Acoustic Echo Cancellation Over Nonlinear Channels

Xiaojian Lu

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

January 2004

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Acoustic echo cancellation (AEC) is an essential component of effective hands-free telephony. Conventional AEC systems employ linear adaptive filters; therefore transmission channel nonlinearities caused by nonlinear components (especially the vocoders in digital networks) can severely degrade performance.

This dissertation examines the performance of popular conventional AEC algorithms based on adaptive filtering theory in nonlinear channels. In order to study the degradation of the algorithms in nonlinear channels, properties of nonlinear devices such as vocoders and loudspeakers are investigated, and a local linearization model is developed for the analysis of the nonlinear devices. This local linearization model is justified by experiments.

A variable step-size adaptive cross-spectral algorithm is proposed so the acoustic echo can be suppressed even during double-talk (DT) periods. This is important since adaptation is frozen during DT periods in order to avoid divergence of the conventional adaptive filtering algorithm; therefore the power of the residual echo may become higher than that of the original echo in nonlinear channels. In addition, the proposed algorithm does not need a DT detector, which is still part of AEC.

To compensate the echo attenuation loss of AEC algorithms caused by channel nonlinearities, post-filtering techniques are exploited. Combined with a linear adaptive filter, post-filters based on various approaches, namely: Wiener-type post-filter, spectral subtraction, subspace method and pitch extraction, are proposed to further attenuate the echo. Experimental results show that the combined AEC system can suppress the acoustic echo to a satisfactory level in the nonlinear channel.

Subband adaptive filtering is also studied to reduce the computational complexity of the AEC system so that it can be implemented in real-time. To this end, an improved simple design of DFT filter banks is proposed. Furthermore, a post-filter is integrated with an adaptive filter in the subband to significantly suppresses the acoustic echo in the presence of channel nonlinearities. This approach also significantly reduces the computational burden.

Finally, a psychoacoustic approach based on the masking of the human ear is exploited in order to mitigate the artifacts resulting from the abovementioned post-filters. Testing indicates that the proposed method significantly reduces the distortion of near-end speech when DT occurs. This makes any audible residual echo sound more natural since it has less musical noise.

Sommaire

L'annulation d'écho acoustique (AEA) est un élément essentiel d'un poste téléphonique mains libres efficace. Cependant, les imperfections du canal de transmission résultant des composants non linéaires, particulièrement les vocodeurs dans les réseaux numériques, dégradent sévèrement la performance d'un système conventionnel de AEA qui utilise principalement un filtre adaptatif linéaire.

Dans cette dissertation, la performance des algorithmes conventionnels de AEA, lesquels sont basés sur la théorie du filtrage adaptatif, a été examinée dans le canal non-linéaire. Pour expliquer la dégradation de la performance des algorithmes, les propriétés des dispositifs non-linéaires tels que les vocodeurs et les haut-parleurs ont été étudiées et un modèle de linéarisation local a été employé pour l'analyse des modèles non-linéaires, ce qui est justifié expérimentalement.

Nous proposons un algorithme adaptatif à pas variables à spectre croisé de sorte que l'écho acoustique puisse être supprimée même pendant la période d'émission simultanée de parole (ESP). Ceci est motivé par le fait que, dans le canal non-linéaire, la puissance de l'écho résiduel peut être plus grande que celle de l'écho original durant la période d'ESP, où l'adaptation a été gelée pour éviter la divergence possible de l'algorithme de filtrage adaptatif conventionnel. De plus, l'algorithme proposé n'a pas besoin d'un détecteur d'ESP, ce qui fait partie intégrante de la AEA.

Pour compenser la perte d'atténuation de l'écho des algorithmes de AEA causée par les non-linéarités du canal, des techniques de post-filtrage sont exploitées. En combinaison avec un filtre adaptatif linéaire, nous proposons des post-filtres basés sur diverses approches pour atténuer davantage l'écho: filtres de types Wiener, technique de soustraction spectrale, méthodes sous-espace et approche basée sur l'estimation de la hauteur (pitch). Les résultats expérimentaux démontrent que le système combiné de AEA peut supprimer l'écho acoustique à un niveau satisfaisant dans le canal non-linéaire.

Le filtrage adaptatif en sous-bandes est également étudié pour réduire la complexité de calculs du système de AEA de sorte qu'il puisse être utilisé en temps réel. À cet effet, nous proposons une conception simple et améliorée d'une batterie de filtres DFT. De plus, un post-filtre est intégré avec un filtre adaptatif en sous-bande, ce qui réduit de manière significative l'écho acoustique quand les non-linéarités du canal ne sont pas négligées. La complexité de calculs est, par le fait même, réduite remarquablement.

Enfin, une approche psychoacoustique basée sur les propriétés de masquage de l'oreille humaine est exploitée pour atténuer les artéfacts résultant des post-filtres mentionnés ci-dessus. Des tests indiquent que la méthode proposée réduit en grande partie la distorsion de la parole locale quand l'ESP se produit, et fait en sorte que l'écho résiduel, s'il peut être entendu, semble plus naturel, c'est-à-dire qu'il y a moins de bruit musical.

Acknowledgments

I would like to thank my supervisor Professor Benoît Champagne for his invaluable advice and guidance during the course of my Ph.D. studies. I have no doubts that my degree could not have been completed in a timely manner without his help and encouragement. I am grateful for the financial support from Professor Champagne via research contracts from Nortel Networks and the grants from Natural Sciences and Engineering Research Council of Canada and McGill University.

I would also like to recognize Mr. Yasheng Qian for his technical expertise in the field of speech coding and the numerous discussions that have influenced the outcome of this work. I would like to thank my fellow Telecommunications and Signal Processing Laboratory graduate students for their close friendship and support.

Many thanks will be given to Linda Wang for her proofreading my thesis in its entirety. Her comments and critiques did much to enhance this thesis. I wish to also thank François Duplessis-Beaulieu and Benoît Pelletier for their help with the translation of my abstract.

Finally, I am sincerely indebted to my wife, Hongyun, for her continuous love and support during my studies. Her belief in me gave me the strength needed to keep me going all these years, and for this I am deeply grateful. Thank you.

Contents

List of Figures

List of Algorithms

Abbreviations and Acronyms

AAC	Advanced Audio Coding
ACS	Adaptive Cross-Spectral
ADPCM	Adaptive Differential Pulse Code Modulation
AEC	Acoustic Echo Cancellation
AES	Acoustic Echo Suppression
AP	Affine Projection
AR	Autoregressive
ARMA	Autoregressive Moving Average
CELP	Code-Excited Linear Prediction
CODEC	Encoder and Decoder
DFT	Discrete Fourier Transform
DT	Double-Talk
ERLE	Echo Return Loss Enhancement
FAP	Fast Affine Projection
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IDFT	Inverse Discrete Fourier Transform
IIR	Infinite Impulse Response
ITU	International Telecommunication Union
KLT	Karhunen-Loève Transform
LEM	Loudspeaker-Enclosure-Microphone
LMS	Least Mean Squares
LPC	Linear Predictive Coding

MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
NLMS	Normalized Least Mean Squares
OPS	Operations Per Sample
PCM	Pulse Code Modulation
PSD	Power Spectral Density
RLS	Recursive Least Squares
SNR	Signal-to-Noise Ratio
SS	Spectral Subtraction
VAD	Voice Activity Detector
VQ	Vector Quantization

Chapter 1

Introduction

When the telephone was first invented, people had to use both hands to make a telephone call, but soon only one hand was needed [1]. Since then, much research has been done to try to free both hands during phone conversations. Figure 1.1 illustrates the evolution of telephones.

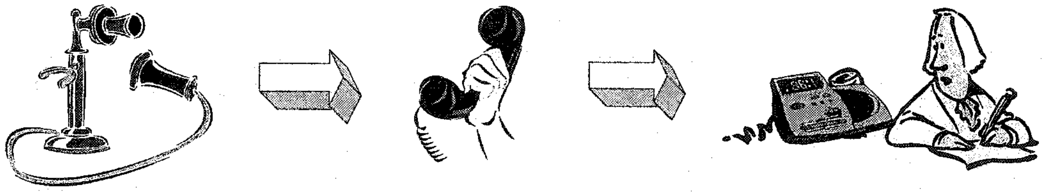

Fig. 1.1 The evolution of the telephones.

The main problem of hands-free devices is that acoustic echo severely degrades the quality of communication. Normally, people hear their own voices through air and bone conduction. This feedback (or echo) is used by the speakers to adjust their volume. During telephone conversations, people find it reassuring to hear an additional echo of their own voices through the earpiece. This is a psychoacoustic phenomenon: most people would find the absence of an echo disturbing, believing that if you cannot hear yourself, then the other person cannot hear you either. However, an echo that is too loud or arrives too late can be annoying to the speaker. In order to avoid any disruption caused by echoes, the time between an original spoken phrase and its echo must be short (normally less than 30 ms), and the echo's level has to be much lower than the original speech's level [2]. From a telephony perspective, this echo of a speaker's own voice through his/her receiver is called *sidetone* [3]. Proper sidetone is desired in the design of telephones. A speaker's tolerance to sidetone depends on the delay time and the level difference between the speaker's voice and the echo signal. Echo tolerance curves can be found in [4].

There are two kind of echoes in telephony: electric echo and acoustic echo. Electric echo (often referred to as talker echo) arises from impedance mismatches in the terminating equipment [5], while acoustic echo is the sound of the speaker's voice returning to the speaker's ear via the acoustic echo path (and via the telephone networks) when a hands-free device is used. Compared to electric echo, acoustic echo has a higher power level and a longer echo path that is also time-varying. Therefore, removing acoustic echo is more challenging than removing electric echo. In fact, electric echo may be regarded as a special case of acoustic echo where the echo path is short and time-invariant. This thesis focuses only on the control of acoustic echoes.

Acoustic echo in hands-free telephones is illustrated in Figure 1.2. The two ends of the communication are referred to as the near-end and the far-end. Without loss of generality, it is assumed here that a hands-free device is used only at the near-end. When a far-end user speaks, his/her voice is played by a loudspeaker at the near-end. Acoustic echo occurs when this loudspeaker signal is picked up by the microphone at the near-end and gets sent back to the far-end. As illustrated in the figure, the loudspeaker signal picked up

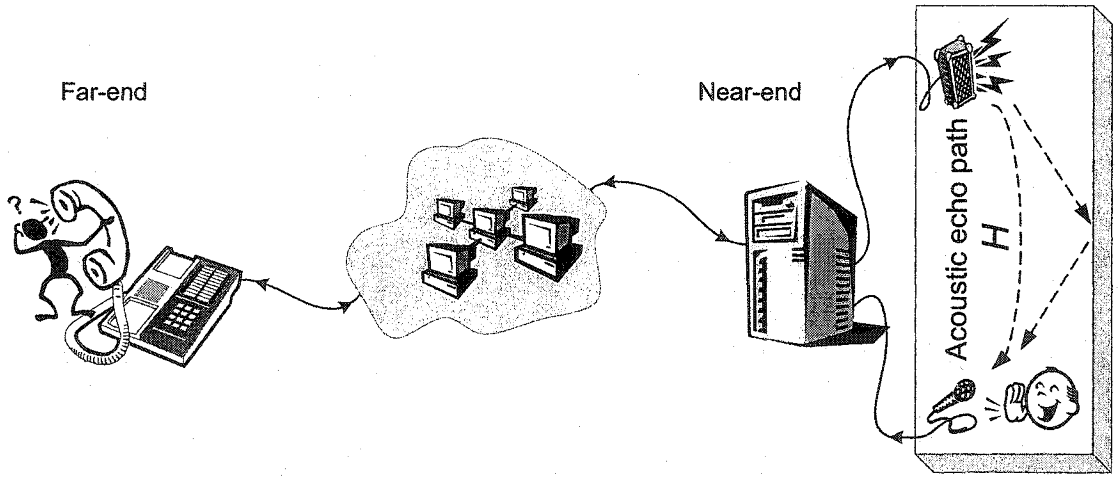

Fig. 1.2 The acoustic echo in the use of the hands-free telephones.

by the microphone consists of both direct sound and reflected sound. The amplitude of the reflected sound is less than the amplitude of the direct sound, because the reflected sound has travelled farther and part of its sound energy has been absorbed by reflecting surfaces [6].

Both the direct sound and the reflections travel back to the far-end with some delay. The delay of the direct sound depends on the distance between the microphone and the loudspeaker, while the delay of the reflections depends on the size of the loudspeaker-enclosure-microphone (LEM) system (for example, an LEM system could be a car compartment in which a hands-free telephone is placed). Since an LEM system usually causes a long delay, the resulting acoustic echo is perceptible to the far-end user, thus the voice quality over the communication channel is noticeably degraded. Furthermore, the acoustic echo feedback may cause howling. As a result, it is extremely important to remove acoustic echo in the use of hands-free devices. Two major approaches to solving this problem have been extensively studied in the past decades: acoustic echo suppression (AES) and acoustic echo cancellation (AEC).

1.1 Acoustic echo suppression (AES)

AES is an intuitive and straightforward approach to suppress acoustic echo. It uses various devices to control the instantaneous levels of the receive and transmit channels. One of the most commonly used AES techniques is voice controlled switching. It changes the insert loss (i.e., negative gains) between both channels according to the direction of main activity, as illustrated in Figure 1.3. As before, the two ends of the communication network in the figure are called the near-end and the far-end, with the hands-free device shown at the near-end. In order to suppress acoustic echo, a large loss is placed in the return path when only the far-end speaker is talking. When the second speaker at the near-end begins talking simultaneously with the first speaker (this situation is called *double-talk*), the inserted loss must be reduced so that the speech of the second speaker is not prevented from reaching the first speaker. This adjustment causes the acoustic echo to be attenuated to some extent [7]. In the use of voice controlled switches with high attenuation, a “comfort noise” is often added to simulate a background noise so the users know that the communication is still on-going. Other AES mechanisms include nonlinear centre-clippers, frequency shifters, and comb filters [8].

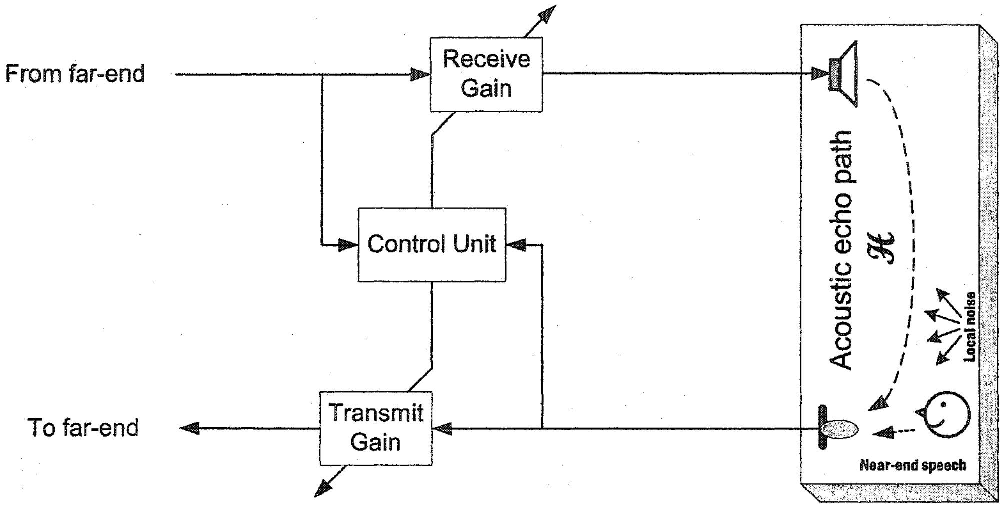

Fig. 1.3 AES with voice controlled switching.

Although AES is simple in structure and not computationally demanding, it has various drawbacks such as speech clipping, frequent loss of syllables as adjustable gains are turned on, user fatigue resulting from the need to synchronize the conversation, and the "pumping effect" in background noise resulting from changes in the instantaneous levels and characteristics of the receive and transmit channels when comfort noise is used.

1.2 Acoustic echo cancellation (AEC)

AEC is a superior method in which an adaptive filter is used to produce a replica of the acoustic echo signal. This replica is then subtracted from the microphone signal prior to its transmission over the communication network so that the acoustic echo is removed. Compared to AES that usually works in half duplex mode, AEC works in full duplex mode. Figure 1.4 shows the principle of AEC with an adaptive filter.

Every hands-free device in a telecommunication system needs its own acoustic echo canceller in order to prevent the acoustic echo produced by that device from transmitting to the other end. Therefore, when hands-free devices are used in both ends of the telecommunication system, two acoustic echo cancellers are needed. Without loss of generality, Figure 1.4 shows only one end with a hands-free device (i.e., the near-end). The signals shown in this system are described as follows:

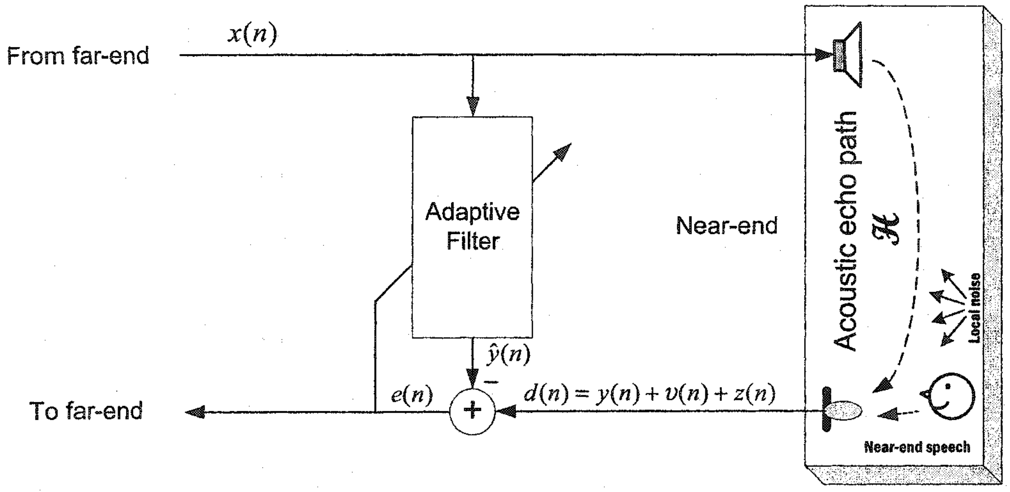

Fig. 1.4 Principle of AEC with an adaptive filter.

- Far-end signal: the signal transmitted from the far-end and received by the AEC system, denoted $x(n)$.
- Near-end signal: the signal transmitted from the near-end and received by the AEC system, denoted $d(n)$. The near-end signal may consist of the following:
  - Acoustic echo: the coupling signal between the loudspeaker and the microphone, denoted $y(n)$.
  - Near-end speech: the signal produced by the near-end user, denoted $\nu(n)$.
  - Background noise: the environment noise signal, denoted $z(n)$.
- Residual signal: the output of the AEC system, i.e., the echo-suppressed signal, which is sent to the far-end, denoted $e(n)$.

As explained before, acoustic echo originates from the coupling between the loudspeaker and the microphone of the hands-free device. Random sound fields composed of direct and reflection waves provide a fundamental model for a finite impulse response (FIR) system [9]. The impulse response of this linear system is the superposition of several delayed and attenuated pulses. In practice, due to continual changes in the acoustic environment (e.g., persons/objects moving), the impulse response is a complicated function of time and the corresponding linear system is time-variant.

As a systematic solution for removing acoustic echo, AEC has been extensively studied in the past decades [10, 8, 11, 12]. AEC is essentially a system identification problem where the Autoregressive Moving Average (ARMA) model can be employed to identify the unknown linear system. In practice, a Finite Impulse Response (FIR) adaptive filter (based on the Moving Average (MA) model) is usually employed in AEC to synthesize the estimated acoustic echo. Infinite Impulse Response (IIR) adaptive filters (based on either the Autoregressive (AR) or the ARMA model) are not as commonly used since they have convergence and stability problems [13], and they do not show obvious advantages over FIR filters [14]. The main challenges of AEC are caused by the following [8]:

- The acoustic echo path is long. For instance, it can be several hundred milliseconds for a typical office room.
- Speech is a highly correlated and non-stationary signal.

To deal with the first challenge, a long FIR adaptive filter is used to compensate the long echo path. This results in high computational cost even for the simplest AEC algorithms, such as the normalized least-mean-square (NLMS) algorithm [15]. In order to reduce computational complexity, methods such as subband adaptive filtering [16, 17, 18] and partially updating the adaptive filter coefficients [19, 20] have been developed. To deal with the second challenge, adaptive filters with faster convergence/tracking speed are needed. Among these, the affine projection (AP) algorithm [21] and its fast version, fast affine projection (FAP) algorithm [22] can be used. These algorithms have fast convergence rates when excited by speech signals.

1.3 AEC over nonlinear channels

Most works on AEC assume that the echo path can be modelled as a slowly time-varying linear system. However, this assumption is no longer valid when nonlinear components along the echo path are taken into account. For analog telephone networks, major nonlinearities of the echo path may be caused by low-quality loudspeakers and overdriven amplifiers.

Speech coding techniques for telephony have been extensively explored in the past decades [23]. Today's third generation digital and wireless networks are employing vocoders (e.g., G.729 [24] and GSM [25]) that significantly lower speech transmission rates with little loss of perceptual quality. However, these codecs can introduce severe distortion in the speech signals. Figure 1.5 shows a centralized AEC system, where the AEC devices are placed in central stations or base stations instead of in user terminals. In such systems, distortions introduced by codecs would result in nonlinearities of the entire echo path for the centralized AEC.

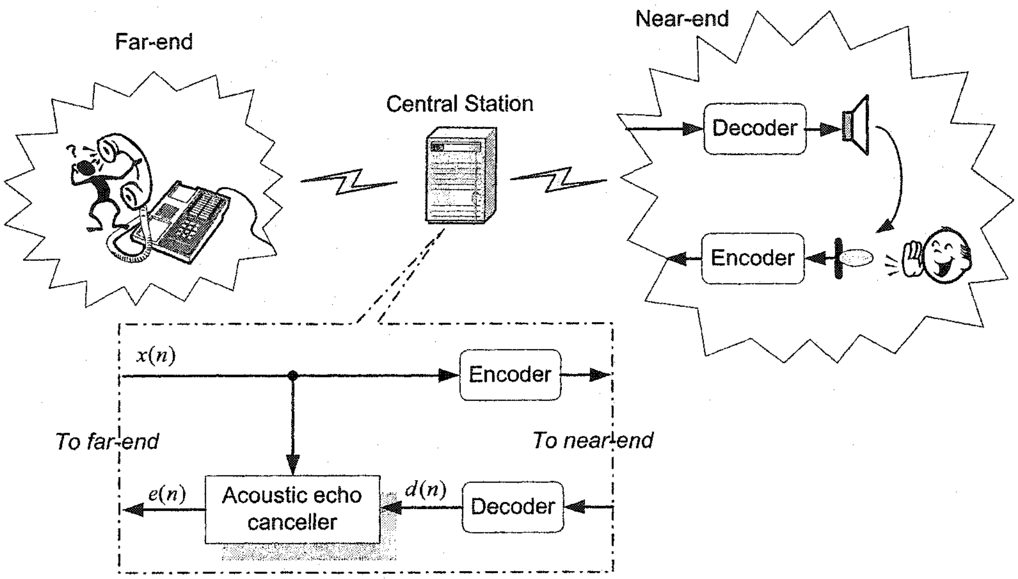

Fig. 1.5 The centralized AEC in the presence of codecs.

It is important to study centralized AEC systems because they have the following advantages:

- They have low system cost since each terminal does not need its own AEC device.
- One AEC device can be dynamically shared by numerous channels.
- They simplify the implementation of the user terminals where the computational capacity and physical size may be limited.

The echo path nonlinearities pose a new challenge to the design of centralized AEC systems, since the nonlinear echo path is very difficult to identify by conventional means, especially when modern vocoders are present.

1.4 Main contributions

This thesis studies the performance of AEC systems over nonlinear channels. The main focus is on cases where vocoders are present along the echo path, since this configuration poses an important and challenging problem in modern digital communication networks. The main contributions of this thesis are summarized as follows:

- A linear approximation approach is developed to model systems with vocoder and loudspeaker nonlinearities. With this approach, a nonlinear system is approximated by a piecewise linear system, with the proviso that the approximating linear system varies as a function of the input signal. The proposed model is verified by applying it to several popular adaptive filtering algorithms in nonlinear channels. Simulation results reveal that the performance of an adaptive filter in a nonlinear channel depends on its tracking capability: the faster an adaptive filter tracks the changes of a linear system, the lower the mean-squared-error (MSE) it achieves, which is consistent with the theoretical analysis.

- Two approaches are proposed to attenuate echo during double-talk (DT) periods. In conventional AEC, adaptive filter coefficients are often fixed during DT in order to prevent the adaptive filtering algorithm from diverging. Unfortunately, stopping adaptation in a nonlinear channel may produce a residual echo that has a higher power level than the original echo signal. The first approach proposed for echo attenuation exploits a pitch analysis technique to extract the pitch information from the residual echo, so that the power level of the residual echo can be largely reduced [26]. The second approach is a variable step-size adaptive cross-spectral algorithm [27]. It exploits the correlation between the far-end signal and the acoustic echo. This algorithm does not need DT detection and it keeps adapting during DT periods.

- A combined AEC system is presented that employs post-filtering to enhance acoustic echo suppression. It is shown that using linear adaptive filters alone in nonlinear channels cannot sufficiently suppress acoustic echoes. When combined with post-filtering, however, results demonstrate that the proposed AEC system achieves significantly better echo suppression in nonlinear channels [28, 29]. Furthermore, a perceptual model of the human ear is incorporated into the combined AEC system [30]. This allows the system to exploit the masking properties of the human ear in order to mitigate musical noise and near-end speech distortion caused by Wiener-type post-filters.

- An effective approach in reducing AEC's computational complexity by using subband adaptive filtering is presented. Reducing computational complexity is important since AEC systems must work in real-time. This work developed a practical method for the rapid design and prototyping of suitable filter banks. Using subband adaptive filters significantly reduces the computational cost of a combined AEC system while achieving desired echo attenuation [31].

1.5 Thesis organization

This thesis is organized as follows:

Chapter 2 investigates popular linear and nonlinear adaptive filters in AEC applications. The performance of these adaptive filters is analyzed.

Chapter 3 describes nonlinear channel characteristics where vocoder and loudspeaker nonlinearities are taken into account. The effects of these nonlinearities on conventional AEC systems (which mainly employ linear adaptive filters) are studied.

Chapter 4 details a robust adaptive filtering algorithm called the variable step-size adaptive cross-spectral algorithm. The derivation of the algorithm is given, and its performance in various situations (e.g., during initialization and double-talk) is investigated.

Chapter 5 explores the use of post-filtering techniques in AEC systems. Various post-filters, namely: Wiener-type post-filter, the spectral subtraction, the subspace method and the pitch extraction approach, are presented. Also their performance is analyzed and evaluated in the nonlinear channels.

Chapter 6 presents subband adaptive filtering algorithms used to reduce AEC's computational complexity. A practical method of designing simplified subband adaptive filters based on DFT filter banks is described. As a result, a combined AEC system with significant computational power reduction is presented.

Chapter 7 incorporates the perceptual model of the human ear into the combined AEC system. The concept of auditory masking and a few popular masking models are introduced. The use of these masking models in AEC is presented, followed by an examination of the performance of this AEC system in nonlinear channels.

Chapter 8 summarizes this thesis and discusses future research directions.

Chapter 2

Fundamentals of AEC: adaptive filtering

AEC is a problem of system identification, where the unknown system (i.e., the echo path) is usually time-variant and in some cases nonlinear. In a conventional AEC system, the echo path is treated as a slowly time-varying linear system where the nonlinearities along the echo path are negligible. For example, the loudspeaker is approximated as a linear device when its volume is sufficiently low. Under such assumptions, linear adaptive filters are used to identify the echo path.

In the cases where the nonlinearities of the echo path cannot be neglected (e.g., when vocoders are cascaded along the echo path), a more complex system model is needed in order to identify the nonlinear echo path. Intuitively, a nonlinear adaptive filter could identify a corresponding nonlinear echo path. In practice, some popular nonlinear adaptive filters are indeed employed in certain AEC applications.

2.1 Introduction

System identification is an experimental approach of modelling a process or a plant, as is illustrated in Figure 2.1. When building an appropriate model for an unknown system, the model need not be a true and accurate description of the system; it only has to be suitable for its intended purpose [32]. In AEC, the unknown system is the echo path, and an adaptive filter is employed to model it. This system identification process is done iteratively with adaptive control algorithms. Let $x(n)$, $y(n)$, $\hat{y}(n)$ and $s(n)$ respectively denote the input signal, the output signal of the unknown system, the output signal of the adaptive filter and the disturbance signal. The summation of $y(n)$ and $s(n)$ results in $d(n)$. The adaptive control algorithm seeks to minimize in some sense the difference signal between $\hat{y}(n)$ and $d(n)$. If the algorithm converges and the output $\hat{y}(n)$ of the adaptive filter is very close to the output $y(n)$ of the plant (i.e., the unknown echo path), then the unknown echo path is identified.

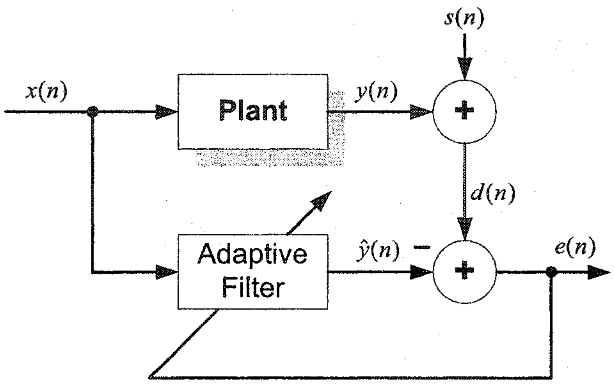

Fig. 2.1 The system identification using an adaptive filter.

Decades of research in adaptive filter theory have produced an enormous number of adaptive filtering algorithms [33, 34]. These algorithms can be categorized as linear or nonlinear. Each category includes a large number of algorithms, but only a few of them are suitable for AEC due to computational complexity, convergence rate, and tracking capability issues. Many popular adaptive filtering algorithms have been studied in white noise excitation. However, when excited by speech signals, the performance of these algorithms is degraded.

AEC has been extensively studied with the focus on identifying the acoustic echo path [8, 35]. Typically, the impulse response of an LEM system is represented by a finite all-zero model because the signal energy becomes so small that it can be neglected after a certain number of reflections [36]. Infinite impulse response (IIR) filter models or autoregressive-moving-average (ARMA) models have also been used [37, 38]. However, IIR models do not appear to offer any advantage over FIR ones in AEC [14].

To obtain a room impulse response, a loudspeaker and a microphone were connected to a PC via a sound card. The loudspeaker played a white noise signal while the microphone picked up the reflected sounds (including the direct sound). Both signals were recorded by a PC for post-processing, in which an adaptive filter was run with the loudspeaker signal as excitation and the microphone signal as the reference. After the steady-state of the adaptive filter was reached, where a very small step-size was used, the filter coefficients were regarded as the room impulse response. For illustration purposes, Figure 2.2 shows an impulse response of a typical office room based on an FIR filter model.

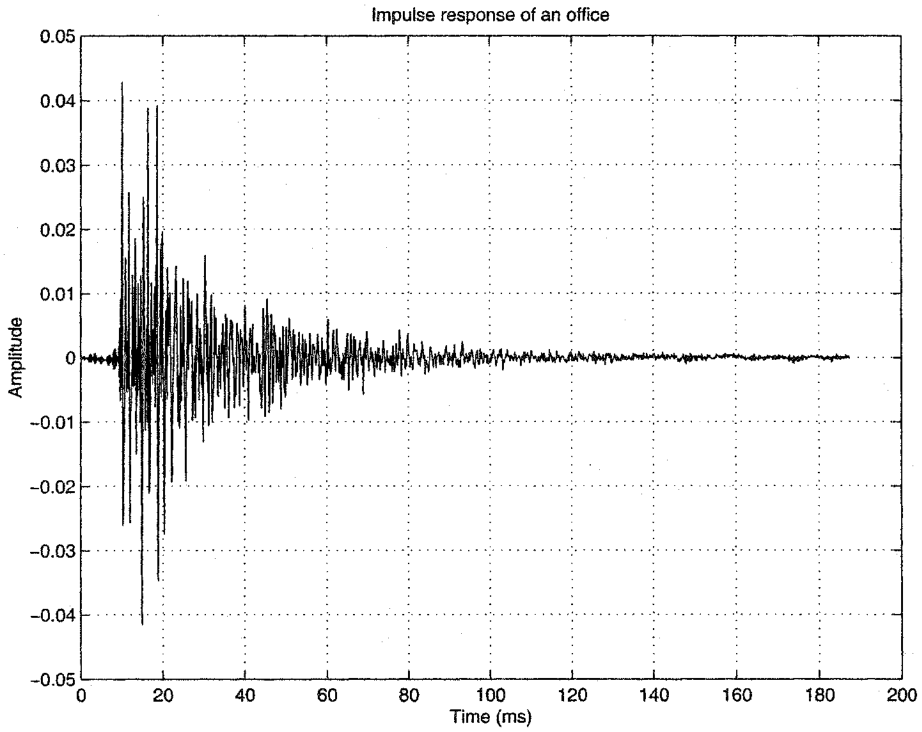

Fig. 2.2 The acoustic impulse response of a typical office room.

In the following sections, linear adaptive filters are discussed first. They are used when the acoustic echo path can be modelled as a linear system. Next, nonlinear adaptive filtering algorithms are investigated. They may be needed when the nonlinearities in the echo path introduced by nonlinear devices such as loudspeakers and vocoders cannot be neglected.

2.2 Linear adaptive filters

A linear adaptive transversal filter is shown in Figure 2.3, where z^{-1} represents a unit delay. It is based on an FIR filter structure. The latter is inherently stable, as opposed to IIR filters, as is thus commonly used in practice. In order to discuss several popular linear adaptive filtering algorithms suitable for AEC, it is necessary to first define the far-end signal vector $\mathbf{x}_n$ and the adaptive filter coefficient vector $\mathbf{w}_n$:

$$\mathbf{x}_n = [x(n), x(n-1), \cdots, x(n-N+1)]^T \tag{2.1}$$

$$\mathbf{w}_n = [w_0(n), w_1(n), \cdots, w_{N-1}(n)]^T, \tag{2.2}$$

where N is the length of the adaptive filter, n is the index of the discrete time and T denotes the transposition operator.

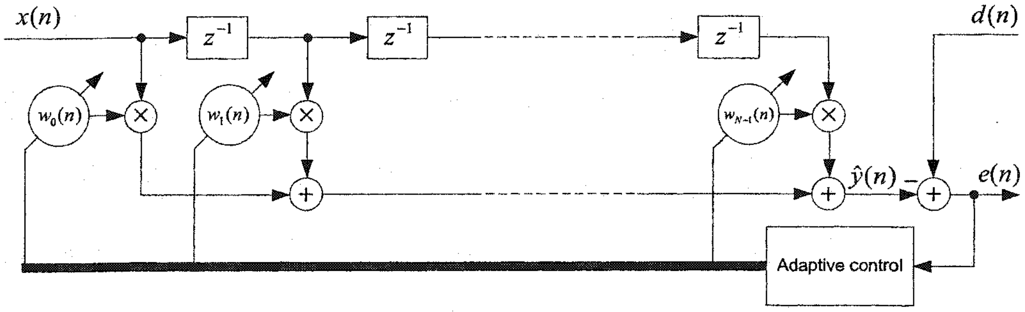

Fig. 2.3 The structure of a FIR adaptive filter.

2.2.1 The normalized least-mean-square (NLMS) algorithm

The normalized least-mean-square (NLMS) algorithm [33] is shown in Algorithm 1. The step-size μ controls the convergence behaviour of the algorithm: the larger the value of μ, the faster the algorithm converges, but this would also cause a greater misadjustment (i.e., larger residual error signal $e(n)$) in steady-state. For the algorithm to be stable, μ must be chosen from $0 < \mu < 2$. The small positive constant δ is introduced in order to prevent the denominator from being too small when the power of the input signal $x(n)$ is very low.

NLMS is one of the most popular algorithms for AEC due to its simplicity of implementation, low computational complexity, and robust behaviour. The computational

Algorithm 1 The normalized least-mean-square (NLMS) algorithm

```
Initialization:
1: $\mathbf{w}_1 = \mathbf{0}$
Recursion:
2: for $n = 1, 2, \ldots$ do
3:     $e(n) = d(n) - \mathbf{w}_n^T \mathbf{x}_n$
4:     $\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{\mathbf{x}_n^T \mathbf{x}_n + \delta} e(n) \mathbf{x}_n$
5: end for
```

complexity of NLMS is $O(2N)$ operations per sample (OPS), where one operation is defined as one real multiplication plus one real addition. The step-size in line 4 of Algorithm 1 is normalized by the input signal power; so the algorithm's convergence behaviour is independent of any variations of the input signal power. However, the algorithm has a slow convergence rate, especially for coloured excitation signals such as speech because of the large eigenvalue spread associated to coloured signals.

2.2.2 The recursive least-squares (RLS) algorithm

The RLS algorithm is relatively unaffected by eigenvalue disparity [13]. RLS is based on minimizing a weighted squared error sum, and thus it can be derived by minimizing the the cost function $J_n(\mathbf{w}_n)$ given below with respect to filter coefficients $\mathbf{w}_n$.

$$J_n(\mathbf{w}_n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2 = \sum_{i=1}^{n} \lambda^{n-i} |d(i) - \mathbf{w}_n{}^T \mathbf{x}_i|^2 \tag{2.3}$$

where λ $(0 < \lambda \leq 1)$ is a weighting factor which gives more weight to the most recent errors. This is useful in non-stationary environments where recent changes in the signal characteristics make the inclusion of old data less appropriate. λ is also known as the forgetting factor.

Differentiating (2.3) with respect to $\mathbf{w}_n$ and equating to zero leads to a set of normal equations:

$$\mathbf{R}_n \mathbf{w}_n = \mathbf{g}_n, \tag{2.4}$$

where the autocorrelation matrix $\mathbf{R}_n$ and cross-correlation vector $\mathbf{g_n}$ are defined as follows:

$$\mathbf{R}_n = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}_i\mathbf{x}_i^T, \tag{2.5}$$

$$\mathbf{g}_n = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}_i d(n). \tag{2.6}$$

Note that $\mathbf{R}_n$ in (2.5) can be updated recursively as

$$\mathbf{R}_n = \lambda\mathbf{R}_{n-1} + \mathbf{x}_i\mathbf{x}_i^T \tag{2.7}$$

After mathematical manipulations and applying the well-known matrix inversion lemma (Sherman-Morrison-Woodbury formula) [39] to (2.7), the RLS algorithm in Algorithm 2 is obtained, where δ is a small positive constant. The value of δ should be chosen so that the ratio of δ and the variance of the data sample $x(n)$ is 0.01 [40].

Algorithm 2 The recursive least-squares (RLS) algorithm

```
Initialization:
  $\mathbf{w}_0 = \mathbf{0}$
  $\mathbf{R}_0^{-1} = \delta^{-1}\mathbf{I}$
Recursion:
  for $n = 1, 2, \ldots$ do
      $\mathbf{\Gamma}_n = \mathbf{R}_{n-1}^{-1}\mathbf{x}_n$
      $\alpha(n) = \frac{1}{\lambda + \mathbf{x}_n^T\mathbf{\Gamma}_n}$
      $e(n) = d(n) - \mathbf{w}_{n-1}^T\mathbf{x}_n$
      $\mathbf{w}_n = \mathbf{w}_{n-1} + \alpha(n)\, e(n)\, \mathbf{\Gamma}_n$
      $\mathbf{R}_n^{-1} = \frac{1}{\lambda}\left[\mathbf{R}_{n-1}^{-1} - \alpha(n)\mathbf{\Gamma}_n\mathbf{\Gamma}_n^T\right]$
  end for
```

Compared to classical descent-based algorithms, such as NLMS, the RLS algorithm is better suited for coloured signal excitations. The RLS algorithm has a much higher convergence rate and a lower steady-state mean-squared error than NLMS. However, a major disadvantage of RLS is its expensive computational load of $O(N^2)$ OPS. Another issue is RLS's numerical sensitivity. In recent years, numerous variations of the RLS algorithm have been developed [15] to lessen its computation burden and/or improve its numerical robustness [41]. As a result, fast versions of RLS, such as the fast a priori error sequential technique (FAEST) [42] and the fast transversal filter (FTF) [43], have a complexity of about $7L$ OPS. Although this is still too high for AEC applications and the fast algorithms suffer from numerical stability problems [44], RLS is still important for this study because of its fast initial convergence.

The forgetting factor λ must be chosen carefully. Usually $\lambda = 1$ is appropriate for stationary data, while $0.95 < \lambda < 0.9995$ has been suggested for other data [45]. The algorithm may become unstable if the value of λ is too small. In practice, it is not clear how to choose a specific value of λ because this depends on the adaptive filter length and the excitation signal.

2.2.3 The affine projection (AP) algorithm

The affine projection (AP) algorithm [21] is very attractive for AEC. It has a faster convergence rate than NLMS for speech excitation, and it is stable and less complex than RLS. The AP algorithm is a generalization of the NLMS algorithm. In the latter, each filter coefficient vector update can be viewed as a one-dimensional affine projection. The AP algorithm generalizes this concept by allowing the projections to be made in multiple dimensions. As the projection dimension increases, so does the convergence speed of the coefficient vector.

In the AP algorithm, the excitation signal matrix $\mathbf{X}_n$, microphone signal vector $\mathbf{d}_n$, and error signal vector $\mathbf{e}_n$ are defined as

$$\mathbf{X}_n = [\mathbf{x}_n, \mathbf{x}_{n-1}, \cdots, \mathbf{x}_{n-p+1}]^T \tag{2.8}$$

$$\mathbf{d}_n = [d(n), d(n-1), \cdots, d(n-p+1)]^T \tag{2.9}$$

$$\mathbf{e}_n = [e(n), e(n-1), \cdots, e(n-p+1)]^T, \tag{2.10}$$

where the input signal vector $\mathbf{x}_n$ is the same as before in (2.1). The relaxed and regularized form of the AP algorithm is given in Algorithm 3.

The vector $\boldsymbol{\varepsilon}_n$ in Algorithm 3 is called the normalized error (or residual echo) vector.

Algorithm 3 The affine projection (AP) algorithm

```
Initialization:
  $\mathbf{w}_1 = \mathbf{0}$
Recursion:
  for $n = 1, 2, \ldots$ do
      $\mathbf{e}_n = \mathbf{d}_n - \mathbf{X}_n \mathbf{w}_n$
      $\boldsymbol{\varepsilon}_n = [\mathbf{X}_n \mathbf{X}_n^T + \delta \mathbf{I}]^{-1} \mathbf{e}_n$
      $\mathbf{w}_{n+1} = \mathbf{w}_n + \mu \mathbf{X}_n^T \boldsymbol{\varepsilon}_n$
  end for
```

Since $\mathbf{X}_n\mathbf{X}_n^T$ may have eigenvalues close to zero (thus creating problems when computing its inverse), a small diagonal matrix $\delta\mathbf{I}$ is added to $\mathbf{X}_n\mathbf{X}_n^T$ before inversion. δ is a small positive constant called the regularization parameter for the excitation signal autocorrelation matrix $\mathbf{X}_n\mathbf{X}_n^T$. δ should be as small as possible while still allowing a well-behaved inverse. The constant $\mu > 0$ sets the step size and is sometimes called the relaxation factor in AP. As in NLMS, AP is stable for $0 < \mu < 2$. The parameter p, called the projection order, defines the dimension of the vectors $\mathbf{d}_n, \mathbf{e}_n, \boldsymbol{\varepsilon}_n$, and the matrix $\mathbf{X}_n$. If p is set to 1, then the AP algorithm reduces to the NLMS algorithm. If p is set to 2, then the AP algorithm reduces to the decorrelation NLMS algorithm [46], which will be discussed in the next section.

There is little difference between the convergence behaviour of NLMS and AP with white noise excitation. However, when they are driven by coloured excitation signals, AP has a much faster convergence rate than NLMS [11], especially as the projection order p is increased.

The computational complexity of AP is $2pN + 7p^2$ OPS. This complexity is significantly reduced in variants of the AP algorithm, such as the block exact fast AP algorithm [47] and the fast affine projection (FAP) algorithm [22]. The computational complexity of FAP is $2N + 14p$ or $2N + 20p$, depending on whether $\mu = 1$ or $\mu < 1$. The fast versions of AP have lately received considerable attentions for AEC applications [48, 11].

2.2.4 The decorrelation NLMS algorithm

In order to overcome the major drawback of NLMS, i.e., the slow convergence rate for speech signal excitation, many decorrelation algorithms have been proposed. The main idea of these algorithms is to pre-whiten speech signals before feeding them to the NLMS algorithm. These algorithms can be classified into two categories: the first is represented by the second order AP algorithm, although algorithms in this category may have been derived from different viewpoints [46, 49]; the second is represented by the conventional NLMS associated with decorrelation filters. This section focuses on the second category since the first has been addressed in Section 2.2.3.

The linear predictive coding (LPC) filter given below can be used as a decorrelation filter:

$$H(z) = 1 - \sum_{i=1}^{M} a_i z^i. \tag{2.11}$$

The LPC filter, represented by its coefficient vector $\mathbf{a}_M = [a_1, a_2, \ldots, a_M]^T$, significantly whitens speech signals by exploiting the properties of speech. It removes short-term redundancies of speech signals, resulting in an output signal with a relatively flat spectral envelope [50].

For simplicity, most LPC work employs an all-pole model (i.e., AR model). Two approaches are often used to obtain the coefficient vector of the LPC filter $\mathbf{a}_M$. The classic least-squares method computes $\mathbf{a}_M$ by minimizing the mean energy in the error signal over a frame of speech data $x(n)$, while the lattice approach involves both a forward and a backward prediction [51]. In practice, the Levinson-Durbin algorithm [33] is used to estimate the LPC coefficients due to its computational efficiency. Although the prediction error decreases monotonically with the order of the LPC filter M, the 10 poles is appropriate for 8 kHz sampled speech, considering the compromise between spectral accuracy and the computational complexity [52]. The length of the window theoretically can be as short as M, but spectral accuracy usually increases with larger length. The typical length corresponds to 10-30 ms.

A possible approach for decorrelation is shown in Figure 2.4, and the corresponding algorithm is given in Algorithm 4. This approach, referred to here as decorrelation scheme 1 (NLMS-DF1), employs an auxiliary loop for adaptation [53, 54]. The adaptive filter coefficients are copied into the echo cancellation filter at each iteration. Since the excitation

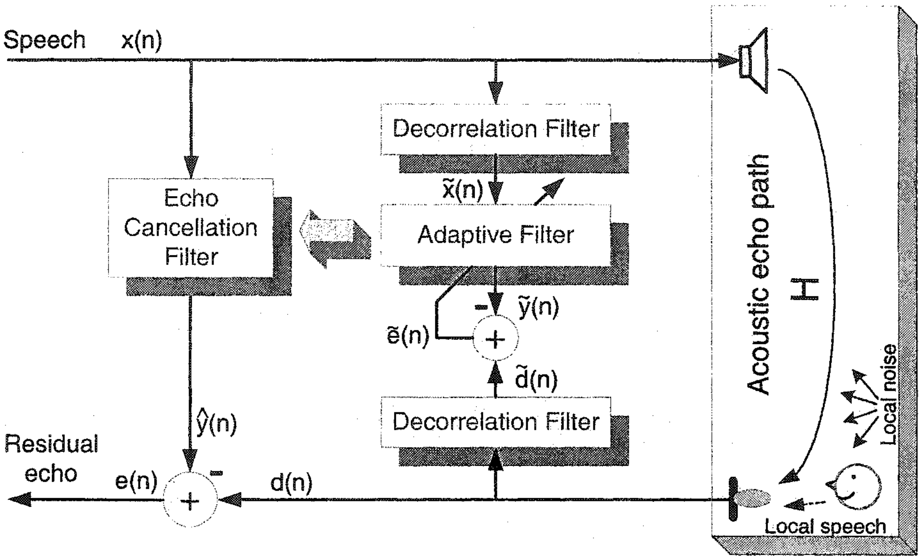

Fig. 2.4 Decorrelation scheme 1 – using an auxiliary loop for adaptation (NLMS-DF1).

signal is pre-whitened by a decorrelation filter, the adaptive filter has a faster convergence rate.

This approach has the shortcoming that the auxiliary path increases computational and memory requirements. The filtering operation of the echo canceller must be performed twice: first for the decorrelated signal, then for the original signal. Furthermore, extra buffers are needed to store the decorrelated excitation signals $\tilde{x}(n)$, $\tilde{d}(n)$ and $\tilde{e}(n)$.

A different approach of using decorrelation filters is depicted in Figure 2.5, and its algorithm is given in Algorithm 5. This is referred to here as decorrelation scheme 2 (NLMS-DF2). It employs an inverse decorrelation filter in order to recover the decorrelated signal, which is either the residual echo or the near-end speech. The inverse decorrelation filter is a recursive filter derived from the LPC filter coefficients. The inverse decorrelation filter is causal and stable because the LPC filter is minimum phase. Compared with NLMS-DF1, the NLMS-DF2 scheme has lower computational complexity and less memory requirement.

NLMS-DF2 also has several shortcomings. First, using a recursive filter as the inverse decorrelation filter may cause instability in fixed-point implementations. Second, the residual echo signal $e(n)$ in NLMS-DF2 differs from the residual echo signal that would result from a standard application of the NLMS, i.e., without the use of decorrelation filters.

Algorithm 4 The decorrelation NLMS algorithm scheme 1 (NLMS-DF1)

```
Initialization:
$\mathbf{w}_1 = \mathbf{0}$
Recursion:
for $n = 1, 2, \dots$ do
    $\mathbf{d}(n-1) = [d(n-1), d(n-2), \dots, d(n-M)]^T$
    $\mathbf{x}(n-1) = [x(n-1), x(n-2), \dots, x(n-M)]^T$
    $\tilde{d}(n) = d(n) - \mathbf{a}_M^T(n)\,\mathbf{d}(n-1)$
    $\tilde{x}(n) = x(n) - \mathbf{a}_M^T(n)\,\mathbf{x}(n-1)$
    $\tilde{\mathbf{x}}_n = [\tilde{x}(n), \tilde{x}(n-1), \cdots, \tilde{x}(n-L+1)]^T$
    $\tilde{e}(n) = \tilde{d}(n) - \mathbf{w}_n^T\tilde{\mathbf{x}}_n$
    $e(n) = d(n) - \mathbf{w}_n^T\mathbf{x}_n$
    $\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{\tilde{\mathbf{x}}_n^T\tilde{\mathbf{x}}_n}\tilde{e}(n)\,\tilde{\mathbf{x}}_n$
end for
```

To demonstrate this difference, decorrelated signals $\tilde{e}(n)$ and $\tilde{\mathbf{x}}_n$ are expressed in terms of original signals $d(n)$ and $\mathbf{x}_n$:

$$\begin{aligned} \tilde{e}(n) &= \tilde{d}(n) - \mathbf{w}_n^T\tilde{\mathbf{x}}_n, \\ &= d(n) - \sum_{i=1}^{M} a_i d(n-i) - \mathbf{w}_n^T\left(\mathbf{x}_n - \sum_{i=1}^{M} a_i \mathbf{x}_{n-i}\right) \\ &= e_0(n) - \sum_{i=1}^{M} a_i\left(d(n-i) - \mathbf{w}_n^T\mathbf{x}_{n-i}\right), \end{aligned} \tag{2.12}$$

where $e_0(n) = d(n) - \mathbf{w}_n^T\mathbf{x}_n$ is the residual echo signal at the output of standard NLMS. Now, let $\tilde{e}_0(n)$ denote the output of the LPC filter (with coefficient vector $\mathbf{a}_M(n)$) when excited by the input $e_0(n)$. That is, $\tilde{e}_0(n)$ is the prediction error associated with the residual

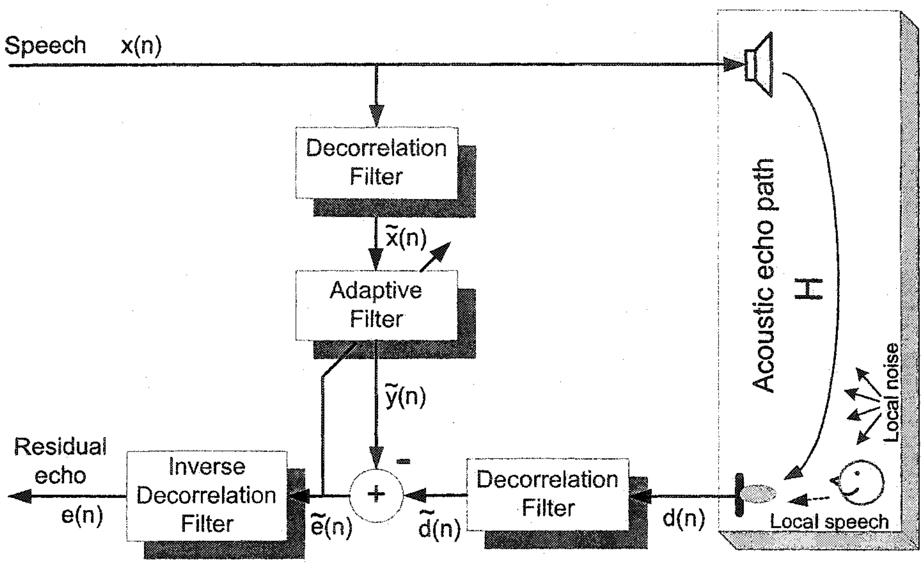

Fig. 2.5 Decorrelation scheme 2 – using an inverse decorrelation filter NLMS-DF2).

echo $e_0(n)$ in NLMS. According to the definition of the LPC filter, $\tilde{e}_0(n)$ is given by

$$\begin{aligned}\tilde{e}_0(n) &= e_0(n) - \sum_{i=1}^{M} a_i e_0(n-i) \\ &= e_0(n) - \sum_{i=1}^{M} a_i \left(d(n-i) - \mathbf{w}_{n-i}^T \mathbf{x}_{n-i}\right).\end{aligned} \tag{2.13}$$

Comparing (2.13) with (2.12), it is clear that $\tilde{e}_0(n)$ is not equal to $\tilde{e}(n)$ in the transient state where $\mathbf{w}_{n-i} \neq \mathbf{w}_n \, (i \geq 1)$. Thus the residual echo $e_0(n)$ cannot be recovered by passing $\tilde{e}(n)$ through an inverse decorrelation filter. In other words, the residual echo $e(n)$ in NLMS-DF2 is different from that in NLMS-DF1 (where $e(n) = e_0(n)$). Although this does not imply that NLMS-DF2 has a poorer performance than NLMS-DF1, it does pose difficulties in analyzing and interpreting NLMS-DF2.

Accurate speech prediction (such that the resulting prediction error signal is maximally whitened) is very important in decorrelation NLMS algorithms. Increasing the prediction order results in a more accurate speech estimate, and thus a whiter error signal at the output of the LPC filter. According to a common geometrical interpretation [21], the

Algorithm 5 The decorrelation NLMS algorithm scheme 2 (NLMS-DF2)

```
Initialization:
$\mathbf{w}_1 = \mathbf{0}$
Recursion:
for $n = 1, 2, \ldots$ do
    $\mathbf{d}(n-1) = [d(n-1), d(n-2), \ldots, d(n-M)]^T$
    $\mathbf{x}(n-1) = [x(n-1), x(n-2), \ldots, x(n-M)]^T$
    $\mathbf{e}(n-1) = [e(n-1), e(n-2), \ldots, e(n-M)]^T$
    $\tilde{d}(n) = d(n) - \mathbf{a}_M^T(n)\,\mathbf{d}(n-1)$
    $\tilde{x}(n) = x(n) - \mathbf{a}_M^T(n)\,\mathbf{x}(n-1)$
    $\tilde{\mathbf{x}}_n = [\tilde{x}(n), \tilde{x}(n-1), \cdots, \tilde{x}(n-L+1)]^T$
    $\tilde{e}(n) = \tilde{d}(n) - \mathbf{w}_n^T\tilde{\mathbf{x}}_n$
    $\mathbf{w}_{n+1} = \mathbf{w}_n + \frac{\mu}{\tilde{\mathbf{x}}_n^T\tilde{\mathbf{x}}_n}\tilde{e}(n)\,\tilde{\mathbf{x}}_n$
    $e(n) = \mathbf{a}_M^T(n)\,\mathbf{e}(n-1) + \tilde{e}(n)$
end for
```

decorrelated input signal $\tilde{\mathbf{x}}_n$ is orthogonal to a hyperplane in the L-dimensional Euclidean space defined by $\{\mathbf{w} : \ \mathbf{w} \in \mathbb{R}^L, \ \mathbf{w}^T\tilde{\mathbf{x}}_n = \tilde{d}(n)\}$. This hyperplane is formed by the adaptive filter coefficient vectors which give the output equal to the decorrelated echo signal $\tilde{d}(n)$. The convergence speed of the coefficient vector greatly depends on the angle between two successive hyperplanes. The fastest convergence rate occurs when the angle is $\pi/2$, which thus requires that the signals $\tilde{\mathbf{x}}_n$ and $\tilde{\mathbf{x}}_{n-1}$ be orthogonal (i.e., the decorrelated signals are ideally whitened). In order to achieve perfect whitening of the input speech signal, the prediction coefficients should be adaptive, and they should be updated as quickly as possible since speech is a non-stationary signal. However, this condition cannot be achieved for AEC, as explained in the following.

Let $d(n) = \mathbf{h}^T\mathbf{x}_n$, where $\mathbf{h}$ represents the impulse response of the acoustic echo path.

Applying the LPC filter (2.11) to the microphone signal $d(n)$, we have

$$\begin{aligned}\tilde{d}(n) &= d(n) - \sum_{i=1}^{M} a_i(n)d(n-i) \\ &= \mathbf{h}^T\mathbf{x}_n - \sum_{i=1}^{M} a_i(n)\mathbf{h}^T\mathbf{x}_{n-i} \\ &= \mathbf{h}^T\left(\mathbf{x}_n - \sum_{i=1}^{M} a_i(n)\mathbf{x}_{n-i}\right) \\ &= \mathbf{h}^T\tilde{\mathbf{x}}_n. \end{aligned} \tag{2.14}$$

The above relation is the basis of decorrelation NLMS algorithms. It is assumed that the prediction coefficients $a_i(n)$ $(i = 1, 2, \cdots, M)$ do not change over a duration comparable to the length of the excitation signal vector $\mathbf{x}_n$. This duration may be hundreds of milliseconds corresponding to thousands of filter taps at an 8 kHz sampling rate. If the coefficients $a_i(n)$ change over such a period of time, then $\tilde{\mathbf{x}}_n$ in (2.14) contains more than one set of $a_i(n)$, which makes the last equality in (2.14) untrue. This means that if the prediction coefficients are adapted too frequently, then using signals $\tilde{x}(n)$ and $\tilde{d}(n)$ instead of $x(n)$ and $d(n)$ will not lead to a correct identification of the unknown system $\mathbf{h}$. Thus, the prediction coefficients must not change too rapidly compared to the impulse response duration. Unfortunately, this is not the case in AEC applications where impulse responses are long (especially in office rooms). Consequently, the accuracy of the predictions is limited.

For these reasons, using a higher order prediction does not lead to increased performance of decorrelation NLMS algorithms. In practice, using a first order LPC filter is often sufficient to achieve a significantly faster convergence rate than a standard NLMS algorithm. A first order decorrelation filter has very low computational complexity, especially a fixed decorrelation filter (a fixed decorrelation filter can exploit the long term characteristic of speech and has a first order autocorrelation coefficient of about 0.9). Using a fixed first-order LPC filter with an adaptive filter of length L, the computational complexity of NLMS-DF1 is about $3L$ OPS, and that of NLMS-DF2 is about $2L$ OPS.

2.3 Nonlinear adaptive filters in AEC

This section investigates some nonlinear adaptive filtering algorithms used for nonlinear echo paths in AEC. Unlike linear systems, which are completely characterized by their unit impulse response, nonlinear systems cannot be characterized in such an unified way. Therefore each nonlinear adaptive algorithm uses its own fitting model. Nonlinear adaptive filters can compensate the saturation nonlinearities caused by loudspeakers [55, 56]. So far, only two types of nonlinear adaptive filters have been used in AEC: polynomial adaptive filters [57, 58] and neural networks [59, 60]. In practice, serious drawbacks of these nonlinear adaptive algorithms limit their applications, even where the nonlinear mechanism is clear [61]. The major shortcomings of nonlinear adaptive algorithms are: high computational complexity, slow convergence speed due to a large number of adaptive filter coefficients, multiple local minima that may halt the adaptation of the algorithm, and numerical sensitivity which requires an extremely careful selection of algorithm parameters in order to avoid divergence.

2.3.1 Volterra filters

Volterra filters belong to the polynomial filter category [62]. Let $x(n)$ represent the input signal and $y(n)$ represent the output signal of a nonlinear system. Then the Volterra series expansion for $y(n)$ using $x(n)$ is given by [34]

$$\begin{aligned} y(n) &= w_0 + \sum_{m_1=0}^{\infty} w_1(m_1)x(n-m_1) \\ &\quad + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} w_2(m_1,m_2)x(n-m_1)x(n-m_2) \\ &\quad + \cdots + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\cdots\sum_{m_p=0}^{\infty} w_p(m_1,m_2,\cdots,m_p) \\ &\quad \cdot x(n-m_1)x(n-m_2)\cdots x(n-m_p) + \cdots \end{aligned} \tag{2.15}$$

where $w_p(m_1, m_2, \cdots, m_p)$ is called the p-th order Volterra kernel of the system. Without loss of generality, one can assume that Volterra kernels are symmetric, meaning the kernels are the same for every permutation of indices (e.g., $w(m_1, m_2) = w(m_2, m_1)$). Since the Volterra series expansion is an infinite series (similar to the Taylor series expansion but with memory), one must work with a truncated form in practice. A Volterra filter's computational complexity increases exponentially with its order p; therefore most applications employ low-order models. The second-order Volterra filter is most commonly used in practice [63, 64].

Adaptive schemes of Volterra filters use linear adaptive filtering criteria such as least-squares and stochastic gradient methods [65, 66]. This work focuses only on the NLMS adaptive Volterra filter [67] because of its simplicity and stability. The NLMS adaptive Volterra filter is obtained through a procedure similar to that used in NLMS, with w_0 neglected (w_0 can often be estimated independently of the adaptive filter structure). To simplify the presentation, the following notations are introduced,

$$\mathbf{W}_n = \left[\mathbf{w}_{n,1}^T \,|\, \mathbf{w}_{n,2}^T \,|\, \cdots\right]^T \tag{2.16}$$

$$\mathbf{w}_{n,1} = [w_{n,1}(0), w_{n,1}(1), \cdots, w_{n,1}(L-1)]^T \tag{2.17}$$

$$\mathbf{w}_{n,2} = [w_{n,2}(0,0), w_{n,2}(0,1), \cdots, w_{n,2}(L-1, L-1)]^T \tag{2.18}$$

$$\cdots$$

and

$$\mathbf{X}_n = \left[\mathbf{x}_{n,1}^T \,|\, \mathbf{x}_{n,2}^T \,|\, \cdots\right]^T \tag{2.19}$$

$$\mathbf{x}_{n,1} = [x(n), x(n-1), \cdots, x(n-L+1)]^T \tag{2.20}$$

$$\mathbf{x}_{n,2} = [x(n)x(n), x(n)x(n-1), \cdots, x(n-L+1)x(n-L+1)]^T \tag{2.21}$$

$$\cdots$$

$\mathbf{W}_n$ is the filter weight vector and $\mathbf{X}_n$ is a modified input signal vector containing all possible cross-products of the input samples. The NLMS adaptive Volterra algorithm, shown as Algorithm 6, is then developed in a way similar to the linear NLMS. Accordingly, the stability region of its step-size μ is the same as in the linear case, i.e., $0 < \mu < 2$.

The adaptive Volterra filter has computational complexity $O(L^p)$, where L is the length of the input signal vector and p is the order of the filter. Using multirate signal processing methods can reduce this computational burden, but certain constraints must be imposed and the filter order is limited [68]. In AEC applications, the filter length L typically varies from several hundred to several thousand taps, so the computational complexity becomes

Algorithm 6 The NLMS adaptive Volterra algorithm

```
Initialization:
    $\mathbf{W}_1 = \mathbf{0}$
Recursion:
    for $n = 1, 2, \ldots$ do
        $e(n) = d(n) - \mathbf{W}_n^T \mathbf{X}_n$
        $\mathbf{W}_{n+1} = \mathbf{W}_n + \frac{\mu}{\mathbf{X}_n^T \mathbf{X}_n} e(n) \mathbf{X}_n$
    end for
```

unmanageable even for a second-order adaptive Volterra filter. In order to make adaptive Volterra filters suitable for AEC, approximations are necessary as discussed below.

From (2.16) - (2.21), it can be seen that a Volterra filter has a linear part ($\mathbf{w}_{n,1}$ and $\mathbf{x}_{n,1}$) and a nonlinear part (the high-order signal cross-terms). Therefore, an adaptive Volterra filter can be implemented as a linear adaptive filter connected in parallel with an adaptive high-order kernel. Since adaptive Volterra filters are known to be effective only in cases of weak nonlinearities [69], the linear component plays a more important role than the nonlinear one. However, the nonlinear component is responsible for most of the computational burden. This suggests that modifying the nonlinear component with approximations can significantly reduce the filter's computational complexity.

A common approximation uses the shortest possible memory length in the nonlinear component, since the correlation between the signal samples decreases with the lag. This leads to a modified second-order Volterra representation with a much shorter memory length for the second-order terms [70]:

$$\begin{aligned} y(n) &= \sum_{m_1=0}^{L-1} w_1(m_1)x(n-m_1) \\ &+ \sum_{m_1=\Delta}^{L_2+\Delta-1} \sum_{m_2=m_1}^{L_2+\Delta-1} w_2(m_1,m_2)x(n-m_1)x(n-m_2), \end{aligned} \tag{2.22}$$

where $L \gg L_2$, and Δ is an adjustable delay introduced in order to focus on tap locations where most nonlinear effects occur.

This approach achieves a few decibels gain in echo-return-loss-enhancement (ERLE) for the compensation of loudspeaker nonlinearities, where white Gaussian noise is used as the excitation signal [70, 58]. However, it is important to note that the higher-order cross-terms in a Volterra adaptive filter (or even in a truncated second order approximate model) introduce valleys in the error surface $J(\mathbf{w}) = \mathrm{E}[e^2(n)]$, so the error surface has multiple local minima. In practice, it is difficult to select an initial filter coefficient vector that guarantees convergence to the global minimum.

2.3.2 Neural networks

In the past two decades, there has been an explosive development of neural networks for various applications [71]. Interest in neural networks for signal processing is motivated by their following properties: nonlinearity, weak statistical assumptions, and learning capability [72].

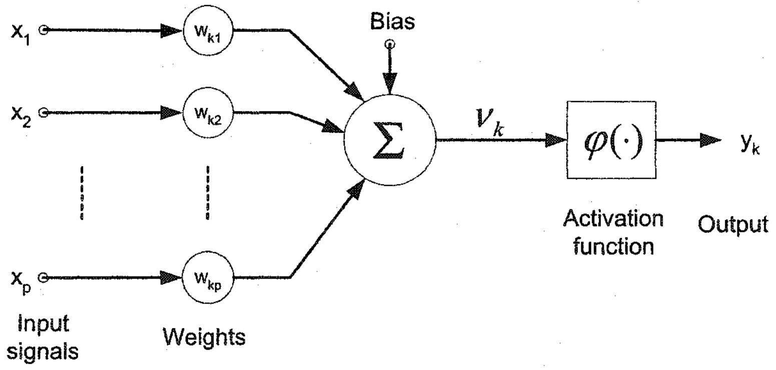

Fig. 2.6 Nonlinear model of a neuron (k–th neuron).

A neural network is an interconnection of nonlinear processing units called neurons. These neurons are connected through a set of connection weights, also called synaptic weights. Figure 2.6 shows the most commonly used model of a neuron. The model consists of a linear combiner followed by a nonlinear activation function $\varphi(\cdot)$. The exact nature of the neuron activation function depends on the neural network model. Two of the most popular activation functions are the nonsymmetric logistic function

$$\varphi(\nu) = \frac{1}{1 + e^{-\beta\nu}}, \tag{2.23}$$

and the symmetric hyperbolic tangent function

$$\varphi(\nu) = \tanh(\frac{\beta\nu}{2}) = \frac{1 - e^{-\beta\nu}}{1 + e^{-\beta\nu}}, \tag{2.24}$$

where β is a positive parameter which determines the slope of the activation function.

There are different structures for neural networks. An important structure for dynamic back-propagation (BP) learning consists of a bank of linear filters feeding a static neural network. The linear filters manage the temporal dependence of the input signal, while the static neural network handles nonlinear processing. A simple case is the time-lagged feedforward network (TLFN) that has a tapped-delay line followed by a multilayer perceptron (MLP) which is trained by using the BP algorithm [72], as illustrated in Figure 2.7. It is called a feedforward network in the sense that the input signals produce a response at the output of the network by propagating in the forward direction only, i.e., there is no feedback in the network. The BP algorithm may be viewed as a generalized form of the well-known LMS algorithm. The MLP consists of an input layer, one or more layers of hidden neurons, and an output layer.

In theory, a nonlinear input-output mapping can be approximated to any desired degree of accuracy by a multilayer perceptron with a single hidden layer, provided that the mapping is continuously differentiable and the hidden layer has enough processing units [73]. For practical reasons, however, one may use a multilayer perceptron with two or more hidden layers, depending on the complexity of the learning task.

Recent research on using neural networks in AEC has shown only limited benefits. For example, an adaptive filter has been proposed consisting of a multi-layer neural network in cascade with a linear FIR filter [59]. When compared with a classical linear adaptive filter, this design only achieved modest gains of a few decibels when there are loudspeaker nonlinearities. Furthermore, only white noise was used as the excitation signal in the simulation, not speech signals. Another similar design uses a two-layer neural network for the nonlinear part of the filter [60], where the nodes' activation functions are hyperbolic tangent functions. White noise is also used as the excitation signal, and results showed limited gains.

Like other nonlinear adaptive filtering algorithms, neural networks cannot guarantee convergence to the global optimum [74]. Furthermore, neural networks are not suitable for real-time systems since they have high computational complexity and slow convergence

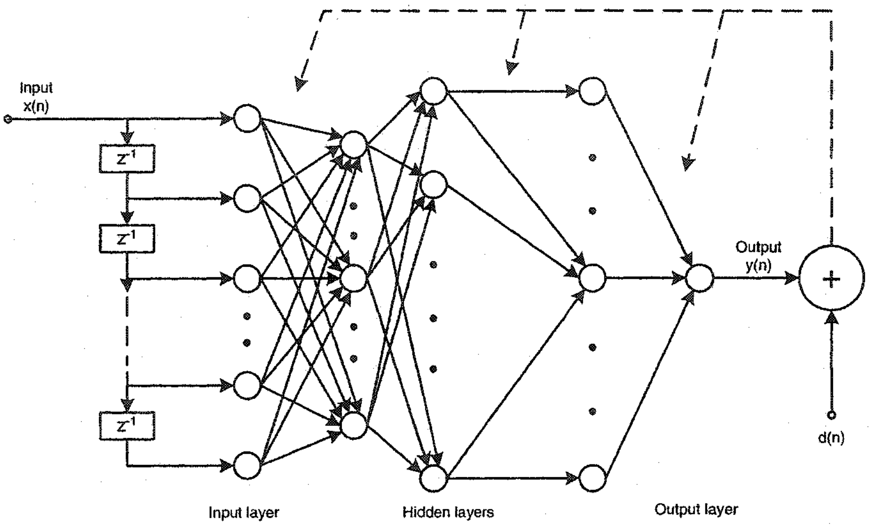

Fig. 2.7 Time-lagged feedforward network (TLFN).

speed. Speech signals also represent a major challenge for neural networks since they can make neural network algorithms unstable. Besides all these difficulties, it is time-consuming to choose a proper neural network model and to develop the corresponding algorithm [75]. Neural networks may still be an option for AEC in the future since there is much to be explored. However, results published to date are not so promising.

Chapter 3

Effects of nonlinear channels

Centralized AEC systems place AEC devices in a central station or a base station instead of user terminals (see Figure 1.5). These systems are attractive in practice since they lower system costs and simplify user terminal implementation. A hands-free terminal (mainly consisting of a loudspeaker and a microphone) is connected to the central station or the base station via a digital link that uses a modern coding scheme, such as G.729 or GSM. However, current echo cancellation technology that mostly uses adaptive transversal FIR filters for echo path identification suffers from nonlinearities introduced by speech codecs and non-ideal loudspeakers. These nonlinearities cause speech signal distortions that cannot be ignored.

In this chapter, the effects of the main nonlinear components, i.e., the low bit-rate speech codec and the loudspeaker, on the conventional AEC are studied. These nonlinear devices significantly degrade the performance of the conventional AEC, resulting in a high residual echo. This greatly impairs the communication quality of the hands-free telephone used in the modern digital networks. To develop a good understanding of the degradation caused by these nonlinearities, a qualitative approach is used throughout this chapter, as a strict closed form of theoretical analysis is almost impossible to obtain due to the complex characteristics of the nonlinear devices.

This chapter is organized as follows. First, the nonlinear mechanism of an important speech codec, i.e., G.729, has been investigated, and a localized linear model is used to analyze the nonlinear character of the codec. Then the nonlinear effects on the conventional AEC that mainly employs a linear adaptive filter have been studied. The tracking capabilities of adaptive filters are also investigated because they are shown as an important aspect of an adaptive filter working in a nonlinear channel. Finally, the nonlinearities of a loudspeaker is studied.

3.1 Characteristics of speech codecs

3.1.1 Speech codecs

Digital coding of speech signals provides efficient and secure signal transmission and storage. Speech is transformed by an encoder into a sequence of bits which is then transmitted over a channel. At the receiver, the bits are converted back into an audible signal using a decoder that acts as an inverse of the encoder. This inverse is only approximate since some information may have been lost during coding (due to analog-to-digital conversions) and data transmission (due to noisy channels).

Speech coding strives to represent speech signals economically while allowing them to be reconstructed with minimal quality loss at the receiver. This is done by exploiting speech signal redundancies as well as the masking properties of auditory perceptions. For instance, natural speech (i.e., speech generated by the human vocal tract, as opposed to speech generated by computers) has the following well known characteristics [50]:

- In general, vocal tract shape (and thus speech spectrum) changes slowly over time relative to the sampling frequency.
- Vocal cords vibrate rapidly but the rate of change of the excitation frequency is slow.
- Successive pitch periods are virtually identical most of the time.
- The vocal tract spectrum varies slowly with frequency, and most of the speech energy is concentrated at low frequency.
- Speech sounds can be modelled as a periodic or noisy excitation passing through a vocal tract filter, and each sound can be represented with a few parameters.
- These parameters have nonuniform probability distributions.

In addition, human ears have the following perception limitations:

- The ear is relatively phase-insensitive.
- The high intensity position of the speech spectrum masks the low amplitude frequencies.

Speech coders can be classified as waveform coders and vocoders. Time-domain waveform coders take advantage of temporal waveform redundancies to allow data compression, while frequency-domain waveform coders exploit the nonuniform frequency distribution of speech information. Examples of waveform coders are PCM, log PCM (G.711) [76], and ADPCM (G.726) [77]. Vocoders are more complex systems that use speech production models. Such models separate speech information into two parts: one that estimates vocal tract shape and one that estimates vocal tract excitation. Popular vocoders include CS-ACELP (G.729) [24] and GSM [25]. In recent years, distinction between the two kinds of speech coders, i.e., waveform coders and vocoders, has become blurred with the design of hybrid systems that code both timing and spectral information. However, a distinction can still be made between waveform coders that reconstruct speech sample-by-sample and vocoders that exploit speech-specific models.

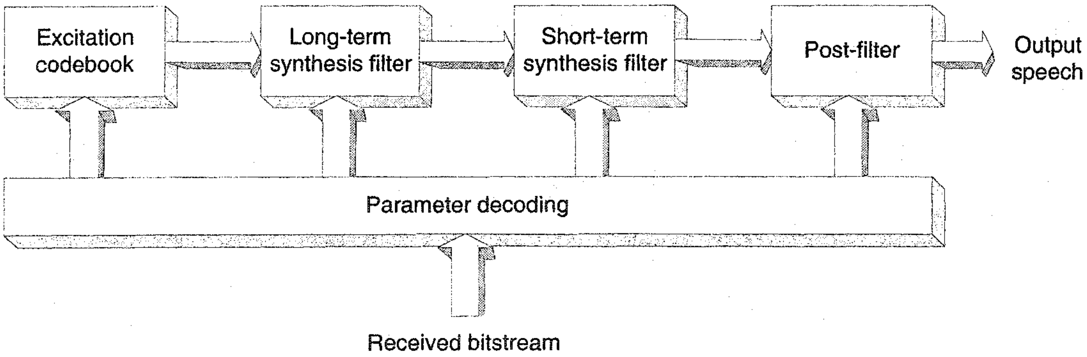

Fig. 3.1 Conceptual synthesis model of CELP vocoders.

This thesis focuses only on vocoders widely used in wireless networks. Specifically, the conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP) coder is chosen for investigation since it is a practical design that typifies the properties of many vocoders.

The CS-ACELP coder follows the 8kb/s ITU-T standard G.729 [24]. It is based on the code-excited linear-prediction (CELP) coding model shown in Figure 3.1. The difference between the standard CELP and CS-ACELP is that the latter does not store residual sample vectors in a vector quantization (VQ) codebook, but rather derives them directly from the transmitted index. This is done by using a simple algebraic transform on the index in order to produce the residual signal used to excite the pitch and the linear-prediction

(LP) synthesizer. In Figure 3.1, the long-term (or pitch synthesis) filter is implemented using the adaptive-codebook approach, while the short-term synthesis filter is a 10th order LP filter. The post-filter further enhances the reconstructed speech. CS-ACELP operates on speech frames of 10 ms corresponding to 80 samples at a sampling rate of 8000 samples per second. 80-bit parameters extracted from every 10 ms frame are used to reconstruct the speech signal. Among these 80 bits, 18 represent line spectrum pairs (LSP) while the others are used for the excitation codebook [24].

Vocoders usually introduce delay because a fixed number of consecutive samples (referred to as a frame) must be collected before the information parameters can be extracted from these samples. CS-ACELP encodes 10-ms frames with a look-ahead of 5 ms, resulting in a total algorithmic delay of 15 ms. Decoders do not introduce algorithmic delay, and their computational delay is negligible with today's DSP chips.

3.1.2 Distortion introduced by vocoders

As explained earlier, some information of the speech signal is irretrievably lost during the coding process. The more information is lost, the more distortion a coder produces. Signal-to-noise ratio (SNR) is a basic measure of the distortion introduced by coders. It is defined as the ratio of the average speech energy to the average energy of the error signal, usually expressed in decibels:

$$\mathrm{SNR}(dB) = 10\,\log_{10}\frac{\sigma_x^2}{\sigma_e^2} = 10\,\log_{10}\frac{E[x^2(n)]}{E[e^2(n)]} = 10\,\log_{10}\frac{\sum_n x^2(n)}{\sum_n e^2(n)}, \tag{3.1}$$

where $x(n)$ is the speech signal and $e(n)$ is the error signal. The operation $E[\cdot]$ represents expectation (averaging) over the entire speech utterance. The temporal variation of SNR can be studied by computing it once for every segment of 128 samples (corresponding to 16ms at an 8-kHz sampling rate). This is called the short-term SNR [78]. Short-term SNR is widely used in speech signals research; it is simply referred to as SNR in this thesis.

It is important to note that SNR is an objective measure of speech quality only for waveform coders. Vocoders distort the original speech signals to the extent that SNR becomes irrelevant as a perceptual measure, so maximizing SNR would not guarantee the best perceptual quality. However, SNR is still an useful tool for AEC systems, since the operation of adaptive filters is based on the waveforms, in either the time or frequency domain.

Figure 3.2 shows how the short-term SNR of reconstructed speech is computed. The original speech signal is first passed through the encoder and the decoder. The output is then compared with a delayed version of the original signal (the delay is introduced to synchronize the two signals), resulting in an error signal. The SNR is computed as the ratio of the average power (over a window of 128 samples) of the original signal and that of the error signal.

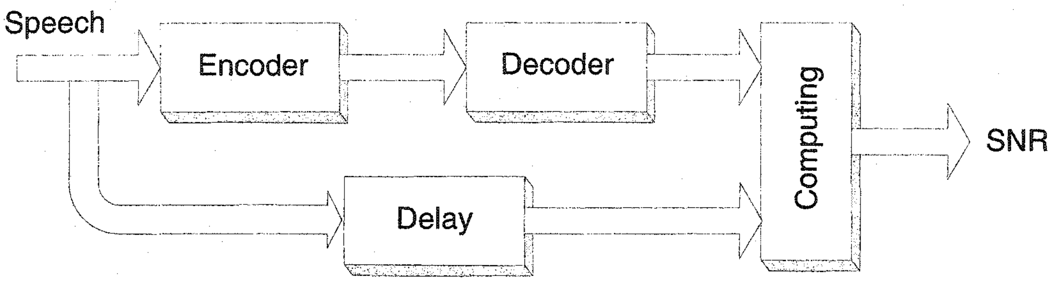

Fig. 3.2 Computational approach for short-term SNR measurement.

Figure 3.3 shows speech distortion (in terms of SNR) produced by the G.729 vocoder for the sample speech signal in Figure 3.4. Although the reconstructed signal has good speech quality when it is heard, the waveform distortion is significant. It can be seen that the maximal SNR is no larger than 10 dB. Comparing Figure 3.3 to Figure 3.4, it is observed that the SNR is approximately proportional to the speech energy. Although the example shown here is a male speech, similar results are obtained for other cases as well.

If white noise is used as the input to the encoder/decoder cascade in Figure 3.2, then vocoder G.729 produces a more severe distortion, as displayed in Figure 3.5. This is because the coder cannot correctly predict parameters (e.g., LPC and pitch coefficients) for white noise, which is an unpredictable waveform. Vocoder G.729 searches the codebook in an attempt to find the optimal parameter values that would allow a reconstruction with minimal distortion, but this effort results in only a few positive decibels of SNR. Since white noise is similar to unvoiced speech, it is obvious that the vocoder introduces more distortion for unvoiced speech than voiced speech.

As an important signal used for testing AEC systems, composite-source-signal (CSS) [79] is a composition of three different signal parts: voiced sound, pseudo-noise signal, and pause. When CSS is used as the input signal to the vocoders, severe distortion of the

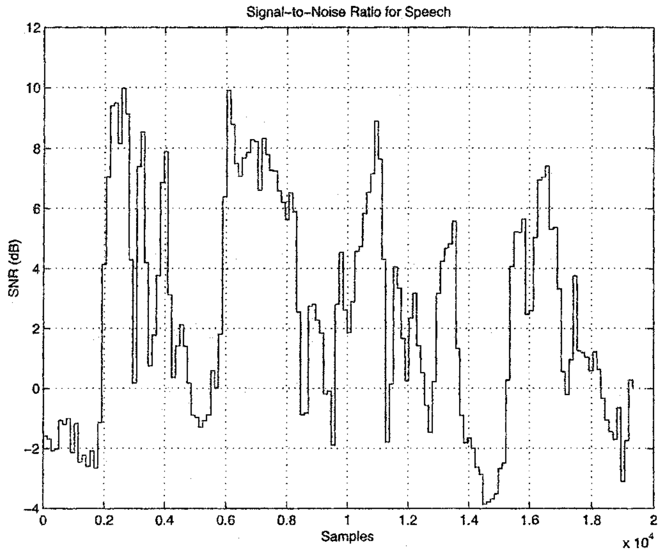

Fig. 3.3 SNR of the reconstructed speech signal from coder G.729.

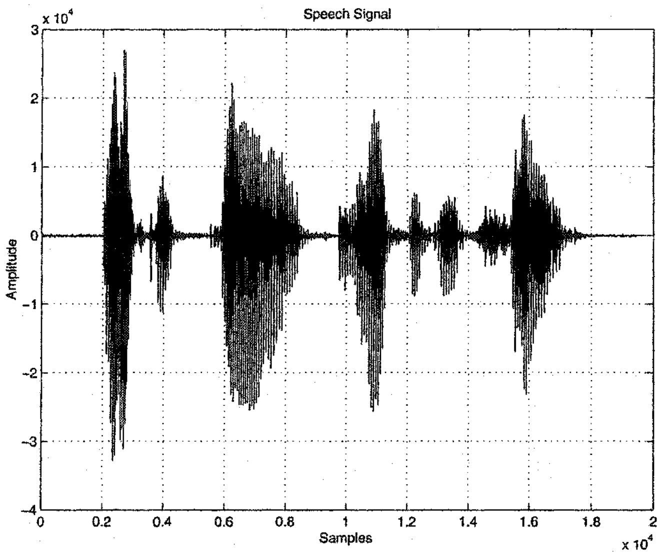

Fig. 3.4 Original speech signal (**male**: *Oak is strong and also gives shade.*)

reconstructed signal is observed at the output. This is because the CSS contains a white noise component. In order to discover the true characteristics of vocoders, real (as opposed

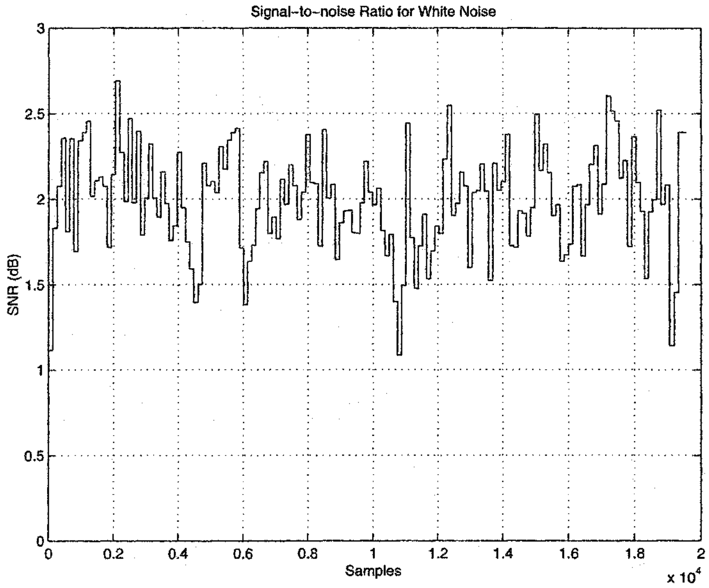

Fig. 3.5 SNR of reconstructed white noise from coder G.729.

to synthesized) speech signals are used throughout this work.

3.1.3 Analysis of vocoders

Both G.729 and GSM speech codecs have coding structures based on ACELP [80, 81]. ACELP is a VQ coding scheme. It searches the best combination of pulses and signs in order to obtain the best matched synthesis signal. The error between the original signal and the synthesized signal, called quantization error, is a nonlinear function of the input signal. Furthermore, a post-filter used to enhance the perceptual quality of the synthesized speech brings in more complex nonlinearities.

As a nonlinear dynamic system, the behaviour of the ACELP vocoder depends not only on the current input signal, but also on the state of the vocoder. The state completely describes the time history of the vocoder. It includes the influence of the past and "future" (look ahead) input and output samples on the current output of the vocoder. The output of the vocoder at a certain time instant is completely described by its state and input sample at that time instant. A nonlinear state space model that takes the state of the system into account can be mathematically described as [82]

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k) + g(\mathbf{x}_k)\mathbf{u}_k \quad (3.2) \\ \mathbf{y}_k &= h(\mathbf{x}_k), \quad (3.3) \end{aligned}$$

where $\mathbf{y}_k \in \mathbb{R}^l$ is the output of the vocoder, $\mathbf{u}_k \in \mathbb{R}^m$ is the input of the vocoder, $\mathbf{x}_k \in \mathbb{R}^p$ is the state of the vocoder, f and g are smooth vector fields, and h is a smooth nonlinear function. Note that this is a discrete-time model with k denoting time, since vocoders always deal with the sampled signal.

To obtain a state space model of a nonlinear dynamic system, it is necessary to estimate the mappings f, g and h using only measurements of the system's input and output. One approach is to use a model structure that can be considered as a general approximation to these mappings. Examples of such structures are neural networks and radial basis function networks [83]. These models are difficult to identify and analyze due to their complex structures. Estimating these models usually involves solving a non-convex nonlinear optimization problem.

A useful technique of nonlinear analysis is linearization, which approximates a nonlinear system by a linear state space model at an operating point [84]. Considering that the number of the vocoder states is very large, a continuous variable $\mathbf{x}$ is used to replace $\mathbf{x}_k$. Without loss of the generality, assume that the nonlinear characteristic of the mapping f is as shown in Figure 3.6. This mapping has an input $\mathbf{x}$ and an output $f(\mathbf{x})$. The operating point is at the input value $\mathbf{x}_0$.

Suppose that a small perturbation, $\Delta\mathbf{x}$, occurs in $\mathbf{x}$ (i.e., $\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x}$). The nonlinear function of Figure 3.6 can be linearized in the vicinity of the operating point $\mathbf{x}_0$ by using the Taylor series expansion:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \left.\frac{df(\mathbf{x})}{dx}\right|_{\mathbf{x}=\mathbf{x}_0} \Delta\mathbf{x} + \left.\frac{d^2 f(\mathbf{x})}{dx^2}\right|_{\mathbf{x}=\mathbf{x}_0} \frac{\Delta\mathbf{x}^2}{2!} + \cdots \quad (3.4)$$

Unfortunately, it is almost impossible to mathematically describe the mapping f and its derivatives in closed forms due to the complex structure of the vocoder. A practical method is to neglect the higher order terms in the expansion of $f(\mathbf{x})$. Consequently, (3.4)

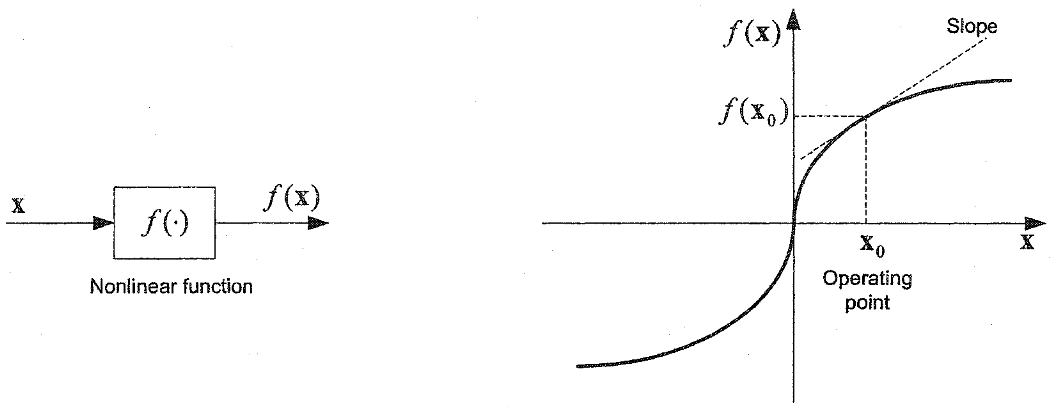

Fig. 3.6 Nonlinear characteristic of a system.

becomes

$$\Delta f(\mathbf{x}) = f(\mathbf{x}) - f(\mathbf{x}_0) \approx \left.\frac{df(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} \Delta\mathbf{x}. \tag{3.5}$$

This equation is linear, since the derivative of a function evaluated at a point $\mathbf{x}_0$ is a constant. Accordingly, the nonlinearities of the vocoder can be approximated by a linear state space model with following structure:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{3.6}$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \tag{3.7}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state. **A**, **B**, **C** and **D** are constant matrices in the vicinity of the operating point $\mathbf{x}_0$. In other words, the dynamic nonlinear system of the vocoder represented by (3.2)-(3.3) can be approximately identified by a linear system (3.6)-(3.7) if the operating point is determined. Note that the state variable $\mathbf{x}_k$ is a function of the input signal $\mathbf{u}_k$, i.e., $\mathbf{x}_k = \mathbf{x}_k(\mathbf{u}_k)$. A linear adaptive filter is able to fulfill this task by tracking changes in the operating point. Error introduced by using a linear model mainly depends on two factors: the tracking capability of the adaptive filter, and the nonlinear characteristic of the vocoder in the vicinity of the operating point. Good tracking capability ensures an accurate operating point at each iteration, while linearity in the vicinity of the operating point makes the linear approximation reasonable.

To verify the accuracy of this approximate linear model for vocoders, a simulation was

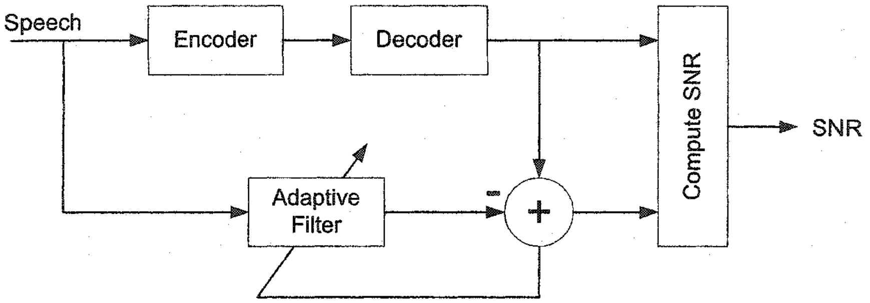

Fig. 3.7 Vocoder identification by an adaptive filter.

conducted using the setup shown in Figure 3.7. The vocoder used is G.729, and the adaptive filter is an FIR filter with 100 taps. The input signal is the same as in Figure 3.4. Results are shown in Figure 3.8, where the SNR is calculated using (3.1). It is seen that the SNR reaches up to 20-25 dB. This means that the local linearization model is suitable for the vocoder. This model provides a reasonable approximation and has been proven to yield satisfactory performance. Therefore, it is used in this thesis for most of the work involving vocoders.

3.2 Effects of vocoder nonlinearities on conventional AEC

3.2.1 Effects of vocoders on "Optimal" AEC

An acoustic echo canceller employs an adaptive filter to estimate the impulse response of the echo path and to generate an echo replica which can then be subtracted from the microphone signal. Ideally, if the impulse response (represented by filter coefficients) of the adaptive filter is exactly the same as that of the echo path, then the acoustic echo will be entirely cancelled; so there is nothing left but the near-end signal at the echo canceller output. However, when vocoders are present along the returned echo path, they will affect echo cancellation. To investigate this aspect, we first consider the use of a fixed filter to mimic the acoustic echo path, which is assumed to be time-invariant.

In the simulation set-up illustrated in Figure 3.9, the coefficients of the fixed filter are identical to the coefficients of the acoustic echo path impulse response. A fixed delay is inserted at the input of the fixed filter in order to compensate for the delay introduced by the vocoders in the echo path. The fixed filter is referred to as an "optimal" filter, since

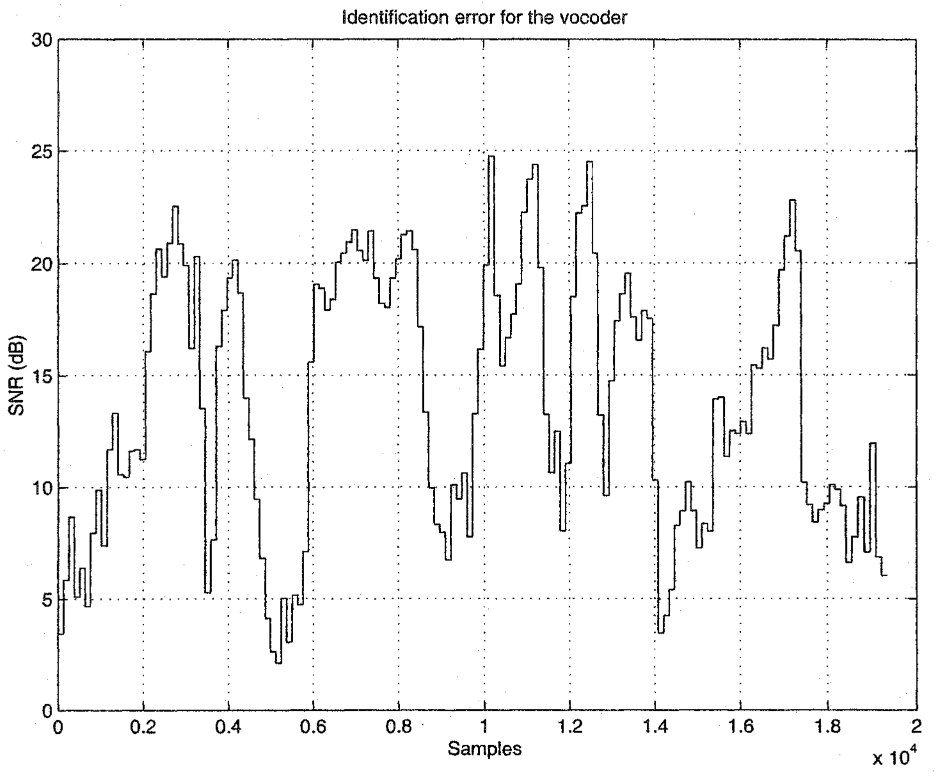

Fig. 3.8 Identification error for a vocoder.

the acoustic echo can be perfectly cancelled by this filter if vocoders are absent and other nonlinearities are neglected. To simplify the discussion, neither noise nor local speech is added to the microphone signal. Simulations are run using many different speech signals and impulse responses. Figure 3.10 shows a typical simulation result. From the results, it can be seen that when the encoders and decoders (G.729) are present, the power of the residual echo signal $e(n)$ is almost as high as that of the acoustic echo $d(n)$.

In practice, a communication path may use multiple coders, where the output of one vocoder provides the input to another. This introduces more distortion to the signal, resulting in a higher residual echo. However, from the simulation results in Figure 3.9, where the fixed filter is identical to the true echo path, it is observed that only the first vocoder added in the echo path has a major effect on the system. Adding more vocoders with the same coding technique only increases the effect slightly. This may be explained by noting that subsequent vocoders have as their input a simplified speech signal that has

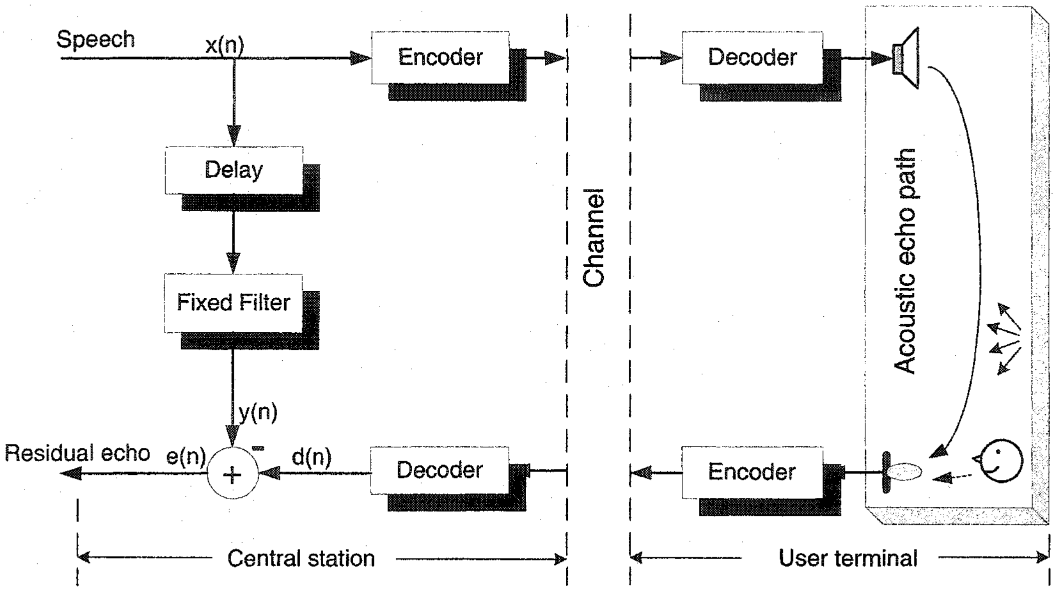

Fig. 3.9 The use of "optimal" (fixed) filter for AEC.

been "cleaned up" by the first vocoder. If vocoders with different coding schemes are used, however, the effect could be more severe because these coders exploit different properties of the signal, thus more information of the original speech could be lost.

An optimal linear echo cancellation that would work perfectly well in a conventional analog network is no longer effective in new digital networks that use speech codecs. This represents a fundamental drawback of the conventional AEC scheme based on adaptive FIR filters in the modern context of digital networks.

3.2.2 Effect of vocoders on NLMS-based AEC

Most AEC systems use adaptive filters since the echo path is time-varying. This section investigates how the performance of such AEC systems is affected by the presence of vocoders. In particular, the performance of adaptive filters is compared with the performance of fixed filters that are matched to the true echo path parameters. This comparison would indicate whether using adaptive filters is better than simply trying to match the acoustic echo path when vocoders are present.

The simulation setup used is the same as the one shown in Figure 3.9, except that the

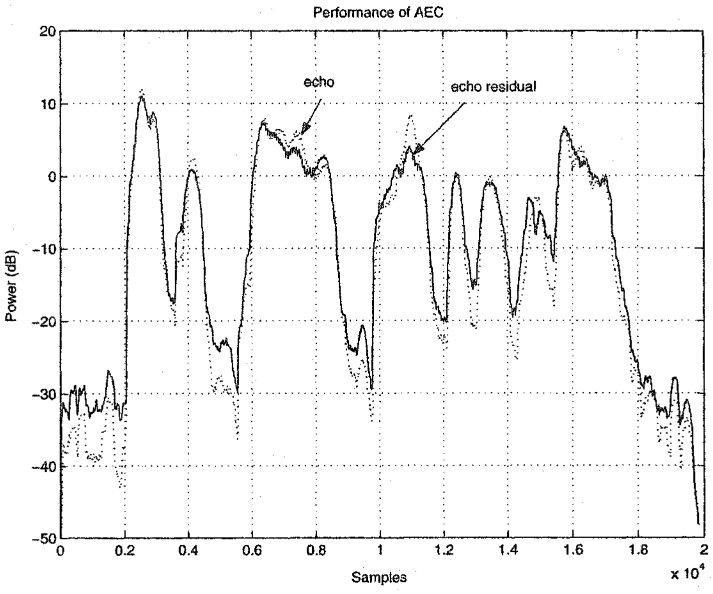

Fig. 3.10 The performance of AEC with the fixed filter in the presence of channel nonlinearities (G.729 codec).

fixed filter and the delay unit are replaced by an adaptive filter. In addition, a white noise is added to the acoustic echo when generating the total microphone signal. This adds local background noise and makes the simulation more realistic. The power of the white noise is adjusted so that the echo-to-noise ratio (ENR) at the microphone is 40 dB. This is a reasonable value when the hands-free telephone is used in a quiet environment.

Among the adaptive filtering algorithms investigated, NLMS is used first in this study because of its simplicity and robustness. This allows the fundamental characteristics of AEC in the presence of vocoders to be easily observed without complication from external factors caused by more complex algorithms.

Figure 3.11 shows the performance of the NLMS-based AEC in terms of MSE both with and without vocoders along the transmission channel. The length of the acoustic echo path used is 200 ms, which is typical for offices. Figure 3.12 shows corresponding results for a shorter acoustic echo path of 40 ms, which is typical for vehicles. Note that a longer time frame is used in the case of Figure 3.11, because the adaptive filter requires more time to converge to steady-state when the impulse response is long. It can be seen that the performance of NLMS is significantly degraded when vocoders are present. Specifically, the power of the residual echo in the nonlinear channel is about 10 to 20 dB higher than that in a linear channel. Similar results are observed for other algorithms such as RLS, AP and the decorrelation NLMS algorithms. The conclusions are summarized as follows:

- When in steady-state, the AEC performance of the adaptive filter is severely degraded by the vocoder nonlinearities regardless of the acoustic echo path length.
- Comparing Figure 3.12 with Figure 3.10, it is clear that the adaptive filter performs much better than a fixed filter whose coefficients are set to the true value of the acoustic echo path.
- Comparing Figure 3.11 with Figure 3.12, and notwithstanding the difference in convergence time, it can be seen that the effects of codec nonlinearities in steady-state are similar for both cases. In particular, because of vocoder nonlinearities, the minimum achievable residual echo power is much higher than that obtained without vocoders (where only background noise and adaptive filter misadjustment affect the minimum

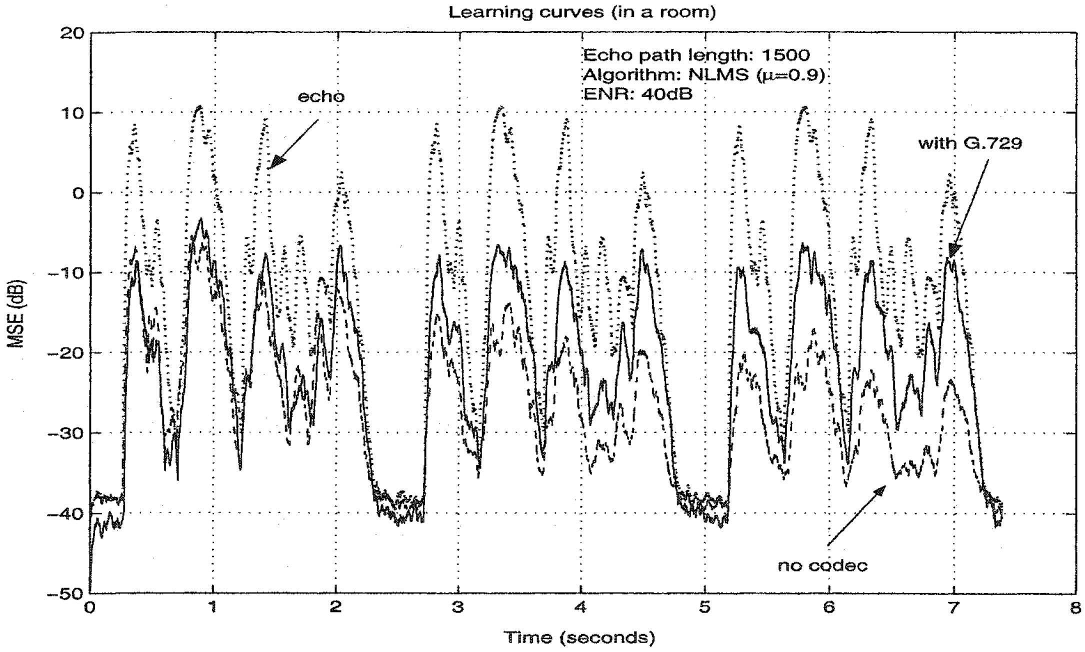

Fig. 3.11 Learning curves of NLMS-based AEC with a long impulse response.

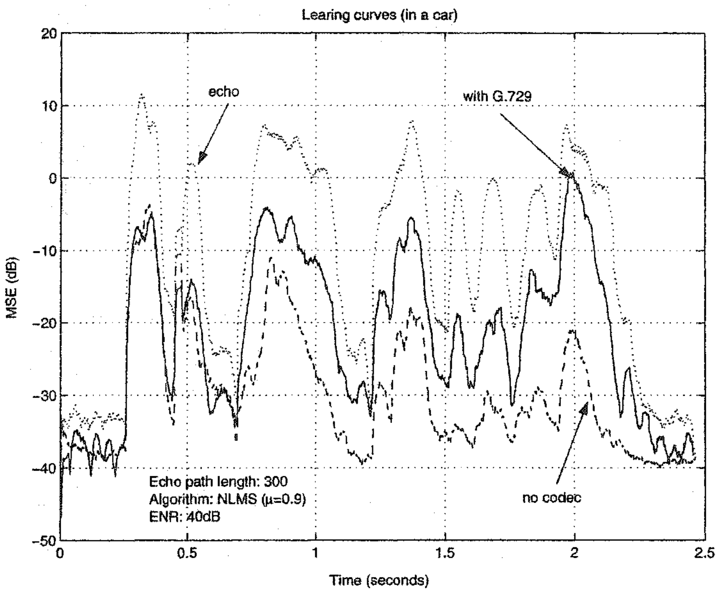

Fig. 3.12 Learning curves of NLMS-based AEC with a short impulse response.

achievable residual echo power).

Because of this “masking effect” of vocoder nonlinearities on adaptive filter performance, the full potential of an adaptive filter may not be completely exploited. The next section discusses the selection of a proper length for an adaptive FIR filter when vocoders are present.

3.2.3 Performance of AEC with different filter lengths

To evaluate the performance of AEC, a common measurement is the echo-return-loss-enhancement (ERLE) defined as

$$\mathrm{ERLE}\,(dB) = 10\,\log_{10}\frac{E[d^2(n)]}{E[e^2(n)]}, \tag{3.8}$$

where $d(n)$ is the microphone signal and $e(n)$ is the residual echo obtained by subtracting the echo cancellation filter output $\hat{y}(n)$ from the microphone signal $d(n)$. There are two ways to compute the expectation in (3.8): (1) using the definition of $E[\cdot]$, it is computed by averaging over the sample space, i.e., the same experiment is repeated many times with different random seeds for white noise excitation, and the average is taken; or (2) based on the ergodicity assumption of the signal under consideration, it is computed by averaging over time in a single experiment, which is much more practical than (1). Although a speech signal is non-stationary in general, it can be treated as a piecewise stationary signal as long as the window length is small enough. The expectation in (3.8) can thus be computed using a window with a proper length. This is the approach taken in most research work on AEC.

In AEC applications, the enclosure system's impulse response which models sound transmission between the loudspeaker (the system's input) and the microphone (the system's output) has in theory an infinite duration because of the sound's multiple reflections. In practice, however, the magnitude of the impulse response decays rapidly with time so that its tail can be discarded once its magnitude is small enough. Suppose the impulse response has a length of N samples, where N is large enough so that the coefficients beyond N can be neglected. If an adaptive filter with length L is excited by a white noise $x(n)$, and no local noise is present, it can be shown that the upper bound of the ERLE is equal to the ratio of the impulse response power to the power of its tail, that is:

$$\mathrm{ERLE}\,(dB) = \begin{cases} 10\,\log_{10} \dfrac{\sum\limits_{i=0}^{N-1} h^2(i)}{\sum\limits_{i=L}^{N-1} h^2(i)}, & \text{if } L < N; \\ \infty, & \text{if } L \geq N, \end{cases} \tag{3.9}$$

where $h(n)$ is the LEM impulse response. Ideally, the ERLE can get very large if the filter length L is large. In practice, the ERLE hardly exceeds 60 dB because of the presence of local noise and the misadjustments of the adaptive filtering algorithm. The contribution of the filter tail to ERLE is masked when L exceeds a certain value. Also, distortions caused by vocoders result in the saturation of ERLE at levels notably lower than that achievable in conventional AEC (i.e., AEC without vocoders), as shown in Figure 3.11 and 3.12. Since only a small portion of the total echo path impulse response power is contained in its tail and ERLE experiences saturation with the presence of vocoders, it appears that adaptive FIR filters that are shorter than the ones in conventional AEC may be used here without degrading the ERLE. Moreover, this would reduce the computational load and increase the initial convergence speed.

In a simulation, an adaptive NLMS filter is excited by a speech signal until its filter coefficients reach steady-state. A local noise is added so that the ENR reaches 40 dB. This simulation is run both with and without vocoders, and for both short and long echo paths. In each case, different values of the filter length L and step size μ are tested. The reason for using different step sizes is that in steady-state, where the filter length approaches the length of the echo path, using smaller step sizes results in a larger ERLE in theory.

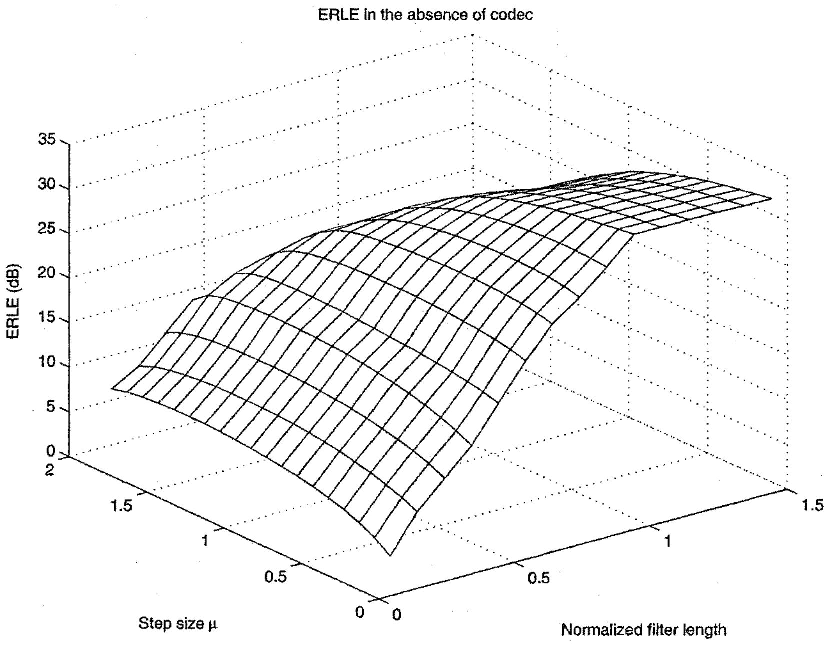

Fig. 3.13 ERLE versus normalized filter length and step-size μ in the steady-state (no vocoders).

Figure 3.13 shows the results for the case where vocoders are absent and the echo path is short. The ERLE values resulting from different filter lengths and step sizes are displayed. Note that the filter length N has been normalized by the true acoustic echo path length L. Figure 3.14 shows the results for the same case but with vocoders present along the channel.

A comparison of Figures 3.13 and 3.14 indicates that vocoder nonlinearities play a more important role than filter truncation in limiting the attainable ERLE (at least for

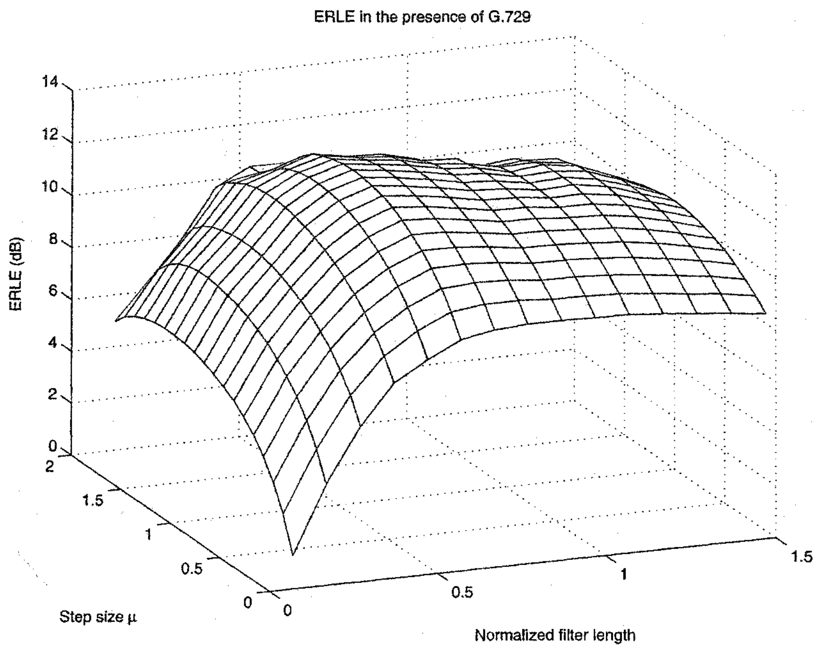

Fig. 3.14 ERLE versus normalized filter length and step-size μ in the steady-state (with G.729 vocoder).

reasonable values of filter length). According to classical adaptive filter theory, if vocoders are absent and the step sizes are small, then the maximum value of ERLE in steady-state should be attained when the normalized filter length is close to one (i.e., when the adaptive filter has the same length as the echo path), and the value of ERLE should remain close to that level when the normalized filter length is larger than one. For larger step sizes, the saturation phenomenon in ERLE would occur at a shorter normalized filter length due to misadjustment noise. This predicted behaviour is consistent with the results in Figure 3.13.

Figure 3.14 shows the different behaviour of ERLE when vocoders are present along the echo path. In this case, the maximum value of ERLE is attained for a much smaller value of normalized filter length, namely $N/L \sim 0.6$. Also, the maximum value of ERLE is only about 12 dB, compared to about 32 dB in Figure 3.13, where $\mu = 0.9$. Another interesting difference is that the ERLE does not remain constant as N/L increases beyond 0.6. Instead of the saturation phenomenon, the ERLE actually decreases as the normalized filter length exceeds 0.6. Results for different echo paths (not shown here) reveal a similar behaviour. This is caused by reduced tracking capability when the filter length is increased.

In conclusion, when vocoders are present, the adaptive filter can be shorter than the acoustic echo path without degrading the ERLE of the AEC system. In fact, this can improve performance since it reduces computational complexity and increases convergence speed.

3.3 Tracking capabilities of adaptive filters

3.3.1 Adaptive filter coefficients in steady-state

It is well known that in order to obtain the smallest misadjustment in steady-state, the step size μ of NLMS should be as small as possible under the constraint of an appropriate initial convergence speed in conventional AEC. However, when vocoders are present, ERLE reaches the maximum when μ is about 1, as seen in Figure 3.14. A larger or smaller value of μ would cause ERLE to decrease. The fact that the ERLE is lower for a smaller value of μ in steady-state implies that the adaptive algorithm does not converge to a certain underlying optimal solution. Instead, the adaptive filter coefficients attempt to track the continual change of the echo path caused by vocoders. As such, the step size should be large enough to track the changes of the system but not too large to produce a significant misadjustment that also reduces ERLE.

Figure 3.15 shows changes in the adaptive filter coefficients in steady-state when vocoders are not present. The adaptive filter used has 300 taps, and the step size μ is set to 0.9. In practice, the filter coefficients can be assumed to have reached steady-state after a long period of adaptation (e.g., 10 seconds). For clarity, the figure shows a downsampled version of the adaptive filter coefficient vector, where only one out of ten coefficients is included. Observing the coefficients along the iteration axis, no significant changes is seen except for a small noise-like disturbance (gradient noise), which is caused by the relatively large μ in steady-state. Clearly, Figure 3.15 shows that the adaptive filter has converged to its optimal solution.

Figure 3.16 displays changes in the adaptive filter coefficients when vocoders are present. All other conditions are the same as before. It can be seen that the coefficients change drastically over time. These results, along with the the analysis in 3.1.3, imply that the echo path in the presence of vocoders can be approximated as a rapidly time-varying linear system. In order to identify such a time-varying system, the adaptive algorithm should have very good tracking properties when excited by speech signals.

The aim of most adaptive filter algorithms is to minimize a cost function, usually defined as the MSE $J = E[e^2(n)]$. As a result, the adaptive filter tries to identify the entire echo path instead of just the LEM impulse response in an AEC system. Therefore, when the echo path changes drastically (as displayed in Figure 3.16 where vocoders are present) while the local linearization model is used, the adaptive filter attempts to track the change of the echo path by minimizing J. Obviously, the faster an adaptive filter can track the change of the echo path, the more acoustic echo can be cancelled.

The fact that the adaptive filter operates in a tracking mode when codecs are present poses no problems during single-talk when only the near-end speech or the far-end speech is active (i.e., they are not active simultaneously). However, the situation is quite different for double-talk. Recall that in conventional AEC, where it is assumed that the acoustic echo path varies slowly over time, one simply needs to freeze the adaptive filter during the double-talk period, and echo attenuation would still be effective. When vocoders are present, however, the adaptive filter only operates effectively in a tracking mode and therefore, freezing the coefficients no longer works well. In fact, freezing the coefficients may cause

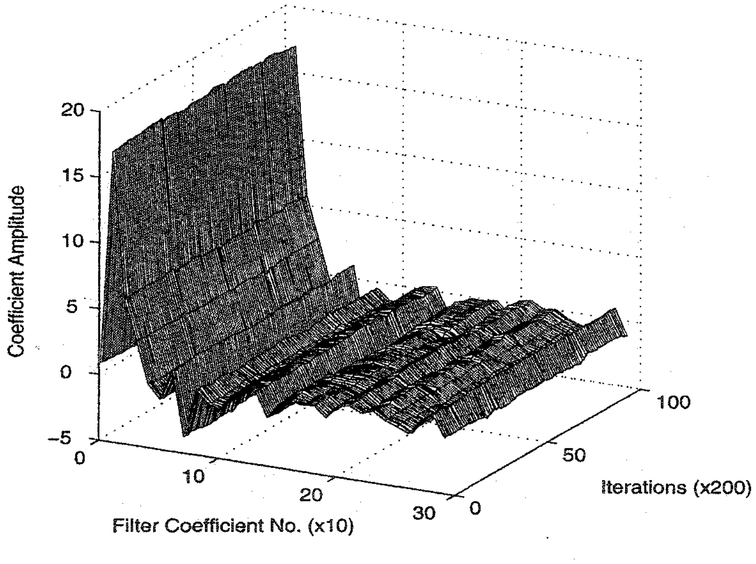

Fig. 3.15 Time variations of adaptive filter coefficients in the steady-state where no vocoder is present.

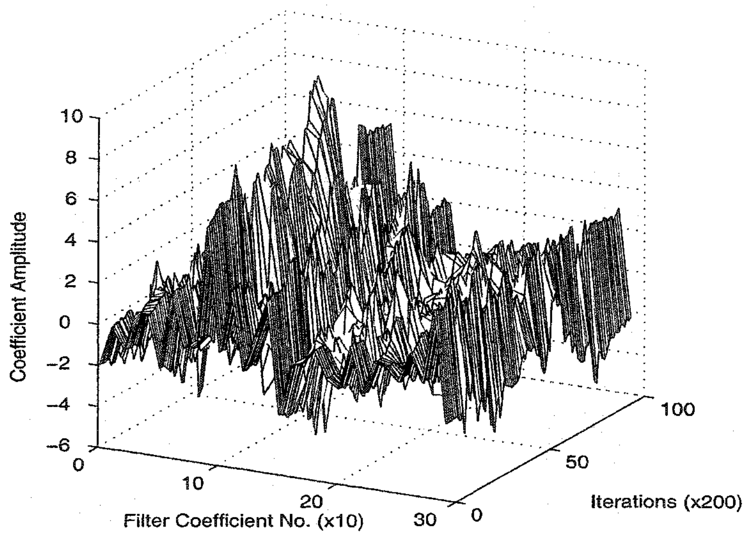

Fig. 3.16 Time variations of adaptive filter coefficients in the steady-state, in the presence of G.729 codec.

the residual power to be temporarily larger than the echo power itself. Thus, when codecs are present, one approach is to simply stop the echo cancellation altogether during double-talk.

3.3.2 Tracking capability versus MSE in the presence of vocoders

Previous sections have investigated the performance of AEC systems based on NLMS in the presence of vocoders. It was shown that vocoder nonlinearities severely degrade the performance of traditional adaptive filtering algorithms. With vocoders, the total echo path becomes a rapidly time-varying system. In fact, the total echo path changes with the speech context depending on the vocoder's coding scheme. Consequently, unlike the traditional case where vocoders are absent, a steady-state solution for acoustic echo cancellation does not exist. From the analysis in the previous sections, it appears that the tracking capability is the most important property of a good acoustic echo canceller in the presence of vocoders.

In order to confirm this viewpoint, the following sets of experiments were carried out: The first set runs the adaptive filtering algorithms in the absence of vocoders, and their tracking behaviour in steady-state is monitored under a sudden change of the echo path.

The second set of experiments run the same algorithms in the presence of vocoders and with a fixed acoustic echo path. For both sets of experiments, the associated learning curves are plotted. By comparing these learning curves, it is possible to confirm whether the faster response of a particular algorithm to a sudden change in impulse response (i.e., a better tracking property) is accompanied by a lower residual echo in the case where vocoders are present. Of particular interest here is the tracking behaviour immediately after the change in the echo path, say in the first 100 ms. When vocoders are present, changes in effective echo path impulse response (due to a change in coding principle) will typically occur over this time frame.

Now we examine the tracking capabilities of some popular AEC algorithms. The experimental conditions are identical for the same algorithm applied in different situations (i.e., in the presence or absence of vocoders). As before, the vocoder used here is G.729, and the ENR is set to 40 dB. The step size for NLMS and AP is 0.9, and the forgetting factor for RLS is 0.999. The projection order of AP is set to 3 (this is chosen since it was observed that a higher order does not improve tracking behaviour).

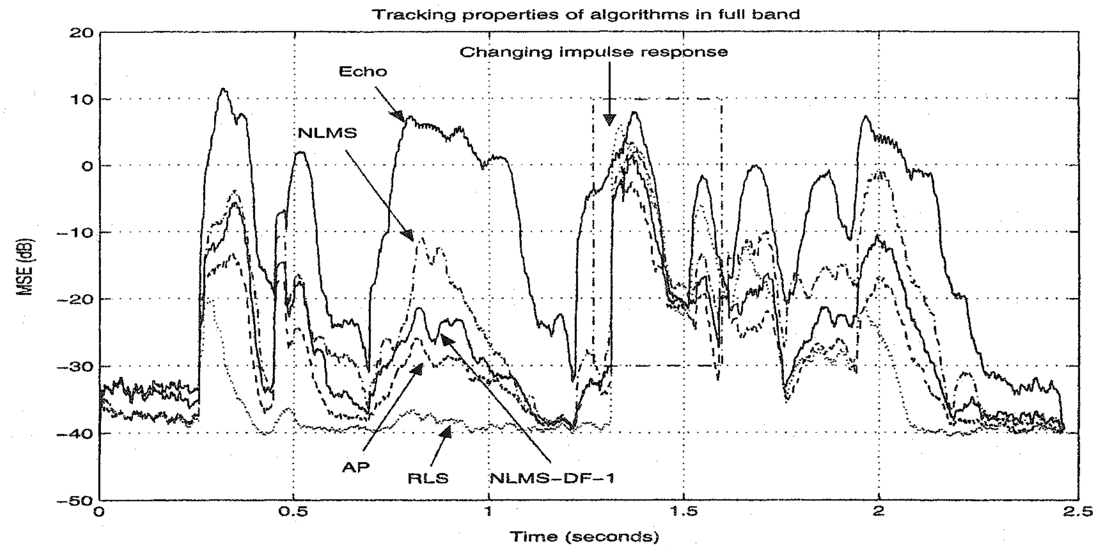

Fig. 3.17 Tracking properties of various full-band adaptive filtering algorithms (ENR=40 dB)

Figure 3.17 shows the tracking properties of popular adaptive filtering algorithms for traditional AEC in a full-band structure without vocoders along the echo path. All these

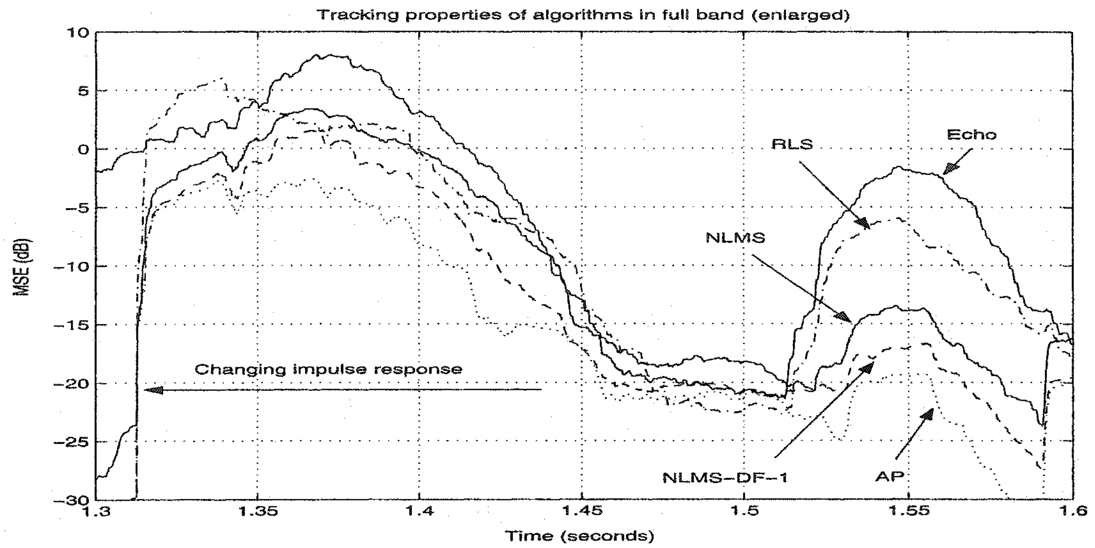

Fig. 3.18 Tracking properties of various full-band adaptive filtering algorithms (enlarged from Figure 3.17).

algorithms have been described in Chapter 2. To simulate a sudden change of the impulse response of the echo path, the impulse response coefficient vector **h** is changed to $-\mathbf{h}$ at about 1.31 s in time. For clarity, the portion in Figure 3.17 within the dash-box is enlarged in Figure 3.18. The results in Figure 3.17 and Figure 3.18 clearly show that the AP algorithm has the strongest tracking capability while RLS has the poorest, although the latter demonstrates the fastest initial convergence speed among all the adaptive algorithms. It is concluded that AP should have the best performance in the presence of vocoders, and RLS the worst one.

To make a clear comparison, the learning curves of the above algorithms when vocoders are present are plotted in Figure 3.19. Comparing the echo cancellation performance of the algorithms in Figure 3.19 to their tracking behaviour in Figure 3.18, one can see that they agree with each other. In other words, AP achieves the lowest MSE in the presence of vocoders because it tracks the changes in the echo path fastest among all these algorithms. In contrast, RLS shows the worst performance in the nonlinear channel. Between these two extreme cases, the NLMS-DF1 achieves a lower MSE than the traditional NLMS.

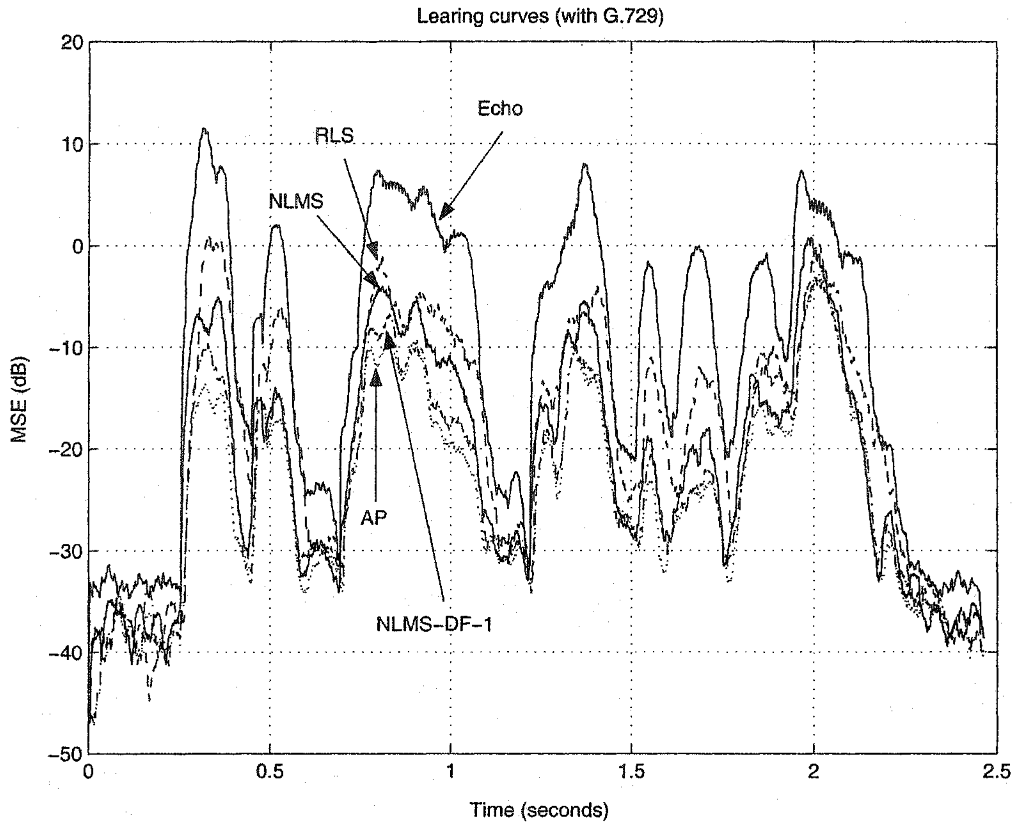

Fig. 3.19 Performance of various full-band adaptive filtering algorithms in the presence of vocoders (ENR=40dB).

3.4 Loudspeaker nonlinearities in AEC

The nonlinearities of a loudspeaker are mainly contributed by its nonlinear cone suspension and uneven magnetic flux densities, especially under high volume [61]. In AEC applications, the loudspeaker nonlinearities may be neglected if the loudspeaker is of high quality and plays at a moderate volume. However, lower quality loudspeakers are usually preferred in hands-free phones in order to reduce cost. Furthermore, the loudspeaker volume may need to be high in a mobile terminal under a noisy environment. In such cases, the entire echo path would have nonlinear characteristics that degrade the performance of a linear AEC device.

3.4.1 Analysis of loudspeaker nonlinearities

When the input power is low, the behaviour of a loudspeaker is usually modelled as a linear system [85]. However, the presence of additional distortion components (e.g., harmonics and intermodulation) indicates that the loudspeaker is a nonlinear system [86]. In fact, these distortions increase rapidly as the input power increases. Nonlinear modelling is required to describe the vibration behaviour and the transfer characteristics of loudspeakers over the entire range of their power handling capacity.

It is difficult to completely describe the nonlinear characteristics of a loudspeaker. However, if the nonlinearities are to be localized, then it is convenient to use the lumped-parameter model [87]. In general, loudspeakers can be modelled in three aspects: the electric, mechanical and acoustic parts of the transducer [88]. The modified lumped-parameter model is described as follows.

First, the electric part of the voice coil can be described in terms of the voltage at the coil terminals and the current flowing in the coil. The latter depends on the wire resistance, the coil inductance, and the transduction of electric quantities (i.e., voltage and current) into mechanical quantities (i.e., force and speed). In the electric part, the most prominent nonlinearities are introduced by the force factor and the electric self-inductance, both depending on the voice coil excursion [87]. The electric part of a loudspeaker can be expressed as

$$e(t) = R_E i(t) + \frac{d(L_E(x)i(t))}{dt} + Bl(x)\frac{dx}{dt}, \tag{3.10}$$

where $e(t)$ is the voice coil's voltage, $i(t)$ is the coil's current, R_E is the coil's resistance, $L_E(x)$ is the coil's inductance, $Bl(x)$ is the force factor, and $x \equiv x(t)$ is the coil excursion.

Second, the mechanical part can be described by the force $F(t)$ which acts on the coil cone assembly. Its dynamic behaviour depends on the viscous resistance, the compliance of the elastic suspension system, the inertial mass, and the transduction of mechanical quantities (i.e., force and cone's velocity) into acoustical quantities (i.e., pressure and volume air velocity). In the mechanical part, the nonlinear suspension stiffness contributes the major nonlinearities. The force is expressed as

$$F(t) = Bl(x)i(t). \tag{3.11}$$

Finally, the acoustic part can be described in terms of several parameters, such as the acoustic pressure in front of the loudspeaker, the acoustic impedance, etc. Combined with the mechanical part, a coupled differential equation is obtained that describes the vibration of the loudspeaker diaphragm by a mass-spring system:

$$F(t) - M_m \frac{d^2x}{dt^2} - R_m \frac{dx}{dt} - K(x)x = p(t)S_r, \tag{3.12}$$

where M_m is the moving mass, R_m is the mechanical damping, $K(x)$ is the stiffness of the mass-spring system, $p(t)$ is the acoustic pressure, and S_r is the effective radiating surface.

Equations (3.10), (3.11) and (3.12) describe the nonlinear vibration of a loudspeaker. Note that the parameters of this model rely on measurement results; therefore it is difficult to accurately describe the nonlinearities. However, because the major nonlinearities heavily depend on the voice coil excursion, they mostly occur near the resonance frequency of the loudspeaker [89]. Therefore, the nonlinear parameters can be approximated by a truncated power series, or Volterra series:

$$\begin{aligned} Bl(x) &= Bl_0 + b_1x + b_2x^2 && (3.13)\\ K(x) &= K_0 + k_1x + k_2x^2 && (3.14)\\ L_E(x) &= L_{E_0} + l_1x + l_2x^2, && (3.15) \end{aligned}$$

where the terms with orders higher than 3 have been discarded. Thus these differential equations (3.10)-(3.12) can be solved and the parameters can be determined by experiments. Examples can be found in [61, 87, 89].

Since the number of the nonlinear parameters is small, the Volterra series expansion is an effective approach to approximately identify the nonlinearities of a loudspeaker. However, when the Volterra expansion cascades with an unknown long FIR filter (such filters are usually used to model the acoustic echo path), then the number of parameters that need to be identified increases greatly. Theoretically, this nonlinear system can be modelled as a Wiener system [90] or a Hammerstein system [91], and may be identified by an adaptive Volterra filter. However, as discussed in Chapter 2, high computational complexity and convergence issues prevent this approach from practical use in AEC.

An alternative approach is to locally linearize the loudspeaker characteristics around an operating point. Because the operating point changes dynamically, tracking this change is crucial. As discussed in Section 3.1.3, an adaptive FIR filter can be employed to approximate the nonlinear system at an operating point, while tracking the operating point can be performed by the adaptation law.

3.4.2 Effects of loudspeaker nonlinearities on conventional AEC

In order to investigate the effects of the loudspeaker nonlinearities in AEC applications, we conduct several experiments as described below.

The experiments are carried out in an office room with the dimension 4(long) $\times$ 3.5(wide) $\times$2.7(high) m^3. A 1.3 GHz Pentium-IV PC is connected with a Midiman Delta 1010 Digital Recording System that has a 10-input and 10-output full-duplex recording interface. A low-cost amplified PC loudspeaker is used to play the far-end speech. The microphone signal is amplified by a Tascam MX-80 microphone/line mixer before it is sent to the recording system, where it is sampled at 8 kHz.

The experiments are run with the loudspeaker in both low volume and in a high volume. Two adaptive filtering algorithms, NLMS and the adaptive Volterra filter (based on NLMS), are evaluated. The filter length of NLMS is 600 taps. The Volterra filter is truncated up to the second order. To reduce computational complexity, the length of the first order vector is 600 taps, and the length of the second order vector is 300 taps. The step-size is set to one for both algorithms.

Figures 3.20 and 3.21 show the performance of the truncated adaptive Volterra filter and the NLMS adaptive filter under low volume and high volume. The results indicate that the Volterra filter slightly outperforms the NLMS filter in terms of ERLE. This is because the weak nonlinearities of the loudspeaker are compensated by the Volterra structure. However, when the volume is high, the loudspeaker has strong nonlinear characteristics that cannot be reasonably approximated by the truncated Volterra series expansion, resulting in notable degradation of the performance. The performance may be improved by employing a higher order Volterra filter, but the computational complexity would increase exponentially. Even the second order truncated Volterra filter used in the experiments had a computational complexity that is about 70 times that of the NLMS filter.

The performance of the NLMS adaptive filter is also degraded when the loudspeaker has high volume. This is because the operating point changes faster and the linear vicinity of the operating point becomes smaller when the low-cost loudspeaker plays in a high volume. Since the performance of the linear adaptive filter is very close to that of the adaptive

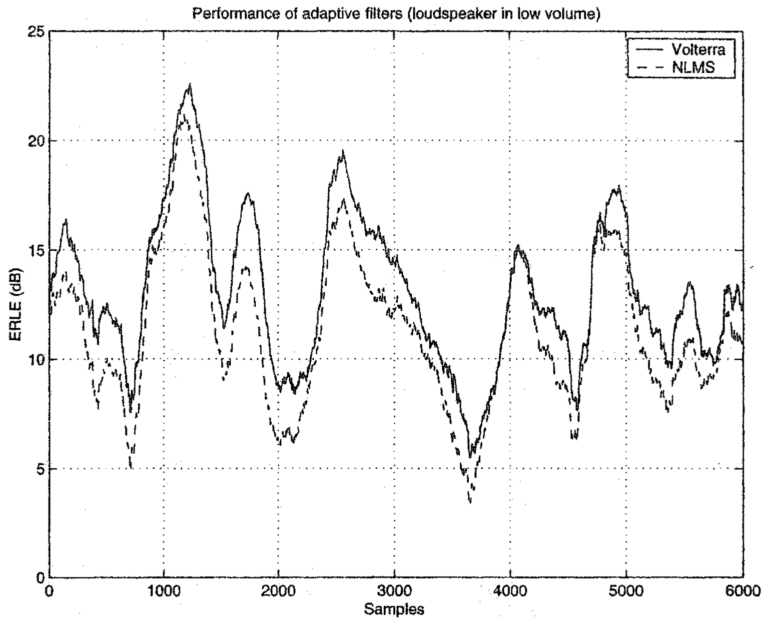

Fig. 3.20 The ERLE (dB) versus samples (loudspeaker in **low** volume): truncated adaptive Volterra filter (solid line); NLMS (dash line).

Volterra filter, and the linear filter requires much less computation power, it is concluded that linear filters should be used when handling loudspeaker nonlinearities.

3.5 Conclusion

The nonlinear devices such as low bit-rate speech codecs and non-ideal loudspeakers present strong nonlinear characteristics. These nonlinearities significantly degrade the performance of the conventional AEC, which mainly employs a linear adaptive filter to identify a time-varying linear echo path.

In this chapter, we have investigated their nonlinear mechanism through the analysis of their structures. Due to the extreme difficulty in modelling these inherent complex nonlinearities, we employed a local linearized model to study the nonlinear characteristics. Based on this simplified yet practical method, we were able to explain the performance degradation of the conventional AEC in the nonlinear channels. Specifically, the length of the adaptive filter and its tracking capability play important roles in the nonlinear channels, especially when vocoders are present.

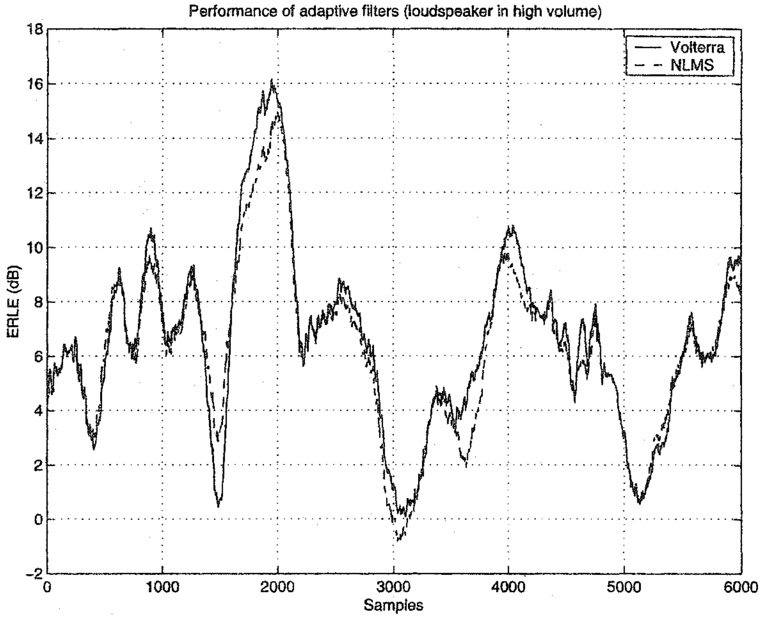

Fig. 3.21 The ERLE (dB) versus samples (loudspeaker in **high** volume): truncated adaptive Volterra filter (solid line); NLMS (dash line).

Beside the theoretical analysis, numerous experiments have been carried out in the research work. Through these experiments, we not only verified the theoretical results, but also developed a better understanding of the characteristics of the nonlinear components and their effects on the conventional AEC system.

Chapter 4

Variable step-size cross-spectral algorithm

In conventional AEC, the adaptation of adaptive filter coefficients is often frozen during a DT period, i.e., both the near-end and the far-end speakers are active simultaneously, in order to prevent the adaptive filtering algorithm from diverging. Unfortunately, stopping adaptation in a nonlinear channel may produce a residual echo that has a higher power level than the original echo signal [26]. In this chapter, we develop a variable step-size cross-spectral algorithm under an assumption of a linear echo path. However, because the proposed algorithm always keeps adapting, it can be used to suppress the acoustic echo in the nonlinear channels, especially during the DT periods.

4.1 Introduction

As discussed before, acoustic echo cancellation (AEC) can be classified as a problem of system identification, where the system is time-varying due to continual changes in the acoustic echo path. Conventional methods of adaptive system identification (e.g., the least-squares algorithms [15]) are affected by local disturbance signals such as the near-end speech and the background noise. When double-talk (DT) occurs, the adaptation of the conventional AEC device has to be stopped to avoid the possible divergence of the adaptive algorithm. Therefore, an accurate DT detector is essential for a conventional AEC device to work properly. Furthermore, tracking changes in the acoustic echo path during the DT period is particularly challenging for AEC.

During the past decade, considerable efforts have been devoted to the research of advanced AEC schemes that can properly handle DT situations. Traditionally, adaptation of the AEC device's filter coefficients is frozen by assigning a very small value (most often zero) to the step-size of the adaptive algorithm during the DT period. In this context, many researchers have focused their efforts on developing accurate and robust DT detectors [92, 93]. Alternatively, instead of completely freezing the adaptation, various schemes with a variable step-size have been attempted to track the change in the acoustic echo path when DT occurs [94, 95, 96, 18]. Unfortunately, due to fundamental difficulties in distinguishing DT from echo path changes, most solutions are not satisfactory in terms of robustness and accuracy.

Okuno *et al.* [97] proposed an adaptive cross-spectral (ACS) algorithm that is particularly attractive for AEC applications. The ACS algorithm exploits the correlation (by time averaging) of the far-end signal and the acoustic echo to estimate the echo path. Consequently, it has the advantage that DT detection is not needed and echo path changes can be tracked during the DT period. To achieve good levels of echo attenuation, the ACS algorithm requires processing many blocks of samples, so that its correlation estimate is reliable. Since the length of each block has to be greater than that of the echo path, which typically ranges from several tens up to a few hundreds of milliseconds in AEC, the ACS algorithm suffers from a relatively slow convergence rate, especially during initialization. Moreover, the non-stationary characteristics of speech signals affect the correlation estimation, leading to insufficient echo suppression during DT periods.

In this chapter, a new variable step-size ACS (VSS-ACS) algorithm is proposed which can achieve a faster convergence rate and a higher acoustic echo suppression during DT. A generalized ACS (GACS) technique is first introduced where a step-size parameter is used to control the magnitude of the incremental correction applied to the coefficient vector of the adaptive filter. Based on the study of the effects of the step-size on the GACS convergence behaviour, a variable step-size ACS (VSS-ACS) algorithm is then proposed. In order to increase the convergence rate while keeping a low misadjustment, the step-size is varied dynamically by a finite state machine that monitors changes in the norm of the ACS correction applied to the adaptive filter coefficients. In addition, a new initial adaptation scheme is adopted, resulting in a significant improvement to the convergence of the algorithm at the early stage when the network connection is established. The advantages of the new algorithm are verified by computer experiments on various sets of speech files.

4.2 The generalized ACS technique

The block diagram of a generic AEC system operating in the discrete-time domain was depicted in Figure 1.4. In the diagram, $x(n)$, $d(n)$, $\hat{y}(n)$ and $e(n)$ respectively denote the far-end speech, the microphone signal, the adaptive filter output, and the residual signal at time index n. For the time being, assume that the acoustic echo path between the loudspeaker and the microphone can be modelled as a linear, time-varying system whose impulse response at time n is represented by the N-tap vector

$$\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{N-1}(n)]^T, \tag{4.1}$$

where the superscript ^{T} denotes transposition. Accordingly, the microphone signal can be expressed as

$$d(n) = \mathbf{h}(n)^T \mathbf{x}(n) + s(n), \tag{4.2}$$

where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$ is a vector of far-end speech samples, the product $\mathbf{h}(n)^T\mathbf{x}(n)$ represents the acoustic echo, and $s(n)$ is a local disturbance signal. $s(n)$ includes both the near-end speech signal $v(n)$ (when active) and a background noise component $z(n)$, as shown in Figure 1.4.

The coefficient vector of the adaptive filter is defined as $\mathbf{w}(n)$. Its length is also assumed to be N taps, so that the adaptive filter output can be expressed as

$$\hat{y}(n) = \mathbf{w}(n)^T \mathbf{x}(n). \tag{4.3}$$

The residual signal sent to the far-end user after echo cancellation is thus given by

$$\begin{aligned} e(n) &= d(n) - \hat{y}(n) \\ &= d(n) - \mathbf{w}(n)^T \mathbf{x}(n). \end{aligned} \tag{4.4}$$

Partitioning the data into consecutive blocks of length M samples and taking the short-term Fourier transform (stDFT) [98] of length $K \geq M$ (zero-padding assumed) on both sides of (4.4), the residual signal is expressed in the frequency domain as

$$E(k;m) = D(k;m) - W(k;m)X(k;m), \tag{4.5}$$

where $E(k;m)$, $D(k;m)$ and $X(k;m)$ are the stDFTs of the signals $e(n)$, $d(n)$ and $x(n)$ over one block, respectively. $W(k;m)$ is the DFT of the vector $\mathbf{w}(n)$ assumed to remain constant within the duration of a block. Note that $W(k;m)$ represents the estimated frequency response of the echo path. In (4.5), the parameter $k \in \{1, 2, \ldots, K\}$ is the index of the frequency bins and $m \in \{1, 2, \ldots\}$ is the block index in the time domain. For the linear convolution in (4.4) to be equivalent to the circular convolution in (4.5), the stDFT size K should be such that $K \geq M + N - 1$.

The cost function is defined as

$$J_k = E[|E(k;m)|^2] = E[E(k;m)E^*(k;m)], \tag{4.6}$$

where $E[\cdot]$ denotes statistical expectation. The update equation of the generalized ACS technique is obtained using iterative steepest descent [13] applied to the cost function J_k, namely:

$$W(k;m+1) = W(k;m) - \frac{\mu'}{2} \frac{\partial E[|E(k;m)|^2]}{\partial W(k;m)}. \tag{4.7}$$

In the frequency domain, the signals and the adaptive filter coefficients are generally complex valued. Splitting the various quantities in (4.5) into real and imaginary components, the following are obtained:

$$\begin{aligned} D(k;m) &= D_R(k;m) + jD_I(k;m) && (4.8)\\ X(k;m) &= X_R(k;m) + jX_I(k;m) && (4.9)\\ W(k;m) &= W_R(k;m) + jW_I(k;m). && (4.10) \end{aligned}$$

where the subscripts R and I respectively denote the real and imaginary parts of the corresponding quantity. It is easy to show that

$$\begin{aligned} E(k;m) = {} & D_R(k;m) - W_R(k;m)X_R(k;m) + W_I(k;m)X_I(k;m) \\ & +j\ [D_I(k;m) - W_R(k;m)D_I(k;m) - W_I(k;m)X_R(k;m)], \end{aligned} \tag{4.11}$$

The partial derivative in (4.7) can then be computed as [13]

$$\begin{aligned}\frac{\partial E[E(k;m)|^2]}{\partial W(k;m)} &= E\left[\frac{\partial|E(k;m)|^2}{\partial W_R(k;m)} + j\frac{\partial|E(k;m)|^2}{\partial W_I(k;m)}\right] \\ &= -2E[X^*(k;m)E(k;m)]. \qquad (4.12)\end{aligned}$$

Applying an NLMS-like procedure of normalization, i.e.,

$$\mu' = \frac{\mu}{E[|X(k;m)|^2]}, \qquad (4.13)$$

the adaptive filter update shown in (4.7) can be written as

$$\begin{aligned}W(k;m+1) &= W(k;m) + \mu\Delta W(k;m) \qquad (4.14)\\ \Delta W(k;m) &= \frac{E[X^*(k;m)E(k;m)]}{E[|X(k;m)|^2]}. \qquad (4.15)\end{aligned}$$

We note that (4.15) can be expanded as

$$\Delta W(k;m) = H(k;m) - W(k;m) + \frac{E[X^*(k;m)S(k;m)]}{E[|X(k;m)|^2]}, \qquad (4.16)$$

where $H(k;m)$ is the frequency response of the true (unknown) acoustic echo path and $S(k;m)$ is the stDFT of the local disturbance signal $s(n)$. During DT, under the assumption that the far-end signal $x(n)$ is uncorrelated with $s(n)$ or the correlation between these signals is very weak [99], the second term in (4.16) can be neglected. Therefore, the local disturbance signal will have no effect on the algorithm.

The main difficulty associated with the implementation of (4.15) is that it requires prior knowledge of the signals' second order statistics. Although such knowledge is usually unavailable, the desired expectations can be estimated by time averaging over the block index m. Among the various types of sliding windows which can be employed for on-line time averaging, the most commonly used are the exponential and rectangular windows, with varying degrees of temporal overlap. Experiments indicate that a very small step-size has to be used with either type of window when there is significant overlap between successive windows, which in turn results in a slower convergence rate. For example, in the case of a rectangular sliding window with length L ($L \gg 1$) blocks and a minimum shift of one block between each update of the time averages, the step-size in (4.14) must be significantly reduced to avoid divergence of the algorithm due to the strong correlation between successive gradient update directions.

Assume a rectangular sliding window of length L with a window shift of Q $(1 \leq Q \leq L)$ blocks for each update of the algorithm. Then the estimated echo path is updated once every Q blocks, with the incremental correction computed as follows based on (4.15):

$$\Delta W(k;p) = \frac{\sum\limits_{i=(p-1)Q+1}^{(p-1)Q+L} X^*(k;i)E(k;i)}{\sum\limits_{i=(p-1)Q+1}^{(p-1)Q+L} |X(k;i)|^2}, \tag{4.17}$$

where $k \in \{1, 2, \ldots, K\}$ and $p \in \{1, 2, \ldots\}$.

In order to avoid processing delay in the algorithm, the operation of the adaptive filter is preferably carried out in the time domain, except for the computation of $\Delta W(k;p)$ in (4.17). Thus, the adaptive filter coefficient vector is updated every MQ samples in the time domain as follows:

$$\mathbf{w}_{p+1} = \mathbf{w}_p + \mu \Delta \mathbf{w}_p, \tag{4.18}$$

where $\Delta \mathbf{w}_p$ is the inverse DFT of $\Delta W(k;p)$. The relationship between the iteration index p and the sample index n is

$$p = \left\lceil \frac{n}{MQ} \right\rceil, \quad n \in \{1, 2, \ldots\}, \tag{4.19}$$

where $\lceil \delta \rceil$ represents the smallest integer $\geq \delta$. Therefore, the coefficient vector of the adaptive filter $\mathbf{w}(n)$ remains constant for MQ consecutive samples.

The generalized ACS (GACS) algorithm is shown in Algorithm 7. It includes equations (4.4), (4.17) and (4.18), together with the associated stDFT computations. The original ACS [97] is a special case of GACS when μ is set to one in (4.18). Introducing variable step-sizes in GACS will provide additional flexibility for improved performance (see the next Section). The step-size μ must be carefully chosen to ensure the convergence of the algorithm (i.e., $0 \leq \mu \leq \mu_{\max}$). In practice, an exact expression for the upper bound $\mu_{\max}$ cannot be derived easily with non-stationary, non-white inputs such as speech signals.

Algorithm 7 The generalized adaptive cross-spectral (GACS) algorithm

```
Initialization:
$\mathbf{w}_1 = \mathbf{0}$
Recursion:
for $n = 1, 2, \ldots$ do
  $p = \lceil \frac{n}{MQ} \rceil$,
  $e(n) = d(n) - \mathbf{w}_p^T \mathbf{x}(n)$
  if $\lceil \frac{n}{M} \rceil = \frac{n}{M}$ then
    $m = \frac{n}{M}$
    $\mathbf{X}(m) = \mathrm{DFT}\{\mathbf{x}(n)\}$
    $\mathbf{E}(m) = \mathrm{DFT}\{\mathrm{e}(n)\}$
    if $m = L + (p-1)Q$ then
      for $k = 1, 2, \ldots, K$ do
        $\Delta W(k;p) = \frac{\sum_{i=(p-1)Q+1}^{(p-1)Q+L} X^*(k;i)E(k;i)}{\sum_{i=(p-1)Q+1}^{(p-1)Q+L} |X(k;i)|^2}$
      end for
      $\Delta \mathbf{w}_p = \mathrm{IDFT}\{\mathbf{\Delta W}(p)\}$
      $\mathbf{w}_{p+1} = \mathbf{w}_p + \mu \Delta \mathbf{w}_p$
    end if
  end if
end for
```

Experimentally, it has been observed during the development of this algorithm that $\mu_{\max}$ depends on the input signal type, the rectangular window length L, and the data reuse factor (i.e., window overlap ratio) $(L - Q)/L$. For all the practical cases tested in this work, the GACS algorithm worked properly with no observed instability when μ is selected within the range $0 \leq \mu \leq 1.5$.

There is a certain degree of similarity between GACS and the standard frequency domain adaptive filter (FDAF) [13]. However, although both GACS and FDAF operate in the frequency domain, the former uses block averaging to estimate the required expected values, while the latter uses instantaneous estimation (i.e., it uses a single block). Therefore, GACS is much more robust than FDAF with disturbance signals, as long as a proper number of blocks is employed in the time averaging operation in (4.17).

Finally, the behaviour of GACS as a function of its step-size μ is consistent with that of other steepest descent adaptive filters (including FDAF). That is, increasing the step-size results in a faster convergence rate at the expense of higher coefficient misadjustment in the steady-state. (See 4.4.2 for results.)

4.3 Variable step-size ACS

4.3.1 Finite state machine

The observed performance of GACS suggests that using variable step-sizes could improve the convergence rate and reduce the misadjustment. A larger value of μ should be used during acoustic echo path changes and a smaller value should be used when the algorithm has converged. Although many adaptive filtering algorithms with variable step-sizes have been developed, there is still a lack of a robust algorithm which can accurately distinguish an echo path change from a DT situation, and also track the echo path change during a DT period.

Here, we propose a variable step-size ACS (VSS-ACS) algorithm based on the GACS algorithm. It incorporates a step-size adjustment mechanism that is regulated by a finite state machine, as shown in Fig. 4.1. The figure identifies three regions of operation that correspond to different states. These regions are explained below.

Region I: This region corresponds to a fast tracking mode where the algorithm needs to estimate or track large changes in the acoustic echo path. Initial operation of the algorithm upon network connection also falls into this region. In order to quickly track echo path changes, it is necessary to have a larger step-size (denoted μ_I).

Region II: This region represents a transient mode of the algorithm. After rapid adaptation in Region I, misadjustment (due to gradient noise) gradually becomes significant while it is still important to keep the adaptive filter updating its weight vector at a reasonable rate. The introduction of two states with smaller step-sizes, μ_{IIa} and μ_{IIb}, in this region ensures a smooth and robust transition from Region I to Region III (see below).

Region III: This region corresponds to the steady-state of the algorithm; a small step-size μ_{III} is proper in this region. Two factors actually limit the choice of the step-size. On the one hand, the step-size should not be too small, otherwise the VSS-ACS will not be able to track the small, ever present fluctuations in the acoustic echo path. On the other hand, it should be small enough so as to keep the misadjustment low and prevent the estimated echo path error from increasing during DT periods. Indeed, while in theory the GACS is not affected by DT, it will be in a practical implementation of Eq. (4.15) based on temporal averaging with non-stationary signals.

The adaptive filter coefficient error could be used in theory to determine the proper state of the algorithm. Unfortunately, it is almost impossible to obtain this information in practice because the acoustic echo path is unknown and time-varying. As an alternative, this work proposes to use the energy ratio of two successive increments of the estimated system impulse response as the basis for a series of tests to decide upon state transition. This energy ratio, denoted $\delta_w(p)$, is formally defined here as

$$\delta_w(p) = \left| 20 \log_{10} \frac{\|\Delta \mathbf{w}_p\|}{\|\Delta \mathbf{w}_{p-1}\| + c} \right|, \tag{4.20}$$

where $\delta_w(p)$ is defined as a non-negative value in order to avoid the fluctuation around 0 (i.e., the sign changes). $\|\mathbf{u}\|$ is the norm of the vector $\mathbf{u}$, and c is a small constant that prevents the division from overflow. Experiments indicate that this energy ratio is strongly indicative of the true (but practically unknown) error in the adaptive filter coefficients.

Figure 4.1 shows the four states of the VSS-ACS algorithm. Transitions between these states are determined by comparing $\delta_w(p)$ with a set of thresholds, labelled $\lambda_1, \lambda_2, \ldots, \lambda_5$. For example, when in state μ_{IIa}, a transition to state μ_{IIb} occurs if $\delta_w(p) < \lambda_2$; otherwise, a transition to state μ_I occurs if $\delta_w(p) > \lambda_5$. If neither case arises, then the algorithm remains in state μ_{IIb}. Considerations in the selection of the thresholds are discussed below.

First, consider the case where the acoustic environment does not change significantly over time. During the adaptation of VSS-ACS, the state of the algorithm is expected to transit from Region I to Region II, or from Region II to Region III. Here, it is straightforward to set the values of λ_1, λ_2 and λ_3. They should be in descending order, representing different stages on the way to the convergence of the algorithm.

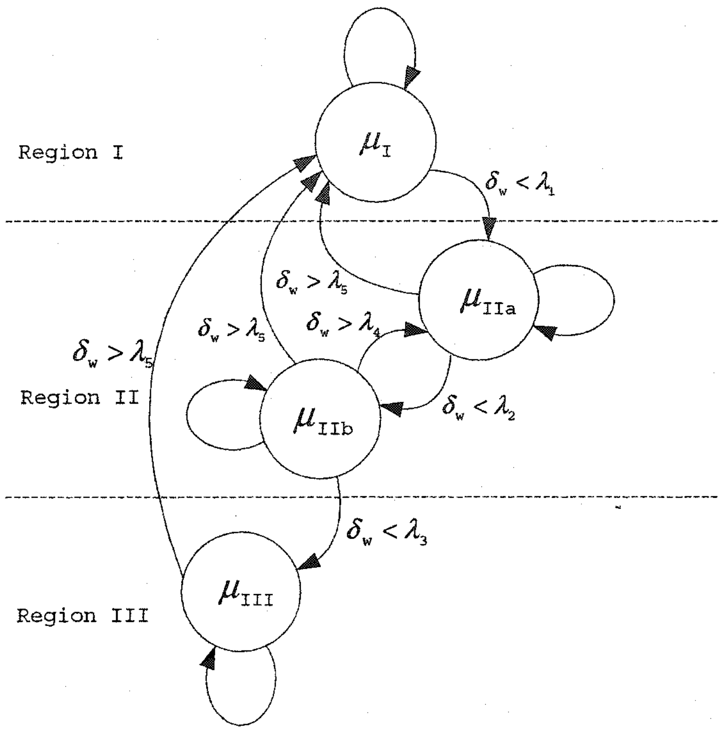

Fig. 4.1 State machine diagram of the VSS-ACS algorithm.

Next, consider a situation where the acoustic echo path is notably changed (including initialization). In this case, the algorithm is expected to track the variations as fast as possible. Hence, the state of the algorithm should jump to Region I, no matter which region it was in previously. One major advantage of the GACS technique over more traditional approaches is its intrinsical ability to distinguish between echo path change and DT. In the DT situation, $\delta_w(p)$ will not increase significantly, whereas a sudden significant change in the echo path will result in a large difference in $\delta_w(p)$. Therefore, the threshold for the change of the acoustic echo path, λ_5, is significantly higher than the other thresholds.

Finally, two states are introduced in Region II to improve the robustness of the algorithm. This is because the adaptive filter coefficient update may be unfavourably affected by a specific segment of speech input. In that case, more time is needed to decide whether the algorithm has reached the steady-state, where μ_{III} should be used. Based on this consideration, it is reasonable to set $\lambda_4 \geq \lambda_2$.

Specific numerical values for the various parameters of the VSS-ACS algorithm (i.e., step-sizes and thresholds) will be given in Section 4.4.1.

4.3.2 Improvement to initial convergence

Among the total LM samples used to compute the incremental correction in (4.17), only MQ samples are new data. Thus, similar to the original ACS, VSS-ACS updates the estimated echo path every MQ samples. At the beginning of adaptation, when the network connection has just been created, the adaptive filter coefficients are initialized to zero, and ACS starts acoustic echo attenuation once it has collected at least QM samples (assume there are $(L-Q)M$ padding zeros) to estimate the echo path. In AEC, the acoustic echo path may be as long as several hundred or even several thousand taps at an 8 kHz sampling rate. Consequently, with Q sufficiently large (say 40 or more [97]) and $M \geq N$, where N is the adaptive filter length, QM samples represent a relatively long time interval. It would be inappropriate not to suppress the acoustic echo during such a long initial period. In order to overcome this drawback, a modification of the VSS-ACS algorithm is described below where different approximation and adaptation schemes are used to compute the filter weight vector during the initial period. The proposed approach is an extension of the technique originally reported in [28] for the ACS algorithm.

The microphone signal in (4.2) in the frequency domain may be expressed as

$$D(k;m) = H(k;m)X(k;m) + S(k;m), \tag{4.21}$$

where $H(k;m)$ and $S(k;m)$ have been previously defined in (4.16). Multiplying both sides of (4.21) by $X^*(k;m)$, then taking the expectation and assuming that $x(n)$ and $s(n)$ are uncorrelated, the frequency response of the acoustic echo path is obtained as

$$H(k;m) = \frac{E[X^*(k;m)D(k;m)]}{E[|X(k;m)|^2]}. \tag{4.22}$$

This is referred to as the cross-spectral technique [100].

Using a similar approximation process as the one leading from (4.15) to (4.17), the expected values in (4.22) can be estimated by time averaging. However, to make full use of the available data so as to improve the initial convergence rate, the desired expected values are estimated by accumulators running from block 1 up to the current block m (instead of the summations over L blocks as in (4.17)). Therefore, the initial adaptation is expressed as

$$W(k;m) = \frac{\sum_{i=1}^{m} X^*(k;i)D(k;i)}{\sum_{i=1}^{m} |X(k;i)|^2}, \tag{4.23}$$

where the block index $m \in \{1, 2, \ldots, L\}$. The coefficients of the estimated acoustic echo path are then obtained by taking an inverse DFT on (4.23). It has been observed that the magnitude of the estimated echo path impulse response may be smaller than that of the real one due to the deformation phenomenon caused by the time window [100]. To reduce such systematic error introduced by the cross-spectral technique, the estimated value needs to be enlarged by a scaling factor. For consistency, this factor is also denoted by μ except that here, $\mu \geq 1$. Hence, during initialization, the adaptive filter coefficient vector is updated every M samples via

$$\mathbf{w}_{1m} = \text{inverse DFT}\{\mu W(k;m)\}. \tag{4.24}$$

Note that the coefficients of the estimated echo path are computed directly during initialization, whereas in subsequent periods they are computed recursively.

The blocking procedure of the VSS-ACS algorithm with modified initialization is illustrated in Fig. 4.2. In the figure, B_i refers to a block of m samples, and F_k refers to a frequency bin within a block.

4.3.3 Computational complexity

The VSS-ACS algorithm involves both real and complex computations. As before, computational complexity is measured in terms of *operations*, where one operation is defined as one real multiplication plus one real addition. Accordingly, four operations are required to realize one complex multiplication and one complex addition.

To efficiently implement the VSS-ACS algorithm, an FFT algorithm can be employed for the K-point stDFT computations. Thus, about $2K \log_2 K$ operations are needed to map each data block from the time domain to the frequency domain, and vice versa [101]. Hence, the computational complexities for (4.4), (4.17) and (4.18) are about LMN, $4(L+Q)K$ and N operations respectively per adaptive filter coefficient update.

In the state machine, the logarithmic scale was used in the definition of $\delta_w(p)$ in (4.20) only to simplify understanding. In practice, both the thresholds and the energy ratio $\delta_w(p)$

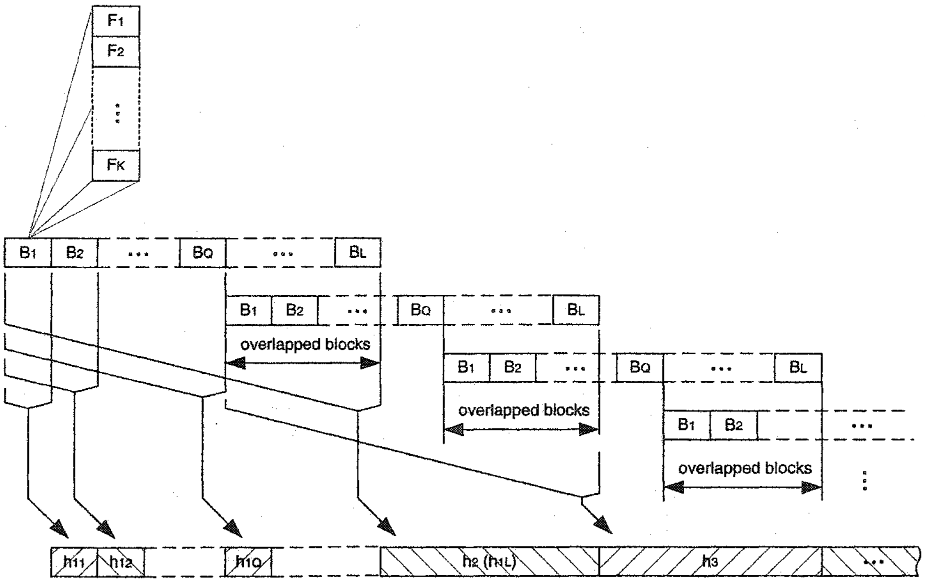

Fig. 4.2 Adaptation of the VSS-ACS algorithm.

can be calculated in the linear scale, so that the computational complexity is reduced. Hence, only the norm of the current coefficient increment needs to be computed, with a computational load of approximately N operations per filter coefficient update.

During the initial period, the implementation can exploit recursive relations to reduce the computational burden. Specifically, $W(k;m)$ in (4.23) can be computed as

$$\begin{aligned} W(k;m) &= \frac{P(k;m)}{Q(k;m)} \\ P(k;m) &= P(k;m-1) + X^*(k;m)D(k;m) \\ Q(k;m) &= Q(k;m-1) + |X(k;m)|^2, \end{aligned} \tag{4.25}$$

where $P(k;m)$ stands for $\sum_{i=1}^{m} X^*(k;i)D(k;i)$, and $Q(k;m)$ stands for $\sum_{i=1}^{m} |X(k;i)|^2$. The computational complexity of the modified initialization scheme is $MN + N + 3K\log_2 K$ operations per coefficient update.

The computational complexity of the VSS-ACS algorithm, in units of operations per sample (OPS), is obtained as follows:

a) During the initial period

During this period, the algorithm updates the adaptive filter coefficient vector every M samples. Hence the computational complexity is

$$\frac{MN + N + 6K\log_2 K + 12K}{M} \text{ OPS.} \tag{4.26}$$

For the case where the block length is the same as the adaptive filter length, i.e., $M = N$, and $K = M + N$ for linear convolution, the computational complexity in the initial period is about $N + 12\log_2 N$ OPS.

b) In the subsequent period

Because the filter coefficients are updated every QM samples, the total computational complexity of the VSS-ACS algorithm in OPS is

$$\frac{LMN + 4(L+Q)K + 2N + 4(L+Q)K\log_2 K}{MQ}. \tag{4.27}$$

For the special case $M = N$ and $K = 2N$, where one half window overlap is assumed (i.e., $Q = L/2$ blocks in Eq. (4.17)), the computational complexity is approximately $2N + 24\log_2 N$ OPS.

Compared to the standard LMS algorithm that approximately requires $2N$ OPS [102], the VSS-ACS algorithm does not require significantly higher computational capacity.

4.4 Computer experiments

4.4.1 Methodology

In the computer experiments, various segments of speech signals including males', females' and children's speech are used as the excitation signals. A coloured noise, obtained by passing a white noise signal through a first-order IIR filter $H(z) = 1/(1 - 0.9z^{-1})$, is added to the microphone signal in order to simulate a local noisy environment, so that the echo-to-noise ratio (ENR) is 30 dB. The impulse response of the acoustic echo path, which has a length $N = 300$ corresponding to 37.5 ms at an 8 kHz sampling rate, has been synthesized to mimic the driver's compartment of a motor vehicle.

To evaluate the performance of the algorithms, the following normalized measure of the estimated impulse response coefficient error at time n is introduced (in dB):

$$\text{Coef\_err}(n) = 20\log_{10}\left(\frac{\|\mathbf{h}(n) - \mathbf{w}(n)\|}{\|\mathbf{h}(n)\|}\right). \tag{4.28}$$

As before, $\mathbf{h}(n)$ is the coefficient vector of the true acoustic echo path, and $\mathbf{w}(n)$ is the coefficient vector of the estimated one.

In the implementation of the GACS and VSS-ACS algorithms, the block size is $M = 300$ samples, $L = 80$ blocks are used for the average in (4.17), and the window shift between updates is set to $Q = 40$ blocks. For the VSS-ACS algorithms, the set of step-sizes in the three regions were: $\mu_I = 1.2$, $\mu_{IIa} = 1.0$, $\mu_{IIb} = 0.8$ and $\mu_{III} = 0.1$. Correspondingly, the thresholds of the state machine were 5.5, 3.0, 1.5, 4.0, and 20.0 for $\lambda_1, \ldots, \lambda_5$.

Motivations guiding the choice of the step-sizes and the corresponding thresholds for the VSS-ACS algorithm have been discussed in Section 4.3.1. In particular, flexibility in the choice of these parameters provides the possibility of a tradeoff between different properties of the algorithm. However, it is difficult to theoretically derive the optimal values of the step-sizes and the thresholds because they depend on many factors such as M, L and Q. The above reported values of the step-sizes and thresholds were determined mainly through experiments (e.g., Figure 4.3 and Figure 4.4). They are not necessarily optimal in a mathematical sense, but they lead to good and reliable performance of the VSS-ACS algorithm. Note that proper values of the thresholds are important to reduce the number of erroneous transitions between the states. Furthermore, experiments showed that the performance of the VSS-ACS algorithm is not very sensitive to the specific choice of these parameters, provided that they are selected in the proper range. The parameter values chosen above may vary within a certain margin, say 5-10%, without incurring significant performance degradation to the algorithm.

Properties of the VSS-ACS algorithm have been tested for three different aspects: behaviour in the DT situation, initial convergence rate, and tracking capability in the presence of near-end speech. For comparison, these tests are also applied to the original ACS algorithm, with same values of M, L and Q as above.

4.4.2 Results

a) Effect of the step-size μ on GACS

To support the previous statement regarding the effect of the step-size μ on the GACS convergence rate, Figure 4.3 presents the time evolution of the coefficient error with various step-sizes. The adaptive filter weight vector is initially set to zero, and speech is used as the excitation signal. It can be seen that increasing the step-size μ in the GACS algorithm leads to faster initial convergence but higher misadjustment in steady-state, as pointed out earlier.

Note that GACS inherits the robustness properties of the original ACS algorithm to local disturbance signals. While ENR was set to 30 dB in Figure 4.3, results of other experiments show that the algorithm convergence behaviour is unaffected when the power of the disturbance signal (e.g., additive noise) varies in a large range, say for ENR as low as 5 to 10 dB.

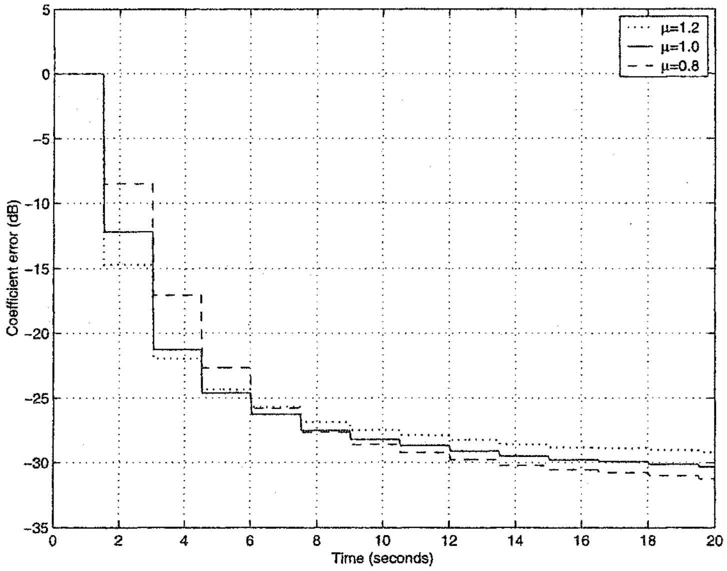

Fig. 4.3 Effects of the step-size on the GACS algorithm.

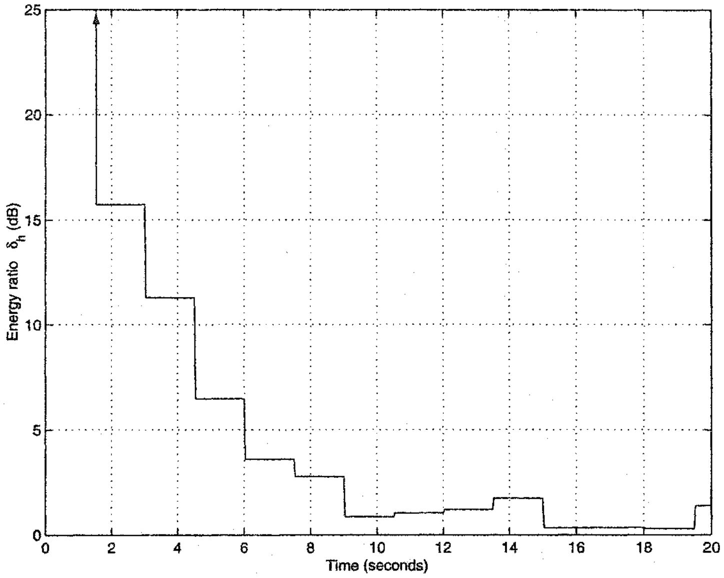

Fig. 4.4 Evolution of the energy ratio of two successive coefficient increments of the adaptive filter (GACS).

b) Evolution of the energy ratio δ_h

The time evolution of the energy ratio of successive coefficient increments, as defined by (4.20), is studied experimentally under similar conditions as in (a). Figure 4.4 illustrates the results for the case where the step-size $\mu = 1$.

Comparing the evolution of the energy ratio in Figure 4.4 with that of the coefficient error in Figure 4.3, it is found that their behaviours are similar but for a difference in scale. In particular, as the adaptation progresses from initialization, both measures decay roughly at the same rate and eventually converge to a steady-state value. In the case of the energy ratio, the limiting values after convergence fluctuate near 0 dB. These fluctuations can be explained as the result of the adaptive filter misadjustment. Indeed, during the steady-state, consecutive incremental corrections are very close to each other in terms of their energy. Thus the incremental correction does not decay any more as the progress of the adaptation. Their amplitudes depend on the specific speech segment being considered since the latter is a non-stationary signal.

Similar results were obtained for other values of the step-size. Hence, it appears justified to use the energy ratio δ_h as an approximation to the coefficient error when testing for state transition.

c) Behaviour of VSS-ACS during DT

In this experiment, a speech signal segment with comparable power as the acoustic echo is added to the microphone signal in order to simulate DT. Moreover, to clearly demonstrate the effect of DT on the algorithms, DT is only allowed to occur after initial convergence of the algorithms.

Figure 4.5 shows the performance of the original ACS algorithm (i.e., GACS with $\mu = 1$) and the VSS-ACS algorithm during the DT period. The microphone signal (its waveform

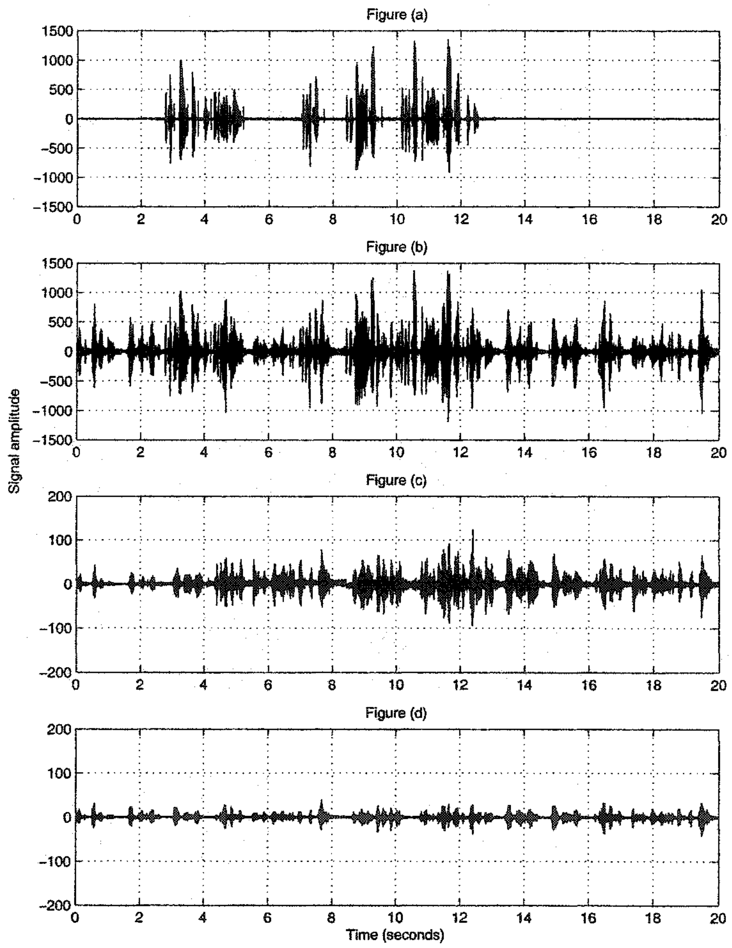

Fig. 4.5 Waveforms in the DT situation: (a) original near-end signal; (b) microphone signal (acoustic echo plus near-end signal); (c) residual echo of ACS; (d) residual echo of VSS-ACS.

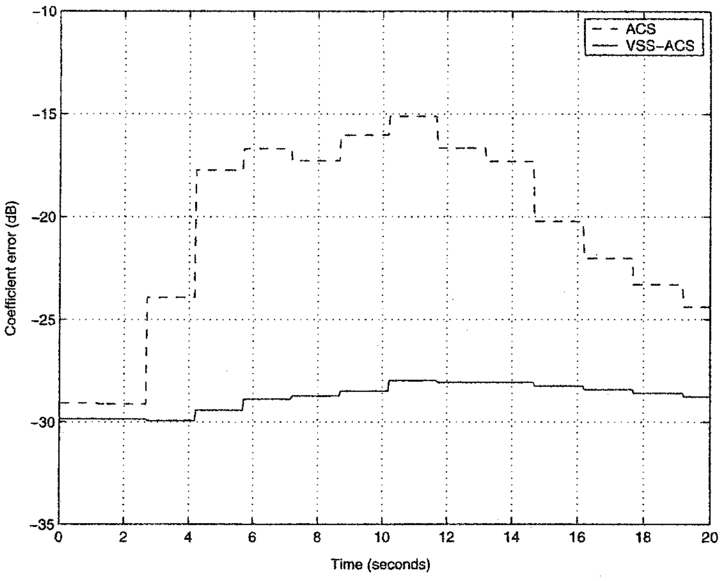

Fig. 4.6 Coefficient errors versus time for the original ACS algorithm (dashed line) and VSS-ACS algorithm (solid line).

is displayed in Fig. 4.5(b)) is comprised of the acoustic echo and the near-end signal. The latter consists of a near-end speech superimposed on a background noise, as plotted in Fig. 4.5(a). The signal waveforms of the residual echo produced by ACS and VSS-ACS are shown in Fig. 4.5(c) and 4.5(d). Note that the near-end signal has been subtracted from the residual signal for clarity. These waveforms demonstrate that VSS-ACS provides much more attenuation to the acoustic echo than ACS during DT.

The error in the estimated coefficients of the acoustic echo path is shown in Figure 4.6 for both the proposed VSS-ACS algorithm and the original ACS algorithm. These results reveal that the new VSS-ACS algorithm has notably smaller coefficient error than ACS during the DT situation, which is in agreement with the results presented in Figure 4.5. Note that the VSS-ACC step-size did not change ($\mu = 0.1$ in Figure 4.6) during the DT period; that is, the disturbance introduced by the presence of the near-end speech did not trigger a transition from state μ_{III} to another state.

d) Improvement to initial convergence

The initial convergence properties of different schemes are examined. The signal waveforms of the acoustic echo, the residual echo of ACS, and the residual echo of VSS-ACS are shown in Figures 4.7(a), 4.7(b) and 4.7(c). The VSS-ACS algorithm achieves a much faster convergence rate when the network connection is created. The time evolution of the echo path coefficient errors for both ACS and VSS-ACS are plotted in Figure 4.8. VSS-ACS shows a significant improvement in acoustic echo suppression as a result of using the modified initialization scheme described in Section 4.3.2.

e) Tracking in the presence of DT

In order to test the tracking capability of the new algorithm, the acoustic echo path is changed from $\mathbf{h}(n)$ to $-\mathbf{h}(n)$ during a DT period after the algorithm has converged. The coefficient errors for the original ACS and VSS-ACS are displayed in Figure 4.9(a), while the corresponding values of the step-size used in VSS-ACS are plotted in Figure 4.9(b). Here, the sudden change in the echo path occurs at time 8.2 seconds. The VSS-ACS state machine demonstrates a satisfactory behaviour in its ability to control the step-size, as dis-

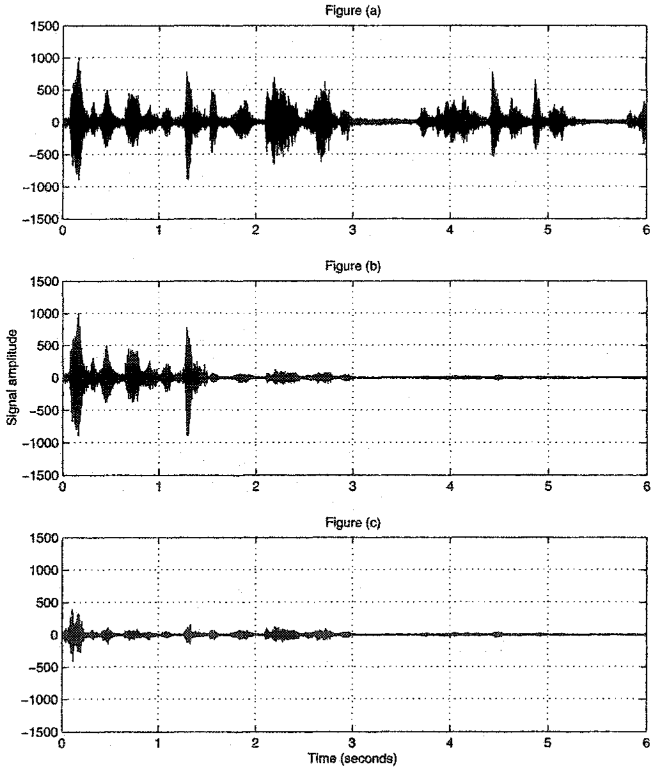

Fig. 4.7 Waveforms in the initial period: (a) acoustic echo signal; (b) residual echo of ACS; (c) residual echo of VSS-ACS.

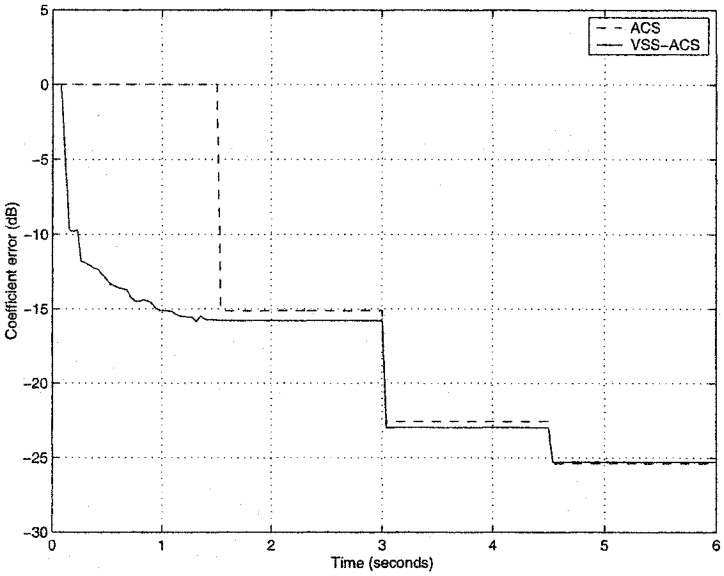

Fig. 4.8 Coefficient error versus time during initial period for the original ACS (dashed line) and VSS-ACS (solid line) algorithms.

cussed in Section 4.3.1. The proposed VSS-ACS algorithm not only inherits the desirable property of ACS (i.e., it can track acoustic echo path changes in the presence of strong disturbance signals), but it also has better performance than ACS.

f) Subjective experiments

Results of informal listening tests suggest that, compared to the original ACS algorithm, the proposed VSS-ACS can suppress the acoustic echo to a satisfactory level even during initialization. During DT, the near-end speech contained in the residual signal, which is sent to the far-end user, is more clearly audible with the VSS-ACS than with ACS because of the lower level of interference signal (i.e., residual acoustic echo). Furthermore, no perceptual distortion of the near-end speech signal is observed. In the case when the acoustic echo path changes, the acoustic echo is suppressed more rapidly by VSS-ACS.

4.5 Conclusion and discussion

A generalized ACS technique was proposed where a step-size parameter is used to control the magnitude of the incremental correction applied to the coefficient vector of the adaptive

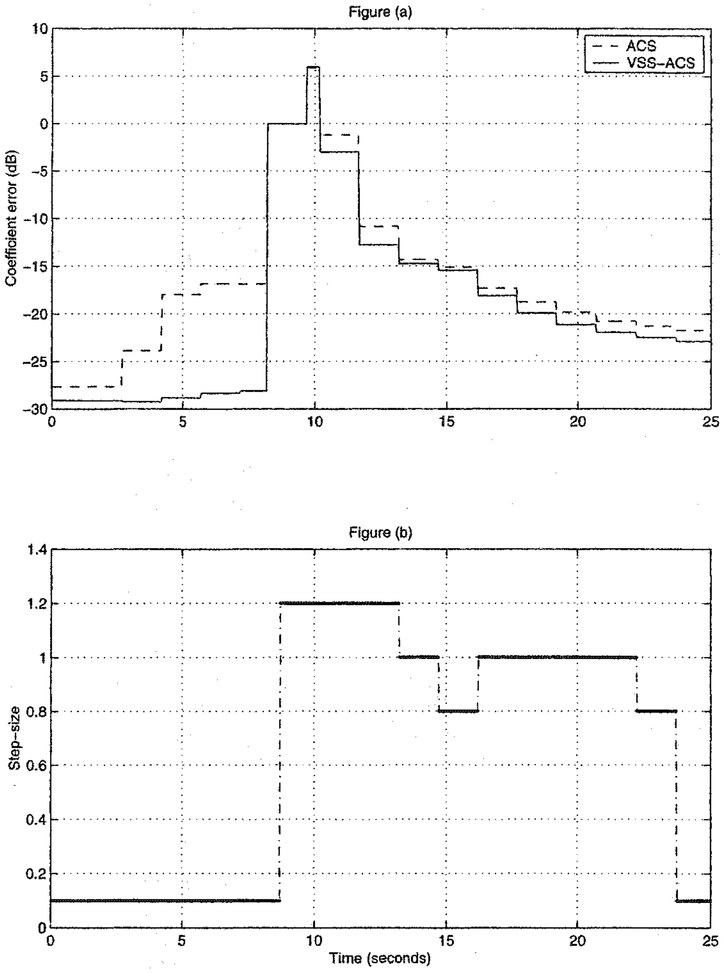

Fig. 4.9 (a) Coefficient errors versus time (acoustic echo path changed from $\mathbf{h}(n)$ to -$\mathbf{h}(n)$ at 8.2s) for ACS (dashed line) and VSS-ACS (solid line); (b) Corresponding step-sizes of VSS-ACS.

filter. Based on the study of the effects of the step-size on the ACS convergence behaviour, a new variable step-size ACS (VSS-ACS) algorithm was developed, where the value of the step-size is set dynamically by a finite state machine, so as to optimize algorithm performance in terms of convergence rate and misadjustment size. Furthermore, the proposed algorithm has a new adaptation scheme that improves the initial convergence rate when the network connection is created.

The proposed VSS-ACS algorithm is attractive in AEC applications because it significantly attenuates acoustic echo without the requirement of DT detection even in the presence of near-end speech and high levels of background noise. The computational complexity of the VSS-ACS algorithm is comparable to that of the standard LMS algorithm; this allows for low-cost real-time implementation with existing DSP technology. The results of computer experiments show that the new VSS-ACS algorithm outperforms the original ACS in terms of the superior acoustic echo suppression during DT periods and faster initial convergence rate.

In the presence of vocoders, the performance of the VSS-ACS algorithm would be severely degraded due to the nonlinearities of the echo path, just like other adaptive filtering algorithms. However, the VSS-ACS algorithm keeps adapting during the DT period, and this results in notably better echo suppression than the other algorithms. Moreover, in a nonlinear channel, the insufficient echo attenuation of the VSS-ACS algorithm may be compensated by a post-filter, which will be discussed in the following chapter.

Chapter 5

Post-filtering technique in AEC

A post-filter usually refers to an auxiliary filter in the last step of speech processing, especially in the fields of speech coding and speech enhancement. In the context of acoustic echo cancellation, a post-filter further attenuates the residual echo after the adaptive filter cancels part of the acoustic echo. In fact, integrating a post-filter in the AEC system is an effective way to suppress the acoustic echo to a satisfactory level when the echo path is nonlinear. This chapter discusses various post-filtering techniques and their combination with conventional adaptive filtering algorithms, which are suitable for use in the nonlinear channels.

5.1 Introduction

As pointed out earlier, the performance of the conventional AEC system is significantly degraded when speech codecs are used in the new generation digital networks [103]. So far, it is clear that a conventional acoustic echo canceller which has a linear structure cannot by itself achieve sufficient attenuation of the acoustic echo in the nonlinear channel.

Recently, several promising approaches have been proposed that combine a conventional AEC system with a post-filter to suppress the acoustic echo to a satisfactory level [104, 105, 106]. The basic idea for these algorithms is to employ a Wiener filter [107] or spectral subtraction [108] combined with a conventional acoustic echo estimator. As a matter of fact, Wiener filtering and spectral subtraction have been widely used in speech enhancement for decades [98, 50]. Figure 5.1 shows the main concept behind these techniques, which is to design an estimator that removes the unwanted component $\delta(n)$ from the contaminated

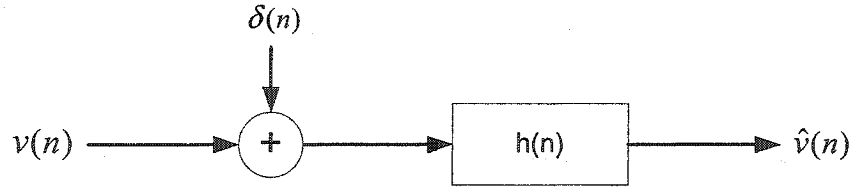

Fig. 5.1 Diagram of the estimator: $\hat{v}(n)$ is the estimate of $v(n)$ from the contaminated signal $v(n) + \delta(n)$.

signal $v(n) + \delta(n)$.

Although Figure 5.1 shows a traditional estimation model which can be traced back to half century ago [109], recent works focus on its application to new situations. In AEC, the near-end speech needs to be extracted from the mixed signal that consists of near-end speech, residual echo and background noise. Since the residual echo may be modelled as a non-stationary background noise, the estimation model is applicable to AEC. This chapter derives several algorithms for AEC based on the minimum mean-squared error (MMSE) criterion.

5.2 The Wiener-type post-filter

Figure 5.2 shows the post-filter configuration in AEC systems used over nonlinear channels. Suppose that the decoded microphone signal, denoted $d(n)$, consists of acoustic echo $y(n)$, near-end speech $v(n)$, and background noise $z(n)$, that is

$$d(n) = v(n) + y(n) + z(n). \tag{5.1}$$

Also let $\hat{y}(n)$ denote the estimated echo signal and let $e(n)$ denote the residual signal, computed as

$$e(n) = d(n) - \hat{y}(n). \tag{5.2}$$

Define the residual echo $\delta(n)$ as

$$\delta(n) = y(n) - \hat{y}(n). \tag{5.3}$$

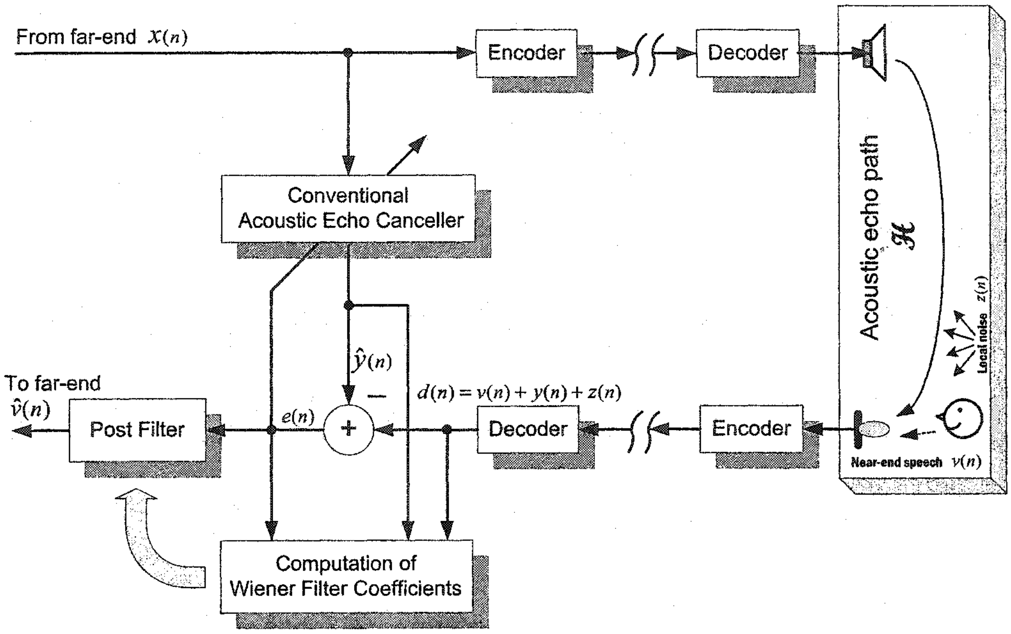

Fig. 5.2 Configuration of the post-filter in AEC over nonlinear channels.

Then (5.2) may be rewritten as

$$e(n) = v(n) + z(n) + \delta(n). \tag{5.4}$$

Now the estimation of $v(n)$, denoted $\hat{v}(n)$, can be obtained by an estimator that is usually implemented by an optimal filter. Because this filter is placed after the conventional acoustic echo canceller, it is also called the post-filter.

5.2.1 The Wiener optimal filter

The cost function $J(n)$ is defined as the mean square error between the near-end speech $v(n)$ and its estimator $\hat{v}(n)$:

$$J(n) = E[|v(n) - \hat{v}(n)|^2]. \tag{5.5}$$

Since $\hat{v}(n)$ is estimated from the output of the conventional acoustic echo canceller $e(n)$, i.e.,

$$\hat{v}(n) = e(n) * h(n), \tag{5.6}$$

minimizing the cost function (5.5) leads to the orthogonality condition [33]

$$E[[v(n) - \hat{v}(n)]e(n)] = 0. \tag{5.7}$$

Substituting (5.6) for (5.7), the Wiener-Hopf equations are obtained [33]:

$$r_{ee}(n) * h(n) = r_{ev}(n), \tag{5.8}$$

where $r_{ee}(n)$ is the autocorrelation function of $e(n)$, and $r_{ev}(n)$ is the cross-correlation function of $e(n)$ and $v(n)$.

If (5.8) is expressed in the frequency domain by a short-term Fourier transform (STFT), then the Wiener filter is given by

$$H(k;m) = \frac{S_{ev}(k;m)}{S_{ee}(k;m)}, \tag{5.9}$$

where $H(k;m)$ denotes the STFT of the filter impulse response $h(n)$ in (5.8), $k = 0, 1, \ldots, K-1$ is the frequency index, and $m = 1, 2, \ldots$ is the time index. In the above, $S_{ee}(k;m)$ denotes the power spectral density (PSD) of $v(n)$, and $S_{ev}(k;m)$ denotes the cross spectral density of $v(n)$ and $e(n)$, respectively defined as

$$\begin{aligned} S_{ee}(k;m) &= \mathrm{STFT}\{r_{ee}(n)\} & (5.10)\\ S_{ev}(k;m) &= \mathrm{STFT}\{r_{ev}(n)\}. & (5.11) \end{aligned}$$

Assuming that the near-end speech $v(n)$, the background noise $z(n)$ and the far-end speech $x(n)$ are mutually uncorrelated, we have

$$\begin{aligned} r_{ev}(n) &= E[e(l+n)v(l)] \\ &= r_{vv}(n) + r_{zv}(n) + r_{\delta v}(n) \\ &= r_{vv}(n), \end{aligned} \tag{5.12}$$

and

$$\begin{aligned} r_{ee}(n) &= E[e(l+n)e(l)] \\ &= r_{vv}(n) + r_{zz}(n) + r_{\delta\delta}(n), \end{aligned} \tag{5.13}$$

where $r_{vv}(n)$, $r_{zz}(n)$ and $r_{\delta\delta}(n)$ denote the autocorrelation functions of $v(n)$, $z(n)$ and $\delta(n)$, respectively. $r_{zv}(n)$ is the cross-correlation function of $z(n)$ and $v(n)$, and $r_{\delta v}(n)$ is the cross-correlation function of $\delta(n)$ and $v(n)$. Taking the STFT of (5.12) and (5.13), the Wiener filter in (5.9) becomes

$$\begin{aligned} H(k;m) &= \frac{S_{vv}(k;m)}{S_{vv}(k;m) + S_{\delta\delta}(k;m) + S_{zz}(k;m)} \\ &= \frac{1}{1 + \dfrac{S_{\delta\delta}(k;m)}{S_{vv}(k;m)} + \dfrac{S_{zz}(k;m)}{S_{vv}(k;m)}}. \end{aligned} \tag{5.14}$$

Three cases arise in using (5.14):

- *Only near-end speech is present.* No signal from the far-end leads to $S_{\delta\delta}(k;m) \approx 0$. Then this becomes a noise reduction case. If the SNR at the near-end is high, i.e., $S_{vv}(k;m) \gg S_{zz}(k;m)$, generally $H(k;m) \approx 1$. So there is almost no distortion in the near-end speech when it is sent to the far-end. If the SNR in the near-end is low, i.e., $S_{vv}(k;m) \ll S_{zz}(k;m)$, then the speech quality is improved by suppressing noise, where the attenuation depends on the SNR in each frequency bin.

- *Only far-end speech is present.* In this case, $S_{vv}(k;m) \approx 0$ so that $H(k;m) \to 0$. This leads to very strong attenuation of the residual echo and the noise. Ideally, the far-end user would not hear anything from the near-end due to the high attenuation. This could make the far-end user uncomfortable because he/she needs to obtain a feedback from the near-end, such as a low level background noise. In order to retain a constant level of natural sounding background noise in the output signal, different optimization objectives have been studied [106].

- *Both near-end and far-end speech are present.* This is the double-talk case. The attenuation of the residual echo in each frequency bin depends on the PSD ratio of the residual echo to the near-end speech. The larger the residual echo's power level compared to the near-end speech, the stronger the attenuation. However, the power level of the residual echo is normally lower than that of the near-end speech for a conventional acoustic echo canceller. Thus, the residual echo suppression may not be sufficient even if the masking effect is taken into consideration.

Because neither $S_{ev}(k;m)$ in (5.9) nor $S_{vv}(k;m)$ in (5.14) can be estimated in practice, $H(k;m)$ needs to be rewritten. From (5.1), first note that

$$\begin{aligned} r_{ev}(n) &= E[e(l+n)v(l)] \\ &= r_{ed}(n) - r_{ey}(n) - r_{ez}(n), \end{aligned} \tag{5.15}$$

where $r_{ed}(n)$, $r_{ey}(n)$ and $r_{ez}(n)$ are the cross-correlation functions of $e(n)$ and $d(n)$, $e(n)$ and $y(n)$, and $e(n)$ and $z(n)$, respectively. Since $E[e(n)y(n)] = 0$ for the conventional adaptive filter in steady-state [13], the term r_{ey} in (5.15) can be neglected. Under the assumption that $z(n)$ is uncorrelated to all signals in (5.4) but itself, it follows that

$$r_{ez}(n) = r_{zz}(n). \tag{5.16}$$

Therefore, the Wiener filter $H(k;m)$ (5.9) is rewritten as

$$H(k;m) = \frac{S_{ed}(k;m) - S_{zz}(k;m)}{S_{ee}(k;m)}. \tag{5.17}$$

If the noise signal $z(n)$ is stationary during a relatively long interval, then $S_{zz}(k;m)$ estimated during a speech silence period can be used when the speech (either near-end speech, echo signal, or both) is active until the next speech silence period arrives. This procedure needs a voice activity detector (VAD) to decide when $S_{zz}(k;m)$ should be estimated [107], but it is difficult to design a reliable VAD that operates under different circumstances. Furthermore, musical noise could be introduced by (5.17) [106] since its numerator is similar to the spectral subtraction method [108].

Since this work focuses on echo cancellation, the second term in the numerator of (5.17) may be discarded so that musical noise is avoided. Hence, the optimal filter (5.17) is approximated by the sub-optimal filter:

$$H(k;m) = \frac{S_{ed}(k;m)}{S_{ee}(k;m)}. \tag{5.18}$$

In practice, the PSDs in (5.18) are estimated recursively using

$$S_{ed}(k;m) = \gamma S_{ed}(k;m-1) + (1-\gamma)E(k;m)D^*(k;m), \tag{5.19}$$

$$S_{ee}(k;m) = \gamma S_{ee}(k;m-1) + (1-\gamma)|E(k;m)|^2, \tag{5.20}$$

where $E(k;m)$ is the STFT of $e(n)$ at block m, and $D(k;m)$ is the STFT of $d(n)$ at block m. The forgetting factor γ can be chosen around 0.8 at the sampling rate of 8 kHz.

In order to analyze the performance of the Wiener filter in (5.18), the equation is rewritten by using the relations in (5.1) and (5.4) and exploiting the 2nd order properties, resulting in

$$H(k;m) = \frac{S_{vv}(k;m) + S_{zz}(k;m)}{S_{vv}(k;m) + S_{\delta\delta}(k;m) + S_{zz}(k;m)}. \tag{5.21}$$

Suppose that the echo attenuation is 20 dB for a conventional acoustic echo canceller, and the echo-to-noise ratio (ENR) is 40 dB at the near-end, then (5.21) becomes

$$H(k;m) \simeq \frac{1 + 10^{-2}\eta}{1+\eta}, \tag{5.22}$$

where the PSD ratio is $\eta = S_{\delta\delta}(k;m)/S_{vv}(k;m)$.

Corresponding to (5.22), Figure 5.3 shows that the attenuation of the residual echo by the filter $H(k;m)$ increases monotonically with the PSD ratio η. For the situation discussed earlier where near-end speech dominates the microphone signal, i.e., $\eta \ll 1$, the residual echo attenuation achieved is the lowest. However, this is acceptable since the power level of the echo is low enough for the echo to be masked by the near-end speech. In the second situation where near-end speech is absent, i.e., $\eta = \infty$, the residual echo attenuation is very high at about 20 dB. Assuming that a typical conventional acoustic echo canceller has 20 dB echo suppression [48], the echo is inaudible for the far-end user with the total 40 dB echo suppression. In the third case of double-talk, the power level of the echo is comparable to that of the near-end speech, i.e., $\eta \approx 1$, the residual echo attenuation is very small. Therefore, the residual echo could be heard by the far-end user in this period.

5.2.2 The over-weighted Wiener filter

As discussed in the previous section, the Wiener filter in (5.18) cannot sufficiently suppress the residual echo when the power level of the latter is relatively high. This case happens

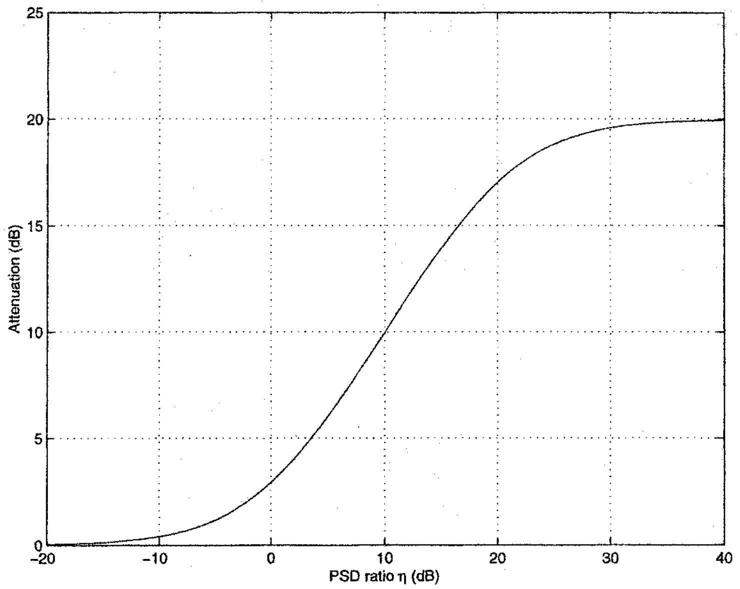

Fig. 5.3 Residual echo attenuation with filter $H(k;m)$ versus PSD ratio of the residual echo $\delta(n)$ to the near-end speech $v(n)$ (ENR=40dB).

with an imperfect acoustic echo canceller (e.g., a conventional linear adaptive filter used when the echo path is non-linear) especially during the double-talk. This section derives another optimal filter that can achieve better attenuation of the residual echo under certain constraints.

Referring to Figure 5.2, the output signal of the conventional acoustic echo canceller $e(n)$ is expressed in the frequency domain as

$$E(\omega) = V(\omega) + \Delta(\omega) + Z(\omega), \tag{5.23}$$

where $D(\omega)$, $V(\omega)$, $\Delta(\omega)$ and $Z(\omega)$ denote the discrete-time Fourier transforms of $e(n)$, $v(n)$, $\delta(n)$ and $z(n)$. Let $\hat{V}(\omega) = H(\omega)E(\omega)$ be a linear estimator of the near-end speech component $V(\omega)$, where $H(\omega)$ is a real-valued frequency weighting function. Making use of (5.23), the error signal associated with this estimator can be expanded as

$$\begin{aligned} \varepsilon(\omega) &= \hat{V}(\omega) - V(\omega) \\ &= [H(\omega) - 1]V(\omega) + H(\omega)\Delta(\omega) + H(\omega)Z(\omega), \end{aligned} \tag{5.24}$$

where the components $V(\omega)$, $\Delta(\omega)$ and $Z(\omega)$ are assumed to be mutually uncorrelated. Let $S_{\varepsilon\varepsilon}$ denote the PSD of the error signal $\varepsilon(\omega)$. It can be verified that

$$S_{\varepsilon\varepsilon} = [H(\omega) - 1]^2 S_{vv}(\omega) + H(\omega)^2 S_{\delta\delta}(\omega) + H(\omega)^2 S_{zz}(\omega), \tag{5.25}$$

where the first term represents the PSD of the signal distortion, the second term represents the PSD of the processed residual echo, and the third term represents the PSD of the processed residual noise. Let $J_v(\omega)$, $J_\delta(\omega)$ and $J_z(\omega)$ denote the three terms in (5.25), i.e.,

$$J_v(\omega) = [H(\omega) - 1]^2 S_{vv}(\omega), \tag{5.26}$$

$$J_\delta(\omega) = H(\omega)^2 S_{\delta\delta}(\omega), \tag{5.27}$$

$$J_z(\omega) = H(\omega)^2 S_{zz}(\omega). \tag{5.28}$$

Then the linear estimator of the near-end speech with constraints on the processed residual echo and the noise is obtained by

$$\begin{aligned} & \min_{H(\omega)} J_v(\omega) \\ \text{subject to :}\quad & J_\delta(\omega) \leq \alpha S_{\delta\delta}(\omega) \\ & J_z(\omega) \leq \beta S_{zz}(\omega), \end{aligned} \tag{5.29}$$

where $0 \leq \alpha, \beta \leq 1$. This estimator is derived with the aim of minimizing the near-end speech distortion while suppressing the processed residual echo and noise to a predefined level.

The optimal estimator in (5.29) can be found by using the Karush-Kuhn-Tucker necessary conditions for inequality constraints [110]. The Lagrangian function is given by

$$L[H(\omega), \mu_1, \mu_2] = J_v(\omega) + \mu_1 [J_\delta(\omega) - \alpha S_{\delta\delta}(\omega)] + \mu_2 [J_z(\omega) - \beta S_{zz}(\omega)], \tag{5.30}$$

where μ_1 and μ_2 are the Lagrange multipliers, and

$$\begin{aligned} \mu_1 [J_\delta(\omega) - \alpha S_{\delta\delta}(\omega)] = 0 \qquad & \text{for } \mu_1 \geq 0, \\ \mu_2 [J_z(\omega) - \beta S_{zz}(\omega)] = 0 \qquad & \text{for } \mu_2 \geq 0. \end{aligned} \tag{5.31}$$

From $\nabla_{H(\omega)}L[H(\omega), \mu_1, \mu_2] = 0$, we have

$$[H(\omega) - 1]^2 S_{vv}(\omega) + \mu_1 H(\omega) S_{\delta\delta}(\omega) + \mu_2 H(\omega) S_{zz}(\omega) = 0. \tag{5.32}$$

There are three feasible points for this optimization problem:

1. The first constraint in (5.29) is active and $\mu_2 = 0$. This is the case where the residual is suppressed and there is no constraint on noise reduction. Then the optimal estimator is

   $$H_{opt1}(\omega) = \frac{S_{vv}(\omega)}{S_{vv}(\omega) + \mu_1 S_{\delta\delta}(\omega)}, \tag{5.33}$$

   and the Lagrange multiplier μ_1 is

   $$\mu_1 = (\frac{1}{\sqrt{\alpha}} - 1)\frac{S_{vv}(\omega)}{S_{\delta\delta}(\omega)}. \tag{5.34}$$

   Note that although this estimator is optimized under the constraint of residual echo suppression, it may also suppress the background noise depending on the PSD ratio of the near-end speech to the residual echo.

2. The second constraint in (5.29) is active and $\mu_1 = 0$. This is a noise reduction case. The optimal estimator is

   $$H_{opt2}(\omega) = \frac{S_{vv}(\omega)}{S_{vv}(\omega) + \mu_2 S_{zz}(\omega)}, \tag{5.35}$$

   and the Lagrange multiplier μ_2 is

   $$\mu_1 = (\frac{1}{\sqrt{\beta}} - 1)\frac{S_{vv}(\omega)}{S_{zz}(\omega)}. \tag{5.36}$$

   The residual echo may also be suppressed by this estimator, and the attenuation depends on the PSD ratio of the near-end speech to background noise.

3. Both constraints in (5.29) are active. It can be verified that the only feasible point exists when $\alpha = \beta$. This means that the constraints are dependent. Hence, the

optimal estimator is given by

$$H_{opt3}(\omega) = \frac{S_{vv}(\omega)}{S_{vv}(\omega) + \mu_1 S_{\delta\delta}(\omega) + \mu_2 S_{zz}(\omega)}, \tag{5.37}$$

and the Lagrange multipliers μ_1 and μ_2 must satisfy

$$\alpha = \frac{S_{vv}^2(\omega)}{[S_{vv}(\omega) + \mu_1 S_{\delta\delta}(\omega) + \mu_2 S_{zz}(\omega)]^2}, \tag{5.38}$$

In this case, both the residual echo and the noise can be suppressed to a predetermined level.

Finally, each of the three estimators above is analyzed to see whether the stationary points so obtained correspond to global minima. This can be shown by evaluating the second derivative of the Lagrangian function:

$$\nabla^2_{H(\omega)H(\omega)} L[H(\omega), \mu_1, \mu_2] = 2S_{vv}(\omega) + 2\mu_1 S_{\delta\delta}(\omega) + 2\mu_2 S_{zz}(\omega). \tag{5.39}$$

Since $\nabla^2_{H(\omega)H(\omega)} L[H(\omega), \mu_1, \mu_2] \geq 0$, and the optimized function as well as the constraints are convex functions for all three cases, the filters are indeed optimal.

Among the above optimal filters, the third one ($H_{opt3}(\omega)$ in (5.37)), which suppresses the residual echo and the noise simultaneously, is the most interesting. Therefore, the rest of this section is exclusively focused on this estimator.

Due to a lack of prior knowledge of signals such as the near-end speech, the residual echo and the background noise, (5.37) must be put into a form that is more easily amenable to practical implementation. Let $S_{ev}(\omega)$ and $S_{vv}(\omega)$ denote the Fourier Transform (FT) of $r_{ev}(n)$ and $r_{vv}(n)$, respectively. From (5.12), we have

$$S_{ev}(\omega) = S_{vv}(\omega). \tag{5.40}$$

Similarly, from (5.15) we have

$$S_{ev}(\omega) = S_{ed}(\omega) - S_{ez}(\omega) - S_{ey}(\omega), \tag{5.41}$$

where $S_{ed}(\omega)$, $S_{ez}(\omega)$ and $S_{ey}(\omega)$ are the FT of $r_{ed}(n)$, $r_{ez}(n)$ and $r_{ey}(n)$, respectively.

Assume that $v(n)$, $z(n)$, $\delta(n)$ and $y(n)$ are mutually uncorrelated; then it is easy to show that

$$S_{ez}(\omega) = S_{zz}(\omega) \tag{5.42}$$

$$S_{ey}(\omega) = 0. \tag{5.43}$$

Finally, it is obtained that

$$S_{vv}(\omega) = S_{ed}(\omega) - S_{zz}(\omega). \tag{5.44}$$

Therefore, (5.37) is rewritten as follows (where, in the sequel, $H(\omega)$ stands for $H_{opt3}(\omega)$ for simplicity):

$$H(\omega) = \frac{S_{ed}(\omega) - S_{zz}(\omega)}{S_{ee}(\omega) + (\mu_1 - 1)S_{\delta\delta}(\omega) + (\mu_2 - 1)S_{zz}(\omega)}. \tag{5.45}$$

Unfortunately, the difficulty in estimating $S_{\delta\delta}(\omega)$ and $S_{zz}(\omega)$ still remains in (5.45). However, since the main goal is to attain a high echo attenuation, some reasonable approximations can be made.

First, if the second term in the numerator of (5.45) (i.e., $S_{zz}(\omega)$) is neglected, then the attenuation is reduced by only a few decibels, since normally $S_{vv}(\omega) \geq S_{zz}(\omega)$ when near-end speech is present. When near-end speech is absent, the effect of neglecting $S_{zz}(\omega)$ in (5.45) is noticeable, but some compensation can be obtained by appropriate modification of the denominator, as explained below.

Second, the denominator in (5.45) can be simplified by setting $\mu_2 = 1$, since noise reduction is not the focus of this work. However, so far we still need the exact information of the processed residual echo $\delta(n)$, which is almost impossible to obtain, for the computation of $S_{\delta\delta}$. Here, we propose to use the estimated echo signal $\hat{y}(n)$ provided by the conventional acoustic echo canceller (see Figure 5.2) to approximately estimate $S_{\delta\delta}$.

The average magnitude spectra of $\delta(n)$ and $\hat{y}(n)$ are compared in Figure 5.4 for two different speech frames using the VSS-ACS algorithm developed in Chapter 4, where speech codecs (i.e., G.729) were present along the echo path. The shapes of the magnitude spectra are alike for both cases. A possible explanation of this phenomenon is discussed below.

Nonlinearity usually brings in new frequency components. Because the spectrum of speech almost occupies the entire 4 kHz bandwidth, the contribution of the system nonlinearities over that range distorts the spectrum by increasing or decreasing the existing frequency components. According to the properties of the vocoder, this spectrum distor-

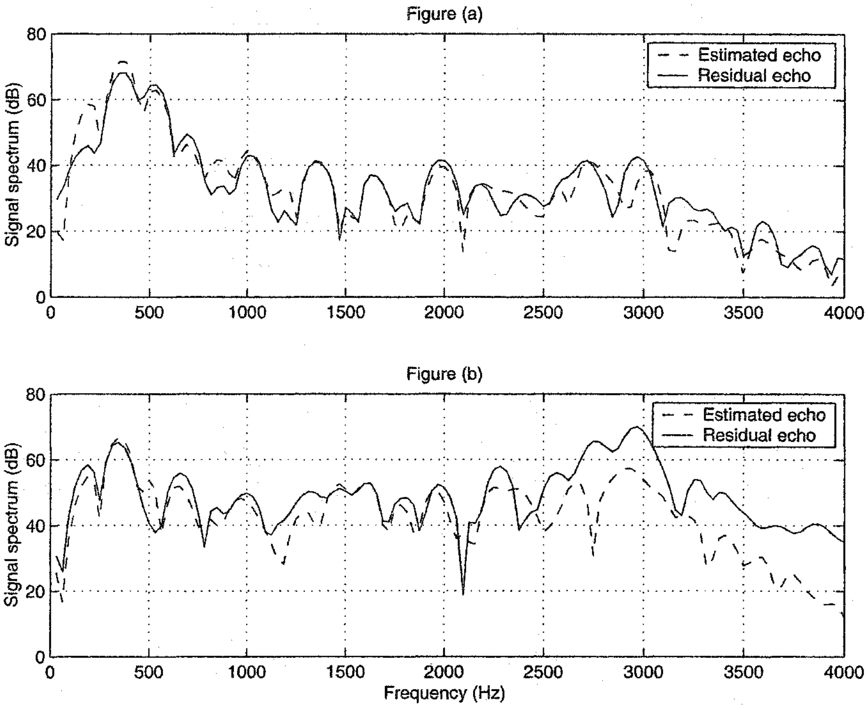

Fig. 5.4 Signal spectra of estimated echo $\hat{y}(n)$ and residual echo $\delta(n)$: (a) vowel-dominated frame, (b) fricative-dominated frame.

tion, especially in the regions of higher energy, is not severe enough to affect auditory perception. The distortion also varies with acoustic phonetics: there is less distortion in the vowel-dominated frame (see Fig. 5.4(a)) than in the fricative-dominated frame (see Fig. 5.4(b)).

Based on these considerations, it appears reasonable to approximate $S_{\delta\delta}(k;m)$ by a scaled version of $S_{\hat{y}\hat{y}}(k;m)$. Hence, the post-filter (5.45) becomes

$$H(\omega) = \frac{S_{ed}(\omega)}{S_{ee}(\omega) + \alpha S_{\hat{y}\hat{y}}(\omega)}, \tag{5.46}$$

where $\mu_1 - 1$ in (5.45) has been replaced by α without loss of generality. In practice, the STFT may be employed to estimate the PSDs in (5.46):

$$S_{ed}(k;m) = \gamma S_{ed}(k,m-1) + (1-\gamma)E(k;m)D^*(k,m), \tag{5.47}$$

$$S_{ee}(k;m) = \gamma S_{ee}(k,m-1) + (1-\gamma)|E(k;m)|^2, \tag{5.48}$$

$$S_{\hat{y}\hat{y}}(k;m) = \gamma S_{\hat{y}\hat{y}}(k,m-1) + (1-\gamma)|\hat{Y}(k;m)|^2. \tag{5.49}$$

As before, $E(k;m)$, $D(k;m)$ and $\hat{Y}(k;m)$ denote the STFTs of $e(n)$, $d(n)$ and $\hat{y}(n)$ at block m, respectively, while k is the frequency index. The forgetting factor γ is chosen to be around 0.7 in the experiment. The over-weighting parameter α in (5.46) can be chosen from a wide range, say from 5 to 40, depending on the requirement of echo suppression and the tolerance of near-end speech distortion. Note that the over-weighted Wiener filter (5.46) reduces to the conventional Wiener filter (5.18) when α is set to zero.

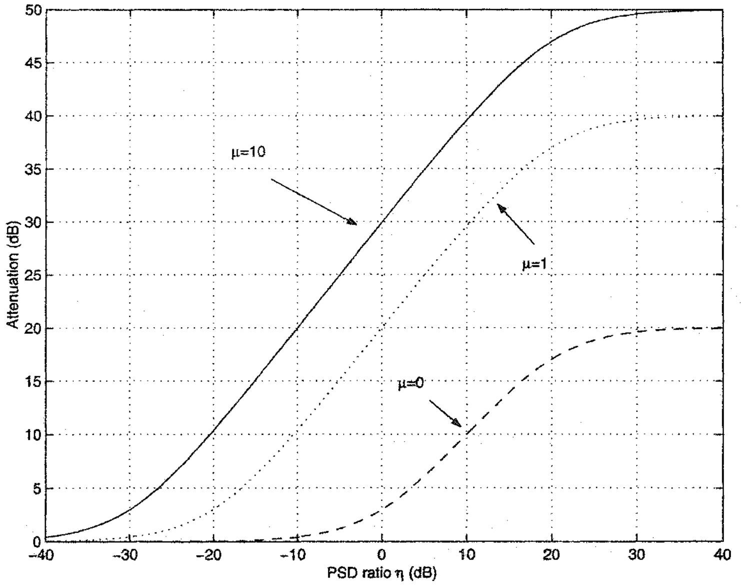

Fig. 5.5 Residual echo attenuation versus the PSD ratio of the residual echo $\delta(n)$ to the near-end speech $v(n)$ for an over-weighted Wiener filter.

Here, (5.46) is rewritten as follows for its performance analysis:

$$H(\omega) = \frac{S_{vv}(\omega) + S_{zz}(\omega)}{S_{vv}(\omega) + S_{\delta\delta}(\omega) + S_{zz}(\omega) + \alpha S_{\hat{y}\hat{y}}(\omega)}. \tag{5.50}$$

Suppose that the attenuation of the echo is 20 dB (i.e., $S_{yy}(\omega) = 10^2 S_{\delta\delta}(\omega)$) for a conventional acoustic echo canceller, and ENR=40 dB (i.e., $S_{yy}(\omega) = 10^4 S_{zz}(\omega)$) at the near-end, (5.50) becomes

$$H(k;m) \simeq \frac{1+10^{-2}\eta}{1+10^{2}\alpha\eta}, \tag{5.51}$$

where the PSD ratio $\eta = S_{\delta\delta}(k;m)/S_{vv}(k;m)$.

The attenuation of the residual echo with the over-weighted Wiener filter (5.46) is shown in Figure 5.5. Compared with the conventional Wiener filter (was $\mu = 0$ shown with dashed line), the over-weighted Wiener filter (shown in solid and dotted lines) suppresses much more residual echo.

5.2.3 Application in AEC systems

In a nonlinear channel, a post-filter works together with a linear adaptive filter to suppress the acoustic echo to a satisfactory level. The adaptive filter performs the primary echo attenuation, as well as providing an estimated echo for the post-filter. Among the possible combinations, the VSS-ACS algorithm combined with a over-weighted Wiener filter is a very attractive one. This is because VSS-ACS is robust in the presence of high disturbance such as double-talk. The main weakness of this algorithm is insufficient acoustic echo suppression, especially when codecs are present along the echo path. This weakness can be compensated by the post-filter. Figure 5.6 illustrates the resulting combined cross-spectral and post-filtering (CSP) algorithm over a non-linear channel in AEC.

In this work, the CSP is implemented in the frequency domain where FFT and IFFT are used for computational efficiency. The indices k and m in Figure 5.6 denote the frequency bins and the index of the data blocks in the time domain. In order to avoid producing noise during the conversion between time domain and frequency domain, the overlap-and-add technique is employed where the input data is segmented into blocks with 50% overlap. The analysis and synthesis windows are Hanning and rectangular windows, respectively, with the length of the windows set to 300 samples (corresponding to 37.5 ms at an 8 kHz sampling rate). In our experiment, the forgetting factor γ in (5.47)-(5.49) is set to 0.7, and the overlap ratio of the VSS-ACS algorithm is 0.5 (i.e., $Q = M/2$), where $M = 40$ (see Chapter 4).

The vocoders used in the simulations (both in the upper and lower branches of Figure 5.6) are software implementations of G.729 [24]. The microphone signal consists of a white noise that simulates background noise, and a speech signal with comparable power as the acoustic echo that simulates near-end speech. The ENR in the near-end was set to

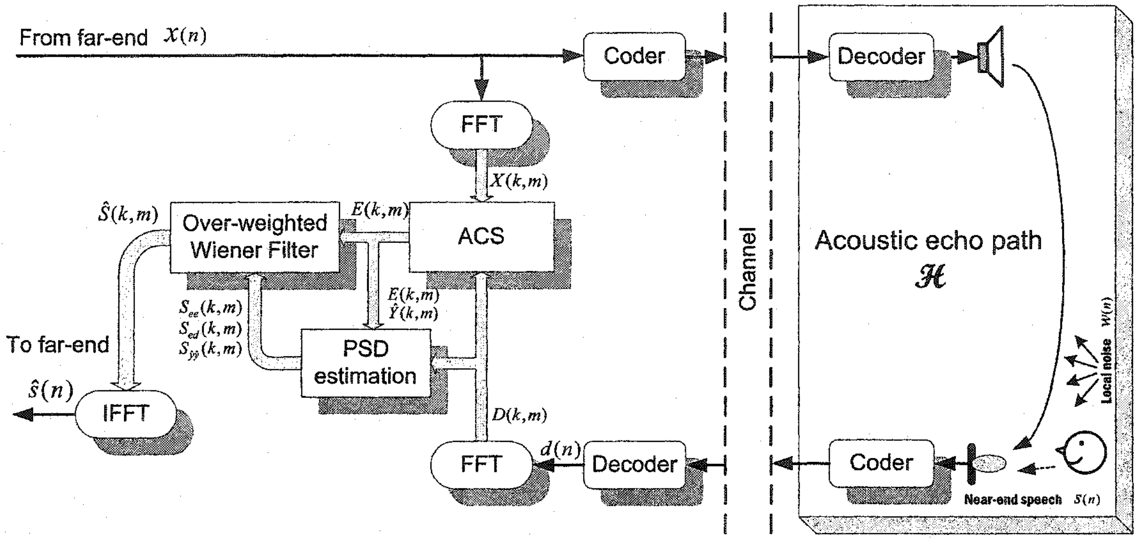

Fig. 5.6 The diagram of the CSP algorithm in AEC.

10 dB to simulate a noisy background.

Figure 5.7(a) shows the microphone signal when double-talk occurs between time 4.5 s and 6.5 s. The error signal produced by the VSS-ACS is shown in Figure 5.7(b). Clearly, the level of echo suppression achieved by this algorithm over the non-linear channel is not sufficient. In comparison, the CSP achieves a much higher echo attenuation as shown in Figure 5.7(c). The original near-end speech signal is plotted in Figure 5.7(d) for reference. The spectrograms in Figure 5.8(d) also demonstrate the superiority of CSP over VSS-ACS in terms of both echo suppression and background noise reduction.

Although the general approach of combining an adaptive echo canceller with noise reduction algorithms, e.g., [111], has been proposed for a while, it is not widely used because it introduces distortion in the near-end speech. This distortion is evidenced by comparing the reconstructed signal in Figure 5.7(c) with the original speech in Figure 5.7(d). Furthermore, a conventional acoustic echo canceller is able to suppress the acoustic echo to an acceptable level in conventional telephone networks, so that a post-filter is not necessary. However, when vocoders are present, a slight speech distortion is acceptable because the vocoders themselves already distort the speech signal.

Based on experimental results and informal listening tests, it is concluded that the CSP algorithm is successful in AEC over non-linear channels. It achieves notable echo

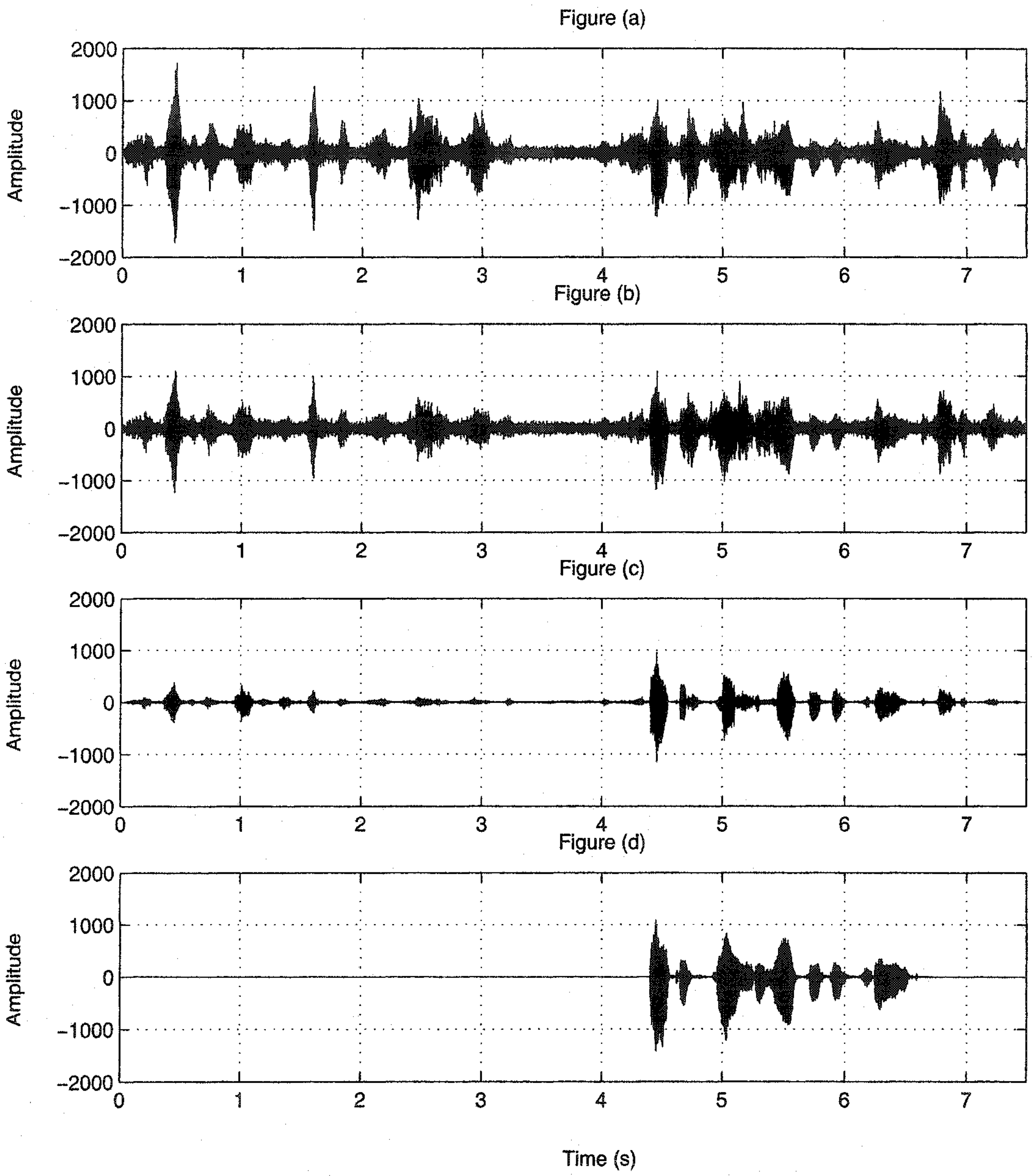

Fig. 5.7 Signal waveforms in the double-talk situation (ENR = 10 dB): (a) microphone signal, including acoustic echo and near-end speech; (b) error signal for the AEC using ACS only; (c) error signal for the AEC using ACS combined with a post-filter; (d) original near-end speech.

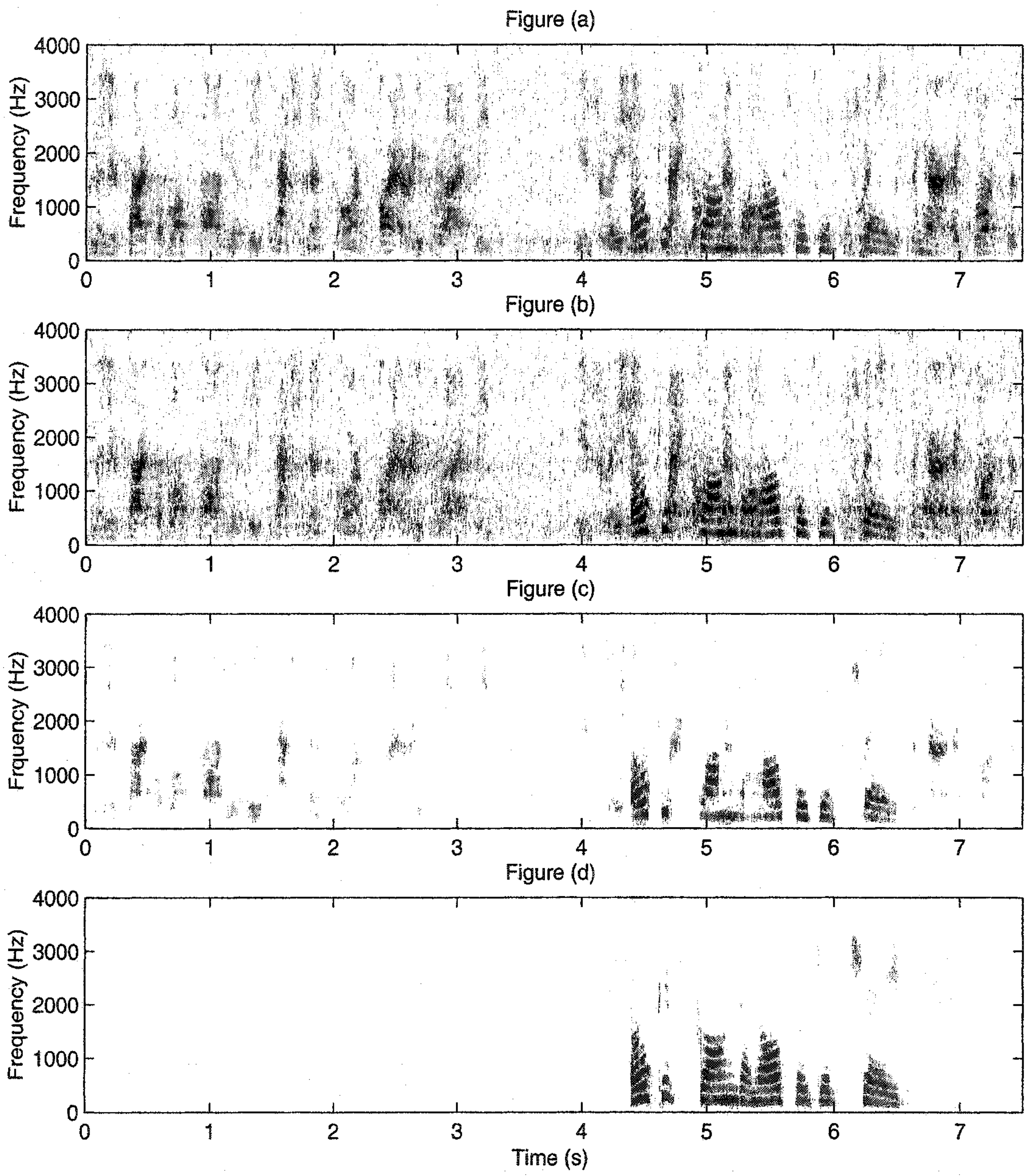

Fig. 5.8 Spectrograms in the double-talk situation (ENR = 10 dB): (a) microphone signal, including acoustic echo and near-end speech; (b) error signal for the AEC using ACS only; (c) error signal for the AEC using ACS combined with a post-filter; (d) original near-end speech.

suppression and is robust to local disturbances, compared to earlier approaches such as [105]. Such robustness is particularly useful in the double-talk situation, since neither a double-talk detector nor voice activity detection is required.

5.3 The spectral subtraction technique in echo suppression

The spectral subtraction technique was developed for speech enhancement [108]. The principle of this technique is to convert both the noisy speech $e(n)$ and an estimate of the noise $z(n)$ into the frequency domain using Fourier transforms, yielding $E(\omega)$ and $Z(\omega)$. The magnitudes of their Fourier transforms are then subtracted to obtain the magnitude of the enhanced speech, i.e.,

$$|\hat{V}(\omega)| = |E(\omega)| - \kappa|Z(\omega)|, \tag{5.52}$$

where $\kappa \geq 1$ is an overestimation factor used to improve noise reduction. Combined with the phase of $E(\omega)$, $\hat{V}(\omega)$ is converted back into the time domain. Since the frequencies where $|\hat{V}(\omega)|$ in (5.52) is negative cannot be recovered, $|\hat{V}(\omega)| = 0$ should be used for those values. A more general form of spectral subtraction is given by [98]

$$\hat{V}(\omega) = \left[|E(\omega)|^{\alpha} - \kappa|Z(\omega)|^{\alpha}\right]^{1/\alpha} e^{j\varphi_{E(\omega)}}, \tag{5.53}$$

where the noisy speech phase $\varphi_{E(\omega)}$ is used as the phase of the enhanced signal. α controls the amount of the noise reduction, and κ controls the amount of speech distortion. Usually, α is set to 1 or 2. When $\alpha = 1$, (5.53) corresponds to the basic linear spectral subtraction; when $\alpha = 2$, it corresponds to the power spectral subtraction.

Equation (5.53) can be rearranged as

$$\hat{V}(\omega) = E(\omega)\left[1 - \kappa\left|\frac{Z(\omega)}{E(\omega)}\right|^{\alpha}\right]^{1/\alpha}, \tag{5.54}$$

or equivalently,

$$\hat{V}(\omega) = E(\omega)H(\omega), \tag{5.55}$$

with

$$H(\omega) = \left[1 - \kappa \left|\frac{Z(\omega)}{E(\omega)}\right|^{\alpha}\right]^{1/\alpha}. \tag{5.56}$$

In (5.56), $H(\omega)$ is a signal dependent filter which reduces the noise power in the noisy signal to improve speech quality.

As previously discussed, the residual echo may be regarded as a noise signal; so the filter in (5.56) can be used in AEC to further attenuate residual echo, if we let $E(\omega)$ denote the FT of the residual signal, and let $Z(\omega)$ represent the FT of the unwanted signal (i.e., residual echo and noise). There are many variants for this corresponding to different approximations.

For instance, let $\alpha = 2$, and substitute $\Delta(\omega) + Z(\omega)$ for $Z(\omega)$ in the general form of the filter (5.56), where $\Delta(\omega)$ represents the Fourier transform of the processed residual echo signal $\delta(n)$. Then the alternative form is obtained as

$$H(\omega) = \sqrt{1 - \kappa \left|\frac{\Delta(\omega) + Z(\omega)}{E(\omega)}\right|^{2}}. \tag{5.57}$$

Expressing (5.57) by using the first-order Taylor expansion in terms of the PSDs, and introducing a spectral floor [105], $H(\omega)$ becomes

$$H(\omega) = \begin{cases} 1 - \sqrt{\dfrac{\kappa S_{\delta\delta}(\omega) + \kappa S_{zz}(\omega)}{S_{ee}(\omega)}} & H(\omega) > \beta_{\omega} \\ \beta_{\omega} & \text{otherwise.} \end{cases} \tag{5.58}$$

The purpose of the spectral floor β_{ω} is to limit the attenuation introduced by the filter $H(\omega)$ to a small positive value in order to preserve speech quality [112].

The spectral subtraction in AEC, which is considered a post-filter, is expressed in (5.58). The performance of this filter is shown in Figure 5.9, where κ is set to one. The echo attenuation curves of the conventional Wiener filter and the over-weighted Wiener filter are plotted as well for comparison. The simulation results indicate that the behaviour of the spectral subtraction in AEC is similar to the over-weighted Wiener filter with a small α (about one).

An alternative way to use the spectral subtraction method (5.54) in AEC is to estimate the near-end speech $\hat{v}(n)$ from the microphone signal $d(n)$ instead of from the output of

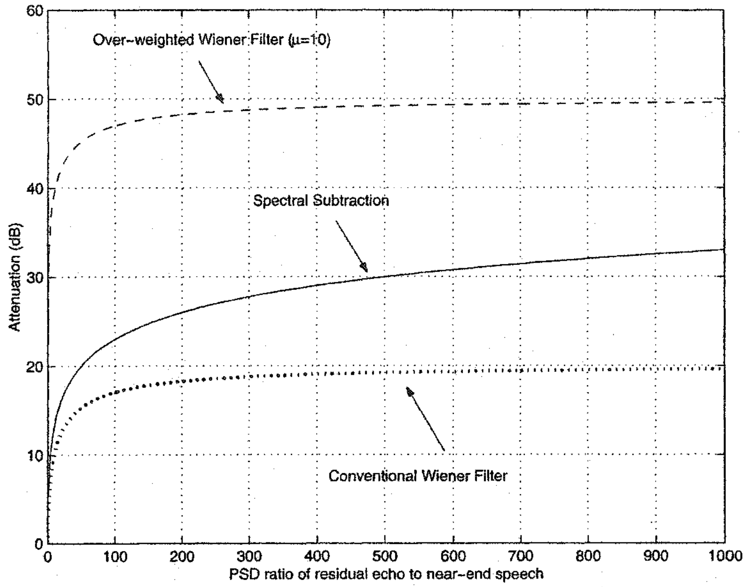

Fig. 5.9 Residual echo attenuation versus the PSD ratio of processed residual echo $\delta(n)$ to the near-end speech $v(n)$ for Spectral Subtraction (solid), over-weighted Wiener filter (dash) and conventional Wiener filter (dot).

the conventional acoustic echo canceller $e(n)$ (see Figure 5.2). The general form of spectral subtraction corresponding to (5.54) can be written as [113]

$$S(\omega) = D(\omega)\left[1 - \kappa\left|\frac{\hat{Y}(\omega)}{D(\omega)}\right|^{\alpha}\right]^{1/\alpha}, \tag{5.59}$$

where $\hat{Y}(\omega)$ and $D(\omega)$ are the Fourier transforms of the estimated echo $\hat{y}(n)$ and the microphone signal $d(n)$. Expanding (5.59) and expressing it in terms of the signals' PSDs, we have

$$\hat{V}(\omega) = [E(\omega) + \hat{Y}(\omega)]\left[1 - \kappa\left(\frac{S_{\hat{y}\hat{y}}(\omega)}{S_{vv}(\omega) + S_{yy}(\omega) + S_{zz}(\omega)}\right)^{\alpha/2}\right]^{1/\alpha}, \tag{5.60}$$

or equivalently,

$$\begin{aligned}\hat{V}(\omega) &= [E(\omega) + \hat{Y}(\omega)]H(\omega) \\ &= E(\omega)H(\omega) + \hat{Y}(\omega)H(\omega), \end{aligned} \tag{5.61}$$

with

$$H(\omega) = \left[1 - \kappa\left(\frac{S_{\hat{y}\hat{y}}(\omega)}{S_{vv}(\omega) + S_{yy}(\omega) + S_{zz}(\omega)}\right)^{\alpha/2}\right]^{1/\alpha}. \tag{5.62}$$

The attenuation curves of $H(\omega)$ in (5.62) for different values of α are plotted in Figure 5.10, where κ is set to one. The results suggest that a smaller α leads to greater echo attenuation.

Note that due to the existence of a second term in (5.61) (i.e., $\hat{Y}(\omega)H(\omega)$), the behaviour of the spectral subtraction approach of (5.60) is significantly different from that in (5.55). Because this term is approximately proportional to the echo signal, echo suppression is significantly degraded. Indeed, our informal subjective experiments suggest that this approach is the least attractive in terms of echo suppression and signal distortion among the post-filters discussed in this chapter.

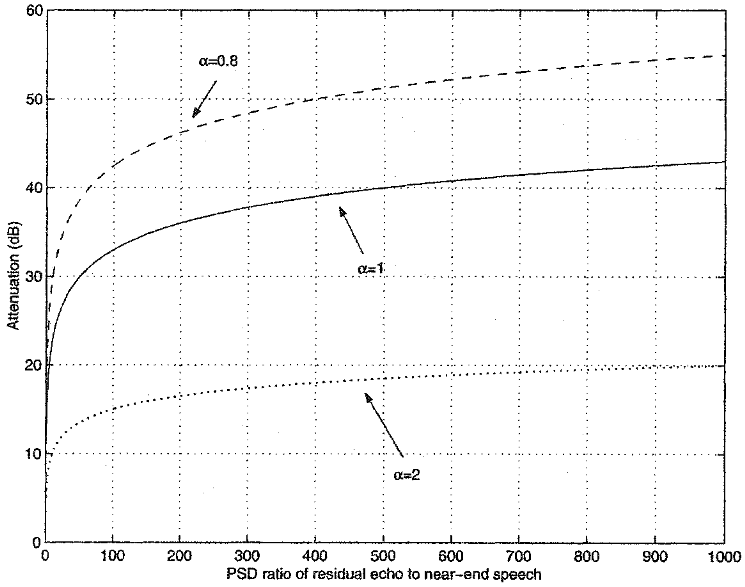

Fig. 5.10 The attenuation of the filter $H(\omega)$ in (5.62) versus the PSD ratio of residual echo to the near-end speech for spectral subtraction in AEC.

5.4 Subspace method

The subspace approach has been widely used in signal processing, including applications in signal enhancement [114, 115]. Based on the assumption that signal vectors lie in a subspace of the Euclidean space of the noisy signal, whereas the white noise vectors occupy the entire space, the vector space of the noisy signal can be decomposed into a signal plus noise subspace and a noise subspace [114]. Removing the noise subspace and extracting the signal from the signal plus noise subspace result in an enhanced signal. In the context of AEC, the microphone signal is decomposed into an echo subspace and a near-end signal plus echo (mixed) subspace. The acoustic echo is suppressed by eliminating the echo subspace and attenuating the non-signal components in the mixed subspace. This section presents an algorithm for subspace echo suppression [29]. The structure of the subspace processing is illustrated in Fig. 5.11.

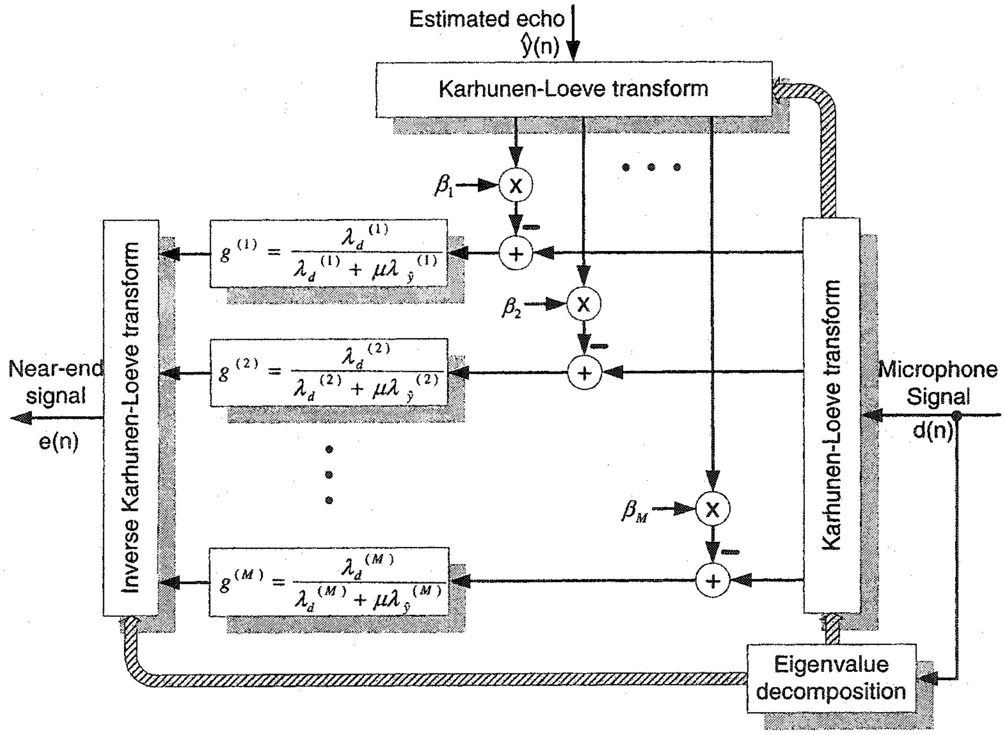
Estimated echo
$\hat{y}(n)$
Karhunen-Loeve transform
β_1
β_2
β_M
$g^{(1)} = \frac{\lambda_d^{(1)}}{\lambda_d^{(1)} + \mu\lambda_{\hat{y}}^{(1)}}$
$g^{(2)} = \frac{\lambda_d^{(2)}}{\lambda_d^{(2)} + \mu\lambda_{\hat{y}}^{(2)}}$
$g^{(M)} = \frac{\lambda_d^{(M)}}{\lambda_d^{(M)} + \mu\lambda_{\hat{y}}^{(M)}}$
Near-end signal
e(n)
Inverse Karhunen-Loeve transform
Karhunen-Loeve transform
Microphone Signal
d(n)
Eigenvalue decomposition

Fig. 5.11 The subspace processing for AEC.

5.4.1 Echo attenuation in subspace

a) Karhunen-Loeve transform (KLT)

The microphone signal in Figure 5.14 is expressed as a K-dimensional vector, defined by

$$\mathbf{d}(n) = [d(n), d(n-1), \ldots, d(n-K+1)]^H, \tag{5.63}$$

and

$$\mathbf{d}(n) = \mathbf{y}(n) + \mathbf{v}(n) + \mathbf{z}(n), \tag{5.64}$$

where $\mathbf{y}(n)$ is the echo signal vector, $\mathbf{v}(n)$ is the near-end speech signal vector, and $\mathbf{z}(n)$ is the background noise vector. Under the assumption that the signals $\mathbf{y}(n)$, $\mathbf{v}(n)$ and $\mathbf{z}(n)$ are mutually uncorrelated, the covariance matrix of $\mathbf{d}(n)$ is obtained as

$$\begin{aligned}\mathbf{R}_d(n) &= E[\mathbf{d}(n)\mathbf{d}^H(n)] \\ &= \mathbf{R}_y(n) + \mathbf{R}_v(n) + \mathbf{R}_z(n),\end{aligned} \tag{5.65}$$

where $\mathbf{R}_y(n)$, $\mathbf{R}_v(n)$ and $\mathbf{R}_z(n)$ are the covariance matrices of $\mathbf{y}(n)$, $\mathbf{v}(n)$ and $\mathbf{z}(n)$, respectively. $E[\cdot]$ denotes the expectation operator. The eigendecomposition of $\mathbf{R}_d(n)$ can be written as

$$\mathbf{R}_d(n) = \mathbf{Q}(n)\mathbf{\Lambda}_d(n)\mathbf{Q}(n)^H, \tag{5.66}$$

where

$$\mathbf{Q}(n) = [\mathbf{q}_1(n), \mathbf{q}_2(n), \ldots, \mathbf{q}_K(n)] \tag{5.67}$$

is an orthonormal matrix of eigenvectors of $\mathbf{R}_d(n)$, and

$$\mathbf{\Lambda}_d(n) = \text{diag}\{\lambda_d^{(1)}(n), \lambda_d^{(2)}(n), \ldots \lambda_d^{(K)}(n)\} \tag{5.68}$$

is a diagonal matrix of eigenvalues of $\mathbf{R}_d(n)$, with the diagonal elements in non-increasing order, i.e.,

$$\lambda_d^{(1)}(n) \geq \lambda_d^{(2)}(n) \geq \ldots \geq \lambda_d^{(K)}(n). \tag{5.69}$$

Hence, $\mathbf{Q}(n)^H\mathbf{d}(n)$ is the KLT of $\mathbf{d}(n)$, which projects the microphone signal $\mathbf{d}(n)$ into the noisy signal basis vectors $\mathbf{q}_i(n)$, $i = 1, 2, \ldots, K$. Similarly, the estimated echo signal $\hat{y}(n)$ from the conventional acoustic echo canceller can be decomposed into the same subspace by KLT, resulting in $\mathbf{Q}(n)^H\hat{\mathbf{y}}(n)$.

b) Echo subtraction in KLT domain

The estimated echo and the microphone signal are decomposed into the subspace by KLT, resulting in $\mathbf{Q}(n)^H\hat{\mathbf{y}}(n)$ and $\mathbf{Q}(n)^H\mathbf{d}(n)$. Subtracting the estimated echo from the microphone signal, the residual signal in the transform domain is obtained as

$$\mathbf{Q}(n)^H\boldsymbol{\epsilon}(n) = \mathbf{Q}(n)^H\mathbf{d}(n) - \boldsymbol{\beta}\mathbf{Q}(n)^H\hat{\mathbf{y}}(n), \tag{5.70}$$

where the underestimation matrix $\boldsymbol{\beta} = \mathrm{diag}(\beta_1, \beta_2, \ldots, \beta_K)$, $\beta_m \in [0,1]$, $m = 1, 2, \ldots, K$, is introduced to reduce the distortion of the near-end speech signal. As mentioned earlier, $\hat{y}(n)$ is not an ideal estimator of $y(n)$ due to loudspeaker non-linearities. Thus, coefficients β_m that are less than one can reduce the effect of the estimation error, which is serious during the double-talk when the adaptation of the AEC device is stopped.

Hence, the residual signal $\boldsymbol{\epsilon}(n)$ in the time domain is

$$\boldsymbol{\epsilon}(n) = \mathbf{v}(n) + \boldsymbol{\delta}_y(n) + \mathbf{z}(n), \tag{5.71}$$

where $\boldsymbol{\delta}_y(n)$ denotes the residual acoustic echo in $\boldsymbol{\epsilon}(n)$, and it is defined by

$$\boldsymbol{\delta}_y(n) = \mathbf{y}(n) - \mathbf{Q}(n)\boldsymbol{\beta}\mathbf{Q}^H(n)\hat{\mathbf{y}}(n). \tag{5.72}$$

c) Echo suppression filter

Let $\mathbf{H}(n)$ be a $K \times K$ matrix, which is the echo suppression filter, and let $\hat{\mathbf{v}}(n) = \mathbf{H}(n)\boldsymbol{\epsilon}(n)$ be the estimator of the near-end speech signal $\mathbf{v}(n)$; then the estimation error $\mathbf{e}_v(n)$ is written as

$$\begin{aligned} \mathbf{e}_v(n) &= \hat{\mathbf{v}}(n) - \mathbf{v}(n) \\ &= [\mathbf{H}(n) - \mathbf{I}]\mathbf{v}(n) + \mathbf{H}(n)[\boldsymbol{\delta}_y(n) + \mathbf{z}(n)], \end{aligned} \tag{5.73}$$

where the first term in (5.73) is the distortion of the near-end speech, and the second term is the further suppression of the echo and the reduction of the background noise. The ideal case is $\mathbf{e}_v(n) = \mathbf{0}$, which means both terms in (5.73) are $\mathbf{0}$. Because the signals $\mathbf{v}(n)$, $\boldsymbol{\delta}_y(n)$ and $\mathbf{z}(n)$ may not be $\mathbf{0}$, the condition $\mathbf{e}_v(n) = \mathbf{0}$ requires that $\mathbf{H}(n) - \mathbf{I} = \mathbf{0}$ and $\mathbf{H}(n) = \mathbf{0}$ simultaneously. Thus, it is impossible to attenuate the echo without any near-end speech distortion.

In this thesis, we propose to minimize the speech distortion in terms of mean-squared error under the constraints of suppressing the acoustic echo and the background noise to a certain level. By employing Kuhn-Tucker necessary conditions [110] and using a procedure similar to that in 5.2.2, the optimal filter is obtained as

$$\mathbf{H}_{opt}(n) = \mathbf{Q}(n)\mathbf{G}(n)\mathbf{Q}^H(n) \tag{5.74}$$

where $\mathbf{G}(n) = \mathrm{diag}\{g^{(1)}(n), g^{(2)}(n), \ldots, g^{(K)}(n)\}$ is a diagonal matrix with

$$g^{(m)} = \frac{\lambda_v^{(m)}}{\lambda_v^{(m)} + \mu[\lambda_{\delta_y}^{(m)} + \lambda_z^{(m)}]} \tag{5.75}$$

where μ is the Lagrange multiplier. In the derivation of (5.74)-(5.75), the approximation has been made that the off-diagonal elements in matrix $\mathbf{Q}(n)\mathbf{R}_v(n)\mathbf{Q}^H(n)$ can be neglected. $\lambda_s^{(m)}$ represent the diagonal elements of $\mathbf{Q}(n)\mathbf{R}_v(n)\mathbf{Q}^H(n)$. Similarly, $\lambda_{\delta_y}^{(m)}$ are the diagonal elements of $\mathbf{Q}(n)\mathbf{R}_{\delta_y}(n)\mathbf{Q}^H(n)$, and $\lambda_z^{(m)}$ are the diagonal elements of $\mathbf{Q}(n)\mathbf{R}_z(n)\mathbf{Q}^H(n)$.

Unfortunately, it is difficult to apply (5.75) in practice, because $\mathbf{R}_v(n)$ and $\mathbf{R}_{\delta_y}(n)$ must be found in order to compute $\lambda_v^{(m)}$ and $\lambda_{\delta_y}^{(m)}$. A voice activity detector is also needed to determine $\mathbf{R}_z(n)$ for $\lambda_z^{(m)}$. Let $\mathbf{R}_{\hat{y}}(n)$ be the covariance matrix of $\hat{\mathbf{y}}(n)$, and $\lambda_{\hat{y}}^{(m)}$ be a diagonal element of matrix $\mathbf{Q}(n)\mathbf{R}_{\hat{y}}(n)\mathbf{Q}^H(n)$. In order to simplify the structure of the subspace processor, a reasonable assumption can be made that $\lambda_{\delta_y}^{(m)}$ is proportional to $\lambda_{\hat{y}}^{(m)}$. Therefore, based on (5.75), we propose a suboptimal acoustic echo suppression filter gain as:

$$g^{(m)} = \frac{\lambda_d^{(m)}}{\lambda_d^{(m)} + \mu\lambda_{\hat{y}}^{(m)}}, \quad m = 1, 2, \ldots, K, \tag{5.76}$$

where the Lagrange multiplier μ controls echo suppression and near-end speech distortion: larger μ implies higher echo attenuation but more signal distortion. Note that $\lambda_v^{(m)}$ has been replaced by $\lambda_d^{(m)}$ in (5.76) since the former cannot be obtained in practice.

d) Dimension of signal subspace

The K-dimensional microphone signal subspace is the noisy (i.e., signal plus noise) subspace. Assuming that the dimension of the signal subspace is M where $M < K$, one can write $\mathbf{Q}(n)$ as

$$\mathbf{Q}(n) = [\mathbf{Q}_v(n), \mathbf{Q}_z(n)], \tag{5.77}$$

where $\mathbf{Q}_v(n)$ constitutes the signal subspace

$$\mathbf{Q}_v(n) = [\mathbf{q}_1(n), \mathbf{q}_2(n), \dots, \mathbf{q}_M(n)], \tag{5.78}$$

and the noise subspace is spanned by $\mathbf{Q}_z(n)$:

$$\mathbf{Q}_z(n) = [\mathbf{q}_{M+1}(n), \dots, \mathbf{q}_K(n)]. \tag{5.79}$$

Since it is very difficult to find an accurate rank of the signal subspace, M, a fixed value based on empirical data is used as the dimension of the signal subspace in this work in order to simplify the algorithm of the subspace echo processor. Hence, only the signals projected in the signal subspace are considered, as illustrated in Fig. 5.11.

e) Estimation of the covariance matrix $\mathbf{R}_d(n)$

In practice, empirical data are used to estimate the covariance matrix $\mathbf{R}_d(n)$. Differing from the approach in [114], many more past samples than future samples are used in this work to estimate $\mathbf{R}_d(n)$ in order to reduce the delay, which is an important issue in acoustic echo cancellation. Referring to the definition (5.63), the covariance matrix with samples from the past $(T-1)th$ frame to the current frame is estimated as

$$\mathbf{R}_d(n) = \frac{1}{TK} \sum_{i=n-TK+1}^{n} \mathbf{d}(i)\mathbf{d}^H(i), \tag{5.80}$$

where K is the length of the vector $\mathbf{d}(i)$, i.e., the frame length.

The estimation of the covariance matrix $\mathbf{R}_d(n)$ is performed frame by frame with a rectangular window, which contains the second order statistics of the samples in the window. After the estimation of the covariance matrix, the $\mathbf{Q}(n)$ and $\boldsymbol{\Lambda}_d(n)$ can be obtained by applying eigendecomposition to $\mathbf{R}_d(n)$.

5.4.2 Experimental Results and Discussion

The setup for the experiments are the same as in the previous chapter. The experiments are carried out in an office room with dimension 4(L)× 3.5(W)×2.7(H) m^3. A 1.3 GHz Pentium-IV PC is connected to a Delta 1010 Digital Recording System which has a 10-input and 10-output full-duplex recording interface. A common amplified PC loudspeaker is used to play the far-end speech. The microphone signal is amplified by a Tascam MX-80 microphone/line mixer before it is sent to the recording system. The computer fans are the main contributors of background noise.

The third-order AP algorithm is employed to estimate the acoustic echo, where the filter length is 1600 taps, corresponding to 200 ms at an 8 kHz sampling rate. The step-size is set to 0.9.

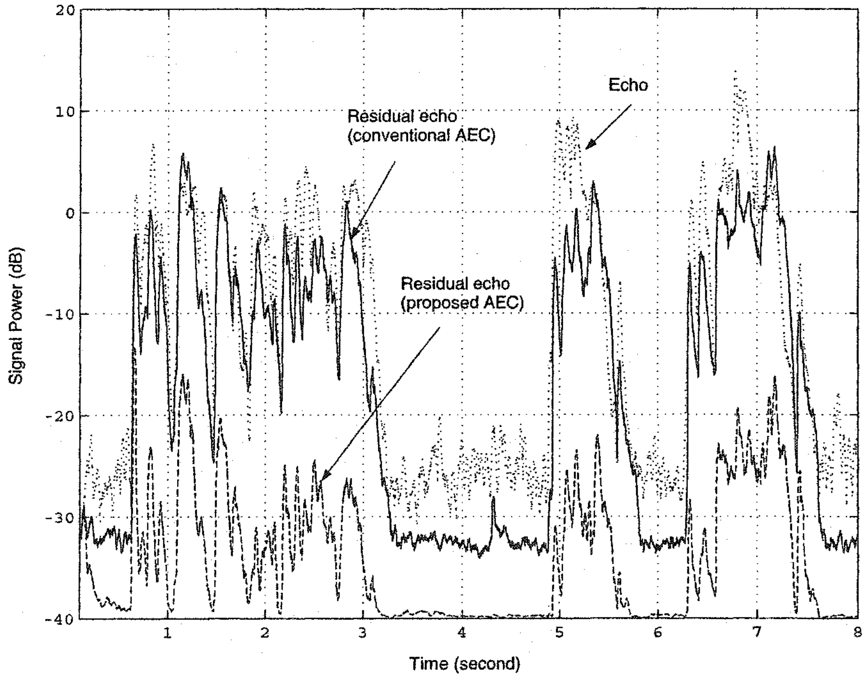

Fig. 5.12 The signal power versus time: dot – acoustic echo; solid – residual echo of a conventional AEC with AP; dash – residual echo of the proposed AEC.

For the subspace echo processor, the dimension of the noisy signal subspace is set to $K = 40$, and the dimension of the signal subspace is determined experimentally as $M = 32$. A Hanning window is used for synthesis, and a rectangular window is used for analysis, both with a 50% frame overlap. $T = 10$ frames are used to estimate the covariance matrix $\mathbf{R}_d(n)$. The parameter μ is set to 10 in order to compromise between echo suppression and signal distortion.

The experimental results are shown in Fig. 5.12. The result from a conventional AEC employing a linear adaptive filter with AP is also plotted for comparison. The proposed AEC outperforms the conventional AEC with an extra 10-20 dB echo suppression and 8 dB background noise reduction. Since the frame size is $K = 40$ samples corresponding to 5 ms, the delay is acceptable in most acoustic echo cancellation applications. Informal listening tests demonstrate that the subspace method achieves a satisfactory level of acoustic echo cancellation in the nonlinear channel.

Note that the subspace echo processor in the proposed AEC system has an open-loop structure, which shows a predictable behaviour. Therefore, it is much more robust than other nonlinear adaptive algorithms, such as Volterra filters and neural networks. Since the eigendecomposition operation in this AEC system leads to high computational complexity, an appropriate subspace tracking technique, e.g. [116, 117], could be used to reduce the computational burden.

5.5 Pitch extraction approach

Echo suppression during double-talk is a serious problem in conventional AEC systems that have echo path nonlinearities. In a conventional AEC system, the coefficients of the adaptive filter are frozen to avoid the divergence of the adaptive filtering algorithm when both near-end and far-end speech are active [96]. However, in nonlinear channels with vocoders, the acoustic echo is difficult to suppress and the residual echo may be larger than the original echo if the adaptation of the conventional AEC system is stopped because of the time-varying behaviour of the nonlinear characteristics [28].

In this section, a new AEC system is proposed for nonlinear channels. Combined with a linear adaptive filter, the new AEC system exploits the speech analysis technique, namely pitch extraction from the residual echo, to further suppress the residual echo produced by the linear adaptive filter. Simulations show that the proposed AEC system significantly suppresses the acoustic echo, especially during double-talk.

5.5.1 Pitch prediction filter

A speech signal is highly correlated, and thus has redundancies in either near-sample or distant-sample [118]. The near-sample redundancies can be removed by the formant filter, while the distant-sample waveform similarities can be removed by the pitch filter.

In AEC applications, the pure acoustic echo path (i.e., from the loudspeaker to the microphone) is modelled as a linear system. That is, the echo is the output of a linear filter representing the LEM system, with the loudspeaker signal as input. Compared to the loudspeaker signal, the formant of the acoustic echo usually changes significantly because the formant is affected by the spectrum of the LEM system. However, the pitch is represented as periodic impulses in the frequency domain; hence the characteristic of the pitch can be preserved for the output signal (i.e., echo) from the linear filter (i.e., LEM system). Consequently, the pitch information of the loudspeaker signal is similar to that of the echo signal, as we have been able to verify experimentally.

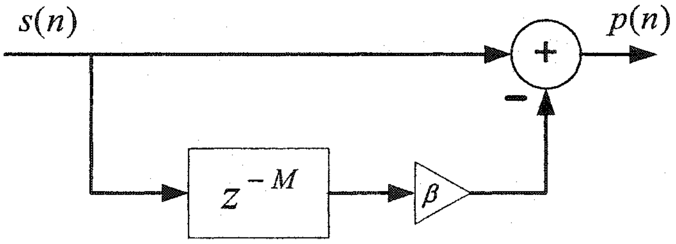

Fig. 5.13 The pitch prediction for a speech signal.

There are different pitch prediction filters such as multi-lag pitch filters and fractional delay pitch filters [118]. This work only examines the one-lag pitch filter shown in Figure 5.13, because of its simplicity and robustness. In this figure, $s(n)$ is the speech signal and $p(n)$ is the prediction error signal. This pitch filter has only one coefficient and is expressed as

$$P(z) = \beta z^{-M}, \tag{5.81}$$

where β is a scaling factor related to the degree of waveform similarity, and the integer M is the estimated period.

The basic method of finding the pitch parameters (the lag M and the correlation coefficient β) is open-loop analysis [118]. Using this approach, the pitch parameters M and β

are chosen to minimize the mean-squared residual $p(n)$ in each N-sample frame:

$$\arg\min_{M,\beta} \sum_{n=0}^{N-1} p^2(n), \tag{5.82}$$

where, from Figure 5.13,

$$p(n) = s(n) - \beta s(n - M). \tag{5.83}$$

In narrow-band speech applications where the sampling rate is typically 8 kHz, the lag M ranges between 20 to almost 150 samples [50]. The pitch coefficient β varies from 0 to 1. For a signal with no detectable periodic structure such as unvoiced speech, β is 0 and M is irrelevant. For a well-structured periodic signal such as steady-state voiced speech, β is close to 1. For other cases, the value of β lies between 0 and 1.

For a given lag M, (5.82) leads to the optimal value of pitch coefficient in terms of M:

$$\beta_{opt}(M) = \frac{\sum_{n=0}^{N-1} s(n)s(n-M)}{\sum_{n=0}^{N-1} s^2(n-M)}. \tag{5.84}$$

Clearly, in order to find the optimal gain β_{opt} among the values of $\beta_{opt}(M)$ and the corresponding M, an exhaustive search in the pitch lag range is necessary. Considering the non-stationary property of the speech signal, the frame size N should not be too large, to avoid reduction of prediction gain [119] and large delays. The frame size cannot be too small either, or it may cause inaccurate estimation of the pitch lag. In this work, different values of the frame size N are used to estimate the pitch lag and the coefficient. A larger frame size $N_2 = 80$ is used for the lag M estimation, while a shorter frame size $N_1 = 40$ is used to find the gain β and to update the output residual.

In order to avoid the pitch multiples issue of the pitch filter, the search range for the parameters is divided into three regions [24]. In practice, the computational complexity of a pitch predictor is significantly reduced by searching the pitch parameters in the following steps.

Loop: for each frame ($N_1 = 40$)

- Step 1: Find three maxima $r(m_i)$, $i = 1, 2, 3$, from the correlations

$$r(m) = \sum_{n=0}^{N_2-1} s(n)s(n-m), \tag{5.85}$$

where the larger frame size $N_2 = 80$ is used to obtain a better estimation of the correlation, and the three ranges are

$$\begin{aligned} i &= 1, \quad 80 \le m \le 147 \\ i &= 2, \quad 40 \le m \le 79 \\ i &= 3, \quad 20 \le m \le 39. \end{aligned} \tag{5.86}$$

- Step 2: Normalize the three candidates

$$\tilde{r}(m_i) = \frac{r(m_i)}{\sqrt{\sum_{n=0}^{N_2-1} s^2(n-m_i)}}, \quad i = 1, 2, 3. \tag{5.87}$$

- Step 3: Search for the proper pitch lag M among the above candidates, where a smaller one is preferred to avoid the pitch multiples:

```
Initialization: $M = m_1$;
Loop: for $i = 2, 3$
        if $\tilde{r}(m_i) \ge \rho\tilde{r}(M)$
          $M = m_i$
        end
     end
```

The weighting parameter ρ is set to 0.85 experimentally.

- Step 4: Compute the pitch gain β by using a shorter frame of N_1 samples:

$$\beta = \frac{\sum_{n=0}^{N_1-1} s(n)s(n-M)}{\sum_{n=0}^{N_1-1} s^2(n-M)}. \tag{5.88}$$

End loop.

5.5.2 Pitch extraction in AEC

Figure 5.14 shows the proposed AEC system for nonlinear channels where low-bit-rate codecs are present along the echo path. It consists of two components: an echo estimator and a pitch extractor. The estimated echo $\hat{y}(n)$ produced by the echo estimator is subtracted from the microphone signal $d(n)$, resulting in the residual echo $e(n)$. The processed residual echo $e_p(n)$ is obtained by attenuating the residual echo $e(n)$ through pitch extraction where the pitch parameters are computed from the estimated echo $\hat{y}(n)$.

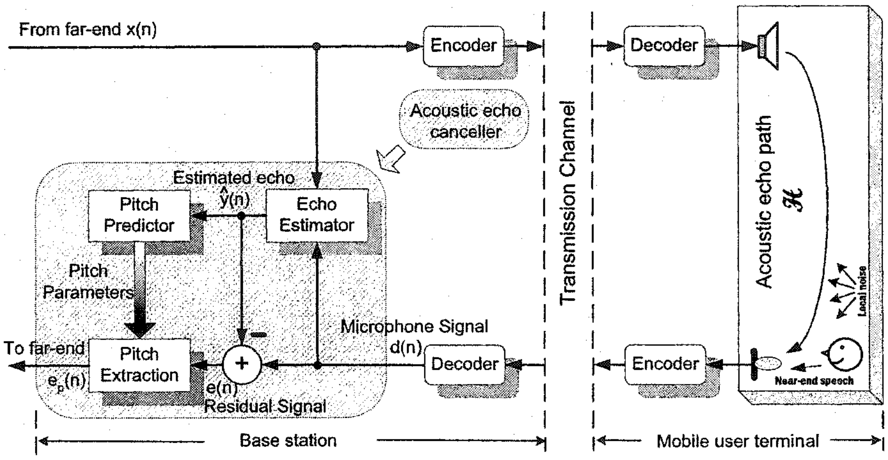

Fig. 5.14 Pitch analysis-based acoustic echo cancellation over a nonlinear channel.

An adaptive filter is used to estimate the echo in most AEC applications, but its performance is significantly degraded by the nonlinearities of the codecs along the echo path. The affine projection (AP) algorithm [21] shows the best performance among popular adaptive filtering algorithms in nonlinear channels [120], therefore it is used to estimate the echo in the new AEC system.

Speech analysis indicates that most of the speech energy is concentrated in the voiced sounds, which have a power that is about 20 dB larger than that of the unvoiced sounds [50]. Furthermore, the voiced sounds have a relative periodicity which is represented by the pitch. Because the content of the residual echo is often recognizable, the residual echo should retain certain speech characteristics. Based on these considerations, the power of the residual echo from a conventional acoustic echo canceller can be further reduced if the pitch of the residual echo is extracted.

In the proposed AEC system shown in Figure 5.14, the acoustic echo is attenuated by subtracting the estimated echo $\hat{y}(n)$ from the microphone signal $d(n)$ before it is further suppressed by the pitch filter. The residual echo signal $e(n)$ may contain the near-end speech and the remaining echo. When the near-end speech is active, it is almost impossible to obtain the correct pitch parameters of the echo component from the residual echo signal.

However, as discussed before, the pitch information of the far-end speech $x(n)$ is similar to that of the echo, but this information cannot be directly applied to the residual echo due to the synchronization problem, i.e., the delay introduced by the codecs and the acoustic echo path. This problem can be solved if the estimated echo $\hat{y}(n)$ is used to obtain the pitch parameters. Then these parameters can be applied to the pitch filter that attenuates the residual echo. This is because the pitch information of the estimated echo $\hat{y}(n)$ and that of the echo component contained in the residual signal $e(n)$ are very similar when the echo estimator is active. Furthermore, based on the assumption that the delay of the entire echo path does not change significantly during double-talk when the coefficients of the echo estimator are frozen, those two signals are still well synchronized.

The pitch parameters (the pitch lag M and the pitch gain β) of the estimated echo $\hat{y}(n)$ are obtained by using the algorithm in section 5.5.1 from step 1 to 4, where $s(n)$ should be replaced by $e(n)$. The pitch of residual echo $e(n)$ is then extracted using (5.83). Similarly, the signals $p(n)$ and $s(n)$ in (5.83) are replaced by $e_p(n)$ and $e(n)$, respectively.

5.5.3 Simulation results

In order to test the proposed AEC system, a simulation is conducted using the platform shown in Figure 5.14, where the codec used is G.729 [24]. A coloured noise, produced by passing a white noise through an IIR filter with the system function $H(z) = \frac{0.1}{1-0.9z^{-1}}$, is added as the background noise so that the ENR is 30 dB. The LEM system of the test platform is simulated to represent the cab of a vehicle. The impulse response is about 40 ms long, corresponding to 300 taps at the sampling rate of 8 kHz. The relaxation factor μ for the AP echo estimator is set to 0.9, and the projection order p is 3 (a higher order would not lead to obvious improvement in this case [28]).

DT occurs between time 0.6 s and 2.7 s. During this period, the coefficients of the echo estimator, i.e., the AP algorithm, are frozen to avoid divergence. The simulation results are shown in Figure 5.15 and Figure 5.16. The residual echo of the conventional AEC system that only employs AP to suppress echo is also plotted in these figures for comparison.

These simulation results show that, in the nonlinear channel, the pitch analysis-based AEC system yields 5 to 10 dB of additional echo attenuation during double-talk compared to the conventional AEC system. In the case of single-talk, the new AEC system obtained the same results as the conventional AEC system, since the periodic similarities in the residual echo have been removed by the echo estimator when the AP algorithm is active. Minor distortions to the near-end speech, which may add a little extra noise to the near-end speech, are introduced by the new AEC system. However, since local background noise exists in most hands-free applications, this distortion is tolerable.

5.6 Conclusion

In this chapter, various post-filtering algorithms have been proposed and discussed in the application of AEC. These algorithms significantly compensate the performance degradation of a conventional AEC system in nonlinear channels. Different combinations of

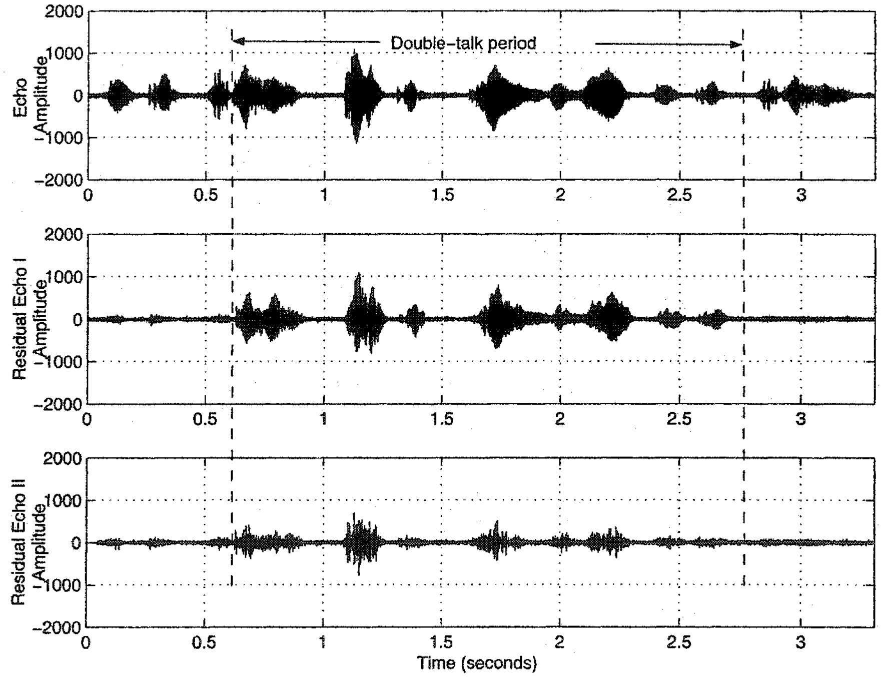

Fig. 5.15 Waveforms of the echo, residual echo I (only AP is employed), and residual echo II (the proposed AEC system).

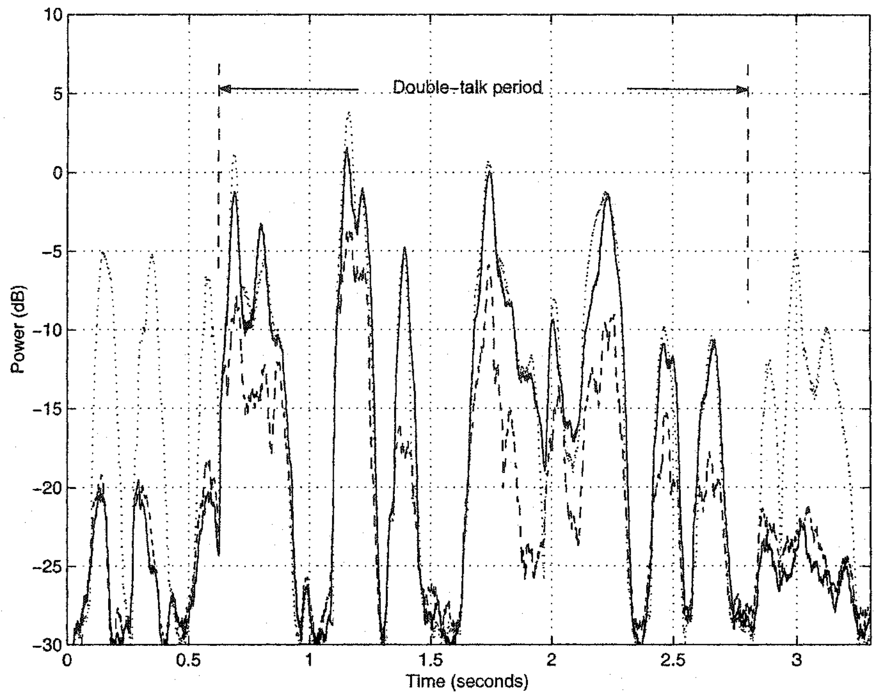

Fig. 5.16 Power versus time for echo (dot), residual echo of AP (solid), and residual echo of the proposed AEC system (dash).

adaptive filters and post-filters can be applied in different situations. For example, the VSS-ACS algorithm combined with an over-weighted Wiener filter can be employed in the case where the the acoustic echo level is moderate. Although the echo attenuation of this combination is not ideal, it has the important advantage that no DT detector is required. In another situation where echo suppression is critical, a combination of the AP algorithm with an over-weighted Wiener filter is a good choice. However, a robust DT detector is necessary in order to freeze the adaptive filter coefficients. In addition, the pitch subtraction technique can also be used to reduce the power of the residual echo during the DT period. Furthermore, an AEC system based on the subspace method is proposed to suppress the acoustic echo in the nonlinear channel. Experimental results show that the acoustic echo can be significantly attenuated by this AEC system.

These post-filters all introduce some distortion in the near-end signal depending on the amount of echo attenuation. However, since signal distortion caused by nonlinear devices is unavoidable in nonlinear channels, adding slightly more distortion with the benefit of significant echo suppression is acceptable. Musical noise is also present when a post-filter is used, but it can be significantly diminished by exploiting a perceptual model of the human ear, as will be discussed later.

Chapter 6

Computational complexity reduction: subband adaptive filtering

In the last chapter, the study of the post-filtering techniques reveals that the latter can significantly suppress the residual echo that resulted from insufficient echo attenuation of a conventional AEC. However, one of the challenging tasks of AEC remains: the long echo path in AEC applications requires numerous taps for an FIR adaptive filter, resulting in very high computational complexity. This may prevent the AEC system from being implemented in real-time. In order to reduce the computational complexity, this chapter introduces an effective approach called subband adaptive filtering. A simplified design of the oversampling subband structure is investigated, and a practical AEC system that seamlessly incorporates the post-filter in the subband is proposed. Compared to its full-band counterpart, the proposed subband algorithm has a much lower computational complexity while achieving a comparable performance in terms of acoustic echo suppression in the nonlinear channel. These observations are supported by experimental results described in this chapter.

6.1 Introduction

For AEC applications requiring very long filters (N in excess of a few thousand), the $2N$ computational complexity of the NLMS algorithm may be unacceptable for low-cost real-time implementations. To go below the barrier of $2N$ operation per sample, some structural modifications to the conventional transversal filter structure are necessary. Possible approaches include block processing [121, 47] and transform domain filtering [122, 123]. In recent years, the subband filtering approach has received considerable attention [124, 125]. Compared with the full-band case, subband adaptive filtering lowers computational complexity by reducing the sampling rate of the subband signals.

Subband adaptive filtering has become a very important scheme for AEC. In fact, many commercial AEC systems available today use subband filtering to reduce the computational complexity so the system can be implemented in real-time.

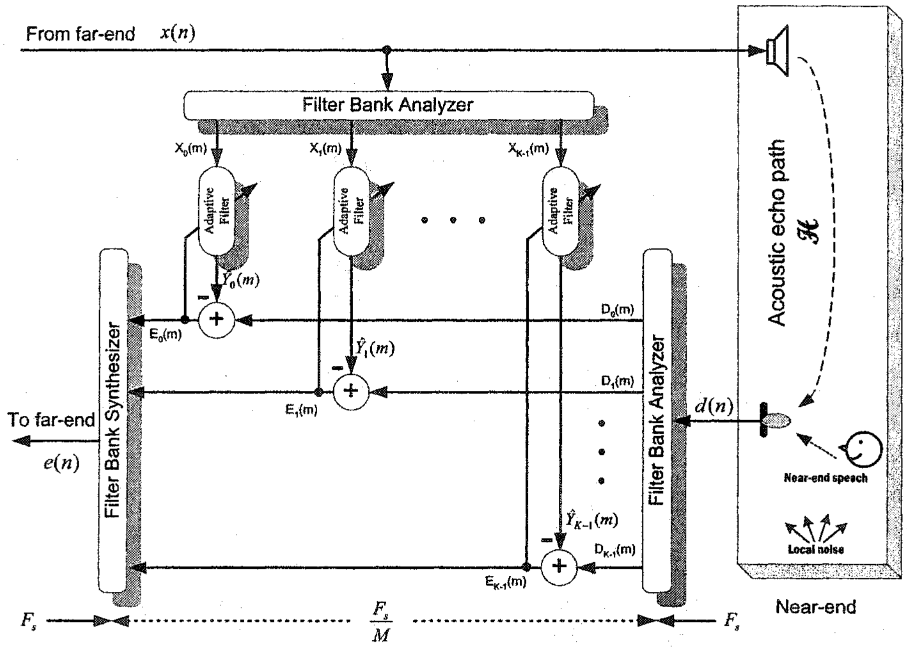

Fig. 6.1 Structure of a subband adaptive filter

Figure 6.1 shows conventional subband filtering for AEC. Both the loudspeaker signal $x(n)$ and the microphone signal $d(n)$ are split into K subband signals by an analysis filter bank, which can be viewed as a set of K band-pass filters that cover the whole spectrum of interest. At the output of the analysis filter bank, the sampling rate of the subband signals is reduced by an integer M, referred to as the decimation factor. In each subband, an adaptive filter is used at a lower sampling rate in order to cancel echo within the subbands. More specifically, each subband adaptive filter uses the corresponding subband component of the loudspeaker as its input, and the corresponding subband component of the microphone as its reference. The input and output are respectively denoted $X_i(m)$ and $D_i(m)$, $i = 0, \cdots, K-1$, where the index m is a time instant at the lower sampling rate.

The components of the residual echo of each subband adaptive filter, denoted by $E_i(m)$, are reconstructed by the synthesis filter bank in order to form the full-band output $e(n)$ at the original sampling rate.

The most important advantage of using subband adaptive filtering is the reduction of the computational complexity, especially when using a long filter (i.e., with a large number of taps). This reduction is made possible by two changes:

- The updating rate of the adaptive filter coefficients in the subband is reduced by a factor of M.
- The length of subband adaptive filters is reduced by the same factor M compared to the corresponding full-band filter.

Thus, by using subband adaptive filters, a total computational gain of up to M^2/K can be achieved under the assumption that the computational requirements of the adaptive filtering algorithm is proportional to the length of the transversal filter. When the additional computations required for the analysis/synthesis banks are taken into account, the computational gain will be slightly lower. Since the reference signal may contain a near-end speech component that will go through the cascade of an analysis and a synthesis bank prior to its transmission, a perfect-reconstruction (PR) or near-PR property is needed for the filter banks so that the near-end signal will not be distorted.

6.2 Uniform DFT filter banks

6.2.1 The basic structure

Uniform DFT filter banks provide a simple method for the rapid design and prototyping of filter banks suitable for AEC. A structural diagram of the proposed K-channel uniform DFT filter banks is depicted in Figure 6.2. In the analysis bank, a complex modulation function W_K^{kn} is applied to the input signal $x(n)$, where $W_K = \exp(j2\pi/K)$, $k = 0, 1, 2, \cdots K$ is the channel index, and n is the discrete-time index at high sampling rate. The analysis filter, represented by its impulse response $h(n)$, is a lowpass filter with cutoff frequency $\omega_c = \pi/K$. Let $H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$ denote the z-transform of $h(n)$, then the ideal low-pass property for $H(z)$ should be

$$|H_{\text{ideal}}(e^{j\omega})| = \begin{cases} 1, & 0 \le |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases} \tag{6.1}$$

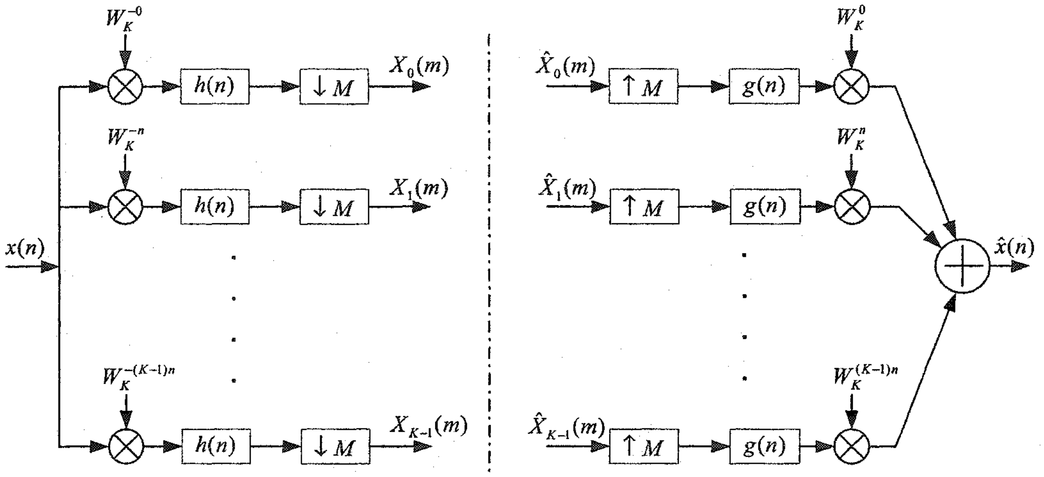

Fig. 6.2 Uniform DFT filter banks

After modulation and lowpass filtering, the signal in kth branch is decimated by an integer factor M, resulting in the desired subband signal, $X_k(m)$, where m is the discrete-time index at low sampling rate. Using well-known properties [126], the subband signal can be expressed in the z-domain as

$$X_k(z) = \frac{1}{M}\sum_{l=0}^{M-1} H(z^{1/M}W_M^{-l})X(z^{1/M}W_M^{-l}W_M^{k}). \tag{6.2}$$

For subsequent discussions, it is convenient to rewrite (6.2) in the form

$$X_k(z) = \frac{1}{M}H(z^{1/M})X(z^{1/M}W_M^{k}) + A_k(z), \tag{6.3}$$

where

$$A_k(z) = \frac{1}{M}\sum_{l=1}^{M-1} H(z^{1/M}W_M^{-l})X(z^{1/M}W_M^{-l}W_M^{k}) \tag{6.4}$$

is viewed as the aliasing component due to decimation in the kth subband signal $X_k(z)$.

In the application of subband adaptive filtering in AEC, this kind of frequency-domain aliasing must be avoided to achieve effective cancellation of acoustic echo.

Note that $A_k(x)$ is affected by the stop-band property of the low-pass filter $H(e^{j\omega})$. Since the ideal low-pass characteristic (6.1) can only be approximated in practice, i.e., in the case of a realizable (causal and stable) filter $H(e^{j\omega})$, a transition band does exist for $H(e^{j\omega})$ and significant aliasing components will generally be created if the critical downsampling factor $M = K$ is used in the subbands. Thus, an oversampling scheme, i.e., $M < K$, is often used in filter banks for subband adaptive filtering so that frequency domain aliasing can be made acceptably small [127, 128].

In the synthesis bank shown in Figure 6.2, each subband signal is first upsampled by the factor M and then passed to a common synthesis filter $g(n)$, or equivalently $G(z)$, which is also a lowpass filter with cutoff frequency $\omega_c = \pi/K$. The output of each filter are modulated by a proper complex modulating function, which is given by W_K^{kn} for the kth subband. Finally they are summed to produce the synthesized output $\hat{x}(n)$, which can be expressed as [16]

$$\hat{x}(n) = \sum_{m=-\infty}^{\infty} g(n-mM)\frac{1}{K}\sum_{k=0}^{K-1}\hat{X}_k(m)W_k^{kn}. \tag{6.5}$$

Referring to Figure 6.2, the channel signals at the output of the analysis bank can be written as

$$X_k(m) = \sum_{n=-\infty}^{\infty} h(mN-n)x(n)W_k^{-kn}, \qquad k = 0, 1, \cdots, K-1. \tag{6.6}$$

When $\hat{X}_k(m) = X_k(m)$ for all m, $k = 0, 1, \cdots, K-1$, it is desired to have $\hat{x}(n) = x(n)$. This condition leads to the following relation between the analysis filter $h(n)$ and the synthesis filter $g(n)$ [129]:

$$\sum_{m=-\infty}^{\infty} g(n-mM)h(mM-n+sK) = \begin{cases} 1, & s=0 \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

for all n.

6.2.2 Prototype filters

Prototype filter designs with better lowpass characteristics and smaller reconstruction errors for a given filter length N can be achieved with computer-aided optimization techniques [130, 131]. One criterion is to minimize an error function E, which is defined as

$$E = \alpha E_s + E_r, \tag{6.8}$$

where α is a real positive weighting factor, while E_s and E_r are given in [16]:

$$E_s = \int_{\omega_s}^{\pi} |H(e^{j\omega})|^2 d\omega, \tag{6.9}$$

$$E_r = \int_0^{2\pi} \left(\sum_{k=0}^{K-1} |H(e^{j\omega} W_K^{-k})|^2 - 1 \right)^2 d\omega, \tag{6.10}$$

where $\omega_s (> \omega_c)$ is the stopband edge which determines the transition band of $H(e^{j\omega})$.

In practice, the error surface E, as a function of the coefficients, has many local minima. An optimization routine can get trapped in these local minima and fail to reach the global minimum. Instead of using complex optimization programs, there is a simpler way to design the prototype filter $h(n)$ for subbanks by performing an I-point interpolation of a two-channel QMF prototype filter $h_0(n)$ (such filters can be found in [16]). The simplicity of this design procedure makes it well-suited for practical engineering applications.

Traditionally, $h(n)$ can be obtained from the I-point interpolation of $h_0(n)$, followed by anti-imaging lowpass filtering. Although this time-domain method is straightforward and well-established in theory, it is not easy to implement in practice. Assume that the original QMF filter has a length of L_0 taps, then the new prototype filter has a length of IL_0. If the anti-imaging lowpass filter is N taps long, then the sequence produced by this filter is $IL_0 + N - 1$ long. The length of the lowpass filter N should be long enough to achieve high stopband attenuation in order to significantly suppress the interpolation imaging. A longer analysis/synthesis filter leads to a larger processing delay which should be avoided in practice. A simple method proposed in [18] reduces the filter length to as short as IL_0.

This work proposes a method to obtain an analysis/synthesis filter by interpolating the QMF filter in the frequency domain, rather than in the time domain. This method, called the DFT method, is accomplished by exploiting the properties of DFT. First, an L_0-point

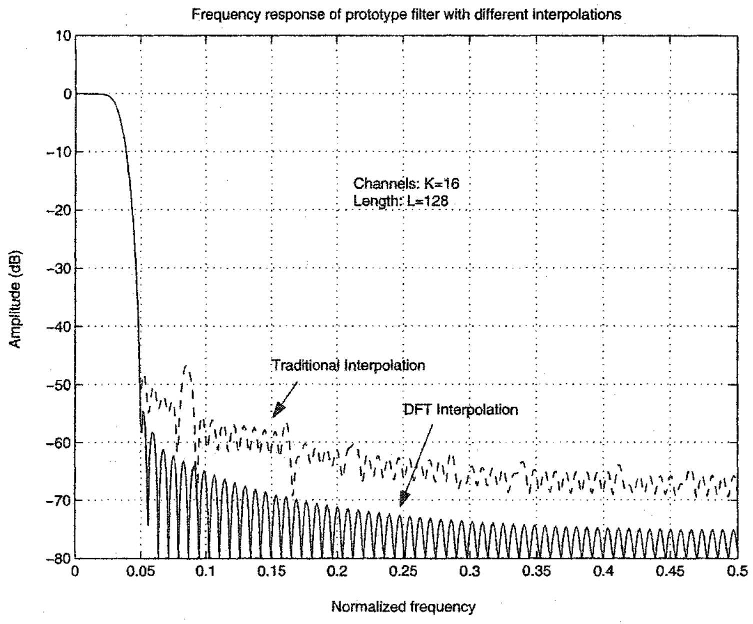

Fig. 6.3 Prototype filter

DFT is applied to the coefficients of the QMF filter $h_0(n)$, resulting in a sequence of length L_0. Next, this sequence is expanded to length IL_0 by padding $(I-1)L_0$ zeros. Finally, IL_0-point inverse DFT is applied to this sequence, so that the filter $h(n)$ is obtained.

As an example, two prototype filters are obtained by interpolating the QMF filter 16A [16], which has 16 parameters, by a factor $I = 8$ using both the method in [18] and the above DFT method. The prototype filters obtained from the two different methods have the same length, which is 128. The frequency responses of the filters are plotted in Figure 6.3. It is found that using the DFT method results in a significantly higher stopband attenuation, by about 10 dB on average.

6.2.3 Reconstruction error

To evaluate the near-PR property of a subband structure, we need to investigate two kinds of distortion, namely: aliasing and amplitude distortions.

Expressing (6.2) in the frequency domain, the decimated channel signals can be written as

$$X_k(e^{j\omega'}) = \frac{1}{M}\sum_{k=0}^{M-1} H(e^{j(\omega'-2\pi l)/M})X(e^{j(\omega'-2\pi l)/M+2\pi k/K}), \tag{6.11}$$

where ω' denotes the frequency with respect to the decimated sampling rate. Combining (6.2) and (6.5), the frequency domain input-output relationship for the filter bank is obtained as

$$\begin{aligned}\hat{X}(e^{j\omega}) &= X(e^{j\omega})\frac{1}{K}\sum_{k=0}^{K-1} G_k(e^{j\omega})H_k(e^{j\omega}) \\ &+\sum_{l=1}^{K-1} X(e^{j(\omega-2\pi l/M)})\frac{1}{K}\sum_{k=0}^{K-1} G_k(e^{j\omega})H_k(e^{j(\omega-2\pi l/M)}),\end{aligned} \tag{6.12}$$

where $H_k(e^{j\omega})$ and $G_k(e^{j\omega})$ are the transforms of $h_k(n) = h(n)W_k^{kn}$ and $g_k(n) = g(n)W_k^{kn}$ respectively.

As is known, the synthesis filter $g(n)$ can be obtained by flipping the analysis filter $h(n)$ of L taps in the time domain:

$$g(i) = h(L-i-1), \quad i = 0, \ldots, L-1. \tag{6.13}$$

If $h(n)$ is symmetric, the synthesis filter $g(n)$ is then identical to the analysis filter $h(n)$. Then (6.12) becomes

$$\begin{aligned}\hat{X}(e^{j\omega}) &= X(e^{j\omega})\frac{1}{K}\sum_{k=0}^{K-1} |H(e^{j\omega}W_K^{-k})|^2 \\ &+\sum_{l=1}^{K-1} X(e^{j\omega}W_M^{-l})\frac{1}{K}\sum_{k=0}^{K-1} H(e^{j\omega}W_K^{-k})H(e^{j\omega}W_K^{-k}W_M^{-l}).\end{aligned} \tag{6.14}$$

Since $h(n)$ is a linear-phase FIR filter, the following conditions must be satisfied for perfect reconstruction (PR):

$$\begin{cases} \frac{1}{K}\sum_{k=0}^{K-1} |H(e^{j\omega}W_K^{-k})|^2 = 1 \\ \sum_{k=0}^{K-1} H(e^{j\omega}W_K^{-k})H(e^{j\omega}W_K^{-k}W_M^{-l}) = 0, \qquad l = 1, 2, \cdots, M-1. \end{cases} \tag{6.15}$$

In practice, a near-PR property is sufficient for the filter banks in AEC applications [18]. The magnitude distortion can be measured from the peak ripple of $T_0(e^{j\omega})$, where $T_0(e^{j\omega})$

is given by

$$T_0(e^{j\omega}) = \frac{1}{K}\sum_{k=0}^{K-1} |H(e^{j\omega}W_K^{-k})|^2. \tag{6.16}$$

The full-band aliasing distortion is given by

$$E_f(e^{j\omega}) = \frac{M-1}{K}\sqrt{\sum_{l=1}^{M-1}\left|\sum_{k=0}^{K-1} H(e^{j\omega}W_K^{-k})H(e^{j\omega}W_K^{-k}W_K^{-l})\right|^2}, \quad |\omega| \le \pi. \tag{6.17}$$

According to (6.4), the subband decimation aliasing distortion is defined as

$$E_s(e^{j\omega}) = \sqrt{\sum_{l=1}^{M-1}\left|H(e^{j\omega/M}W_M^{-l})\right|^2}, \quad |\omega| \le \pi. \tag{6.18}$$

Note that the absence of the full-band aliasing distortion does not guarantee that there is no decimation aliasing in the subbands; however, aliasing distortion in the full-band leads to decimation aliasing in subbbands.

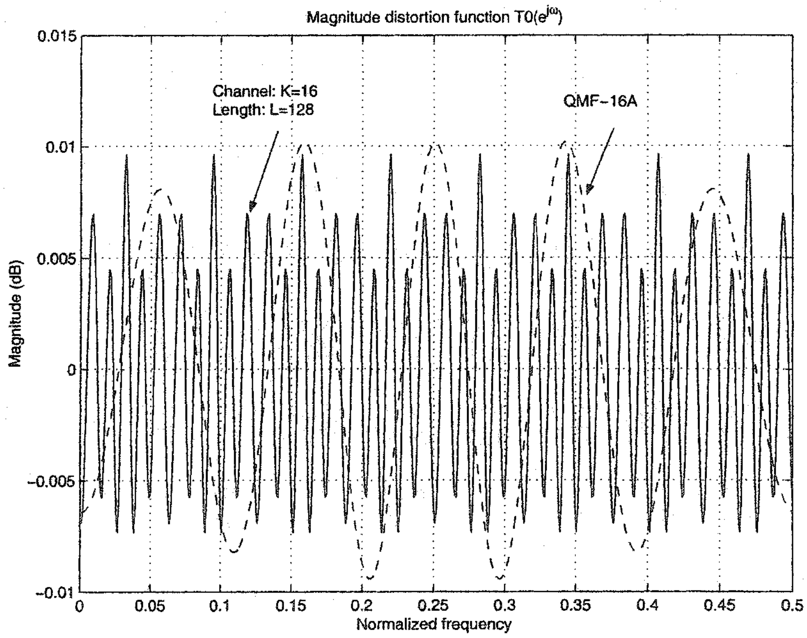

Fig. 6.4 Magnitude distortion function $T_0(e^{j\omega})$

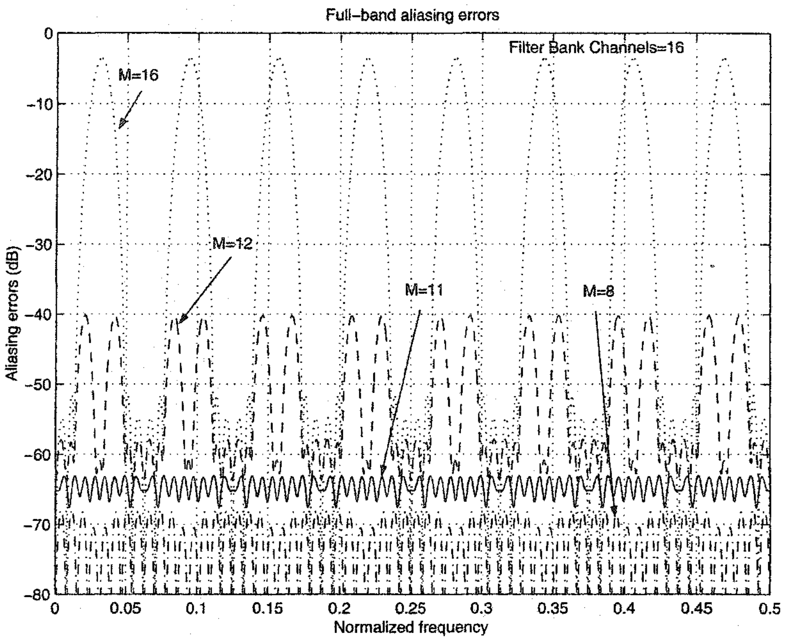

Fig. 6.5 Full-band aliasing error

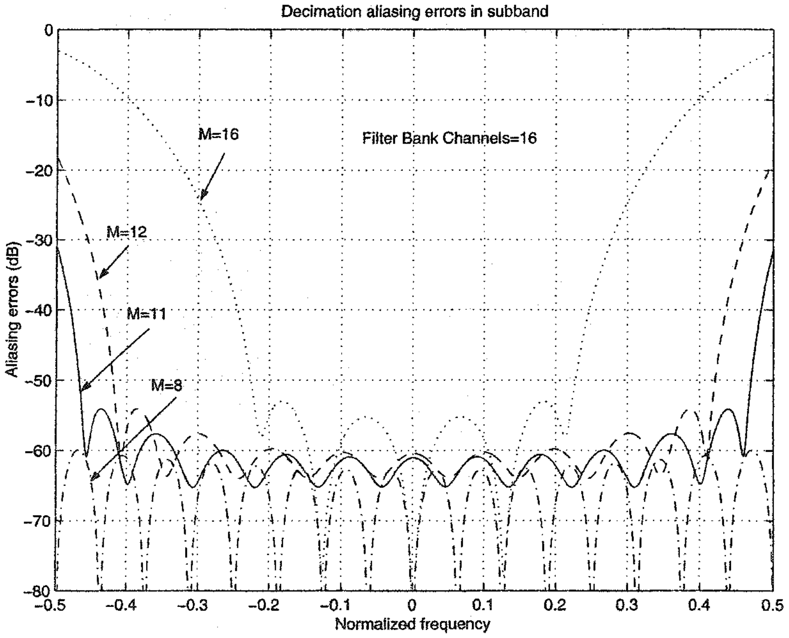

Fig. 6.6 Decimation aliasing error

As an example, consider the prototype QMF filter 16A in [132]. This filter has 16 symmetrical coefficients and its stopband attenuation is about 60 dB. To design a uniform DFT subband, shown in Figure 6.2, with $K = 16$ channels, an analysis/synthesis filter is obtained by using the DFT method. The length of this filter is therefore 128 taps. The magnitude distortion (6.16) of the subband using this filter is shown in Figure 6.4. Apparently, so obtained filter banks inherit the low magnitude distortion of the original QMF filter banks.

Figure 6.5 shows the full-band aliasing error (6.17) for the same filter banks with different decimation factors. Figure 6.6 shows the corresponding decimation aliasing error in (6.18). Because the analysis/synthesis filter is not ideal, aliasing errors are unavoidable. Both the full-band and the decimation aliasing errors increase when the decimation rate increases. From the results, it is found that the decimation factor $M = 11$ or $M = 12$ is a reasonable choice for a 16-channel filter bank.

6.2.4 Realization in weighted overlap-add filter structure

There are two basic structures for realizing DFT filter banks: polyphase structure and weighted overlap-add structure. The latter is based on an interpretation of the DFT filter bank in terms of a block-by-block transform analysis (or synthesis) of the signal. It is more general than the polyphase structure in that it can be more easily applied to cases where the channel decimation ratio M is unrelated to the number of subbands K. In other words, there are no restrictions to the relation between M and K. In this thesis, we use the weighted overlap-add structure.

To develop the weighted overlap-add structure, one starts with the basic filter bank model as illustrated in Figure 6.2. The output signal $X_k(m)$ (6.6) for the kth channel of the filter bank analyzer can be expressed as

$$X_k(m) = W_K^{-kmM} \tilde{X}_k(m), \tag{6.19}$$

where

$$\tilde{X}_k(m) = \sum_{r=0}^{K-1} \tilde{x}_m(r) W_K^{-kr}, \tag{6.20}$$

and

$$\tilde{x}_m(r) = \sum_{l=-\infty}^{\infty} h(-r-lK)x(r+lK+mM), \qquad r = 0, 1, , \cdots, K-1. \tag{6.21}$$

This structure is considerably more efficient than a direct implementation of the DFT filter bank because of the use of the direct-form decimation structure, of the same windowing process shared with all the subbands, and of the FFT for the modulation [16]. Similarly, the reconstructed signal for the synthesizer in (6.5) can be expressed in the form

$$\hat{x}(r+mM)|_{m=m_0} = g(r)\hat{\tilde{x}}(r)|_{m=m_0} + \quad (\text{terms} \quad \text{for} \;\; m \neq m_0), \tag{6.22}$$

where

$$\hat{\tilde{x}}_m(r) = \frac{1}{K}\sum_{k=0}^{K-1} \hat{\tilde{X}}_k(m) W_K^{kr}, \tag{6.23}$$

and

$$\hat{\tilde{X}}_k = \hat{X}_k(m) W_k^{kmM}. \tag{6.24}$$

Just like the analysis structure, the synthesis structure performs the DFT filter bank synthesis by using efficient interpolation, sharing filter computation among channels, and using a fast inverse transform algorithm to do the modulation.

As pointed out earlier, the adaptive filter operates in each subband, resulting in a significant reduction of computational complexity. Moreover, when an adaptive filter is implemented in a uniform DFT subband, the filter bank structure can be simplified. Specifically (6.19) and (6.24) are no longer needed for certain adaptive filtering algorithms. The derivation can be found in Appendix A.

Similar to [48], the main steps of modified implementation of the analysis bank are summarized as follows:

1. At block m in the full-band, input the vector consisting of M consecutive samples of $x(n)$ (i.e., $[x(mM), x(mM-1), \ldots, x(m-1)M+1)]$) into a shift register of length L, where L is the length of the analysis filter $h(n)$.

2. Multiply the above vector by the analysis filter $h(n)$, resulting in a windowed sequence $y_m(r) = h(r)x(mM-r)$.

3. Partition $y_m(r)$, $r = 0, 1, \ldots, L-1$ into $N_b = L/K$ blocks of K consecutive samples.

4. Add these N_b blocks together in order of sequence, resulting in a new sequence $\tilde{x}_m(r)$ of K samples. In other words, the first sample of the new sequence is the sum of the first samples of the N_b blocks, and the second sample is the sum of the second samples of the N_b blocks, etc.

5. Apply a K-point FFT to the sequence $\tilde{x}_m(r),\ r = 0, 1, \ldots, K-1$, resulting in the set of subband signals $\tilde{X}_k(m),\ k = 0, 1, \ldots, K-1$. These signals are used as the excitation signals for the subband adaptive filters.

The procedure of the synthesis bank is almost the reverse of the above steps. The implementation of the synthesis bank described in (6.22)-(6.23) is stated as:

1. At time m, apply a K-point IFFT to the subband signals $\hat{\hat{X}}_k(m),\ k = 0, 1, \ldots, K-1$, resulting in a sequence $\hat{\hat{x}}_m(r),\ r = 0, 1, \ldots, K-1$.

2. Extend the sequence $\hat{\hat{x}}_m(r)$ periodically into a new sequence of L samples, which is also denoted as $\hat{\hat{x}}_m(r)$, but with the index r ranging from $r = 0$ to $L-1$.

3. Multiply the new sequence $\hat{\hat{x}}_m(r)$ by the analysis filter $h(n)$, resulting in $\hat{y}_m(r) = \hat{\hat{x}}_m(r)h(r),\ r = 0, 1, \ldots, L-1$.

4. Perform an overlap-add operation on $\hat{y}_m(r)$. Details can be found in [16].

5. Shift the output buffer to the left by M samples, and the resulting block of M samples outside is $\hat{x}(mM - r),\ r = 0, 1, \ldots, M-1$.

In AEC applications, all excitation signals are always real-valued so that certain symmetry relations hold among the subband signals:

$$X_k(m) = X^*_{K-k}(m), \quad k = 1, 2, \ldots, K/2 - 1. \tag{6.25}$$

Note that $X_0(m)$ and $X_{K/2}(m)$ are real-valued. Because these symmetry relations are not affected by subband filtering, they can be exploited to further reduce computational complexity. For instance, as a result of (6.25), the outputs of subbands $K/2+1, \ldots, K-1$ do not need to be computed any more.

6.3 Adaptive filter combined with post-filter in subband

The previous chapter showed a combined AEC system that significantly suppresses acoustic echo in non-linear channels. The main drawback of this system is its high computational complexity, which is caused by both the computational requirement of the full-band adaptive filtering algorithm, and by the use of FFT for transforming the signals between the time and the frequency domains. In order to reduce the computational complexity for practical implementation, a combined adaptive filter and post-filtering algorithm employed in the subband is preferable. A combination of these two components results in significant computational savings [31].

6.3.1 Nonlinear effects on the subband filter

Before we study the performance of the subband filter in the nonlinear channel, it is important to understand its behaviour in the conventional linear channel, as depicted in Figure 6.1.

First, simulations with different excitation signals (both white noise and speech) are conducted to compare the performance of the subband adaptive filtering algorithm with

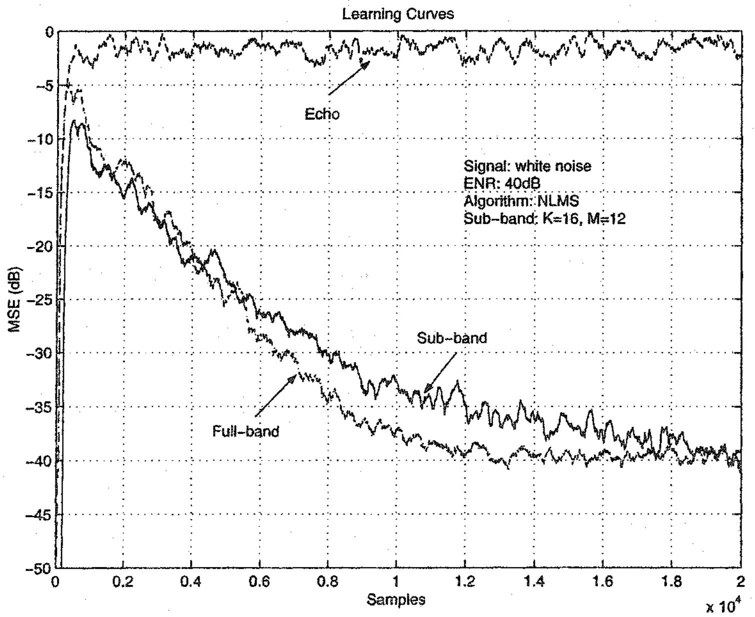

Fig. 6.7 Learning curves for white noise excitation

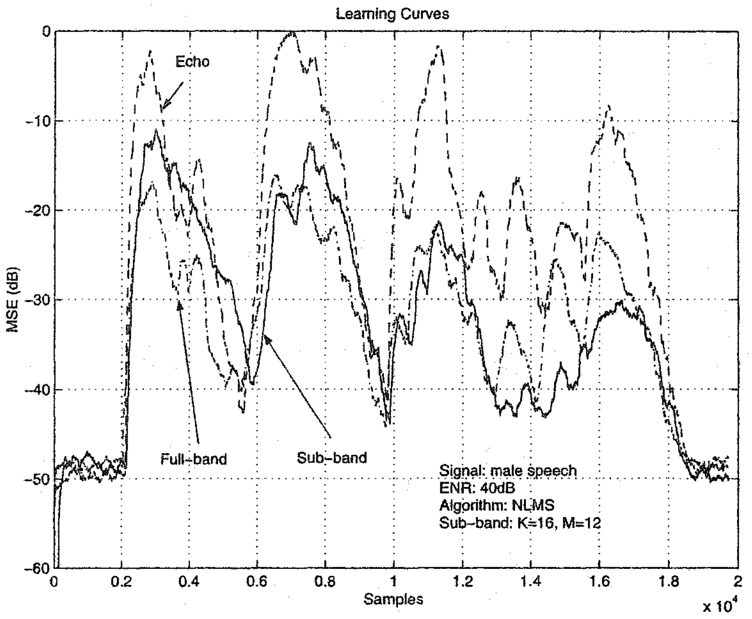

Fig. 6.8 Learning curves for a speech signal

its full-band counterpart, where the channel for now is assumed to be linear. The NLMS algorithm is used for adaptive filtering since it is simple and predictable. For the proposed subband structure, the number of filter banks used is $K = 16$, and the decimation rate is $M = 12$. The full-band adaptive filter length is $L = 300$ taps. Accordingly, the subband adaptive filter length is $L/M = 300/12 = 25$. The step-size μ is set to 0.9 both in full-band and subband.

Figures 6.7 and 6.8 respectively display the learning curves with white noise excitation and with speech excitation. The performance of both the full-band and subband structures are similar when excited by white noise. When excited by speech, the results were not as straightforward since the subband NLMS performance depends on the properties of the speech segment. However, in general, the performance of the subband NLMS is comparable to that of full-band NLMS in the conventional linear channel.

Secondly, because the NLMS and the AP have their own distinguishing merits (i.e., the former has the lowest computational complexity while the latter has the best tracking property), they are also implemented in the subband structure in order to see if they still work well with significantly less computational burden. With simulations (where the echo path is linear) paralleling those in the full-band case in 3.3.2, results shown in Figures 6.9 to

6.10 imply that the tracking capabilities of both AP and NLMS are worse for the subband scheme than for the full-band scheme. Accordingly, the performance of the subband scheme shouldn't be better than the full-band scheme.

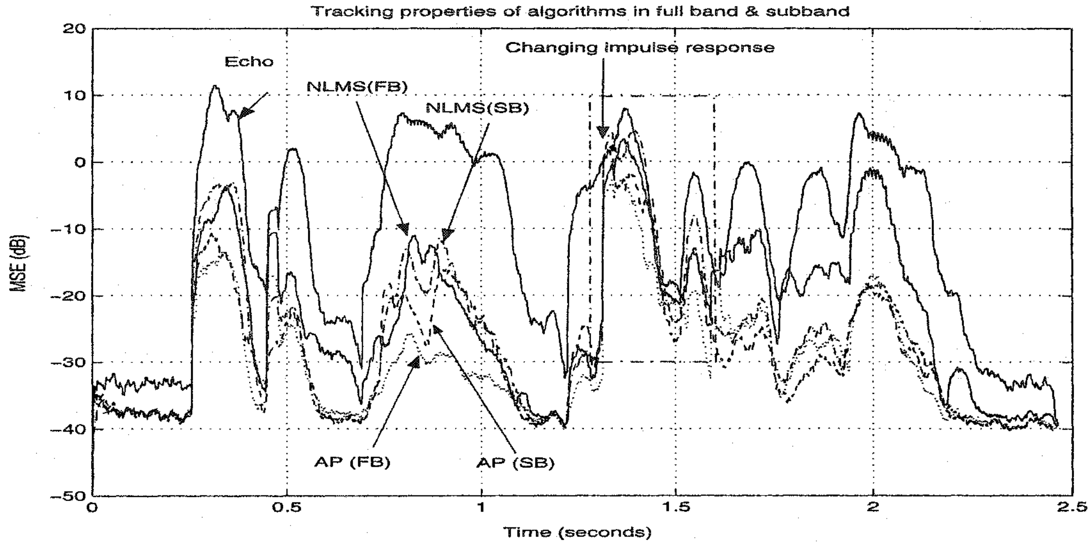

Fig. 6.9 Tracking properties of AP and NLMS in full/subband (ENR=40 dB).

The subband algorithms' poorer tracking capabilities may be explained by the fact that the signals are downsampled by a factor of M when they are fed into the subbands. This makes adaptation in the subband slower than in the full-band; therefore the tracking capabilities of the adaptive filters in the subband structure are degraded. According to the relation between the tracking capability and the achievable MSE in the nonlinear channel (see 3.3.2), the acoustic echo canceller in the subband structure has a poorer performance than its full-band counterpart in the nonlinear channel.

Finally, when vocoders are present along the echo path, the results are plotted in Figure 6.11. It is observed that the MSE of each subband algorithm is larger than that of its full-band counterpart, which is consistent with the above analysis. Compared to their full-band counterparts, the MSE is about 10 dB higher for the subband AP, and 5 dB higher for the subband NLMS.

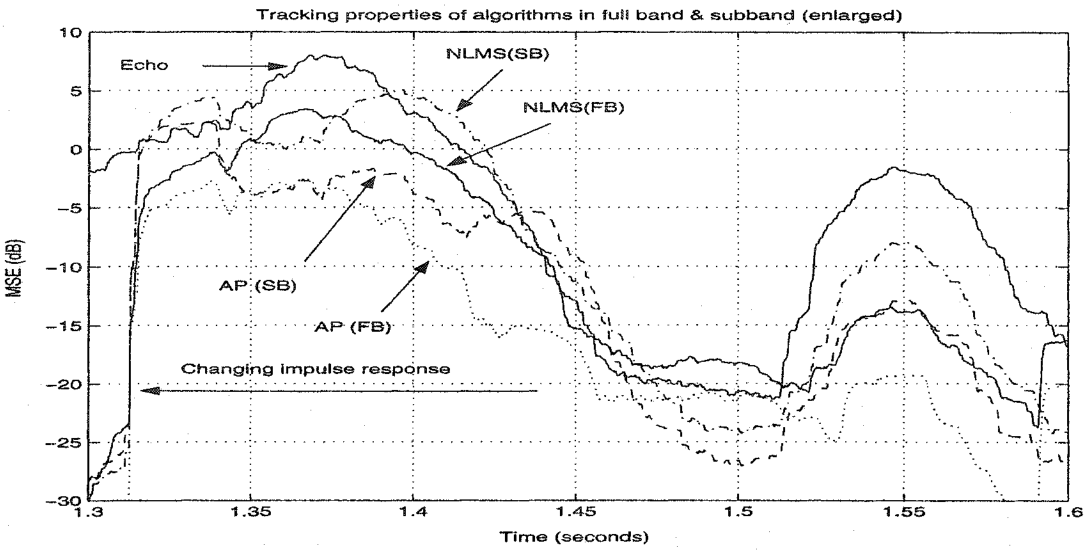

Fig. 6.10 Tracking properties of the AP and the NLMS in full/subband (enlarged from Figure 6.9).

6.3.2 The structure of the combined AEC system

Figure 6.12 shows the combined AEC in subband, where both the far-end signal $x(n)$ and the near-end signal $d(n)$ are split into subbands by analysis banks. The adaptive filtering algorithm is applied in each subband at a decimated rate, and the subband residual echo signal is further attenuated by individual over-weighted subband Wiener filters before they are recombined by a synthesis bank to create a full-band output signal $\hat{s}(n)$ at the original rate. Since the microphone signal may contain a near-end speech component that will go through the cascade of an analysis and a synthesis bank prior to its transmission, a near-PR property is necessary for the filter banks [18].

Uniform DFT filter banks are employed in the combined AEC system since they provide a simple method for designing filter banks suitable for subband AEC applications. Important design requirements include arbitrary oversampling in decimation, near-PR property of the combined analysis and synthesis banks, low complexity of implementation, and low processing delay. Additional details can be found in [48, 16].

The NLMS algorithm is very popular in practical AEC applications due to its predictable robust behaviour, simple implementation and low computational requirement. In order to compensate for the echo attenuation loss caused by the nonlinearities of the echo

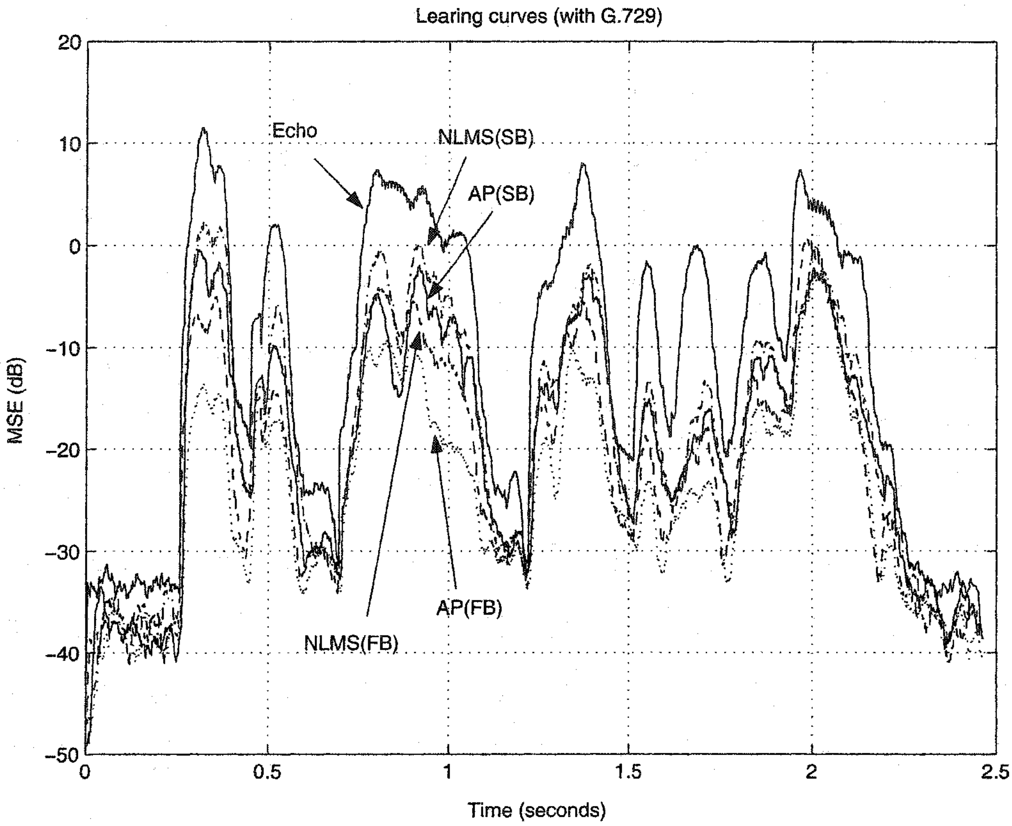

Fig. 6.11 Performance of the AP and NLMS in full/subband in the presence of vocoders (ENR=40 dB)

path and to reduce the computational complexity for the use of low-cost real-time DSP processors, we integrate NLMS and the post-filter in the subband structure. The algorithm is summarized in Algorithm 8, where all the variables are the components of signals in subband, and the adaptive filter in the kth bank is assumed to have a length of L_k, which may be different for each bank. The parameters μ, γ and α denote the step-size, the forgetting factor and the attenuation factor, respectively. A small constant ρ is introduced to prevent the division from overflow. Note that other adaptive filtering algorithms, e.g., the affine projection algorithm, may also be employed in the structure shown in Fig. 6.12. However, more computational capacity may be required.

Note that the subband signals at the output of the uniform DFT analysis banks satisfy a conjugate symmetric property: the ith subband signal is the complex conjugate of the

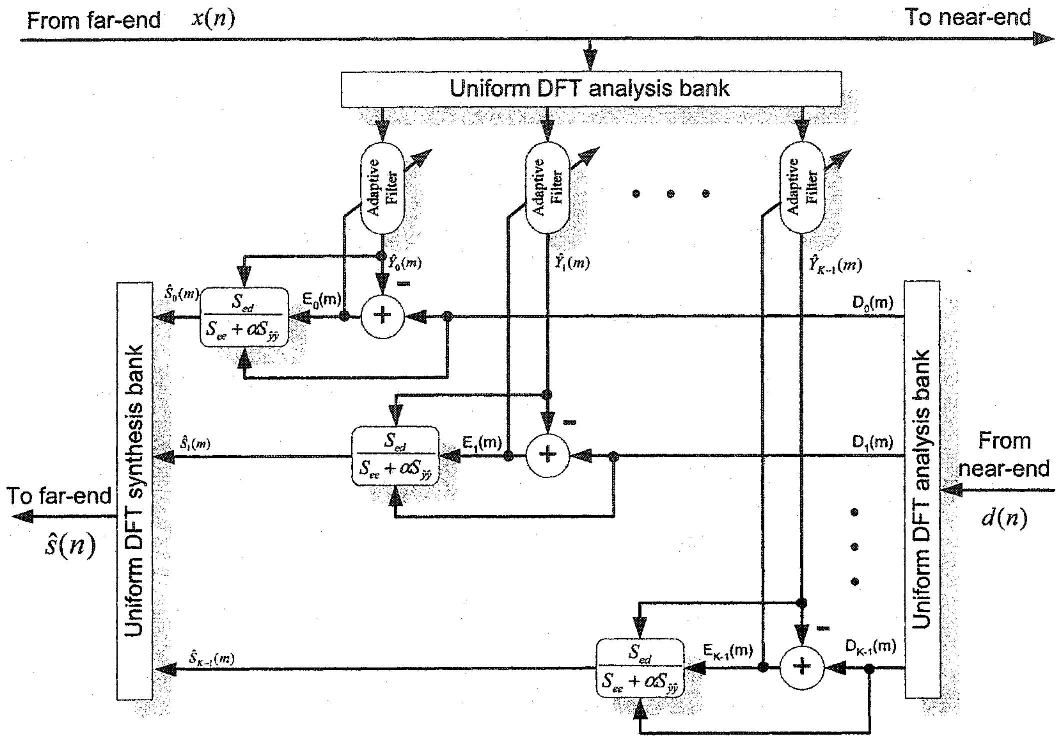

Fig. 6.12 The diagram of the combined AEC in a subband.

$(K+2-i)$th for $1 < i \leq K/2$ (assuming the number of the filter banks, K, is even), while the signals in the first and the $(K/2+1)$th banks are real. Thus, there are a total of $K/2+1$ independent subband signals: two real and $K/2-1$ complex, which only need $K/2+1$ adaptive filters running in these subbands. Furthermore, FFT is used to perform the DFT, resulting in more computational savings.

As before, computational complexity is measured in terms of operations, where one operation is defined as one real multiplication plus one real addition. The total computational complexity of the proposed AEC system consists of three components: analysis/synthesis filtering, adaptive filtering, and post-filtering in each subband.

Suppose that the decimation factor is M $(M < K)$ and that the number of taps of all subband filters is chosen equal to L/M, where L is the length of the full-band adaptive filter, which is assumed to match the duration of the echo path. Then the operations per every M input samples for the above three components are respectively $3K(L_0/2 + 2\log_2 K)$,

Algorithm 8 NLMS plus post-filtering in a subband

```
Initialization:
$\mathbf{W}_k(1) = \mathbf{0},\ k = 1, 2, \ldots, K$, where
$\mathbf{W}_k(m) = [W_k^{(1)}(m), W_k^{(2)}(m), \ldots, W_k^{(L_k)}(m)]^T$
Recursion:
for $m = 1, 2, \ldots$ do
    for $k = 1, 2, \ldots, K$ do
        $\mathbf{X}_k(m) = [X_k(m), X_k(m-1), \ldots, X_k(m - L_k + 1)]^T$
        $\hat{Y}_k(m) = \mathbf{W}_k^H(m)\mathbf{X}_k(m)$
        $E_k(m) = D_k(m) - \hat{Y}_k(m)$
        $\mathbf{W}_k(m+1) = \mathbf{W}_k(m) + \frac{\mu}{\mathbf{X}_k^H(m)\mathbf{X}_k(m) + \rho} E_k^*(m)\mathbf{X}_k(m)$
        $S_{ed}^{(k)}(m) = \gamma S_{ed}^{(k)}(m-1) + (1-\gamma)E_k(m)D_k^*(m)$
        $S_{\hat{y}\hat{y}}^{(k)}(m) = \gamma S_{\hat{y}\hat{y}}^{(k)}(m-1) + (1-\gamma)|\hat{Y}_k(m)|^2$
        $\hat{V}_k(m) = \frac{S_{ed}^{(k)}(m)E_k(m)}{S_{ee}^{(k)}(m) + \alpha S_{\hat{y}\hat{y}}^{(k)}(m)}$
    end for
end for
```

$4(K-1)L/M$ and $6(K-1)$. Therefore, the computational complexity of the proposed algorithm (based on NLMS), in common units of operations per sample (OPS), is obtained:

$$\frac{3K(L_0/2 + 2\log_2 K) + 4(K-1)L/M + 6(K-1)}{M}. \tag{6.26}$$

We note that the new algorithm achieves remarkable computational savings, if compared to the full-band counterpart which usually needs $2L + O(L_p)$ OPS, where L_p denotes the size of the DFTs, as required in the practical full-band realization of the post-filter (5.46). Furthermore, the seamlessly integrated post-filter in a subband only requires the slightly extra computational capacity of $6(K-1)/M$ OPS, compared to the pure subband NLMS.

6.3.3 Experiments

Three AEC algorithms, namely, the subband NLMS, and NLMS plus post-filter, both in full-band and subband, were tested on a platform where a low-cost loudspeaker played in a high volume so that the entire echo path presented certain nonlinearity.

In the implementation of the subband adaptive filtering algorithms, the number of filter banks and the decimation factor were $K = 16$ and $M = 12$, respectively. The prototype filter was obtained by using the DFT method to interpolate the QMF filter 16A [16], which has $L_0 = 16$ parameters, by a factor $K/2 = 8$. Hence, the length of the prototype filter is 128. For simplicity, the adaptive filters in subband were set to the same length, i.e., $L/M = 300/12 = 25$, where $L = 300$ was the length of the adaptive filter in full-band. The parameters of α and γ in Algorithm 8 were 5.0 and 0.8, respectively. The step-size μ was set to 0.9 for all the algorithms.

Under the above conditions, the computational complexity of the proposed algorithm, as shown in (6.26), can be calculated, resulting in about 200 OPS, that is: about 4% more than the pure subband NLMS, but only about 1/3 the complexity of its full-band counterpart (NLMS with post-filtering).

The results shown in Fig. 6.13 reveal that the proposed algorithm outperforms the pure subband NLMS in terms of significant acoustic echo suppression, and especially when the nonlinearity of the echo path cannot be neglected. Furthermore, the proposed algorithm exhibits a level of echo attenuation that is comparable to its full-band counterpart. A little distortion of the near-end speech is introduced by the use of the post-filtering technique, both in the full-band and in the subband, when the double-talk occurs. However, the distortion is not significant, as illustrated in Fig. 6.14, and thus it is almost imperceptible for the far-end user when he/she is talking at the same time.

As it is shown, the proposed algorithm significantly reduces the computational complexity, compared to its full-band counterpart. However, similarly to other subband adaptive filtering algorithms, some processing delay is introduced by the proposed scheme, which is 16 ms in the above experimental implementation. We note that this amount of delay is acceptable in most AEC applications.

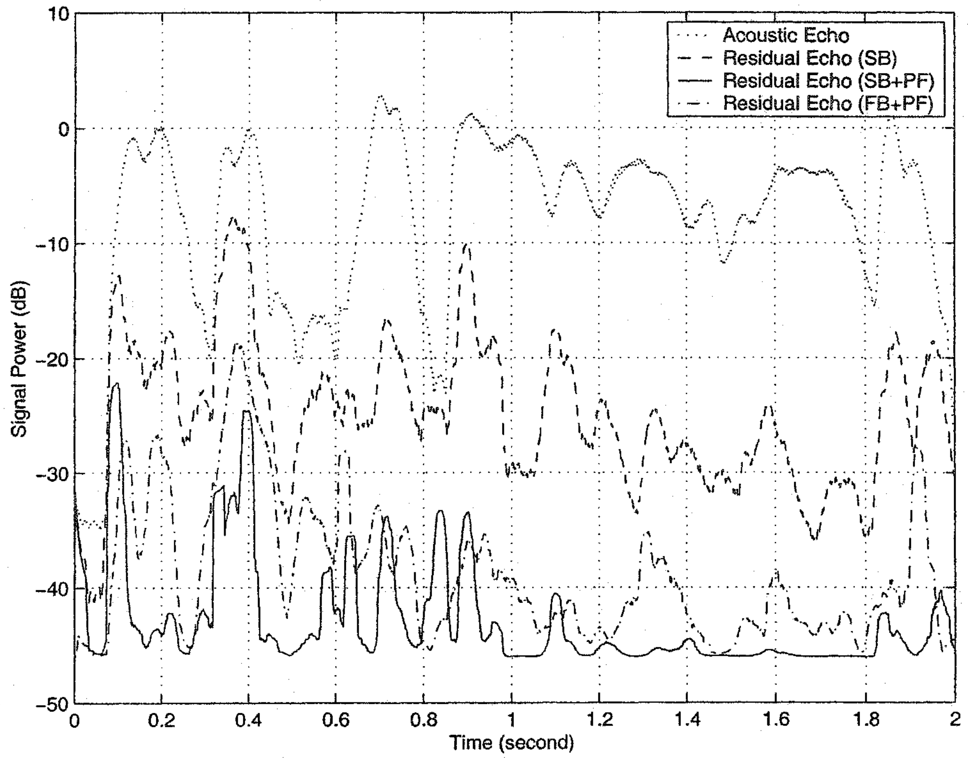

Fig. 6.13 Performance comparison of AEC system (based on NLMS): NLMS in subband (SB), NLMS plus post-filter in full-band (FB+PF) and NLMS plus post-filter in subband (SB+PF).

6.4 Conclusion

Filter banks with oversampling rates in the subbands are suitable for subband adaptive filtering in AEC. In this chapter, the use of NLMS algorithms with uniform DFT filter banks has been thoroughly studied. A practical method for prototype filter design is presented that satisfies the following properties: (a) near-PR property with oversampling; (b) very simple design procedure. Also, modifications are made to simplify the implementation of the weighted overlap-add subband structure, where the NLMS or AP algorithm is employed in the subband. Compared to the full-band AEC system, the subband system reduces computational complexity by a factor of M^2/K (where M is the decimation rate and K is the number of filter bank channels).

The performance of the subband AEC system and its full-band counterpart is com-

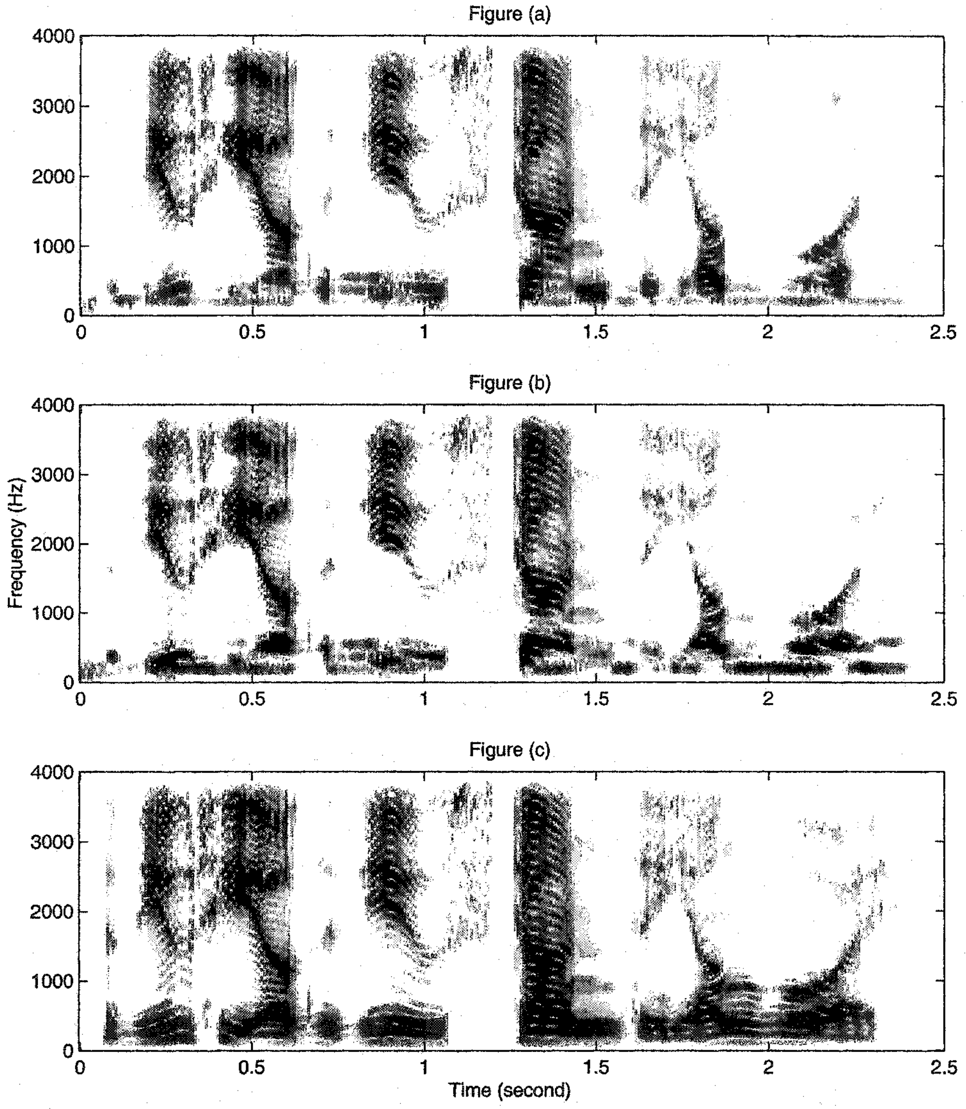

Fig. 6.14 Spectrograms: (a) output of combined subband system; (b) output of combined full-band system; (c) original (clean) near-end speech (*The beauty of the view stunned the young boy*).

pared. It is observed that in the nonlinear channels, the full-band adaptive filter has better performance in terms of a lower MSE because of its better tracking capability.

A combined subband AEC system proposed in this chapter significantly attenuates acoustic echo in the nonlinear channels. Furthermore, the proposed AEC system seamlessly integrates the post-filtering technique with the adaptive filtering technique in subband, resulting in a much higher echo attenuation with only a small increase in computational complexity compared to conventional AEC systems.

Similarly to other subband adaptive filtering algorithms, some processing delay is introduced by the proposed scheme, which is 16 ms in our experimental implementation. However, we note that this amount of delay is acceptable in most AEC applications.

Chapter 7

Psychoacoustic approach in the AEC system

Various post-filtering techniques used in AEC have been discussed in Chapter 5. A post-filter combined with a conventional acoustic echo canceller can significantly suppress the acoustic echo in nonlinear channels. However, Wiener-type post-filters introduce undesirable musical noise. In order to alleviate this problem, masking properties of the human ear can be exploited in speech enhancement [133]. This psychoacoustic approach attempts to minimize distortion by attenuating the audible components and smoothing the peaks of the enhanced signal by spectral and temporal averaging. This chapter proposes a combined AEC scheme that incorporates a robust post-filter to exploit the masking properties of the human ear. The new AEC system has the following advantages: significant acoustic echo suppression, no perceptual musical residual, and low perceptual distortion in nonlinear channels.

7.1 The human auditory system

The human ear converts sound waves into nerve impulses, and these impulses are interpreted by the brain as sound. The ear can perceive sounds in the range of 20 to 20,000 Hz. While the human auditory system has many functions, this work only focuses on the masking property of the ear as applicable to AEC.

The human ear consists of three parts: the outer ear, the middle ear, and the inner ear, as illustrated in Figure 7.1. The outer ear channels sound waves through the ear canal to the eardrum. The eardrum is a thin membrane stretched across the inner end of the canal. Air pressure changes in the ear canal cause the thin membrane to vibrate. These vibrations are transmitted to three small bones called ossicles. The ossicles are located in the air-filled middle ear and conduct the vibrations across the middle ear to another thin membrane called the oval window. The oval window separates the middle ear from the fluid-filled inner ear. The inner ear houses the cochlea, a spiral-shaped structure that contains the Organ of Corti - the most important component of hearing. The Corti sits in an extremely sensitive membrane called the basilar membrane. Whenever the basilar membrane vibrates, small sensory hair cells inside the Corti are bent, which stimulates the sending of nerve impulses to the brain.

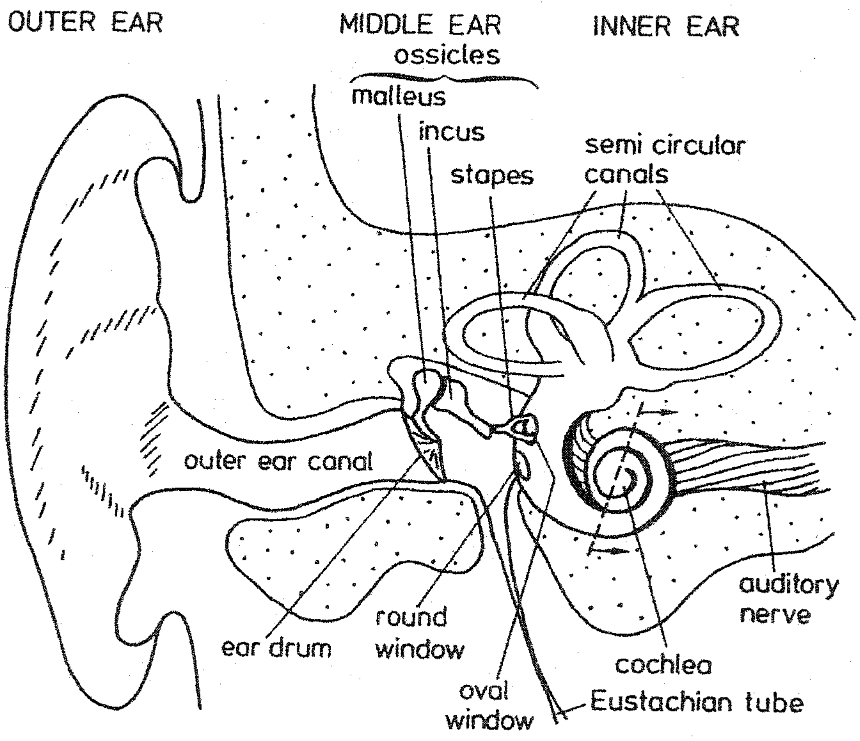

Fig. 7.1 The structure of the peripheral auditory system [134].

The outer ear includes two parts: an external part called the pinna, and the ear canal called the external auditory meatus [135]. The pinna funnels the sound, and by its asymmetric shape makes the ear more receptive to sounds in front of the head than behind. The external auditory meatus acts as a quarter wave resonator to boost the high frequencies of the sound it receives. The length of a male adult's canal is approximately 2.5 cm, and thus the first resonance is about 3440 Hz. A female or child's ear canal would probably be shorter than 2.5 cm, resulting in resonance at higher frequencies. This resonance amplifies sounds in the 3-5 kHz range by up to 15 dB [136]. High frequency emphasis provided by the outer ear is helpful for perception of sounds that have important information in the frequency range above 2 kHz (e.g., fricatives).

The eardrum, called the tympanic membrane, marks the beginning of the middle ear that contains three ossicles, namely: malleus, incus and stapes. These ossicles are connected to one another, forming the ossicular chain. The eardrum is responsive to small pressure variations across a wide range of frequencies. On the internal side of the eardrum is the ossicular chain which bridges the space between the eardrum and the cochlea in the inner ear. The major function of the middle ear is to efficiently transfer sound from the air to the fluids in the cochlea. In other words, the middle ear acts as an impedance-matching device, which is accomplished mainly by the difference in effective areas of the eardrum and the oval window, as well as the level action of the ossicles [137].

The most important part of the ear is the cochlea located in the inner ear. It is a tube filled with incompressible fluids and has bony rigid walls. The cochlea is divided along its length by two membranes, namely, the Reissner's membrane and the basilar membrane. One end of the cochlea where the oval window is situated is called the base, while the other end, called the apex, is the inner tip. A small opening called the helicotrema between the basilar membrane and the walls of the cochlea at the apex ensures that the fluid can flow between the two main chambers of the cochlea. Pressure variations applied by the stapes rocking in the oval window are translated into pressure variations within the fluids of the cochlea, resulting in displacements of the basilar membrane. On the basilar membrane lies the Organ of Corti, which contains rows of sensory hair cells. When hairs at the top of the hair cells are bent by the movement of the basilar membrane, nerve impulses are stimulated in the nerve fibers, and then transmitted to the brain via the auditory nerves.

The behaviour of the basilar membrane is of primary interest to this work. The change in stiffness from the base to the apex of the basilar membrane creates a mechanical sound analyzer. As shown in Figure 7.2, high frequency sound causes the narrow base end of the membrane to vibrate; medium frequencies cause the membrane in the middle cochlea to vibrate; low frequencies cause the whole membrane to vibrate. The cochlea is able to map frequencies onto certain locations on the basilar membrane. The sensation of pitch is a function of the vibration location on the basilar membrane.

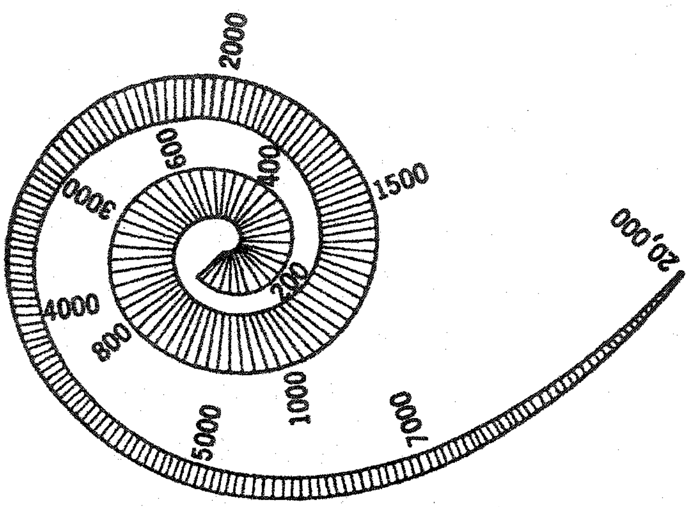

Fig. 7.2 The diagram of the basilar membrane with maximum amplitude of response to different frequencies in Hz. [135]

Each point on the basilar membrane can be regarded as a bandpass filter with a certain centre frequency and bandwidth. Experiments in hearing indicate that this bandwidth is not constant across all the points, but increases in proportion with the centre frequency [138]. The basilar membrane behaves like a Fourier analysis that breaks a complex sound into its sinusoidal components when the frequency differences of these components are large enough [137]. However, when the component frequencies are too close, the basilar membrane is no longer able to distinguish them. The frequency separation necessary for the resolution of two tones is proportional to the centre frequency.

7.2 Masking

Masking is a very important phenomenon where the perception of one sound, known as the maskee, is obscured by the presence of another, known as the masker [50]. For example, low volumes are sufficient for conversations in quiet environments, but not in noisy environments. This is because strong noises mask the sounds. Two types of masking effects have been identified: simultaneous masking and temporal masking [134]. The former occurs when two sounds are simultaneously present, while the latter occurs when a short intervening delay is introduced between the two sounds. Simultaneous masking is more prominent, but temporal masking is also considered in sophisticated perceptual models [139].

7.2.1 Critical bands

Critical bands were originally introduced by Fletcher [140]. This concept explains the masking of a narrow band (sinusoidal) signal by a wideband noise source. Based on the basilar membrane behaviour (i.e., each location on the basilar membrane responds to a limited range of frequencies), Fletcher suggested that the peripheral auditory system can be considered as a bank of non-uniform bandpass filters. The human ear groups the received sounds into bands of frequencies, and sounds within a critical band blur together.

A large number of experiments have all confirmed similar estimations for both the absolute width of a critical band and the way the critical band varies as a function of frequency [134]. Critical bandwidth approximately corresponds to 1.3 mm spacing along the whole length of the 32 mm basilar membrane. 24 critical bands can model the basilar membrane well, covering the audible frequency range to 16 kHz. To convert acoustic frequency resolution to perceptual frequency resolution, the Bark scale (also called the critical band rate) is introduced, where one Bark covers one critical bandwidth. Table 7.1 indicates the measured critical bandwidths and the corresponding critical band rates.

Since our research focuses on narrow-band speech signals, only critical bands up to 4 kHz are listed in Table 7.1. It can be shown that the critical bandwidth, denoted $\mathrm{BW}(f)$ where f is the frequency, is approximated by the following expression:

$$\mathrm{BW}(f) = \begin{cases} 100\ \mathrm{Hz}, & f < 500\ \mathrm{Hz} \\ 0.2f\ \mathrm{Hz}, & f \geq 500\ \mathrm{Hz} \end{cases} \tag{7.1}$$

Similarly, conversion between the Bark frequency scale and real frequencies can be approximated using the analytical expression [134]

$$z = 13\arctan(0.00076f) + 3.5\arctan((f/7500)^2), \tag{7.2}$$

where z is frequency in Bark and f is frequency in Hertz.

Table 7.1 Critical band rate, Centre frequencies and Bandwidths [134].

Critical band rate (Bark)	Centre frequencies (Hz)	Critical Bandwidths (Hz)
1	50	100
2	150	100
3	250	100
4	350	100
5	450	110
6	570	120
7	700	140
8	840	150
9	1000	160
10	1170	190
11	1370	210
12	1600	240
13	1850	280
14	2150	320
15	2500	380
16	2900	450
17	3400	550
18	4000	700

7.2.2 Masking effects

As explained earlier, masking effects consist of simultaneous masking and temporal masking. Classical experiments that measure masking have explored all combinations of maskers and maskees (tone or noise), where the tone could be a pure tone or a complex tone, and the noise could be broadband, narrowband or low/highpass.

Properties of simultaneous masking are summarized as follows:

- Lower frequencies tend to mask higher frequencies at high levels; this is called the upward spread of masking [141]. This behaviour is reversed at low levels.
- A narrow band noise masker is more effective than a tonal masker, which can be explained by the rate of intensity fluctuation [142].
- The tonal masker and the narrow band noise masker have different masking patterns: for the tonal masker, the slope at lower frequencies becomes more steep with increas-

ing masker level, whereas the narrow band noise masker has relatively invariant low frequency slopes.

Temporal masking (also called nonsimultaneous masking) consists of postmasking and premasking. Postmasking occurs when the maskee follows the masker, while premasking occurs when the maskee precedes the masker. The prominence of nonsimultaneous masking is affected by the following factors [137]:

- *The delay between the masker and the maskee.*
  The shorter this delay is, the more amount of masking occurs. Furthermore, nonsimultaneous masking mostly occurs within about ±100 ms of the masker onset or termination.

- *The frequencies of the signals.*
  Similar to simultaneous masking, nonsimultaneous masking is also influenced by the relation between the frequencies of the masker and the maskee.

- *The duration of the masker.*
  The amount of masking increases with longer masker duration within the range 1 $\sim$ 20 ms, but beyond this range it is approximately independent of the masker duration.

In addition, if the time interval between masker and maskee is very short, more postmasking than premasking occurs. Moreover, increasing the masker intensity does not produce a corresponding increase in the amount of nonsimultaneous masking; this is different from the case of simultaneous masking.

7.2.3 Masking threshold

Below the masking threshold, the maskee would become inaudible in the presence of the masker. Masking models of the human ear are used to calculate the masking threshold. Some perceptual models, such as the Johnston model [143], only take simultaneous masking into account since it is more prominent. More sophisticated perceptual models, such as the PEAQ model [139], consider both simultaneous and nonsimultaneous masking.

In the context of speech coding, a more sophisticated model usually results in a better performance in terms of perception. However, in the application of AEC, we prefer to start with a simple model, i.e., the Johnston model, due to its easy implementation. Later, more perceptual models will be employed in AEC for comparison. Here, we describe the calculation of the masking threshold based on the Johnston model as follows:

a) Frequency analysis in the Bark scale

Let $v(n)$ be the masker signal. Its complex spectrum, denoted $V(k;m)$, is obtained by performing DFT analysis (FFT is used in practice) where k and m represent the indices in the frequency domain and the time domain. Next, the energy spectrum, i.e., $|V(k;m)|^2$, is added up in each critical band, where the number of the frequency bins for each critical band is determined by Table 7.1. Hence, the energy in each critical band is

$$B_i(m) = \sum_{k=k_i^{(l)}}^{k_i^{(u)}} |V(k;m)|^2, \tag{7.3}$$

where $B_i(m)$ is the energy in critical band i, and $k_i^{(l)}$ and $k_i^{(u)}$ are the lower boundary and the upper boundary of critical band i, respectively. The specific values of $k_i^{(l)}$ and $k_i^{(u)}$ depend on the length of the DFT. A typical case can be found in [133].

b) Masking between different critical bands

In addition to signals being masked by other signals within the same critical band, the effect of masking across critical bands also needs to be taken into account. This can be performed by convolving the critical band energy with a spread function SF_{ij}, resulting in the spread critical band spectrum

$$C_i(m) = SF_{ij} * B_i(m), \tag{7.4}$$

where $*$ denotes the convolution operation. At intermediate speech levels, the lower skirt of the spread function has a slope of of +25 dB per critical band, and the upper skirt has a slope of -10 dB per critical band. The spread function can be analytically expressed as [144]:

$$10\log_{10} SF_{ij} = 15.81 + 7.5(|i-j| + 0.474) - 17.5\sqrt{1 + (|i-j| + 0.474)^2}, \tag{7.5}$$

where i is the bark frequency of the masker, and j is the bark frequency of the maskee. For wide-band audio, SF_{ij} is calculated for $|i-j| \leq 25$, while for narrow-band speech, $|i-j| \leq 18$.

c) Spread masking threshold

The masking effects of tone masking noise and noise masking tone are different. The former is estimated to be $(14.5+i)$ dB below the spread critical band spectrum $C_i(m)$ [144], while the latter is about 5.5 dB less than $C_i(m)$ [142]. Other cases fall between these two situations (i.e., neither entirely tonelike nor completely noiselike). To deal with all cases, an offset for the masking energy in critical band i is introduced by Johnston [143]:

$$O_i(m)(dB) = \alpha(14.5+i) + (1-\alpha)5.5, \tag{7.6}$$

where $\alpha \in [0,1]$ is a weight for different signals: $\alpha = 0$ for noise masking a tone, and $\alpha = 1$ for tone masking noise; other values of α indicate signals in between.

Since noise has a flat spectrum while tone has a peak-shaped spectrum, the Spectral Flatness Measure (SFM) can be used to determine if the signal is noiselike or tonelike:

$$SFM(dB) = 10\log_{10}\frac{Gm}{Am}, \tag{7.7}$$

where Gm and Am denote the geometric mean and the arithmetic mean of the power spectrum, respectively. Hence, the coefficient α can be defined as

$$\alpha = \min(\frac{SFM}{SFM_{max}}, 1), \tag{7.8}$$

where SFM_{max} is set to -60 dB. The spread masking threshold in critical band i is obtained by subtracting the offset from the spread critical band spectrum:

$$T_i(m) = 10^{\log_{10}(C_i(m)) - O_i(m)/10}. \tag{7.9}$$

d) Renormalization

The spread masking threshold must be deconvolved back to the Bark domain. In practice, renormalization is used instead of deconvolution since the spread function increases the energy estimates in each band due to spreading. The renormalized masking threshold is obtained by multiplying each $T_i(m)$ with the inverse of the energy gain, assuming that each band has uniform energy.

Figure 7.3 illustrates the signal power of a speech segment and the corresponding threshold calculated using the Johnston model. The speech signal is quantized with 16 bits at an 8kHz sampling rate. The length of the FFT windows was 256 samples.

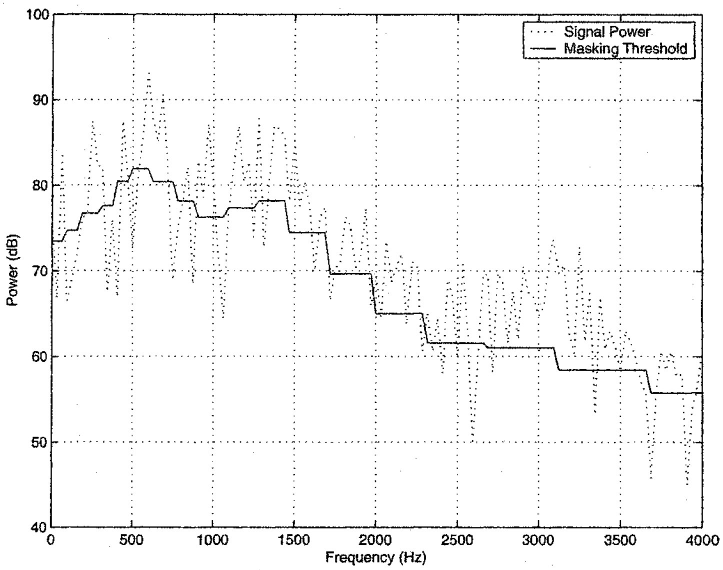

Fig. 7.3 Example of masking threshold for a segment of speech.

The simple perceptual model described here provides the basis for more sophisticated ear models such as the MPEG perceptual model [145] and the PEAQ model [139]. Both models have also been studied in our research.

7.3 Exploiting masking properties with post-filtering

As discussed earlier, a weak signal is inaudible when it is masked by a strong signal. This suggests that the near-end signal can be used to mask the nonlinear acoustic echo, as long as the echo power spectral density (PSD) is lower than the masking threshold. In other words, it is only necessary to suppress the part of the echo with a higher PSD until it falls below the masking threshold. Therefore, the post-filter can multiply the input signal spectrum by a set of variable gains, where the values of the gains are determined by both the masking threshold and the PSD of the acoustic echo. Figure 7.4 illustrates the psychoacoustic post-filter used in AEC.

Let $S_{\delta\delta}(k;m)$ denote the PSD of $\delta(n)$, which is the residual echo as defined in (5.3), where k is the index of frequency bins, and m is the index of frames in the time domain.

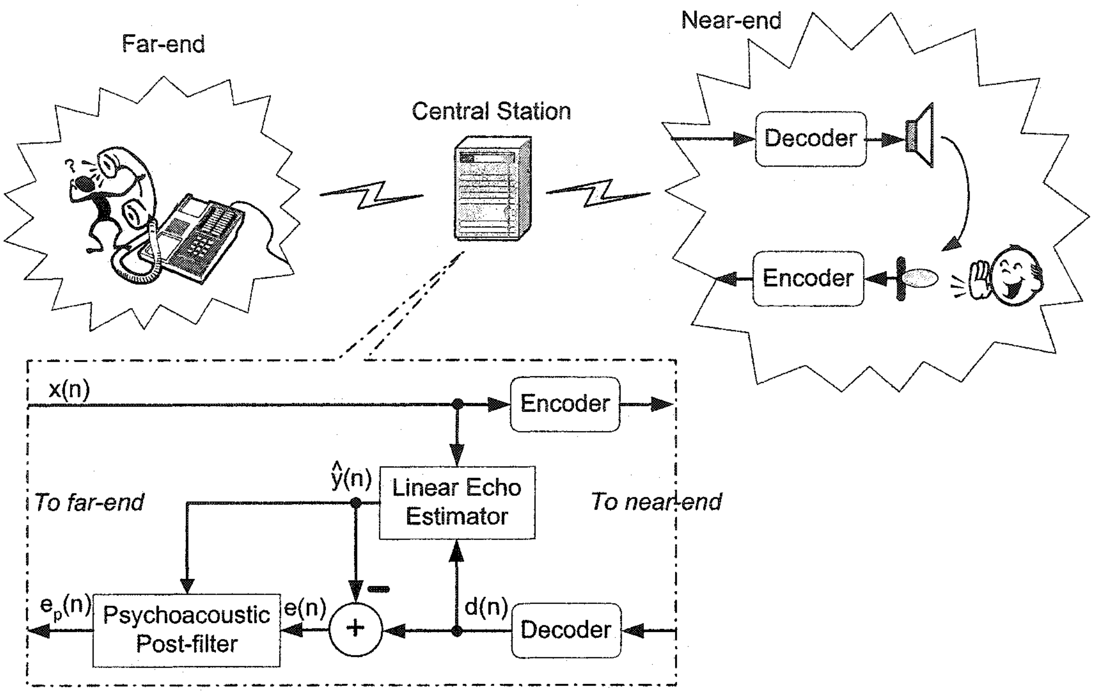

Fig. 7.4 Psychoacoustic post-filter in AEC.

Given the masking threshold $T(k;m)$, the gain of the post-filter is

$$G(k;m) = \min\left(\sqrt{\frac{T(k;m)}{S_{\delta\delta}(k;m)}}, 1\right). \tag{7.10}$$

The residual echo is then attenuated by passing the signal through the post-filter, resulting in a echo-suppressed signal given by

$$E_p(k;m) = G(k;m)E(k;m), \tag{7.11}$$

where $E(k;m)$ is the DFT of $e(n)$, the residual signal as shown in Figure 7.4. Finally, $E_p(k;m)$ is transferred back to the time domain and sent to the far-end user.

Ideally, the masking threshold is calculated from the near-end speech signal, but the exact near-end speech $\nu(n)$ and the PSD of the residual echo $\delta(n)$ are unknown. However, the information can be estimated from the available data. Here we propose to use the estimated near-end speech for the computation of both the masking threshold and PSD of

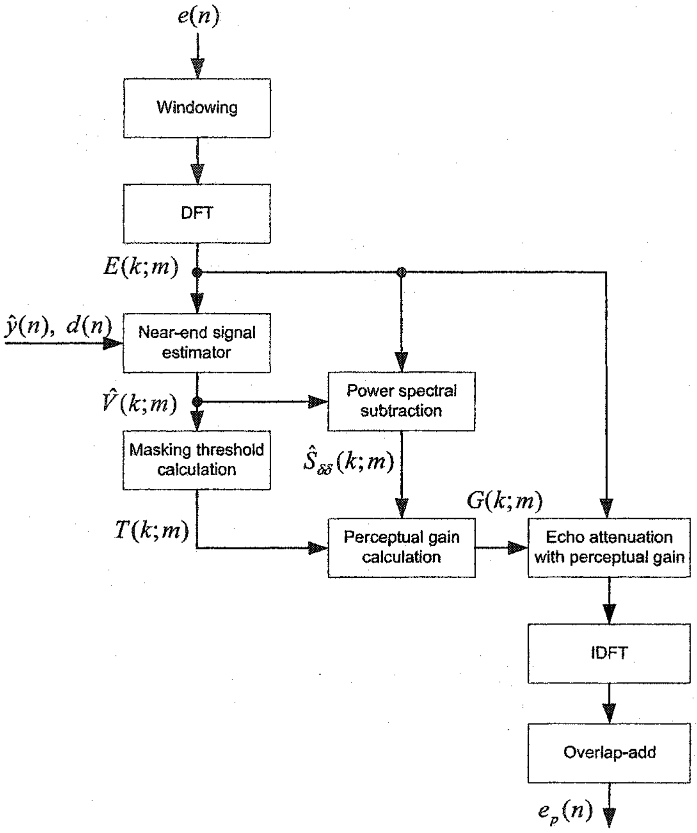
$e(n)$
Windowing
DFT
$E(k;m)$
$\hat{y}(n)$, $d(n)$
Near-end signal estimator
$\hat{V}(k;m)$
Power spectral subtraction
Masking threshold calculation
$\hat{S}_{\delta\delta}(k;m)$
$G(k;m)$
$T(k;m)$
Perceptual gain calculation
Echo attenuation with perceptual gain
IDFT
Overlap-add
$e_p(n)$

Fig. 7.5 Flow chart of psychoacoustic post-filter scheme.

the residual echo. Figure 7.5 shows the flow chart of the psychoacoustic post-filter scheme. The important steps are explained as follows.

a) Estimation of the near-end speech

As explained earlier, the estimated near-end speech is obtained by passing the residual signal through the optimal filter in (5.46), i.e.,

$$\hat{V}(k;m) = H(k;m)E(k;m). \tag{7.12}$$

Since most perceptual models are established in the frequency domain, $\hat{V}(k;m)$ does not need to be transferred back to the time domain at this stage.

b) Calculation of the masking threshold

The Johnston model [143] is chosen here to compute the masking threshold for its simplicity, but more sophisticated auditory models can also be employed for this application as well. The masking threshold $T(k;m)$ is obtained with the input of $\hat{V}(k;m)$.

c) Estimation of the PSD of the residual echo

Due to the echo path nonlinearities, estimating the PSD of the residual echo by conventional means for linear systems [146] is inaccurate. However, the estimated near-end speech signal can be used to find an approximation of the PSD $S_{\delta\delta}(k;m)$.

Using (5.13) and neglecting the background noise, $S_{ee}(k;m)$ is given by

$$S_{ee}(k;m) = S_{\nu\nu}(k;m) + S_{\delta\delta}(k;m). \tag{7.13}$$

Replacing $S_{\nu\nu}(k;m)$ by its estimation yields

$$\hat{S}_{\delta\delta}(k;m) = S_{ee}(k;m) - \hat{S}_{\nu\nu}(k;m). \tag{7.14}$$

In fact, this procedure is the power spectral subtraction algorithm. Similar to (5.20), the PSD of $\nu(n)$ is computed as

$$\hat{S}_{\nu\nu}(k;m) = \gamma\hat{S}_{\nu\nu}(k;m-1) + (1-\gamma)\hat{V}(k;m)\hat{V}^{*}(k;m) \tag{7.15}$$

d) Reduction of masking distortion

Since the residual signal $e(n)$ contains significant amounts of nonlinear and linear acoustic echo, the echo should be aggressively attenuated by choosing a large attenuation factor α in (5.46). On the other hand, the shape difference between $S_{\hat{y}\hat{y}}(k;m)$ and $S_{\delta\delta}(k;m)$ caused by the vocoder nonlinearity may over-attenuate the near-end speech, resulting in distortion.

The perceptual post-filter uses spectral and temporal averaging to smooth the residual signal, which effectively reduces the saliency of musical noise. Unfortunately, this may also impair the intelligibility of the near-end speech, because both the masking threshold and the PSD of residual echo are computed based on the estimated "clean" near-end speech which could be distorted by the optimal filter (5.46), as pointed out above, due to the over-attenuation. In order to reduce this masking distortion, a psychoacoustic post-filter is proposed that stops attenuating the residual acoustic echo a few decibels above the masking threshold. Consequently, the signal spectrum will be shaped to improve intelligibility. The modified post-filtering gain is

$$G(k;m) = \min\left(\sqrt{\frac{10^{P/10}T(k;m)}{S_{\delta\delta}(k;m)}}, 1\right), \tag{7.16}$$

where P (in dB) is the relaxation factor that depends on the attenuation factor μ: a larger μ results in a larger P. Experimental results show that P may be chosen between 0 and 10.

7.4 Results

To compare the performance of the proposed psychoacoustic post-filter with that of a Wiener-type post-filter (5.46) and other psychoacoustic post-filters such as [146] in the nonlinear channel, experiments were conducted based on the platform shown in Figure 7.4, where G.729 was chosen to be the vocoder. The performance was evaluated in terms of both objective and subjective criteria.

For the tests, two segments of real speech were used as the near-end speech and far-end signal. The LEM system of the test platform was simulated to represent the inside of a vehicle. The impulse response was about 40 ms long, corresponding to 300 taps at a sampling rate of 8 kHz. The linear acoustic echo estimator used the modified adaptive cross-spectral algorithm [27]. Other parameters were set as $\mu = 40$ and $P = 5$ dB.

For a realistic scenario, three situations were considered, namely: far-end single-talk, double-talk and near-end single talk. Fig. 7.6 displays the waveforms of the signals in these situations. The near-end signal $d(n)$ that consists of $y(n)$ and $v(n)$ is shown in Fig. 7.6(a). We note that the real $v(n)$ cannot be obtained in the double-talk situation when $y(n)$ and $v(n)$ are mixed, due to the non-linearities of the vocoder. However, $d(n)$ is a reasonable approximation of $v(n)$ when the acoustic echo is absent, which is plotted in Fig. 7.6(d).

The echo-suppressed signal $e_p(n)$, shown in Fig. 7.6(c), is the output of the proposed AEC system. Because the estimation of the residual echo plays the key role in the performance of the psychoacoustic post-filter, $S_{\delta\delta}(k;m)$ was also estimated by the approach in [146] for comparison. This led to a different echo-suppressed signal $e_p^r(n)$, which is dis-

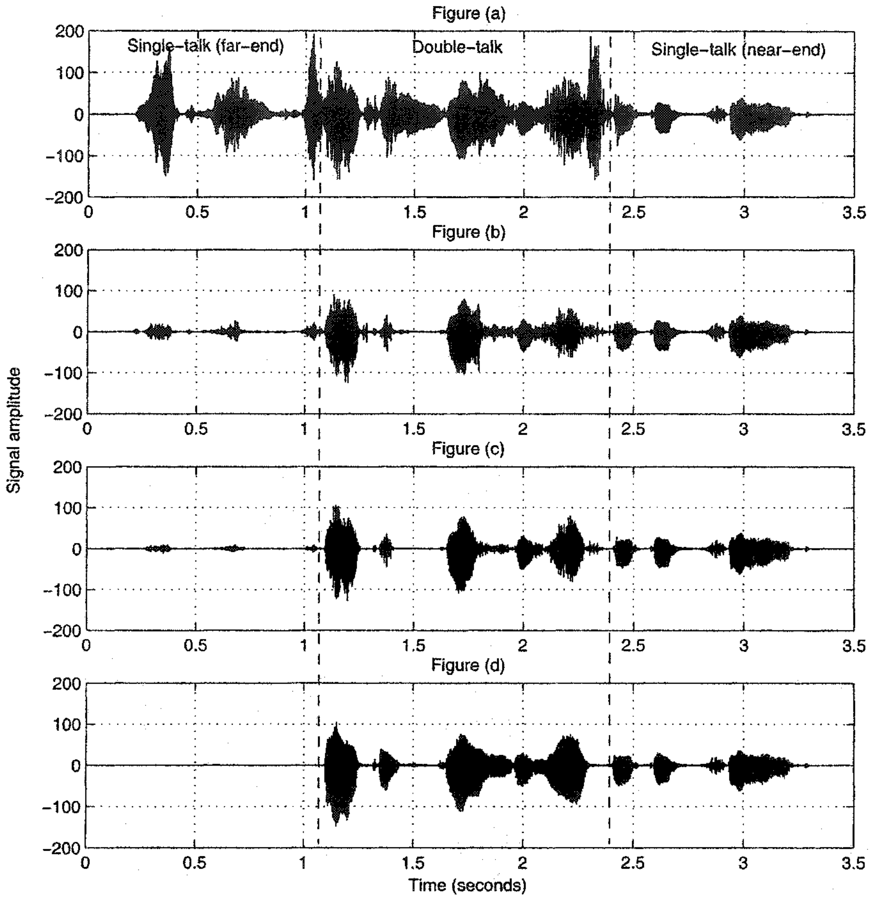

Fig. 7.6 Waveforms of signals: (a) near-end signal $d(n)$, (b) echo-suppressed signal $e_p^r(n)$, (c) echo-suppressed signal $e_p(n)$, (d) approximate near-end speech $v(n)$.

played in Fig. 7.6(b). Comparing $e_p(n)$ and $e_p^r(n)$ with the approximate $\nu(n)$ in Fig. 7.6, one can find that, in the presence of vocoders, the proposed AEC has advantages in terms of higher echo suppression and of less distortion to the near-end speech during the DT period. Moreover, we note that the psychoacoustic post-filter has very similar performance in terms of ERLE, compared to the Wiener-type post-filter.

Furthermore, informal listening tests were also conducted. The Wiener-type post-filter produced strong musical noise, although it remarkably suppressed the residual echo. On the contrary, the musical noise produced by the proposed psychoacoustic post-filter was almost imperceptible. A listening comparison between $e_p(n)$ and $e_p^r(n)$ was also done. It was found that $e_p(n)$ has better quality in terms of less residual echo, less musical noise and less distortion to the near-end speech during the DT period.

We note that the proposed AEC still brought somewhat perceptual distortion to the near-end speech, although it did not affect the intelligibility. This distortion can also be observed in the spectrograms shown in Fig. 7.7. However, in the practical scenario, this may not be a critical issue since the far-end user is not sensitive to the speech quality of the near-end user when he/she is speaking.

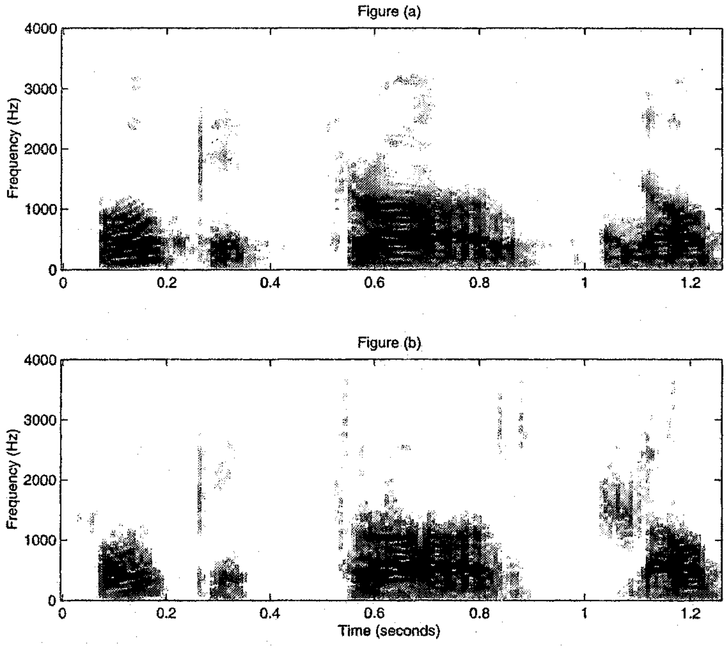

Fig. 7.7 Spectrograms of signals during the double-talk period: (a) original near-end speech, (b) echo-suppressed signal.

In this chapter, only the results with the Johnston perceptual model have been shown. However, some sophisticated perceptual models, e.g., PEAQ, which consider the temporal masking as well as the spectral masking, were also tested in our research. We note that no obvious improvement has been observed for the advanced models in this application. Therefore, the Johnston model is more suitable for the AEC since it has a simpler structure and lower computational complexity than those sophisticated models.

7.5 Conclusion

Based on the masking effect of the human auditory system, a new AEC system has been presented that combines a psychoacoustic post-filter with a conventional echo estimator for echo suppression over nonlinear channels, where codecs are cascaded along the echo path. Compared to the post-filters discussed in Chapter 5, the psychoacoustic post-filter significantly mitigates musical noise while achieving a similar amount of echo suppression. Simulation results and informal listening tests show that the proposed AEC achieves significant echo attenuation with little distortion to the near-end speech.

Chapter 8

Conclusion

This thesis studied the performance of AEC systems over nonlinear channels, where the channel nonlinearities are contributed by vocoders and loudspeakers. While this work mostly focused on the vocoder nonlinearities, the loudspeaker nonlinearities were also examined. This chapter provides a summary of the research done and gives directions for future work.

8.1 Summary of research contributions

This thesis first investigated the effects of vocoder nonlinearities on the performance of AEC systems. The AEC systems under study consisted of a full-band transversal finite impulse response (FIR) filter structure with their coefficients adjusted by means of several popular algorithms. Two common coding schemes were considered as part of the communication channel, namely CS-ACELP (G.729) and GSM. Simulations verified that these codecs produced similar nonlinear effects on AEC. Furthermore, it was found that the codec nonlinearities significantly reduce the achievable ERLE by 10 to 20 dB. However, closer investigation revealed that an adaptive echo canceller can achieve higher ERLE than a fixed echo canceller that has an impulse response identical to the LEM system's impulse response. Further simulation and analysis indicated that the codec nonlinearities caused the adaptive filter to drift away from the true echo path of the LEM. This may cause the echo residual to be larger than the original echo when the adaptation of the acoustic echo canceller is frozen, which is the case during DT in traditional AEC systems. In order to avoid this situation during DT in nonlinear channels, the AEC should be stopped (i.e., no cancellation) instead of simply freezing the adaptation of the AEC coefficients.

The adaptive filter behaviour led to the adoption of a local linearized model for analyzing the nonlinear echo path. Using this model, the AEC performance was linked to the tracking capabilities of the employed adaptive filtering algorithms: the faster an algorithm can track changes in the system response, the lower MSE it achieves. Comparative evaluation of some popular adaptive filtering algorithms showed that the AP algorithm (in its full-band version) achieves the lowest MSE in the presence of codecs. Furthermore, a projection order of two or three is sufficient for AP (in fact, no improvement was observed with higher orders). To further reduce the computational cost, the FAP algorithm can be used.

A saturation phenomenon was observed with the behaviour of ERLE versus the adaptive filter length. With codecs present, the largest gain in ERLE occurred at about 60% of the filter length. Therefore, there is no need to use a long adaptive filter in nonlinear channels. This is consistent with the theoretical analysis which showed that longer adaptive filters have worse tracking capabilities.

DT presents a more serious problem in nonlinear channels than in linear ones, hence it is important to find a solution for echo suppression during double-talk periods in nonlinear channels. For this purpose, the adaptive cross-spectral technique was investigated since it can exploit the correlation between the far-end signal and the acoustic echo, and it is robust to strong disturbances in the local region. Using this technique, a new variable step-size adaptive cross-spectral algorithm (VSS-ACS) was derived that employs a finite state machine to control the step-size. Simulation results showed that the new algorithm can suppress echo during DT without the need for a DT detector. However, similar to other adaptive filtering algorithms, this algorithm's effectiveness is degraded by the channel nonlinearities so that the echo suppression is insufficient.

In order to sufficiently suppress acoustic echo in nonlinear channels, post-filtering techniques were extensively studied in this research. Combined with VSS-ACS or conventional adaptive filters, post-filters can further reduce the residual echo by using the following approaches.

The Wiener-type post-filter: based on the MMSE criteria, we derived a Wiener-type post-filter with the aim of minimizing the near-end speech distortion while suppressing the processed residual echo to a predefined level. Experiments showed that the proposed post-filter remarkably compensated the insufficient echo attenuation of the adaptive filters caused by channel nonlinearities.

The spectral subtraction method: apart from speech enhancement, we explored the spectral subtraction method, which is based on the ML criteria, in the application of AEC. The performance of this method was analyzed in the viewpoint of post-filtering, and evaluated experimentally.

The subspace approach: we introduced the subspace method in the context of AEC, where the microphone signal is decomposed into an echo subspace and a near-end signal plus echo (mixed) subspace. The acoustic echo is suppressed by eliminating the echo subspace and attenuating the non-signal components in the mixed subspace. The corresponding structure and algorithm was proposed and tested. Significant echo attenuation was achieved.

The pitch extraction method: based on an auditory property, i.e., people are more sensitive to voiced speech than to unvoiced speech, we proposed a new post-filter. This post-filter exploits the speech analysis technique, namely pitch extraction from the residual echo, to further suppress the residual echo produced by the linear adaptive filter. Experiments showed that the residual acoustic echo was remarkably suppressed, especially during DT.

Among these post-filters, the Wiener-type filter and the spectral subtraction are easy to implement, while the subspace approach and the pitch extraction method need more computational capacity. The Wiener-type filter and the subspace method achieve more echo attenuation and bring in less distortion of the near-end speech.

This thesis also investigated using subband adaptive filters to reduce computational complexity. An improved design of the uniform DFT filter bank was proposed. In nonlinear channels, the performance of subband algorithms is not as good as full-band ones since subband algorithms have lesser tracking capabilities. To overcome this drawback, a Wiener-type post-filter was seamlessly integrated into the filter bank. Experimental results showed that this combination not only significantly suppresses the acoustic echo resulting from the channel nonlinearities, but it also greatly reduces computational complexity.

Finally, a psychoacoustic approach was explored to mitigate musical noise resulting from post-filtering. It was found that the psychoacoustic post-filter only needs to suppress the echo signal component that is at a higher level than the masking threshold, since any echo component below the masking threshold cannot be perceived by human ear. This approach was shown to significantly reduce near-end speech distortion while achieving good echo suppression.

8.2 Future research directions

This section presents future research directions on acoustic echo cancellation over nonlinear channels. The main issues to be considered include reducing computational complexity, improving acoustic echo attenuation, and reducing near-end speech distortion. Also, attention should be placed on studying new types of channel nonlinearities.

One interesting possibility is to investigate how the side information of vocoders may be used to improve the acoustic echo control system. New mechanisms could be developed to increase ERLE and reduce the computational complexity of echo cancellation algorithms.

Specific issues to be investigated in the future include the following:

- In the present AEC setup, decorrelation is used to increase the convergence rate of adaptive filters. Instead of trying to decorrelate the reconstructed signal at the output of the codec, it might be better to use the vocal tract excitation signal generated internally by the decoding algorithm (e.g., from the codebook) as the input to an NLMS adaptive filter.

- The vocal tract shaping filter, which uses parameters available from the codec, could be exploited in AEC. One possible approach is to use a secondary (or modified) error signal for adapting the transversal filter coefficients. This secondary error signal could be obtained by subtracting the adaptive filter output from a filtered version of the microphone signal. The filtering applied to the microphone signal could be derived from (and matched to) the vocal tract shaping of the loudspeaker signal. This way, any unwanted signal components originating from the local site (e.g, local noise or near-end talker) could be removed. This approach is motivated by the fact that the acoustic echo signal picked up by the microphone preserves many of the spectral and temporal attributes of the original loudspeaker signal.

- The use of psychoacoustic perceptual models in AEC brings in two issues, namely: significantly increased computational complexity and difficulty in effectively evaluating the performance. There are two possible ways to reduce this complexity: one approach is to develop a subband ear model suitable for AEC (an approximate model may be acceptable); another approach is to find a simple way to design a non-uniform (Bark scale) subband structure with low computational complexity, so the perceptual post-filter can be integrated with the adaptive filter in the subband. For the second issue, the method of perceptual evaluation of speech quality (PESQ) [147] may be investigated for the use of the AEC testing.

- Blind separation (ICA) techniques may be explored to separate acoustic echo and near-end speech, so that only the near-end speech is sent out. Some results of this approach have been presented for linear echo paths. However, reducing ICA's high computational complexity and adapting it for nonlinear channels remain challenging.

Moreover, applications of internet communications such as voice over IP from a personal computer are rapidly growing. For these applications, research on low-cost real-time implementations for general-purpose CPUs is very important. Today's PCs have processors that are powerful enough to handle a complete AEC system. In this case, various sources of nonlinearities need to be considered, such as loudspeakers, microphones, non-ideal A/D and D/A converters, audio amplifiers, and speech codecs. Furthermore, nonlinearities resulting from packet loss in voice over IP could be investigated.

In order to simplify the design and implementation of hands-free user terminals, more and more AEC devices are being implemented in a base station or a central station. To improve efficiency, however, signals will no longer be decoded in the base or central stations in the near future. Only bit-streams will be available to AEC devices. Exploiting information from these bit-streams will pose a new challenge for AEC.

Appendix A

Simplified structure of adaptive filter in subband

Intuitively, (6.19) and (6.24) in Chapter 6 may be removed so that the structure of the adaptive filter in subband can be simplified, resulting in reduced computational complexity. Here, rigorous proofs are given in the case of the NLMS algorithm and the AP algorithm, respectively. Two sets of symbols are used to be consistent with Chapter 6. Namely, symbols with "$\sim$" represent the signals without the multiplications in (6.19) and (6.24); while symbols without "$\sim$" represent the signals after the operations of (6.19) and (6.24).

A.1 NLMS in the DFT subband

Recap NLMS shown in Algorithm 1 in Chapter 2 as the more general expression (complex value) in the filter bank k:

$$e_k(m) = d_k(m) - \mathbf{w}_k^H(m)\mathbf{u}_k(m) \tag{A.1}$$

$$\mathbf{w}_k(m+1) = \mathbf{w}_k(m) + \frac{\mu}{\mathbf{u}_k^H(m)\mathbf{u}_k(m) + \delta} e_k^*(m)\mathbf{u}_k(m) \tag{A.2}$$

where the superscripts ^{H} and * denote the Hermitian transpose and complex conjugate, respectively. In order to distinguish the input signal of the adaptive filter from the signal in the time domain, $\mathbf{u}_k(m)$ is used to denote the vector of the input signal, defined by

$$\mathbf{u}_k(m) = [X_k(m), X_k(m-1) \cdots, X_k(m-L+1)]^T \tag{A.3}$$

where the superscript ^{T} indicates the transpose, and L denotes the length of the adaptive filter in subband.

Referring to (6.19), the vector of input signal $\mathbf{u}_k(m)$ can be written as

$$\begin{aligned} \mathbf{u}_k(m) &= \left[W_K^{-kmM}\tilde{X}_k(m), W_K^{-k(m-1)M}\tilde{X}_k(m-1), \cdots, W_K^{-k(m-L+1)M}\tilde{X}_k(m-L+1)\right]^T \\ &= W_K^{-kmM}\mathbf{\Lambda}_k\tilde{\mathbf{u}}_k(m) \end{aligned} \tag{A.4}$$

where

$$\begin{aligned} \tilde{\mathbf{u}}_k(m) &= [\tilde{x}_k(m), \tilde{x}_k(m-1) \cdots, \tilde{x}_k(m-L+1)]^T & \text{(A.5)} \\ \mathbf{\Lambda}_k &= \text{diag}\left[1, W_K^{kM}, \cdots, W_K^{k(L-1)M}\right] & \text{(A.6)} \end{aligned}$$

Similarly, the desired signal $d_k(m)$ can be expressed as

$$d_k(m) = W_k^{-kmM}\tilde{d}_k(m) \tag{A.7}$$

Referring to (6.24), we also have

$$\tilde{e}_k(m) = W_k^{kmM}e_k(m) \tag{A.8}$$

Then (A.1) becomes

$$W_K^{-kmM}\tilde{e}_k(m) = W_k^{-kmM}\tilde{d}_k(m) - \mathbf{w}^H(m)W_K^{-kmM}\mathbf{\Lambda}_k\tilde{\mathbf{u}}_k(m) \tag{A.9}$$

or

$$\tilde{e}_k(m) = \tilde{d}_k(m) - (\mathbf{\Lambda}_k^H\mathbf{w})^H\tilde{\mathbf{u}}_k(m) \tag{A.10}$$

It is easy to derive from (A.4) that

$$\begin{aligned}\mathbf{u}_k^H(m)\mathbf{u}_k(m) &= \left[W_K^{-kmM}\mathbf{\Lambda}_k\tilde{\mathbf{u}}_k(m)\right]^H\left[W_K^{-kmM}\mathbf{\Lambda}_k\tilde{\mathbf{u}}_k(m)\right] \\ &= \cancel{W_K^{kmM}}\tilde{\mathbf{u}}_k^H(m)\cancel{\mathbf{\Lambda}_k^H}\cancel{\mathbf{\Lambda}_k}\cancel{W_k^{-kmM}}\tilde{\mathbf{u}}_k(m) \\ &= \tilde{\mathbf{u}}_k^H(m)\tilde{\mathbf{u}}_k(m) \qquad (A.11)\end{aligned}$$

where we have used the fact that $\mathbf{\Lambda}_k^H = \mathbf{\Lambda}_k^{-1}$, then (A.2) becomes

$$\mathbf{w}_k(m+1) = \mathbf{w}_k(m) + \frac{\mu}{\tilde{\mathbf{u}}_k^H(m)\tilde{\mathbf{u}}_k(m) + \delta}\tilde{e}_k^*(m)\mathbf{\Lambda}_k\tilde{\mathbf{u}}_k(m) \qquad (A.12)$$

or

$$\mathbf{\Lambda}_k^H\mathbf{w}_k(m+1) = \mathbf{\Lambda}_k^H\mathbf{w}_k(m) + \frac{\mu}{\tilde{\mathbf{u}}_k^H(m)\tilde{\mathbf{u}}_k(m) + \delta}\tilde{e}_k^*(m)\tilde{\mathbf{u}}_k(m) \qquad (A.13)$$

Replacing $\mathbf{\Lambda}_k^H\mathbf{w}_k(m)$ by $\tilde{\mathbf{w}}_k(m)$, (A.10) and (A.13) are written as

$$\begin{aligned}\tilde{e}_k(m) &= \tilde{d}_k(m) - \tilde{\mathbf{w}}_k^H(m)\tilde{\mathbf{u}}_k(m) \qquad (A.14) \\ \tilde{\mathbf{w}}_k(m+1) &= \tilde{\mathbf{w}}_k(m) + \frac{\mu}{\tilde{\mathbf{u}}_k^H(m)\tilde{\mathbf{u}}_k(m) + \delta}\tilde{e}_k^*(m)\tilde{\mathbf{u}}_k(m) \qquad (A.15)\end{aligned}$$

Obviously, (A.14) and (A.15) are equivalent to (A.1) and (A.2), respectively. In other words, if $\mathbf{u}_k(m)$ is replaced by $\tilde{\mathbf{u}}_k(m)$ and $d_k(m)$ is replaced by $\tilde{d}_k(m)$, the output $e_k(m)$ should be substituted by $\tilde{e}_k(m)$. Therefore, (6.19) and (6.24) can be discarded when the NLMS algorithm is implemented in the uniform DFT subband.

A.2 AP in the DFT subband

The original AP algorithm needs to be modified when it is used in subband because of the complex signals. Referring to Algorithm 3 in Chapter 2, the complex version of AP is expressed as [148]:

$$\begin{aligned}\mathbf{e}_k(m) &= \mathbf{d}_k(m) - \mathbf{U}_k(m)\mathbf{w}_k^*(m) \qquad (A.16) \\ \mathbf{w}_k(m+1) &= \mathbf{w}_k(m) + \mu\mathbf{U}_k^+(m)\mathbf{e}_k^*(m) \qquad (A.17)\end{aligned}$$

where $\mathbf{U}_k(m)$, as defined in Chapter 2, is the excitation signal matrix consisting of input signal vectors, and $\mathbf{U}_k^{+}(m)$ is the conjugate of the pseudo-inverse of $\mathbf{U}_k(m)$ [39], defined as

$$\mathbf{U}_k^{+}(m) = \left\{\mathbf{U}_k^H(m)\left[\mathbf{U}_k(m)\mathbf{U}_k^H(m)\right]^{-1}\right\}^* \tag{A.18}$$

Using (A.4), we have

$$\begin{aligned}\mathbf{U}_k(m) &= \begin{bmatrix}\mathbf{u}_k^T(m)\\ \mathbf{u}_k^T(m-1)\\ \vdots\\ \mathbf{u}_k^T(m-p+1)\end{bmatrix} = \begin{bmatrix}W_K^{-kmM}\tilde{\mathbf{u}}_k^T(m)\mathbf{\Lambda}_k\\ W_K^{-k(m-1)M}\tilde{\mathbf{u}}_k^T(m-1)\mathbf{\Lambda}_k\\ \vdots\\ W_K^{-k(m-p+1)M}\tilde{\mathbf{u}}_k^T(m-p+1)\mathbf{\Lambda}_k\end{bmatrix}\\ &= W_K^{-kmM}\mathbf{\Psi}_k\tilde{\mathbf{U}}_k(m)\mathbf{\Lambda}_k\end{aligned} \tag{A.19}$$

where p is the projection order; $\tilde{\mathbf{U}}_k(m)$ and $\mathbf{\Psi}_k$ are respectively defined as

$$\tilde{\mathbf{U}}_k(m) = \left[\tilde{\mathbf{u}}_k(m), \tilde{\mathbf{u}}_k(m-1), \cdots, \tilde{\mathbf{u}}_k(m-p+1)\right]^T \tag{A.20}$$

$$\mathbf{\Psi}_k = \mathrm{diag}\left[1, W_K^{kM}, \cdots, W_K^{k(p-1)M}\right] \tag{A.21}$$

By the similar procedure, the vectors $\mathbf{d}_k(m)$ and $\tilde{\mathbf{e}}_k(m)$ are obtained from (6.19) and (6.24), respectively

$$\mathbf{d}_k(m) = W_K^{-kmM}\mathbf{\Psi}_k\tilde{\mathbf{d}}_k(m) \tag{A.22}$$

$$\tilde{\mathbf{e}}_k(m) = W_K^{kmM}\mathbf{\Psi}_k^*\mathbf{e}_k(m) \tag{A.23}$$

where, $\tilde{\mathbf{d}}_k(m)$ and $\tilde{\mathbf{e}}_k(m)$ are the vectors of the desired signal and the error signal, respectively defined as

$$\tilde{\mathbf{d}}_k(m) = \left[\tilde{d}_k(m), \tilde{d}_k(m-1)\cdots, \tilde{d}_k(m-L+1)\right]^T \tag{A.24}$$

$$\tilde{\mathbf{e}}_k(m) = \left[\tilde{e}_k(m), \tilde{e}_k(m-1)\cdots, \tilde{e}_k(m-L+1)\right]^T \tag{A.25}$$

Note that $\mathbf{\Psi}_k^H = \mathbf{\Psi}_k^* = \mathbf{\Psi}_k^{-1}$, and replace $\mathbf{U}_k(m)$, $\mathbf{d}_k(m)$ and $\mathbf{e}_k(m)$ by (A.20), (A.22) and (A.23), then (A.16) can be written as

$$\tilde{\mathbf{e}}_k(m) = \tilde{\mathbf{d}}_k(m) - \tilde{\mathbf{U}}_k(m)\mathbf{\Lambda}_k\mathbf{w}_k^*(m) \tag{A.26}$$

From (A.19), one can derive that

$$\begin{aligned}
\mathbf{U}_k^{+}(m) &= \left\{\mathbf{U}_k^H(m)\left[\mathbf{U}_k(m)\mathbf{U}_k^H(m)+\delta\mathbf{I}\right]^{-1}\right\}^* \\
&= \left\{\left[W_K^{kmM}\boldsymbol{\Lambda}_k^H\tilde{\mathbf{U}}_k^H(m)\boldsymbol{\Psi}_k^H\right]\left[\cancel{W_K^{-kmM}}\boldsymbol{\Psi}_k\tilde{\mathbf{U}}_k(m)\cancel{\boldsymbol{\Lambda}_k}\cancel{W_K^{kmM}}\cancel{\boldsymbol{\Lambda}_k^H}\tilde{\mathbf{U}}_k^H(m)\boldsymbol{\Psi}_k^H+\delta\mathbf{I}\right]^{-1}\right\}^* \\
&= \left\{W_K^{kmM}\boldsymbol{\Lambda}_k^H\tilde{\mathbf{U}}_k^H(m)\cancel{\boldsymbol{\Psi}_k^H}\cancel{\boldsymbol{\Psi}_k}\left[\tilde{\mathbf{U}}_k(m)\tilde{\mathbf{U}}_k^H(m)+\delta\mathbf{I}\right]^{-1}\boldsymbol{\Psi}_k^{-1}\right\}^* \\
&= W_K^{-kmM}\boldsymbol{\Lambda}_k\tilde{\mathbf{U}}_k^{+}(m)\boldsymbol{\Psi}_k \qquad \text{(A.27)}
\end{aligned}$$

where we have used the facts that $\boldsymbol{\Lambda}_k^H = \boldsymbol{\Lambda}_k^{-1}$ and $\boldsymbol{\Psi}_k^H = \boldsymbol{\Psi}_k^{-1}$, and a small positive δ is added to the diagonal elements in case of the inverse of an ill-conditioned matrix.

Hence, (A.17) becomes

$$\boldsymbol{\Lambda}_k^H\mathbf{w}_k(m+1) = \boldsymbol{\Lambda}_k^H\mathbf{w}_k(m) + \mu\tilde{\mathbf{U}}_k^{+}(m)\tilde{\mathbf{e}}_k^*(m) \qquad \text{(A.28)}$$

Similar to the procedure in A.1, substitute $\tilde{\mathbf{w}}_k(m)$ for $\boldsymbol{\Lambda}_k^H\mathbf{w}_k(m)$, (A.26) and (A.28) can be written as

$$\begin{aligned}
\tilde{\mathbf{e}}_k(m) &= \tilde{\mathbf{d}}_k(m) - \tilde{\mathbf{U}}_k(m)\tilde{\mathbf{w}}_k^*(m) \qquad \text{(A.29)} \\
\tilde{\mathbf{w}}_k(m+1) &= \tilde{\mathbf{w}}_k(m) + \mu\tilde{\mathbf{U}}_k^{+}(m)\tilde{\mathbf{e}}_k^*(m) \qquad \text{(A.30)}
\end{aligned}$$

Comparing (A.29) and (A.30) with (A.16) and (A.17), one can find that when $\mathbf{u}_k(m)$ and $\tilde{\mathbf{u}}_k(m)$ are respectively replaced by $d_k(m)$ and $\tilde{d}_k(m)$, the output $e_k(m)$ should be substituted by $\tilde{e}_k(m)$. Therefore, (6.19) and (6.24) can be discarded when the AP algorithm is implemented in the uniform DFT subband.

References

[1] G. Shiers, ed., *The Telephone : An Historical Anthology.* Historical studies in telecommunications, New York: Arno Press, 1977.

[2] ITU-T Recommendation G.310, *Transmission Characteristics for Telephone Band (300-3400 Hz) Digital Telephones.* International Telecommunication Union, May 2000.

[3] A. M. Noll, *Introduction to Telephones and Telephone Systems.* Boston: Artech House, 3rd ed., 1998.

[4] ITU-T Recommendation G.131, *Control of Talker Echo.* International Telecommunication Union, Aug. 1996.

[5] Members of the technical staff, *Transmission Systems for Communications.* Holmdel: Bell Telephone Laboratories, 5th ed., 1982.

[6] J. Kang, *Acoustics of Long Spaces: Theory and Design Guidance.* London: Thomas Telford, 2002.

[7] ITU-T Recommendation G.164, *Echo Suppressors.* International Telecommunication Union, Nov. 1988.

[8] E. Hänsler, "The hands-free telephone problem - an annotated bibliography," *Signal Processing*, vol. 27, pp. 259–271, June 1992.

[9] M. Tohyama and T. Koike, *Fundamentals of Acoustic Signal Processing.* San Diego: Academic Press, 1998.

[10] K. Murano, S. Unagami, and F. Amano, "Echo cancellation and applications," *IEEE Communications Magazine*, vol. 28, pp. 49–55, Jan. 1990.

[11] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control: An application of very-high-order adaptive filters," *IEEE Signal Processing Magazine*, vol. 16, pp. 42–69, July 1999.

[12] S. L. Gay and J. Benesty, eds., *Acoustic Signal Processing for Telecommunication.* Boston: Kluwer Academic Publishers, 2000.

[13] P. M. Clarkson, *Optimal and Adaptive Signal Processing.* Boca Raton: CRC Press, 1993.

[14] A. Liavas and P. Regalia, "Acoustic echo cancellation: Do IIR models offer better modeling capabilities than their FIR counterparts," *IEEE Trans. Signal Processing*, vol. 46, pp. 2499–2504, Sept. 1998.

[15] G. Glentis, K. Berberidis, and S. Theodoridis, "A unified view: Efficient least squares adaptive algorithms for FIR transversal filtering," *IEEE Signal Processing Magazine*, vol. 16, pp. 13–41, July 1999.

[16] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing.* Englewood Cliffs: Prentice Hall, 1983.

[17] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1862–1875, Aug. 1992.

[18] Q.-G. Liu, B. Champagne, and K. C. Ho, "Simple design of oversampled uniform DFT filter banks with applications to subband acoustic echo cancellation," *Signal Processing*, vol. 80, pp. 831–847, June 2000.

[19] T. Aboulnasr and K. Mayyas, "Complexity reduction of the nlms algorithm via selective coefficient update," *IEEE Trans. Signal Processing*, vol. 47, pp. 1421–1424, May 1999.

[20] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Proc. Asilomar Conf. Signals, Systems, Computers*, vol. 1, (Pacific Grove, USA), pp. 394–398, Nov. 1998.

[21] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol. 67-A, no. 5, pp. 19–27, 1984.

[22] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'95)*, (Detroit, USA), pp. 3023–3026, May 1995.

[23] B. E. Keiser and E. Strange, *Digital Telephony and Network Integration.* New York: Van Nostrand Reinhold, 2nd ed., 1995.

[24] ITU-T Recommendation G.729, *Coding of Speech at 8 kbits/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP).* International Telecommunication Union, Mar. 1996.

[25] GSM, Draft EN 300 724 V6.0.0, *Digital Cellular Telecommunications System; ANSI-C Code for the GSM Enhanced Full Rate (EFR) Speech Codec (GSM 06.53 Version 6.0.0 Release 1997).* European Telecommunications Standards Institute, Jan. 1999.

[26] X. Lu and B. Champagne, "Pitch analysis-based acoustic echo cancellation over a non-linear channel," in *Proc. XI European Signal Processing Conference (EUSIPCO'02)*, vol. 1, (Toulouse, France), pp. 159–162, Sept. 2002.

[27] X. Lu and B. Champagne, "A variable step-size adaptive cross-spectral algorithm for acoustic echo cancellation," *IEICE Trans. Fundamentals*, vol. E86-A, pp. 2812–2821, Nov. 2003.

[28] X. Lu and B. Champagne, "Acoustic echo cancellation over a non-linear channel," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'01)*, (Darmstadt, Germany), pp. 139–142, Sept. 2001.

[29] X. Lu and B. Champagne, "Subspace approach for the suppression of the nonlinear acoustic echo introduced by loudspeakers," in *Proc. 21st Biennial Symposium on communications*, (Kingston, Canada), pp. 512–515, June 2002.

[30] X. Lu and B. Champagne, "A centralized acoustic echo canceller exploiting masking properties of the human ear," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'03)*, vol. 5, (Hong Kong, China), pp. 377–380, Apr. 2003.

[31] X. Lu and B. Champagne, "Acoustic echo cancellation with post-filtering in subband," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'03)*, (New Paltz, USA), pp. 29–32, Oct. 2003.

[32] L. Ljung, *System Identification: Theory for the User.* Upper Saddle River: Prentice Hall PTR, 2nd ed., 1999.

[33] S. Haykin, *Adaptive Filter Theory.* Upper Saddle River: Prentice-Hall, 4th ed., 2002.

[34] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, vol. 8, pp. 10–26, July 1991.

[35] W. Kellermann, "Echoes and noise with seamless acoustic man-machine interfaces - the challenge persists," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'99)*, (Pocono Manor, USA), pp. 1–7, Sept. 1999.

[36] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, pp. 943–950, Apr. 1979.

[37] Y. Haneda, S. Makino, and Y. Kaneda, "Common acoustical pole and zero modeling of room transfer functions," *IEEE Trans. Speech and Audio Processing*, vol. 2, pp. 320–328, Apr. 1994.

[38] J. Mourjopoulos and M. A. Paraskevas, "Pole and zero modeling of room transfer functions," *J. Sound and Vibration*, vol. 146, pp. 281–302, Apr. 1991.

[39] G. H. Golub and C. F. V. Loan, *Matrix Computations.* Baltimore: Johns Hopkins University Press, 3rd ed., 1996.

[40] N. E. Hubing and S. T. Alexander, "Statistical analysis of initialization methods for RLS adaptive filters," *IEEE Trans. Signal Processing*, vol. 39, pp. 1793–1804, Aug. 1991.

[41] A. Benallal and A. Gilloire, "A new method to stabilize fast RLS algorithms based on a first-order of the propagation of numerical errors," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'88)*, vol. 3, (New York, USA), pp. 1373–1376, Apr. 1988.

[42] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, pp. 1394–1402, Dec. 1983.

[43] J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 304–337, Apr. 1984.

[44] D. T. M. Slock and T. Kailath, "Numerically stable fast recursive least-squares transversal filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'88)*, vol. 3, (New York, USA), pp. 1365–1368, Apr. 1988.

[45] S. T. Alexander, *Adaptive Signal Processing Theory and Applications.* New York: Springer-Verlag, 1986.

[46] H. Yasukawa and S. Shimada, "An acoustic echo canceller using subband sampling and decorrelation methods," *IEEE Trans. Signal Processing*, vol. 41, pp. 926–930, Feb. 1993.

[47] M. Tanaka, S. Makino, and J. Kojima, "A block exact fast affine projection algorithm," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 79–86, Jan. 1999.

[48] B. Champagne and Q. G. Liu, "Adaptive filters for acoustic echo cancellation," Tech. Rep. 96-15, INRS-Telecommunications, June 1996.

[49] M. Rupp, "A family of adaptive filter algorithms with decorrelating properties," *IEEE Trans. Signal Processing*, vol. 46, pp. 771–775, Mar. 1998.

[50] D. O'Shaughnessy, *Speech Communications: Human and Machine.* New York: IEEE Press, 2nd ed., 2000.

[51] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs: Prentice-Hall, 1978.

[52] J. P. Campbell and T. E. Tremain, "Voiced/unvoiced classification of speech with applications to the U.S. government LPC-10E algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'86)*, vol. 11, (Tokyo, Japan), pp. 473–476, Apr. 1986.

[53] S. Yamamoto, S. Kitayama, J. Tamura, and H. Ishigami, "An adaptive echo canceller with linear predictor," *Trans. IEICE of Japan*, vol. E 62, pp. 851–857, Dec. 1979.

[54] R. Frenzel and M. Hennecke, "Using prewhitening and stepsize control to improve the performance of the LMS algorithm for acoustic echo compensation," in *Proc. IEEE Int. Sym. Circuits and Systems (ISCAS'92)*, vol. 4, (San Diego, USA), pp. 1930–1932, May 1992.

[55] B. S. Nollett and D. L. Jones, "Nonlinear echo cancellation for hands-free speakerphones," in *Proc. IEEE Workshop on Nonlinear Signal and Image Processing (NSIP'97)*, (Michigan, USA), Sept. 1997.

[56] A. Stenger and W. Kellermann, "Nonlinear acoustic echo cancellation with fast converging memoryless preprocessor," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'00)*, vol. 2, (Istanbul, Turkey), pp. 805–808, June 2000.

[57] G. L. Sicuranza, A. Bucconi, and P. Mitri, "Adaptive echo cancellation with nonlinear digital filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'84)*, vol. 9, (San Diego, USA), pp. 130–133, Mar. 1984.

[58] J. Chen and J. Vandewalle, "Study of adaptive nonlinear echo canceller with Volterra expansion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'89)*, vol. 2, (Glasgow, UK), pp. 1376–1379, May 1989.

[59] L. Ngia and J. Sjöberg, "Using cascaded neural net filter in non-linear identification of acoustic echo path," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'99)*, (Pocono Manor, USA), pp. 172–175, Sept. 1999.

[60] A. Birkett and R. Goubran, "Nonlinear loudspeaker compensation for hands free acoustic echo cancellation," *Electronics Letters*, vol. 32, pp. 1063–1064, June 1996.

[61] F. X. Y. Gao and W. M. Snelgrove, "Adaptive linearization of a loudspeaker," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'91)*, vol. 5, (Toronto, Canada), pp. 3589–3592, Apr. 1991.

[62] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications.* Boston: Kluwer Academic Publishers, 1990.

[63] A. Fermo, A. Carini, and G. L. Sicuranza, "Simplified Volterra filters for acoustic echo cancellation in GSM receivers," in *Proc. X European Signal Processing Conference (EUSIPCO'00)*, vol. 4, (Tampere, Finland), pp. 2413–2416, Sept. 2000.

[64] M. V. Dokic and P. M. Clarkson, "On the performance of a second-order adaptive Volterra filter," *IEEE Trans. Signal Processing*, vol. 41, pp. 1944–1947, May 1993.

[65] A. J. Chaturvedi and G. Sharma, "A new family of concurrent algorithms for adaptive volterra and linear filters," *IEEE Trans. Signal Processing*, vol. 47, pp. 2547–2551, Sept. 1999.

[66] J. Lee and V. Mathews, "A fast recursive least squares adaptive second-order Volterra filter and its performance analysis," *IEEE Trans. Signal Processing*, vol. 41, pp. 1087–1102, Mar. 1993.

[67] S. Kalluri and G. Arce, "A general class of nonlinear normalized adaptive filtering algorithms," *IEEE Trans. Signal Processing*, vol. 47, pp. 2262–2272, Aug. 1999.

[68] S. Kinoshita, Y. Kajikawa, and Y. Nomura, "Volterra filters using multirate signal processing and their application to loudspeaker systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'01)*, vol. 6, (Salt Lake City, USA), pp. 3497–3500, May 2001.

[69] P. Chang, C. Lin, and B. Yeh, "Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks," *J. Acoust. Soc. Am.*, vol. 95, pp. 3400–3408, June 1994.

[70] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'99)*, vol. 2, (Phoenix, USA), pp. 877–880, Mar. 1999.

[71] J. Hwang, S. Kung, M. Niranjan, and J. Principe, "The past, present, and future of neural networks for signal processing," *IEEE Signal Processing Magazine*, vol. 14, pp. 28–48, Nov. 1997.

[72] S. Haykin, *Neural Networks: A Comprehensive Foundation.* Upper Saddle River: Prentice-Hall, 2nd ed., 1999.

[73] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series." in *The Handbook of Brain Theory and Neural Networks*, M.A. Arbib Ed., MIT Press, 1995.

[74] L. Yin, J. Astola, and Y. Neuvo, "A new class of nonlinear filters - neural filters," *IEEE Trans. Signal Processing*, vol. 41, pp. 1201–1222, Mar. 1993.

[75] M. Bouchard, "New recursive-least-squares algorithms for nonlinear active control of sound and vibration using neural networks," *IEEE Trans. Neural Networks*, vol. 12, pp. 135–147, Jan. 2001.

[76] ITU-T Recommendation G.711, *Pulse Code Modulation (PCM) of Voice Frequencies.* International Telecommunication Union, Nov. 1988.

[77] ITU-T Recommendation G.726, *40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM).* International Telecommunication Union, Dec. 1990.

[78] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video.* Englewood Cliffs: Prentice-Hall, 1984.

[79] CCITT, COM XII-67-E, *Measurement of the Transfer Functions of Hands-Free Telephones, Comparison Between Results Measured with Artificial Voice and a Composite Source Signal.* International Telegraph and Telephone Consultative Committee, Aug. 1990.

[80] L. Besacier, S. Grassi, A. Dufaux, M. Ansorge, and F. Pellandini, "GSM speech coding and speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'00)*, vol. 2, (Istanbul, Turkey), pp. 1085–1088, July 2000.

[81] F.-K. Chen and J.-F. Yang, "Maximum-take-precedence ACELP: a low complexity search method," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'01)*, vol. 2, (Salt Lake City, USA), pp. 693–696, May 2001.

[82] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control.* New York: Springer, 1999.

[83] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang, "Nonlinear black-box models in system identification: Mathematical foundations," *Automatica*, vol. 31, pp. 1725–1750, Dec. 1995.

[84] C. L. Phillips and R. D. Harbor, *Feedback Control Systems.* Upper Saddle River: Prentice Hall, 4th ed., 2000.

[85] R. H. Small, "Direct-radiator loudspeaker system analysis," *J. Audio Eng. Soc.*, vol. 20, pp. 383–395, June 1972.

[86] W. Klippel, "Nonlinear large-signal behavior of electrodynamic loudspeaker at low frequencies," *J. Audio Eng. Soc.*, vol. 40, pp. 483–496, June 1992.

[87] A. J. M. Kaizer, "Modeling of the nonlinear response of an electrodynamic loudspeaker by a Volterra series expansion," *J. Audio Eng. Soc.*, vol. 35, pp. 421–433, June 1987.

[88] A. Bellini, G. Cibelli, E. Ugolotti, A. Farina, and C. Morandi, "Non-linear digital audio processor for dedicated loudspeaker systems," *IEEE Trans. Consumer Electronics*, vol. 44, pp. 1024–1031, Aug. 1998.

[89] W. Frank, R. Reger, and U. Appel, "Realtime loudspeaker linearization," in *IEEE Winter Workshop on Nonlinear Digital Signal Processing*, (Tampere, Finland), pp. 2.1_3.1–2.1_3.5, Jan. 1993.

[90] W. Greblicki, "Nonparametric identification of Wiener systems," *IEEE Trans. Information Theory*, vol. 38, pp. 1487–1493, Sept. 1992.

[91] W. Greblicki and M. Pawlak, "Nonparametric identification of Hammerstein systems," *IEEE Trans. Information Theory*, vol. 35, pp. 409–418, Mar. 1989.

[92] S. Kim, H. Kwon, K. Bae, K. Byun, and K. Kim, "Performance improvement of double-talk detection algorithm in the acoustic echo canceller," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'01)*, vol. 5, (Salt Lake City, USA), pp. 3249–3252, May 2001.

[93] J. Cho, D. Morgan, and J. Benesty, "An objective technique for evaluating doubletalk detectors in acoustic echo cancelers," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 718–724, Nov. 1999.

[94] C. Breining, "A robust fuzzy logic-based step-gain control for adaptive filters in acoustic echo cancellation," *IEEE Trans. Speech and Audio Processing*, vol. 9, pp. 162–167, Feb. 2001.

[95] S. J. Pushparajah and J. A. Chambers, "A smarter method for acoustic echo cancellation in the presence of double talk," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'99)*, (Pocono Manor, USA), pp. 184–186, Sept. 1999.

[96] P. Heitkämper, "An adaptation control for acoustic echo cancellers," *IEEE Signal Processing Letters*, vol. 4, pp. 170–172, June 1997.

[97] T. Okuno, M. Fukushima, and M. Tohyama, "Adaptive cross-spectral technique for acoustic echo cancellation," *IEICE Trans. Fundamentals*, vol. E82-A, pp. 634–639, Apr. 1999.

[98] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-time Processing of Speech Signals.* New York: IEEE Press, 2000.

[99] M. Asharif, T. Hayashi, and K. Yamashita, "Correlation lms algorithm and its application to double-talk echo cancelling," *Electronics Letters*, vol. 35, pp. 194–195, Feb. 1999.

[100] M. Fukushima, T. Okuno, H. Yanagawa, and K. Kido, "Improvement of the accuracy in attenuation constant estimation using the cross-spectral technique," *IEICE Trans. Fundamentals*, vol. E82-A, pp. 626–633, Apr. 1999.

[101] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time Signal Processing.* Upper Saddle River: Prentice Hall, 2nd ed., 1999.

[102] C. W. K. Gritton and D. W. Lin, "Echo cancellation algorithms," *IEEE ASSP Magazine*, vol. 1, pp. 30–38, Apr. 1984.

[103] X. Lu and B. Champagne, "Acoustic echo cancellation in the presence of codec nonlinearities," technical report A0-19, INRS-Telecommunications, Oct. 2000.

[104] V. Turbin, A. Gilloire, and P. Scalart, "Comparison of three post-filtering algorithms for residual acoustic echo reduction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'97)*, (Munich, Germany), pp. 307–310, Apr. 1997.

[105] P. Dreiseitel and H. Puder, "A combination of noise reduction and improved echo cancelation," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC'97)*, (London, UK), pp. 180–183, Sept. 1997.

[106] S. Gustafsson, P. Jax, A. Kamphausen, and P. Vary, "A postfilter for echo and noise reduction avoiding the problem of musical tones," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'99)*, vol. 2, (Phoenix, USA), pp. 873–876, Mar. 1999.

[107] C. Beaugeant and P. Scalart, "Combined system for noise reduction and echo cancellation," in *Proc. IX European Signal Processing Conference (EUSIPCO'98)*, vol. 2, (Island of Rhodes, Greece), pp. 957–960, Sept. 1998.

[108] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 27, pp. 113–120, Apr. 1979.

[109] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 3rd ed., 1991.

[110] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming.* New York: McGraw-Hill, 1996.

[111] R. Martin and J. Altenhöner, "Coupled adaptive filters for acoustic echo control and noise reduction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'95)*, vol. 5, (Detroit, USA), pp. 3043–3046, May 1995.

[112] L. Arslan, A. McCree, and V. Viswanathan, "New methods for adaptive noise suppression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'95)*, vol. 1, (Detroit, USA), pp. 812–815, May 1995.

[113] C. Avendano and G. Garcia, "STFT-based multi-channel acoustic interference suppressor," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'01)*, vol. 1, (Salt Lake City, USA), pp. 625–628, May 2001.

[114] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Trans. Speech and Audio Processing*, vol. 3, pp. 251–266, July 1995.

[115] G. Doblinger, "Performance analysis of an adaptive subspace filter for signal enhancement," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'95)*, vol. 2, (Seattle, USA), pp. 1086–1089, May 1995.

[116] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol. 43, pp. 95–107, Jan. 1995.

[117] B. Champagne, "On the asymptotic convergence and numerical stability of the Proteus EVD trackers," *IEEE Trans. Signal Processing*, vol. 48, pp. 242–246, Jan. 2000.

[118] R. P. Ramachandran and R. J. Mammone, eds., *Modern Methods of Speech Processing.* Boston: Kluwer Academic Publishers, 1995.

[119] R. P. Ramachandran and P. Kabal, "Pitch prediction filter in speech coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 467–478, Apr. 1989.

[120] Y. Huang and R. A. Goubran, "Effects of vocoder distortion on network echo cancellation," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME'00)*, vol. 1, (New York City, USA), pp. 437–439, Aug. 2000.

[121] J. Benesty and P. Duhamel, "A fast exact least mean square adaptive algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 2904–2920, Dec. 1992.

[122] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, pp. 14–37, Jan. 1992.

[123] D. Linebarger, B. Raghothaman, D. Begusic, E. Dowling, R. DeGroat, and S. Oh, "Low rank transform domain adaptive filtering," in *Proc. Asilomar Conf. Signals, Systems, Computers*, vol. 1, (Pacific Grove, USA), pp. 123–127, Nov. 1997.

[124] O. Tanrikulu, B. Baykal, A. G. Constantinides, and J. A. Chambers, "Residual echo signal in critically sampled subband acoustic echo cancellers based on IIR and FIR filter banks," *IEEE Trans. Signal Processing*, vol. 45, pp. 901–912, Apr. 1997.

[125] M. R. Petraglia, R. T. B. Vasconcellos, and R. G. Alves, "Performance comparison of adaptive subband structures applied to acoustic echo cancelling," in *Proc. Asilomar Conf. Signals, Systems, Computers*, vol. 2, (Pacific Grove, USA), pp. 1535–1539, Nov. 2001.

[126] P. P. Vaidyanathan, "Multirate digital filters, filter banks,polyphase networks, and applications: A tutorial," *Proc. of the IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[127] Z. Cvetkovic and M. Vetterli, "Oversampled filter banks," *IEEE Trans. Signal Processing*, vol. 46, pp. 1245–1255, May 1998.

[128] N. Grbic, J. M. de Haan, I. Claesson, and S. Nordholm, "Design of oversampled uniform DFT filter banks with reduced inband aliasing and delay constraints," in *Proc. Int. Symposium Signal Processing and its Appications*, vol. 1, (Kuala Lumpur, Malaysia), pp. 104–107, Aug. 2001.

[129] M. R. Portnoff, "Time-frequency representation of digial signals and systems based on short-time fourier analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 55–69, Feb. 1980.

[130] C. D. Creusere and S. K. Mitra, "A simple method for designing high-quality prototype filters for M-band pseudo QMF banks," *IEEE Trans. Signal Processing*, vol. 43, pp. 1005–1007, Apr. 1995.

[131] T. Q. Nguyen, "Digital filter bank design quadratic-constrained formulation," *IEEE Trans. Signal Processing*, vol. 43, pp. 2103–2108, Sept. 1995.

[132] J. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP'01)*, vol. 5, (Denver, USA), pp. 291–294, Apr. 1980.

[133] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 126–137, Mar. 1999.

[134] E. Zwicker and H. Fastl, *Psychoacoustics–Facts and Models.* New York: Springer, 2nd ed., 1999.

[135] G. J. Borden and K. S. Harris, *Speech Science Primer: Physiology, Acoustics, and Perception of Speech.* Baltimore: Williams & Wilkins, 2nd ed., 1984.

[136] C. Giguère and P. C. Woodland, "A computational model of the auditory periphery for speech and hearing research," *J. Acoust. Soc. Am.*, vol. 95, pp. 331–349, Jan. 1994.

[137] B. C. J. Moore, *An Introduction to the Psychology of Hearing.* London: Academic Press, 4th ed., 1997.

[138] G. von Békésy, *Experiments in Hearing.* New York: McGraw-Hill, 1960.

[139] ITU-R Recommendation BS.1387-1, *Method for Objective Measurements of Perceived Audio Quality.* International Telecommunication Union, Nov. 2001.

[140] H. Fletcher, "Auditory patterns," *Reviews of Modern Physics*, vol. 12, pp. 47–65, Jan. 1940.

[141] J. P. Egan and H. W. Hake, "On the masking pattern of a simple auditory stimulus," *J. Acoust. Soc. Am.*, vol. 22, pp. 622–630, 1950.

[142] R. Hellman, "Asymmetry of masking between noise and tone," *Perception and Psychophysics*, vol. 11, pp. 241–246, 1972.

[143] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Selected Areas in Communications*, vol. 6, pp. 314–323, Feb. 1988.

[144] M. R. Schroeder, B. S. Atal, and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. Acoust. Soc. Am.*, vol. 66, pp. 1647–1652, Dec. 1979.

[145] K. Brandenburg and G. Stoll, "ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio," *J. Acoust. Soc. Am.*, vol. 42, pp. 780–792, Oct. 1994.

[146] S. Gustafsson, R. Martin, P. Jex, and P. Vary, "A psychoacoustic approach to combined acoustic echo cancellation and noise reduction," *IEEE Trans. Speech and Audio Processing*, vol. 10, pp. 245–256, July 2002.

[147] ITU-T Recommendation P.862, *Perceptual Evaluation of Speech Quality (PESQ), An Objective Method for End-to-end Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs.* International Telecommunication Union, Feb. 2001.

[148] Q. G. Liu, B. Champagne, and K. C. Ho, "On the use of a modified fast affine projection algorithm in subbands for acoustic echo cancelation," in *Proc. IEEE Digital Signal Processing Workshop*, (Loen, Norway), pp. 354–357, Sept. 1996.