ACOUSTIC ECHO CANCELLATION STRUCTURES BASED ON PERCEPTUAL HEARING CRITERIA

submitted by

Michael Oliver Vukadinovic, B. Sc. (Eng)

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Masters of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
May 4, 1996

ISBN 0-612-16473-X

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

Hands-free communication techniques are becoming increasingly important in the world of modern telephony. The convenience of hands-free links is invaluable in teleconferencing situations, automobiles and even in personal communications. The loudspeaker and microphone on a hands-free set are inherently coupled acoustically, producing an echo which degrades the quality of conversation over such a link. The conventional acoustic echo cancellation method is gain control which provides a half-duplex communication link. Research into adaptive filters which can provide a full-duplex communication link is intense but transparent transmission of speech over a hands-free link remains an elusive goal.

Conventional acoustic echo cancellers are concerned with minimizing an objective measure of the level of echo. This thesis presents a study of two adaptive filtering structures which are based on the goal of providing the best subjective echo cancellation possible. The adaptive filtering algorithms attempt to give the echo the same spectral shape as the absolute hearing threshold curve, in order to minimize the echo's audibility. The filtered-E LMS algorithm, considered first, was found not to perform the kind of frequency weighting desired. A proposed subband adaptive filter with perceptually-based adaptive tap assignment, when trained with natural speech, is capable of providing a noticeable improvement in audible echo cancellation; however, when trained with bandlimited speech, the algorithm's performance was found to be only slightly better or comparable to the fixed tap assignment algorithm in terms of audible echo cancellation. The subband adaptive filtering algorithm is evaluated by means of simple listening tests performed by the author.

Acknowledgements

I would like to thank NSERC and the University of Ottawa for the financial support that I have received over the past two years. I would like to thank Dr. Aboulnasr for her guidance and support and Heping Ding and Scott McLennon at Bell Northern Research for the speech and room response data and the helpful discussions. I also express my gratitude to my colleagues for their innumerable tidbits of advice, especially Cris Lambiri, computer guru extraordinaire. Last but not least, I would especially like to thank my family for their love and caring.

Contents

Figures

Tables

Notation

Various symbols and abbreviations used frequently in this thesis are summarized below. All notation is fully defined where it first arises in the text. Bold font is used to indicate vector quantities.

Symbols

n	discrete time
ω	digital frequency
i	weight vector tap number
M	transform size
m	subband number
L	weight vector length
Δ	delay
$x(n)$	input sequence
$d(n)$	desired response sequence
$y(n)$	filter output sequence
$e(n)$	error sequence

$h_d(n)$	plant time domain impulse response
$\mathbf{W}(n)$	weight vector
$w_i(n)$	tap i of the weight vector $\mathbf{W}(n)$
$\mathbf{X}(n)$	input memory vector
$\|\cdot\|$	Euclidean norm operator

Abbreviations

AEC	acoustic echo cancellation
BLMS	block least mean square
CO	central office
DCT	discrete cosine transform
DFT	discrete Fourier transform
DSP	digital signal processing
EEQ	error equalizing intersubband tap assignment algorithm
ERLE	echo return loss enhancement
FDAF	frequency domain adaptive filter
FELMS	filtered-E least mean square
FIR	finite impulse response
FNTF	fast Newton transversal filters
FPAF	fixed pole adaptive filters
FRLS	fast recursive least squares

FTA	standard fixed tap assignment subband adaptive filter
FTF	fast transversal filters
FXLMS	filtered-X least mean square
IIR	infinite impulse response
KLT	Karhunen-Loeve transform
LEC	line echo cancellation
LMS	least mean square
LSL	least squares lattice
MSE	mean square error
NLMS	normalized least mean square
RLS	recursive least squares
RTD	round-trip delay
RX	receive
SFTF	stabilized fast transversal filters
SNR	signal to noise ratio
SPL	sound pressure level
SUG	Sugiyama adaptive intersubband tap assignment algorithm [58]
TIP/TP	total impulse response power to tail response power ratio
TX	transmit
WEEQ	weighted error equalizing intersubband tap assignment algorithm

CHAPTER 1

Introduction

Acoustic echo cancellation is an integral part of hands-free telephony. A hands-free communication device is any device that allows the user to speak into the device without having to employ an extra peripheral device, such as a hand-set, for doing so. Such devices contain a loudspeaker for playing received communication and a microphone to pick up communication that is to be sent. The most common hands-free communication device is a speakerphone or hands-free telephone. A hands-free telephone placed in a room will inherently send echoes back on the send line whenever it receives a signal on the receive line and plays it over its loudspeaker. This is an artifact of the acoustic coupling present between the loudspeaker and the microphone on the speakerphone. Such echoes are unpleasant for users of the communication system and in the competitive world of telecommunications, an echo-free speakerphone would be a most desirable product. Anyone who has used a phone can imagine the convenience of replacing a hand-set telephone with a speakerphone.

Most generally, an acoustic echo canceller is some device that seeks to reduce the level of the echo returned on the send line by a hands-free communication device. On a conventional speakerphone, a reduction in the echo level is achieved by switching attenuation in and out of the receive and send paths. This reduces the communication channel to half-duplex. With the advent of DSP chips, research into adaptive signal processing algorithms to cancel acoustic echoes has intensified. The goal of adaptive

filtering algorithms is to provide a full-duplex communication link between one or more users of hands-free devices. The adaptive filtering algorithm is designed to minimize the difference between the output of the microphone (which has been converted to a digital representation) and the output of the adaptive filter. Currently, only high-end conference phones use DSP chips but telephone products are beginning to introduce DSP chips and adaptive echo cancelling algorithms. The adaptive filter and the loudspeaker share the same input: the speech signal on the receive line.

For many reasons acoustic echo cancellation is a difficult field and there is much ongoing research into providing efficient and effective signal processing algorithms to do the job. Standard adaptive filtering algorithms, such as the LMS and RLS algorithms, suffer from performance limitations in terms of convergence speed, modelling error, tracking performance and stability that prevent them from performing entirely satisfactorily in the arena of acoustic echo cancellation [3, 4, 5]. Newer algorithms such as transform domain adaptive filters [40, 41, 42] and subband adaptive filters [50, 51, 53] seek to improve upon the performance of LMS and RLS algorithms and are often based on these "basic" algorithms.

The speed at which an adaptive filter converges to the desired solution and the level of echo cancellation provided by the adaptive filter after convergence are the primary yardsticks by which acoustic echo canceller performance is measured. These are both objective measures and an adaptive filter structure is said to achieve improved performance if it achieves better echo cancellation or faster convergence than previous adaptive filtering structures. The structure against which most new algorithms are compared is the LMS algorithm. This thesis will focus, for the most part, on the level of echo cancellation. The objective measure of echo cancellation is simply a ratio of the power in the echo to the power in the input signal. Such a measure gives no consideration to the frequency content of these two signals. Performance can only be improved objectively by cancelling more of the echo. However if the frequency content of the echo is affected by the adaptive filtering algorithm somehow, the echo will be perceptibly different. The human ear does not have a flat frequency response. It is possible for two echo signals to

have the same power level yet markedly different perceptual annoyance levels. It makes sense to try to tune the performance of the adaptive filter to the human ear, thus getting the best subjective performance possible for a given level of objective echo cancellation.

In this thesis, an overview of current acoustic echo cancellation techniques is presented and some improvements which can be applied to many different adaptive filtering set-ups based on psychoacoustic properties are proposed. Psychoacoustic properties refer to properties of the human ear including how sensitive the ear is to sounds of particular frequencies. A modified LMS adaptive filtering algorithm is first investigated. This algorithm, known as the filtered-E LMS algorithm, adds a filter to the update path of the LMS algorithm structure in order to allow the error signal to be frequency weighted. The intention of weighting the error thus is to bias the performance of the LMS algorithm in a frequency dependent sense. The possibility of obtaining better performance at psychoacoustically important frequencies at the expense of degraded performance at other frequencies is investigated. Next a subband adaptive filtering algorithm which makes use of some psychoacoustic properties to improve the subjective performance of the adaptive filter is implemented. The performance of the new subband adaptive filtering algorithm is evaluated by means of some simple listening tests conducted by the author and objectively by means of echo spectra after adaptive filter convergence and mean square error convergence curves.

The thesis is divided into five chapters. Chapter 2 provides an introduction to the field of acoustic echo cancellation and gives a broad overview of current acoustic echo cancellation techniques. Chapter 3 investigates the performance of a modified LMS adaptive filtering structure known as the filtered-E LMS algorithm. Chapter 4 presents the subband adaptive filtering structure with frequency weighting capabilities mentioned above. This algorithm is evaluated both objectively and by means of simple listening tests performed by the author. Finally, Chapter 5 presents conclusions on the thesis material along with a discussion of areas requiring further study.

CHAPTER 2

Acoustic Echo Cancellation

This chapter presents an overview of the field of acoustic echo cancellation. Acoustic echoes are a problem encountered more and more frequently as the communications industry strives to manufacture communication systems that require hands-free operation. Some devices that fall under this category are hands-free telephones, inter-com systems and teleconferencing systems.

In Section 2.1, the source of the acoustic echoes is considered. Other sources of speech degradation are also presented briefly although the objective of this thesis is to cover the topic of acoustic echoes and techniques to overcome them. Section 2.2 covers the form of the conventional echo canceller. The structure presented here serves as the benchmark against which all other echo canceller structures are compared. Also described briefly at the beginning of Section 2.2 is the method of gain control. This method is perhaps the oldest attempt to cancel acoustic echoes. It differs from all the other schemes presented here in that it does not use an adaptive filter. It is covered here in order to provide a background for the introduction of adaptive filters into the field of acoustic echo cancellation. Section 2.3 discusses the limitations of the conventional acoustic echo canceller. Sections 2.4 to 2.9 cover other echo canceller structures. All the methods considered here use adaptive filters and the differences in the various methods are due to the specific algorithm used to update the adaptive filter as well as the adaptive filter structure.

2.1 Sources of Speech Degradation

There are several sources of speech degradation in a hands-free communication system. Any attempt to produce a high quality hands-free system must make efforts to control all sources of degradation. However, it is most practical to consider these sources of degradation independently when trying to develop a solution to the problem. These sources of degradation can be divided into three main categories. The categories are loudspeaker-microphone acoustic coupling, echo generated by the speakerphone user within the room and picked up by the microphone (room reverberation), and ambient noise. The focus of this thesis is on loudspeaker-microphone acoustic coupling but all three sources of speech degradation will be considered briefly in this section before concentrating solely on loudspeaker-microphone acoustic coupling. The problem of "howling" is also considered briefly.

2.1.1 Loudspeaker-microphone Acoustic Coupling

Acoustic echoes appear when a communication system uses audio terminals where the loudspeaker is acoustically coupled to the microphone [3, 5]. Such a set-up exists with hands-free telephone sets and teleconferencing systems. Consider for the moment a typical hands-free telephone set-up in an office. Figure 2.1 shows a schematic for one end (which shall be referred to as the near end) of the communication system before the addition of any echo cancellers. When the user on the remote end is talking, speech $x(t)$ is diffused into the room by the loudspeaker. Due to the acoustic paths between the loudspeaker and the microphone (including the direct path from the loudspeaker to the microphone, reflections off walls, furniture and people, and mechanical coupling through the casing of the telephone set itself) an echo signal $d(t)$ is picked up by the microphone and fed back to the remote user [50]. Thus the remote user will perceive an echo while he/she is speaking.

The disruptive effect that this echo has on the remote user's speech will be dependent on the magnitude of the echo signal as well as the delay between the original speech

and the echo generated by it [4]. The delay is a very critical factor and can greatly influence the amount of attenuation of the echo required to make the echo inaudible to the remote user. Greater delays between the echo and the original speech signal make the echo much worse subjectively to the remote user. The magnitude of the echo may be influenced by such factors as the size and architecture of the room and the type of materials used in the construction of the room, as well as the positioning of the loudspeaker and the microphone.

The transfer function of the loudspeaker, room and microphone, including any acoustical coupling present, will be referred to as the speakerphone-room response in this thesis. At times the term room response may be used to refer to the entire speakerphone-room response. It should be understood, however, that the responses considered in this thesis are not simply room responses. Many acoustical paths exist between the loudspeaker and microphone which are not due to the room itself.

2.1.2 Howling

Referring again to Figure 2.1, it must be noted that the hybrid will leak some of the send out signal back through to the receive side. The acoustic coupling between the loudspeaker and the microphone put together with the leakage of the hybrid provides a feedback path for echo signals [11]. If the gain of this feedback loop exceeds one for some frequencies, then the circuit becomes unstable. "Whistling" or "howling" noises may then be perceived by the user. The phenomenon of "howling" most often occurs during idle periods when neither user is speaking. During such idle periods, adaptation of the adaptive filter halts [11]. However, there may still be low-level signals present at the microphone input, such as ambient noise. If the room response changes during this "idle period" of the acoustic echo canceller, then the acoustic echo canceller may no longer be effectively

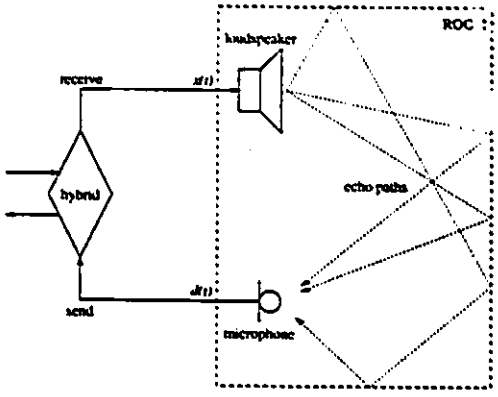

Figure 2.1: Loudspeaker-microphone acoustic coupling.

modelling the room. The possibility then exists for "howling" since there is a signal present in the system (the low-level noise, for example) and there is a feedback path which is not necessarily being attenuated by the adaptive filter. (When the adaptive filter models the room response accurately, the adaptive filter effectively attenuates the feedback path.)

2.1.3 Room Reverberation in the Far-end Room

Room reverberation in the far-end room refers to echo that is generated in the room where the remote user is speaking rather than echo generated in the near-end room (where the hands-free communication set of interest is located), as in the case of loudspeaker-microphone acoustic coupling discussed above. This far-end echo is not the result of loudspeaker-microphone coupling but rather the result of the multiple acoustic paths between the far-end speaker's mouth and the microphone [6]. The speech picked up by the microphone is composed of speech which has propagated directly from the speaker as well as speech which has reflected off walls, furniture and other things in the room. This problem must be considered separately from loudspeaker-microphone acoustic coupling since the transfer function between the loudspeaker and the microphone will not be the same as the transfer function between the speaker's mouth and the microphone.

2.1.4 Ambient Noise

Ambient noise is another obvious problem with a hands-free communication system. In an office environment, the microphone can pick up the sounds of other conversations, photocopiers and any other present noise. This noise will be added to the speech picked up by the microphone and will cause a degradation in the speech quality. Hands-free telephones are more susceptible to ambient noise than ordinary telephones due to the different kind of microphone set-up required. An ordinary telephone with a handset has a fairly directional set-up. Since the receiver is placed directly in front of the speaker's

mouth, the microphone tends to pick up the speech signal better than other undesired signals. A hands-free communication set-up, on the other hand, must have a fairly omnidirectional microphone set-up in order to pick up the user's speech. The user may be located at some distance from the microphone (compared to an ordinary telephone) and may not be talking directly into the microphone. A microphone capable of picking up the speech signal under such conditions will also be very good at picking up ambient noise as well. In spite of these adverse conditions, the hands-free communication system must still be capable of transmitting a clear speech signal for it to be a viable means of communication.

2.2 Conventional Echo Cancellers

The conventional echo canceller uses an FIR filter to model the response of the room. The coefficients of the FIR filter are determined by minimizing a measure of the difference between the actual output of the filter and the desired output of the filter. The most common measure is that of the mean square error (MSE). Before considering the details of the adaptive algorithm, an older method of removing the acoustic echo is considered. The method of gain control is considered in order to provide a background for the reasoning behind the use of adaptive filters in acoustic echo cancellation.

2.2.1 Gain Control

The method of gain control consists of the alternate insertion of attenuation blocks in the send and receive path. This reduces the normally full-duplex transmission of a communication system to half-duplex [3]. Figure 2.2 shows the placement of the attenuation blocks in the two paths. When speech is being received from the far end, the send attenuation block is switched on and any echo picked up by the near-end microphone is attenuated. The receive attenuation block is switched off to allow reception of the incoming signal. Typically, an attenuation of approximately 50 dB is required. When the near-end user is speaking, the send attenuation block must be switched out

of the path so that the near-end signal can be sent to the remote end and the receive attenuation block is switched on to ensure a loop loss of greater than one.

Proper functioning of the method of gain control relies on fast switching of the attenuation blocks and accurate determination of which user is talking, far or near-end so that the appropriate attenuation is enabled in both send and receive paths. In practice, these requirements cannot be fulfilled. The attenuation blocks can not be switched arbitrarily quickly since audible clicking noises become apparent when switching is performed too quickly. Thus the attenuation must be ramped from 50 dB up to 0 dB and vice versa. The problem with this scheme then, is that bits of speech may be lost. If the near-end user stops speaking and the remote user begins to speak, the remote user's first bit of speech may be lost as the attenuating block in the send path of the remote end is switched off and the receive path attenuation in the near-end is switched off. Bits of conversation such as "yes", "ummm" and other short expressions which are essential to a normal, free-flowing conversation may be inaudible. The accurate determination of which user is talking is also problematic. If both users are attempting to talk, one of them will be cut off. It will not be immediately apparent which user has been cut off. Each user will be talking while only one of them will be hearing what the other user is saying. A conversation under such conditions will be quite strained.

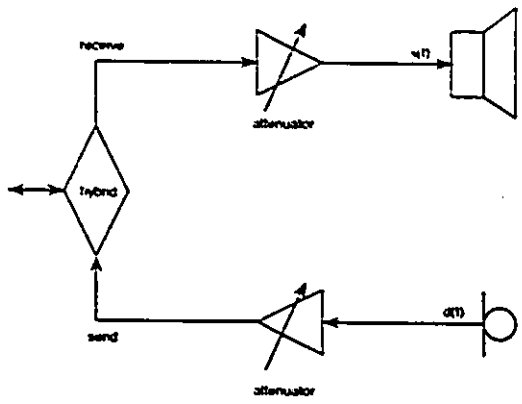

Figure 2.2: Switched loss gain control.

The use of adaptive filtering in acoustic echo cancellation can relax the requirements on gain control considerably or ultimately, eliminate them altogether. Using an adaptive filter to model the echo path allows for full-duplex transmission. An adaptive filter attempts to model the room response on a continuous basis with no need for switching. Of course, the problems described above which so limit the method of gain control

also provide some difficulties for adaptive filters, but these difficulties appear more surmountable in the case of adaptive filters.

2.2.2 Adaptive Filter Echo Cancellers

The configuration of the conventional acoustic echo canceller is shown in Figure 2.3. The adaptive filter here operates in the system identification set-up [13]. The unknown system of interest is the acoustic echo path between the loudspeaker and the microphone. The adaptive filter tries to predict the response of the room to the given input signal $x(n)$. Obviously, the adaptive filter cannot adapt under all of the possible operating conditions of the hands-free system. If the input signal $x(n)$ is not present or if there is another signal $s(n)$, representing near-end speech, present at the microphone then adaptation of the adaptive filter must cease. Once adaptation of the adaptive filter has ceased the coefficients of the adaptive filter are held constant and it is hoped that the adaptive filter will provide a good model of the room response until adaptation of the adaptive filter can begin anew.

FIR filters are generally used as the preferred structure because FIR filters are inherently stable and adaptive algorithms are relatively simple to implement with FIR filters. IIR filters are avoided due to problems with stability, although IIR algorithms will be discussed briefly in Section 2.9. The FIR filter $W(z)$ shown in Figure 2.3 will be updated when the remote user is talking and his/her speech signal $x(n)$ is present on the receive side of the hands-free set and the only signal present at the microphone input is $d(n)$, the

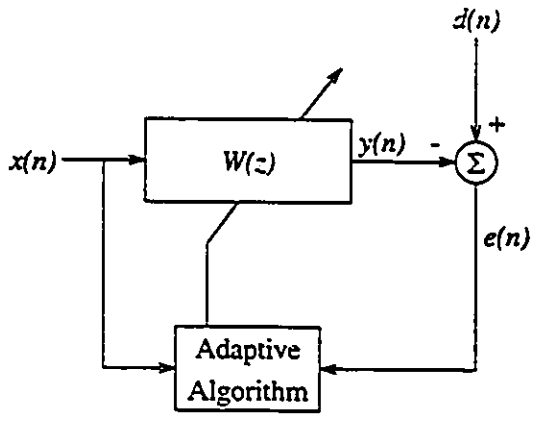

Figure 2.3: Structure of the conventional FIR adaptive filter.

echo generated by the remote user's speech. $\mathbf{W}(n)$ is the weight vector given by

$$\mathbf{W}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \cdots & w_{L-1}(n) \end{bmatrix}^T. \tag{2.1}$$

where ^{T} represents transposition. The input signal vector $\mathbf{X}(n)$, composed of the last L samples of $x(n)$, is defined as

$$\mathbf{X}(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-L+1) \end{bmatrix}^T. \tag{2.2}$$

The output of the adaptive filter at sampling instant n can be computed as

$$y(n) = \mathbf{W}(n)^T\mathbf{X}(n). \tag{2.3}$$

The error $e(n)$ is the difference between the desired response $d(n)$ and the output of the adaptive filter $y(n)$:

$$e(n) = d(n) - y(n). \tag{2.4}$$

The most common means of updating the weight vector is with the least mean square algorithm, commonly referred to as the LMS algorithm [13]. In the LMS algorithm, the mean square error $\overline{e^2(n)}$ is minimized by updating the weight vector $\mathbf{W}(n)$ at each sampling instant n according to the following update equation

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu e(n)\mathbf{X}(n) \tag{2.5}$$

where μ is the adaptation step size. The step size μ governs the compromise between the rate of convergence and the final mean square error after convergence [12, 13]. The allowable values of μ are bounded by

$$0 < \mu < \frac{2}{LE(x^2(n))} \tag{2.6}$$

where $E(\cdot)$ is the expectation operator.

2.3 Limitations of Conventional Echo Cancellers

The limitations of the LMS algorithm are described in this section. This discussion will also provide a motivation for many of the algorithms discussed in the next section.

The LMS algorithm is the simplest adaptive algorithm and is used as the benchmark against which all other algorithms are compared.

The criteria by which an acoustic echo canceller is evaluated are by no means fixed. These criteria vary with respect to the size and layout of the room. However, as an example of a typical teleconferencing situation it is reported in [4] that for a room with a reverberation time of 400 ms and a communication system with a round trip delay time of 100 ms, an echo return loss enhancement (ERLE) of 40 dB is required. The ERLE of an adaptive filter is defined as

$$\text{ERLE} = 10 \cdot \log_{10}\left(\frac{\overline{d^2(n)}}{\overline{e^2(n)}}\right) \tag{2.7}$$

where $d(n)$ and $e(n)$ are the desired response signal and the error signal, respectively, as shown in Figure 2.3. The ERLE required is based on a subjective assessment of the quality of the system. Round trip delays greater than 100 ms may require an ERLE even greater than 40 dB to achieve acceptable speech quality.

2.3.1 Speakerphone-room Response Length

The job of the echo canceller is made very difficult by the length of typical speakerphone-room responses [5]. In order for the echo canceller to achieve an ERLE of 40 dB or thereabouts the echo canceller must use an FIR filter with several thousand taps. Figures A.1 and A.4 in Appendix A show two speakerphone-room responses sampled at 8 kHz. These two speakerphone-room responses will be used as plant responses later in the simulations of this thesis.

For the speakerphone-room response shown in Figure A.1, an echo canceller with 300 taps and white noise as input can theoretically achieve an ERLE of 14.3 dB based on a TIP/TP [7, 8] calculation. The equation for TIP/TP is given in Appendix A. The TIP/TP factor was proposed by Knappe [7, 8] as a means of estimating adaptive filter performance for a given length of adaptive filter. Of course, 14.3 dB of cancellation is achieved only if the echo canceller can match the first 300 points of the speakerphone-room response exactly. In a real situation an echo canceller will not be capable of this.

Even if the input signal to the echo canceller is white noise, which is the best training signal, there will exist some misadjustment of the coefficients of the adaptive filter after convergence, which will prevent the echo canceller from achieving the theoretical maximum ERLE. The rate of convergence of the adaptive filter is also of paramount importance. While the echo canceller is in a transient state the room echo will not be cancelled effectively and the quality of the communication system will suffer. The rate of convergence of the LMS algorithm is dependent on the step size μ used. From Equation 2.6, it can be seen that the step size for the LMS algorithm must be chosen to be inversely proportional to the length of the adaptive filter. Longer adaptive filters thus converge much slower than shorter adaptive filters.

2.3.2 Adaptive Filter Input Characteristics

The performance of the LMS algorithm is also hampered by the type of input signal present in communication systems, namely speech [50]. Speech is a highly correlated signal with a spectrum that is far from flat. As mentioned earlier, white noise is considered the ideal input for training because its spectrum is flat and its statistics do not vary with time. In this regard, speech is also a poor training signal since it is a nonstationary signal. The energy in a speech signal has many bursts and lulls with respect to time. The speed of convergence of adaptive filtering algorithms is related to the eigenvalue spread of the autocorrelation matrix of the input [41], defined as:

$$\mathbf{R_{XX}} = E(\mathbf{X}(n)\mathbf{X}^T(n)). \tag{2.8}$$

The matrix $\mathbf{R_{XX}}$ is an $L \times L$ matrix. If the autocorrelation coefficients are defined as

$$r_i = E(x(n)x(n-i)), \qquad i = 0, 1, 2, \cdots, L-1 \tag{2.9}$$

then for a stationary input signal $x(n)$, $\mathbf{R_{XX}}$ will have the following form:

$$\mathbf{R_{XX}} = \begin{bmatrix} r_0 & r_1 & \cdots & r_{L-1} \\ r_1 & r_0 & \cdots & r_{L-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{L-1} & r_{L-2} & \cdots & r_0 \end{bmatrix}. \tag{2.10}$$

As mentioned above, the rate of convergence of the adaptive filter is related to the eigenvalue spread of $\mathbf{R_{XX}}$. Eigenvalues of $\mathbf{R_{XX}}$ are denoted as λ_i. The condition number of the autocorrelation matrix is the ratio $\lambda_{max}/\lambda_{min}$ and is a measure of the eigenvalue spread of $\mathbf{R_{XX}}$. When this ratio is high the adaptive filter converges slowly [41]. When $x(n)$ is white, $\mathbf{R_{XX}}$ will be a diagonal matrix with all diagonal elements equal to r_0, as defined above. The eigenvalues for such a matrix will all have the same value and thus the ratio $\lambda_{max}/\lambda_{min}$ will be unity. This is the best possible scenario. To illustrate how the performance of the LMS algorithm deteriorates when $x(n)$ is speech, consider Table 2.1 from [41] which shows estimated autocorrelation coefficients for a speech signal. In this case L is 16 and the coefficients r_i are estimated from 6.4 seconds of speech, spoken by a male speaker and sampled at 10 kHz. As well, the coefficients have been normalized according to

$$r_i = \frac{E(x(n)x(n-i))}{E(x^2(n))} \tag{2.11}$$

Table 2.2, also from [41], gives the eigenvalues for the autocorrelation coefficients of Table 2.1.

For this data, the ratio of $\lambda_{max}/\lambda_{min}$ is given by λ_0/λ_{15} and is equal to 1874. For a white input signal with a diagonal autocorrelation matrix, this ratio would be one. It can be seen that the autocorrelation matrix deviates greatly from the ideal case of white noise, in terms of eigenvalue disparity, and slow convergence of the adaptive filter will result. In Sections 2.4 to 2.9, algorithms which seek to improve upon the performance of the LMS algorithm are introduced. The main aim of several of these algorithms, including transform domain adaptive filtering algorithms and subband adaptive filtering algorithms, is to reduce the eigenvalue disparity of the autocorrelation matrix of the input for highly correlated inputs such as speech.

i	r_i	i	r_i
0	1.00000000	8	-0.09794104
1	0.91189350	9	-0.21197350
2	0.75982820	10	-0.30446960
3	0.59792770	11	-0.34471370
4	0.41953610	12	-0.34736840
5	0.27267350	13	-0.32881280
6	0.13446380	14	-0.29209750
7	0.00821722	15	-0.24512650

Table 2.1: Autocorrelation Coefficients of a Speech Signal.

2.3.3 Nonstationarity of Speakerphone-room Responses

Another very important characteristic of the acoustic echo cancellation problem is that the system being modelled, the room (in conjunction with the speakerphone), is nonstationary itself [3, 10, 50]. This means that the adaptive algorithm must be capable of tracking changes in the room response. If the adaptive filter cannot adapt quickly enough to changes in the room response then one can never hope to achieve good echo cancellation. The room response will change when objects in the room are moved or even when people move about a room. The ability of an algorithm to adapt to changes in the speakerphone-room response is known as its tracking performance. Several works [7, 9, 10] investigate the tracking performance of adaptive filtering algorithms. Nonstationarity in a speakerphone-room response can be created by waving an object back and forth in front of the speakerphone while recording primary and reference signals (see Appendix A for a brief discussion of room response measurement techniques and further references).

i	λ_i	i	λ_i
0	6.1069	8	0.0821
1	6.0082	9	0.0625
2	1.8629	10	0.0531
3	0.5082	11	0.0371
4	0.2784	12	0.0329
5	0.1051	13	0.0214
6	0.0841	14	0.0074
7	0.0821	15	0.0033

Table 2.2: Eigenvalues of $\mathbf{R_{xx}}$.

2.3.4 Loudspeaker Nonlinearities

Loudspeakers (such as are used on speakerphones) are nonlinear devices [7, 24]. Most adaptive filtering algorithms (LMS and RLS based) are based on linear theory. Their performance can be expected to suffer if significant nonlinearities exist in the speakerphone-room response. Such nonlinearities may impose a ceiling on the possible ERLE which can be achieved by linear adaptive filtering. Adaptive filtering techniques to overcome nonlinear limitations such as neural networks [24] are investigated in the literature.

This thesis concentrates on linear adaptive filtering techniques and ignores nonlinear effects. The simulations later in this thesis are based on speakerphone-room responses

that are assumed to be linear.

2.3.5 Summary

In summary, many of the factors which limit the performance of the LMS algorithm have been mentioned here. The ability to achieve low modelling error (an ERLE of 40 dB or more) and to converge quickly, given such an unsuitable training signal as speech, is beyond the LMS algorithm. These requirements work against each other. To achieve a high ERLE requires an adaptive filter with several thousand taps, which makes for very slow convergence, thus preventing satisfaction of the goal of fast convergence. In an acoustic echo cancellation simulation presented in [20], the NLMS algorithm is used in a teleconferencing situation. The input is split into two bands, each sampled at 8 kHz with the lower band covering the frequency range of 0.15 kHz - 4 kHz (the range of interest here). The adaptive filter has 3840 taps in this frequency band. With white noise as an input, the adaptive filter takes more than 6 seconds to converge to a final ERLE of about 30 dB. When the adaptive filter is trained with a speech signal as the input, convergence takes more than 30 seconds.

2.4 Normalized LMS

The LMS algorithm is ill-suited to adaptive filtering when the input is a nonstationary signal such as speech, due to the large fluctuations in input power over time. The fixed step size of the LMS algorithm will not be well tuned to such fluctuations in the input signal power. The normalized LMS adaptive algorithm (NLMS) [12, 14] is a slight variation on the LMS algorithm. In the NLMS algorithm, the step size is adjusted automatically at each iteration to account for the power in the input signal $x(n)$. If the input $x(n)$ is a stationary signal then the NLMS algorithm will perform almost identically to the LMS algorithm but when $x(n)$ is not stationary, the NLMS algorithm will be able to outperform the LMS algorithm by attempting to match the step size to the expected value of $x(n)$ at each iteration. In fact, with input signals

such as speech it becomes necessary to use NLMS in place of LMS in order to achieve any kind of reasonable performance. In the NLMS algorithm, the LMS weight vector update Equation 2.5 is modified to give

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu}{\|\mathbf{X}(n)\|^2} e(n)\mathbf{X}(n) \tag{2.12}$$

where $\|\mathbf{X}(n)\|^2$ is the Euclidean norm of the vector $\mathbf{X}(n)$ and is defined as

$$\|\mathbf{X}(n)\|^2 = x^2(n) + x^2(n-1) + \cdots + x^2(n-L+1). \tag{2.13}$$

The bounds on μ are given by

$$0 < \mu < 2 \tag{2.14}$$

for the NLMS algorithm.

2.5 Modified LMS Algorithms

The LMS algorithm is simple and effective. Unfortunately, its performance suffers when the input signal is highly correlated and when the adaptive filter is very long. Both of these conditions are encountered in the problem of acoustic echo cancellation.

The first class of modified LMS algorithms considered attempts to decorrelate the input signal in order to improve the speed of convergence. In this algorithm, the input signal $x(n)$ is filtered by a "whitening" filter before being used in the adaptive filter update equation. The error $e(n)$ must also be filtered by the same "whitening" filter. Figure 2.4 shows the signal paths of this modified LMS algorithm, known as a coupled LMS algorithm [21]. The structure is identical to the filtered-X LMS structure [13, 27], save that the additional filter of the coupled LMS algorithm is an adaptive predictor and is used specifically to whiten the input. With a filtered-X LMS structure, the additional filter is a fixed filter and represents a model of some error path component which cannot be removed from the adaptive filtering problem [26]. A variation of the filtered-X LMS structure, known as the filtered-E LMS adaptive filter, has also been

used to serve the purpose of weighting the final error, by appropriate choice of the fixed filter $P(z)$ [28].

In [21], the filter $\mathbf{P}(n)$, where $\mathbf{P}(n)$ is the vector containing the coefficients of the filter $P(z)$, is an adaptive predictor. It is updated using the LMS algorithm as is the main adaptive filter $\mathbf{W}(n)$ (corresponding to $W(z)$). Note that the signal path through the echo channel and through the echo cancelling filter $\mathbf{W}(n)$ is not changed by the addition of the "whitening" filter $\mathbf{P}(n)$. The update equations for the coupled LMS algorithm are very similar to the LMS update equations. The equations given below correspond to the fixed step size LMS algorithm. $\mathbf{W}(n)$ remains as given by Equation 2.1 and $d(n)$ is the desired response as before. The filtered sequence $x_f(n)$ is determined from $x(n)$ by

$$x_f(n) = x(n) - \mathbf{P}^T(n)\mathbf{X}_P(n) \tag{2.15}$$

where

$$\mathbf{P}(n) = \begin{bmatrix} p_1(n) & p_2(n) & \cdots & p_N \end{bmatrix}^T \tag{2.16}$$

and

$$\mathbf{X}_P(n) = \begin{bmatrix} x(n-1) & x(n-2) & \cdots & x(n-N) \end{bmatrix}^T. \tag{2.17}$$

N is the number of taps in the adaptive predictor filter $\mathbf{P}(n)$. Sequences $y(n)$ and $e(n)$ are determined as in Equations 2.3 and 2.4 respectively. The same filter $\mathbf{P}(n)$ is applied to $e(n)$:

$$e_f(n) = e(n) - \mathbf{P}^T(n)\mathbf{E}_P(n) \tag{2.18}$$

where $E_P(n)$ takes the same form as $\mathbf{X}_P(n)$ in Equation 2.17 above. Finally, both the adaptive predictor and the main adaptive filter are updated according to the following two equations:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu e_f(n)\mathbf{X}_f^*(n) \tag{2.19}$$

$$\mathbf{P}(n+1) = \mathbf{P}(n) + \eta x_f(n)\mathbf{X}_P^*(n) \tag{2.20}$$

where the notation $(\cdot)^*$ indicates complex conjugation of each component of $(\cdot)$. $\mathbf{X}_f(n)$ has the same form as $\mathbf{X}(n)$ in Equation 2.2 and η is the step size for the adaptive

predictor. The aim of this algorithm is to provide the LMS algorithm with an input signal $x_f(n)$ which is less correlated than the original signal $x(n)$. In other words the autocorrelation matrix of $x_f(n)$, $\mathbf{R}_{\mathbf{x}_f\mathbf{x}_f}$, will have a smaller eigenvalue spread than $\mathbf{R}_{\mathbf{xx}}$.

In [21], it is reported that the coupled LMS algorithm converges faster than the LMS algorithm when the input is a complex sinusoid added to complex white noise. Theoretically, the structure should converge faster than the LMS algorithm for highly correlated inputs such as speech in an application like acoustic echo cancellation. However, several questions are left unanswered. The effect of misadjustment of the coefficients of $\mathbf{P}(n)$ on the error was not discussed. With a nonstationary signal such as speech, one expects that the adaptive predictor coefficients will vary around the optimum and this will have an adverse effect on the adaptation of the main adaptive filter. This problem has been confirmed by the author in some simple system identification simulations with a speech input. The choice of the predictor step size η and its effect on the stability of the system has also not been quantitatively evaluated. In short, it is unclear whether significant gains can be achieved by the use of the coupled LMS adaptive algorithm. This algorithm does not add to the computational complexity of the LMS algorithm significantly since it requires the addition of only two filters which are usually chosen to be very short.

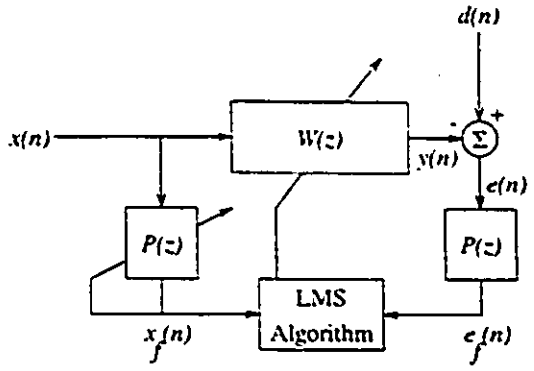

Figure 2.4: Coupled-LMS adaptive filter.

A very similar algorithm to the coupled LMS above is presented in [22]. The differences are that in [22] the "whitening" filter is periodically re-evaluated about every 16 ms using the Levinson-Durbin algorithm and the prewhitening filter is applied to $x(n)$ and $d(n)$ rather than to $x(n)$ and $e(n)$ as in [21]. No analysis of the algorithm was undertaken and no comparisons with the LMS algorithm or any other algorithm are given in this paper.

Another modification made to the LMS algorithm is to use a weighted step size [10, 20]. This approach is based on the statistics of typical room responses. Room responses are often characterized as being exponentially decaying. This may not be true for some room responses but every room obeys a decaying response of some kind. Since the room response is a decaying function, it makes sense to "force" the filter to have a decaying impulse response as well. This can be done by having earlier taps updated with a larger step size than later taps. This should bias the adaptive filter somewhat to look for an impulse response more in line with typical room responses. Theoretically, the step sizes for the adaptive filter take on an exponential profile but in practice one can use a set of precomputed discrete step sizes in order to avoid the on-line calculation of an exponential function or the storage of a table of values of an exponential function. A microprocessor can perform multiplications and additions very easily, but the evaluation of complex functions requires the use of series expansions and/or look-up tables. A typical step size versus filter tap plot is shown in Figure 2.5.

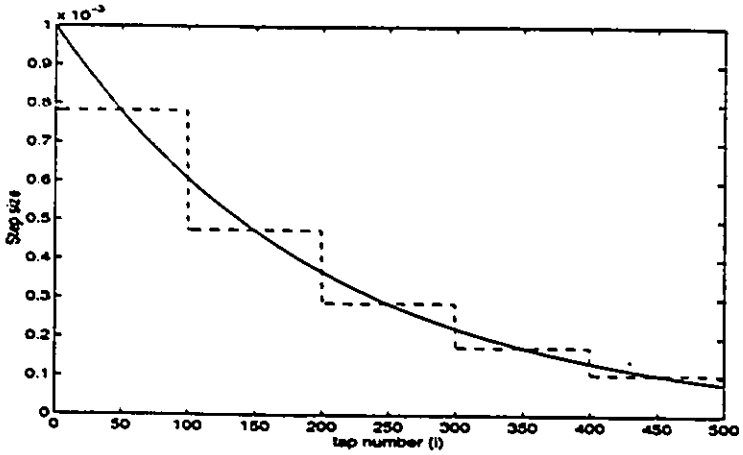

Figure 2.5: Exponentially weighted step sizes.

In summary, modified LMS algorithms achieve some gains in convergence speed over the LMS algorithm. The steady state performance of the modified LMS algorithms is very similar to the LMS algorithm itself. However, the gains in convergence speed are not great and there is much left to be desired in terms of speeding up the convergence of the adaptive filter.

2.6 Recursive Least Squares Algorithms

RLS algorithms are known for their fast convergence. For a stationary input signal, an RLS algorithm will converge to the optimum solution in about L iterations where L is the number of taps in the adaptive filter. However, RLS algorithms require a large number of computations for each sample. In fact the complexity of RLS adaptive filters is on the order of L^2 [12]. For a large adaptive filter, such as would be used in acoustic echo cancellation, the computational burden of the RLS algorithm becomes excessive when compared to the LMS algorithm (with a computational complexity on the order of L.). There are several algorithms which seek to decrease the computational complexity of the RLS algorithm. These algorithms are known collectively as "Fast RLS" (FRLS) algorithms. Some examples of FRLS algorithms are the fast Kalman filter [31], fast transversal filters (FTF) [32, 33] and their stabilized versions (SFTF), least squares lattice (LSL) algorithms [12],and fast Newton transversal filters [34]. FRLS algorithms have computational complexities which are significantly less than a direct application of the RLS algorithm. In fact, the fastest FRLS algorithms have a computational complexity on the order of L.

RLS algorithms have yet to be widely implemented in acoustic echo cancellation applications due to inherent instability problems. Instability in FRLS algorithms results from unpredictable rounding errors in the calculations required by a FRLS algorithm and from singularities in the input autocorrelation matrix [12, 35]. The problem of singularities in the input autocorrelation matrix can be minimized by proper choice of the initial variables. The numerical rounding errors are influenced by the accuracy of arithmetic used. If an FRLS algorithm is implemented with finite precision arithmetic, it will eventually go unstable [35]. The current methods to stabilize FRLS algorithms involve periodically re-initializing the internal state variables of the FRLS algorithm. These re-initializing procedures add at least L operations per sample to the computational complexity of the FRLS algorithms.

In [35], a new method of stabilizing an FRLS algorithm is presented. This method involves multiplying one of the internal variables of the FRLS algorithm by a factor close to but slightly less than one. This method requires only one additional multiplication per sample. Results are presented that show that the use of the above procedure avoids the instability that would otherwise have occurred in the FRLS algorithms. The FRLS algorithms are then shown to perform fairly closely to the RLS algorithm itself.

FRLS algorithms typically employ a predictor section of the same length as the length of the adaptive FIR filter itself. In [36], a new fast RLS algorithm is proposed to reduce the computational complexity of an FRLS algorithm by employing a predictor section of a shorter length. Given that the length of the adaptive filter is probably a few thousand taps great savings in the computational complexity are expected if a predictor of 5 to 20 taps is used. A predictor of such a length should be sufficient to model speech. An algorithm which is less than twice as computationally intense as the NLMS algorithm is shown to perform nearly as well as a standard FRLS algorithm when the predictor is of length 20. The algorithm converges considerably faster than the NLMS algorithm as long as the length of the predictor is chosen to be greater than four.

The LMS algorithm uses the instantaneous value of $e^2(n)$ as its measure of the MSE. The RLS algorithm, on the other hand, uses information about the input data and error data extending back to the time the algorithm was started [12]. Usually, a "forgetting factor" is applied to the error measure of the RLS algorithm so that very old data are forgotten, and the adaptive filter can follow the statistical variations in the desired response when the adaptive filter operates in a nonstationary environment. The ability of the adaptive filter to follow these statistical variations is known as "tracking" performance. The error measure for the RLS algorithm at sampling instant n is defined as

$$J_{RLS}(n) = \sum_{i=0}^{n} \beta(n,i) \cdot |e(i)|^2. \tag{2.21}$$

A commonly used weighting factor $\beta(n,i)$ is the exponential weighting factor, given by

$$\beta(n,i) = \lambda^{n-i}, \qquad i = 1, 2, \cdots, n. \tag{2.22}$$

The effective "memory" of the exponential weighting factor is given by $(1-\lambda)^{-1}$. If the effective memory of the weighting factor is K, then the error measure of the RLS algorithm is essentially the sum of the past K values of the squared error. $\lambda = 1$ corresponds to the case of the infinite memory RLS. It is the infinite memory RLS which is capable of making great gains in convergence speed over the LMS algorithm. However, the infinite memory RLS has very poor tracking performance since every value of the error from time 0 to time n is used to determine the current adaptive filter weight vector. Changes in the system being modelled will not be followed very well by the RLS algorithm since the RLS algorithm is still using "statistically outdated" error measurements. The use of the forgetting factor allows the RLS to track better, but convergence speed suffers as a result. In fact, for the exponentially weighted RLS, the factor $(1-\lambda)$ plays a role very similar to that of μ in the LMS algorithm. Results in [37, 38] indicate that, with both the LMS and the RLS algorithms tuned to minimize misadjustment, the RLS algorithm has worse tracking performance than the LMS algorithm.

Although the convergence properties of RLS algorithms are very attractive the computational complexities of these algorithms remains excessive. Even the fastest FRLS algorithms, except some more recently proposed algorithms [36], are still about four times more computationally intense than the NLMS algorithm. As well, instability problems due to the causes described above require additional calculations and may affect the performance of the algorithm to some extent. The convergence gains possible with the RLS algorithm become less attractive when the performance of the RLS algorithm in a nonstationary environment is considered. The acoustic echo cancellation environment is indeed highly nonstationary, as described previously, and the performance of the RLS algorithm will suffer under such conditions. For these reasons the bulk of research in the field of acoustic echo cancellation remains dedicated to the use of LMS algorithms and, save the odd exception, the rest of this chapter will deal specifically with various algorithms all having as their basic method of parameter optimization an LMS algorithm.

2.7 Transform Domain Algorithms

Transform domain implementations, also known as frequency domain implementations, of adaptive filtering algorithms have two basic advantages over time domain implementations. The first advantage is the potential computational savings over a time domain adaptive filter if an efficient transform algorithm such as the Fast Fourier Transform (FFT) is used [40]. The second advantage is that the input signal is split into transform components which are approximately uncorrelated and thus the adaptive filter coefficient corresponding to each input transform component may be updated using its own time varying step size [41]. Transform domain algorithms use an LMS algorithm for the update of their coefficients although the use of an algorithm such as the RLS is not precluded. The discussion here will be limited to the LMS algorithm.

The basic operation involved in a transform domain algorithm is the transformation of the input signal $x(n)$ into a more desirable form before adaptive processing. The transform used is a fixed transform and can be considered as a preprocessing step. Typically a Discrete Fourier Transform (DFT) or a Discrete Cosine Transform (DCT) is used. The basic blocks involved in a transform domain adaptive algorithm are shown in Figure 2.6. Transform sequences $X(k), Y(k), D(k)$ and $E(k)$ are parallel signals with either M or $2M$ parallel coefficients each where M is the size of the transform used. $2M$ coefficients are needed to implement linear convolution whereas if M coefficients are used, circular convolution will be implemented. The choice of structure from Figure 2.6 depends upon the particular application. Note that the first structure in Figure 2.6 has more delay (two transform blocks) in the error path than the second structure. This added delay may cause some deterioration in performance under some circumstances.

In Section 2.3, the effect of eigenvalue spread of the autocorrelation matrix on the rate of convergence of an adaptive filter was discussed. It can be shown that an orthogonal transform such as the DFT or DCT can greatly reduce the eigenvalue spread and hence result in improved convergence of the adaptive filter [41]. Estimated eigenvalues for a 16×16 autocorrelation matrix of a speech signal were given in Table 2.2. The eigenvalue

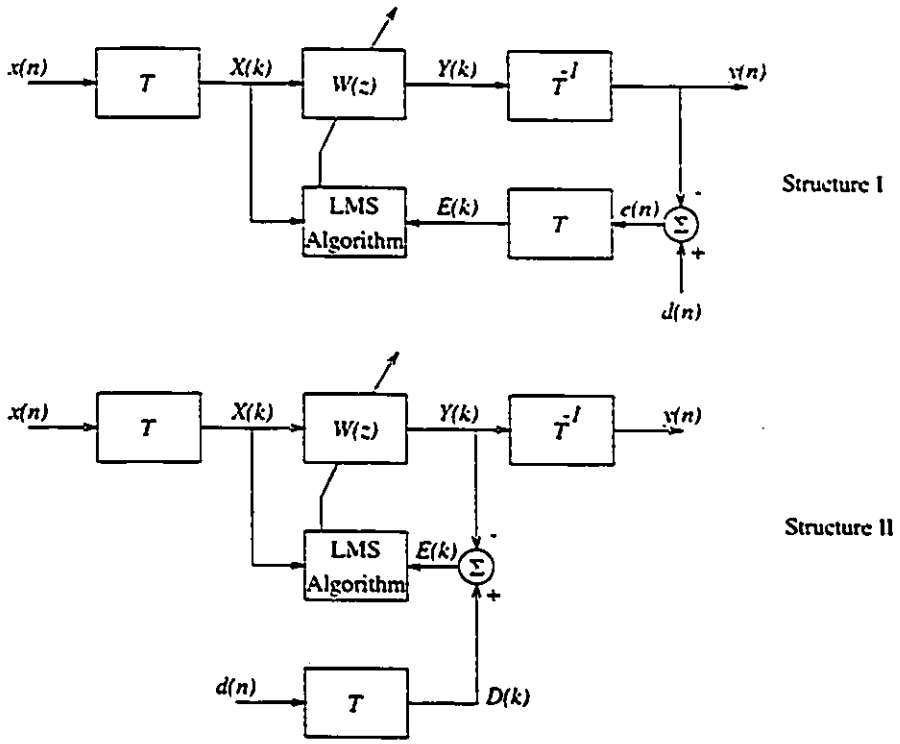

Figure 2.6: Structure of the transform domain adaptive filter.

spread of this matrix was shown to be very large. In order to illustrate the advantages of a transform domain implementation of an adaptive filter, Table 2.3 shows eigenvalue spreads for several different transforms [41]. The Karhunen-Loeve Transform (KLT) is the ideal transform in that the KLT of a signal has completely orthogonal components. The KLT is a time varying signal dependent transform that is difficult to compute in practice. The autocorrelation matrices of the transformed signals are indeed much better conditioned than the original signal. The DCT in particular appears to be a very appropriate transform for use with speech signals.

There are essentially two types of transform domain algorithms. The first type implements the adaptive filter using a block processing approach [40]. Its main advantage is the computational savings afforded when very long adaptive filters are used. The second type applies the transform at every sampling instant and is known as a "sliding window" transform domain algorithm [40, 41]. These two forms of the transform domain adaptive filter will be considered in turn.

Transform	λ_{min}	λ_{max}	$\lambda_{max}/\lambda_{min}$
Identity	0.0032585	6.1069	1874
DFT	0.0288863	3.5476	123
DCT	0.6331885	1.6733	3
KLT	1.0000000	1.0000	1

Table 2.3: Eigenvalue spreads for different transforms (M = 16) of the speech signal in Tables 2.1 and 2.2.

The standard time domain LMS adaptive filter can be implemented using a block approach. The block LMS algorithm (BLMS) achieves computational reductions over the LMS algorithm [19, 39] by only updating the weight vector once every N samples rather than at every sampling point. In this case, the estimated gradient is computed at every sampling point but the weight vector is not updated at every sampling point. Instead, the gradient estimates are averaged over the block size N and the weight vector is updated once every N samples using the averaged gradient estimate. The block processing is in effect a sort of smoothing process. The block transform domain adaptive filter implements the above block processing scheme in the transform domain. This algorithm is commonly called the Frequency Domain Adaptive Filter (FDAF) since the DFT is used to implement the convolutions required by the algorithm. The FDAF can be implemented using either of the structures of Figure 2.6. When the DFT is used to implement the convolutions required by the algorithm, constraints must be introduced in order to obtain linear convolution. An FDAF which implements linear convolution is known as a constrained FDAF. For a block size of N, the DFT size used is $2N$. The linear convolutions may be performed using either the overlap-add or the overlap-save methods [1]. In either case, the FDAF algorithm will require storing old data blocks and appending blocks of zeros to current data before the computations of the size $2N$ transforms. A gradient constraint section requiring two DFT blocks is also required in the weight vector update section in order to "zero" the last N points of the gradient

FDAF Algorithm	Filter Size L						
	16	32	64	128	256	512	1024
Linear Convolution	2.063	1.188	0.672	0.375	0.207	0.062	0.033
Unconstrained Gradient	1.438	0.813	0.453	0.250	0.137	0.040	0.021
Circular Convolution	0.625	0.359	0.203	0.113	0.062	0.019	0.010

Table 2.4: FDAF computational complexity ratios (relative to LMS) [40].

estimate so that the $2N$ weight vector coefficients correspond to the N point time domain weight vector. The unconstrained FDAF is obtained if the gradient constraint condition above is removed. As such the $2N$ point weight vector will not correspond directly to the equivalent time domain vector as it did in the case of the constrained FDAF. The dropping of the gradient constraint enables the unconstrained FDAF to save a number of computations over the constrained FDAF. If N point DFT's are used in place of $2N$ point DFT's then the FDAF will implement circular convolutions. Its advantage over the constrained and unconstrained FDAF's is the less complex transform required.

The weight vector update of the FDAF uses a time varying independent step size for each weight vector coefficient. This is possible since each weight vector coefficient represents the frequency domain component of the adaptive filter and thus different weight vector coefficients are approximately uncorrelated. These independent step sizes for different frequency bands lead to faster convergence for highly correlated inputs as indicated by Table 2.3. The FDAF is also computationally less intense than a fullband LMS filter for longer adaptive filters. The computational savings arise from the use of an efficient FFT algorithm for the implementation of the DFT. Table 2.4 from [40] shows the ratio of FDAF computational complexity to time domain LMS adaptive filter complexity for a block of N samples for several different filter lengths.

The disadvantages of the FDAF algorithms stem from the use of block weight vector updates and from the inexact convolutions implemented by the unconstrained FDAF and the circular convolution FDAF algorithms. For the case of acoustic echo cancella-

tion, adaptive filters with several thousand taps may be required. An FDAF algorithm with a filter of such a length would have a very long block length. Such a long block length would seriously hamper the ability of the algorithm to track any nonstationarities in the room response. As well the FDAF would perform poorly with a nonstationary signal such as speech which might have more quickly varying statistics than the FDAF would be able to pick out, given a block length of a few thousand samples. Since the data is processed every N samples, the entire FDAF will have an end to end delay of N samples as well. Such a long delay may be unacceptable for applications such as telephony. The unconstrained FDAF and the circular convolution FDAF would also have inferior steady state error performances as compared to the fullband LMS adaptive filter and as already pointed out the steady state error performance required for echo free transmission is as yet an elusive goal for all adaptive filters.

"Sliding window" transform domain algorithms will now be examined. These algorithms are "non-block" transform domain algorithms, in contrast with the FDAF's discussed above. As such, they avoid the performance problems associated with long block lengths discussed above. However, they do so at the expense of computational complexity. These algorithms are more computationally complex than a time domain LMS adaptive filter. The main goal of these transform domain adaptive filters is the gain in convergence rate possible as demonstrated above (via the reduction in eigenvalue disparity of the input autocorrelation matrix). These algorithms function as depicted in Figure 2.7. The input $x(n)$ is fed to a delay line which stores the current input and the last $M-1$ values of it. An M point transform such as the DFT or the DCT is computed at every sample. Each transform component is weighted by one adaptive coefficient and the weighted transform components are added to form the current output $y(n)$. The error is obtained from $d(n)-y(n)$ and used to update the weight vector. Each coefficient of the weight vector is updated using its own step size determined from:

$$\mu_i = \frac{\mu}{E(z_i^2(n))}, \qquad i = 0, 1, \cdots, M-1. \tag{2.23}$$

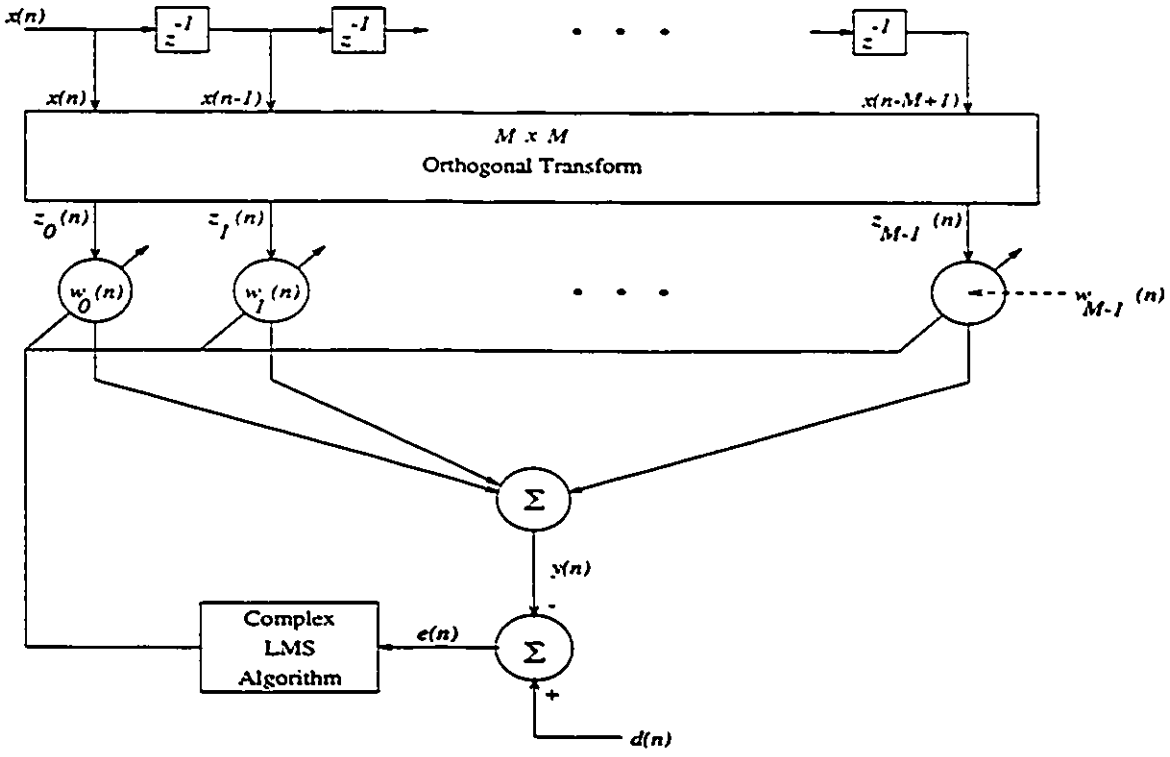

Figure 2.7: Transform domain adaptive filter.

The combination of the tapped delay line and the $M \times M$ orthogonal transform can be implemented as a filter bank. To calculate a large transform at every sample would be very computationally expensive, although efficient methods do exist for the computation of transforms such as the DFT and DCT. An adaptive channel equalization example from [42] compared the performance of the LMS algorithm and the sliding window transform domain adaptive filter. The transform domain filter required twice as many computations as the LMS adaptive filter. In [42], a DCT of size 32 was used. In acoustic echo cancellation, much larger transforms would be required and this would greatly multiply the additional computational complexity required by the sliding window transform domain adaptive filter. In spite of the lure of faster convergence such a structure is simply not feasible for acoustic echo cancellation.

Variants of the above structure which employ more than one tap per transform component have been introduced [42, 43]. The structure proposed in [42] is based on

the use of a transform with a sparse adaptive filter in each band. The sparse adaptive filters, whose spectrums repeat themselves within the fullband $(0 - 2\pi)$, make use of the narrowband nature of the transform signals in each frequency band. This structure achieves a speed of convergence close to that of a transform domain LMS structure while saving considerably on the computations. The reduced computational complexity, relative to the sliding window transform domain adaptive filter, is a result of the use of a much smaller transform size M. For example, a time domain adaptive filter with L taps can be replaced by a sliding window transform domain adaptive filter with a transform of size $M = L$. The adaptive filter structure of [42] could use a transform of size N with K taps for every transform component, such that $N \cdot K \geq L$. This allows the possibility of N being considerably smaller than M. However, the structure still requires considerably more computations than the fullband LMS structure. The adaptive filter structure of [42] also has the advantage that it adds less delay to the signal path than the transform domain LMS structure, again due to the smaller transform size used. As already discussed, applications such as telephony may have very stringent requirements for the added delay of the adaptive filter. Typically, one would want to restrict the delay to less than 10 ms which is 80 samples at a sampling rate of 8 kHz.

2.8 Subband Adaptive Filtering Algorithms

The use of subband adaptive filtering is motivated by two goals: faster convergence and reduced computational complexity [50, 51, 55]. Subband adaptive filtering is based on the use of an M band filter bank. A basic M band filter bank is shown in Figure 2.8. The bank of filters $H_i(z)$ form the analysis filter bank. The outputs of each of the filters in the analysis filter bank are decimated by a factor R where

$$R \leq M. \tag{2.24}$$

The second filter bank, composed of the filters $G_i(z)$ in Figure 2.8, is called the synthesis filter bank. The inputs to the filters $G_i(z)$ are upsampled by R before filtering. The

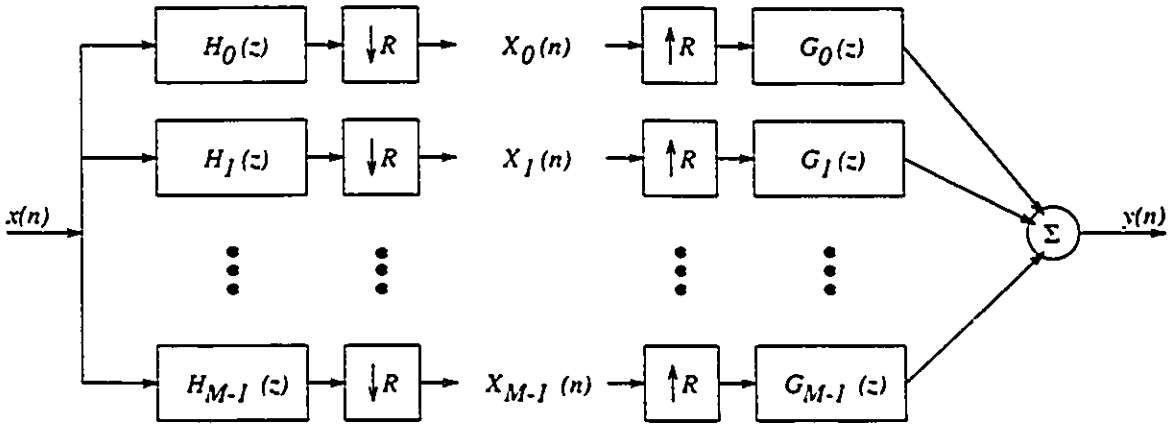

Figure 2.8: Basic M-band filter bank structure.

outputs of the filters $G_i(z)$ are added together to form the output of the filter bank $y(n)$. The relationship of Equation 2.24 must be observed in order for the output $y(n)$ to be an alias-free approximation to the original signal $x(n)$. In fact it is possible to choose the filters $H_i(z)$ and $G_i(z)$ so that $y(n)$ is merely a delayed version of $x(n)$. Filter banks satisfying this property are called perfect reconstruction (PR) filter banks [44, 47].

The basic structure of a subband acoustic echo canceller is shown in Figure 2.9. The speech input is split into subbands by the analysis filter bank as is the echo signal $d(n)$. The subband signals $D_i(n)$ and $X_i(n)$ are decimated by R before going through the adaptive filter. The NLMS algorithm is performed in each subband independently on the decimated subband signals. The output $e(n)$ of the system is formed by first upsampling the subband error signals $E_i(n)$ by R and then passing the upsampled $E_i(n)$'s through the synthesis filter bank.

Faster convergence than the LMS algorithm is expected when the input $x(n)$ is a highly correlated signal. This can be expected intuitively since within each subband of the subband adaptive filter, the input for that subband $X_i(n)$ contains only a portion of the spectrum of the original input. It is likely that this portion of the spectrum is closer to being white, in other words flat, than the whole spectrum when considered all at once. The process of decimating aliases the signal and spreads out the originally bandlimited signal into a more or less fullband signal. This is as well a sort of "whitening" process

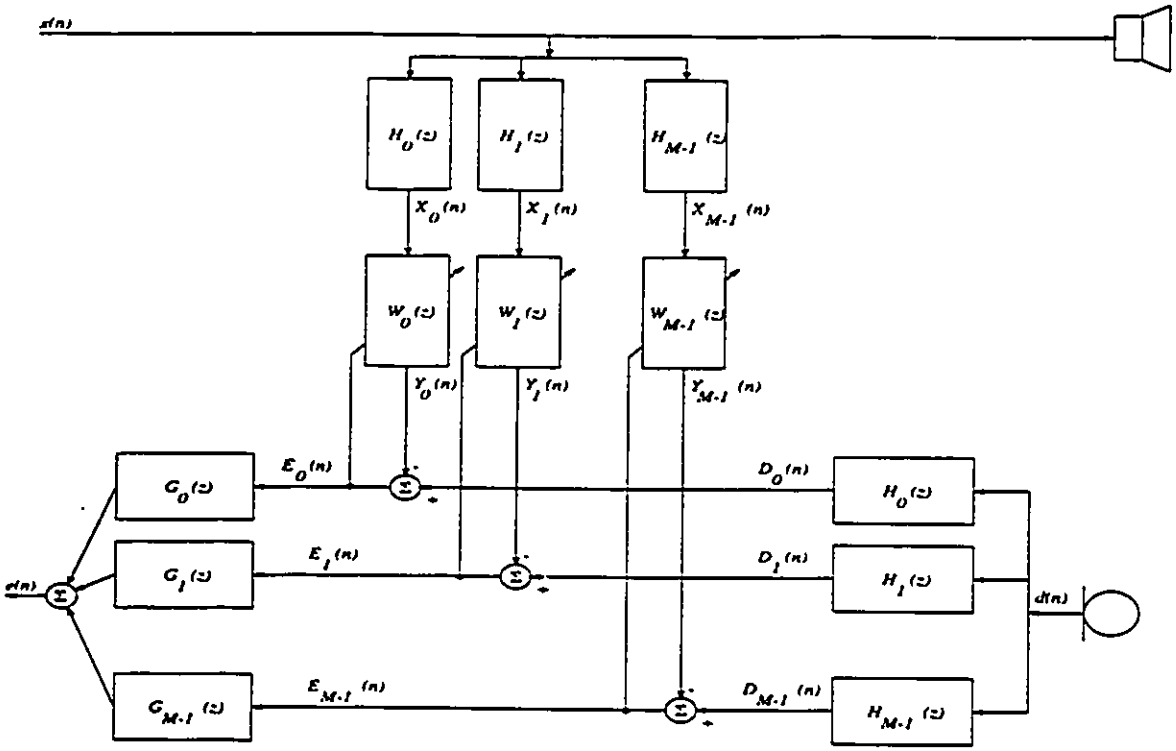

Figure 2.9: Subband acoustic echo canceller.

which may serve to speed up the convergence of the adaptive filters in each band. The length of the adaptive filter in each subband is usually chosen to span the same length of time as the equivalent fullband adaptive filter. Since the subband adaptive filters are operating at a lower sampling rate fewer taps are required in each subband. This also provides the opportunity for faster convergence since it has already been mentioned that convergence speed is inversely proportional to adaptive filter length. Results in [48] show a much faster rate of decay for the error for a subband adaptive filter as compared to a fullband adaptive filter for a correlated input signal. The input signal used was a stationary noise signal with a speech-like spectrum.

The savings in computational complexity of a subband adaptive filter over a fullband adaptive filter are dependent on the the sampling rates used [50]. If the subband signals are maximally decimated ($R = M$), then the subband structure will provide the most savings. Obviously at some decimation rate less than the maximum possible ($R <$

M), the subband structure becomes more computationally intense than the fullband LMS algorithm, and if there is no decimation whatsoever ($R = 1$), then the subband adaptive filter will require more than M times as many computations per sample as the fullband LMS algorithm. One must also not forget the computational burden of the three filter banks required by the subband adaptive filter. Multirate filter banks may be implemented as polyphase structures which reduces their computational burden considerably [46].

Current subband adaptive filters suffer mostly from one problem. The steady state error or misadjustment of the subband adaptive filter is significantly greater than that of the fullband LMS filter. Attempts to overcome this limitation have been only partially successful and usually sacrifice performance in other areas.

The misadjustment problems of the subband adaptive filter are due mainly to the nonideal nature of the filter bank structure itself. The subbands are in fact not completely independent of each other. The filters used in the filter banks have finite stopband attenuation and adjacent filters overlap in the region of their transition bands. In general, the more independent the subbands are with respect to each other, the better the subband adaptive filters will perform. The overlapping of adjacent filters in the frequency domain leads to aliasing in the output error spectrum if the subband adaptive filter is decimated maximally ($R = M$) [50, 51]. A look at the error spectrum after convergence will show that the adaptive filter has failed to identify the room response in the frequency regions which correspond to the overlapping areas of adjacent filters in the filter bank. The maximally decimated structure is the most desirable structure due to its computational savings and due to the fact that the process of maximum decimation leads to the most "white" signals in each of the subbands. A solution to this problem of aliasing proposed in [49] and [50] is to use adaptive cross-filters between adjacent subbands. It was shown that the inclusion of such cross-filters in the subband adaptive filter structure cancelled the aliasing and led to an adaptive filter capable of achieving misadjustment levels similar to a fullband LMS scheme. However, the cross-filters significantly degraded the convergence speed of the subband adaptive filter to the

point where convergence was slower than the fullband LMS [50]. As well, the addition of cross-filters counteracts the computational savings originally claimed for a maximally decimated subband adaptive filter. The filters making up the filter bank may also be optimized [53, 54, 62, 64]. The optimization criterion used is usually the maximization of the stopband attenuation for a given transition bandwidth. Improved misadjustment may result if optimized filters are used but aliasing will not be cancelled completely and the need for cross-filters remains.

This leaves subband adaptive filters with decimation rates lower than M, known as oversampled filter banks, as the way to go [51, 63]. With such a scheme aliasing, as described above, can be avoided and some decrease in the computational complexity over a fullband adaptive filter can still be achieved [50]. There are still problems with such a structure. In Section 2.3, convergence speed was linked to the eigenvalue spread of the input autocorrelation matrix. In a subband structure the autocorrelation matrix of a subband signal may in fact have a larger eigenvalue spread than the fullband signal due to bandpass filtering. The decimation by $R \leq M$ does not "whiten" the signal since there are only R copies of the spectrum which has a bandwidth of $2\pi/M$ and $2\pi R/M < 2\pi$. According to [5], the greater eigenvalue spread is not necessarily that detrimental since most of the eigenvalue disparity is due to the regions of the spectrum near the transition bands. In other words, it helps that the input and desired response are bandlimited by the same filter. The subband adaptive filters converge well in their midband regions and converge poorly in their transition band regions, however the poorer convergence in these transition band regions is masked somewhat in the overall error performance since the synthesis filter bank tends to attenuate the subband signals at these frequency points. The deterioration in convergence of subband adaptive filters near the transition bands is more clearly detailed in [55]. It is there proposed to use analysis filters with larger bandwidths than the synthesis filters. Thus, the analysis filters are very "overlapping" and the synthesis filters are not so "overlapping". The transition band of the analysis filters, where the adaptive filters perform poorly, will be more completely filtered out by the synthesis filters and the convergence of the system

as a whole should be improved.

All the subband adaptive filter structures discussed up until now have used the LMS algorithm. A subband acoustic echo canceller using an RLS algorithm is proposed in [61]. This structure seeks to achieve convergence performance comparable to a fullband RLS adaptive filter while avoiding the massive amounts of computations usually required by an RLS algorithm. The savings in computations are afforded by the nature of the subband adaptive filter. If enough bands are used, the decimation rate is great enough, and a fast RLS algorithm is used, a subband RLS structure with a computational complexity comparable to a fullband LMS adaptive filter can be implemented. Additionally, the speed of convergence of the subband RLS algorithm will be superior to that of the fullband LMS algorithm. Of course, the stability of the fast RLS algorithm used is another issue to be resolved and as mentioned previously, the RLS algorithm may not track the desired response statistics as well as the LMS algorithm.

2.9 IIR Adaptive Filtering Algorithms

The vast majority of work on acoustic echo cancellers has concentrated on FIR adaptive filters. This is because the adaptation of FIR filters is much better understood than the adaptation of IIR filters. In a problem such as acoustic echo cancellation where the room transfer functions being modelled are very long, one might expect that an adaptive IIR filter would achieve similar or better performance than an FIR filter while using significantly fewer coefficients. However, the efficiency of IIR filters is associated with other disadvantages when implemented in an adaptive mode.

IIR adaptive filters fall into two basic categories: equation error (EE) and output error (OE) algorithms. OE algorithms are a direct implementation of an IIR filter and as such they have numerous problems with stability, local minima and convergence speed [67]. The EE algorithms on the other hand result in a biased estimate of the desired response [67]. The estimates become biased when the desired response $d(n)$ contains not only the response of the room to the input speech $x(n)$ but also another

uncorrelated signal. An uncorrelated noise signal is always present in a hands-free communication system and thus the EE algorithm will be unsuitable for acoustic echo cancellation. This leaves the OE approach as the only viable way to implement an IIR adaptive filter and this has yet to be used in an acoustic echo cancellation situation with satisfactory results.

An approach that has received some attention [69, 70] is the use of fixed pole adaptive filters (FPAF). In such filters the number and locations of the poles must be assigned based on a priori information. This would require knowing the resonance properties of the room to be modelled or making several measurements of the room transfer function prior to any adaptive modelling. Some gains in computational complexity and convergence speed were claimed in [70] for a FPAF scheme. The poles were calculated by measuring the room response ten times with the loudspeaker-microphone orientation changed for each measurement. After the poles had been calculated the adaptive FIR part of the filter could function to identify the room response. Such a scheme would require significant overhead in terms of calculations of the pole locations. As well, results in [70] are for a very limited bandwidth because at low sampling frequencies the poles are expected to have lower orders. It is unclear how the performance of this system would be affected by the use of the typical telephone sampling rate of 8 kHz. Stability of the system is another concern which has not been fully addressed.

Another question that must be asked is how much gain in performance can be expected from the use of IIR adaptive filters? This issue has been addressed in [68] with the conclusion that for a given number of parameters an OE IIR model can perform only marginally better than an FIR model with the same number of parameters. The comparison was based on the ERLE achieved and says nothing about comparative convergence rates. The performance of the EE IIR modelling was found to be worse than the FIR model. In summary it must be concluded that due to the performance problems of IIR adaptive filters and due to the uncertainty about the gains to be achieved by the use of IIR adaptive filters, FIR adaptive filters remain the preferred choice for the problem of acoustic echo cancellation.

2.10 Subjective Hearing Criteria

The motivation for this thesis is presented in this section. The goal of developing an adaptive filtering algorithm which can weight the frequency spectrum of the error will be undertaken. The reasons for doing so will be exposed shortly.

2.10.1 Subjectively Based Coding Algorithms

There have been several efforts in the field of audio coding to make use of subjective properties of the ear in order to maximize coding gain while preserving the subjective quality of the signal being coded [72, 73, 75]. If a signal can be coded, transmitted and reconstructed with no loss in subjective quality then the coding system is known as a "transparent" coder. The goal of the audio coding algorithms is to retain the transparency of the coder while exploiting certain properties of the human ear in order to reduce the bit rate as much as possible.

There are two essential properties of the ear which provide for this exploitation. These properties are the ear's sensitivity to sounds of different frequencies and secondly, the inability of the ear to distinguish between sounds of two different frequencies if these frequencies are sufficiently close to one another. The first property is illustrated by the absolute hearing threshold curve of Figure 4.9 on page 124. This curve shows the tone sound pressure level (SPL), in dB, required for audibility (by human listeners) versus tone frequency [72]. As the tone frequency decreases (below 1 kHz), the tone must have a greater SPL to be audible. As the tone frequency increases (above 5 kHz), the same is true. Thus, this curve indicates that the human ear is most sensitive to signal frequencies in the range of 1 kHz to 5 kHz. The second property, known as the masking threshold, indicates that if a strong signal is present at some signal then the ear will be unable to hear another signal with a nearby frequency if that second signal does not exceed some relative value. In other words, weaker signals are masked by stronger signals to a degree inversely proportional to the frequency difference between the weaker signal and the stronger signal.

Several efforts in the domain of audio coding [72, 73, 75] have used models of the above two properties of the ear to achieve significant coding gains for transmission of audio signals. The coding gain is defined as the ratio of the reduced bit rate to the original bit rate. In [72], a coding gain of about 7 is achieved using a masking threshold model of the human ear. The basic idea behind these coding algorithms is to split the original signal into subbands, usually using a maximally decimated, perfect reconstruction filter bank, and to then assign bits to each subband for the quantization of each subband signal on the basis of the masking model. In bands where there is significant input energy one may be able to use fewer bits since quantization noise generated by the use of a small number of bits will be masked. The masking model also states that a strong signal in one band will mask signals in the next band so that the calculation of the masking threshold for each band is based on a spectral analysis of the input signal. The masking threshold for each band is the basis by which the number of bits for quantization is assigned.

This algorithm is also applicable to speech coding. However, for telephone speech, the bandwidth of interest is from 0 Hz to 4 kHz whereas with audio the bandwidth was 0 Hz to 20 kHz. Results for speech coding using this method have not yet been reported.

2.10.2 Frequency Weighting Adaptive Filters

The psychoacoustically based coding algorithms described above give motivation to apply some of the same psychoacoustic principles to the area of acoustic echo cancellation for hands-free telephony. The idea remains essentially the same: to concentrate the error (echo) in frequency regions where the human ear is less sensitive to them. This requires the development of an adaptive filtering algorithm which shapes the echoes spectrum in accordance with the sensitivity of the human ear. The development of such an algorithm will be the focus of the bulk of this thesis. The adaptive filters described thus far seek only to minimize the mean square error in a mathematical sense.

It is impractical to employ the masking threshold model of the frequency response of the human ear for the acoustic echo cancellation case. Calculation of the short-

time Fourier transform and the masking thresholds for each subband would present an enormous computational burden to the adaptive filtering algorithm. These calculations may be quite justifiable in the case of coding since all of the complexity takes place on the coding side of transmission. Decoding remains a simple process and thus a system where one complex, powerful coder is capable of performing these calculations while an infinite number of simple decoders can decode the transmissions is possible. Such is not the case with acoustic echo cancellation. DSP chips used to implement acoustic echo cancelling algorithms have a limited amount of usable memory and a limited operation speed and any adaptive filter implemented on the chip is restricted in complexity by these constraints. Thus, computational simplicity is a goal in adaptive filtering algorithm design.

The other problem, and an equally if not more daunting one at that, is that the masking threshold levels are constantly changing. The masking threshold spectral curve at any given instant is directly linked to the input signal statistics at that time. Adaptive filters take a finite amount of time to converge and if the parameters of the adaptive filter are constantly changing in order to match the nonstationary characteristics of the masking threshold curve, the adaptive filter may not be able to track the input as well and the performance of the adaptive filter is expected to suffer.

Thus, for the purposes of this thesis, a simpler model for the frequency weighting must be derived in order to apply perceptual hearing criteria to the echo spectrum of the adaptive filter. The absolute hearing threshold curve in Figure 4.9 on page 124 provides this model. Unlike the masking thresholds discussed earlier, this curve is simply a measure of the ear's sensitivity to sounds of different frequencies and is not related to input signal statistics. Thus, the same weighting will be applied to the adaptive filter error at all times and this should not harm the convergence of the adaptive filter.

Two algorithms are proposed and investigated for the purpose of weighting the echo spectrum according to the psychoacoustic principles discussed above. In Chapter 3, the filtered-E LMS algorithm is considered. The filtered-E LMS algorithm is simply the full-band LMS algorithm with a weighted error. The object of weighting the error is to give

echoes with frequencies deemed more important (subjectively speaking) a higher weighting than less important echoes. In the investigation of the filtered-E LMS algorithm, it will be discovered that the weighting filter applied to the error does not significantly affect the shape of the final echo spectrum when modelling a speakerphone-room response while changing the convergence properties of the adaptive filter. A subband adaptive filtering structure is then proposed in Chapter 4 which successfully achieves frequency weighted echoes by adaptively allocating the available number of filter taps to the M bands of the subband adaptive filter. The performance of the adaptive tap assignment subband adaptive filter is investigated from both objectively and by means of simple listening tests performed by the author. The proposed algorithm's performance is compared with an existing adaptive tap assignment algorithm and with the standard fixed tap assignment subband adaptive filter.

2.11 Summary

This chapter has presented an overview of the field of acoustic echo cancellation. The nature of the acoustical coupling between the loudspeaker and the microphone of a hands-free communication set was discussed. Other problems inherent in the implementation of a hands-free communication link were also briefly discussed. Perhaps the oldest method of acoustic echo cancellation, gain control, was described before presenting acoustic echo cancellation in an adaptive filtering context. The basic structure of the adaptive filter-based acoustic echo canceller was presented and the terminology and filtering equations for this structure were given. The LMS algorithm was presented as the most common approach to updating the coefficients of the adaptive filter.

The limitations of the LMS based adaptive filter in an acoustic echo cancellation application were discussed on the basis of convergence speed and steady state mean square error. Three main factors which impede the LMS algorithm from achieving satisfactory convergence and steady state error were introduced. These were the length of typical room responses, the statistics of the input (speech) in an acoustic echo cancel-

lation environment, and loudspeaker nonlinearities. The tracking performance adaptive filters was introduced as another area of comparison between various adaptive filtering algorithms.

Algorithms which seek to improve upon the convergence speed of the LMS algorithm were introduced and compared to the LMS algorithm in terms of improvement in convergence speed and tracking performance. The algorithms covered were the normalized LMS (NLMS) algorithm, modified LMS algorithms, recursive least squares (RLS) algorithms, transform domain algorithms, subband adaptive filtering algorithms, and IIR adaptive filtering algorithms.

The chapter ended with a section on the motivation for this thesis. That motivation is to develop an algorithm which seeks to minimize the audibility of the adaptive filter error rather than just minimize the adaptive filter error in a mathematical sense. Coding algorithms which use models of the human auditory system to achieve coding gains were presented as a backdrop to the introduction of similar ideas in the field of adaptive filtering. The two algorithms that will be used to try to shape the error in accordance with the perceptual criteria of the human ear were introduced. They were the filtered-E LMS algorithm and the subband adaptive filtering algorithm with adaptive tap assignment.

CHAPTER 3

Filtered-E LMS Algorithm

In the conventional LMS algorithm the adaptive filter error $e(n)$, as given by Equation 2.4, is used directly to update the weight vector as in Equation 2.5. In this chapter, the feasibility of weighting the error in the frequency domain (based on psychoacoustic criteria) so that the bulk of the error energy is placed in less audible frequency ranges is investigated. This would improve the subjective quality of the degraded speech signal regardless of the effect on the SNR. The advantage of using this scheme is its simplicity. A filter would be used to weight the error, requiring only a few more computations per sample in addition to the computations normally required by the LMS algorithm.

Simulation results for this algorithm will be presented and its performance discussed in due course. This algorithm will be applied to two adaptive filtering environments: line echo cancellation (LEC) and acoustic echo cancellation (AEC). In the LEC simulations, the proposed approach provides significant performance improvement over the LMS algorithm in terms of frequency weighting. However, in the AEC simulations, the filtered-E LMS algorithm did not achieve better echo cancellation at emphasized frequencies than the LMS algorithm. Further simulations show that the performance of the filtered-E LMS algorithm is very dependent on the type of response being modelled. Important factors in the filtered-E LMS algorithm performance are the level of undermodelling and the dominant features of the plant response (such as its smoothness and whether or not it contains regions of transition). The performance of the filtered-E

LMS algorithm will be examined from several perspectives. Henceforth, the filtered-E LMS algorithm will also be referred to simply as the FELMS algorithm.

The unknown responses and input speech sequences used in the simulations of this thesis are displayed in Appendices A and B. Two speakerphone-room responses were used, and these will be denoted as rooms *WBN13* and *mc3023*. Descriptions of these speakerphone-room responses are given in Appendix A. The speakerphone-room responses were sampled at 8 kHz. The method of speakerphone-room impulse response measurement is described in Appendix A. One hybrid response was used and this will be denoted as hybrid *hybrid1* (also sampled at 8 kHz). Three different speech input sequences were used, denoted as speech sequences *m98*, *m54* and *htf*. These speech sequences are plotted in Appendix B along with their average power spectra. All speech sequences were recorded in an anechoic room and obtained from Bell Northern Research [65]. The sampling rate for the recordings was 16 kHz. The sequences were subsequently decimated by 2 using the program `decimate` from `Matlab`. The program `decimate` first filtered the sequences with an anti-aliasing, lowpass, linear phase FIR filter and then decimated the signals to bring the sampling rate to 8 kHz. The labels *m98*, *m54* and *htf* are simply used to differentiate between the three speech sequences.

3.1 Origins of Filtered-E LMS

The filtered-E LMS algorithm has its origins in the filtered-X LMS algorithm. This section discusses the filtered-X LMS algorithm and the motivation for the modifications to the filtered-X LMS algorithm which make it the filtered-E LMS algorithm. The development of the filtered-X LMS algorithm arose from work in the field of inverse control. In inverse control, an adaptive filter is placed after the plant

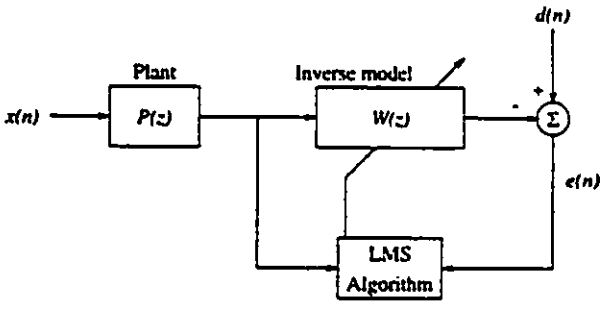

Figure 3.1: Adaptive inverse modelling of a plant $P(z)$ with LMS.

(unknown system) and is updated so as to provide an inverse to the plant, as illustrated in Figure 3.1 (from [13]). However, if the estimate of the plant output is very noisy and this noise is uncorrelated with the plant input, the adaptive solution will not be a close approximation to the delayed inverse. In such cases where the noise is significant and may cause unsatisfactory error in the inverse solution, the filtered-X LMS approach may be applied. In the filtered-X LMS approach, the adaptive inverse filter is placed in front of the plant as shown in Figure 3.2. The input to the LMS algorithm must also be filtered by the plant transfer function (see [13]).

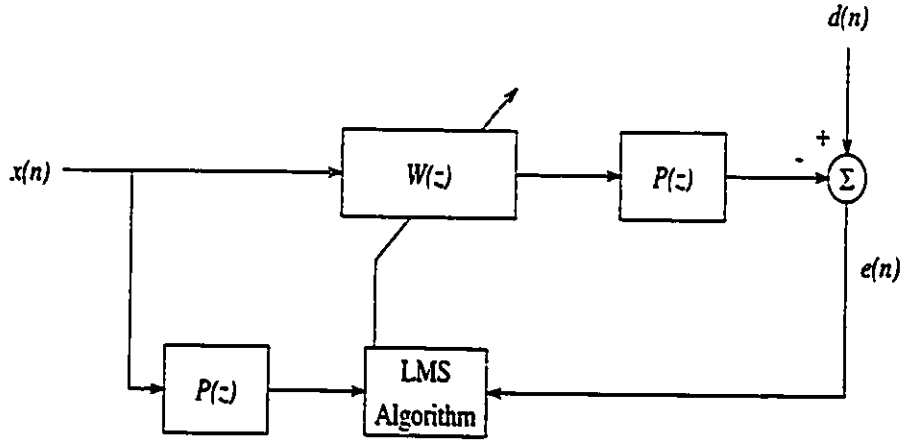

Figure 3.2: Adaptive inverse modelling of a plant $P(z)$ with filtered-X LMS.

In the inverse control application discussed above, the desired response $d(n)$ is simply a delayed version of the input: $d(n) = x(n - \Delta)$. The delay is required since both the plant $P(z)$ and the adaptive inverse model $W(z)$ are causal systems. The filtered-X LMS algorithm has come to be used in other applications as well, most notably, in active noise control. The adaptive filtering set-up for active noise control is shown in Figure 3.4. Note that the notation of Figure 3.4 is somewhat different than that of Figure 3.1. What can be inferred from Figures 3.1 and 3.4 is that the term "filtered-X LMS" really refers to a modified LMS algorithm which has an additional transfer function applied to the adaptive filter output $y(n)$ and to the reference input to the LMS algorithm $x(n)$.

The performance of the filtered-X LMS algorithm in inverse control is claimed to be very comparable to that of the LMS algorithm [13]. In the case of active noise control, the use of the filtered-X LMS algorithm is required. The cancellation path transfer function is an estimate of the transfer function between the adaptive filter loudspeaker (which tries to cancel the noise) and the error microphone (which picks up the remaining noise at some location physically distinct from the loudspeaker), as shown in Figure 3.3. An example of a noise cancellation application is the cancellation of motor noise inside a car. Here, the noise source is of course the car motor and the input to the adaptive filter might be taken from under the hood. The loudspeaker and error microphone would be placed inside the car. The purpose of the loudspeaker is to cancel noise inside the car (the noise is generated outside of the car). The error microphone picks up the resultant signal and it is the objective of the algorithm to minimize the resultant signal. However, since the loudspeaker and error microphone are physically distinct, there is a transfer function between the two and the actual error at the loudspeaker is modified before it is picked up by the error microphone. An estimate of this transfer function must also be applied to the input, in accordance with the derivation of [13]. The performance of the filtered-X LMS algorithm depends upon the accuracy of the estimate of the cancellation path transfer function, as investigated in [26, 27].

A modification to the filtered-X LMS algorithm, called the filtered-E LMS algorithm, has been proposed in [28]. The filtered-E LMS algorithm calls for an additional filter $F(z)$ to be added to the adaptive filtering set-up of Figure 3.4, as shown in Figure 3.5. Two different filters are used in place of $F(z)$ in [28]. One mimics the response of the human ear to various frequencies and the other is a simple bandpass filter. The frequency range of interest in [28] is from 0 to 1000 Hz. The purpose of the filter $F(z)$ is to weight the error in order to emphasize psychoacoustically important frequencies. This is intended to make the algorithm perform better at these

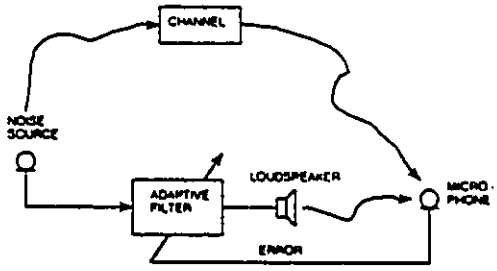

Figure 3.3: Active noise cancellation set-up.

frequencies (achieve lower error at these frequencies). Several figures in [28] show that the filtered-E LMS algorithm has achieved better noise cancellation than the filtered-X LMS algorithm at frequencies emphasized by the weighting filter $F(z)$ at the expense of less noise cancellation at frequencies de-emphasized by $F(z)$. However, the unknown system modelled in [28] is a simple low pass filter response and not a real noise cancellation set-up. It is precisely this type of frequency weighting capability that is the goal of this thesis and a variant of the filtered-E LMS algorithm, suitable for acoustic echo cancellation, will be described in the next section.

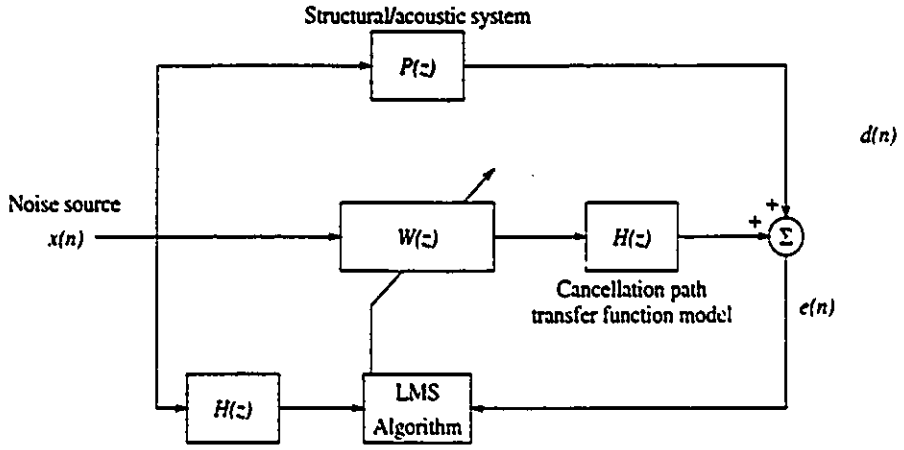

Figure 3.4: Active noise control system using the filtered-X LMS algorithm.

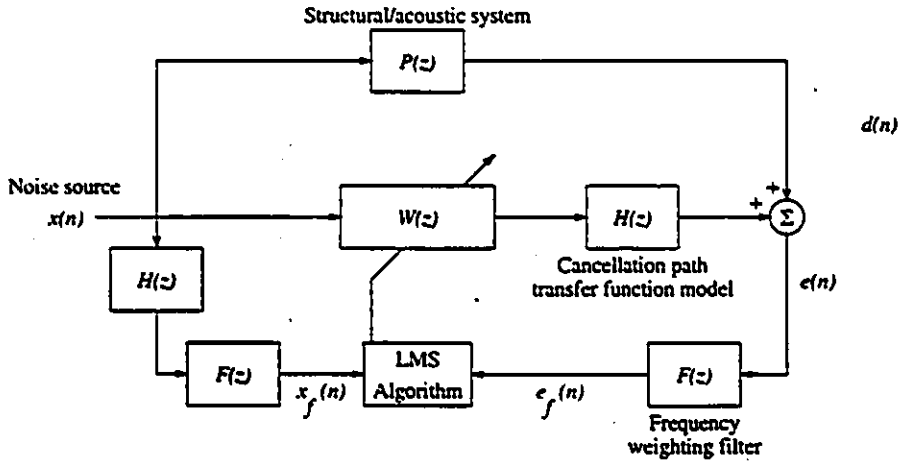

Figure 3.5: Active noise control system using the filtered-E LMS algorithm.

3.2 Filtered-E LMS Acoustic Echo Canceller

In an acoustic echo cancellation application, there is no fixed filter $H(z)$ which forms a necessary part of the error path of the adaptive filtering set-up, as there was for the active noise control application discussed above (see Figure 3.4). Thus the resultant filtered-E LMS adaptive filtering set-up, shown in Figure 3.6, is essentially identical to the structure of Figure 3.5, except that there is no cancellation path transfer function $H(z)$. The filter $F(z)$ takes the same role in both algorithms. The frequency response of $F(z)$ will reflect the relative importance of frequencies from a psychoacoustic point of view. Similar structures (such as the coupled LMS structure of [21]) have come to be known as filtered-X LMS structures in the adaptive signal processing literature though this is a misleading moniker due to the fact that the filtered-X LMS algorithm is a necessary structure in active noise cancellation due to the cancellation path transfer function. In reality, the filtered-X LMS algorithm in active noise control is the counterpart of the LMS algorithm in acoustic echo cancellation. If one used the simple LMS algorithm in active noise control, one would essentially be using the wrong error to update the adaptive filter. No such parallel exists in acoustic echo cancellation. In this thesis, the term "filtered-E LMS" will be used to describe the LMS algorithm with additional update path weighting filters.

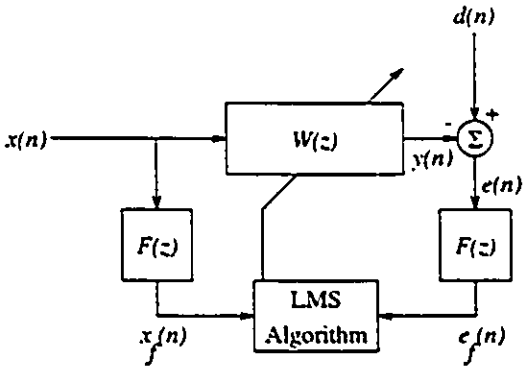

Figure 3.6: Structure of the filtered-E LMS adaptive filter.

A derivation of the weight vector update equation needed for the filtered-E LMS algorithm is now presented. In the conventional LMS algorithm the objective function to be minimized is the squared error:

$$J_{LMS}(n) = e^2(n). \tag{3.1}$$

In the FELMS case a weighted error must be minimized so the objective function be-

comes:

$$J_{FELMS}(n) = (F[e(n)])^2 \tag{3.2}$$

where $F[\cdot]$ is used to represent a transform or filtering operation on the sequence $e(n)$. For a gradient adaptive filtering algorithm such as the LMS or FELMS algorithm, the weight vector update can be expressed in a general form as:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu}{2}\frac{\partial J(n)}{\partial \mathbf{W}(n)}. \tag{3.3}$$

In Equation 3.3, $\frac{\partial J(n)}{\partial \mathbf{W}(n)}$ is an estimate of the gradient of the objective function. The step size factor $\mu/2$ controls the rate at which the adaptive filter searches the objective function surface in its attempt to reach the bottom where the objective function will be minimized. The factor 2 in the step size is chosen merely for convenience. The gradient estimate for the new objective function $J_{FELMS}(n)$ is

$$\begin{aligned}\frac{\partial J_{FELMS}(n)}{\partial \mathbf{W}(n)} &= \frac{\partial}{\partial \mathbf{W}(n)}(F[e(n)])^2 \\ &= 2F[e(n)]\frac{\partial}{\partial \mathbf{W}(n)}F[e(n)].\end{aligned}$$

The order of evaluation of the derivative and the transform above can be reversed since both are linear operators, giving

$$\begin{aligned}\frac{\partial J_{FELMS}(n)}{\partial \mathbf{W}(n)} &= 2F[e(n)]F[\frac{\partial}{\partial \mathbf{W}(n)}e(n)] \\ &= 2F[e(n)]F[\frac{\partial}{\partial \mathbf{W}(n)}(d(n) - y(n))] \\ &= -2F[e(n)]F[\frac{\partial}{\partial \mathbf{W}(n)}\mathbf{W}^T(n)\mathbf{X}(n)] \\ &= -2F[e(n)]F[\mathbf{X}(n)].\end{aligned}$$

The elements of $\mathbf{X}(n)$ were given in Equation 2.2. Further defining

$$e_f(n) = F[e(n)] \tag{3.4}$$

$$x_f(n) = F[x(n)], \tag{3.5}$$

the filtered-E LMS weight vector update equation can then be expressed as:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu e_f(n)\mathbf{X}_f(n) \quad . \tag{3.6}$$

where

$$\mathbf{X}_f(n) = [\ x_f(n) \quad x_f(n-1) \quad \cdots \quad x_f(n-L+1)\ ]^T. \tag{3.7}$$

The above algorithm is a fixed step size filtered-E LMS algorithm. These results can easily be extended to create the normalized filtered-E LMS algorithm (NFELMS) which has the following weight vector update equation:

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu}{\|\mathbf{X}_f(n)\| + \sigma} e_f(n)\mathbf{X}_f(n) \tag{3.8}$$

where $\|\mathbf{X}_f(n)\|$ is the Euclidean norm of the vector $\mathbf{X}_f(n)$, defined in a manner analogous to Equation 2.13 and $\sigma > 0$. In Subsection 2.2.2, the bounds on the step size of the NLMS algorithm were stated as $0 < \mu < 2$. Experimental results to be presented in Section 3.5 will show that the same step size bounds do not apply to the filtered-E LMS algorithm. In fact, the filtered-E LMS algorithm tends to have a tighter bound on step sizes than the LMS algorithm. A theoretical bound on step size for the filtered-X LMS algorithm is derived in [27], and given by:

$$0 < \mu < \frac{2}{m_x^{(2,2)}(tr(\mathbf{\Lambda}) + 2\lambda_{max})\sum_{i=0}^{M-1} h_i^2} \tag{3.9}$$

where $m_x^{(2,2)}$ is the joint fourth order moment of the input process, $\mathbf{\Lambda}$ is the input autocorrelation matrix, λ_{max} is the largest eigenvalue of the input autocorrelation matrix, and h_i are the coefficients of the cancellation path transfer function $H(z)$ (as in Figure 3.4). Since the filtered-E LMS algorithm is virtually identical in structure to the filtered-X LMS algorithm, Equation 3.9 can also be used to predict the step size limits of the filtered-E LMS algorithm. The step size bound on the fixed step size LMS algorithm, in a similar form (from [13]), is given by:

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{3.10}$$

where λ_{max} is again the largest eigenvalue of the input autocorrelation matrix. What is evident from Equations 3.9 and 3.10 is that the step size of the filtered-X (and thus the filtered-E) LMS algorithm has several additional factors which may influence its step size bound.

The filter $F(z)$ in Figure 3.6 in effect causes a delay in the update path of the adaptive filter. Such delays in the update path can cause stability problems [16, 25]. Thus, it is desirable to use weighting filters with short delays in order to minimize the effects of delayed updates. In addition to stability problems, the delayed updates may harm the convergence and tracking performance of the adaptive filter [16].

The filtered-E LMS algorithm is expected to achieve better echo cancellation than the LMS algorithm at frequencies emphasized by the weighting filter $F(z)$. The filter $F(z)$ will create an error signal in which psychoacoustically important frequencies are dominant. Thus the weight vector updates of the filtered-E LMS algorithm are expected to focus on these important frequencies. The resultant echo is expected to be less disturbing to the human ear than the echo produced by the LMS algorithm since echo will be concentrated in less audible frequencies.

3.3 Choice of Weighting Filter

In this section the choice of the weighting filter $F(z)$ in Figure 3.6 is discussed. The filter $F(z)$ is to be chosen to emphasize frequencies of interest and de-emphasize other frequencies. The frequencies of interest are chosen according to the application at hand. For acoustic echo cancellation, the final echo signal is evaluated by the human ear and thus frequencies to which the ear is sensitive should be emphasized (to achieve better echo cancellation at these frequencies) and frequencies to which the ear is less sensitive should be de-emphasized. The levels of emphasis for the various frequencies should be chosen in accordance with empirically determined hearing sensitivity curves for the human ear. Since the application of interest is acoustic echo cancellation for hands-free telephones, the frequency range of interest is from 0 Hz to 4 kHz. One such curve is the "C-message" frequency weighting curve, shown in Figure 3.7. This curve is the result of extensive listening tests with a variety of talkers and listeners. This curve shows the sensitivity of the human ear to noise at specific frequencies. For instance, the curve shows that a noise signal at 200 Hz is 25 dB less disturbing than a similar signal at

1 kHz. Note that the C-message weighting curve is a combination of a typical telephone set frequency response with the low frequency response of the human ear [2]. This means that the signal to which the test subjects listened was played through telephone sets and thus these signals were modified by the transfer function of the telephone set.

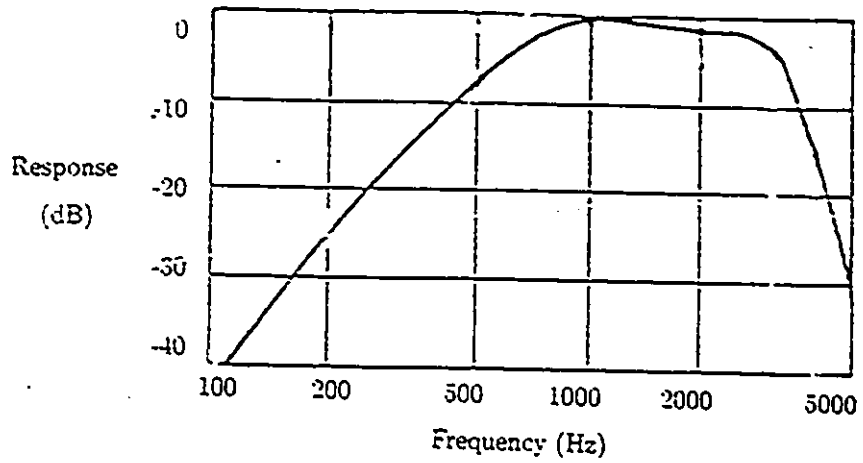

Figure 3.7: C-message frequency weighting response. (Figure A-1 from [30].)

In order to make use of the C-message frequency weighting curve as the weighting filter, a filter $F(z)$ which has a frequency response which matches or attempts to match the C-message weighting curve must be constructed. A 17 tap linear phase FIR filter designed using the multiband FIR design program `fir2` in `Matlab` is used. The time response and frequency response of the designed filter are shown in Figures 3.8 and 3.9, respectively. The frequency response of the filter $F(z)$ does not match the C-message weighting curve exactly. Note that the gain of the C-message approximating filter has been boosted to give the filter a gain of one for white inputs. In particular it does not fall off as quickly for low frequencies. One could make up for this somewhat by using a longer filter $F(z)$ but a longer filter is avoided here since longer filters produce a tighter bound on the step size for the FELMS algorithm, as discussed in Section 3.5.

The filtered-E LMS algorithm, as described above, has been used in a line echo cancellation application [29]. Some promising results in terms of frequency dependent performance of the filtered-E LMS algorithm were reported in this case. The filtered-E LMS algorithm will be evaluated using both weighting filters $F^{(1)}(z)$ and $F^{(2)}(z)$ (given in Equation 3.11). As well, the use of the filtered-E LMS algorithm will be considered in both line echo cancellation and acoustic echo cancellation applications.

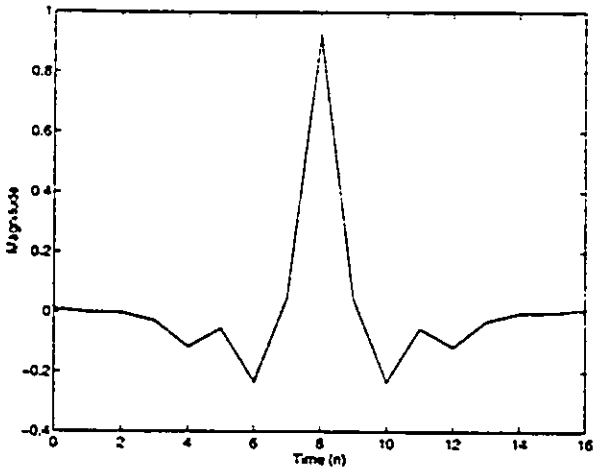

Figure 3.8: Time response of 17 tap C-message approximation filter.

In [29], the weighting filter used was a simple, two tap, high pass filter given by:

$$F^{(2)}(z) = \frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}} z^{-1}. \tag{3.11}$$

The frequency response of this filter is given in Figure 3.10.

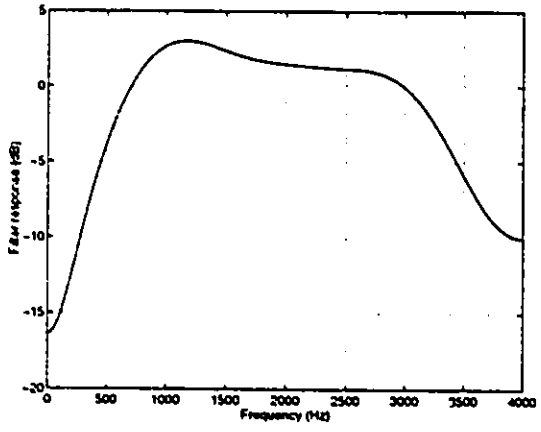

Figure 3.9: Frequency response of 17 tap C-message approximation filter.

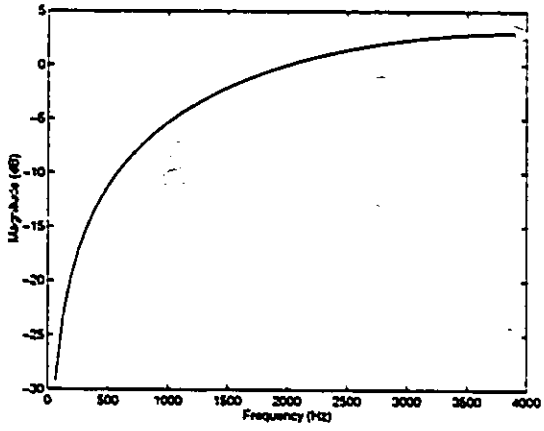

Figure 3.10: Frequency response of 2 tap high pass weighting filter.

3.4 Frequency Performance of Filtered-E LMS

In this section, the performance of the filtered-E LMS algorithm with white noise as input is examined in terms of the error spectrum after convergence. It was initially expected that where the weighting filter weighted the error heavily, better echo cancellation would be obtained than previously with a simple LMS algorithm. Consistent with this, the performance of the FELMS algorithm was expected to suffer, in comparison with the LMS algorithm, at frequencies where the error was de-emphasized.

In Figures 3.11, 3.12 and 3.13 error spectra after convergence for both the FELMS algorithm and the LMS algorithm are displayed. The adaptive filters of both algorithms were 300 taps long for the speakerphone-room modelling cases and 32 taps long for the hybrid modelling case. The unknown response was *WBN13* in Figures 3.11 and 3.12 and *hybrid1* in Figure 3.13. The LMS algorithm is of course the benchmark against which the performance of the FELMS algorithm will be compared. These three figures consider the performance of the FELMS algorithm for the two weighting filters $F^{(1)}(z)$ and $F^{(2)}(z)$ discussed above and for a room response and a hybrid response. Step sizes were chosen for both algorithms so that the FXLMS and LMS algorithms had the same steady state MSE after convergence. For the speakerphone-room modelling cases, the algorithms converged to the same MSE when the step sizes were equal. In the hybrid modelling case, the FELMS algorithm needed a step size of $7.0 \cdot 10^{-3}$ to converge to the same MSE as the LMS algorithm with a step size of $1.0 \cdot 10^{-2}$. When modelling response *hybrid1*, the LMS adaptive filter was allowed to adapt for 10000 iterations and the FELMS for 30000 iterations. When modelling response *WBN13* and *mc3033*, LMS was allowed to adapt for 10000 iterations and FELMS for 50000 iterations.

Note that in the case of the room response *WBN13*, the frequency weighting performed on the adaptive filter error has failed to give a significant frequency weighting to the final error spectrum. In the case of the hybrid however, a very significant weighting has taken place. Figures 3.14 shows the error spectrum when room *WBN13* from Figure 3.11 is replaced with room *mc3033*. Note that room *WBN13* has a more lowpass

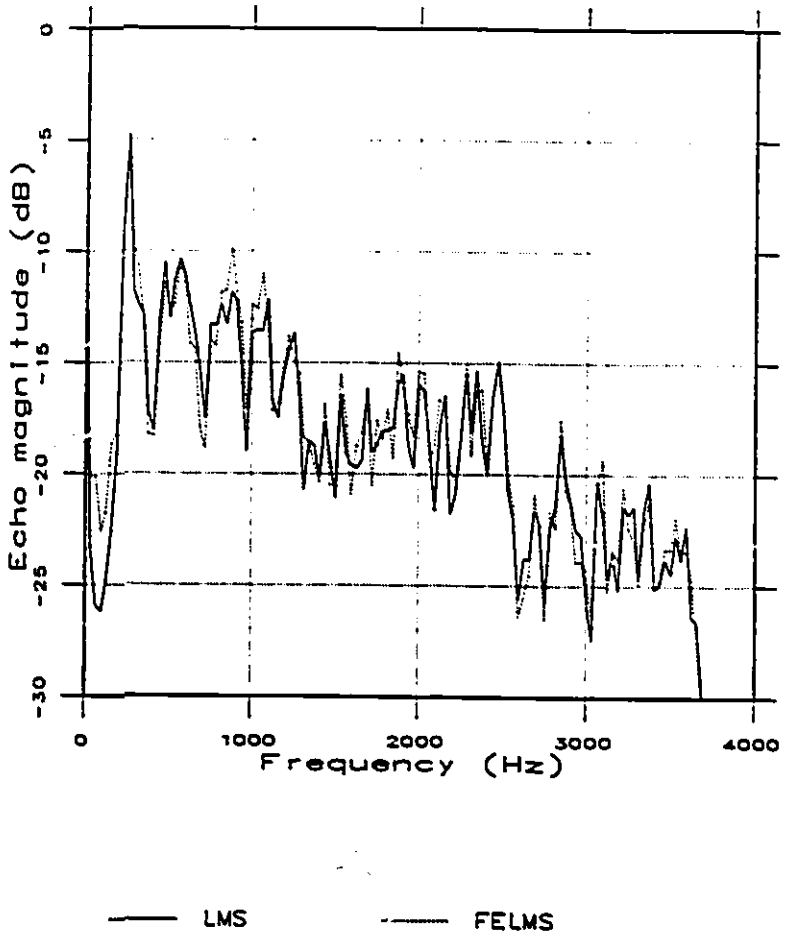

Figure 3.11: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: *WBN13*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$ (both algorithms).

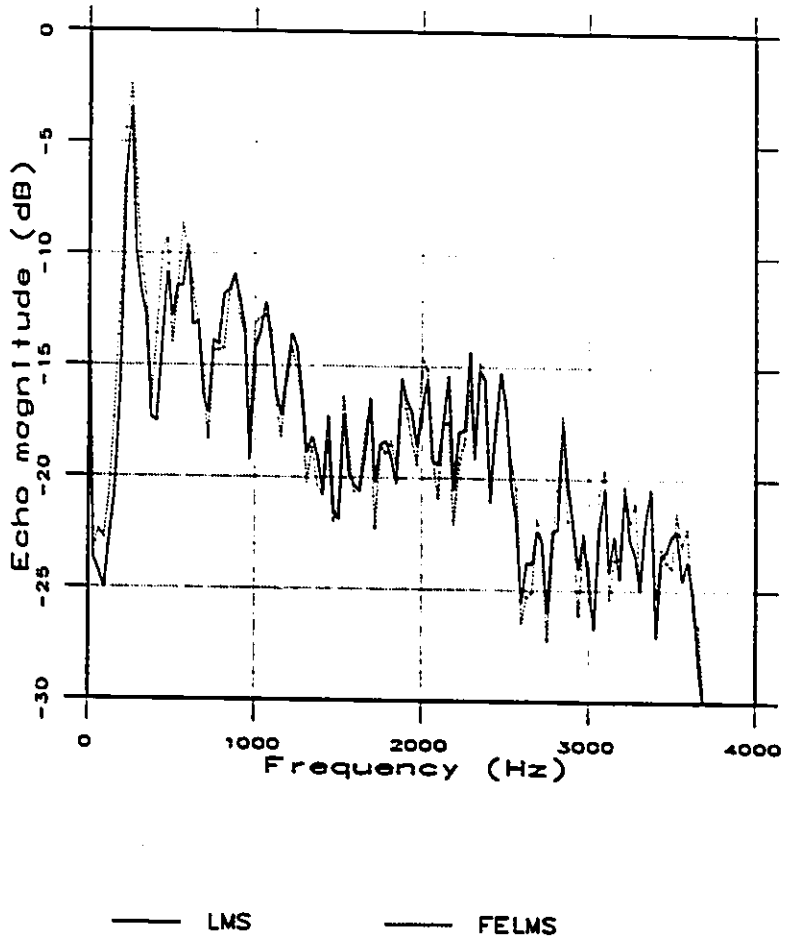

Figure 3.12: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: *WBN13*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$ (both algorithms).

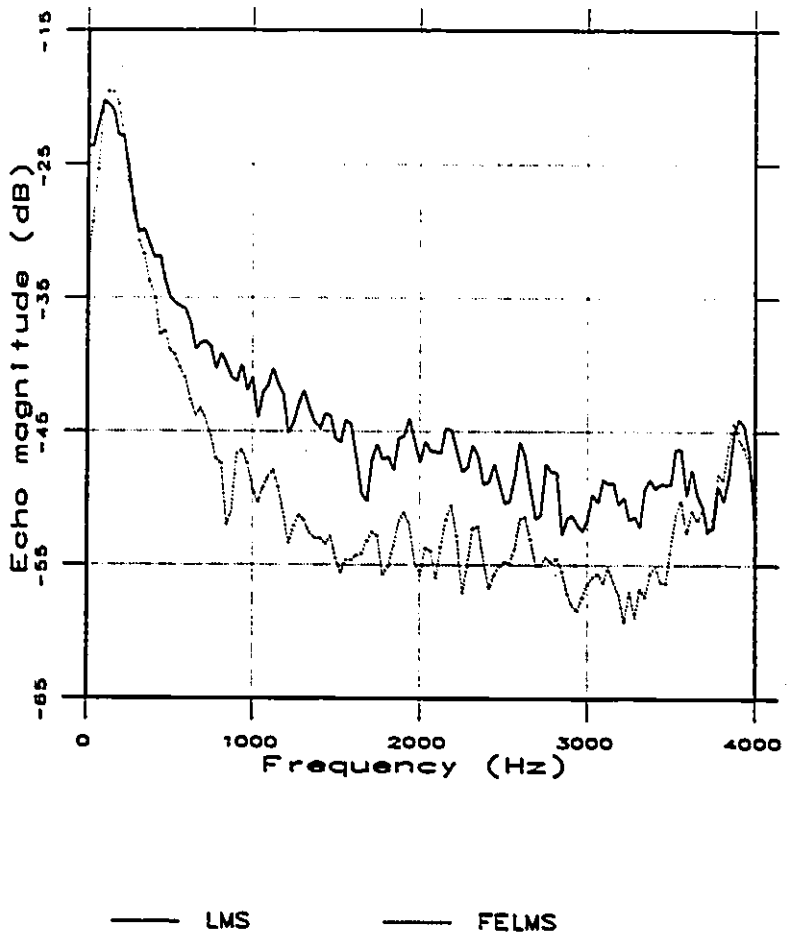

Figure 3.13: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-2}$ (LMS) and $\mu = 7.0 \cdot 10^{-3}$ (FELMS), $L = 32$ (both algorithms).

frequency response characteristic than room *mc3033*. In this case as well, the FELMS algorithm fails to provide any frequency weighting of the error. The effect of the frequency response characteristics of the unknown system will surface in the discussions of Sections 3.7 and 3.8.

3.5 Convergence of Filtered-E LMS

In Section 3.4, the FELMS algorithm was unable to effect any significant frequency weighting to the error spectrum in the case of the speakerphone-room responses. If the convergence speed of the FELMS algorithm relative to the LMS algorithm is examined, it can be seen that the effect of the frequency weighting filter $F(z)$ has been to slow the convergence of the weight vector for de-emphasized frequencies and speed up convergence for emphasized frequencies. Of course, since there is still a bound on the maximum step size, the convergence cannot be sped up significantly for any frequency and thus the main effect of the FXLMS algorithm is to harm the convergence of the weight vector for de-emphasized frequencies while having little effect elsewhere.

In order to examine the rate of convergence for the filtered-E LMS algorithm, relative to the LMS algorithm, it is useful to examine the rate of convergence in specific frequency bands. This can be done by filtering the error signal by a bandpass filter, as illustrated in Figure 3.15. By appropriate selection of the filter $B(z)$, any frequency range of interest in the error $e(n)$ can be isolated to give the signal $e_b(n)$. The error signal of the LMS algorithm can be filtered in an analogous manner. Below, the rates of convergence for both the FELMS and the LMS algorithms are considered in various frequency ranges utilizing the method described here. Note that step sizes are again chosen so that the FELMS and LMS algorithms reach the same steady state MSE (fullband MSE that is).

The slowed convergence of the FELMS algorithm for de-emphasized frequencies is demonstrated in Figures 3.16 to 3.19 for the case of modelling speakerphone-room *mc3033* with a C-message frequency weighting filter. In Figures 3.16 and 3.18, the convergence of frequencies in the range of 0 Hz to 500 Hz is considered for the LMS and FELMS

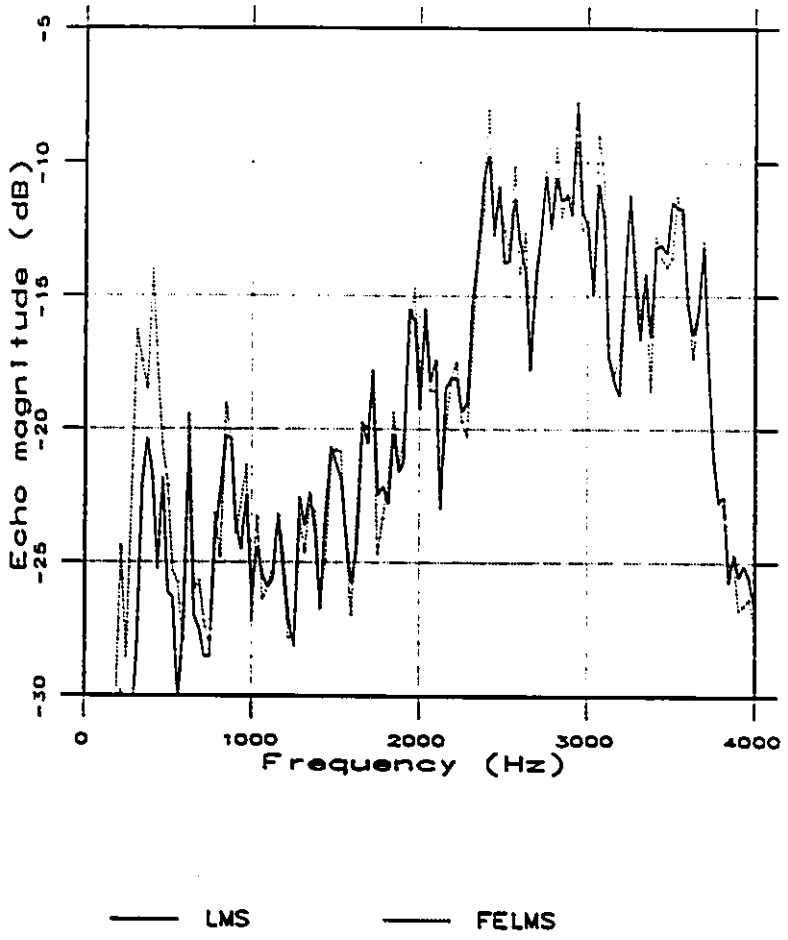

Figure 3.14: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 4.0 \cdot 10^{-4}$, $L = 300$ (both algorithms).

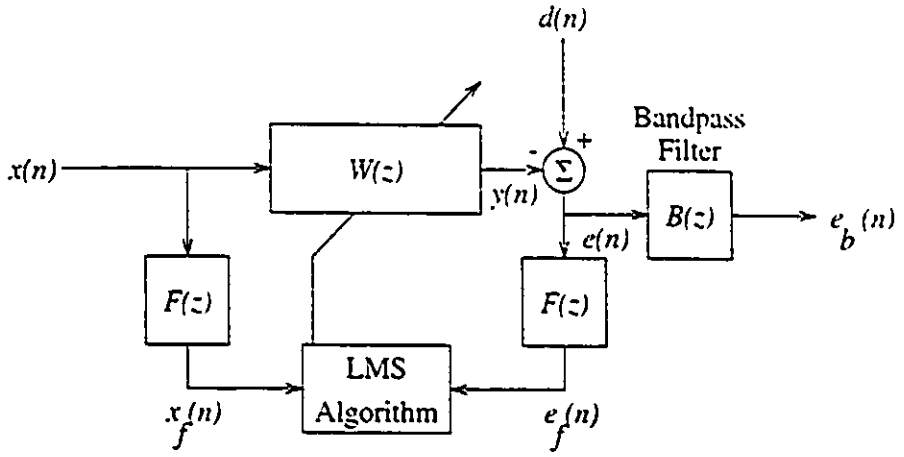

Figure 3.15: Filtered-E LMS structure to examine specific error frequencies.

algorithms, respectively. Note that this corresponds to a heavily attenuated region in the C-message frequency weighting curve. Figures 3.17 and 3.19 show the convergence of frequencies in the range of 1 kHz to 1.5 kHz. This region passes through the C-message approximate weighting filter $F^{(1)}(z)$ with a gain of slightly less than 1.5 (see Figure 3.9). Note from Figures 3.16 and 3.17 that the rate of convergence of the error for the LMS algorithm is identical for the two frequency ranges with the adaptive filter taking approximately 5000 iterations to converge. From Figures 3.18 and 3.19 it can be seen that this is not at all the case when the FELMS algorithm is used with a C-message approximate weighting filter. The convergence is much slower for frequencies in the 0 Hz to 500Hz range, corresponding to the de-emphasized frequencies. It can be seen from Figure 3.18 that the filtered-E LMS adaptive filter takes approximately 20000 iterations to converge (note the longer time scale used in Figure 3.18). The 1 - 1.5 kHz frequency region of the filtered-E LMS algorithm, on the other hand, has converged very quickly to its optimum solution, taking about 3000 iterations. This makes sense since it can be seen from Figure 3.9 that this region of the adaptive filter frequency response is amplified by approximately 1.5.

In Figure 3.13 it was seen that in the line echo cancellation application, the filtered-E LMS algorithm achieved a desirable frequency weighting of the error compared to the LMS algorithm. However, the convergence of the algorithm is still affected in the

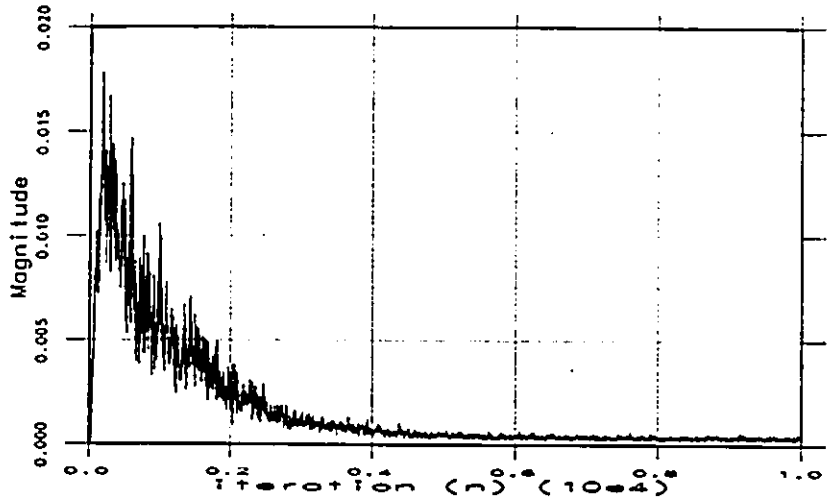

Figure 3.16: Convergence of the LMS adaptive filter for the frequency range 0 - 500 Hz. Unknown system: room *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

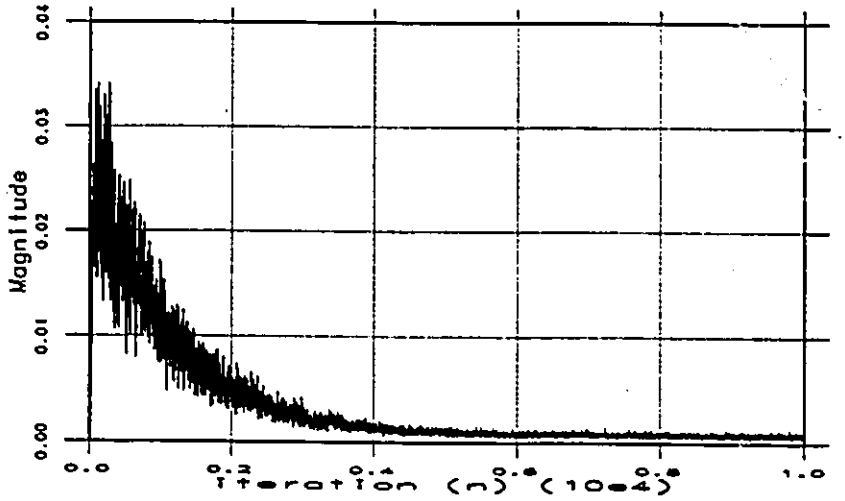

Figure 3.17: Convergence of the LMS adaptive filter for the frequency range 1 - 1.5 kHz. Unknown system: room *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

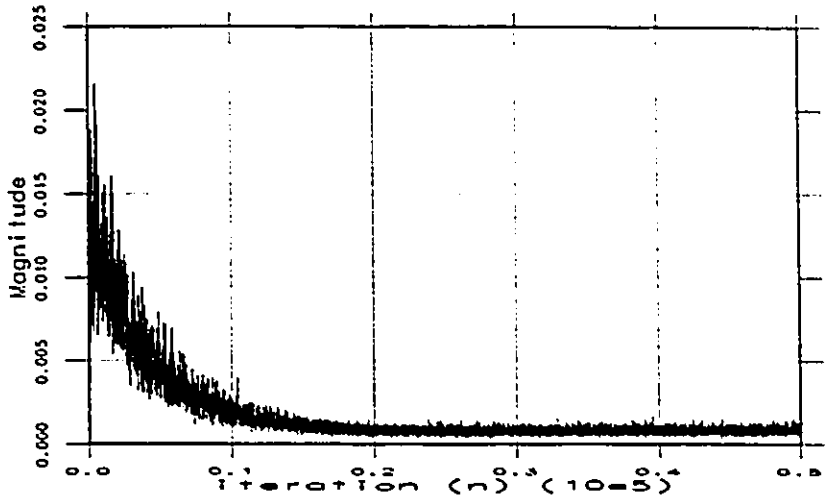

Figure 3.18: Convergence of the filtered-E LMS adaptive filter for the frequency range 0 - 500 Hz. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: room *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

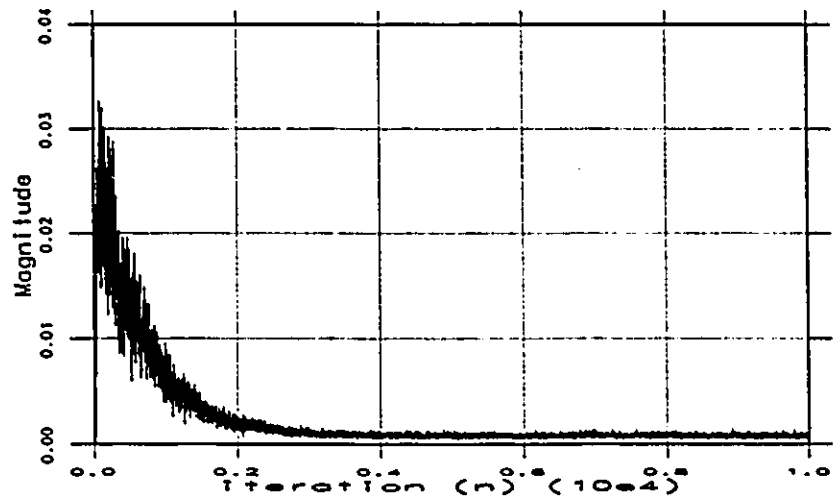

Figure 3.19: Convergence of the filtered-E LMS adaptive filter for the frequency range 1 - 1.5 kHz. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: room *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

same manner as the room modelling case described above. Figures 3.20 to 3.23 show convergence curves for the hybrid modelling case with the simple high pass weighting filter. Since a different frequency weighting filter is being used the convergence is considered in different frequency regions than considered above in the room modelling case. Figures 3.20 and 3.22 consider the convergence of the adaptive filters in the frequency range of 0 to 0.5 kHz and Figures 3.21 and 3.23 show the convergence of the weight vectors in the frequency range 3.5 to 4 kHz. Again for the LMS algorithm, the convergence is similar in the two frequency bands. Figures 3.20 and 3.21 show that both frequency ranges converge in approximately 400 iterations. For the filtered-E LMS, the convergence curve of Figure 3.23 shows that the filtered-E LMS algorithm converges as fast or perhaps slightly faster than the LMS algorithm in the high frequency region. This is not unexpected as the frequency response of the weighting filter $F^{(2)}(z)$ shows an amplification of approximately 1.5 in the passband. Figure 3.22 shows significantly slower convergence for the filtered-E LMS algorithm in the low frequency region with convergence taking approximately 30000 iterations. Figure 3.10 shows that this region is quite heavily attenuated; thus, the slow convergence is expected.

In terms of convergence, the bounds on the step size μ for the FELMS algorithm must also be discussed. Experimentally, it was found that the FELMS algorithm suffered from a tighter bound on allowable step sizes than the LMS algorithm. Table 3.1 outlines the experimentally determined maximum step sizes for the filtered-E LMS algorithms in the simulations above. Note that the bound on step size for the LMS algorithm is well known [13], and is given by

$$\mu_{LMS}^{max} = \frac{2}{LE(x^2(n))}. \tag{3.12}$$

In the case here the input is white noise of variance 1.0 so that $E(x^2(n)) = 1.0$. The theoretical filtered-E LMS step size bounds given in Table 3.1 were determined using Equation 3.9 from [27].

There are two fairly obvious factors which influence the step size bounds for the filtered-E LMS algorithm. These are: (1) the gain of the weighting filter $F(z)$ and (2)

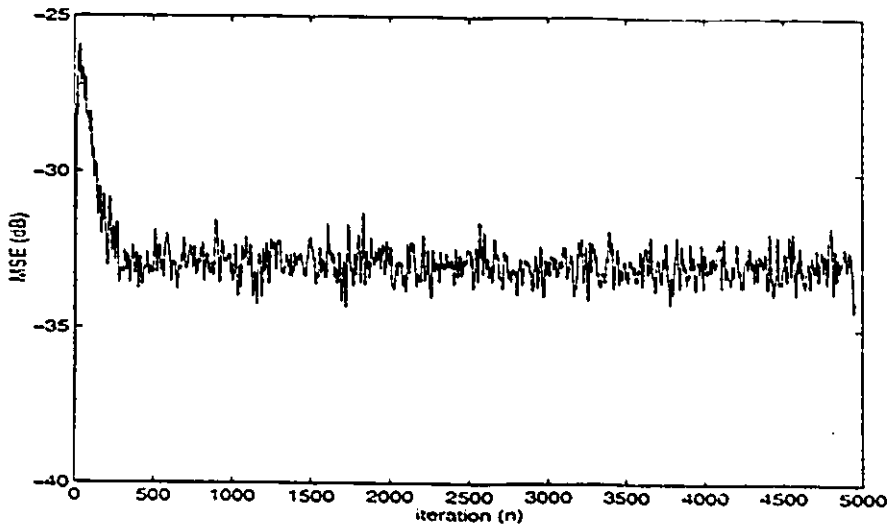

Figure 3.20: Convergence of the LMS adaptive filter for the frequency range 0 - 500 Hz. Unknown system: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-2}$, $L = 32$.

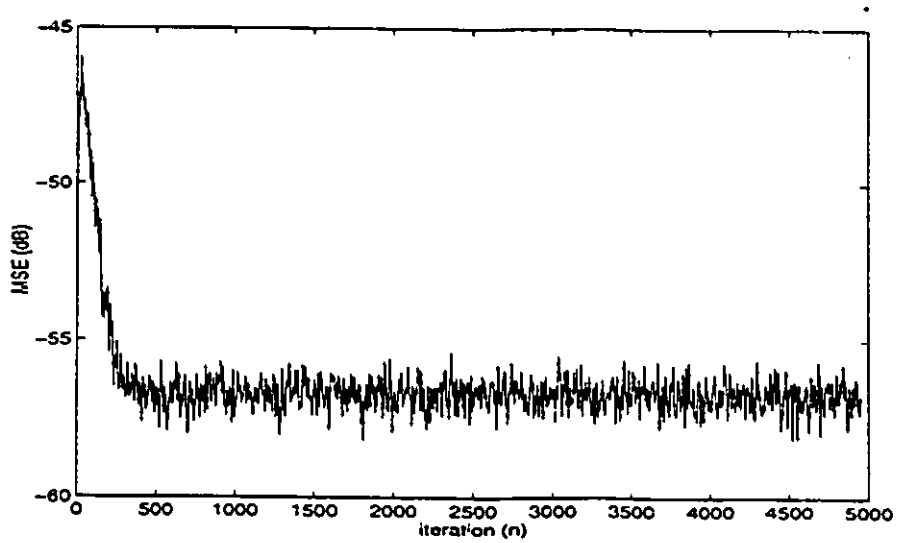

Figure 3.21: Convergence of the LMS adaptive filter for the frequency range 3.5 - 4 kHz. Unknown system: room *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-2}$, $L = 32$.

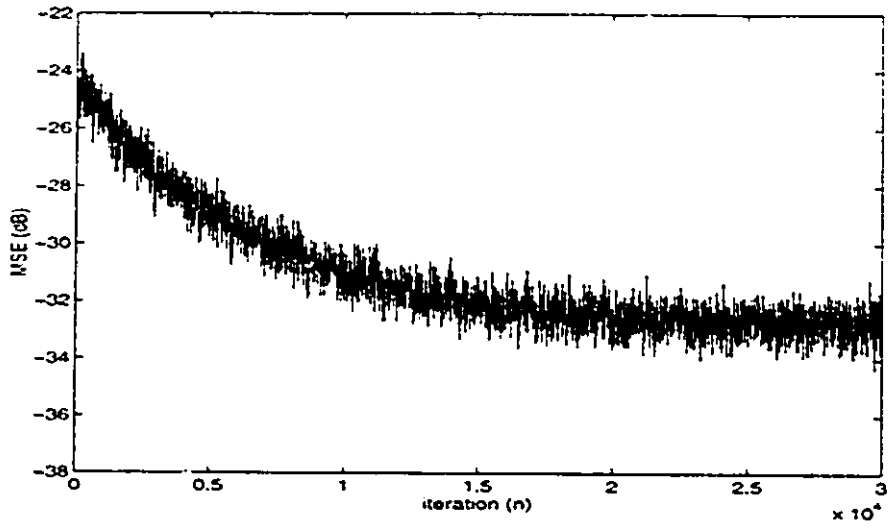

Figure 3.22: Convergence of the filtered-E LMS adaptive filter for the frequency range 0 - 500 Hz. Unknown system: *hybrid1*. Input: white noise of variance 1.0. $\mu = 7.0 \cdot 10^{-3}$, $L = 32$.

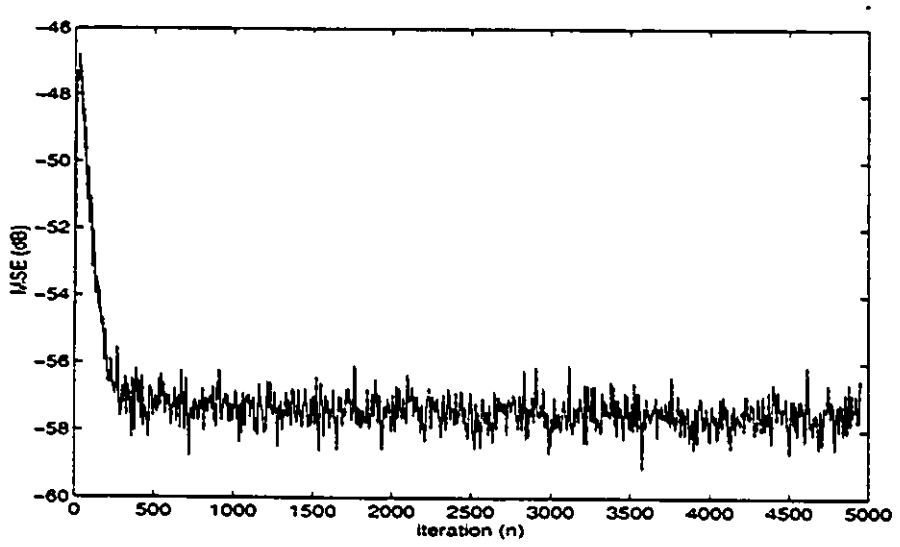

Figure 3.23: Convergence of the filtered-E LMS adaptive filter for the frequency range 3.5 - 4 kHz. Unknown system: *hybrid1*. Input: white noise of variance 1.0. $\mu = 7.0 \cdot 10^{-3}$, $L = 32$.

Unknown system	$F(z)$	L	μ_{LMS}^{max}	μ_{FELMS}^{max}	$\mu_{experimental}^{max}$
hybrid1	$F^{(2)}(z)$	32	$6.25 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$
hybrid1	$F^{(1)}(z)$	32	$6.25 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$
WBN13	$F^{(2)}(z)$	300	$6.7 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$4.0 \cdot 10^{-3}$
WBN13	$F^{(1)}(z)$	300	$6.7 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$
mc3033	$F^{(2)}(z)$	500	$6.7 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$
mc3033	$F^{(1)}(z)$	500	$6.7 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$

Table 3.1: Experimental step size bounds for the filtered-E LMS algorithm.

the delay of the filter $F(z)$. The effect of the delay can be explained by referring to the analysis of [16] where the delayed LMS (DLMS) algorithm is analyzed. The DLMS algorithm is identical to the LMS algorithm save that weight vector updates are delayed. The weight vector update for the DLMS and normalized DLMS (NDLMS) algorithms can be expressed as

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu e(n-\Delta)\mathbf{X}(n-\Delta) \quad (3.13)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu}{\|\mathbf{X}(n-\Delta)\| + \sigma} e(n-\Delta)\mathbf{X}(n-\Delta), \quad (3.14)$$

respectively. The DLMS algorithm is really a special case of the filtered-E LMS algorithm where the weighting filter $F(z)$ is a simple delay, ie. $F(z) = z^{-\Delta}$. The analysis in [16] shows that the effect of the delay factor Δ is to limit the range of step sizes μ for which the algorithm is stable. In [16] the bound on step size for a given delay of Δ can be found by determining the roots of a polynomial of order $\Delta + 1$. No exact step size bounds are calculated for the filtered-E LMS algorithm used here but the results presented in [16] serve to back up the observations of Table 3.1.

The argument presented above; namely, that the DLMS algorithm is really a special case of the filtered-E LMS algorithm, helps to provide some insight into the effect of the length of $F(z)$ on the step size bounds. As mentioned earlier, a detailed analysis of the filtered-X LMS algorithm is presented in [27]. The step size bound calculated in [27] was given earlier as Equation 3.9. This equation can be evaluated for the cases of

interest here since white noise training signals were used throughout. For comparison, the step size bound for the LMS algorithm is $6.7 \cdot 10^{-3}$ for the room modelling case and $6.25 \cdot 10^{-2}$ for the hybrid modelling case. To evaluate Equation 3.9. the fourth order moment ($m_x^{(2,2)}$) is required. For a white noise process, this moment is given as $m_x^{(2,2)} = 3\sigma^4$ where σ^2 is the variance. Since the variance is one for the simulations here, the fourth order moment is then three. The weighting filters in this thesis were both selected to satisfy $\sum_{i=0}^{N-1} f_i^2 = 1.0$, where f_i are the coefficients of the filters $F(z)$. Thus, the term involving h_i in Equation 3.9 is one. The term $tr(\Lambda)$ will be equal to L and the eigenvalues of the autocorrelation matrix of a white noise process are all equal to one (if the variance is one). Evaluating all of these terms gives the following bounds on step size for the FELMS algorithm: $2.2 \cdot 10^{-3}$ for the room modelling case and $1.9 \cdot 10^{-2}$ for the hybrid modelling case. The maximum step sizes for the filtered-E LMS algorithm, as determined experimentally and given in Table 3.1, turn out to be slightly greater than the values predicted by Equation 3.9, but the values predicted by Equation 3.9 are convincing in terms of the expectation that the filtered-E LMS algorithm will suffer from tighter bounds on step size than the LMS algorithm, given the same adaptive filtering parameters. The simulations of [28] also show slower convergence for the filtered-E LMS algorithm than for the filtered-X LMS algorithm.

From Table 3.1 it can be seen that the C-message weighting filter restricts the step size slightly more than the simple two tap high pass filter. This is due to the longer length of the C-message filter. As for the gain of the frequency weighting filter note that both the C-message weighting filter and the high pass weighting filter have a maximum gain of approximately 1.4. So, effectively, for frequencies in the passband of these filters the step size is increased by the factor of 1.4. Between this and the effect of the weighting filter delay (note that both of the weighting filters used here are linear phase), it is natural for the filtered-E LMS algorithm to suffer from a tighter bound on step sizes.

3.6 Level of Undermodelling

Earlier, response *hybrid1* was modelled with an adaptive filter of length 32 taps. The plant response was 100 taps long. Thus, the adaptive filter was undermodelling the plant response by 66 taps. The level of undermodelling is dependent on the number of unmodelled taps as well as the energy content of those taps. Thus, two plant responses undermodelled by the same number of taps may not have the same level of undermodelling.

The filtered-E LMS algorithm was able to achieve significant frequency weighting relative to the LMS algorithm when undermodelling plant response *hybrid1* with 32 taps. The ability of the filtered-E LMS algorithm to effect frequency weighting on the final echo relative to the LMS algorithm is dependent on undermodelling. In fact, undermodelling the plant response is a necessary though not sufficient condition to achieve frequency weighting of the echo spectrum via the FELMS algorithm. This section will demonstrate the dependence of the weighting capabilities of the FELMS algorithm on the level of undermodelling using simulations with response *hybrid1*.

Figure 3.13 shows the FELMS and LMS echo spectra when *hybrid1* was modelled using 32 tap adaptive filters. Note the significant frequency weighting obtained by the FELMS algorithm. Figures 3.24 and 3.25 show FELMS and LMS echo spectra when *hybrid1* is modelled using adaptive filters of lengths 60 and 80, respectively. It is obvious from these three figures that the frequency weighting achieved by FELMS relative to LMS decreases as the level of undermodelling decreases. In Figure 3.25 the FELMS algorithm achieves very little weighting relative to the LMS algorithm.

In Figures 3.11, 3.12 and 3.14 the FELMS algorithm achieved virtually no weighting of the echo spectrum relative to the LMS algorithm. Still the speakerphone-room responses were undermodelled significantly and thus, as stated above, undermodelling the plant response is a necessary but not sufficient condition to achieve frequency weighting via the FELMS algorithm. In fact, the performance of the FELMS algorithm is heavily dependent on the nature of the plant response being modelled. This dependency on the

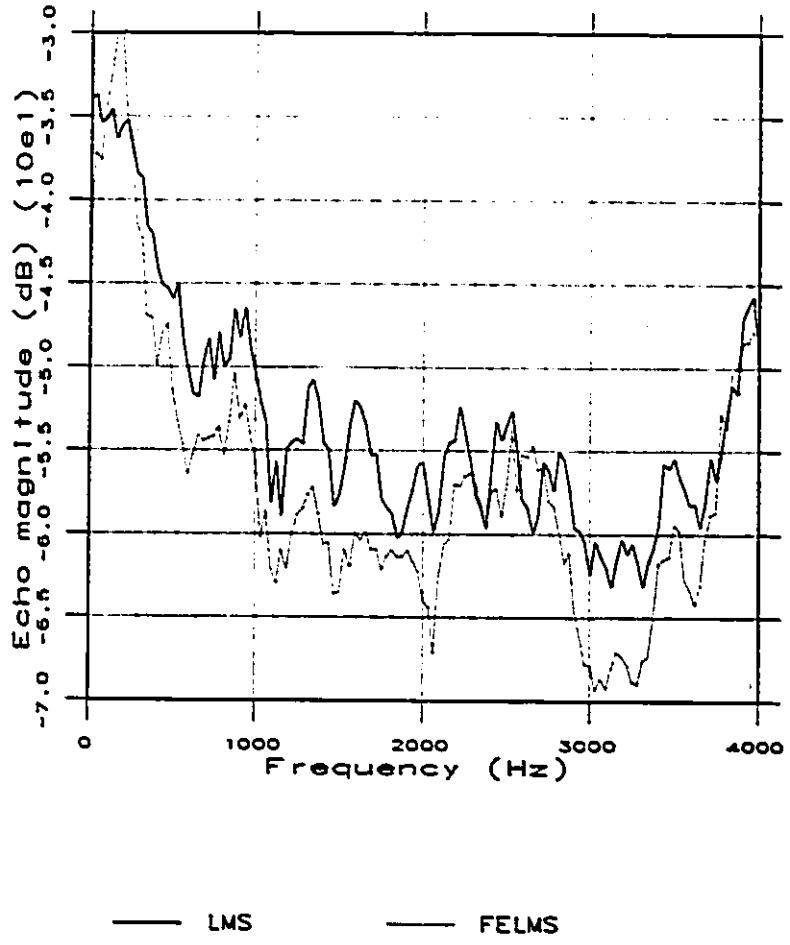

Figure 3.24: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 8.0 \cdot 10^{-3}$ (LMS) and $\mu = 2.5 \cdot 10^{-3}$ (FELMS), $L = 60$ (both algorithms).

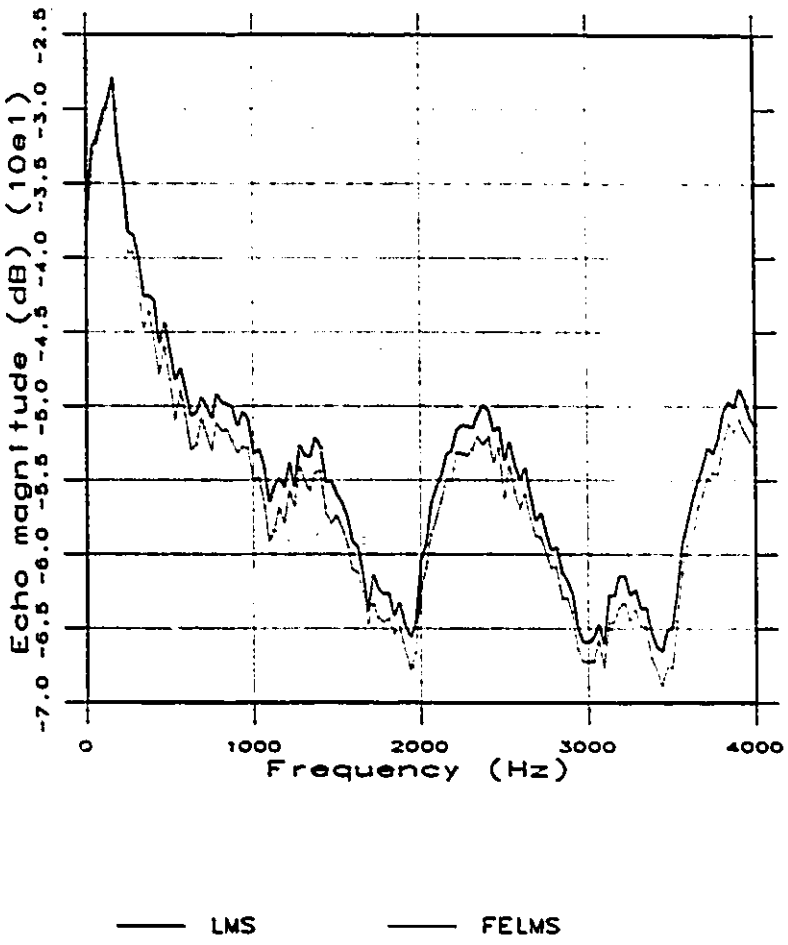

Figure 3.25: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 3.0 \cdot 10^{-3}$ (LMS) and $\mu = 1.0 \cdot 10^{-3}$ (FELMS), $L = 80$ (both algorithms).

nature of the plant response will be investigated in the following sections.

3.7 Nature of Plant Response

It was seen above that the frequency weighting capabilities of the FELMS depend on the plant response to be modelled. FELMS performed well when modelling response *hybrid1* but not when modelling the speakerphone-room responses. The frequency response of *hybrid1* (Figure A.8) is very smooth. It does not contain any steep transition regions or any random characteristics (as the spectrum of a white noise signal would). On the other hand, the frequency responses of the two speakerphone-room responses (Figures A.2 and A.5) are quite random and these frequency responses are nearly always in transition, are never smooth, and resemble some sort of coloured noise sequence spectrum. Results in this section are reproduced based on the work of Mayyas and Aboulnasr [80]. To illustrate the effect of the nature of the plant frequency response on FELMS performance relative to LMS performance, simulations with a lowpass filter as plant response are considered. A lowpass filter has both a smooth, non-varying frequency response (in its passband) and a response with a sharp transition region (around the frequency cut-off).

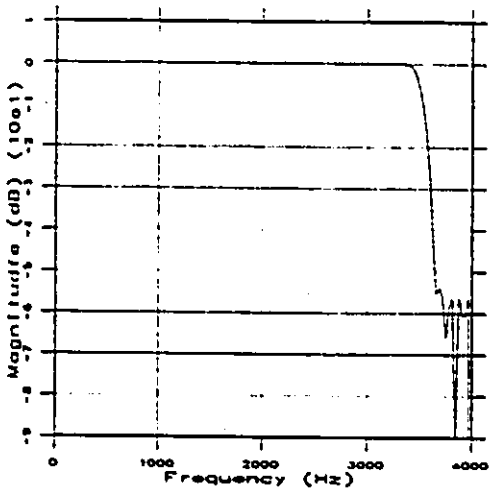

Figure 3.26: Frequency response of 100 tap lowpass filter with cut-off at 3500 Hz.

Figure 3.26 shows the frequency response of the lowpass filter to be modelled. This filter is a 100 tap, linear-phase filter designed using the `fir1` program in `Matlab`. The adaptive filters are given 66 taps so that there is significant undermodelling and the FELMS algorithm will have a chance to weight the output. The FELMS and LMS algorithms are used with equal step sizes so that the final errors of the FELMS and LMS will

not necessarily be the same. Both algorithms were allowed to run for 20000 iterations with the spectra being computed over the last 3000 iterations using 256 pt. DFT's averaged over those last 3000 iterations (a program implemented with the Khoros simulation system).

Averaged echo spectra for the LMS and FELMS (with three different weighting filters) are shown in Figures 3.27 to 3.29. In Figure 3.27, weighting filter $F^{(1)}(z)$ is used with the FELMS algorithm, resulting in some reduction of the echo for frequencies emphasized by $F^{(1)}(z)$. Note that weighting filter $F^{(1)}(z)$ (in Figure 3.9) attenuates frequencies above 3 kHz which includes the transition band of the lowpass filter being modelled. Figure 3.28 shows echo spectra for the LMS and FELMS algorithms with weighting filter $F^{(2)}(z)$. Here the FELMS algorithm does not improve the echo cancellation at any frequency relative to LMS. The weighting filter $F^{(2)}(z)$ does not attenuate the transition region of the lowpass filter at all. Finally Figure 3.29 shows the echo spectra for the LMS and the FELMS algorithms with weighting filter $F^{(3)}(z)$. Weighting filter $F^{(3)}(z)$ is given by:

$$F^{(3)}(z) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot z^{-1}. \qquad (3.15)$$

This weighting filter is the "mirror image" of filter $F^{(2)}(z)$ and its frequency response can be easily discerned from the response of $F^{(2)}(z)$ in Figure 3.10. In Figure 3.29, the FELMS once again achieves better echo cancellation than the LMS for emphasized frequencies. Note that, like Figure 3.27, the weighting filter attenuates the transition band of the plant response (near 3500 Hz).

Figures 3.27 to 3.29 show that the ability of the FELMS to reduce the echo relative to the LMS for emphasized frequencies depends upon what part of the frequency response is emphasized and what part is de-emphasized. In particular, de-emphasizing a frequency region with a steep transition provides for better echo cancellation elsewhere whereas emphasizing the transition region does not provide any better cancellation for the FELMS than the LMS. To relate these results to those with *hybrid1*, *WBN13* and *mc3033*, note that response *hybrid1* has a fairly steep transition at low frequencies and then falls off quite smoothly and fairly slowly. Since the weighting filter $F^{(2)}(z)$ attenu-

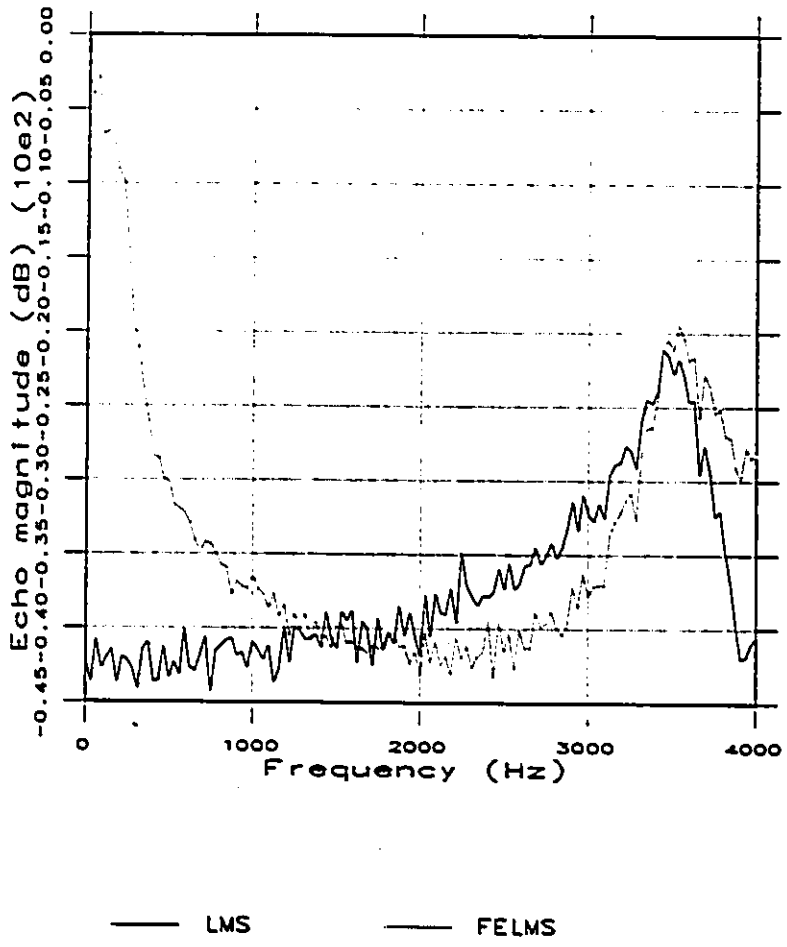

Figure 3.27: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

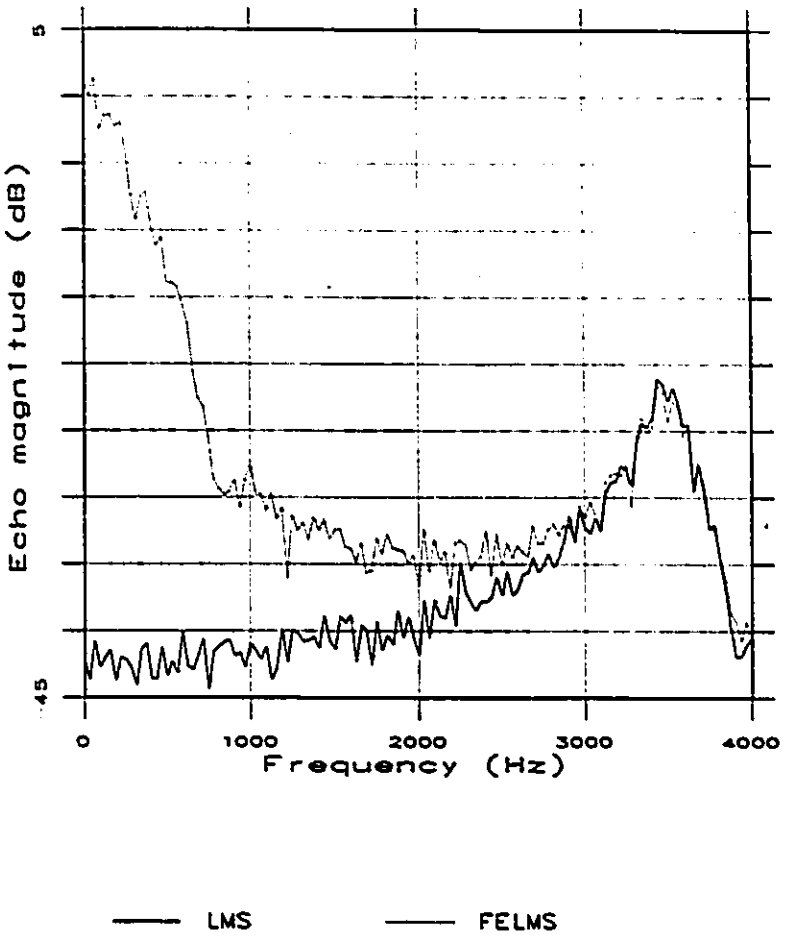

Figure 3.28: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

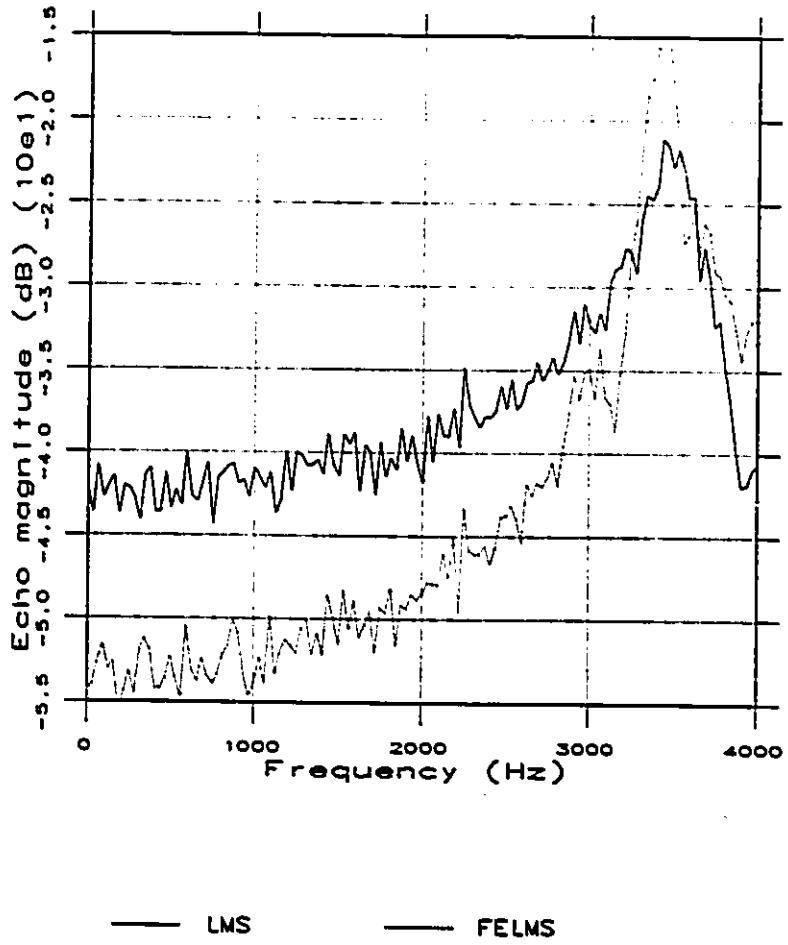

Figure 3.29: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(3)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

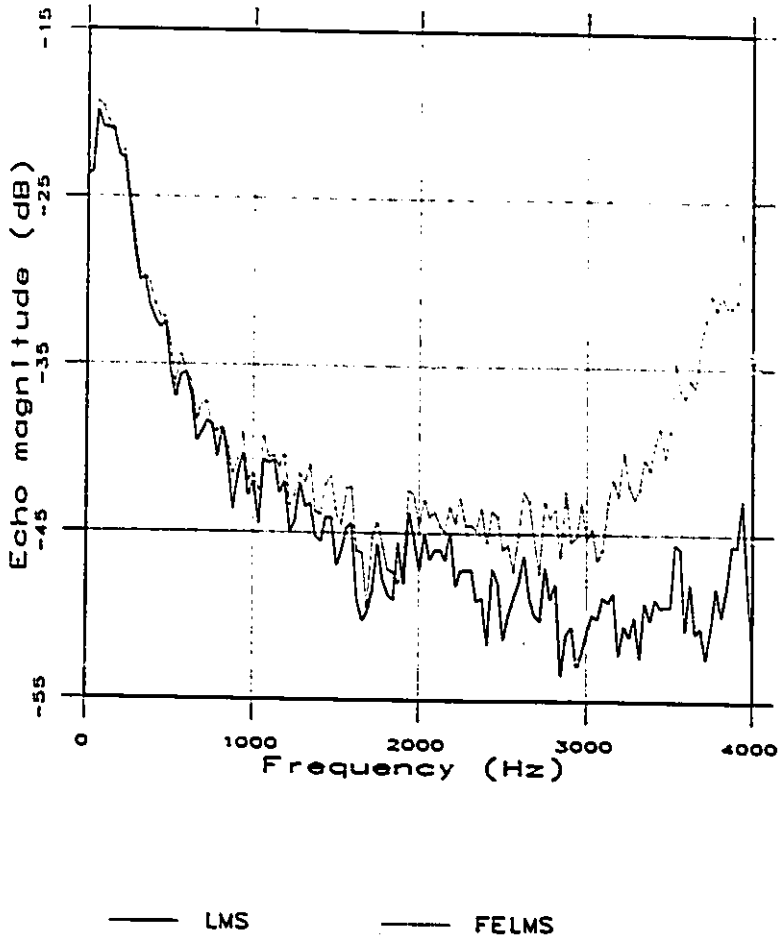

Figure 3.30: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(3)}(z)$. Unknown system: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 8.0 \cdot 10^{-3}$, $L = 32$ (both algorithms).

ates the transition region, better echo cancellation at higher frequencies is obtained by FELMS over LMS. If one were to try the opposite weighting ($F^{(3)}(z)$). FELMS would be unable to achieve better echo cancellation at emphasized frequencies than LMS. This is illustrated in Figure 3.30 where weighting filter $F^{(3)}(z)$ is used with *hybrid1*. Here, weighting filter $F^{(3)}(z)$ tries to filter out the smooth part of the response in order to provide better cancellation in the transition region but no better cancellation is achieved. Responses *WBN13* and *mc3033* both exhibit responses which are never smooth. There are no smooth regions of the response where better cancellation could be obtained by emphasizing that region.

To illustrate the effect of a plant response which is not smooth, FELMS and LMS are used to model a "white noise" plant response, shown in Figure 3.31. The white noise process of Figure 3.31 is 100 taps long, with a variance of 1.0, and will be modelled using adaptive filters 66 taps long. Figures 3.32 to 3.34 show echo spectra for FELMS and LMS where weighting filters $F^{(1)}(z)$, $F^{(2)}(z)$ and $F^{(3)}(z)$ are used with the FELMS, respectively. Note that the FELMS achieves no better echo cancellation for any frequency region than LMS for any of the weighting filters. The white noise plant response of course contains no smooth frequency regions.

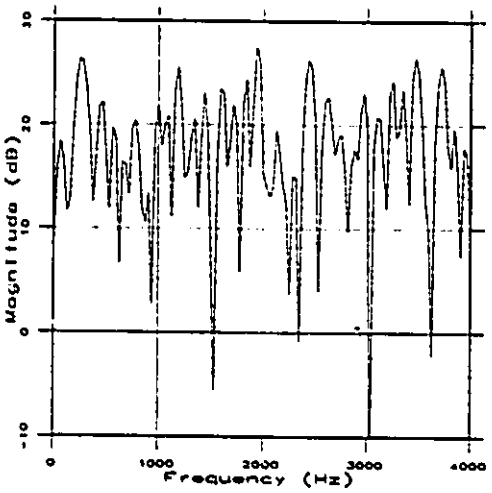

Figure 3.31: Frequency response of white noise plant.

Having discussed the relationship between FELMS performance and plant response some comment can now be made about the results in [28]. Although it is claimed in [28] that the FELMS can achieve better noise cancellation at emphasized frequencies than FXLMS, results are only given for the case where the plant response is a simple lowpass filter with a cut-off frequency of 950 Hz. The sampling rate used in [28] is 2 kHz so that the lowpass filter has a transition band near half the sampling rate, just as the

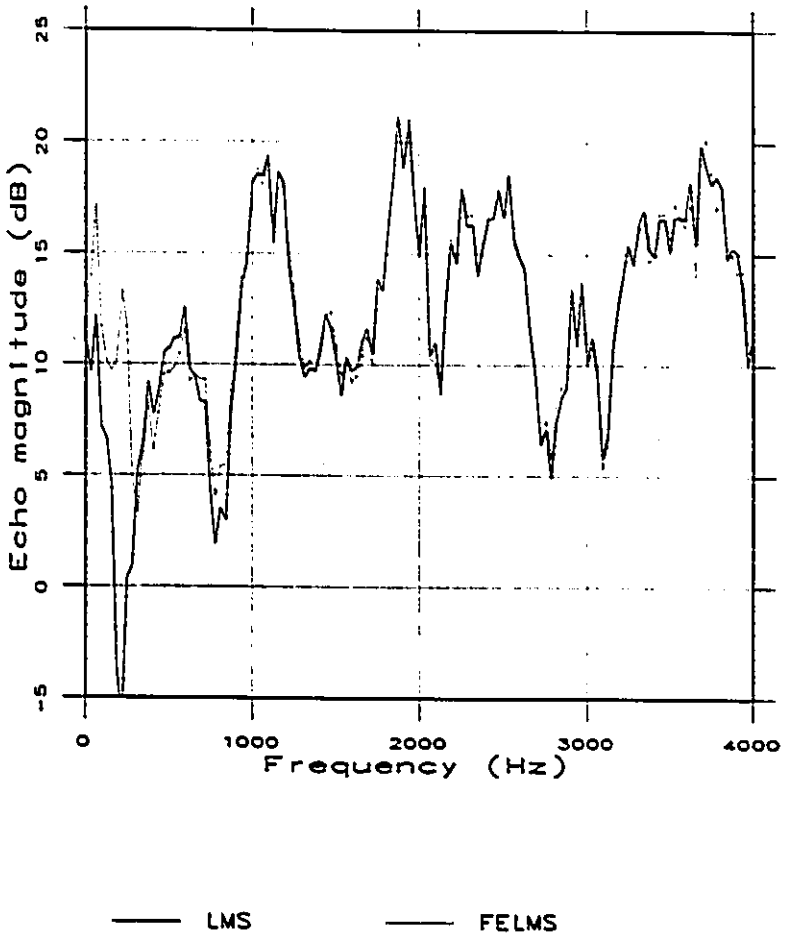

Figure 3.32: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: white noise plant. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

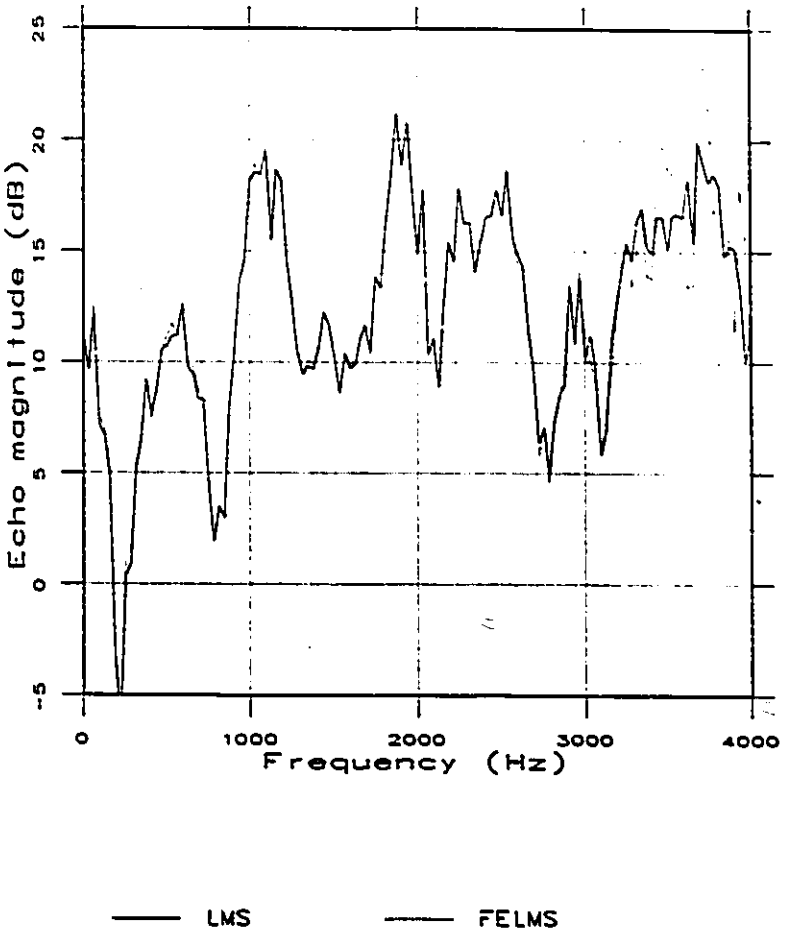

Figure 3.33: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: white noise plant. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

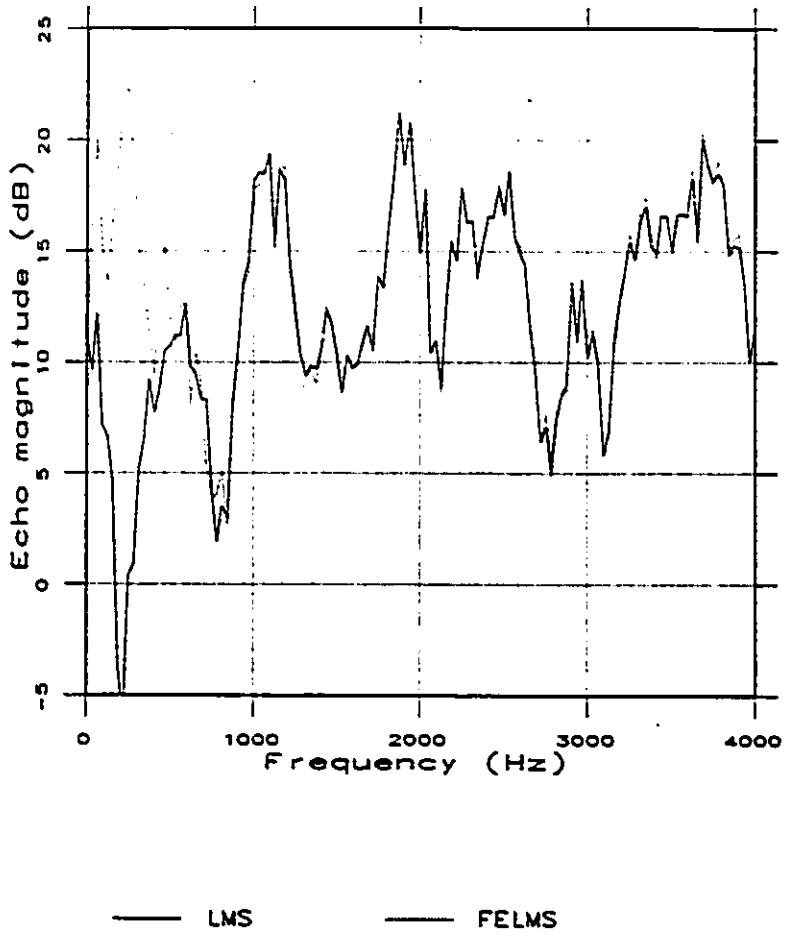

Figure 3.34: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(3)}(z)$. Unknown system: white noise plant. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

3500 Hz filter used in the simulations above is. The simulations above showed that the FELMS could only achieve better performance when the weighting filter de-emphasized the transition region. This is exactly what is done in [28]. The two weighting filters used in [28] both attenuate the transition region of the lowpass filter being modelled, resulting in improved noise cancellation at emphasized frequencies elsewhere. Although one of the early paragraphs in [28] claims that results for a real acoustic noise cancellation case will be presented, this promise is not fulfilled.

Figures 3.35 and 3.36 show the frequency responses of the FELMS and LMS weight vectors (after 20000 iterations) and the lowpass plant response of Figure 3.26 for frequency regions 0 to 3000 Hz and 3000 to 4000 Hz, respectively. The FELMS weight vector shown corresponds to the simulation with weighting filter $F^{(3)}(z)$ (lowpass weighting). By using a weighting filter which attenuates the transition band, the FELMS weight vector no longer needs to fit as closely to the difficult part of the frequency response curve (the transition band). Thus, since the rest of the frequency response curve of the lowpass filter is easy to model, the FELMS algorithm can fit the frequency response better there. The LMS algorithm spends more effort fitting the curve near the transition band and the fit of the LMS curve to the plant response elsewhere suffers. Consider the situation when weighting filter $F^{(2)}(z)$ is used. The plant frequency response remains just as hard to model as originally, since the steep transition (hard to fit a curve to) is still there. Figures 3.37 and 3.38 illustrate this situation. Note here that performance has worsened at de-emphasized frequencies but essentially the FELMS can still not fit the curve of the plant response in the transition region any better than the LMS algorithm.

To further show that FELMS performance is dependent on the plant response being modelled, the *hybrid1* response is altered by removing its unmodelled tail (taps 33 to 100) and replacing these with taps 1133 to 1200 of speakerphone-room response *mc3033*. The new response is shown in Figure 3.39 with corresponding frequency response in Figure 3.40. Error spectra for LMS and FELMS are shown in Figure 3.41. Now, the FELMS does not achieve better echo cancellation at the weighted frequencies.

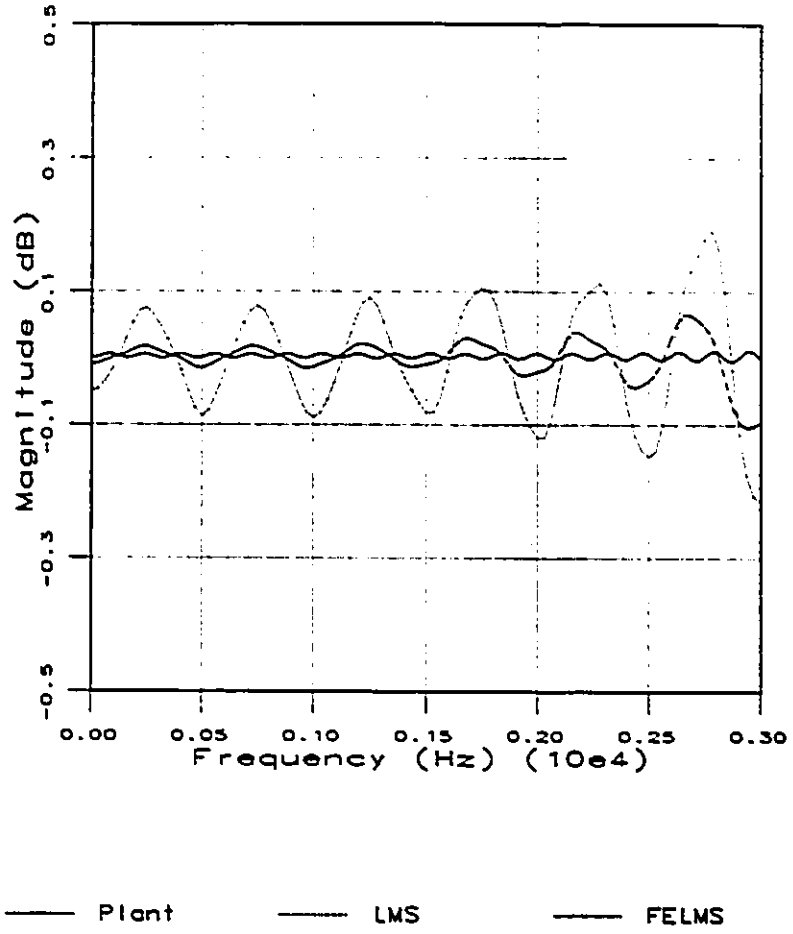

Figure 3.35: Frequency responses of lowpass plant,LMS and FELMS weight vectors from 0-3000 Hz. FELMS weighting filter: $F^{(3)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

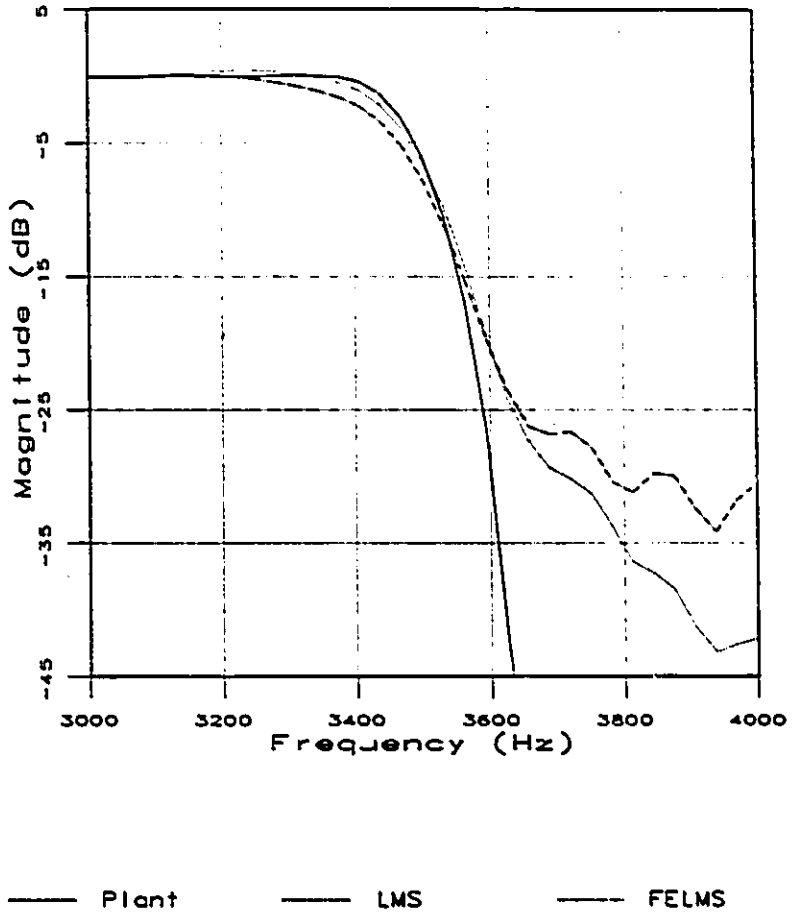

Figure 3.36: Frequency responses of lowpass plant,LMS and FELMS weight vectors from 3000-4000 Hz. FELMS weighting filter: $F^{(3)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

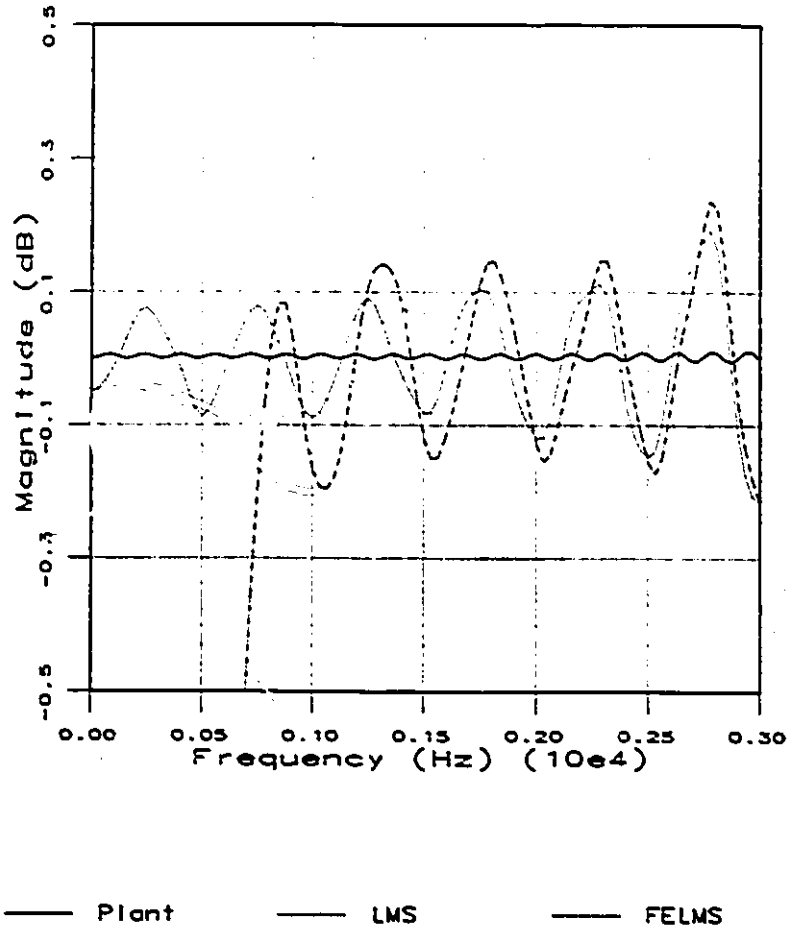

Figure 3.37: Frequency responses of lowpass plant,LMS and FELMS weight vectors from 0-3000 Hz. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

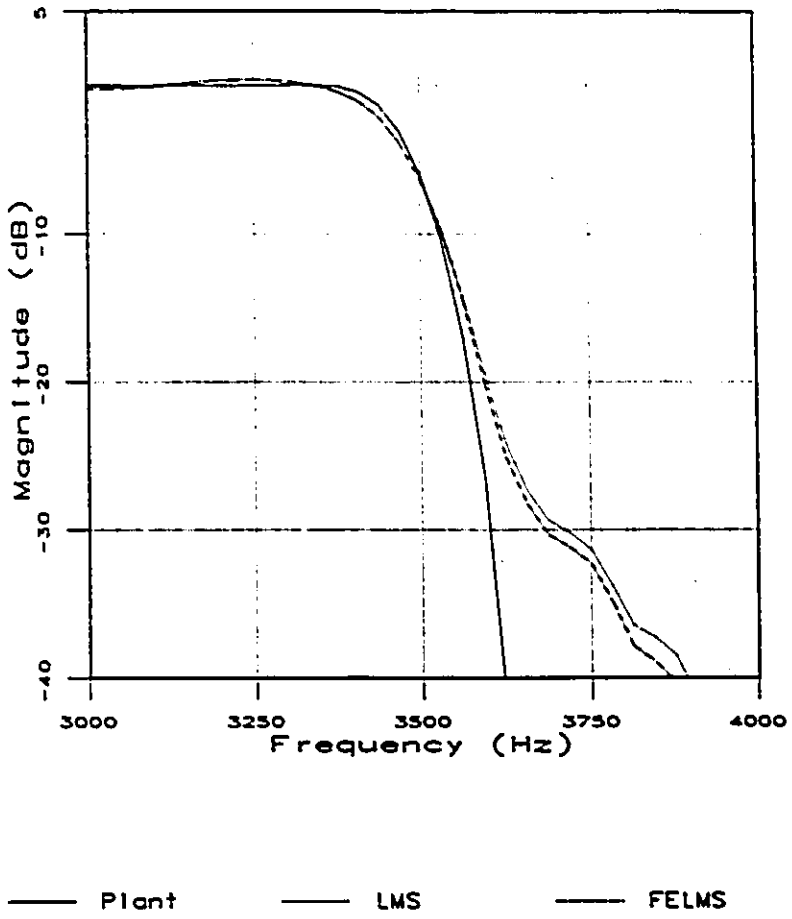

Figure 3.38: Frequency responses of lowpass plant,LMS and FELMS weight vectors from 3000-4000 Hz. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: 3500 Hz lowpass filter. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-3}$, $L = 66$ (both algorithms).

Figure 3.40 shows that although the new frequency response retains a similar shape to the orginal hybrid of Figure A.8. it is not nearly as smooth and has lost some of its dominant low frequency content. Thus, there is more information at all frequencies of the plant response and the response is no longer smooth.

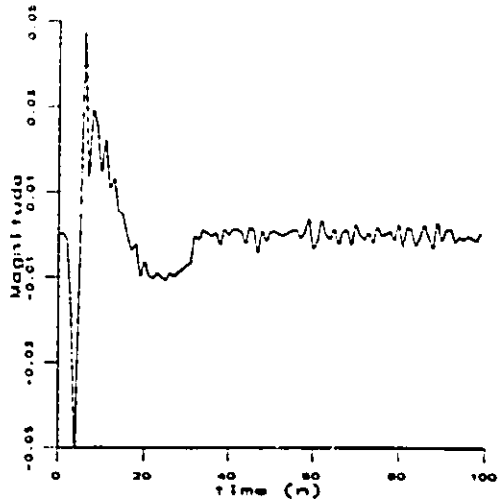

Figure 3.39: Time response of altered *hybrid1* response.

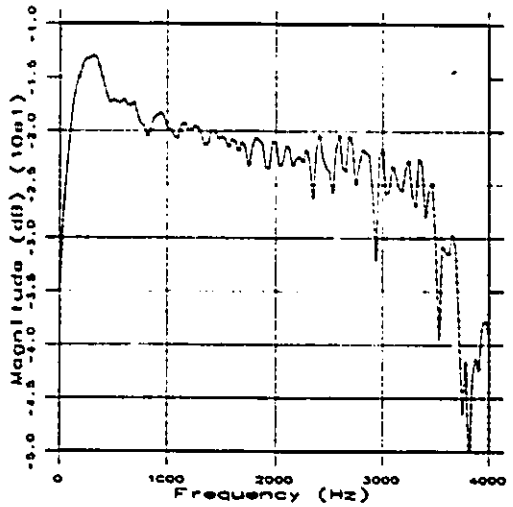

Figure 3.40: Frequency response of altered *hybrid1* response.

Next, speakerphone-room response *mc3033* was altered by truncating taps 301 to 1200 and then adding the tail of *hybrid1* (taps 33 to 100) to it to form a new 368 tap response. Figures 3.42 and 3.43 show the time domain response and frequency response of the new plant response, respectively. Now, with the new tail, the FELMS algorithm can achieve better echo cancellation at emphasized frequencies than the LMS algorithm as shown in Figure 3.44. In this case, Figure 3.43 does not show a smooth part. However, the tail response tacked on is indeed smooth and just does not have great enough amplitude to be seen over the frequency components of *mc3033* which fall in the first 300 taps. Note here that the gains achieved by the FELMS over the LMS are smaller than they were when the original *hybrid1* was modelled. This is quite understandable as now the undermodelling is at a significantly lower level. With *hybrid1*, the unmodelled taps rep-

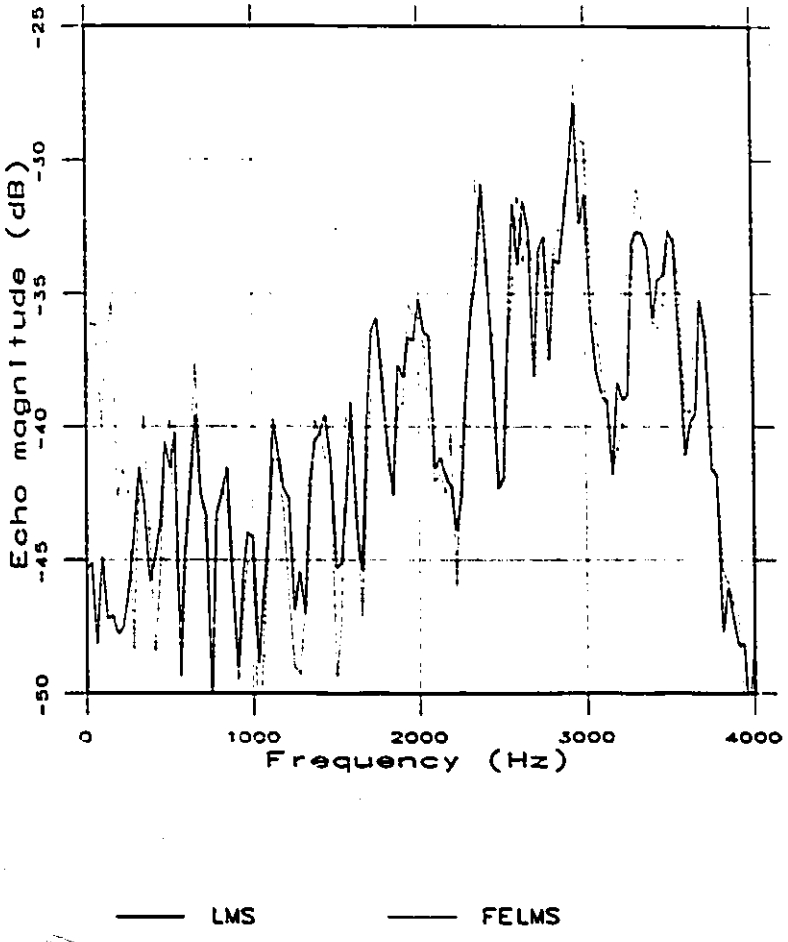

Figure 3.41: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(2)}(z)$. Unknown system: altered *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-2}$ (LMS) and $\mu = 7.0 \cdot 10^{-3}$ (FELMS), $L = 32$ (both algorithms).

resented about 68 % of the total length of the plant response whereas here they only represent about 19 % of the total length of the plant response.

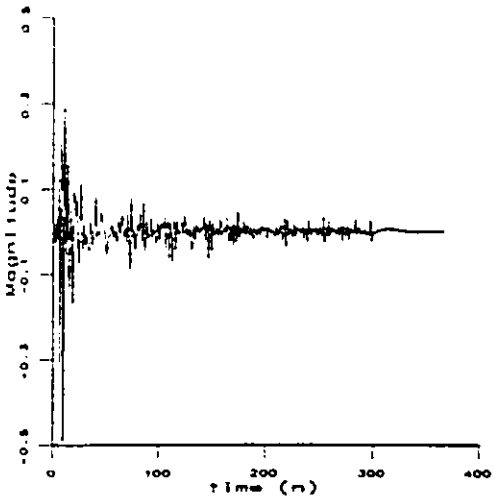

Figure 3.42: Time response of altered *mc3033* response.

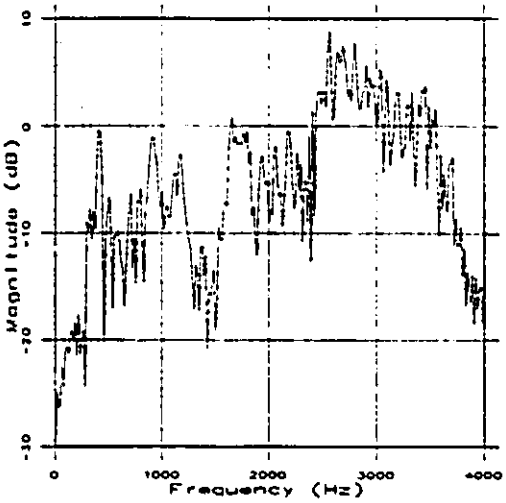

Figure 3.43: Frequency response of altered *mc3033* response.

3.8 Filtered-E LMS Weight Vector Misadjustment

The dependency of FELMS performance on the combination of the nature of the plant response and the weighting filter can be examined from another perspective. In this section, the weight vector misadjustment of both the LMS and FELMS algorithms will be examined. This will provide some insight into the operation of the filtered-E LMS algorithm and help explain why it works as it does in the line echo cancellation simulations and not in the acoustic echo cancellation simulations.

This section will show that the weighting filter $F(z)$ of the filtered-E LMS algorithm has a significant effect on weight vector misadjustment. This is one reason that the filtered-E LMS algorithm was able to shape the error spectrum in the desired manner for the hybrid modelling case of Figure 3.13. The misadjustment of the weight vector is influenced by the nature of the unknown system and *hybrid1* has a very low frequency

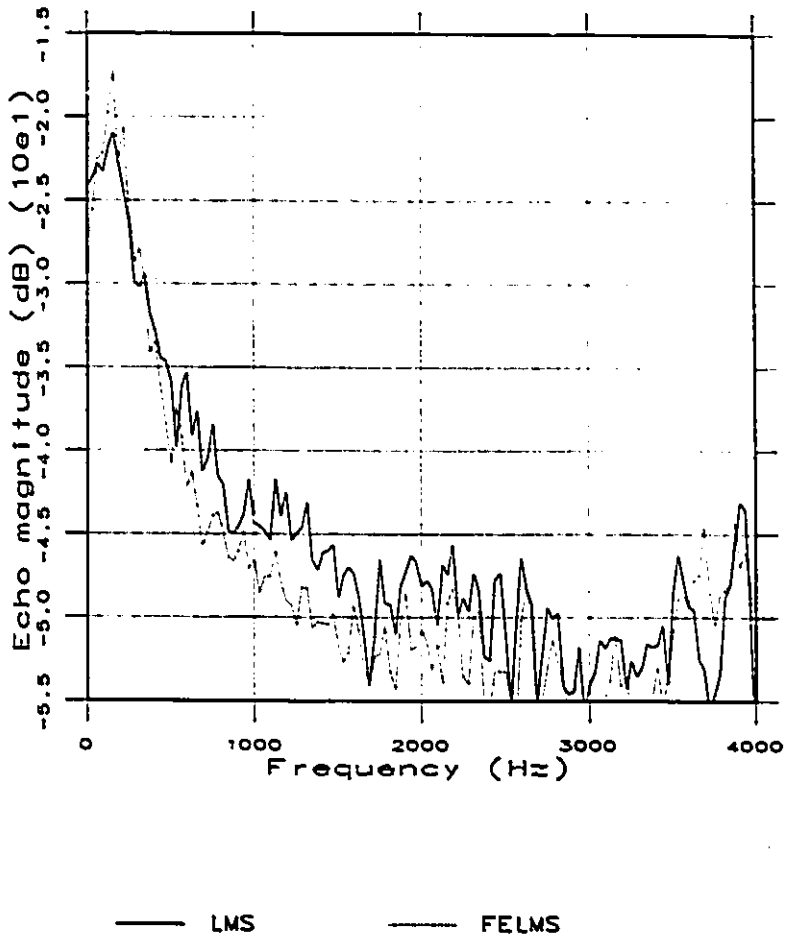

Figure 3.44: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: altered *mc3033*. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$ (both algorithms).

tail, as seen in Figure A.7. The filtered-E LMS algorithm, using the high pass weighting filter $F^{(2)}(z)$, was able to filter out most of the misadjustment, resulting in lower error at most frequencies. This effect did not occur for the room response modelling cases since the frequency responses of rooms *WBN13* and *mc3033* do not have any predominant frequency components in their tails.

In general, the length of the weight vector L chosen to model some unknown system will not be as long as the actual length of the unknown system. The nature of the acoustic echo cancellation problem was already described in Chapter 2. Room responses are indeed very long, as depicted in Figures A.1 and A.4 of Appendix A. For a typical commercial acoustic echo canceller, an adaptive filter of perhaps 300 or 400 taps might be used to model the speakerphone-room response. Most of the power in the speakerphone-room response will be contained in the initial reverberation allowing the adaptive filter to provide significant cancellation, yet the tail of the room response dies out slowly and the room response is effectively being severely undermodelled. In the case of line echo cancellation one can again expect to undermodel the hybrid response significantly. In the hybrid modelling simulations presented thus far, a 32 tap adaptive filter has been used to model the hybrid response (which is truncated at 100 taps).

Consider the error in estimation caused by the samples of the unknown system which cannot be modelled by the FIR weight vector. Little is said of the nature of this error other than that the misadjustment of the weight vector is inversely proportional to the weight vector length [13]. This means that the more some system is undermodelled, the more the weight vector tends to oscillate about its optimum solution.

In order to examine the misadjustment of the weight vector for the filtered-E LMS algorithm (and the LMS algorithm), a measure of misadjustment is required. To do this, one can use a measure of the distance (in a geometric sense) of the current weight vector $\mathbf{W}(n)$ from the optimum weight vector $\mathbf{W}^{opt}$. This measure is known as the weight vector misadjustment ($mis(n)$) and is a function of the sampling instant n. It is calculated according to the following equation:

$$mis(n) = (\|\mathbf{W}^{opt} - \mathbf{W}(n)\|)^{\frac{1}{2}} \tag{3.16}$$

where the coefficients of $\mathbf{W}^{opt}$,

$$\mathbf{W}^{opt} = [\ w_0^{opt} \quad w_1^{opt} \quad \cdots \quad w_{L-1}^{opt}\ ]^T. \tag{3.17}$$

are determined as follows:

$$w_i^{opt} = h_d(i). \qquad \text{for} \quad i = 0, 1, \cdots, L-1. \tag{3.18}$$

Note that $h_d(i)$ are the coefficients of the unknown system response. It is a fact of mean square analysis that the lowest MSE is achieved for an L tap weight vector $\mathbf{W}$ when those L taps match the first L taps of the unknown response $h_d(i)$ exactly [13]. Appendix C shows that Equation 3.18 corresponds to the Wiener solution and is thus the optimum weight vector in a mean square error sense.

The error caused by the undermodelling of the unknown system is highly dependent on the nature of the unknown system. Later in this section it will be shown that in the case of the hybrid, the part of the hybrid response that is not modelled has a predominantly low frequency response. Thus the hybrid response has a low frequency tail which is not being modelled at all by the adaptive filter. The coefficients of the tail are still an integral part of the desired response $d(n)$ and will; therefore, also show up in the error $e(n)$. Since the adaptive filter has no hope of modelling these tail coefficients, the component of $e(n)$ caused by the tail coefficients becomes a noise which will cause some modelling error in the weight vector. This modelling error shows up as reduced echo cancellation compared to the echo cancellation that could be expected if the weight vector converged to the Wiener solution. Applying the filtered-E LMS algorithm to the hybrid modelling case significantly reduces the undermodelling error since the weighting filter effectively filters out most of the tail residual power. This shows up as significantly less misadjustment and thus better echo cancellation. Such misadjustment reduction was not realized in the room modelling cases because the room responses do not have a predominant low frequency tail. Thus the success of the filtered-E LMS algorithm in the line echo cancellation application is really partially due to the nature of the hybrid response *hybrid1*.

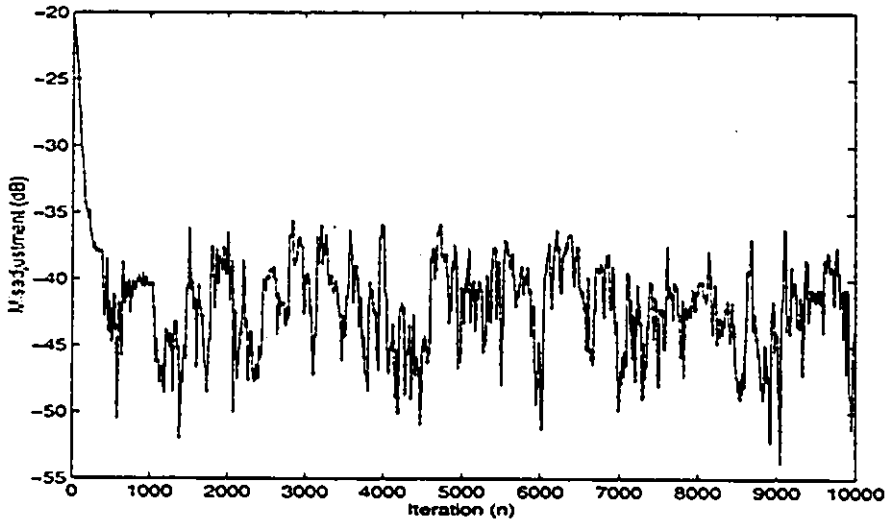

Figure 3.45: Misadjustment in the LMS trained weight vector. Unknown response: *hybrid1*. Input: white noise of variance 1.0. Parameters: $\mu = 1.0 \cdot 10^{-2}$, $L = 32$.

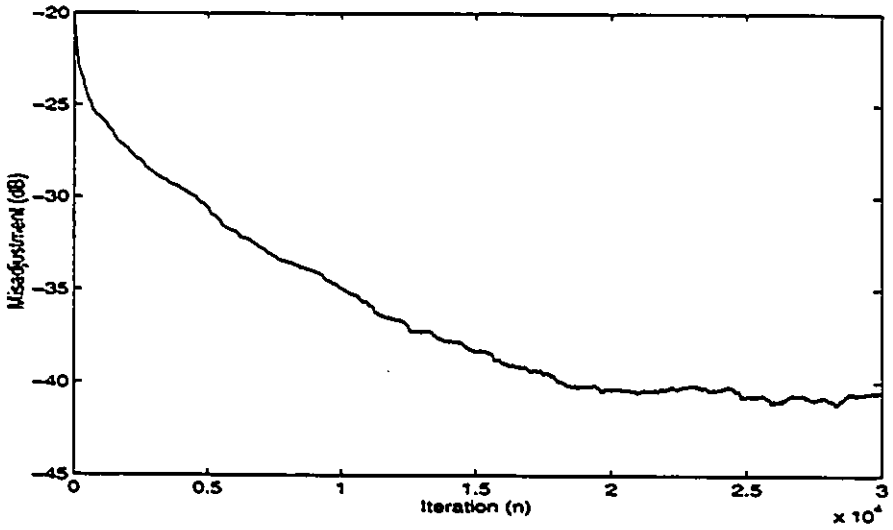

Figure 3.46: Misadjustment in the filtered-E LMS trained weight vector. Unknown response: *hybrid1*. Weighting filter $F^{(2)}(z)$. Input: white noise of variance 1.0. Parameters: $\mu = 7.0 \cdot 10^{-3}$, $L = 32$.

Figure 3.45 plots the misadjustment of the LMS trained adaptive filter, as defined above, for the hybrid modelling case and Figure 3.47 is the misadjustment for the room modelling case. In the case of the hybrid, the simple high pass weighting filter considered heavily attenuates the low frequency component of the error. This significantly alters the frequency content of the error since the hybrid has such a dominant low frequency response characteristic. Consider Figures 3.49 and 3.50 which show spectra of the modelled and unmodelled parts of the hybrid response of Figure A.7. The weight vector misadjustment, as defined in Equation 3.16 is significantly reduced for the FELMS trained weight vector when compared with the LMS trained weight vector as shown in Figure 3.46. However, for the speakerphone-room responses, which do not have such a predominant low frequency tail response (see Figures 3.51 and 3.52), the frequency weighting filter does not have as big an impact on the weight vector misadjustment, as shown in Figure 3.48.

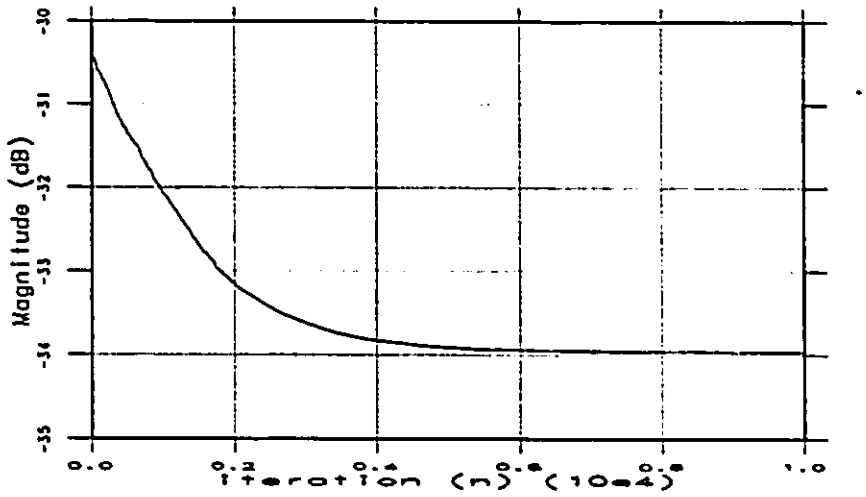

Figure 3.47: Misadjustment in the LMS trained weight vector. Unknown response: *mc3033* Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

Note that the misadjustment for the LMS trained weight vector in the hybrid modelling case (Figure 3.45) is very erratic (it has a lot of higher frequency error). After the application of the high pass weighting filter, $F^{(2)}(z)$, the misadjustment becomes very

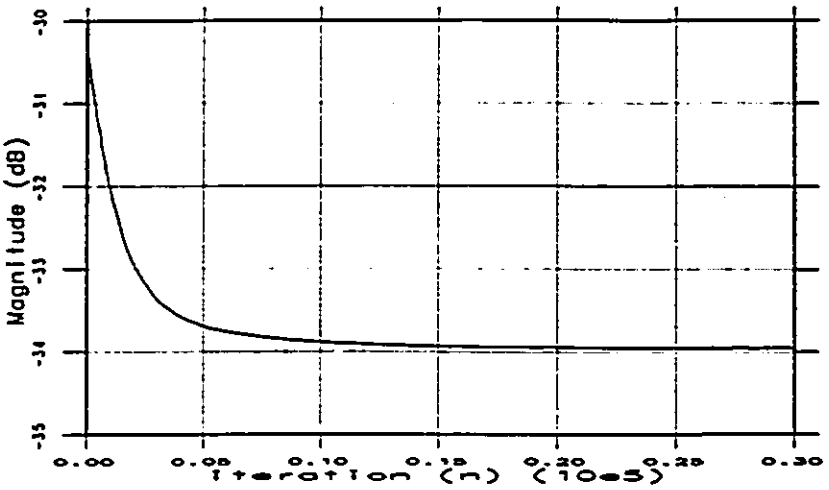

Figure 3.48: Misadjustment in the filtered-E LMS trained weight vector. Unknown response: *mc3033*. Weighting filter: $F^{(1)}(z)$. Input: white noise of variance 1.0. Parameters: $\mu = 5.0 \cdot 10^{-4}$, $L = 300$.

smooth and there is little "high frequency" misadjustment error. This confers with the error spectrum plot, Figure 3.13, which shows that there is much less higher frequency

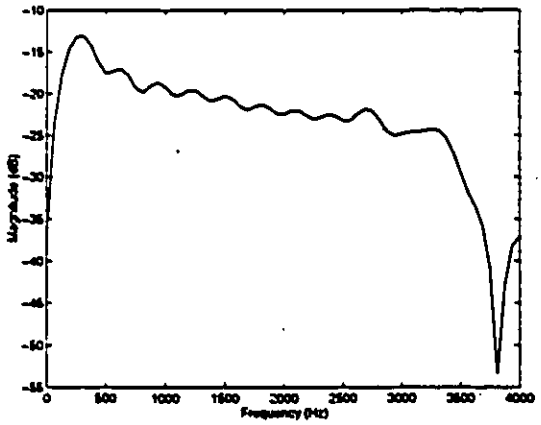

Figure 3.49: Frequency response of modelled part of *hybrid1* response.

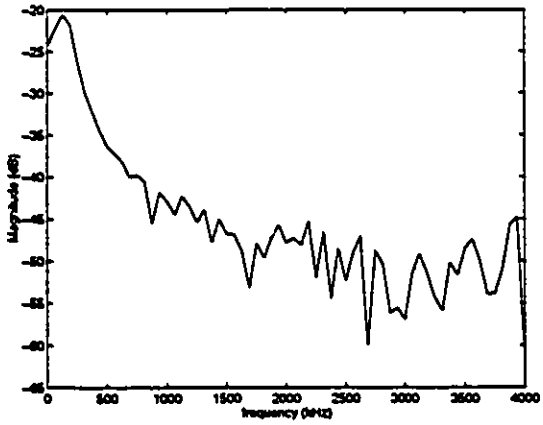

Figure 3.50: Frequency response of unmodelled part of *hybrid1* response.

error with the FELMS algorithm than with the LMS algorithm. In the speakerphone-room response modelling case, it can be seen that the misadjustment is not so erratic in

the first place (Figure 3.47). The weighting filter cannot have such a big effect in this case and indeed that is what was found (Figures 3.47 and 3.48).

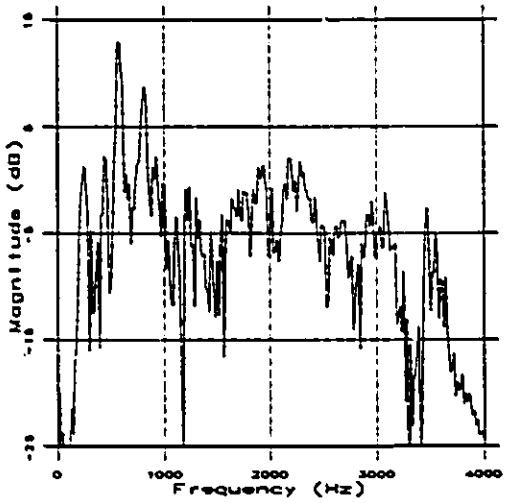

Figure 3.51: Frequency response of unmodelled part of *WBN13* response.

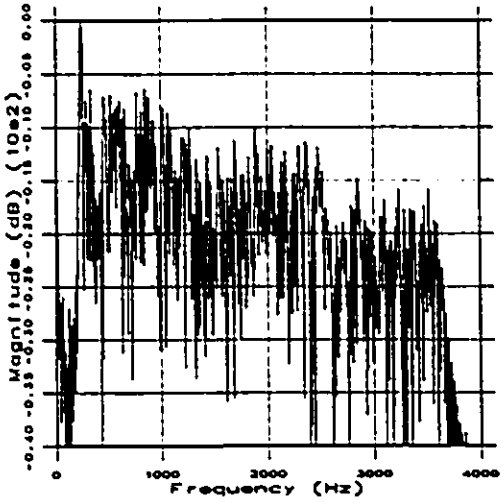

Figure 3.52: Frequency response of unmodelled part of *WBN13* response.

Thus, one can conclude that two of the main effects of the filtered-E LMS algorithm on adaptive filter performance relative to the LMS algorithm are:

1. Frequency dependent convergence rate.

2. Filtering of the undermodelling error.

These effects occur simultaneously and cannot really be separated. It can be seen most easily in the case of the hybrid modelling application how the weighting filter worked to filter out the undermodelling error, but the effect of frequency dependent convergence was also evident as slower convergence of the weight vector for low frequencies as in Figure 3.20. Perhaps the misadjustment argument can be understood more clearly by referring to Figure 3.53. In Figure 3.53, the plant response has been split into two parts: the modelled part ($P_m(z)$) and the unmodelled part ($P_u(z)$). The modelled part is of course the same length as the adaptive filter weight vector $W(z)$. $W(z)$ is capable

of perfectly matching the modelled part of the plant $P_m(z)$ (when trained with white noise) except for the misadjustment which is directly proportional to the step size and the level of undermodelling [13]. Note that the unmodelled part of the plant response can be viewed as an injected noise $r(n)$ into the adaptive filter error path where $r(n)$ represents the convolution of the input $x(n)$ and the unmodelled plant transfer function $P_u(z)$. With the filtered-E LMS algorithm, this injected noise is filtered by the weighting filter $F(z)$. Thus, if the noise $r(n)$ is predominantly of a frequency attenuated by the weighting filter (as it is in the hybrid modelling case) the amount of noise injected into the adaptive filter error path will be less and the weight vector will thus have less misadjustment. On the other hand, if the injected noise $r(n)$ is not significantly attenuated by the weighting filter $F(z)$ then misadjustment levels will be much the same for both the LMS and the FELMS algorithms.

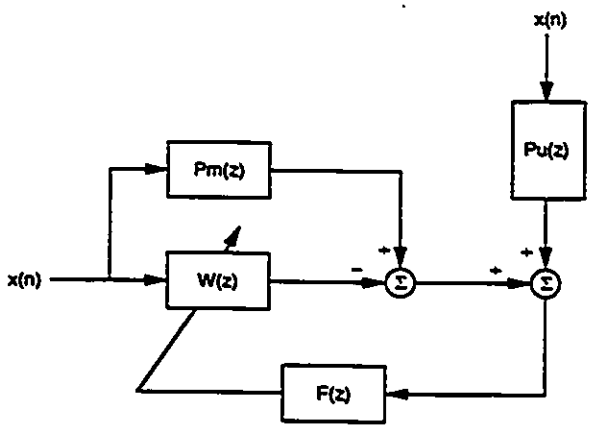

Figure 3.53: Unmodelled plant response "noise" in error path.

3.9 Steady State Performance of Filtered-E LMS

Some further insight into the filtered-E LMS adaptive filter performance can be gained by considering its steady state performance. Assume that the weight vector of Figure 3.6 has converged to the optimum solution as given by Equation 3.18 and that the step size μ is extremely small. The filter blocks of Figure 3.6 can be approximately

considered as linear time invariant blocks and thus they are commutable. The blocks can be re-arranged to produce an equivalent "steady state" filtered-E LMS adaptive filter structure as in Figure 3.54. What is shown in Figure 3.54 is simply an LMS adaptive filter with a coloured input $x_f(n)$. If the colouring filter $F(z)$ is not too severe (there is sufficient energy content at all frequencies) then the weight vector should converge to the Wiener solution as given by Equation C.2. In other words the filtered-E LMS algorithm cannot be expected to change the optimum weight vector to which the adaptive filter tries to converge.

Since the optimum weight vector is the same for both the LMS and the filtered-E LMS algorithm, the only factors over which the filtered-E LMS algorithm can exert some control are the rate of convergence and the weight vector misadjustment. These facets of the adaptive filter performance have already been discussed and their behaviour under the filtered-E LMS algorithm was uncovered. It was seen that given an unknown system with an unmodelled tail response with a dominant frequency characteristic, the filtered-E LMS algorithm could significantly reduce weight vector misadjustment, resulting in a frequency weighted echo spectrum.

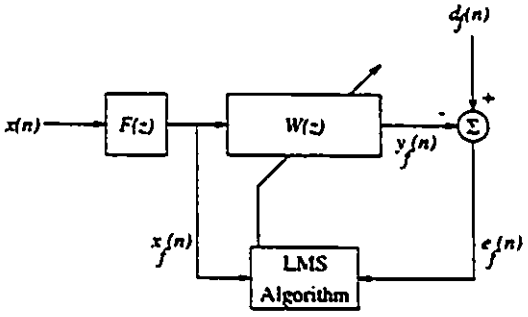

Figure 3.54: Equivalent filtered-E LMS adaptive filter structure for steady-state.

As mentioned earlier, the filtered-X LMS structure has already been applied to the acoustic echo cancellation application, albeit with slightly different goals. In [21] the filtered-X LMS algorithm is used with an adaptive $F(z)$. The purpose of the algorithm presented in [21] is to speed up convergence of the weight vector for coloured inputs. The analysis indicates that the weight vector will still converge to the same optimum weight vector as the LMS algorithm. Another adaptive filtering structure with the same basic block diagram as the filtered-E LMS structure is presented in [23]. Again the purpose of the algorithm presented in [23] is to speed up convergence of the weight vector with the expectation that the weight vector will reach the same optimum weight vector as in

the LMS case.

3.10 Effective Plant Response Length

In Section 3.9 it was seen that the weighting filter $F(z)$ effectively colours the input. By colouring the input, one is effectively changing the plant response being modelled. The question is: can this colouring process lead to less error in the effective band of the coloured input? In the cases studied here, a white noise process is the input to the adaptive filter. Thus, the weighting filter acts to colour the white noise before filtering by the plant response. In effect, the white noise is input to a new plant response which is the convolution of the colouring (weighting) filter and the original plant response. When the colouring filter is simply the impulse response (no colouring) the LMS algorithm

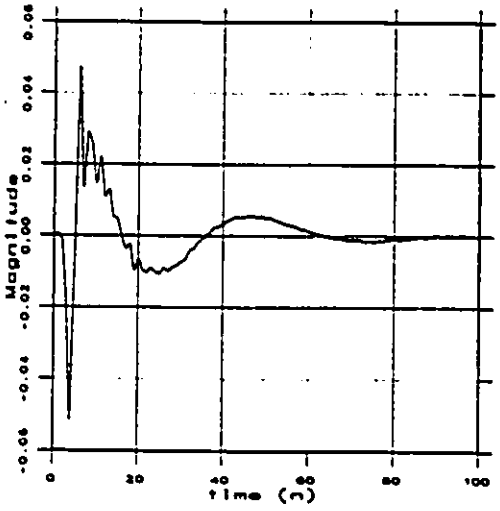

Figure 3.55: *hybrid1* response.

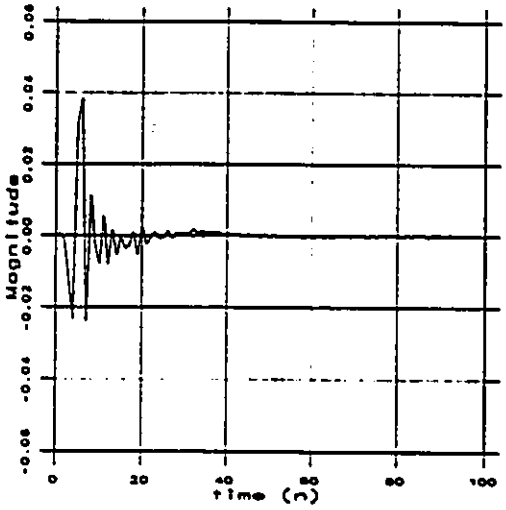

Figure 3.56: *hybrid1* convolved with $F^{(2)}(z)$.

is the result. Figures 3.55 and 3.56 show response *hybrid1* by itself and filtered by the high pass weighting filter $F^{(2)}(z)$, respectively. It is evident that the weighting filter $F^{(2)}(z)$ has effectively shortened the plant response significantly. This illustrated by the

TIP/TP plots in Figure 3.57 showing that the TIP/TP for the weighted hybrid response rises much quicker initially. The FELMS algorithm is able to achieve reduced in-band echo (in-band signifying the band of the weighting filter) relative to the LMS algorithm due to the effectively shorter plant response. Results in this section are reproduced based on the work of Mayyas and Aboulnasr [80].

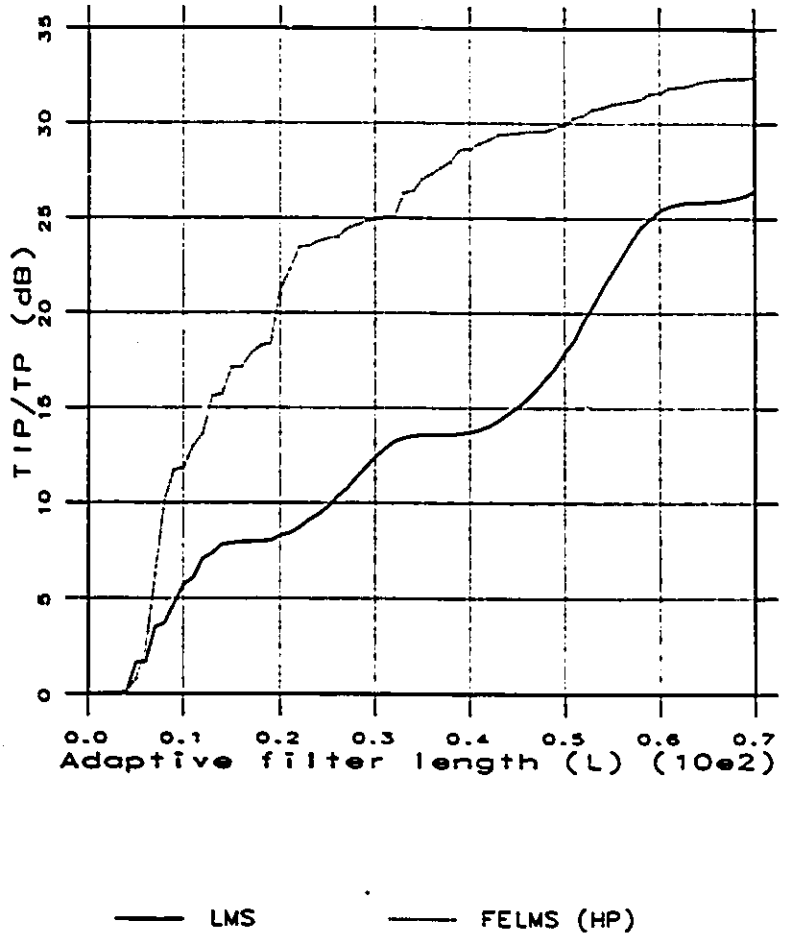

Figure 3.57: TIP/TP of weighted and unweighted *hybrid1*.

Response *hybrid1* was effectively shortened considerably by the weighting filter. Figures 3.58 and 3.59 show the weighted and unweighted response of speakerphone-room

response *WBN13*. Obviously, the weighting filter has done little to effectively shorten the plant response. Thus the FELMS algorithm has to work just as hard as the LMS algorithm and for the same number of taps cannot achieve a reduction of in-band echo. Figure 3.60 shows the TIP/TP curves for *WBN13*. Note that these TIP/TP curves fall virtually on top of each other for adaptive filter lengths up to 400 taps or so. For longer adaptive filters it appears that some very small gains may be achieved in in-band echo cancellation by the FELMS algorithm over the LMS algorithm. To illustrate this, Figure 3.61 shows echo spectra after convergence for the two algorithms. The adaptive filter lengths of both algorithms are set to 900 taps. Step sizes were chosen so that the two algorithms converged to the same steady state MSE. Note that this means choosing a significantly lower step size for FELMS and hence the FELMS algorithm suffers from very slow convergence. The in-band gains in echo cancellation of FELMS over LMS are evident from Figure 3.61 but these gains are indeed very slight, as indicated by the TIP/TP curves of 3.60.

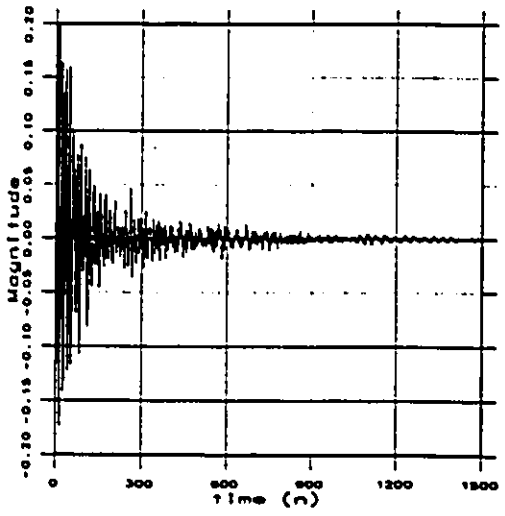

Figure 3.58: *WBN13* response.

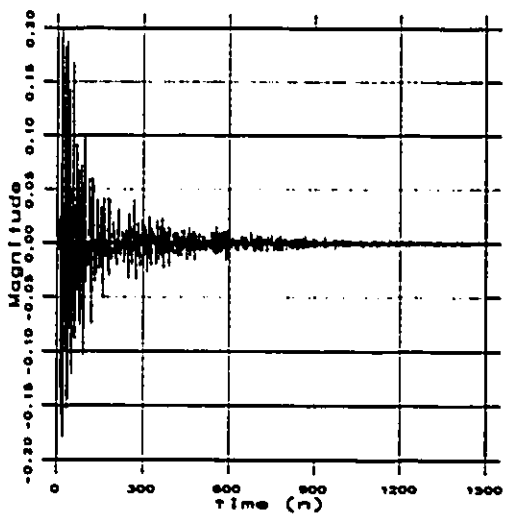

Figure 3.59: *WBN13* convolved with $F^{(1)}(z)$.

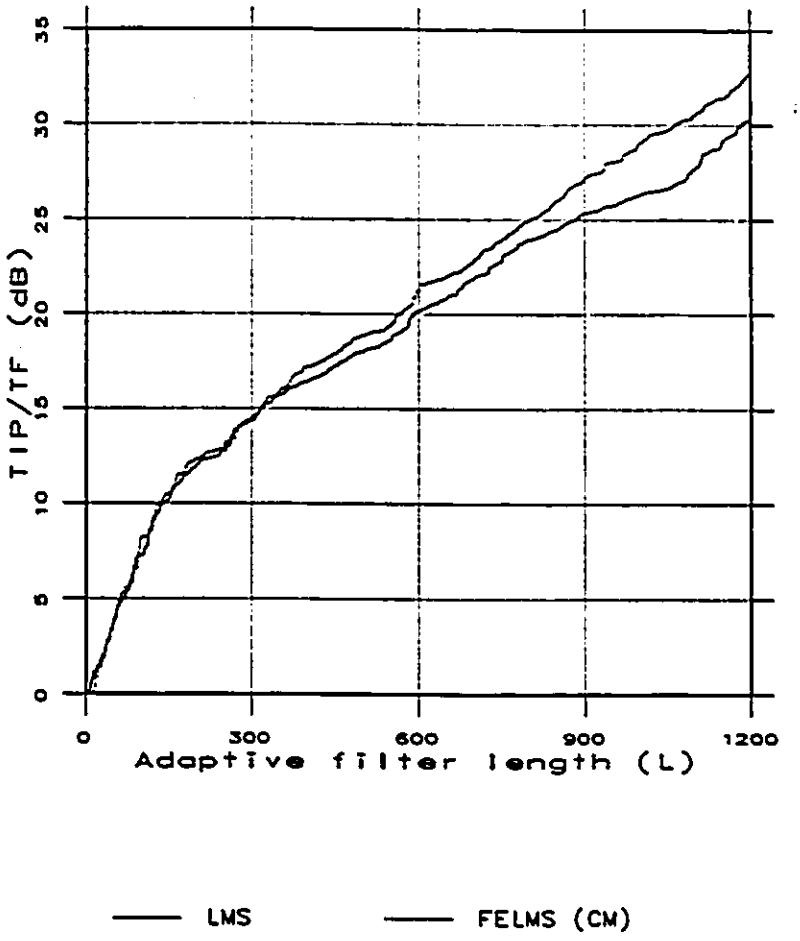

Figure 3.60: TIP/TP of weighted and unweighted *WBN13*.

In [28] several simulations with the filtered-E LMS algorithm were presented which showed that FELMS was achieving the desired spectral shaping over the FXLMS algorithm. However, it must be noted that the "unknown response" being modelled in [28] was a simple lowpass filter and not a true acoustic environment. The lowpass filter modelled in [28] had a cut-off frequency of 950 Hz. The two weighting filters used in [28] both applied attenuation to the frequency region near the cut-off frequency. By attenuating the region of the lowpass unknown response near the cut-off frequency, the effective response of the lowpass unknown response was considerably shortened. This is due to the fact that for a lowpass filter, long filter lengths are needed to achieve a sharp cut-off. The frequency information of the filter is concentrated near the cut-off frequency and filtering out this region corresponds to reducing the information in the original response. Figures 3.62 and 3.63 show the time responses of a lowpass filter of 100 taps with a cut-off frequency of 3500 Hz with and without a weighting filter, respectively. The weighting filter used is a 2 tap lowpass filter $F^{(3)}(z)$ (described earlier). Figure 3.64 shows the TIP/TP curve for the 100 tap lowpass filter and for the 100 tap lowpass filter convolved with weighting filter $F^{(3)}(z)$.

The discussion in this section has focussed on the in-band effect of the FELMS algorithm over the LMS algorithm. Here, in-band refers to the weighting filter $F^{(i)}(z)$'s band. Out of band frequency components of the echo have to be viewed differently since the weighting filter is actually applied after the plant response has been excited at all frequencies (for white noise inputs). Thus, all frequency components of the plant response are present in the echo unless attenuated by the adaptive algorithm (Figure 3.6). Thus frequencies which are attenuated heavily by the weighting filter do not really participate in the adaptive filter updates but still show up in the final echo. This means that, effectively, out-of-band echo for the FELMS algorithm is basically at an uncancelled level whereas in-band echo experiences cancellation dependent on the in-band effective plant response length, as discussed in this section.

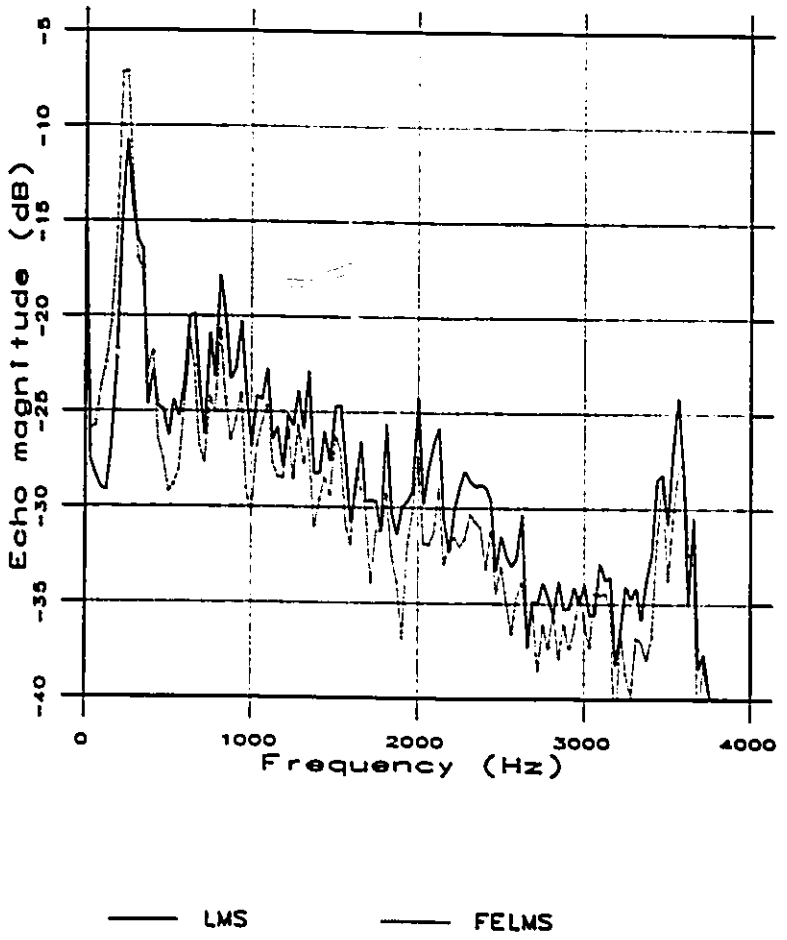

Figure 3.61: Error spectra after convergence for the LMS and FELMS algorithms. FELMS weighting filter: $F^{(1)}(z)$. Unknown system: *WBN13*. Input: white noise of variance 1.0. Parameters: $\mu = 1.5 \cdot 10^{-4}$ (FELMS), $\mu = 1.0 \cdot 10^{-3}$ (LMS), $L = 900$ (both algorithms).

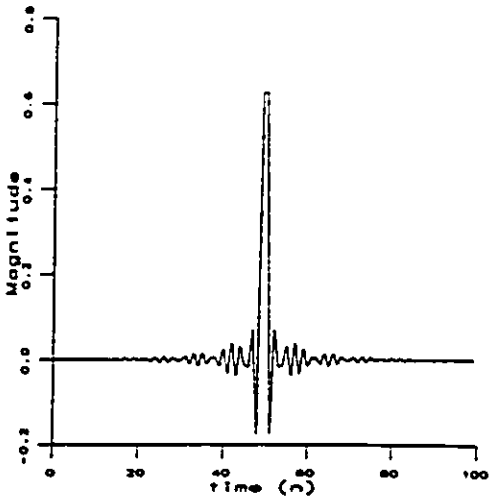

Figure 3.62: Lowpass filter response.

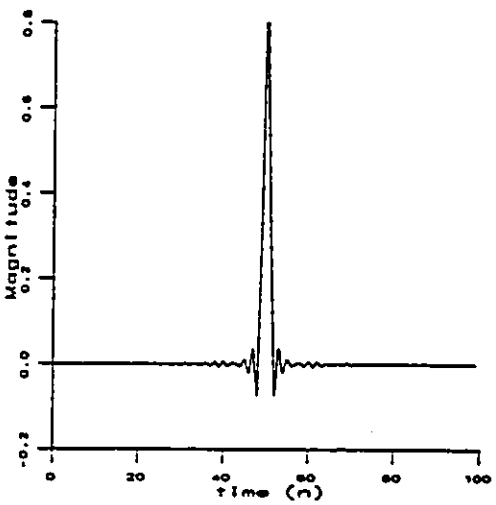

Figure 3.63: Lowpass filter convolved with $F^{(3)}(z)$.

3.11 Summary

This chapter has investigated the feasibility of weighting the error in the frequency domain in order to obtain better echo cancellation at psychoacoustically important frequencies. The structure proposed for this task was the filtered-E LMS structure. The filtered-E LMS algorithm has been applied in the areas of active noise control and line echo cancellation with promising results in terms of frequency weighting capabilities. The structure of the filtered-E LMS algorithm was described and the weight vector update equation was derived.

The choice of the filtered-E LMS weighting filter $F(z)$ was described. The motivation for the choice of this weighting filter was psychoacoustic principles. The chosen filter should emphasize frequencies to which the human ear is sensitive and de-emphasize frequencies to which the human ear is not very sensitive. The C-message frequency weighting curve was proposed as a suitable curve for this task. The C-message curve reflects the sensitivity of the human ear to sounds of frequencies in the range 0 to 5 kHz.

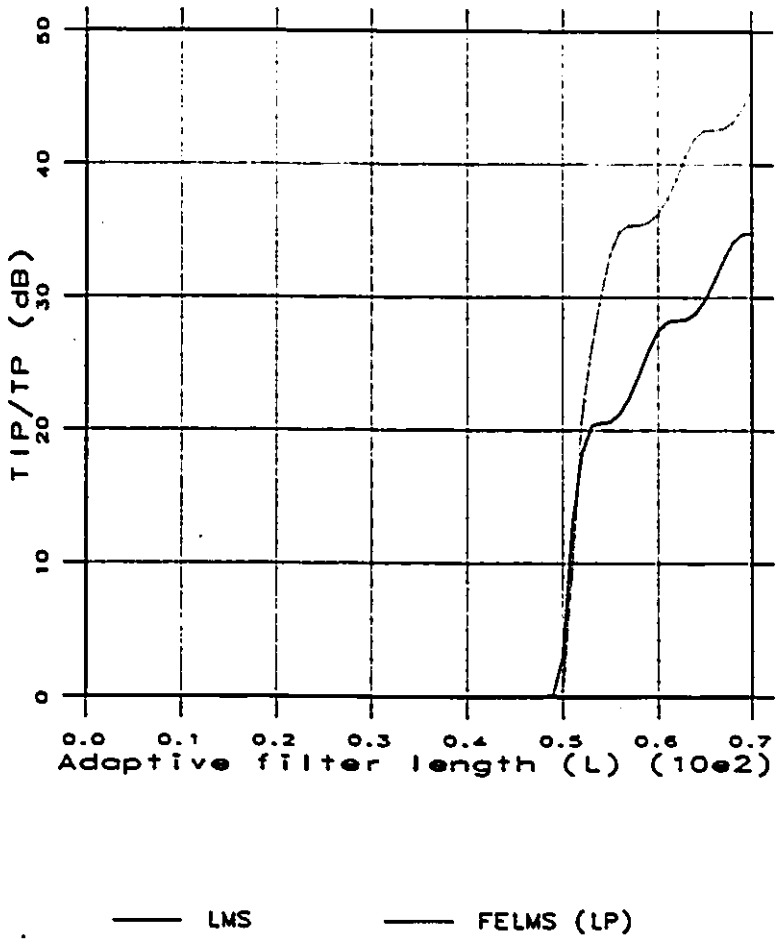

Figure 3.64: TIP/TP of weighted and unweighted lowpass filter.

The performance of the filtered-E LMS algorithm was examined from several points of reference: level of undermodelling, nature of the plant response, weight vector misadjustment, rate of convergence and effective plant response length. The filtered-E LMS algorithm was found to converge much more slowly than the LMS algorithm for frequencies which are heavily attenuated by the weighting filter. Frequencies which were emphasized by the weighting filter converged as fast or faster under the filtered-E LMS algorithm as compared with the LMS algorithm. In effect, the filtered-E LMS algorithm implements the LMS algorithm with a frequency dependent step size. The frequency response of the weighting filter determines the step sizes for each of the frequencies.

The weighting filter of the filtered-E LMS algorithm was found to have a profound effect on weight vector misadjustment in a line echo cancellation application. The unknown response being modelled, *hybrid1*, had a very low frequency tail which died out slowly and was not modelled by the relatively short adaptive filter. Thus, this tail caused a lot of weight vector misadjustment, resulting in significantly greater echo (especially at higher frequencies) for the LMS algorithm. The filtered-E LMS algorithm, using a high pass weighting filter, effectively attenuated the tail echo power resulting in less weight vector misadjustment and thus better performance at higher frequencies.

Unfortunately the filtered-E LMS structure did not provide any good results in the acoustic echo cancellation simulations. The promising results obtained in the fields of active noise control [28] and line echo cancellation [29] did not carry over to the field of acoustic echo cancellation. The filtered-E LMS structure was unable to achieve any frequency weighting of the error when the unknown system was one of the room responses *WBN13* or *mc3033*.

The filtered-E LMS algorithm performed as expected, achieving better echo cancellation than the LMS algorithm at emphasized frequencies, when the plant response was characterized by dominant frequency regions, as *hybrid1* was and as a lowpass filter such as modelled in [28] was. By de-emphasizing the dominant frequency region, better echo cancellation could be obtained at other frequencies. However, better performance could not be achieved by emphasizing the dominant frequency region. In effect, by de-

emphasizing the dominant frequency region, the plant response length was shortened considerably, allowing the FELMS algorithm to achieve better echo cancellation for emphasized frequencies. With the dominant frequency region de-emphasized, the FELMS algorithm no longer had to try as hard to match the frequency characteristics of the "hard-to-match" portion of the plant frequency response and could achieve better plant frequency response matching at "easier-to-match" frequencies. In the cases of responses *WBN13* and *mc3033* there were no "easier-to-match" frequency response sections and therefore, no matter what part of the frequency region was de-emphasized, the FELMS algorithm could not better the echo cancellation achieved by the LMS algorithm. The effect of de-emphasizing and not de-emphasizing the dominant frequency region of a plant response was clearly demonstrated in Section 3.7.

Filtered-E LMS performance was also examined from the perspective of weight vector misadjustment. This is essentially another way of looking at the same thing. Plant responses which are good candidates for the FELMS algorithm have unmodelled tails (those coefficients in the plant response which extend beyond the length of the adaptive filter) with a very dominant frequency characteristic. Response *hybrid1*, for example, contains a dominant low frequency tail. If the weighting filter used with the FELMS algorithm serves to filter out that part of the error caused by the plant unmodelled tail response then weight vector misadjustment is significantly reduced and better echo cancellation can be attained at emphasized frequencies.

In conclusion, the speakerphone-room responses *WBN13* and *mc3033* studied in this thesis were not good candidates for the FELMS algorithm. They had no dominant frequency characteristics and their frequency responses show no smooth regions where one might be able to obtain better echo cancellation by emphasizing that region. Their frequency responses can be seen to be constantly changing (always in transition) which may be characteristic of effectively having transfer functions with many, many poles.

CHAPTER 4

Adaptive Tap Assignment

In the pursuit of an adaptive filter which can arbitrarily shape the spectrum of the echo, it is natural to turn to a subband adaptive filter. Originally, subband adaptive filters have been given a fixed tap assignment. That is, if there are a total of $M \cdot L$ taps to be distributed among the subbands then each of the M subbands will have an adaptive filter L taps long. It can be expected, that if the $M \cdot L$ taps are distributed in a nonuniform fashion, the spectrum of the echo will be shaped in some manner. Giving more taps to one band than to another should mean more echo cancellation in the band with more taps and less echo cancellation in the band with less taps. The trick, of course, is to find a way in which to adaptively distribute the taps and to find a criterion upon which this adaptive tap distribution can be based.

4.1 Subband Adaptive Filter Structure

The generic subband adaptive filter structure was presented earlier in Figure 2.9 of Chapter 2. The filter structure remains the same for the work here. Some of the parameters of the adaptive filter bank structure can be specified immediately. $M = 8$ has been chosen as the number of subbands. The use of the subband adaptive filter structure gives one a piece-wise way of shaping the echo spectrum. Eight subbands were chosen so that a fair degree of flexibility would be available when shaping the echo

spectrum. The rest of the parameters of the filter bank structure, such as the analysis filters used and the decimation rate R will be considered in Sections 4.2 and 4.3.

4.2 Filter Bank Design

The filter banks used in this thesis were chosen to satisfy the perfect reconstruction (PR) property. That is, if no processing were done in between the analysis filter bank and the synthesis filter bank then a signal passed through the entire structure would suffer no distortion. The only effect of the structure would be to delay the signal by $L_{AS} - 1$, where L_{AS} is the length of the analysis and synthesis filters. Note that all of the analysis and synthesis filters are of identical length in the filter bank structure used here.

The design procedure for the analysis and synthesis filters used here is given in [44]. A brief summary of it is presented here. Initially, one designs a "prototype" lowpass analysis filter for a two channel filter bank. In order to generate filter banks with more channels, the simple two channel filter bank must be decomposed. The restriction in this procedure is that the final number of channels in the filter bank must be a power of 2. A filter bank generated from a simple two channel filter bank is known as a "tree structured" filter bank. The perfect reconstruction property is preserved through any number of decompositions.

In order to generate an M-channel filter bank one can also use a direct M-band decomposition method, as in [47]. The M-channel decomposition techniques require nonlinear optimization procedures in order to design the constituent filters of the filter banks. Tree structured filter banks have a much simpler design procedure and were chosen for this reason.

Below, the procedure for implementing a four channel tree structure filter bank is described. This procedure can easily be generalized to the design of an eight channel filter bank. Figure 4.1 shows the tree structure for $M = 4$. Note that only the analysis filter bank is considered here. The synthesis filter bank can be generated in an analogous

manner [44]. This tree structure can be implemented as is or it can be implemented in "parallel" form. The parallel form is illustrated in Figure 4.2.

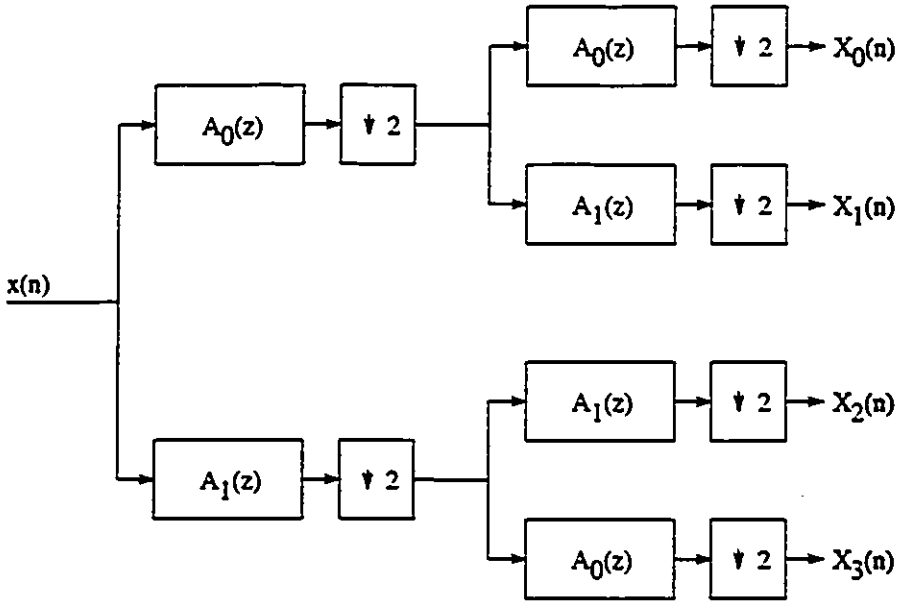

Figure 4.1: Tree structured analysis filter bank for $M = 4$.

In order to generate the parallel form analysis filter bank from the tree structured analysis filter bank one makes use of the noble identities [46]. These identities are illustrated in Figure 4.3. Once the filter bank is in the parallel form structure all of the decimation takes place after filtering. The literature on tree structured filter bank design uses maximally decimated filter banks. That is, the rate of decimation in each subband is equal to the total number of subbands. For the four band case being described here, the rate of decimation is four. However, in Chapter 2 some problems with using maximum decimation for adaptive filtering were mentioned. It is desirable then to use a decimation rate which is less than the maximum. It turns

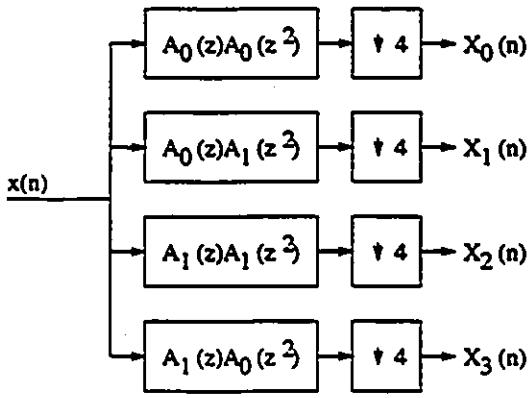

Figure 4.2: Parallel form analysis filter bank for $M = 4$.

out that one can use the same filter banks as were designed for the case of maximal decimation.

Given a set of analysis and synthesis filters and a decimation rate, an "aliasing component" or AC matrix can be constructed [45]. The given filter bank satisfies the PR properties if its AC matrix is unitary on the unit circle of the z-plane [47]. The AC matrix is a function of the filters which make up the filter bank and the decimation rate of the filter bank structure. Thus in order to design a perfect reconstruction filter bank with a decimation rate less than the maximum what one need do is design the perfect reconstruction filter bank with maximum decimation rate according to [44], then construct the AC matrix for the designed filters and the desired decimation rate. If the AC matrix is unitary on the unit circle of the z-plane then the filter bank satisfies the properties of perfect reconstruction. For a tree structured perfect reconstruction filter bank with M a power of two, the decimation rate R must also be a power of two. For the eight band filter bank to be used in this thesis, the possible rates of decimation are 1 (no decimation), 2, 4 and 8 (maximal decimation).

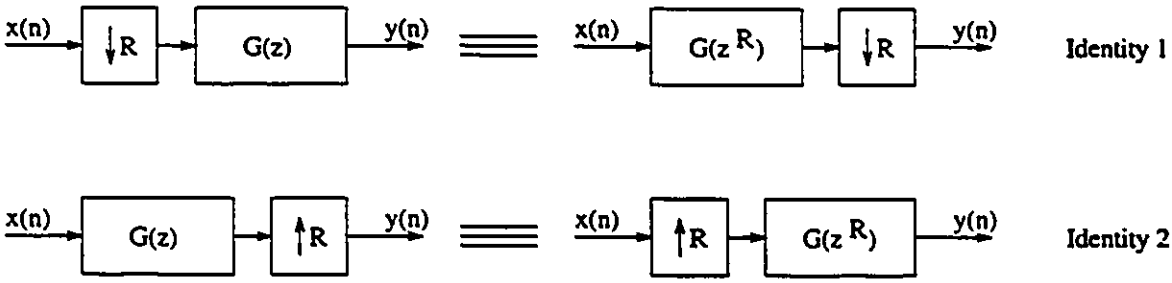

Figure 4.3: The noble identities for multirate systems.

In order to design an 8 band filter bank a 16 tap low-pass prototype filter given in [44] was employed. This prototype filter has 40 dB of stopband attenuation. The procedure described above for generating the parallel form filter bank was then used to produce an eight band filter bank structure. In the parallel form, each of the eight bands of the analysis filter bank has 106 taps and 40 dB of stopband attenuation. The decimation rate R was chosen to be one. That is, no decimation was applied to the subband signals, as discussed in Section 4.3.

4.3 Decimation Rate

In Section 2.8, subband adaptive filtering was discussed. Mention was made of aliasing effects which occurred when the filter banks were maximally decimated. Maximal decimation refers to having the decimation rate equal to the number of subbands. While attractive from a computational savings point of view, the performance of the subband adaptive filter is unacceptable when these aliasing effects are present. One is left to choose a decimation rate R less than M [5, 63].

However, even non-maximally decimated filter banks suffer from some performance degradations. These effects and ways to overcome them are studied in [59, 62, 63, 64]. This thesis will focus on the problem at hand, that of applying psychoacoustic criteria to acoustic echo cancellation, rather than get lost in the quagmire of subband adaptive filtering. Thus, the filter bank structure used in this thesis, in common with the work of [57, 58], does not decimate any of the subband signals ($R = 1$). Having bypassed the issue of filter bank performance, the issues underlying the implementation of an adaptive filter with frequency-dependent performance can be examined more directly.

It should be noted however that the use of a subband adaptive filter without decimation is unattractive from a computational complexity point of view. An eight band subband adaptive filter with no decimation will require more than eight times as many multiplications per sample as a fullband adaptive filter. Even a non-maximally decimated subband adaptive filter could provide some computational savings over the fullband case if the decimation rate is high enough [50]. In the case here, given the restrictions on the choice of the decimaton rate R, it would be necessary to bring the decimation rate up to 4 in order to bring the computational complexity of the subband adaptive filter in line with the fullband case. However, this issue is not covered in the work of this thesis.

4.4 Nonstationary Performance of NLMS

4.4.1 Fullband NLMS

Before considering the design of the subband adaptive filtering algorithm there are some issues related to the performance of the NLMS algorithm with which to deal. The NLMS update equation was presented earlier as Equation 2.12. Note that the effective step size of this update equation can be written as:

$$\mu_{eff} = \frac{\mu}{\sigma + P_x}. \tag{4.1}$$

Very little is said in the literature about the choice of the parameter σ. The factor σ is merely suggested as a "fudge factor" to be used to restrict μ_{eff} from becoming excessively large. As such it has been suggested to choose a "very small number" for σ. Such a choice is merely meant to control the step size when the input power estimate P_x becomes very small or zero. Such a problem is virtually nonexistent when dealing with a white noise training signal and thus such a problem must be studied using a nonstationary signal such as speech as input. Some recent literature has begun to address this issue [17].

The technique presented in [17] is to interrupt the updates of the weight vector when P_x falls below a certain level. In this thesis a very similar technique is used. The effective step size is restricted from exceeding a certain level. If the effective step size tries to exceed this level then it is clamped at this level. This is akin to placing a cap on the effective step size as follows:

$$\mu_{eff} < \mu_{max}. \tag{4.2}$$

With such a technique, σ becomes an unnecessary factor. It is illustrative to view a plot of μ_{eff} versus P_x for the various methods of restricting the values of μ_{eff}. In Figure 4.4, three variations on the NLMS update given above are considered. These three cases are:

Case 1. $\mu_{eff} = \mu/P_x$.

Case 2. $\mu_{eff} = \mu/(\sigma + P_x)$.

Case 3. $\mu_{eff} = \mu/P_x$ iff $\mu_{eff} < \mu_{max}$ else $\mu_{eff} = \mu_{max}$.

The NLMS update technique of [17] is a fourth option, given by

Case 4. $\mu_{eff} = \mu/P_x$ iff $\mu_{eff} < \mu_{max}$ else $\mu_{eff} = 0$.

In Figure 4.4, the choice of the parameters is meant to reflect typical values found in the speech signals used throughout the simulations of this thesis. The average power found in a speech signal was approximately $3 \cdot 10^6$. (Note that this average signal power is based on a 16 bit integer representation of the speech signals. The speech signals were normalized before quantization so that the signal powers specified here bear no relation to any phyiscal quantity.) The cap μ_{max} placed on μ_{eff} was chosen according to the theoretical maximum step size for the LMS algorithm. This gives

$$\mu_{max} = \frac{2}{L \cdot E(x^2(n))} \approx 6.7 \cdot 10^{-7}. \tag{4.3}$$

For the purpose of comparison, L was set equal to one in Equation 4.3 for Figure 4.4. The step size μ was also set to one as it was for most simulations. Only the portion of the curve for which the differences in the three methods of NLMS update are apparent is shown. It is evident that μ_{eff} grows without bound for decreasing input power if no limit is placed on it. The third method of capping the effective step size is perhaps the most attractive since it uses the theoretical NLMS effective step size, μ/P_x exactly except when this quantity is deemed to be too large. In Case 2, where σ is used, the effective step size approaches the theoretical NLMS effective step size when P_x is large, but for much of the input signal power range the effective step size remains below the theoretical NLMS effective step size. In Figure 4.4, σ was chosen to be 10^6, or about the same order of magnitude as the power of the input speech sequences used in this thesis. Such a large choice of σ does not concur with the general advice in the signal processing literature "to make σ a very small number". If σ is chosen to be a very small number (about 50 dB lower than the expected input power), then the weight vector misadjustment is very erratic, as in Figure 4.5. A larger value of σ gives lower

misadjustment and leads to better echo cancellation. It was determined experimentally that values of σ on the order of the input signal power gave the best performance in terms of reducing weight vector misadjustment. Of course, if σ is chosen to be too large then the convergence speed of the NLMS algorithm will be harmed considerably.

Initial results using the NLMS algorithm with speech input were quite perplexing. The error of the adaptive filter was found to vary quite erratically. This was later attributed to the use of the NLMS update with a small value of σ, as described above. The factor σ had always been chosen to be a small number, usually about 50 dB below the level of the input signal. Such a choice conformed to the literature's advice to pick a very small number for σ. It was later found that even with a small value of σ, the effective step size became very large when the input signal fell to zero and the weight vector tended to deviate from the optimum solution. A measure that can be used to assess the performance of an adaptive filter is the misadjustment, abbreviated as $mis(n)$. This quantity was defined in Equation 3.16.

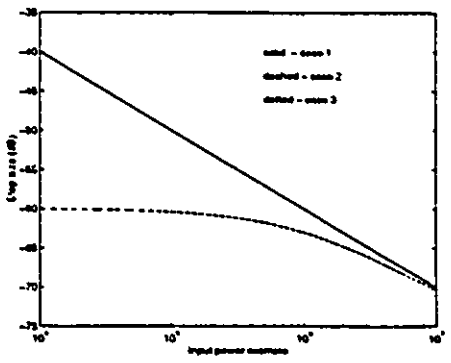

Figure 4.4: Limited and unlimited μ_{eff}.

In Figure 4.5, $mis(n)$ of a 100 tap fullband adaptive filter is plotted. The input is speech signal *htf* and the room response is *WBN13*. The parameters of the adaptive filter are $\mu = 1$ and $\sigma = 10$. Note that σ is approximately 50 dB below the input power level. Note how $mis(n)$ shoots up at certain points. It can be shown that the points at which $mis(n)$ shoots up are the points at which P_x reaches a minimum. When P_x nears zero and the input $x(n)$ begins to rise again, the NLMS algorithm is at a vulnerable point. Due to the fact that P_x is the sum of the past L values of $x^2(n)$ there is an inherent delay in the power estimate and there are times when the relationship $P_x \propto L \cdot E(x^2(n))$ may not hold. Figure 4.6 shows the relationship between $mis(n)$, $P_x(n)$ and $x(n)$. The region of time plotted corresponds to the point where $mis(n)$ takes its biggest jump

in Figure 4.5, around iteration 11500. Clearly $mis(n)$ jumps up when $P_x(n)$ reaches a minimum and then $x(n)$ begins to rise again.

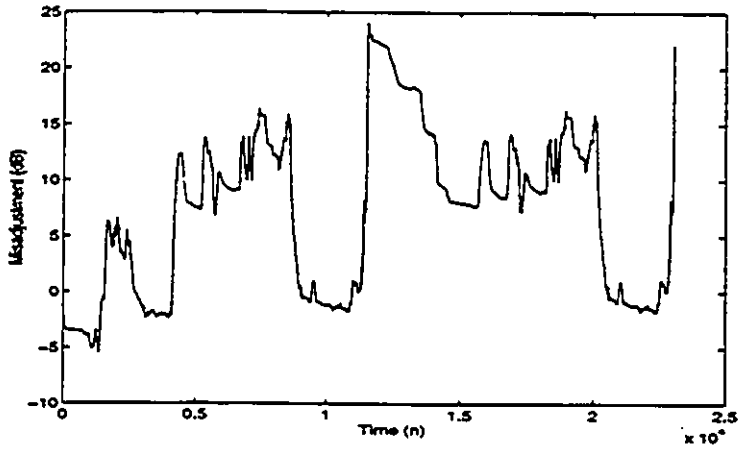

Figure 4.5: $mis(n)$ for an NLMS trained adaptive filter. Parameters: $\mu = 1.0, L = 100, \sigma = 10$. Unknown response: *WBN13*. Input: *htf*.

The approach used to resolve this misadjustment problem has already been given. The effective step size of the NLMS update equation must be limited. Of the two methods described earlier (in Figure 4.4) the second one remains closer to the true NLMS algorithm and thus it will be the approach used throughout the remainder of the thesis. The approach of simply limiting the effective step size rather than interrupting adaptive filter updates when the effective step size reached the limit as in [17] was chosen since this approach is closer to the 'true" NLMS algorithm (no step size limitations whatsoever). Informal simulations showed that both approaches effectively prevented large jumps in weight vector misadjustment, although the step size limiting approach limited the jumps in weight vector misadjustment slightly more than the step size interruption approach. In the step size limiting approach, μ_{eff} is chosen according to the following criterion:

$$\mu_{effective} = \begin{cases} \frac{\mu_{NLMS}}{P_x(n)} & \text{if} \quad \frac{\mu_{NLMS}}{P_x(n)} < \mu_{max} \\ \mu_{max} & \text{otherwise} \end{cases} \tag{4.4}$$

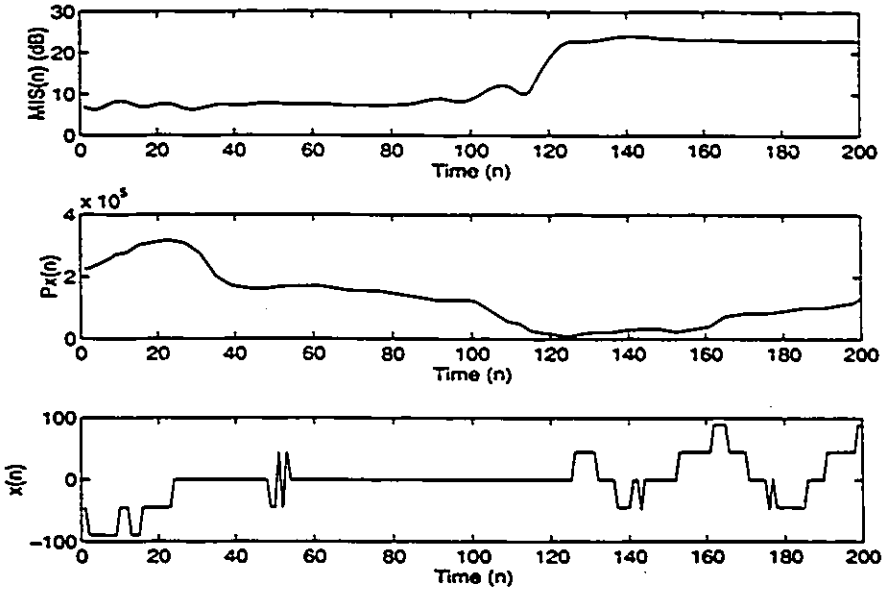

Figure 4.6: Relationship between $mis(n)$, $P_x(n)$ and $x(n)$.

where

$$\mu_{max} = \frac{2}{L \cdot E(x^2(n))}. \tag{4.5}$$

Note that μ_{max} corresponds to the theoretical maximum step size for the fixed step size LMS algorithm. For the simulations in this thesis, an a priori estimate of the input signal power ($E(x^2(n))$) is required by all algorithms. To derive such an estimate, the average input power was calculated for each of the speech sequences used in this thesis (*m98, m54* and *htf*) and these average powers were themselves averaged to produce a "typical" input signal power level. This "typical" input signal power level was then used to calculate one μ_{max} to be used for all simulations. In Figure 4.7, $mis(n)$ is plotted for the same adaptive filter as above with the new restriction on μ_{eff}. All other parameters remain the same. In order to illustrate the effect of the limit on μ_{eff}, Figure 4.8 shows the averaged error spectra for the same adaptive filter using the small σ value as in Figure 4.5 and using the limit on μ_{eff} as in Figure 4.7. Note that limiting μ_{eff} has resulted in a drastic reduction in the misadjustment of the weight vector and a corresponding reduction in the echo level.

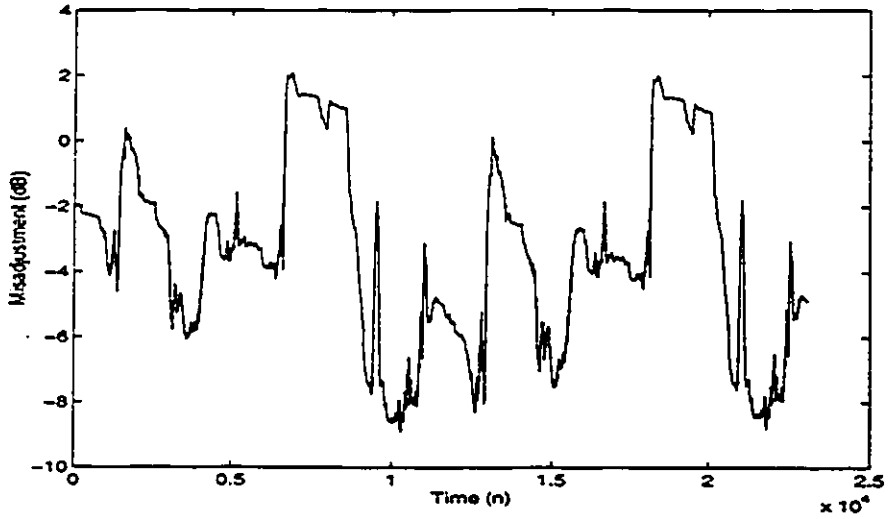

Figure 4.7: $mis(n)$ for an NLMS trained adaptive filter. Parameters: $L = 100, \mu = 1.0, \mu_{max} = 6.7 \cdot 10^{-9}$. Input: *htf*. Unknown response: *WBN13*.

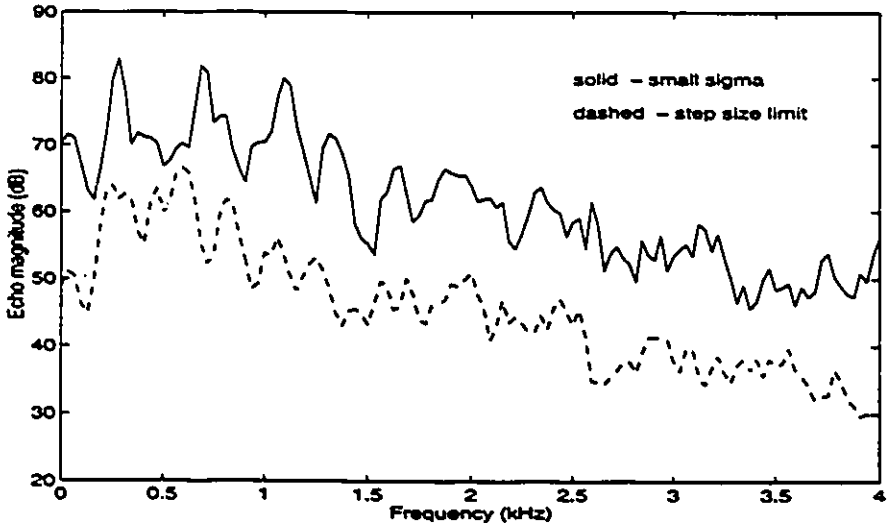

Figure 4.8: Error spectra after convergence for adaptive filters with small sigma and step size limit. Parameters: $L = 100, \mu = 1.0, \sigma = 10$, $\mu_{max} = 6.7 \cdot 10^{-9}$. Input: *htf*. Unknown response: *WBN13*.

4.4.2 Subband NLMS

The discussion of misadjustment in the weight vector for the NLMS algorithm above has focussed on a fullband adaptive filter. In order to make use of the step size limiting factor, this needs to be generalized to the subband adaptive filtering case. This is quite simple, since a subband adaptive filter can be considered as M independent fullband adaptive filters. One need then only estimate the step size limiting factor, μ_m^{max}, for each band m. It would be somewhat impractical to estimate the M μ_m^{max} factors on an adaptive basis so a set of average estimates for these factors based on several different speech inputs will be used. It would also be possible to use long term averages of the speech power in each band in order to determine the step size limits but for simplicity, the work of this thesis uses fixed factors.

Table 4.1 tabulates the speech power by band for the three speech signals used in this thesis (see Appendix B), the average speech power by band, and the step size limiting factors μ_m^{max} for each of the bands as calculated from Equation 4.5. Note that Table 4.1 is for the case $M = 8$ and for weight vectors of length one. Since the step size of the LMS algorithm is inversely proportional to adaptive filter length, a corrective factor must be applied to each of the μ_m^{max} factors of Table 4.1 when these μ_m^{max} factors are applied to a subband adaptive filter which has adaptive filters of various lengths in each of its bands. Assuming that $\mu_m^{max}(1)$ is the maximum step size for band m as given by Table 4.1 (for an adaptive filter of length one), and the length of the adaptive filter in band m is L_m then the actual step size limit for band m will be given by:

$$\mu_m^{max}(L_m) = \frac{\mu_m^{max}(1)}{L_m}. \tag{4.6}$$

4.5 Adaptive Tap Assignment

Earlier, the expectation of being able to control the level of echo cancellation in each band of a subband adaptive filter by choosing an appropriate tap assignment was mentioned. In order to do this effectively, the weight vector lengths in each subband

Band	Speech signal power			Average power	μ_m^{max}
	m98	*m54*	*htf*		
0	$2.2 \cdot 10^6$	$2.5 \cdot 10^6$	$1.6 \cdot 10^6$	$2.1 \cdot 10^6$	$2.0 \cdot 10^{-6}$
1	$8.5 \cdot 10^5$	$7.7 \cdot 10^5$	$1.4 \cdot 10^6$	$1.0 \cdot 10^6$	$2.0 \cdot 10^{-6}$
2	$2.1 \cdot 10^4$	$5.3 \cdot 10^4$	$2.9 \cdot 10^5$	$1.2 \cdot 10^5$	$2.0 \cdot 10^{-5}$
3	$6.3 \cdot 10^4$	$2.5 \cdot 10^4$	$5.5 \cdot 10^4$	$4.8 \cdot 10^4$	$2.0 \cdot 10^{-4}$
4	$3.6 \cdot 10^4$	$1.2 \cdot 10^4$	$3.7 \cdot 10^4$	$2.8 \cdot 10^4$	$2.0 \cdot 10^{-4}$
5	$1.5 \cdot 10^4$	$7.7 \cdot 10^3$	$1.6 \cdot 10^4$	$1.3 \cdot 10^4$	$2.0 \cdot 10^{-4}$
6	$9.1 \cdot 10^3$	$9.3 \cdot 10^3$	$8.8 \cdot 10^3$	$9.1 \cdot 10^3$	$2.0 \cdot 10^{-4}$
7	$2.1 \cdot 10^4$	$8.6 \cdot 10^3$	$6.9 \cdot 10^3$	$1.2 \cdot 10^4$	$2.0 \cdot 10^{-4}$

Table 4.1: Average speech power by band and corresponding factors μ_m^{max} for $L = 1$.

should be adaptively assigned. This will allow the algorithm to work well for different inputs and different room responses. Two adaptive tap assignment algorithms will be presented in this section. The first is an adaptive tap assignment algorithm proposed in [57] and subsequently modified in [58]. The second is an algorithm developed as part of the work of this thesis. Both algorithms seek to improve upon the echo cancellation obtained by a fixed tap assignment subband adaptive filter. A fixed tap assignment adaptive filter is of course one which assigns an equal number of taps to the adaptive filter of each subband. Both of the adaptive tap assignment algorithms achieve better echo cancellation by adaptively assigning taps so as to equalize the error power amongst all of the bands of the subband adaptive filter. The two adaptive tap assignment algorithms use different algorithms to distribute the taps, with the algorithm developed for this thesis being more suitable for the application of a psychoacoustic weighting criterion. A psychoacoustic weighting criterion will be applied to the adaptive tap assignment algorithm in Section 4.6.

4.5.1 Sugiyama Algorithm

The Sugiyama algorithm is described in [58]. This algorithm is designed to equalize the error levels in each of the subbands, corresponding to a flat echo spectrum. In the case of a typical speech spectrum and a typical room spectrum (see Appendices A and B), the desired response exhibits a predominantly lowpass frequency characteristic. Using the algorithm presented in [58], most of the available taps will be concentrated in the lowest frequency subbands. The simulation examples in [58] concern themselves only with a two channel subband adaptive filter although the algorithm is presented for the general case of an M channel subband adaptive filter. Results using an 8 band version of the algorithm from [58] will be presented in Subsection 4.8.1. Note that in common with the work here, the authors of [58] have chosen to ignore the issue of decimation rates and do not apply any decimation to their filter bank.

Full details of the Sugiyama algorithm are presented in [58] but a short summary will be presented here. The length of the adaptive filter in a given band m is denoted $L_m(n)$. The dependence on sample n indicates the adaptive nature of these filter lengths. Adaptive filter lengths are updated every S iterations. The reallocation of taps consists of subtracting $R(n)$ taps from every subband and redistributing these taps according to:

$$L_m(n+1) = L_m(n) - R(n) + M \cdot R(n) \cdot \Phi_m(n) \tag{4.7}$$

for each of the M subbands. In the above equation, $\Phi_m(n)$ is a measure of the error in subband m relative to the total error of the subband adaptive filter. This error measure is given by:

$$\Phi_m(n) = \mathrm{INT}\left[\frac{\sum_{p=n-S+1}^{n} P_m(p) \cdot P_m^{tail}(p)}{\sum_{m=0}^{M-1}\sum_{p=n-S+1}^{n} P_m(p) \cdot P_m^{tail}(p)}\right] \tag{4.8}$$

where

$$P_m(n) = \sum_{i=0}^{L_m(n)-1} X_m^2(n-i) \tag{4.9}$$

and

$$P_m^{tail}(n) = \sum_{i=L_m(n)-P}^{L_m(n)-1} W_m^2(i). \tag{4.10}$$

The notation INT indicates that each of the M error measures $\Phi_m(n)$ must be rounded to an integer value. The factor $P_m(n)$ is the NLMS power estimate of the input signal in band m at sampling instant n and $P_m^{tail}(n)$ is the power in the last P coefficients of the weight vector of band m. $P_m^{tail}(n)$ is an estimate of the tail power of the weight vector and is used as an indication of the length of the room impulse response in band m. $R(n)$ is a function of n since the value of R is variable. $R(n)$ is incremented by one if band m has the largest $\Phi_m(n)$ for more than r consecutive tap reassignments, otherwise $R(n)$ is decremented by one to a minimum of one.

4.5.2 Error Equalizing Algorithm

The adaptive tap assignment algorithm that will be used for the work of this thesis is presented now. In the derivation of this algorithm, the same labels for variables as in the Sugiyama algorithm above will be used when such variables are identical in function. The tap assignment is recalculated every S iterations, as above. In this case, an error measure for each band is defined as

$$P_m^E(n) = \sum_{i=n-S+1}^{n} E_m^2(i) \tag{4.11}$$

where $E_m(n)$ is the error output of the adaptive filter in band m at sampling instant n. This error measure is essentially a measure of the average error in each subband over the past S iterations. At tap reassignment time, these error measures are sorted to produce a vector $\mathbf{V}_E(n)$ which contains the largest of $P_m^E(n)$, for $1 \leq m \leq M-1$, as its first element and the smallest of $P_m^E(n)$, for $1 \leq m \leq M-1$, as element $M-1$.

$$\mathbf{V}_E(n) = [\ (P_m^E(n))^{max} \ \cdots \ (P_m^E(n))^{min}\ ] \tag{4.12}$$

A vector $\mathbf{V}_M(n)$ corresponding to $\mathbf{V}_E(n)$ is constructed as well. The elements of $\mathbf{V}_M(n)$ are the numbers of the subbands which correspond to the entries in $\mathbf{V}_E(n)$. Thus, the first element of $\mathbf{V}_M(n)$ is the m for which $P_m^E(n)$ is greater than all other $P_m^E(n)$. The algorithm now consists of simply adding R taps to the subband referred to by the first entry in $\mathbf{V}_M(n)$ and subtracting R taps from the subband referred to by the last

entry in $\mathbf{V}_M(n)$. If the subband referred to by the last entry in $\mathbf{V}_M(n)$ already has the minimum number of taps possible then the subband referred to by the second last entry in $\mathbf{V}_M(n)$ has R taps subtracted from its length. To summarize this algorithm: at each tap reassignment time the band with the highest error accumulated over the past S iterations is accorded an additional R taps and the band with the least error accumulated over the past S iterations loses R taps. The weight vector lengths in all of the other bands remain unchanged.

A few general comments about the two adaptive tap assignment algorithms are in order. The Sugiyama algorithm requires each adaptive filter to be at least P (where P was the length of the tail) coefficients long in order for the algorithm to be able to calculate the required tail power estimate $P_m^{tail}(n)$. The error equalizing algorithm (EEQ) presented above does not have such a restriction and it is possible for an adaptive filter to have a zero length if there is little error in that band. Nevertheless, a minimum number of taps per band parameter will shortly be added to the EEQ algorithm. The error equalizing algorithm adjusts the weight vector lengths in only two bands at each tap reassignment whereas the Sugiyama algorithm changes the weight vector length in each and every band. Though this approach would slow down the convergence of the weight vector lengths of the EEQ algorithm, it will be seen that it also has a smoothing effect on the weight vector lengths; in short, a desirable effect. The error equalizing algorithm deals directly with the error in each subband and thus, if one wishes to weight the errors of the subbands relative to each other, weighting factors may be applied directly to errors. The Sugiyama algorithm has a more complicated error measure and weighting factors applied directly to the error measure may not produce the desired results. Comments on the specific choice of parameters are deferred until the simulation results are presented.

4.6 Psychoacoustic Criteria

The presentation of the adaptive tap assignment algorithms above has been from a strictly objective point of view. The given algorithms were designed to equalize the error power throughout the eight bands of the subband adaptive filter. In Section 2.10, some ideas on how the performance of an echo canceller might be improved if the echo spectrum were shaped to fit some hearing curve or masking curve were presented. This is the real reason for developing an adaptive tap assignment algorithm. Such an algorithm can be used to shape the echo spectrum so that the echo is as inaudible as possible. Hearing curves such as the absolute hearing threshold curves indicate that a flat echo spectrum may not be the most efficient shape to give the echo. In fact the absolute hearing threshold curve for the human ear indicates lower sensitivity to sounds with frequencies below 1 kHz and the highest sensitivity to sounds with frequencies in the range of 2 kHz to 4 kHz. This leads one to suspect that one should try to shape the echo spectrum similarly to the shape of the absolute hearing threshold curve for the human ear. In this way one will be making the best use of the available adaptive filter taps.

The EEQ algorithm presented above needs some minor modifications in order to meet the perceptual weighting needs. The algorithm, as it was presented, merely tries to give the echo spectrum a flat shape. It is desired to give the echo a shape more in accordance with the lowest curve of Figure 4.9, which is a curve of the absolute hearing threshold of the human ear. The upper curves of Figure 4.9 represent masking threshold curves and are not considered in this thesis. To do this, a piece-wise estimate of the absolute hearing threshold curve is made. Since the subband adaptive filtering structure of this thesis has eight bands, there are eight piece-wise segments with which to approximate the absolute hearing threshold curve. With these piece-wise estimates of the absolute hearing threshold curve an error-weighting vector is constructed. This vector has eight elements: one for every subband. The frequency range of interest here is from 0 Hz to 4 kHz. The piece-wise estimate of the absolute hearing threshold curve and the corresponding error weighting factors are given in Table 4.2.

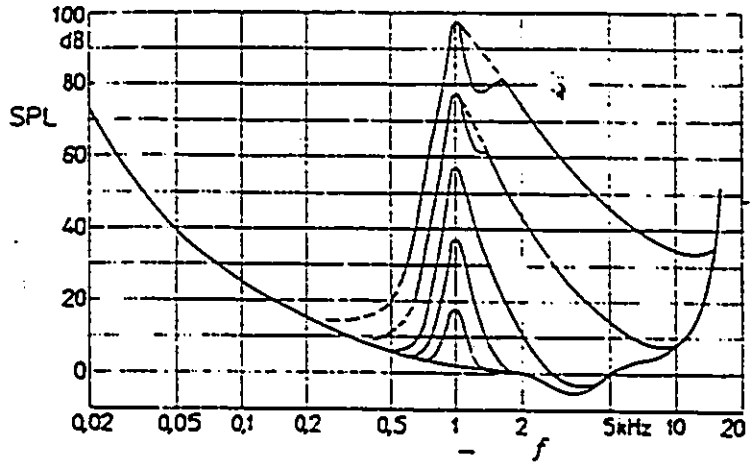

Figure 4.9: Absolute hearing threshold (lowest curve) versus frequency for the human ear. (Figure 1 of [72], originally from [78].)

Frequency (kHz)	Curve estimate(dB)	Error weight (dB)	Error weight
0 - 0.5	6	-6	0.25
0.5 - 1.0	4	-4	0.4
1.0 - 1.5	2	-2	0.63
1.5 - 2.0	0	0	1
2.0 - 2.5	0	0	1
2.5 - 3.0	-2	2	1.6
3.0 - 3.5	-5	5	3.2
3.5 - 4.0	-5	5	3.2

Table 4.2: Piece-wise estimate of the absolute hearing threshold curve and corresponding error weighting factors for the adaptive tap assignment algorithm.

The fourth column of Table 4.2 gives the linear error weighting factors. These are the factors which make up the error weighting vector as follows:

$$\mathbf{V}^{weight} = [\ 0.25 \quad 0.4 \quad 0.63 \quad 1 \quad 1 \quad 1.6 \quad 3.2 \quad 3.2\ ]. \tag{4.13}$$

In order to turn the adaptive tap assignment algorithm described above (EEQ) into an algorithm which tries to shape the echo spectrum to the same shape as the absolute hearing threshold curve, a new error measure $P_m^{E'}(n)$ is defined as follows:

$$P_m^{E'}(n) = v_m^{weight} \cdot \sum_{i=n-S+1}^{n} E_m^2(i) \tag{4.14}$$

where v_m^{weight} is coefficient m of the weighting vector $\mathbf{V}^{weight}$ in Equation 4.13. Vectors $\mathbf{V}'_E(n)$ and $\mathbf{V}'_M(n)$ are then constructed in a manner analogous to vectors $\mathbf{V}_E(n)$ and $\mathbf{V}_M(n)$ of the error equalizing algorithm. The effect of this weighting is simply to emphasize the higher frequencies and de-emphasize the lower frequencies in the echo. The end result should be an echo with a spectral profile similar to the absolute hearing threshold curve.

The numbers of Table 4.2 were deduced in the following manner. A reference point was selected on the absolute hearing threshold curve and considered as the 0 dB point. In Table 4.2, this corresponds to bands 3 and 4 or the frequency range 1.5 kHz to 2.5 kHz. The other frequency bands were then assigned magnitudes relative to the 0 dB point. Thus, the human ear is most sensitive to frequencies in the 3 kHz to 4 kHz range and least sensitive to lower frequencies. These relative numbers were then inverted to give the desired weighting factors. The numbers must of course be inverted since one wants to de-emphasize the lower frequency echoes and emphasize the higher frequency echoes.

The adaptive tap assignment algorithms presented above must be evaluated in two ways. Firstly, it must be determined if they are performing as they were designed to do. This means that these algorithms must achieve the desired spectral shape. In the case of the original error equalizing algorithm of Subsection 4.5.2, from here on EEQ algorithm, the echo spectrum should be nearly flat. In the case of the perceptually based algorithm,

from here on referred to as the weighted error equalizing algorithm (WEEQ), the echo spectrum should resemble the absolute hearing threshold curve. The algorithm from [58] shall be known as the Sugiyama algorithm (SUG). Secondly, it must be determined if the perceptually based weighting applied to the errors (in the case of the WEEQ algorithm) leads to less audible echoes when compared to the fixed tap assignment algorithm and to the original adaptive tap assignment algorithms.

4.7 Minimum Tap Assignment

An issue, mentioned previously, is whether or not to allow the weight vector length of any band m to go to zero. Alternatively, one could enforce a minimum number of taps per band $min_m(n)$ parameter which could as well be adaptive. Firstly, the reasons for which such a parameter might be necessary must be examined. To do this a performance measure known as the Total Impulse response Power to Tail response Power ratio" (TIP/TP), proposed by Knappe [7, 8], is useful. To explain it briefly one can say that the TIP/TP for a given room response gives the maximum achievable ERLE for a given fullband adaptive filter with L taps when the input signal is white noise. Figures A.3 and A.6 give the TIP/TP curves for the two room responses that have been considered:

The initial portions of the TIP/TP curves for the speakerphone-room responses *WBN13* and *mc3033* are steeper than the latter stages. Subbands of the adaptive filter which have lengths varying in value near zero may experience somewhat erratic cancellation performance.

One solution to this problem is to use a minimum number of taps per band parameter, denoted $min_m(n)$. This parameter is a function of n but it is not updated continuously. Rather, $min_m(n)$ will be updated every $N_{min} \cdot S$ iterations and will take on one of two values: 0 or 60 taps. The reasons for these two values of $min_m(n)$ are:

1. If a subband has very little error and $min_m(n)$ is not really required then it is undesirable to reserve any taps for this band since these taps could be assigned to

another band where they will do more good.

2. When $min_m(n)$ is required for a band, it is chosen according to $min_m(n) = 60$ because this will place the weight vector length at a significant distance from zero.

Every $N_{min} \cdot S$ iterations $min_m(n)$ must be updated for all M bands according to:

$$min_m(n) = \begin{cases} 60 & \text{if} \quad L_m^{max} \geq 65 \quad \text{and} \quad L_m^{min} < 35 \\ 0 & \text{if} \quad L_m^{max} \geq 65 \quad \text{and} \quad L_m^{min} \geq 35 \\ 0 & \text{if} \quad L_m^{max} < 65 \end{cases} \tag{4.15}$$

where

$$L_m^{max} = \text{maximum of} \quad L_m(i) \quad \text{for} \quad n - N_{min} \cdot S < i \leq n \tag{4.16}$$

$$L_m^{min} = \text{minimum of} \quad L_m(i) \quad \text{for} \quad n - N_{min} \cdot S < i \leq n. \tag{4.17}$$

To illustrate the effect of this minimum number of taps per band parameter, simulations are presented in Figures 4.10 and 4.11 for the WEEQ algorithm for runs with and without the minimum number of taps per band parameter. Figures 4.12 and 4.13 show the corresponding weight vector length curves to illustrate how the minimum number of taps per band parameter has constrained the weight vector lengths. Figure 4.12 shows a case where the minimum number of taps per band criteria aids the tap allocation in one band from oscillating too much. Figure 4.13 shows a case where the minimum number of taps per band criteria outlined above deems it unnecessary to allocate a minimum number of taps to the band in question and thus the band receives virtually no taps with or without the minimum number of taps per band strategy outlined above. The point of Figure 4.13 is to show that the minimum number of taps per band strategy is not rigid and will not allocate a minimum number of taps to a band if it is not necessary.

4.8 Objective Performance

An objective performance evaluation of the adaptive tap assignment algorithms presented above must be concerned with several factors:

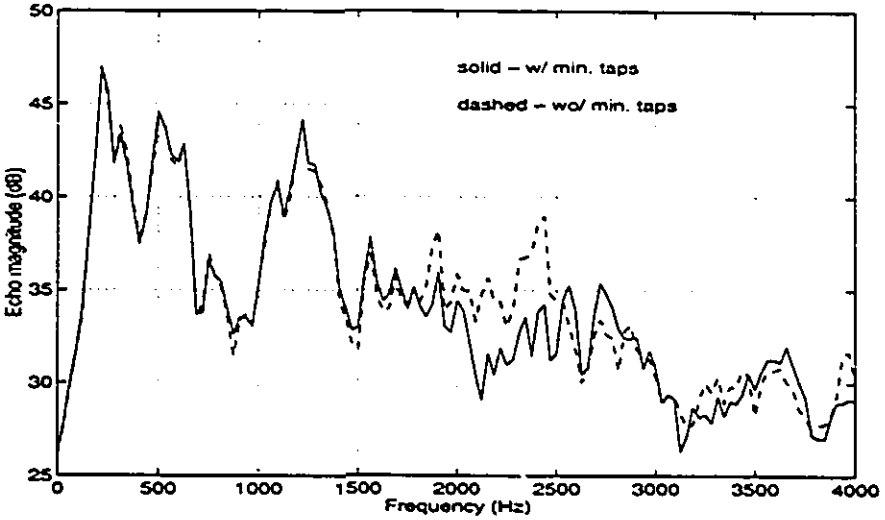

Figure 4.10: Error spectra for WEEQ with and without the minimum number of taps per band parameter. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (both), $S = 100$ (both), $1 \leq R \leq 5$ (both)).

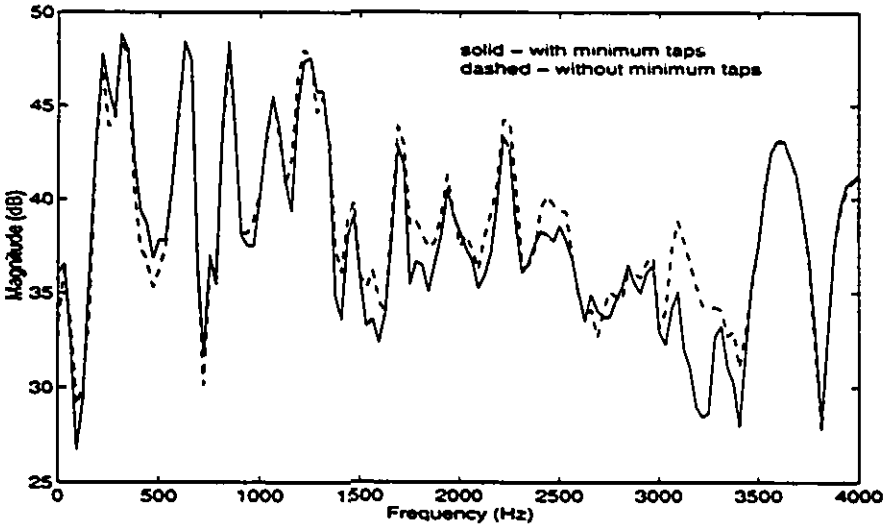

Figure 4.11: Error spectra for WEEQ with and without the minimum number of taps per band parameter. Unknown response: *WBN13*. Input: *htf*. Parameters: $M = 8$, $\mu = 1.0$ (both), $S = 100$ (both), $1 \leq R \leq 5$ (both).

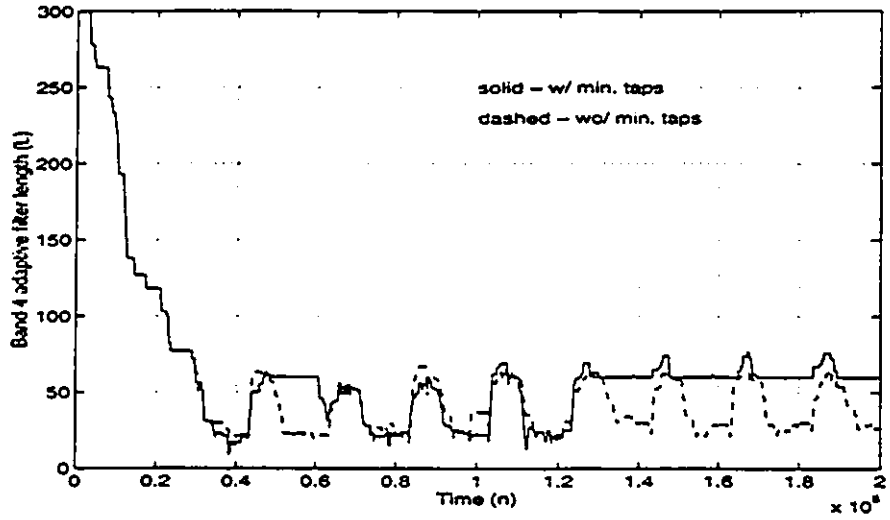

Figure 4.12: Weight vector lengths of the WEEQ algorithm with and without the minimum number of taps per band parameter versus time for band 4. Unknown response: *WBN13*. Input: *m54*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (both), $S = 100$ (both), $1 \leq R \leq 5$ (both).

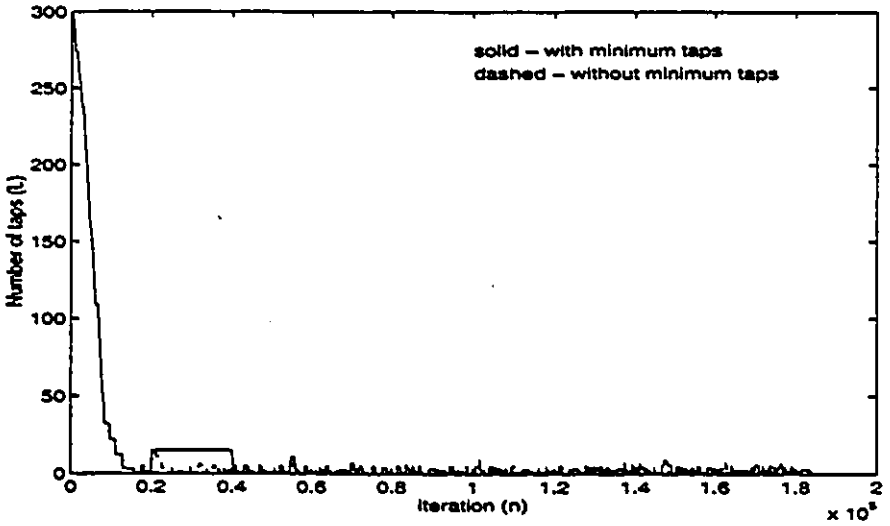

Figure 4.13: Weight vector lengths of the WEEQ algorithm with and without the minimum number of taps per band parameter versus time for band 7. Unknown response: *WBN13*. Input: *htf*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (both), $S = 100$ (both), $1 \leq R \leq 5$ (both).

1. The convergence time of the weight vector lengths in each of the bands. In other words; how long does it take the weight vector length in band m to reach a steady state value.

2. The overall convergence rate of the fullband error of the subband adaptive filter.

3. The echo spectrum after weight vector convergence. Weight vector convergence in this case means both convergence of the individual taps and convergence of the weight vector lengths. How well does the final echo spectrum match the desired spectral shape?

4. Choice of the adaptive tap assignment algorithm parameters and their effect on the performance measures discussed above. The effect of the parameters discussed above (S, R) must be examined.

Each of these criteria will be examined in the subsections that follow.

4.8.1 Sugiyama and Error Equalizing Algorithms

The Sugiyama algorithm and the error equalizing algorithm both seek to equalize the error amongst all subbands. This section compares the two algorithms on the following basis:

1. Echo spectra after weight vector convergence.

2. Speed of weight vector convergence.

3. Stability of weight vector lengths.

Steady-state Echo Spectra

Figures 4.14 through 4.16 show the error spectra after convergence for three different input and room response combinations. Figures 4.14 through 4.16 show the visual aspect of the "flattening" properties of the two adaptive tap assignment algorithms, but a more objective comparison would also be helpful. A measure of the flatness of an

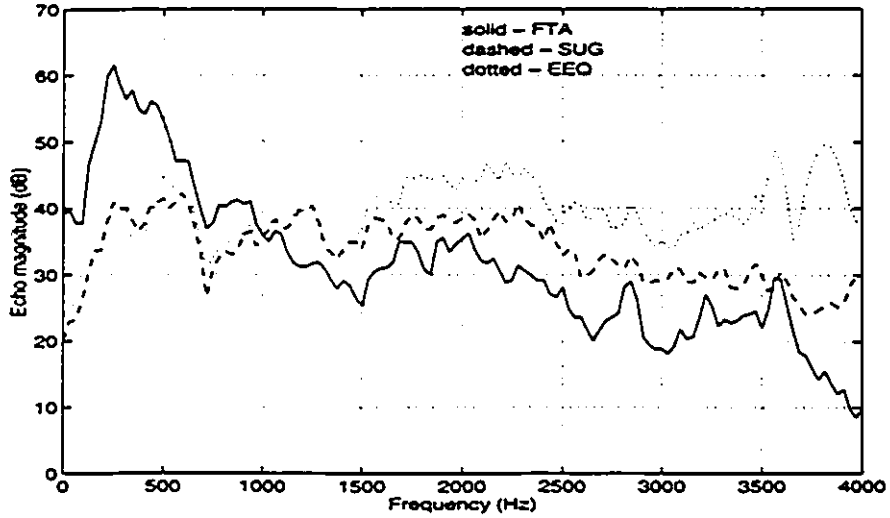

Figure 4.14: Error spectra for FTA, SUG and EEQ algorithms. Unknown response: *WBN13*. Input: *m98*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG/EEQ algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ), $P = 32$ (SUG).

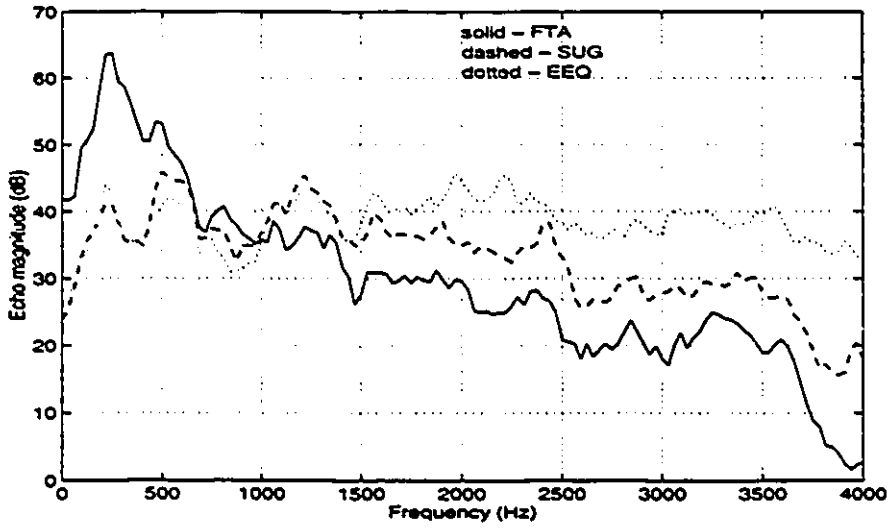

Figure 4.15: Error spectra for FTA, SUG and EEQ algorithms. Unknown response: *WBN13*. Input: *m54*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG/EEQ algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ), $P = 32$ (SUG).

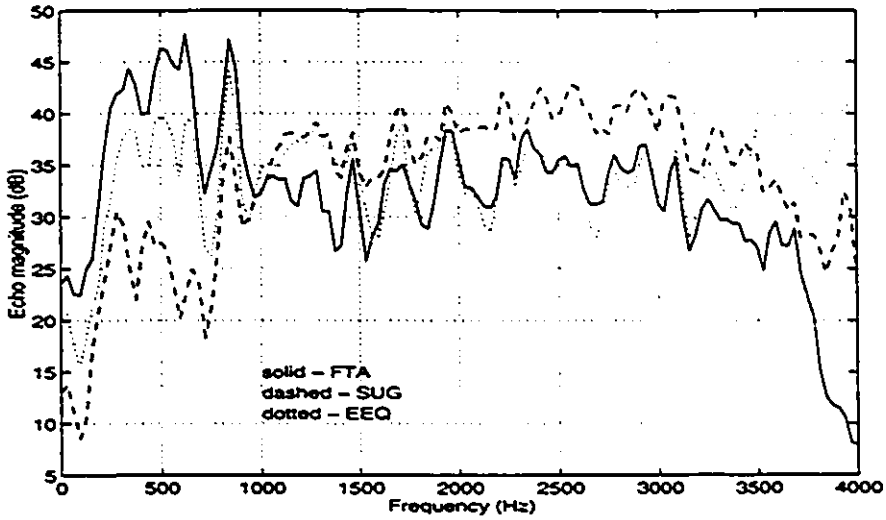

Figure 4.16: Error spectra for FTA, SUG and EEQ algorithms. Unknown response: *mc3033* Input: *htf.* Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG/EEQ algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ), $P = 32$ (SUG).

echo spectrum would be the second central moment of the averaged spectrum sequence (in other words, the variance of the spectrum). The spectra plotted in Figures 4.14 through 4.16 are actually averaged spectra computed using 256 point DFT's and the Welch method of averaging. This function is available as the `spectrum` function in `Matlab` (see Appendix B). The result is a 129 point sequence (covering the 0 Hz to 4 kHz frequency range in the case here) which can be plotted as the power spectrum magnitude. Computing the second central moment (variance) of the power spectrum sequence gives a measure of the flatness of the spectrum. Consider a power spectrum sequence given by $X(k)$ for $k = 0, 1, \cdots, (N/2-1)$ where N is the size of the DFT used in the calculation of the averaged spectrum. The second central moment of the power spectrum $X(k)$ is defined as:

$$m_X^2 = \frac{2}{N} \cdot \sum_{k=0}^{\frac{N}{2}-1} (X(k) - \mu_X)^2 \tag{4.18}$$

algorithm	Input and Room Response		
	m98-WBN13	*m54-WBN13*	*htf-mc3033*
	m_X^2	m_X^2	m_X^2
FTA	$3.34 \cdot 10^{10}$	$9.57 \cdot 10^{10}$	$9.68 \cdot 10^{7}$
SUG	$1.3 \cdot 10^{7}$	$5.8 \cdot 10^{7}$	$2.38 \cdot 10^{7}$
EEQ	$2.9 \cdot 10^{8}$	$5.06 \cdot 10^{7}$	$1.14 \cdot 10^{7}$

Table 4.3: Second central moments of the error spectra.

where μ_X is the mean of $X(k)$, defined as:

$$\mu_X = \frac{2}{N} \cdot \sum_{k=0}^{\frac{N}{2}-1} X(k). \tag{4.19}$$

Note that the symbols used here are somewhat arbitrary, and are not necessarily based on convention. Table 4.3 gives the second central moments (m_X^2) for the simulations of Figures 4.14 through 4.16. Table 4.3 indicates that the two adaptive tap assignment algorithms, in all cases, have flattened the echo spectrum with respect to the fixed tap assignment algorithm (FTA).

In summary, both algorithms flatten the echo spectrum considerably, relative to the fixed tap assignment algorithm. By flattening the echo spectrum, the maximum echo spectral components are lower in the adaptive tap assignment algorithms than in the fixed tap assignment algorithm.

Convergence Speed

The term ERLE was defined earlier in this thesis. The ERLE is a measure of the echo cancellation. When adaptive filters are trained with highly nonstationary signals such as speech, some sort of short-term averaging must be applied to the ERLE in order to make it a useful criterion for measuring the performance of the adaptive filter. The ERLE measure that will be used in this section to assess the convergence rate of the subband adaptive filters will be denoted ERLE(n) to indicate that it is computed at

each iteration n. The following equation gives the ERLE measure:

$$\text{ERLE}(n) = \frac{\sum_{i=n-N_{ERLE}+1}^{n} d^2(i - L_{AS} + 1)}{\sum_{i=n-N_{ERLE}+1}^{n} e^2(i)}. \tag{4.20}$$

N_{ERLE} is the length of the window over which the average ERLE is computed. The choice of N_{ERLE} represents a trade-off between being able to see the fluctuations of the ERLE and having a smooth ERLE curve. $(L_{AS} - 1)$ is the delay of the filter bank structure, as discussed in Chapter 2. The ERLE is defined as the ratio of the power of the desired response (ie. the echo with absolutely no cancellation) and the power of the echo. Note that in the subband adaptive filtering structure of Figure 2.9, $d(n)$ is not delayed at all while $e(n)$ sees the full delay $(L_{AS} - 1)$ of the filter bank. Thus, in order for $d(n)$ to represent the echo with no attenuation, it must be delayed by $L_{AS} - 1$. This is the reason for the different indices in Equation 4.20.

Figures 4.17 and 4.18 show plots of the ERLE measure for two of the simulations above. The factor N_{ERLE} was set equal to 1000 in order to give a decent amount of smoothing to the ERLE measure. The Sugiyama and EEQ algorithms initially converge to the same ERLE level as the fixed tap assignment algorithm, but then as the new tap distribution begins to take effect, the adaptive tap assignment algorithms converge to a new ERLE.

The most important thing to be recognized from Figures 4.17 and 4.18 is that the adaptive tap assignment algorithms are not losing anything, in terms of convergence, to the fixed tap assignment algorithm. Initial convergence is as quick for the adaptive tap assignment algorithms, and the adaptive tap assignment algorithms continue to converge to a better ERLE.

Weight Vector Length Convergence

Weight vector convergence, in the context discussed below, refers to the speed with which the weight vector lengths reach some sort of steady state value. The stability of the weight vector lengths refers to the level of oscillation of the weight vector length about its steady state average. Figures 4.19 to 4.22 show the weight vector lengths versus

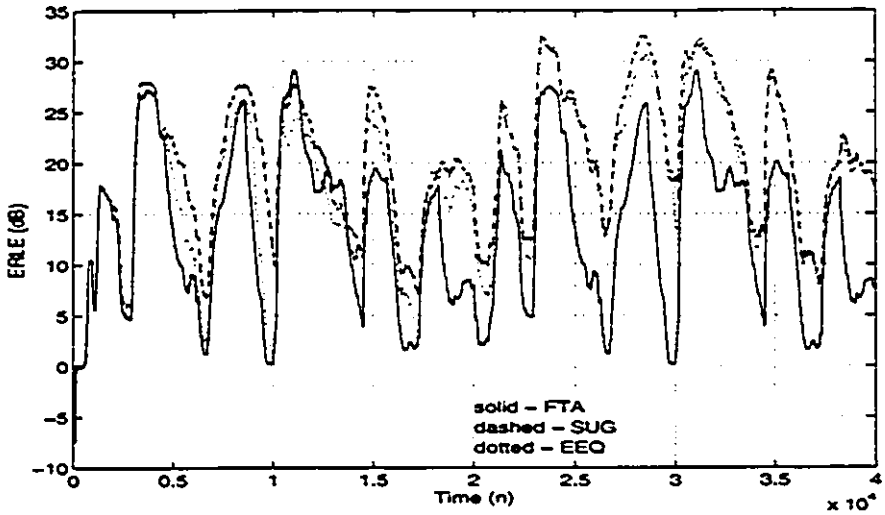

Figure 4.17: ERLE curves for FTA, SUG and EEQ algorithms.Unknown response: *WBN13*. Input: *m54*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG/EEQ algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ).

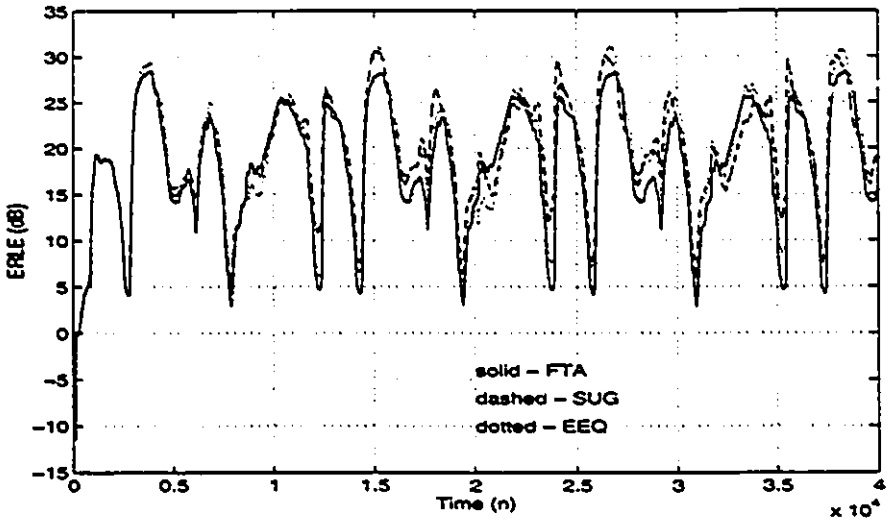

Figure 4.18: ERLE curves for FTA, SUG and EEQ algorithms. Unknown response: *mc3033*. Input: *htf*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG/EEQ algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ).

time for four bands of interest taken from the same simulations as used for the echo spectra in Figures 4.14 to 4.16. The weight vector lengths of the SUG algorithm have a faster initial convergence rate than those of the EEQ algorithm. What is somewhat surprising is the discrepancy in final weight vector lengths of the two algorithms for the cases in Figures 4.21 and 4.22. These weight vector length curves correspond to the simulation of Figure 4.16 where the SUG algorithm actually seems to be assigning too many taps to the lowest two subbands (as evidenced by Figure 4.21 and two few taps to higher bands (as evidenced by Figure 4.22) resulting in an echo which is not as flat as the EEQ echo of Figure 4.16.

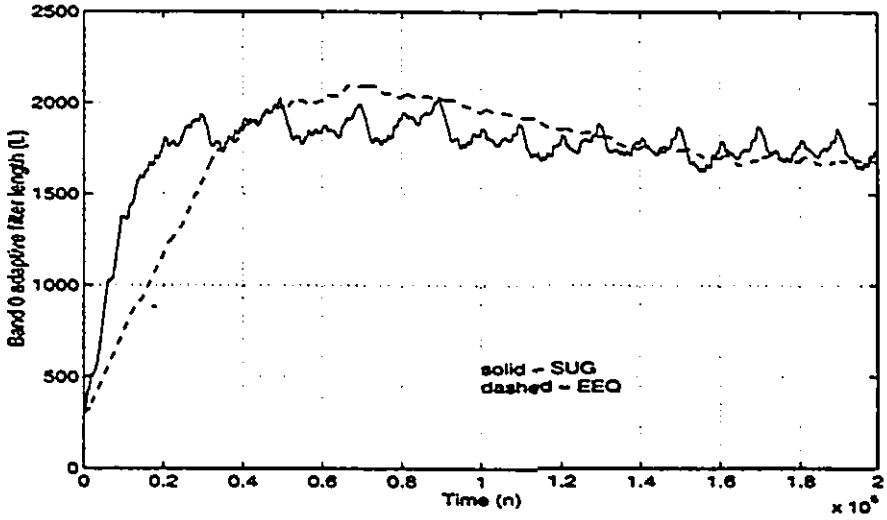

Figure 4.19: Weight vector lengths of the EEQ and SUG algorithms versus time for band 0. Unknown response: *WBN13*. Input: *m54*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (both algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ).

Effect of Parameters S and R

The effect of these parameters will be first considered for the EEQ algorithm. These parameters are considered in conjunction since they can both be used to achieve similar effects. Intuitively, one expects faster convergence of the weight vector lengths for

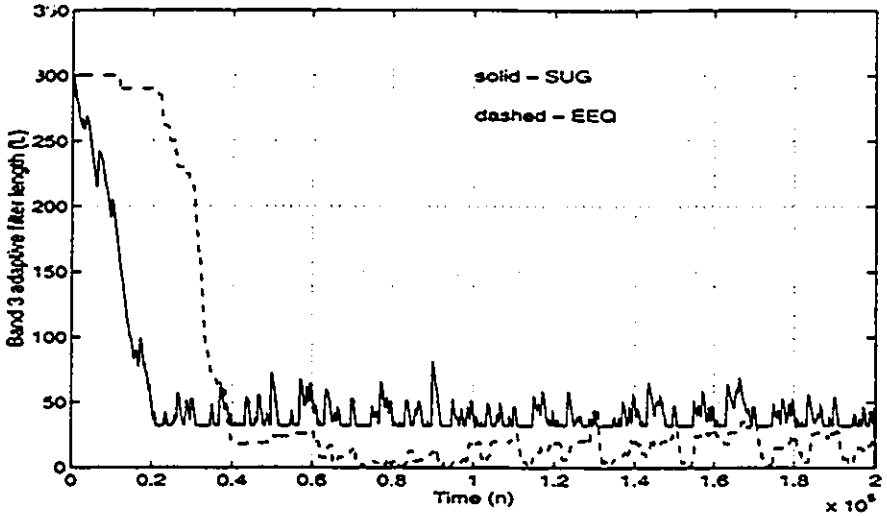

Figure 4.20: Weight vector lengths of the EEQ and SUG algorithms versus time for band 1. Unknown response: *WBN13*. Input: *m54*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (both algorithms), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (EEQ).

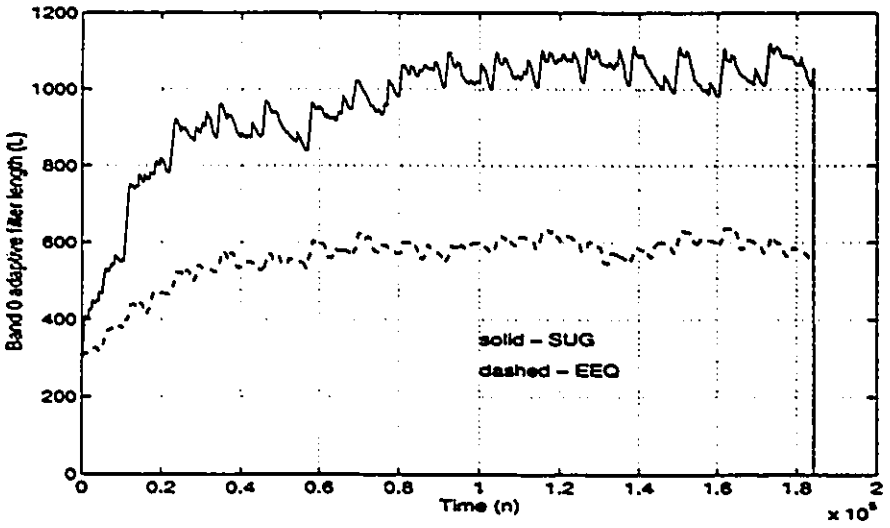

Figure 4.21: Weight vector lengths of the EEQ and SUG algorithms versus time for band 0. Unknown response: *mc3033*. Input: *htf*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (both algorithms), $S = 100$ (both algorithms), $1 \leq R \leq 3$ (SUG) and $1 \leq R \leq 5$ (EEQ).

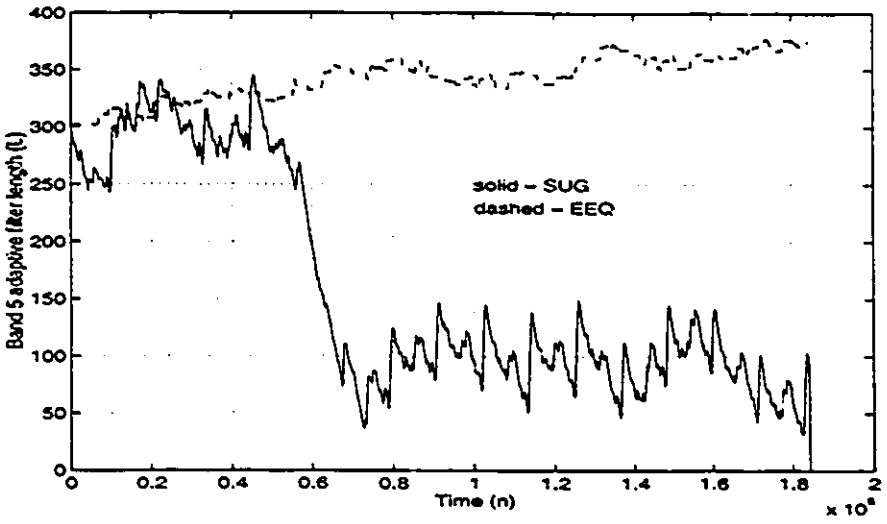

Figure 4.22: Weight vector lengths of the EEQ and SUG algorithms versus time for band 3. Unknown response: *mc3033*. Input: *htf*. Initial tap assignment: 300 taps per band. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (both algorithms), $1 \leq R \leq 3$ (SUG) and $1 \leq R \leq 5$ (EEQ).

smaller S and for larger R. It is R_{max} rather than R which is truly of interest since R is always allowed to vary within the range:

$$1 \leq R \leq R_{max}. \tag{4.21}$$

It can be intuitively expected that the weight vector lengths will be somewhat erratic if an R_{max} that is too large is used. To illustrate this, Figure 4.23 shows the weight vector length curve of band 0 for three different values of R_{max} with $S = 100$.

Figure 4.24 shows the weight vector length curve of band 0 for three different values of S with $R_{max} = 5$. One can see that the effect of the choice of R_{max} and S is very similar to the intuitive expectation outlined earlier. When the weight vector length tries to converge too quickly, either through a small choice of S or a large choice of R_{max}, there tends to be significant overshoot in the weight vector length for bands that are gaining a significant number of taps. Figures 4.23 and 4.24 provide the evidence of this. The choice of $S = 100$ and $R_{max} = 5$ provides a good compromise between weight vector length convergence and smoothness of the weight vector length (overshoot).

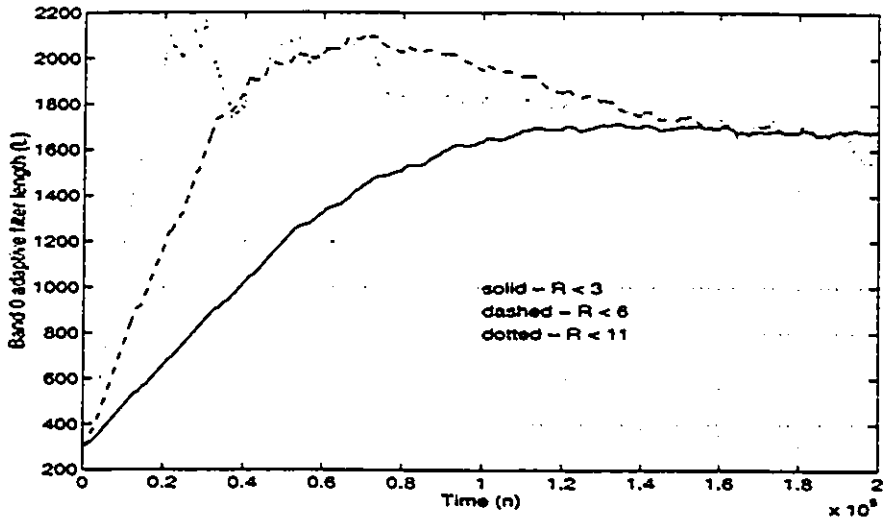

Figure 4.23: Weight vector lengths of the EEQ algorithm versus time for band 0. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$, $S = 100$, $R_{max} = 2, 5, 10$.

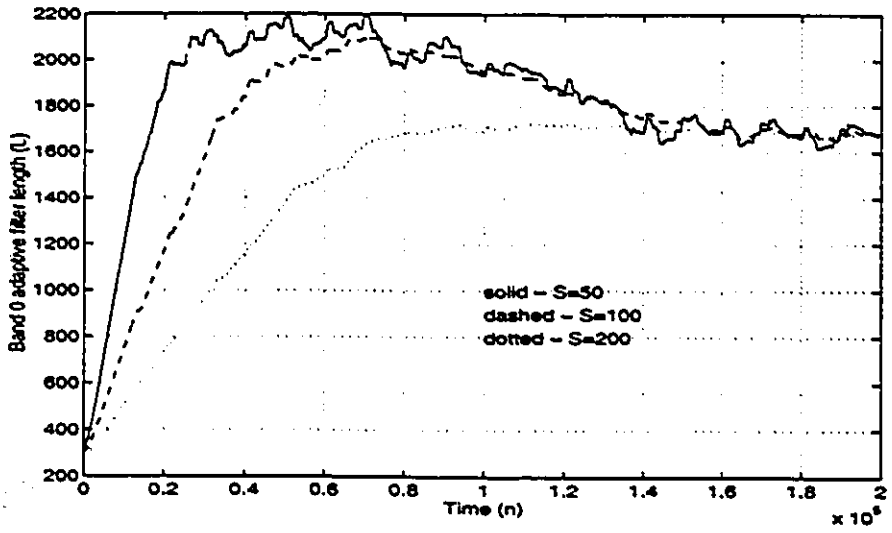

Figure 4.24: Weight vector lengths of the EEQ algorithm versus time for band 0. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$, $S = 50, 100, 200$, $1 \leq R \leq 5$.

The choice of these parameters is slightly different for the Sugiyama algorithm. While S functions identically for both algorithms, R_{max} is significantly different. In the EEQ algorithm, up to R_{max} taps are added or subtracted from two bands every S iterations. In the Sugiyama algorithm, up to $R_{max} \cdot M$ taps are available for redistribution every S iterations. Thus, it is possible for a band to receive up to $R_{max} \cdot S$ taps although no band will lose more than R_{max} taps at any given iteration. The result is that R_{max} cannot be set as high for the Sugiyama algorithm as it can for the EEQ algorithm. Figure 4.25 shows that choosing $R_{max} = 5$ leads to significant overshoot in the convergence of the weight vector length. The value of $R_{max} = 3$, used in the earlier simulations, represents a compromise between smooth convergence and fast convergence.

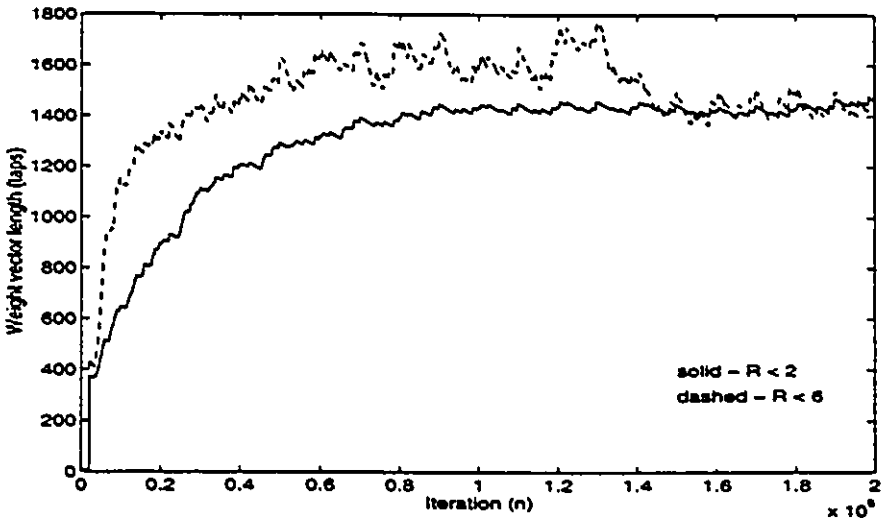

Figure 4.25: Weight vector lengths of the Sugiyama algorithm versus time for band 0. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$, $S = 100$, $R_{max} = 1, 5$.

Conclusion

The main point of this subsection has been to compare the EEQ algorithm and the Sugiyama algorithm. The two algorithms perform quite similarly. There are, of course, small differences in their performances; yet, nothing that could be construed to imply

that one algorithm is markedly superior to the other.

4.8.2 Natural Speech Results

Now, an evaluation of the psychoacoustically based WEEQ algorithm performance compared to the performance of the fixed tap assignment (FTA) algorithm and the Sugiyama algorithm is presented. The performance of the WEEQ algorithm is identical to the performance of the EEQ algorithm except for the weighting applied to the echo. Many of the performance measures discussed in the previous subsection; namely, the convergence speed, weight vector length and stability, the effect of the S and R parameters, and the algorithm complexity, do not need to be discussed again. Instead, this subsection will concentrate on the echo spectrum shaping obtained by the WEEQ algorithm. In this subsection, the concern is the performance of the adaptive filtering algorithms with natural, full bandwidth speech. This does not take into account any of the bandlimiting that a telephone channel may apply to the spectrum of the speech signal. Simulations in Subsection 4.8.4 will consider the influence of the telephone channel frequency response on the performance of the adaptive tap assignment algorithms.

Steady-state Echo Spectra

It will be shown that the WEEQ algorithm has shaped the echo spectrum in the desired manner. Figures 4.26 to 4.31 show echo spectra after convergence for six different input-room response combinations. The WEEQ algorithm usually finds itself in between the FTA and SUG algorithms as far as level of echo in a given frequency band goes. The use of the WEEQ algorithm results in an echo with more echo content at lower frequencies and less echo content at higher frequencies, in accordance with the shape of the absolute hearing threshold curve. Due to the minimum number of taps requirement, the Sugiyama algorithm will provide a minimum depth of cancellation in every band, even if that band contains little energy. The WEEQ algorithm has a minimum number of taps per band parameter as well, but this parameter is only applied to a band when it is found that the echo in that band is significant and is causing the WEEQ algorithm

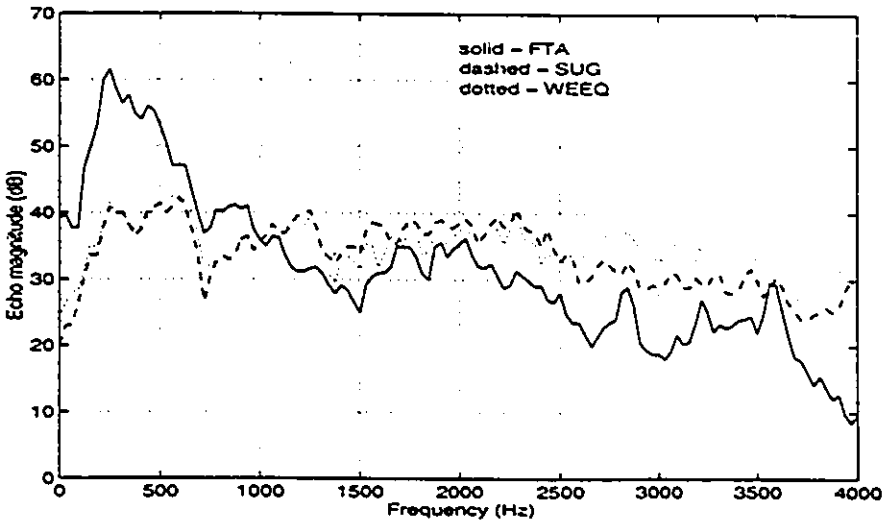

Figure 4.26: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *m98*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

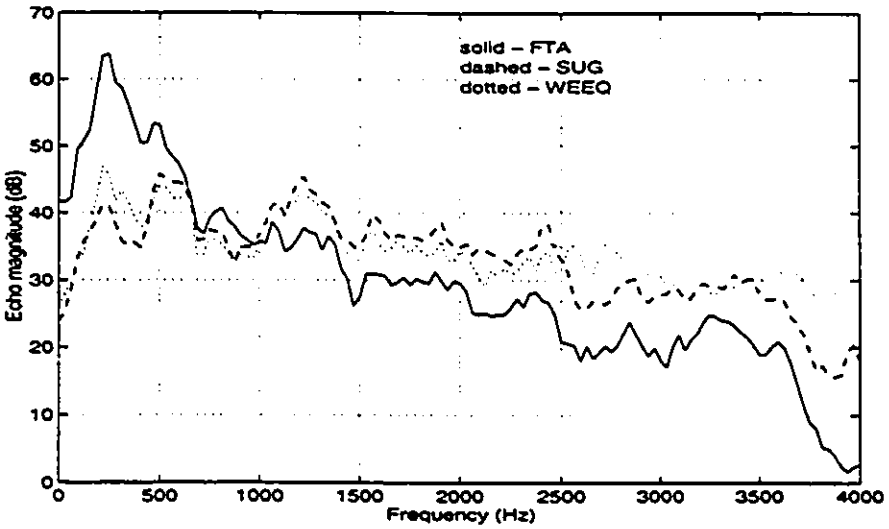

Figure 4.27: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

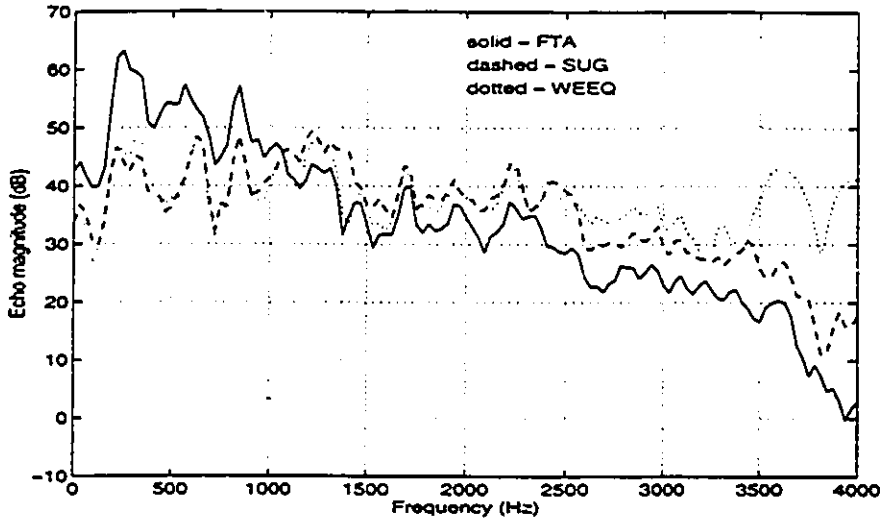

Figure 4.28: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *htf*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

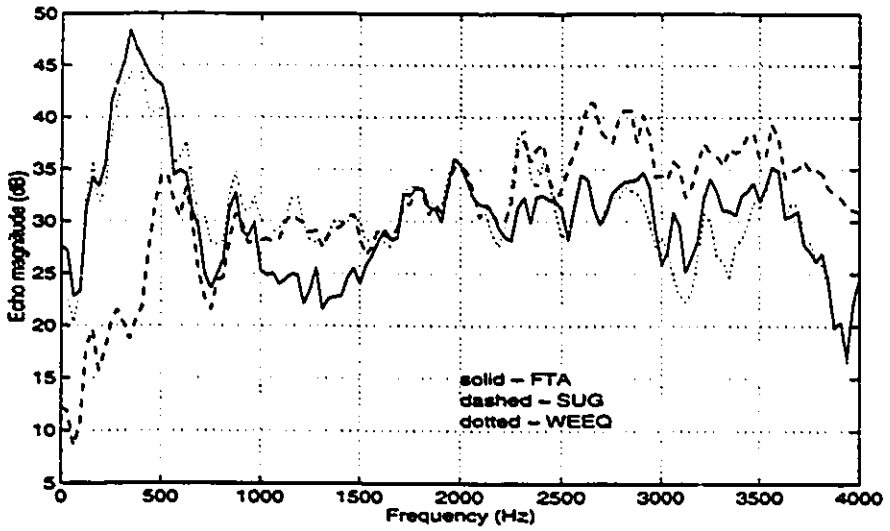

Figure 4.29: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *m98*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

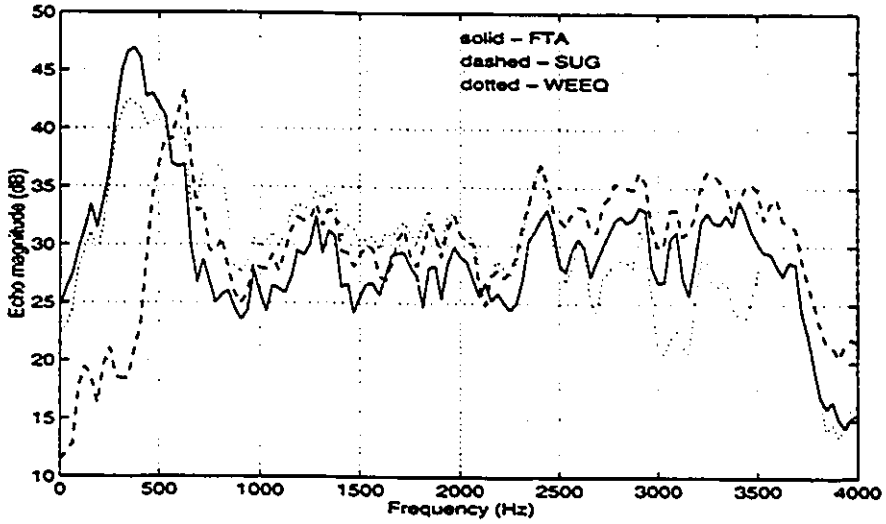

Figure 4.30: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

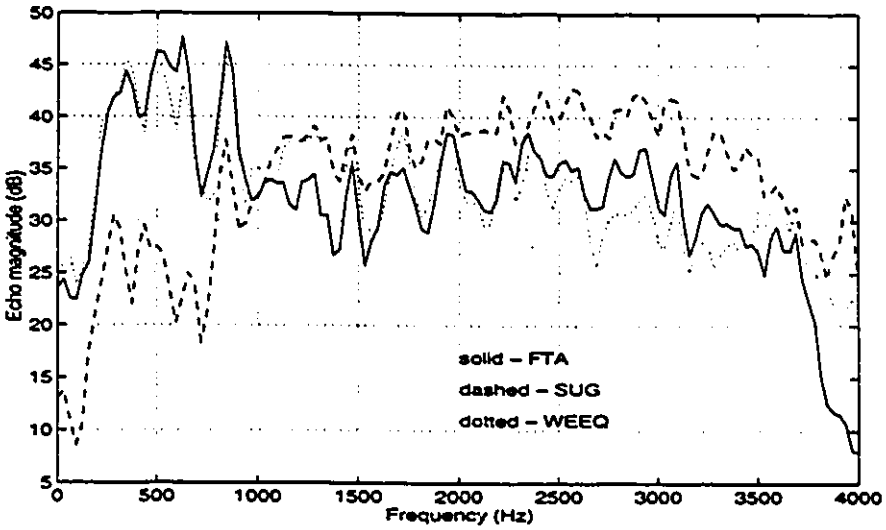

Figure 4.31: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *htf*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

to not achieve the desired level of cancellation in that band. The real test of the WEEQ algorithm can only be a subjective hearing test. From an objective point of view, the WEEQ echo usually has more error than the Sugiyama (or EEQ) algorithm. However, as mentioned previously, psychoacoustic principles imply that the sensitivity of the ear is a more important criterion than the absolute ERLE obtained.

In Figures 4.26 to 4.29, the Sugiyama and WEEQ algorithms have shaped the echo spectra in very similar manners. Given that the desired echo spectral shape of the Sugiyama algorithm is flat, it is the Sugiyama algorithm which is deviating more from its intended performance. The WEEQ algorithm appears to be achieving a spectral shape in line with its criteria. Table 4.4 in Subsection 4.8.5 shows that for these three cases, almost all of the available taps are being assigned to the first two bands, both for the Sugiyama algorithm and for the WEEQ algorithm. For the Sugiyama algorithm in particular, the rest of the bands receive an average number of taps very close to the absolute minimum number of taps allowed per band. This indicates that the Sugiyama algorithm is doing everything it can to reduce the echo in the first two bands at the expense of higher echo everywhere else. However, Figures 4.26 to 4.29 show that the echo level is still higher at low frequencies than at high frequencies. Thus, if the Sugiyama algorithm had even more taps to work with, it would allocate these to the low bands in an effort to achieve a flat echo spectrum. Thus the reason that the Sugiyama and the WEEQ algorithm have achieved the same echo spectral shaping for these cases is due to the fact that there are simply not enough available taps for both algorithms to achieve their desired shape. With more available taps the Sugiyama algorithm would flatten the echo spectrum even more while the WEEQ algorithm would maintain the psychoacoustically-derived emphasis on better echo cancellation in the higher frequency bands.

Convergence Speed

As mentioned previously, the performance of the WEEQ algorithm in terms of convergence speed is identical to that of the EEQ algorithm, save the weighting applied to

the echo. This implies that the WEEQ algorithm will reach its steady-state ERLE as quickly as the EEQ algorithm, but that this steady-state ERLE will likely be somewhat worse (though no not necessarily so) than that of the EEQ algorithm. Figure 4.32 shows the ERLE measure of the WEEQ algorithm and the EEQ algorithm for the simulation of Figure 4.27. In Figure 4.27 it can be seen that the Sugiyama and WEEQ algorithms have achieved virtually the same echo spectral shape. It is thus not surprising that Figure 4.32 shows virtually identical ERLE performance for the WEEQ and EEQ algorithms. Remember that the EEQ and Sugiyama algorithms perform very similarly (the convergence rate of the Sugiyama algorithm was found to be almost identical to that of the EEQ algorithm in Subsection 4.8.1). In cases such as Figure 4.30 and Figure 4.31 where the WEEQ algorithm achieves an echo spectrum with more low frequency echo (and thus more echo on the whole) than the Sugiyama (or EEQ) algorithm, one can expect the WEEQ ERLE performance to be slightly inferior to that of the Sugiyama or EEQ algorithms.

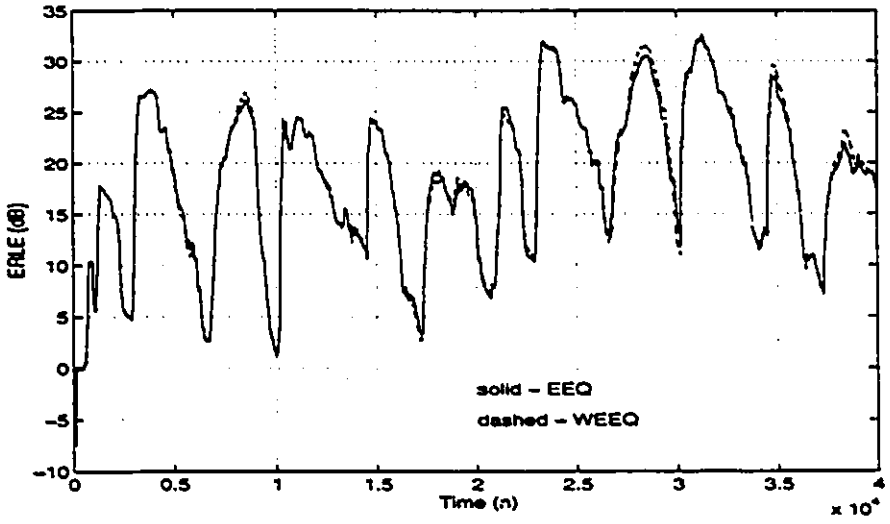

Figure 4.32: ERLE curve for the EEQ and WEEQ algorithms. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (EEQ, WEEQ), $1 \leq R \leq 5$ (EEQ, WEEQ).

Conclusion

The weighting of the error applied by the WEEQ algorithm results directly in a shaped echo spectrum. The WEEQ algorithm allows more low frequency echo and less high frequency echo than the EEQ algorithm. This weighting is in accordance with the absolute hearing threshold curve. The test of the performance of the psychoacoustically-based WEEQ algorithm is necessarily a subjective one and this test is the subject of Section 4.10.

4.8.3 Telephone Channel Bandlimiting

In order to make the simulations as realistic as possible, the effect of transmission over a telephone channel on the spectra of the input signals must be considered. Telephone networks use a sampling rate of 8 kHz. Thus, there is a maximum bandwidth available of 4 kHz. Due to practical considerations, such as immunity from low frequency noise, separation of DC and AC function in telephone electronics and the use of practical, low-cost analogue circuits, less than the full 4 kHz bandwidth is transmitted. The characteristics of the telephone channel can be approximated by a filter. The filter used to approximate the telephone channel response in this thesis is a high quality bandpass filter with flat passband from 300 to 3400 Hz (the effective bandwidth of telephone transmission). High attenuation is applied to out of band frequency components by this filter, as shown by its frequency response in Figure 4.33.

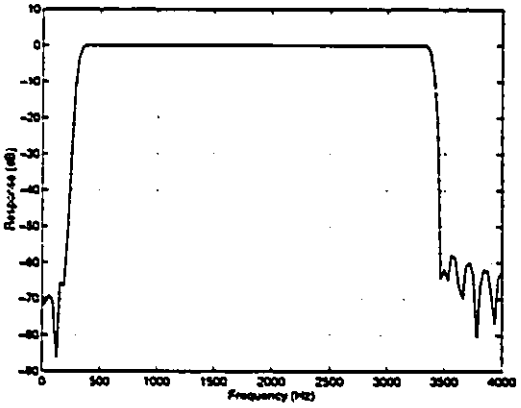

Figure 4.33: Telephone channel bandlimiting filter.

To illustrate the effect that the telephone channel has on the frequency response of speech signals, Figure 4.34 shows the spectrum of speech signal *m98* before and after being filtered by the bandlimiting filter. A comment about the spectral estimation

method used here is now in order. One will certainly notice that the bandpass filter of Figure 4.33 has a DC attenuation of about 70 dB. However, Figure 4.34 does not show a corresponding drop in the DC component of the bandpass-filtered speech signal. The reason is due to the spectrum function's spectral estimation method. As mentioned previously, the spectrum function averages together enough shifted 256 pt. DFT's to cover the entire speech sequence. Each of the DFT's is first windowed by a Hanning window. The effect of this is to introduce some error, especially at very low frequencies, because not all components of the speech sequence are weighted equally due to the windowing. The author did verify, using one large DFT, that the actual DC content of the bandlimited speech sequence does indeed concur with the DC attenuation shown in Figure 4.33.

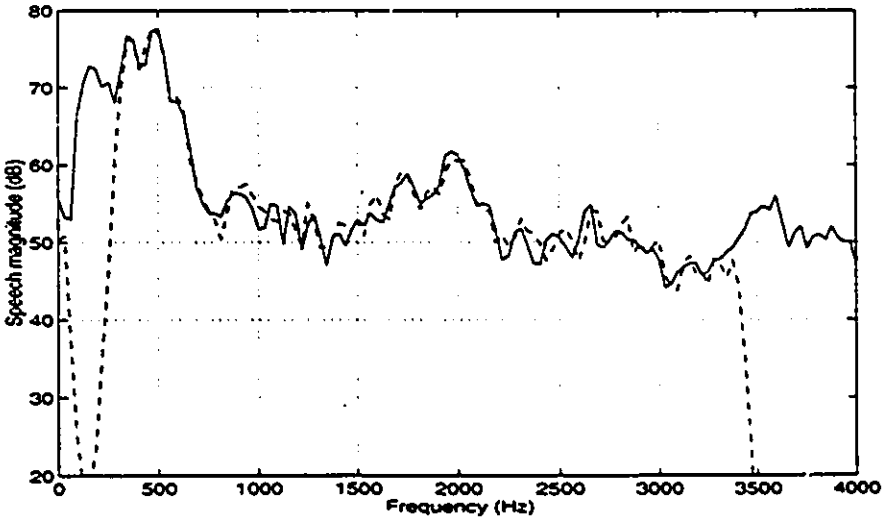

Figure 4.34: Bandlimiting of the speech spectrum.

In the simulations that follow, the speech signals of Appendix B will be filtered with the bandlimiting filter of Figure 4.33 before being used in the simulations. This should ensure that the input signals used are a good approximation to actual speech signals encountered in telephone transmissions. Figure 4.35 shows a simple model of a telephone communication link between two users on hands-free communication sets. Note that the input speech to the adaptive filter ($x_{NEAR}(n)$) is the original speech ($x_{FAR}(n)$) filtered

by the far end telephone set's transmit response, local loop responses, office switch responses and the near end telephone set's receive response, and the echo which reaches the far-end user's ear ($e_{FAR}(n)$) is the error of the adaptive filter ($e_{NEAR}(n)$) filtered by the same elements as ($x_{FAR}(n)$), except in the reverse order. Figure 4.35 indicates that the frequency shaping applied to speech signals by the telephone network occurs in many stages. Frequency shaping occurs at the user's telephone (TX and RX responses of the set), between the phone set and the central office (CO), at the central office and at any links and switches between the central offices of the near and far-end users (if any).

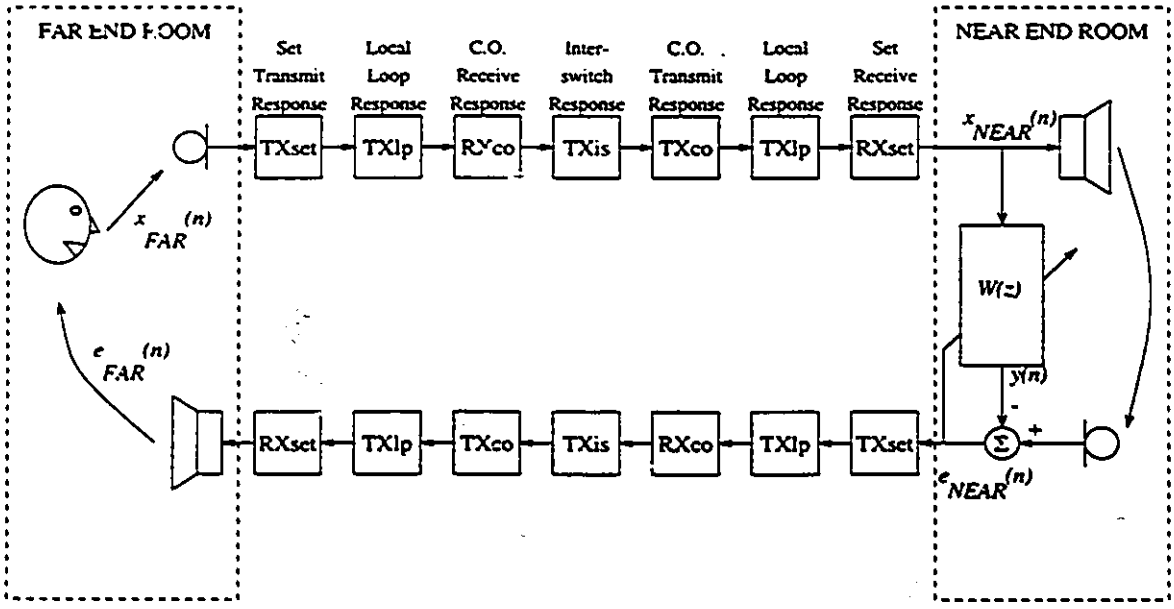

Figure 4.35: Filtering of signals by the telephone network.

4.8.4 Bandlimited Speech Results

The simulations presented in Subsection 4.8.2 are presented again, but with bandlimited speech. As discussed in Subsection 4.8.3, the speech signals in a hands-free communication link are filtered by the telephone channel. This bandlimits the speech, altering the spectrum of the transmitted signals. This will have an effect on the echo spectra of the various subband adaptive filtering algorithms, and this effect is studied

in this subsection. It is the echo signal ($e_{FAR}(n)$) which reaches the far-end user's ear which must be evaluated subjectively. Note that the echo signal generated by the adaptive filtering simulations is signal $e_{NEAR}(n)$. In order to simulate the transmission of $e_{NEAR}(n)$ to the far end user, where it becomes $e_{FAR}(n)$, $e_{NEAR}(n)$ must be filtered by the appropriate telephone channel responses. However, since this in effect would just further attenuate the already extremely low level out of band signal components while not affecting in band components significantly, this step is omitted in the following work.

Steady State Echo Spectra

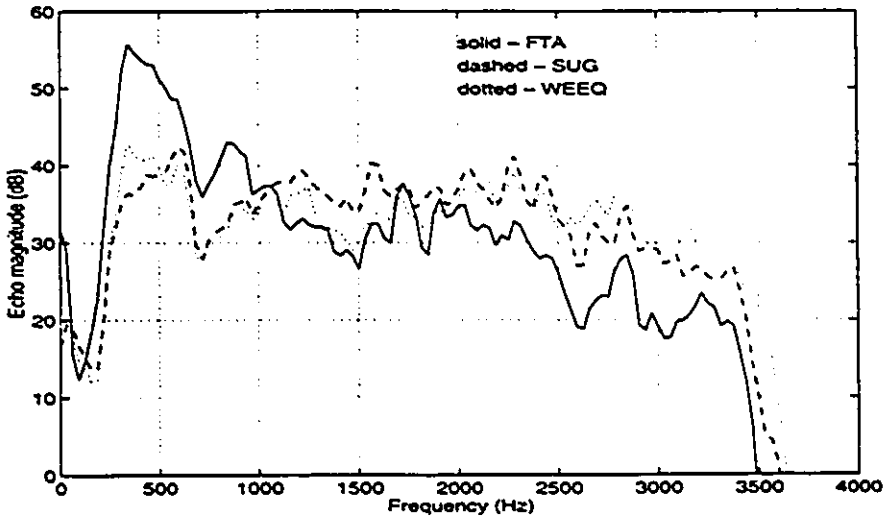

Figure 4.36: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *m98*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

Figures 4.36 to 4.41 show bandlimited echo spectra after convergence for six different input-room response combinations. The results are different from Subsection 4.8.2 due to the bandlimiting effect input speech. In particular, the fixed tap assignment echoes are no longer as dominated by low frequency echo (echo below 1 kHz) for speakerphone-room response *WBN13*. In Subsection 4.8.2, the adaptive tap assignment algorithms reduced the low frequency echo considerably for *WBN13* while allowing echo everywhere

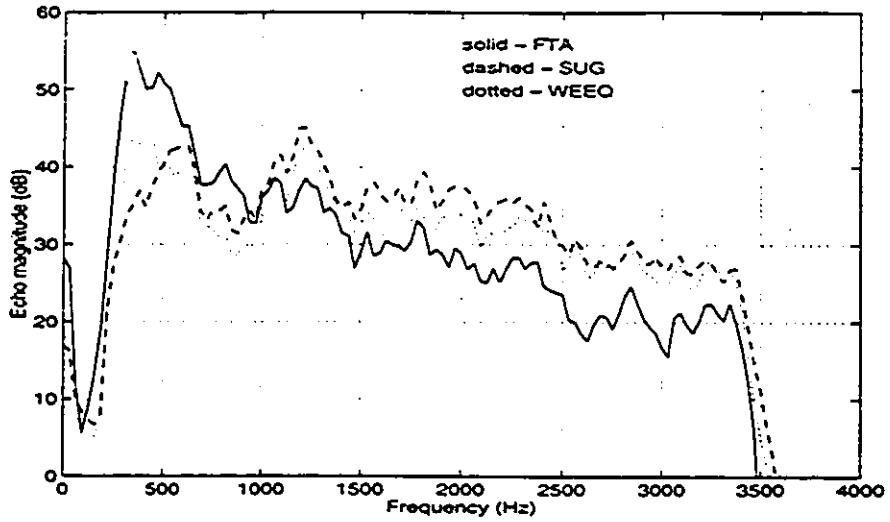

Figure 4.37: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

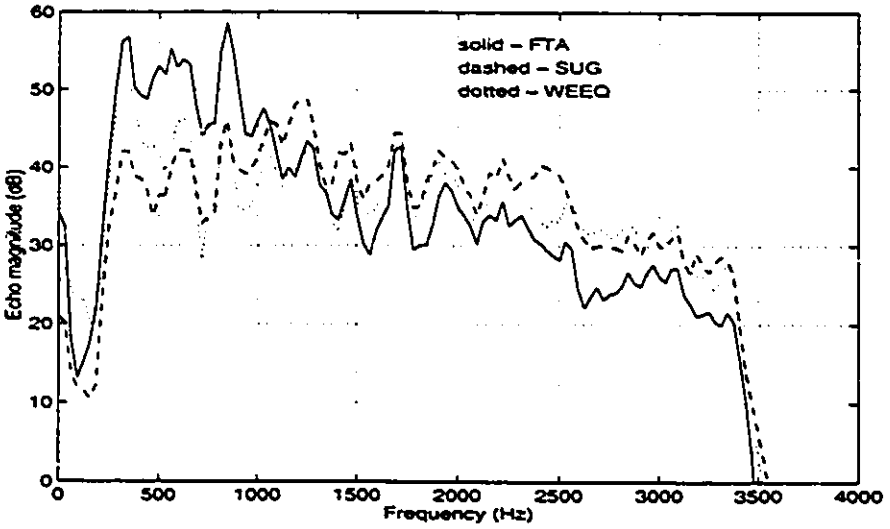

Figure 4.38: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *WBN13*. Input: *htf*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

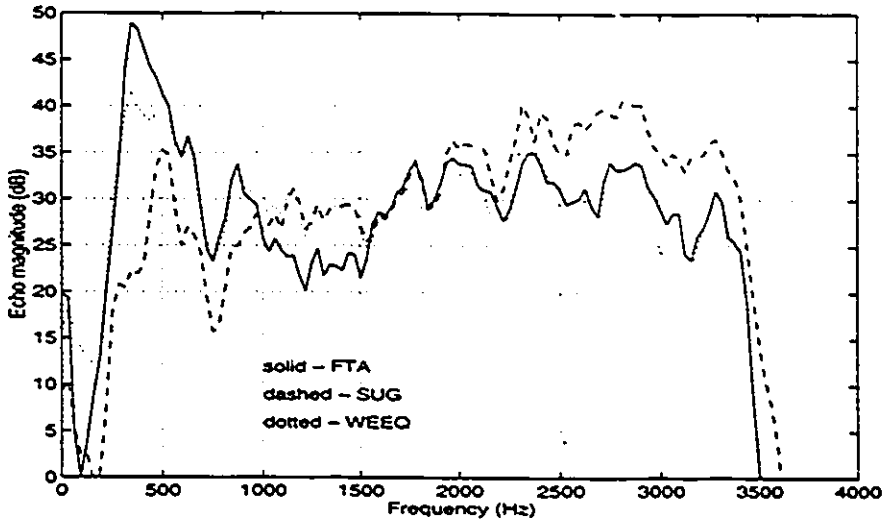

Figure 4.39: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *m98*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

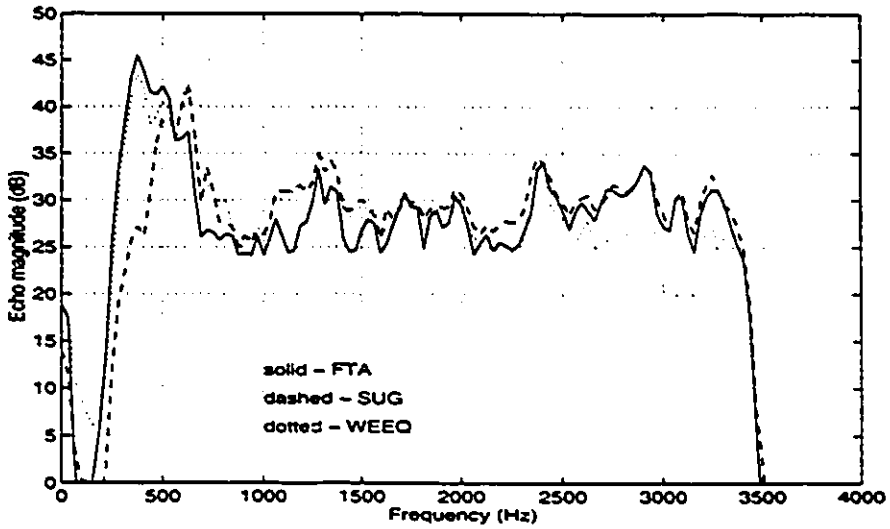

Figure 4.40: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

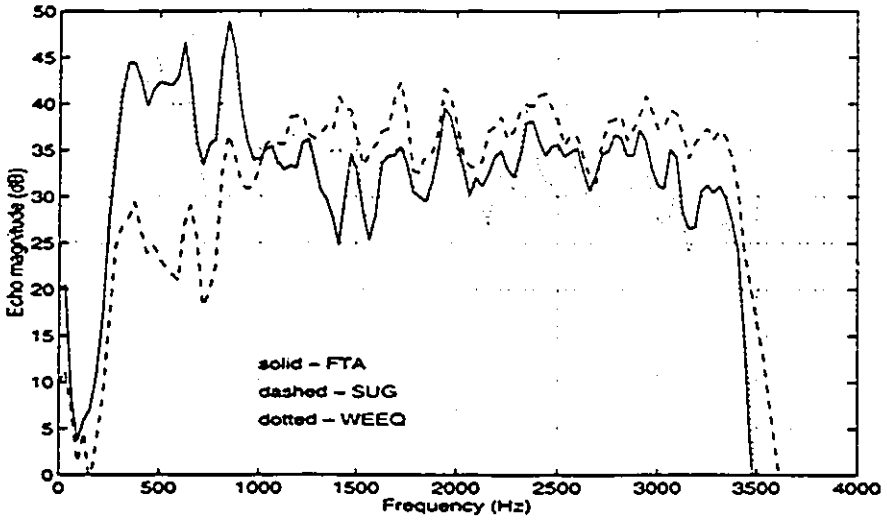

Figure 4.41: Error spectra for FTA, SUG and WEEQ algorithms. Unknown response: *mc3033*. Input: *htf*. Parameters: $M = 8$, $\mu = 1.0$ (all bands), $S = 100$ (SUG, WEEQ), $1 \leq R \leq 3$ (SUG), $1 \leq R \leq 5$ (WEEQ).

else to rise somewhat, in compensation. With bandlimited speech, the adaptive tap assignment algorithms do not have as dramatic an effect. The fixed tap assignment echo, after filtering by the bandlimiting filter, is closer to the desired spectral shape than it was when natural speech was used as input. Thus, the adaptive tap assignment algorithms do not effect as much shaping to the echo as they did for the natural speech case. For the simulations with speakerphone-room response *mc3033*, the bandlimiting has not had as much effect as it did for *WBN13* since *mc3033* is essentially a high-pass response and the low frequencies of the speech signals were already attenuated.

Conclusion

The WEEQ algorithm still shapes the echo to fit the curve of the absolute hearing threshold, but there is less difference between the echo of the fixed tap assignment algorithm and the WEEQ algorithm for bandlimited speech than there was for natural speech. Due to the low frequency attenuation of the bandlimiting filter, the resultant echo of the Sugiyama algorithm now tends to contain more high frequency echo than

	m98-WBN13		*m54-WBN13*		*htf-WBN13*		*m98-mc3033*		*m54-mc3033*		*htf-mc3033*	
band	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ
0	1500	1480	1750	1550	1400	1280	1150	490	1170	540	1030	260
1	600	600	400	500	730	750	500	190	300	200	1000	490
2	60	65	40	80	100	130	80	70	60	70	70	70
3	60	80	40	80	50	70	280	280	110	90	70	220
4	60	65	40	65	50	70	200	260	120	210	60	360
5	32	10	40	20	32	10	80	330	150	370	80	460
6	32	20	40	65	32	10	50	460	150	590	40	410
7	40	20	40	20	40	0	100	300	110	300	32	80

Table 4.4: Average distribution of taps for the SUG and WEEQ adaptive tap assignment algorithms after weight vector length convergence (natural speech).

low frequency echo.

4.8.5 Initial Tap Assignment

At first, the taps were assigned equally among the bands and then allowed to converge to whatever solution the algorithm found. After the weight vector lengths of the algorithm had settled, average weight vector lengths were computed for each band. These steady state weight vector lengths are tabulated in Table 4.4 for natural speech and Table 4.6 for bandlimited speech. (Note that the averages may not add up exactly to $M \cdot L = 8 \cdot 300 = 2400$ taps.) For natural speech, it is clear that most of the available taps end up in the lower frequency bands. This allows one to make a better choice of initial starting weight vector lengths. Choosing weight vector lengths judiciously might allow the algorithm to find the optimal tap distribution more quickly.

It can be seen from Table 4.4 that band 0 typically receives the lion's share of the taps while bands 4 to 7 receive fewer taps when response *WBN13* was being modelled. It makes sense to start with an initial tap assignment which favours the likely outcome of the adaptive tap assignment procedure. Table 4.5 gives an initial tap assignment, based on the numbers of Table 4.4, for both the Sugiyama and WEEQ algorithms. Note that the initial tap assignment given in Table 4.5 is not as radical as the values of Table 4.4. In order to avoid assigning zero taps to any band as an initial starting point, each band was allowed at least 100 taps, thus restricting the radicality of the initial tap assignment. To illustrate the effect of the initial tap assignment, Figure 4.42 shows the weight vector length curve for band 0 of the WEEQ algorithm for the initial tap assignment of Table 4.5 and for equal initial tap assignment. Apparently, the initial tap assignment has not really led to faster weight vector length convergence. The biased initial tap assignment has simply led to greater overshoot in the weight vector length convergence curve. For response *mc3033*, the final tap distributions of Table 4.4 are not even that different from the initial tap assignment of 300 taps per band so that a biased initial tap assignment would appear to be an unnecessary addition to the algorithm.

band	SUG	WEEQ
0	1000	800
1	600	300
2	300	300
3	100	200
4	100	200
5	100	200
6	100	200
7	100	200

Table 4.5: Initial tap assignment for SUG and WEEQ.

When the algorithms were simulated using bandlimited speech, as in Subsection 4.8.4, the final weight vector lengths were different from those of Table 4.4, due to the different spectral content of the input signals and the different weighting applied to the subband errors (in the case of the WEEQ algorithm). Table 4.6 shows the approximate final weight vector lengths for the Sugiyama and WEEQ algorithms when trained with bandlimited speech. Here, the final tap distribution is not even as radical as it was for most cases in Table 4.4 so that the implementation of a biased initial tap assignment really does not appear justified.

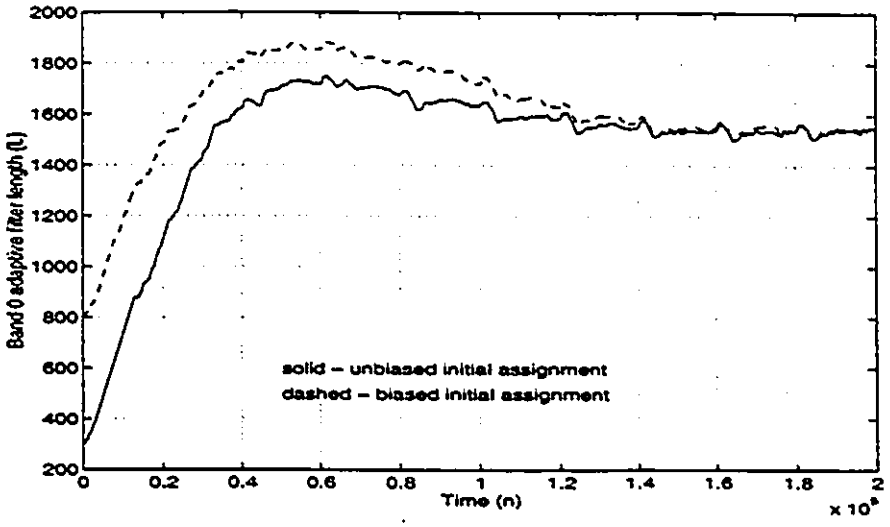

Figure 4.42: Weight vector lengths of the WEEQ algorithm with biased and unbiased initial tap assignments versus time for band 0. Unknown response: *WBN13*. Input: *m54*. Parameters: $M = 8$, $\mu = 1.0$, $S = 100$, $1 \leq R \leq 5$.

	m98-WBN13		*m54-WBN13*		*htf-WBN13*		*m98-mc3033*		*m54-mc3033*		*htf-mc3033*	
band	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ	SUG	WEEQ
0	1500	1050	1580	1080	1080	700	1080	580	1000	460	830	200
1	620	740	580	780	1050	940	640	330	290	270	1050	510
2	60	190	70	170	100	360	90	80	80	80	90	100
3	60	210	60	170	70	110	270	350	210	190	80	290
4	60	80	50	70	70	170	160	300	210	300	80	400
5	32	20	40	65	40	40	100	320	240	470	180	450
6	32	20	40	80	32	70	40	360	240	540	40	410
7	32	10	32	10	32	0	32	20	32	20	32	10

Table 4.6: Average distribution of taps for the SUG and WEEQ adaptive tap assignment algorithms after weight vector length convergence (bandlimited speech).

4.9 Subband NLMS versus Fullband NLMS

This section presents a brief comparison of the objective performance of the subband adaptive filtering algorithms with the performance of the NLMS algorithm. Figure 4.43 and Figure 4.44 show steady-state error spectra and averaged ERLE performance, respectively, for the NLMS algorithm and the fixed tap assignment algorithm for input speech *m54* and room response *WBN13*. A direct comparison between the NLMS algorithm and the adaptive tap assignment algorithms is not given, but such a comparison is not necessary since the adaptive tap assignment algorithms can merely be measured up against the fixed tap assignment algorithm, and their performance against the NLMS algorithm then simply inferred from that.

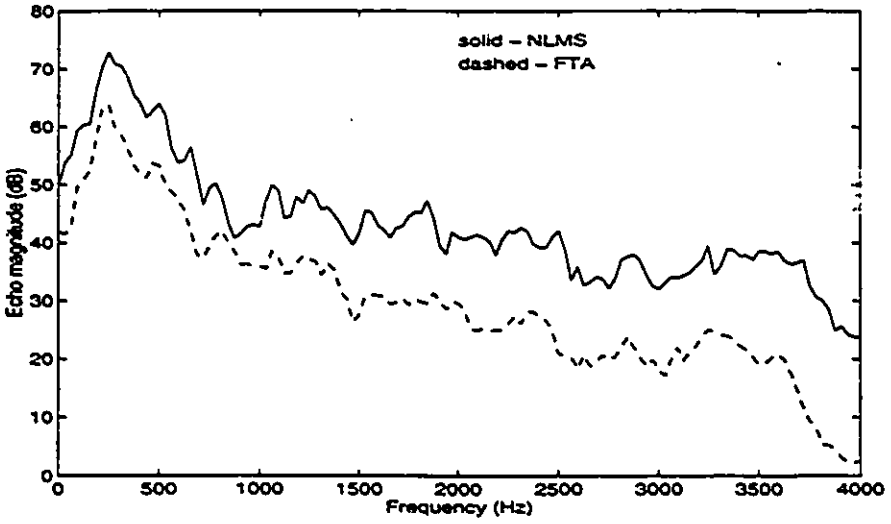

Figure 4.43: Steady state error spectra for the FTA algorithm and the NLMS algorithm. Input: *m54*. Unknown response: *WBN13*. $\mu = 1.0$ (both algorithms). Tap assignment: 300 taps/band (FTA) and 300 taps (NLMS).

The lengths of the adaptive filters of each algorithm were chosen to span the same length of time. Thus, since the subband adaptive filter used no decimation, each of the subbands in the subband adaptive filter has an adaptive filter of 300 taps, or the same length as the full-band adaptive filter. Figure 4.43 shows that the NLMS echo contains significantly more echo than the FTA echo at all frequencies. The ERLE performance

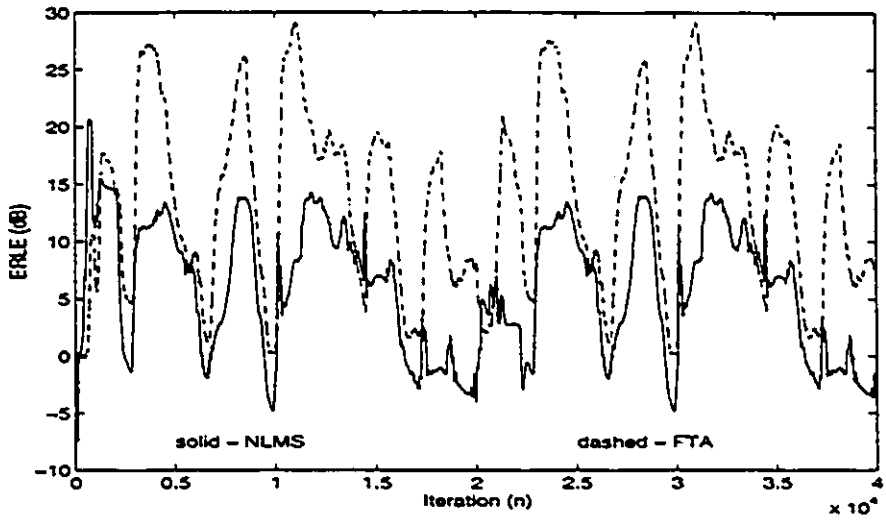

Figure 4.44: Averaged *ERLE* for the FTA algorithm and the NLMS algorithm. Input: *m54*. Unknown response: *WBN13*. $\mu = 1.0$ (both algorithms). Tap assignment: 300 taps/band (FTA) and 300 taps (NLMS).

of the subband adaptive filter here is superior to that of the NLMS algorithm although this improvement comes at the expense of a good deal of additional computational complexity, as outlined earlier.

4.10 Listening Test Results

The ultimate goal of this thesis was to implement an algorithm which would make use of the hearing properties of the human ear to shape the error in the least (subjectively speaking) disturbing manner possible. By shaping the error spectrum to achieve this goal the performance of the given adaptive filter structure would be maximized. The weighted error equalizing algorithm (WEEQ algorithm), based on the absolute hearing threshold curve, has been proposed as just such an algorithm. The objective results presented thus far have shown how this algorithm was, for the most part, successful in shaping the error spectrum in the desired manner. The acid test for this algorithm; however, must be entirely subjective since the whole concept of the WEEQ algorithm is based

on subjective hearing criteria. To evaluate the performance of the WEEQ algorithm, informal listening tests were conducted by the author of this thesis. The listening test set-up and the results of the tests are presented below. These tests were performed using both natural speech and bandlimited speech. Note that these listening tests were by no means rigorous and should not be construed as a subjective listening test. In particular, since the author was the only participant in the tests, the results should only be interpreted as an indication of the performance of the said algorithms. Further tests involving a large group of unbiased listeners and taking into consideration such factors as listener learning and fatigue would be necessary to make any definitive statements about the subjective performance of the algorithms studied here.

4.10.1 Listening Test Methods

Ideally, to test any acoustic echo canceller from a subjective point of view, the algorithm must be implemented in real-time on a speakerphone. Figure 4.45 shows the layout of a real-time testing system.

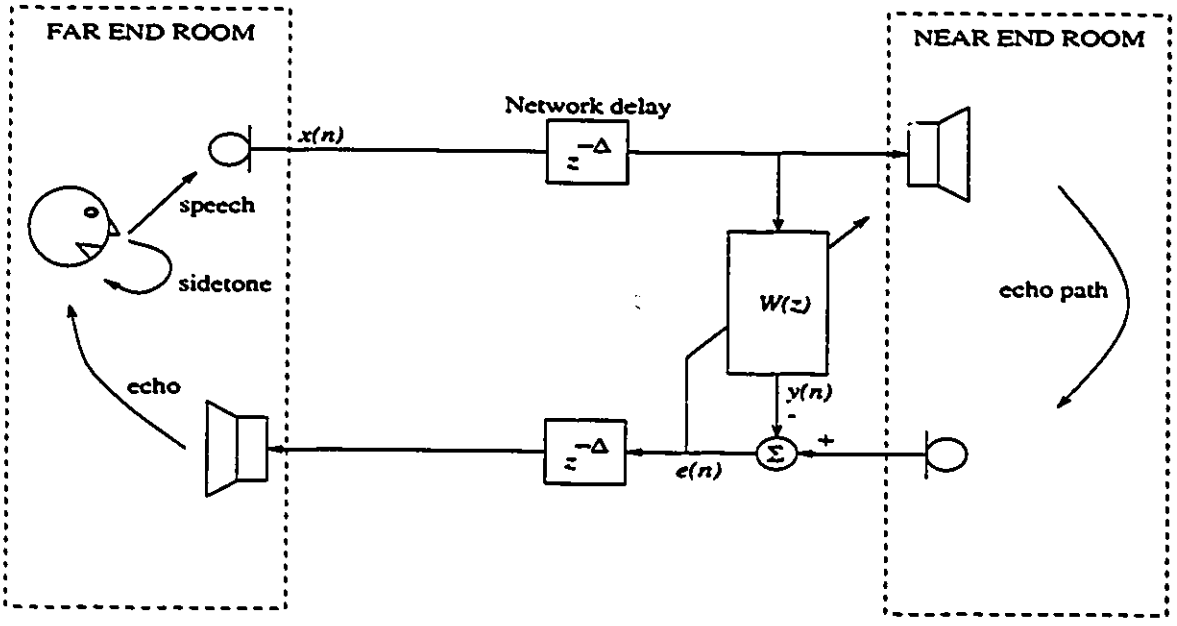

Figure 4.45: Real-time hands-free communication link set-up.

Without access to a real-time set-up, a good approximation is to run simulations

in "batch" mode and then conduct listening tests as follows: the adaptive filter error sequence, obtained from the non-real-time simulation, is added to the original input sequence and the composite signal, which shall be called $s_{test}(n)$, is evaluated for subjective quality in a listening test. It must be emphasized that there remains some difference between this "batch" mode of testing and a real-time test. In the real-time set-up diagrammed in Figure 4.45 the echo is perceived by the far-end user while he/she is speaking whereas in the "batch" listening test, the listener passively listens to a composite signal, without doing any speaking. The rationale for adding the input sequence to the error sequence for the batch listening tests is that when a person speaks they inherently hear their own voice as they speak. This speech that is heard by the speaker is known as "sidetone". The addition of the input to the error sequence in the batch tests is a crude way of adding sidetone to the echo signal. The method of generating the composite test signal $s_{test}(n)$ is presented in Figure 4.46. Sidetone at a person's ear may be at a level 12 dB below that at their mouth (the point of origination) [81]. Thus $x(n)$ in Figure 4.46 is attenuated by 12 dB before addition to the echo in order to promote the greatest similarity possible to a real time set-up.

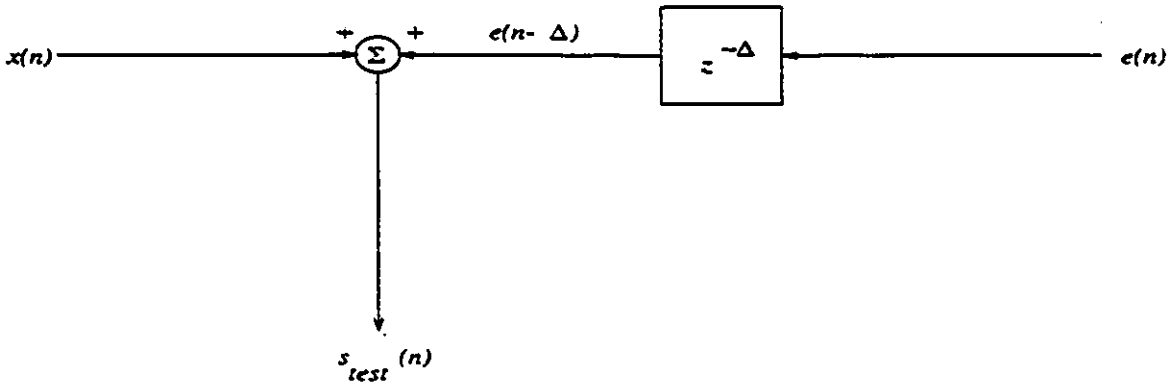

Figure 4.46: Nonreal-time listening test signal generation.

Once a composite test signal $s_{test}(n)$ was generated for a given input, room response and adaptive filtering algorithm, its subjective quality was evaluated by simply listening to the signal and comparing it to a benchmark composite signal for that input and room response. All listening tests in this thesis were conducted using a single speaker on a

Pentium PC equipped with a sound blaster card. All audio signals were sampled at 8 kHz using 16 bit quantization. The speakers provide surprisingly good quality sound considering their size and were sufficient to evaluate the quality of telephone bandwidth speech. The benchmark composite signal, against which other composite signals were compared, was the fixed tap assignment (FTA) algorithm composite signal. The other composite signals were generated by the WEEQ algorithm, the Sugiyama algorithm and the NLMS algorithm.

The Δ factor shown in Figure 4.45 is the delay involved in sending a signal across the communication network. The round-trip delay is $2 \cdot \Delta$ and the echo generated by the near-end speakerphone in a hands-free communication link will be delayed by $2 \cdot \Delta$, relative to the far-end input speech which generated it, by the time it reaches the far-end speaker's ears. This factor varies depending upon the type of communication link used to connect the two ends of the hands-free communication link. For local telephone calls, Δ is a negligible factor whereas if the two ends of the communication system are linked using a satellite link, Δ may be on the order of 300 ms. It was mentioned in Chapter 2 that as the round-trip delay (RTD) of the communication link increases, the echo becomes subjectively worse even though the ERLE remains the same. To fully test the effectiveness of the frequency weighting schemes, the composite signals produced by the various algorithms were evaluated with three different RTD's: 0 ms (ie. no delay - local connection), 100 ms (long distance connection), and 300 ms (satellite link).

In order to quantify the results of the listening tests, the quality of the composite signal for a given adaptive algorithm was classified using one of the following relative descriptors:

1. SB, significantly better
2. B, slightly better
3. E, equal
4. W, slightly worse

5. SW, significantly worse.

All of the relative descriptors above refer to the quality of the composite signal for the given adaptive algorithm relative to the quality of the composite signal for the fixed tap assignment subband adaptive filtering algorithm. It should be noted that if two composite test signals received the same rating, these composite test signals do not necessarily sound the same. It is quite possible for one to contain more low frequency echo and the other more high frequency echo. What is indicated, is simply that the listener (in this case, the author) could not definitely conclude that one composite test signal had better quality than the other. In evaluating the performance of the adaptive tap assignment algorithms on an objective basis above, six different input and room combinations have been considered. The listening test shall be applied to each of these six combinations in Subsection 4.10.2.

4.10.2 Natural Speech Results

The listening test results for natural speech are presented in three segments below. Each of Tables 4.7, 4.8 and 4.9 gives listening test results for a particular value of Δ. Table 4.7 gives the results for $\Delta = 0$, Table 4.8 for $\Delta = 100ms$ and Table 4.9 for $\Delta = 300ms$. The entry in the table columns under each particular algorithm gives the listening test rating of the composite echo signal in accordance with the rating guidelines outlined above.

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	SB	SB
m54-WBN13	SW	SB	SB
htf-WBN13	SW	SB	SB
m98-mc3033	SW	W	E
m54-mc3033	SW	E	E
htf-mc3033	SW	W	E

Table 4.7: Listening test results for $\Delta = 0ms$ (natural speech).

The adaptive tap assignment algorithms (Sugiyama and WEEQ) consistently outperformed the fixed tap assignment algorithm in terms of subjective echo quality for speakerphone-room response *WBN13*. For speakerphone-room response *mc3033*, the performance of the SUG algorithm was typically slightly worse than the FTA algorithm

while the WEEQ algorithm performed equal to the FTA algorithm except when speech signal *m54* was used in as input. In this case, the WEEQ algorithm managed to slightly outperform the FTA algorithm. All of the subband adaptive filtering algorithms outperformed the NLMS algorithm in terms of subjective echo quality. The difference in subjective performance between simulations with speakerphone-room response *WBN13* and simulations with speakerphone-room response *mc3033* is quite marked. The frequency responses of *WBN13* and *mc3033* are given in Appendix A. Note that *mc3033* has a somewhat high pass type of response while *WBN13* tends more towards low-pass. One can also see in the Figures from Subsection 4.8.2 that the simulations with *WBN13* show that the SUG and WEEQ algorithms achieve a large reduction in low frequency echo relative to the FTA algorithm whereas when considering the simulations with *mc3033* not as much difference in the spectra of the SUG, WEEQ and FTA algorithms is apparent. It is this large difference in low frequency echo cancellation that provides for the echo cancellation quality improvements for simulations with *WBN13*. The spectral shaping effected by the adaptive tap assignment algorithms in simulations with *mc3033* is, for the most part, not significant enough to be reliably perceived by a human listener.

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	SB	SB
m54-WBN13	SW	SB	SB
htf-WBN13	SW	SB	SB
m98-mc3033	SW	W	E
m54-mc3033	SW	E	B
htf-mc3033	SW	W	E

Table 4.8: Listening test results for $\Delta = 100ms$ (natural speech).

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	SB	SB
m54-WBN13	SW	SB	SB
htf-WBN13	SW	SB	SB
m98-mc3033	SW	W	E
m54-mc3033	SW	W	B
htf-mc3033	SW	W	E

Table 4.9: Listening test results for $\Delta = 300ms$ (natural speech).

4.10.3 Bandlimited Speech Results

The listening test results for bandlimited speech are presented in three segments below. Each of tables 4.10, 4.11 and 4.12 gives listening test results for a particular value of Δ. Table 4.10 gives the results for $\Delta = 0$, Table 4.11 for $\Delta = 100ms$ and Table 4.12 for $\Delta = 300ms$. The entry in the table columns under each particular algorithm gives the listening test rating of the composite echo signal in accordance with the rating guidelines outlined above.

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	B	B
m54-WBN13	SW	B	B
htf-WBN13	SW	B	B
m98-mc3033	SW	W	E
m54-mc3033	SW	E	B
htf-mc3033	SW	W	E

Table 4.10: Listening test results for $\Delta = 0ms$ (bandlimited speech).

The listening test results for bandlimited speech are somewhat different than the results for natural speech. The subband adaptive filtering algorithms continued to outperform the NLMS algorithm but the adaptive tap assignment algorithms did not outperform the fixed tap assignment algorithm significantly. For simulations with speakerphone-room response *WBN13*, the adaptive tap assignment algorithms continued to outperform the FTA algorithm though the echoes of the SUG and WEEQ algorithms were no longer significantly better than the FTA echo, just slightly better than the FTA echo. For simulations with *mc3033*, the picture remained much the same as for natural speech. Due to the bandlimiting applied to the speech, some low frequency content was lost. This lead to echoes with less low frequency content and the gains of the previous subsection were lost somewhat since there was less low frequency echo to cancel in the *WBN13* case. For the *mc3033* case the bandlimiting had less effect since low frequencies were already quite attenuated by the high pass *mc3033* response. The echo spectra plotted in Subsection 4.8.4 also show that the SUG and WEEQ algorithm do not effect as much change to the spectrum of the FTA algorithm at low frequencies for *WBN13* as they did in Subsection 4.8.2.

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	B	B
m54-WBN13	SW	B	B
htf-WBN13	SW	B	B
m98-mc3033	SW	W	E
m54-mc3033	SW	E	B
htf-mc3033	SW	W	E

Table 4.11: Listening test results for $\Delta = 100ms$ (bandlimited speech).

Input and room	Algorithm		
	NLMS	SUG	WEEQ
m98-WBN13	SW	B	B
m54-WBN13	SW	B	B
htf-WBN13	SW	B	B
m98-mc3033	SW	W	E
m54-mc3033	SW	E	B
htf-mc3033	SW	W	E

Table 4.12: Listening test results for $\Delta = 300ms$ (bandimited speech).

4.11 Summary

This chapter has investigated subband adaptive filtering from a psychoacoustic viewpoint. The basic design process of the tree-structured perfect reconstruction filter bank was briefly summarized. Some facets of the operation of the NLMS algorithm in a nonstationary environment were explored. Specifically, the tendency of the misadjustment of the weight vector to vary erratically when the power estimate became small was investigated. This erraticness in misadjustment was controlled by limiting the step size of the NLMS algorithm. The limit on step size was calculated according to the theoretical maximum step size for the fixed step size LMS algorithm. By employing this limit on step size, misadjustment of the weight vector was reduced and echo cancellation was improved. The use of this step size limit was employed both for the fullband NLMS algorithm and for the subband adaptive filtering algorithm, which employed the NLMS algorithm for updating the weight vectors of the individual subbands.

Three adaptive tap assignment algorithms were presented. The first, known as the Sugiyama algorithm, was proposed in [58]. The equations of this algorithm were presented. A similar algorithm, known as the EEQ algorithm, was proposed and described. The Sugiyama and EEQ algorithms share the goal of equalizing the error power among the subbands. Objective performance evaluations of the two algorithms showed that

they perform similarly in terms of steady-state echo cancellation and convergence rate. The absolute hearing threshold curve was introduced as the perceptual hearing criterion by which the echo should be shaped. The WEEQ algorithm, requiring only a slight modification to the EEQ algorithm, was introduced as the psychoacoustically based adaptive filtering algorithm which would give the echo the same spectral shape as the absolute hearing threshold.

Objectively, the WEEQ algorithm created an echo with more echo at lower frequencies than at high frequencies. The spectral shape of the WEEQ echoes complied approximately with the shape of the absolute hearing threshold curve. The Sugiyama and EEQ algorithms created echoes with flatter spectra than that of the WEEQ algorithm. The spectra achieved by the FTA algorithm depended upon whether or not the input speech was bandlimited or not. With natural speech, the FTA echo was very predominantly low frequency in most cases. When the input speech was bandlimited (characterizing the frequency response of the telephone channel), the fixed tap assignment echo was not as predominantly low frequency. The parameters of the adaptive tap assignment algorithms and their effect on filter performance was discussed.

A method of subjectively evaluating the quality of echo cancellation of various adaptive filtering algorithms was presented. Simple listening tests were applied to the simulations of this chapter. With natural speech as the input to the adaptive filters, the adaptive tap assignment algorithms outperformed the fixed tap assignment algorithm in terms of subjective echo quality for speakerphone-room response *WBN13* while for speakerphone-room response *mc3033* there was no clear performance benefit. When the input speech was bandlimited, the adaptive tap assignment echoes remained of better quality than the FTA echoes for *WBN13* but not by as great a margin. In some cases with bandlimited speech and speakerphone-room response *mc3033*, the Sugiyama algorithm echoes were worse than the fixed tap assignment echo due to the presence of more high frequency echo. In all cases, natural or bandlimited speech, the WEEQ algorithm echo was always as good or better than the echoes of all the other algorithms, subjectively speaking.

The reason for the different results when using bandlimited speech as opposed to natural speech was discussed. The improvement in subjective performance of the adaptive tap assignment algorithms over the fixed tap assignment algorithms for the case of natural speech was due to the very predominant low frequency echo of the fixed tap assignment algorithm. The adaptive tap assignment algorithms greatly reduced the low frequency echo and spread it among the rest of the frequency region. This large reduction in echo at a very specific spot in the frequency range led to a very perceptible improvement in perceptive echo cancellation. With bandlimited speech however, the fixed tap assignment echo was not as predominantly low frequency and thus the adaptive tap assignment algorithms did not greatly change the echo spectrum in any particular place. Without making a large difference to the echo spectrum, the changes effected by the adaptive tap assignment algorithms were much less audible. This is a facet of the perceptual abilities of the human ear. Change must be very significant for the human ear to discern the difference between two signals with the same content (the same sentence of speech).

In conclusion, the adaptive tap assignment algorithms capable of frequency weighting presented here provide improved quality (subjectively speaking) echo cancellation under certain circumstances. In particular, the nature of the room response is very important. Significant gains were obtained with a low pass type speakerphone-room response (*WBN13*) while little or no gains were obtained with a high pass type speakerphone-room response (*mc3033*). The significant gains obtained with natural speech and *WBN13* were also diminished somewhat when the speech was bandlimited to the frequency range 300 to 3400 Hz to represent the limited bandwidth of conventional telephony.

CHAPTER 5

Conclusions

5.1 Discussion of Results

The objective of this thesis has been to design an adaptive filtering algorithm which could shape the echo spectrum in accordance with perceptual hearing criteria of the human auditory system. The specific perceptual criterion used was the absolute hearing threshold curve. The goal was to produce an echo which was as inaudible as possible to the human ear rather than an echo which had the minimum mathematical error possible. Two algorithms were proposed to accomplish this frequency weighting of the echo.

A simple modified LMS adaptive filtering algorithm, the filtered-E LMS algorithm, was presented. Initial hopes that the algorithm would provide frequency weighting of a speakerphone-room echo were unfulfilled but much energy was concentrated on the understanding of the workings of the filtered-E LMS algorithm. These results were somewhat baffling since the filtered-E LMS algorithm had been used in two other applications (active noise control and line echo cancellation) to implement frequency weighting of the error. The ability of the filtered-E LMS to implement frequency weighting of the error in some simulations but not others was linked to the nature of the plant response being modelled. It was found that plant responses with a dominant feature, such as the low frequency tail of *hybrid1* and the transition band of a lowpass filter, were good candidates for filtered-E LMS. By de-emphasizing the dominant feature of the plant

response (the low frequencies of *hybrid1* and the transition region of the lowpass filter), the filtered-E LMS algorithm was able to achieve better echo cancellation at other frequencies. However, if one tries to emphasize the dominant feature of the plant response, no additional echo cancellation is gained there. For fairly "white" plant responses with no dominant feature, such as *WBN13* and *mc3033*, the FELMS algorithm could not improve the cancellation in any frequency regions by weighting the error.

Filtered-E LMS performance was investigated from several perspectives. It was shown that the use of the FELMS algorithm reduced weight vector misadjustment significantly relative to the LMS algorithm when the plant response was *hybrid1* but not when it was *WBN13* or *mc3033*. This was linked to the dominant, unmodelled low frequency tail of *hybrid1* which caused misadjustment in the LMS weight vector but not the FELMS weight vector since it was filtered out by the FELMS weighting vector. This effect did not occur for *WBN13* or *mc3033* since these plant responses had no dominant feature causing misadjustment in the LMS weight vector which could be filtered out by the FELMS weighting vector. It was also shown that in cases where FELMS "worked", the weighting filter essentially gave the adaptive filter a shorter plant response for emphasized frequencies. In cases where FELMS did not "work", the effective plant response length remained the same as the original plant response length even after filtering by the FELMS weighting filter.

An adaptive tap assignment subband adaptive filtering algorithm has been designed which can implement an "absolute hearing threshold curve" frequency weighting on the adaptive filter error. By weighting the error according to the absolute hearing threshold curve for the human ear, the performance of the subband adaptive filter was improved relative to the standard fixed tap assignment subband adaptive filter when the input was natural speech and the unknown system being modelled was speakerphone-room response *WBN13*. However, when the unknown response was speakerphone-room response *mc3033* the adaptive tap assignment algorithms failed to improve the subjective quality of the echo. In fact, the SUG algorithm worsened the echo by passing considerably more high frequency echo than either the FTA or WEEQ algorithm. The

unknown response *mc3033* can be characterized as a high pass response and this offset the low frequency dominant characteristics of the speech sequences used in this thesis. The resultant FTA echo was already fairly close to the shape of the absolute hearing threshold and the WEEQ algorithm did not provide much additional shaping, resulting in an echo of virtually inaudible difference with the FTA echo. With bandlimited speech, the fixed tap assignment algorithm provided an echo of equal subjective quality to the WEEQ algorithm for *mc3033* and only slightly worse in the case of *WBN13*. The reduction in subjective improvement for bandlimited speech versus that of natural speech is due to the extra attenuation of the low frequency components in the speech signal by the bandlimiting effect. The gain in subjective echo quality in the case of natural speech was due to the fact that the fixed tap assignment algorithm had a predominantly low frequency echo and the adaptive tap assignment algorithm cancelled much of this predominantly low frequency echo, redistributing it more evenly, resulting in a less disturbing echo. With bandlimited speech, much of the low frequency content of the speech was attenuated by the telephone channel response. Thus, the fixed tap assignment echo was not as predominantly low frequency as before. The adaptive tap assignment algorithms performed less shaping on the echo in the bandlimited speech case and the resultant echoes were, for the most part, subjectively equal to the fixed tap assignment echoes. The error weighting based on the absolute hearing threshold curve was found to be superior to a simple unweighted error equalizing algorithm in terms of subjective quality of the resultant echo. The unweighted adaptive tap assignment algorithms (the Sugiyama and EEQ algorithms) were found to pass slightly more high frequency echo than the WEEQ algorithm in some cases. All of the subband adaptive filtering algorithms were found to outperform the fullband LMS algorithm in terms of the level of echo cancellation and subjective echo quality.

The EEQ adaptive tap assignment algorithm (upon which the WEEQ algorithm was based) derived was compared to a similar algorithm in [58] and the performance of the two algorithms was found to be similar. The algorithm presented here had some small advantages as far as flexibility in tap assignment was concerned.

In the course of the development of the adaptive tap assignment algorithm for subband adaptive filtering, several observations were made on the nature of the normalized LMS algorithm in a nonstationary environment. Some solutions (not necessarily original) to the problem of nearly unbounded effective step sizes in the NLMS algorithm were discussed and used throughout the simulations of the subband adaptive filtering algorithms.

5.2 Future Research

The work of this thesis opens up several avenues of research which can be pursued, most related to the adaptive filtering algorithms themselves as opposed to the idea of applying psychoacoustic criteria to acoustic echo cancellation. This idea remains valid but adaptive filtering algorithms need to be improved in order to make better use of the fundamental idea of customizing the objective performance function to fit the human ear.

The unstable nature of the step size in the NLMS algorithm under a nonstationary environment was discussed in Section 4.4 and some working solutions were proposed and used to combat the problem. More fundamental research is required into this problem and perhaps a mathematical analysis can provide more insight into the nature of the NLMS step size. A more concretely based proposal for controlling the weight vector misadjustment in a nonstationary environment might be feasible.

The subband adaptive filters used in this thesis did not decimate any of the subband signals. Without decimation, the problems associated with aliasing, as described in Chapter 2 were avoided. The cost of this is the greatly increased computational complexity of the resultant filtering structure. Without any subband decimation, it is unlikely that the subband adaptive filter structure is a likely candidate to be employed in a commercial acoustic echo canceller, due to the high computational cost.

Noncritical subsampling was proposed as a way to forge a compromise between computational complexity and performance but, in the investigations of this thesis, it

was found that the performance of the subband adaptive filter was degraded if any decimation was applied at all. The effects of the decimation became worse and worse as the decimation rate increased up to the maximal decimation rate of 8.

The decimation rate possible is influenced by the filter bank used and in this case it is likely that better filters are needed in order to decimate further. The 40 dB stopband attenuation provided by the filters may be inadequate for acoustic echo cancellation considering the spectrums of some typical speech signals (see Appendix B). However, it must be pointed out that the length of the filters used here was already 106 taps and in order to provide better filters, even more taps might be needed. The delay inherent in a filter bank with long filters is already an issue of concern. It would be worth considering filter bank design procedures which employ direct M-band decomposition rather than the tree-structured filter bank design procedure employed here. Direct M-band decomposition filter banks might provide better filter characteristics for a given filter length than the tree structured filter bank filters. Obviously this is a very open field of study right now and requires further work.

In this thesis, M was chosen to be 8 so that frequency weighting could be investigated adequately. However, filter banks with fewer bands would have a smaller computational complexity, given that the subband signal will not be critically decimated. As well, better analysis/synthesis filters could be designed for small filter banks for the same length of analysis/synthesis filter as used here. The adaptive tap assignment algorithms will likely converge more quickly with a smaller number of bands since the order of the system is smaller. In [58], it can be seen that for an $M = 2$ subband adaptive filter, the convergence of the weight vector lengths is indeed very fast. Of course, some sort of compromise between filter bank size and the ability of the subband adaptive filter to effect the desired spectral shape of the echo would have to be reached. It is possible that a filter bank with $M = 4$ or $M = 5$ would be sufficient. With some of the newer M-channel filter bank design procedures in the literature, filter banks can be designed for an arbitrary number of channels. In fact techniques do exist to design nonuniform filter banks. A nonuniform filter bank need not have equally spaced frequency bands

in addition to having an arbitrary number of channels. Nonuniform filter bank design techniques could be used to design a subband adaptive filter with its subbands having frequency ranges tailor-made to suit the frequency response of the human ear.

The idea of trying to weight the spectral shape of the echo is quite general and possibilities exist to apply this algorithm to other adaptive filtering structures. Efforts to apply this idea to a modified LMS adaptive filter (ie. the filtered-E LMS algorithm) were unsuccessful in an acoustic echo cancellation application but this does not rule out other algorithms such as transform domain algorithms.

All of the simulations in this thesis were for telephone bandwidth speech (0 to 4 kHz). However, in the future, with communication networks destined to be designed for higher quality transmission, wideband speech (0 to 8 kHz) may become more and more common. The adaptive tap assignment algorithms proposed here may provide more improvement over the fixed tap assignment algorithm if the input is wideband speech. The improvement would come from the fact that the speech spectrum from 4 kHz to 8 kHz has considerably less energy than the speech spectrum from 0 kHz to 8 kHz. Thus, fixed tap assignment would give the same number of taps to all of the bands, right from 0 kHz all the way to 8 kHz. The adaptive tap assignment algorithm, on the other hand, would have more taps to work with, in the sense that the upper half of the frequency range would not require many taps. Another important fact to consider is that, with an 8 kHz bandwidth, a larger portion of the hearing range of the human ear would be under consideration. In the absolute hearing threshold curve of Figure 4.9 on page 124, the portion used up until now has consisted of the 0 to 4 kHz range. In this range, the most noticeable fact is that the human ear's sensitivity is decreased for low frequencies. From 1 kHz to 4 kHz, the human ear is essentially at its maximum sensitivity. This frequency range covers most of the range of telephone bandwidth speech and when considering the fact that the telephone channel frequency response cuts out much of the lowest frequency content, there is not much room left for the WEEQ algorithm to work with. However, from 5 kHz to 8 kHz the absolute hearing threshold shows that the sensitivity of the human ear is again decreasing. A wideband WEEQ algorithm

could make use of this to push error into both higher and lower frequencies and really concentrate its echo cancelling power on the most audible frequency region of 1 kHz to 5 kHz. In summary, the use of the adaptive tap assignment algorithms is expected to provide more subjective echo quality improvement in the case of wideband speech than in the current case of narrowband speech.

APPENDIX A

Speakerphone-room and Hybrid Responses

This appendix contains plots of the two speakerphone-room responses and the hybrid response that have been used in the simulations of this thesis. Also included are the approximate frequency responses of the speakerphone-room set-ups and the hybrid. Note that all responses are given for a sampling frequency of 8 kHz. For each speakerphone-room response the TIP/TP curve, defined by equation A.1, has also been plotted. The TIP/TP, or Total Impulse response Power to Tail Power ratio, curve is a measure of the possible depth of cancellation versus the adaptive filter length [7, 8]. The speakerphone-room frequency responses plotted here were obtained by taking 2048 pt. DFT's of the speakerphone-room time impulse responses.

$$TIP/TP = \frac{\sum_{i=0}^{\infty} h_d(i)}{\sum_{i=L}^{\infty} h_d(i)}. \tag{A.1}$$

Speakerphone-room impulse responses were obtained by placing a modified speakerphone inside a conference room and measuring primary and reference signals which were used off-line as inputs to an adaptive filtering algorithm. The resultant adaptive filter taps, after filter convergence, were used as a stationary model of the impulse response of the particular speakerphone-room set-up. The reference signal was bandlimited white

noise (300 to 3400 Hz) sent to the receive line of the speakerphone. The resultant echo, measured on the transmit line of the speakerphone during and after the application of the reference signal, was recorded as the primary signal. The techniques used to measure speakerphone-room impulse responses are described more fully in [7, 9, 10].

The first speakerphone-room impulse response, referred to as *WBN13* and plotted in Figure A.1, was obtained by placing a high quality prototype speakerphone set (from Bell Northern Research) in a medium sized conference room. The dimensions of the room were; width: 3.5 m, length: 5.5 m and height: 2.5 m. The room contained a large central table, several cushioned seats and hard walls. This response was originally obtained by Knappe [7] and the reader is referred there for the source of the information presented here. The frequency response of *WBN13* is plotted in Figure A.2 and the TIP/TP curve for *WBN13* is plotted in Figure A.3.

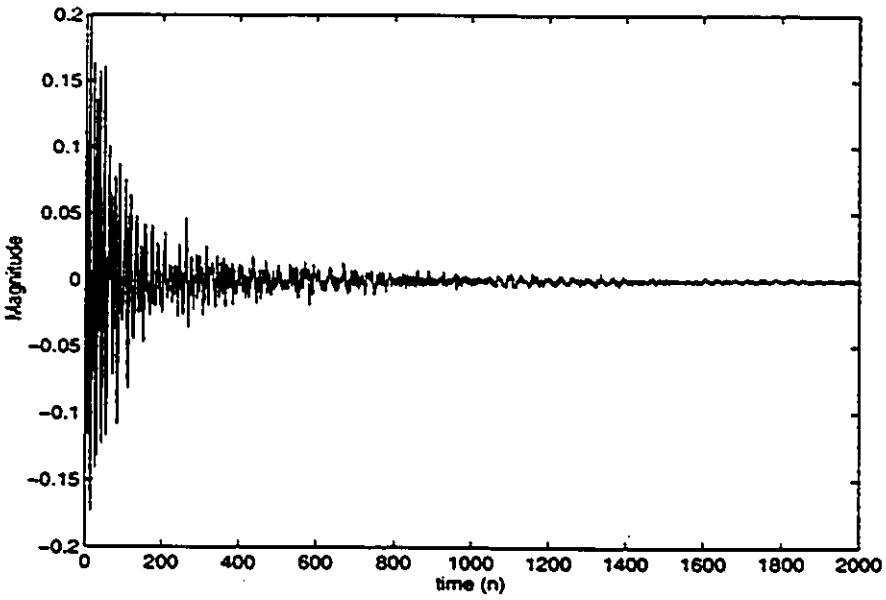

Figure A.1: Impulse response of high-quality prototype speakerphone in conference room *WBN13*.

The second speakerphone-room impulse response, referred to as *mc3033* and plotted in Figure A.4, was obtained by placing a Northern Telecom speakerphone (model NT8B04AA) in conference room 3033 of the Minto Building for Advanced Studies in Engineering at

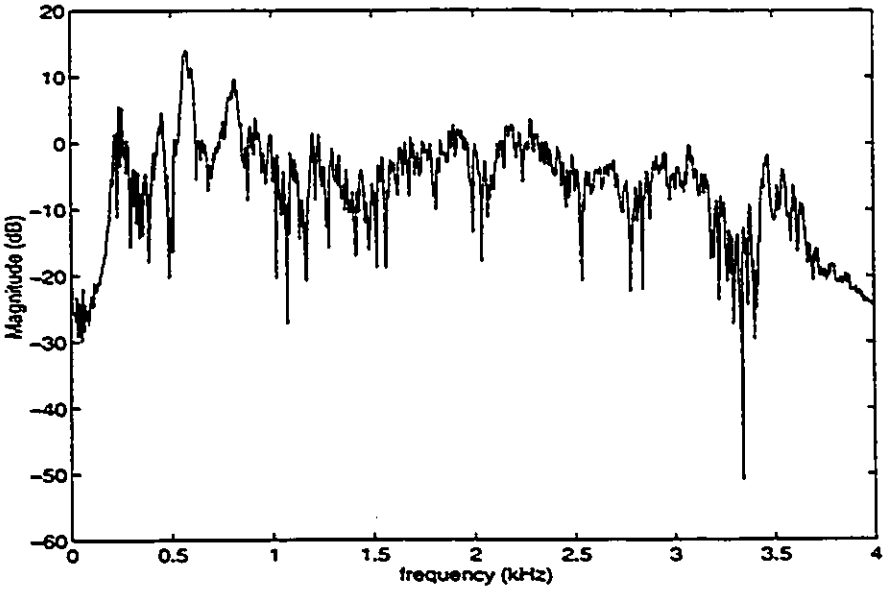

Figure A.2: Frequency response of speakerphone-room *WBN13* set-up.

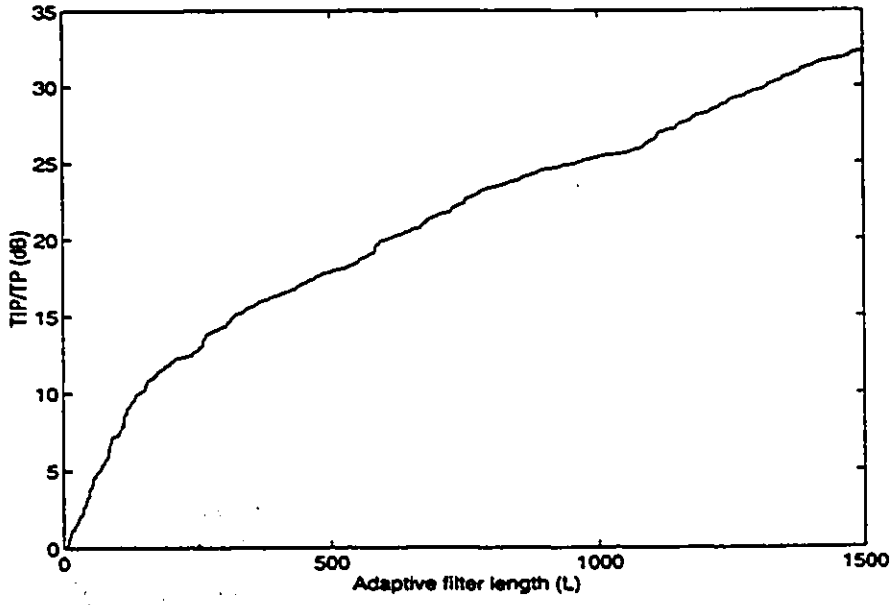

Figure A.3: TIP/TP versus L for speakerphone-room *WBN13* set-up.

Carleton University. The dimensions of the room were: width: 12 ft., length: 18 ft, and height: 10 ft. The room contained two tables, upon one of which the speakerphone was placed, and six padded chairs arranged around the table without the speakerphone. The walls and floor were reflective and the ventilation fan was turned on. This response was originally obtained by Webster [9] and the reader is referred there for the source of the information presented here. The frequency response of *mc3033* is plotted in Figure A.5 and the TIP/TP curve for *mc3033* is plotted in Figure A.6.

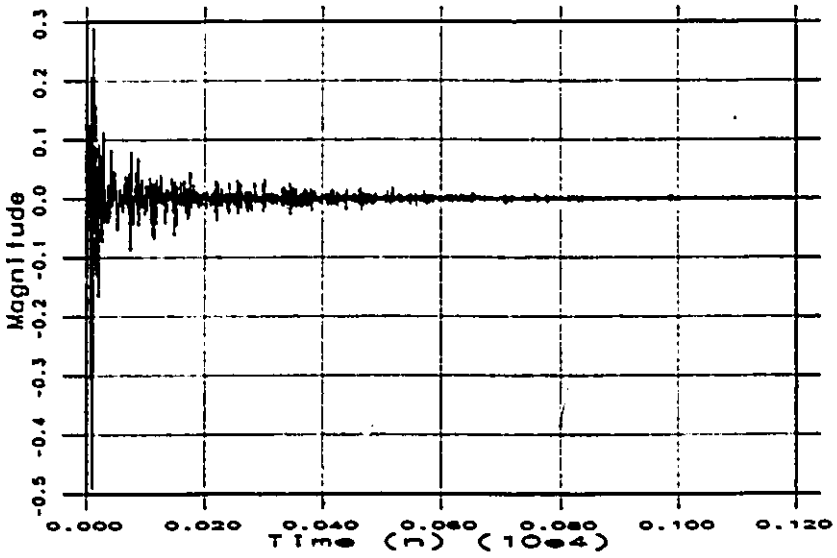

Figure A.4: Impulse response of NT speakerphone in conference room *mc3033*.

The hybrid impulse response *hybrid1* was obtained from Meek [29] and is plotted in Figure A.7. The frequency response of *hybrid1* is plotted in Figure A.8.

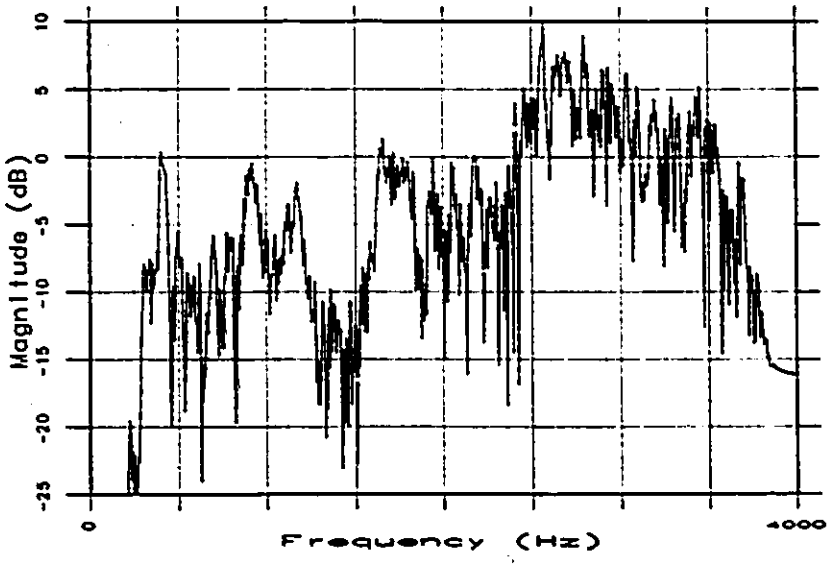

Figure A.5: Frequency response of speakerphone-room *mc3033* set-up.

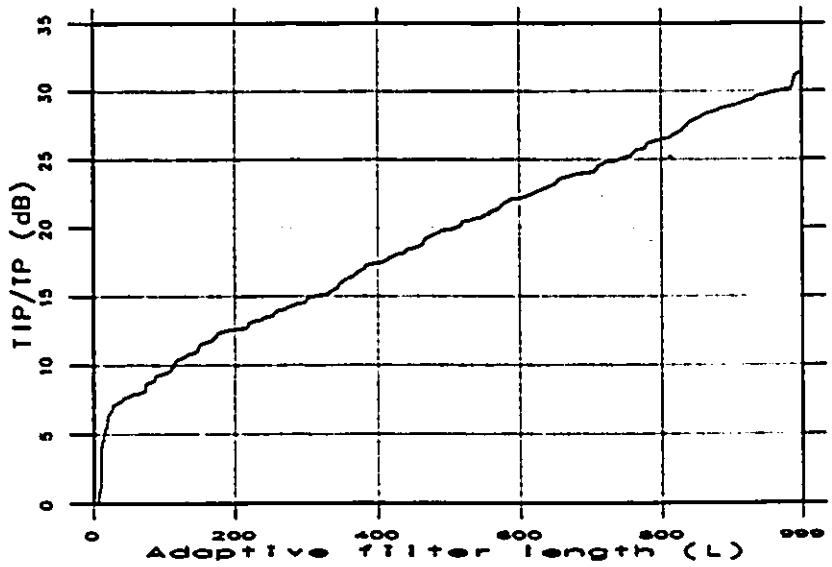

Figure A.6: TIP/TP versus L for speakerphone-room *mc3033* set-up.

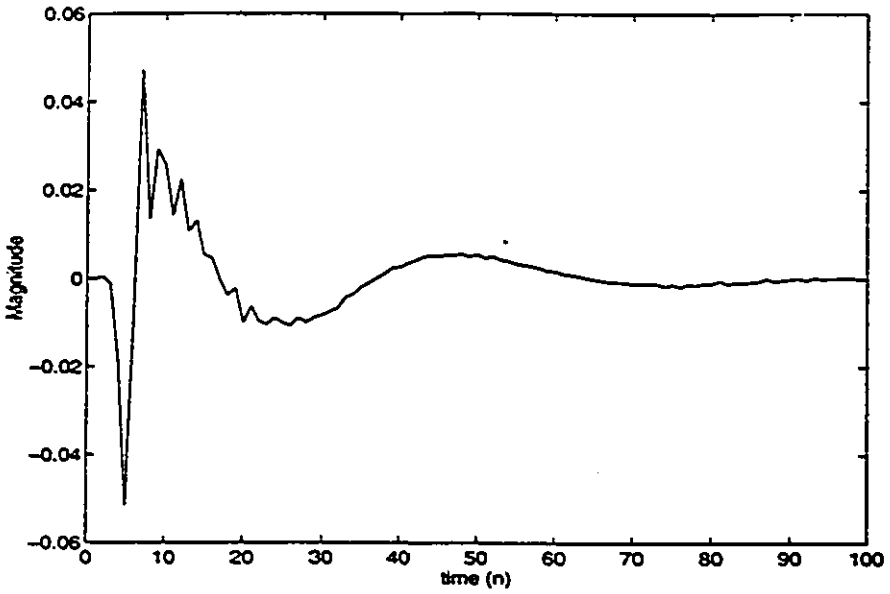

Figure A.7: Impulse response of *hybrid1*.

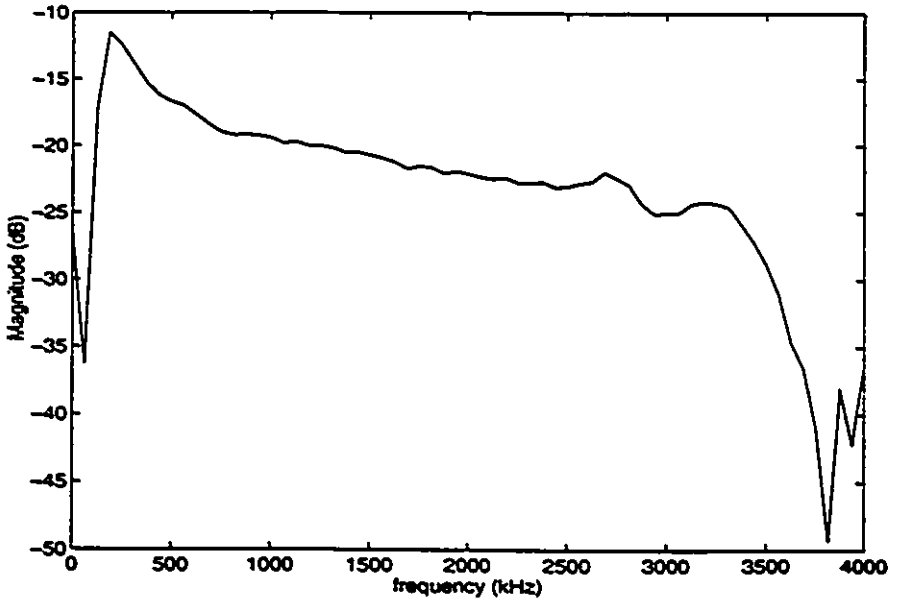

Figure A.8: Frequency response of *hybrid 1*.

APPENDIX B

Speech Sequences

This appendix contains plots of the three speech sequences that have been used in the simulations of this thesis. Also included are the averaged frequency spectrums of the speech sequences. Note that all speech sequences are given for a sampling frequency of 8 kHz. All speech signals used were obtained from Bell Northern Research [65] and were recorded using 16 bit quantization in an anechoic room. The speech sequences shown in the following plots are shown in 16 bit integer format with full scale being 2^{15}. Longer speech sequences were obtained by concatenating several of the following three speech sequences together. This technique of obtaining a longer speech sequence was also used in [9]. The three different speech sequences were not mixed in this process so that a concatenated version of *m98* refers to a sequence containing several copies of sequence *m98* and so on. The lengthened version of *m98* is shown in Figure B.7. Lengthened versions of the other two speech signals are not plotted but take very similar forms. Frequency responses of the speech signals were obtained using the `spectrum` program available with the `Matlab` signal processing package. The `spectrum` program computes N-pt. DFT's over the length of the given sequence and calculates the averaged square value of the DFT's to give the resultant power spectrum. For this thesis, 256 pt. DFT's were used. Note that this method of computing the power spectrum was used throughout the thesis when spectra of echo and input speech signals were required.

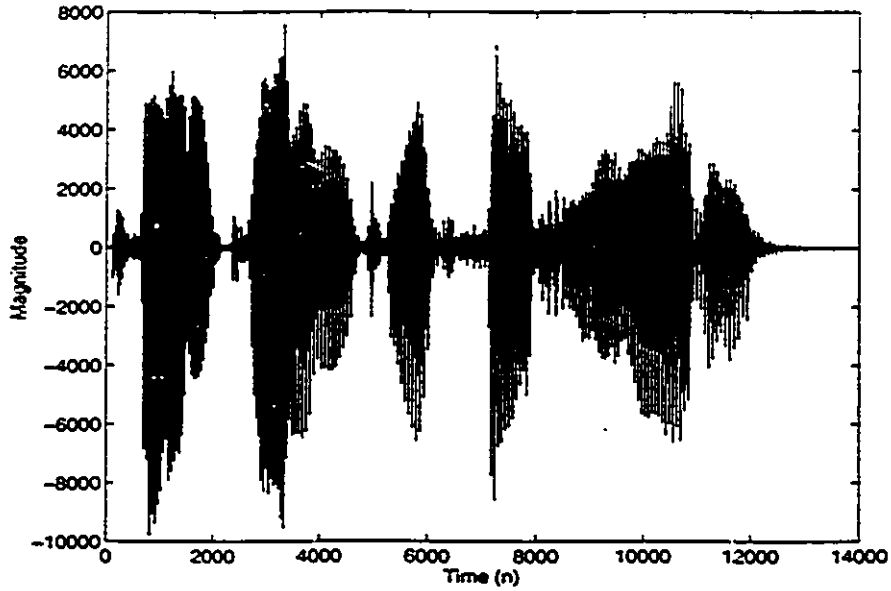

Figure B.1: Speech sequence *m98*: male uttering the phrase "ten pins were set in order".

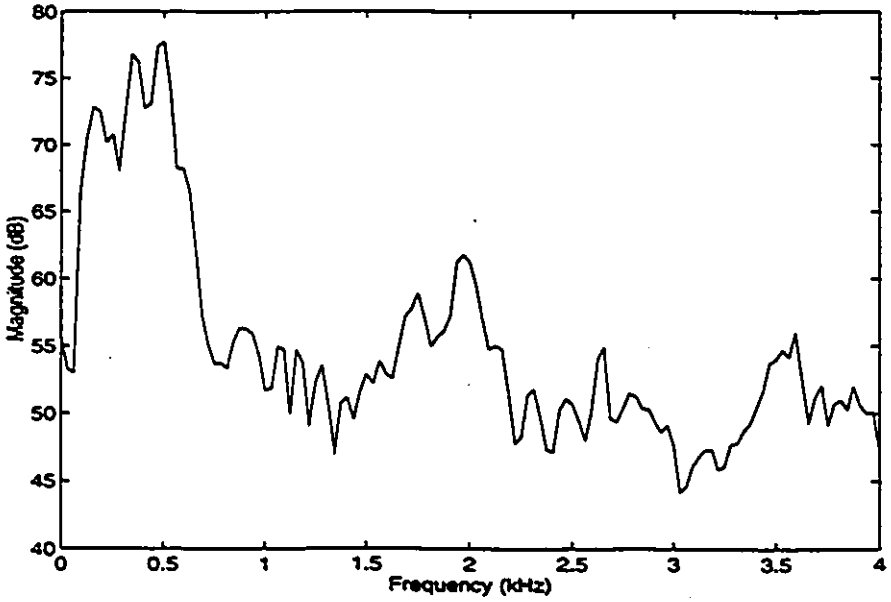

Figure B.2: Power spectrum of speech sequence *m98*.

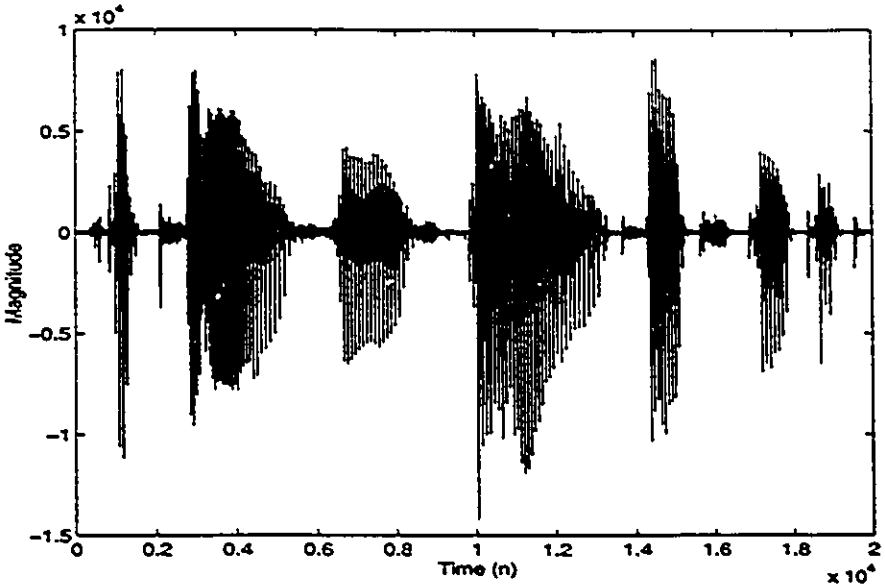

Figure B.3: Speech sequence *m54*: male uttering the phrase "the play seems dull and quite stupid".

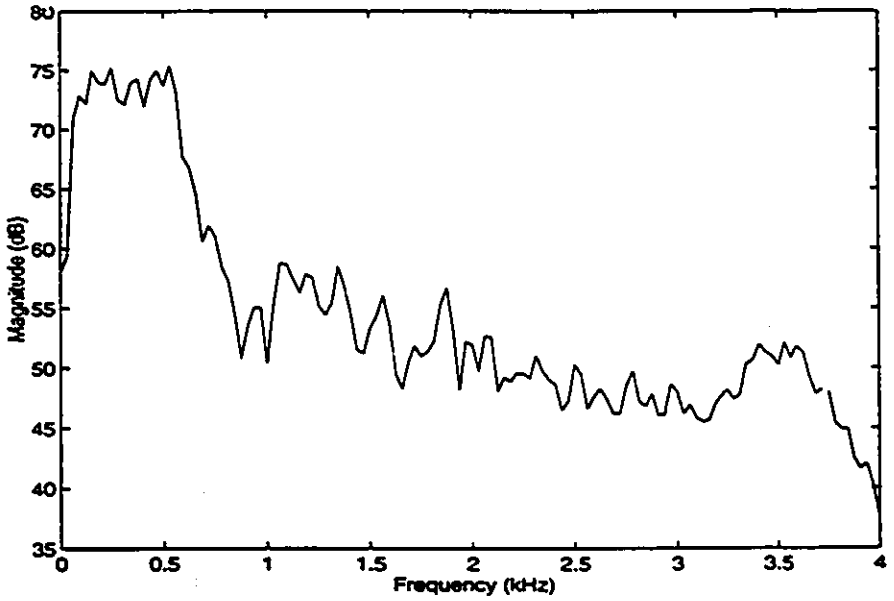

Figure B.4: Power spectrum of speech sequence *m54*.

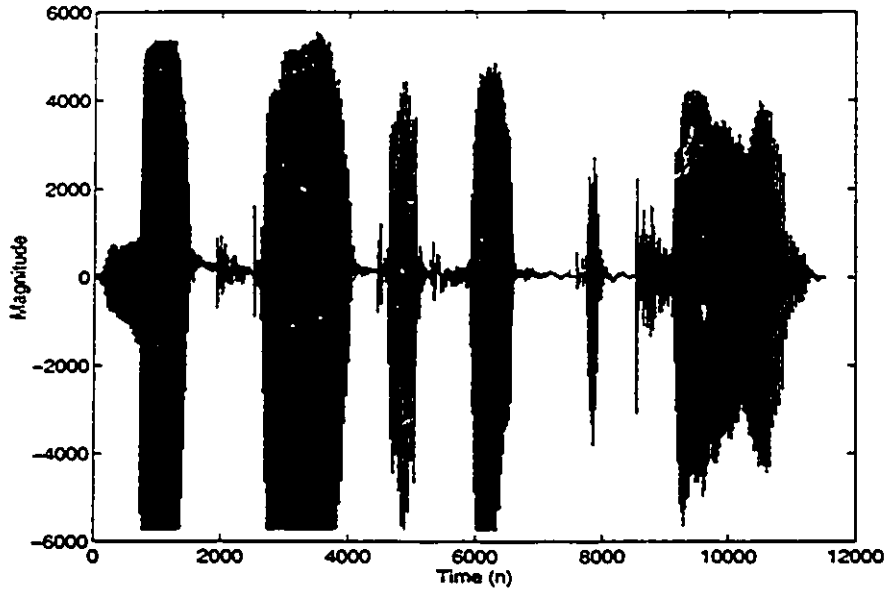

Figure B.5: Speech sequence *htf*: female uttering the phrase "looks like it hit the tree".

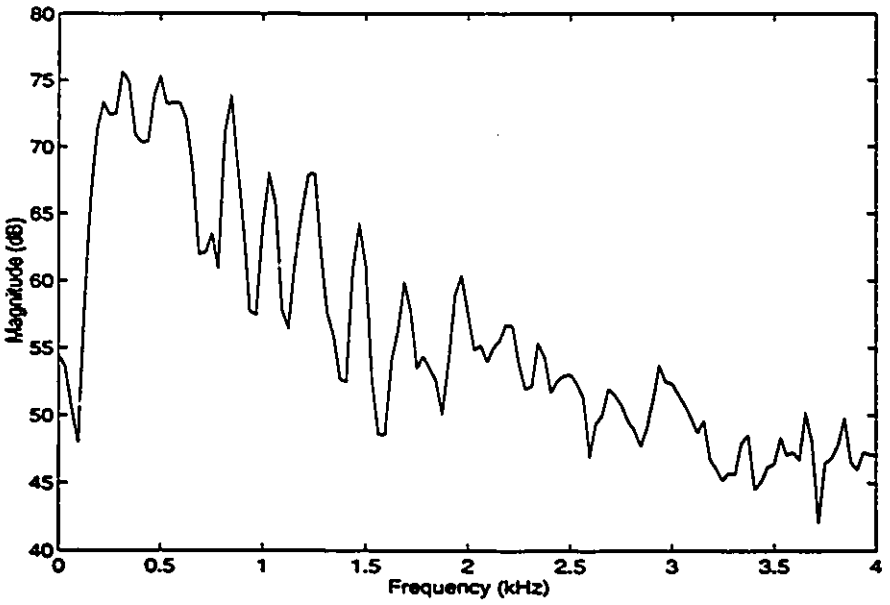

Figure B.6: Power spectrum of speech sequence *htf*.

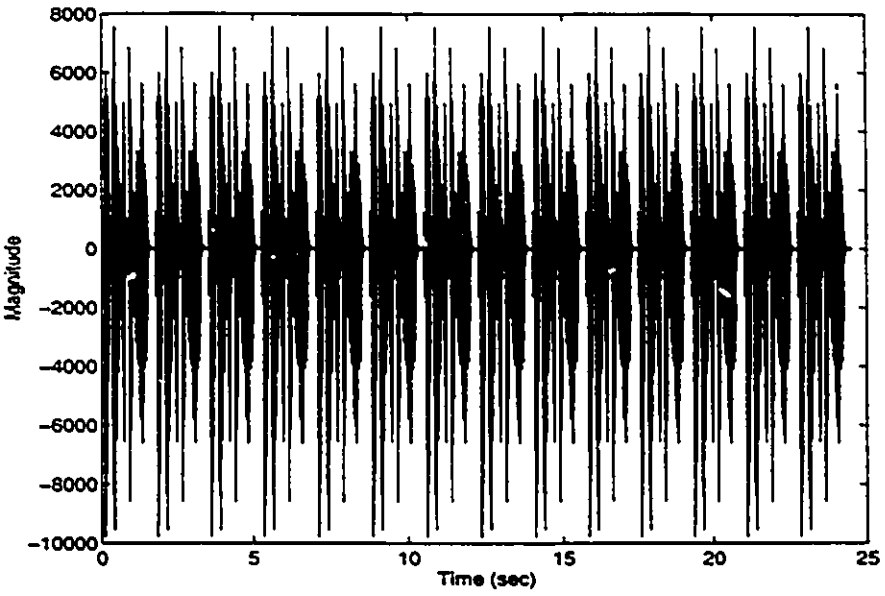

Figure B.7: Concatenated speech sequence *m98*.

APPENDIX C

A Measure of Weight Vector Misadjustment

For an FIR weight vector modelling some linear unknown system, where the unknown system has an impulse response given by:

$$h_d(i), \qquad \text{for} \quad 0 < i < \infty, \tag{C.1}$$

the optimum L tap weight vector is given by:

$$\mathbf{W}^{opt} = \mathbf{R}_{\mathbf{XX}}^{-1}\mathbf{P}. \tag{C.2}$$

This solution is known as the Wiener solution. The autocorrelation matrix $\mathbf{R}_{\mathbf{XX}}$ was defined in Chapter 2. The second vector $\mathbf{P}$ is the cross-correlation between the input vector $\mathbf{X}(n)$ and the desired response $d(n)$, defined as follows:

$$\mathbf{P} = E(\mathbf{X}(n) \cdot d(n)). \tag{C.3}$$

It can be shown that if the input $x(n)$ is an uncorrelated white noise process that the coefficients of the Wiener optimum weight vector $\mathbf{W}^{opt}$ can also be determined from the following equation:

$$w_i^{opt} = h_d(i), \qquad \text{for} \quad i = 0, 1, \cdots, L-1. \tag{C.4}$$

First, equation C.2 is rewitten as:

$$\mathbf{R_{XX}W}^{opt} = \mathbf{P}. \tag{C.5}$$

If $x(n)$ is an uncorrelated white noise process then the autocorrelation matrix $\mathbf{R_{XX}}$ is simply a diagonal matrix with all of its diagonal elements equal to the variance of the noise:

$$\mathbf{R_{XX}} = \begin{bmatrix} \sigma_x^2 & 0 & \cdots & 0 \\ 0 & \sigma_x^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_x^2 \end{bmatrix}$$

which reduces to:

$$\mathbf{R_{XX}} = \sigma_x^2 \cdot \mathbf{I}$$

where $\mathbf{I}$ is the identity matrix. Equation C.5 can then be easily replaced by:

$$\sigma_x^2 \cdot \mathbf{W}^{opt} = \mathbf{P}. \tag{C.6}$$

The definition of $\mathbf{P}$ was given in equation C.3. Expanding equation C.3 gives:

$$\begin{aligned} \mathbf{P} &= E(d(n) \cdot [\ x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)\ ]^T) \\ &= [\ E(d(n) \cdot x(n)) \quad E(d(n) \cdot x(n-1)) \quad \cdots \quad E(d(n) \cdot x(n-L+1))\ ]^T. \end{aligned}$$

Equating coefficient i of the left hand side of equation C.6 with coefficient i of the right hand side of equation C.6 gives:

$$\sigma_x^2 \cdot w_i^{opt} = E(d(n) \cdot x(n-i)). \tag{C.7}$$

It is now helpful to express $d(n)$ in terms of $x(n)$ and $h_d(i)$ as follows:

$$d(n) = \sum_{j=0}^{\infty} x(n-j) \cdot h_d(j). \tag{C.8}$$

Using equation C.8 in equation C.7 gives:

$$\sigma_x^2 \cdot w_i^{opt} = E(x(n-i) \cdot \sum_{j=0}^{\infty} x(n-j) \cdot h_d(j))$$

$$
\begin{aligned}
&= E(\sum_{j=0}^{\infty} x(n-i)\cdot x(n-j)\cdot h_d(j)) \\
&= \sum_{j=0}^{\infty} E(x(n-i)\cdot x(n-j)\cdot h_d(j)) \\
&= \sum_{j=0}^{\infty} h_d(j)\cdot E(x(n-i)\cdot x(n-j)) \qquad \text{(C.9)}
\end{aligned}
$$

Since $x(n)$ is an uncorrelated white noise process, the following relationship holds:

$$
E(x(n-i)\cdot x(n-j)) = \begin{cases} \sigma_x^2, & i=j \\ 0, & i\neq j. \end{cases} \qquad \text{(C.10)}
$$

Due to the relationship of equation C.10, only one term in the summation of equation C.9 is non-zero. Using the relationship of equation C.10 in equation C.9 gives the desired result:

$$
\begin{aligned}
\sigma_x^2\cdot w_i^{opt} &= \sigma_x^2\cdot h_d(i). \\
w_i^{opt} &= h_d(i). \qquad \text{(C.11)}
\end{aligned}
$$

Thus for an uncorrelated white noise process equations C.2 and C.4 give the same optimum weight vector $\mathbf{W}^{opt}$.

Bibliography

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

[2] W. D. Gayler, *Telephone Voice Transmission: Standards and Measurement.* Englewood Cliffs, New Jersey: Prentice Hall.

[3] E. Hansler, "The hands-free telephone problem: An annotated bibliography update," *Annales des Télécommunications,* 1994, pp. 360-367.

[4] H. Yasukawa, M. Ogawa and M. Nishino, "Echo return loss required for an acoustic echo controller based on a subjective assessment," *IEICE Transactions,* vol. E-74, 1991, pp. 692-705

[5] M. M. Sondhi and W. Kellermann, "Adaptive echo cancellation for speech signals," *Advances in Speech Signal Processing,* (S. Furui and M. M. Sondhi, Eds.) New York: Marcel Dekker, 1992.

[6] P. Naylor, J. Alcazar, J. Boudy and Y. Grenier, "Enhancement of hands-free telecommunications," *Annales des Télécommunications,* 1994, pp. 373-379.

[7] M. E. Knappe, "Acoustic Echo Cancellation: Performance and Structures," M. Eng. Thesis, Carleton University, July 25, 1992.

[8] M. E. Knappe and R. Goubran, "Steady-state performance limitations of full-band acoustic echo cancellers", *Proc. IEEE ICASSP'94,* vol. II, pp. 73-76.

[9] T. G. D. Webster, "Tracking Performance of Acoustic Echo Cancellers", M. Eng. Thesis, Carleton University, 1995.

[10] H. Yuan, "Dynamic Behaviour of Acoustic Echo Cancellation", M. Eng. Thesis, Carleton University, 1994.

[11] S. M. Kuo and J. Chen, "New adaptive IIR notch filter and its application to howling control in speakerphone system," *Electronics Letters,* vol. 28, 1992, pp. 764-766.

[12] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[13] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[14] R. Bitmead and B. Anderson, "Performance ofadaptive estimation algorithms in dependent random environments," *IEEE Trans. on Automatic Control,* vol. AC-25, pp. 788-794.

[15] D. R. Morgan, "Slow asymptotic convergence of LMS acoustic echo cancellers," *IEEE Trans. on Speech and Audio Processing,* vol. 3, Mar. 1995, pp. 126-136.

[16] T. Ernst and A. Kaelin, "Analysis of the LMS algorithm with delayed coefficient update," *Proc. IEEE ISCAS'95,* Seattle, vol. II, pp. 1247-1250.

[17] M. Fukumoto, H. Kubota and S. Tsuji, "Improvement in stability and convergence speed on normalized LMS algorithm," *Proc. IEEE ISCAS'95,* Seattle, vol. II, pp. 1243-1246.

[18] J. T. Cilke and D. M. Etter, "A new adaptive algorithm to reduce weight fiuctuations caused by high variance data," *IEEE Trans. on Signal Processing,* vol. 40, no. 9, Sept. 1992, pp. 2324-2327.

[19] G. Clark, S. Mitra and S. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-29, June 1982, pp. 744-752.

[20] S. Makino, Y. Kaneda and N. Koizumi, "Exponentially weighted stepsize NLMS adaptive filter based on the statistics of a room impulse response," *IEEE Trans. on Speech and Audio Processing*, vol. 1, Jan. 1993, pp. 101-108.

[21] M. Mboup, M. Bonnet and N. Bershad, "LMS coupled adaptive prediction and system identification: A statistical model and transient mean analysis," *IEEE Trans. on Signal Processing*, vol. 42, Oct. 1994, pp. 2607-2615.

[22] R. Frenzel and M. E. Hennecke, "Using prewhitening and stepsize control to improve the performance of the LMS algorithm for acoustic echo compensation," *Proc. IEEE ISCAS'92*, pp. 1930-1932.

[23] M. de Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," *Proc. IEEE ICASSP'95*, Detroit, vol. II, pp. 985-988.

[24] A. N. Birkett and R. A. Goubran, "Acoustic echo cancellation using NLMS-Neural network structures", *Proc. IEEE ICASSP'95*, Detroit, vol. V, pp. 3035-3038.

[25] P. L. Feintuch, N. J. Bershad and A. K. Lo, "A frequency domain model for filtered LMS algorithms - stability analysis, design and elimination of the training mode," *IEEE Trans. on Signal Processing*, vol. 41, No. 4, April 1993, pp. 1518-1531.

[26] S. D. Snyder and C. H. Hansen, "The effect of transfer function estimation errors on the filtered-X LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 42, no. 4, April 1994, pp. 950-953.

[27] E. Bjarnason, "Analysis of the filtered-X LMS algorithm," *Proc. IEEE ICASSP'93*, Minneapolis, vol. III, pp. 511-514.

[28] S. M. Kuo and J. Tsai, "Residual noise shaping technique for active noise control systems," *Journal of the Acoustical Society of America*, vol. 95, no. 3, March 1994, pp. 1665-1668.

[29] Q. Meek, *Private communication*, May 1, 1995.

[30] CCITT Recommendation 0.41.

[31] D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. on Communications*, vol. COM-26, 1978, pp.1439-1446.

[32] G. Carayannis, D. Manolakis and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-31, 1983, pp. 1394-1402.

[33] J. Cioffi and T. Kailath, "Fast, recursive least squares transversal filters for adaptive filtering," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-32, 1984, pp. 304-337.

[34] G. Moustakides and S. Theodoris, "Fast Newton transversal filters - A new class of adaptive estimation algorithms," *IEEE trans. on Signal Processing*, vol. 39, Oct. 1991, pp. 2184-2193.

[35] H. Schütze, "Stabilized fast adaptation algorithms for acoustic echo control," *Proc. IEEE ISCAS'92*, pp. 1926-1929.

[36] A. Gilloire, T. Pétillon and S. Theodoridis, "Acoustic echo cancellation using fast RLS adaptive filters with reduced complexity," *Proc. IEEE ISCAS'92*, pp. 2065-2068.

[37] N. Bershad and O. Macchi, "Comparison of RLS and LMS algorithms for tracking a chirped signal," *Proc. IEEE ICASSP'89*, Glasgow, pp. 896-899.

[38] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady state performance of RLS adaptive filter algorithms," *IEEE Trans. on Acoustics, Speech and Signal Processing,* vol. ASSP-34, 1986, pp. 1097-1110.

[39] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEE Trams. on Signal Processing,* vol. 43, Feb. 1995, pp. 444-453.

[40] J. J. Shynk, "Frequency domain and multirate adaptive filtering," *IEEE Signal Processing Magazine,* pp. 14-37, Jan. 1992.

[41] S. S. Narayan, A. M. Peterson and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. on Acoust., Speech, and Signal Processing,* vol. ASSP-31, pp. 609-615, June 1983.

[42] M. R. Petraglia and S. K. Mitra, "Adaptive FIR filter structure based on the generalized subband decomposition of FIR filters," *IEEE Trans. on Circuits and Systems,* vol. 40, pp. 354-361, June 1993.

[43] H. Ochi and N. J. Bershad, "A new frequency domain LMS adaptive filter with reduced sized FFT's," *Proc. IEEE ISCAS'95,* Seattle, vol. III, pp. 1608-1611.

[44] M. J. T. Smith and T. P. Barnwell, III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-34, June 1986, pp. 434-441.

[45] M. J. T. Smith and T. P. Barnwell, III, "A unifying framework for analysis/synthesis systems based on maximally decimated filter banks," *Proc. of IEEE ICASSP'85,* Tampa, pp. 521-524.

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

[47] P. P. Vaidyanathan, "Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect reconstruction prop-

erty," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-35, Apr. 1987, pp. 476-492.

[48] A. Gilloire, "Experiments with subband acoustic echo cancellers for teleconferencing," *Proc. IEEE ICASSP'87,* Dallas, pp. 2141-2144.

[49] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands," *Proc. IEEE ICASSP'88,* New York, pp. 1572-1575.

[50] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and applications to acoustic echo cancellation," *IEEE Trans. on Signal Processing,* vol. 40, Aug. 1992, pp. 1862-1875.

[51] W. Kellermann. "Analysis and design of multirate systems for cancellation of acoustical echoes," *Proc. IEEE ICASSP'88,* New York, pp. 2570-2573.

[52] J. Chen, H. Bes, J. Vandewalle and P. Janssens, "A new structure for subband acoustic echo cancellers," *Proc. IEEE ICASSP'88,* New York, pp. 2574-2577.

[53] M. R. Petraglia and S. K. Mitra, "Performance analysis of adaptive filter structures based on subband decomposition," *Proc. IEEE ISCAS'93,* Chicago, pp. 60-63.

[54] Q. Jin, K. M. Wong, Z. Q. Luo, "Design of an optimum wavelet for cancellation of long echoes in telephony," *IEEE Intl. Workshop on Time-Freq. and Time-Scale Analysis,* Oct. 1994, pp. 488-491.

[55] D. M. Etter, N. C. Carlson, V. A. Margo, and J. H. Gross, "Adaptive system modelling with critically decimated multirate filters," *Proc. IEEE ISCAS'93,* Chicago, pp. 68-71.

[56] K. Anandakumar and S. A. Kassam, "Adaptive filtering based on wavelet decomposition," *IEEE Intl. Workshop on Time-Freq. and Time-Scale Analysis,* Oct. 1994, pp. 260-263.

[57] Z. Ma, K. Nakayama and A. Sugiyama, "Automatic tap assignment in subband adaptive filter," *IEICE Trans. on Communications*, vol. E76-B, July 1993, pp. 751-754.

[58] A. Sugiyama and F. Landais, "A new adaptive intersubband tap assignment algorithm for subband adaptive filters," *Proc. IEEE ICASSP'95*, Detroit, vol. V, pp. 3051-3054.

[59] P. L. De León II and D. M. Etter, "Experimental results with increased bandwidth analysis filters in oversampled, subband acoustic echo cancellers," *IEEE Signal Processing Letters*, vol. 2, Jan. 1995, pp. 1-3.

[60] H. Yasukawa and S. Shimada, "An acoustic echo canceller using subband sampling and decorrelation methods," *IEEE Trans. on Signal Processing*, vol. 41, Feb. 1993, pp. 926-930.

[61] B. Hätty, "Recursive least squares algorithms using multirate systems for cancellation of acoustical echoes," *Proc. IEEE ICASSP'90*, pp. 1145-1148.

[62] O. Tanrikulu, B. Baykal, A. G. Constantinides, J. A. Chambers and P. A. Naylor, "Finite-precision design and implementation of all-pass polyphase networks for echo cancellation in subbands," *Proc. IEEE ICASSP'95*, Detroit, vol. V, pp. 3039-3042.

[63] M. R. Petraglia, "Convergence analysis of a subband adaptive filter structure," *Proc. IEEE ISCAS'95*, Seattle, vol. II, pp. 1251-1254.

[64] H. Ochi, Y. Higa and S. Kinjo, "A subband adaptive filter bank with the optimum analysis filter bank," *Proc. IEEE ICASSP'95*, Detroit, vol. II, pp. 993-996.

[65] H. Ding, *Private Communication*, May 29, 1995.

[66] E. Diethorn, "Perceptually optimum adaptive filter tap profiles for subband acoustic echo cancellers," *Proc. ICSPAT'95*, Boston, vol. I, pp. 290-293.

[67] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, Apr. 1989, pp. 4-21.

[68] S. Gudvangen and S. J. Flockton, "Comparison of of pole-zero and all-zeros modelling of acoustic transfer functions," *Electronics Letters,* vol. 28, 1992, pp. 1976-1978.

[69] S. Zimmermann and G. A. Williamson, "Performance properties of fixed pole adaptive filters," *Proc. ISCAS'93,* Chicago, pp. 56-59.

[70] Y. Haneda, S. Makino and Y. Kaneda, "Common acoustical pole and zero modelling of room transfer functions," *IEEE Trans. on Speech and Audio Processing,* vol. 2, Apr. 1994, pp. 320-328.

[71] J. Chao, S. Kawabe and S. Tsuji, "A new IIR adaptive echo canceller: GIVE," *Proc. IEEE ICSE'92,* pp. 547-551.

[72] R. N. J. Veldhuis, M. Breeuwer and R. G. van der Waal, "Subband coding of digital audio signals," *Philips Journal of Research,* vol. 44, 1989, pp. 329-343.

[73] J. D. Johnston, "Transform coding of audio signals using perceptual noise criterion," *IEEE Journal on Selected Areas in Communications,* vol. 6, Feb. 1988, pp. 314-323.

[74] D. Sinha and A. H. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Trans. on Signal Processing,* vol. 41, Dec. 1993, pp. 3463-3479.

[75] N. Jayant, J. Johnston and R. Safranik, "Signal compression based on models of human perception," *Proceedings of the IEEE,* vol. 81, no. 10, Oct. 1993, pp. 1385-1422.

[76] A. Gersho, "Advances in speech and audio compression," *Proceedings of the IEEE,* vol. 82, no. 6, June 1994, pp. 900-918.

[77] A. S. Spanias, "Speech coding: A tutorial review," *Proceedings of the IEEE,* vol. 82, no. 10, Oct. 1994, pp. 1541-1582.

[78] E. Zwicker and R. Feldtkeller. *Das Ohr als Nachrichtenempfanger*. Hirzel, Stuttgart, 1967.

[79] E. Zwicker and U. T. Zwicker, "Audio engineering and psycho-acoustics. Matching signals to the final receiver, the human auditory system." *Journal of the Audio Engineering Society*, vol. 39, no. 3, Mar. 1991, pp. 115-126.

[80] K. Mayyas and T. Aboulnasr, *Private Communication*, December 1995.

[81] D. dal Farra, *Private Communication*, October 1995.