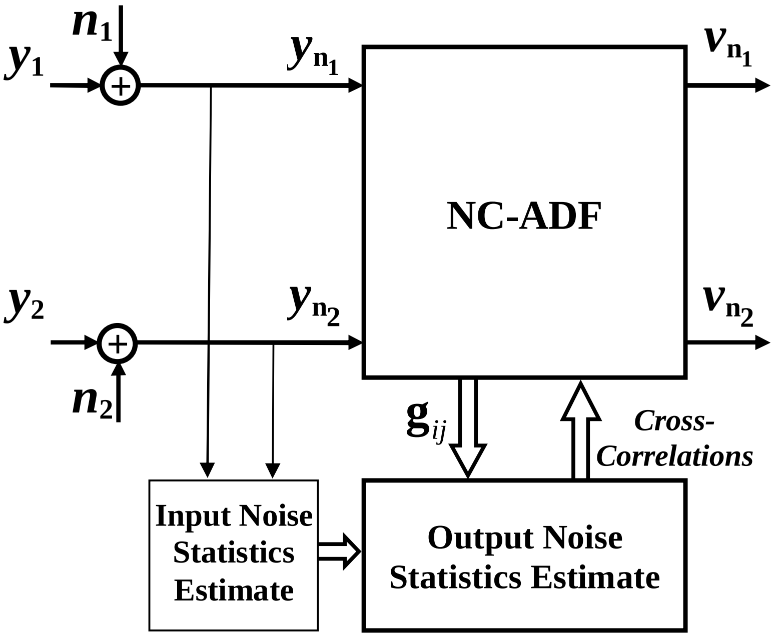


Figure 5.2: Noise-compensated ADF (NC-ADF) system.

The proposed system for improving the separation performance of ADF in noise is shown in Figure 5.2, where the noise effects on the adaptation procedure, including the step-size computation, are to be reduced to achieve improved speech separation. Noise cross-correlations in (5.10) at the output of ADF are estimated by the module of Output Noise Statistics Estimation based on input noise statistics provided by the module of Input Noise Statistics Estimation as well as on estimates of separation filters provided by the module of NC-ADF. The noise cross-correlation statistics are used in NC-ADF for noise compensation.

For the computation of step-sizes, the source energy based VSS technique of presented in Chapter 4 is extended to include noise compensation of system output powers. Since the gradient-adaptive gain factor discussed in Section 4.2 is very sensitive to background noise, it is not applicable to the noisy scenario.

Analysis of the effect of unequal source energies on filter estimation errors in Section 4.3 revealed that the lower the relative strength of jth source, the higher the estimation error will be for the filter $\mathbf{g}_{ij}$ [61]. To reduce ADF estimation error caused by such unbalanced source energies, step-sizes can be scaled down by relative short-term powers of ADF outputs. Specifically, the source energy based step size (4.25) is applied to balance adaptation between unequally excited sources, denoted as

$$\mu_{ij}(t) = \mu(t) \cdot \hat{\sigma}^2_{v_j}(t)/\hat{\sigma}^2_{av}(t), \tag{5.11}$$

where the normalizing gain factor $\mu(t)$ was given by

$$\mu(t) = \gamma/\big(N(\sigma^2_{y_{n1}}(t) + \sigma^2_{y_{n2}}(t))\big), \tag{5.12}$$

with $\sigma^2_{y_{ni}}(t)$ the short-term power of the i-th input, and γ $(0 < \gamma < 1)$ the constant gain factor that controls convergence speed. The estimated average speech output power $\hat{\sigma}^2_{av}(t)$ is

$$\hat{\sigma}^2_{av}(t) = \big(\hat{\sigma}^2_{v_1}(t) + \hat{\sigma}^2_{v_2}(t)\big)/2 \tag{5.13}$$

Noise compensation to output power is made by subtracting noise power from the power of noisy ADF output by

$$\hat{\sigma}^2_{v_j} = \hat{r}_{v_j v_j}(0) = \hat{r}_{v_{n_j} v_{n_j}}(0) - \hat{r}_{\eta_j \eta_j}(0), \tag{5.14}$$

where the output noise power is obtained from (5.6) as

$$\hat{r}_{\eta_j \eta_j}(0) = \hat{r}_{n_j n_j}(0) - 2\mathbf{g}^T_{ji}\hat{\mathbf{r}}_{n_j \mathbf{n}_i} + \mathbf{g}^T_{ji}\hat{\mathbf{R}}_{\mathbf{n}_i \mathbf{n}_i}\mathbf{g}_{ji}. \tag{5.15}$$

5.3 Fast Implementation of NC-ADF

5.3.1 Update of compensation terms

Direct computations of noise cross-correlation vectors in NC-ADF adaptation (5.9) are not feasible for real-time applications since the terms (5.10) require matrix-vector multiplications for every time sample. For fixed speaker locations, in general the changes of ADF filters are small within short time intervals (e.g., around $30ms$). The slow-change of ADF parameters and the short-term stationarity of input noise make it possible to update compensation terms in a block-wise fashion, reducing the update rate by a factor of K (block-length). To speed up NC-ADF, we first reduce the update rate for compensation terms and then utilize the Toeplitz structures of both system and correlation matrices to derive FFT-based estimation of (5.10).

The ADF adaptation bias estimate (5.10) can be rewritten as

$$\hat{\mathbf{r}}_{\eta_i \boldsymbol{\eta}_j} = \hat{\mathbf{r}}_{n_i \mathbf{n}_j} - \mathbf{a}_{ij} - \mathbf{b}_{ij} + \mathbf{c}_{ij}, \tag{5.16}$$

with

$$\mathbf{a}_{ij} = \mathbf{G}_{ji} \hat{\mathbf{r}}_{n_i \tilde{\mathbf{n}}_i}, \tag{5.17}$$

$$\mathbf{b}_{ij} = \hat{\mathbf{R}}_{\mathbf{n}_j \mathbf{n}_j} \mathbf{g}_{ij}, \tag{5.18}$$

$$\mathbf{c}_{ij} = \mathbf{G}_{ji} \mathbf{d}_{ij}, \tag{5.19}$$

$$\mathbf{d}_{ij} = \hat{\mathbf{R}}_{\tilde{\mathbf{n}}_i \mathbf{n}_j} \mathbf{g}_{ij}. \tag{5.20}$$

The computations of $\mathbf{a}_{ij}$ and $\mathbf{c}_{ij}$ share the same structure. The components of vector $\mathbf{a}_{ij}$, i.e., $a_{ij}(k)$, $k = 0, ..., N-1$, can be expressed as the last N samples, in reversed order, of the convolution $g_{ji}(n) * \xi^a_{ij}(n)$, i.e.,

$$a_{ij}(k) = g_{ji}(n) * \xi^a_{ij}(n)|_{n=2N-2-k}, \tag{5.21}$$

where

$$\xi_{ij}^{a}(n) = \hat{r}_{n_i\tilde{n}_i}(2N-2-n) \tag{5.22}$$

is the $(2N-1)$-point reverse of $\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i}$. A detailed derivation for (5.21) and (5.22) were given in Appendix C. Similarly, components of $\mathbf{c}_{ij}$ are obtained by

$$c_{ij}(k) = g_{ji}(n) * \xi_{ij}^{c}(n)|_{n=2N-2-k}, \tag{5.23}$$

with

$$\xi_{ij}^{c}(n) = d_{ij}(2N-2-n). \tag{5.24}$$

The vectors $\mathbf{b}_{ij}$ and $\mathbf{d}_{ij}$ also have a similar structure, where

$$b_{ij}(k) = g_{ij}(n) * \xi_{ij}^{b}(n)|_{n=k+N-1} \tag{5.25}$$

with

$$\xi_{ij}^{b}(n) = \hat{r}_{n_j\tilde{n}_j}(n-N+1), \tag{5.26}$$

and

$$d_{ij}(k) = g_{ij}(n) * \xi_{ij}^{d}(n)|_{n=k+N-1} \tag{5.27}$$

with

$$\xi_{ij}^{d}(n) = \hat{r}_{n_i\tilde{n}_j}(N-1-n). \tag{5.28}$$

Based on such convolutive expressions, the N-point sequences $a_{ij}(k)$, $b_{ij}(k)$, and $c_{ij}(k)$ can be computed by N_F-point FFTs ($N_F > 2N-1$). For modularity, the $(2N-1)$-point sequence $d_{ij}(k)$ can be decomposed into two N-point sub-sequences and computed with two N_F-point FFT-IFFT modules. In this way, all the sequences above only need to be zero-padded to length N_F, because only N-point results are required in each module. The rest points with aliasing are irrelevant and are discarded.

From (5.14)-(5.16), the noise-free ADF output powers used in VSS computation are estimated by

$$\hat{\sigma}_{v_j}^2 \approx \mathbf{v}_{n_j}^T \mathbf{v}_{n_j}/N - \hat{r}_{n_j n_j}(0) + 2\mathbf{g}_{ji}^T \hat{\mathbf{r}}_{n_j \mathbf{n}_i} - \mathbf{g}_{ji}^T \mathbf{b}_{ji}. \tag{5.29}$$

5.3.2 Fast block implementation of ADF

The sample-wise procedures of filtering (5.2) and adaptation (2.39) of ADF are also modified for a block-wise implementation to enable fast noise compensation. The fast computation of (5.2) can use the standard overlap-add fast convolution method [58] under the approximation that filters are constant within each block.

Assuming that a constant step-size is used within each block, a block-adaptive procedure for filter update can be obtained. For noise-free ADF, consider the m-th block covering samples from t_m to $t_m + K - 1$, and let $\mathbf{g}_{ij}^m = \mathbf{g}_{ij}(t_m)$'s denote the filters of the current block, estimated from the previous block. After obtaining ADF outputs of the m-th block by fast convolution filtering, the step-size μ_{ij}^m's can be estimated to update filters using the output data in the entire current block. The newly adapted ADF filters will be used for the separation filtering in the next block ($(m+1)$-th).

By summing up both sides of the baseline ADF adaptation (2.39) for $t = t_m, ..., t_m + K - 1$, the new filters for the next block, $\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}(t_m + K)$, can be estimated as

$$\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}^m + \mu_{ij}^m K \hat{\mathbf{r}}_{v_i \mathbf{v}_j}^m. \tag{5.30}$$

The cross-correlation estimate

$$\hat{\mathbf{r}}_{v_i \mathbf{v}_j}^m = \hat{\mathbf{r}}_{v_i \mathbf{v}_j}(t_m) = \frac{1}{K} \sum_{k=0}^{K-1} v_i(t_m + k)\mathbf{v}_j(t_m + k). \tag{5.31}$$

can be computed by an FFT-based fast implementation [58]. The fast algorithm for (5.30) and (5.31) will be called fast ADF (FADF) in subsequent discussions.

Similar to (5.30), the block-wise NC-FADF in noisy conditions is obtained from (5.9) as

$$\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}^{m} + \mu_{ij}^{m} K(\hat{\mathbf{r}}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}}^{m} - \hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}), \tag{5.32}$$

where $\hat{\mathbf{r}}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}}^{m}$ is defined by replacing v_i and v_j with their noisy counterparts in (5.31), $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}$ is from (5.16), and the block step-size μ_{ij}^{m} is computed by the block-wise counterpart of (5.11). The normalization gain factor μ^m, corresponding to (5.12), uses ADF input powers that are estimated from samples of both current and previous blocks. To prevent over-compensation in NC-FADF, $\hat{\sigma}_{v_j}^2$ in (5.29) is set to zero when negative values occur. The denominator in (5.11) is also added a small positive number to avoid divide-by-zeros. Triangular windows $w(n) = (N-n)/N, n = 0, ..., N-1$ are applied to both correlation estimate $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}$ and ADF adaptation vectors $\hat{\mathbf{r}}_{v_i\mathbf{v}_j}^{m}$ to prevent instability.

The overlap-add method requires that $N \leq K \leq 2N$. When K=N and the FFT length N_F=$2N$, the computation of $2N$-point FFTs are distributed to the block of length N, resulting in a complexity of $O(\log N)$ per time-sample for NC-FADF, in contrast to $O(N^2)$ for direct estimation of NC terms required by matrix-vector multiplications.

5.4 Speech Separation Experiments

5.4.1 Experimental setup

The data used in the experiments clean speech data, acoustic impulse response data, and both simulated and real recorded diffuse noise. The clean speech mixtures were generated with the same procedure as in Section 3.4 with the same acoustic conditions. The speech capturing sensor pair used in this experiment was chosen to be microphones 3 and 15 in the circular array in Figure 3.9.

For the evaluation of NC-ADF and NC-FADF algorithms, both simulated and real diffuse noises were tested. The simulated noises were generated by the following procedure to be speech-shaped in spectrum

$$n_1(t) = 0.65 \sum_{k=1}^{2} a_k^{(1)} n_1(t-k) + 0.35 n_2(t) + \varepsilon_1(t), \tag{5.33}$$

$$n_2(t) = 0.6 \sum_{k=1}^{3} a_k^{(2)} n_2(t-k) + 0.4 n_1(t) + \varepsilon_2(t), \tag{5.34}$$

where $\varepsilon_i(t)$'s are white Gaussian excitations and $a_k^{(i)}$'s are linear prediction coefficients (LPC) estimated from clean TIMIT data. Real diffuse noises were recorded in a computer lab with a pair of omnidirectional microphones placed on a conference table in the center of the lab, where the microphones were the same distance apart as that of the array microphone pair. The ventilation and air-conditioning systems and 8 desktop workstations were working simultaneously, generating diffuse noises that fit the stationary assumption. The noise data are added to the clean speech mixtures obtained above to simulate noisy mixtures.

The basic setup for ADF were N=400 and γ=0.01. In all cases, preemphasis (1-z^{-1}) was applied to mixtures to remove the speech long-term spectral tilt and to reduce eigen-value dispersion for faster convergence [52]. Preemphasis enhances perceptually important speech components, and it also alters input noise properties as well as relative strengths of noise and speech measured in signal-to-noise ratio (SNR): $SNR = 10 \log_{10}(P_S/P_N)$, where P_S and P_N are speech and noise powers, respectively. In fact, the simulated speech-shaped noise spectrum was flattened by preemphasis, resulting in a loss of SNR of approximately 3dB; the recorded diffuse noise retained a significant amount of coloration and spatial correlation after preemphasis that increased SNR by 12dB through suppressing strongly correlated low frequency noise components. In subsequent discussions, SNR and target-to-interference-ratio (TIR) were evaluated on preemphasized input and output component. For FADF and NC-FADF, the block length was K=400 and FFT length was N_F=1024. Since VSS without NC would corrupt ADF adaptation at high levels of noise, it was not applied to baseline ADF (2.39) and FADF (5.30).

Figure 5.3 illustrates the power spectra of the simulated and real diffuse noises. The noise cross-power spectral densities are shown in Fig. 5.4, before and after preemphasis processing. We can see that noises were spatially correlated and there were strong colorations contained in the real diffuse noise spectrum. A 5-second segment of noise-only data preceding the speech was used to estimate input noise statistics required by NC-ADF.

5.4.2 Convergence performances

The convergence performance was evaluated in relative ADF filter estimation error recorded for each block m

$$e(m) = \frac{1}{2}\left(\frac{\|\mathbf{g}_{12}^m - \mathbf{g}_{12}^o\|^2}{\|\mathbf{g}_{12}^o\|^2} + \frac{\|\mathbf{g}_{21}^m - \mathbf{g}_{21}^o\|^2}{\|\mathbf{g}_{21}^o\|^2} \right), \tag{5.35}$$

where $\mathbf{g}_{ij}^o$'s are the ideal values of separation filters. The initial values of filters were set to be zeros. Compared with FADF, the steady-state error of NC-FADF was significantly reduced. Figures 5.5 and 5.6 show the convergence history of ADF error $e(m)$ for the separation under simulated and real diffuse noise cases, respectively, with various SNR conditions. Both results indicate significant improvements in adaptation robustness of NC-FADF over FADF. As a result, the steady state filter estimation accuracy was significantly enhanced. It is observed that as the noise level increases, the advantages of NC-FADF over baseline becomes more significant.

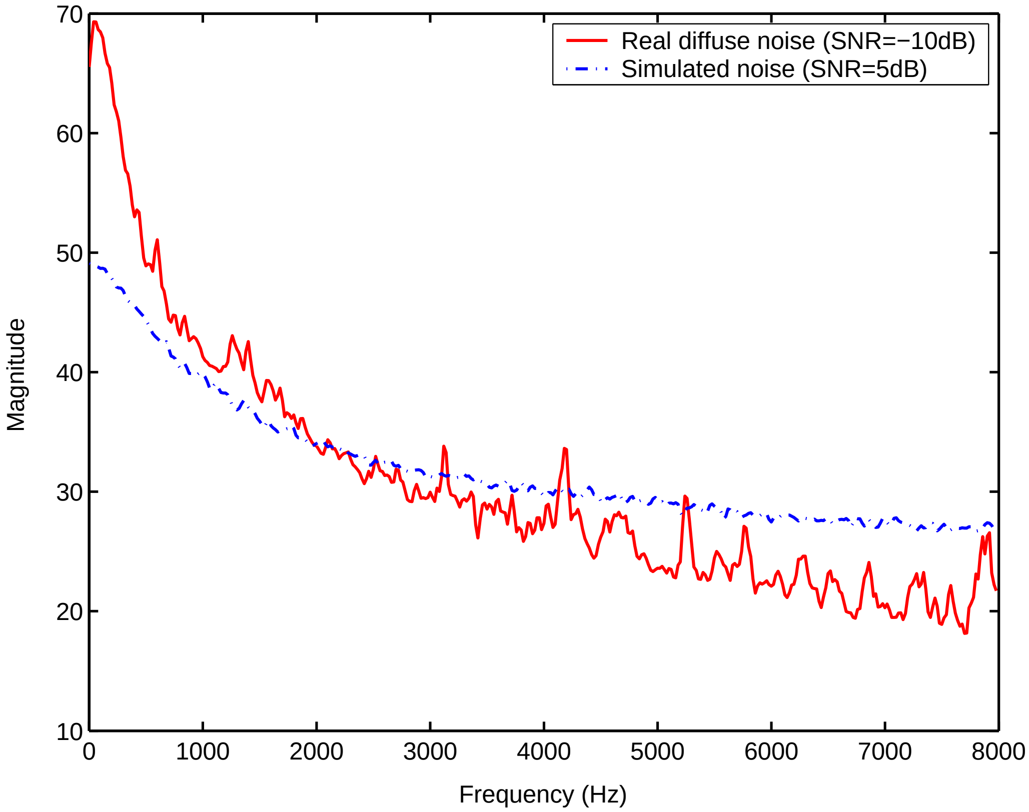


Figure 5.3: Power spectra of two types of diffuse noises: simulated speech-shaped noise and real lab noise.

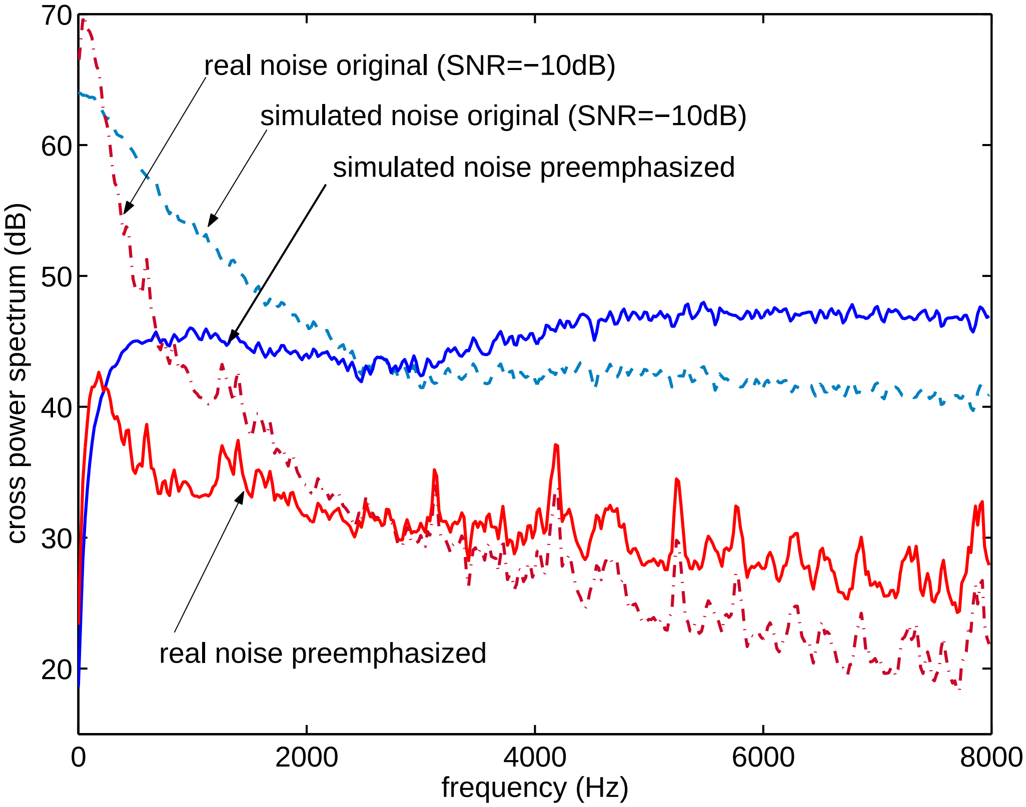


Figure 5.4: Cross power spectra of two types of diffuse noises: simulated speech-shaped noise and real lab noise.

original	preemphasized	baseline	FADF	NC-FADF
SNR	SNR(y_1, y_2)	v_1, v_2	v_1, v_2	v_1, v_2
3dB	$0.2, -1.3$	$1.7, 2.1$	$1.7, 2.0$	$7.3, 8.4$
9dB	$6.2, 4.7$	$3.0, 3.9$	$2.8, 3.6$	$9.2, 9.5$
15dB	$12.2, 10.7$	$4.7, 5.6$	$4.4, 5.2$	$10.3, 10.0$
21dB	$18.2, 16.7$	$6.3, 6.8$	$5.9, 6.3$	$10.8, 10.1$
27dB	$24.2, 22.7$	$7.5, 7.6$	$6.9, 6.9$	$11.0, 10.2$

Table 5.1: Gain in TIR (dB) by NC-FADF under simulated speech-shaped noise.

original	preemphasized	baseline	FADF	NC-FADF
SNR	SNR(y_1, y_2)	v_1, v_2	v_1, v_2	v_1, v_2
-12dB	$0.2, 0.3$	$3.1, 3.9$	$3.1, 3.6$	$7.5, 8.5$
-6dB	$6.2, 6.3$	$4.2, 5.6$	$1.5, 5.4$	$9.6, 9.5$
0dB	$12.2, 12.3$	$6.3, 7.7$	$6.2, 6.9$	$10.5, 10.0$
6dB	$18.2, 18.3$	$7.7, 7.9$	$7.2, 7.3$	$10.9, 10.1$
12dB	$24.2, 24.3$	$8.1, 8.1$	$7.5, 7.4$	$11.1, 10.2$

Table 5.2: Gain in TIR (dB) by NC-FADF under real diffuse noise.

5.4.3 Separation performances

Separation performances were evaluated by system TIR gains in dB, ΔTIR, defined in (1.7). In Tables 5.1 and 5.2, the TIR gains of NC-FADF outperform those of the baseline for both types of noises, at the cost of a slightly decreased SNR, as shown in Tables 5.3 and 5.4. It is interesting to observe that under severe noise conditions, e.g., SNR=-12 dB (original), baseline ADF actually increased SNR. This is consistent with the analysis in [72] that in correlated noise, baseline ADF tends to divert from speech separation to noise cancellation. The TIR gain in Tables 5.1 and 5.2 and the SNR results in Tables 5.3 and 5.4 demonstrate that the NC algorithm can force ADF to focus on speech separation, rather than noise cancellation.

Analysis on the waveforms of ADF filter estimates obtained with and without NC processing shows that in very severe noise conditions, the baseline ADF adaptations is shown to have been distracted from the speech separation task to do some type of noise cancellation. The truncated ADF filter waveforms showing coefficients of the

original SNR	preemphasized SNR(y_1, y_2)	baseline v_{n_1}, v_{n_2}	FADF v_{n_1}, v_{n_2}	NC-FADF v_{n_1}, v_{n_2}
3dB	$0.2, -1.3$	$-0.3, -1.5$	$-0.1, -1.3$	$-1.1, -3.4$
9dB	$6.2, 4.7$	$5.3, 3.4$	$5.5, 3.7$	$4.6, 2.5$
15dB	$12.2, 10.7$	$10.8, 8.6$	$11.0, 8.9$	$10.5, 8.5$
21dB	$18.2, 16.7$	$16.3, 14.0$	$16.5, 14.3$	$16.4, 14.5$
27dB	$24.2, 22.7$	$22.0, 19.7$	$22.2, 19.9$	$22.3, 20.5$

Table 5.3: Output SNR (dB) for NC-FADF under simulated speech-shaped noise.

original SNR	preemphasized SNR(y_1, y_2)	baseline v_{n_1}, v_{n_2}	FADF v_{n_1}, v_{n_2}	NC-FADF v_{n_1}, v_{n_2}
-12dB	$0.2, 0.3$	$3.8, 2.4$	$4.2, 3.2$	$0.6, -1.7$
-6dB	$6.2, 6.3$	$6.1, 5.9$	$6.5, 4.9$	$6.3, 4.2$
0dB	$12.2, 12.3$	$12.3, 11.4$	$12.8, 11.6$	$12.3, 10.1$
6dB	$18.2, 18.3$	$17.9, 16.4$	$18.0, 16.5$	$18.2, 16.0$
12dB	$24.2, 24.3$	$23.4, 21.7$	$23.6, 21.9$	$24.1, 22.0$

Table 5.4: Output SNR (dB) for NC-FADF under real diffuse noise.

beginning 10ms, estimated under various SNR conditions, are illustrated in Figures 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, and 5.16. When noise level was very high, the ADF filters estimated with adaptations without NC terms exhibited more similarities to the filters obtained from noise cancellation adaptations than the separation filters. For example, the strong beginning peaks of the solid curves in the (c) and (d) sub-figures of Figures 5.7 and 5.8 are characteristics of cancellation filter. As a contrast, the second peaks that are characteristics of the separation filters are much weaker than their true values. This shows a strong distracting effect of noise on the estimation of ADF filters. As a comparison, the adaptive NC processing enhances separation performances by effectively preventing the distraction of noise and reducing such type of errors, as shown in Figures 5.7-(a,b) and 5.8-(a,b).

5.5 Special Case of Uncorrelated Noise

The above derivations and analyses are for general noise. When the degradation comes only from the special case of noises, such as uncorrelated sensor noise, compensation algorithm of simplified form could be obtained so that implementation complexities could be further reduced. However, in the following, we only discuss the simplified NC-ADF algorithm [67] without providing corresponding fast implementations.

5.5.1 Simplification of NC algorithm

When the input noises are sensor noise only and assumed to be white and uncorrelated with each other, the noise correlation matrix is

$$\mathbf{R}_{\tilde{\mathbf{n}}\tilde{\mathbf{n}}} = blkdiag\left(\sigma_{n1}^2 \mathbf{I}_{(2N-1)\times(2N-1)}, \sigma_{n2}^2 \mathbf{I}_{(2N-1)\times(2N-1)}\right), \tag{5.36}$$

where σ_{n1}^2 and σ_{n2}^2 are powers of input noises. With the I/O relations of correlation matrices (5.2), the output noise correlation matrix becomes

$$\mathbf{R}_{\mathbf{v}_\eta \mathbf{v}_\eta} = \mathbf{R}_{\mathbf{vv}} + \begin{bmatrix} \sigma_1^2 \mathbf{I} + \sigma_2^2 \mathbf{G}_{12}\mathbf{G}_{12}^T & -\sigma_1^2[\mathbf{I}\ \ \mathbf{0}]\mathbf{G}_{21} - \sigma_2^2 \mathbf{G}_{12}[\mathbf{I}\ \ \mathbf{0}]^T \\ -\sigma_2^2[\mathbf{I}\ \ \mathbf{0}]\mathbf{G}_{12} - \sigma_1^2 \mathbf{G}_{21}[\mathbf{I}\ \ \mathbf{0}]^T & \sigma_2^2 \mathbf{I} + \sigma_1^2 \mathbf{G}_{21}\mathbf{G}_{21}^T \end{bmatrix}, \tag{5.37}$$

where the auto-correlation block of $\mathbf{R}_{\mathbf{v}_\eta \mathbf{v}_\eta}$,

$$\mathbf{R}_{\mathbf{v}_{\eta i} \mathbf{v}_{\eta i}} = \mathbf{R}_{\mathbf{v}_i \mathbf{v}_i} + \sigma_i^2 \mathbf{I} + \sigma_j^2 \mathbf{G}_{ij}\mathbf{G}_{ij}^T. \tag{5.38}$$

Eq. (5.38) shows that the effects of input noises on ADF outputs could be classified into two types: those propagated by direct paths and those propagated by cross-channel paths. The direct path noise remains to be white with the same power as input, while the cross-channel noise is colorized by the de-coupling filters $\mathbf{g}_{ij}$'s. The noise effects on outputs depend on the state of ADF system. The input noise components have the least effects at the outputs in the initial and trivial cases of $\mathbf{g}_{ij} = \mathbf{0}$, where only direct-path noises are present; input noises have the strongest effects when $\mathbf{g}_{ij}$'s are close to the ideal ADF separation filters, under the speech source mixing condition that the direct paths are close to the cross-coupling paths [72] such that the magnitudes of $\mathbf{G}_{ij}(f)$'s are close to 1. In such extreme case, the output level of noise energy nearly doubles that of the input noise energy, which deteriorate the SNR conditions for the outputs.

The cross-correlation matrices between noisy output vectors are

$$\mathbf{R}_{\mathbf{v}_{\eta i}\mathbf{v}_{\eta j}} = \mathbf{R}_{\mathbf{v}_i\mathbf{v}_j} - \sigma_i^2[\mathbf{I}\ \ \mathbf{0}]\mathbf{G}_{ji} - \sigma_j^2\mathbf{G}_{ij}[\mathbf{I}\ \ \mathbf{0}]^T, \tag{5.39}$$

which describes the contribution of noise on output cross correlations degrades ADF filter adaptation. The same relation in vector form can be written as

$$\mathbf{r}'_{\mathbf{v}_i\mathbf{v}_j} = \mathbf{r}_{\mathbf{v}_{\eta i}\mathbf{v}_{\eta j}} + \sigma_i^2 g_{ji}(0)\mathbf{e}_1 + \sigma_j^2\mathbf{g}_{ij}, \tag{5.40}$$

where $\mathbf{e}_1 = [1, 0, \cdots, 0]^T$. Similar to the derivation of general NC-ADF algorithms, the simplifed NC-ADF algorithm can be derived with the noise-compensated decorrelation criterion function

$$J'_{ij} = \frac{1}{2}(\mathbf{r}'_{\mathbf{v}_i\mathbf{v}_j})^T\mathbf{r}'_{\mathbf{v}_i\mathbf{v}_j}. \tag{5.41}$$

The gradient vectors of J'_{ij} can be represented by

$$\nabla_{\mathbf{g}_{ij}} J'_{ij} = -\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j} \cdot \left(\mathbf{r}_{\mathbf{v}_{\eta i}\mathbf{v}_{\eta j}} + \sigma_i^2 g_{ji}(0)\mathbf{e}_1 + \sigma_j^2\mathbf{g}_{ij}\right), \tag{5.42}$$

where $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_{n_j}\mathbf{v}_{n_j}} - \sigma_j^2\mathbf{I}$. By the same approximation technique used for the derivations of baseline ADF algorithms, we could also omit the multiplying correlation matrix in (5.42) under the assumption that $\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j}$ being positive-definite (in the sense that its quadratic forms being non-negative over all real non-negative vectors). Another simplification could be made from the observation that $g_{ij}(0) \approx 0$. This is because that the lengths of the direct acoustic paths and that of the cross-channel acoustic paths are usually different. The cross-correlation vectors $\mathbf{r}_{v_{n_i}\mathbf{v}_{n_j}}$'s can be estimated by the corresponding instantaneous correlation vectors $v_{n_i}(t)\mathbf{v}_{n_j}(t)$'s. Therefore, the simplified implementation is obtained as follows.

$$\mathbf{g}_{ij}^{(t+1)} = \mathbf{g}_{ij}^{(t)} + \mu(t)\left(v_{n_i}(t)\mathbf{v}_{n_j}(t) + \sigma_j^2\mathbf{g}_{ij}^{(t)}\right), \tag{5.43}$$

where the normalizing gain $\mu(t)$ takes the same basic form as (1.6).

$$\mu(t) = \frac{2\gamma}{N\left(\sigma_{y_{n_1}}^2 + \sigma_{y_{n_2}}^2\right)}. \tag{5.44}$$

5.5.2 Speech separation simulations

Speech mixing procedures are the same as previous sections. Speech corruption was simulated with the uncorrelated white noise signals under various levels of SNRs. The ADF adaptation used $N = 400$ and $\gamma = 0.01$. The noise powers σ_1 and σ_2 are assumed to be known, where in practice they can be measured during speech in active periods.

The convergence performances of simplified NC algorithm are shown in the ADF error curve in Figure 5.17. Associated with error reduction in filter estimation, the NC-ADF algorithm provided improvements to TIR gains over baseline methods, as shown in Table 5.5.

Both FADF and NC-FADF modules alone achieved real-time with Matlab implementations. FADF had a performance similar to that of baseline ADF, with only a slight degradation.

SNR dB	Baseline ADF (v_1, v_2)	Simplified NC-ADF (v_1, v_2)
0	(3.72, 3.36)	(7.02, 6.89)
5	(5.01, 4.58)	(7.52, 7.35)
10	(6.13, 7.78)	(7.76, 7.56)
15	(7.00, 6.73)	(7.90, 7.71)
20	(7.56, 7.34)	(7.97, 7.80)
25	(7.84, 7.66)	(8.00, 7.84)
30	(7.96, 7.79)	(8.02, 7.86)

Table 5.5: Target-to-interference (TIR) gains in dB for simplified NC-ADF.

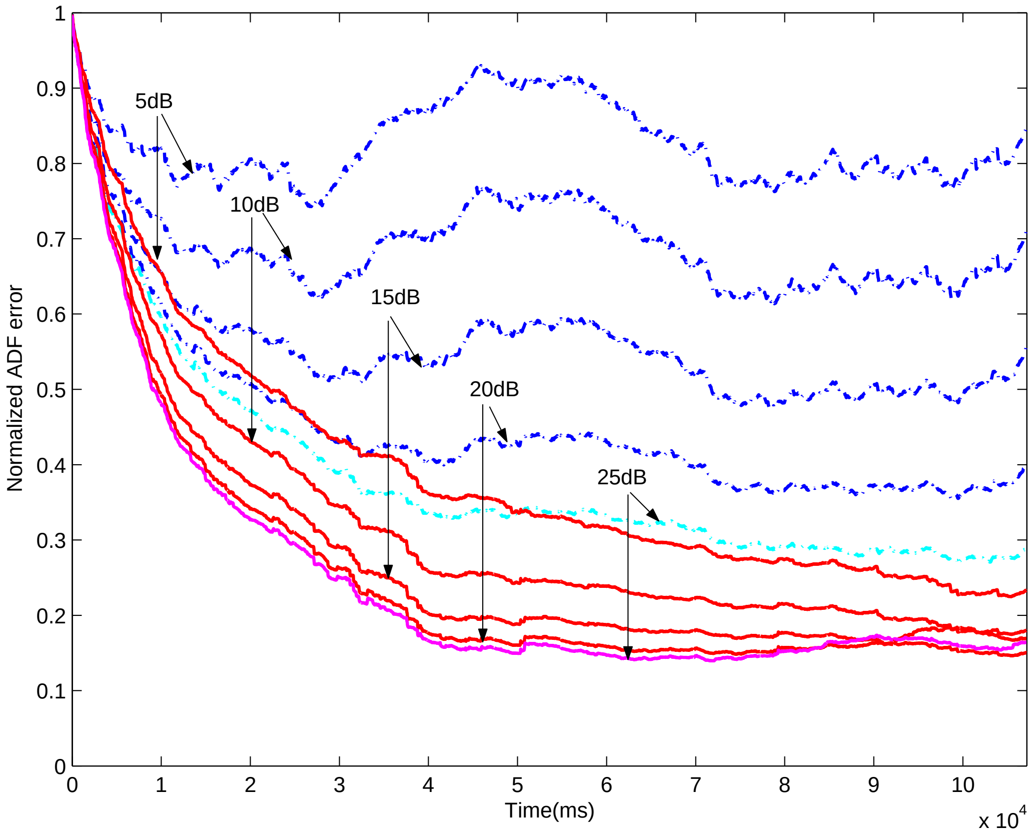


Figure 5.5: Convergence of ADF estimation error in simulated noise (dash-dot: FADF; solid: NC-FADF).

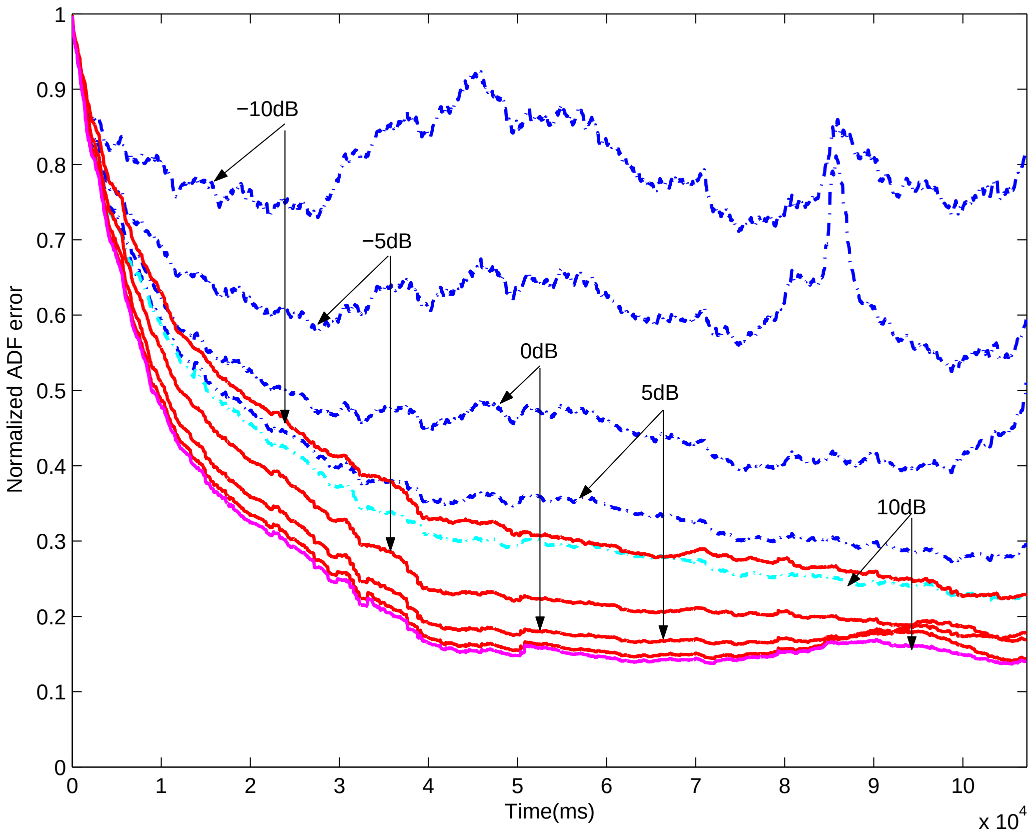


Figure 5.6: Convergence of ADF estimation error in real diffuse noise (dash-dot: FADF; solid: NC-FADF).

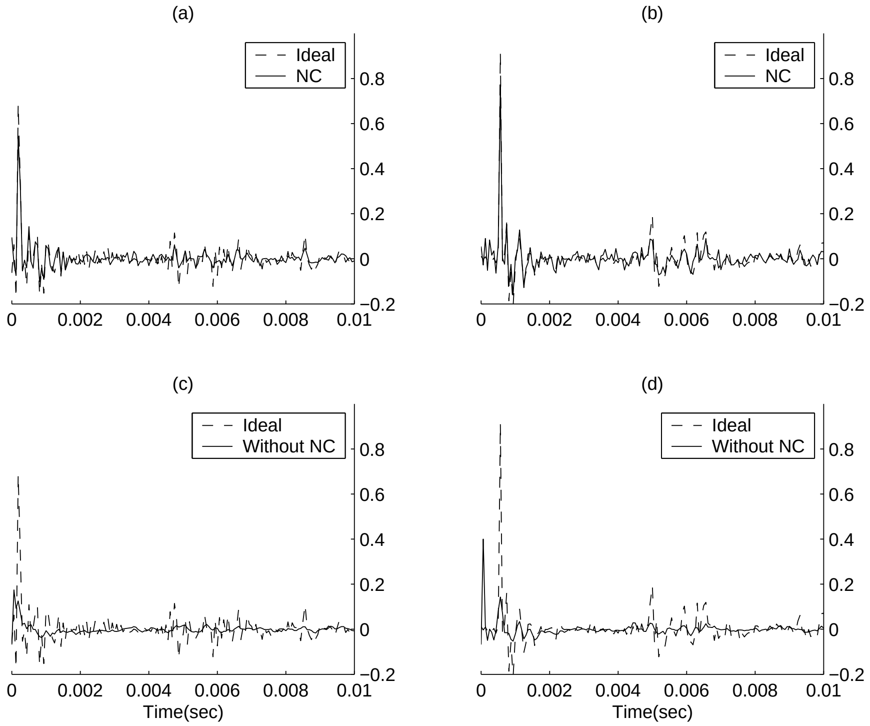


Figure 5.7: Filter estimates under simulated noise with SNR=5dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

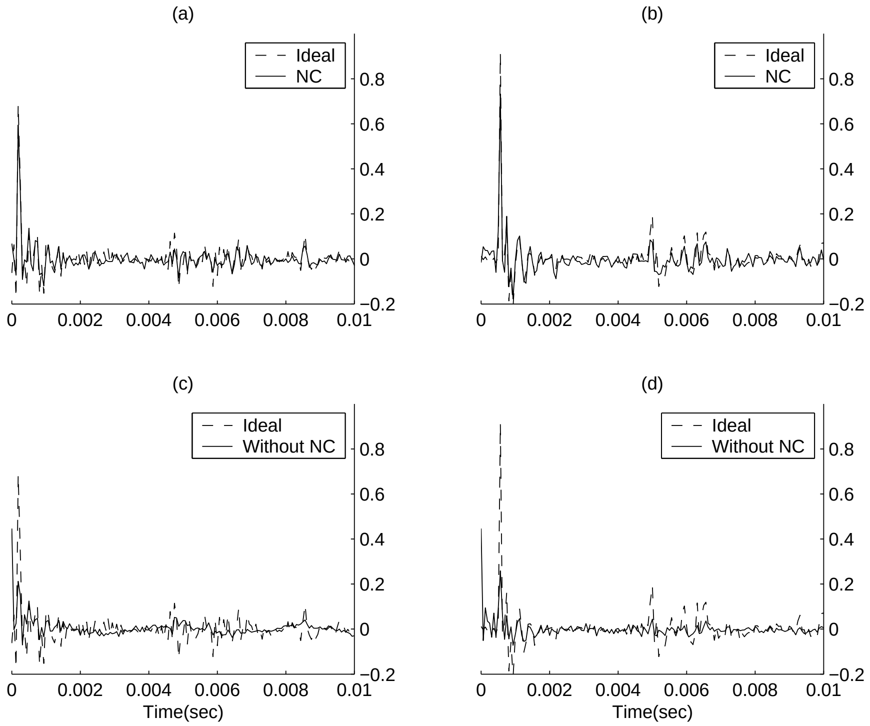


Figure 5.8: Filter estimates under real diffuse noise with SNR=−10dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

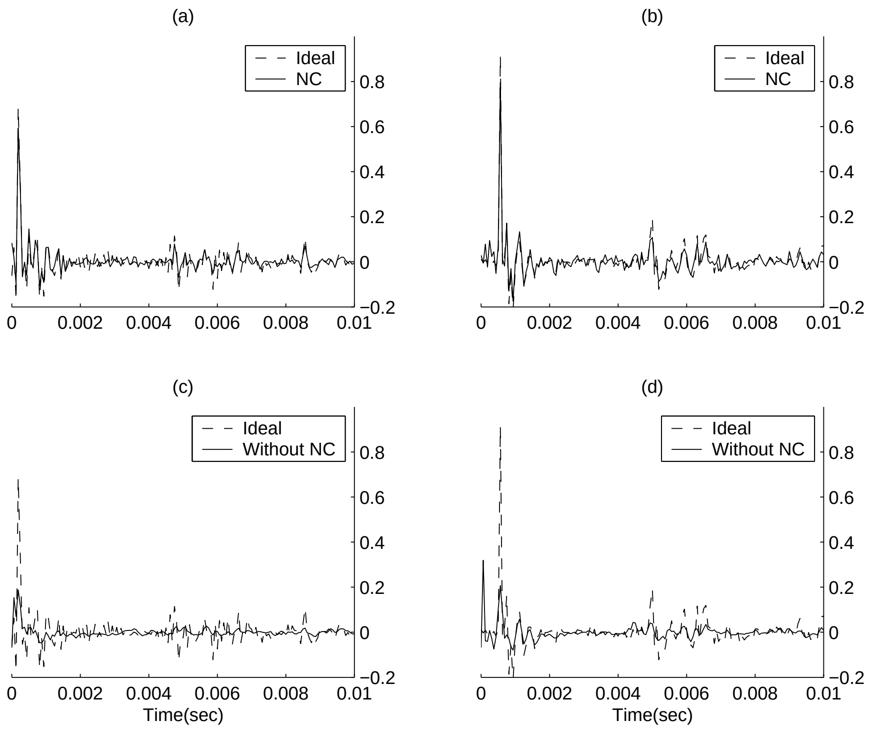


Figure 5.9: Filter estimates under simulated noise with SNR=10dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

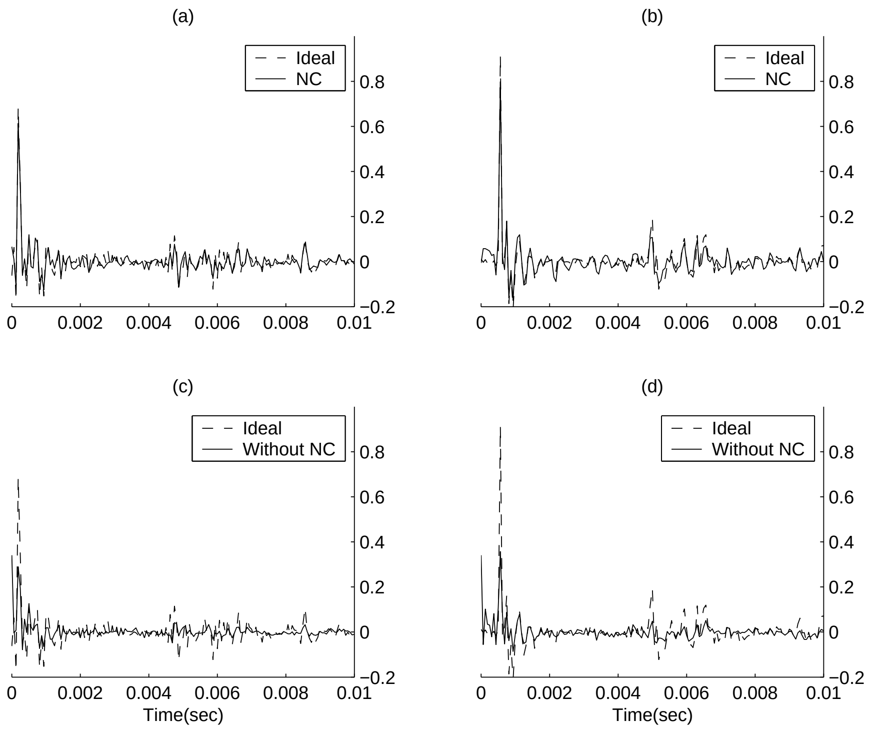


Figure 5.10: Filter estimates under real diffuse noise with SNR=−5dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

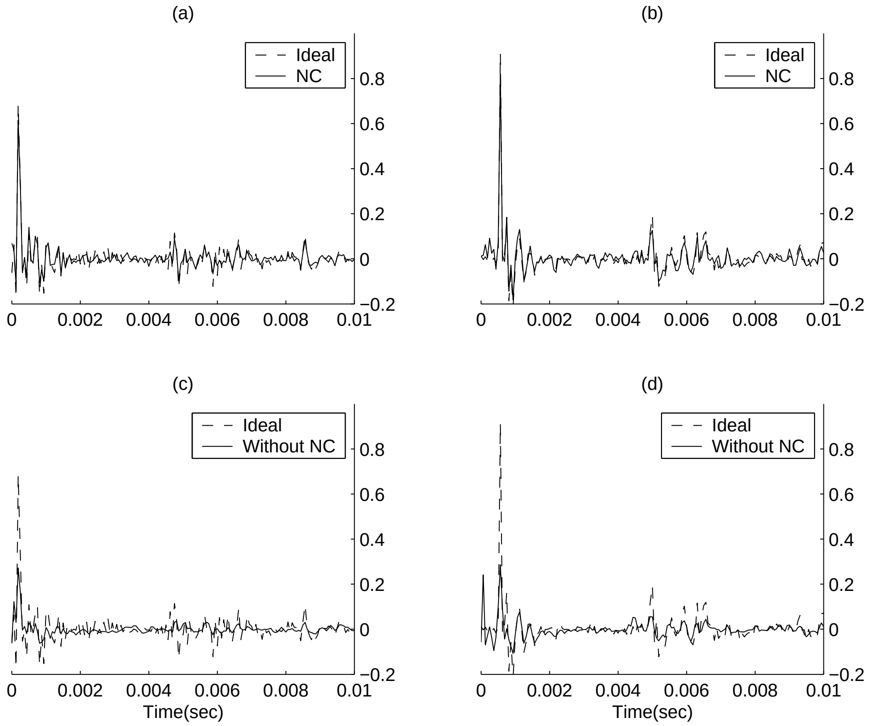


Figure 5.11: Filter estimates under simulated noise with SNR=15dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

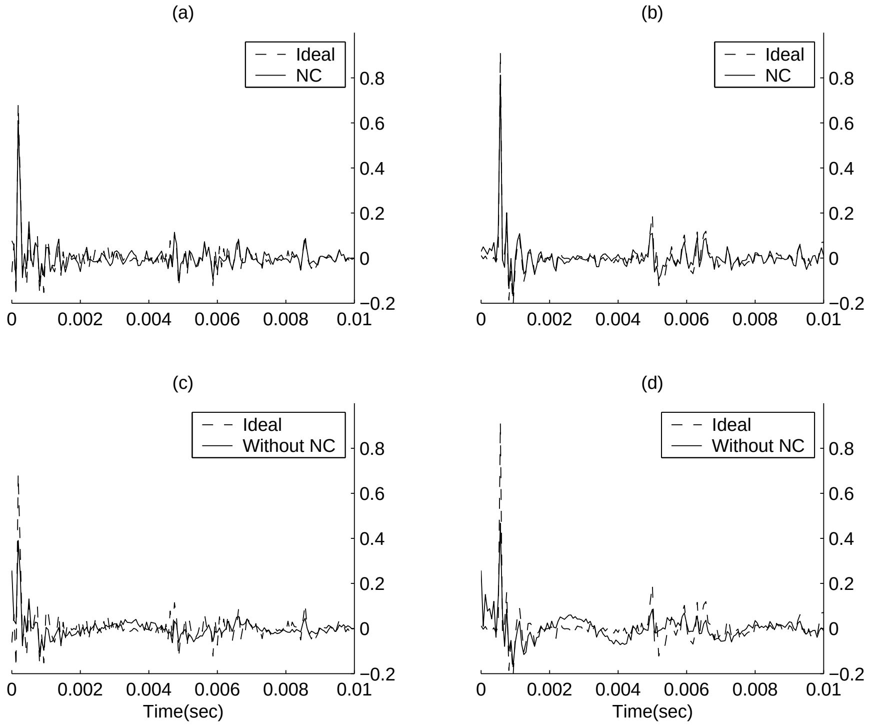


Figure 5.12: Filter estimates under real diffuse noise with SNR=0dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

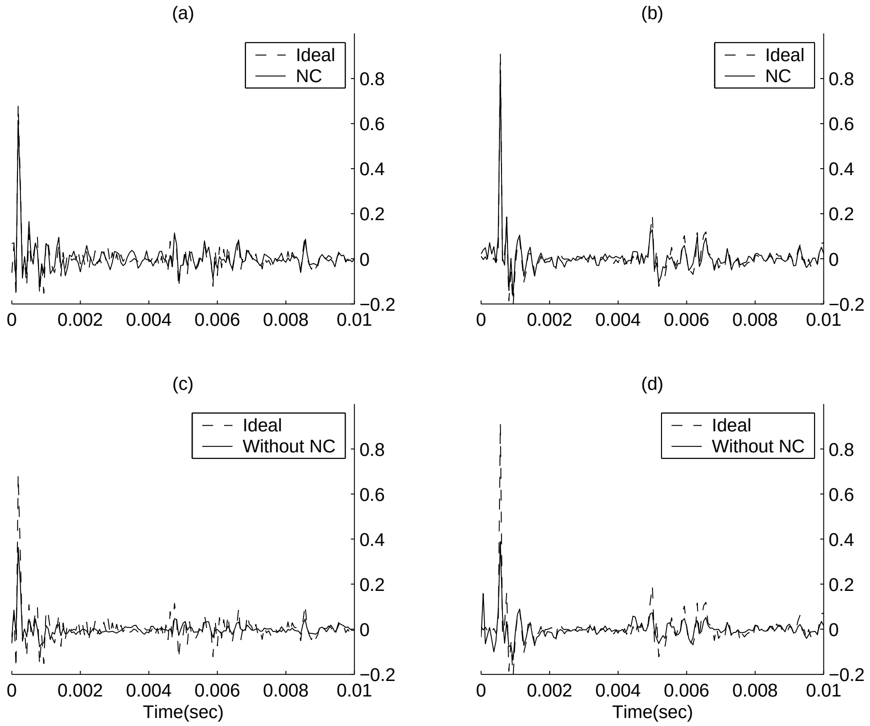


Figure 5.13: Filter estimates under simulated noise with SNR=20dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

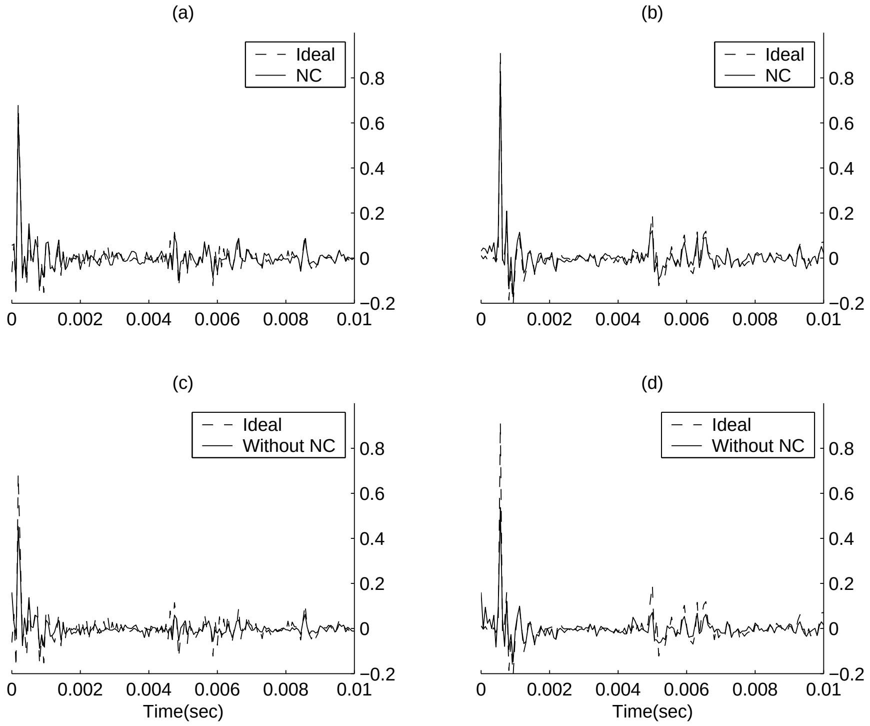


Figure 5.14: Filter estimates under real diffuse noise with SNR=5dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

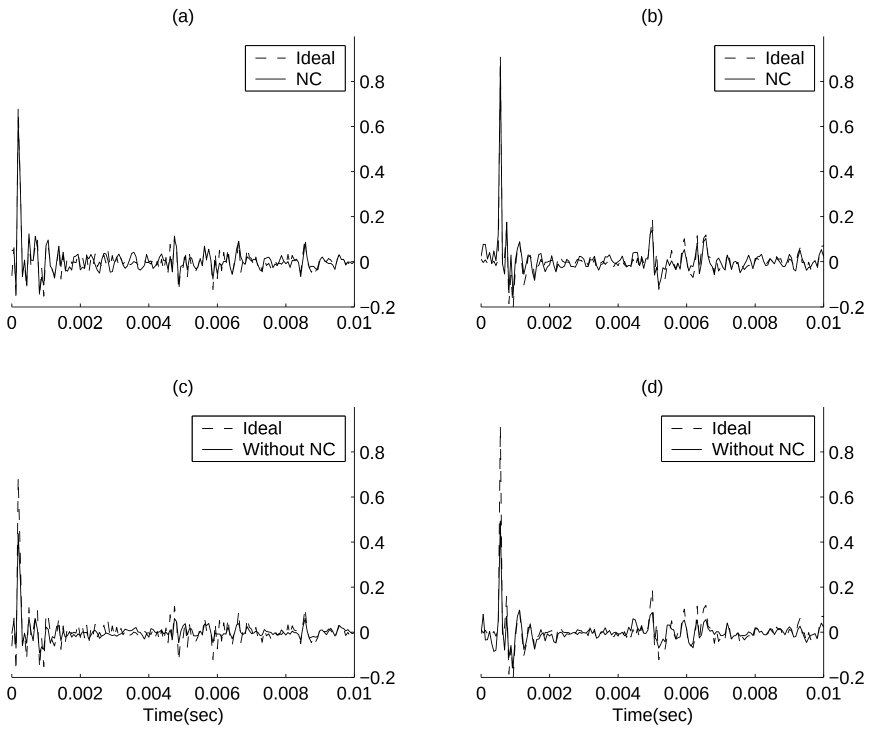


Figure 5.15: Filter estimates under simulated noise with SNR=25dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

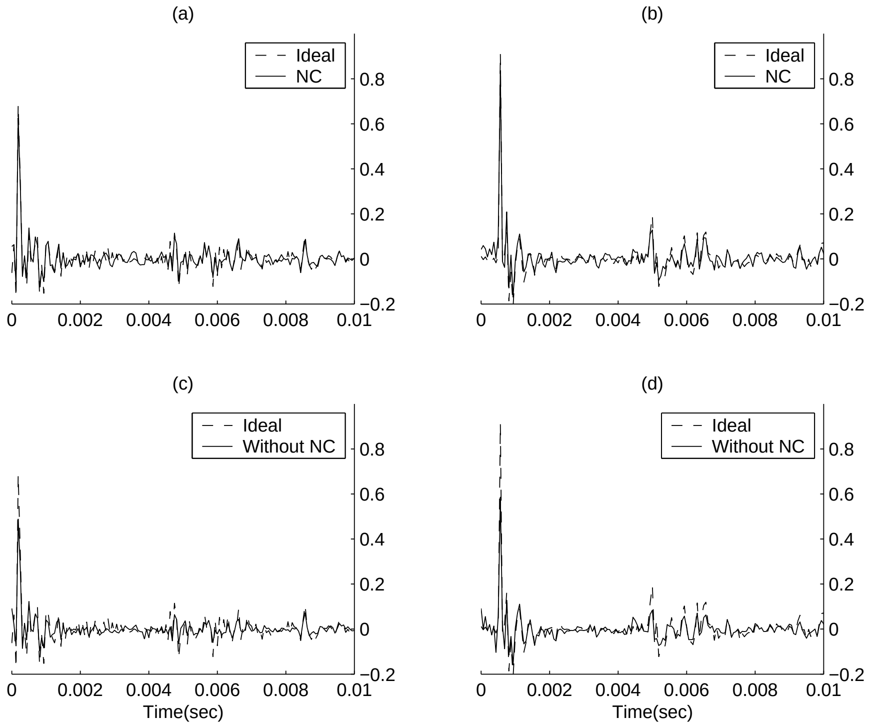


Figure 5.16: Filter estimates under real diffuse noise with SNR=10dB. (a) and (b) are estimates of $\mathbf{g}_{12}$ and $\mathbf{g}_{21}$ with NC, respectively; (c) and (d) are without NC.

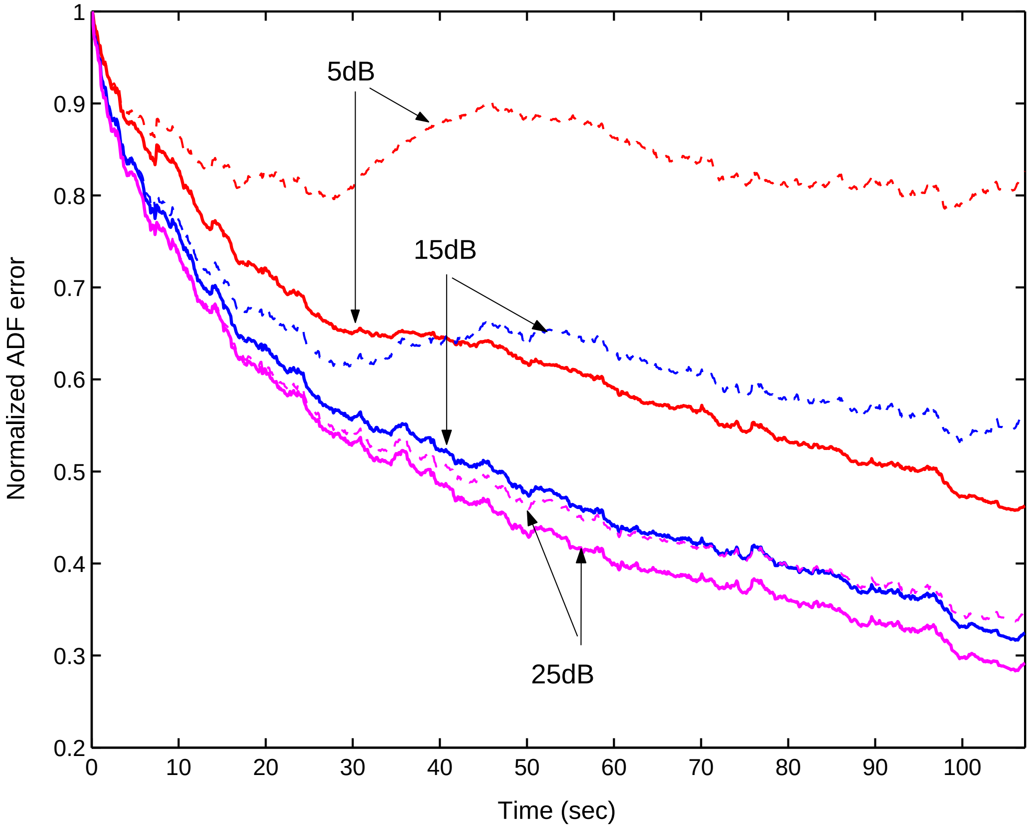


Figure 5.17: Convergence of ADF estimation errors of simplified NC-ADF implementation under uncorrelated noise (dash: without NC; solid: with simplified NC).

Chapter 6

ADAPTIVE SPEECH ENHANCEMENT FOR ADF MODEL

In this chapter, we discuss the adaptive speech enhancement techniques for reducing the output noise of ADF model. With the fast algorithm of tracking ADF output noise statistics presented in Chapter 5, noise reduction methods are integrated with NCADF to enhance the separated speech.

6.1 Enhancement as Post-Processing

Although NC-FADF improves the speech separation performance in noise, the separation outputs $\mathbf{v}_{\mathbf{n}_i}$ are still contaminated by noise. For the improvement of speech qualities in ADF processing, speech enhancement or noise reduction techniques are possible solutions. In this respect, there are seemingly two different options. One is to reduce noise as a pre-processor prior to ADF separation system, as shown in Figure 6.1. The other configuration is doing speech enhancement as post-processing, as illustrated in Figure 6.2. An adaptive noise canceler (ANC) was applied in [73] to

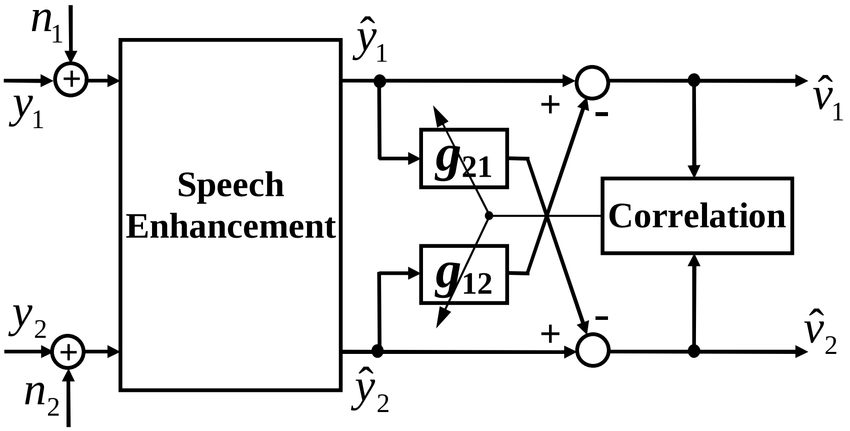


Figure 6.1: Speech enhancement prior to ADF processing.

reduce noise levels prior to the source separation of foetal electrocardiogram (EGC) signal, with the help of additional noise sensing reference input. In [72], a noise reducing pre-filtering was tested for ADF model under noise with limited success in speech application. However, the pre-filtering in the first choice deteriorates the working conditions of the subsequent speech source separation, due to the distortions introduced by speech enhancement. Therefore, it will be more favorable to enhance the speech after separation processing.

6.2 Tracking of Output Noise Auto-Correlations

Speech enhancement post-processing should be integrated with NC-ADF to reduce noise in each output. In the following, we assume that the enhancement processing are also implemented in blocks to facilitate fast algorithm. Usually, speech enhancement algorithms require statistics of the noise to be removed. In the case of ADF, we need to track the time-varying output noise statistics as filters evolve from block to block, which can be accomplished by a fast computation of (5.6).

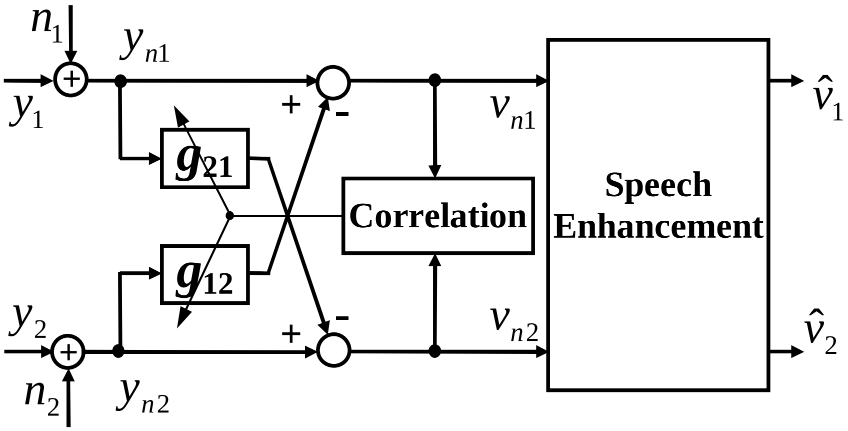


Figure 6.2: Speech enhancement after ADF processing.

Similar to the derivations of (5.16) and Appendix C, we obtain auto-correlation of ADF output noise for the m-th block

$$\hat{\mathbf{r}}^m_{\eta_i\boldsymbol{\eta}_i} = \hat{\mathbf{r}}_{n_i\mathbf{n}_i} - \mathbf{a}^m_{ii} - \mathbf{b}^m_{ii} + \mathbf{c}^m_{ii}, \tag{6.1}$$

where

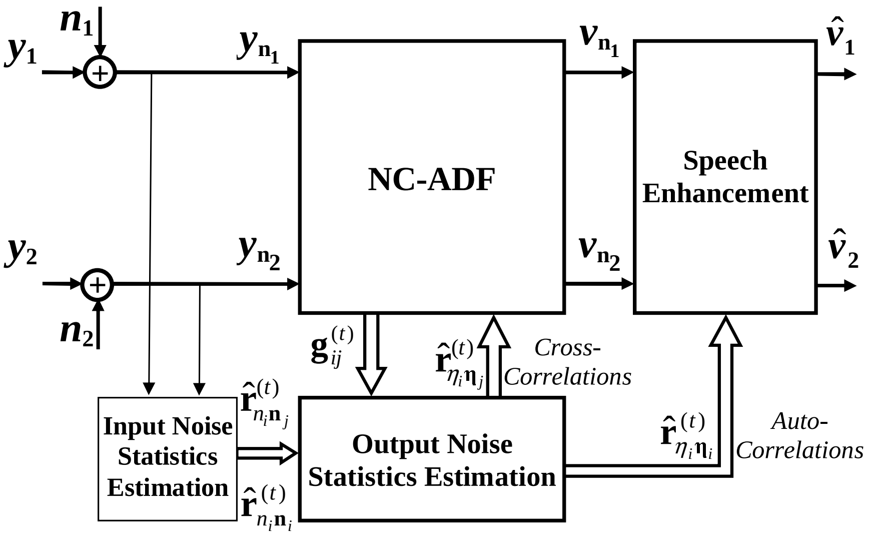
$$\mathbf{a}^m_{ii} = \mathbf{G}^m_{ij}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_j}, \tag{6.2}$$

$$\mathbf{b}^m_{ii} = \hat{\mathbf{R}}_{\mathbf{n}_i\mathbf{n}_j}\mathbf{g}^m_{ij}, \tag{6.3}$$

$$\mathbf{c}^m_{ii} = \mathbf{G}^m_{ij}\mathbf{d}^m_{ii}, \tag{6.4}$$

$$\mathbf{d}^m_{ii} = \hat{\mathbf{R}}_{\tilde{\mathbf{n}}_j\mathbf{n}_j}\mathbf{g}^m_{ij}. \tag{6.5}$$

Since input noise is stationary, its auto and cross correlations can be measured *a priori* during speech inactive period. The fast mappings from input noise correlations to output noise auto-correlation, depending only on current system parameters $\mathbf{g}^m_{ij}$'s and $\mathbf{G}^m_{ji}$'s, are implemented as fast convolutions of the following signal sequences:



y_1
n_1
y_{n_1}
NC-ADF
v_{n_1}
Speech Enhancement
$\hat{v}_1$
y_2
n_2
y_{n_2}
v_{n_2}
$\hat{v}_2$
$g_{ij}^{(t)}$
$\hat{r}_{\eta_i \eta_j}^{(t)}$
Cross-Correlations
$\hat{r}_{n_i n_j}^{(t)}$
Input Noise Statistics Estimation
$\hat{r}_{n_i n_i}^{(t)}$
Output Noise Statistics Estimation
$\hat{r}_{\eta_i \eta_i}^{(t)}$
Auto-Correlations

Figure 6.3: Noise-compensated ADF and adaptive speech enhancement system.

$$a_{ii}^m(k) = g_{ij}^m(n) * \xi_{ii}^a(n)|_{n=2N-2-k}, \tag{6.6}$$

$$\xi_{ii}^a(n) = \hat{r}_{n_i \tilde{n}_j}(2N-2-n), \tag{6.7}$$

$$c_{ii}^m(k) = g_{ij}^m(n) * \xi_{ii}^c(n)|_{n=2N-2-k}, \tag{6.8}$$

$$\xi_{ii}^c(n) = d_{ij}^m(2N-2-n), \tag{6.9}$$

$$b_{ii}^m(k) = g_{ij}^m(n) * \xi_{ii}^b(n)|_{n=k+N-1}, \tag{6.10}$$

$$\xi_{ii}^b(n) = \hat{r}_{n_i \tilde{n}_j}(N-1-n), \tag{6.11}$$

$$d_{ii}^m(k) = g_{ij}^m(n) * \xi_{ii}^d(n)|_{n=k+N-1}, \tag{6.12}$$

$$\xi_{ii}^d(n) = \hat{r}_{n_j \tilde{n}_j}(N-1-n). \tag{6.13}$$

6.3 Adaptive Enhancement of Separated Speech

Utilizing the adaptively estimated noise statistics $\hat{\mathbf{r}}_{\eta_i \boldsymbol{\eta}_i}^m$, many speech enhancement algorithms can be considered for post-enhancement of ADF outputs. Two single channel speech enhancement methods, spectral subtraction and generalized subspace methods, are compared in the current work for reducing ADF output noises. A basic spectral subtraction approach is included for two reasons: 1) it is simple to implement and suitable for fast algorithm 2) to some extent, it provides a performance lower-bound for enhancement algorithms with higher complexities.

6.3.1 Spectral subtraction

The spectral subtraction (SS) algorithm [74] is taken in the basic form. For block m, the estimate of clean speech amplitude is given by

$$\left|\hat{V}_i^m(f)\right| = \begin{cases} \left(\left|V_{n_i}^m(f)\right|^2 - E\left\{\left|\Phi_i^m(f)\right|^2\right\}\right)^{\frac{1}{2}}, if \frac{E\left\{\left|\Phi_i^m(f)\right|\right\}^2}{\left|V_{n_i}^m(f)\right|^2} \leq 1 \\ 0, \qquad otherwise, \end{cases} \tag{6.14}$$

and the phase of $\hat{V}_i{}^m(f)$ is set to be equal to that of $V_{n_i}^m(f)$. The noise power spectral density required in (6.14) at each block m is directly transformed from the short-term correlation vectors $\boldsymbol{\phi}_{\boldsymbol{\eta}_i}^m$ as

$$E\left\{\left|\Phi_i^m(f)\right|^2\right\} = \boldsymbol{FFT}\left(E\left\{\boldsymbol{\phi}_{\boldsymbol{\eta}_i}^m\right\}\right), \tag{6.15}$$

where, for a signal vector of length N in the m-th block,

$$\phi_{\boldsymbol{\eta}_i}^m(n) = \sum_{t=t_m}^{t_m+N-1} \eta_i(t)\eta_i(t-n),$$

which relates to the average vector $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_i}^m$ by

$$E\left\{\boldsymbol{\phi}_{\boldsymbol{\eta}_i}^m\right\} = N\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_i}^m. \tag{6.16}$$

No further processing is made to suppress musical noise, because the we are focused on machine recognition of speech rather than human speech reception.

6.3.2 Subspace noise reduction

For subspace based speech enhancement, we choose the time domain constrained (TDC) type of generalized subspace (GSub) method by Hu and Loizou [75], because of its ability to handle colored noise. TDC-GSub processing is applied to every block of ADF outputs. This method requires the noise auto-correlation matrix $\mathbf{R}^m_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i}$, which can be constructed by forming a symmetric Teoplitz matrix from the output auto-correlation vector in (6.1). Specifically, $\hat{\mathbf{r}}^m_{\eta_i\boldsymbol{\eta}_i}$ constitutes the first column and the first row of $\mathbf{R}^m_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i}$. Another information the TDC-GSub algorithm takes is the auto-correlation matrix of noisy ADF output, $\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}}$, which is estimated from ADF outputs of the current block. The TDC-GSub processing are performed on each non-overlapping sub-frame of length $L = 40$ and the major steps are the same as in [75]:

Step 1. Do eigen-decomposition $\boldsymbol{\Sigma}_i\mathbf{U}{=}\mathbf{U}\boldsymbol{\Lambda}$ for matrix $\boldsymbol{\Sigma}_i{=}(\mathbf{R}^m_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i})^{-1}\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}} - \mathbf{I}$, with $\boldsymbol{\Lambda} = diag\left[\lambda^1, ...\lambda^M, 0, ...0\right]$, and M is the number of positive eigen-values.

Step 2. Compute the optimal speech estimator $\mathbf{H}{=}\mathbf{U}^{-T}diag\left[\alpha_1, ..., \alpha_M, 0, ..., 0\right]\mathbf{U}^T$, where the eigen-domain filtering gains are obtained by $\alpha_k = \lambda^k/(\lambda^k+\beta)$,$k = 1, ..., M$, and β determined from

$$\beta = \begin{cases} 5 & SNR_{dB} \le -5, \\ 1 & SNR_{dB} \ge 20, \\ 4.2 - (SNR_{dB})/6.25 & otherwise, \end{cases} \tag{6.17}$$

with $SNR_{dB} = 10\log_{10}\left(\sum_{k=1}^{M}\lambda^k/L\right)$.

Step 3. Enhance the ith ADF output by $\hat{\mathbf{v}}^m_i = \mathbf{H}\mathbf{v}^m_{n_i}$.

The computations of matrix inversion, multiplication, and eigen-decomposition are in the order of the cube of the matrix dimension L and are usually time consuming. They become acceptable only when the small value of L (corresponding to $2.5ms$) is used. In addition, a measure is taken to speed up TDC-GSub by utilizing the

Computation	Complexity estimates		Gain
	Direct	Fast	
ADF filtering	$2N$	$(8N_F \log_2 N_F)/K$	$N/(8\log_2(2N))$
ADF adapt	$2N$	$(8N_F \log_2 N_F)/K$	$N/(8\log_2(2N))$
$\hat{\mathbf{r}}_{\eta_i \boldsymbol{\eta}_j}$'s	$10N^2$	$(40N_F \log_2 N_F)/K$	$N^2/(8\log_2(2N))$
$\hat{\mathbf{r}}_{\eta_i \boldsymbol{\eta}_i}$'s	$10N^2$	$(40N_F \log_2 N_F)/K$	$N^2/(8\log_2(2N))$
SS	$8K_F \log_2 K_F/K$, $(K_F \geq K)$		
TDC-GSub	$O(L^2)$		

Table 6.1: Counts of real multiplications for NC-ADF and adaptive speech enhancement algorithms.

short-term stationary property of separated speech signals $\mathbf{v}_{n_i}$'s. Within $20ms$, the variations of $\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}}$'s are relatively small, obviating the need for updating their eigen-decompositions in every sub-frames. In practice, the computation rate for both steps 1 and 2 are thus reduced to every $12.5ms$, without introducing significant degradations.

6.4 Complexity Analysis

The complexity of major computation steps in terms of average number of real multiplications per time-sample are listed in Table 6.1. Steps causing trivial computation overheads, e.g., VSS and its compensation, are ignored. The gain of fast over direct implementations are evaluated for N=K and N_F=$2N$. The counts for FFT are based on regular radix-2 method. It is possible to further reduce complexities. For example, more efficient FFTs can be used, and it is obvious that $\mathbf{a}$'s and $\mathbf{c}$'s in (5.16) and (6.1) can further share some FFT/IFFT computations, which is not considered here. In Table 6.1, only a coarse complexity estimate is made for TDC-GSub, based on direct implementations of matrix operations. Faster techniques for TDC-GSub algorithms and complexity analysis are out of the scope of this dissertation.

Chapter 7

SPEECH RECOGNITION EXPERIMENTS AND EVALUATIONS

The data and setup for speech mixing and separation experiments using the proposed methods have been discussed in previous chapters. The separation performances were evaluated with gains of TIRs in those experiments. However, to be applied for real speech recognition system, the enhancement abilities of those methods should eventually be evaluated by speech recognition performances. In this chapter, speech recognition experiments are presented for the evaluation of those enhancing algorithms discussed in previous chapters. Automatic speech recognition (ASR) experiments were carried out for the following methods: pre-whitening processing, block-iterative ADF implementation, multi-ADF integration and post-filtering, VSS-ADF, FADF, NCFADF, and adaptive speech enhancement. ASR training and testing configurations are presented, enhancement techniques for ADF speech separation model are compared with reference models or baseline algorithms, and experimental results evaluated based on corresponding results.

7.1 ASR Experiments

7.1.1 ASR setup and training

The ASR system was based on hidden Markov modeling (HMM) [1] of phone units with state observation probability density functions of Gaussian mixture densities. Phone recognition were performed for separated target speech signals obtained from various ADF separation algorithms. Speech signal was represented by a sequence of 39-dimensional feature vectors. The feature vectors were obtained from short-time analysis windows of $20ms$ with 50% overlapping ($10ms$). Speech feature components included 13 cepstral coefficients and their first and second-order time derivatives. The feature vector sequence extracted from both training and test data were processed with spectral mean subtraction. There were 39 phone units, defined by the phone grouping scheme of [2]. Two types of acoustic models were employed: monophones and crossword triphones. The former models a phone unit independent of its neighboring phone context, while the latter takes into account both left and right neighboring phones that reflect the coarticulation effect of speech production. Phone bigram was used as "language model."

Acoustic models were trained from the entire training set of TIMIT database, consisting of 4620 sentences and their phone transcriptions. The HMM topology had three left-to-right emitting states, where the state transitions were nonskipping for speech units and the middle state was allowed to be skipped for silence. For monophone models, each emitting state of HMM was modeled by a size-8 Gaussian mixture density. For triphone HMMs, 9667 crossword triphones were initially obtained based on the 39-phone set, and to improve reliability of the triphone models, phonetic decision tree (PDT) based state tying was performed to reduce the triphone states to 1607 tied states [76]. The tied triphone states were each modeled by a size-8 Gaussian

Iteration	$N = 200$	$N = 400$	$N = 600$	$N = 800$	$N = 1000$
1	38.7	37.2	37.0	36.5	36.9
2	43.3	42.2	40.0	40.0	38.8
3	43.5	43.6	42.5	×	×
4	43.8	44.8	42.5	×	×
5	44.4	45.3	42.6	×	×
6	44.3	45.1	43.5	×	×

Table 7.1: Phone recognition accuracies (%) with monophone acoustic models versus filter length for multi-iterations of baseline ADF, with "×" denoting divergence.

mixture density. The resulting model set, including 6153 crossword triphone models shared by 57799 logical triphones , was then used in speech decoding.

Given the knowledge sources of acoustic and language models, namely the phone or triphone HMMs and phone bigrams, the Viterbi time-synchronous beam search based decoding engine generates a sequence of output phone label strings for the input sequence of feature vectors. The processing of feature extraction, the training of HMM models and the decoding of speech were implemented with HTK toolkit [76]. Recognition performance was measured by the standard string alignment algorithm also provided in HTK [76].

7.1.2 Effects of separation filter length

Theoretically, the length of the separation filters should be long enough to cover the length of the impulse responses of the acoustic paths. The impulse responses of the measured acoustic paths were on the order of 2000 samples, indicating that long FIR filters would be required. However, in reality, it is not feasible for ADF algorithms to estimate such long filters under strong room reverberation. Proper lengths for the FIR separation filters were evaluated by phone recognition accuracy with monophone models for baseline ADF. The results are summarized in Table 7.1, where the adaptation gain in (1.6) was set as $\gamma = 0.005$ and the parameter β was set as 0.8. It is observed that filter length of 200 to 400 taps yielded good results. With

Target source $s_1(t)$	Target remote $s_1(t) * h_{11}(t)$	mixture $y_1(t)$	ideal separation $v_1^o(t)$
68.9	59.7	29.1	52.0

Table 7.2: Phone recognition accuracies (%) of reference cases for clean target source/mixture, tested with monophone models.

longer filters, divergence occurred within a few iterations of ADF estimation (shown as “×”). The separation filter length was fixed as $N = 400$ in the subsequent phone recognition experiments.

7.1.3 Recognition accuracies of reference cases

Phone recognition experiments were first performed for several reference cases so that a) better references of performance gains could be established; b) upper and lower bounds could be set to ADF separation processing. The phone accuracy of the clean TIMIT target speech $s_1(t)$ gives the recognition upper bound for all other cases. The clean mixture $y_1(t)$ captured by target microphone 15 provides an accuracy lower bound for any ADF separation algorithms. The ideally separated speech signal using true ADF separation filters (1.3) gives an upper bound for all ADF algorithms. The accuracy differences between the clean target and the remotely captured target containing no interference speech ($s_1(t) * h_{11}(t)$) indicates a 9.2% absolute percentage degradation caused by room reverberation.

Table 7.2 lists those basic facts for recognition experiments measured by phone recognition with monophone models. As a contrast, recognition tests using triphone models were also conducted for the clean target $s_1(t)$ and the speech mixture $y_1(t)$, and their results are 71.9% and 26.8%, respectively. It is observed that the triphone model set led to improved phone accuracy for clean speech and deteriorated phone accuracy for convolutive mixture, compared with monophone models (68.9% and 29.1% repectively). This is due to the fact that the acoustic models were trained from clean speech, and the residue interference speech and processing distortions in ADF separated speech make recognition testing condition mismatch with the training condition. Under a matched training-test condition, where test speech was clean, the contextual details of triphone models enhanced discrimination of phonetic sounds. Under an unmatched training-test condition, where test speech was corrupted by interference speech, coarse monophone models could tolerate the interference components better. Therefore, monophone models are more robust for interferences and noises, and the recognition experiments in noisy scenarios, e.g., NC-ADF, will only be based on monophone models.

7.2 Speech Recognition Evaluation Results

Phone recognition experiments for evaluation of the proposed enhancement techniques were carried out for the following cases:

1. baseline batch ADF;
2. batch ADF with preemphasis;
3. batch ADF with inverse-PSD prewhitening;
4. block-iterative ADF with inverse-PSD;
5. multi-ADF post-filtering and integration by using multiple pairs of microphones for 1) throught 4);
6. comparative convolutive BSS algorithm of Douglas and Sun [12], denoted as DS-BSS;
7. comparative delay-and-sum beam forming to post-process multiple pairs of ADF outputs;

8. VSS-ADF algorithm;

9. NC-ADF, FADF, and NC-FADF under diffuse noise;

10. integration of NC-FADF with adaptive speech enhancement;

The results are grouped in several topics for comparison and presented below.

7.2.1 Prewhitening and block iteration

Prewhitening processing significantly improves the convergence performance and increases the speech recognition accuracies for subsequent ADF processing methods of various types. Examples are shown in Figure 7.1 for the single pair ADF processing on the mixtures acquired by microphone pair 15-3 and the 3-pair ADF post-filtering integration. The whitening pre-processing makes improvement for both single and multiple ADF applications, with the inverse-PSD slightly better than preemphasis.

Table 7.3 compares the phone recognition accuracies in the first iteration pass of ADF algorithm with various number of ADF pairs. It indicates the advantages of doing prewhitening for ADF. Consistent improvements were observed for both preemphasis and inverse-PSD processing in both single pair and multiple ADFs. Table 7.3 also shows the advantage of block-iterative ADF implementation. By adjusting the number of iterations locally within a relatively long time block, it utilizes the information contained in input data more efficiently than the sample-wise processing of batch ADFs, and it is not necessary to repeat the block-iterative ADF with multiple passes of iterations. However, it is necessary for batch processing of ADFs to achiever their upper limits in separation performances. The phone accuracy results of batch ADF processing after 10 iterations are listed in Table 7.4.

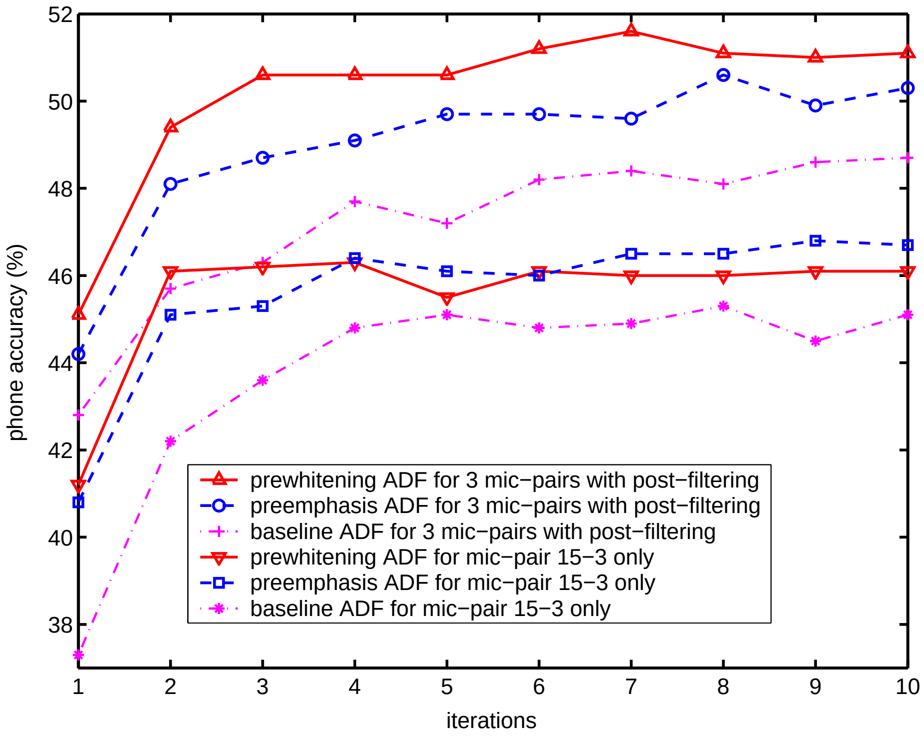


Figure 7.1: Phone accuracies (%) with and without prewhitening pre-processing and multi-ADF post-processing for 10 iterative passes of $N = 400$, $\gamma = 0.005$, $\beta = 0.8$, and recognition under monophone acoustic model.

Number of microphones	baseline	preemphasis	inverse-PSD	block-iterative+ inverse-PSD
One pairs	36.2/37.2	38.7/40.8	39.3/41.0	44.3/43.7
Two pairs	40.3/41.5	41.8/43.1	41.1/42.6	50.2/46.3
Three pairs	42.8/43.0	45.7/44.4	45.8/44.9	50.1/47.9
Four pairs	43.5/44.6	44.8/45.4	46.5/46.4	50.9/49.2

Table 7.3: Comparison of phone recognition accuracies (%) with triphone/monophone models for the first iteration pass of ADF algorithms on different number of microphone pairs, with and without prewhitening and block-iterative ADF.

Number of microphones	baseline	preemphasis	inverse-PSD
One pairs	44.3/45.1	46.5/46.9	46.3/46.6
Two pairs	50.9/48.2	51.7/49.2	51.6/48.7
Three pairs	49.9/48.6	52.2/50.2	51.6/51.1
Four pairs	50.8/49.1	52.5/50.6	51.8/51.3

Table 7.4: Comparison of phone recognition accuracies with (%) triphone/monophone models for the tenth iteration pass of batch ADF processing on different number of microphone pairs, with and without prewhitening.

	iteration pass 1	iteration pass 10
Phone accuracy	43.6/43.3	49.5/49.2

Table 7.5: Phone accuracy (%) with triphone/monophone models of comparative experiment on delay-and-sum beamforming for the integration of multi-ADF outputs.

7.2.2 Post-processing and multi-ADF integration

The phone accuracy results for ADF processing with and without multi-ADF post-processing are also shown in Figure 7.1 for 10 iterative passes of baseline/prewhitened ADFs. The combination of inverse-PSD prewhitening and multiple block-iterative ADF processing achieved highest recognition accuracy with only one iteration pass, as shown in Table 7.3. The advantages of multi-ADF post-filtering are also demonstrated by the recognition results from multi-pass processing of batch ADF in Table 7.4. The phone accuracies are consistent with the results of TIR gains presented in Table 3.1 of Section 3.4.

A simple comparative experiment of delay-and-sum beam-forming was performed, using four pairs of ADF processing with inverse-PSD prewhitening on speech mixtures prior to ADF, as a reference to the performances of post-filtering. The beamforming delay parameters were determined by first computing the pair-wise cross-correlations between an ADF output target speech with the reference output target speech which was associated with microphone pair $15-3$, and then peak-picking from the lags of cross-correlation functions. The resulting delay parameters were -1, 0, 0, 1 for the microphone pairs $16-2$, $15-3$, $14-4$ and $13-5$, respectively. Phone recognition accuracy for the beamforming scheme is shown in Table 7.5, with phone recognitions based on both monophone and triphone models. Comparing the phone recognition results of Table 7.5 with those in the bottom row of inverse-PSD in both Tables 7.3 and 7.4, one observes that although beamforming also improved phone accuracy over the cases of one pair of microphones, the proposed post filtering method was more effective in integrating ADF outputs. For example, with ten iteration passes, Beamforming led to phone accuracy of 49.5% and 49.2% for triphone and monophone models, whereas post filtering led to accuracy of 51.8% and 51.3% for the two models. The superior performance of the post-filtering method is due to the fact that post-filtering not only does time alignment on multiple ADF outputs, but it also reduces spectral distortions of these outputs.

7.2.3 Comparative BSS and recognition experiments

The convolutive BSS method of Douglas and Sun (DS-BSS) [12] that is based on a mutual information criterion was implemented as a comparison with ADF separation algorithms. The block-updated natural-gradient [77] based adaptations of separation filter matrices $\mathbf{B}(n), n = 0, \cdots, N-1$ are listed below.

$$\mathbf{B}^{(k+1)}(n) = \mathbf{B}^{(k)}(n) + \mathbf{\Gamma}(k)\left[\mathbf{A}^{(k)}(n) - \mathbf{f}\left(\mathbf{y}(k-N)\mathbf{u}^T(k-n)\right)\right], \tag{7.1}$$

where the non-linear function $f(\cdot)$, defined on each elements of the input signal vector $\mathbf{y}$, is chosen to be

$$f(y) = -\frac{\partial \ln p(y)}{\partial y}. \tag{7.2}$$

Based on the assumption that the speech signals follow the Laplacian distribution, it is simplified to

$$f(y) = sign(y) \tag{7.3}$$

	iteration pass 1	iteration pass 10
Phone accuracy	30.5/34.1	39.0/39.0

Table 7.6: Phone accuracy (%) with triphone/monophone models of comparative experiment on DS-BSS.

The elements of the diagonal matrix of step-sizes, $\mathbf{\Gamma}(k)$, is computed similar to ADF, using a normalization by input speech signal powers as

$$\gamma_i(k) = \frac{\gamma_0}{\beta + \sum_{q=N+1}^{2N} y_i(k-q) f(y_i(k-p))}, \tag{7.4}$$

where γ_0 is a constant gain factor and $\beta > 0$ is used to avoid divide-by-zero and improve the robust ness of the algorithm. The (i, j)-th entry of $\mathbf{A}^{(k)}(n)$ in 7.1 is

$$a_{ij}^{(k)}(n) = f\left(y_i(k-N)\right) u_{ij}(k-n), \tag{7.5}$$

with

$$u_{ij}(k) = \sum_{q=0}^{N} b_{ij}^{(k)}(N-q) y_i(k-q). \tag{7.6}$$

The comparative experiment on the DS-BSS method was performed as follows. The algorithm was implemented by block-wise adaptation of demixing FIR filters with all tuning parameters (e.g., adaptation step size) empirically chosen to achieve best results. The phone recognition accuracies achieved from one and ten iteration passes are shown in Table 7.6, where recognition used both monophone and triphone models. Comparing the recognition results of Table 7.6 with those of single pair of microphones of ADF in Tables 7.3 and 7.4 reveal that under the studied experimental conditions, ADF is more effective than DS-BSS in separating speech sources for automatic speech recognition. The somewhat simpler approach of ADF to the source separation problem as compared with the DS-BSS approach may account for the superior performance of ADF. First, although the mixing system used by DS-BSS,

Mixture	Baseline + Inverse-PSD	Block-iterative + Inverse-PSD	VSS-ADF
29.1	41.0	43.7	47.8

Table 7.7: Phone accuracy (%) comparison of VSS-ADF methods with baseline and block-iterative ADF, recognition based on monophone models.

shown in Figure 1.1, contains four filters, the ADF system only employs two filters for decorrelating output signals, whereas DS-BSS attempts to estimate all the four filters. For the latter method, adaptation may become less focused on the cross-coupled acoustic path filters that are important for source separation. Second, ADF and DS-BSS utilizes different separation criteria and adaptation methods, i.e., decorrelation versus mutual information miminization, and stochastic approximation versus gradient descent, and in each aspect ADF is simpler and hence may be more reliable for processing a limited amount of data.

7.2.4 Variable step-sizes

The phone recognition results for VSS-ADF on single microphone pair $15-3$ are shown in Table 7.7. The separation parameters for VSS-ADF were introduced in Section 4.4. The comparison between VSS-ADF and other ADF methods shows that the separation performance of one-pass ADF is significantly improved by VSS techniques.

However, further attempts to integrate the VSS-ADF algorithm for multi-ADF post-processing proved not promising. This is due to the disadvantage of the GAS gain adaptation, which is very sensitive to the selection of GAS step size ε. Although VSS-ADF achieved significant enhancement in separation performance for single pair of ADF processing, it is difficult to find a set of parameters that could guarantee the optimal adaptations for multiple pairs of ADF. Therefore, efforts should still be devoted for the improvement of VSS-ADF in this respect so that the overall optimal separation performances could be achieved when multiple sensors are available.

7.2.5 NC-ADF and adaptive speech enhancement

For phone recognitions in diffuse noise conditions, monophone models were used for their robustness and less complexity compared with triphones. Phone accuracy results in simulated and real diffuse noise cases are shown in Figure 7.2, respectively. In a comparative experiment for the adaptive speech enhancement, TDC-GSub algorithm was implemented without update rate reduction for matrices computations in subspace decomposition and transformation mentioned in Section 6.3.2. Instead, the rate of subspace update was set the same as the frame rate and L was set to be $25ms$, which makes the noise reduction extremely time consuming. The corresponding phone accuracy results are illustrated in Figure 7.3. However, comparing the results in Figure 7.2 with 7.3, we only see a slight degradation in phone accuracy caused by those practical speedup techniques for TDC-GSub method.

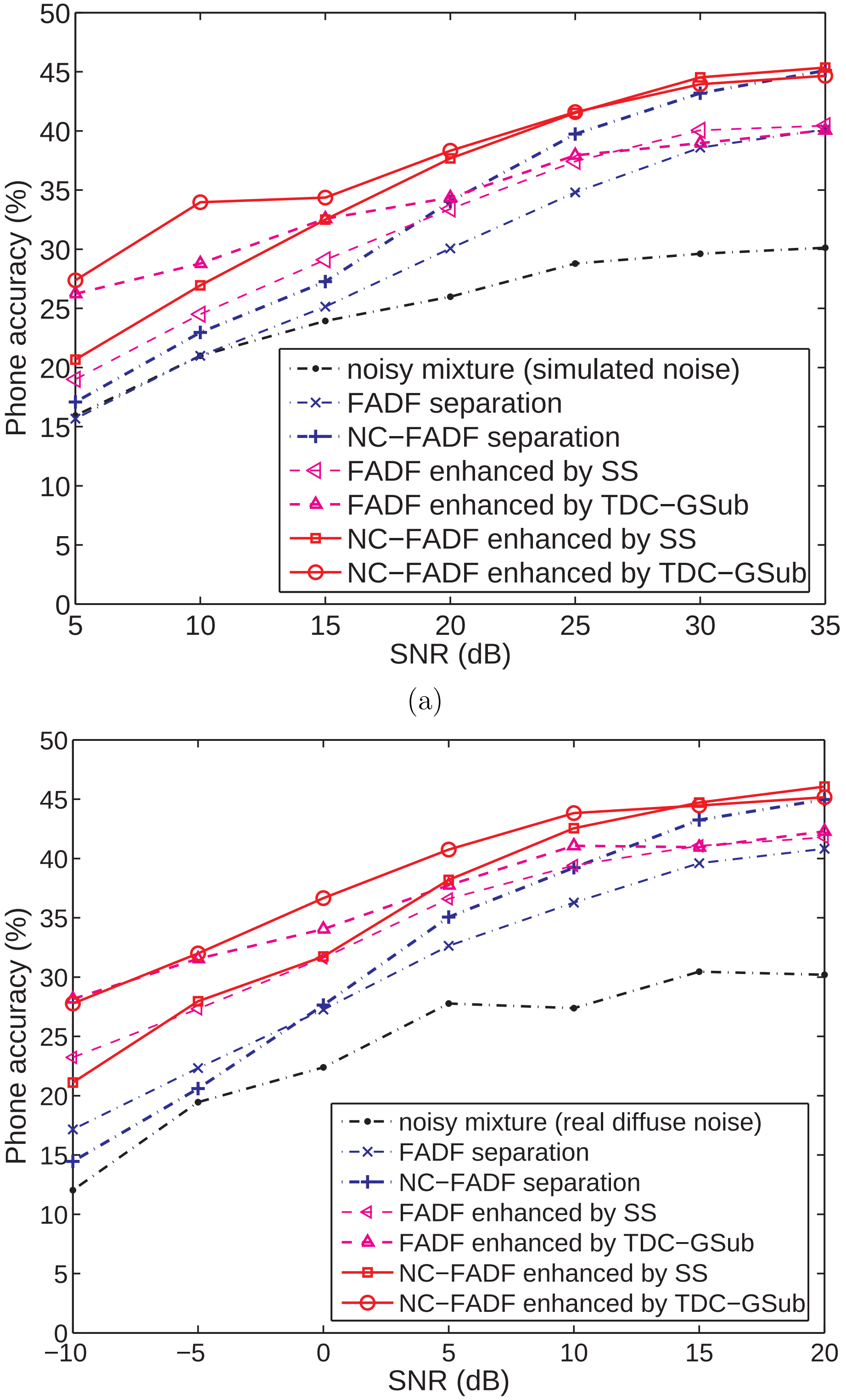
It also is observed that the adaptive enhancement techniques significantly improved the phone recognition accuracy of the ADF separation outputs. The combination of NC-FADF with TDC-GSub achieved highest performance. At low SNR's, the gains of phone recognition accuracy are mainly provided by speech enhancement; at high SNR's, the improvement of accuracy comes mainly from better noise compensated speech separation.

For simplified NC-ADF under assumptions of uncorrelated white noises, Table 7.8 lists the results of a simplified equivalent recognition experiment. To reduce complexity, the adaptive speech enhancement was not performed, and the recognition experiments were carried out on clean speech mixtures separated by ADF filters obtained from baseline ADF and simplified NC-ADF, where the phone accuracy result measures the equivalent separation performances of the ADF algorithms that obtains

SNR (dB)	Baseline ADF	Simplified NC-ADF
0	32.9	39.6
5	35.8	40.6
10	37.3	41.9
15	39.3	43.5
20	42.3	43.5
25	42.4	43.8
30	43.5	44.0

Table 7.8: Phone accuracies (%) with simplified NC-ADF method under uncorrelated white noise, based on monophone models.

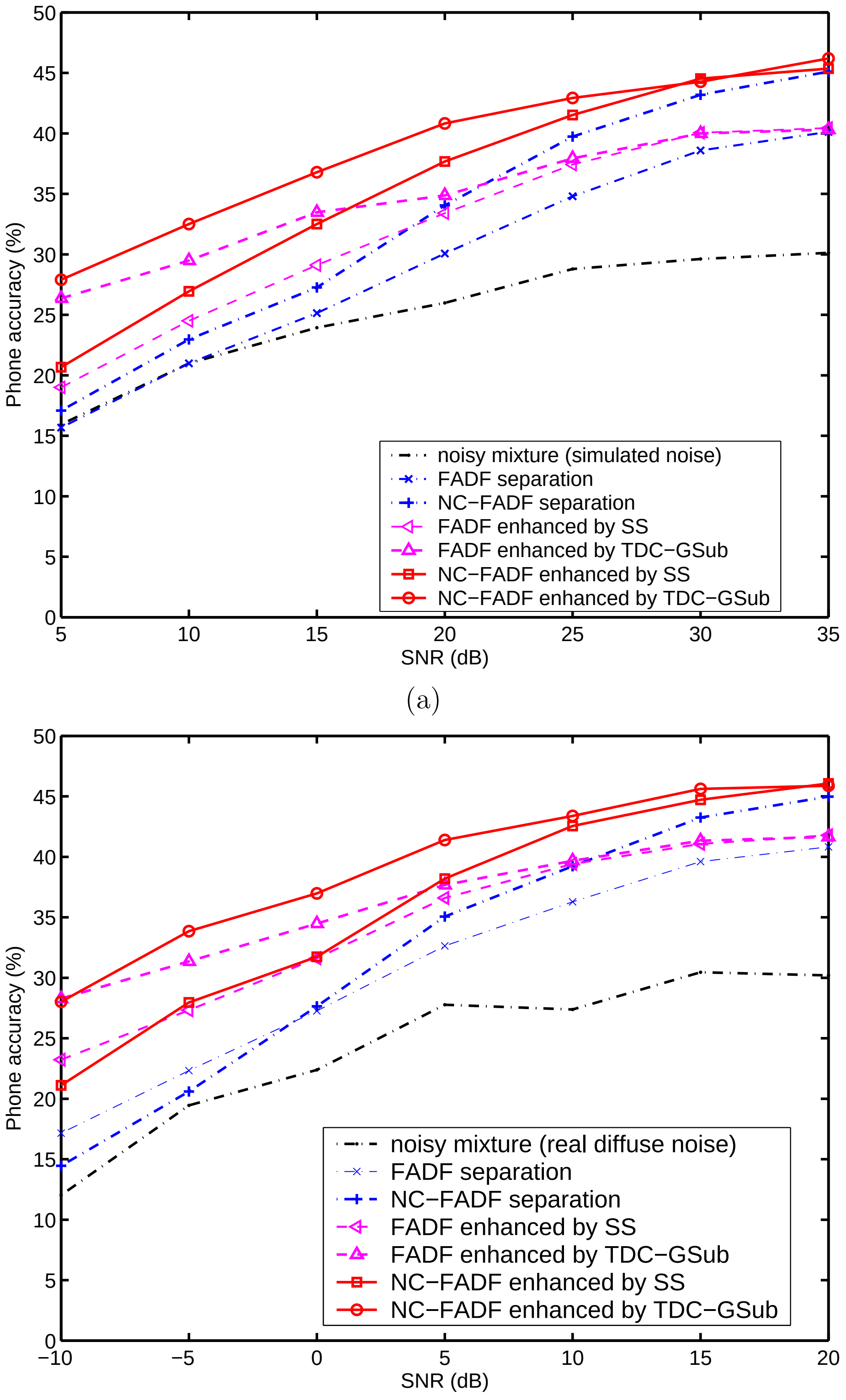
their separation filters. It is also observed that the improvement by simplified NC-ADF was more significant when SNR was at lower levels.



(a)

(b)

Figure 7.2: Phone accuracies under simulated noise (TDC-GSub implemented with reduction of update rate for subspace computations), with $L = 40(2.5ms)$: (a) simulated noise; (b) real diffuse noise.



(a)

(b)

Figure 7.3: Phone accuracies under simulated noise (TDC-GSub implemented without reducing the update rate of subspace computations), with $L = N = 400(25ms)$: (a) simulated noise; (b) real diffuse noise.

Chapter 8

CONCLUSIONS AND FUTURE WORKS

In this dissertation, enhancement algorithms of ADF separation model for the application of automatic speech recognition (ASR) are discussed. Experiments of speech separation and recognition performed in both clean and noisy conditions demonstrated the effectiveness of the proposed algorithms.

8.1 Conclusions

The proposed techniques are successful in the enhancement of ADF separation model for robust speech recognition. Their performances are summarized as follows.

1. The prewhitening processing of inverse-PSD and preemphasis improves the robustness of ADF adaptation and reduced steady state filter estimation error, with only a slight computational overhead. They are easy to implement and suitable to be integrated with other enhancement techniques for ADF; the block-iterative implementation significantly improves the convergence speed of ADF

adaptation for ASR applications by utilizing mixture information more efficiently with longer delay tolerable in ASR. The multi-ADF post-filtering proved its advantages in suppressing residual interferences and reverberations contained in separated speech by single ADFs.

2. VSS-ADF algorithm that used combination of GAS gain factor and the source energy based gain factor obtained fast convergence rate and low steady-state filter estimation error at noise free conditions. However, when GAS technique was incorporated, VSS-ADF demonstrated sensitivity to the choice of parameters and did not work well under noisy conditions.

3. The combination of the techniques of FADF and NC-FADF with adaptive speech enhancement significantly improved the phone recognition accuracies of target speech corrupted by jammer speech and diffuse background noise.

4. The integration of NC-FADF with TDC-GSub technique achieved highest performances under both types of noises. The combinations of a set of practical speedup techniques significantly improved the implementation speed for TDC-GSub to be integrated with NC-FADF. At low SNRs, the gains of phone accuracy are mainly provided by speech enhancement; at high SNRs, the improvement of accuracy comes mainly from better noise compensated speech separation.

8.2 Future Works

The potential research topics for further improvement of ADF separation algorithm for the task of ASR are listed as follows.

1. Further efforts are required to decrease the sensitivity of current VSS-ADF algorithms to the choice of parameters. There are potentials for multi-ADF processing to be integrated with VSS techniques so that more streamlined integration of multi-ADF post-filtering may be implemented with better overall separation performance. Again, ideas could be borrowed from similar techniques in LMS adaptive filtering, because of the similarity between two types of algorithms.

2. Evaluation of the integration of NC-FADF with adaptive types of Input Noise Statistics Estimation module for non-stationary noises with faster time-varying statistics. An accurate estimation of noise statistics from the noisy speech simultaneously will be the important to the success of the whole system.

3. In addition to the current speedup techniques used in TDC-GSub implementation for noisy ADF outputs, faster subspace-based adaptive speech enhancement methods may exist, in diffuse noise fields. Potential choices of techniques are: subspace tracking algorithms for faster update of eigen-decomposition of signal covariance matrices, fast inversion techniques for Toeplitz matrices, etc. For example, recursive estimation methods similar to [78] might exist for the update of the eigen-subspace. Fast tracking algorithm design and Testing should focus on how to speedup the subspace method without significant degradation of performances for speech enhancement. For the perspective of utilizing Toeplitz structures to speedup matrix inversions, many techniques could be applied for the fast solution of symmetric Toeplitz system [79].

4. Evaluation of other speech enhancement techniques for noisy ADF outputs, e.g., algorithms incorporating human perceptual models, etc. It is also possible to incorporate more knowledge about structure of speech signals for better removal of residual interferences.

5. Modification of post-filtering techniques for better integration of multi-microphone ADF pair outputs. Reverberation effects in ADF outputs still contribute to a large portion of degradation in phone accuracy. There are still potentials in improving the ASR accuracies by countering reverberation with either multi-channel or single channel methods. For example, microphone array processing techniques, such as generalized sidelobe canceller (GSC) [80] or other adaptive beam-forming methods [11] could be tested in new data sets to combine multiple ADF outputs. Due to channel mismatches in the experimental data, array processing could not be performed based on the current acoustic data set.

6. More compact integration of separation with ASR task are potentially achievable by considering the interaction between ADF separation model and models used in speech recognition, either in feature domain or acoustic model level. Effects of speech separation on feature extraction and acoustic modeling could be analyzed for potential compensation in feature domain or adaptation in acoustic model parameter space.

7. Generalization and evaluation of the proposed enhancement algorithms for MIMO scenarios with $M > 2$. The basic ideas of the proposed methods are still valid for MIMO cases. However, the exact forms of the resulting algorithms for MIMO models could be different from that of the TITO model.

Appendix A. Frequency Sampling Method for Inverse-PSD Prewhitening Filter Design

The computation procedures for the design of FIR filter coefficients [54] that implement the inverse-PSD processing in Section 3.1 are listed below.

Let L be the number of frequency samples, the desired response vector $\mathbf{h}_d \in \mathbb{C}^{L\times 1}$ are sampled at the frequency points $[\omega_1, \cdots, \omega_L]$. The coefficient vector $\mathbf{w} = [w_0, \cdots, w_{p-1}]^T$ of the p-tap FIR filter $W(z)$ to be designed from these frequency samples are computed in a vector form by

$$\mathbf{w} = \left(\mathbf{F}_c^T \mathbf{F}_c\right)^{-1} \mathbf{F}_c^T \mathbf{h}_c, \tag{A.1}$$

where

$$\mathbf{h}_c = \left[\mathbf{h}_r^T, \mathbf{h}_i^T\right]^T, \tag{A.2}$$

$$\mathbf{F}_c = \left[\mathbf{F}_r^T, \mathbf{F}_i^T\right]^T, \tag{A.3}$$

with $\mathbf{h}_r$ and $\mathbf{h}_i$ the real and imaginary components, respectively, of the desired response vector $\mathbf{h}_d$. Matrices $\mathbf{F}_r$ and $\mathbf{F}_i$ are the real and imaginary parts of the constant matrix

$$\mathbf{F}_w = \left[\mathbf{f}_w(e^{j\omega_1}), \cdots, \mathbf{f}_w(e^{j\omega_L})\right], \tag{A.4}$$

respectively, with

$$\mathbf{f}_w(e^{j\omega_k}) = \left[1, e^{-j\omega_k}, \cdots, e^{-j(p-1)\omega_k}\right]^T{}_{,k=1,\cdots,L} \,. \tag{A.5}$$

Appendix B. Derivation of Simplified ADF Error Analysis

From (4.10), the mixing system output has the correlation matrix

$$\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}} = \begin{bmatrix} \mathbf{R}_{\tilde{\mathbf{y}}_1\tilde{\mathbf{y}}_1} & \mathbf{R}_{\tilde{\mathbf{y}}_1\tilde{\mathbf{y}}_2} \\ \mathbf{R}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_1} & \mathbf{R}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_2} \end{bmatrix} = \tilde{\mathbf{H}} \cdot \mathbf{R}_{\bar{\mathbf{s}}\bar{\mathbf{s}}} \cdot \tilde{\mathbf{H}}^T. \tag{B.1}$$

where the auto- and cross-correlations are reduced by (4.13) as

$$\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j} = p_i\tilde{\mathbf{H}}_{ji}\tilde{\mathbf{H}}_{ji}^T + p_j\tilde{\mathbf{H}}_{jj}\tilde{\mathbf{H}}_{jj}^T, \tag{B.2}$$

$$\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_i} = p_j\tilde{\mathbf{H}}_{jj}\tilde{\mathbf{H}}_{ij}^T + p_i\tilde{\mathbf{H}}_{ji}\tilde{\mathbf{H}}_{ii}^T. \tag{B.3}$$

The correlation relationship (2.12) derived from the analysis of ADF system based on (2.2) shows that

$$\mathbf{R}_{\mathbf{y}_j\mathbf{v}_j} = \mathbf{R}_{\mathbf{y}_j\mathbf{y}_j} - \mathbf{R}_{\mathbf{y}_j\tilde{\mathbf{y}}_i}\mathbf{G}_{ji}^T, \tag{B.4}$$

where the input correlation matrices $\mathbf{R}_{\mathbf{y}_j\mathbf{y}_j}$ and $\mathbf{R}_{\mathbf{y}_j\tilde{\mathbf{y}}_i}$ are sub-matrices of $\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j}$ and $\mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_i}$, respectively, i.e.,

$$\mathbf{R}_{\mathbf{y}_j\mathbf{y}_j} = \left[\mathbf{I}_N \ \mathbf{0}_{N\times(N-1)}\right] \mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j} \left[\mathbf{I}_N \ \mathbf{0}_{N\times(N-1)}\right]^T, \tag{B.5}$$

$$\mathbf{R}_{\mathbf{y}_j\tilde{\mathbf{y}}_i} = \left[\mathbf{I}_N \ \mathbf{0}_{N\times(N-1)}\right] \mathbf{R}_{\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_i}. \tag{B.6}$$

Substituting (B.2) and (B.3) into (B.5) and (B.6) respectively, and then both into (B.4), we obtain (4.15) together with (4.16) and (4.17).

Similarly, correlation analysis for (2.2) also has

$$\mathbf{r}_{y_j \mathbf{v}_j} = \mathbf{r}_{y_j \mathbf{y}_j} - \mathbf{G}_{ji} \mathbf{r}_{y_i \tilde{\mathbf{y}}_i}, \tag{B.7}$$

where, under the assumption of white uncorrelated sources, from (B.2) and (B.3), we have

$$\mathbf{r}_{y_i \mathbf{y}_j} = p_i \tilde{\mathbf{H}}_{ji} \tilde{\mathbf{h}}_{ii} + p_j \tilde{\mathbf{H}}_{jj} \tilde{\mathbf{h}}_{ij}, \tag{B.8}$$

$$\mathbf{r}_{y_i \tilde{\mathbf{y}}_j} = p_i \tilde{\mathbf{H}}_{ii} \begin{bmatrix} \mathbf{I}_{2N-1} \\ \mathbf{0}_{(2N-2)\times(2N-1)} \end{bmatrix} \tilde{\mathbf{h}}_{ii} + p_j \tilde{\mathbf{H}}_{ij} \begin{bmatrix} \mathbf{I}_{2N-1} \\ \mathbf{0}_{(2N-2)\times(2N-1)} \end{bmatrix} \tilde{\mathbf{h}}_{ij}. \tag{B.9}$$

The relations of (B.8) and (B.9) reduce (B.7) to (4.18) together with (4.19) and (4.20).

Appendix C. Derivation of the Fast Implementation of Compensation Terms

The FFT implementation of the compensation term introduced in Section 5.3.1 are derived as follows. Without loss of generality, we only list the derivation of Eqs (5.21) and (5.22) for the fast implementation of $\mathbf{a}_{ij} = \mathbf{G}_{ji}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i}$. The detailed derivations of other terms are omitted.

From the definition of system matrix (2.4), we have

$$\mathbf{a}_{ij} = \mathbf{G}_{ji}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i} = \begin{bmatrix} \sum_{n=0}^{N-1} g_{ji}(n) r_{n_in_i}(n) \\ \sum_{n=0}^{N-1} g_{ji}(n) r_{n_in_i}(n+1) \\ \vdots \\ \sum_{n=0}^{N-1} g_{ji}(n) r_{n_in_i}(n+N-1) \end{bmatrix}, \tag{C.1}$$

with the k^{th} component

$$a_{ij}(k) = \sum_{n=0}^{N-1} g_{ji}(n) r_{n_in_i}(n+k)_{,k=0,\cdots,N-1}\,. \tag{C.2}$$

Let $k = 2N - 2 - q$ (i.e. $q = 2N - 2 - k$, $q = N - 1, \cdots, 2N - 2$) and reverse the order of the sequence $r_{n_i n_i}(n)$,

$$\xi_{ij}^a(n) = \{r_{n_i n_i}(2N-2), \cdots, r_{n_i n_i}(N-1), \cdots, r_{n_i n_i}(0)\} = r_{n_i n_i}(2N-2-n), \quad \text{(C.3)}$$

we can rewrite (C.2) as

$$a_{ij}(2N-2-q) = \sum_{n=0}^{N-1} g_{ji}(n) r_{n_i n_i}\left(2N-2-(q-n)\right), \quad \text{(C.4)}$$

which becomes

$$a_{ij}(2N-2-q) = \sum_{n=0}^{N-1} g_{ji}(n) \xi_{ij}^a(q-n)_{,q=N-1,\cdots,2N-2} \quad \text{(C.5)}$$

by further utilizing (C.3). Denote R.H.S. of (C.5) as $z(q)$, then it could be represented as linear convolution

$$a_{ij}(2N-2-q) = z(q) = g_{ij}(q) * r_{n_i n_j}(q)|_{q=N-1,\cdots,2N-2} \quad \text{(C.6)}$$

Since $k = 2N - 2 - q$, changing the variable back, we have

$$a_{ij}(k) = z(q)|_{q=2N-2-k} = g_{ij}(q) * r_{n_i n_j}(q)|_{q=2N-2-k,k=0,\cdots,N-1}. \quad \text{(C.7)}$$

Similarly, Eqs. (5.23)-(5.27) could also be obtained by exploiting the Toeplitz structure of correlation/system matrices.

Bibliography

[1] L.R. Rabiner and B.H. Juang, *Fundamentals of speech recognition*, Prentice Hall, Englewood Cliffs, NJ, 1800.

[2] K.F. Lee and H.W. Hon, "Speaker independent phone recognition using hidden markov models," *IEEE Trans. ASSP*, vol. 37, pp. 1641–1648, Nov. 1989.

[3] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Prentice Hall PTR, 2001.

[4] Y. Gong, "Speech recognition in noisy environment: a survey," *Speech Commun.*, vol. 16, pp. 261–291, 1995.

[5] P. J. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. dissertation, Carnegi Mellon University, 1996.

[6] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. ASSP*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[7] J. H. L Hansen and M. A. Clements, "Constrained iterative speech enhancement with application to speech recognition," *IEEE Trans. SP*, vol. 39, no. 4, pp. 795–805, Apr. 1991.

[8] B. E. D. Kingsbury, *Perceptually Inpired Signal-processing Strategies for Robust Speech Recognition in Reverberant Environments*, Ph.D. dissertation, University of California-Berkeley, 1998.

[9] N. Roman and D. Wang, "Pitch-based monaural segregation of reverberant speech," *J. Acoust. Soc. Am.*, vol. 120, no. 1, pp. 458–469, July 2006.

[10] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent component analysis*, John Wiley, 2001.

[11] M. Brandstein and D. Ward (Eds.), *Microphone arrays: signal processing techniques and applications*, Springer-Verlag, Berlin, 2001.

[12] S.C. Douglas and S. Sun, "Blind separation of acoustical mixtures without timedomain deconvolution or decorrelation," in *Proc. NNSP*, 2001, pp. 323–332.

[13] R. Aichner and H. Buchner and W. Kellermann, "Convolutive blind source separation for noisy mixtures," in *Proc. CFA/DAGA*, 2004, *http://www.lnt.de/LMS/publications/web/lnt2004_2.pdf*.

[14] H. Buchner and R. Aichner and W. Kellerman, "Trinicon: A versatile framework for multichannel blind signal processing," in *Proc. ICASSP*, 2004, vol. 3.

[15] S. Araki, S. Makino, A. Blin, and et. al., "Underdetermined blind separation for speech in real environment with sparseness and ica," in *Proc. ICASSP*, 2004, vol. 3, pp. 881–884.

[16] T-W. Lee and A. Ziehe and R. orglmester and T. J. Sejnowski, "Combining time-delayed decorrelation and ica: Towards solving the cocktail party problem," in *Proc. ICASSP*, 1998, pp. 1249–1252.

[17] E. Weinstein, M. Feder, and A. V. Oppenheim, "Multi-channel signal separation by decorrelation," *IEEE Trans. SP*, vol. 43, pp. 405–413, Oct. 1993.

[18] D. Yellin and E. Weinstein, "Criteria for multichannel signal separation," *IEEE Trans. SP*, vol. 42, pp. 2158–2168, Aug. 1994.

[19] D. Yellin and E. Weinstein, "Multichannel signal separation: methods and analysis," *IEEE Trans. SP*, vol. 44, pp. 106–118, Jan. 1996.

[20] K. Yen and Y. Zhao, "Adaptive co-channel speech separation and recognition," *IEEE Trans. SAP*, vol. 7, pp. 138–151, 1999.

[21] K. Yen and Y. Zhao, "Adaptive decorrelation filtering for separation of co-channel speech signals from $M > 2$ sources," in *Proc. of ICASSP*, 1999, pp. 801–804.

[22] K. Yen and Y. Zhao, "Lattice-ladder structured adaptive decorrelation filtering for cochannel speech separation," in *Proc. of ICASSP*, 2000, pp. 388–391.

[23] Y. Zhao, K. Yen, S. Soli, S. Gao, and A. Vermiglio, "On application of adaptive decorrelation filtering to assistive listening," *Journal of Acoust. Society America*, vol. 111, no. 2, pp. 1077–1085, 2002.

[24] F. Ehlers and H.G. Schuster, "Blind separation of convolutive mixtures and an application in automatic speech recognition in a noisy environment," *IEEE Trans. SP*, vol. 45, no. 10, pp. 2608–2612, Oct. 1997.

[25] S. V. Gerven and D. V. Compernolle, "Signal separation by symmetric adaptive decorrelation: Stability, convergence, and uniqueness," *IEEE Trans. SP*, vol. 43, pp. 1602–1612, July 1995.

[26] U. Lindgren, T. Wigren, and H. Broman, "On local convergence of a class of blind separation algorithms," *IEEE Trans. SP*, vol. 43, pp. 3054–3058, Dec. 1995.

[27] U. Lindgren, H. Sahlin, and H. Broman, "Source separatin using second order statistics," in *Proc. EUSIPCO*, 1996, pp. 699–702.

[28] U. Lindgren and H. Broman, “Source separation using a criterion based on secondorder statistics,” *IEEE Trans. SP*, vol. 46, no. 7, pp. 1837–1850, Jul. 1998.

[29] D. Chan, P. Rayner, , and S. Godsill, “Multi-channel signal separation,” in *Proc. ICASSP*, 1996, pp. 649–652.

[30] C. Jutten, L. N. Thi, E. Dijkstra, E. Vittoz, , and J. Caelen, “Blind separation of sources: An algorithm for separation of convolutive mixtures,” in *Proc. Int. Signal Processing Workshop High Order Statist.*, Jul. 1991.

[31] L. N. Thi and C. Jutten, “Blind source separation for convolutive mixtures,” *Signal Processing*, vol. 45, pp. 209–229, 1995.

[32] A. Mansour, C. Jutten, , and P. Loubaton, “Subspce method for blind separation of sources in convlutive mixture,” in *Proc. EUSIPCO*, 1996, pp. 2081–2084.

[33] A. Gorokhov and O. Loubaton, “Subspace based techniques for second order blind separation of convolutive mixtures with temporally correlated sources,” *IEEE Trans. on Circuits and Systems*, vol. 44, no. 9, pp. 813–820, Sept. 1997.

[34] A. Mansour, C. Jutten, , and P. Loubaton, “Adaptive subspace algorithm for blind separation of independent sources in convolutive mixture,” *IEEE Trans. SP*, vol. 48, no. 2, pp. 583–586, Feb. 2000.

[35] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, “A blind source separation technique using second-order statistics,” *IEEE Trans. SP*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[36] H. Robbins and S. Monro, “A stochastic approximation method,” *Annals of Math. Stat.*, vol. 22, pp. 400–407, 1951.

[37] J. R. Blum, “Multidimensional stochastic approximation methods,” *Annals of Math. Stat.*, vol. 25, pp. 736–744, 1954.

[38] B. B. Bauer, “Stereophonic earphones and binaural loudspeakers,” *J. Acoust. Soc. Am.*, vol. 9, no. 2, pp. 148–151, 1961.

[39] A. Sibbald, “Transaural crosstalk cancellation,” [Online]. Available: *http://www.sensaura.com/whitepapers/pdfs/dev009.pdf.*

[40] M. R. Schroeder, “Models of hearing,” *Proc. IEEE*, vol. 63, no. 9, pp. 1332–1350, Sept. 1975.

[41] M. R. Bai, C. W. Tung, and C. C. Lee, “Optimal design of loudspeaker arrays for robust cross-talk cancellation using the taguchi method and the genetic algorithm,” *J. Acoust. Soc. Am.*, vol. 117, no. 5, pp. 2802–2813, May 2005.

[42] J. A. Abel, “Crosstalk canceler,” *US Patent 6668061.*

[43] C. Kyriakakis, “Fundamental and technological limitations of immersive audio systems,” *Proc. IEEE*, vol. 86, pp. 941–951, 1998.

[44] S. Srinivasan, Y. Shao, Z. Jin, and D.L. Wang, “A computational auditory scene analysis system for robust speech recognition,” in *Interspeech 2006 - ICSLP, Pittsburg, USA*, Sept. 2006, pp. 73–76.

[45] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, “Super-human multi-talker speech recognition: the ibm 2006 speech separation challenge system,” in *Interspeech 2006 - ICSLP, Pittsburg, USA*, Sept. 2006, pp. 97–100.

[46] A. J. W. Kouwe, D. Wang, and G. J. Brown, “A comparison of auditory and blind separation techniques for speech segregation,” *IEEE Trans. SAP*, vol. 9, no. 3, pp. 189–195, Mar. 2001.

[47] Y. Ephraim and I. Cohen, "Recent Advancement in Speech Enhancement," http://ece.gmu.edu/ yephraim/ephraim.html.

[48] L. L. Beranek, *Acoustical measurements*, American Institute of Physics (for the Acoustical Society of America), 1988.

[49] S. T. Neely and J. B. Allen, "Invertibility of a room impulse response," *J. Acoust. Soc. Am.*, vol. 66, no. 1, pp. 165–169, July 1979.

[50] H. Sahlin and H. Groman, "MIMO signal separation for FIR channels: a criterion and performance analysis," *IEEE Trans. SP*, vol. 48, no. 3, pp. 642–649, Mar. 2000.

[51] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1991.

[52] Y. Zhao, R. Hu, and X. Li, "Speedup convergence and reduce noise for enhanced speech separation and recognition," *IEEE Trans. ASLP*, vol. 14, pp. 1235–1244, July, 2006.

[53] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[54] M. J. Link and M. Buckley, "Prewhitening for intelligibility gain in hearing aid arrays," *J. Acoustical Society of America*, vol. 93, no. 4, pp. 2139–2145, Apr. 1993.

[55] L.R. Rabiner and R. W. Schafer, *Digital Processing of Speech*, Prentice Hall, Englewood Cliffs, NJ, 1978.

[56] G.C. Carter, "Coherence and time delay estimation," *Proc. IEEE*, vol. 75, no. 2, pp. 236–255, 1987.

[57] R. L. Bouquin and G. Faucon, "Using the coherence function for noise reduction," *IEE Proceedings-I*, vol. 139, no. 3, pp. 276–280, 1992.

[58] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[59] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, “Numerical Recipes in C: the Art of Scientific Computing ,” Cambridge University Press, 1992.

[60] “RWCP Sound Scene Database in Real Acoustic Environments,” ATR Spoken Language Translation Research Lab, Kyoto, Japan, 2001.

[61] R. Hu and Y. Zhao, “Variable step size adaptive decorrelation filtering for competing speech separation,” in *Eurospeech’05*, Sept. 2005, vol. I, pp. 2297–2300.

[62] V. J. Mathews and Z. Xie, “A stochastic gradient adaptive filter with gradient adaptive step size,” *IEEE Trans. SP*, vol. 41, no. 6, pp. 2075–2087, Jun. 1993.

[63] W. P. Ang and B. F. Boroujeny, “A new class of gradient adaptive step-size LMS algorithms,” *IEEE Trans. SP*, vol. 49, no. 4, pp. 805–810, Apr. 2001.

[64] H. Kesten, “Accelated stochastic approximation methods,” *Annals of Math. Stat.*, pp. 41–49, 1958.

[65] S. C. Douglas and A. Cichocki, “Adaptive step size techniques for decorrelation and blind source separation,” in *Conference Record of the 32nd Asilomar Conf. on Signals, Sys. and Comp.*, Nov. 1995, vol. 2, pp. 1191–1195.

[66] H. C. Woo, “Improved stochastic gradient adaptive filter with gradient adaptive step size,” *Electronics Letters*, vol. 34, no. 13, pp. 1300–1301, Jun. 1998.

[67] R. Hu and Y. Zhao, “Adaptive decorrelation filtering algorithm for speech source separation in uncorrelated noises,” in *ICASSP*, 2005, vol. I, pp. 1113–1116.

[68] R. Hu and Y. Zhao, “Fast noise compensation for speech separation in diffuse noise,” in *ICASSP*, 2006, vol. 5, pp. 865–868.

[69] S. Araki, S. Makino, R. Mukai, and H. Saruwatai, "Equivalence between frequency domain blind source separation and frequency domain adaptive null beamformers," in *Proc. Eurospeech*, Sept. 2001, vol. 4, pp. 2595–2598.

[70] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Trans. SAP*, vol. 11, pp. 109–116, Mar. 2003.

[71] F. Asano, S. Hayamizu, T. Yamada, and S. Nakamura, "Speech enhancement based on the subspace method," *IEEE Trans. SAP*, vol. 8, no. 5, pp. 497–507, Sept. 2000.

[72] K. Yen, J. Huang, and Y. Zhao, "Co-channel speech separation in the presence of correlated and uncorrelated noises," in *ESCA Eurospeech'99*, 1999, pp. 2587–2589.

[73] M. G. Jafari and J. A. Chambers, "Adaptive noise cancellation and blind source separation," in *Proc. Int. Symp. on Independent Component Analysis and Blind Source Separation (ICABSS03), Nara, Japan*, Apr. 2003, pp. 627–632.

[74] J.S. Lim and A.V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proc. of the IEEE*, vol. 67, no. 12, pp. 1586–1604, Dec. 1979.

[75] Y. Hu and C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Trans. SAP*, vol. 11, no. 4, pp. 334–341, July 2003.

[76] S. Young, D. Kershawl, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "HTK Speech Recognition Toolkit," [Online]. Available: http://htk.eng.cam.ac.uk/docs/docs.shtml, 1999.

[77] S. Amari, T. P. Chen, and A. Cichocki, "Nonholonomic orthogonal learning algorithms for blind source separation," *Neural Computation*, vol. 12(6), pp. 1463–1484, June 2000.

[78] G. Mathew, V. U. Reddy, and S. Dasgupta, "Adaptive estimation of eigensubspace," *IEEE Trans. SP*, vol. 43, no. 2, pp. 401–411, Feb. 1995.

[79] G. S. Ammar and W. B. Gragg, "Superfast solution of real positive definite toeplitz systems," *SIAM J. Matrix Anal. Appl.*, , no. 9, pp. 61–67, 1988.

[80] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Trans. on SAP*, vol. 12, no. 6, pp. 561–571, Nov. 2004.

List of Publications

- Rong Hu, and Yunxin Zhao, ”Fast noise compensation and adaptive enhancement for speech separation,” submitted to IEEE Trans. ASLP.

- Yunxin Zhao, Rong Hu, and Xiaolong Li, ”Speedup convergence and reduce noise for enhanced speech separation and recognition,” IEEE Trans. ASLP, vol. 14, no. 4, July, 2006, pp. 1235-1244.

- Rong Hu and Yunxin Zhao, ”Adaptive speech enhancement for speech separation in diffuse noise,” in Interspeech’06, Pittsburgh, USA, Sept. 2006, pp. 2618-2621.

- Rong Hu and Yunxin Zhao, ”Fast noise compensation for speech separation in diffuse noise,” ICASSP’06, Toulouse, France, May 14-19, 2006, Vol V, pp.865-868.

- Rong Hu and Yunxin Zhao, ”Variable step size adaptive decorrelation filtering for competing speech separation,” in Eurospeech’05, Sept. 2005, vol. I, pp. 2297-2300.

- Rong Hu and Yunxin Zhao, ”Adaptive decorrelation filtering algorithm for speech source separation in uncorrelated noises,” ICASSP’05, Vol. 1, Philadelphia, Mar. 18-23, 2005, pp. 1113-1116.

- Yunxin Zhao, Rong Hu, and Xiaolong Li, ”Enhanced speech source separation for robust speech recognition,” Joint Workshop on Hands-Free Speech Commun. and Microphone Arrays (HSCMA’2005), Piscataway, New Jersey, Mar. 17-18, 2005.

- Yunxin Zhao and Rong Hu, ”Fast convergence speech source separation in reverberant acoustic environment,” ICASSP’04, vol.3, May 17-21, 2004 pp. 897-900.

- Yunxin Zhao, Rong Hu, and Satoshi Nakamura ”Whitening processing for blind

separation of speech signals," in Proc. of ICABSS'03 (*4th Int. Symp. on Independ. Component Analysis and Blind Signal Separation*), pp.331-336, 2003.

Vita

Rong Hu was born in Taixing, Jiangsu Province, P. R. China, on February 8, 1972. He received the B.S. and M.S. degrees in electronic engineering from Nanjing University of Aeronautics and Astronautics (NUAA), China, in 1994 and 1997, respectively. He is currently a Ph.D. candidate in the Department of Computer Science, University of Missouri, Columbia.

From 1997 to 2001, he was a Lecturer with the Department of Electronic Engineering, Shanghai Jiaotong University, China. He was a Summer Research Intern in the Speech Technology Group, Microsoft Research, Redmond, WA, from May 2006 to August 2006. His research interests include speech separation and enhancement, audio processing, adaptive signal processing, radar and array processing, automatic speech recognition, pattern recognition, and multimedia applications.