A Multi-band Acoustic Echo Canceller for Vehicular Handsfree Telephone

by

Mingxi Fan

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degrees of

Bachelor of Science in Electrical Science and Engineering

and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 5, 1999

[June 1999]

Author____________________________________
Department of Electrical Engineering and Computer Science
May 5, 1999

Certified by____________________________________
G. D. Forney
Thesis Supervisor

Accepted by____________________________________
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

A Multi-band Acoustic Echo Canceller for Vehicular Hands-free Telephone

by

Mingxi Fan

Submitted to the
Department of Electrical Engineering and Computer Science

May 5, 1999

In Partial Fulfillment of the Requirement for the Degree of
Bachelor of Science in Electrical Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

A novel subband acoustic echo canceller is designed and implemented after thorough analysis and investigation of prior echo cancellation techniques. It applies different suitable adaptation algorithms to non-uniform frequency bands according to the amount of echo present within each band. It also incorporates non-uniform resolution analysis and anti-aliasing filters in subband division. A new double-talk detector that is entirely based on correlations is developed and verified to complete the echo canceller package. The entire system has exhibited excellent echo energy reduction and convergence performance in realistic vehicular environments. The echo canceller has been implemented on a fixed-point DSP. The DSP implementation will be further optimized for future hands-free products.

Thesis Supervisor: G. D. Forney
Title: Adjunct Professor, MIT (Laboratory for Information and Decision Systems)

Acknowledgments

During all phases of this research project, I have been very fortunate to receive valuable suggestions, assistance and support from many of my professors, supervisors and colleagues. At Hughes Network Systems (HNS), I would like to first thank Allan Lamkin, my VI-A thesis supervisor, who gave me the opportunity to work on the project. Allan granted me a great deal of flexibility in developing and implementing the algorithm and provided very helpful comments and insights at all stages of the project. I would also like to thank many of my co-workers and mentors at HNS, including Michael Castello, Peter Sobczeck, Frank Onochie, T.J. Vishwarthan, Romeo Velarde, Sharath Anand, Kan Zhu, Luong Le, On-Wa Yeung and Alfred Ibrahim. They always enthusiastically and patiently shared with me their technical expertise and experiences, which enlightened me in resolving a number of technical difficulties. I would also like to express my appreciation toward Nancy Neigus, my VI-A advisor at HNS, who has spent a great deal of effort to make my experiences at the company worthwhile. Nancy also provided me with encouragement and useful advice throughout the last three years.

In writing this thesis, my special thanks goes to Dr. Dave Forney, my thesis advisor at MIT, for his support and guidance throughout the project. Dr. Forney has spent many precious hours proofreading my writing, and his suggestions and guidance have been invaluable in enhancing the overall presentation of the thesis, particularly in theoretical aspects. I would also like to thank Prof. Mildred Dresselhaus, my academic advisor, who provided me very useful suggestions in terms of thesis planning and organization. I also want to thank Prof. Donald Troxel, my VI-A advisor at MIT, and Ms. Lydia Wereminsky in the VI-A office for their effort in making my overall VI-A experiences a successful one.

Finally, my deepest appreciation goes my family. Their love and support provided me the strength to conquer all the challenges during my undergraduate and graduate years at MIT. In particular, I would like to dedicate this thesis to my mother, in a hope that her joy in receiving it will speed up her recovery from cancer.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

I. Introduction

Since the late 1980s, the loudspeaker telephone has become popular because it provides users the convenience of teleconferencing and hands-free telephony applications. In such systems, however, several troublesome phenomena can severely degrade the quality of speech communications, among which acoustic echo is perhaps the most serious.

Acoustic echo is a problem caused by acoustic coupling between microphone and loudspeaker of the hands-free telephone. As the far-end speech is played by the loudspeaker, part of this speech will be reflected by the surroundings and collected by the microphone and subsequently transmitted back to the other end [1][2]. This echo is annoying because it causes the far-end talker to hear a delayed and distorted version of his/her own previous speech.

The best way to solve the acoustic echo problem appears to be adaptive echo cancellation. In this solution, an acoustic echo canceller (AEC) adaptively builds a model of the echo path in the form of a transversal filter and subtracts the output of the filter from the signals picked up by the microphone [7]. The uncancelled echo residual is then suppressed and transmitted to the other end. The echo residual is also used to update coefficients of the impulse response of the estimated echo path. Figure 1 shows a typical acoustic echo cancellation scenario [1].

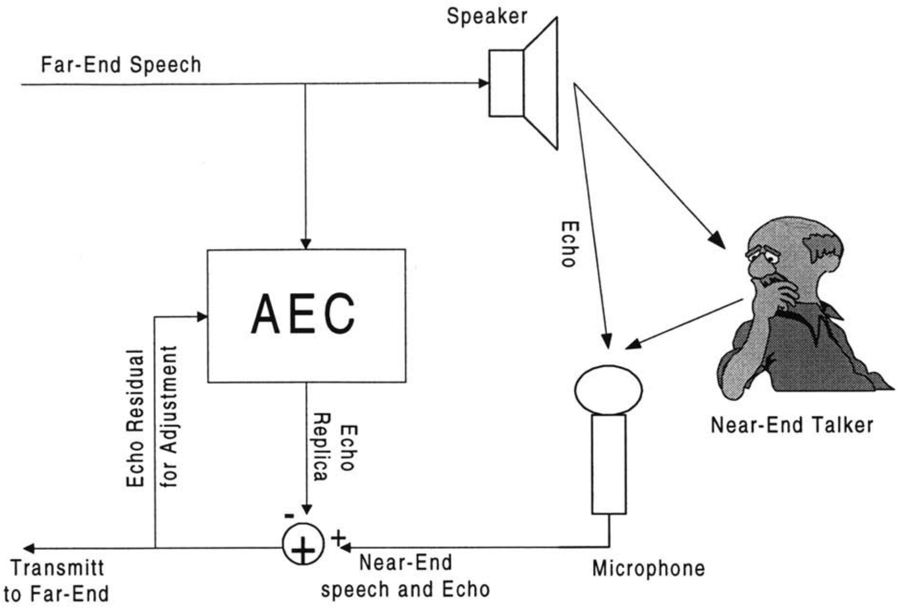

Figure 1. Block Diagram of Hands-free Telephone Environment

Even though numerous echo cancellation and suppression solutions have been proposed and implemented in the past, achieving efficient acoustic echo attenuation in certain settings is still very challenging. One example of such an environment is wireless hands-free telephony in a vehicle. The long delays of wireless transmissions, especially in a geo-mobile satellite system, impose a very stringent requirement on echo cancellation performance. Echo suppression techniques based on simple gain switching are not suitable for vehicular speakerphone, as they may introduce an unpleasant non-uniform background noise level heard by the far-end talker. Thus, a significant portion of this high echo attenuation must be obtained through the adaptive filter. Other difficulties include the adverse effect of engine noise and the large reverberation time of a typical vehicular interior. Excessive background noise can greatly perturb the adaptation process. The reverberation time of the car interior may exceed 100-150ms. In order to achieve powerful echo cancellation, the number of taps for conventional adaptive finite impulse response (FIR) filters may become very large (> 1000 taps), thus increasing the overall implementation complexity and cost [8].

The purpose of this project is to investigate, design and implement an efficient solution for full-duplex acoustic echo cancellation in a vehicular environment, particularly for geo-mobile telephony. The first stage of the task involves studying existing algorithms and their tradeoffs concerning level of echo suppression, convergence time, and computational complexity. The lessons learned from the previous methods are then applied to the design of a novel echo cancellation algorithm at the second stage of the project. Finally, the new echo canceller is simulated and implemented on a fixed-point DSP for performance verification and comparison with previous methods.

This research was carried out as an R & D project for Hughes Network Systems, Inc. as part of an MIT VI-A Internship Program. Hughes Network Systems will use the new echo canceller package on a vehicular docking adapter and possibly a fixed docking adapter for the company's geo-mobile satellite system products. Geo-mobile systems, which have long round-trip delays (400 ms one-way) are therefore the context assumed for this research.

This document first introduces the basic notations for and performance requirements of an acoustic echo canceller, and then analyzes and compares several existing and representative adaptation algorithms. The analyses lead to the design of a new multi-band echo cancellation algorithm. Simulation performance results are presented. Advantages of the multi-band echo canceller over its predecessors are discussed. The design of auxiliary devices, such as filter banks and double-talk detectors, is also considered. Suggestions for future research are included before the conclusion.

II. Essential Notation and Specifications

2.1 Echo-path Modeling

The echo-path of the far-end talker echo can be modeled as a linear system with time-varying impulse response $h_i[n]$, where i denotes the coefficient index of h at time n. The validity of the linear assumption is supported by the fact that atmospheric pressure is roughly 194 dB SPL, while that the threshold of pain to human hearing is about 120 dB SPL. Consequently, the nonlinearity of speech propagation through the air and into the human ear can be ignored [1].

Given the speech samples x[n], the resultant acoustic echo, y[n], is

$$y[n] = \sum_{i=0}^{\infty} h_i[n]x[n-i]$$

where $h_i[n]$ models the impulse response of the echo path (i.e., from microphone to loudspeaker) at time n [3].

In order to eliminate this echo, an echo replica, $\hat{y}[n]$, needs to be generated and subtracted from the echo y[n]. This echo replica can be found by feeding the input x[n] through an adaptive FIR filter with coefficients $c_i[n]$, 0<i<N-1, where N is the length of the filter so that $\hat{y}[n] = \sum_{i=0}^{N-1} c_i[n]x[n-i]$. The echo residual, defined as e[n] = y[n] - $\hat{y}[n]$, is transmitted to the far-end. To ensure the stability of the filtering process, only FIR filters are considered.

2.2 Relevant Design Considerations

It is important to specify how much reduction in echo energy this Acoustic Echo Canceller (AEC) must achieve for the hands-free phone system to reach satisfying performance. The required echo reduction depends on the signal levels of both the near-end and far-end talkers.

Reduction in echo is measured by a parameter called Echo Return Loss Enhancement (ERLE). ERLE is defined as the following: [1]

$$ERLE = 10\log_{10}\frac{\sigma_y^2}{\sigma_e^2}$$

where σ_y^2 and σ_e^2 are echo and residual variances, respectively. ERLE is therefore a measure (in dB) of the power of the echo residual e[n] compared to that of the unprocessed acoustic echo y[n] received by the microphone .

In a typical hands-free telephone conversation, the signal level of the near-end speaker ranges approximately between 55 dB and 70 dB, depending on the distance between the talker and the microphone. The acoustic echo of the far-end speech can go up to 70 dB. Therefore, the returned far-end talker echoes may be 15 dB higher than the near-end talker signal. It has been subjectively determined in the industry that the near-end talker level must be at least 10 dB higher than the returned far-end talker echoes [1]. Therefore, our AEC must achieve at least 25 dB of echo reduction; in other words, ERLE should be at least 25 dB. We are aiming for 25-30 dB ERLE in our design.

Two other characteristics of the echo canceller are as important as ERLE: rate of convergence and computational complexity. The echo residual in most situations needs to converge quickly to the desired level. The acoustic echo path is a non-stationary channel. The movement of the speaker and other objects in the surrounding environment can drastically modify the impulse response of the acoustic echo path. The rapid time-varying character of the echo path requires fast convergence of the echo residual to ensure accurate tracking ability [8]. For the HNS geo-mobile product, the desired convergence time is less than 1 second. For most conventional algorithms, the echo energy reduction is inversely proportional to the convergence speed. Thus, the trade-off between convergence rate and ERLE must be addressed in designing an echo canceller. Due to the stringent ERLE requirement of geo-mobile systems and the rapidly time-varying nature of the acoustic echo path in a vehicular environment, the objective of our echo canceller design is to maximize echo energy reduction while keeping convergence time within 1 second.

Relatively simple computational complexity is desired because it will allow algorithmic implementation on low-cost DSP platforms. The amount of echo energy reduction and convergence time is general inversely proportional to the computational complexity, which depends on the number of adaptive filter taps. The number of adaptive filter taps necessary in turn depends on the reverberation time of the environment in which the phone is used [2]. (Reverberation time is the time interval during which the reverberation level drops by 60 dB, and is typically 100 ms or more for a large vehicular interior.) Consequently, an acoustic echo canceller usually has a very heavy computational burden, due to the large number of taps required.

Since the acoustic echo path is characterized by its impulse response, the number of taps for the representation of the impulse response is of the same order as the product of the sampling frequency and the reverberation time of the environment where the hands-free phone is installed. In our case, the reverberation time of a large vehicle compartment is about 0.07-0.15 s. To get more than 30 dB of echo reduction, we need a process window that is at least half the reverberation time. The sampling frequency is chosen to be 8 kHz. Therefore, in order to achieve an echo energy attenuation of 30 dB with a process window of 60 ms, more than 500 taps must be implemented in an AEC using a FIR filter [8].

Our approach in this product was to focus our efforts on optimizing ERLE and reducing computational complexity, while keeping the convergence time under one second.

Table 1 summarizes the desired characteristics of our acoustic echo canceller:

Table 1
Acoustic Echo Canceller Performance Specifications

Requirement	**Value**
ERLE (w/o echo suppressor)	25 – 30 dB
ERLE (w/ echo suppressor)	45 – 60 dB
Convergence Time	1 second
Process window	60 ms
DSP MIPS (Fixed Point)	20 MIPs
Program Memory Usage	16 K Words
Data Memory Usage	4 K Words

III. Analysis of Existing Methods

Since numerous echo cancellation techniques have already been proposed and implemented, it is important to fully understand their advantages and weaknesses in order to propose an elegant design of acoustic echo canceller. During the preliminary stage of this project, several popular and representative adaptive filtering methods have been carefully investigated, simulated, and compared. These methods can be roughly divided into three categories – time-domain full-band algorithms, transform-domain methods, and subband approaches. This chapter presents the underlying concepts of these algorithms, and Chapter 5 compares the performance of these algorithms by simulation both among themselves and against the new multi-band design.

3.1 An Overview on Time-Domain Full-band Algorithms

From the early 1970s through the mid-1980s, full-band adaptive echo cancellation techniques were popular. Some of these techniques include Least Mean Squares (LMS) adaptation [3][6][14] (a typical example of the Stochastic Gradient (SG) approach [7]), the Recursive Least Squares (RLS) algorithm [3] and the Affine Projection (AP) method [6]. Researchers proposed these algorithms either for implementation simplicity or for large reductions in echo energy while keeping a desired convergence rate. Some of these algorithms will be discussed in this section.

3.1.1 The Basics of Stochastic Gradient Type Algorithms

The most commonly used implementation technique is the family of Least Mean Squares (LMS) algorithms [6], since these routines are easy to implement. This stochastic-gradient-type MSE algorithm is based on the following adaptive filtering scenario [3]:

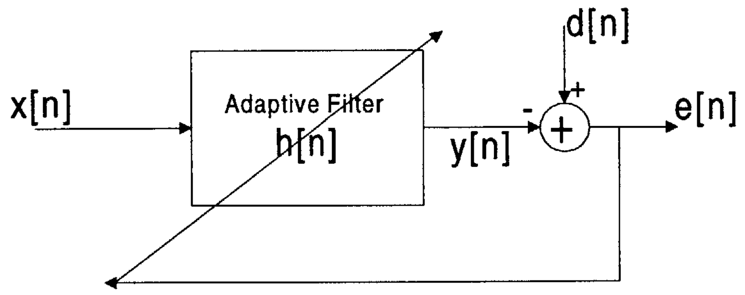

Figure 2. Adaptive Filter Diagram

Let's denote the auto-correlation matrix of $\mathbf{x}[n] = [x[n], x[n-1], x[n-2], \ldots\ldots x[n-N+1]]^T$ as $\mathbf{R}_{xx}$, and cross-correlation matrix between d[n] and **x**[n] as $\mathbf{R}_{dx}$. We know that

$$y[n] = h_i[n] * x[n] = \mathbf{h}[n]^T\mathbf{x}[n] \qquad \textbf{(Eq. 1)}$$

where $\mathbf{h}[n] = [h_0, h_1, h_2, \ldots\ldots h_{N-1}]^T$ denotes the N coefficients of the adaptive filter at time n.

For y[n] to resemble d[n] as closely as possible, one approach would be to minimize the mean squared error (MSE), i.e. $E[e^2[n]]$. We can express the MSE in terms of our specified correlation matrices by the following steps:

$$\begin{aligned} E[e^2[n]] &= E[(d[n] - y[n])^2] \\ &= E[d^2[n] - 2d[n]y[n] + y^2[n]] \\ &= E[d^2[n]] - 2E[d[n]\,\mathbf{h}[n]^T\mathbf{x}[n]] + \mathbf{h}[n]^T\mathbf{R}_{xx}\mathbf{h}[n] \\ &= E[d^2[n]] - 2\mathbf{h}[n]^T\mathbf{R}_{dx} + \mathbf{h}[n]^T\mathbf{R}_{xx}\mathbf{h}[n] \end{aligned}$$

From the principle of orthogonality, we know that at minimum $E[e^2[n]]$ for linear estimation, e[n] must be orthogonal to any vector-valued linear function of the input data (i.e., **x**[n]) in Hilbert space. In other words, at optimal **h**[n] (defined as $\mathbf{h}_o[n]$), we have

$E[\mathbf{x}[n]e[n]] = 0$. Since $E[\mathbf{x}[n]e[n]] = E[\mathbf{x}[n](\mathbf{d}[n] - \mathbf{x}^T[n]\mathbf{h}_o)] = \mathbf{R}_{xd} - \mathbf{R}_{xx}\mathbf{h}_o$, we can then set $\mathbf{R}_{dx} - \mathbf{R}_{xx}\mathbf{h}_o = 0$, which implies that the optimum set of filter coefficients for the adaptive filter is [3]

$$\mathbf{h}_o = \mathbf{R}_{xx}^{-1}\,\mathbf{R}_{dx}$$

From this we obtain $E_{min} = E[d^2[n]] - \mathbf{R}_{xd}^T\mathbf{R}_{xx}^{-1}\mathbf{R}_{dx} = E[d^2[n]] - \mathbf{h}[n]^T\mathbf{R}_{xx}\mathbf{h}[n]$

This $\mathbf{h}_o$ is the Wiener solution, and it will lead to the optimal estimation of d[n] based on $\mathbf{x}[n]$ in the least-mean-squares sense. However, in terms of implementation, this approach is impractical, since the correlation matrices are usually unavailable, and evaluating inverses of matrices requires a large amount of computations. [7]

A reasonable alternative, which does not involve computing matrix inverses, would be to minimize MSE via a incremental gradient-type process, such as Newton's method. Through this process, the coefficients of **h** will eventually converge to the Wiener solution. The MSE surface can be derived as the following:

$$\begin{aligned} E[e^2[n]] &= E[d^2[n]] - 2\mathbf{h}[n]^T\mathbf{R}_{dx} + \mathbf{h}[n]^T\mathbf{R}_{xx}\mathbf{h}[n] \\ &= E_{min} + \mathbf{h}_o^T\mathbf{R}_{dx} - 2\mathbf{h}[n]^T\mathbf{R}_{dx} + \mathbf{h}[n]^T\mathbf{R}_{xx}\mathbf{h}[n] \\ &= E_{min} - (\mathbf{h}[n] - \mathbf{h}_o)^T\mathbf{R}_{dx} + \mathbf{h}[n]^T\mathbf{R}_{xx}(\mathbf{h}[n] - \mathbf{h}_o) \\ &= E_{min} - (\mathbf{h}[n]-\mathbf{h}_o)^T\mathbf{R}_{xx}(\mathbf{h}[n]-\mathbf{h}_o) \\ &= E_{min} - (\mathbf{h}[n]-\mathbf{h}_o)^T\mathbf{Q}^T\Lambda\mathbf{Q}(\mathbf{h}[n]-\mathbf{h}_o) \\ &= E_{min} - \mathbf{V}^T\Lambda\mathbf{V} \\ &= E_{min} - \sum_{i=0}^{N}\lambda_i V_i^2 \end{aligned}$$

Through the above diagonalization process (i.e. let $\mathbf{R}_{xx} = \mathbf{Q}^T\Lambda\mathbf{Q}$, where **Q** contains eigenvectors of $\mathbf{R}_{xx}$, and Λ is a diagonal matrix containing eigenvalues of $\mathbf{R}_{xx}$), the MSE surface has been expressed in terms of scaled eigenvectors of the input correlation matrix. In other words, these eigenvectors are the principal axes of the MSE surface [3].

One efficient gradient-type algorithm is the Steepest Descent Algorithm, which incrementally converges toward the minimum values of the MSE surface. This algorithm is expressed as follows:

$$\mathbf{h}[n+1] = \mathbf{h}[n] - \mu\nabla^{T}_{\mathbf{h}}[n]$$

where $\nabla_{\mathbf{h}}[n] = \dfrac{\partial E[e^2[n]]}{\partial \mathbf{h}} = -2\mathbf{R}_{dx}{}^{T} + 2\mathbf{h}^{T}\mathbf{R}_{xx}$. Optimal filter coefficients can be determined by setting this gradient to zero.

Under this algorithm, the filter coefficients **h**[n] will converge to the Wiener solution if $0 < \mu < \lambda_{max}{}^{-1}$, where λ_{max} is the maximum eigenvalue of $\mathbf{R}_{xx}$. The steepest descent algorithm has a transient behavior with convergence time constant $\tau_i \approx \dfrac{1}{4\mu\lambda_i}$ along each principal axis. If $\lambda_i = \lambda_{min}$ and $\mu < \lambda_{max}{}^{-1}$, where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\mathbf{R}_{xx}$, respectively, then the convergence time constant is at least $\dfrac{\lambda_{max}}{4\lambda_{min}}$, so this algorithm will be slow for an input correlation matrix $\mathbf{R}_{xx}$ with a large spread of eigenvalues [3].

3.1.2 LMS-Type Algorithms

A problem in the steepest descent algorithm is how to estimate the correlation vectors and matrices, $\mathbf{R}_{dx}$ and $\mathbf{R}_{xx}$. A reasonable and yet readily obtainable estimate comes from instantaneous correlations, i.e., $\mathbf{R}_{dx} = d[n]\mathbf{x}[n]$ and $\mathbf{R}_{xx} = \mathbf{x}[n]\mathbf{x}[n]^{T}$. In this case, the gradient $\nabla_{\mathbf{h}}[n]$ is replaced by the so-called stochastic gradient

$$\begin{aligned}\hat{\nabla}_{\mathbf{h}}[n] &= -2d[n]\mathbf{x}[n]^{T} + 2\mathbf{h}[\mathbf{n}]^{T}\mathbf{x}[\mathbf{n}]\mathbf{x}[\mathbf{n}]^{T} \\ &= -2(d[n] - \mathbf{h}[\mathbf{n}]^{T}\mathbf{x}[\mathbf{n}])\mathbf{x}[\mathbf{n}]^{T} \\ &= -2e[n]\mathbf{x}[n]^{T}\end{aligned}$$

and now we can update the coefficients **h** as follows:

$$\mathbf{h}[n+1] = \mathbf{h}[n] + 2\mu e[n]\mathbf{x}[\mathbf{n}]$$

which leads to the basic Least Mean Squared (LMS) algorithm.

The general LMS algorithm is summarized as follows: [1]

Table 2: General LMS Algorithm

1) Initialization: $\mathbf{x}[0] = \mathbf{h}[0] = [0\ 0\ 0\ \ldots\ 0]^T$

Do for n > 0:

2) Update Error: $e[n] = d[n] - \mathbf{h}^T[n]\mathbf{x}[n]$
3) Update Filter: $\mathbf{h}[n+1] = \mathbf{h}[n] + 2\mu e[n]\mathbf{x}[n]$

The convergence properties of LMS algorithm are stochastically the same as those of the steepest-descent algorithm, i.e. **h** will converge if $0 < \mu < \lambda_{max}^{-1}$, where λ_{max} is the maximum eigenvalue of $\mathbf{R_{xx}}$, and the transient time constant for the algorithm at each eigenvector mode is $\tau_i \approx \frac{1}{4\mu\lambda_i}$.

We can conclude from the LMS convergence property that the choice of μ is critical, since it directly influences the convergence time. It would be best if we could increase the convergence speed without relying too much on the characteristics of the input correlation matrix. An intuitive strategy is to choose μ to maximize the reduction of the squares of instantaneous error. [3]

The instantaneous error squared can be expressed as:

$$\begin{aligned} e^2[n] &= (d[n] - y[n])^2 \\ &= d^2[n] + \mathbf{x}^T[n]\mathbf{h}[n]\mathbf{h}^T[n]\mathbf{x}[n] - 2d[n]\mathbf{h}^T[n]\mathbf{x}[n] \end{aligned}$$

If we introduce an update in the filter coefficients, i.e. $\mathbf{h}^{\backprime}[n] = \mathbf{h}[n+1] = \mathbf{h}[n] + \Delta\mathbf{h}^{\backprime}[n]$,

where $\Delta\mathbf{h}^{\backprime}[n] = 2\mu e[n]\mathbf{x}[n]$, then the change in corresponding error is

$$\begin{aligned} e^{\backprime 2}[n] &= e^2[n] + 2\Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]\mathbf{x}^T[n]\mathbf{h}[n] \\ &\quad + \Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]\mathbf{x}^T[n]\Delta\mathbf{h}^{\backprime}[n] - 2d[n]\Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n] \end{aligned}$$

Hence, we can express the instant squared error reduction as

$$\begin{aligned} \Delta e^2[n] &= e^{\backprime 2}[n] - e^2[n] \\ &= 2\Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]\mathbf{x}^T[n]\mathbf{h}[n] + \Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]\mathbf{x}^T[n]\Delta\mathbf{h}^{\backprime}[n] \\ &\qquad - 2d[n]\Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n] \\ &= -2\Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]e[n] + \Delta\mathbf{h}^{\backprime T}[n]\mathbf{x}[n]\mathbf{x}^T[n]\Delta\mathbf{h}^{\backprime}[n] \\ &= -4\mu e^2[n]\mathbf{x}^T[n]\mathbf{x}[n] + 4\mu^2 e^2[n][\mathbf{x}^T[n]\mathbf{x}[n]]^2 \end{aligned}$$

To minimize $\Delta e^2[n]$ with respect to μ, we set $\frac{\partial \Delta e^2[n]}{\partial \mu} = 0$, and find that the optimum value of μ is given by

$$\mu = \frac{1}{2\mathbf{x}^T[n]\mathbf{x}[n]}$$

With this choice of variable convergence factor, the updating equation for LMS algorithm is given by

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \frac{e[n]\mathbf{x}[n]}{\mathbf{x}^T[n]\mathbf{x}[n]}$$

This leads to the Normalized LMS (NLMS) algorithm, which is summarized below: [3]

Table 3: Normalized LMS Algorithm

1) Initialization: $\mathbf{x}[0] = \mathbf{h}[0] = [0\ 0\ 0\ \ldots\ 0]^T$

Choose μ_0 in the range of $0 < \mu_0 < 2$ for convergence,
Set γ = small constant

Do for $n > 0$:

2) Update Error: $e[n] = d[n] - \mathbf{h}^T[n]\mathbf{x}[n]$
3) Update Filter: $\mathbf{h}[n+1] = \mathbf{h}[n] + \dfrac{\mu_0 e[n]\mathbf{x}[n]}{\gamma + \mathbf{x}^T[n]\mathbf{x}[n]}$

A fixed convergence factor μ_0 is introduced in the algorithm to control the error misadjustment, since all the derivations are based on instantaneous values of squared errors. Also, a small constant γ is included in the denominator to avoid large step sizes when the input becomes small.

The LMS algorithm in general has a simple computational structure, which is very suitable for implementation purposes. However, the drawback of this type of approach is that the convergence speed of the LMS algorithm is related to the statistical property (eigenvalue spread) of the input signal. LMS type algorithms and their convergence properties are derived under the assumption that the input x[n] is white, i.e. $E[x[i]x[j]] = k\delta[i-j]$, where k is a scaling constant, and that the d[n] are uncorrelated with all past values of x[n]. In the case of acoustic echo cancellation, however, the assumption is rather a dubious one, since we are dealing with speech signals, which in nature are usually highly correlated in time and have a wide spread of eigenvalues for their auto-correlation matrices. Speech is not very well suited for LMS also because it is in general non-stationary and has a non-flat power spectrum. [8] This is why LMS algorithms do not perform optimally in many echo cancellation scenarios, even though they are simple to implement.

3.1.3 Time Domain Enhancement of LMS algorithms – Affine Projection

The sub-optimality of LMS-type algorithms regarding processing speech can be alleviated if a whitening process can be applied to the far-end input. This leads to several alternative classes of algorithms. One is a transform-domain LMS algorithm that projects input speech signals onto a complete and orthonormal basis, such as Fourier transformations, which we will discuss in the later section. The other is the so-called

Affine Projection algorithm, which causes the present far-end input to become approximately uncorrelated with past samples by successive projections.

Data that best resembles the current input but is yet uncorrelated with all past inputs would be the orthogonal component of the current input vector when it is projected onto the vector spaces generated by past input samples. We can subtract this projection from the current input to obtain this "innovation" component. This innovation process is the key to the Affine Projection (AP) algorithm.

In the AP algorithm, we have a N-element vector for the current N inputs, denoted by **x**[n] = [x[n], x[n-1], x[n-2], ..., x[n-N-1]]^T, and the N-element vector for the preceding input vector **x**[n-1] = [x[n-1], x[n-2], ..., x[n-N]]^T. Let **z**[n] denote the estimate of the component of **x**[n] that is orthogonal to the space generated by past inputs, in the following fashion,

$$\mathbf{z}[\mathrm{n}] = \mathbf{x}[\mathrm{n}] - \frac{\mathbf{x}[n]^T \mathbf{x}[n-1]}{\mathbf{x}[n-1]^T \mathbf{x}[n-1]} \mathbf{x}[n-1]$$

We can then use the **z**[n] as inputs to the NLMS algorithm instead of the **x**[n]. This approximation of whitening process, along with NLMS, determines the Affine Projection Algorithm, summarized in Table 4: [6]

Table 4: Affine Projection Algorithm

1) Initialization: $\mathbf{x}[0] = \mathbf{h}[0] = [0\ 0\ 0\ \ldots\ 0]^T$

Choose α in the range of $0 < \alpha < 2$ for convergence,
γ = small constant
Do for $n > 0$:

2) Update Error: $e[n] = d[n] - \mathbf{h}^T[n]\mathbf{x}[n]$
3) Innovation Process: $\mathbf{z}[n] = \mathbf{x}[n] - \dfrac{\mathbf{x}[n]^T\mathbf{x}[n-1]}{\mathbf{x}[n-1]^T\mathbf{x}[n-1]}\mathbf{x}[n-1]$
4) Update Filter: $\mathbf{h}[n+1] = \mathbf{h}[n] + \dfrac{\alpha e[n]\mathbf{z}[n]}{\gamma + \mathbf{x}^T[n]\mathbf{z}[n]}$

It has been shown that this algorithm has better performance than NLMS in acoustic echo cancellation [6], but at the expense of increased storage requirements and computational complexity, since we need to compute the projections.

An issue in this algorithm is the optimum choice of the step-size α. In order to achieve a faster rate of convergence α needs to be large, but large α leads to increased mean squared error. One proposal is to set α as a variable loop gain constant depending on past values of y[n] and e[n], as follows:

$$\alpha = \kappa \frac{\sum_{i=0}^{N-1} e^m[n-i]}{\sum_{i=0}^{N-1} y^m[n-i]}$$

where κ is a scaling factor. If m = 1, we are using the average value of the echo, and if m is 2, we are using the average echo power. For computational simplicity, m is usually chosen to be 1. This is the essential idea of the so-called Variable Loop Gain (VL) algorithms proposed by Yasukawa, Shimada, and Furukawa in 1987 [6]. This algorithm, unfortunately, did not outperform Affine Projection algorithm with a preset α in our simulation, as will be shown later.

3.1.4 Computation Complexity of Full-band Algorithms

An important issue to consider from an implementation point of view is the number of operations required per cycle to execute the algorithm. The computational complexity for each full-band algorithm per iteration are summarized below for an adaptive filter **h** with N taps:

Table 5.
Computational Complexity of Common Full-band Algorithms

Algorithm	**Number of Real Multiplications**	**Number of Real Additions**	**Number of Real Divisions**	**Total Number of Operations***
General LMS	2N + 1	2N + 1	0	4N + 2
NLMS	3N + 1	3N + 2	1	6N + 19
AP	6N + 1	6N + 2	2	12N + 35
VL (m = 1)	6N + 2	8N + 2	3	14N + 52

* Assuming that one 16-bit division takes 16 operations.

In terms of implementation, all of these algorithms except for general LMS are considered to require too many computations in the acoustic echo cancellation environment. For example, to cancel echo in a large vehicular compartment, the adaptive filter length should be no less than 512. For N = 512, general LMS takes about 16 MIPS (Millions of Instructions Per Second), NLMS takes about 25 MIPS, whereas AP requires 50 MIPS and VL requires 58 MIPS. For this project, we desire an echo canceller with robust performance with the total number of operations under 20 MIPS.

Many of the full-band algorithms in reality are therefore either unsuitable for practical implementation or do not yield high ERLE with speech input.

3.2 Frequency-Domain Subband and Block-Processing Algorithms

Consequently, we investigated two further possibilities: one is to adapt signals in blocks or groups to save computations, while the other is to whiten the input signal during preprocessing without affecting the statistical properties of the final output.

Two types of algorithms are therefore considered. One uses orthogonal transforms to project the input signal onto a new orthonormal basis, such as discrete Fourier Transform (DFT) or discrete cosine transform (DCT), and then applies transform-domain adaptive filtering for each component, or group of components. This method typically enhances ERLE and convergence rate compared to full-band algorithms. The second approach applies subband decomposition on input signals by using analysis filter banks and performs adaptation within each frequency band, and then forms the output through a synthesis filter bank. This class of algorithms is mainly advantageous in terms of savings in computation and optimizing convergence rate. We will explore transform-domain algorithms in Section 3.2.1 and 3.2.2, and subband approaches in Section 3.3.

3.2.1 Transform-Domain Adaptive Filtering

Transform-domain algorithms are mainly techniques to increase the rate of convergence of full-band algorithms when the input signal is highly correlated. The basic idea is to decorrelate the input vector before adaptive filtering. The full-band algorithm could become more effective if we could project the input speech signals onto an orthogonal basis first and then allow filters within each subspace to be adapted "almost" independently [10].

In transform-domain algorithms, the input signal vector **x**[n] is transformed into a more convenient vector **s**[n], by applying an orthonormal transform, i.e.

$$\mathbf{s}[n] = \mathbf{Tx}[n]$$

where $\mathbf{TT}^T = I$. As a result, the surface of MSE and the input correlation matrix, as described in the previous section, are rotated to a new set of axes after the transform. The rotation does not change the eigenvalue characteristics and spread of these surfaces. However, in this new setting, we can apply power normalization along each of the new principal axes to reduce the eigenvalue spread and hence reduce correlation among input signals. In this way, the update factor μ can be chosen independently in each subspace. The power normalization is performed in the following updating formula of LMS as an example, where the signal **s**[n] are normalized by their power, denoted as $\sigma_i^2[n]$: [8]

Table 6: Transform Domain LMS Algorithm

```
1) Initialization: $\mathbf{x}[0] = \mathbf{h}[0] = [0\ 0\ 0\ \ldots\ 0]^T$

Choose $\mu_n$ in the range of $0 < \mu < 2$ for convergence,
$\gamma$ = small constant,
$0 < \lambda < 0.1$
Do for n > 0:

2) Transform:     $\mathbf{s}[n] = \mathbf{Tx}[n]$, $\delta[n] = \mathbf{T}d[n]$

3) Update Error:  $e[n] = \delta[n] - \mathbf{h}^T[n]\mathbf{s}[n]$

4) Update Power:  $\sigma_i^2[n] = \lambda s_i^2[n] + (1 - \lambda)\sigma_i^2[n]$   for $0 < i < N$
5) Predict Filter: $h_i[n+1] = h_i[n] + \dfrac{\mu e[n] s_i[n]}{\gamma + \sigma_i^2[n]}$   for $0 < i < N$
```

$\sigma_i^2[n]$ here estimates the power for $s_i[n]$ and λ is the forgetting factor, which places weights on current input.

A number of real transform techniques are readily available, such as the discrete cosine transform (DCT) and discrete Hartley transform [3]. However, even though most of them can be computed via a fast algorithm or can be implemented in recursive frequency-domain format, the computational complexity of most of them is still unsuitable for practical implementation if executed every cycle. This is where block processing of signals in the frequency domain comes into play. The most efficient and readily obtainable transform is the discrete-time Fourier Transform (DFT), which we will use here.

3.2.2 Fast LMS Algorithms – Frequency-Domain Block Processing

A special case of transform-domain adaptive filtering is the fast LMS algorithm, (sometimes called the frequency-domain block LMS algorithm). This approach is a frequency-domain implementation of a block LMS algorithm, which updates filter coefficients for every block of input data, instead of on a sample-by-sample basis. The DFT, carried out via the Fast Fourier Transform (FFT), is used here for two purposes. First, the efficiency of the FFT reduces the complexity of linear convolution and correlation. Second, it allows an individual normalization of adaptation gains within each frequency bin, so as to optimize the rate of convergence.

Since the fast LMS algorithm is based on block LMS (BLMS) algorithm, it is necessary to briefly describe BLMS before understanding how we use the FFT for further computation simplicity. As we did in the time-domain analysis, let the far-end signal vector of length M be $\mathbf{x}[n] = [x[n], x[n-1], \ldots\ldots, x[n-M+1]]^T$, and $\mathbf{h}[k]$ denote a vector of filter coefficients of same length. Furthermore, let L be the block number, and define

$$n = kL + i, \qquad i = 0, 1, \ldots\ldots, M\text{-}1$$

It is apparent that the echo estimates y[n] can be calculated as

$$y[n] = \mathbf{h}[k]^T\mathbf{x}[n] = \mathbf{h}[k]^T\mathbf{x}[kL + i].$$

Let d[n] denote the near-end echo as reference signal. Then we can obtain e[n] as

$$e[n] = d[n] - y[n]$$

or

$$e[kL + i] = d[kL + i] - y[kL + i]$$

Thus, our input, reference signal, and error signal are all sectioned into blocks of L elements in a synchronous manner. For each block of data we use the M values of the error signal in adaptation by summing the product of **x**[kL + i]e[kL + i] over all possible values of i to obtain the following updating formula:

$$\mathbf{h}[n+1] = \mathbf{h}[n] + \mu\sum_{i=0}^{M-1}\mathbf{x}[kL+i]e[kL+i] \ = \mathbf{h}[n] + \mu\phi[n]$$

A popular choice of the block size M is to be the same as the number the adaptive filter taps.

Knowing the basics of the block LMS algorithm, we just need the following three basic facts to derive the fast LMS algorithm:

1) Circular convolution is linear convolution aliased in the time-domain.

2) Circular convolution in time domain yields multiplication in DFT domain.

3) Linear correlation is basically reversed from linear convolution in time domain.

First, we note that the echo estimate y[n] is obtained by linear convolution of **h**[k] and **x**[n]. Using fact 1) above, we can implement the linear convolution in terms of circular convolution using the overlap save method with 50 percent overlap. Then we can use fact 2) and apply the FFT to transform signals into the frequency domain.

Under this method, let the N-by-1 vector **H**[k] denote the FFT coefficients of the fifty-percent zero-padded length-M vector **h**[n], i.e. $\mathbf{H}[k] = \mathrm{FFT}\begin{bmatrix}\mathbf{h}[n] \\ \mathbf{0}\end{bmatrix}$, where **0** is a M-by-1 null vector. Note that the frequency-domain adaptive filter is twice as long as the filter in the time domain, i.e. N = 2M. Correspondingly, for the kth block define **X**[k] as the N-by-N diagonal matrix derived from x[n] as follows:

$$\mathbf{X}[k] = \mathrm{diag}\{\mathrm{FFT}[x[kM - M], \ldots, x[kM - 1], x[kM], \ldots, x[kM + M - 1]]\}$$

Note that the first half of the vector is the previous block (the (k-1)th block). Hence, by applying the overlap save method, we can find echo estimates y[n] as

$$\mathbf{y}[n] = [y[kM]\ y[kM + 1], \ldots, y[kM + M - 1]]^T = \text{last M elements of } \mathrm{IFFT}[\mathbf{X}[k]\mathbf{H}[k]]^T$$

where IFFT denotes Inverse FFT. Since the first M elements in the IFFT correspond to the aliased region in circular convolution, only the last M elements are obtained.

Consider next the linear correlation between x[n] and e[n] in the coefficient updating formula. For the kth block, define the M-by-1 reference vector

$$\mathbf{d}[n] = [d[kM], d[kM + 1], \ldots, d[kM + M - 1]]^T$$

and the corresponding M-by-1 error signal vector

$$\mathbf{e}[n] = [e[kM], e[kM + 1], \ldots, e[kM + M - 1]]^T = \mathbf{d}[n] - \mathbf{y}[n]$$

To transform the error signal vector into the frequency domain, we apply the FFT again as

$$\mathbf{E}[k] = \mathrm{FFT}\begin{bmatrix}\mathbf{0} \\ \mathbf{e}[n]\end{bmatrix}$$

Now we can use fact 3), recognizing that to obtain linear correlation we just need to use a "reversed" form of linear convolution, and get

$$\phi[n] = \text{first M elements of } \mathrm{IFFT}[\mathbf{U}^H[k]\mathbf{E}[k]]$$

Note that $\mathbf{U}^{\mathbf{H}}[\mathrm{k}]$, the complex conjugate transpose of $\mathbf{U}[\mathrm{k}]$, is used to account for the effect of "reversal" in time-domain. Since in linear convolution the first M elements are discarded, here we discard the last elements of the IFFT.

Finally, to update our filter coefficient in the Frequency domain, we may use the following updating formula:

$$\mathbf{H}[\mathrm{k}+1] = \mathbf{H}[\mathrm{k}] + \mu\mathrm{FFT}\begin{bmatrix}\phi[n]\\ \mathbf{0}\end{bmatrix}$$

One final step is to choose the adaptation gain. As mentioned in the previous section on transform-domain LMS, we need to normalize the step-size μ for each subspace here for each frequency component to reduce the spread of eigenvalues of the input autocorrelation matrix. Using the same procedure as the last section, here we define

$$\sigma_i^2[\mathrm{k}] = \lambda|X_i[\mathrm{k}]|^2 + (1-\lambda)\sigma_i^2[\mathrm{k}] \quad \text{for } 0 < i < N$$

and

$$\mathrm{P}[\mathrm{k}] = \mathrm{diag}[\sigma_0^{-2}[\mathrm{k}], \sigma_1^{-2}[\mathrm{k}], \ldots, \sigma_{2M-1}^{-2}[\mathrm{k}]]$$

Consequently, redefine

$$\phi[\mathrm{n}] = \text{first M elements of IFFT}[\mathrm{P}[\mathrm{k}]\mathbf{U}^{\mathrm{H}}[\mathrm{k}]\mathbf{E}[\mathrm{k}]];$$

this leads to

$$\mathbf{H}[\mathrm{k}+1] = \mathbf{H}[\mathrm{k}] + \mu\mathrm{FFT}\begin{bmatrix}\phi[n]\\ \mathbf{0}\end{bmatrix}.$$

The Fast LMS or so-called Complex LMS Algorithm is summarized in Table 7: [3][17]

Table 7: Fast Block LMS Algorithm

1) Initialization:	$\mathrm{H}[0]$ = 2M-by-1 null vector $\sigma_0^2[0] = 1$, and $\sigma_i^2[0] = 0$ for $i > 0$.
2) Update Error:	$\mathbf{X}[k] = \mathrm{diag}\{\mathrm{FFT}[x[kM - M], ..., x[kM - 1], x[kM], ..., x[kM + M - 1]]\}$ $\mathbf{y}[n]$ = last M elements of $\mathrm{IFFT}[\mathbf{X}[k]\mathbf{H}[k]]^T$ $\mathbf{e}[n] = \mathbf{d}[n] - \mathbf{y}[n]$ $\mathbf{E}[k] = \mathrm{FFT}\begin{bmatrix} \mathbf{0} \\ \mathbf{e}[n] \end{bmatrix}$
3) Calculate Power:	$\sigma_i^2[k] = \lambda\lvert X_i[k]\rvert^2 + (1 - \lambda)\sigma_i^2[k]$ for $0 < i < N$ $P[k] = \mathrm{diag}[\sigma_0^{-2}[k], \sigma_1^{-2}[k], ..., \sigma_{2M-1}^{-2}[k]]$
4) Predict Echo Path:	$\phi[n]$ = first M elements of $\mathrm{IFFT}[P[k]\mathbf{U}^H[k]\mathbf{E}[k]]$ $\mathbf{H}[k + 1] = \mathbf{H}[k] + \mu\mathrm{FFT}\begin{bmatrix} \phi[n] \\ \mathbf{0} \end{bmatrix}$

The computational complexity of Fast LMS can be compared with that of the standard LMS algorithms. For each block of M data, five FFT or IFFT procedures are used. If N is power of 2, each length-N FFT requires $N \log_2 N$ real multiplications and $N \log_2 N$ real additions, where in our case $N = 2M$. Also, computation of frequency domain output vectors require 4N real multiplications and 3N real additions, and so does the correlation output. In the power computation, we used 2 multiplications and 1 additions per block, as well as N divisions. Therefore the total computational complexity of the Fast LMS algorithm is:

Multiplication:	$5N \log_2 N + 8N + 2 = 10M \log_2 M + 26M + 2$
Addition:	$5N \log_2 N + 6N + 1 = 10M \log_2 M + 22M + 1$
Division:	$N = 2M$

The total number of operations is therefore $20M \log_2 M + 80M + 3$ per block of M input data, or $20 \log_2 M + 80 + 3/M$ per iteration. If $M = 512$, as in our case, this yields 2.1 MIPS, which is a great reduction compared to the conventional LMS algorithm, while its performance is better than full-band LMS under many circumstances, as we will see in the simulation section. The main disadvantage of this method, however, is that it diverges during the simulation in a low-SNR environment.

3.3 Time-Domain Subband Adaptive Filtering

This class of algorithms involves decomposing input speech or echo signals into disparate frequency subbands using analysis filter banks, applying decimation, and then performing LMS or other time-domain adaptive algorithms to cancel the echo within each subband. At the final stage, echo residuals from all frequency bands are combined together through synthesis filter banks and transmitted to the other end [4]. This scheme has been shown to be very useful in reducing computational complexity and enhancing convergence speed compared to the earlier echo cancellation techniques of Section 3.1.

In the following sections, we will analyze the loop structures and filter bank implementations for subband acoustic echo cancellers, as well as the advantages and disadvantages of this approach.

3.3.1 General Structure of Subband AEC

The basic structure of a subband acoustic echo canceller is shown in Figure 3 below [12]:

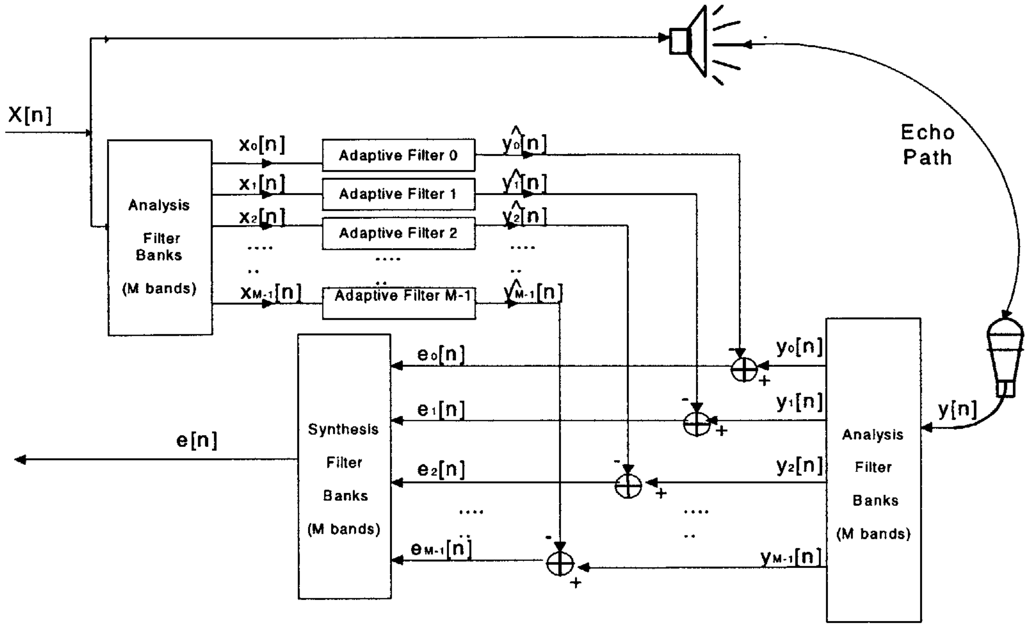

Figure 3. A Generic Subband Acoustic Echo Canceller

This prototype AEC works in the following fashion: the far-end speech x[n] is first divided into M subbands by the analysis filter bank and then decimated into each band to form $x_0[n]$ … $x_{M-1}[n]$. Subsequently, the signals in each subband go through an adaptive filter (normally LMS) that outputs echo estimates for the corresponding frequency band. At the same time, the input to the microphone, which consists of acoustic echo and near-end speech plus noise, also goes through the same analysis filter banks and decomposes into subbands in similar fashion. The echo estimate in each subband is then subtracted from these echo inputs. The resulting echo residuals are summed together through a synthesis filter bank and transmitted to the far end.

3.3.2 Filter Bank Design

A critical part of this subband structure is the design of analysis and synthesis filter banks. Since the echo residual will ultimately be transmitted to the far-end, the filter banks need to be perfect-reconstruction (PR), or near PR in nature. Using the vast development of multirate signal processing techniques starting in the 1980s, it is possible to use lossless (LL) PR filter banks for frequency-domain separation. Quadrature Mirror Filter (QMF) banks [4][7], uniform DFT banks [4] and discrete wavelet transforms (DWTs) [5][13] are among the most commonly used schemes. A two-band PR filter bank is presented below:

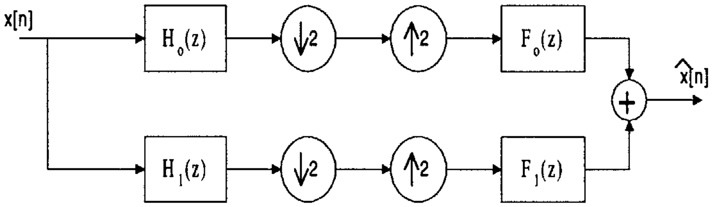

Figure 4. A Two Channel Multirate System

From the figure, we can derive a set of conditions for H(z) and F(z) so that $\hat{x}$[n] is a perfect reconstruction of x[n]. We know that

$$\hat{X}(z) = \frac{1}{2}(H_0(z)F_0(z) + H_1(z)F_1(z))X(z) + \frac{1}{2}(H_0(-z)F_0(z) + H_1(-z)F_1(z))X(-z)$$

For the system to be PR, we want $\hat{x}$[n] to differ from x[n] only by a pure delay, i.e. $\hat{X}(z) = z^{-l}X(z)$. The most important task is to eliminate aliasing within the system. The aliasing portion is the second part of above equation. Hence, we need to set

$$H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0$$

and subsequently

$$H_0(z)F_0(z) + H_1(z)F_1(z) = 2z^{-l}$$

At the same time, we also want to minimize the overlap of information between different frequency bands, so we would like to design a set of orthogonal filter banks as well, i.e. $<h_i[n]\ h_j[k]> = \delta[i-j]\delta[n-k]$.

From these sets of relationships, we can derive the following properties for two-channel PR filter banks [13]:

Table 8: PR FILTER BANK PROPERTIES

a) The filter length is even, L = 2k

b) The filters satisfy the power complementary or Smith-Barnwell condition:

$$\left|H_0(e^{j\omega})\right|^2 + \left|H_0(e^{j(\omega+\pi)})\right|^2 = 2,\ \left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2 = 2$$

c) The high-pass filter is specified (up to an even shift and sign change) by the low-pass filter as

$$H_1(z) = -z^{-2k+1} H_0(z^{-1})$$

d) To cancel aliasing, we can specify synthesis filters according to analysis filters as follows:

$$F_0(z) = z^{-(2k+1)} H_0(z^{-1})$$
$$F_1(z) = z^{-(2k+1)} H_1(z^{-1})$$

e) If the lowpass filter has a zero at π, that is, $H_0(-1) = 0$, then $G_0(1) = \sqrt{2}$.

From these properties, we see that as long as we can find a low-pass prototype for an analysis filter bank that satisfies property b), then all other filters are determined. There are a number of procedures to design $H_0(z)$. A common technique is spectral factorization, as used by Smith and Barnwell, as well as Daubechies' family of maximally flat filters that lead to a set of continuous wavelet basis. An alternative procedure is based on lattice structures as proposed by Vaidyanathan and Hoang [15]. Some of these techniques will be analyzed in the later algorithm design section.

To design a multi-channel PR system, there exist several techniques. One uses block and lapped orthogonal transforms, which leads to, for example, cosine-modulated or pseudo-QMF banks. Another method is to use tree-structured filter banks by cascading the two-channel filter banks, which leads to either spectrum division similar to that of the short-time Fourier transform, or a family of orthogonal bases such as wavelet packets. The latter is used in our multi-band echo canceller design due to its suitable non-uniform nature in analyzing speech signals, which we will discuss in Chapter 4.

3.3.3 Why Do Adaptive Filtering in Subbands

The subband echo canceller provides several major advantages over the full-band scheme. First of all, this structure is computationally more efficient. For a full-band algorithm that has computational complexity linear in N, this complexity can be reduced to $\frac{N}{M^2}$ if a uniform M-band subband implementation is used within a processor that allows parallel-processing. There are two reasons for the reduction in computations. First, signals in each band are decimated by a factor of M and thus only need to be processed at a rate F_s/M, where F_s is the sampling frequency. Second, the length of adaptive filter in the frequency band is reduced by a factor of M, due to the decimation.

Second, subband structure takes into account the frequency-dependent characteristics of speech signals and echo. This means that adaptation speed could be significantly enhanced by adjusting the adaptation gain in each band according to the energy within the band. Researchers have applied different adaptation constants to different bands both to save computations and to improve convergence characteristics [10]. At the same time, the length of adaptive filters in each subband can be matched to the average echo level in that band, which enables further reduction of computational burden while maintaining the overall performance of the echo canceller [6].

It has also been proposed and investigated that the subband structure may also help in auxiliary devices such as double-talk detectors, by using specific subband data instead of the full-band signal. This possibility was studied during simulation.

However, there are disadvantages to subband approaches as well. Since the frequency responses of the filters in analysis bank are overlapping, when critical downsampling is applied, the output of these filter banks may contain undesirable aliasing components that can impair the adaptation ability of the algorithm. Although the aliasing problem is in principle solved in PR systems, this is done only at the synthesis stage. Consequently, any operation performed in the transform domain would suffer from aliasing [7][9].

Many techniques have been proposed to resolve this problem of spectral overlap. Vetterli and Gilloire have introduced a full matrix of cross filters between the analysis and synthesis filter banks [12], Yasukawa, Shimada and Furukawa suggested using fewer overlapping filter banks [6], and Sumayajulu proposed an adaptive structure with decimated auxiliary subbands [9]. However, these methods become either computationally more intensive, or do not allow critical subsampling, or tend to introduce spectral holes in the output signals of the adaptive filter [9][10].

A promising approach has been the use of wavelet filters. Wavelets have been observed to reduce the correlation between subbands; they also allow the design of sharper cut-off filters, both FIR and IIR [4][5][13]. However, a problem associated with wavelets is the length of the filters. As the number of frequency bands increases, the order of the filters used in the filter bank increases dramatically [9].

IV. Algorithm Design and Development

As shown in the previous section, many of the existing echo-cancellation algorithms are either suboptimal or inelegant. The objective of our design is to incorporate the advantages of past designs while trying to avoid their inefficient aspects.

4.1 Overall Design of the Multi-band Acoustic Echo Canceller

Our design takes into account the advantages and disadvantages of existing full-band algorithms in a subband structure. After careful consideration and myriad simulations and tests, a non-uniform subband algorithm, the multi-band acoustic echo canceller, has been developed.

The design of this multi-band algorithm is based on traditional subband methods, which focus on optimizing convergence performance independently within each frequency band. However, we make two important improvements over existing subband algorithms.

First, unlike most subband algorithms which use uniform subband division, we use non-uniform analysis filter banks to take advantage of the non-uniform distribution of human speech energy. Although the frequency content of human speech ranges from 200 to 4000 Hz and above, most of the signal energy is concentrated at low frequencies. Therefore a multi-resolution analysis with more emphasis on low-frequency signals is desirable. On the other hand, experimental recordings inside vehicles show that under usual circumstances the vehicular compartment tends to amplify the low-frequency components of the echo and near-end speech. Therefore, instead of dividing the speech band into uniform subbands as in many traditional subband echo cancellers, we use non-uniform orthogonal filter banks (which eventually leads to a continuous-time wavelet basis) to perform subband coding. This technique is explained in Section 4.2.

Second, unlike all existing subband method that apply the same algorithm (normally LMS-type) to all frequency bands, our multi-band echo canceller applies different adaptation methods to distinct subbands according to the amount of echo in each band, since the orthogonality between frequency bands can be assumed. For example, for several critical subbands that have large amount of echo contents, such as the band between 300 and 1000 Hz, algorithms that yield high ERLE and rapid convergence are used at the expenses of increasing computational power and precision. For subbands that do not have much echo energy, such as between 3 and 4 KHz, we use only a conventional LMS algorithm, which does not require extensive computation but on the other hand does not yield rapid convergence. This parallel-processing scheme improves overall convergence performance without significantly increasing the computational complexity because of the decimations involved in subband division. The assignments of algorithms to different subbands are explained in Section 4.3, while the reduction in computations is presented in Section 4.5. Auxiliary devices such as noise filters are also important for the convergence behavior; these are discussed in section 4.4.

An overall design of the multi-band acoustic echo canceller with six frequency bands is shown in Figure 5:

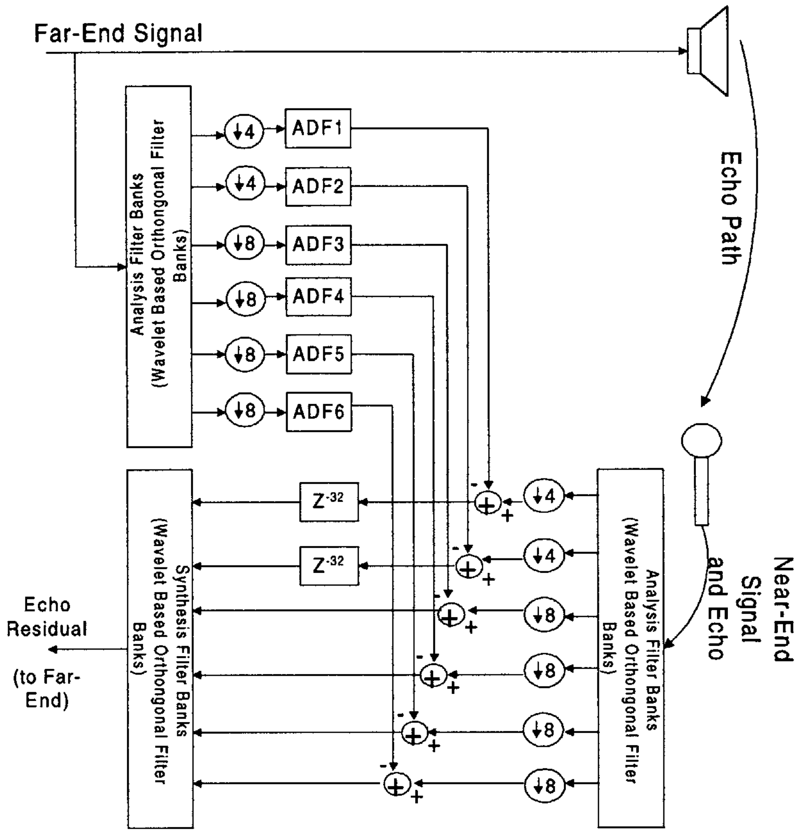
Far-End Signal
Echo Path
Analysis Filter Banks (Wavelet Based Orthongonal Filter Banks)
↓4
↓4
↓8
↓8
↓8
↓8
ADF1
ADF2
ADF3
ADF4
ADF5
ADF6
Near-End Signal and Echo
Analysis Filter Banks (Wavelet Based Orthongonal Filter Banks)
Synthesis Filter Banks (Wavelet Based Orthongonal Filter Banks)
Z-32
Z-32
Echo Residual
(to Far-End)
-
+

Figure 5. Overall Design of the Multi-band Acoustic Echo Canceller

4.2 Derivation of Wavelet-Based Perfect-Reconstruction Filter Banks

Examination of the spectrogram of more than a dozen male and female speakers, as well as their echo recorded in a vehicular environment, shows that both speech and echo has very little energy above 3.3 KHz; most of the energy is between 400 and 2000 Hz. Therefore, we chose the following six non-uniform bands: 0 - 0.5 KHz, 0.5 – 1 KHz, 1 – 1.5 KHz, 1.5 – 2 KHz, 2 – 3 KHz, and 3 - 4 KHz. The analysis filter banks are constructed by cascading two-channel PR systems in the fashion illustrated in Figure 6, with matching synthesis filter banks.

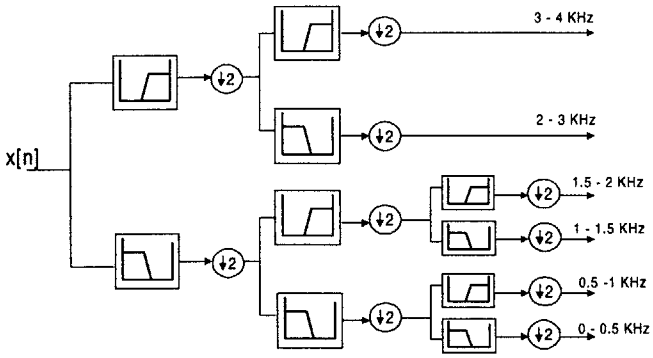

Figure 6. Tree-Structured Multi-Resolution Analysis Filter Bank Design

The design of the two-channel PR systems is based on the filter bank properties summarized in Section 2.5.2 and the spectral factorization techniques used in Daubechies' family of maximally flat filters.

To construct a length-2K filter, first define $P(z) = H_0(z)H_0(z^{-1})$, where $H_0(z)$ is a prototype low-pass filter. According to property (b) of Section 2.5.2,

$$P(z) + P(-z) = 2, \qquad (3.2.1)$$

Because Daubechies' objective was that the filters should correspond to continuous-time wavelet bases, the design procedure amounts to finding orthogonal low-pass filters with a maximum number of zeros at $\omega = \pi$. A length-2K low-pass filter consequently has to have an autocorrelation P(z) satisfying (3.2.1) and having maximum possible number of zeros at $\omega = \pi$. This leads to

$$P(z) = (1 + z^{-1})^k(1+z)^kR(z), \qquad (3.2.2)$$

where R(z) is symmetric, i.e. $R(z^{-1}) = R(z)$, and positive on the unit circle. In general we would like R(z) to have minimal order, i.e. R(z) has powers of 2 from $2^{(-k+1)}$ to $2^{(k-1)}$. R(z) can then be found by substitution into (3.2.1) and solution of a system of linear equations. Once the solution to this constrained problem is found, a spectral factorization of R(z) yields part of the desired filter $H_0(z)$, which automatically has k zeros at π. As always with spectral factorization there is a choice of taking zeros either inside or outside the unit circle. If we systematically take zeros from inside the unit circle, we obtain Daubechies' family of minimum-phase filters.

In our system, in order to create sharp-cutoff filters while not increasing computational complexity significantly, k is chosen to be 16. A length-32 filter was derived with the help of Matlab, and its frequency response is plotted below in Figure 7:

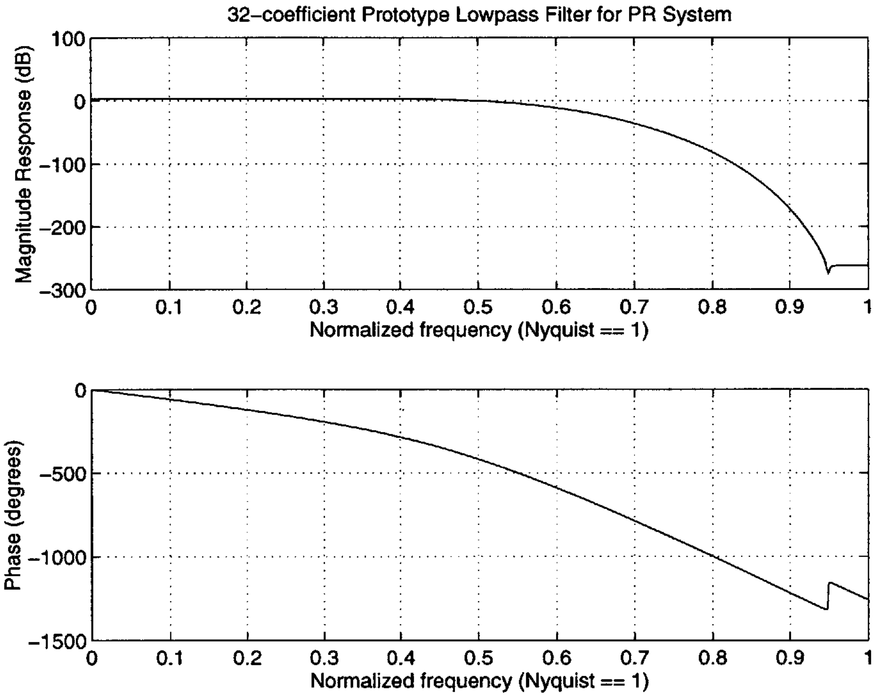

Figure 7. Frequency Response of Daubechies-Family PR Low-pass Filter

4.3 Assignment of Algorithms and Weights to Subbands

The design of Figure 5 has the advantage of optimizing echo cancellation performance and minimizing the required number of operations by assigning different adaptive filter schemes to different subbands according to amount of echo within each. The assignment of conventional adaptation algorithms to subbands is thus critical and is performed after careful examination of the spectrum of different speakers and frequency responses of the vehicle interior. A sample spectrogram of male and female speakers' echo is shown in Figure 8. The frequency response of a large vehicle (Cadillac) is shown in Figure 9.

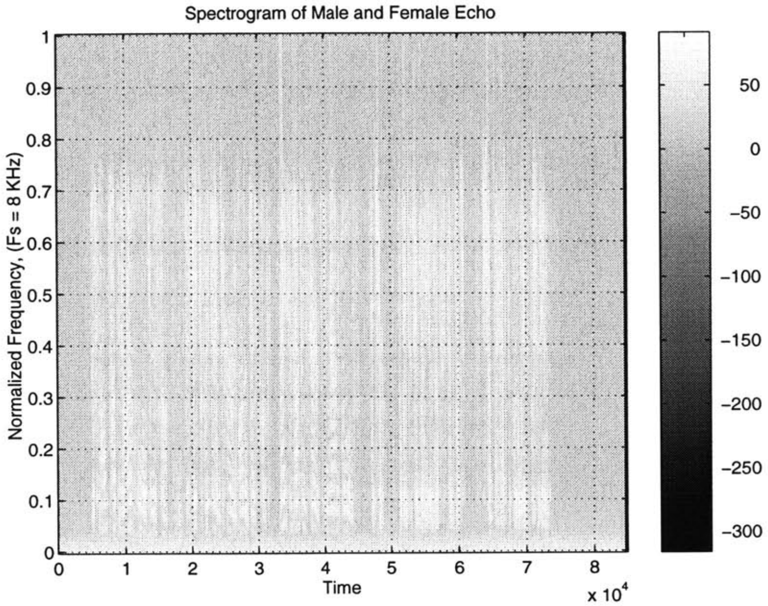

Figure 8. Spectrograms of Echo from Male and Female Speakers

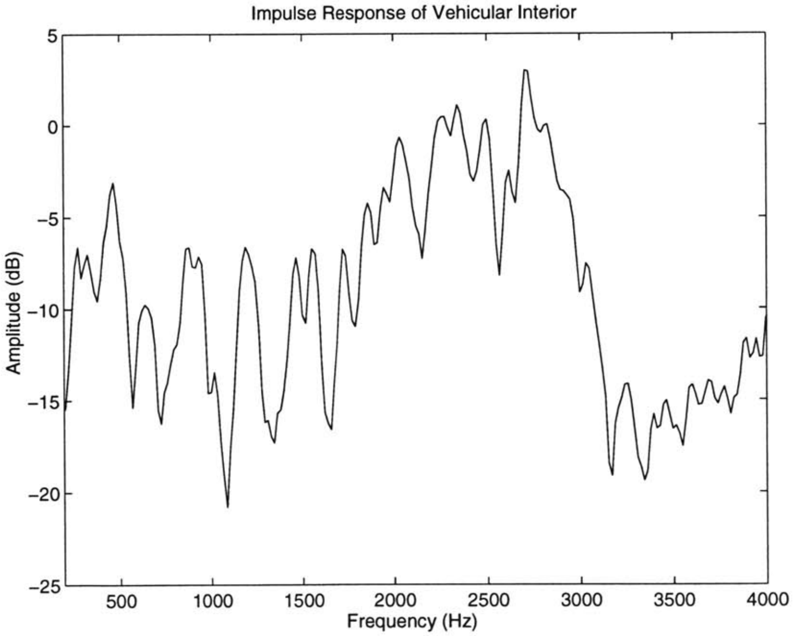

Figure 9. Frequency Response of a Large Vehicular Interior

We see from Figures 8 and 9 that most of the echo and speech energy is concentrated above 300 Hz and below 2 KHz. These two figures also represent echo characteristics and frequency responses of most large GM vehicles, which are typical settings for HNS vehicular hands-free geo-mobile phone. Therefore, the Affine Projection algorithm is assigned to these bands due to its superior performance in comparison to LMS, with the exception of 1-1.5 KHz, where echo energy is not as strong and therefore NLMS is applied. The band between 2 to 3 KHz not only contains a certain amount of echo, but also echo can be amplified by the vehicular environment, so an AP algorithm with reduced filter length was assigned to this band. Finally, for frequencies above 3 KHz, little echo or speech was present, and therefore the NLMS adaptive algorithm with reduced number of filter taps was applied to increase computational efficiency. The number of filter taps needed are determined based on the fact that 512 taps are required for full-band echo cancellers to effectively cancel the echo. Since each subband signal has been downsampled by different factors, the number of filter taps required in that band is then 512 divided by the decimation ratio, with the exception of the band above 3 KHz, in which little echo energy exists. The assignment of algorithms and filter lengths is shown in Table 9:

Table 9.
Algorithm Assignments to Frequency Bands

Frequency Band	Algorithm Used	Adaptive Filter Length
0 – 500 Hz	Affine Projection	64
500-1000Hz	Affine Projection	64
1000-1500 Hz	NLMS	64
1500-2000 Hz	Affine Projection	64
2000-3000 Hz	Affine Projection	128
3000-4000 Hz	NLMS	64

Another important practical issue here is the amount of noise present in each frequency band. Both AP and NLMS share the property of LMS-type algorithms that the misadjustment in MSE correlates inversely with the adaptation step-size (or the convergence factor) and the amount of noise in the reference signal (in this case the near-end speech and echo). Due to the frequency response of the vehicle and the nature of speech, the relative noise level in each subband is different. Therefore, it is desirable to have a different adaptation step in each band. For lower frequency bands, such as the two bands under 1 KHz, the relative noise level is usually high because engine noise is generally low-frequency. Consequently, a smaller step-size is assigned to these bands to compensate for the potential large misadjustment ratio caused by a possibly high noise-level. Thus, the convergence rate in these bands is a little slower than that of the higher frequency bands.

It may also seem from Table 10 that further dividing the frequency band between 2 and 3 KHz could lead to more reduction in computations. However, the problem of aliasing arises with increased frequency bands, which could lead to a significant degradation in the overall ERLE performance. Anti-aliasing filtering techniques in are described in the next section.

4.4 Further Refinements

So far, our acoustic echo canceller uses 6 bands with Nyquist sampling rate applied to each band. As mentioned in Section 2.5.3, with critical decimation, aliased versions of original and reference signals are generated in the subbands due to frequency overlaps of the PR filter banks, which causes degradation in the overall cancellation performance. This problem is addressed by introducing an auxiliary component that can help to adequately attenuate echo signals without too much computational complexity or near-end signal distortion.

The basic idea of this component is "filtering on demand." [17] When no near-end speech is detected, we relax our PR constraints by pre-filtering both far-end and echo signals with sharp band-stop (notch) filters to reduce the frequency components that overlap at band edges. When a near-end signal is detected, however, the pre-filters initially are removed allow near-end speech to be transmitted without distortion. A noise-elimination filter, namely is a high-pass filter with 400 Hz cutoff frequency, is also introduced to attenuate the noise level in low frequency spectrum.

This "filter on demand" scheme introduces drawbacks in practice, however. In a typical telephone conversation, when there is no near-end speech, an echo suppressor is turned on, which causes an additional 5-15 dB attenuation of near-end signal to ensure that minimum echo is transmitted to the far-end. When near-end speech is detected, the echo suppressor is turned off. This is the time when we really need flawless performance from our AEC. However, the removal of pre-filters at these times will undoubtedly weaken the echo cancellation performance.

It seems very difficult to highly attenuate echoes without distorting the near-end signal. As a compromise, pre-filters with less attenuation are introduced even when near-end talking is detected. The anti-noise low-pass filter introduced now has a cutoff at 350 Hz. Furthermore, not all band edges need notch filters. Aliasing is more significant for band edges at earlier dividing stages than later, due to the different resolutions involved. Therefore, notch filters are introduced only at 2 KHz and 3 KHz to avoid further degradation of PR requirements.

An common concern with a subband acoustic echo canceller is the long delay that is associated with it. In our case, the delay is approximately 96 samples because of the analysis and synthesis filter banks, which corresponds to 12 ms of delay. Since our device will be used in a geo-mobile system that has a one-way delay of 400 ms, the delay of our AEC is relatively insignificant.

4.5 Computational Complexity

The computational efficiency of this subband AEC has been enhanced greatly as a result of decimation and parallel processing. The total number of operations required is shown in Tables 2 and 3. For a band with an N-tap AP adaptive filter, the required number of computations is 6N+1 multiplications, 6N + 2 additions, and 2 divisions (16 operations per division). Likewise, for a band using an N-tap NLMS adaptive filter, 3N + 1 multiplications, 3N + 2 additions, and 1 division are required. The total MIPs required for each band is summarized in Table 10 below:

Table 10.
Number of Operations Needed in Each Subband

Frequency Band	Algorithm Used	MIPs Required
0 – 500 Hz	Affine Projection	0.8
500-1000Hz	Affine Projection	0.8
1000-1500 Hz	NLMS	0.4
1500-2000 Hz	Affine Projection	0.8
2000-3000 Hz	Affine Projection	3.14
3000-4000 Hz	NLMS	0.8

This leads to about 6.74 total MIPS required for adaptive filtering. We also need to take into account the computations involved in the filter banks. Each filter has 32 coefficients, which leads to 32 multiplications and 32 additions per filter cycle. We have a total of 2 filters working every cycle, 4 filters working every two cycles, and 4 filters working every 4 cycles; therefore the number of MIP required for the filter banks is

$$(2 \times (32 + 32) + 4 \times (32 + 32) / 2 + 4 \times (32 + 32) / 4) \times 8000/1000000 = 2.56 \text{ MIPs}$$

Therefore, the total number of operations needed for our subband AEC is 9.3 MIPS, which is a significant enhancement in comparison to any of the full-band algorithms described previously, and is yet more robust than frequency-domain algorithms during simulation, as we will see in the next section.

V. Simulation Development and Results

Our simulations were programmed in the C language and performed in two stages. In the first stage, we evaluated the algorithms for suitability for implementation on a Texas Instrument DSP. In the second stage, we further optimized the chosen algorithms in terms of convergence performance and computation reduction.

5.1 Simulation Stage One – Predecessor Algorithms

The initial simulation considered seven different algorithms: NLMS, AP, variable-loop gain (VA), Fast LMS, Polyphase-Based Adaptive Structure [9], 2-band subband algorithm, and a Frequency domain AP algorithm, which is just an implementation of the Affine Projection algorithm using DFT via the same principles as used in the derivation of the fast LMS algorithm. The variable loop gain algorithm was described in chapter 3, and Polyphase-Based Adaptive Structure is developed and simulated because of its computational efficiency. Its detail is described in [9]. The test results of these algorithms run in different scenarios were used for performance comparison.

The first category of tests is the ideal case, in which the echo path is simulated as a pure delay scaled by a gain factor (i.e. with no echo reverberation involved.) Real speech signals are used as test input. In this case, almost all algorithms have superior performance, as expected. The ERLE performance curve for the seven algorithms is plotted in Figure 10:

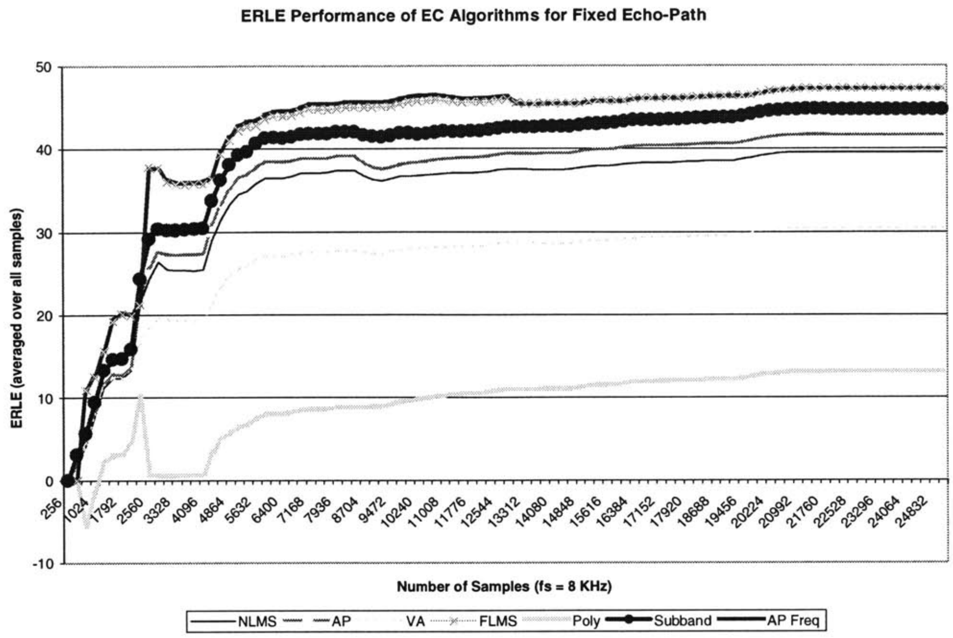

Figure 10: ERLE Performance Curves of Seven Algorithms with Fixed-Path Echo

The next step challenges the algorithms a little. A longer LTI filter with 256 coefficients representing echo reverberation process has been added into the echo path. In this scenario, the performance for most algorithms drops from the ideal case: some demonstrate more robustness than others, however. The strong candidates in this case are AP, subband, and frequency-domain LMS and AP. The performance curves for algorithms canceling long echo are shown in Figure 11:

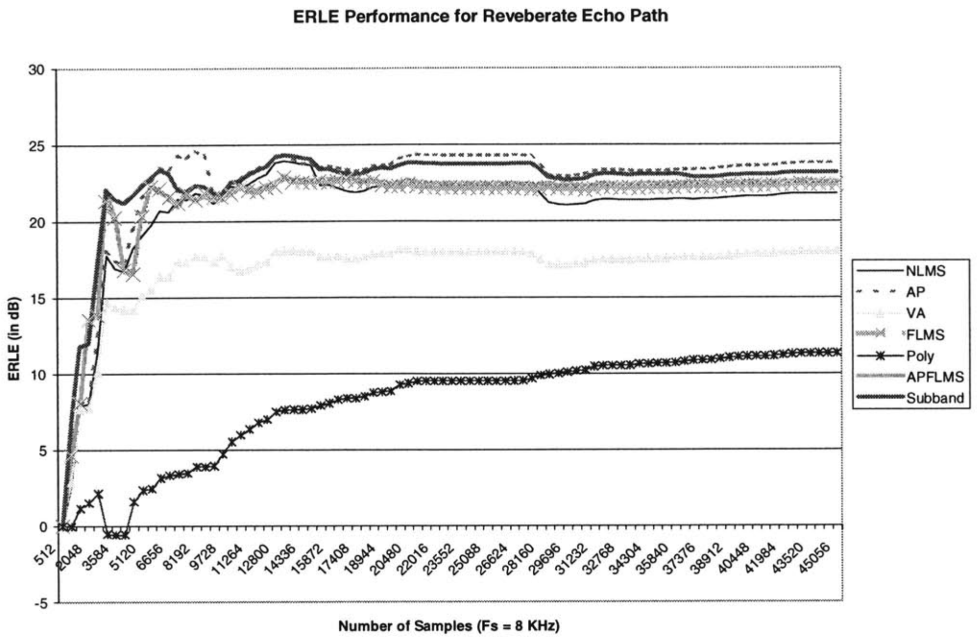

Figure 11: ERLE Performance Reverberate Echo Path

What really differentiates the algorithms is the third simulation step, during which field data recorded in actual vehicles are used. The car recording was done in both a medium-sized car (Toyota Corolla Wagon) and a large vehicle (Chevrolet Lumina Van). Speech recordings were broadcast from a typical speaker for a hands-free phone, and echo was recorded via both omni-directional and noise-canceling microphone. Different cases, such as closed/open window, engine turned on/off, and presence/absence of near-end speech, were used in different recordings. In many of these scenarios, the frequency-domain algorithms suffer from severe instability. Due to limited time, the causes have not been investigated thoroughly.

As a result, we have narrowed our choices to NLMS, Affine Projection, and subband approach. Affine Projection initially demonstrated the best performances, with 2-band subband following second. The ERLE performance of AP algorithm is depicted below:

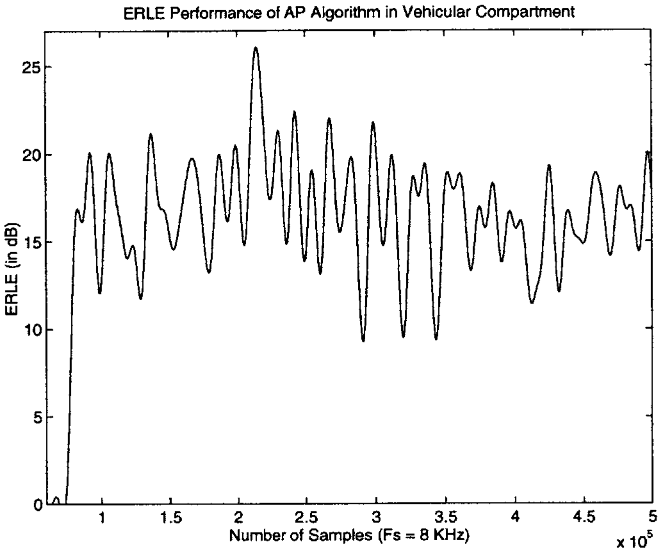

Figure 12. Performance of AP Algorithm in Lumina Van

5.2 Simulation Stage Two – Multi-band Acoustic Echo Canceller

The choice now remains between Affine Projection and Subband schemes. The AP algorithm indeed performs better than the 2-band subband algorithm. However, the amount of computations involved in AP algorithm, as demonstrated in Section 2.3.4, is very large. It is thus unsuitable for a DSP implementation, in which we need to keep the total operations under 20 MIPS. This makes the subband approach our final algorithmic choice.

Our final design in Figure 5 arrives at last after optimization and testing in three main scenarios: 1) white noise input, 2) realistic vehicular settings with minimal noise, and 3) noisy vehicular environment. Its ERLE performances in these scenarios are discussed in the following sections.

5.2.1. White Noise

The behavior of LMS-type algorithms is nearly optimal when the input is white, since the orthogonality of direct inputs to the adaptive filter is already present. It has also been asserted in some literature that subband filtering is not only redundant but may also cause degradation for cases in which the input is already white noise. This may well be true. Due to the non-ideal characteristics of the filter banks, small correlations will be present in each subband and between overlapping bands, even if the input is white. This is usually reflected by a degradation in performance of the subband algorithms compared to that of the LMS algorithms. This assertion is confirmed by the ERLE performance curves of our subband AEC, in comparison with LMS and AP schemes, as plotted below:

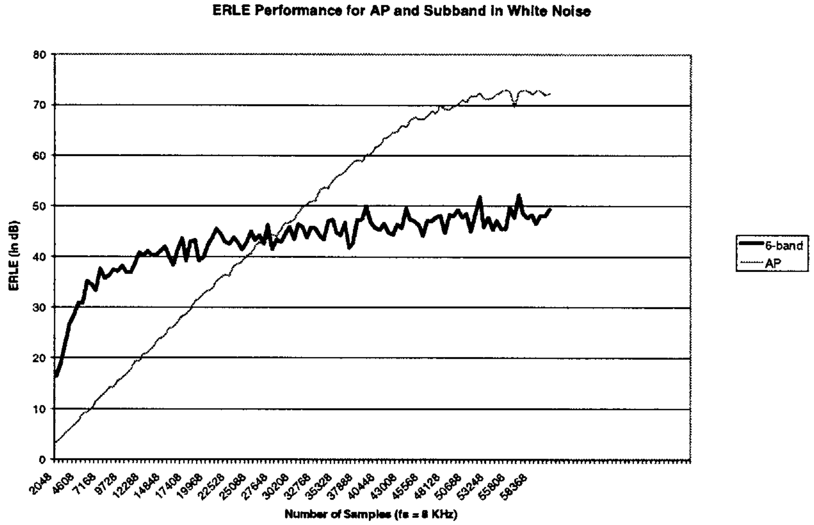

Figure 13: Algorithmic Performance in White Noise

However, one thing we should note here is that the adaptation speed of multi-band algorithm is much quicker than that of Affine Projection method.

5.2.2 Vehicular Environment

The performance of the 6-band echo canceller is significantly better than Affine Projection algorithm and LMS-type algorithms when colored inputs and time-variant situations are encountered. Vehicular recordings of real speech and echo were used as test data to determine this result. The ERLE performance of subband algorithm in the Chevrolet Lumina van with window closed and engine turned off (same environment as the AP algorithm plotted in Figure 12) is presented below:

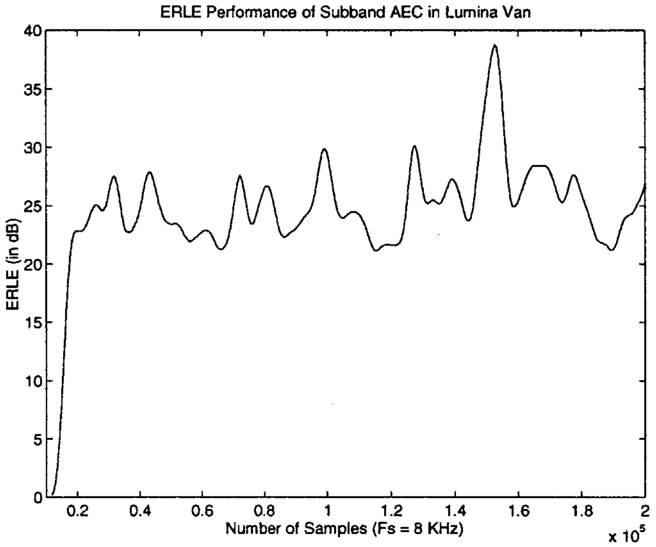

Figure 14: Subband AEC Single-Talk ERLE Performance in Vehicle with Engine off

We should also be aware that a vehicular speakerphone is usually used in a running car. In this scenario, the presence of engine noise could severely degrade the adaptation process of an LMS-type algorithm. Our subband echo canceller, however, is much more immune to engine noise, which is mainly low-frequency in nature. In higher-frequency bands, the noise level is not very high, and thus echo cancellation is unlikely to be affected. In lower-frequency bands where the noise is concentrated, the anti-noise filters significantly reduce the noise level and thus retain a similar quality of convergence and MSE misadjustment performance.

When engine is turned on, the AP algorithm performs 5-10 dB worse than it does in engine off scenario, while ERLE performance of the subband algorithm with engine turned on is plotted below; we see that it is actually better.

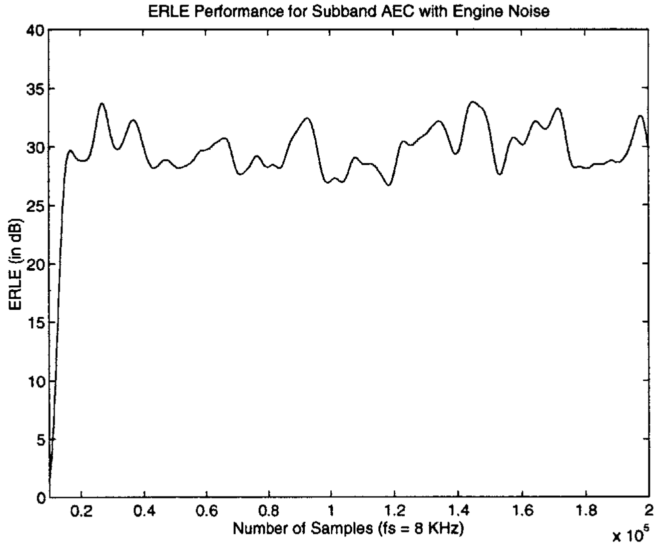

Figure 15: Subband ERLE Performance with Engine on

VI. Near-End Speech Detector

6.1 Function of Near-End Speech Detector

So far, we have investigated and developed an acoustic echo canceller under the assumption that near-end speech is absent. Unfortunately, this assumption is not true for full-duplex transmission. Once near-end speech is present, such as in the case of double-talk, during which both near-end and far-end speakers are simultaneously talking, serious degradation to echo canceller performance and possible divergence of adaptive filter coefficients can occur. We thus need to add an important auxiliary device to our system. This is the near-end speech detector, or as it is commonly called, a double-talk detector.

If the echo canceller keeps updating its filter coefficients during double-talk periods, the coefficients of the adaptive filter will start to vary sharply and may eventually diverge. Therefore, once near-end speech is detected, the algorithms must stop updating the coefficients of the adaptive filters. Also, in the absence of a near-end talker, echo suppressors are usually turned on to further suppress the transmission of echoes to the far end. The echo suppressor should be turned off, however, when near-end speech is detected.

A functional block diagram of a double-talk detector is depicted below: [1]

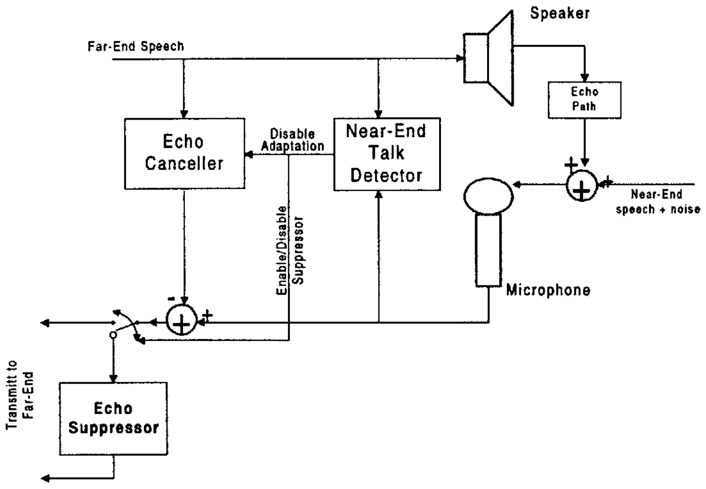

Figure 16. Near-end Talker Detector Functional Block Diagram

6.2 Traditional Double-Talk Detection Methods

The essential requirement for a double-talk detector is to detect the occurrence of double-talk quickly and accurately. At the same time, it should have the ability to distinguish double-talk conditions from echo path variations.

A traditional detection scheme is a comparison of the power of the near-end signal y[n] with that of the far-end signal x[n] [2][11]. If the near-end power exceeds the far-end power by certain threshold, then a double-talk situation is declared, with a certain holding time. This approach has two problems. First, the threshold is difficult to set for the hands-free phone scenario, since it is very dependent on the echo-path loss between microphone and speaker [21]. Second, the power of the near-end signal increases during both echo-path variation and double-talk [22]. A power comparison may be insufficient to distinguish these two situations.

An alternative detection method is based on principle of orthogonality for LLSE estimation [22]. Since the far-end signal vector **x**[n] is essentially an input to a least-squares estimation process, it should be ideally uncorrelated with the cancellation error signal e[n], i.e. E[e[n]**x**[n]] = 0, if we assume the adaptive filter has converged close to its MSE solution. At this time, even if the existence of double-talk increases the near-end signal powers, the cross-correlation should still be very small. The cross-correlation increases only during an echo-path variation. This scheme therefore updates the adaptive filter coefficients only if the cross-correlation between far-end and residual echo is greater than a certain threshold, which indicates that the cancellation filter is sub-optimum. This approach detects double-talk more accurately than the power comparison method, since it prevents the echo canceller being perturbed by double-talk interference after it has converged. However, this method is still sub-optimal, since the performance of the adaptive filter can severely degrade if an echo path variation occurs during double-talk.

A number of other detectors can be found in the literatures [11], [19], [20], [21]. These approaches vary from comparing the power spectrum density (PSD) of near-end and far-end signals, to comparing the ERLE of the echo canceller within a specific band, to statistical fuzzy-logic operations. However, these detectors are either not very robust in a low-SNR environment, or are computationally complex and therefore unsuitable for implementation.

6.3 Double-Talk Detector Design

The double-talk detector designed in this project is based on the Neyman-Pearson criterion for binary hypothesis testing. It aims to maximize the probability of detection while maintaining the probability of false alarm under a desirable threshold for a designed test. The test uses the relationship between three estimated parameters to determine whether an incoming near-end signal consists of near-end speech:

1) $C_{y\hat{y}}[n]$: the cross-correlation coefficient between the near-end signal y[n] and echo estimate $\hat{y}[n]$.

2) $C_{ye}[n]$: the cross-correlation coefficient between y[n] and the residual error e[n].

3) $C_{xe}[n]$: the cross-correlation coefficient between the far-end signal x[n] and e[n].

These three correlation coefficients are estimated as follows using exponential weighting recursive equations: [21][22]

1) $$C_{y\hat{y}} = \frac{P_{yh}[n]}{\sqrt{P_y[n]P_h[n]}}$$

where

$$P_y[n] = \lambda P_y[n-1] + (1-\lambda)\, y^2[n]$$
$$P_h[n] = \lambda P_h[n-1] + (1-\lambda)\, \hat{y}^2[n]$$
$$P_{yh}[n] = \lambda P_{yh}[n-1] + (1-\lambda)\, y[n]\,\hat{y}[n]$$

2) $$C_{ye} = \frac{P_{ye}[n]}{\sqrt{P_y[n]P_e[n]}}$$

where

$$P_y[n] = \lambda P_y[n-1] + (1-\lambda)\, y^2[n]$$
$$P_e[n] = \lambda P_e[n-1] + (1-\lambda)\, e^2[n]$$
$$P_{ye}[n] = \lambda P_{ye}[n-1] + (1-\lambda)\, y[n]e[n]$$

3) $$C_{xe} = \frac{P_{xe}[n]}{\sqrt{P_x[n]P_e[n]}}$$

where

$$P_x[n] = \lambda P_x[n-1] + (1-\lambda)\, x^2[n]$$
$$P_e[n] = \lambda P_e[n-1] + (1-\lambda)\, e^2[n]$$
$$P_{xe}[n] = \lambda P_{xe}[n-1] + (1-\lambda)\, x[n]e[n]$$

The use of these three measurements is to clearly distinguish the difference between double-talk and echo-path variation. The basic justification is the following three observations, assuming a relatively high-SNR environment and approximately uncorrelatedness between near-end speech and far-end speech:

1) If the adaptive filter converges close to its MSE solution, then $C_{y\hat{y}}$ will be very large, while C_{xe} will be close to zero [22].

2) If there is no echo path variation, then the presence of near-end speech should increase C_{ye} and decrease $C_{y\hat{y}}$, but should not significantly affect C_{xe} [21].

3) During an echo-path variation, C_{xe} and C_{ye} will be very large, while $C_{y\hat{y}}$ will be small. The presence of near-end speech at this time will decrease C_{xe} and $C_{y\hat{y}}$ while increasing C_{ye}. This is the key observation that can help us detect double-talk during echo-path variation.

Using these three observations, a set of decision rules was established based on comparison of the three parameters against different thresholds under different scenarios. Near-end speech is declared if these threshold levels are exceeded. The detector declares near-end speech by setting a near-end-alert flag in the program. The decision is made in the following steps:

1) Checks to see if the near-end-alert flag is set presently.

2) If the near-end-alert flag is not set, check whether $C_{y\hat{y}} < A$ and $C_{ye} > B$. If this is true, check to see if $C_{xe} < D$, if so set near-end-alert flag, where A, B, D are certain thresholds. Otherwise, don't do anything.

3) If the near-end-alert flag is presently set, then check if $C_{y\hat{y}}$ > E and C_{ye} < F, where E and F are also thresholds. If this is true, clear the near-end-alert flag. Otherwise check if C_{xe}> G. If so, clear the near-end-alert flag. Otherwise nothing is changed.

The thresholds A, B, D, E, F, G were modified and optimized after many simulations and trials. The thresholds all range between 0 and 1.

Since double-talk detection has to take place before the echo canceller updates its coefficients, a separate LMS detector with reduced length is used solely to calculate $\hat{y}$ [n] and e[n]. In this way the main adaptive filter will is likely to be perturbed by the first group of samples from the near-end speech.

The overall block diagram of the double-talk detector design is depicted below:

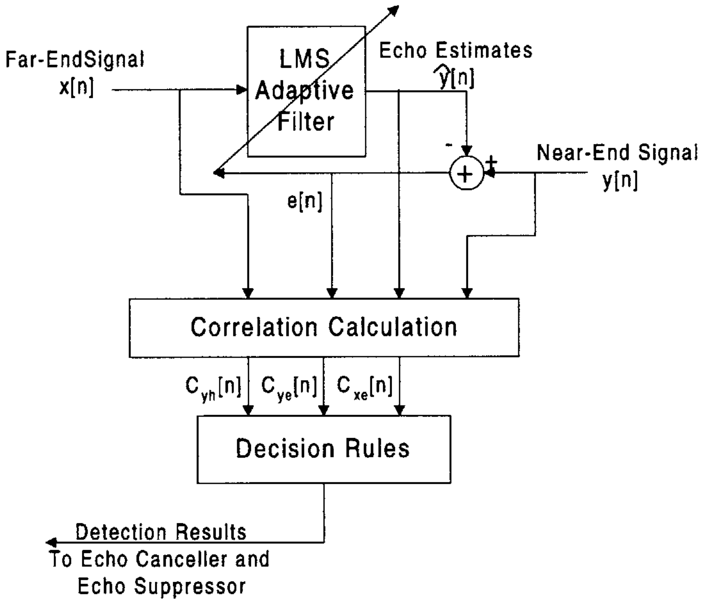

Figure 17. Block Diagram of Double-Talk Detector

6.4 Detection Performance

Two sets of tests were given to test the robustness of the double-talk detector. The first set of tests consists of recordings of actual telephone conversations, during which the energy of the near-end speech input is equivalent or greater than the transmitted far-end speech. In this case, the probability of detection is ~100%, and the probability of false alarm (detecting near-end speech when it's not there) is ~ 15%. We observed that false alarms happen almost always in cases when both near-end speech and far-end speech are absent. This will not degrade echo cancellation performance.

The second set of test cases correspond more closely to the worst case of real vehicle speaker-phone conversation scenarios, during which the energy of near-end speech input is sometimes weaker than far-end speech and echo path variation occurs frequently. Simulation of several traditional double-talk detection schemes showed that a significant portion of this type of speech is not detected, and sometimes the near-end speech caused divergence in the filter coefficients. Our double-talk detector, however, exhibited outstanding robustness for this case. Two-minute speech files containing continuous double-talk in this scenario were used as test data. More than 90% of the double-talk was detected, and the missed portion occurred mostly during either extremely weak speech signals that are comparable to background noise, or during nulls in near-end speech. Neither of these events will cause echo canceller coefficients to diverge. The probability of false alarm increased a little from the previous case to 23%. However, as before, most of the false alarms involve situations where there were nulls in far-end speech. Vestiges of undetected echo are attenuated at least 20 dB by the echo canceller before transmission.

Finally, it is also important to make sure that the addition of the double-talk detector does not degrade the ERLE performance of our echo canceller, especially during the case of double-talk. The ERLE performance during a section of the worst-case double talk scenario is depicted below; it shows little deviation from the ERLE level during single talk case.

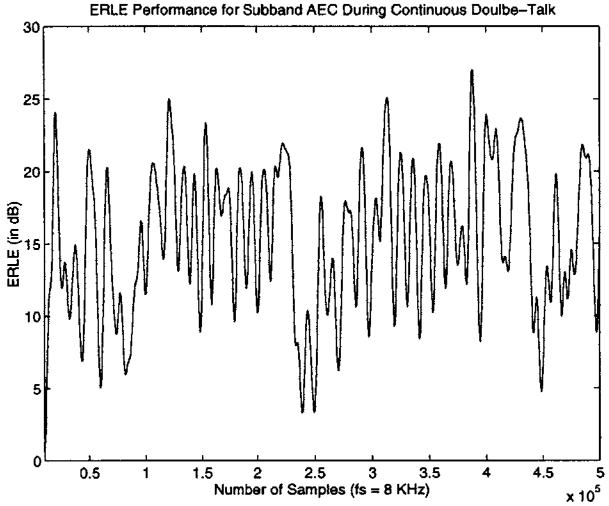

Figure 18: AEC ERLE Performance during Long Double Talk

VII. Further Development

7.1 DSP Implementation

Simulations are not enough to truly verify the robustness of our acoustic echo canceller. It is necessary to implement the algorithm on a DSP processor to prove its performance. The multi-band algorithm and correlation-based double-talk detector are required to be implemented on a fixed point DSP chip (TI TMS320C54X).

This implementation was a major challenge to the algorithm because quantization noise occurs when converting from floating-point to fixed-point arithmetic. Since an inappropriately chosen quantization scale could seriously degrade the echo canceller performance and cause arithmetic overflow, all intermediate-stage calculations are carefully scaled and rounded to an appropriate level. The overall ERLE degradation of the fixed-point DSP compared to the floating-point simulation for the multi-band algorithm has been successfully kept within 1 to 3 dB.

The successful DSP implementation has shown the practicality of our multi-band acoustic echo canceller. The main improvement that need to be made in the future is to further reduce the overall number of operations and MIPs (from the current 20 MIPs, including both echo cancellation and double-talk detector) so as to further decrease the cost of implementation.

7.2 Suggestions for Further Research

Due to limited time, the ERLE performance of this acoustic echo canceller, though better than many commercial ones, is still not optimal. It could be further enhanced to 30 ~ 35 dB by more frequency-band division and appropriate choice of subband algorithms and noise filters.

Less distortion of near-end speech can also be achieved. A possible method for improvement is the post-filtering process, as mentioned in a number of papers.

Also, instead of dividing near-end input into subbands, we could divide residual error signal into subbands instead, as several papers have suggested [16], though synchronization of far-end signal and echo residuals will be more difficult in this case.

Another possible improvement is the method of assigning algorithms to frequency bands. Currently, the adaptation algorithm in each subband is pre-determined subjectively by examining spectrograms of speakers and their echo in a vehicular environment. The echo canceller would work more effectively if the algorithm assignments were made adaptively according to the individual characteristics of the near-end speech and environment. For instance, all subband algorithms could start with LMS, and then after several seconds of conversation some bands could change to the Affine Projection adaptation algorithm if the echo concentration exceeds a certain threshold. Specific methods for determining echo concentration and thresholds need to be developed.

Furthermore, a number of alternative echo canceller structures could be explored for geo-mobile communication systems. First, frequency-domain algorithms require very little computation and yet yield sound ERLE performance. The drawback is that some of these algorithms suffer from instability at low SNRs. It is therefore necessary to declare a scheme that prevents this unwanted instability. Second, we have restricted our full-band algorithm choices to gradient-based algorithms for reasons of computational complexity. It would also be beneficial to explore other families of adaptive filters such as the recursive least square (RLS) approach [3] and fast Kalman algorithms. These methods are currently computationally unsuitable for DSP implementation, but they do yield better performance than LMS-type algorithms.

For double-talk detectors, further refinement of the thresholds in our decision rules may lead to possible improvements in false-alarm rate. Instead of using strict threshold levels, a simpler fuzzy-logic based detection mechanism, as suggested in [21], could also be tried here.

VIII. Conclusion

A complete multi-band acoustic echo canceller design has been presented. It takes advantage of the approximate orthogonality between frequency bands, and applies appropriate and different algorithms to each band in order to minimize the computational burden and maximize echo cancellation performance. Its performance has been tested through many simulations, and has been shown to surpass that of many existing echo cancellation algorithms. A double-talk detector based on correlation, which is different from conventional methods, has also been designed and simulated, with satisfactory performance. These designs have been modified from floating-point to fixed-point arithmetic for DSP implementation and real-time testing. More advanced methods for selecting subband algorithms and prefiltering techniques, as well as the possibility to use frequency-domain algorithms, have been suggested for future research.

Patent applications are in process for the multi-band algorithm and the double-talk detector.

References

[1] Hsu, W., D.A. Hodges, and D. G. Messerschmitt. "Acoustic Echo Cancellation for Loudspeaker Telephones," in *GLOBECOM '87 Conf. Rec.*, Nov. 1987, pp. 1955-1959.

[2] Hsu, W., F. Chui, and D. A. Hodges. "An Acoustic Echo Canceller," *IEEE Journal of Solid-state Circuits*, Vol. 24, No. 6, December 1989, pp.1639-1646.

[3] Haykin, Simon. *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ (3rd Edition, 1996).

[4] Vaidyanathan, P.P. *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, NJ, 1993.

[5] Chan, Y. T. *Wavelet Basics*, Kluwer Academic Publishers, Norwell, MA, 1996.

[6] Yasukawa, H., S. Shimada, and I. Furukawa. "Acoustic Echo Canceller with High Speech Quality," *Proc. Intl. Conf. Acoust. Speech. Signal Process.*, Dallas, TX, April, 1987, vol. 4, pp. 2125-2128.

[7] Gilloire, A. "Experiments with Subband Acoustic Echo Cancellers for Teleconference," *Proc. Intl. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, April, 1987, vol. 4, pp. 2141-2144.

[8] Chen, J. D. Z., H. Bes, J. Vanderwalle, I. Evers, and P. Janssens. "A Zero-Delay FFT-Based Subband Acoustic Echo Canceller for Tele-conferencing and Hands-Free Telephone Systems," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 43, October 1996, pp. 713-717.

[9] Iyer, U., M. Nayeri, and H. Ochi, "Polyphase Based Adaptive Structure for Adaptive Filtering and Tracking," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 43, March 1996, pp. 220-233.

[10] de Courville, M., and P. Duhamel, "Adaptive Filter in Subbands Using a Weighted Criterion," *IEEE Transactions on Signal Processing*, Vol. 46, September, 1998, pp. 2359-2371.

[11] Gansler, T., M. Hansson, C. Ivarsson, and G. Salomonsson, "A Double-Talk Detector Based on Coherence," *IEEE Transactions on Communications*, Vol. 44, November 1996, pp. 1421-1427.

[12] Gilloire, A. and M. Vetterli, "Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments, and Application to Acoustic Echo Cancellation," *IEEE Transactions on Signal Processing*, Vol. 40, August, 1992, pp. 1862-1875.

[13] Vetterli, M. and C. Herley, "Wavelets and Filter Banks: Theory and Design," *IEEE Transactions on Signal Processing*, Vol. 40, September 1992, pp. 2207-2232.

[14] Farhang-Boroujeny, B., "Fast LMS/Newton Algorithms Based on Autoregressive Modeling and Their Application to Acoustic Echo Cancellation," *IEEE Transactions on Signal Processing*, Vol. 45, August 1997, pp. 1987-2000.

[15] Vetterli, M. and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[16] Morgan, D. R. and J. C. Thi, "A Delayless Subband Adaptive Filter Architecture," *IEEE Transactions on Signal Processing*, Vol. 43, August 1995, pp. 1819-1830.

[17] Amano, F., H. P. Meana, A. de Luca, and G. Duchen, "A Multirate Acoustic Echo Canceller Structure," *IEEE Transactions on Communications*, Vol. 43, July 1995, pp. 2172-2176.

[18] Tahernezhadi, M., S. Manapragada, J. Liu, and G. Miller, "A Subband-Based Acoustic Echo Canceller for a Hands-Free Telephone," *IEEE 45th Vehicular Technology Conference*, Vol. 2, 1995, pp. 728-732.

[19] Carlemalm, C. and A. Logothetis, "On Detection of Double Talk and Changes in the Echo Path Using a Markov Modulated Channel Model," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, 1997, pp. 3869-3872.

[20] Carlemalm, C., F. Gustafsson, and B. Wahlberg, "On the Problem of Detection and Discrimination of Double Talk and Change in Echo Path," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, 1996, pp. 2742-2745.

[21] Ryu, G., D. Kim, J. Choe, D. Kim, S. Kim, H. Bae, B. Yuan, and X. Tang, "Double Talk Detection in Adaptive Echo Canceller Using the Fuzzy Logic," *Third International Conference on Signal Processing*, Vol. 2, 1996, pp. 1643-1646.

[22] Ye, H. and B. Wu, "A New Double Talk Detection Algorithm Based on the Orthogonality Theorem," *IEEE Transactions on Communications*, Vol. 39, November 1991, pp. 1542-1545.