

University of Windsor

Scholarship at UWindsor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2009

Efficient time delay estimation and compensation applied to the cancellation of acoustic echo

Tomi Hila

University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Hila, Tomi, "Efficient time delay estimation and compensation applied to the cancellation of acoustic echo" (2009). *Electronic Theses and Dissertations*. 7938.

<https://scholar.uwindsor.ca/etd/7938>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

NOTE TO USERS

This reproduction is the best copy available.

UMI

Efficient Time Delay Estimation and Compensation

Applied to the Cancellation of Acoustic Echo

by

Tomi Hila

A Thesis

Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2009

© 2009 Tomi Hila



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-57585-7
Our file Notre référence
ISBN: 978-0-494-57585-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

■ ♦ ■
Canada

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

The system identification problem is notably dealt with using adaptive filtering approaches. In many applications the unknown system response consists of an initial sequence of zero-valued coefficients that precedes the “active” part of the response. The presence of these coefficients introduces a flat delay in the incoming signals which can take significantly large values. When most adaptive approaches attempt to model such a system, the presence of flat delay impairs their operation and performance. The approach introduced in this thesis aims to model the flat delay and “active” part of the unknown system separately. An efficient system for time delay estimation (TDE) is introduced to estimate the flat delay of an unknown system. The estimated delay is then compensated within the adaptive system thus allowing the latter to cover the active part of the unknown system. The proposed system is applied to the Acoustic Echo Cancellation (AEC) problem.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. Kwan for his guidance, support and encouragement. During the period that I have worked under his direction, he has been a major influence in my development as a graduate student. Dr. Kwan has directed my research efforts and stimulated my interest in several areas – including time delay estimation and compensation - through his advice and instruction. I would also like to thank my internal reader, Dr. Narayan Kar and my external examiner Dr. Robin Gras for their valuable feedback. Many thanks also to Ms. Andria Turner for her prompt help and support.

My appreciation also goes to the students and friends at the University of Windsor, especially the people at the ISP Lab. They have been a constant source of knowledge, support and entertainment. I wish the best to all of them in their future endeavours.

Last but not least, I take this chance to express my profound appreciation to my family. None of this would be possible without their love and sacrifice.

TABLE OF CONTENTS

AUTHOR'S DECLARATION OF ORIGINALITY	iii
ABSTRACT	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF SYMBOLS	xiv
LIST OF ACRONYMS	xvi
 1. INTRODUCTION.....	 1
1.1 Background and Problem Specification	1
1.1.1 System Identification and Acoustic Echo Cancellation	1
1.1.2 Properties of Acoustic Channels and the Need for Time Delay Estimation ..	5
1.1.3 Properties of Speech Signals	8
1.2 Motivation	9
1.3 Objective.....	9
 2. TIME DELAY ESTIMATION.....	 10
2.1 Time Delay Estimation via Cross-Correlation	11
2.2 Time Delay Estimation via Generalized Cross Correlation	13

2.2.1 Prefiltering	15
2.3 Time Delay Estimation in Acoustic Echo Cancellation - Reverberation	18
3. ADAPTIVE FILTERING FOR ACOUSTIC ECHO CANCELLATION.....	26
3.1 Adaptive Filtering for Single Channel Acoustic Echo Cancellation.....	26
3.2 Least Mean Squares Adaptive Filter	27
3.3 The Normalized Least Mean Squares Adaptive Filter	31
3.4 Subband Adaptive Filtering.....	32
3.4.1 Subband Filter Structures for SAF	34
3.5 Tracking of Echo Path Impulse Responses	38
4. EFFICIENT TIME DELAY ESTIMATION FOR REAL TIME	
OPERATION.....	44
4.1 Parallel Search of Cross Correlation Values	44
4.2 Preprocessing.....	47
4.3 Structures for Time Delay Estimation.....	53
4.4 Rough Estimate of the Delay.....	54
4.5 Exact Estimate of the Delay	59
4.6. Theoretical Analysis.....	62
4.7. Operation and Robustness	64
4.8. Computational Cost.....	66
4.9 Combination of the TDE Schemes with Adaptive Filtering Algorithms	68
4.9.1 TDE and Compensation in Series with Adaptive Filtering	68

4.9.2 TDE and Compensation Exploiting SAF Structures	70
5. SIMULATIONS.....	72
5.1 Performance Measures	72
5.2 Test Signals.....	74
5.3 Acoustic Echo Path Responses.....	76
5.4 Simulation Results.....	77
5.4.1 Performance Comparison of TDE Algorithms.....	78
5.4.2 Time Delay Compensation with Adaptive Filtering.....	84
5.4.3 Real Time Operation	90
6. CONCLUSIONS AND FUTURE WORK.....	95
APPENDICES.....	98
A. Statistical Preliminaries	98
REFERENCES.....	106
VITA AUCTORIS	112

LIST OF TABLES

Table 2-1 GCC weighing functions	17
Table 3-1 The Gradient Descent algorithm.....	30
Table 3-2 The LMS algorithm	31
Table 3-3 Computational complexity of fullband and subband adaptive filters.....	37
Table 4-1 Rough estimate of the Time Delay – steps.....	57
Table 5-1 Percentage of correct identification of the optimal delay.....	84

LIST OF FIGURES

Figure 1-1 The System Identification problem	1
Figure 1-2 SISO System Identification using adaptive filtering.....	2
Figure 1-3 Setup for acoustic echo cancellation	4
Figure 1-4 Impulse response of room with low reverberation.....	6
Figure 1-5 Impulse response of highly reverberant room	6
Figure 1-6 Compensation of the flat delay into the signal path	8
Figure 2-1 Cross correlation function between delayed white noise signals.....	12
Figure 2-2 Cross correlation function for delayed speech signals.....	13
Figure 2-3 GCC for delayed speech signals.....	18
Figure 2-4 Propagation of sound in acoustic environment	19
Figure 2-5 Cross correlation between filtered white noise signals	21
Figure 2-6 Cross correlation between filtered speech signals.....	22
Figure 2-7 GCC for filtered speech – reverberation present	22
Figure 2-8 Cepstral prefiltering for TDE	23
Figure 2-9 TDE based on AED.....	24
Figure 3-1 AEC with adaptive filtering	27
Figure 3- 2 Analysis (right) and synthesis (left) filter banks	32
Figure 3-3 Subband adaptive filtering	33
Figure 3-4 Analysis (top) and synthesis (bottom) GDFT modulated filter banks	36
Figure 3-5 Echo path response (to be estimated)	38

Figure 3-6 Estimated echo path response using 2000 sample long FIR NLMS	39
Figure 3-7 Estimated echo path response using 512 sample long FIR NLMS	39
Figure 3-8 Estimated echo path response using 256 sample long FIR NLMS	40
Figure 3-9 Flat delay estimation and compensation for AEC.....	41
Figure 3-10 Estimated echo path response using 2000 sample long FIR NLMS with delay compensation	41
Figure 3-11 Estimated echo path response using 512 sample long FIR NLMS with delay compensation	42
Figure 3-12 Estimated echo path response using 256 sample long FIR NLMS with delay compensation	42
Figure 4-1 Simplified diagram of the proposed TDE scheme	45
Figure 4-2 Result of TDE when choosing wrong search interval (Case 1)	46
Figure 4-3 Result of TDE when optimal delay lies between two consecutive checked delays (Case 2)	47
Figure 4-4 Serial (top) vs. parallel (bottom) search for TDE.....	48
Figure 4-5 Preprocessing	49
Figure 4-6 Frequency domain result of downsampling for $M=2$	50
Figure 4-7 Result of decimation	50
Figure 4-8 Original speech signal	52
Figure 4-9 Spectrum of the original signal	52
Figure 4-10 Frequency response of the decimation filter $B(\omega)$	52
Figure 4-11 Spectrum of the preprocessed signal	53

Figure 4-12 TDE scheme for approximating the delay to the nearest M samples.....	58
Figure 4-13 Structure for estimating the correlation values	58
Figure 4-14 Structure for estimating the delay to the nearest sampling period.....	61
Figure 4-15 Structure for estimating th cross correlation values.....	62
Figure 4-16 Result of TDE (first algorithm) when M is excessively large.....	65
Figure 4-17 Adaptive filtering with TDE and time delay compensation.....	69
Figure 4-18 TDE combined with SAF for AEC.....	70
Figure 5-1 Recorded speech signal	74
Figure 5-2 Speech signal <code>dft_speech.wav</code>	75
Figure 5-3 Spectrogram of the speech signal in Figure 5-1	76
Figure 5-4 Experiment setup to test TDE algorithm performance.....	78
Figure 5-5 TDE algorithm performance – cross correlation.....	80
Figure 5-6 TDE algorithm performance - GCC.....	80
Figure 5-7 TDE algorithm performance – proposed (M=8)	80
Figure 5-8 TDE algorithm performance – proposed (M=10)	81
Figure 5-9 TDE algorithm performance – proposed (M=16)	81
Figure 5-10 TDE algorithm performance – cross correlation.....	82
Figure 5-11 TDE algorithm performance – GCC	82
Figure 5-12 TDE algorithm performance – proposed (M=8)	82
Figure 5-13 TDE algorithm performance – proposed (M=10)	83
Figure 5-14 TDE algorithm performance – proposed (M=16)	83

Figure 5-15 Average ERLE as a function of the compensated delay	85
Figure 5-16 Maximum ERLE as a function of the compensated delay	85
Figure 5-17 Average and maximum ERLE as a function of the adaptive filter length (fullband)	86
Figure 5-18 Average and maximum ERLE as a function of the adaptive filter length (subband)	87
Figure 5-19 Average ERLE as a function of the compensated delay	88
Figure 5-20 Maximum ERLE as a function of the compensated delay	89
Figure 5-21 Average ERLE as a function of the compensated delay	89
Figure 5-22 Maximum ERLE as a function of the compensated delay	90
Figure 5-23 Operation of the proposed scheme for rough delay estimation.....	91
Figure 5-24 Operation of the proposed scheme for rough delay estimation – detailed....	92
Figure 5-25 Operation of the proposed scheme for exact delay estimation.....	93
Figure 5-25 Operation of the proposed scheme for exact delay estimation.....	93
Figure 5-26 Operation of the proposed scheme for exact delay estimation – detailed.....	94

LIST OF SYMBOLS

Symbol	Definition
t	Continuous time index
n	Discrete time index
$h(n)$	Impulse response of filter
$H(\omega)$	Frequency response of filter
$w(n)$	Noise signal
$r(n)$	Echo signal
$s(n)$	Speech signal
$\delta(n)$	Kroenecker delta function
$x(n)$	Discrete signal
$x_d(n)$	Delayed (and filtered) version of $x(n)$
Δ	Delay (in samples)
$E\{ \cdot \}$	Expected value operator
$R_{xx_d}(n, d)$	Cross correlation function between $x(n)$ and $x_d(n)$ calculated at time instant n for a delay d
$\rho_{x_d x}(n, k)$	Cross correlation coefficient, or normalized cross correlation
$\Phi_{yy_d}(n, \omega)$	Cross power spectral density between $y(n)$ and $y_d(n)$ calculated at time instant n and frequency ω
$\Psi(\omega)$	Frequency domain weighing function for GCC prefiltering
$J(n)$	Cost function
$\nabla\{ \cdot \}$	Gradient operator

$\{\cdot\}^T$	Transpose operator
$H_k(\omega)$	k -th analysis filter (for subband adaptive filter)
$G_k(\omega)$	k -th synthesis filter (for subband adaptive filter)
L_p	Length of analysis and synthesis filters
M	Number of channels (for subband adaptive filter)
D	Downsampling factor
m	Discrete time index of the preprocessed signals
$x'(m)$	Prefiltered input signal $x(n)$
$x'_d(m)$	Prefiltered desired signal $x_d(n)$
$B(\omega)$	Bandpass filter utilized for prefiltering
$\Delta_{\text{opt1}}MT_s$	Optimal delay to be estimated by the first proposed TDE structure
$\Delta_{\text{opt2}}T_s$	Optimal delay to be estimated by the second proposed TDE structure
α, β	Smoothing constants (for correlation function)
K	Number of iterations (for TDE algorithms)
P, L	Number of delays to be checked for optimality (for TDE algorithms)

LIST OF ACRONYMS

Acronym	Definition
TDE	Time Delay Estimation
AEC	Acoustic Echo Cancellation
AF	Adaptive Filter
SISO	Single Input Single Output
MIMO	Multiple Inputs Multiple Outputs
MISO	Multiple Inputs Single Output
SIMO	Single Input Multiple Outputs
FIR	Finite Impulse Response
LEM	Loudspeaker Enclosure Microphone
IIR	Infinite Impulse Response
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
SNR	Signal-to-Noise Ratio
ERLE	Echo Return Loss Enhancement
MSE	Mean Square Error
LMS	Least Mean Squares
NLMS	Normalized Least Mean Squares
GCC	Generalized Cross Correlation
PSD	Power Spectral Density
AEVD	Adaptive Eigen Value Decomposition

CEP-GCC	Cepstral GCC Method (for TDE)
ATDE	Adaptive Time Delay Estimation
FB	Filter Banks
MDFB	Modulated DFT Filter Bank
GDFT	Generalized DFT
MGDFB	Generalized DFT Modulated Filterbanks
SAF	Subband Adaptive Filter
NLMS-TDE	Combined NLMS and TDE Algorithm
SNLMS-TDE	Combined Subband NLMS and TDE Algorithm
pmf	Probability Mass Function
pdf	Probability Density Function
CDF	Cumulative Distribution Function
WSS	Wide Sense Stationary

CHAPTER 1

INTRODUCTION

1.1 Background and Problem Specification

1.1.1 System Identification and Acoustic Echo Cancellation

System identification refers to the procedure of dynamically estimating the parameters of an unknown time-varying system [1] when only the input and the output (plus additive noise) to the unknown system are measurable (Figure 1.1).

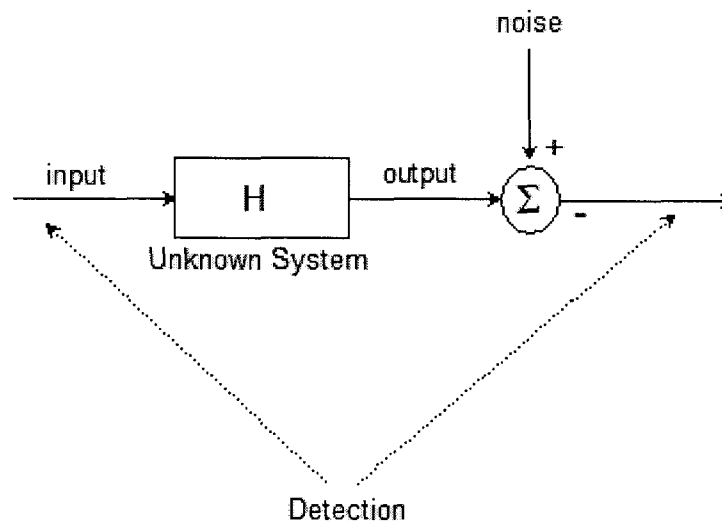


Figure 1-1 The System Identification problem

The unknown system in Figure 1-1 can be a transmission channel, an acoustic environment etc. The diagram in Figure 1.1 depicts a Single Input Single Output (SISO) system. Other arrangements include: Multiple Input Single Output (MISO), Single Input Multiple Output (SIMO) and Multiple Input Multiple Output (MIMO). One popular

approach to estimate the unknown system involves adaptive filtering techniques. In these approaches, the response of the unknown channel is modeled using a digital filter. The parameters of the digital filter are found such as to provide the best fit to the unknown system behavior. In [38] as well as in most adaptive filtering techniques, finite impulse response (FIR) filters are used. The coefficients of such filters are dynamically optimized using gradient-based iterative techniques. A typical scenario is depicted in Figure 1-2.

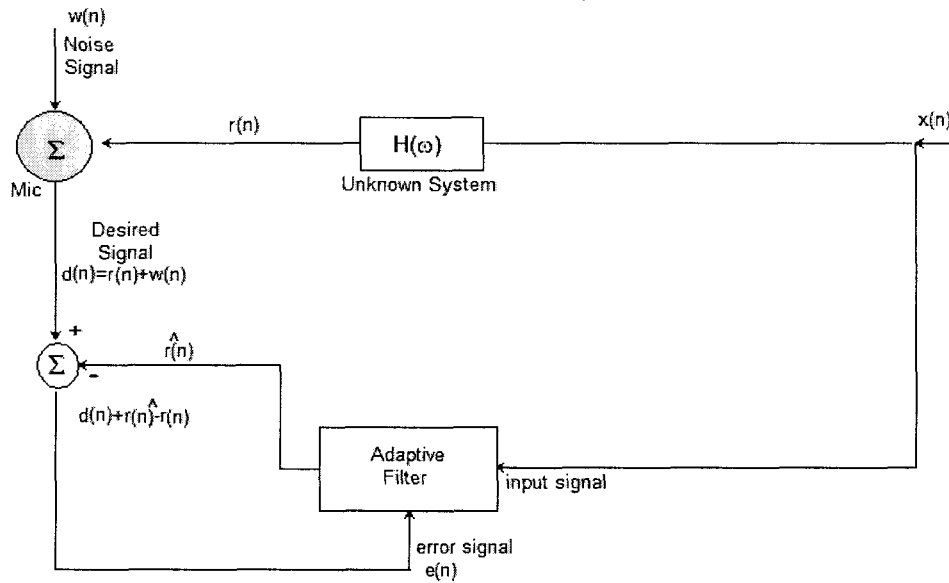


Figure 1-2 SISO System Identification using adaptive filtering

The aim of the adaptive filter in Figure 1-2 is to model the unknown channel such that the error signal is minimized. In this way, the output of the filter $\hat{r}(n)$ tries to estimate the output $r(n)$ of the unknown system. Notice that both the adaptive filter and the unknown system are driven by the same input, while the output of the unknown system serves as the “desired” output of the adaptive filter.

The performance of the adaptive system is usually measured in terms of the following parameters [2]:

Convergence speed – speed with which the adaptive filter reaches the desired solution;

1. *Misadjustment* – amount by which the final value of the mean squared error (MSE) deviates from the MSE produced by the optimal Wiener filter;
2. *Tracking* – ability to track statistical variations of the unknown system;
3. *Robustness* – resistance to small disturbances;
4. *Computational requirements* – depends on number of operation required to complete a single iteration of the algorithm and memory requirements.

Several factors decide the performance of the adaptive algorithms such as [2]:

1. The type of adaptive approach used (LMS, RMS, time vs. frequency vs. subband domain)
2. The nature of the unknown channel (long vs. short impulse response, linear vs. nonlinear, time varying vs. time invariant, etc)
3. The nature of the signals (white vs. coloured, wideband vs. narrowband, time varying vs. time invariant, etc).

Acoustic Echo Cancellation (AEC) is a scenario where system identification techniques are applied to estimate the response of an unknown acoustic environment, or acoustic path. The acoustic environment to be modeled is often termed the loudspeaker-enclosure-microphone (LEM) system (Figure 1-3). Acoustic echo occurs in cases when a direct path between the transmitter (microphone) and receiver (loudspeaker) exists. In such a case, the signal propagates from the receiver, through the acoustic environment and back to the transmitter. Some of the applications when acoustic echo is present are teleconferencing, VOIP and other hands-free applications [3]-[9], [12]-14], [18], [35]-[37]. In addition,

such applications introduce into the signal delays that are inherent to the communication system. A general configuration for single-channel AEC is shown in Figure 1-3.

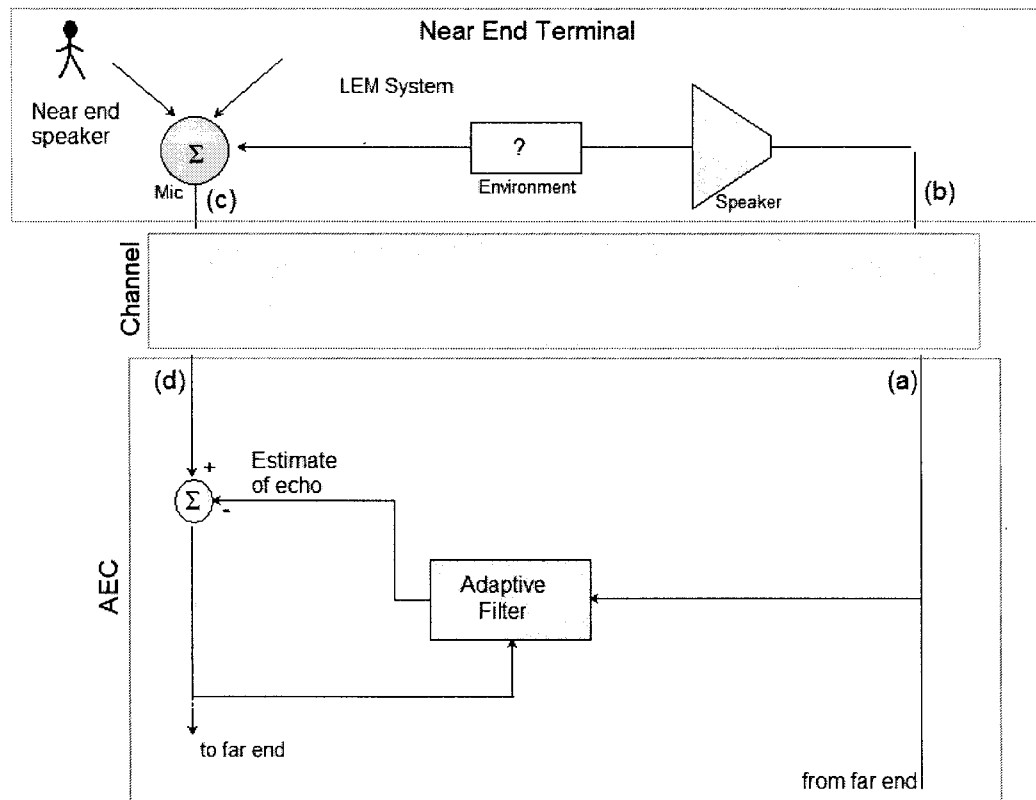


Figure 1-3 Setup for Acoustic Echo Cancellation

The echo signal corrupts the quality of the communication since it is transmitted back to the far end speaker. An AEC system is employed to estimate the echo signal and subtract it from the signal received at the microphone.

1.1.2 Properties of Acoustic Channels and the Need for Time Delay Estimation

The acoustic channels considered in this thesis are represented by a long impulse response - sometimes thousands of samples long - and are typical for medium to large-sized rooms [10], [11]. Two typical impulse responses of acoustic environments are shown in Figures 1-4 and 1-5. The acoustic impulse responses of Figure 1-4 and 1-5 can be divided by observation into three sections. The first section is composed of consecutive zero-valued coefficients. The presence of these coefficients introduces the so-called *flat delay* [10] into the input signal (see Figure 1-5). In many applications such as teleconferencing and VOIP, the flat delay can take very large values due to the channel delays, packetizing jitter buffering etc. The next section in the impulse response is composed of only a few coefficients that have relatively large values. These coefficients represent the direct path as well as early reflections between the receiver and the transmitter which are located at each boundary of the echo path. The last section is composed of a large number of quasi-exponentially-decreasing coefficients. These coefficients are due to reflections as well as absorption and attenuation of the signal from the loudspeaker caused by the walls and other objects that are present within the acoustic environment. As a result of these coefficients, the acoustic channel introduces *reverberation* which causes spectral distortion of the signals [12], [18]. It can be deduced by visual inspection that the echo path signal in Figure 1-5 has higher reverberation than the one in Figure 1-4, i.e. the environment is more reverberant. The two acoustic echo path responses in Figures 1-4 and 1-5 are the ones used for simulations throughout this thesis.

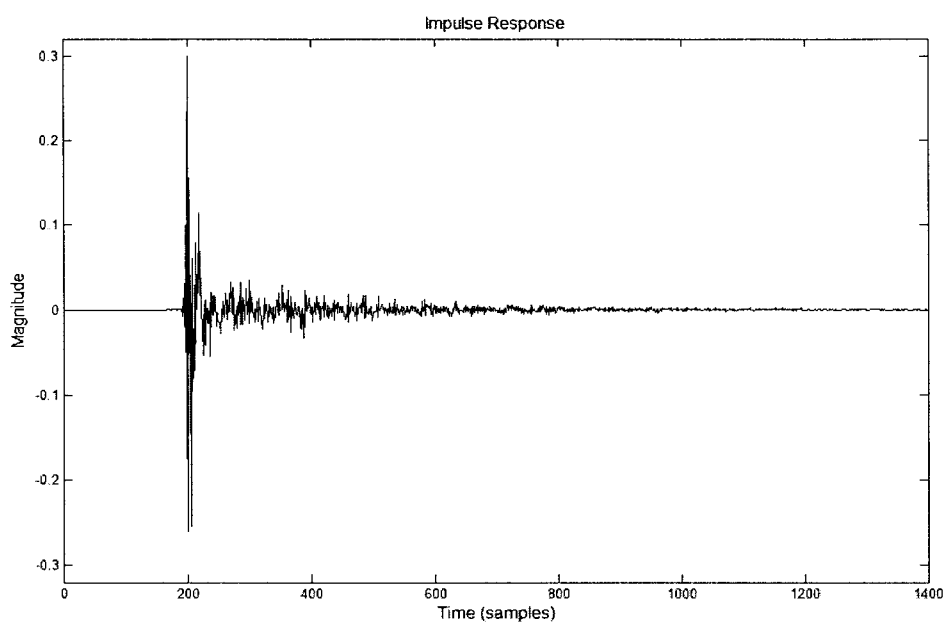


Figure 1-4 Impulse response of room with low reverberation.

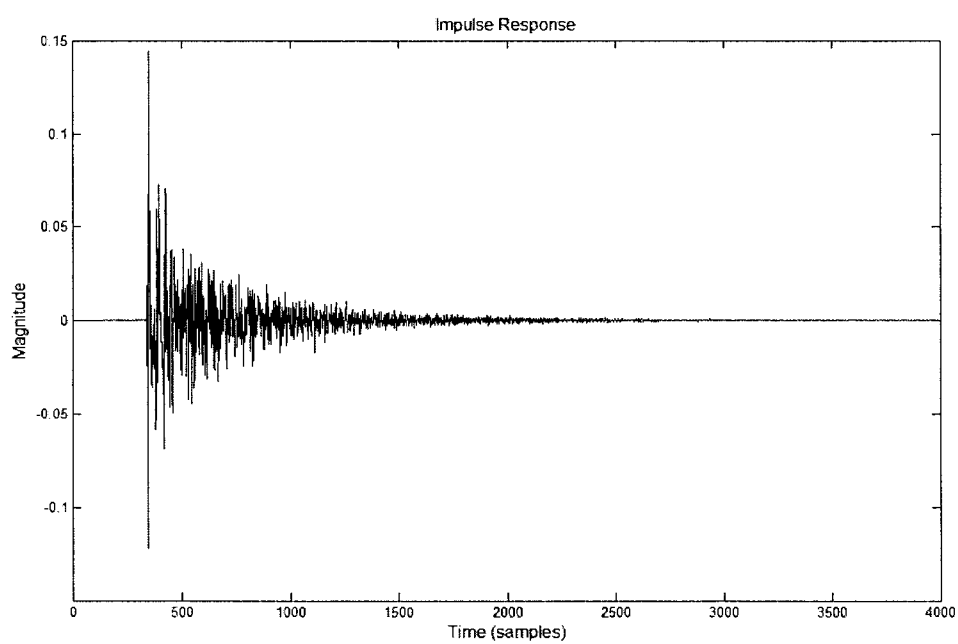


Figure 1-5 Impulse response of highly reverberant room.

When modeling acoustic responses (echo paths) such as the ones in Figures 1-4 and 1-5, one of the main obstacles that adaptive filtering techniques are faced with is the possible excessive length of the acoustic echo path response. In such a scenario, a FIR adaptive filter could need thousands of coefficients to achieve the desired mean square error (MSE) performance. If a flat delay is also present, the FIR adaptive filter will “spend resources” trying to model this section of the echo path; furthermore this sections is not usually modeled exactly as a pure delay due to the random nature of the coefficient search in the adaptive filters.

The approach introduced in this thesis aims at estimating the flat delay present in the echo path and incorporating this delay in the signal path accordingly such that the reference signal is “centered” with respect to the input signal. In this way the adaptive filter will only have to model the active part of the impulse response of the unknown channel (Figure 1-6).

The impulse response of acoustic environments such as the ones treated in this thesis can change due to minimal variations in the position of receivers, transmitters and objects that are placed within the environment [10], [12], [14]. This makes such systems highly time-varying. Hence the need for continuous estimation of the flat delay arises.

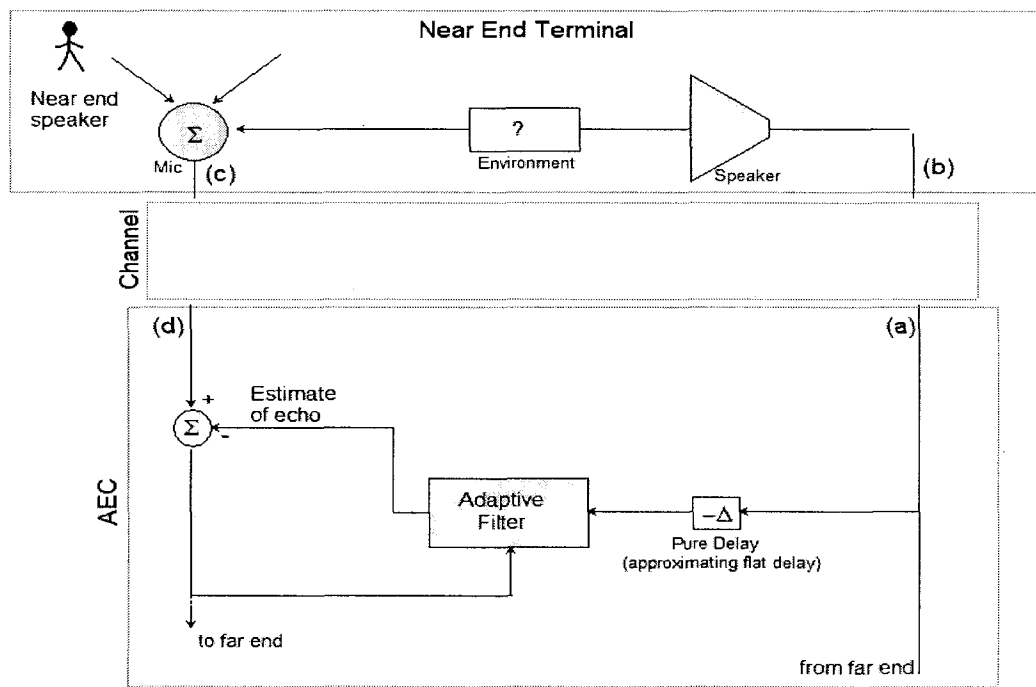


Figure 1-6 Compensation of the flat delay into the signal path

1.1.3 Properties of Speech Signals

As mentioned in Section 1.1.1, the performance of adaptive algorithms depends heavily on the nature of the input signals. In the AEC applications treated in this project, speech signals are mostly used. Identifying systems when speech signals are present is especially difficult [10] - [14].

Speech signals are wideband signals with frequency range 100 – 8000 Hz. The frequencies 300 – 3400 Hz are of main interest since they contain most of the information and energy of the signal. Speech signals are highly time-varying. Nonetheless, short-term stationarity may be assumed for intervals of about 20ms. Speech signals are also highly correlated, which poses additional problems for adaptive filtering and time delay estimation approaches [13], [14], [18].

1.2 Motivation

Although a number of approaches for TDE exist, many of them must compromise between complexity and performance. The scale of difficulty increases especially when factors such as reverberation, noise, coloured signals and time-variance need to be accounted for. When real-time operation constraints are introduced, the task becomes even more difficult. In acoustic echo cancellation, all of the constraints mentioned above are present.

The current project was motivated by the need to find a TDE scheme that combines robustness in the presence of noise and coloured signals with an aptitude for real-time operation – i.e. low complexity and high speed of convergence.

1.3 Objective

The objective of this thesis is to introduce a time delay estimation algorithm that can be applied in real time for speech signals and time-varying environments for the estimation of the flat delay present in acoustic environments. The proposed algorithm will operate in tandem with adaptive filtering techniques thus improving their performance and introducing computational savings.

CHAPTER 2

TIME DELAY ESTIMATION

Consider the two discrete signals:

$$x(n) \text{ and } x_d(n) = \alpha x(n - \Delta) + w(n) \quad (2-1)$$

where $w(n)$ is a noise signal uncorrelated with $x(n)$ and α is a constant. The purpose of a TDE system is to determine the value of the delay Δ when only $x(n)$ and $x_d(n)$ are given.

Note that:

$$x_d(n) = \alpha x(n) * \delta(n - \Delta) + w(n) \quad (2-2)$$

where $*$ denotes convolution. The scenario depicted in (2-1) and (2-2) represents the ideal situation when the signals in question are simply shifted versions of each other. A more general case – which is more applicable to real world situations – will be treated later on as this chapter progresses.

TDE is used in applications such as source localization [15], [16], [19]-[24] and echo cancellation [10], [11]. In the following sections, several techniques to estimate the optimal delay are presented. For an introduction of related statistical concepts such as cross correlation, power spectral density and so on, the reader is referred to Appendix A.

2.1 Time Delay Estimation via Cross-Correlation

When $x(n)$ and $x_d(n)$ in (2-1) are both real-valued white signals, the method of finding the delay Δ using cross-correlation is straight forward. Simply put, the optimal delay between two such signals is that for which their cross-correlation function is maximized. A simple derivation illustrates this point:

$$R_{xxd}(n, d) = E\{x(n-d)x_d(n)\} = E\{\alpha x(n-d)x(n-\Delta)\} + E\{x(n-d)w(n)\} = \alpha \delta(n-\Delta+d) \quad (2-3)$$

where $R_{xxd}(n, d)$ is the cross correlation function between $x(n)$ and $x_d(n)$, $E\{\cdot\}$ denotes the expected value operation and $\delta(n)$ is the Kroenecker delta function. It is evident from (2-3) that the cross correlation function above is maximized at $d=\Delta$.

In general, the cross correlation approach to TDE is applied by finding values of the cross correlation $R_{xxd}(n, d)$ between $x(n)$ and $x_d(n)$ for several values of d . The optimal value for the delay is the value of d for which the value of the cross correlation is maximized, that is:

$$\Delta = \arg_d\{\max(R_{xxd}(n, d))\} \quad (2-4)$$

The time index n is included in the expression for the cross correlation to imply that it can be time varying. Note that (2-4) is a formulation of a parameter estimation problem.

As discussed in Appendix A, for ergodic signals, ensemble averages can be approximated using time averages. We assume that the signals in this thesis are all created from ergodic processes, which is the case for many real-world signals. With that in mind, the cross correlations of interest are computed using the time averages:

$$R_{xx_d}(n, d) = E\{x(n-d)x_d(n)\} = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=-K/2}^{K/2} x(n-d-k)x_d(n-k) \quad (2-5)$$

The expression in (2-5) is not computable due to the infinite time window lengths. To make practical implementations possible, the time averaging should be performed over a finite time window of size K samples:

$$R_{xx_d}(n, d) = \frac{1}{K} \sum_{k=0}^{K-1} x(n-d-k)x_d(n-k) \quad (2-6)$$

The cross correlation between a white signal and its delayed version is shown in Figure 2-1. Notice that the correlation function is simply an impulse at the delay point.

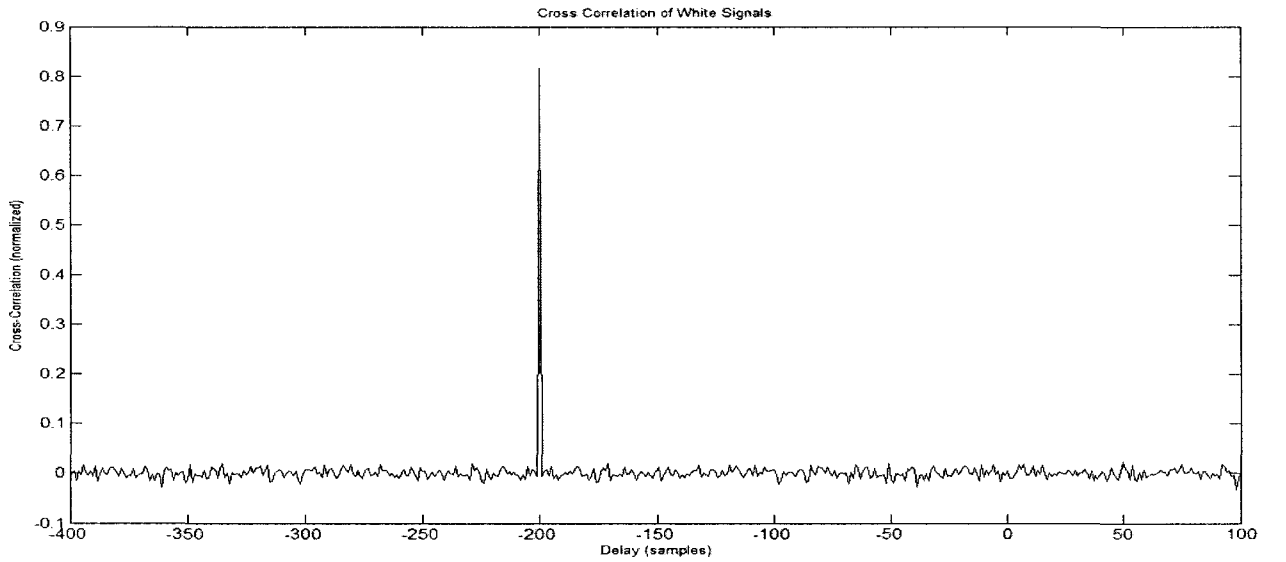


Figure 2-1 Cross correlation function between delayed white noise signals ($\Delta = 200$ samples)

2.2 Time Delay Estimation via Generalized Cross Correlation

When dealing with colored signals such as speech, the accuracy of TDE using cross correlation deteriorates significantly. The cross-correlation between a speech signal and its delayed version is a highly varying function with several maxima and sometimes the optimal delay cannot even be extracted [10], [19]. The cross correlation function between two speech signals – one of which is simply a delayed version of the other - is shown in Figure 2-2.

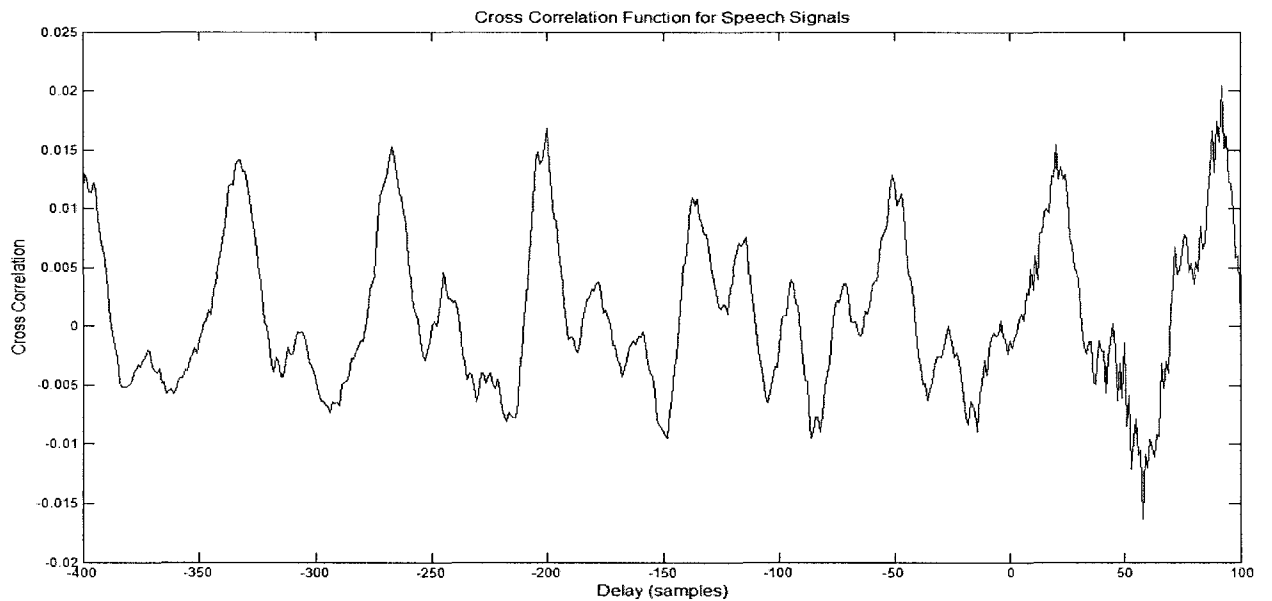


Figure 2-2 Cross correlation function for delayed speech signals ($\Delta = 200$ samples)

The cross correlation function in Figure 2-2 was calculated using (2-6) with a window of 62ms ($K = 5000$ samples, sampling rate $F_s = 8000\text{Hz}$). The optimal delay (at -200 samples) is not clear, and the task of estimating it is made even harder by the presence of other local maxima in its vicinity. Thus, when it comes to real-time implementations, the straight-forward cross correlation approach becomes unfeasible.

One proposed solution to the problem just presented is based on the Generalized Cross Correlation (GCC) technique [19], [20]. The GCC method is based on a prefiltering operation which is performed on each input signal. The prefiltering operation modifies the spectral characteristics of the signals and reduces the correlation between subsequent samples [12]. In a sense, the prefiltering operation can be seen as having the purpose of making the signals resemble white noise. The GCC method is the most popular method for TDE and is known to perform well in environments with moderate noise and low reverberation.

Given two speech signals $x(n)$ and $x_d(n) = ax(n-\Delta) + w(n)$, two new signals $y(n)$ and $y_d(n)$ are derived by prefiltering $x(n)$ and $x_d(n)$:

$$y(n) = x(n) * h_1(n) \quad (2-7a)$$

and:

$$y_d(n) = x_d(n) * h_2(n) = ax(n-\Delta) * h_2(n) + w(n) * h_2(n) \quad (2-7b)$$

where $\mathbf{h}_1 = [h_1(0), \dots, h_1(N_1-1)]$ and $\mathbf{h}_2 = [h_2(0), \dots, h_2(N_2-1)]$ are the coefficients of the prefilters. These prefilters are designed such that the delay between the resulting signals $y(n)$ and $y_d(n)$ is easier to detect compared to the delay between the initial signals $x(n)$ and $x_d(n)$. The GCC approach then consists of finding the value of the delay that maximizes the cross correlation between $y(n)$ and $y_d(n)$, that is:

$$\Delta = \arg_d \{ \max(R_{yy_d}(n, d)) \} \quad (2-8)$$

The following procedure helps in determining the properties of the filters \mathbf{h}_1 and \mathbf{h}_2 .

2.2.1 Prefiltering

Knowing that the delay between two signals is easier to estimate if they are similar to white noise, the prefiltering step aims at modifying the spectrum of $x(n)$ and $x_d(n)$ such that they are made to resemble the spectrum of white noise.

The cross correlation functions for each pair of signals can be approximated as follows:

$$R_{xx_d}(n, d) = \frac{1}{K} \sum_{k=0}^{K-1} x(n-d-k)x_d(n-k) \quad (2-9)$$

and:

$$R_{yy_d}(n, d) = \frac{1}{K} \sum_{k=0}^{K-1} y(n-d-k)y_d(n-k) \quad (2-10)$$

The cross power spectral density (PSD) of $y(n)$ and $y_d(n)$ can be approximated by using the periodogram (see Appendix A):

$$\Phi_{yy_d}(n, \omega) = \sum_{d=-\infty}^{\infty} R_{yy_d}(n, d)e^{-j\omega d} \approx \frac{Y_T(\omega)Y_{dT}(\omega)}{T} \quad (2-11)$$

In (2-11), T is the observation time and $Y_T(\omega)$ and $Y_{dT}(\omega)$ are the Fourier transforms of the signals $y(n)$ and $y_d(n)$ observed over a period T . Furthermore:

$$\begin{aligned}
\Phi_{yy_d}(n, \omega) &= \sum_{d=-\infty}^{\infty} R_{yy_d}(n, d) e^{-j\omega d} \approx \frac{Y_T(\omega) Y_{dT}(\omega)}{T} = \\
&= \frac{X_T(\omega) H_1(\omega) X_{dT}(\omega) H_2(\omega)}{T} = \\
&= \frac{H_1(\omega) H_2(\omega) X_T(\omega) X_{dT}(\omega)}{T} = \\
&= \frac{\Psi(\omega) X_T(\omega) X_{dT}(\omega)}{T} = \\
&= \Psi(\omega) \Phi_{xx_d}(n, \omega)
\end{aligned} \tag{2-12}$$

$X_T(\omega)$ and $X_{dT}(\omega)$ are the Fourier transforms of the signals $x(n)$ and $x_d(n)$ observed over a period T . $H_1(\omega)$ and $H_2(\omega)$ are the frequency responses of the filters \mathbf{h}_1 and \mathbf{h}_2 .

Equation (2-12) states that the cross PSD of the prefiltered signals $y(n)$ and $y_d(n)$ can be approximated by frequency-weighting the PSD $\Phi_{xx_d}(n, \omega)$ of the input signals. The frequency-domain weighing function $\Psi(\omega) = H_1(\omega) H_2^*(\omega)$ provides the relation between $\Phi_{xx_d}(n, \omega)$ and $\Phi_{yy_d}(n, \omega)$. Taking a step back, the cross correlation in (2-10) is the inverse Fourier transform of (2-11), that is:

$$\begin{aligned}
R_{yy_d}(n, d) &= \int_{-\infty}^{\infty} \Phi_{yy_d}(n, \omega) e^{j\omega d} d\omega = \\
&= \frac{1}{T} \int_{-\infty}^{\infty} \Psi(n, \omega) X_T(n, \omega) X_{dT}^*(n, \omega) e^{j\omega d} d\omega = \\
&= \frac{1}{T} \int_{-\infty}^{\infty} \Psi(n, \omega) \Phi_{xx_d}(n, \omega) e^{j\omega d} d\omega
\end{aligned} \tag{2-13}$$

The expression in (2-13) can be computed using DFT in real systems by changing the integration to a summation over a finite number of terms.

In order to perform GCC, the frequency responses of the prefilters $H_1(\omega)$ and $H_2(\omega)$ must be determined. Two factors help at determining $H_1(\omega)$ and $H_2(\omega)$:

1. The PSD $\Phi_{yy_d}(n, \omega)$ must be similar to that of white noise – which is a constant over all frequencies;
2. $H_1(\omega)$ and $H_2(\omega)$ are identical.

Many choices are available for prefilters depending on the desired result and the criteria to be optimized. Table 2-1 summarizes some of the possible choices.

Table 2-1 GCC Weighing Functions

Name	Expression for $\Psi(n, \omega)$
<i>Cross Correlation</i>	1
<i>PHAT</i>	$1/ \Phi_{xx_d}(n, \omega) $
<i>SCOT</i>	$1/\sqrt{\Phi_x(n, \omega)\Phi_{x_d}(n, \omega)}$
<i>Roth</i>	$1/\Phi_{x_d}(n, \omega)$

The PHAT prewhitening technique [19] is the method of choice for most applications of GCC. Figure 2-3 depicts the GCC function for the same speech signal as in Figure 2-2 (with a window of 62ms, $K = 5000$ samples, $F_s = 8000\text{Hz}$). The difference in performance is clear, but it comes at a higher computational cost.

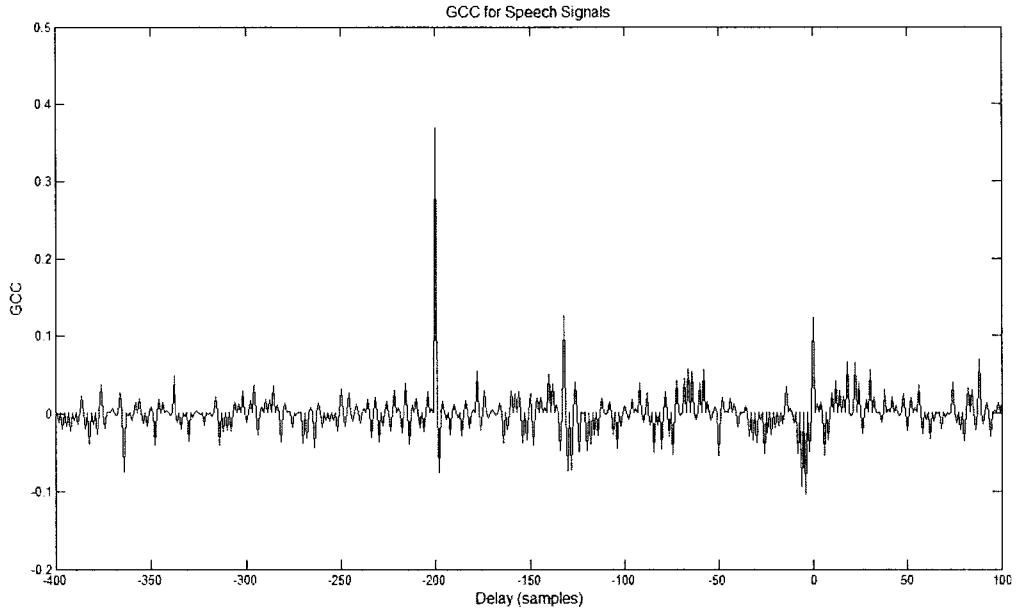


Figure 2-3 GCC for delayed speech signals

2.3 Time Delay Estimation in Acoustic Echo Cancellation - Reverberation

In the discussions so far, it was assumed that the two signals are simply delayed versions of each other (plus noise) having a relationship such as in (2-1). In a scenario such as AEC, and many other real-world situations, the original signal goes through reflections, attenuation and absorption effects caused by the environment (Figures 1-4, 1-5). This results in the final signal being a weighted sum of delayed versions of the original signal plus an added noise:

$$x_d(n) = \sum_{k=0}^{N-1} h(k)x(n-k) + w(n) = \mathbf{h}^T \mathbf{x}_n + w(n) \quad (2-14)$$

where $\mathbf{x}_n = [x(n), x(n-1), \dots, x(n-N+1)]^T$ is the input vector, and $\mathbf{h} = [h(0), h(1), \dots, h(N-1)]^T$ can be seen as the impulse response of the echo path.

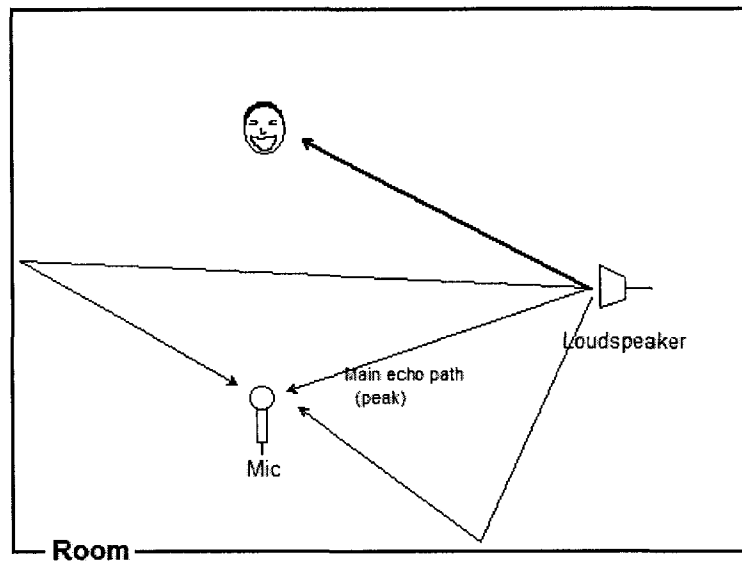


Figure 2-4 Propagation of sound through the acoustic environment

Thus a “linear filtering” operation such as the one in (2-14) is performed by the environment on the input signal. This filtering operation accounts for the following effects:

1. *Introduction of a flat delay on the input signal* – which is caused by the initial zero-valued coefficients present in the echo path impulse response. The value of the delay is the distance (in samples) between the first sample of the echo path response and the peak of this response. This peak corresponds to one of the first samples of the “active” section of the echo path response and is due to the direct path between the signal source (loudspeaker) and receiver (microphone) – see Figures 1-4 and 1-5. The value for this particular delay must be estimated.
2. *Reverberation* – is caused by the section of the echo path response composed of coefficients having quasi-exponentially-decreasing magnitude. Reverberation

makes it very difficult for techniques such as cross correlation or GCC to estimate the delay [12], [14], [16].

When the signals in question are white, one can proceed similarly to the derivation in (2-3):

$$\begin{aligned}
 R_{xx_d}(n, d) &= E\{x(n-d)x_d(n)\} = \\
 &= E\{x(n-d)\sum_{k=0}^{N-1} h(k)x(n-k)\} + E\{x(n-d)w(n)\} = \\
 &= \sum_{k=0}^{N-1} h(k)E\{x(n-d)x(n-k)\} + E\{x(n-d)w(n)\} = \\
 &= \sum_{k=0}^{N-1} h(k)\delta(n-k+d) = h'(n)
 \end{aligned} \tag{2-15}$$

where:

$$h'(n) = \begin{cases} 0 & \text{for } n < d \\ h(n-d) & \text{for } d \leq n < N+d \\ 0 & \text{for } n \geq N+d \end{cases} \tag{2-16}$$

The expression in (2-15) is maximized at the same point where the impulse response h is maximized. For an impulse response such as the one in Figures 1-4 or 1-5, the maximal point indicates the direct echo path. The cross correlation between a white signal and its filtered version using a filter with impulse response from Figure 1-4 is shown in Figure 2-5. For this simulation, the correlation was calculated over a window of 0.5s, the cross correlation window is $K=4000$ samples with sampling rate $F_s=8000\text{Hz}$ and the flat delay is 200 samples. Notice the peaks in the cross correlation corresponding to the time inverted peaks in the impulse response.

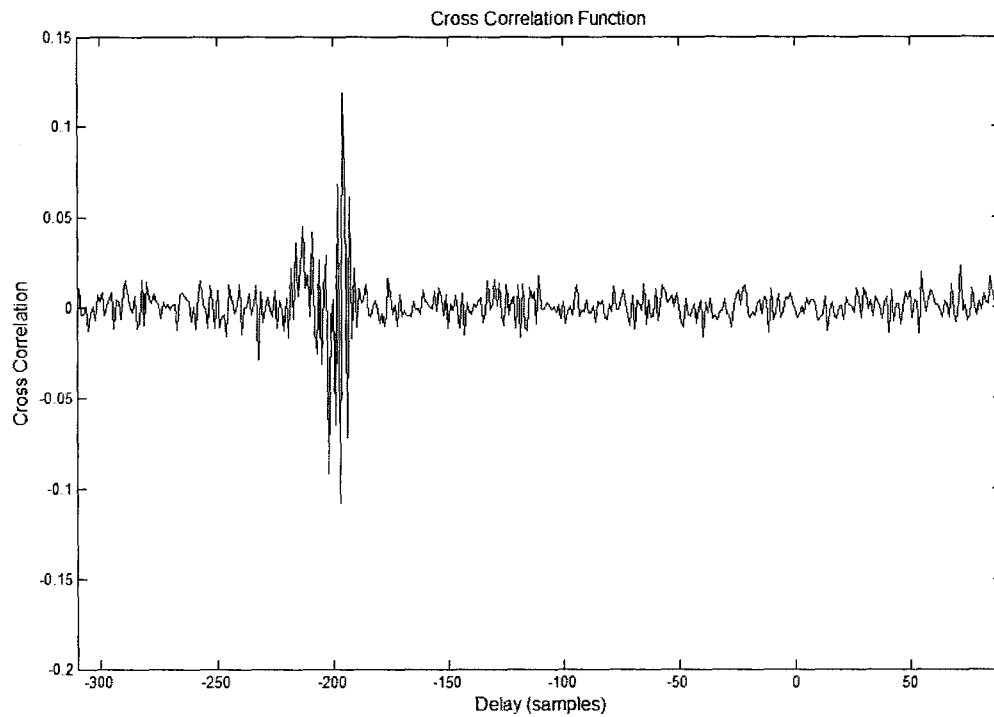


Figure 2-5 Cross correlation between filtered white speech signals

The development in (2-15) and (2-16) assumes ideal uncorrelated white signals. If the same echo path response is used with speech signals, the result is much different (Figure 2-6). In Figure 2.6, window sized of 4000 and 8000 samples were used to calculate the cross correlation function. The two cross correlated signals are filtered versions of each other (using the impulse response in Figure 1.4). Even with a window size of 1s or 8000 samples, the peaks are hard to distinguish from one another and the main peak is almost impossible to estimate.

In Section 2.2, GCC was introduced as a technique for dealing with delayed speech signals. Although quite effective in the case when the signals are delayed versions of each other, GCC's performance deteriorates in the presence of reverberation [12], [21] - [23], [30]. Nevertheless, GCC outperforms the classical cross correlation approach. This

improvement in performance comes at the cost of higher computation. Figure 2-7 depicts the GCC function for two speech signals. The second speech signal is derived by filtering the first speech signal using the impulse response in Figure 1.4. The parameters used are: GCC time window 65ms, $K=500$ samples, sampling rate $F_s=8000\text{Hz}$.

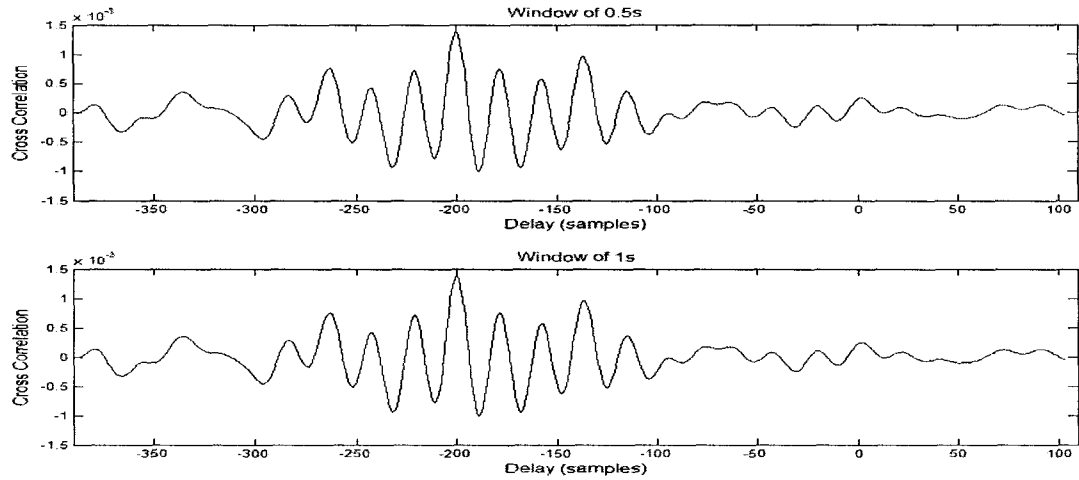


Figure 2-6 Cross correlation between filtered speech signals ($\Delta = 200$ samples)

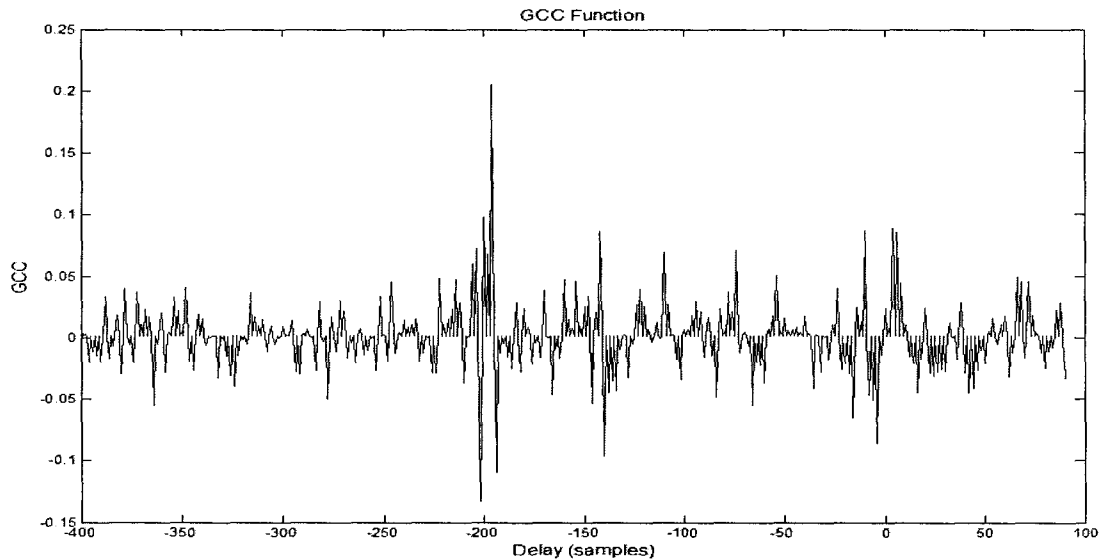


Figure 2- 7 GCC for filtered speech - reverberation present ($\Delta = 200$ samples)

New techniques for TDE have been introduced to account for reverberant environments. The cepstral method of processing was introduced in [21] and further refined in [24] – [26]. This method consists of a cepstral prefiltering stage applied to the signals in question which is used in conjunction with the GCC (Figure 2-8).

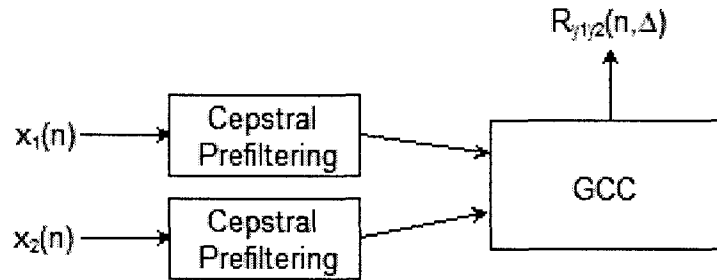


Figure 2-8 Cepstral Prefiltering for TDE

The main purpose of cepstral prefiltering is to alleviate the effects of reverberation on the performance of GCC. The TDE system resulting from the use of cepstral prefiltering and GCC is named GCC-CEP. The convolution operation of the input signal with the echo path response in (14) can be transformed into an additive component in the cepstral domain [27]. This component can then be estimated and extracted via linear filtering. Although the CEP-GCC method provides better performance than GCC in reverberant environments, its computational cost is prohibitively high.

Another more recent approach termed Adaptive Eigenvalue Decomposition (AED) was introduced in [29]. The AED algorithm estimates the impulse responses of different paths within the same acoustic environment (Figure 2-9). From the estimated acoustic impulse responses, the time delay can be calculated as the time difference between the main peak

(direct path) of the two impulse responses or as the peak of the correlation function between the two impulse responses. Since only the time difference between the main peaks (direct path) of the impulse responses is required, it is therefore not necessary to estimate the complete acoustic impulse responses [31].

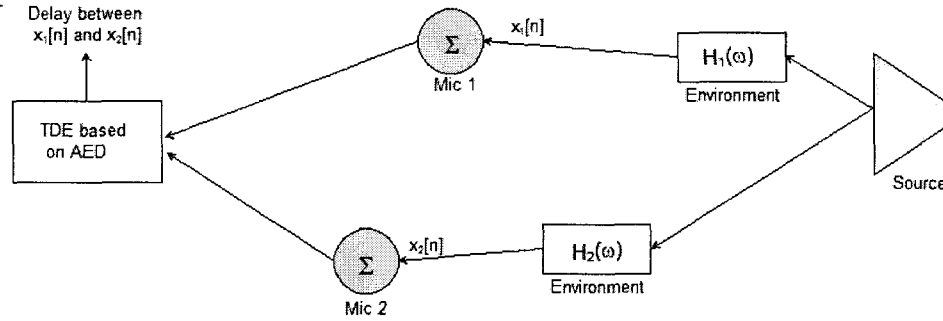


Figure 2-9 TDE based on AED

The adaptive EVD algorithm for TDE performs much better in highly reverberant environments than the GCC based methods. The algorithm proposed in [29] is only valid if either no noise or if spatiotemporally white noise is present. In [31], the AED algorithm was generalized to deal with colored noise. The AED algorithm estimates the desired impulse responses by approximating the *eigenvector of the correlation matrix of the input signal* which corresponds to the minimum eigenvalue of this matrix. In the case when two impulse responses of two different paths are estimated, the time delay between the two signals is the difference (in samples) between the peaks of the two impulse responses. The AED is a generalization of Adaptive Time Delay Estimation (ATDE) proposed in [33], [34] and [46]. In ATDE, a single adaptive filter is used to estimate the impulse path response and the delay is estimated by detecting the peak of the impulse response. Since

the purpose is only to estimate the flat delay, the adaptive filters used do not have to be large, and should only cover the part of the impulse response up to the peak value.

Nevertheless, AED and ATDE approaches are computationally intensive and ATDE does not usually present satisfactory performance [10].

The TDE approach used in this thesis is based on the approach undertaken in [10] and [11], but differs in some crucial aspects which will be explained in later chapters. This approach is based on time-domain prefiltering and cross correlation and has relatively low computational complexity.

CHAPTER 3

ADAPTIVE FILTERING FOR ACOUSTIC ECHO CANCELLATION

It was mentioned earlier in this thesis that AEC is a fundamentally a System Identification procedure. Adaptive filtering techniques are commonly used to perform AEC. Some topics from adaptive filter theory will be treated in this chapter. Since it is not the main focus of the current thesis, the treatment of adaptive filtering will be limited in scope and depth to include only the relevant material.

3.1 Adaptive Filtering for Single Channel Acoustic Echo Cancellation

A common setup for single channel AEC using adaptive filtering is shown in Figure 3-1. The purpose of the adaptive filter is to model the unknown echo path response. In this way the unwanted echo signal can be estimated and subtracted from the signal that is picked up at the microphone [35]-[37]. Non linear processing (NLP) is used to deal with non-linearity in the signals, while the double talk detector freezes the operation of the adaptive filter when double talk is perceived. There are many types of adaptive filtering algorithms that can be used for AEC. They differ in complexity and performance. For a complete background on adaptive signal processing, the reader is referred to [2], [32] and [54] which offer an excellent treatment of the topic.

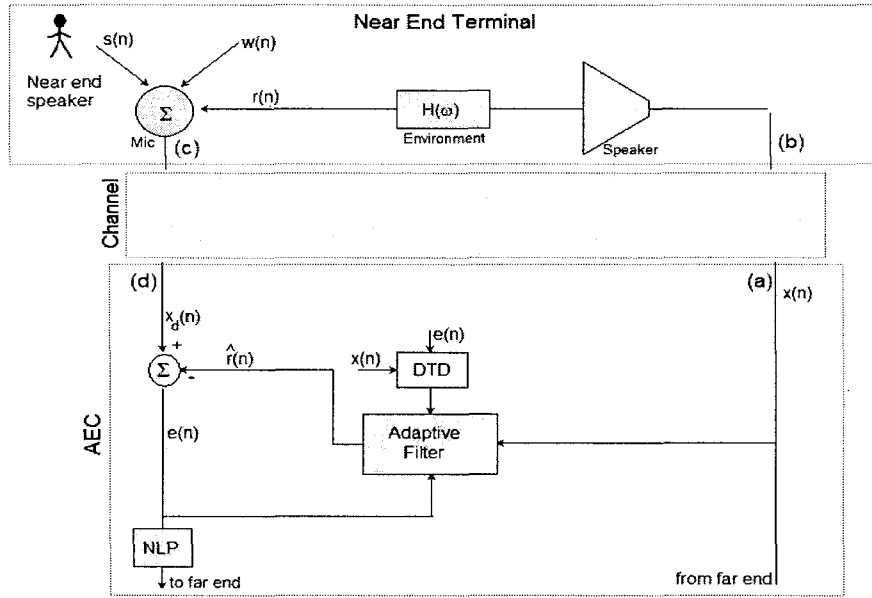


Figure 3-1 AEC with Adaptive Filtering

3.2 Least Mean Squares Adaptive Filter

Least Mean Square (LMS) adaptive filters are the most widely used adaptive filters due to their low complexity and relatively good performance. The LMS filter models the unknown system using an FIR filter with coefficients that are optimized recursively. The operation of the LMS filter is based on minimizing the mean square value of the error (MSE) - otherwise called “cost function” (see Figure 3-1) [2]:

$$\text{minimize } J_h(n) = (E\{|e(n)|^2\}) \text{ w.r.t } \mathbf{h}' \quad (3-1)$$

where $E\{\cdot\}$ is the expected value operator and \mathbf{h}' is the coefficient vector of the adaptive filter. Note that:

$$e(n) = d(n) - \hat{r}(n) \quad (3-2)$$

is the error signal and:

$$\hat{r}(n) = \mathbf{x}^T(n) \mathbf{h}' \quad (3-3)$$

Assuming real-valued signals, the cost function is:

$$\begin{aligned} J_{h'}(n) &= E\{|d(n) - \hat{r}(n)|^2\} \\ &= E\{d(n)^2 - d(n)\mathbf{h}'^T \mathbf{x}(n) - \mathbf{x}^T(n)\mathbf{h}' d(n) - \mathbf{x}^T(n)\mathbf{h}'\mathbf{h}'^T \mathbf{x}(n)\} \\ &= E\{d(n)^2\} - E\{\mathbf{h}'^T \mathbf{x}(n)d(n)\} - E\{\mathbf{x}^T(n)\mathbf{h}' d(n)\} - E\{\mathbf{h}'^T \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}'\} \\ &= \sigma_d^2 - 2\mathbf{h}'^T E\{\mathbf{x}(n)d(n)\} - \mathbf{h}'^T \mathbf{R}_x(n)\mathbf{h}' \end{aligned} \quad (3-4)$$

where σ_d^2 is the variance of $d[n]$ and $\mathbf{R}_x(n)$ is the autocorrelation matrix:

$$\mathbf{R}_x(n) = E \left\{ \begin{bmatrix} x(n)x(n) & x(n)x(n-1) & \dots & x(n)x(n-N+1) \\ x(n-1)x(n) & x(n-1)x(n-1) & \dots & x(n-1)x(n-N+1) \\ \dots & \dots & \dots & \dots \\ x(n-N+1)x(n) & x(n-N+1)x(n-1) & \dots & x(n-N+1)x(n-N+1) \end{bmatrix} \right\} \quad (3-5a)$$

or:

$$\mathbf{R}_x(n) = \begin{bmatrix} r_x(0) & r_x(1) & \dots & r_x(N-1) \\ r_x(-1) & r_x(0) & \dots & r_x(N-2) \\ \dots & \dots & \dots & \dots \\ r_x(-N+1) & r_x(-N+2) & \dots & r_x(0) \end{bmatrix} \quad (3-5b)$$

where:

$$r_x(k) = E\{x(n)x(n-k)\} \quad (3-6)$$

is the autocorrelation function of $x(n)$ at lag k .

To minimize the cost function, its gradient of the cost function in (3-4) w.r.t. \mathbf{h}' is equated to zero, leading to:

$$\nabla J_{h'}(n) = \frac{\partial J_{h'}(n)}{\partial \mathbf{h}'} = -2E\{\mathbf{x}(n)d(n)\} + 2R_x\{\mathbf{x}(n)\mathbf{h}'\} = 0 \quad (3-7a)$$

or:

$$\mathbf{h}' = R_x^{-1}(n)E\{\mathbf{x}(n)d(n)\} \quad (3-7b)$$

The expression in (3-7b) is the so called optimal Wiener solution [2].

One is faced with two main obstacles when computing the optimal weight (3-7b):

1. The correlation matrix might be large and hence its inverse used in (3-7b) becomes hard to calculate, especially in real-time;
2. The expected value operator $E\{\}$ cannot be computed in practical situations.

The first problem was addressed by the introduction of the steepest descent algorithm.

The steepest descent algorithm attempts to find the optimal value that minimizes the cost function (3-4) by recursively adapting the coefficients of the weight vector. At each iteration, the weight vector \mathbf{h}' is updated by *incrementing it with a vector pointing to the opposite direction of the gradient of the cost function* [2]. The recursion process is summarized in Table 3-1. The second problem mentioned above was addressed by Widrow [38]. He proposed that instantaneous values be used instead of expected values.

In this case:

$$E\{\mathbf{x}(n)d(n)\} \approx \mathbf{x}(n)d(n) \quad (3-8a)$$

and:

$$r_x(k) \approx x(n)x(n-k) \quad (3-8b)$$

Table 3-1 The gradient descent algorithm

1.	Initialize the weight vector $\mathbf{h}'(n)$
2.	Compute gradient vector $\nabla J_{h'}(n)$
3.	Update weight vector $\mathbf{h}'(n+1) = \mathbf{h}'(n) + 1/2\mu[-\nabla J_{h'}(n)]$
4.	Go to step 2

Using instantaneous values instead of the statistical expectations causes the values to oscillate randomly around the optimal solution. This means that the optimal solution is never likely to be found, instead it is only approximated with a certain (acceptable) margin of error. By using the approximations in (3-8a), (3-8b) together with the gradient descent algorithm, the LMS adaptive filtering method was constructed [38]. The gradient of the cost function is now approximated using:

$$\nabla J_{h'}(n) \approx -2\mathbf{x}(n)d(n) + 2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{h}' = -2\mathbf{x}(n)(d(n) - \mathbf{x}^T(n)\mathbf{h}') = 2\mathbf{x}(n)e(n) \quad (3-9)$$

The LMS adaptive filtering algorithm is summarized in Table 3-2. Although the filtering operation is performed by a FIR filter, the presence of feedback from the weight update makes the LMS filter prone to instability issues. In general, to avoid instability the value for μ must be chosen from the interval [2]:

$$0 < \mu < 2/\lambda_{\max} \quad (3-10)$$

where λ_{\max} is the largest eigenvalue of the correlation matrix $R_x(n)$. The larger the value for μ , the faster the convergence, but the larger the excess mean square error. This means that the approximated solution will oscillate around the optimal value.

Table 3-2 The LMS Algorithm

1.	Initialize: $\mathbf{h}'(0), \mathbf{x}(0)$
2.	Calculate adaptive filter output: $\hat{r}(n) = \mathbf{x}^T(n)\mathbf{h}(n)$
3.	Compute error: $e(n) = d(n) - \hat{r}(n)$
4.	Update weight vector : $\mathbf{h}'(n+1) = \mathbf{h}'(n) + 1/2\mu e(n)\mathbf{x}(n)$
5.	Go to step 2.

3.3 The Normalized Least Mean Squares Adaptive Filter

According to the weight update step in the LMS algorithm, the value of the weight vector depends on the input signal. If the input fluctuates this might cause instability or inaccurate convergence behaviour. To account for fluctuations in the input signal, the normalized LMS (NLMS) algorithm was introduced [2]. The only difference with the regular LMS algorithm is in the weight update step (see step 4 in Table 3-2). For the NLMS algorithm, this particular step becomes:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu}{2\|\mathbf{x}(n)\|} e(n)\mathbf{x}(n) \quad (3-11)$$

3.4 Subband Adaptive Filtering

Subband Adaptive Filters (SAF) are an alternative to fullband methods (such as NLMS) for AEC. This is mainly due to their computational efficiency and better performance in the presence of coloured signals such as speech [47] – [50].

SAF is based on subband processing of signals using filter banks [41]. Figure 3.2 depicts an analysis and synthesis filterbank. The analysis filter bank splits the input signal $x(n)$ into the subband signals $x_0(m)$, $x_1(m)$, ... $x_{M-1}(m)$. By processing each of those signals separately, one can selectively act on different parts of the spectrum of $x(n)$. Note that the sampling rate of each of the signals $x_0(m)$, $x_1(m)$, ... $x_{M-1}(m)$ is D times lower than that of $x(n)$. This allows for more efficient processing in the subbands. The synthesis filter band does the opposite, i.e. it combines the subband signals to form the output signal $y(n)$.

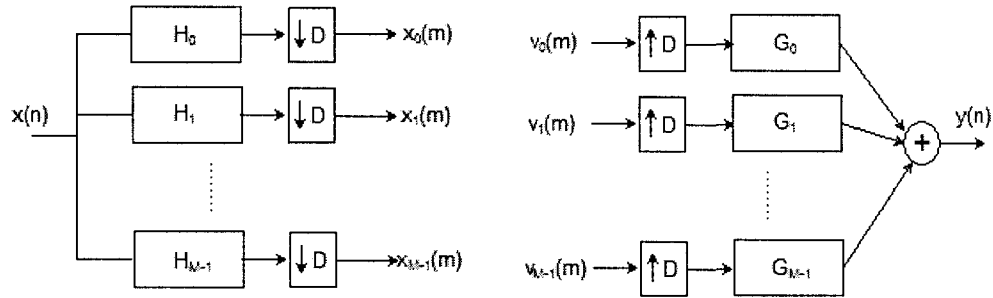


Figure 3-2 Analysis (right) and synthesis (left) filter banks

The analysis filters $H_0(\omega)$, $H_1(\omega)$, ... $H_{M-1}(\omega)$ are bandpass filters with passband of width π/M each of which covers a different part of the full spectrum. Similarly the synthesis filters $G_0(\omega)$, $G_1(\omega)$, ... $G_{M-1}(\omega)$.

The structure of a SAF is depicted in Figure 3.3.

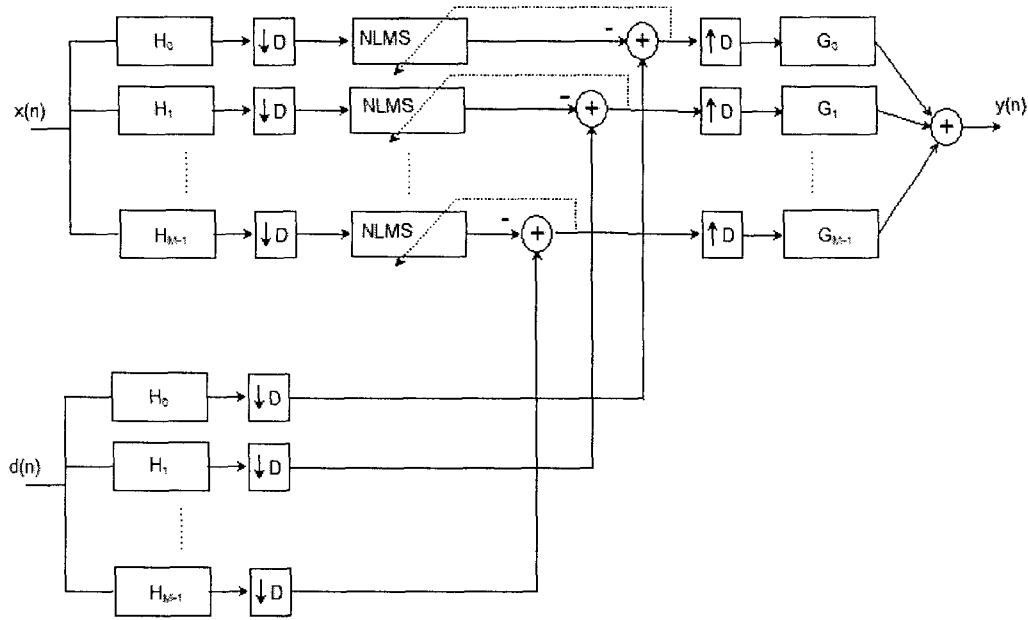


Figure 3-3 Subband adaptive filtering

In this setup, the signals are essentially split into subbands and adaptive filtering is subsequently applied on each subband. The separation of the signals into subbands is a reason for the greater flexibility of subband adaptive filtering when compared to fullband approaches such as NLMS.

In addition to providing the possibility for complexity reduction, downsampling spreads the signal spectrum, which makes subband adaptive filtering potentially more effective for coloured signals such as speech. In Figure 3.3, the input and desired signals $x(n)$ and $d(n)$ are split into M subbands by means of an identical analysis filterbank. When the decimation factor D is equal to the number of channels (subbands) M , the filterbank is *critically downsampled*. The problem with critically downsampled subband filtering is that, if the analysis and synthesis filters are not ideal (which is always the case), distortion

due to aliasing will be present in the reconstructed signals [47], [48]. Aliasing is the result of the spectrum of the signal in the transition bands of each of the analysis filters being reflected back into the baseband. To correct the aliasing problem, *oversampled filterbanks* are often used. In this type of filterbanks, the bandwidth π/M of each analysis filter is smaller than π/D thus making $M > D$. Note that, compared to the critically sampled case, an increase in the number of channels (hence filters) implies an increase in computational cost.

3.4.1 Subband Filter Structures for SAF

Modulated DFT filter banks (MDFB) are a popular design which combines the efficiency of polyphase decomposition and FFT-s to reduce the computational complexity of SAF. The topic of filterbank design is quite complex and will not be treated in this thesis. Some concepts however, are required to guarantee a logical continuity of the material and will be presented in the following sections.

Each of the analysis filters in the MDFB is generated by frequency-shifting a lowpass prototype filter with a given transfer function $H_0(z)$. The amount of shift for the k -th filter is in the frequency domain is constant $2\pi k/M$. Given the prototype filter response:

$$H_0(z) = \sum_{n=0}^{\infty} h_0[n]z^{-n} \quad (3-12)$$

The k -th analysis filter is a $2\pi k/M$ shifted version of $H_0(z)$ i.e.:

$$H_k(z) = H_0\left(z \cdot e^{\frac{-j2\pi k}{M}}\right) = \sum_{n=0}^{\infty} h_0(n) \left(z \cdot e^{\frac{-j2\pi k}{M}}\right)^{-n} = \sum_{n=0}^{\infty} h_k(n) z^{-n} \quad (3-13)$$

for $k = 1, \dots, M-1$, and

$$h_k(n) = h_0(n) e^{-j2\pi kn / M} \quad (3-14)$$

In the frequency domain:

$$H_k(\omega) = H_0\left(\omega - \frac{2\pi k}{M}\right) \quad (3-15)$$

for $k = 1, \dots, M-1$.

Similarly, the synthesis filters are obtained by frequency-shifting the prototype synthesis filter $g_0(n)$:

$$g_k(n) = g_0(n) e^{j2\pi kn / M} \quad (3-16)$$

Note that the frequency shift is the reverse of the one applied to the prototype analysis filter.

The Generalized DFT (GDFT) kernel [51] is defined as:

$$W_M^{(m+m_0)(k+k_0)} = e^{j\frac{2\pi}{M}(n+n_0)(k+k_0)} \quad (3-17)$$

By using the GDFT instead of the DFT, the centres of each subband filter can be shifted to locations other than the origin. The time offset in the GDFT also allows the subband filters to be linear phase. The choice for the time and frequency offset are:

$$n_0 = (L_p - 1)/2 \text{ and } k_0 = -1/2 \quad (3-18)$$

respectively, where L_p is the length of the polyphase filter prototype.

A direct implementation of the uniform Modulated GDFT filter bank (MGDFB) is shown in Figure 3-4. Note that the MGDFB reduces to the MDFB for $n_0=0$ and $k_0=0$.

The computation of the subband signals can be performed much more efficiently using MGDFB-s if polyphase decomposition and FFTs are employed. Specifically, *computation of all subband signals can be performed with one convolution and one weighed FFT of order M* . Furthermore, all the above operations have to be performed only once every D samples, where D is the downsampling rate [52].

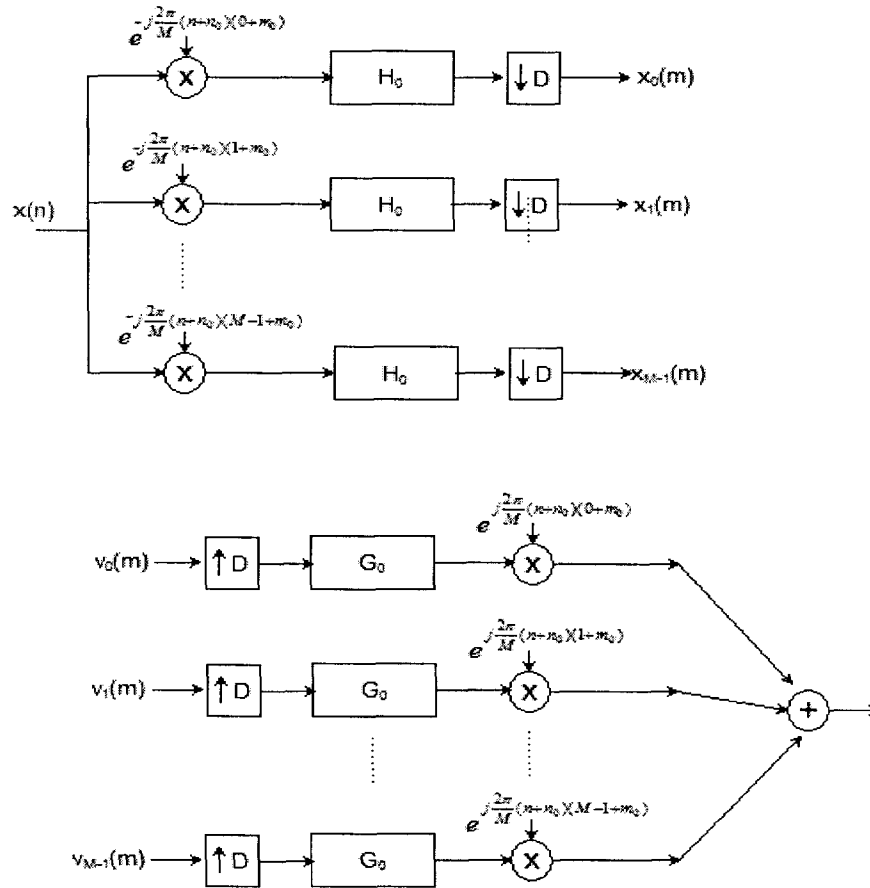


Figure 3-4 Analysis (top) and synthesis (bottom) GDFT modulated filter banks

Using the efficient structures such as the one in [53], the total number of multiplications for each output sample is:

$$Cost_{subband} = \frac{1}{D}(4M \log_2 M + 6M + L_p) + 2N_{subband} \quad (3-19)$$

$$Cost_{fullband} = 2N_{fullband} \quad (3-20)$$

where L_p is the length of the analysis and synthesis filters in the filterbank, $N_{subband}$ is the length of the subband adaptive filter and $N_{fullband}$ is the length of the fullband adaptive filter. For the sake of comparison, consider the adaptive filters with the parameters given in Table 3-2. These filters give a similar performance - actually the subband adaptive filter is superior as will be empirically demonstrated in Chapter 5.

Table 3-3 Computational complexity of fullband and subband adaptive filters compared

Subband	Fullband
NLMS	NLMS
$L_p=192, M=16, D=8$	N/A
$N_{subband}=92$	$N_{fullband}=512$
Total multiplications per output sample:	
252	1024

3.5 Tracking of Echo Path Impulse Responses

The echo path responses that are dealt with in this project are similar to the ones depicted in Figure 1-4 and 1-5. As discussed in Section 1.1.2, this type of impulse response is characterized by the presence of a number of zero-valued coefficients that correspond to the time delay of the direct path of the echo. When faced with the task of identifying such an impulse response, adaptive filters fail to identify the flat delay part as zero-valued coefficients [39], [40]. Figure 3-5 depicts the actual echo path while Figures 3-6 to 3-8 depict the resulting weight vector at the end of an adaptation period using NLMS adaptive filtering.

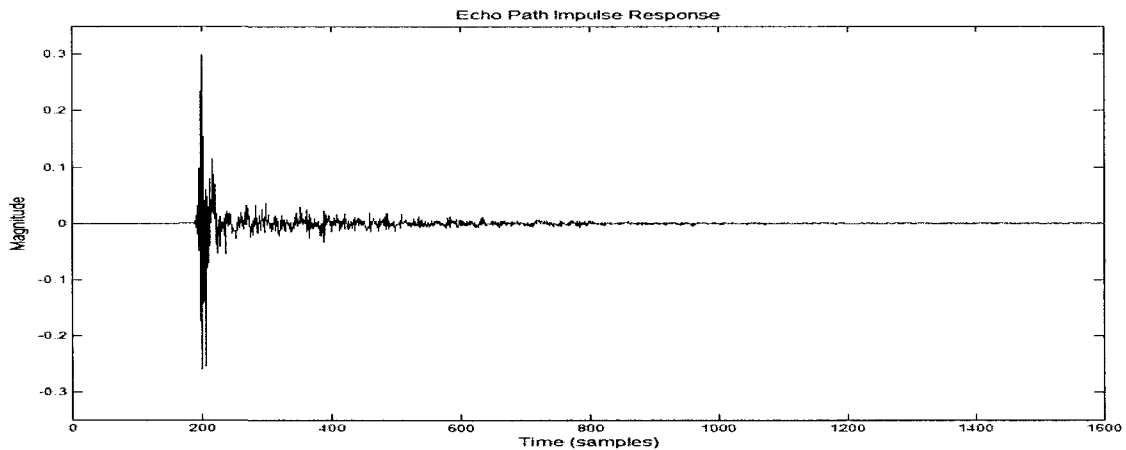


Figure 3-5 Echo path response (to be estimated)

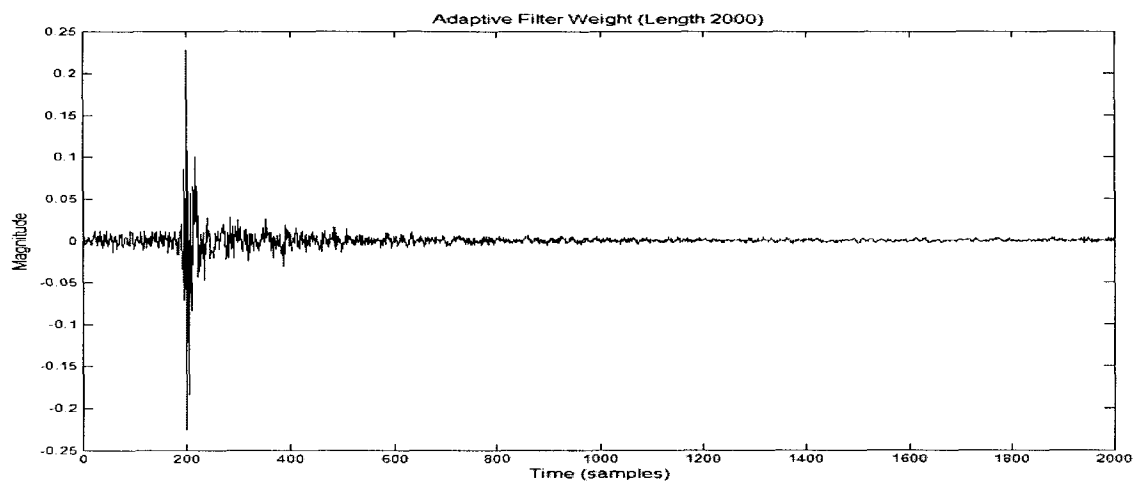


Figure 3-6 Estimated echo path response using 2000 sample long FIR NLMS

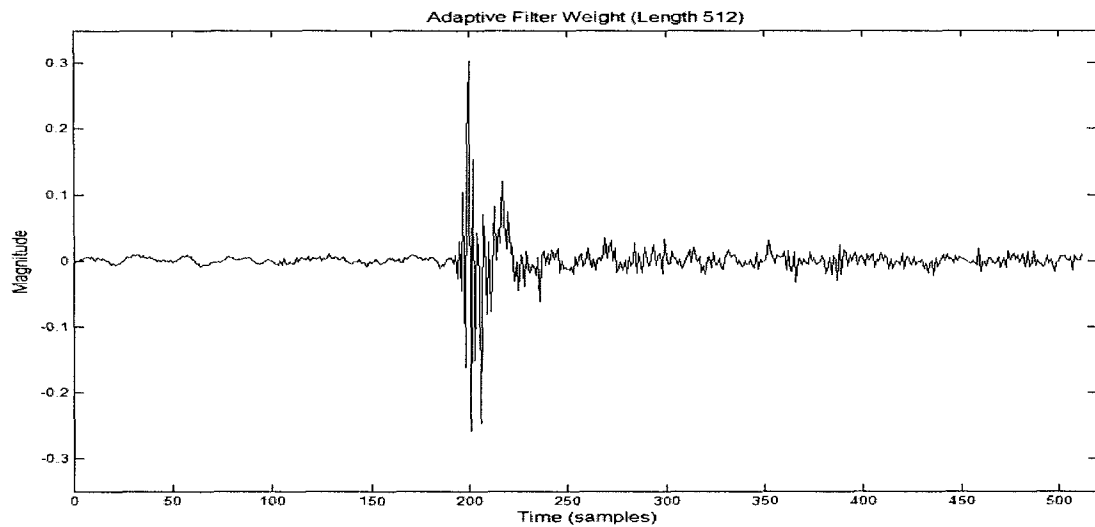


Figure 3-7 Estimated echo path response using 512 sample long FIR NLMS

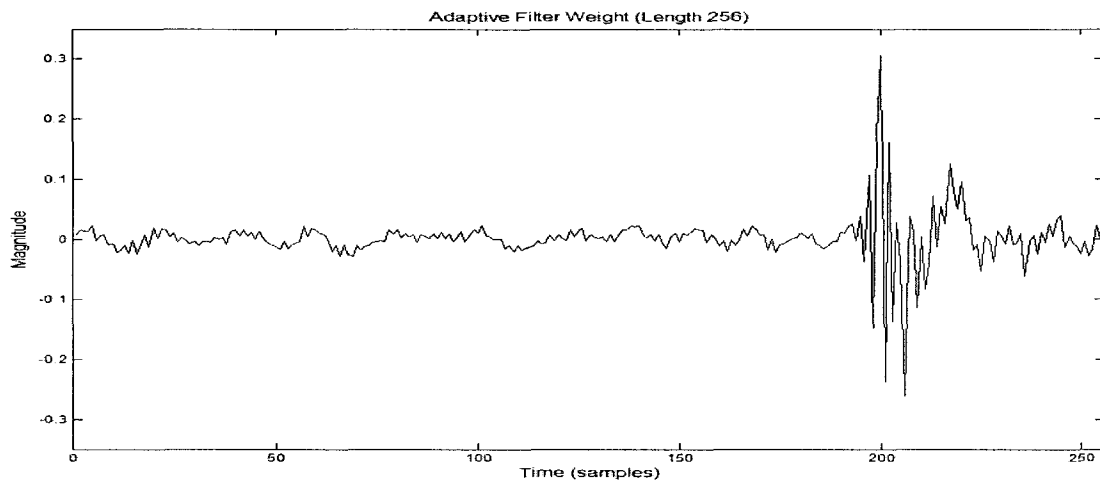


Figure 3-8 Estimated echo path response using 256 sample long FIR NLMS

Notice how in Figures 3-5 to 3-8, the NLMS adaptive filter attempts to model the first 200 zero-valued coefficients of the echo path impulse response. As it becomes obvious from the figures above, the presence of the flat delay impairs the NLMS adaptive filter's ability to model the active part of the echo path response. Similarly, when using NLMS in a subband scheme, the zero-valued coefficients of the echo path impulse response need to be accounted for.

A solution to this problem is to account for the flat delay by incorporating it in the input signal prior to the latter being used in the adaptive filter. The setup for this approach is depicted in Figure 3-9. The results of fullband NLMS adaptation when flat delay is compensated into the input signals are shown in Figures 3-10 to 3-12.

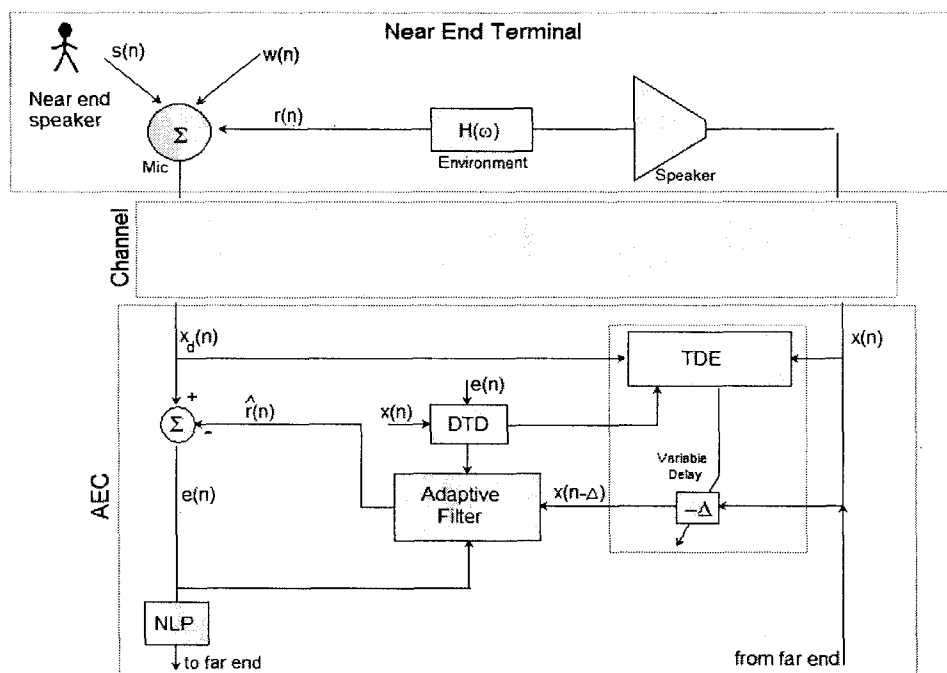


Figure 3-9 Flat delay estimation and compensation for AEC

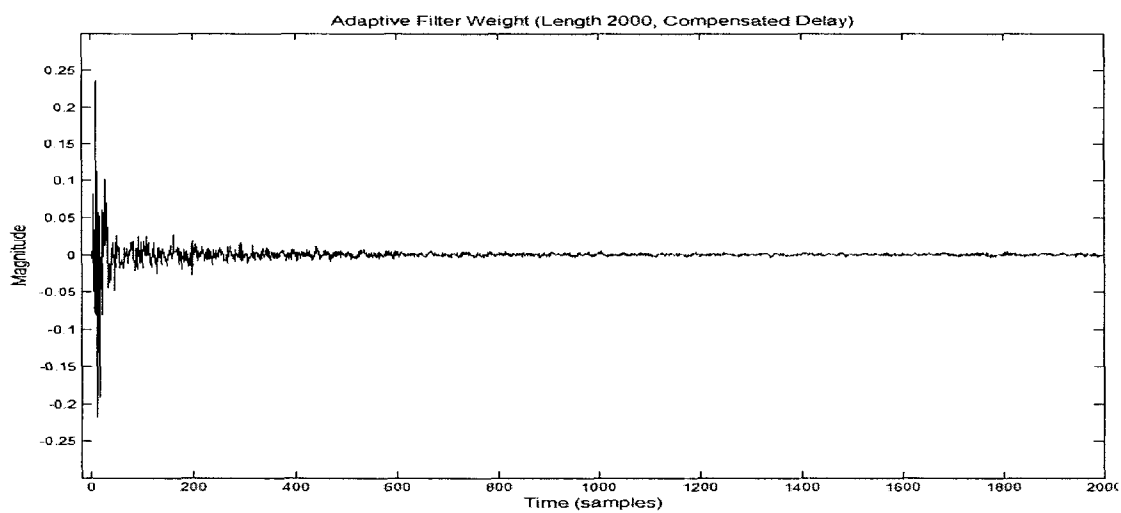


Figure 3-10 Estimated echo path response using 2000 sample long FIR NLMS with delay compensation

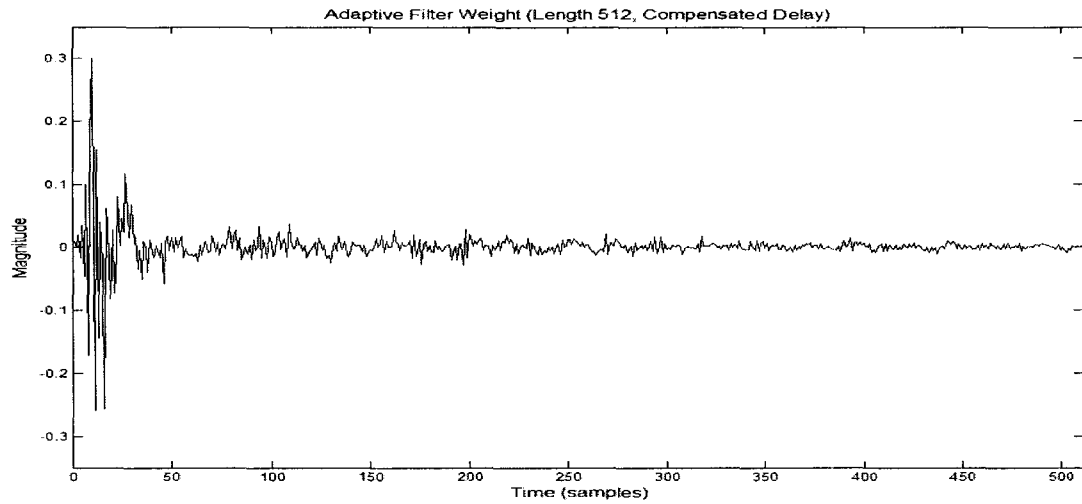


Figure 3-11 Estimated echo path response using 512 sample long FIR NLMS with delay compensation

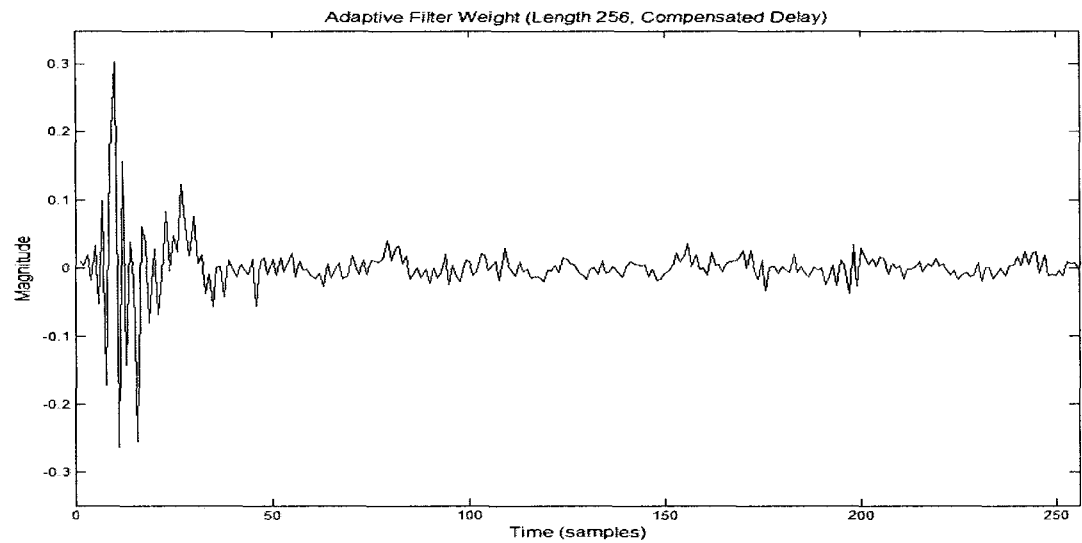


Figure 3-12 Estimated echo path response using 256 sample long FIR NLMS with delay compensation

It is obvious from these results that by compensating for the flat delay significant gains can be achieved in terms of performance as well as computational savings by reducing the

size of the adaptive filters. By calculating the flat delay separately, the coefficients of the adaptive filter can be dedicated exclusively to estimating the active part of the echo path impulse response.

The purpose of the upcoming chapters is to present an appropriate TDE scheme that estimates the flat delay in real time and compensates this delay into the input path.

CHAPTER 4

EFFICIENT TIME DELAY ESTIMATION FOR REAL TIME OPERATION

The TDE schemes proposed in this chapter are based on the cross correlation approach. It is assumed that the signals being processed are speech signals. As it was mentioned in Section 2.2, simple cross correlation does not provide a good insight into the relative time delays between speech signals. That is why, similarly to GCC, a preprocessing routine is added to the TDE scheme in order to make signals easier to handle by cross correlation. The proposed TDE scheme is intended for real-time operation and time-varying environments. Hence, besides robustness, convergence speed and computational expenses are two major factors that influenced its design.

4.1 Parallel Search of Cross Correlation Values

A generalized diagram depicting a high-level operation of the proposed TDE scheme is shown in Figure 4-1. The approach employed in this scheme approximates the modified cross correlation values between two signals corresponding to several delays. The optimal delay is the one corresponding to the maximum value of the cross correlation computed. The signals are preprocessed prior to being used for estimating their cross correlation. Note that the values of the cross correlation are computed concurrently for each of the delays checked. Consequently the method for calculating the cross correlation values in this approach is analogous to a parallel search method. This makes the search process

more immune to the presence of local maxima in the correlation function and hence more robust.

In one single “scan”, the proposed TDE scheme does the following:

1. Compute the cross-correlation for a finite number of delays $d_1 < d_2 < \dots < d_P$;
2. Find the maximum value of correlation from step 1. The corresponding delay is the “most optimal” among the delays that were evaluated;

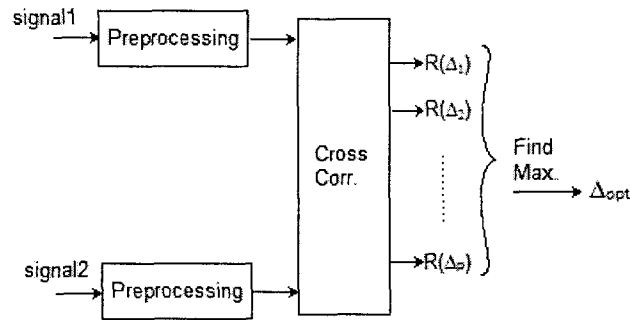


Figure 4-1 Simplified diagram of the proposed TDE scheme

The proposed TDE scheme performs a sampling of the modified cross correlation function. If one of the delays checked in step 1 is the optimal delay, then one “scan” is sufficient. If, on the other hand, the optimal delay is not one of the delays checked and is therefore not correctly detected, one of the following is true:

Case 1. The optimal delay is outside the search interval, i.e. $d_{opt} < d_1$ or $d_{opt} > d_P$ (Figure 4-2);

Case 2. The optimal delay lies between two consecutive delay values checked, i.e. $d_k < d_{opt} < d_{k+1}$ for any $1 < k < P$ (Figure 4-3);

A solution to *case 1* is to perform a new search in a new interval, that is, to evaluate a new set of delays. The solution to *case 2* is to perform a “fine” (higher resolution) search, thus evaluating a new set of delays that are closer together than the set of delays in the first search. In this thesis, it is assumed that the optimal delay lies inside the first search interval, that is:

$$d_l < d_{opt} < d_p;$$

Two versions of the TDE search scheme are proposed in this thesis: the first one allows for coverage of a wide search range, the second one performs a finer search, thus achieving a better search resolution. Assuming that the global maximum in the cross correlation function is distinct and the delay resolution is appropriate, the optimal delay can be promptly found by employing the proposed TDE scheme.

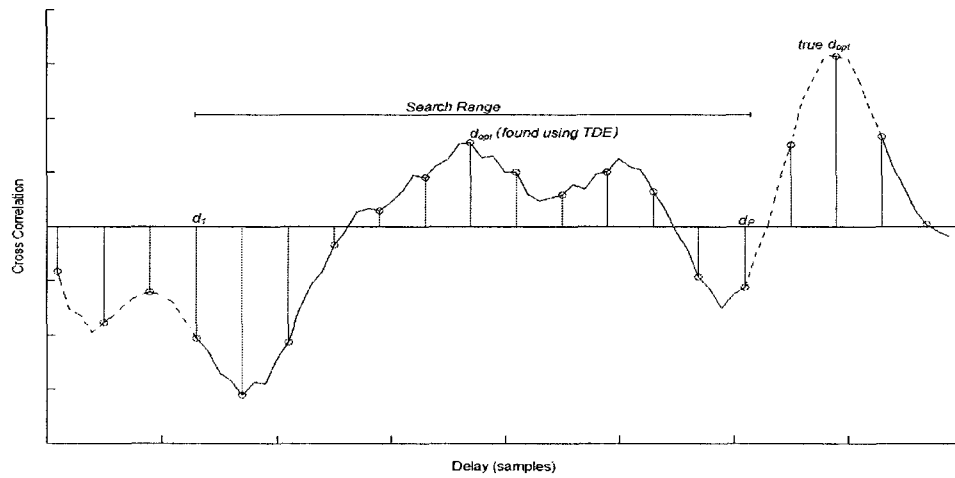


Figure 4-2 Result of TDE when choosing wrong search interval (Case 1)

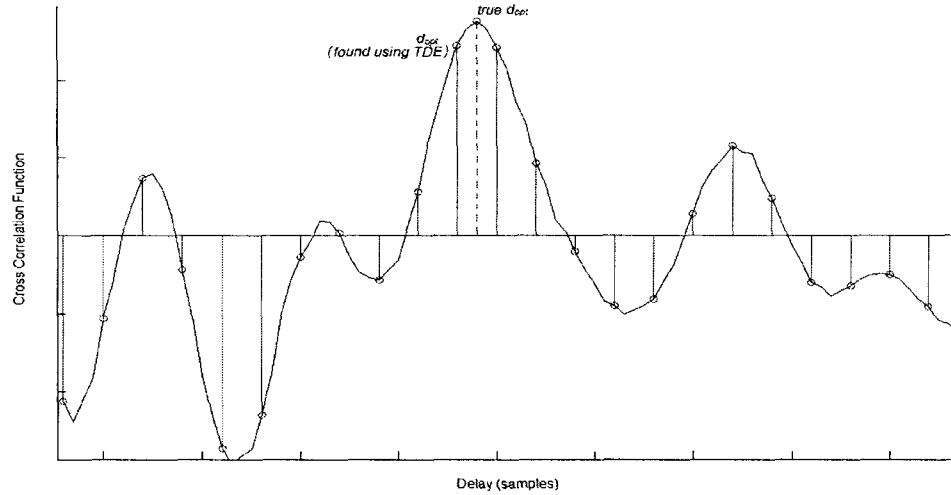


Figure 4-3 Result of TDE when optimal delay lies between two consecutive checked delays (Case 2)

As mentioned earlier in the section, the proposed method performs a parallel search of the correlation function. An advantage of this approach is that it provides increased robustness with respect to time varying environments and signals. This is due to the fact that the TDE scheme uses signal segments that are extracted within approximately the same time intervals to calculate the cross correlation for different delays. This last property is especially important in real-time processing. The difference between the sequential and parallel (proposed) approaches in calculating the cross correlation at different delays is depicted in Figure 4-4.

4.2 Preprocessing

GCC is a very popular TDE method due to its robustness and relatively low computational complexity. As discussed in Section 2.2, GCC is a TDE technique used to handle time-varying and colored signals by prefiltering them before processing. Similarly,

in this project, the signals are preprocessed before being used to compute their relative time delay. The procedure presented in this project is different from the preprocessing used in GCC but its purpose is the same: making the input signals resemble white signals by widening their bandwidth, so that the maxima of their correlation function becomes easier to detect. The preprocessing step employed in this project is similar to the one used in [10] and [11] (Figure 4-5).

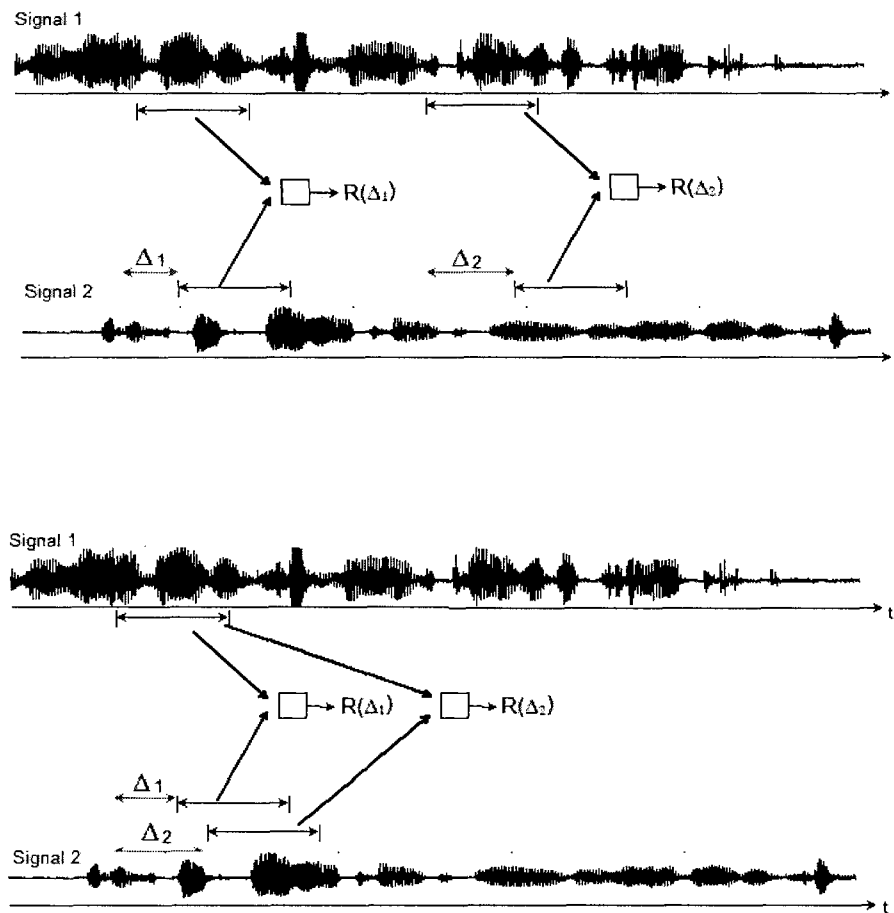


Figure 4-4 Serial (top) vs. parallel (bottom) search for TDE

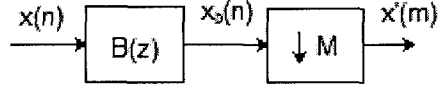


Figure 4-5 Preprocessing

The preprocessed signals are:

$$x'(m) = x_b(n) = \mathbf{x}^T(n)\mathbf{b}, \quad n = 0, M, 2M, \dots \quad (4-1a)$$

and

$$x'_d(m) = x_{bd}(n) = \mathbf{x}_d^T(n)\mathbf{b}, \quad n = 0, M, 2M, \dots \quad (4-1b)$$

where $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-N+1)]$ is the input vector, $\mathbf{b}^T = [b(0), b(1), \dots, b(N-1)]$ is the decimation filter coefficient vector and $m = n/M$ is the sample index of the preprocessed signals.

The sampling rate of $x'_b(m)$ is M times lower than that of $x_b(n)$. The frequency domain representation of $x'_b(m)$ is related to that of $x_b(n)$ by:

$$X'_b(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X_b\left(e^{j\frac{\omega - 2\pi k}{M}}\right) \quad (4-2)$$

It can be deduced from (4-1) and (4-2) that downsampling has two (related) consequences that are of relevance to the current thesis. First, from a time-domain perspective, the number of computations is reduced since the number of input samples is decreased by a factor of M . Secondly, from a frequency domain perspective, the resulting spectrum is composed of “ M uniformly shifted (by $2\pi/M$) and stretched (by M) versions of the original signal frequency response $X(\omega)$ ” [42]. The bandwidth increase is of special

interest in this case. Figure 4-6 shows the frequency domain effect of downsampling a signal by a factor of 2.

As it is made apparent from Figure 4-6, if the bandwidth of the original signal $x_b(n)$ is larger than $2\pi/M$, then aliasing will occur. In order to avoid aliasing, a filter $B(\omega)$ with bandwidth $2\pi/M$ should be placed in series with the downsampler. (Figure 4-7).

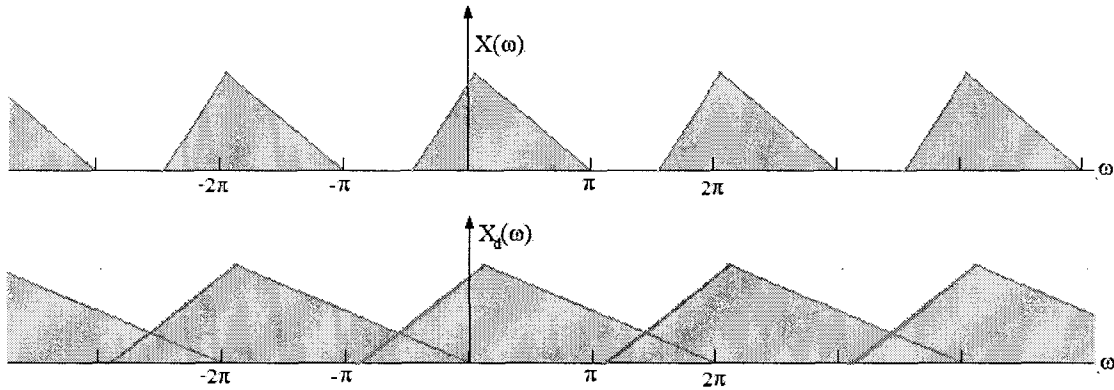


Figure 4-6 Frequency domain result of downsampling with M=2

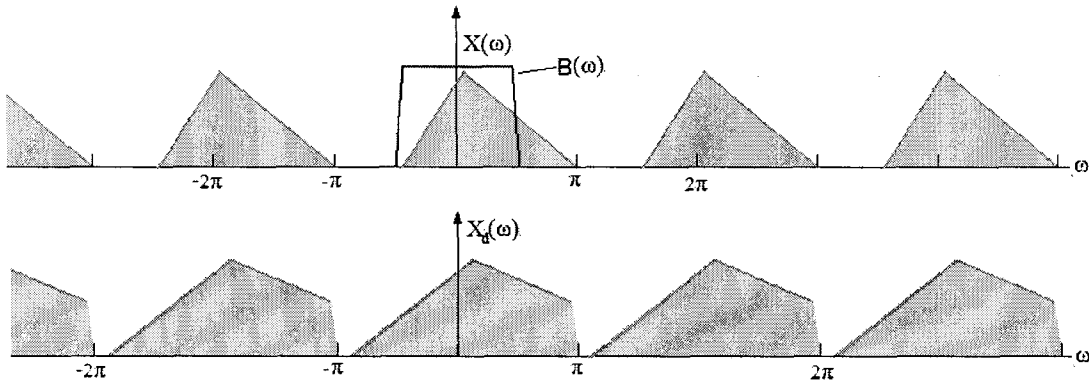


Figure 4-7 Result of decimation

Similarly, in this thesis, a bandlimiting filter is used. The filter $B(\omega)$ in Figure 4-4 has to perform three functions for the purposes of the current thesis:

1. Antialiasing – hence should have a bandwidth of $2\pi/M$;
2. Selecting a portion of the signal spectrum with relatively high energy – hence its center frequency must be chosen accordingly.
3. Select a portion of the signal spectrum that is not too fluctuating – such that the spectrum resulting from the downsampled speech signal resembles a white signal.

The spectrum of the input signal $x(n)$ is related to that of the preprocessed signal $x'_b(n)$ by [41]:

$$X'_b(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} B\left(e^{j\frac{(\omega-2\pi k)}{M}}\right) X\left(e^{j\frac{(\omega-2\pi k)}{M}}\right) \quad (4-3)$$

To find a center frequency for $B(\omega)$, the frequency distribution of speech signals must be considered. For such signals most of the energy is concentrated within the first 4kHz. A good choice for the center frequency is 750Hz [10].

Figures 4-8 and 4-9 depict the original signal sampled at 8000Hz and its spectrum respectively. The preprocessing is performed on this signal using the filter shown in figure 4-10 which has center frequency at 750Hz and bandwidth of 1800Hz. The downsampling rate used is $M=8$. The resulting spectrum is shown in Figure 4-11. The spectrum in Figure 4-11 is much wider than the one in Figure 4-9.

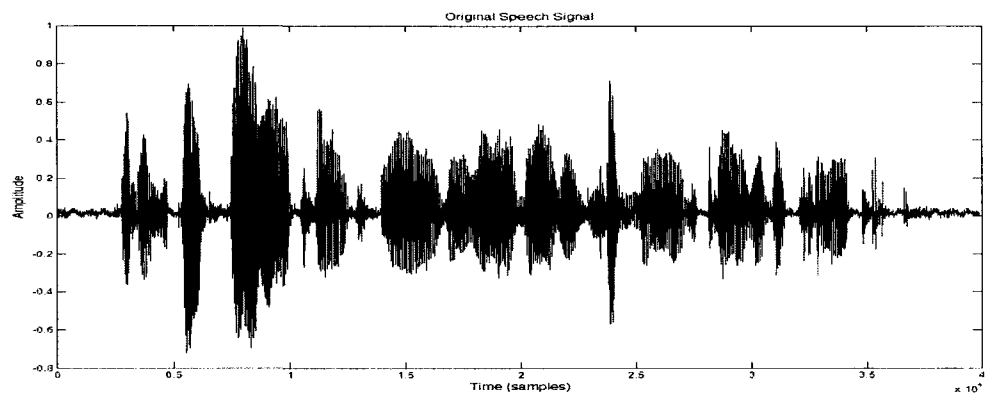


Figure 4-8 Original speech signal

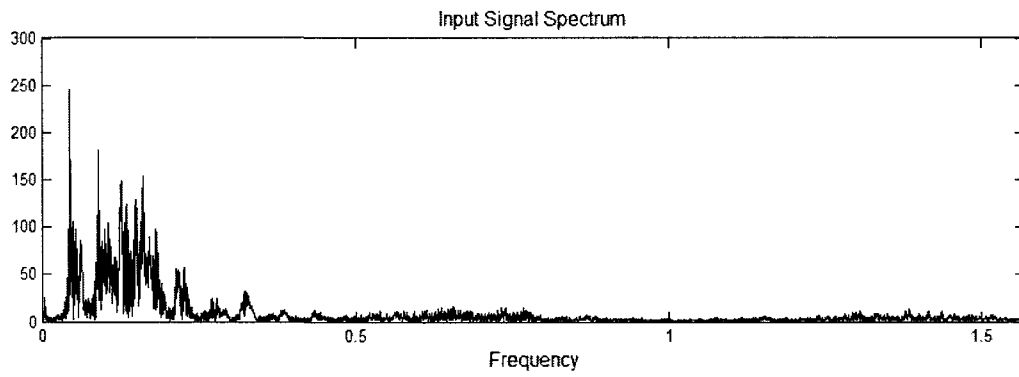


Figure 4-9 Spectrum of original signal

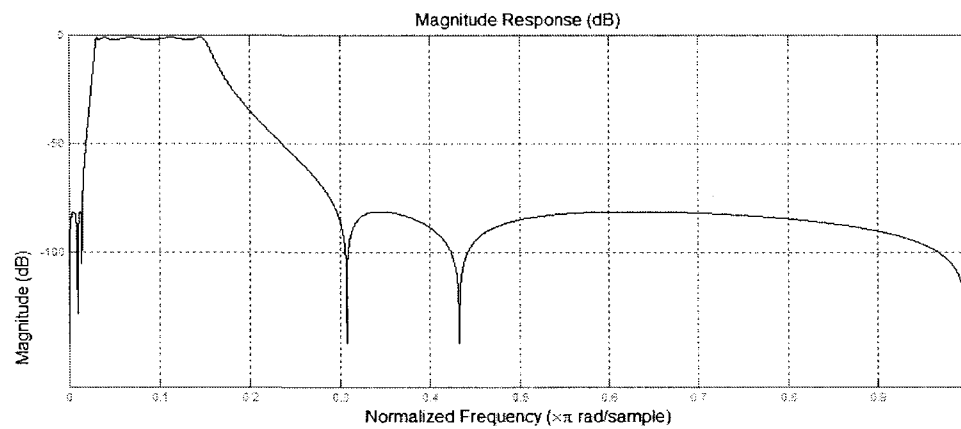


Figure 4-10 Frequency response of the decimation filter $B(\omega)$

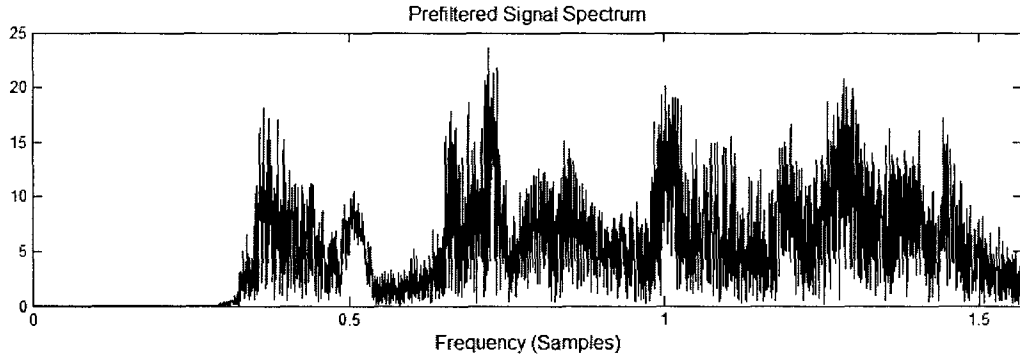


Figure 4-11 Spectrum of the preprocessed signal

4.3 Structures for Time Delay Estimation

Two different structures for real-time TDE are introduced in this project. The first one is used to obtain a “rough” estimate of the delay to the nearest M -sampling periods. Thus the use of this structure warrants the following assumption to be made:

$$\text{Optimal delay estimated from first structure} = \Delta_{\text{opt1}} M T_s \quad (4-4a)$$

where Δ_{opt1} is an integer and M is the (integer > 1) downsampling factor.

The second structure on the other hand, can be employed to estimate the time delay to the nearest sampling period:

$$\text{Optimal delay estimate from second structure} = \Delta_{\text{opt2}} T_s. \quad (4-4b)$$

where Δ_{opt2} is an integer. Both structures use preprocessed signals as introduced in Section 4.2. The preprocessed signals are then cross correlated for different delays and the delay corresponding to the maximal cross correlation value is extracted.

The next paragraphs treat each structure separately. Section 4-4 discusses the rough estimate of the time delay to the nearest M -sampling periods using the first structure. Section 4-5, on the other hand, introduces the second structure employed to estimate the time delay to the nearest sampling period. The two structures can be used separately or in combination depending on the application.

4.4 Rough Estimate of the Delay

The structure introduced in this section is used for estimation of the delay between two signals $x(n)$ and $x_d(n)$ to the nearest M -sampling periods, where M is an integer larger than one. In this approach it is assumed that the delay $\Delta_{opt}MT_s$ in (4-4a) can take one of the values:

$$0, MT_s, 2MT_s, \dots, PMT_s, \quad (4-5)$$

where T_s is the sampling period and P is an integer. Both signals are first preprocessed to obtain $x'(m)$ and $x'_d(m)$ respectively, that is:

$$x'(m) = x'(n/M) = \mathbf{x}^T(n)\mathbf{b} \text{ for } n = 0, M, 2M, \dots \quad (4-6a)$$

and

$$x'_d(m) = x'_d(n/M) = \mathbf{x}_d^T(n)\mathbf{b} \text{ for } n = 0, M, 2M, \dots \quad (4-6b)$$

where $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-N+1)]$ is the input vector, $\mathbf{b}^T = [b(0), b(1), \dots, b(N-1)]$ is the decimation filter coefficient vector and $m = n/M$ is the sample index of the preprocessed signals. To determine which one of the delays in (4-5) is the most optimal,

the cross correlation values between $x'(m-k)$ and $x'_d(m)$ are approximated for $k = 0, 1, \dots, P$. Notice that the time index n in the preprocessed signals $x'(m)$ and $x'_d(m)$ is a multiple of M due to downsampling. This implies that the sampling rates of $x'(m)$ and $x'_d(m)$ are M times smaller than those of $x(n)$ and $x_d(n)$.

To approximate the cross correlation between two signals $x'(m)$ and $x'_d(m)$ with relative delay kM samples at time instant m , the proposed structures use a slightly modified version of the expression (see Appendix A):

$$\rho_{x'_d x'}(m, kM) = \frac{R_{x'_d x'}(m, kM)}{\sqrt{R_{x'}(m-k)R_{x'_d}(m)}} \text{ for } k = 0, 1, 2, \dots, P \quad (4-7)$$

The expressions in the numerator and denominator in (4-7) are computed using the iterative formulas:

$$R_{x'_d x'}(m, kM) = (1 - \alpha)R_{x'_d x'}(m-1, kM) + \alpha x'_d(m)x'(m-k) \quad (4-8a)$$

$$R_{x'_d}(m) = (1 - \beta)R_{x'_d}(m-1) + \beta x'_d(m)x'_d(m) \quad (4-8b)$$

$$R_{x'}(m) = (1 - \beta)R_{x'}(m-1) + \beta x'(m)x'(m) \quad (4-8c)$$

The proposed structure generates $P+1$ cross correlation values corresponding to $P+1$ delays using the following - more convenient - expression instead of the one in (4-7):

$$\rho_{x'_d x'}(m, kM) = \frac{R_{x'_d x'}(m, kM)}{R_{x'}(m-k)} \text{ for } k = 0, 1, 2, \dots, P \quad (4-9)$$

Thus the cross correlation is computed for the delays kM for $k = 0, 1, \dots, P$. The sampling rate T_s is assumed to be equal to 1. The highest of the $P+1$ cross correlation values

generated using (4-9) corresponds to the “most optimal” among the $P+1$ delays checked.

The reason for excluding $R_{x_d, x_d}(n)$ (4-9) is simply that all the $P+1$ cross correlation values generated using (4-7) have $R_{x_d, x_d}(n)$ in their denominator. Since all the cross correlation values are to be divided by the same number, this operation would make no difference in the final result when choosing the highest value, hence it is irrelevant.

The number of iterations performed when computing (4-8a) to (4-8c) and (4-9) must be large enough to give meaningful values of the cross correlation. This also implies that there must be a “wait time” during which the TDE system has to remain idle until the values of cross correlation are indicative of the true relative cross-correlation values.

Assuming the iterations are run K times, the wait time for the system is approximately:

$$\tau = KMT_s, \quad (4-10)$$

Another implication of this method of computing cross correlation values is that it must be assumed that the cross-correlation function does not change during the wait time, i.e. the environment system is assumed to be time invariant throughout a period of time equal to τ .

After the cross correlation values are computed, a control algorithm simply picks the value for the delay that corresponds to the largest of the $P+1$ cross correlation values. The structure used for preprocessing and to calculate the values in (4-9) is shown in Figures 4-12 and 4-13. The algorithm for finding the rough delay to the nearest M sampling periods is summarized in Table 4-1.

Table 4-1 Rough estimate of the delay – individual steps

1.	<p>Initialize cross correlation values: $R_{x'dx'}(0, kM)$, $R_{x'd}(-k)$ for $k = 0, 1, \dots, P$; Initialize $count = 1$;</p>
2.	<p>Preprocess incoming signals to get $x'(n)$ and $x'_d(n)$:</p> $x'(n) = \mathbf{x}^T(n)\mathbf{b}(n) \text{ for } n = 0, M, 2M, \dots$ $x'_d(n) = \mathbf{x}_d^T(n)\mathbf{b}(n) \text{ for } n = 0, M, 2M, \dots$ <p>Increment $count$;</p>
3.	<p>Calculate cross correlation values:</p> $R_{x'dx'}(m, kM) = (1 - \alpha)R_{x'dx'}(m - 1, kM) + \alpha x'_d(m)x'(m - k)$ $R_{x'}(m) = (1 - \beta)R_{x'}(m - 1) + \beta x'(m)x'(m)$ $\rho_{x'dx'}(m, kM) = \frac{R_{x'dx'}(m, kM)}{R_{x'}(m - k)}$ <p>for $k = 0, 1, \dots, P$</p>
4.	<p>If $count \leq K$, go to step 2, otherwise go to step 5</p>
5.	<p>Extract the optimal delay:</p> $k_{opt} = \arg_k \{ \max \{ \rho_{x'dx'}(m, kM) \} \} \text{ for } k=0, 1, \dots, P$ <p>The optimal delay is $k_{opt}M$</p>

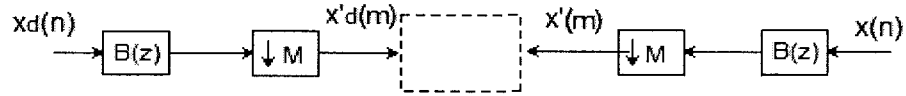


Figure 4-12 TDE scheme for approximating the delay to the nearest M samples

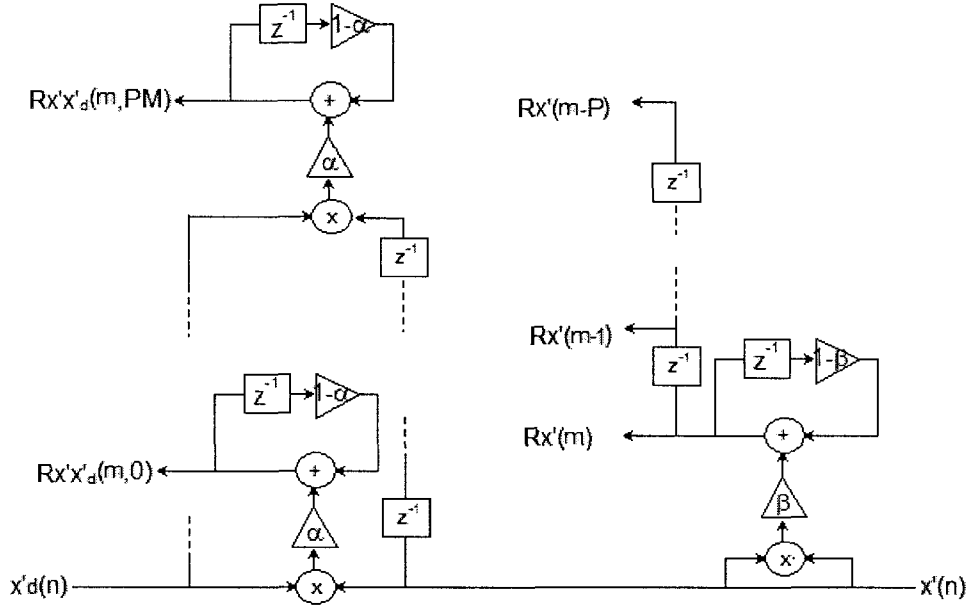


Figure 4-13 Structure for estimating the correlation values (dotted box in Figure 4-10)

The structure proposed in this section can find a rough value for the delay “in a single run” if it is assumed that the optimal delay $\Delta_{opt} MT_s$ is within $[0, MPT_s]$. The parameters that can be controlled to optimize the performance of the structure proposed in this section are:

K – (number of iterations) the larger its value, the better the approximation for the cross correlation, but its increase results in a longer “wait time” to generate the needed values of cross correlation.

P – (number of delays that are “checked”) increasing the value for this parameter results in a wider search range. As a side effect, the increase of P results in an increase in computation and complexity.

M – (downsampling factor) the larger its value, the larger the search range. On the other hand, increasing its value reduces resolution in the search space thus making it possible for the maximum point to not be detected. Also, a large M requires longer preprocessing filters in order to achieve sufficiently good response.

4.5 Exact Estimate of the Delay

The structure introduced in Section 4.4 was used to find the optimal delay to the nearest M sampling periods. In order to achieve a better “delay resolution”, a new structure is introduced in this section. The proposed structure finds the optimal delay to the nearest sampling period between two signals $x(n)$ and $x_d(n)$. In this case, it is assumed that the optimal delay Δ_{opt2} in can take one of the $2L+1$ values:

$$0, T_s, \dots, 2LT_s. \quad (4-11)$$

This structure uses the preprocessed signals $x_l'(m)$ and $x_d'(m)$.

$$x_l'(m) = \mathbf{x}^T(n-l)\mathbf{b} \text{ for } n = 0, M, 2M, \dots; l = 0, 1, 2L \quad (4-12a)$$

and

$$x_d'(m) = \mathbf{x}_d^T(n)\mathbf{b} \text{ for } n = 0, M, 2M, \dots \quad (4-12b)$$

where $\mathbf{x}^T(n-l) = [x(n-l), x(n-l-1), \dots, x(n-l-N+1)]$ and $\mathbf{x}_d^T(n) = [x_d(n), x_d(n-1), \dots, x_d(n-N+1)]$ are the input vectors, $\mathbf{b}^T = [b(0), b(1), \dots, b(N-1)]$ is the decimation filter coefficient vector and $m = n/M$ is the sample index of the preprocessed signals. To find the most optimal between the delays in (4-11), the cross correlation between $x'_l(m)$ and $x'_d(m)$ is approximated for $l = 0, 1, \dots, 2L$. The time index n is a multiple of M due to downsampling.

The values for the cross correlation calculated using this structure are estimated as:

$$\rho_{x'_d x'}(m, l) = \frac{R_{x'_d x'}(m, l)}{R_{x'}(m - l)} \quad (4-13)$$

where:

$$R_{x'_d x'}(m, l) = (1 - \alpha)R_{x'_d x'}(m - 1, l) + \alpha x'_d(m) x'_l(m) \quad (4-14a)$$

$$R_{x'}(m - l) = (1 - \beta)R_{x'}(m - l - 1) + \beta x'_l(m) x'_l(m) \quad (4-14b)$$

for $l = 0, 1, \dots, 2L$.

The structure used for preprocessing and to calculate the cross correlation values in (4-14a), (4-14b) and (4-13) is illustrated in Figures 4-12 and 4-13.

Note that if $2L = M-1$, the new structure performs a polyphase decomposition of $x'(n)$.

The parameters that can be used in this structure to optimize the performance are:

K – (number of iterations) the larger its value, the better the approximation for the cross correlation, but its increase results in a longer “wait time” to generate the needed values of cross correlation.

L – (determines the number of delays that are “checked”) increasing the value for this parameter results in a wider search range. As a side effect, the increase of P results in an increase in computation and complexity.

M – does not affect the search resolution but it affects the signals and preprocessing filters.

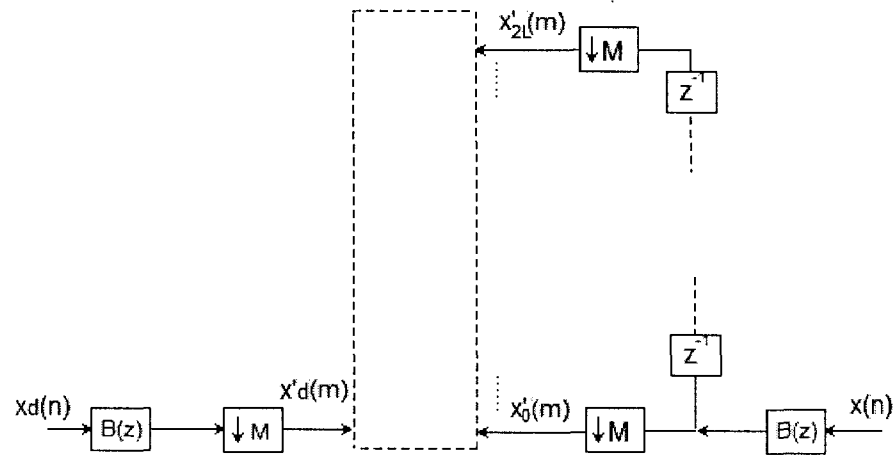


Figure 4-14 Structure for estimating the delay to the nearest sampling period

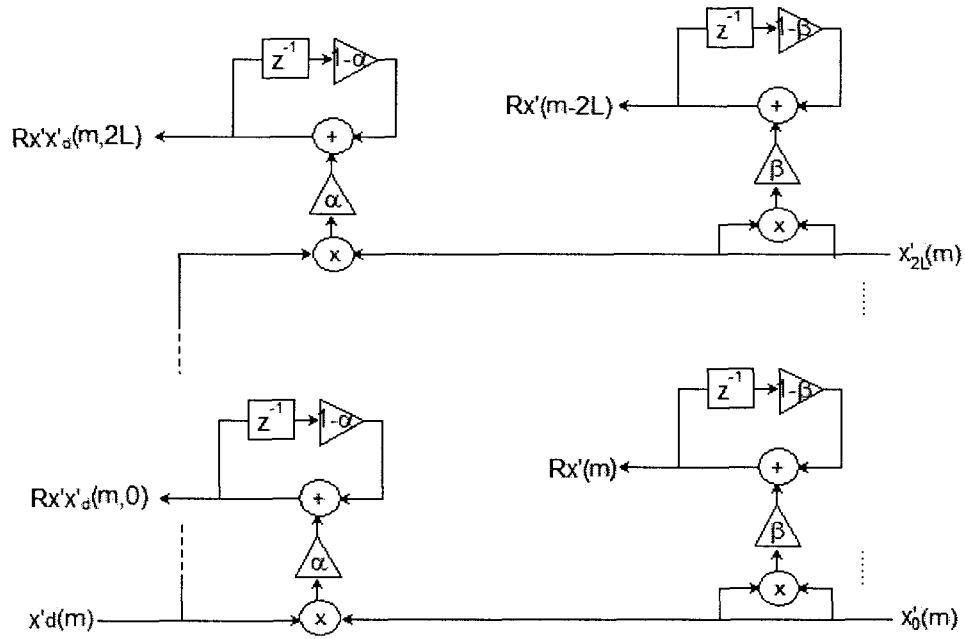


Figure 4-15 Structure for estimating the cross correlation values (dotted box in Figure 4-12)

4.6. Theoretical Analysis

In this section, the effect of preprocessing on the signals and its inherent advantages related to TDE are examined. In order to do so, we start of from the frequency domain expressions of $x(n)$ and $x_d(n)$ (see Figure 3-6), which are: $X(\omega)$ and $X_d(\omega)$ respectively. Note that:

$$X_d(\omega) = X(\omega) H(\omega) + W(\omega) \quad (4-15)$$

where $H(\omega)$ and $W(\omega)$ are the frequency responses of the echo path and additive noise respectively. In this project, it is assumed that the echo path is made up of an initial flat delay of length Δ_{opt} which precedes the active part of the impulse response (Figure 1-5), that is:

$$H(\omega) = e^{-j\omega\Delta_{opt}} H_{act}(\omega) \quad (4-16)$$

with $e^{-j\omega\Delta_{opt}}$ and $H_{act}(\omega)$ representing the frequency response of the flat delay and the active part of the echo path respectively. The purpose of the proposed TDE schemes is to estimate Δ_{opt} such that the AEC system can estimate $H_{act}(\omega)$ separately. Assuming ergodicity for all random processes, the cross correlation function between $x(n)$ and $x_d(n)$ is:

$$R_{xx_d}(d) = \sum_{n=-\infty}^{\infty} x(n)x_d(n-d) \quad (4-17)$$

Taking the Fourier transform of (4-17):

$$\begin{aligned} \Phi_{xx_d}(\omega) &= \sum_{d=-\infty}^{\infty} R_{xx_d}(d)e^{-j\omega d} = \sum_{d=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(n)x_d(n-d)e^{-j\omega d} \\ &= \sum_{d=-\infty}^{\infty} x(n)e^{-j\omega n} \left[\sum_{n=-\infty}^{\infty} x_d(n-d)e^{-j\omega(n-d)} \right]^* \\ &= X(\omega)X_d^*(\omega) \end{aligned} \quad (4-18)$$

$\Phi_{xx_d}(\omega)$ in (4-18) is the cross power spectral density (PSD) of $x(n)$ and $x_d(n)$.

In general, the wider the bandwidth of $\Phi_{xx_d}(\omega)$, the narrower is $R_{xx_d}(d)$ (as an extreme example, the cross PSD of two white signals is a constant while their cross correlation is an impulse located at the lag between the signals). So, naturally to be able to identify its peak, it helps to have $R_{xx_d}(d)$ as “narrow” as possible. This implies making the bandwidth of $\Phi_{xx_d}(\omega)$ as “wide” as possible.

The frequency response of the preprocessed signals is:

$$X'(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} B\left(e^{j\frac{\omega(2\pi k)}{M}}\right) X\left(e^{j\frac{\omega(2\pi k)}{M}}\right) \quad (4-19a)$$

$$X'_d(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} B\left(e^{j\frac{\omega(2\pi k)}{M}}\right) X_d\left(e^{j\frac{\omega(2\pi k)}{M}}\right) \quad (4-19b)$$

for $x'(n)$ and $x'_d(n)$ respectively.

The filter $B(\omega)$ selects a part of the spectrum of $X(\omega)$ and $X_d(\omega)$ where a relatively high portion of the respective energy is concentrated, while suppressing other parts of their spectrum. $X'(\omega)$ and $X'_d(\omega)$ result from the addition of shifted copies of $B(\omega)X(\omega)$ and $B(\omega)X_d(\omega)$ as indicated in (4-19a) and (4-19b). Assuming the filter $B(\omega)$ to be properly chosen, the spectrums of $X'(\omega)$ and $X'_d(\omega)$ are wider than those of $X(\omega)$ and $X_d(\omega)$, thus their cross correlation function should have a more defined peak than the cross correlation function of $X(\omega)$ and $X_d(\omega)$. Therefore the downsampling and bandpass filtering operations applied on each of the signals make their frequency response more similar to that of white noise and hence their cross correlation peak easier to detect.

4.7. Operation and Robustness

The structures for TDE proposed in this thesis sample the modified cross-correlation function at intervals of MT_s (first structure) or T_s (second structure). Ideally, if the sampling is done at an appropriate resolution and in the appropriate range, the peak can be identified. Both structures have a “search range” within which the optimal delay is assumed to lie for each search. The first algorithm (structure) is proposed to cover a wide range, while its resolution depends on the downsampling factor M . If the value of M is too

large, the resolution might not be good enough and the algorithm might “miss” a whole region where the maximum is located (Figure 4-14). The purpose of the first algorithm is not to necessarily find the true optimal delay, but rather to identify the “neighborhood” where the optimal delay is located (see Figure 4-14). It can do so by identifying a delay that is (roughly) close to the optimal one but not necessarily it. If needed, the second algorithm (structure) can be employed to scan the vicinity of the delay resulting from the first algorithm and identify the true optimal delay with better precision.

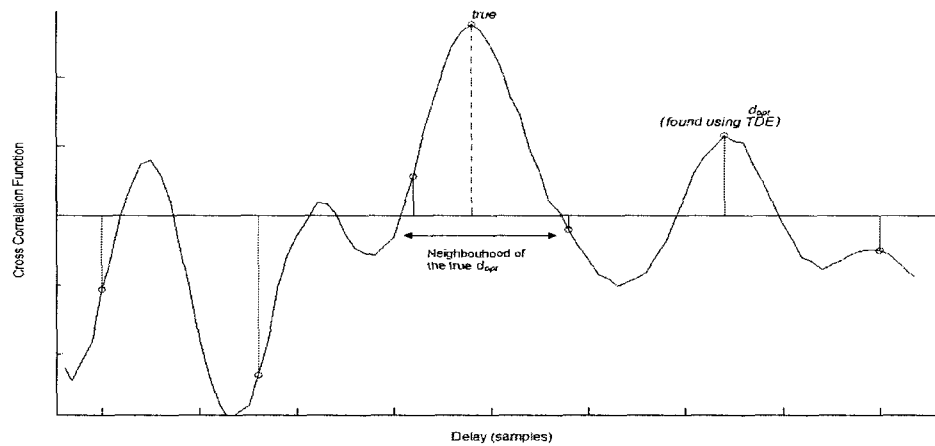


Figure 4-16 Result of TDE (first algorithm) when M is excessively large

Note that the search range is potentially limitless for both structures, but must be confined since it directly affects computational complexity.

The two structures can be used in the following possible combinations:

1. To find delay within nearest sample, when the search range is sufficiently small, the second algorithm can be used directly.
2. To find the rough value for the optimal delay, the first algorithm can be used directly if a good approximation of the optimal delay is not needed.

3. To find delay to the nearest sample when the search range is large, the first algorithm is first used to find a rough estimate for the optimal delay. The second algorithm can be subsequently used to refine the search around the first rough estimate.
4. For continuous operation applied for time varying environments - where the delay changes only by few samples - the approach in “3” above can be used to find the delay. Once the delay is determined for the first time, it is assumed that the rough estimate of the delay does not change, but rather the delay changes by a few sampling periods due to the non stationary nature of the acoustic environment. This situation can be handled by keeping the rough delay constant while updating the exact delay periodically.

The parallel search approach provides a good robustness for TDE (Figure 4-4). This is due to the fact that it introduces less signal-dependency than a sequential approach would. It is expected that the TDE algorithms perform well in the presence of noise – especially white noise, while double talk significantly impairs their operation.

4.8. Computational Cost

In this section, the computational cost for: 1. the traditional cross correlation method; 2. the GCC method; and 3. the proposed method are compared. For each case, P different cross correlation values - one for each possible delay - must be computed. It is also assumed that the computation of each cross correlation value requires a time window of K samples in order to achieve a meaningful estimate. According to [10], for calculating:

$$\rho_{x_d x}(n, k) = \frac{R_{x_d x}(n, k)}{\sqrt{R_x(n-k)R_{x_d}(n)}} \quad (4-20)$$

where $k=0,1,\dots,P-1$, one needs P multiplications, P square roots and P divisions. A division and square root operation require 16 multiplications each. Thus the total number of multiplication operations to estimate (4-20) is $33P$. Also, for each delay, one multiplication and addition operation is required to update the value for the cross correlation:

$$R_{x_d x}(n, k) = (1 - \alpha)R_{x_d x}(n-1, k) + \alpha x_d(n)x(n-k) \quad (4-21)$$

The operation (4-21) is performed K times for each of the three correlation values in (4-20). Therefore the total cost for calculating P different cross correlation values is:

$$(33 + 3K)P \quad (4-22)$$

When using (time-domain) GCC, two time-varying prefilters of length N are employed. Ignoring the computational cost for the weight update of the prefilters, the cost of GCC based TDE is:

$$2N + (33 + 3K)P \quad (4-23)$$

For the proposed structure, the two prefilters are assumed to be of length N . Also, the square root in the denominator is omitted as is one of the correlations in the denominator. The cost for TDE based on the proposed structure is:

$$\frac{2N + (16 + 2K)P}{M} \quad (4-24)$$

where M is the downsampling factor. As a result, the proposed structures for TDE have the potential of drastically reducing the number of computations when compared to GCC or even the regular cross correlation TDE scheme.

4.9 Combination of the TDE Schemes with Adaptive Filtering Algorithms for AEC

The next two sections outline possible combinations of the TDE and adaptive filtering modules. The two modules are independent and can hence be implemented using parallel processing techniques in single or multi-processor systems. Typically, the TDE module should be run before adaptive filtering to identify the flat delay. Once the flat delay is determined, AEC with adaptive filtering can proceed normally. The operation of the latter can be interrupted and TDE can “jump in” in case the flat delay has changed and needs to be estimated again.

4.9.1 TDE and Compensation in Series with Adaptive Filtering

The pure delay Δ introduced into the path of the input signal is the estimate of the flat delay present in the acoustic echo path. By inserting this delay in series with the adaptive filter, the latter is “spared” the task of estimating and compensating the flat delay itself. In a sense, the input signal is “centered” with respect to the echo signal. Another way of viewing this problem is by considering the pure delay and active part of the acoustic echo path as two separate systems connected in series. This arrangement is expressed in (4-16) and repeated here for convenience:

$$H(\omega) = e^{-j\omega\Delta_{opt}} H_{act}(\omega)$$

where $H(\omega)$ is the frequency response of the acoustic echo path and $H_{act}(\omega)$ is the frequency response of the active section of the acoustic echo path. The approach proposed in this thesis consists of:

1. Estimating the value for Δ is using TDE and compensating this pure delay into the input signal path;
2. Estimating the impulse response of $H_{act}(\omega)$ using adaptive filtering and compensating this into the input signal path in series with the pure delay above.

The general structure used for combining TDE and compensation with adaptive filtering was given in Figure 3-9. Figure 4-15 provides a more transparent view of this structure.

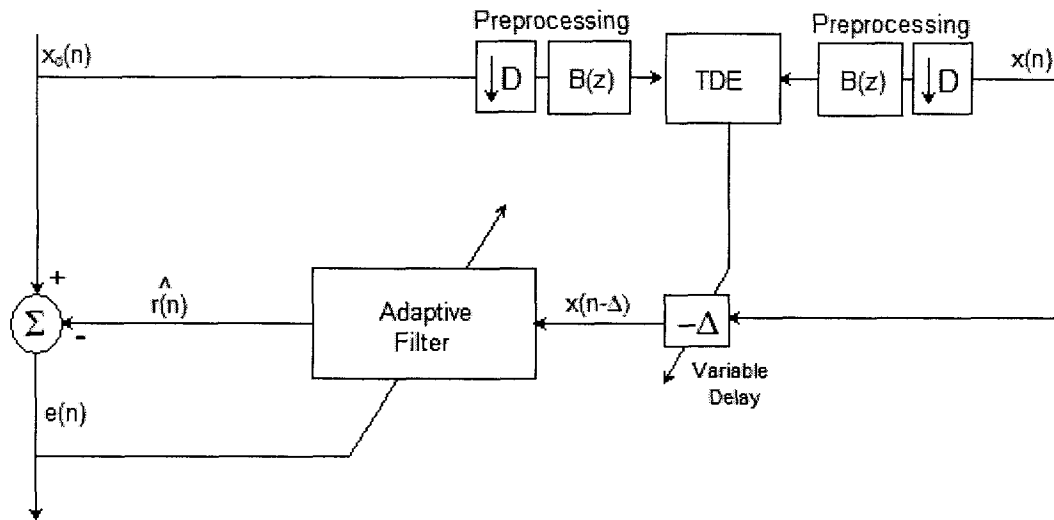


Figure 4-17 Adaptive filtering with TDE and time delay compensation

4.9.2 TDE and Compensation Exploiting SAF Structures

Subband Adaptive Filtering (SAF), which was treated in Chapter 3, involves splitting the input signals into a number of subbands. The operations performed to obtain each of the resulting subband signals using analysis filter banks are similar to the preprocessing step used in the proposed TDE structure. This implies that when adaptive filtering is performed using SAF, the subband signals could be directly used as preprocessed signals. In this way the preprocessing step can be omitted and further computational savings can be achieved. The combination of the SAF with TDE is depicted in Figure 4-18.

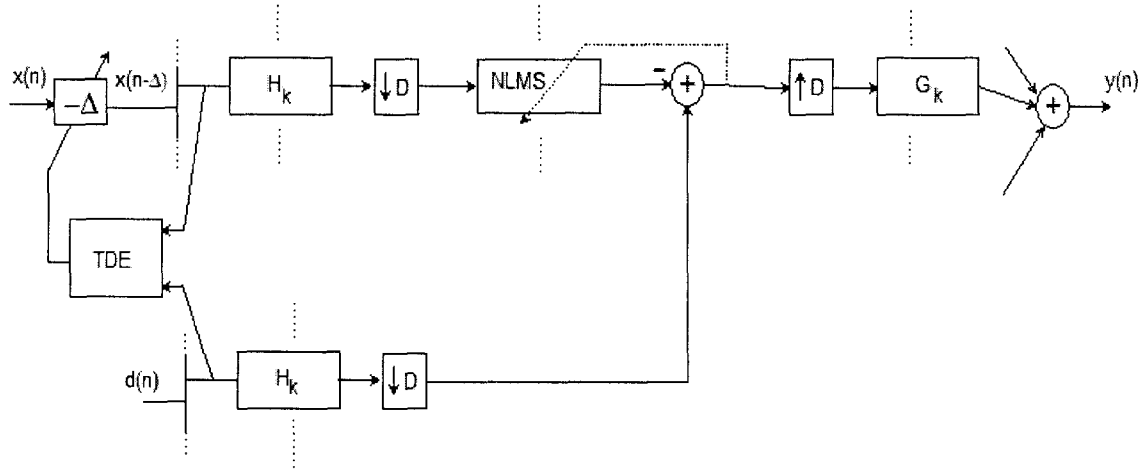


Figure 4-18 TDE combined with SAF for AEC

In Figure 4-18, the k -th subband of each signal was chosen as input to the TDE algorithm. The value of k is chosen to include the section of the signal spectrum with the highest energy. Note that the role of the preprocessing filter $B(z)$ is played by the analysis filter $H_k(z)$.

Although the approach discussed in this section allows for a drastic decrease in computation, it has two limitations: first, it is only applicable when SAF are being

employed; second, it does not allow for as much control over the selection of the preprocessing functions since the downsampling operation and the decimation filter are both dependent on the SAF structure.

CHAPTER 5

SIMULATIONS

5.1 Performance Measures

To quantify the performance of the proposed TDE system coupled with the AEC system, a few performance metrics must first be introduced.

Echo Return Loss Enhancement (ERLE) – a popular performance measure for Echo cancellation systems [14]. The ERLE is defined as the ratio of the energy of the incoming echo with the energy of the residual echo, that is:

$$\text{ERLE} = 10 \log_{10} \frac{E\{d^2(n)\}}{E\{e^2(n)\}} \quad (5-1)$$

where $d(n)$ is the incoming echo signal and $e(n)$ is the residual echo signal after AEC is performed. The ERLE is approximated in practice using sliding windows:

$$\text{ERLE}(n) = 10 \log_{10} \frac{\sum_{k=0}^{N_w-1} d^2(n-k)}{\sum_{k=0}^{N_w-1} e^2(n-k)} \quad (5-2)$$

In this case, $d(n)$ is the microphone input, $e(n)$ is the error signal coming from the AEC system and N_w is the window size.

System distance – indicates the quality of the convergence of the adaptive system by measuring the difference between the tap weight vector and the ideal Wiener solution:

$$sd(n) = 10 \log_{10} \left(\frac{\|\mathbf{h} - \mathbf{h}(n)\|^2}{\|\mathbf{h}\|^2} \right) \quad (5-3)$$

In this thesis, instead of the optimal Wiener solution, the given acoustic echo path impulse response will be used. Specifically, given the actual impulse response to be:

$\mathbf{h}=[h_0, \dots, h_{L-1}]$ and the approximated response: $\hat{\mathbf{h}}=[\hat{h}_0, \dots, \hat{h}_{N-1}]$, then the system distance is approximated using the expression:

$$sd(n) = 10 \log_{10} \left[\sum_{i=0}^{N-1} (h_i - \hat{h}_i(n))^2 \right] \quad (5-4)$$

Reverberation time – is an indicator of the amount of reverberation present in an acoustic environment. It is defined as the time it takes for the impulse response of the acoustic environment to fall 60dB from its peak value.

Signal-to-Noise Ratio (SNR) - defined as:

$$SNR = 10 \log_{10} \frac{\sum_{k=0}^{N-1} s^2(n-k)}{\sum_{k=0}^{N-1} w^2(n-k)} \quad (5-5)$$

where $s(n)$ is the signal and $w(n)$ is the additive noise.

5.2 Test Signals

The test signals used in this thesis were speech signals taken from two sources:

1. The .wav file in Matlab called *dft_speech.wav* located in the *Matlab\toolbox\dspblks* folder in Matlab. This is a 2-channel .wav file with sampling frequency of 22050Hz (Figure 5-2).
2. Recordings using a PC platform and a microphone using *Audacity* V. 1.2.6 digital audio editor [43]. The speech signals were recorded at 44.1kHz and resampled down to 16kHz. The resampling was performed in Matlab using the function: $y = \text{resample}(x, p, q)$ which resamples the sequence in vector x at p/q times the original sampling rate, using a polyphase filter implementation [44].

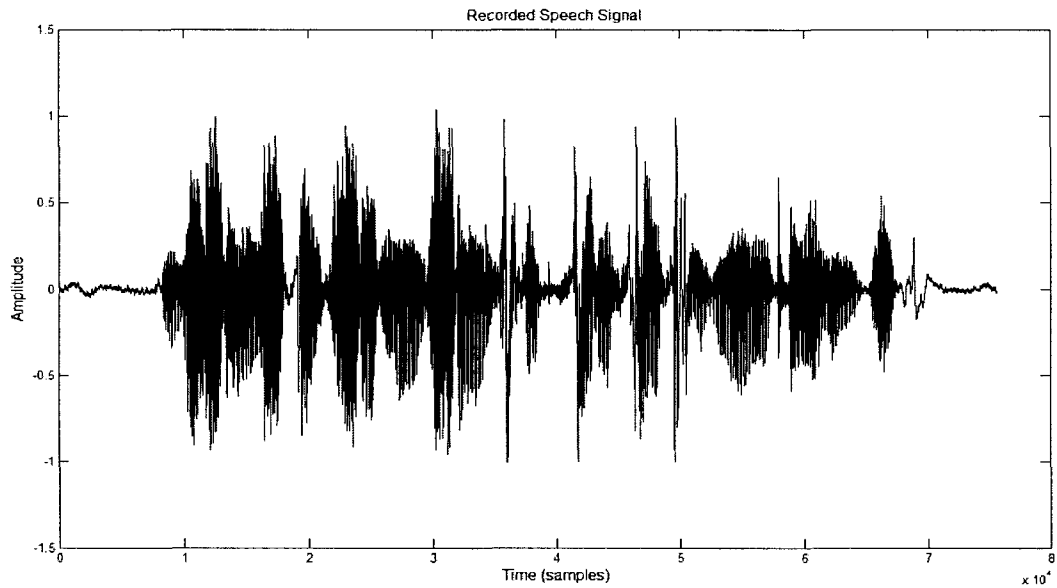


Figure 5-1 Recorded speech signal

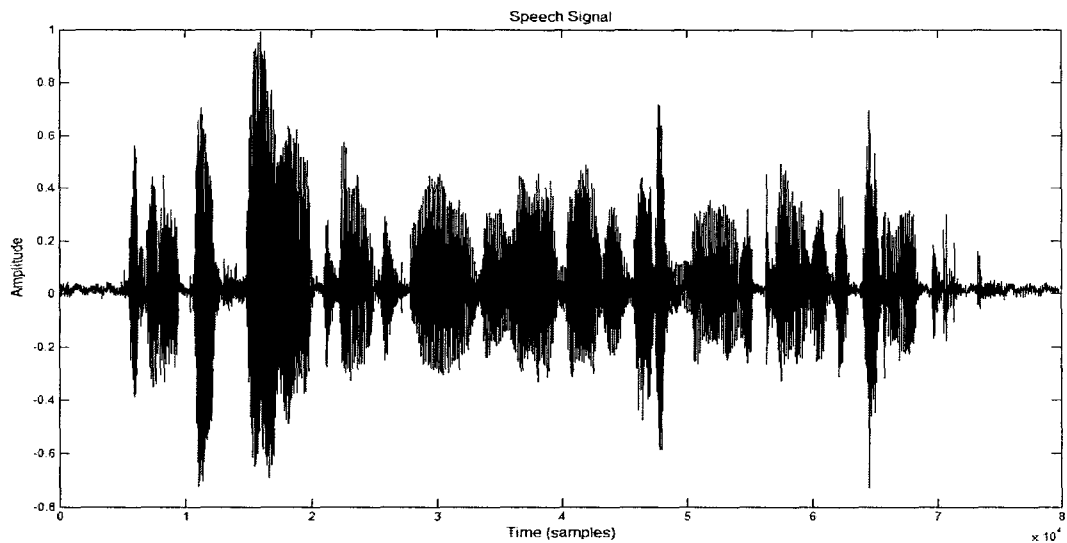


Figure 5-2 Speech signal *dft_speech.wav*

The time-varying frequency response of the speech signal can be visualized using the spectrogram. This computes the magnitude of the windowed discrete-time Fourier transform of a signal using a sliding window. The spectrogram of the speech signal in Figure 5-1 is depicted in Figure 5-3 using 1024 sample time windows.

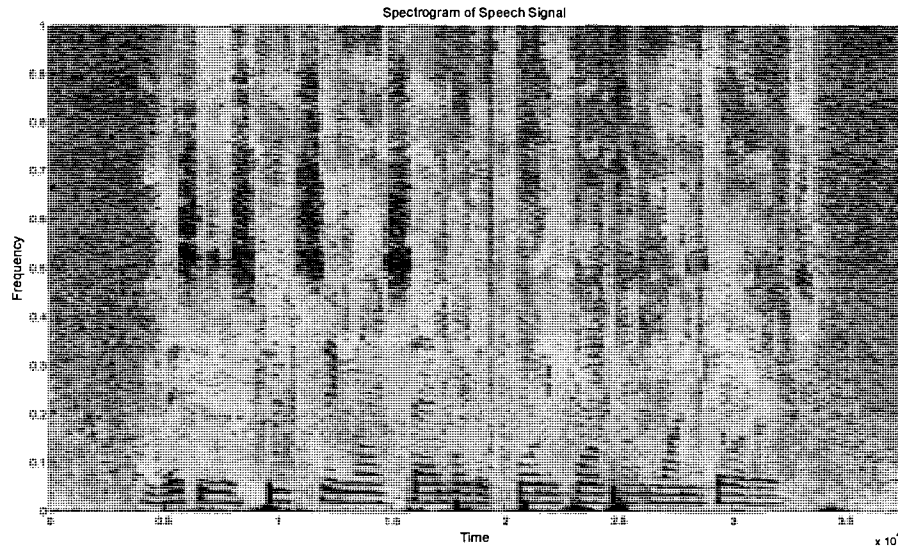


Figure 5-3 Spectrogram of the speech signal in Figure 5-1

5.3 Acoustic Echo Path Responses

The main purpose of the AEC and TDE systems is to model the active part and flat delay part of the acoustic echo path response respectively. Acoustic echo paths vary with time due to factors such as the relative motion of receiver and transmitter as well as motion of objects within the acoustic environment [13], [14], [18]. It is assumed that only the first factor (relative motion of receiver and transmitter) influences the flat delay region of the acoustic echo path response significantly enough to require “full” TDE processing (i.e. employing TDE in a large search range). The effect the second factor - motion of objects within the acoustic environment - is assumed to have minimal influence on the echo path and as a result only require a search within a small range of possible delays (i.e. it is assumed that the flat delay only changes by a small number of samples, thus only requiring the use of the second proposed algorithm).

Four different echo path responses are used to test the performance of the AEC-TDE system: `office1.mat`, `office2.mat`, `room_highreverb.mat`, `room_highreverb1.mat`. The impulse responses differ in length, reverberation time and flat delay. The first two impulse responses are characteristic of small rooms and offices (Figure 1-4 depicts `office2.mat`). The next two impulse responses are characteristic of larger rooms and have more reverberation than the first two (Figure 1-5 depicts `room_highreverb.mat`). The impulse responses are generated in Matlab and their respective flat delay is artificially altered as needed.

5.4 Simulation Results

This chapter illustrates the results from a series of simulations which were performed to test the robustness and speed of convergence of the proposed algorithms as well as their dependence on certain parameters. The performance of the proposed algorithm is tested through three stages, each of which deals with one aspect of the algorithms. In the first stage, the quality of the preprocessing procedure is tested. The ability of the proposed algorithms to identify the optimal flat delay is demonstrated in this stage. The proposed TDE algorithm is compared to two other candidate algorithms for real-time TDE: cross correlation and GCC. In the second series of simulations, the performance gain in AEC using fullband and subband NLMS in combination with the proposed algorithms is demonstrated. The last set of simulations demonstrates real-time operation of the proposed algorithms.

All the simulations are performed using *Matlab* version 7 on a computer working with an *Intel Centrino Duo T2450* processor at 2GHz with 2GB of RAM.

5.4.1 Performance Comparison of TDE Algorithms

In this section, the proposed TDE algorithm is compared to the performance of:

1. The classical cross-correlation based TDE, and
2. GCC-based TDE.

The above algorithms are the only ones which are suitable for real-time and single-channel performance.

The setup for the experiment is shown in Figure 5-4. The TDE algorithm must identify the flat delay that the echo path introduces into the input signal.

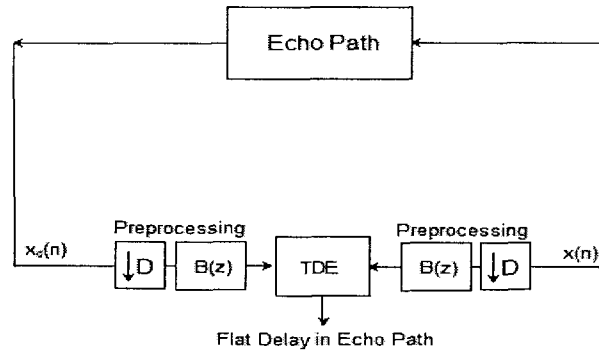


Figure 5-4 Experiment setup to test TDE algorithm performance

The results are shown in the form of histograms that depict how many times the optimal delay is identified as being within a range of width 10 samples. To achieve the desired results, the following data were used:

Two echo path impulse responses, the first one with a lower reverberation than the second. The impulse responses are *office2.mat* which has a flat delay of 200 samples, and *room_highrerb.mat* which has a flat delay of about 350 samples;

Thirty eight (38) different speech segments extracted from each of eight (8) speech signals. Four of the speech signals originate from a male speaker and four from a female speaker. The result is three hundred and four (304) different speech segments, all of which were used on each echo path impulse response. Each speech segment has a sampling rate of 16kHz and a length of 512 samples (32 ms).

The following parameters were used when evaluating the ability of each algorithm to detect the optimal flat delay:

The optimal delay is searched for within a window of length 200 samples, e.g. if the optimal flat delay is 200 samples, the TDE algorithms are applied for delays 100 to 300; To approximate the cross correlation function for each of the delays within the interval above, a time window of 512 samples (32 ms) is used.

Figures 5-5 to 5-9 depict the histograms resulting from testing the respective TDE algorithm on 140 of the speech segments which were “filtered through” the echo path response `office2.mat`. The proposed algorithm is tested for different values of the downsampling factor ($M = 8, 10$ and 16).

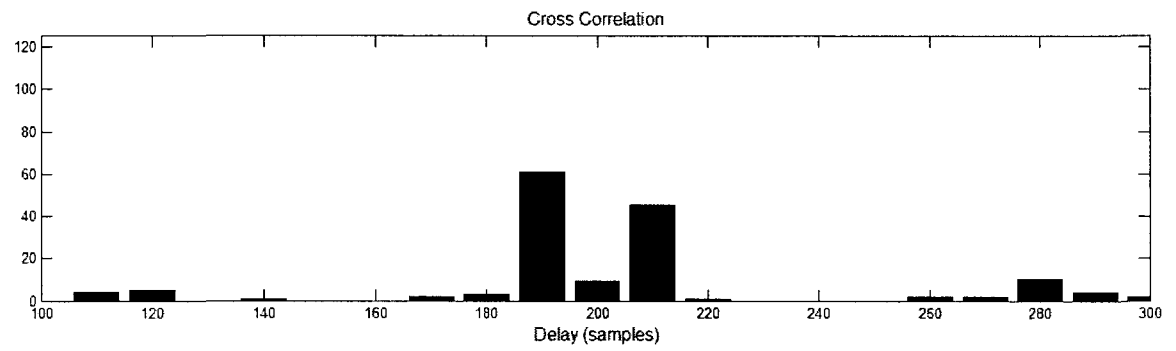


Figure 5-5 TDE algorithm performance – cross correlation

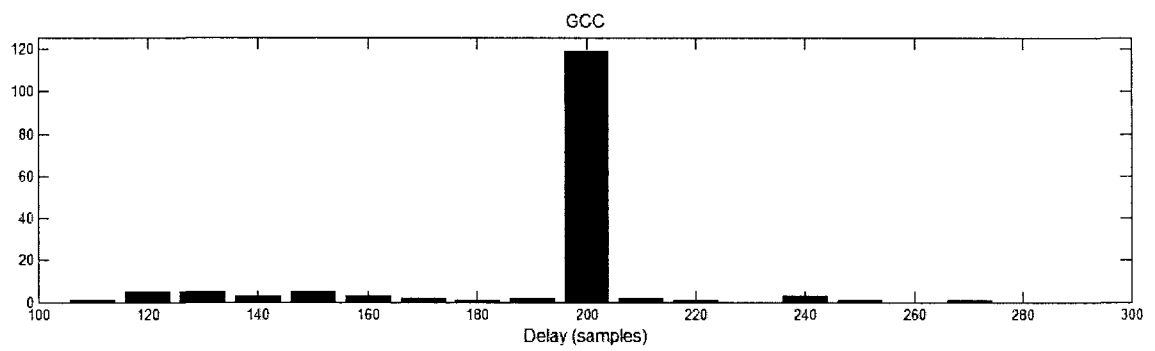


Figure 5-6 TDE algorithm performance – GCC

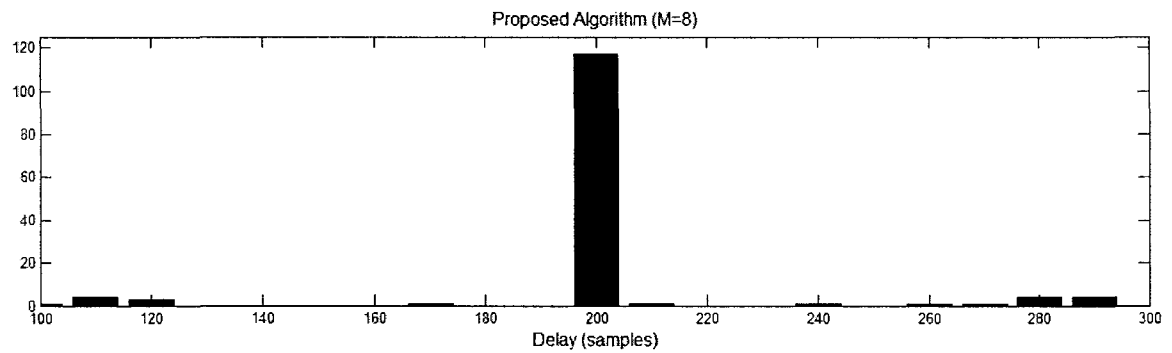


Figure 5-7 TDE algorithm performance – proposed (M=8)

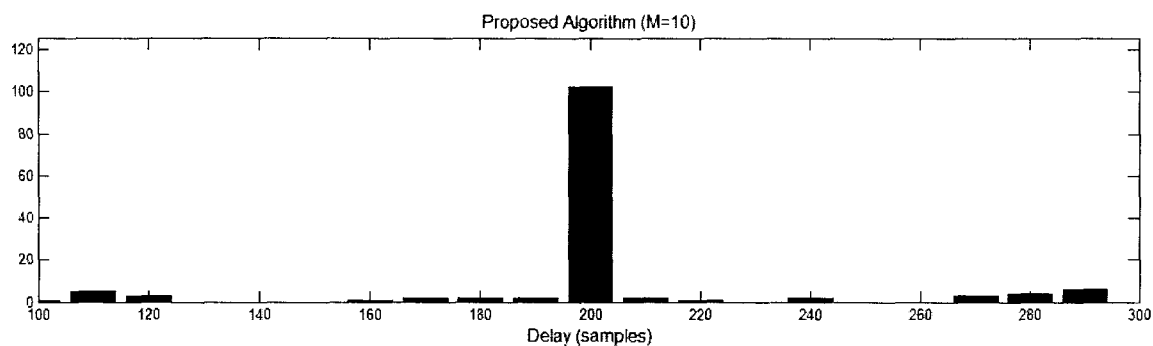


Figure 5-8 TDE algorithm performance – proposed (M=10)

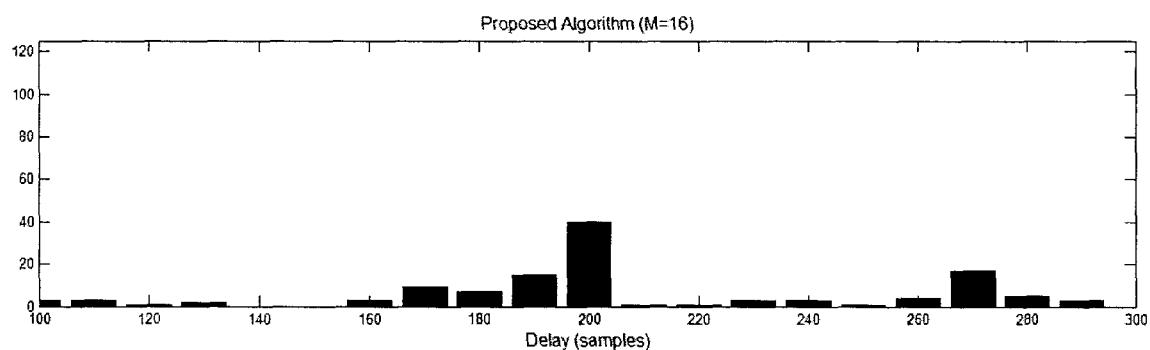


Figure 5-9 TDE algorithm performance – proposed (M=16)

Figures 5-10 to 5-14 depict the histograms resulting from testing the respective TDE algorithms on 140 of the speech segments which were “filtered through” the echo path response `room_highreverb.mat`. The proposed algorithm is again tested for different values of the downsampling factor ($M = 8, 10$ and 16).

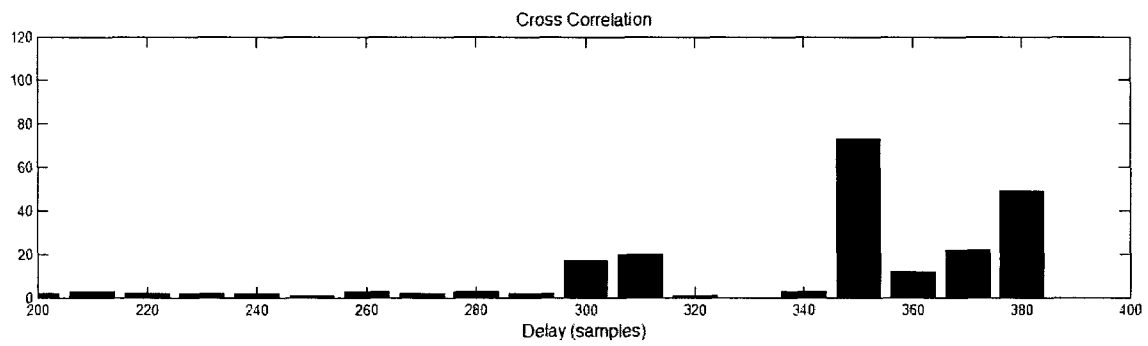


Figure 5-10 TDE algorithm performance – cross correlation

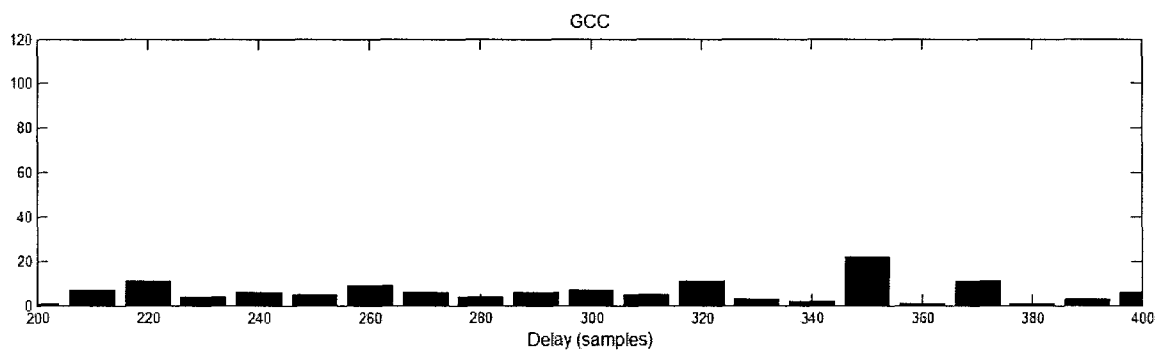


Figure 5-11 TDE algorithm performance – GCC

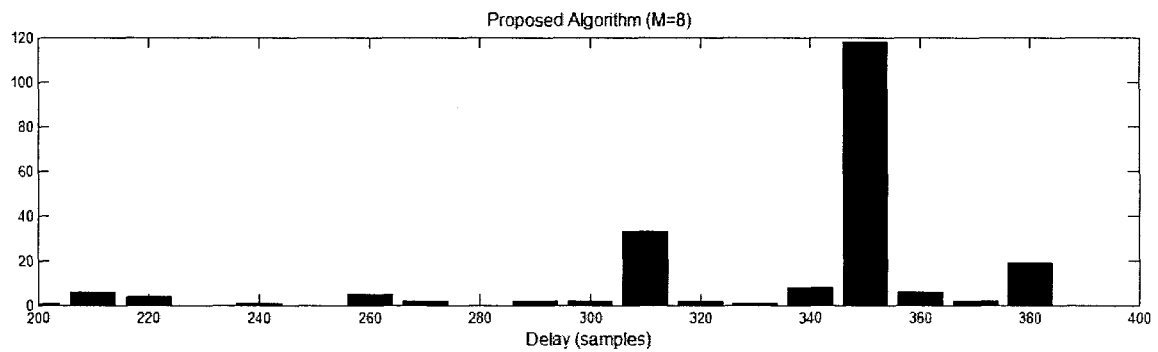


Figure 5-12 TDE algorithm performance – proposed (M=8)

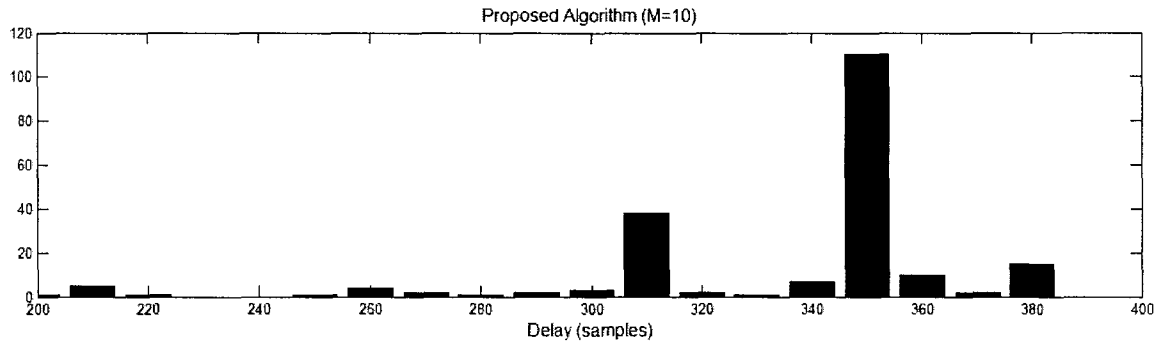


Figure 5-13 TDE algorithm performance – proposed (M=10)

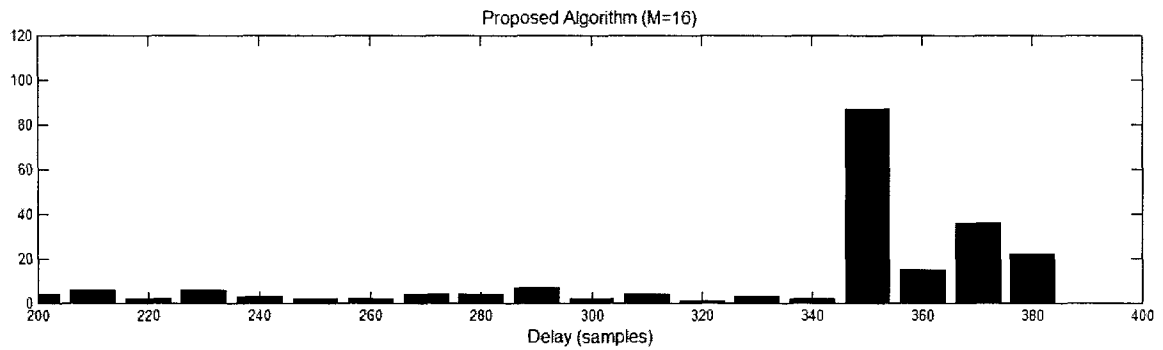


Figure 5-14 TDE algorithm performance – proposed (M=16)

When considering the environment with low reverberation (implemented by `office2.mat`), the proposed algorithm with $M=8$ and $M=10$ outperforms the classical cross correlation algorithm and performs similarly to the GCC.

On the other hand, in the reverberant environment (implemented by `room_highreverb.mat`), the proposed algorithm with $M=8$ $M=10$ and $M=16$ outperforms all other algorithms. A summary of the results is given in Table 5-1.

Table 5- 1 Percentage of correct identification of the optimal delay

	<i>Cross-Corr.</i>	<i>GCC</i>	<i>Proposed Algorithm</i>		
			<i>M=8</i>	<i>M=10</i>	<i>M=16</i>
<i>Low</i> <i>Reverb.</i>	44%	85%	85%	73%	29%
<i>High</i> <i>Reverb.</i>	50%	16%	85%	79%	62%

5.4.2 TDE and Time Delay Compensation with Adaptive Filtering

The experimental results illustrated in this section demonstrate the performance gain of using time delay estimation and compensation for AEC. The experiment performed in this stage consists of observing the performance of the adaptive filter for different values of the compensated delay Δ (see Figure 4-17).

First, white Gaussian noise is used as input and its filtered version (through the echo path `office2.mat`) is used as the desired signal. Adaptive filtering is performed using:

Fullband adaptive NLMS of length 512 taps;

Subband adaptive NLMS with $M=16$ $D=8$. The NLMS filter in each channel is of length 64 taps; the analysis and synthesis filters are of length 12 samples.

The maximum and average ERLE are plotted as a function of the value of the compensated delay (Figure 5-15 and 5-16) which is incorporated into the input signal.

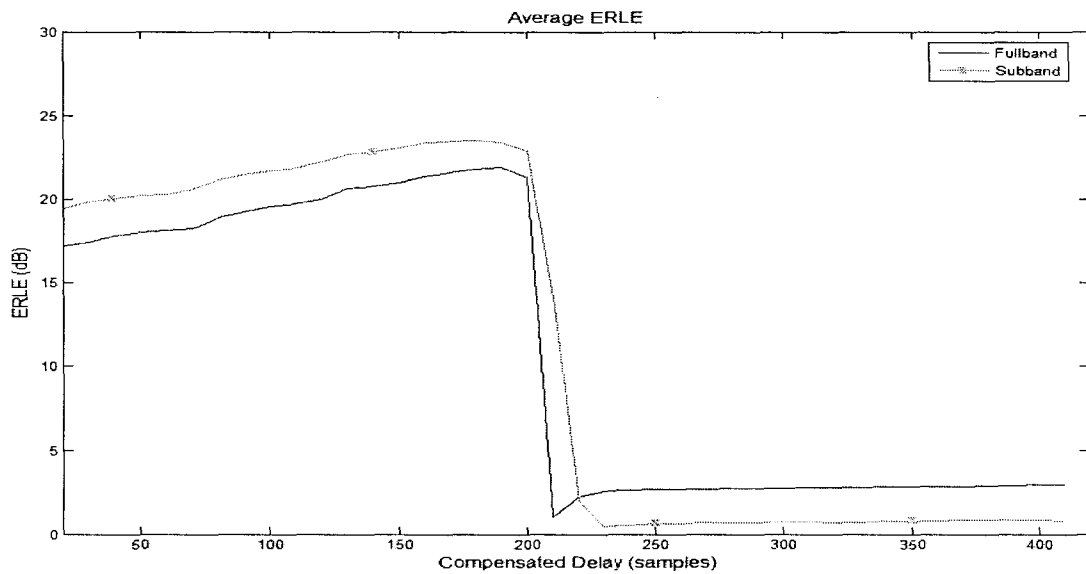


Figure 5-15 Average ERLE as a function of the compensated delay

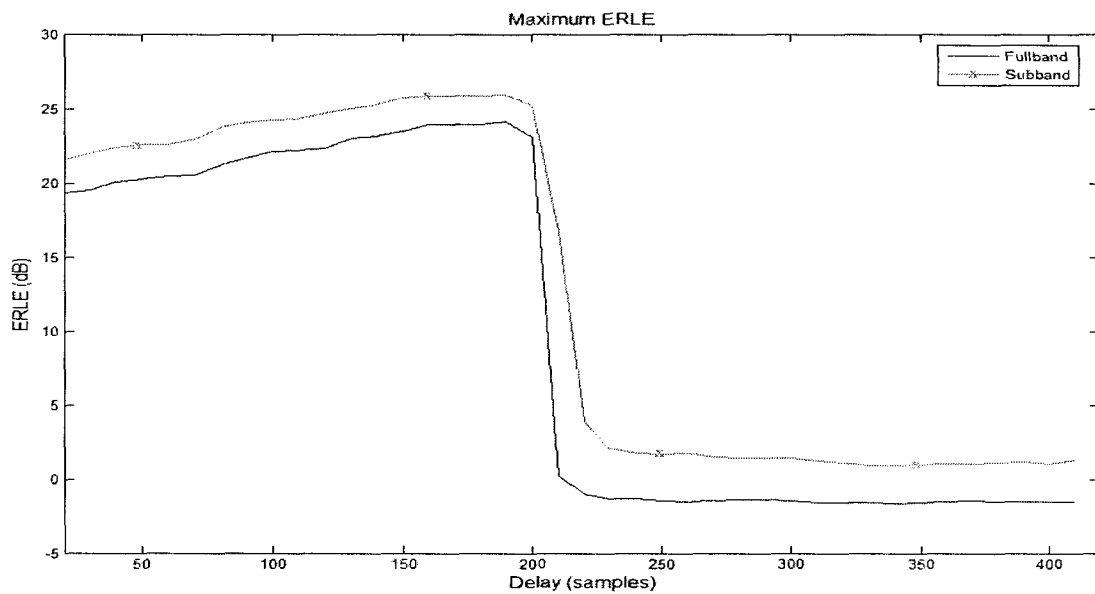


Figure 5-16 Maximum ERLE as a function of the compensated delay

The results above show the increase in AEC performance as the compensated delay approaches the optimal flat delay. After the compensated delay exceeds the value of the

optimal flat delay, the ERLE quickly drops. This happens because the AEC coefficients fail to cover the beginning of the active part of the echo path response, which are the coefficients with the highest amplitude. Thus care must be taken not to overestimate the value of the flat delay.

In order to better visualize the possible gain of using time delay compensation, the dependence of the ERLE on the length of the adaptive filter was analyzed. Figures 5-17 and 5-18 depict the maximum and average ERLE as a function of the filter length for fullband and subband adaptive filters respectively.

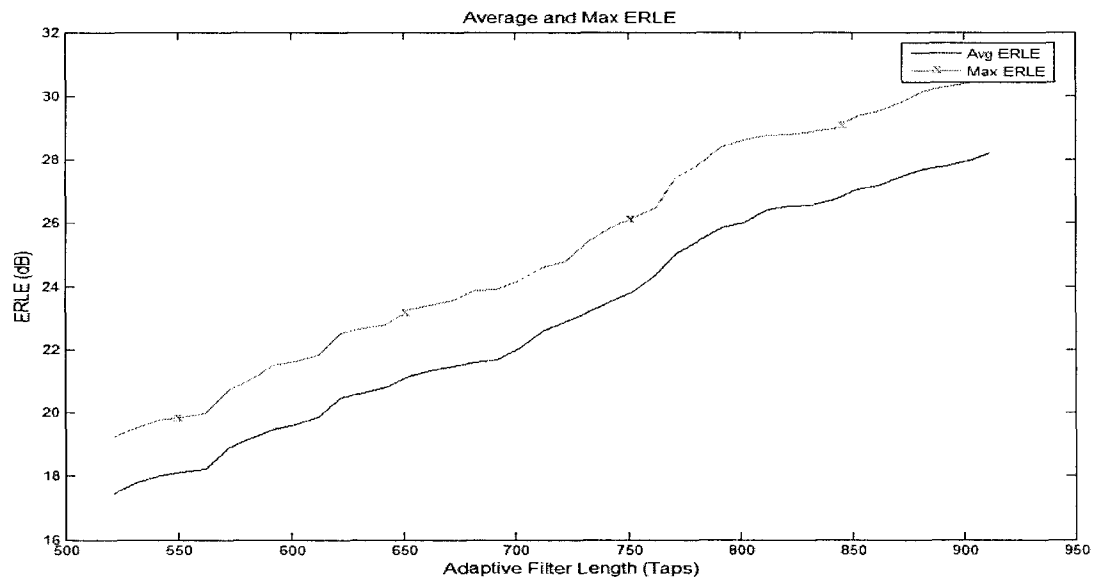


Figure 5-17 Average and maximum ERLE as a function of the adaptive filter length (fullband)

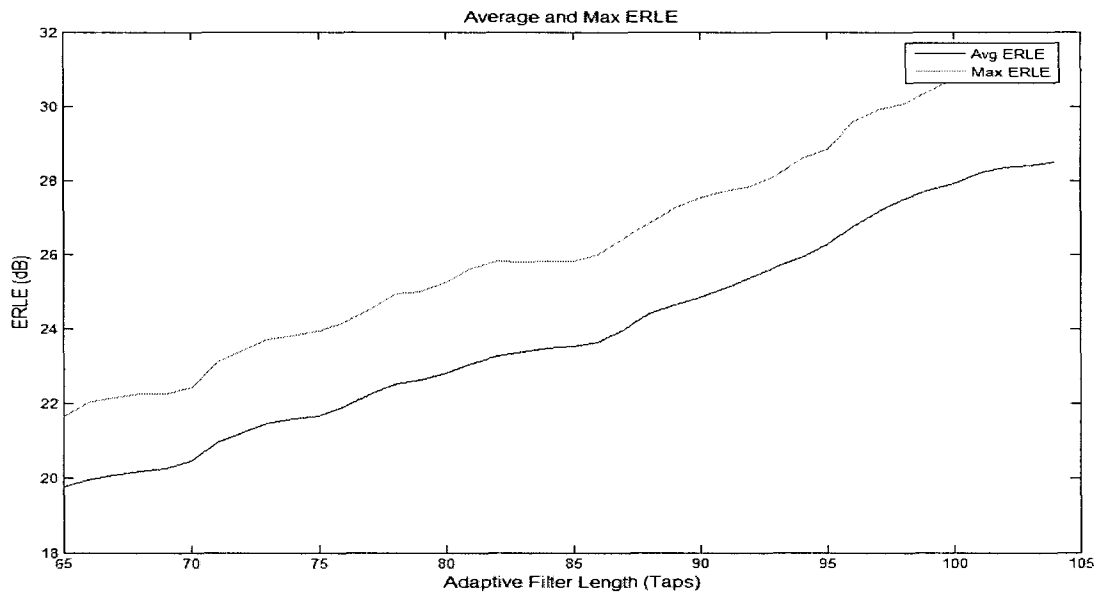


Figure 5-18 Average and maximum ERLE as a function of the adaptive filter length (subband)

Note that, to achieve the same ERLE improvement that was achieved by time delay compensation, an additional 190 taps was needed for the fullband adaptation and 20 taps for the subband adaptation. This is equivalent to an 18% and 31% saving in filter coefficients respectively.

Similarly to the approach used so far, the following experiments will demonstrate the ERLE improvement when the flat delay present in the echo path is compensated. In these simulations, however, speech signals will be used as opposed to white noise for the evaluation of the algorithms in question. The results are generated using four speech signals and two different echo path impulse responses (office2.mat and room_highreverb.mat). Gaussian white noise is inserted to achieve an SNR = 10dB.

Adaptive filtering is performed using:

1. Fullband adaptive NLMS of length 512 taps;

2. Subband adaptive NLMS with $M=16$ channels and decimation factor $D=8$. The NLMS filter in each channel is of length 92 taps; the analysis and synthesis filters are of length 192 samples.

Figures 5-19 to 5-22 depict the maximum and average ERLE values as a function of the compensated delay.

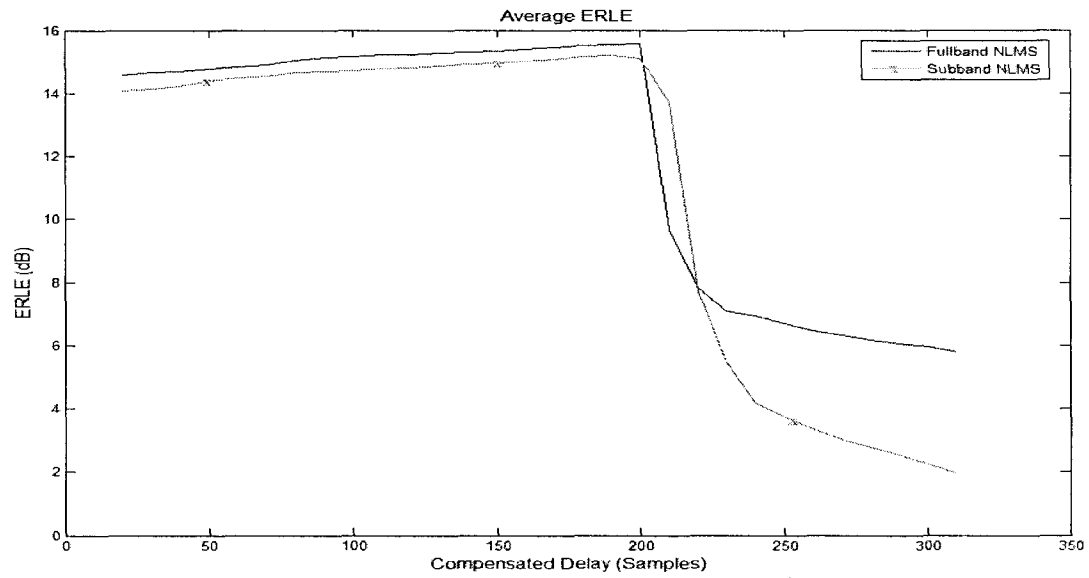


Figure 5-19 Average ERLE as a function of the compensated delay (office2.mat)

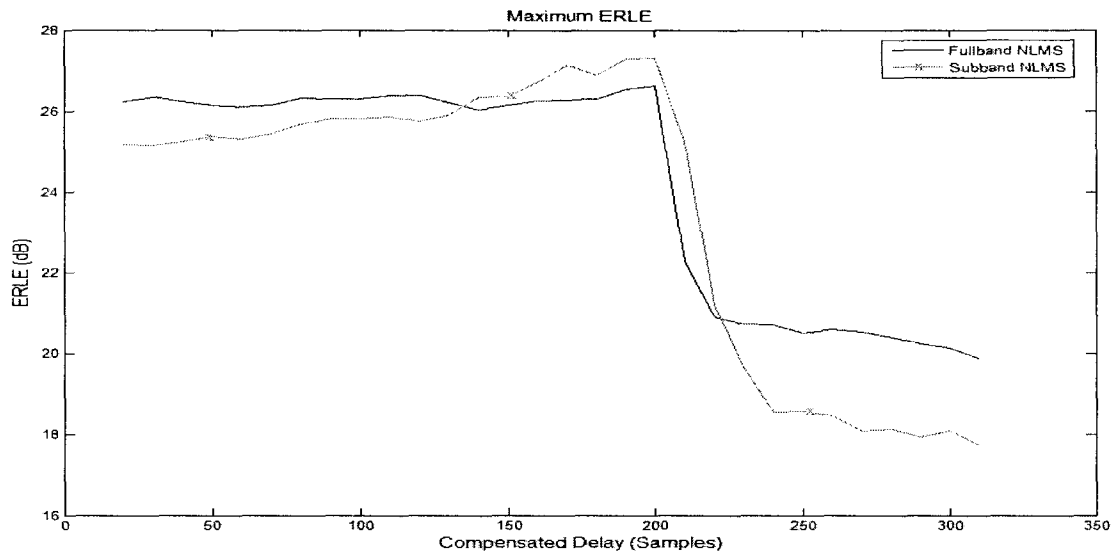


Figure 5-20 Maximum ERLE as a function of the compensated delay (office2.mat)

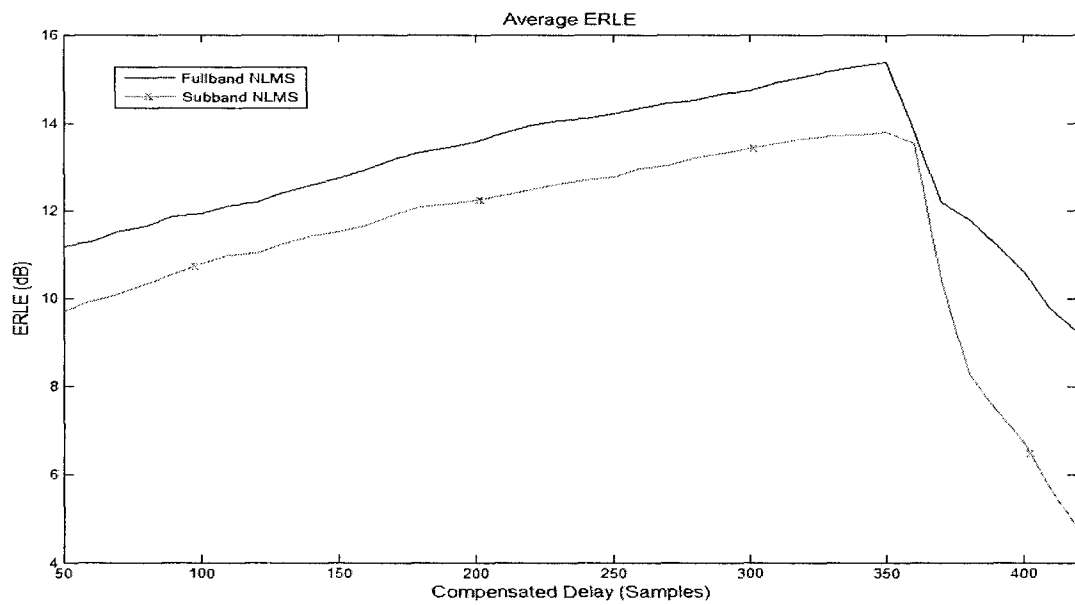


Figure 5-21 Average ERLE as a function of the compensated delay (room_highreverb.mat)

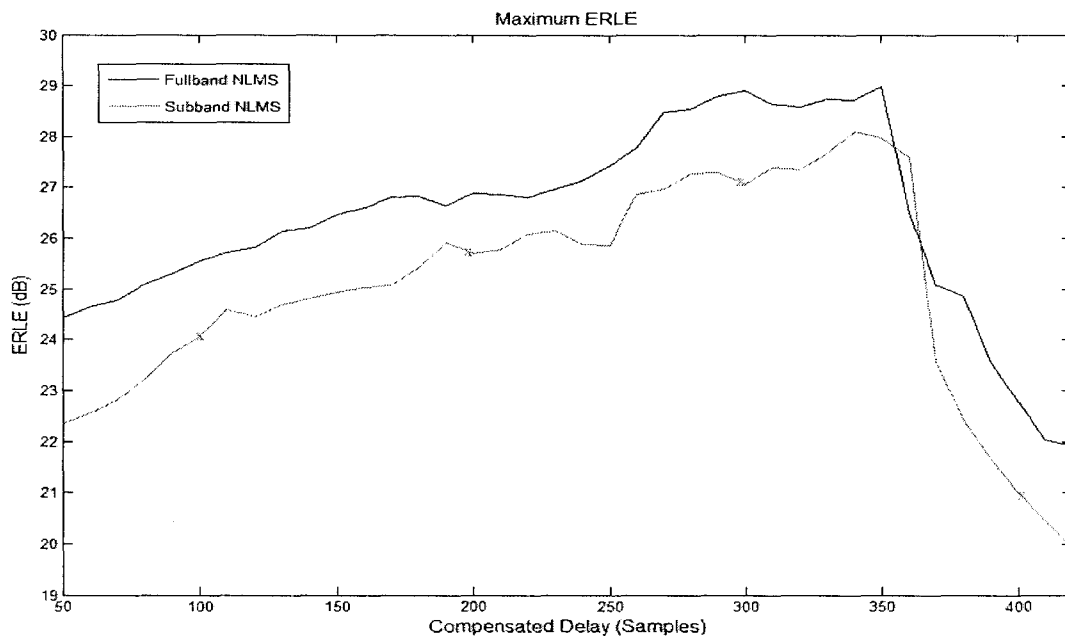


Figure 5-22 Maximum ERLE as a function of the compensated delay (room_highreverb.mat)

Based on the results of the experiments conducted, the following conclusions can be drawn:

Compensating the flat delay section of the echo path into the input signal improves the performance of the AEC;

1. Employing TDE and time delay compensation allows for the use of smaller adaptive filters to achieve a given AEC performance

5.4.3 Real Time Operation

To visualize the operation of the rough TDE algorithm, the following experiment shows the result of using the proposed algorithm in real time with the flat delay abruptly changing values three times (at respectively 200,250,300 and 150 samples).

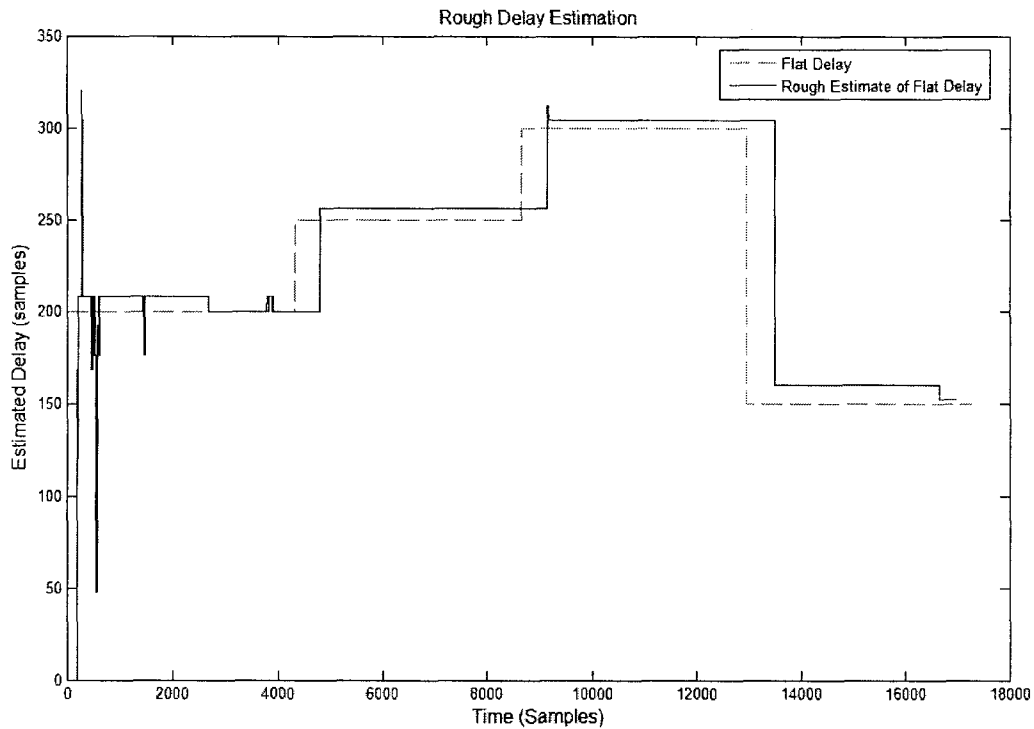


Figure 5-23 Operation of the proposed scheme for rough delay estimation

There is an inherent delay in the algorithm above due to the preprocessing step and the computation of the cross correlation. Nonetheless, it is apparent that the proposed method is robust and apt for real-time operation. The proposed TDE algorithm roughly estimates the optimal flat delay. The exact TDE scheme can be employed to obtain a more refined estimate of the flat delay.

In order to get a better view at how the estimated delay evolves in real time, Figure 5-24 depicts a zoomed-in view of a section of Figure 5-23. Note that the TDE module is activated at sample 199. This means that in this case it took $10 \times 8 = 80$ samples - or 10ms - to achieve an estimate of the rough delay value.

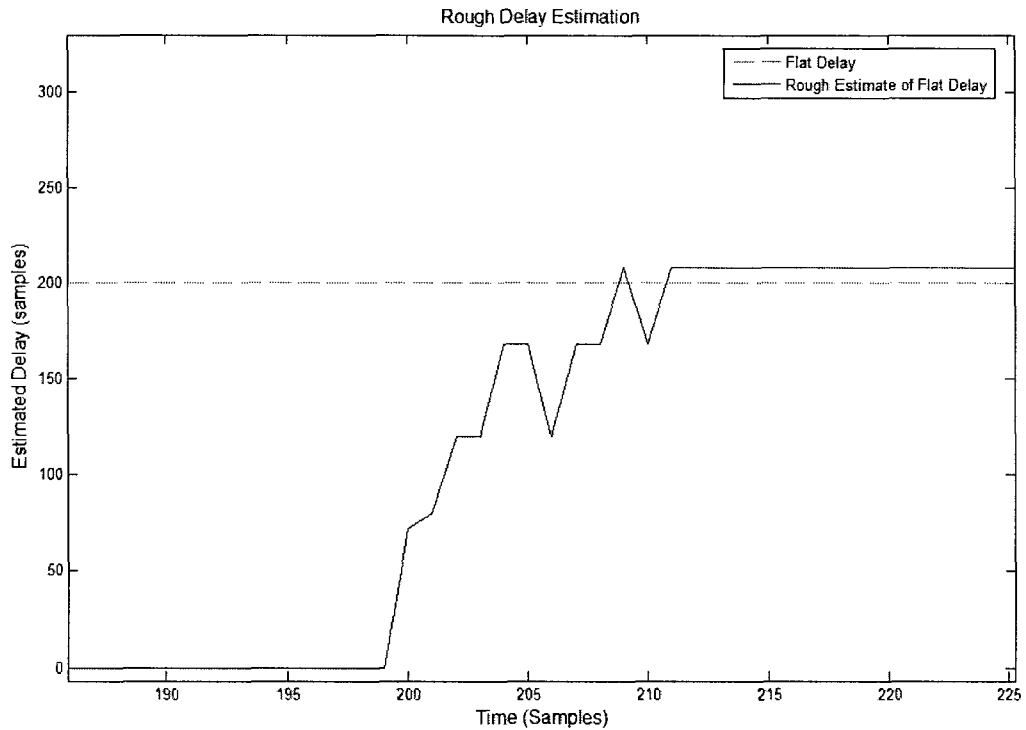


Figure 5-24 Operation of the proposed scheme for rough delay estimation – detailed view

Next, the operation of the exact delay estimation algorithm is examined in conditions that simulate real-time operation. This experiment consists of applying the proposed algorithm in real time with the flat delay abruptly changing values three times (at respectively 200, 199, 202 and 203 samples). Figures 5-25 and 5-26 depict the result of the aforementioned experiment. A close-up view of the initial stages of the process of estimating the exact delay is given in Figure 5-27. It is evident from this figure that the exact TDE algorithm takes much longer to achieve the optimal estimate than its rough-delay counterpart. In this case, a period longer than 1s is needed to achieve the desired estimate - which might not be practical if the optimal delay changes within this 1s interval.

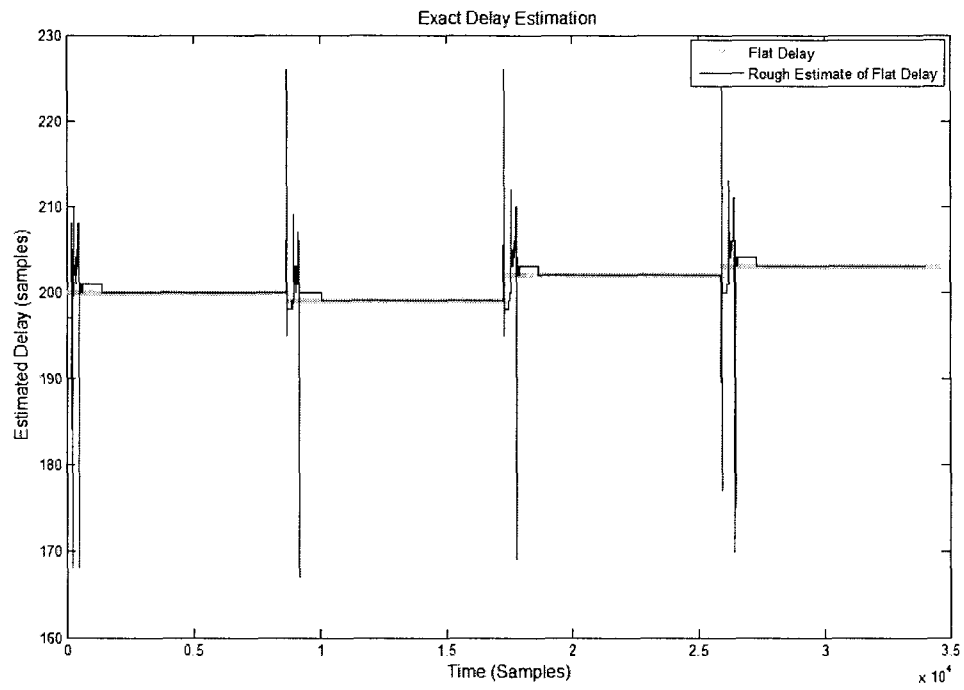


Figure 5-25 Operation of the proposed scheme for exact delay estimation

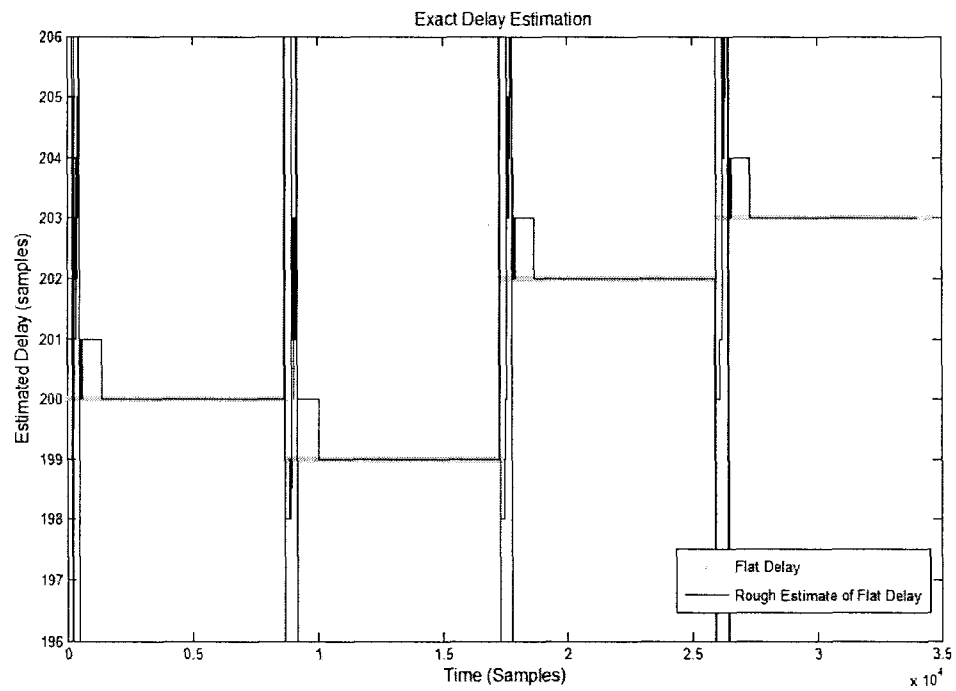


Figure 5-26 Operation of the proposed scheme for exact delay estimation

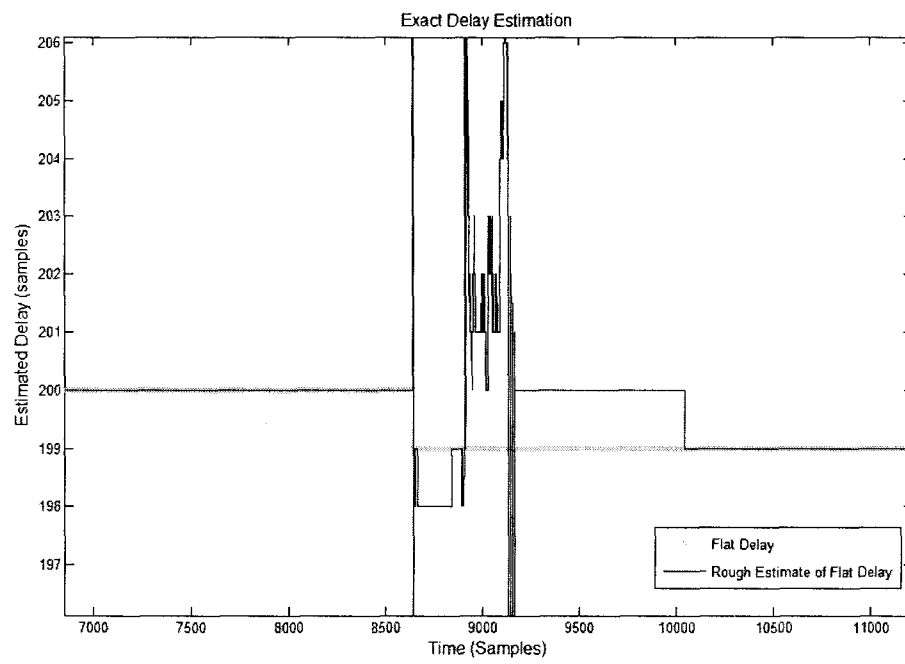


Figure 5-27 Operation of the proposed scheme for exact delay estimation – detailed view

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis introduced a TDE algorithm which is based on a parallel search of the cross correlation function. The signals whose relative time delay needs to be estimated are initially preprocessed. The preprocessing stage is meant to increase the accuracy of the TDE algorithm by modifying the spectrum of the signals in question. In addition, preprocessing works at a sampling rate which is a fraction of that of the input signals, thus the possibility for computational savings is introduced. For the purposes of this thesis, the signals in question – whose respective time delay needs to be estimated – correspond to a speech signal and its filtered version plus noise. The intended application is estimating the pure delay which an acoustic echo path – which is in fact modelled by a filter – introduces into a speech signal. The value of this delay – also known as the “flat delay” - is then compensated into the path of the signal thus “centering” the signals. This procedure saves effort and resources which would have otherwise been - inaccurately - employed by the AEC adaptive filter itself.

The algorithm proposed in this thesis focuses on two aspects:

1. Accuracy – the presence of speech signals and reverberation in general deteriorates the accuracy of TDE algorithms and calls for additional “fixes” at the cost of computational complexity;
2. Low computational cost – the proposed TDE algorithm is employed in real-time AEC. This calls for low computation and fast convergence.

The first issue above is addressed by the introduction of the preprocessing step. The second issue is addressed by introducing downsampling and parallel search. The resulting algorithm represents a step forward when compared to existing approaches such as cross correlation and GCC.

Despite its improved features, the proposed TDE algorithm has several points that could be developed further. Some of these are familiar issues which TDE algorithms working under the same circumstances are often faced with. These issues, with possible future work to improve them are mentioned below:

1. Time of convergence – the TDE algorithm sometimes takes a relatively long time to converge to the accurate estimates of the delay. This is especially true for the exact TDE algorithm. Given that the delay to be estimated might vary with time, it is important to estimate its value before it changes. A general improvement of the algorithm should naturally diminish the time of convergence.
2. Robustness in the presence of speech – one of the reasons why the TDE algorithm does not offer 100% robustness is due to the presence of speech signals and their use for estimating the delays. It was mentioned previously in this thesis that speech signals are inherently time-varying. Possible improvements to the algorithm include the employment of a time-varying preprocessing step or the use of perceptual models of speech such that more information is available to the TDE algorithm.
3. Robustness in the presence of reverberation – reverberation is another major factor that negatively influences the performance of correlation-based TDE algorithms such as the one proposed in this thesis. While - as mentioned in Chapter 2 - solutions have been proposed to deal with reverberation, they tend to

be costly from a computational point of view. In general, accounting for the presence of reverberation implies that the TDE algorithm needs to somehow identify (at least partially) the impulse response of the unknown echo path. In this way the sample of the impulse response with the highest amplitude is selected and assumed to be the limiter of the flat delay. One possible way to improve this aspect of the TDE algorithm is to include prior information about the expected shape of the echo path impulse response.

Although improvements are possible by undertaking the research efforts suggested above, one must keep in mind that due to the presence of random processes and variables and the use of several approximated values, the TDE algorithms cannot achieve 100% robustness. Instead, one must formulate expected thresholds for performance and computational costs based on the application at hand. This information, in turn, allows the user to choose the most appropriate algorithm.

APPENDICES

APPENDIX A

Statistical Preliminaries

A.1 Random Variables and Random Processes

A probability system is composed of the set of possible outcomes of an experiment and their associated probability. For example, the experiment of tossing a coin has two outcomes, or events $\{H, T\}$, each of which has an associated probability equal to 0.5.

Random variables are functions that map the sample space – i.e. possible outcomes - of an experiment to the real axes. For the coin toss example, the possible outcomes $\{H, T\}$ can be mapped into $\{-1, 1\}$. As a result, random variables are associated with a probability distribution that can be represented graphically. For the coin toss example, the probability distribution is shown in Figure A-1.

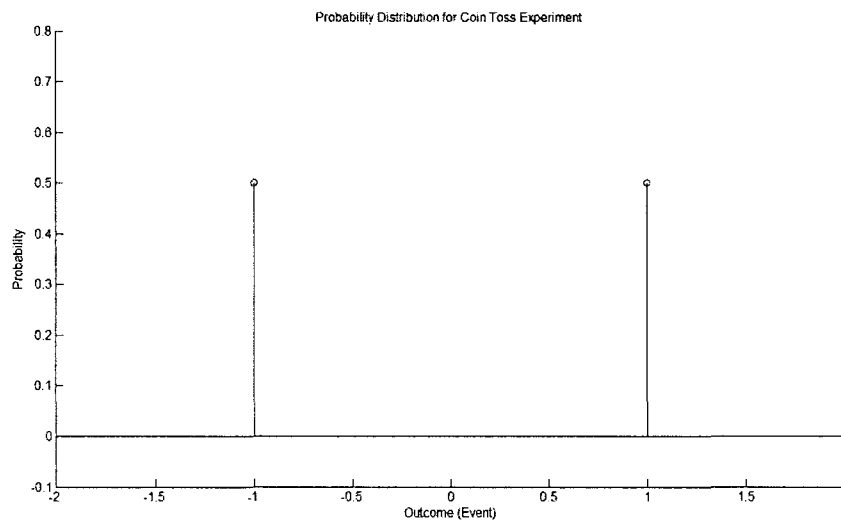


Figure A-1 Probability distribution of the coin toss experiment

The result of the coin toss experiment can only take discrete values. Random variables can also take a continuous range of values. Therefore a distinction arises between discrete and continuous random variables. Their probability distributions are distinguished as probability mass function (*pmf*) and probability density function (*pdf*) for discrete and random variables respectively. The *pmf* of the coin tossing experiment was depicted in Figure A-1, the pdf of a continuous Gaussian random variable is depicted in Figure A-2.

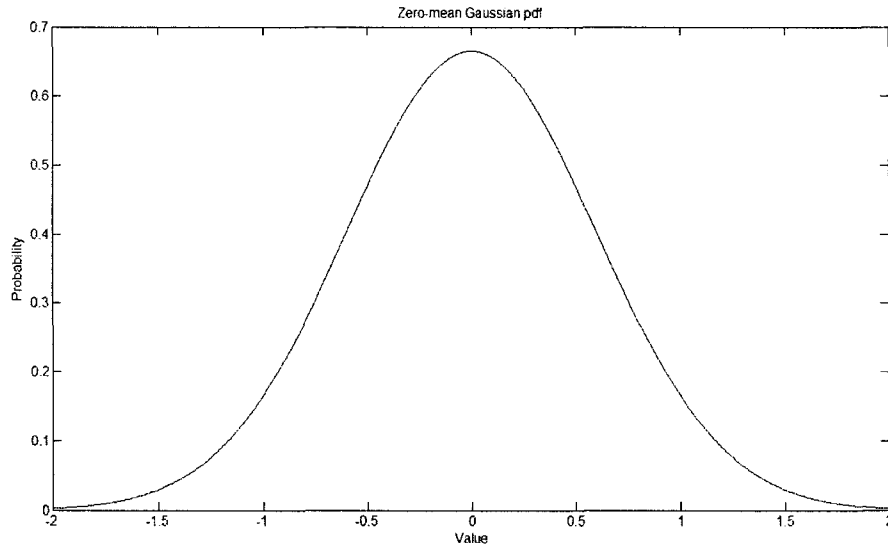


Figure A-2 Pdf of Gaussian random variable

The cumulative distribution function (CDF) of a random variable x is defined as:

$$F(x) = \sum_{i=-\infty}^{\infty} P(x_i) \quad (\text{A-1})$$

In this case x_i is the possible value of the outcome and $P(x_i)$ is the probability of that outcome. For the case of continuous random variables, the CDF is defined as:

$$F(x) = \int_{-\infty}^x f(x)dx \quad (\text{A-2})$$

where $f(x)$ is the pdf of the continuous random variable x . Multivariate probability distributions can be computed that depict the relation between multiple random variables, for example: $f(x_1, x_2, \dots, x_N)$ is the joint pdf of x_1, x_2, \dots, x_N , or $P(x_1, x_2, \dots, x_N)$ is the joint probability of x_1, x_2, \dots, x_N when they are discrete variables.

A.2 Ensemble Averages and Moments

Given a random variable x , a new random variable can be created as $y=h(x)$ where $h(.)$ is a function that maps the random variable x into the new random variable y . The ensemble average of x is defined as:

$$E(x) = \bar{x} = \int_{-\infty}^{\infty} xf(x)dx \quad (\text{A-3})$$

for continuous x , and

$$E(x) = \bar{x} = \sum_{i=-\infty}^{\infty} x_i P(x_i) \quad (\text{A-4})$$

for discrete x .

$E(x)$ is commonly referred to as the *expected value* of x , or simply as the *mean* of x . The r -th moment of x taken about the point x_0 is defined as:

$$E((x - x_0)^r) = \int_{-\infty}^{\infty} (x - x_0)^r f(x)dx \quad (\text{A-5})$$

so the mean is the first moment taken about the origin and the variance σ^2 is the second moment taken about the mean, that is:

$$E((x - \bar{x})^2) = \int_{-\infty}^{\infty} (x - \bar{x})^2 f(x) dx \quad (\text{A-6})$$

The standard deviation σ is simply the square root of the variance.

A.3 Random Processes

Random variables map events into constants, while random (or stochastic) processes map events into functions of time. Random processes can be thought of as time-indexed collections of random variables. For example, the random variable $x_n = x(t_n)$ is extracted from the random process x at time instant t_n . Their probability distribution can be represented over two axes, time and probability. Random processes may be classified into continuous and discrete, depending on the nature of the associated random variables. In signal processing, each sample in a sequence of data is the outcome of a random variable and is associated with a certain probability.

A random process is defined as being *stationary* of order N if for any t_1, t_2, \dots, t_N :

$$f(x(t_1), x(t_2), \dots, x(t_N)) = f(x(t_1+t_0), x(t_2+t_0), \dots, x(t_N+t_0)) \quad (\text{A-7})$$

that is, a random process is stationary of order N if its N -th order statistics do not change by a shift in time.

Another distinction of random processes is based on the property of ergodicity. A random process is ergodic if all time averages of any sample function are equal to the

corresponding ensemble averages. For example, a random process $x(t)$ is ergodic in the mean if:

$$E(x(t)) = \langle x(t) \rangle \quad (\text{A-8})$$

that is if:

$$\int_{-\infty}^{\infty} x f_x(x) dx = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) dt \quad (\text{A-9})$$

Ergodicity is often assumed for digital signals since only time samples are measured.

A.4 The Correlation Function

The autocorrelation function of a process $x(t)$ for a delay τ is defined as:

$$R_{xx}(t, t + \tau) = E[x(t)x(t + \tau)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_t x_{t+\tau} f_x(x_t, x_{t+\tau}) dx_t dx_{t+\tau} \quad (\text{A-10})$$

The cross-correlation function is similar to the above except it is defined for two different random processes $x(t)$ and $y(t)$.

$$R_{xy}(t, t + \tau) = E[x(t)y(t + \tau)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_t y_{t+\tau} f_{xy}(x_t, x_{t+\tau}) dx_t dy_{t+\tau} \quad (\text{A-11})$$

In the case when $x(t)$ and $y(t)$ are jointly ergodic, the cross correlation function can be expressed as a time average:

$$R_{xy}(t, t + \tau) = \langle x(t)y(t + \tau) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)y(t + \tau) dt \quad (\text{A-12})$$

The auto correlation function is computed in a similar manner. The above formulas can easily be extended to discrete signals by changing the integration to summation. The cross correlation coefficient is used when taking into account changes in the energy levels of the signals. The cross correlation coefficient can be considered as a normalized version of the cross correlation and is defined as:

$$\rho_{xy}(t, t + \tau) = \frac{R_{xy}(t, t + \tau)}{\sqrt{R_{xx}(t, t)R_{yy}(t + \tau, t + \tau)}} \quad (\text{A-13})$$

Stationarity has implications in measures of statistical averages. Specifically, the mean of a stationary process $x(t)$ is a constant, that is:

$$E[x(t)] = \int_{-\infty}^{\infty} x_{t_1} f_x(x_{t_1}) dx_{t_1} = \int_{-\infty}^{\infty} x_{t_2} f_x(x_{t_2}) dx_{t_2} = \text{const.} \quad (\text{A-14})$$

The autocorrelation of a stationary random process $x(t)$ is:

$$\begin{aligned} R_{xx}(t_1, t_1 + \tau) &= E[x(t_1)x(t_1 + \tau)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_1} x_{t_1 + \tau} f_x(x_{t_1}, x_{t_1 + \tau}) dx_{t_1} dx_{t_1 + \tau} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_2} x_{t_2 + \tau} f_x(x_{t_2}, x_{t_2 + \tau}) dx_{t_2} dx_{t_2 + \tau} = R_{xx}(\tau) \end{aligned} \quad (\text{A-15})$$

that is, the correlation function is only a function of the delay.

If conditions (A-14) and (A-15) are true for a random process $x(t)$, then this process is said to be a Wide Sense Stationary (WSS) process. Wide Sense Stationarity is a weaker

type of stationarity and is commonly used because it is relatively easy to test. A process is WSS if it is stationary in the strict sense, the converse is not necessarily true.

Two random processes $x(t)$ and $y(t)$ are *uncorrelated* if:

$$R_{xy}(\tau) = E[x(t)y(t)] = E[x(t)]E[y(t)] \quad (\text{A-16})$$

and vice-versa. On the other hand, two random processes $x(t)$ and $y(t)$ are *orthogonal* if:

$$R_{xy}(\tau) = 0 \quad (\text{A-17})$$

A.5 Power Spectral Density

To represent the frequency distribution of the power for a random process, the concept of Power Spectral Density (PSD) is introduced. Given a random process $x(t)$, its PSD is defined as:

$$\Phi_{xx}(\omega) = \lim_{T \rightarrow \infty} \left(\frac{|X_T(\omega)|^2}{T} \right) \quad (\text{A-18})$$

Where $X(\omega)$ is obtained by taking the Fourier transform of all the truncated sample functions as:

$$X_T(\omega) = \int_{-T/2}^{T/2} x(t)e^{-j\omega t} dt \quad (\text{A-19})$$

Using the Wiener-Khintchine theorem, the PSD can also be calculated by simply taking the Fourier transform of the autocorrelation of $x(t)$ with respect to the time lag:

$$\Phi_{xx}(\omega) = \int_{-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau} d\tau \quad (\text{A-20})$$

A.6 Approximating the Correlation and Power Spectral Density for Discrete Signals

In most DSP application, only a finite number of samples of a random signal are available at any time. This means that the expressions in (A-10) to (A-12) and (A-20) are impossible to compute in practice. Thus, when measuring the correlation function or the PSD, approximations are needed. Specifically, the cross correlation of two discrete random processes that generate the respective random signals $x(n)$ and $y(n)$ is approximated using a finite time window as:

$$R_{xy}(\tau) = \frac{1}{L} \sum_{l=0}^{L-1} x(n-\tau-l)y(n-l) \quad (\text{A-21})$$

where τ must be an integer.

The PSD is approximated in practice using the periodogram:

$$\Phi_{xy}(k) = \left(\frac{|X_T(k)|^2}{T} \right) \quad (\text{A-22})$$

where $X_T(k)$ is the k -th sample of the DFT of $x(n)$, and T is the length of the DFT.

REFERENCES

- [1] Ljung, Lennart. *System Identification: A Theory for the User*. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1998.
- [2] Haykin, Simon *Adaptive Filter Theory*. 4th ed. Prentice Hall, 2002.
- [3] Trevor Yensen and Rufik Goubrun, “An Acoustic Echo Cancellation System for Synthetic Surround Sound”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp.3237 – 3240, May 2001.
- [4] A. Gilloire, M. Vetterli, “Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation”, *IEEE Trans. Signal Processing*, vol. 40, pp.1862 – 1875, Aug. 1992.
- [5] M. Sondhi, D. Morgan and J. Hall, “Stereophonic acoustic echo cancellation - An overview of the fundamental problem,” *IEEE Signal Processing Lett.*, Vol. 2, No. 8, pp. 148-151, Aug. 1995.
- [6] J. Benesty, D. Morgan, J. Hall and M. Sondhi, “Synthesized stereo combined with acoustic echo cancellation for desktop conferencing”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 853-856.
- [7] M. Tahernezehadi, J. Liu, G. Miller, X. Kong, “Acoustic echo cancellation using subband technique for teleconferencing applications”, *IEEE Global Telecommunications Conference*, vol.1, pp. 243 – 247, Nov. 28-Dec. 2, 1994.
- [8] P. Eneroth, S.L. Gay, T. Gansler, J. Benesty, “A real-time implementation of a stereophonic acoustic echo canceler”, *IEEE Transactions on Speech and Audio Processing*, vol. 9, Issue 5, pp. 513 – 523, July 2001.

- [9] T.N. Yensen, R.A. Goubran, I. Lambadaris, “Synthetic stereo acoustic echo cancellation structure for multiple participant VoIP conferences”, *IEEE Transactions on Speech and Audio Processing*, vol. 9, Issue 2, pp. 168 – 174, Feb. 2001.
- [10] Y. Lu, R. Fowler, W. Tian, and L. Thompson, “Enhancing Echo Cancellation via Estimation of Delay”, *IEEE Trans. Signal Processing*, vol. 53, no. 11, Nov. 2005.
- [11] D. L. Duttweiler, “Subsampling to estimate delay with application to echo canceling,” *IEEE Trans. Signal Processing*, vol. 31, no. 10, pp. 1090–1099, Oct. 1999.
- [12] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*, Springer-Verlag, Berlin, Germany, 2006.
- [13] K. Eneman, “Subband and Frequency Domain Adaptive Filtering Techniques for Speech Enhancement in Hands-Free Communication”, Masters Thesis, Katholieke Universiteit Leuven, March 2002.
- [14] J. Benesty and Y. Huang, *Adaptive Signal Processing - Applications to Real-World Problems*. Springer-Verlag, Berlin, Germany, 2003.
- [15] M. Branstein, H. Silverman, “A Practical methodology for speech source localization with microphone arrays”, *Comput., Speech Language*, vol. 2, pp. 91-126, Nov. 1997.
- [16] Y. Huang, J. Benesty, G. W. Elko, “Adaptive eigenvalue decomposition algorithm for realtime acoustic source localization system”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 937-940, 1998.
- [17] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice-Hall, 4th ed., 2007.

- [18] C. Breining, P. Dreiscitel, E. Hansler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, J. Tilp, "Acoustic echo control, an application of very-high-order adaptive filters", *IEEE Signal Process. Magazine*, vol. 16, issue 4, pp. 42–69, July 1999.
- [19] G. C. Carter, *Coherence and Time Delay Estimations*. New York: IEEE, 1993.
- [20] M. Azaria, D. Hertz, "Time delay estimation by generalized cross correlation methods", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 32, issue 2, pp. 280-285, Apr. 1984.
- [21] A. Stephenne, B. Champagne, "Cepstral prefiltering for time delay estimation in reverberant environments", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3055-3058, May 1995.
- [22] R. Parisi, R. Gazzetta, E. Di Claudio, "Prefiltering approaches for time delay estimation in reverberant environments", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2997-3000, May 13-17, 2002.
- [23] S. Doclo, M. Moonen, "Robust adaptive time delay estimation for speaker localization in noisy and reverberant acoustic environments", *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 11, pp. 1110-1124, 2003.
- [24] Q. G. Liu, B. Champagne, P. Kabal, "A microphone array processing technique for speech enhancement in reverberant space", *Speech Communication*, vol. 18, pp. 317-334, 1996.
- [25] S. Subramaniam, A. P. Petropulu, C. Wendt, "Cepstrum-based deconvolution for speech dereverberation", *IEEE Trans. On Speech Audio Processing*, vol. 4, no. 5, pp. 392-396, 1996.

- [26] A. St  phenne and B. Champagne, "A new cepstral prefiltering technique for estimating time delay under reverberant conditions," *Signal Processing*, vol. 59, no. 3, pp. 253-266, 1997.
- [27] D. Childers, et.al, "The Cepstrum: A Guide to Processing", *Proc. of the IEEE*, vol. 65, pp. 1428- 1443, 1977.
- [28] A. P. Oppenheim, R. W. Schaffer, *Discrete-Time Signal Processing*, Second Edition, Prentice-Hall, New Jersey, 1999.
- [29] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 384–391, 2000.
- [30] B. Champagne, S. B  dard, and A. St  phenne, "Performance of time-delay estimation in the presence of room reverberation", *IEEE Trans. Speech, and Audio Processing*, vol. 4, no. 2, pp. 148–152, 1996.
- [31] J. Chen, J. Benesty, Y. Huang, "Time delay estimation in room acoustic environments: an overview", *EURASIP Journal on Applied Signal Processing*. vol. 2006, no. 12. Pp. 19-39, 2006.
- [32] O.L. Frost III, "An algorithm for linearly constrained adaptive array processing", *IEEE Proceedings*, Vol. 60, pp. 926-935, August 1972.
- [33] F. A. Reed, P. L. Feintuch, N. J. Bershad, "Time delay estimation using the LMS adaptive filter static behaviour", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 3, pp 561-571, June 1981.
- [34] D. H. Youn, N. Ahmed, G. C. Carter, "On using the LMS algorithm for time delay estimation", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, no. 5, pp 798-801, Oct. 1982.

- [35] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*, Springer-Verlag, Berlin, Germany, 2001.
- [36] S. L. Gay, J. Benesty, *Acoustic signal processing for telecommunication*, Kluwer Academic Publishers, Hingham, MA, 2000.
- [37] A. Gilloire, E. Moulines, D. Slock, and P. Duhamel, *State of the Art in Acoustic Echo Cancellation*. Boston, MA: Kluwer, 1995.
- [38] B. Widrow, S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, N.J. 1985.
- [39] S. Kawamura, M. Hatori, “A tap selection algorithm for adaptive FIR filters with sparse taps”, IEICE Trans. Fundamentals, vol. E77-A, no. 4, pp. 681-687, Apr. 1994.
- [40] A. Sugiyama, S. Ikeda, A. Hirano, “A fast convergence algorithm for sparse-tap adaptive FIR filters identifying an unknown number of dispersive regions”, *IEEE Trans. Signal Processing*, vol. 50, no. 12, pp 3008-3017, Dec. 2002.
- [41] P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [42] S. Mitra, *Digital Signal Processing: A computer based approach*, 2nd ed. McGraw-Hill, 2002.
- [43] “Audacity – Free audio editor and recorder”. Available online at <http://audacity.sourceforge.net/>
- [44] “Matlab Online Documentation”, 1994-2008 The MathWorks, Inc.
<http://www.mathworks.com/access/helpdesk/help/helpdesk.html>
- [45] ITU –T G.168 standard

- [46] H. C. So, P. C. Ching, and Y. T. Chan, “A new algorithm for explicit adaptation of time delay”. *IEEE Transactions on Signal Processing*, vol. 42, no. 7, pp. 1816–1820, 1994.
- [47] W. Kellermann. “Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes”, *IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*, vol. 5, pp. 2570-2573, Apr. 1988.
- [48] A. Gilloire, M. Vetterli, “Adaptive Filtering in Subbands with Critical Sampling : Analysis, Experiments and Application to Acoustic Echo Cancellation”, *IEEE Trans. Signal Processing*, vol. 40, pp. 1862-1875, Aug. 1992.
- [49] M. Harteneck, S. Weiss, R. W. Stewart, “Design of near perfect reconstruction oversampled filter banks for subband adaptive filters”, *IEEE Trans. Circuits Systems II: Analog and Digital Signal Processing*, vol. 46, no. 8, pp. 1081-1085, 1999.
- [50] S. Weiss, A. Stenger, R. Stewart, R. Rabenstein, “Steady-state performance limitations of subband adaptive filters”, *IEEE Trans. Signal Processing*, vol. 49, pp. 1982-1991, Sept. 2001.
- [51] R. E. Crochiere, L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, 1983.
- [52] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. New York, Wiley, 2004.
- [53] S. Weiss, L. Lampe, R. Stewart, “Efficient implementations of complex and real-valued filter banks for comparative subband processing with an application to adaptive filtering”, *Proceedings of the First International Symposium on Communication Systems and Digital Signal Processing*, vol. 1, pp. 32-35, 1998.
- [54] A. Sayed, *Fundamentals of Adaptive Filtering*. Wiley, 2003.

VITA AUCTORIS

NAME

Tomi Hila

DATE OF BIRTH

May, 18 1982

PLACE OF BIRTH

Shkoder, Albania

EDUCATION

M. A. Sc.

Department of Electrical and Computer Engineering

University of Windsor,

Windsor, ON, Canada

2007-2009

B. App. Sc.

Department of Electrical and Computer Engineering

University of Windsor,

Windsor, ON, Canada

2002-2006