IMPLEMENTATION AND EVALUATION OF ECHO CANCELLATION ALGORITHMS

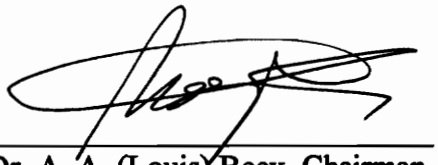
by

Sundar G. Sankaran

Thesis submitted to the Faculty of the
Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

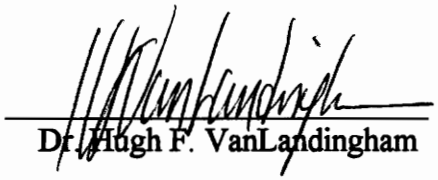
Master of Science
in
Electrical Engineering

APPROVED



Dr. A. A. (Louis) Beex, Chairman

Dr. Dushan Borojevic



Dr. Hugh F. VanLandingham

December 1996
Blacksburg, Virginia

Keywords: Hands-Free Telephony, Adaptive Filtering, Echo Canceler, Noise Canceler

IMPLEMENTATION AND EVALUATION OF ECHO CANCELLATION ALGORITHMS

by

Sundar G. Sankaran

Dr. A. A. (Louis) Beex, Chairman

Bradley Department of Electrical Engineering

(ABSTRACT)

Echo in telephones is generally undesirable but inevitable. There are two possible sources of echo in a telephone system. The impedance mismatch in hybrids generates network (electric) echo. The acoustic coupling between loudspeaker and microphone, in hands-free telephones, produces acoustic echo. Echo cancelers are used to control these echoes.

In this thesis, we analyze the Least Mean Squares (LMS), Normalized LMS (NLMS), Recursive Least Squares (RLS), and Subband NLMS (SNLMS) algorithms, and evaluate their performance as acoustic and network echo cancelers. The algorithms are compared based on their convergence rate, steady state echo return loss (ERL), and complexity of implementation. While LMS is simple, its convergence rate is dependent on the eigenvalue spread of the signal. In particular, it converges slowly with speech as input. This problem is mitigated in NLMS. The complexity of NLMS is comparable to that of LMS. The convergence rate of RLS is independent of the eigenvalue spread, and it has the fastest convergence. On the other hand, RLS is highly computation intensive. Among

the four algorithms considered here, SNLMS has the least complexity of implementation, as well as the slowest rate of convergence.

Switching between the NLMS and SNLMS algorithms is used to achieve fast convergence with low computational requirements. For a given computational power, it is shown that switching between algorithms can give better performance than using either of the two algorithms exclusively, especially in rooms with long reverberation times.

We also discuss various implementation issues associated with an integrated echo cancellation system, such as double-talk detection, finite precision effects, nonlinear processing, and howling detection and control. The use of a second adaptive filter is proposed, to reduce near-end ambient noise. Simulation results indicate that this approach can reduce the ambient noise by about 20 dB.

A configuration is presented for the real time single-chip DSP implementation of acoustic and network echo cancelers, and an interface between the echo canceler and the telephone is proposed. Finally, some results obtained from simulations and implementations of individual modules, on the TMS320C31 and ADSP 2181 processors, are reported. The real time NLMS DSP implementations provide 15 dB of echo return loss.

Acknowledgments

Words can not express my appreciation to Dr. A. A. (Louis) Beex for being an excellent advisor and an extraordinary professor. I consider myself to be very fortunate to pursue research under his guidance. His encouragement, support, and invaluable suggestions have made this work possible.

I am deeply indebted to Dr. Dushan Borojevic for advising me during my first year at Virginia Tech. I thank Dr. Hugh VanLandingham for his time and effort in reviewing this work.

I would also like to thank the members of the Center for Wireless Telecommunications, especially Mr. Willard Farley, for providing me an opportunity to work with them.

This work would not have been possible without the love and support provided by my parents and my sister Dr. Anitha A. Sankaran. I thank my uncle, Dr. Jayavel Sounderpandian, for being a constant source of inspiration for me.

I would like to thank my friends Desikan, Sriram, and Carlos for all the interesting discussions we had about almost everything under the sun.

I thank the Center for Innovative Technology, Herndon VA, and Comdial Corporation, Charlottesville VA, for their support in this research.

Finally, I thank God for giving me a nice family and advisor.

Table of Contents

List of Figures

1. Introduction

Echo is inevitable in telephones; echo is the delayed and possibly distorted version of the transmitted sound reflected back to the sender. There are two possible sources of echo in a telephone system. Figure 1.1 illustrates the generation of *network echo* (also referred to as electric echo). While the local channel from the phone to the central office is a *two-wire* bi-directional channel, the channel between the central offices is a *four-wire* channel comprising two unidirectional two-wire channels. The *hybrid* is the device that couples the two-wire and the four-wire channels. Ideally, the hybrid should transfer all the energy from the incoming branch of the four-wire channel to the two-wire channel. There should not be any coupling between the outgoing and the incoming branches of the four-wire channel. However, certain impedance mismatch problems in the hybrid can cause a part of the energy on the incoming branch to be transferred to the outgoing branch, thereby producing an echo.

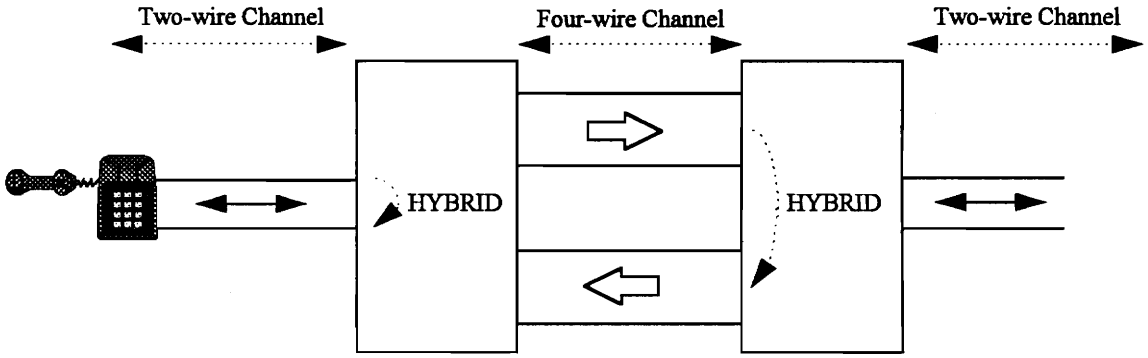


Fig. 1.1 Generation of Network Echo.

Figure 1.2 shows the generation of *acoustic echo* as it occurs in hands-free telephones. Unlike in handset-based telephones, in hands-free telephony the human communicator is separated from the telephone. This separation of the handset from the ear and the mouth brings the room itself into the communication system. There will be multiple reflections of the signal from the loudspeaker back to the microphone. There can also be a direct acoustic coupling between the loudspeaker and the microphone. These signals reach the other end as echoes.

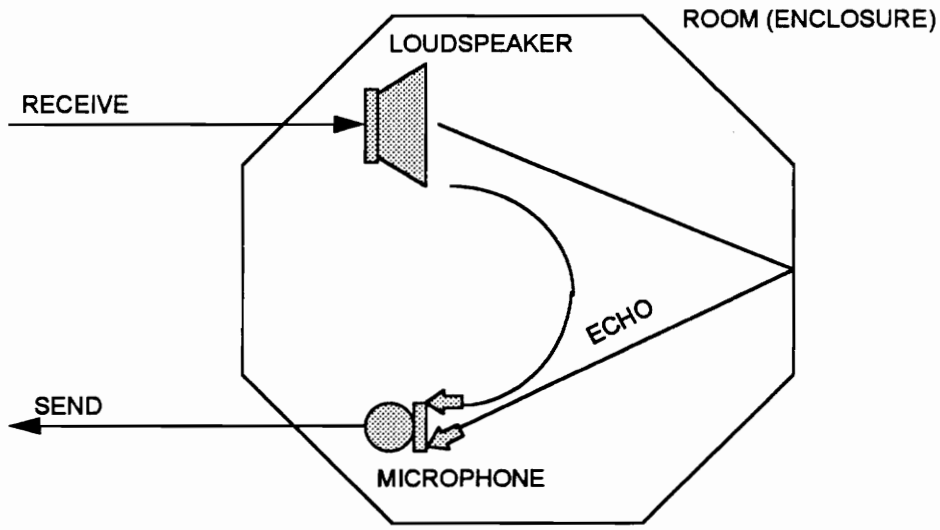


Fig. 1.2 Generation of Electric Echo.

Furthermore, the channel, including the acoustic and electric feedback paths, forms a closed loop. High volume level output at the loudspeaker can cause the loop to get into an oscillatory mode, usually referred to as howling. This limits the maximum volume of the speaker output.

The echo needs special treatment if the round-trip delay (echo with respect to original) exceeds 40 milliseconds [1]. When the calls are routed through satellites, the round-trip delay is 600 milliseconds, which exceeds the 40 msec threshold. Even some land-based long distance calls exceed that threshold.

Common current solutions control the echo using an echo suppressor (also called vari-losser or switched loss device). The echo suppressor, shown in Figure 1.3, dynamically alters the gains in the send and receive paths. The control unit (voice activity detector) continuously monitors the signals in the send and receive paths. The gain of the amplifier on the path with the higher energy signal is increased (to a high positive value), while the gain of the other amplifier is reduced (to a high negative value). Thus the echo suppressor, inherently, allows only one-way communication (half duplex), usually favoring the louder speaker [2].

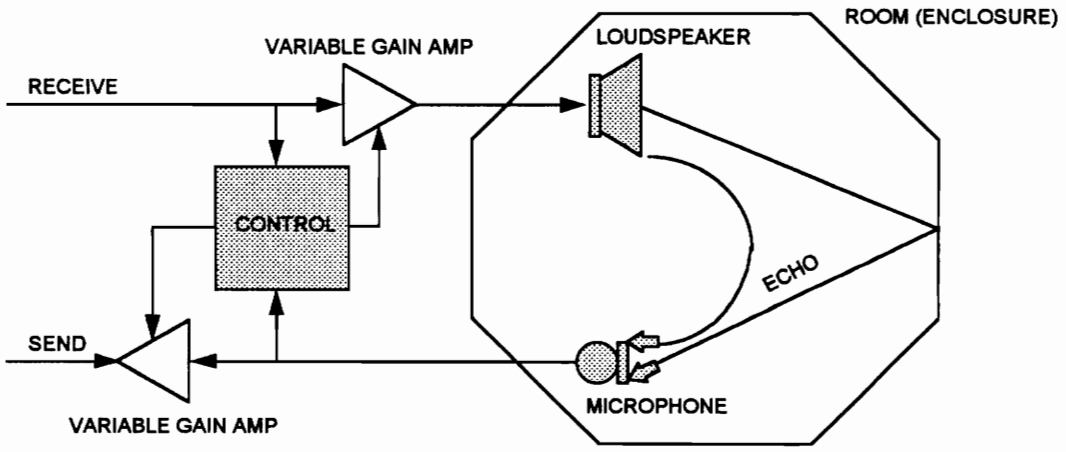


Fig. 1.3 Echo Suppressor.

While the echo suppressor is simple to implement, this solution clips speech and impairs interruptions. For example, if the near-end talker is initially listening to the far-end talker but suddenly wants to interject a point, it is quite likely that the switch preventing his speech from being transmitted will not close quickly enough, and the far-end talker may not receive all of the interrupting message [1]. At the same time, the near-end talker is not receiving all of the message from the far-end.

Attempts have been made to reduce the echo by using directional microphones [3]. This solution reduces the acoustic echo by placing the loudspeaker in the null of the microphone, thereby eliminating the direct path of the acoustic coupling between them. The performance of this system is highly sensitive to the positioning of the microphones, and the speaker.

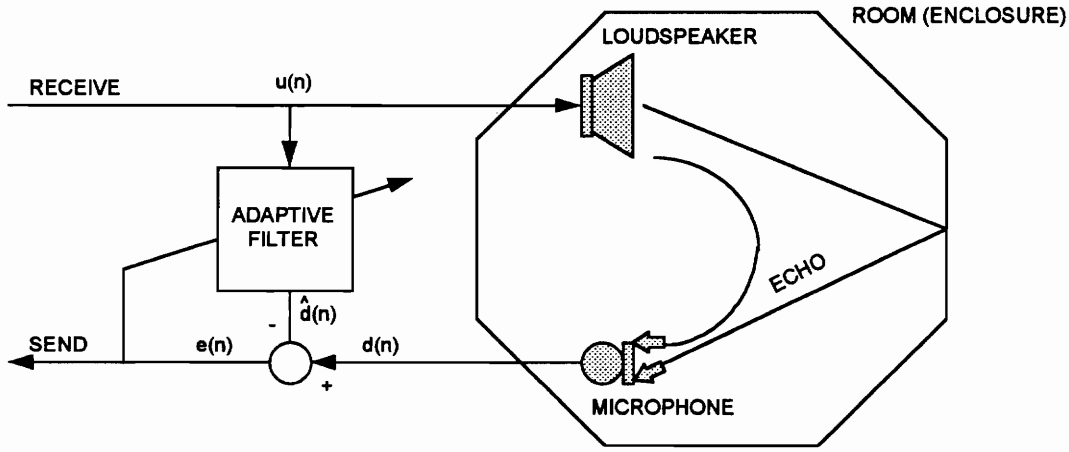


Fig. 1.4 Echo Canceler.

In this thesis, we study a full-duplex solution that uses echo cancelers to control the echo. The echo canceler first tries to estimate the impulse response of the echo path, and then generates a replica of the echo. The estimated echo is then subtracted from the received signal, as shown in Figure 1.4.

Adaptive filtering is required to obtain a good replica of the echo, since the echo path is usually unknown and time varying. The echo canceler must accurately estimate the impulse response of the echo path, and rapidly adapt to its variations. The recommendation ITU-T : G.167 specifies the performance requirements of acoustic echo control devices, while ITU-T : G.165 specifies the requirements of network echo control devices. While details are given in Appendices A and B, typical quantities used to specify performance are [5]:

- echo return loss in single talk and double talk modes,
- initial convergence time,
- recovery time after echo path variation.

The performance of the echo canceler depends on the choice of the adaptive filtering algorithm. In this thesis, we analyze the performance of various adaptive filtering algorithms and investigate the feasibility of using them for echo cancellation. We illustrate the performance of the adaptive filtering algorithms using simulation results and actual system measurements. For simulations, we modeled the echo path as an all-pass type system with poles at $0.995e^{\pm j\pi/6}$ and $0.996e^{\pm j2\pi/3}$, and zeros at $1.005e^{\pm j\pi/6}$ and $1.004e^{\pm j2\pi/3}$. The impulse response of this system is shown in Figure 1.5.

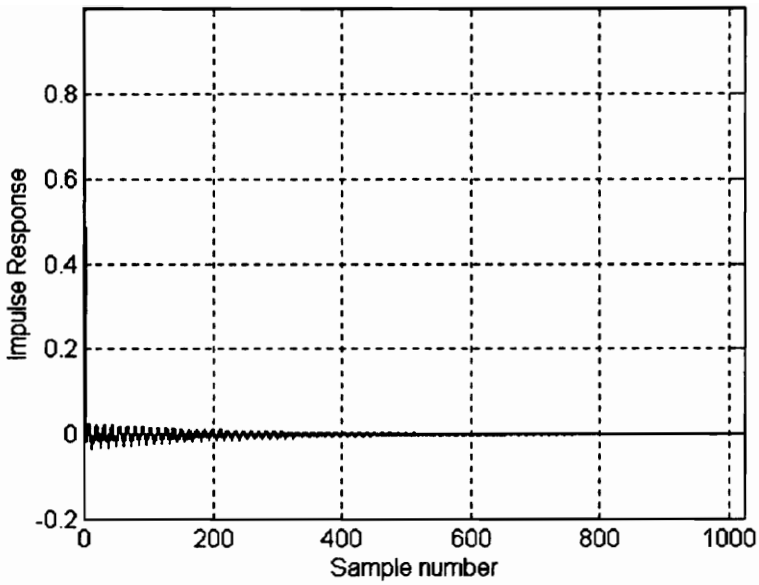


Fig. 1.5 Echo Path Impulse Response for Simulations.

The impulse response beyond 1024 sample points was assumed to be zero. We used two sets of input signals; white Gaussian noise, and speech. Due to the difficulty of defining a speech test signal, the performance requirements are specified, in the ITU-T recommendations, for a white noise input signal. However, "proper performance with speech input signal" is a part of the requirements [16]. We used the speech in a Mylanta commercial as the speech test signal. Figure 1.6 shows the speech test signal used for simulations.

We used *Echo Return Loss* as the performance index of the algorithms. Echo Return Loss (ERL) is defined as the ratio of the energy in the residual echo $e(n)$ to the energy in the original echo $d(n)$. We estimated energy using the 16 point moving average

of the instantaneous squared amplitudes. The convergence rate can be studied using a plot of ERL versus sample number. These plots are usually referred to as the *learning curves* of the algorithm.

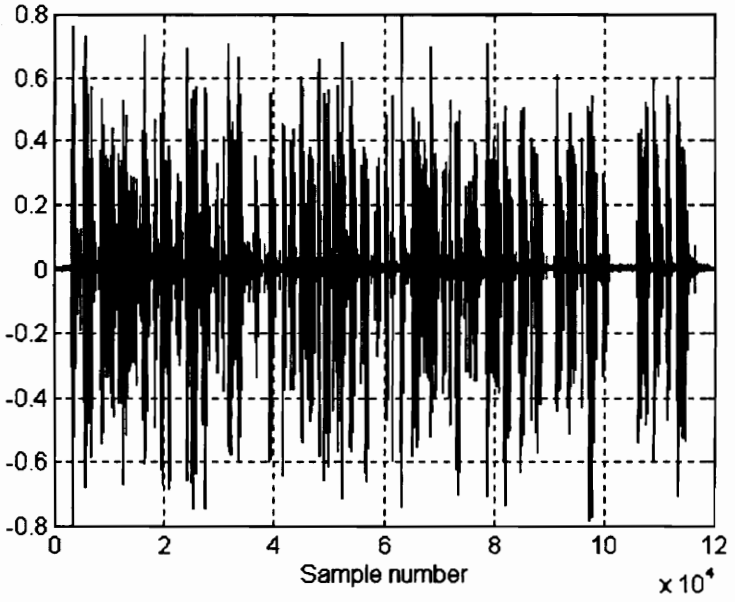


Fig. 1.6 Speech Test Signal for Simulations.

Chapters 2-5 discuss different adaptive filtering algorithms. Chapter 6 discusses some of the practical issues involved in implementing the echo canceler. Chapter 7 summarizes the results and Chapter 8 provides the conclusion.

2. Least Mean Square Algorithm

The Least Mean Square (LMS) algorithm is a widely used adaptation algorithm. The popularity of this algorithm is to a large extent due to its computational simplicity. This algorithm, as the name suggests, attempts to minimize the expected value of the squared error (residual echo). The transversal structure is chosen for the adaptive filter due to its simplicity. The algorithm starts from some initial (arbitrary) value for the tap weight vector, and the weights, adapted according to the stochastic steepest descent algorithm, improve with the number of iterations. The final value so computed for the tap weight vector converges in the mean to the Wiener Solution.

The operation of the LMS algorithm is descriptive of a feedback control system. Basically, it consists of a combination of two basic processes [6]:

1. An *adaptive process*, which involves the automatic adjustment of tap weights.
2. A *filtering process*, which involves (a) forming the inner product of a set of tap inputs and the corresponding set of tap weights emerging from the adaptive process to produce an estimate of a desired response, and (b) generating an estimation error by comparing this estimate with the actual value of the desired response. The estimation error is in turn used to actuate the adaptive process, thereby closing the feedback loop.

The block diagram of Figure 2.1 illustrates these basic components.

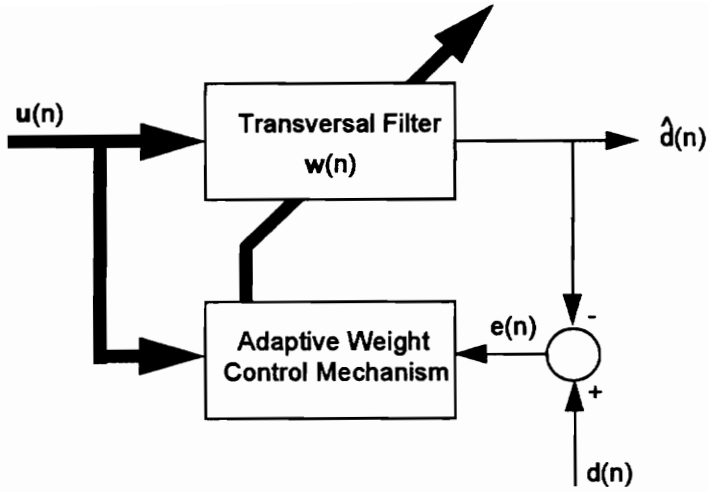


Fig. 2.1 Block Diagram of Adaptive Transversal Filter.

2.1 Description

Let $\mathbf{u}(n)$ denote the vector of tap inputs at time n, and $\hat{d}(n)$ denote the estimate of the desired response at the filter output. By comparing this estimate with the true desired response $d(n)$, we generate the estimation error $e(n)$. Thus, we may write

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n) \\ &= d(n) - \mathbf{w}^H(n)\mathbf{u}(n) \end{aligned} \tag{2.1}$$

where the term $\mathbf{w}^H(n)\mathbf{u}(n)$ is the inner product of the tap weight vector $\mathbf{w}(n)$ and the tap input vector $\mathbf{u}(n)$. The expanded form of the tap-weight vector is described by

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \dots & w_{M-1}(n) \end{bmatrix}^T \tag{2.2}$$

and that of the tap input vector by

$$\mathbf{u}(n) = \begin{bmatrix} u(n) & u(n-1) & \dots & u(n-M+1) \end{bmatrix}^T \tag{2.3}$$

The criterion function used in the LMS algorithm is the expected value of the squared error, and it may be written as

$$J(n) = E\left[e^2(n)\right] \tag{2.4}$$

Assuming that the tap input vector $\mathbf{u}(n)$ and the desired response $d(n)$ are jointly stationary, the mean squared error $J(n)$ is a convex function of the tap weight vector $\mathbf{w}(n)$ with a unique minimum. Wiener filter theory can be used to compute the optimum tap weight vector [6]. The adaptive process attempts to locate the optimum point using the steepest descent method. According to this method, the updated value of the tap weight vector at time $n + 1$ is computed by using the simple recursive relation [6]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu\left[-\nabla\left(J(n)\right)\right] \tag{2.5}$$

where $\nabla\left(J(n)\right)$ denotes the value of the gradient vector at time n, and μ is a positive real valued constant. The factor $\frac{1}{2}$ is used merely for convenience. The gradient vector $\nabla\left(J(n)\right)$ is given by [6]

$$\nabla\left(J(n)\right) = -2\mathbf{p} - 2\mathbf{R}\mathbf{w}(n) \tag{2.6}$$

where $\mathbf{p}$ is the cross-correlation vector between the tap input vector $\mathbf{u}(n)$ and the desired response $d(n)$, and $\mathbf{R}$ is the correlation matrix of the tap input vector $\mathbf{u}(n)$. Exact knowledge of $\mathbf{p}$ and $\mathbf{R}$ is usually not available. Instead they have to be estimated. The simplest choice of estimators for $\mathbf{p}$ and $\mathbf{R}$ is to use instantaneous estimates as shown in (2.7) and (2.8) respectively.

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^{H}(n) \tag{2.7}$$

$$\hat{\mathbf{p}}(n) = \mathbf{u}(n)d^{*}(n) \tag{2.8}$$

Substituting (2.7) and (2.8) in the steepest descent algorithm, as defined by (2.5), we get the following recursive relation for updating the tap weight vector:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)\left[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)\right] \tag{2.9}$$

The hat over the symbol for the tap weight vector distinguishes it from the value obtained using the steepest descent algorithm. We may write (2.9) in the form of three basic relations as follows [6]:

1. *Filter output*:

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \tag{2.10}$$

2. *Estimation error*:

$$e(n) = d(n) - y(n) \tag{2.11}$$

3. *Tap weight adaptation*:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n) \tag{2.12}$$

The algorithm described by (2.10) to (2.12) is the Least Mean Square or LMS algorithm.

2.2 Convergence behavior

Due to the presence of feedback in the LMS algorithm, there exists a possibility of it becoming unstable. The stability of the algorithm depends on the step size parameter μ. For the LMS algorithm to *converge in the mean*, which means that the expected value of

the tap weight vector $\hat{\mathbf{w}}(n)$ approaches the optimum (Wiener) solution $\mathbf{w}_0$ as the number of iterations n approaches infinity, μ should satisfy [6]:

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{2.13}$$

where λ_{max} is the largest eigenvalue of the correlation matrix $\mathbf{R}$.

Another form of convergence that is of interest to us is *convergence in the mean square*. This means that the final (steady-state) value of the mean squared error $J(\infty)$ is finite. This occurs if, and only if [6]:

$$\sum_{i=1}^{M} \frac{\mu\lambda_i}{2-\mu\lambda_i} < 1 \tag{2.14}$$

where λ_i, i = 1, 2, ... , M, are the eigenvalues of the correlation matrix $\mathbf{R}$ and M is the number of taps.

The tap weight vector $\hat{\mathbf{w}}(n)$ does not converge exactly to the minimum point of $J(n)$. Instead, the algorithm executes a random motion around the minimum point due to the presence of gradient noise. This results in a steady state error $J(\infty)$ which is always greater than the minimum mean squared error J_{min} that corresponds to the Wiener solution. The difference between the final value $J(\infty)$ and the minimum value J_{min} is

called the *excess mean squared error* $J_{ex}(\infty)$. The ratio of $J_{ex}(\infty)$ to $J_{\min}$ is called the *misadjustment* $\mathcal{M}$, which is a measure of the deviation of the solution, computed by the LMS algorithm, from the Wiener solution. The misadjustment $\mathcal{M}$ is given by [6]:

$$\mathcal{M} \approx \frac{\mu M \lambda_{av}}{2} \tag{2.15}$$

where λ_{av} is the averaged eigenvalue of the underlying correlation matrix $\mathbf{R}$ of the tap inputs.

$$\lambda_{av} = \frac{1}{M}\sum_{i=1}^{M}\lambda_i \tag{2.16}$$

Suppose that the ensemble averaged learning curve of the LMS algorithm is approximated by a single exponential with time constant $\tau_{mse,av}$. The average time constant for the LMS algorithm is given by [6]:

$$\tau_{mse,av} \approx \frac{1}{2\mu\lambda_{av}} \tag{2.17}$$

Equations (2.15) and (2.17) show that the misadjustment $\mathcal{M}$ is directly proportional to the step size parameter μ, whereas the average time constant $\tau_{mse,av}$ is inversely proportional to μ. Therefore if μ is reduced so as to reduce the misadjustment $\mathcal{M}$, the settling time is increased and vice versa.

2.3 Suitability to echo cancellation

Due to its inherent simplicity, the LMS algorithm is easy to implement. In most Digital Signal Processors (DSPs) it takes approximately 3M instruction cycles for execution. The algorithm is further known to be robust against implementation (numerical) errors introduced by finite register length. These properties make the LMS algorithm attractive for echo cancellation applications.

However, the speed of convergence of the mean squared error depends on the spread of the eigenvalues of **R**. When the eigenvalue spread is large, the convergence slows down. For white noise, the eigenvalues are equal; the LMS algorithm has a high convergence rate. In most echo cancellation applications, speech is the input signal. Speech signals have a large eigenvalue spread and this slows the convergence. The LMS learning curves in Figure 2.2 show that the LMS algorithm converges much slower with speech as input than with white noise as input. Here the echo return loss is computed using a 16 point moving average, as explained in Chapter 1. Note that the learning curve depends on the number of points used in averaging. For example, Figure 2.3 shows the learning curve obtained using the same speech signal, but with a 128 point moving average

to compute the ERL. Note that with longer averaging the minimum steady state ERL appears to be larger. We know that the actual instantaneous ERL is the same in both the cases. In the sequel, the echo return loss is computed using a 16 point moving average.

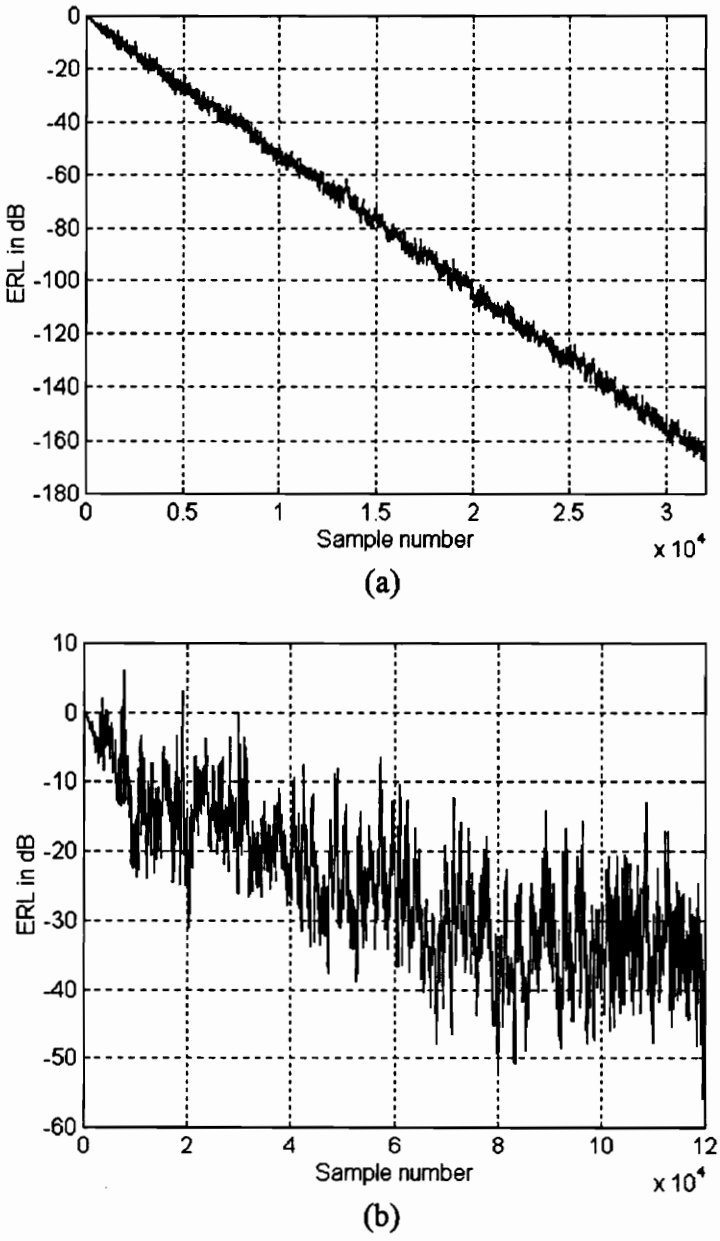


Fig. 2.2 Learning Curves of the LMS Algorithm with
(a) White Noise Input (b) Speech Input.

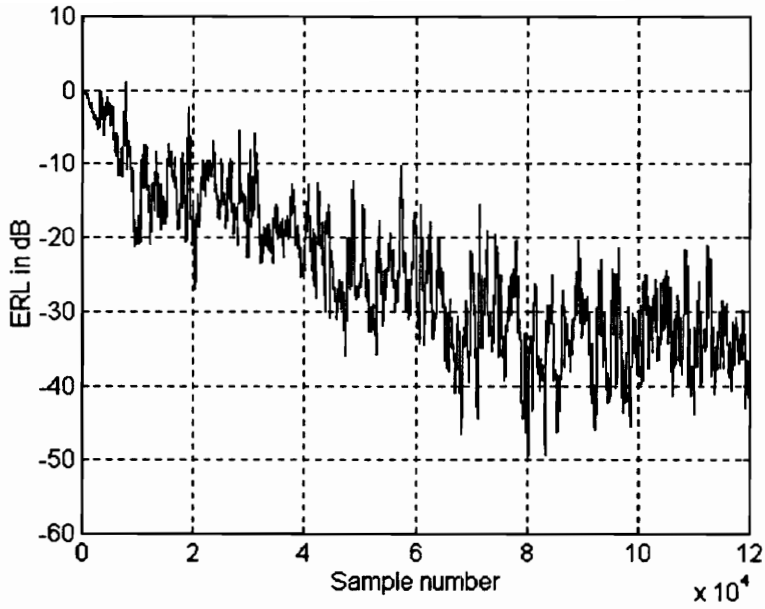


Figure 2.3 Learning Curve of LMS with White Noise Input
(Using 128 Point Moving Average for ERL Computation)

The undesirable dependence of the LMS convergence rate on the eigenvalue spread has motivated the investigation of other adaptive filtering algorithms, with signal independent convergence characteristics, for echo cancellation.

3. Normalized Least Mean Square Algorithm

The Normalized Least Mean Square algorithm is a modified version of the LMS algorithm. In the LMS algorithm, the correction factor to the tap weight vector $\mathbf{w}(n)$ is computed as $\mu\mathbf{u}(n)e^*(n)$. Since this quantity is directly proportional to the tap input vector $\mathbf{u}(n)$, the error in the gradient estimate gets magnified for large $\mathbf{u}(n)$. This problem can be avoided by normalizing the correction factor by the squared Euclidean norm of the tap input vector $\mathbf{u}(n)$. This variant of the LMS algorithm, with the normalized correction factor, is called the Normalized LMS (NLMS) algorithm.

3.1 Description of the algorithm

The update produced by the NLMS algorithm can be interpreted as the solution to the following optimization problem [7]:

$$\min_{\mathbf{w}(n)} \left\{ \left\| d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \right\|^2 + \left(\frac{1}{\tilde{\mu}} - 1 \right) \left\| \mathbf{u}(n) \right\|^2 \left\| \mathbf{w}(n) - \mathbf{w}(n-1) \right\|^2 \right\} \qquad (3.1)$$

Thus, for $\tilde{\mu} \in [0,1]$, the new estimate of the tap weight vector produced by the NLMS algorithm is a compromise between the fit to the new desired signal and the deviation from the prior estimate.

The following steps constitute the NLMS algorithm [6]:

1. *Filter output*:

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \tag{3.2}$$

2. *Estimation error*:

$$e(n) = d(n) - y(n) \tag{3.3}$$

3. *Tap weight adaptation*:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}\mathbf{u}(n)e^*(n)}{\|\mathbf{u}(n)\|^2} \tag{3.4}$$

The division by $\|\mathbf{u}(n)\|^2$ in (3.4) can lead to numerical problems, when the input signal is small. This problem can be surmounted by adding a small positive value a to the norm $\|\mathbf{u}(n)\|^2$. Hence, the tap weight adaptation equation (3.4) is modified as follows:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}\mathbf{u}(n)e^*(n)}{a + \|\mathbf{u}(n)\|^2} \tag{3.5}$$

We used $a = 0.001$ in our simulations and implementation.

3.2 Convergence behavior of the NLMS algorithm

Under certain simplifying assumptions the necessary and sufficient condition for the convergence of the NLMS algorithm is [7, 8]

$$\tilde{\mu} \in (0,2). \tag{3.6}$$

Unlike the convergence condition for the LMS algorithm, given in (2.13), the above condition is independent of the signal characteristics. The fastest convergence occurs when [7]

$$\tilde{\mu} = 1 \tag{3.7}$$

The LMS algorithm does not guarantee such optimum step size constant. A step size, optimum for a certain class of signals, may not be the optimum for others, or may even result in divergence of the algorithm. This forces one to use a conservative value of the step size constant resulting in a slower convergence. The NLMS algorithm may be interpreted as the LMS algorithm with a time varying step size given by

$$\mu(n) = \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2}. \tag{3.8}$$

Equivalently, the NLMS algorithm chooses the optimum step size depending on the input signal.

The convergence rates of the NLMS and LMS algorithms are comparable for white $\mathbf{u}(n)$. However with colored signals like speech, the NLMS algorithm converges faster than the LMS algorithm. The learning curves of the LMS and NLMS algorithms are shown in Figure 3.1, for our speech signal as input. Figure 3.2 shows the learning curves of LMS and NLMS, obtained by ensemble averaging the learning curves from 15 different speech signals. We see that the NLMS algorithm yields a faster convergence rate and a better steady state ERL than the LMS algorithm.

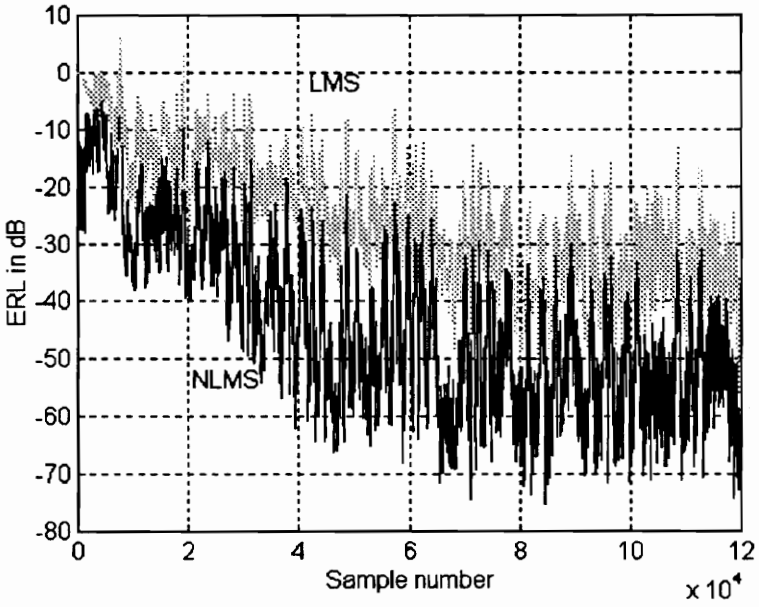


Fig. 3.1 Learning Curves of the LMS and NLMS Algorithms with Speech as Input.

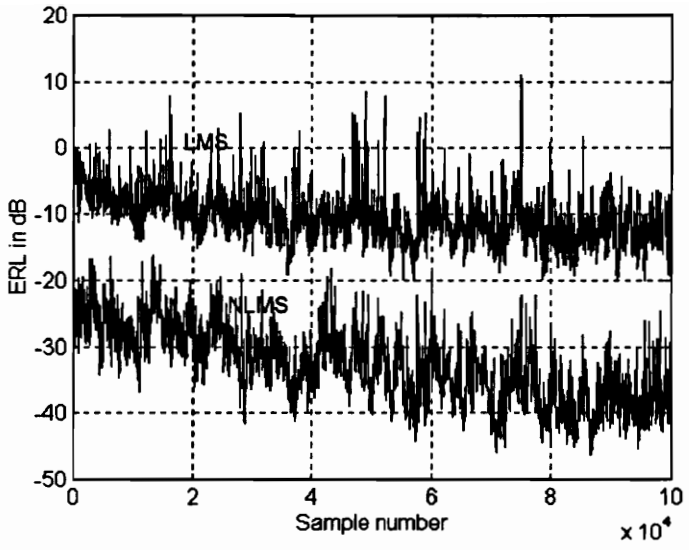


Fig. 3.2 Ensemble Averaged Learning Curves of
the LMS and NLMS Algorithms with Speech as Input.

3.3 Suitability for echo cancellation

The complexity of the NLMS algorithm is comparable to that of the LMS algorithm. The faster convergence of the NLMS algorithm with speech signals as input makes it more suitable than the LMS algorithm for echo cancellation.

Even though NLMS is a very simple algorithm, the computational power of present day DSPs limits the maximum order of the NLMS adaptive filter to a few hundred (details are given in Chapter 7). While such an order is sufficient for network echo cancelers, it is insufficient for acoustic echo cancelers operating in rooms with long reverberation time constants. The acoustic echo path is usually very long; it ranges from several hundreds of taps for hands-free telephones, to a few thousands for modern teleconferencing systems [9]. This has prompted research into algorithms that are computationally more efficient than the NLMS algorithm.

4. Least Squares Algorithm

While the LMS algorithm attempts to minimize the mean squared error, the *Least Squares* algorithm minimizes the weighted sum of squared errors. This may be viewed as an alternative to Wiener filter theory. Wiener filters are designed based on ensemble averages, and hence the result obtained is the same for different operating conditions, which assume a stationary environment. On the other hand, the least squares algorithm uses time averages. Here the optimum filter obtained depends on the number of samples used for averaging.

4.1 Description

In the least squares method, we minimize the cost function

$$\xi(n) = \sum_{i=-\infty}^{k} |e(i)|^2 \omega(k-i) \tag{4.1}$$

where $\{\omega(0), \ \omega(1), \ \omega(2), \ .....\}$ is a sequence of weights. The weighting factor $\omega(k-i)$ can be used to ensure that the filter forgets the past data, and tracks the statistical variations of the data. This enables the filter to operate in a nonstationary environment. The weighting sequence is chosen based on the type of time variation expected for the echo path. Commonly used weighting sequences are

$$\omega(i) = \lambda^i, \qquad \text{where } 0 < \lambda < 1; \tag{4.2}$$

and

$$\omega(i) = \begin{cases} 1, & \text{for } 0 \le i \le M-1; \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

The weighting described by (4.2) is called an *exponential weighting*. Here the effect of the past data fades exponentially. The weighting given in (4.3) is called a *sliding window weighting*. Here only the most recent M samples are used, with equal weights, for estimation. Both these weighting methods can be used to handle slow time variation in the echo path [1]. In the sequel, we shall only consider the exponential weighting (4.3).

The optimum value of the tap weight vector $\hat{\mathbf{w}}(n)$, which minimizes the cost function $\xi(n)$, is given by the *normal equations*, written in matrix form [6]:

$$\Phi(n)\hat{\mathbf{w}}(n) = \theta(n) \tag{4.4}$$

The M-by-M correlation matrix $\Phi(n)$ is defined by

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)\mathbf{u}^{H}(i) \tag{4.5}$$

The M-by-1 cross-correlation $\theta(n)$ between the tap inputs of the transversal filter and the desired response is defined by

$$\theta(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}(i)d^{*}(i) \tag{4.6}$$

Equation (4.5) can be written in a recursive form as shown below:

$$\Phi(n) = \lambda\Phi(n-1) + \mathbf{u}(n)\mathbf{u}^{H}(n) \tag{4.7}$$

Similarly, (4.6) can be written in a recursive form as

$$\theta(n) = \lambda\theta(n-1) + \mathbf{u}(n)d^{*}(n) \tag{4.8}$$

To solve for the optimal tap weight vector $\hat{\mathbf{w}}(n)$ in accordance with (4.4), we need to compute the inverse of the correlation matrix $\Phi(n)$. However, computing the inverse of a matrix is a complex operation, especially when the order M is high. The complexity can

be reduced by using a basic result in matrix algebra known as the *matrix inversion lemma*. This leads to the following recursive equation for the inverse of the correlation matrix [6]:

$$\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)\mathbf{u}(n)\mathbf{u}^{H}(n)\Phi^{-1}(n-1)}{1+\lambda^{-1}\mathbf{u}^{H}(n)\Phi^{-1}(n-1)\mathbf{u}(n)} \tag{4.9}$$

The above equation may be rewritten, for convenience of computation, as

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^{H}(n)\mathbf{P}(n-1) \tag{4.10}$$

where

$$\mathbf{P}(n) = \Phi^{-1}(n) \tag{4.11}$$

and

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1+\lambda^{-1}\mathbf{u}^{H}(n)\mathbf{P}(n-1)\mathbf{u}(n)} \tag{4.12}$$

Using (4.4), (4.8), (4.10), and (4.11), we can get the following recursive equation for updating the least squares estimate $\hat{\mathbf{w}}(n)$ for the tap weight vector [6].

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n) \tag{4.13}$$

where $\alpha(n)$ is the a priori estimation error defined by

$$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n) \tag{4.14}$$

Equations (4.12), (4.14), (4.13), and (4.10), in that order, constitute the *Recursive Least Squares* (RLS) algorithm, as summarized below:

$$\mathbf{k}(n) = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{1 + \lambda^{-1}\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{u}(n)}$$

$$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n)$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1)$$

4.2 Convergence Behavior

The recursion of (4.10) of the RLS algorithm needs to be started with a value of $\mathbf{P}(0)$ that assures the nonsingularity of the correlation matrix $\Phi(n)$. This is usually done by setting

$$\Phi(0) = \delta\mathbf{I} \tag{4.15}$$

where δ is a small positive constant. Correspondingly, the initial value of $\mathbf{P}(n)$ is set at

$$\mathbf{P}(0) = \delta^{-1}\mathbf{I} \tag{4.16}$$

This introduces a bias in the estimate of the tap weight vector $\hat{\mathbf{w}}(n)$ produced by the RLS algorithm [6]. Hence, we may write the mean value of $\hat{\mathbf{w}}(n)$ as

$$E[\hat{\mathbf{w}}(n)] = \mathbf{w}_0 + \mathbf{b}(n) \tag{4.17}$$

where $\mathbf{w}_0$ is the optimal tap weight vector, and the bias $\mathbf{b}(n)$ is given by [6]

$$\mathbf{b}(n) = -\frac{\delta}{n}\mathbf{R}^{-1}\mathbf{w}_0, \qquad \text{for large } n \tag{4.18}$$

Here $\mathbf{R}$ is the M-by-M ensemble averaged correlation matrix of the tap input vector $\mathbf{u}(n)$. Equation (4.18) shows that the bias $\mathbf{b}(n)$ converges to zero as the number of iterations n approaches infinity. Thus, RLS produces an asymptotically unbiased estimate of the optimal tap weight vector. That is, the RLS algorithm is convergent in the mean.

The RLS algorithm exhibits exponential convergence depending on the forgetting factor λ. The time constants of the process, given by [10]

$$\tau_i = \frac{1}{1-\lambda}, \tag{4.19}$$

are the same for all coordinates i, $1 \le i \le N$. Unlike for the LMS algorithm, RLS convergence is independent of the eigenvalue spread of the autocorrelation matrix.

The RLS algorithm converges in the mean square in about $2M$ iterations. This means that the RLS algorithm usually converges an order of magnitude faster than the LMS algorithm.

Under certain simplifying assumptions, the misadjustment in the estimates produced by the LMS algorithm is given by [10]

$$\mathbf{M} = \frac{1-\lambda}{1+\lambda} M \tag{4.20}$$

This shows that the misadjustment M increases, as the memory λ decreases. That is, fast adaptation leads to noisy estimates. In a stationary environment, the best steady-state performance results from slow adaptation, which corresponds to $\lambda = 1$.

4.3 Suitability to Echo Cancellation

The RLS algorithm has the fastest convergence rate among the algorithms we have seen so far. The learning curves, obtained from simulations, shown in Figure 4.1, corroborate this. In addition, as mentioned in Section 4.2, the convergence rate does not depend on the statistics of the signal. Figure 4.2 shows the learning curve of the RLS algorithm with our speech signal as input. We see that the RLS algorithm still retains its fast convergence. These features make RLS highly suited for the echo cancellation application.

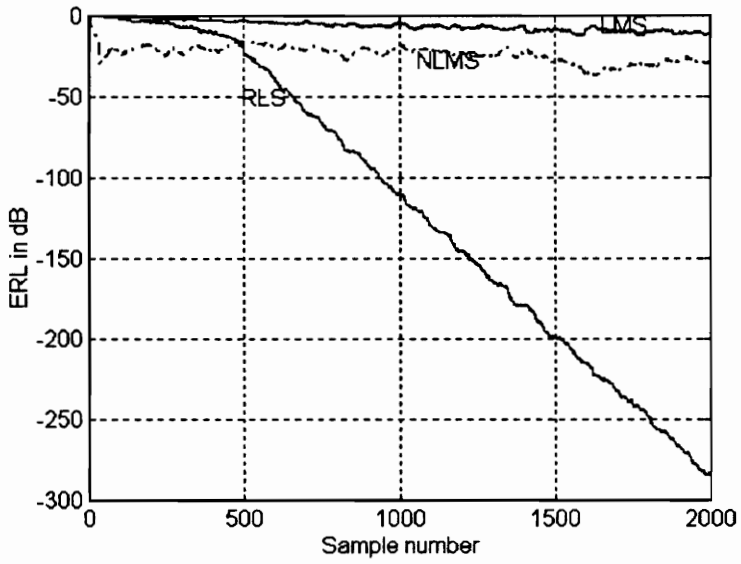


Fig. 4.1 Learning Curves of the RLS, LMS, and NLMS Algorithms with White Noise as Input.

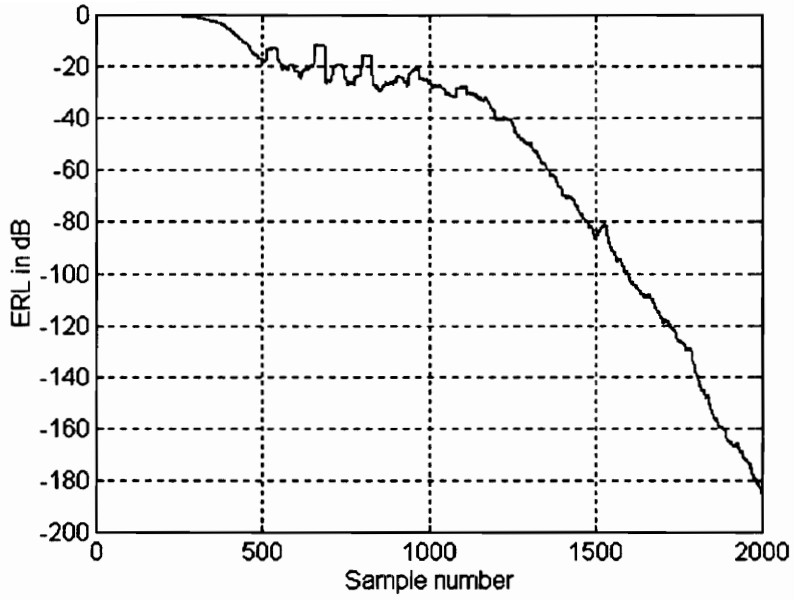


Fig. 4.2 Learning Curve of the RLS Algorithm
with Speech as Input.

However, numerical instabilities or finite precision problems pose potential problems for any RLS adaptation algorithm, whether transversal or lattice [10]. The transversal RLS algorithms tend to become unstable, especially when λ is less than 1. On the other hand, the convergence of the RLS algorithm is slowed for values of λ close to 1.

The lattice RLS algorithms are known to be more robust to numerical problems than the transversal RLS algorithms. The downside is that the lattice RLS algorithms are highly computation intensive. The maximum order of the filter that can be implemented on present day DSPs is limited to a few tens (details are provided in Chapter 7). Due to these limitations, the RLS algorithm is not a practically viable option for echo cancellation.

5. Subband NLMS Algorithm

As mentioned in Chapter 3, the acoustic echo canceler needs an adaptive filter with a few thousands of taps. The computational burden associated with the adaptive filtering algorithms discussed in Chapters 2 - 4 precludes them from being used for acoustic echo cancellation in rooms with a long impulse response. Subband techniques may be used to reduce the computational burden [12]. By processing the signals in subbands, both the number of taps and the weight update (adaptation) rate can be decimated in each subband. This reduces the computational burden by approximately the number of subbands.

5.1 Description

Figure 5.1 shows the architecture of the subband adaptive filter. The far-end signal $u(n)$, and the residual error signal $e(n)$, are split into N subband signals using contiguous single sideband bandpass filters $F_0, F_1, \ldots, F_{N-1}$ that span the signal bandwidth. The signals in each subband are decimated by a factor D. An adaptive filter is used in each subband and the subband adaptive weights for each are computed by the complex NLMS algorithm. The adaptive weights in each subband are then transformed (by FFT) into the frequency domain, appropriately stacked, and inverse transformed to obtain the wideband filter coefficients [11]. The wideband filter convolution can be efficiently computed by using orthogonal transform techniques such as the FFT.

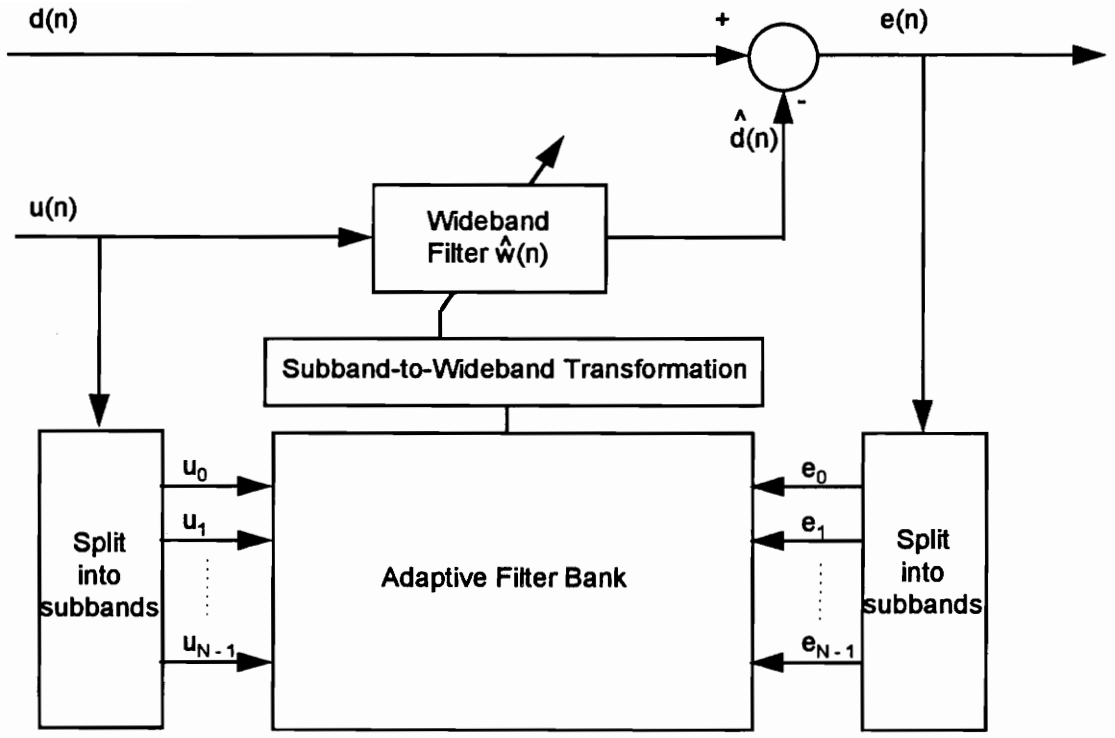


Fig. 5.1 Subband Filter Architecture.

5.1.1 Subband Analysis Filter

The wideband signal is split into subbands using a subband analysis filter. We employ the *polyphase FFT technique* to perform this subband splitting. This technique realizes N contiguous single sideband bandpass filters whose outputs are downsampled by a factor $D = N/2$ to produce N complex subband signals [11]. Since the signals involved in an acoustic echo canceler are real, it is sufficient to process only half of the complex subbands.

The output of the mth bandpass filter can be mathematically expressed as

$$x_m(n) = \sum_{k=0}^{K-1} a_k e^{j2\pi \frac{mk}{N}} u(n-k) \tag{5.1}$$

where the a_k are the coefficients of the K-point prototype FIR filter. K is chosen such that it is an integral multiple of the number of subbands N. Here we have written the output as the convolution of the wideband input signal with the frequency shifted prototype filter.

The K-point filter can be split into a set of N smaller filters, called polyphase filters, with impulse responses

$$h_k(l) = a_{k+lN} \qquad k = 0, 1, \ldots, N-1 \qquad l = 0, 1, \ldots, L-1 \tag{5.2}$$

where $L = K/N$. Modifying (5.1) as the convolution of the wideband signal with the frequency shifted prototype filters, defined in (5.2), we get

$$\begin{aligned} x_m(n) &= \sum_{k=0}^{N-1} \left[\sum_{l=0}^{L-1} a_{k+lN} e^{j2\pi \frac{m(k+lN)}{N}} u(n-k-lN) \right] \\ &= \sum_{k=0}^{N-1} \left[\sum_{l=0}^{L-1} a_{k+lN} e^{j2\pi \frac{mk}{N}} u(n-k-lN) \right] \end{aligned} \tag{5.3}$$

Equation (5.3) may be rewritten as

$$x_m(n) = \sum_{k=0}^{N-1} e^{j2\pi\frac{mk}{N}} \sum_{l=0}^{L-1} a_{k+lN} u(n-k-lN) \tag{5.4}$$

The quantity in the inner summation may be recognized as the output of the polyphase filters. This shows that the (inverse) FFT of the polyphase filtered wideband signal also gives the bandpass filtered output. This, in fact, is a computationally more efficient way.

Since the bandpass filter outputs are decimated by a factor $D = N/2$, we perform the inverse FFT operation only after every batch of $N/2$ new samples of the far end signal $u(n)$ are received.

5.1.2 Wideband Convolution

The most computation intensive task in a long subband NLMS adaptive filter is the convolution of the far end signal $\mathbf{u}(n)$ with the wideband filter coefficients $\hat{\mathbf{w}}(n)$ to produce $\hat{d}(n)$, an estimate of the echo. This can be performed efficiently by using orthogonal transform techniques such as the FFT. However, the usual FFT based block convolution approach [13], though very efficient, introduces a block delay. This problem can be eliminated by dividing the wideband filter coefficients into segments of equal

length. We process the first segment by direct convolution, while the remaining segments are processed by fast convolutions using FFTs, as illustrated in Figure 5.2 [11].

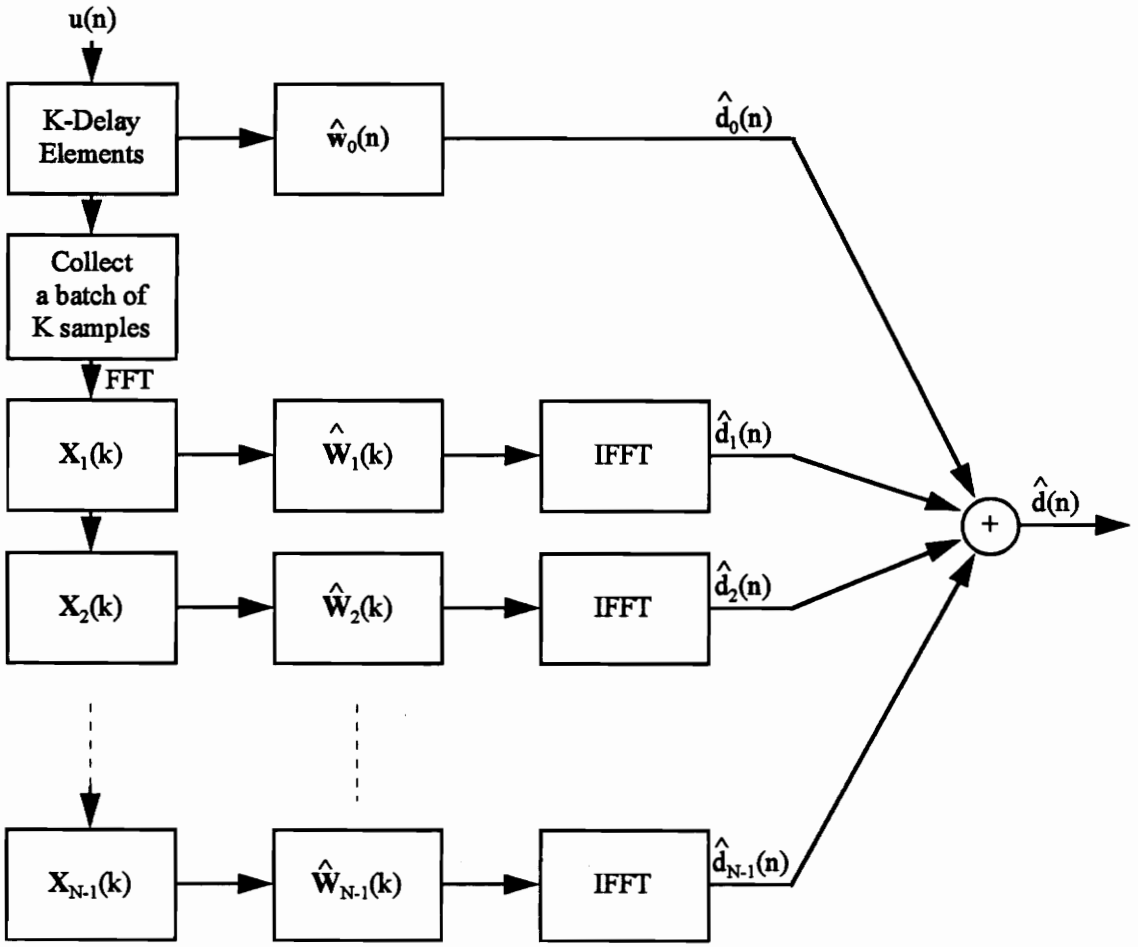


Fig. 5.2 Implementation of Wideband Convolution.

We divide the M-point wideband filter into N filters, with the kth filter having the following $K\ (= M/N)$ taps:

$$\hat{w}_k(n) = \hat{w}(kK + n) \qquad n = 0, 1, 2, \ldots, K - 1 \tag{5.5}$$

These weights are modified after every 4096 samples, as explained in Section 5.2. As shown in Figure 5.2, the estimated echo signal $\hat{d}(n)$ is the sum of the nth outputs of these N filters. The nth output of the kth filter depends only on the far end signal at and prior to $(n - kK)$ i.e., for all values of $k \neq 0$, the input depends only on samples received at least K time steps before. Hence the filtering operation can be performed in the frequency domain, without introducing any delay, for all the filters, except the first one. We pad K zeroes, to avoid circular convolution, at the end of the K input samples to each filter and use the overlap-add method to convolve using a $2K$ point FFT [13]. This procedure reduces the number of computations by approximately the number of segments.

5.2 Convergence Behavior

Since each of the NLMS subband adaptive filters operates on a decimated narrow-band signal, which then occupies the entire band, the ill-effects of the colored signals, on the convergence of NLMS, are mitigated. The subband adaptive filter achieves a reduction in the number of computations by decimating the number of updates. However, the decimation of the updates slows the convergence, since the number of weight updates per unit time decreases. Hence, the subband adaptive filter has a poor convergence rate when the decimation rate is large [14]. We can effect compromise between performance and complexity by choosing the decimation rate, and hence the number of subbands.

Figure 5.3 shows the learning curve of a subband NLMS adaptive filter with 128 subbands and 1024 taps. A 511th order low pass filter with cutoff frequency at 1/64, designed using the Remez algorithm (*remez*(511, [0,1/128,1/64,1], [1,1,0,0], [1,10]) *in Matlab*), was used as the prototype subband analysis FIR filter. The decimation factor was chosen to be 64. White Gaussian noise was used as the far end signal. The subband-to-wideband transformation of the weights was performed only after every 64 updates of the subband weights. Thus, for every 64 x 64 input samples the filter is actually changed. These 4096 sample blocks are discernible in Figure 5.3. We observe that the convergence rate of the subband NLMS algorithm is slow compared to that of the algorithms discussed in Chapters 2 - 4.

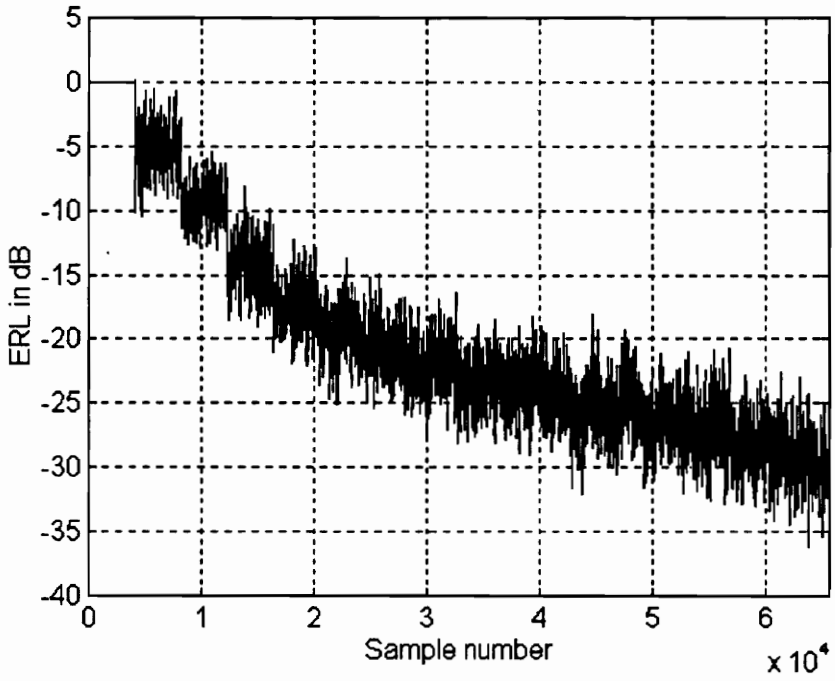


Fig. 5.3 Learning Curve of Subband NLMS Algorithm with White Noise as Input.

5.3 Suitability to Echo Cancellation

The low computational requirements of the subband NLMS algorithm make it very attractive for the acoustic echo cancellation application. We can readily implement adaptive filters with a few thousands of taps, using the subband NLMS algorithm, on low cost DSPs (details provided in Chapter 7).

The only disadvantage of the subband NLMS algorithm is its slow convergence rate. This problem can be surmounted by judicious switching between adaptive filter structures [15]. A detailed discussion will be presented in Section 6.4.

6. Implementation Issues

One of the underlying assumptions in the discussions in Chapters 2 - 5 is that the near-end signal $d(n)$ is the echo signal. In practice, this is not always true. When the near-end talker starts talking, there will be a large additive interference. Further, the (acoustic) background noise, if there is any, contaminates the echo signal. These interfering signals can lead to misconvergence of adaptive filtering algorithms. In this chapter, we discuss techniques to combat the ill effects of these interfering signals. We also discuss the limitations on the achievable ERL due to finite word length of the processors used for implementation. A computationally efficient *switched adaptive filter structure*, which can provide fast initial convergence and high steady state ERL, is proposed. Nonlinear processing of the residual echo, to mitigate the problems due to nonlinearities in the echo path, is proposed. Finally, a method to detect and control howling is discussed.

6.1 Double-talk

The condition where the near-end talker begins speaking, while the far-end talker is still active, is usually referred to as double-talk (both parties are talking simultaneously). The near-end signal $d(n)$, under this condition, is composed of a mixture of the echo of the received signal and the near-end speech, as depicted in Figure 6.1. The echo canceler may interpret the near-end speech as a new echo signal and attempt to adapt to it. This will lead to the misconvergence of the adaptive filter weights $\hat{\mathbf{w}}(n)$ because of the reduced

correlation of the near-end speech and the received (reference) signal $u(n)$. This results in a reduced echo return loss, as well as distorted transmitted speech, thereby seriously degrading the subjective quality of the connection.

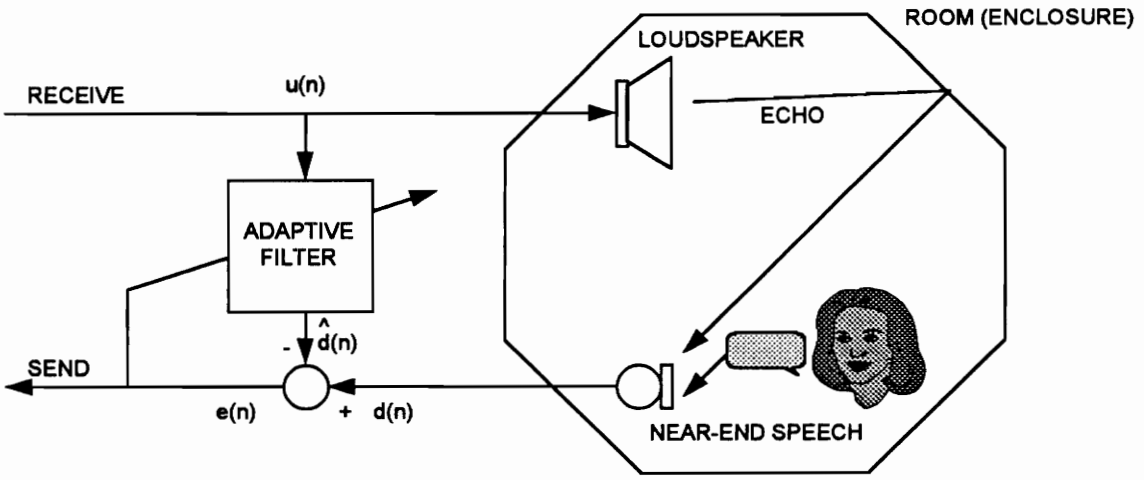


Fig. 6.1 Double-talk Condition.

These problems can be mitigated by stopping the weight adaptation during the double-talk condition. This necessitates the use of a double-talk detector. The double-talk detector should be reliable, since the lack of double-talk detection generally impairs performance. ITU-T : G.165 recommends that the echo canceler should favor break-in at the expense of adaptive operation on false echo [16]. Too many false detections however, slows down the adaptation. Furthermore, the detection time delay should be very short.

Some of the techniques proposed for double-talk detection rely on finding the spectral distance between the near-end signal $u(n)$ and the received signal $d(n)$ [9, 21].

A small spectral distance between the two spectra is a strong indication that only the far-end talker is active. A large distance indicates the presence of near-end talker activity.

The Itakura distance [19] may be used as a measure to estimate the distance between the two spectra. For this, the far-end and the near-end speech signals are divided into 20 msec frames (160 samples) with 50% overlap. Let $\hat{\alpha}(m)$ be the linear prediction (LP) coefficients corresponding to the frame of the far-end signal $u(n;m)$, spanning the time range $n = m - 159, \ldots, m$. Usually 12 coefficients are used, since they are known to sufficiently closely model speech spectra. Let $\hat{\mathbf{R}}_u(m)$ be the 13 × 13 correlation matrix defined as

$$\hat{\mathbf{R}}_u(m) = \begin{pmatrix} \hat{r}_{uu}(0) & \hat{r}_{uu}(1) & . & . & \hat{r}_{uu}(12) \\ \hat{r}_{uu}(1) & \hat{r}_{uu}(0) & . & . & \hat{r}_{uu}(11) \\ . & . & . & . & . \\ . & . & . & . & . \\ \hat{r}_{uu}(12) & \hat{r}_{uu}(11) & . & . & \hat{r}_{uu}(0) \end{pmatrix} \tag{6.1}$$

where the $\hat{r}_{uu}(k)$ are estimates for the autocorrelation of the far-end signal $u(n;m)$ at lag k found using the short-term unbiased estimator shown below:

$$\hat{r}_{uu}(k) = \frac{1}{160 - |k|} \sum_{n=m-159+|k|}^{m} u(n)u(n-k) \tag{6.2}$$

The correlation matrix $\hat{\mathbf{R}}(m)$, defined in (6.1), is a symmetric matrix; it is always nonnegative definite, and almost always positive definite [6]. For a random speech signal, it is guaranteed to be positive definite. Hence $\forall \mathbf{x} \neq 0$, we have the following:

$$\mathbf{x}^{\mathbf{T}}\hat{\mathbf{R}}(m)\mathbf{x} > 0 \tag{6.3}$$

We use the above result in the subsequent discussion to assume the existence of the logarithm of $\mathbf{x}^{\mathbf{T}}\hat{\mathbf{R}}(m)\mathbf{x}$, $\forall \mathbf{x} \neq 0$, without explicitly stating so.

Let $\hat{\beta}(m)$ be the LP coefficients of the corresponding near-end speech frame $d(n;m)$. Then the Itakura (spectral) distance measure for the far and near-end speech spectra is given by [19]

$$d_I(m) = \log \frac{\hat{\beta}^T(m)\hat{R}_u(m)\hat{\beta}(m)}{\hat{\alpha}^T(m)\hat{R}_u(m)\hat{\alpha}(m)} \tag{6.4}$$

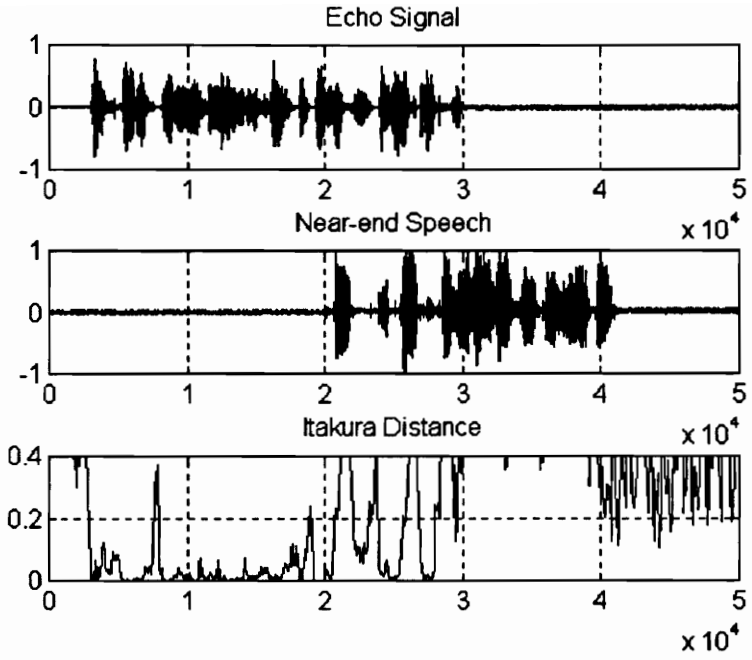
The quantity $\hat{\alpha}^T(m)\hat{R}_u(m)\hat{\alpha}(m)$ is the mean square error (MSE) associated with the prediction of the far-end speech frame $u(n;m)$ using the parameters $\hat{\alpha}(m)$ [19]. If we attempt to predict the far-end speech frame $u(n;m)$ with the coefficients $\hat{\beta}(m)$, the MSE associated with this prediction will be $\hat{\beta}^T(m)\hat{R}_u(m)\hat{\beta}(m)$. We note that

$$\hat{\alpha}^T(m)\hat{R}_u(m)\hat{\alpha}(m) \le \left[\hat{\beta}_{st}{}^T(m)\hat{R}_u(m)\hat{\beta}_{st}(m)\right] \le \left[\hat{\beta}_{dt}{}^T(m)\hat{R}_u(m)\hat{\beta}_{dt}(m)\right] \qquad (6.5)$$

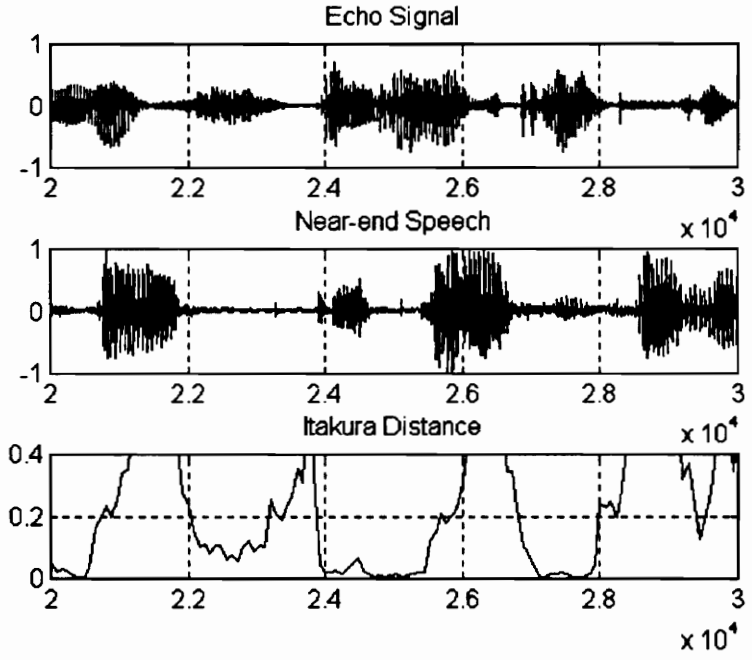
The first inequality in (6.5) is true since the coefficients $\hat{\alpha}(m)$ are the optimal LP coefficients for the speech frame $u(n;m)$, and $\hat{\alpha}^T(m)\hat{R}_u(m)\hat{\alpha}(m)$ is the best possible MSE. In the absence of near-end speech, there will be a high correlation between the far-end and the near-end signals. However, the correlation between the far and near-end signals will be low, when there is near-end speech. Hence we can expect that the coefficients $\hat{\beta}(m)$, derived from the near-end speech $d(n;m)$, would predict the far-end speech $u(n;m)$ better when there is single-talk than when there is double-talk.

Figure 6.2 shows the Itakura distance between the far-end and near-end speech signals. We see that the Itakura distance is mostly less than 0.2, when there is far-end single-talk, while it is mostly more than 0.2 under all the other conditions.

We should inhibit adaptation whenever there is double-talk, since this is the (only) condition that leads to either misconvergence or divergence. Also, we should adapt when there is far-end single talk. From Figure 6.2, we see that 0.15 is a reasonable threshold for double talk detection. Any threshold less than this will result in inhibition of adaptation during far-end single talk, which is undesirable. We also observe that occasionally the filter is adapted during double-talk (for example for *n* between 24000 and 24500 in Figure 6.2 (b)). This happens whenever the far-end speech is stronger than the near-end speech. Thus the Itakura distance is not very sensitive to near-end speech.



(a)



(b)

Fig. 6.2 Itakura Distance Measure for Double-talk Detection.

Another approach to double-talk detection is based on the double-talk detection statistic (DTDS), defined as [18]

$$DTDS(m) = \frac{E_u(m)E_e(m)}{E_u{}^2(m) + E_{\hat{d}}{}^2(m)} \tag{6.6}$$

where $E_u(m)$ is the energy contained in the mth frame of the far-end speech $u(n;m)$ and $E_e(m)$ and $E_{\hat{d}}(m)$ are the energy contained in the corresponding error signal frame $e(n;m)$ and the estimated signal frame $\hat{d}(n;m)$ respectively.

In the absence of near-end speech, the energy in the far-end signal $u(n)$ will not be lower than in the near-end signal $d(n)$, since the echo path is passive. Also, the energy in the residual echo $e(n)$ will be less than the energy in the near-end signal $d(n)$. Hence, the following inequality holds.

$$0 \le E_e(m) \le E_d(m) \le E_u(m) \tag{6.7}$$

Consequently, in the absence of local talk, DTDS has an upper bound of unity. However, when there is double-talk, the energy in the desired signal $d(n)$, and hence the energy in the error signal $e(n)$, grows, while the denominator of (6.6) remains essentially unaltered. This results in an increase of DTDS, roughly proportional to the energy in the near-end speech.

Unlike for the Itakura distance, it is computationally simple to compute the DTDS. Hence, the DTDS can be computed using a frame that slides by one sample rather than by the frame overlap, which jumps by 80 samples in the computation of the Itakura distance. This reduces the lag in the detection process.

Figure 6.3 shows the DTDS corresponding to single-talk and double-talk conditions. We computed DTDS using a single-sample sliding frame of length 32. We see that the DTDS is less than 0.5 when there is far-end single-talk, and more than 0.5 elsewhere. By choosing an appropriate detection threshold for DTDS, D_{Th}, the adaptation of the weights can be controlled as indicated below:

if $E_u E_e \leq D_{Th}\left[E_u^{\,2} + E_{\hat{d}}^{\,2}\right]$

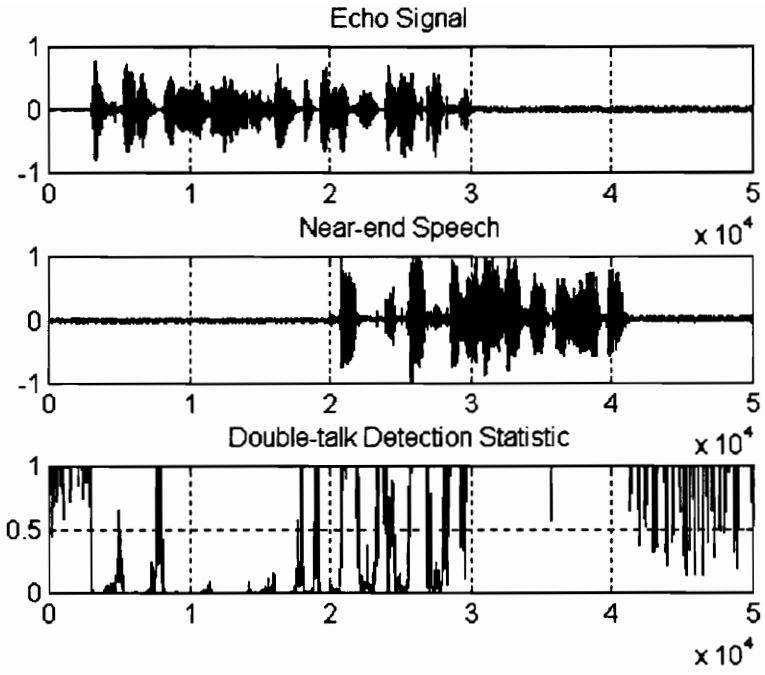
Adapt ;

else

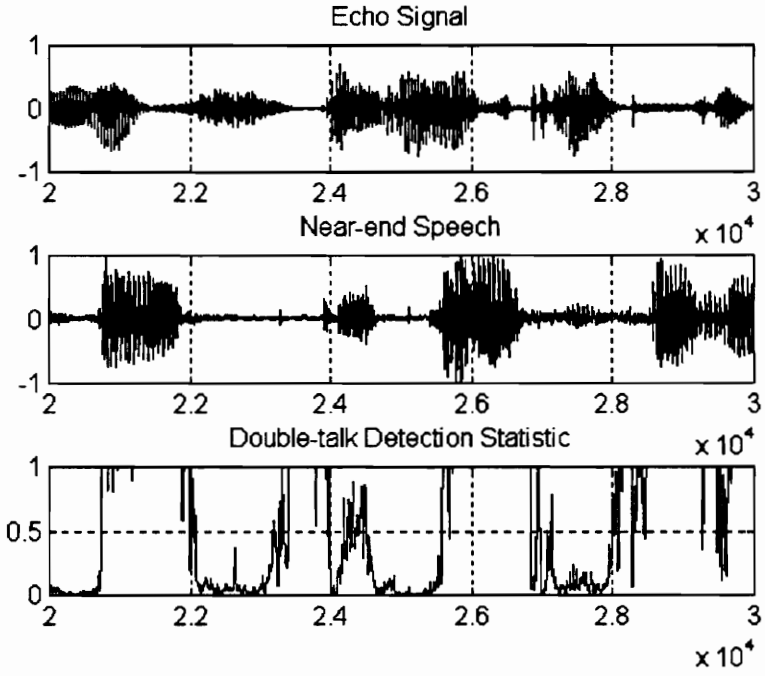
Stop Adapting ;

end ;

The computational simplicity of this approach makes it very attractive for double-talk detection. For the example shown in Figure 6.3, we see that the threshold D_{Th} can be chosen to be 0.5.



(a)



(b)

Fig. 6.3 DTDS for Double-talk Detection.

6.2 Ambient Noise

In the presence of ambient noise, such as when using a mobile hands-free telephone in a noisy vehicle, the near-end signal $d(n)$ is always contaminated with noise. The near-end speech and noise are usually uncorrelated. Hence, the adaptive filter can not cancel this ambient noise and the residual echo signal $e(n)$ always contains the noise. This noisy error signal, when fed to the adaptation process, results in a high misadjustment $\mathcal{M}$, thereby degrading the performance of the echo canceler. Furthermore, the near-end noise causes inconvenience to the far-end listener. Here we propose a procedure to reduce the background noise $v(n)$ in the near-end signal.

Some procedures to reduce the ambient noise were recently proposed [20, 23]. The two-microphone system proposed by Martin and Altenhoner, as the name suggests, uses two microphones [23]. Furthermore, this procedure uses 4 adaptive filters, apart from a time varying filter, and hence needs a lot of computational power. Our approach uses only one microphone and it is less complex since we use only 2 adaptive filters. The approach presented by Ayad, Faucon, and Jeannes uses short-term spectral amplitude estimates of the signal and noise to derive the noise reduction filter [20]. Very good silence detection algorithms are needed for short-term spectral amplitude estimation of noise. However, the silence detection algorithms do not perform reliably unless the SNR is high [19]. We do not rely on silence detection to estimate the statistics of noise.

We employ an adaptive noise canceling technique to filter the noise from the near-end signal. The proposed solution uses two adaptive filters, one to cancel the acoustic

echo (Acoustic Echo Canceler - AEC) and another to cancel the acoustic noise (Acoustic Noise Canceler - ANC), as shown in Figure 6.4. We need an adaptive filter to cancel the noise, since the statistics of the noise and that of the echo (speech) are usually unknown and time-varying.

The echo canceling filter attempts to minimize the mean square residual echo signal $v'(n)$. It does so by converging (in the mean) to the impulse response of the echo path. Once the AEC filter converges, the residual echo signal $v'(n)$ consists of mainly the ambient noise and some amount of residual echo due to the misconvergence and misadjustment. We cancel the ambient noise, by adding another adaptive filter.

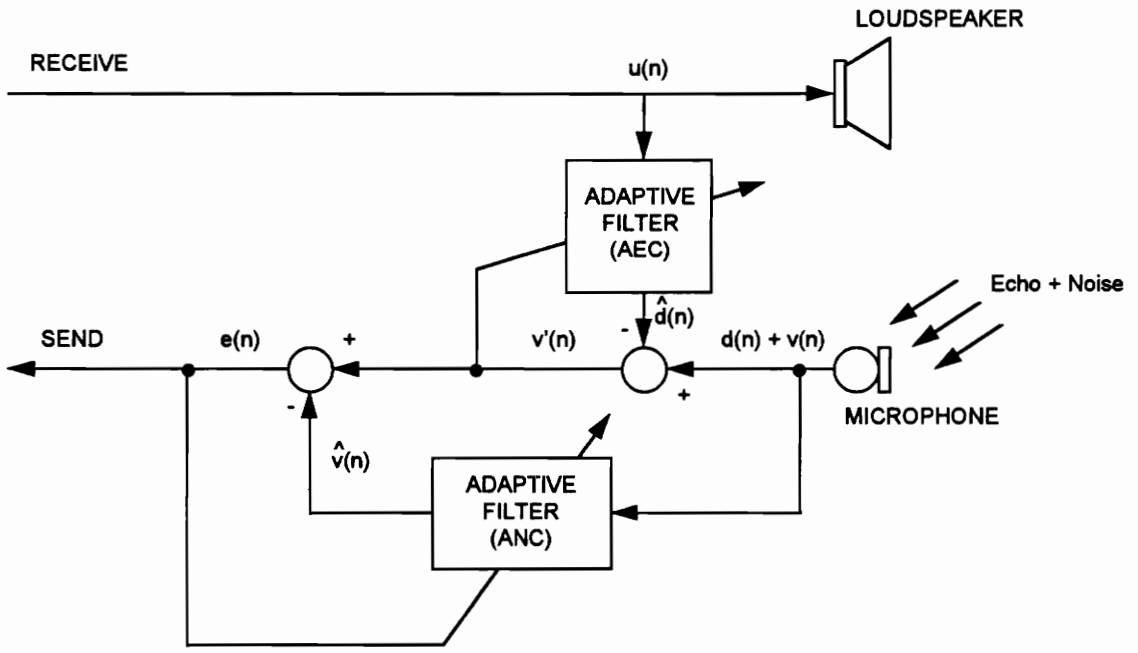


Fig. 6.4 Sequential Echo and Noise Cancellation.

The noise canceling filter attempts to minimize the mean square residual noise signal $e(n)$. It does so by converging to the causal optimal (Wiener) solution for estimating the noise from the noisy near-end signal $d(n)+v(n)$.

During the double-talk and near-end single-talk conditions, we inhibit the adaptation of both filters. Frozen weights are used to continue canceling the echo and the noise.

Let $w(n)$ be the impulse response of the echo path, $\hat{w}(n)$ be the weights to which the acoustic echo canceling filter converges, and $\hat{h}(n)$ be the weights to which the acoustic noise canceling filter converges. Then, we have the following:

$$\begin{aligned} D(z) &= U(z)W(z) \\ \hat{D}(z) &= U(z)\hat{W}(z) \\ \hat{V}(z) &= \left[D(z)+V(z)\right]\hat{H}(z) \\ V'(z) &= D(z)+V(z)-\hat{D}(z) \\ E(z) &= V'(z)-\hat{V}(z) \end{aligned} \tag{6.8}$$

The Fourier transform of the residual echo signal can be rewritten as

$$V'\left(e^{j\omega}\right) = U\left(e^{j\omega}\right)\left\{W\left(e^{j\omega}\right)-\hat{W}\left(e^{j\omega}\right)\right\}+V\left(e^{j\omega}\right) \tag{6.9}$$

From (6.9), we see that the mean square residual echo signal will be minimized when

$$\hat{W}\left(e^{j\omega}\right) = W\left(e^{j\omega}\right) \tag{6.10}$$

Assuming that the acoustic echo canceling filter converges in the mean to the above optimal solution, the Fourier transform of the residual noise signal is

$$E\left(e^{j\omega}\right) = U\left(e^{j\omega}\right)W\left(e^{j\omega}\right)\hat{H}\left(e^{j\omega}\right) + V\left(e^{j\omega}\right)\left\{1 - \hat{H}\left(e^{j\omega}\right)\right\} \tag{6.11}$$

The mean square residual noise signal is minimized if the noise canceling filter converges in the mean to the causal optimal Wiener solution, shown below, to estimate noise from the noisy near-end echo signal $z(n)$ [22].

$$\hat{\mathbf{h}}(n) = \mathbf{R}_{zz}{}^{-1}(n)\mathbf{r}_{zv}(n) \tag{6.12}$$

Thus, $z(n) = d(n) + v(n)$, $\mathbf{R}_{zz}(n)$ is its autocorrelation matrix, and $\mathbf{r}_{zv}(n)$ is the cross-correlation vector, needed to define the Wiener filter solution.

Assuming that the echo and the near-end speech have similar power spectral densities, the optimal Wiener solutions to filter noise from the noisy echo and from the noisy near-end speech are not very different. Hence the noise canceling filter is nearly optimal even under the near-end single-talk condition.

An advantage of this procedure is that the noise and echo canceling filters are completely decoupled, after they converge. However, the adaptation of the echo

canceling filter by a noisy error signal results in a large misadjustment, and consequently in a low ERL.

An alternative will be to filter the noise from the near-end signal and use the noise-free signal for AEC adaptation, as shown in Figure 6.5. Here, both the echo and noise canceling filters attempt (together) to minimize the residual error signal $e(n)$.

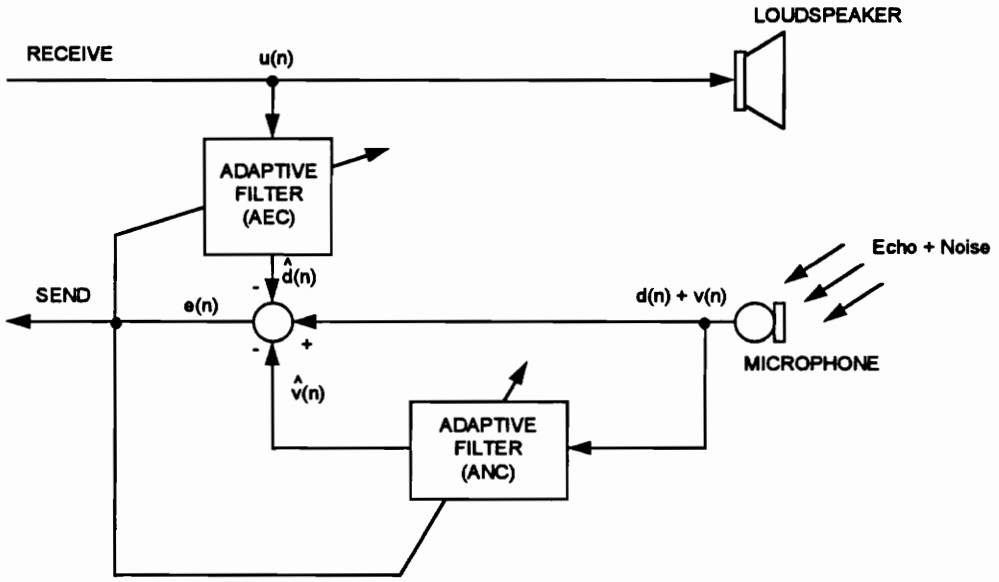


Fig. 6.5 Simultaneous Echo and Noise Cancellation.

Here, we have the following:

$$\begin{aligned} D(z) &= U(z)W(z) \\ \hat{D}(z) &= U(z)\hat{W}(z) \\ \hat{V}(z) &= \left[D(z)+V(z)\right]\hat{H}(z) \\ E(z) &= D(z)+V(z)-\hat{D}(z)-\hat{V}(z) \end{aligned} \tag{6.13}$$

Combining the results in (6.13), and substituting $z = e^{j\omega}$ we get

$$E\left(e^{j\omega}\right) = U\left(e^{j\omega}\right)\left\{W\left(e^{j\omega}\right)\left[1-\hat{H}\left(e^{j\omega}\right)\right]-\hat{W}\left(e^{j\omega}\right)\right\}+V\left(e^{j\omega}\right)\left\{1-\hat{H}\left(e^{j\omega}\right)\right\} \qquad (6.14)$$

The objective is to minimize the mean square residual error signal $e(n)$. The first term of (6.14) is the Fourier transform of the echo related component of the residual error signal, while the second term is due to the noise component. Thus we see that the quantity $\left\{1-\hat{H}(j\omega)\right\}$ plays a role in both the echo and the noise cancellations. That is, the acoustic noise canceling filter affects both the echo and the noise cancellations.

We have assumed that the acoustic echo and the near-end speech have similar power spectral densities. Hence, the acoustic noise canceling filter cancels the near-end speech to some extent, in addition to canceling some of the acoustic noise, during near-end single-talk. However, we pay this trade-off price to get a better echo return loss than with the echo cancellation followed by noise cancellation approach. Since both adaptive filters in the simultaneous echo and noise cancellation approach are adapted using the noise-free error signal, misadjustment is reduced.

We simulate the noise cancellation schemes using orthogonal echo and noise signals. This choice of signals is made to make it easy to observe the effect of the different noise cancellation approaches on the echo and the noise. The signal exists in the frequency

range $(0, 0.25)$ and the noise exists in the range $(0.75, 1)$. Both consist of filtered white noise. Their power spectral densities are shown in Figure 6.6.

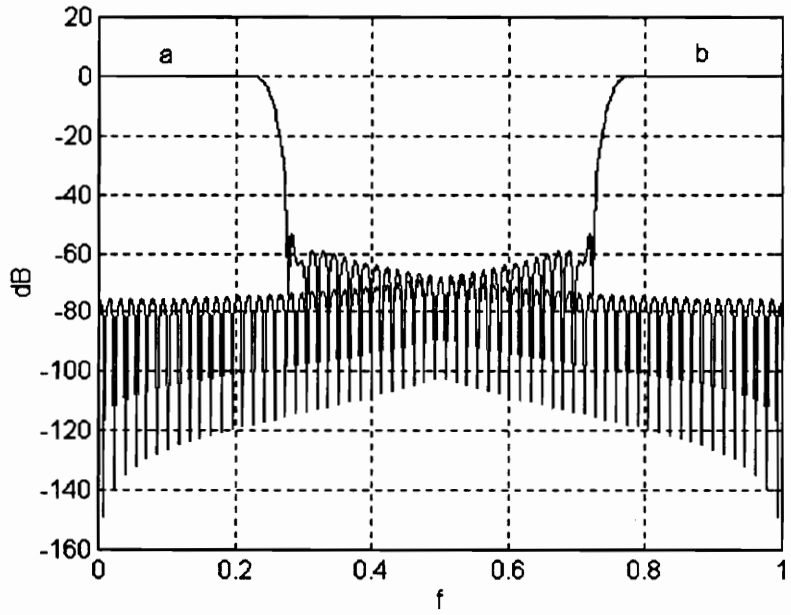
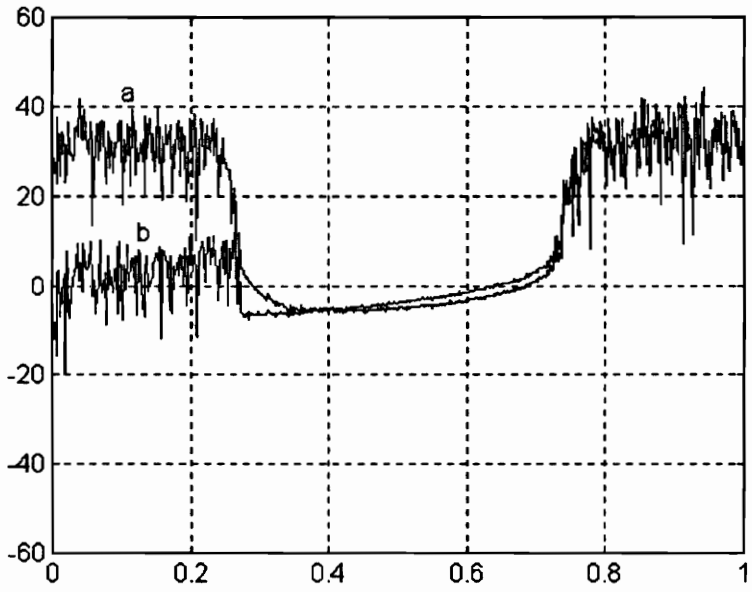


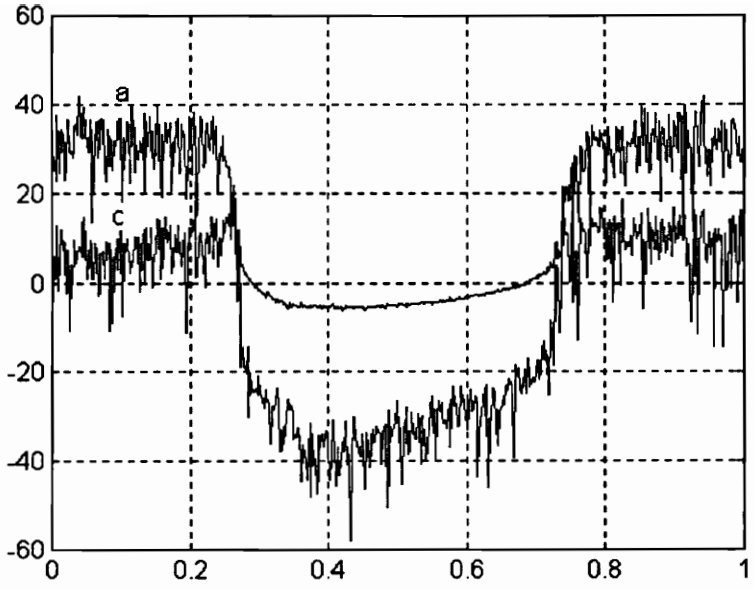
Fig. 6.6 Power Spectral Density of (a) Echo (b) Noise.

We use 1024th order NLMS adaptive filters for both AEC and ANC. Both noise cancellation approaches were adapted using the same set of 64,000 samples. While simulating the simultaneous echo and noise cancellation approach, we initially adapted only the echo canceler. Once the echo canceler converges, the residual error signal $e(n)$ is predominantly the ambient noise. We start adapting the noise canceler only after the echo canceler has converged to its steady state ERL of 30 dB. This helps to reduce the dependence of the echo and noise cancelers explained earlier.

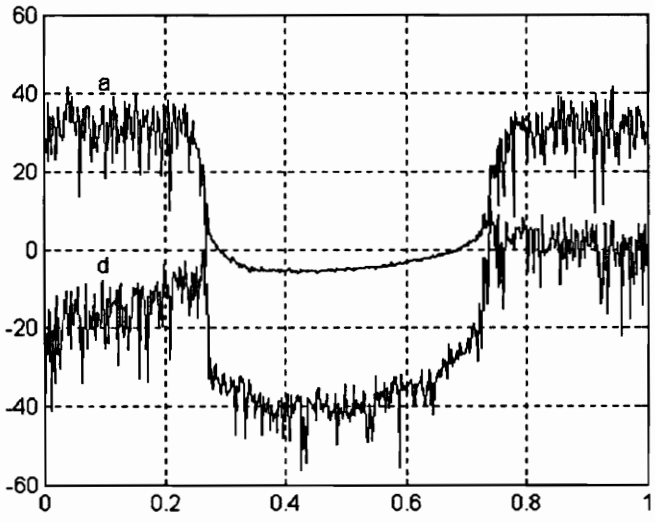
In Figure 6.7, we show the power spectral densities of the original echo and the residual signal $e(n)$ obtained with and without noise cancellation. We see that while the residual error in the procedure without noise cancellation has the same amount of noise as the original echo, the noise cancellation approaches have reduced the noise by at least 20 dB. This reduction in noise improves the perceived quality of the communication. We also see that the simultaneous echo and noise cancellation approach yields at least 10 dB less echo than the sequential cancellation approach. This corroborates the fact that we can achieve better ERL with simultaneous echo and noise cancellation than with their sequential cancellation.



(i)



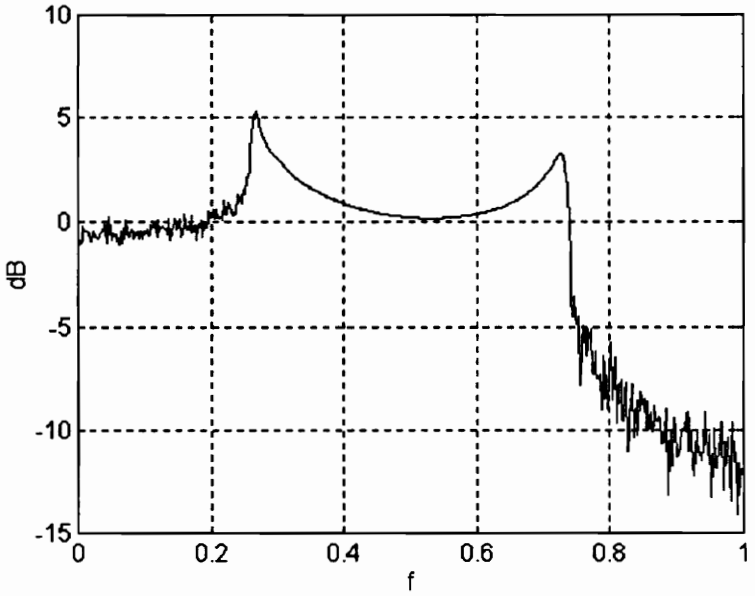
(ii)



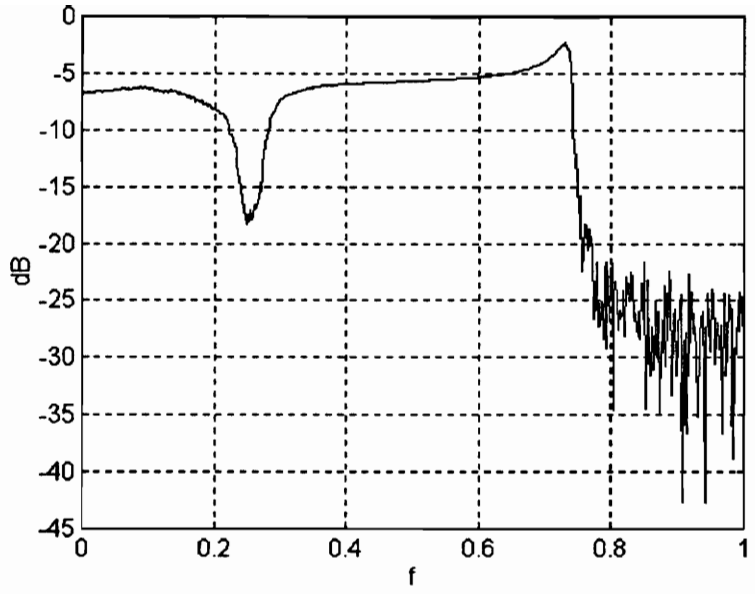
(iii)

Fig. 6.7 Power Spectral Density of
(a) Original Echo and Noise, and Residual Error (b) Without Noise Cancellation,
(c) Using the Echo Cancellation Followed by Noise Cancellation Approach
(d) Using the Simultaneous Echo and Noise Cancellation Approach.

Now we take a look at the distortion introduced by the noise canceler on the near-end speech. According to our assumption, the near-end speech also exists in the frequency range $(0, 0.25)$. The acoustic noise canceler, effectively, convolves the near-end speech with the sequence $\left[\delta(n) - \hat{h}(n)\right]$. Figure 6.8 shows this transfer function for the sequential cancellation and the simultaneous cancellation approaches. From Figure 6.8 we see that the simultaneous cancellation approach distorts the near-end speech, in the frequency range $(0, 0.25)$, more than the sequential cancellation approach. This is a disadvantage of the simultaneous cancellation approach over the sequential cancellation approach.



(a)



(b)

Fig. 6.8 Distortion Introduced by Noise Canceler $\left(1-\hat{H}(j\omega)\right)$ for (a) Sequential Cancellation (b) Simultaneous Cancellation

6.3 Finite Precision Effects

The implementation of the adaptive filtering algorithms on a fixed point digital computer invariably introduces quantization noise, due to the finite length of the registers used. The finite precision effect manifests itself in three different ways viz. signal quantization noise, change in frequency response due to the quantization of filter coefficients, and round-off noise (arithmetic quantization noise).

Figure 6.9 shows the nonlinear model of the quantization effects in the filtering and adaptation processes of the NLMS algorithm. Here $Q_B[.]$ is a B bit (1 sign and $(B-1)$ magnitude bits) round-off operator.

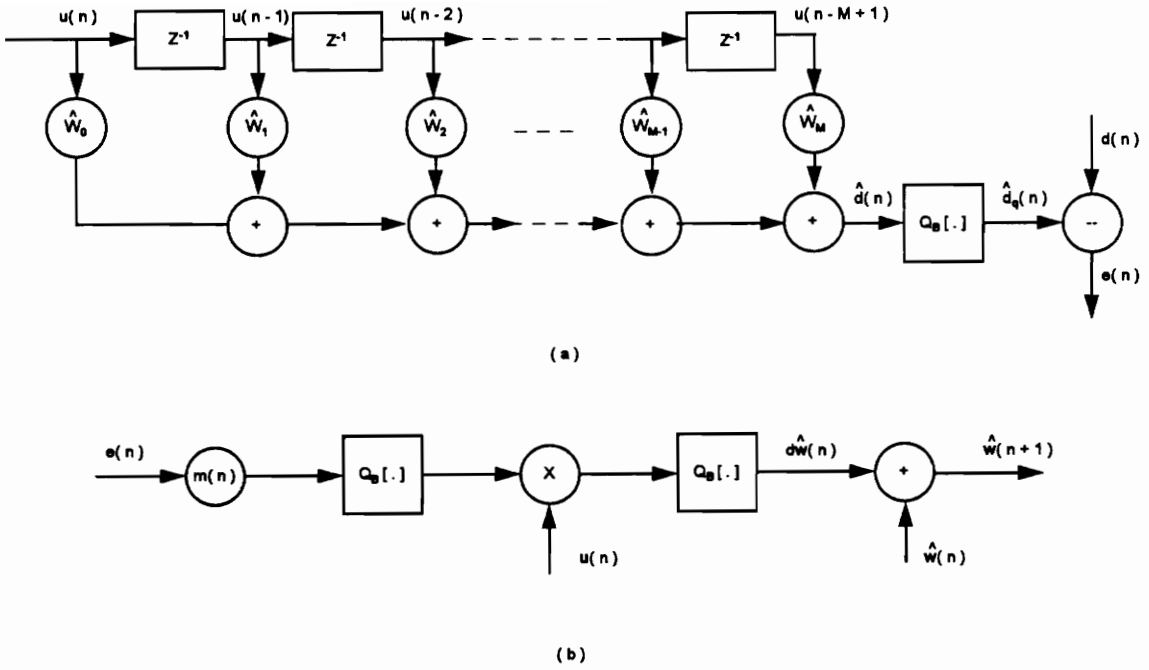


Fig. 6.9 Quantization Effects on the NLMS Adaptive Filter
(a) Filtering Process (b) Adaptation Process.

As illustrated in Figure 6.9(a), the filtering process is implemented as a multiply-accumulate (MAC) operation. The accumulator, usually, has $(2B + 8)$ bits accuracy. This double length accumulator eliminates the need to round-off the results from the individual multiplies. The accumulated result is, finally, rounded-off to B bits. Figure 6.9(b) shows the round-offs involved in the adaptation process. This can be written, mathematically, as

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + Q_B\left[Q_B\left[\mu(n)e(n)\right]\mathbf{u}(n)\right] \tag{6.15}$$

Figure 6.10(a) shows the results of simulation, with white noise input, for different word lengths. We observe that the performance of the echo canceler degrades as the

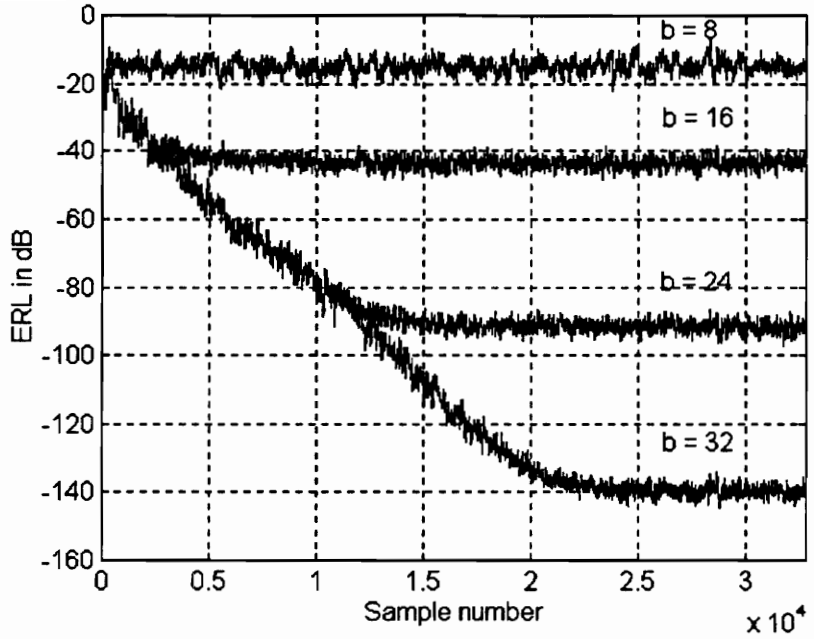
number of bits is reduced. This is mainly due to the *early stopping* of the adaptation. As the error decreases, the result of the product $\mu(n)e(n)$ drops below the Bth bit and the output of the quantizer goes to zero. This stops the adaptation.

The early stopping can be delayed, if not eliminated, by not quantizing the output of the first multiplier. This modifies (6.15) as shown below:

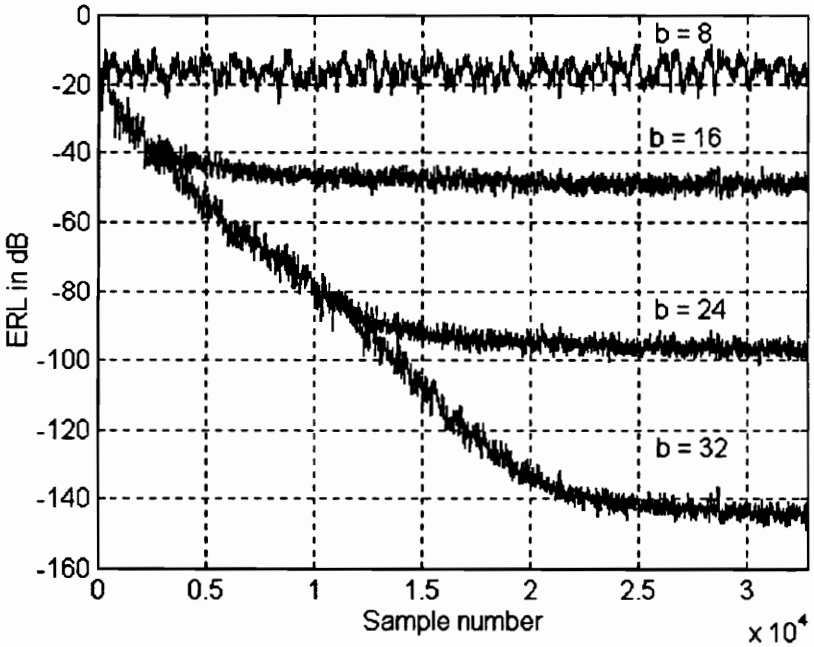
$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + Q_B\left[\mu(n)e(n)\mathbf{u}(n)\right] \tag{6.16}$$

The results of the simulation, with white noise input, for different word lengths is shown in Figure 6.10(b). We see that, for word lengths 16 and 24, the steady state ERL obtained with this modified approach is at least 6 dB better than that with the original approach. This improved performance is obtained at the cost of slightly increased complexity.

Our simulations indicate that the coefficient quantization does not have any significant influence on the performance of the adaptive filter.



(a)



(b)

Fig. 6.10 Variation of ERL for NLMS with White Noise as Input, at:
(a) Single Precision (b) Enhanced Precision.

6.4 Switching Adaptive Filter Structures

To obtain very fast initial convergence and high steady state ERL, judicious switching between the RLS and LMS algorithms has been suggested for network echo cancellation [15]. Even though this approach gives very good performance, the complexity of the RLS algorithm prohibits its implementation on present day DSPs, especially for acoustic echo cancellation in rooms with long reverberation time constants. Here we propose a new switching strategy which gives acceptable convergence rate and high steady state ERL, as well as reduced computational complexity.

As mentioned in Chapter 5, while the NLMS algorithm has a fast convergence rate compared to the subband NLMS algorithm, the complexity of NLMS is higher than for its subband counterpart. This motivates us to switch between NLMS and subband NLMS.

We, at first, use the NLMS algorithm to obtain fast initial convergence. However, the complexity of the NLMS algorithm prevents it from being used for high order adaptive filters as needed to mimic a long echo impulse response. Hence, the NLMS algorithm we use will have a lower order than that required. This results in a low steady state ERL from the NLMS adaptive filter by itself. We thus switch to the subband NLMS algorithm, after NLMS reaches its steady state, to obtain a better steady state ERL. The echo path estimated by the NLMS algorithm is zero-padded, to make it as long as the order of the subband adaptive NLMS filter, and used as the initial guess for the wideband filter of the subband NLMS algorithm. The wideband filter weights are transformed into subband filter weights by performing the inverse of the subband-to-wideband transformation described in Section 5.1. That is, the Fourier transforms of the subband filter weights are

obtained by appropriate grouping (unstacking) of the Fourier transform of the wideband filter weights. The Fourier transforms of the subband filter weights are inverse transformed to obtain subband filter weights. The low complexity of subband NLMS allows implementation of high order adaptive filters. This high order filter can capture the *tail portion of the impulse response* that was left unmodeled by the low order NLMS filter, thereby providing higher steady state ERL.

The divergence of the adaptive filter can also be used as an indicator for double-talk [15]. The adaptive filter can diverge either due to double-talk or due to a change in the echo path. In either case, we stop adapting the weights of the subband NLMS adaptive filter and use the frozen weights to remove the echo. Meanwhile, we switch to the NLMS algorithm and start adapting its weights. If the divergence is caused by a change in the echo path, NLMS will track the changes and reduce the echo. We, thus, exploit the fast tracking capability of the NLMS algorithm. Once the NLMS weights converge, we switch back to subband NLMS and continue. However, if the divergence is due to double-talk, NLMS continues to diverge. This can be used to detect double-talk.

We simulated the NLMS/SNLMS switching algorithm assuming that the maximum order of the NLMS adaptive filter that can be implemented is 256 and that of the subband NLMS adaptive filter is 1024. Figure 6.11 shows the learning curve of the 256th order NLMS algorithm with white noise as input. While NLMS(256) exhibits a fast convergence rate, the steady state ERL achieved is low. Figure 6.12 shows the learning curve of the 1024th order subband NLMS algorithm for the same input. We see that the convergence is too slow.

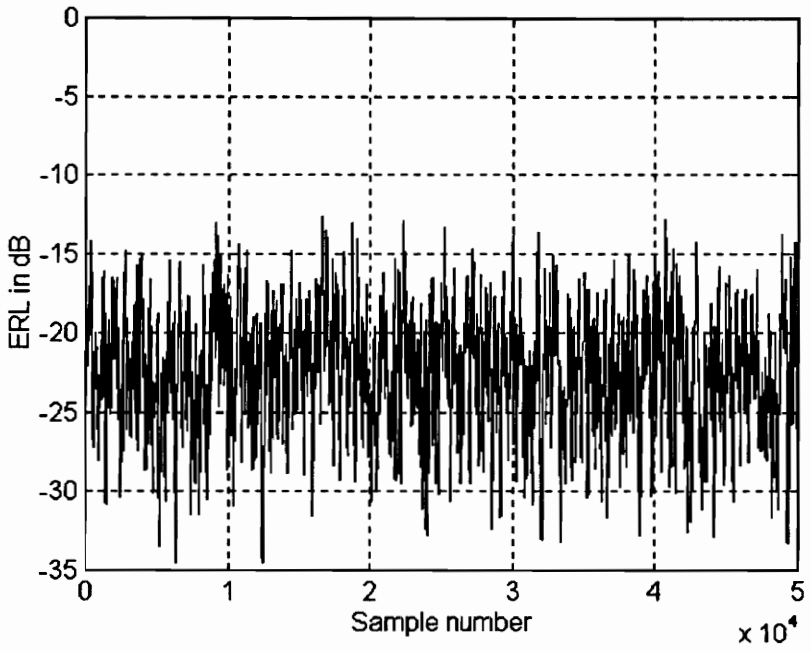


Fig. 6.11 Learning Curve of the NLMS Adaptive Filter of Order 256 with White Noise as Input.

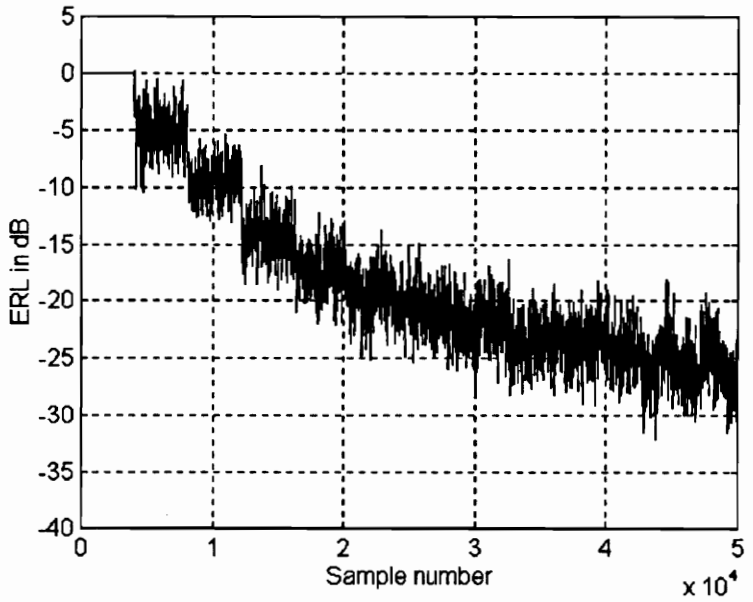


Fig. 6.12 Learning Curve of the Subband NLMS Adaptive Filter of Order 1024 with White Noise as Input.

Figure 6.13 shows the learning curve obtained from the switching of the algorithms for the same input. We switched from NLMS to subband NLMS at n = 4096. We see that NLMS(256)/SNLMS(1024) exhibits fast convergence as well as a steady state ERL of more than 50 dB.

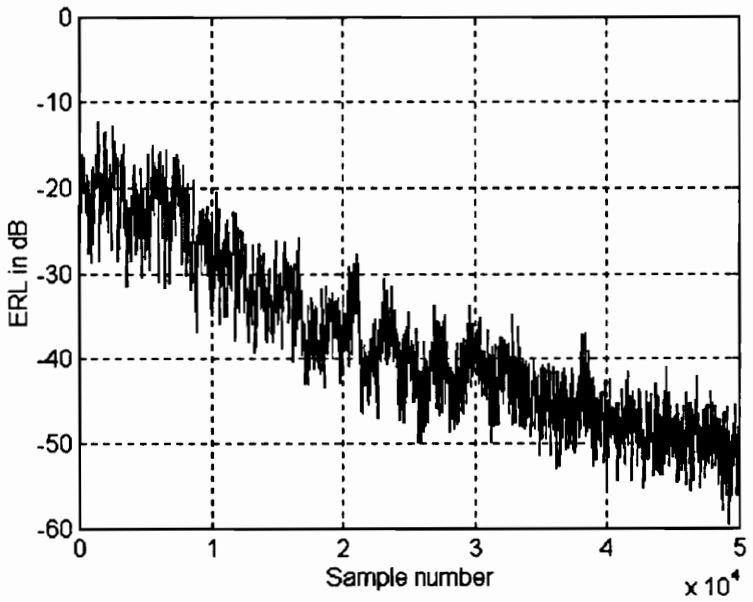


Fig. 6.13 Learning Curve with Switching Between NLMS(256) and Subband NLMS(1024) with White Noise as Input (switched at n = 4096).

6.5 Nonlinear Processor

In the discussions so far, we assumed that the echo path can be accurately modeled using a linear filter. However, in practice, the echo path has significant levels of nonlinear distortion due to quantization distortion from the codecs, companding, nonlinearities in the amplifier, etc. This often prevents echo cancelers from achieving the necessary echo return loss by using linear cancellation techniques alone. The nonlinear effects motivate

the use of a suitable nonlinear processor, cascaded with the adaptive filter, to obtain better performance.

An ideal nonlinear processor should not distort the near-end speech. However such a nonlinear processor is not practically realizable. Hence ITU-T recommends the disabling of the nonlinear processor under the double-talk and near-end single-talk conditions [16]. This suggests that the echo canceler must not rely excessively on the nonlinear processor and that the adaptive filter should be able to provide sufficient echo return loss to prevent objectionable echo under double-talk conditions.

One of the simplest nonlinear processors is the center clipper. The transfer functions of two variants of center clipper are shown in Fig. 6.14. An appropriate clipping level can be chosen, depending on the echo return loss provided by the adaptive filter.

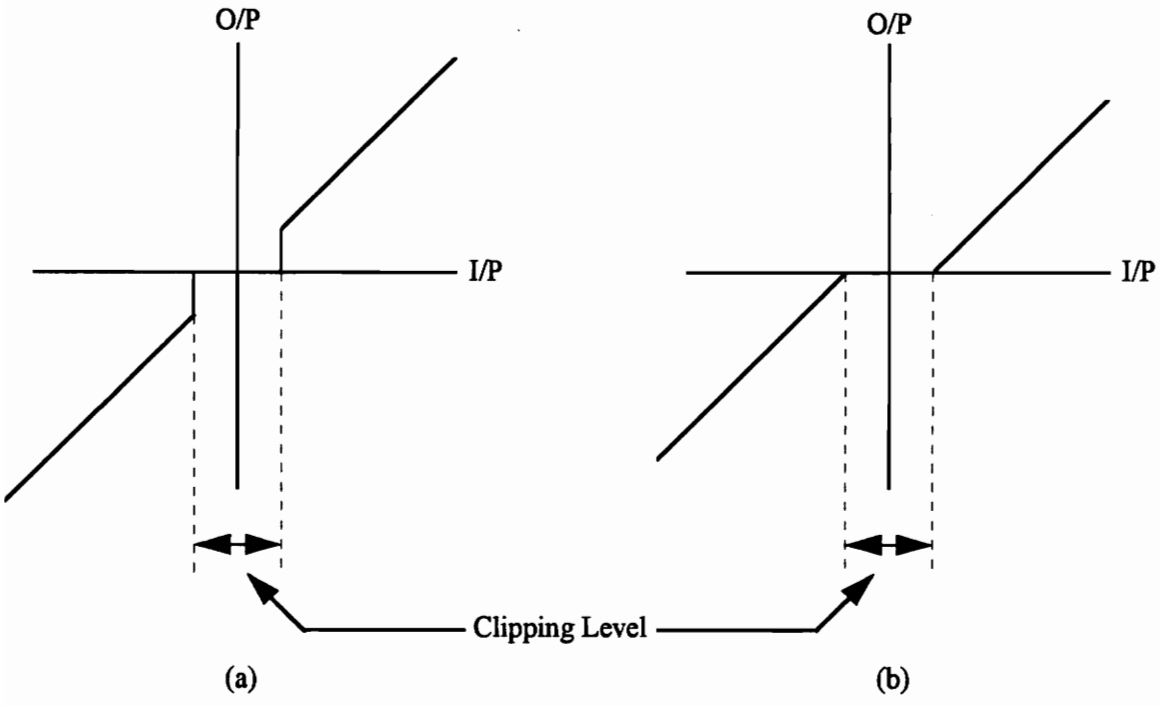


Fig. 6.14 Input/Output Characteristics of
(a) Discontinuous Center Clipper (b) Continuous Center Clipper.

Figure 6.15 shows the residual echo after nonlinear processing. Here we used clipping levels of 0.003, which is 30 dB below the original echo level. This level is sufficient to cancel the residual echo left by the adaptive filter. Clipping using the continuous center clipper is less annoying to the listener than using the discontinuous center clipper, possibly due to the less abrupt changes in the residual echo.

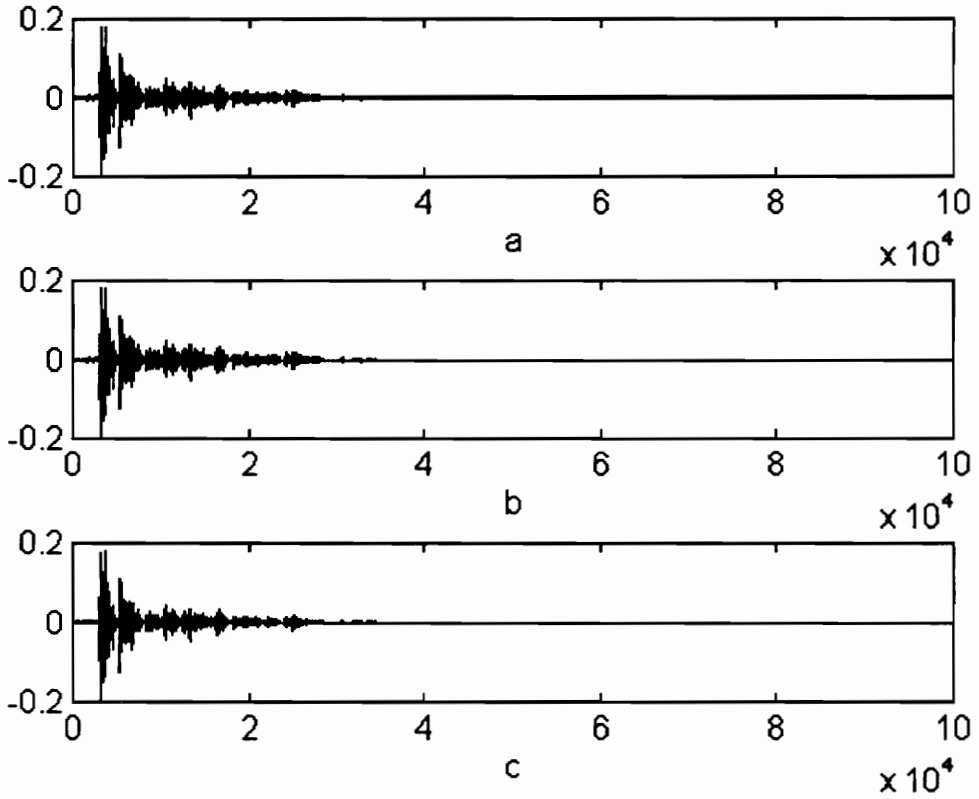


Fig. 6.15 Residual Echo (a) Before Nonlinear Processing, and After (b) Discontinuous and (c) Continuous Center Clipping.

6.6 Howling Detection and Control

The channel, including the acoustic and network echo paths, forms a closed loop. High volume level output at the loudspeaker can cause the loop to get into an oscillatory mode, usually referred to as howling. This is undesirable and this problem can be combated by detecting the howling and increasing the attenuation in the loop if howling is detected.

As shown later in this section, under the howling condition the near-end and received signals consist of a strong (unwanted) sinusoid, at the oscillation frequency, along with noise/speech. An adaptive second order IIR notch filter [29] is used to detect howling. Figure 6.16 shows the block diagram of the howling detector.

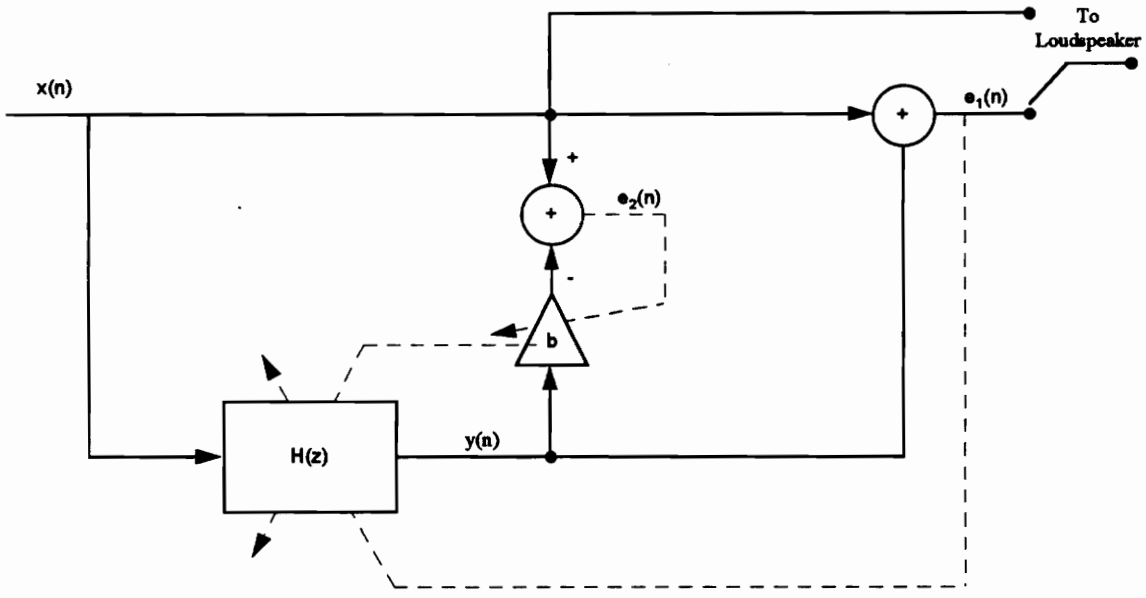


Fig. 6.16 Howling Detector.

The transfer function of the bandpass filter $H(z)$ shown in Figure 6.16 is

$$H(z) = \frac{\left[\frac{1-r^2}{1+r^2}\right] wz^{-1} - \left(1-r^2\right)z^{-2}}{1 - wz^{-1} + r^2 z^{-2}} \tag{6.17}$$

The pole radius r determines the bandwidth of the filter. As r approaches unity, the bandwidth becomes narrower. The coefficient w determines the center frequency of the bandpass filter and the frequency response equals unity at this frequency. The above filter is guaranteed to be stable if $0 < r < 1$ and $|w| < 2r$. Figures 6.17 and 6.18 show the magnitude response of this bandpass filter for different values of w and r.

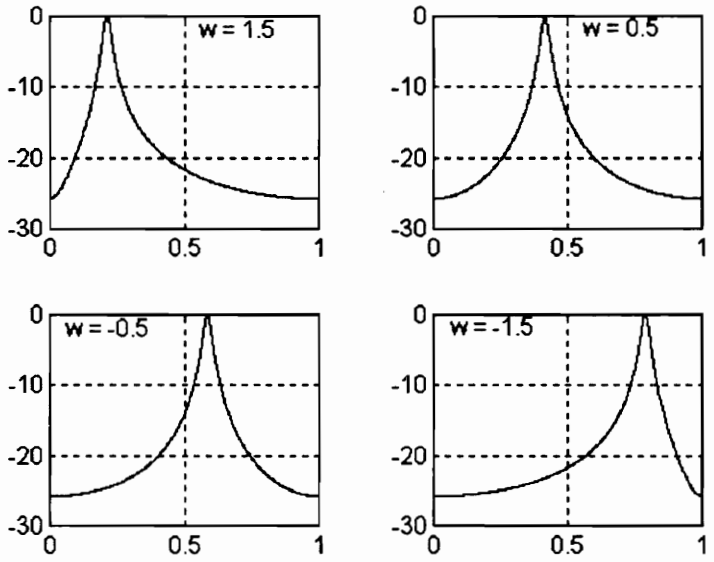


Fig. 6.17 Magnitude Response of $H(z)$ with r = 0.95 for Different Values of w.

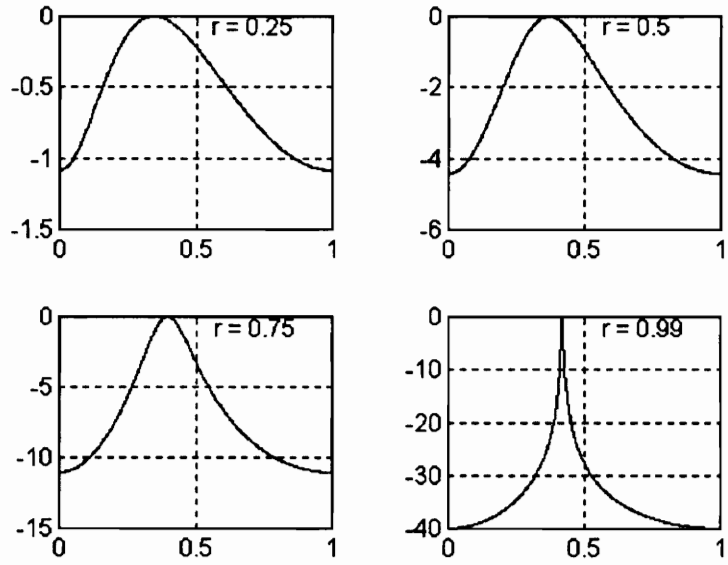


Fig. 6.18 Magnitude Response of $H(z)$ with $w = 0.5$ for Different Values of r.

The output of the adaptive filter $H(z)$ is given by

$$y(n) = \left[\frac{1-r^2(n)}{1+r^2(n)}\right] w(n)x(n-1) - \left[1-r^2(n)\right]x(n-2) + w(n)y(n-1) - r^2(n)y(n-2) \quad (6.18)$$

The value of $w(n)$ is chosen, so that the mean squared error $E\left[e_1^2(n)\right]$ is minimized, using the LMS algorithm. Thus, the update equation for $w(n)$ can be written as

$$w(n+1) = w(n) + \mu e_1(n)\alpha(n) \tag{6.19}$$

where μ is the step size and $\alpha(n)$ is the partial derivative of the estimated output $y(n)$ with respect to $w(n)$, which is given by the recursive equation

$$\alpha(n) = w(n)\alpha(n-1) - r^2(n)\alpha(n-2) + \left[\frac{1-r^2(n)}{1+r^2(n)}\right]x(n-1) + y(n-1) \tag{6.20}$$

The pole radius $r(n)$ is controlled by the detection parameter b. The parameter b is also adapted using LMS, as follows:

$$b(n+1) = b(n) + \nu e_2(n)y(n) \tag{6.21}$$

Here the step size is ν and the error $e_2(n)$ is defined according to

$$e_2(n) = x(n) - b(n)y(n) \tag{6.22}$$

A value of b close to unity indicates that the output of the bandpass filter almost equals the original signal. In that case, we reduce the bandwidth of the filter so that the bandpass

filter passes only the howling signal through it. The pole radius $r(n)$ is adapted as follows:

$$\text{If } b(n) \geq \beta \text{ and } r(n) \leq r_{max} \text{ then } r(n+1) = r(n) + \delta \tag{6.23}$$

$$\text{If } b(n) < \beta \text{ and } r(n) \geq r_{min} \text{ then } r(n+1) = r(n) - \delta \tag{6.24}$$

The threshold β, which controls the attack time of the howling detector, is selected based on the SNR. However, it has been found that a general value for β of 0.5 or less gives a satisfactory performance [29].

If $r(n)$ exceeds 0.95, we decide that howling exists and increase the variable loss (reduce the volume of the loudspeaker output) in the receive path.

The results from the simulation of the howling detector are shown in Figures 6.19 - 6.22. For this simulation, we use the signal obtained from sampling the far-end signal of a howling telephone.

Figure 6.19 shows the far-end signal under the howling condition. From the power spectral density of this signal shown in Figure 6.20, it is evident that the far-end signal consists of a strong sinusoid at approximately 2550 Hz. Note that in the howling detector error output the sinusoid at the howling frequency is attenuated, by about 20 dB, within 1000 samples. Figure 6.19 (c) shows the variation of the estimated pole radius with respect to time. We see that the pole moves closer to the unit circle when howling exists.

We also see that the threshold radius of 0.9 is reached within 4000 samples. At this time the input to the loudspeaker is switched to the error output of the howling detector. Sometimes this might lead to oscillations of howling detector parameters. If such oscillations are detected, the attenuation of the variable gain amplifier in the receive path is increased to control howling.

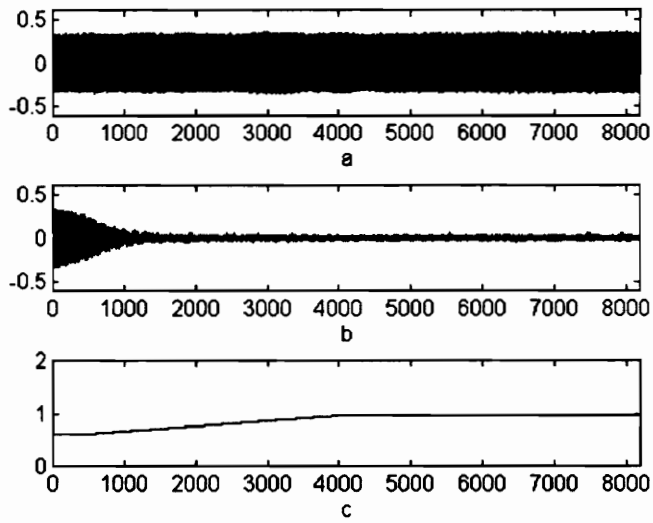


Fig. 6.19 (a) Far-end Signal Under Howling (b) Error Output of Howling Detector (c) Pole Radius.

Figure 6.21 shows the far-end signal under the howling condition along with speech, the error signal output of the howling detector, and the estimated pole radius. From the power spectral densities shown in Figure 6.22, we see that the howling now occurs at approximately 2750 Hz. The shift in the howling frequency, relative to the case shown in Figure 6.20, is due to the echo path change that occurred between the two measurements. Here also the estimated pole radius exceeds the threshold radius of 0.9 within 4000 samples.

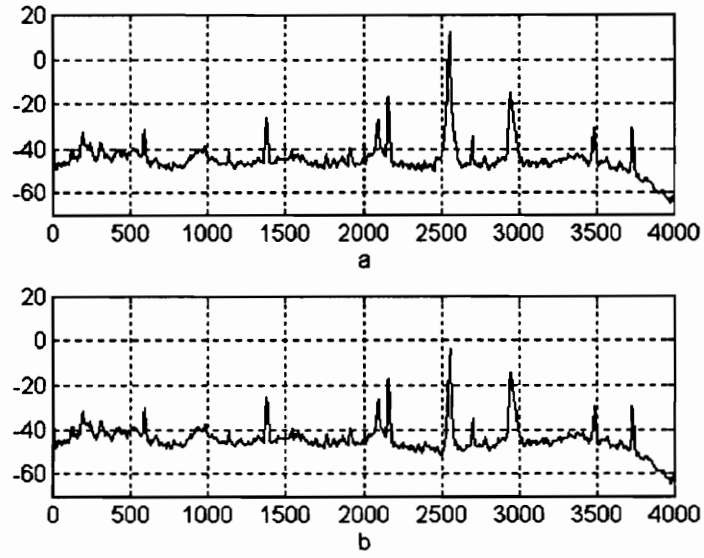


Fig. 6.20 Power Spectral Densities of
(a) Far-end Signal and (b) Error Signal from Howling Detector.

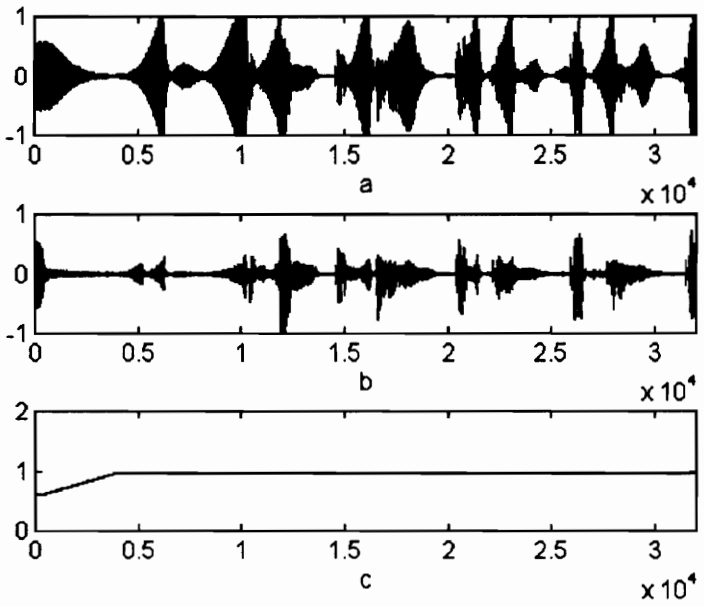


Fig. 6.21 (a) Far-end Signal Under Howling with Speech
(b) Error Signal Output of Howling Detector (c) Pole Radius.

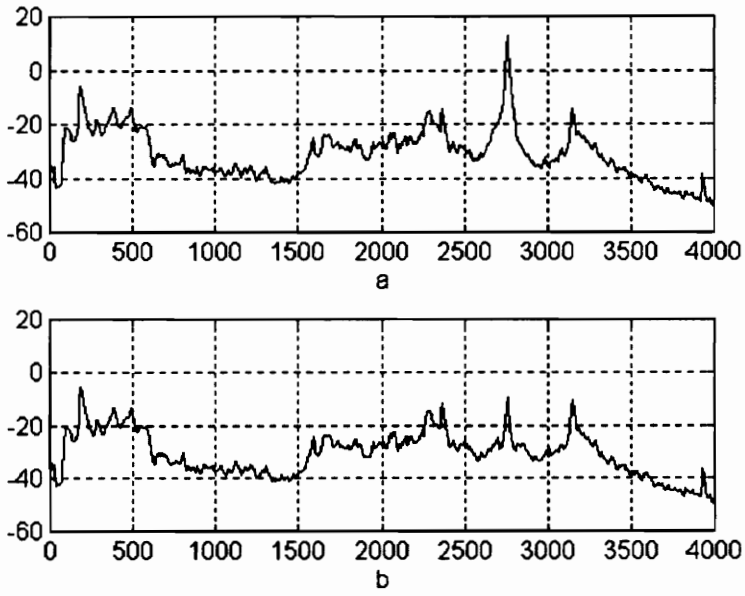


Fig. 6.22 Power Spectral Densities of
(a) Far-end Signal and (b) Error Signal from Howling Detector.

Thus the second order adaptive IIR notch filter can be used as the howling detector. The computational simplicity of this approach makes it very attractive for practical implementation.

7. Results

As mentioned in Chapter 1, echo arises in hands-free telephony due to impedance mismatch in the hybrid (network echo) and due to acoustic feedback from the loudspeaker to the microphone (acoustic echo). In this chapter, we present a configuration for the real time single-chip implementation of combined network and acoustic echo cancelers. Each of the functional modules in the proposed configuration is described. Strategies are proposed to interface the echo canceler with digital and analog telephones. Finally, we report the performance measures obtained for some modules from simulations and real time implementations on the Texas Instruments floating point processor TMS320C31 and the Analog Devices fixed point processor ADSP 2181.

7.1 Echo Canceler Configuration

Figure 7.1 shows the proposed configuration for real time implementation of the echo canceler on a single Digital Signal Processor (DSP). Digital Signal Processors are programmable microcomputers optimized for digital signal processing operations.

The acoustic and network echo estimators form the heart of the configuration. The implementation of the different functional modules that constitute the echo canceler are discussed in the following section.

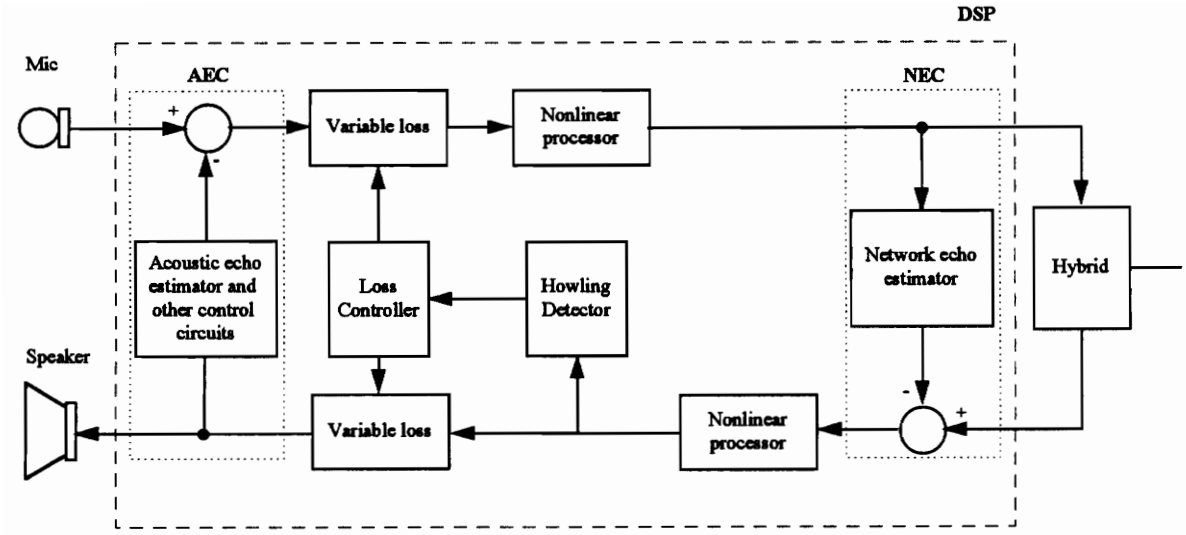


Fig. 7.1 Echo Canceler Configuration.

7.2 Functional Modules

7.2.1 Acoustic Echo Estimator

The acoustic echo estimator consists of an adaptive filter. It could use any adaptive filtering algorithm, in particular one of those described in Chapters 2 - 6. The adaptive filter converges to the impulse response of the acoustic echo path and produces an estimate of the acoustic echo. The estimated echo is subtracted from the microphone signal and the residual signal is transmitted to the far-end. The order (length) of the adaptive filter needed here depends on the reverberation time constant of the room. Typical values for the order range from 1024 to 2048. While underestimating the order results in a low ERL due to modeling error, overestimating the order results in a low ERL due to estimation error. Hence, the selected order should neither be too low nor too high.

We recommend using the NLMS adaptive filter for the adaptive echo canceler. If the computational power of the DSP prohibits the implementation of the NLMS filter of the required order, we suggest switching the adaptive filtering algorithms between low order NLMS and required order subband NLMS, as explained in Section 6.4.

We also recommend storing the echo impulse response found during any use of the telephone and using the stored weights as the initial guess during the next use, to achieve faster convergence in most cases.

7.2.2 Network Echo Estimator

The adaptive filter used for the network echo estimation converges to the impulse response of the echo generating hybrid and produces an estimate for the network echo. The estimated echo is subtracted from the received signal and the residual signal is fed to the speaker (as well as to the acoustic echo canceler). The network echo impulse response is usually short and it can be modeled using an adaptive filter of order between 128 and 256. Here also, neither overestimation nor underestimation of the order is desirable.

The low order required for the adaptive filter readily allows using the NLMS algorithm for network echo estimation. Since the network echo path does not vary with time, after the connection gets established, we propose inhibiting the adaptation of the weights once the adaptive filter has converged. Only the filtering (estimation) and echo removal (subtraction) processes are continued. This reduces the computational burden on the processor.

If the implementation is done on either a 16 bit or 24 bit fixed point processor, we suggest using the modified quantization procedure described in Section 6.3.

7.2.3 Control Circuit

The control circuit consists of the double-talk detector. The double-talk detector should detect double-talk fast enough to prevent large divergence. Whenever the double-talk condition is detected, adaptation of both the acoustic and network echo cancelers is inhibited.

We recommend using the double-talk detection statistic for double-talk detection. Since the double-talk detection statistic can be used as the near-end activity detector, as shown in Section 6.1, we use it to control the nonlinear processor as well.

7.2.4 Nonlinear Processor

The maximum echo return loss that can be achieved by the echo cancelers is limited due to the nonlinearities in the echo path, quantization errors, and modeling errors. The nonlinear processing block is intended to reduce the residual echo level further and this is done after both acoustic and network echo cancellation are operational. The AEC nonlinear processor should be disabled when near-end speech is present, since the near-end speech should not be distorted due to the nonlinear processing. The NEC nonlinear processor should be disabled when far-end speech is present, since the far-end speech should not be distorted due to the nonlinear processing.

We recommend using the continuous center clipper, discussed in Section 6.5, for nonlinear processing. The nonlinear processor is controlled on the basis of the DTDS. The clipping level is chosen based on the attenuation provided by the adaptive filters.

7.2.5 Variable Loss

This unit suppresses the echo by inserting variable losses on the received and/or transmitted audio signals. The variable loss device, apart from being a fail-safe echo suppressor, helps in starting up the echo canceler, as will be explained in Section 7.3. The attenuation level is varied depending on the energies of the loudspeaker and microphone signals. The attenuations in the send and receive paths, corresponding to different energy levels, are shown in Figure 7.2. The variable losses are disabled after the echo canceling filters converge.

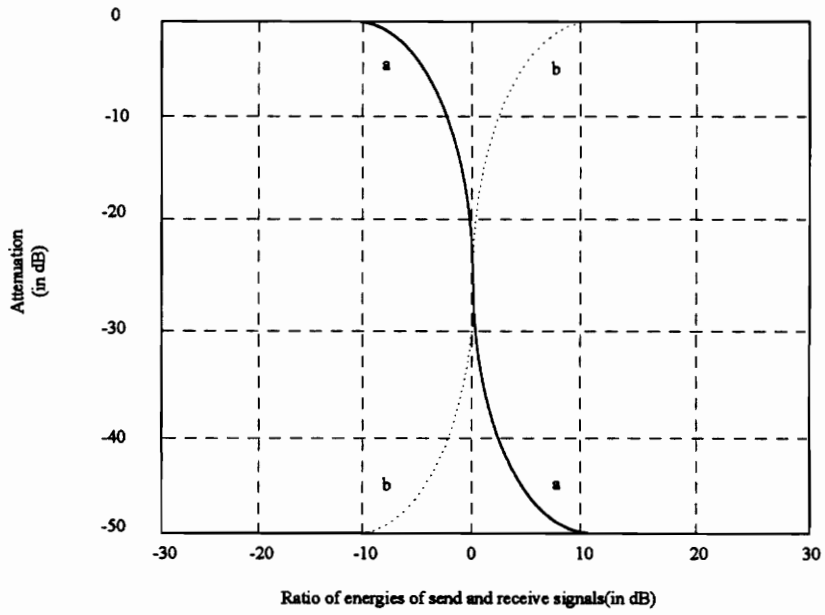


Fig. 7.2 Variable Loss in (a) Receive Path (b) Send Path.

7.2.6 Howling Detector

When the volume level output at the loudspeaker is high, the closed loop formed by the channel, including the acoustic and network echo paths can oscillate. These oscillations, usually referred to as howling, are detected using a howling detector.

The howling detector uses an adaptive second order IIR notch filter. The operation of the howling detector is explained in Section 6.6. If howling is detected, the input to the loudspeaker is switched to the output of the notch filter, thereby suppressing only the oscillation. Sometimes this might lead to oscillation of the howling detector parameter itself. In that case, the attenuation in the receive path would be increased and the input to the loudspeaker switched back to the received signal. This would reduce the loop gain and hence suppress the oscillations (howling), together with the rest of the signal.

7.3 Start-up Procedure

The acoustic feedback path along with the network echo path forms a closed loop. Since the echo cancelers do not provide sufficient echo cancellation initially, the loop gain might exceed unity and this results in howling. We aim to avoid howling by starting the echo canceler in the soft half-duplex mode using the variable loss devices. Meanwhile, we adapt the network echo canceler. Once the NEC converges, it provides some amount of attenuation in the loop. After the network echo canceler converges, we start adapting the acoustic echo canceler. Once both echo cancelers converge and reach their steady states, the loop gain can never exceed unity.

7.4 Interfacing

While the signals received from the microphone, and those fed into the loudspeaker, are analog, the received and transmitted signals are either analog or digital, depending on whether the telephone is analog or digital. The echo canceler performs all the processing in digital format and produces residual error signals in digital format. Here we present approaches to interconnect these signals, with minimum additional circuitry. We use the serial port of the echo canceling DSP to receive and transmit data, due to its simplicity in interfacing. CODECs with serial digital interface are used for analog-to-digital and digital-to-analog data conversion.

7.4.1 Digital Telephone

In digital telephony, the signals transmitted to and received from the far-end are usually in *M*-ary phase shift keying (MPSK) format [24]. A universal digital line transceiver (UDLT) is used to perform both the modulation and retrieval of the pulse code modulated (PCM) data. The PCM data are, sometimes, $\mu-$ or $A-$ law companded. The master UDLT at the telephone exchange transmits the timing and framing information along with the data, which is retrieved by the slave UDLT in the telephones. The retrieved data, along with the timing and framing information, provides a seamless interface between the UDLT and DSP, through the serial port (SPORT0) of the DSP, as shown in Figure 7.3.

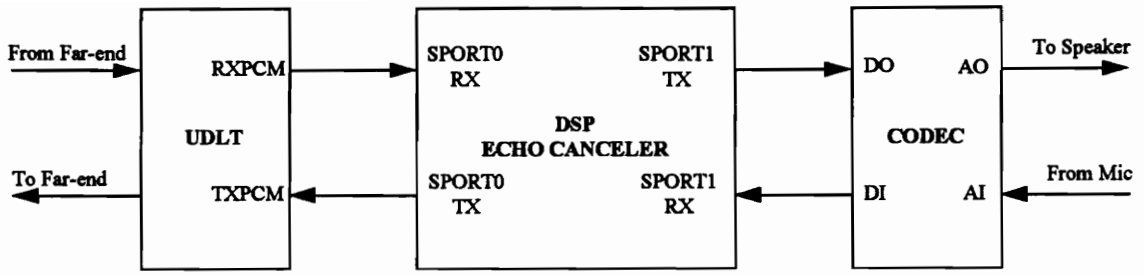


Fig. 7.3 Interfacing for Digital Telephones.

The signal from the microphone is digitized using a CODEC and fed into the DSP through the second serial port (SPORT1). The acoustic echo in the microphone signal is canceled and the residual is transmitted to the far-end. Similarly, the network echo in the signal received from the far-end is subtracted and the residual is fed to the loudspeaker, after digital-to-analog conversion by the CODEC.

Even though this approach necessitates the use of a DSP with at least two serial ports, we need to add only the DSP to the telephone in order to perform echo cancellation, since the UDLT and CODEC are already present in digital telephones.

7.4.2 Analog Telephone

The signals transmitted to and received from the far-end are analog in an analog telephone. Here we use a stereo CODEC for the A/D and D/A conversion. This allows us to receive and transmit two analog signals. The stereo CODEC has a serial digital interface where the left and right channel data are time division multiplexed. Figure 7.4 shows the approach used for interfacing.

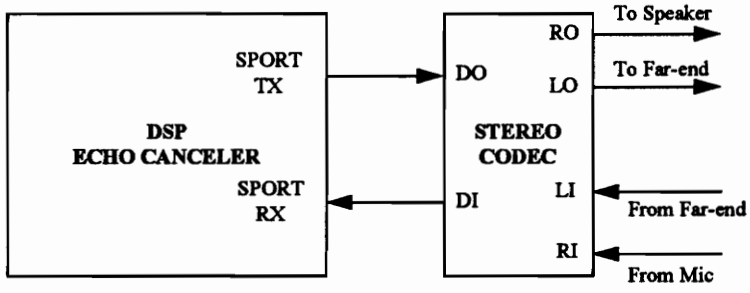


Fig. 7.4 Interfacing for Analog Telephones.

The analog signals from the far-end and microphone are fed into the left and right channels of the stereo CODEC, where they are digitized. The digitized signals are fed into the DSP. The echo canceler processes these signals and the echo canceled outputs are fed back to the CODEC. The analog outputs of the CODEC are transmitted to the loudspeaker and to the far-end.

This approach uses a DSP with at least one serial port and a stereo CODEC to interface with the analog telephone.

7.5 Simulation Results

The proposed NLMS echo estimation algorithm is simulated in Matlab and its performance measured according to the tests listed in ITU-T : G.167 [17].

7.5.1. Measurement Conditions

The echo path was simulated using the finite impulse response filter discussed in Chapter 1. White noise, and the real speech signal shown in Figure 1.6, were used to test the echo canceler. A 1024th order NLMS adaptive filter provided the echo estimate. The

clipping level was set at 0.003, which corresponds to 30 dB below the original echo level. The time variation of the echo path is modeled by linearly changing the echo path impulse response from one all pass filter to another, as shown in (7.1).

$$\mathbf{w}(n) = \alpha(n)\mathbf{w}_1 + (1-\alpha(n))\mathbf{w}_2 \tag{7.1}$$

Here $\mathbf{w}_1$ is the impulse response shown in Figure 1.5 and $\mathbf{w}_2$ is the impulse response of the new all pass system with poles at $0.995e^{\pm\pi/5}$ and $0.996e^{\pm 3\pi/5}$, and zeros at $1.005e^{\pm\pi/5}$ and $1.004e^{\pm 3\pi/5}$. $\alpha(n)$ is a linear mapping from $[N_1, N_2]$ to $[0,1]$ with $\alpha(0) = N_1$ and $\alpha(1) = N_2$ (the echo path variation starts at N_1 and ends at N_2). For these tests, we assume that the network echo and its canceler are absent.

7.5.2 Total Echo Return Loss - Single Talk (TERLwst)

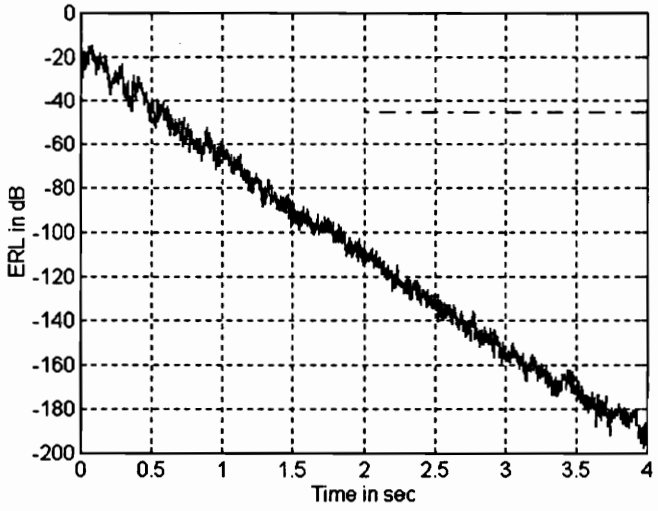
Test Procedure

All the AEC functional units are initially reset. A signal is applied at the far-end for a sufficient time so that the different functional units reach their steady states. No other (speech) signal than the acoustic return from the loudspeaker is applied to the microphone. TERLwst is the difference between the echo levels before and after the enabling of the AEC.

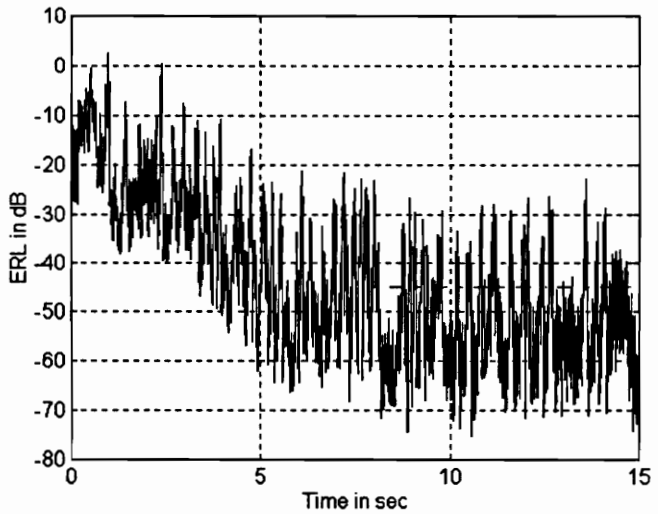
Result

While the G.167 required minimum value of TERLwst is 45 dB, the simulation yields at least 180 dB, for white noise input, without the nonlinear processor. The

simulation result is shown in Figure 7.5. If the nonlinear processor is enabled, its steady state output is 0. Under this condition, TERLwst is infinite. With speech as input, TERLwst obtained is about 40 dB.



(a)



(b)

Fig. 7.5 TERLwst Test Result with
(a) White Noise Input (b) Speech Input.

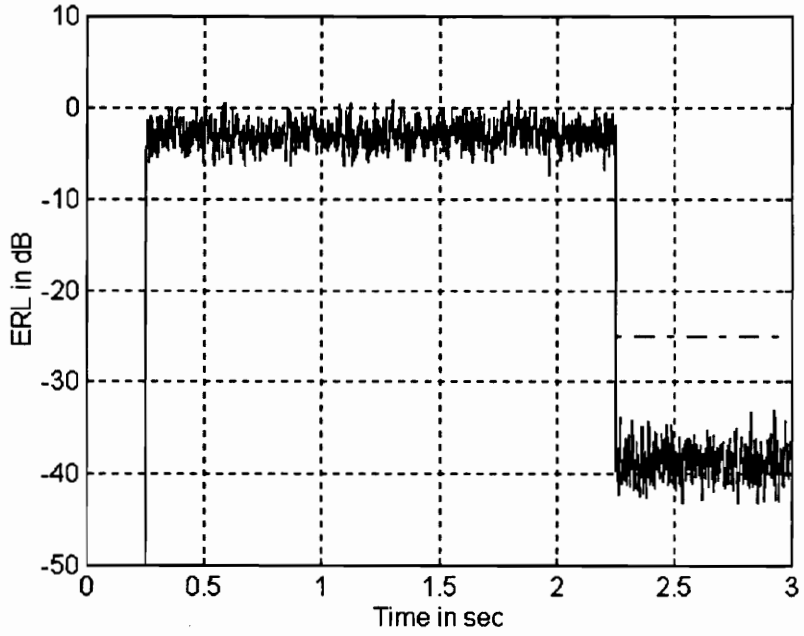
7.5.3 Total Echo Return Loss - Double Talk (TERLwdt)

Test Procedure

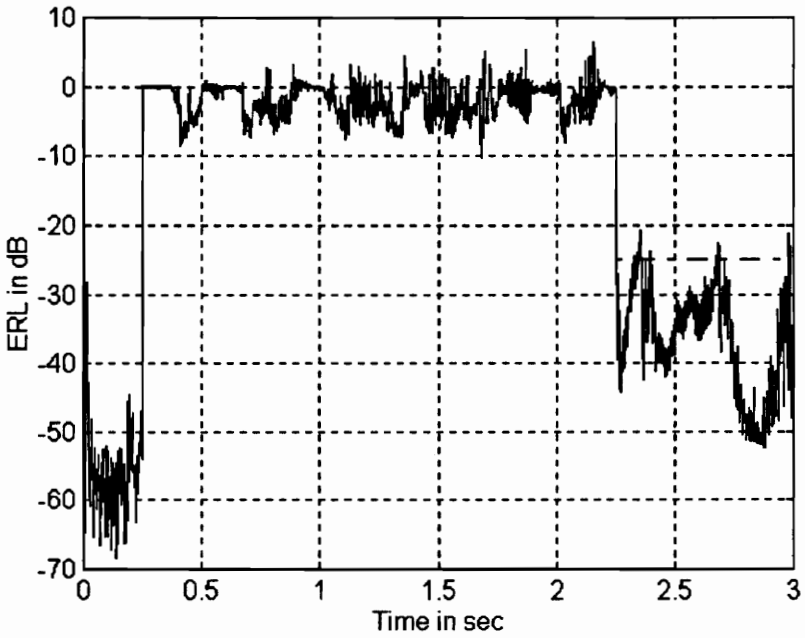
The AEC is first operated in the TERLwst test. After the steady state is reached, an acoustic signal simulating the near-end user's speech is applied at the microphone input for 2 seconds. The processing unit is then frozen, and the simulated near-end speech removed. TERLwdt is the difference between the level of the echo signal before the enabling of the AEC and now.

Result

The G.167 required value of TERLwdt is at least 25 dB. The simulation achieves 38 dB and about 30 dB, with white noise and speech respectively as input, as shown in Figure 7.6.



(a)



(b)

Fig. 7.6 TERLwdt Test Result with
(a) White Noise Input (b) Speech Input.
(Double-talk applied between t = 0.25 and t = 2.25
and adaptation stopped at t = 2.25.)

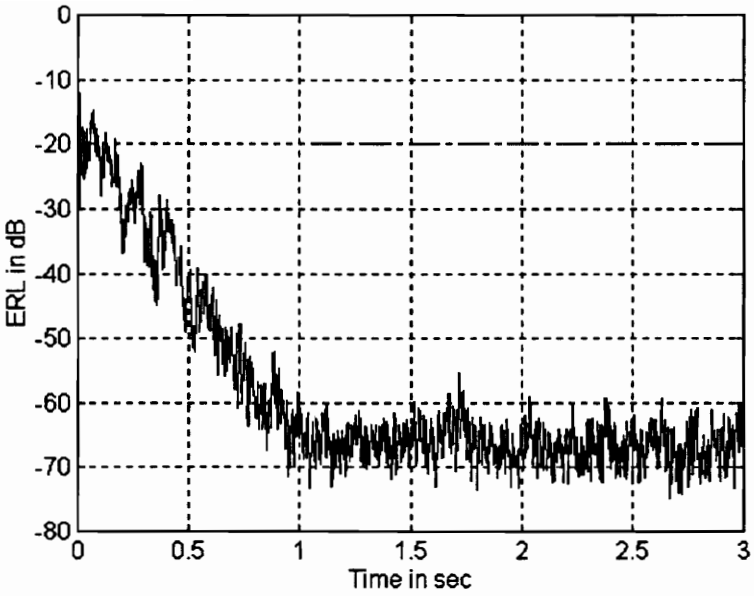
7.5.4 Initial Convergence Time (Tic)

Test Procedure

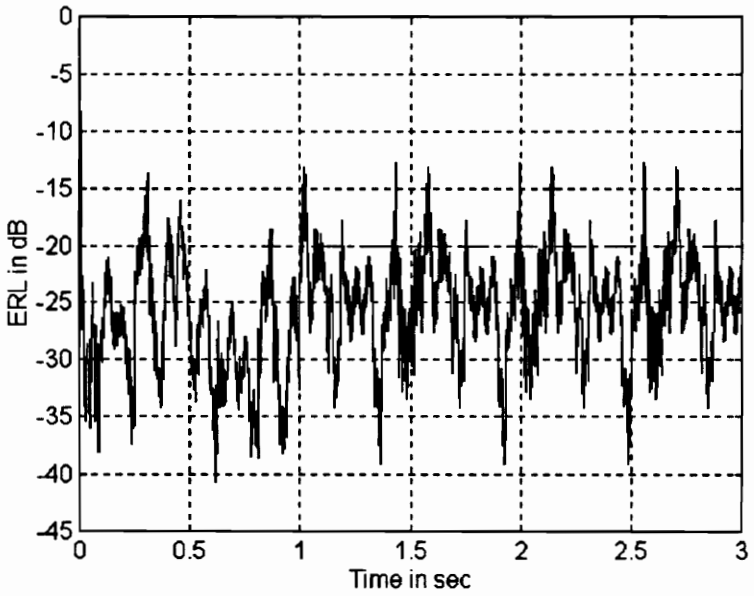
All the AEC functional units are initially reset and then enabled. A signal is applied at the far-end and a timer is started. The attenuation of the residual echo at the end of 1 second is measured.

Result

While the G.167 required attenuation is at least 20 dB, the simulation yields an attenuation of more than 60 dB and 20 dB with white noise and speech respectively as input, at the end of 1 second, as shown in Figure 7.7.



(a)



(b)

Fig. 7.7 Tic Test Result with
(a) White Noise Input (b) Speech Input.
(Adaptation stopped at t = 1.)

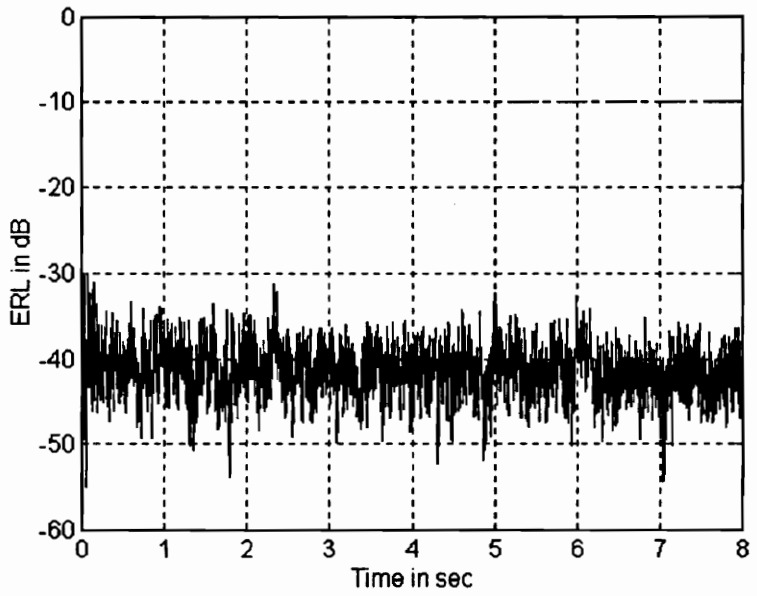
7.5.5 Echo Return Loss during Echo Path Variation (TERLwpv)

Test Procedure

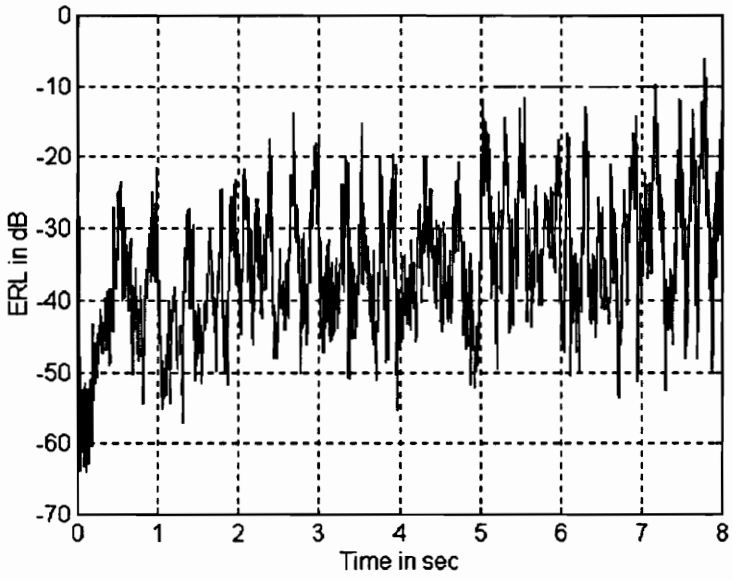
The AEC is initially operated as in the TERLwst test. After the steady state is reached, a simulated echo path variation is applied for 5 seconds. At the end of the echo path variations, the processing unit is frozen, and the residual echo level is measured. TERLwpv is the difference between the echo level before enabling the AEC and the measured value.

Result

While the G.167 required value of TERLwpv is 10 dB, the simulation yields about 37 dB and 20 dB, with speech and white noise respectively as input. The result is shown in Figure 7.8.



(a)



(b)

Fig. 7.8 TERLwpv Test Result with
(a) White Noise Input (b) Speech Input.
(Echo path variation applied between t = 0 and t = 5
and adaptation stopped at t = 5.)

7.5.6 Recovery Time after Echo Path Variation (Trpv)

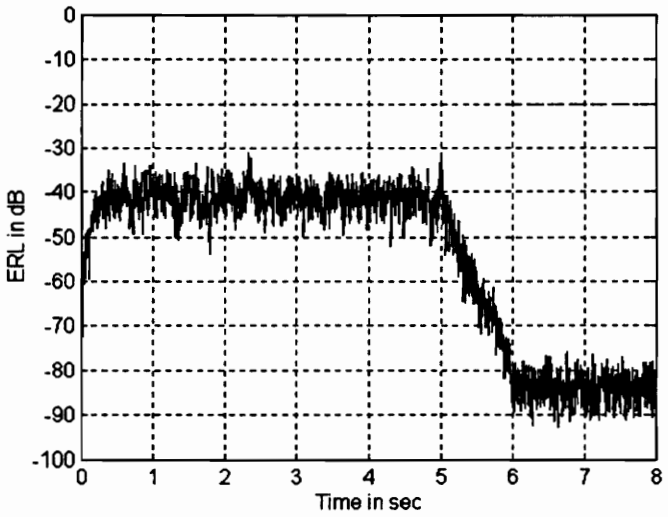
Test Procedure

The AEC is initially operated as in the TERLwst test. After the steady state is reached, a simulated echo path variation is applied for 5 seconds. At the end of the echo path variation a timer is started. After 1 second, the processing unit is frozen and the residual echo level is measured.

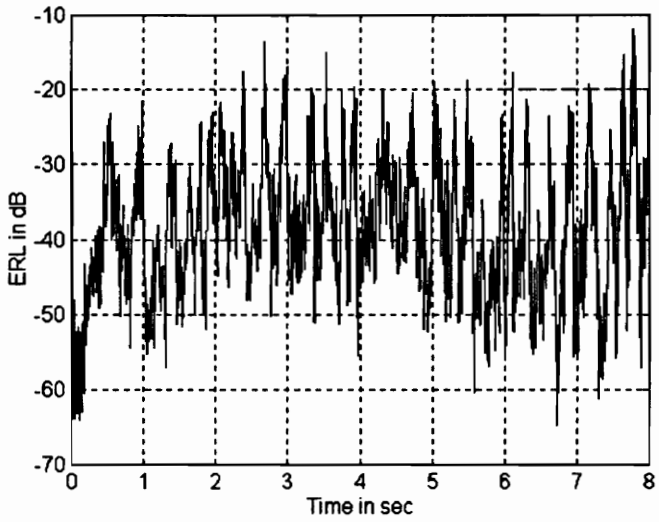
Result

While the required attenuation of the echo is at least 20 dB, the simulation yields more than 80 dB and about 30 dB, with white noise and speech respectively as input, at

the end of 5 seconds of path variation, followed by 1 second of adaptation, as shown in Figure 7.9.



(a)



(b)

Fig. 7.9 Trpv Test Result with
(a) White Noise Input (b) Speech Input
(Echo path variation applied between t = 0 and t = 5
and adaptation stopped at t = 6.)

7.6 Implementation Results

The echo estimator algorithms were implemented on the floating point processor TMS320C31 [25] and/or the 16 bit fixed point processor ADSP 2181 [26].

7.6.1 Floating Point Implementation

The adaptive filtering algorithms discussed in Chapters 2 - 5 were implemented on the TMS320C31. The TMS320C31 is a 32 bit, 16 MIPS floating point processor with one serial port. The ELF development platform [27] was used for the implementation. The analog interfacing approach was used to interface with the telephone. The speech signals were sampled at 8 kHz. The maximum order of the adaptive filters, that can be implemented on the TMS320C31 using different algorithms, is shown in Table 7.1.

Table 7.1 Maximum Order for Different Adaptation Algorithms.
(C31 Implementation)

Adaptation Algorithm	Maximum Order
LMS	300
NLMS	300
RLS	15
SNLMS*	1024

* 128 subbands were used

The echo canceler was tested using white noise as the far-end signal and a first order RC filter, with 2 kHz cut-off frequency, acting as the echo path. The ERL here was estimated as the ratio of the average amplitude of the original and residual echo signals, i.e. not taking into account specific spectral content. Estimated this way, LMS(256) and

NLMS(256) provide an ERL of at least 20 dB within 1 second. The steady state ERL is nearly 37 dB for both algorithms. The RLS(15) algorithm implementation converges to its steady state ERL of 37 dB within 1 second. SNLMS(1024) provides a steady state ERL of 37 dB after nearly 8 seconds. The power spectral densities of the original and residual echo signals from NLMS(256) are shown in Figure 7.10. From this figure, we see that the ERL is around 37 dB.

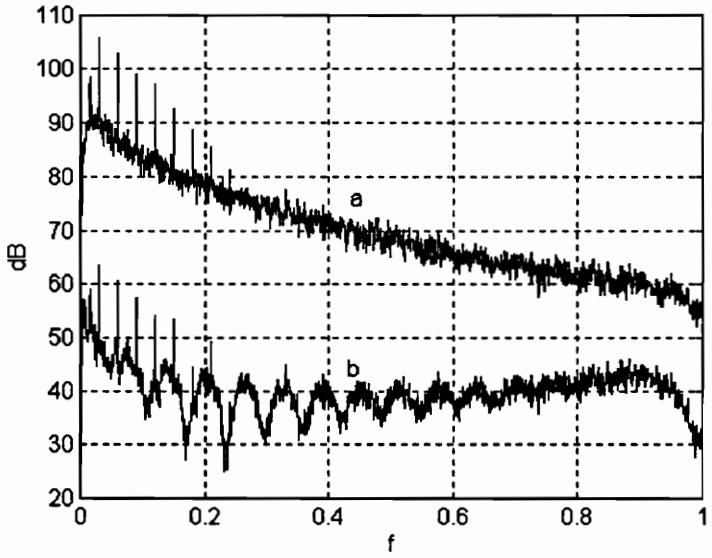


Fig. 7.10 Power Spectral Density of (a) Original Echo (b) Residual Echo from C31 Implementation of NLMS(256) with RC filter as Echo Path.

The echo canceler was also tested using real speech as the far-end signal and the actual room as echo path. While LMS(256) takes nearly 3 seconds to converge, NLMS(256) converges in approximately 1 second. Both algorithms provide a maximum ERL of about 15 dB. Figure 7.11 shows the original and residual echo signals obtained by using the LMS(256) adaptive filter. The corresponding time averaged (4096 point blocks

with 75% overlap) power spectral densities are shown in Figure 7.12. The original and residual echo signals obtained from the NLMS(256) adaptive filter are shown in Figure 7.13. Figure 7.14 shows the corresponding time averaged (4096 point blocks with 75% overlap) power spectral densities.

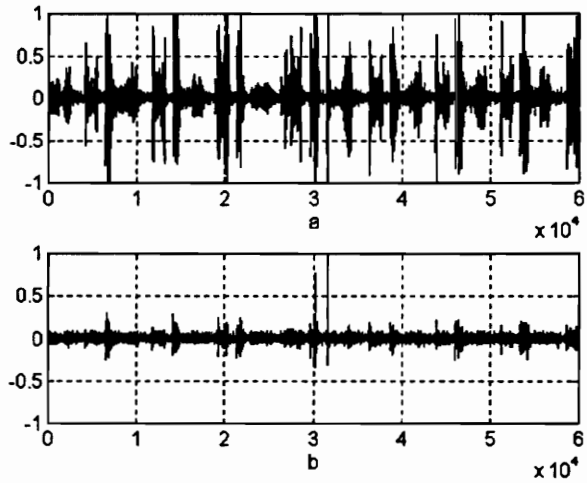


Fig. 7.11 (a) Original and (b) Residual Echo Signals
from C31 Implementation of LMS(256).

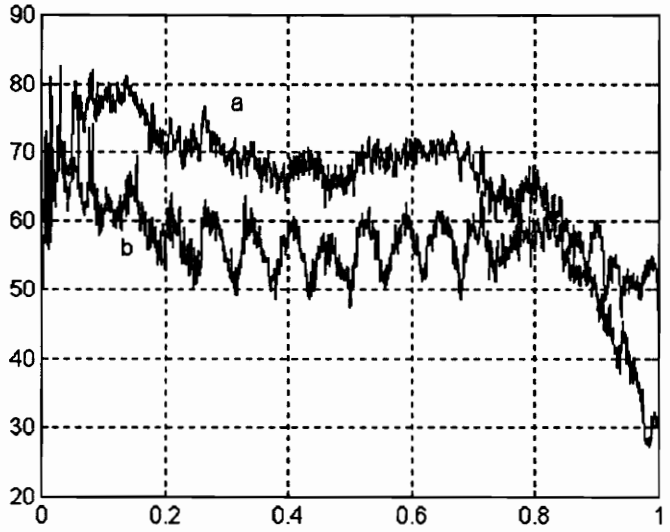


Fig. 7.12 Power Spectral Density of
(a) Original Echo (b) Residual Echo
from C31 Implementation of LMS(256).

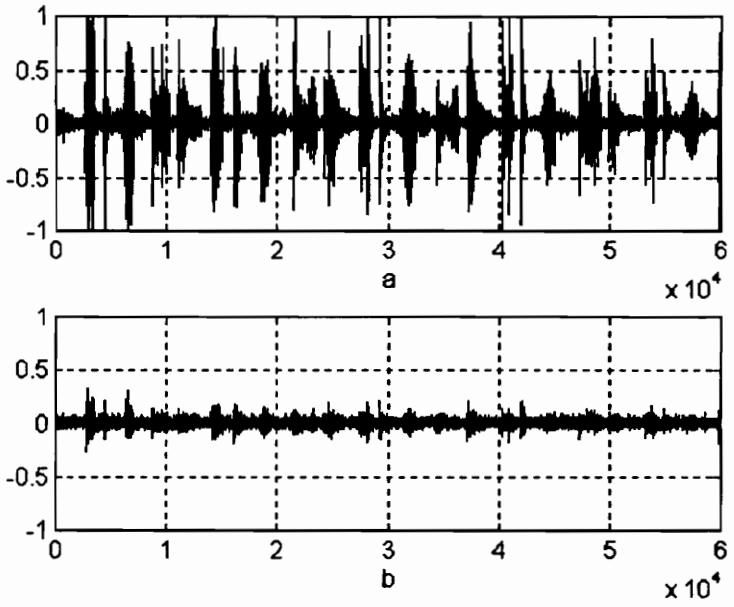


Fig. 7.13 (a) Original and (b) Residual Echo Signals from C31 Implementation of NLMS(256).

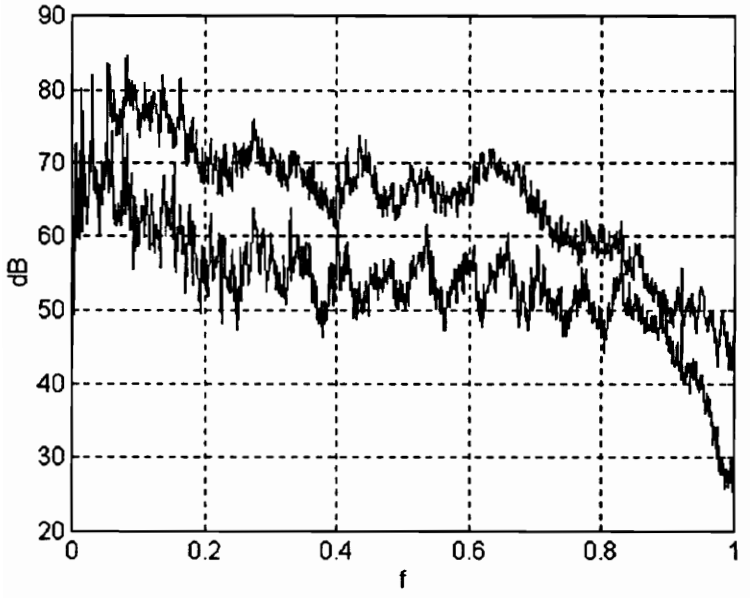


Fig. 7.14 Power Spectral Density of (a) Original Echo (b) Residual Echo from C31 Implementation of NLMS(256).

Figure 7.15 shows the result of the TERLwst test performed on the 256th order NLMS adaptive filter implemented on the TMS320C31. For this test, white noise was used as the far-end input. We see that the TERLwst, provided by the adaptive filter, is about 15 dB. Additional echo return loss can be achieved using the variable loss device and the nonlinear processor.

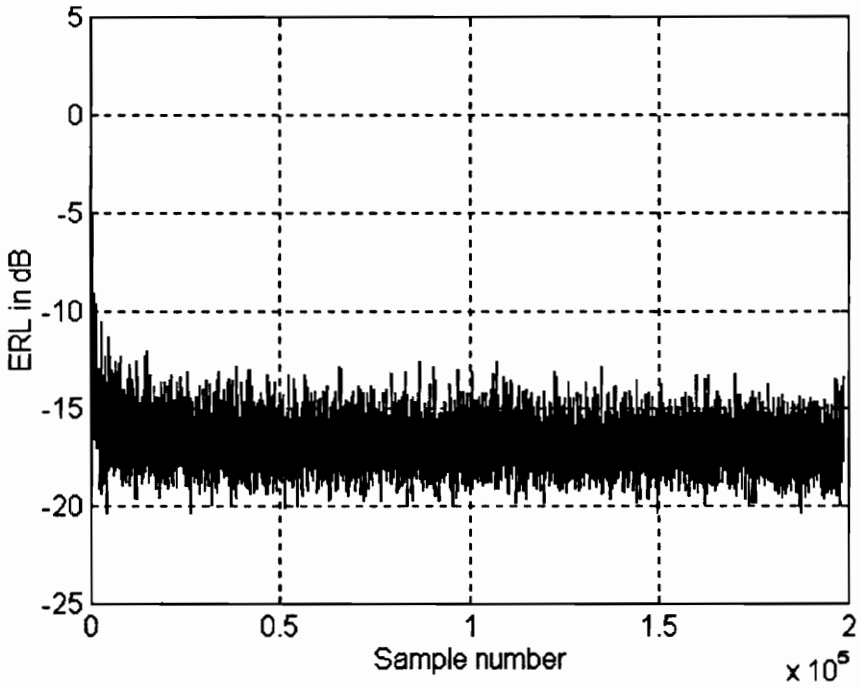


Fig. 7.15 Result of TERLwst Test on C31 Implementation of NLMS(256) Echo Canceling Algorithm.

7.6.2 Fixed Point Implementation

The NLMS adaptive filter was implemented on the ADSP-2181, which is a 16 bit, 32 MIPS fixed point processor with two serial ports. The EZ-KIT Lite development board [28] was used for this implementation. The digital interfacing was verified on this

processor. All the measurements were taken with analog interfacing. We used an 8 kHz sampling rate to sample the speech signals. The microphone was placed 2 cm away from the edge of the loudspeaker.

While this DSP allows implementation of the NLMS adaptive filters having orders up to 1024, we found that an order of 256 is sufficient to model the impulse response of the room used for testing. The adaptive filter converges within 1 second. The estimated impulse response of the room is shown in Figure 7.16. Figure 7.17 shows the original and residual echo signals, with real speech as the far-end input. The time averaged (4096 point blocks with 75% overlap) power spectral densities of the original and residual echo signals are shown in Figure 7.18. We see that the maximum echo attenuation is almost 20 dB.

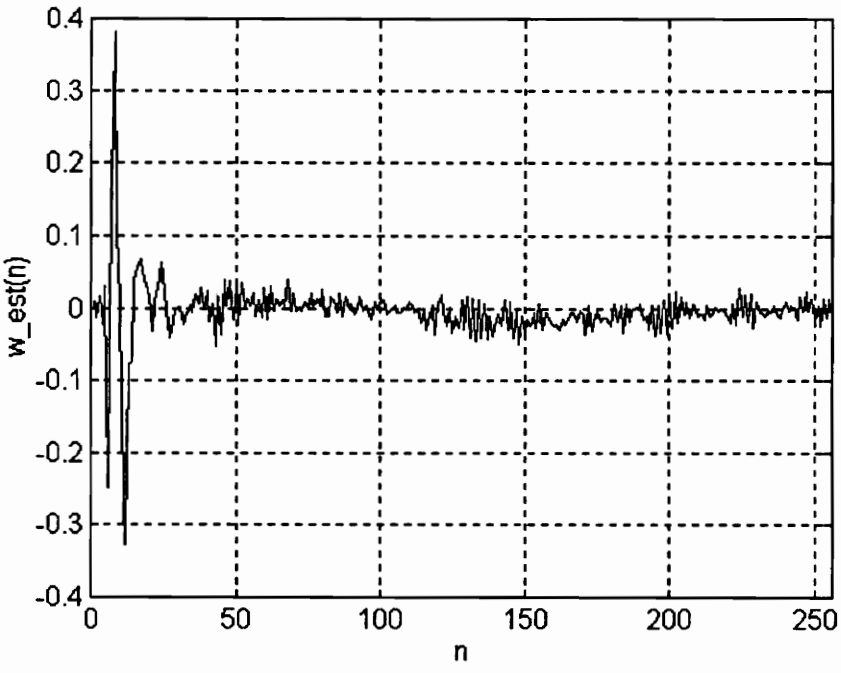


Fig. 7.16 Estimated Impulse Response of the Room from ADSP Implementation of NLMS(256).

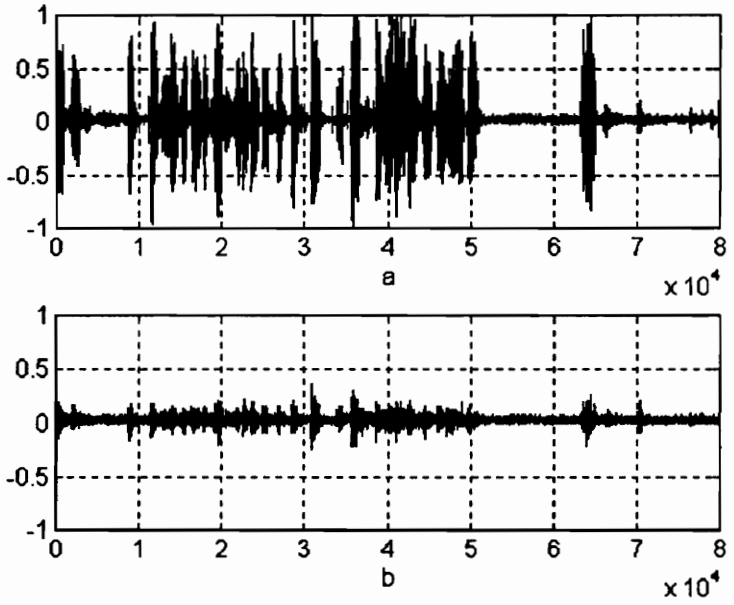


Fig. 7.17 (a) Original and (b) Residual Echo Signals from ADSP Implementation of NLMS(256).

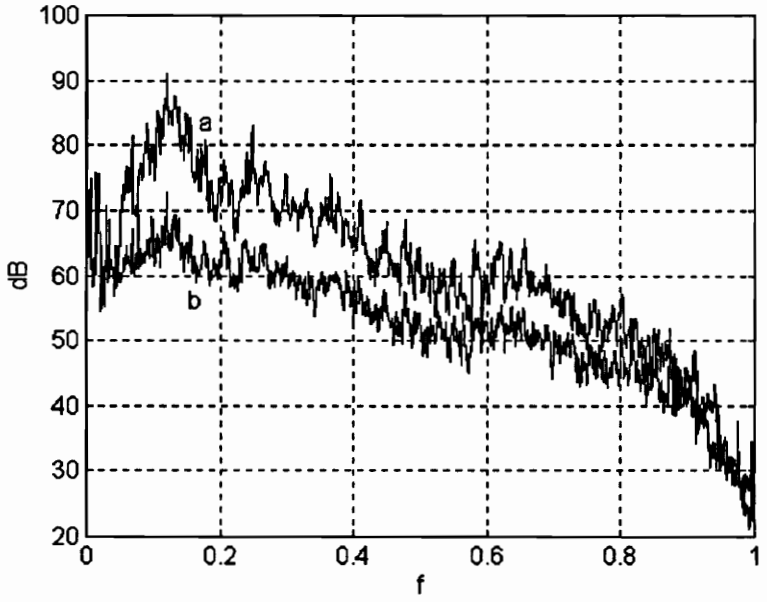


Fig. 7.18 Power Spectral Density of (a) Original Echo (b) Residual Echo from ADSP Implementation of NLMS(256).

Figure 7.19 shows the result of the TERLwst test performed on the 256th order NLMS adaptive filter implemented on the ADSP-2181. For this test, white noise was used as the far-end input. We see that TERLwst, provided by the adaptive filter, is about 15 dB. Additional echo return loss can be achieved using the variable loss device and the nonlinear processor.

Figure 7.20 shows the result of the initial convergence time test performed on the ADSP-2181 implementation. Here also, white noise was used as the far-end input. The adaptation was stopped at t = 1 sec. We see that the implementation yields an echo return loss of about 12 dB.

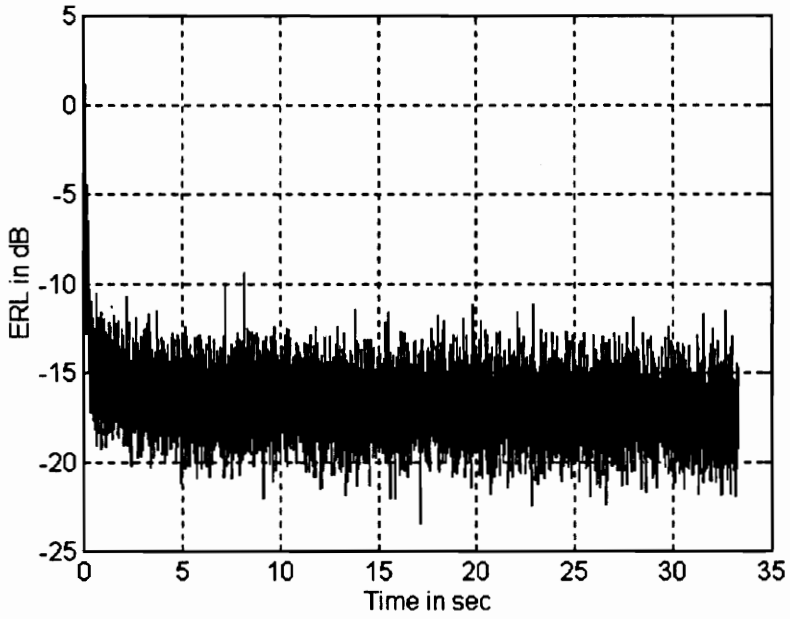


Fig. 7.19 TERLwst Test Result on ADSP Implementation of NLMS(256) with White Noise as Input.

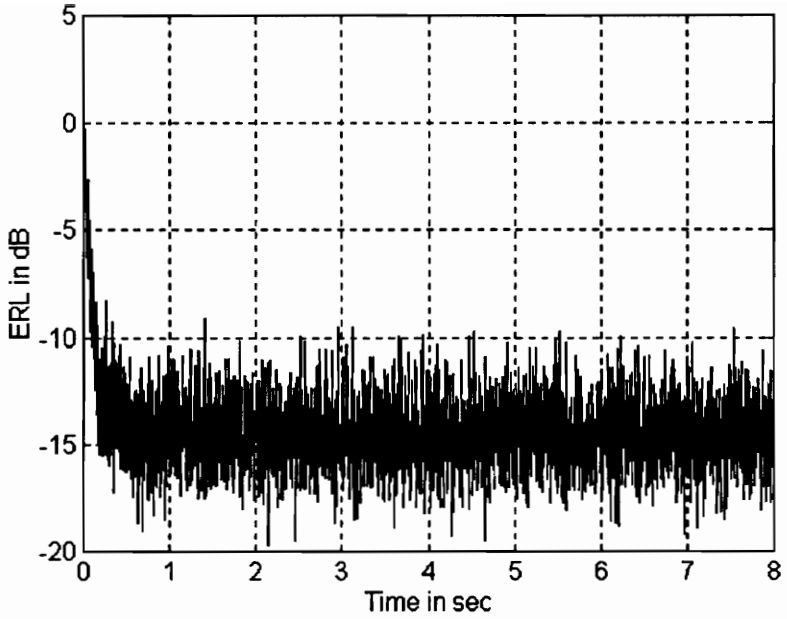


Fig. 7.20 Tic Test Result on ADSP Implementation of NLMS(256) with White Noise as Input (Adaptation stopped at t = 1).

7.6.3 Validation of Implementation Results

The echo path was replaced by a 256th order digital filter, having impulse response as shown in Figure 7.21 (a), internal to the DSP. This echo path is the same as one obtained in Figure 7.16, in order to have the spectral characteristics of the test room. With this digital echo path, phenomena due to the modeling of an analog echo path (room) are eliminated. White noise was used as the far-end input signal. The digital filter output was used as the near end echo signal. The impulse response estimated by the NLMS algorithm is shown in Figure 7.21 (b). The difference between the true and estimated impulse responses is shown in Figure 7.21 (c). We see that the estimate is very close to the actual impulse response. From Figure 7.22, we see that the echo return loss is more than 40 dB. Here the echo return loss is computed from the 4096 samples of the steady

state near-end and residual echo signals downloaded directly from the DSP's memory. This eliminates analog measurement error. The power spectral densities of the steady state original and residual echo signals are shown in Figure 7.23. The echo return loss is limited to 40 dB, which coincides with the fixed point simulation of quantization effects described in Section 6.3.

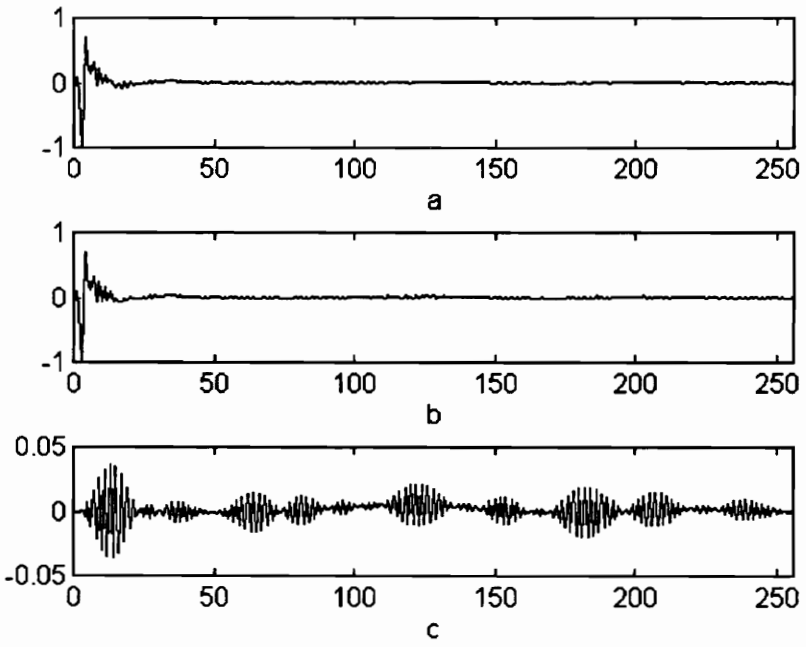


Fig. 7.21 Digital Echo Path Test on ADSP Implementation of NLMS(256)
(a) True Impulse response (b) Estimated Impulse Response (c) Estimation Error

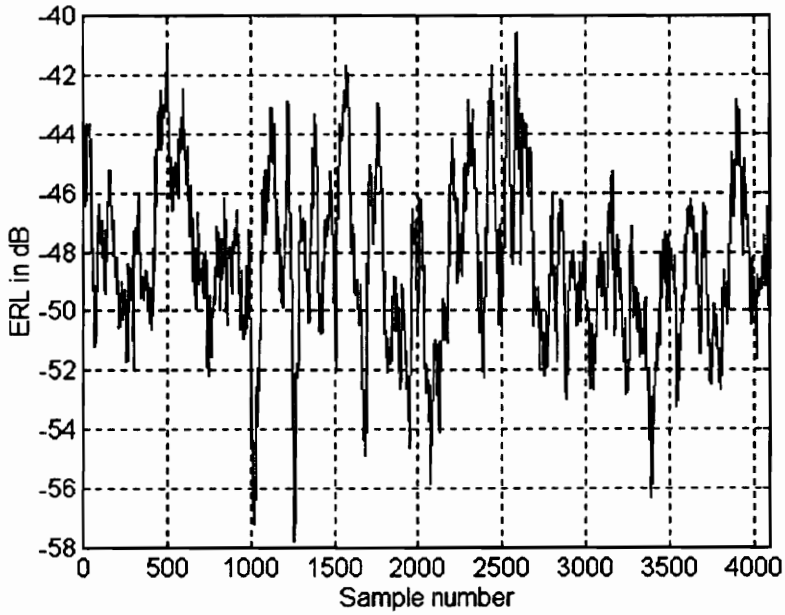


Fig. 7.22 Steady State Echo Return Loss with Digital Echo Path from ADSP Implementation of NLMS(256).

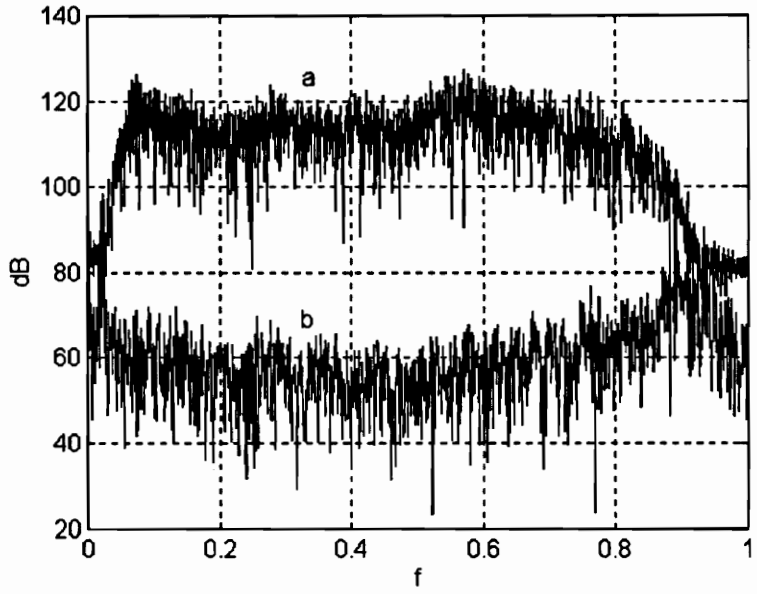


Fig. 7.23 Power Spectral Densities of
(a) Near End Echo (b) Residual Echo from Digital Echo Path Test on ADSP Implementation of NLMS(256).

The far-end signal, as well as the near-end echo produced by the room, were recorded. An NLMS adaptive filter, simulated in Matlab, was adapted using these recorded signals. The echo paths estimated by the ADSP-2181 implementation and the Matlab simulation are shown in Figure 7.24. The two estimates match well except for small differences in the tail portion, which may be due to misadjustment.

Figure 7.25 shows the learning curves obtained from implementation and simulation. We see that the simulation yields better ERL than the implementation. The only difference between the simulation and implementation scenarios here is the word length of the arithmetic; the signals were quantized to 16 bits in both cases.

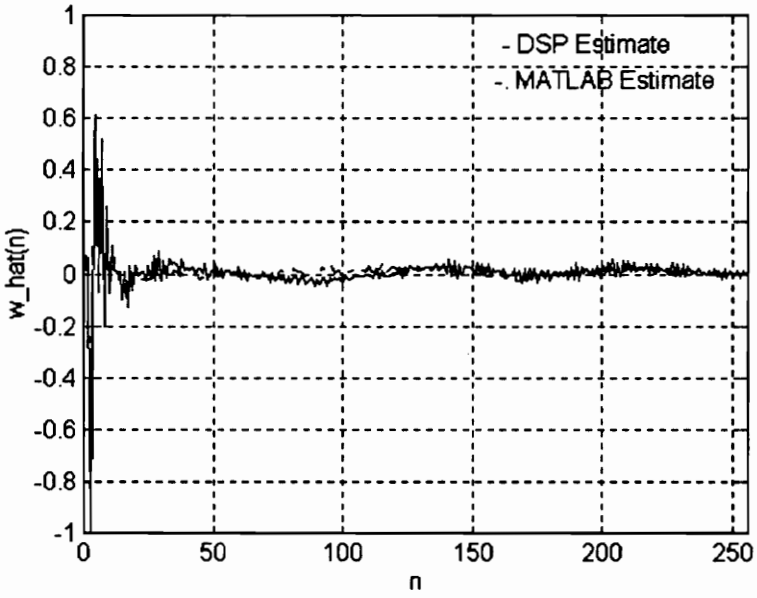


Fig. 7.24 Comparison of Impulse Response Estimates from Implementation (ADSP-2181) and Simulation (Matlab) of NLMS(256).

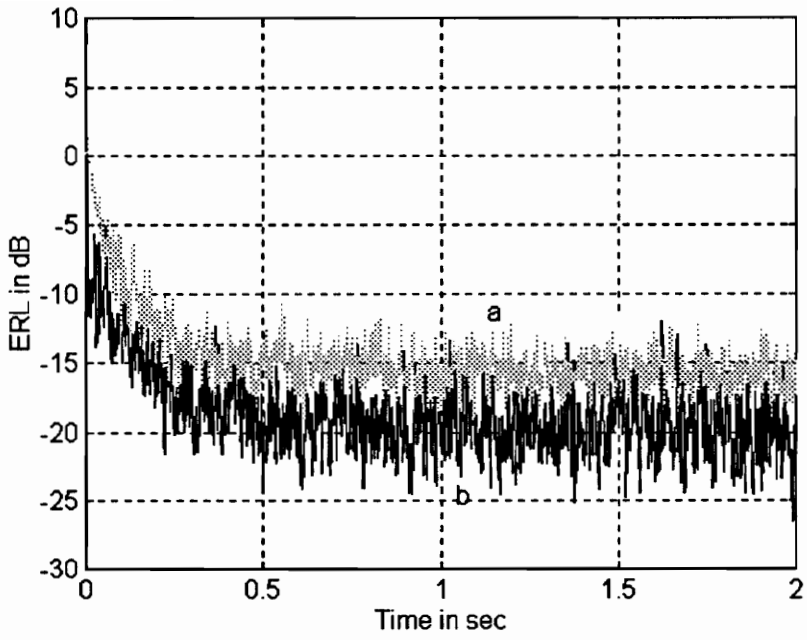


Fig. 7.25 Comparison of Learning Curves Obtained from
(a) Implementation (ADSP-2181) (b) Simulation (Matlab) of NLMS(256).

From the digital echo path test, we see that the performance of the DSP is limited only by the finite register length effects, in the absence of modeling error. Comparison of the implementation result with the simulation result indicates that the DSP performs the best it can, with the analog echo path (room). Recall that similar results were obtained when the continuous time RC filter was used as the echo path. Hence, for the 2181, the limited echo return loss achieved, with the room as echo path, seems to be due to artifacts incurred in sampling of the continuous time echo path.

7.7 Summary

The following table summarizes the results of the ITU-T: G.167 tests simulated on the NLMS(1024) acoustic echo canceling algorithm. We used white noise or speech as input signals for these tests.

Table 7.2 Test Results from Simulation of NLMS(1024).

TEST	REQUIRED VALUE	WHITE NOISE	SPEECH
TERLwst	45	180	40
TERLwdt	25	38	30
Tic	20	60	20
TERLpv	10	37	20
Trpv	20	80	30

All values specified in dB

We note that the test results are due to the AEC algorithms only. Therefore, the only additional loss (of 5 dB for TERLwst) is easily provided by the standard variable loss device and/or nonlinear processor.

With an electronic RC network as echo path, the TERLwst test yields about 37 dB and 15 dB for the C31 and 2181 implementations respectively. Finite arithmetic together with filter coefficient quantization seem to be responsible for differences in performance in this case.

Both real time DSP implementations of the NLMS(256) acoustic echo canceling algorithm, with an actual room as echo path, yield a TERLwst of about 15 dB. The echo return loss achieved here is possibly limited due to artifacts incurred in sampling of the continuous time echo path.

8. Conclusions and Recommendations for Future Work

The LMS, NLMS, RLS, and SNLMS adaptive filtering algorithms are compared based on their convergence rates, steady state ERL, and complexity. LMS, while simple to implement, has a poor convergence rate when the eigenvalue spread of the signal is high. NLMS mitigates this problem, with a slight increase in complexity. RLS provides the fastest convergence. On the other hand, it is highly computation intensive. SNLMS reduces the complexity at the cost of convergence rate.

The low complexity of SNLMS makes it a candidate for a practically feasible adaptation algorithm for acoustic echo cancellation, especially for rooms with long reverberation time constants. We show that the convergence is expedited by using NLMS initially, and subsequently switching to SNLMS to achieve high steady state ERL.

While echo cancellation algorithms perse are the major building blocks of hands-free full duplex telephones, several additional modules often are incorporated. Double-talk can be detected well with the Itakura distance and DTDS double-talk indicators. The DTDS is easier to evaluate than the Itakura distance, and can track changes faster. Ambient noise picked up by the microphone (and transmitted to the far-end) can be reduced by using an adaptive filter to cancel the noise. Our simulation results indicate that this approach can reduce the ambient noise by about 20 dB. The early stopping of adaptation, due to finite length register effects, can be delayed by storing the product $\mu(n)e(n)$ in

double-word format. This enhanced precision computation was shown to provide an additional 5 dB echo return loss over that achieved with single precision computation.

We also find that the performance of the NLMS echo canceling algorithm is highly dependent on the particular speech signal used for the tests. Further research to adaptively optimize the adaptive filter structure, to obtain signal independent performance, would be relevant.

Further research to reduce the complexity of the RLS algorithm would be worthwhile due to the desirability of its fast convergence at higher filter orders than presently implementable on DSPs. As mentioned in Chapter 7, neither underestimation nor overestimation of the order of the echo causing system is desirable. The addition of adaptively estimating the order required for echo cancelers would be useful as future work. By using IIR filters to model the echo path, the computational complexity of the filtering process can be reduced. Future work on IIR adaptive filtering algorithms that can guarantee stability of the estimated model would seem useful.

Acoustic echo canceler performance can be measured by the standard tests specified in the ITU-T : G.167 requirements. These tests were simulated on a 1024th order NLMS echo canceling algorithm. We used white noise or speech as input signals for these tests. The simulation results indicate that the echo canceling algorithm meets all the requirements for white noise input. With speech as input, while most specifications are met, the ERL provided by the echo canceling algorithm by itself is 5 dB below the

required value. The total ERL can be increased by the use of variable loss devices and nonlinear processors.

The real time DSP implementation, with a digital echo path, performs nearly as well as the Matlab simulation predicts. However, the real time DSP implementations of the NLMS acoustic echo canceler algorithm with the room as echo path yield a TERLwst of only about 15 dB; this is the same for white noise and speech input signals and does not seem to depend on the particular speech sample used. While in practice the total echo return loss can be increased by the use of variable loss devices and nonlinear processors, it would - for the future - be desirable to develop acoustic echo canceler algorithms capable of providing the required TERLwst on their own. This points to the need to understand the reason for the limited echo return loss achieved in real time DSP implementations involving the sampling of continuous time systems, in particular when a room constitutes the echo path.

While the tests and DSP implementations for individual aspects of an echo canceler reported here are encouraging, the eventual goal is to implement the integrated acoustic and network echo canceler proposed in Section 7.1 in its entirety on DSPs. Only this will show whether acceptable performance can be achieved. The ultimate future development would be of a combined DSP based acoustic echo canceler/controller incorporating double talk detection, howling detection and control, that would meet the required test values for any speech test signal and under exclusively full-duplex operation.

Appendix A Performance Requirements for Network Echo Cancelers

The recommendation G.165 of ITU-T (Telecommunication Standardization Sector of International Telecommunication Union, formerly known as CCITT) specifies the performance requirements of a network echo canceler. It defines the network echo canceler as a *voice operated device* placed in the 4-wire portion of a circuit and used for reducing near-end echo on the send path by subtracting an estimate of the echo from the near-end echo. Note that in this thesis, we place the network echo canceler in the 2-wire portion of the circuit, that is in the telephone itself. This appendix lists some of the salient requirements specified in G.165.

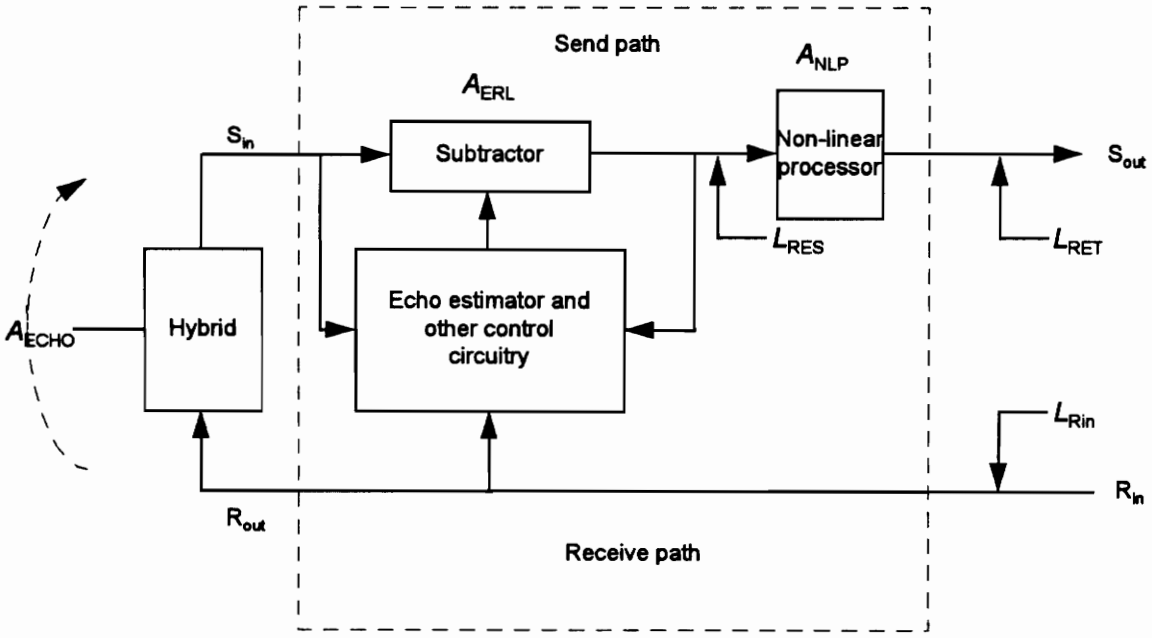


Fig. A.1 Network Echo Canceler.

The performance requirements are given in terms of tests specified by applying signals to R_{in} and S_{in} of the echo canceler, and measuring the S_{out} signals. Band-limited white (300-3400 Hz) noise is used as the receive input test signal.

With the nonlinear processor disabled for all values of receive input signal levels such that $L_{Rin} \geq -30$ dBm0 and ≤ -10 dBm0 and for all values of echo loss ≥ 6 dB, the residual echo level should be less than or equal to that shown in Figure A.2. When the nonlinear processor is enabled the returned echo level must be less than -65 dBm0.

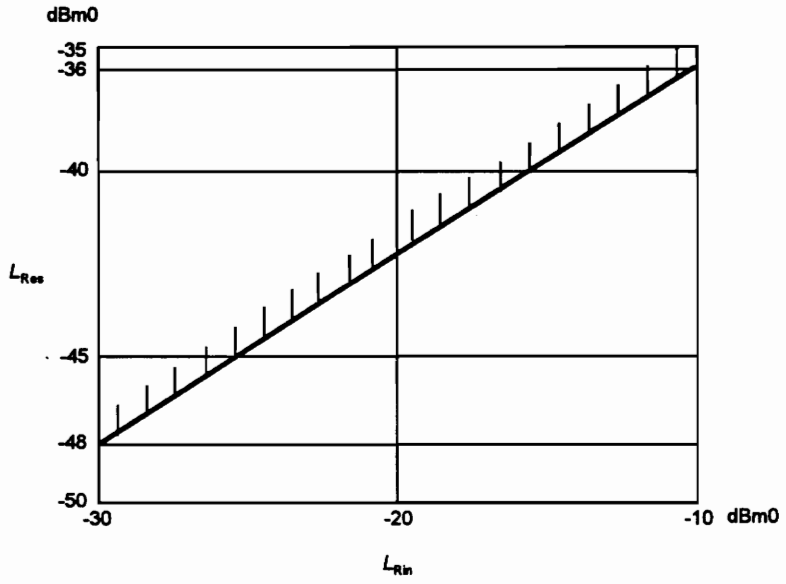


Fig. A.2 Echo Return Loss Requirements.

With the adaptive filter weights initially set to zero, for all values of input signal levels such that $L_{\mathrm{Rin}} \geq -30$ dBm0 and ≤ -10 dBm0 and for all values of echo loss ≥ 6 dB the combined loss ($A_{COM} = A_{ECHO} + A_{ERL} + A_{NLP}$) should be ≥ 27 dB.

The double talk detection should not be so sensitive that echo and low level near-end speech falsely cause operation of the double talk detector to the extent that adaptation does not occur. The double talk detector should be sufficiently sensitive and it should operate sufficiently fast to prevent large divergence during double-talk.

The nonlinear processor should be active when L_{RES} is at a significant level, because it is intended to further reduce L_{RES}. Also, the nonlinear processor should be inactive when near-end speech is present, since it should not distort the near end speech. When these two guidelines conflict, it is recommended that the control function favors the second.

Appendix B Performance Requirements for Acoustic Echo Cancelers

The recommendation G.167 of ITU-T (Telecommunication Standardization Sector of International Telecommunication Union, formerly known as CCITT) specifies the performance requirements of an acoustic echo canceler. It defines the acoustic echo canceler as a *voice operated device* installed in audio terminals on the customer premises, used for the purpose of eliminating acoustic echoes and protecting the communication from howling due to acoustic feedback from loudspeaker to microphone. This appendix lists some of the salient requirements specified in G.167.

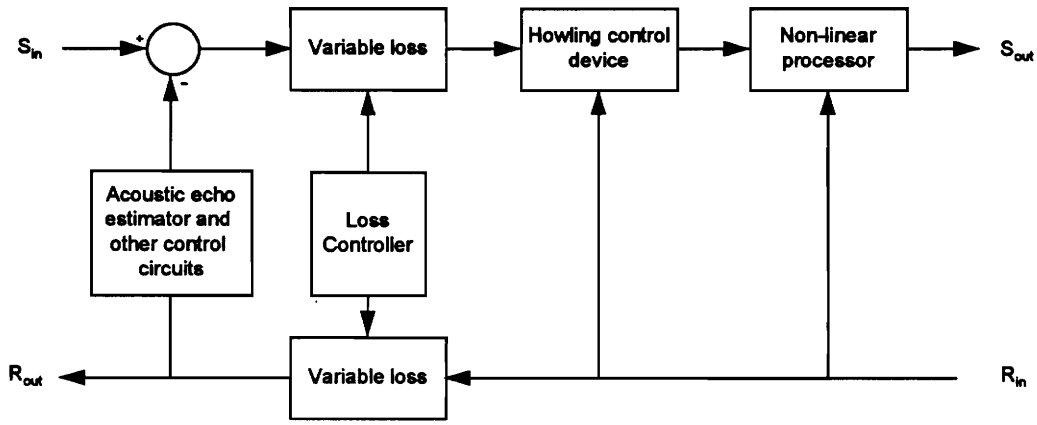


Fig. B.1 Functional Block Diagram of a Typical Acoustic Echo Canceler

G.167 recommends the use of a non-stationary signal (e.g. real speech) to test the performance of the echo canceler. The use of real rooms or enclosures with appropriate acoustic characteristics is recommended.

B.1 Total echo return loss - single talk (TERLwst)

A test signal is applied at R_{in} for a sufficient time (to be defined, under study) so that the different functional units reach their steady states. No other speech signal than the acoustic return from the loudspeaker is applied to the microphone. The value TERLwst is the difference (in dB) between the signal level at S_{out} before and after the enabling of the acoustic echo canceler. For hands-free telephones, TERLwst shall be at least 45 dB.

B.2 Total echo return loss - double talk (TERLwdt)

The acoustic echo canceler is operated as in the TERLwst test. After the echo return loss has attained TERLwst, an acoustic signal simulating the local user's speech is applied at the S_{in} point for 2 seconds. The processing unit is frozen and then the local speech is removed. The maximum total echo return loss that this frozen AEC can provide is the TERLwdt, which shall be at least 25 dB.

B.3 Initial Convergence Time (Tic)

All the AEC functional units are initially reset and then enabled. A signal is applied at R_{in} and the processing unit is frozen after 1 second. The signal measured at S_{out} should be down by at least 20 dB, after 1 second.

B.4 Echo Return Loss during Echo Path Variation (TERLwpv)

The AEC is initially operated as in the TERLwst test. After the steady state is reached, a simulated echo path variation is applied for 5 seconds. At the end of the echo path variations, the processing unit is frozen, and the residual echo level is measured. TERLwpv is the difference between the level of S_{out} before enabling the AEC and the measured value. The required value of TERLwpv is at least 10 dB.

B.5 Recovery Time after Echo Path Variation (Trpv)

The AEC is initially operated as in the TERLwst test. After the steady state is reached, a simulated echo path variation is applied for 5 seconds. At the end of the echo path variation a timer is started. After 1 second, the processing unit is frozen and the residual echo level is measured. The echo should be attenuated by at least 20 dB, at the end of 1 second.

Bibliography

[1] C. W. K. Gritton and D. W. Lin, *Echo Cancellation Algorithms*, IEEE ASSP Magazine, pp. 30 - 37, April 1984.

[2] A. A. (Louis) Beex, S. G. Sankaran, A. Padmanabhan, and K. Rangarajan, *Full-Duplex Speakerphone*, Project Report Submitted to CIT, DSP Research Laboratory, Virginia Tech, March 1996.

[3] J. C. Baumhauer, S. H. Early, J. H. Fikus, S. L. Gay, and M. A. Zuniga, *Audio Technology Used in AT&T's Equipment*, AT&T Technical Journal, vol. 77, no. 2, pp. 57 - 70, March/April 1995.

[4] K. Murano, S. Unagami, and F. Amano, *Echo Cancellation and Applications*, IEEE Communications Magazine, pp. 49 - 55, January 1990.

[5] A. Gilloire, *Performance evaluation of acoustic echo control : required values and measurement procedures*, Annals de Telecommunications, vol. 49, no. 7 - 8, pp. 368 - 372, 1994.

[6] Simon Haykin, Adaptive Filter Theory, Prentice Hall Inc., 1991.

[7] D. T. M. Slock, *On the Convergence Behavior of the LMS and the Normalized LMS algorithms*, IEEE Transactions on Signal Processing, vol. 41, no. 9, pp. 2811 - 2825, September 1993.

[8] M. Tarrab and A. Feuer, *Convergence and Performance Analysis of the Normalized LMS Algorithm with Uncorrelated Gaussian Data*, IEEE Transactions on Information Theory, vol. 34, no. 4, pp. 680-691, July 1988.

[9] J. Prado and E. Moulines, *Frequency-Domain Adaptive Filtering with Applications to Acoustic Echo Cancellation*, Annals de Telecommunications, vol. 49, no. 7 - 8, pp. 414-428, 1994.

[10] E. Eleftheriou and D. D. Falconer, *Tracking Properties and Steady-State Performance of RLS Adaptive Filter Algorithms*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-34, no. 5, pp. 1097 - 1110, October 1986.

[11] D. R. Morgan and J. C. Thi, *A Delayless Subband Adaptive Filter Architecture*, IEEE Transactions on Signal Processing, vol. 43, no. 8, pp. 1819 - 1830, August 1995.

[12] A. Gilloire, *Experiments with Subband Acoustic Echo Cancelers for Teleconferencing*, Proceedings of ICASSP, Dallas, pp. 2141 - 2144, April 1987.

[13] L. B. Jackson, Digital Filters and Signal Processing, Kluwer Academic Publishers, 1996.

[14] K. Ashihara, K. Nishikawa, and H. Kiya, *Improvement of Convergence Speed for Subband Adaptive Digital Filters Using Multirate Repeating Method*, Proceedings of ICASSP, Detroit, pp. 989 - 992, May 1995.

[15] G. Zakaria, *Switching Adaptive Filter Structures for Improved Performance*, M. S. Thesis, Virginia Tech, December 1993.

[16] CCITT, *Recommendation G.165 Echo Cancelers*, CCITT Blue Book, Volume III, Fascicle III.1, pp. 221 - 243, November 1988.

[17] ITU-T, *Recommendation G.167 Acoustic Echo Cancelers*, 1993.

[18] J. M. Paez and M. G. Otero, *On the implementation of a partitioned block frequency domain adaptive filter for long acoustic echo cancellation*, Signal Processing, vol. 27, no. 3, pp. 301 - 315, June 1992.

[19] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, Discrete-Time Processing of Speech Signals, Macmillan Publishing Company, New York, 1993.

[20] B. Ayad, G. Faucon, and R. L. Bouquin-Jeannes, *Optimization of a Noise Reduction Preprocessing in an Acoustic Echo and Noise Controller*, Proceedings of ICASSP, Atlanta, pp. 953 - 956, May 1996.

[21] F. Capman, J. Boudy, and P. Lockwood, *Acoustic Echo Cancellation using a Fast QR-RLS Algorithm and Multirate Scheme*, Proceedings of ICASSP, Detroit, pp. 969 - 972, May 1995.

[22] J. M. Mendel, Lessons in Estimation Theory for Signal Processing, Communications, and Control, Prentice Hall P T R, New Jersey, 1995.

[23] R. Martin and J. Altenhoner, *Coupled Adaptive Filters for Acoustic Echo Control and Noise Reduction*, Proceedings of ICASSP, Detroit, pp. 3043 - 3046, May 1995.

[24] H. Taub and D. L. Schilling, Principles of Communication Systems, Second Edition, Tata McGraw-Hill Publishing Company Limited, New Delhi, 1994.

[25] Texas Instruments, TMS320C3X User's Guide, Revision F, Texas Instruments, July 1992.

[26] Analog Devices, ADSP-2100 Family User's Manual, Third Edition, Analog Devices, September 1995.

[27] ASPI, Elf DSP Application Developer's Toolkit - Instruction Manual, ASPI, Atlanta, 1993.

[28] Analog Devices, ADSP-2100 Family EZ-KIT Lite - Reference Manual, First Edition, Analog Devices, May 1995.

[29] S. M. Kuo and J. Chen, *New Adaptive IIR Notch Filter and its Application to Howling Control in Speakerphone System*, Electronics Letters, vol. 28, no. 8, pp. 764 - 766, April 1992.

Vita

Sundar G. Sankaran was born in Madurai, India. He received the Bachelor of Engineering degree in Electronics and Communication Engineering from Anna University, Madras, India in 1992. From 1992 to 1994, he was at Infosys Technologies Limited, Bangalore, India, as a Systems Analyst, working on digital signal processing hardware design and embedded software development. Since August 1994, he is a graduate student at Virginia Polytechnic Institute and State University, Blacksburg, VA, currently pursuing his Ph.D. in Electrical Engineering. His research interests are in the areas of digital signal processing, adaptive signal processing, stochastic signal processing, and information theory.

He is a student member of the IEEE Signal Processing and Information Theory societies.