INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Transform Domain Echo Cancellation and Noise Reduction

by Jeffrey P. Lariviere, M. Eng.

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

Ottawa-Carleton Institute of Electrical and Computer Engineering

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario

May 8, 2000

© copyright 2000,

Jeffrey Lariviere

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-57614-0

Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

“Transform Domain Echo Cancellation and Noise Reduction”

submitted by Jeffrey Lariviere, B.Eng./M.Eng.
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Chair

Department of Systems and Computer Engineering

Thesis Supervisor

External Examiner

Carleton University

May 8, 2000

Abstract

This thesis examines aspects of transform-domain echo cancellation (EC) and noise reduction (NR). Two particular transform domains are examined for EC and NR - the frequency domain and the linear predictive coding (LPC) domain.

Within the frequency domain an existing algorithm, the Generalized Multi-Delay Frequency Domain (GMDF) adaptive filter is enhanced in three key areas. That is, the throughput delay of the algorithm due to block processing is rendered adjustable (down to no delay), the performance in terms of rate of convergence and depth of convergence is improved through control of the algorithm's reconstruction filters, (mean echo return loss enhancement (ERLE) > 2.5 dB) and a noise reduction component with improved performance compared to existing noise reduction methods is integrated into the algorithm with very little additional computational expense. (4% extra computation allows for between 6dB and 9dB increase in ERLE compared with using GMDF without a noise reduction component.) The GMDF is compared and contrasted with the Normalized Least-Mean-Square (NLMS) algorithm and the Fast Affine Projection (FAP) algorithm in terms of performance and computational complexity.

In addition, the effect of the GMDF noise-reduction component on the error rate of an automatic speech recognition (ASR) system in the presence of additive car noise is examined. Results show a decrease in word classification errors from 15.3 % to 7.1%, using a state-of-the-art commercial speech recognition engine in the presence of car noise at 0dB SNR, when preprocessed with the noise reduction method. Lesser, but still significant, improvements are obtained from 5dB to 15dB SNR.

The LPC domain is chosen for examination due to its prevalence in existing

mobile coding applications. The relationships between LPC parameters before and after a loudspeaker-room-microphone transfer function are derived for stationary and non-stationary far-end speech. For the non-stationary case a computationally simplified structure is introduced and analysis and simulations are performed showing considerable improvement in convergence depth and rate over the traditional NLMS approach ($> 14\text{dB}$ in system distance, $> 8\text{dB}$ in ERLE), where system distance is a normalized measure of impulse response coefficient error, and is useful for evaluating adaptation algorithms in simulation experiments with known impulse responses.

Acknowledgments

Foremost, I would like to thank my supervisor, Dr. R. Goubran for his insight, guidance and assistance during this work. I would also like to thank the following organizations for the financial assistance during the course of my studies: Carleton University, the Natural Sciences and Engineering Research Council of Canada (NSERC), Communications and Information Technology Ontario (CITO), Nortel Networks Corporation and the Government of Ontario's Graduate Scholarship (OGS) program.

I would also like to thank my fellow lab residents for their useful suggestions and technical assistance on numerous occasions. In particular, Trevor Yensen, Gilad Helman, and Danny Lemay deserve special mention.

Finally, I would like to give special thanks to my daughter Chelsea and my wife Christine for their patience and support during this endeavour.

Table of Contents

Abstract	iii
Acknowledgments	v
List of Figures	viii
List of Tables	x
List of Variables	xi
List of Abbreviations	xvi
Introduction	1
Problem Statement	3
Contributions	6
Generalized Multi-Delay Frequency Domain Adaptive Filter (GMDF)	7
Description of Algorithm	8
Generation of Filter Output Samples	8
Updating of Filter Coefficients	17
Asymptotic Computational Complexity	21
Low-delay and Delayless GMDF	23
GMDF with Adaptive Reconstruction Filters	31
Frequency Response and Adaptation over Time of ARF Coefficients	43
Noise Reduced GMDF	49
Generation of Filter Output Samples	49
Generation of NR Filter Output Samples	49
Updating of NR Filter Weights	51
Convergence of the Noise Reduction Filter	54
Performance Evaluation	55
Linear Predictive Coding Based Acoustic Echo Cancellation	60
General & Simplified Structure for LPC-AEC (without long-term predictor) ..	61
Case 1. Stationary Far-end Speech, General Structure	63
Case 2. Non-stationary Far-end Speech, General Structure	66
Simplified Structure for LPC-AEC (with long-term predictor)	73
Case 1. Estimate of $\hat{X}_e(n)$ for Stationary Signal	75

Case 2. Estimate of $\hat{X}_e(n)$ for Non-stationary Signal	77
Derivation of Adaptation Equations	77
Computer Simulations	82
Conclusions	89
Future Work	92
References	93
Appendix A: Recording room characteristics	97

List of Figures

Figure 1	Acoustic echo, noise and near-end speech return to the far-end listener . .	2
Figure 2	Depiction of FIR filter convolution.	9
Figure 3	Subdivision of filter into 3 smaller filters gives equivalent results.	10
Figure 4	Frequency domain “overlap-save” version of subdivided filter.	11
Figure 5	Example of output sample generation for $R=4$, $\alpha=1,2$ and 4.	16
Figure 6	Average ERLE vs. throughput delay for NLMS, FAP (10 th order) and GMDF with different block sizes (indicated by R).	24
Figure 7	Schematic representation of the relationship in time between the error vector, e_s , and the output sample y_m , with $R=4$ and $\alpha=R$	27
Figure 8	ERLE for different throughput delay amounts using $R=16$, $\alpha=8$ for filtered white noise and a simulated conference room impulse response of 640 taps.	28
Figure 9	ERLE for delayless GMDF and NLMS	30
Figure 10	Error variance vs. intra-block index for unconstrained GMDF.	32
Figure 11	Error variance vs. intra-block index for eight speakers (Males shown black, females shown red.)	34
Figure 12	Example of the relationship between $\hat{y}_s$, $\hat{y}_{s;i}$ and $v_j^{(s)}$ for the case of $R=8$, $\alpha=4$	37
Figure 13	ERLE for 2.3 seconds of synthetic speech recorded in a real conference room for a delayless GMDF system using a moving average reconstruction filter and an adaptive reconstruction filter.	40
Figure 14	ERLE for 2.3 seconds of synthetic speech recorded in a real conference room for moving average, SARF and MARF.	42
Figure 15	Average ERLE vs. block size for multiple adaptive reconstruction filters (red) and moving average reconstruction filters (black) for $\alpha = \text{block size}$,	

Figure 16	Adaptation over time of ARF coefficients.	46
Figure 17	Adaptation over time of the magnitude of the normalized frequency response of an ARF.	47
Figure 18	Normalized histogram of one second of the adaptive reconstruction filter's frequency response (shown in red) and the input signal's frequency components (shown in black).	48
Figure 19	Schematic of Noise-Reduced GMDF.	52
Figure 20	Details of NR filter for N spectral components	53
Figure 21	Mean ERLE vs SNR, with NR complex-valued tap weights (solid lines), with NR real-valued tap weights (dotted lines) and without NR (dashed lines) for car noise (circles), office noise (diamonds), street noise (stars) and air-conditioning noise (squares).	57
Figure 22	Error percent vs. SNR for utterances with and without NR-GMDF processing	59
Figure 23	General structure for LPC-AEC.	62
Figure 24	Simplified structure for LPC-AEC.	64
Figure 25	Average normalized distortion vs. impulse response truncation point for 3 impulse responses, truncated at 100, 256, 512 and 640 taps.	72
Figure 26	Synthesis filters used to estimate the echo.	74
Figure 27	Equivalent filter structures.	76
Figure 28	Condition number estimate vs time for LPC-AEC system and NLMS system (traditional AEC).	81
Figure 29	Proposed structure for LPC-AEC.	83
Figure 30	ERLE vs time for NLMS (traditional AEC) and LPC-AEC.	84
Figure 31	System Distance vs time for NLMS and LPC-AEC.	85
Figure 32	System Distance vs time for NLMS, Xf based LPC-AEC and LPC-AEC with true impulse response of 64 taps.	87
Figure 33	System Distance vs time for NLMS, Xf based LPC-AEC and LPC-AEC with true impulse response of 256 taps.	88

List of Tables

Table 1	Real mults and adds for NLMS, FAP and GMDF calculated per output sample.	21
----------------	---	----

List of Variables

(For chapter 2)

y_n	scalar output value at time n
$\mathbf{w}$	vector of FIR tap coefficients
$\mathbf{x}_n$	vector of input values at time n
N	length of FIR filter
K	number of subfilters
R	length of each subfilter
$\mathbf{w}_k$	vector of k th subfilter's coefficients
M	size of DFT
$\mathbf{x}_k^{(M)}$	vector of input values used by the k th subfilter, extended by zeros to a size of M
$\mathbf{X}_k$	frequency domain version of $\mathbf{x}_k^{(M)}$
$\mathbf{F}_M$	discrete Fourier transform operator of size M by M
$\mathbf{F}_M^{-1}$	inverse discrete Fourier transform operator of size M by M
$\mathbf{w}_k^{(M)}$	vector of k th subfilter's coefficients, zero-extended to size M
$\mathbf{W}_k$	frequency domain version of $\mathbf{w}_k^{(M)}$
$\mathbf{0}_{n;k}$	zero-filled matrix of size n by k
$\tilde{\mathbf{y}}_k$	circularly convolved output vector for the k th subfilter
$\hat{\mathbf{y}}_k$	linearly convolved output vector for the k th subfilter
$\hat{\mathbf{y}}$	linearly convolved output vector for all subfilters combined through summation
α	GMDF parameter controlling the amount of overlap at each iteration

P	the number of new samples seen at each iteration
$\mathbf{g}$	the reconstruction filter's vector of coefficient values
$\hat{y}_{m;n}$	the n th element of the output vector of the m th iteration
$\hat{\mathbf{y}}_s$	the final output vector for the s th iteration
$\hat{\mathbf{v}}_s$	estimate of the gradient vector at iteration s
$\hat{v}_{s;j}$	j th element of the gradient vector estimate at iteration s
$\mathbf{\chi}_s$	diagonal matrix of blocked input values at iteration s
$\mathbf{e}_s$	error vector at iteration s
$\mathbf{E}_s$	frequency domain version of the error vector at iteration s
$\otimes$	element by element multiplication operator for vectors or matrices
$\mathbf{C}$	GMDF "constraint" operator (converts from circular to linear convolution)
$\mathbf{T}_s$	vector of inverse power levels at iteration s
D	amount of delay
f_s	sampling frequency
$\mathbf{v}_j^{(s)}$	vector of estimates corresponding to the same time location at iteration s
$\mathbf{V}^{(s)}$	matrix of $\mathbf{v}_j^{(s)}$, j from 1 to P , at iteration s
μ_R	step-size parameter for the adaptive reconstruction filter
e_n	reconstruction error
$\mathbf{G}^{(s)}$	matrix of reconstruction vectors, $\mathbf{g}_j^{(s)}$, j from 1 to P , at iteration s
$\hat{e}$	echo estimate
$\hat{\mathbf{e}}_s$	estimate of a vector of echo values, at iteration s
z	system output
m	microphone signal
z_{nr}	noise reduced system output

m_{nr}	noise reduced microphone signal
$M_{nr}(k)$	k th frequency bin spectral component of the noise-reduced microphone signal
$M(k)$	k th frequency bin spectral component of the microphone signal
$S(k)$	k th frequency bin spectral component of the noise signal
$NR(k)$	k th frequency bin spectral component of the noise reduction signal
$\hat{E}(k)$	k th frequency bin spectral component of the echo estimate
$AECerr(k)$	k th frequency bin spectral component of the acoustic echo cancellation error
$NRerr(k)$	k th frequency bin spectral component of the noise reduction error
$J(k)$	mean-square-error cost function for the k th frequency bin
$NRtaps(k)$	noise reduction filter tap weight for the k th frequency bin
$NR\mu$	step size for the noise reduction filter

(For chapter 3)

$e(n)$	echo signal
$\hat{e}(n)$	estimate of the echo signal
$s(n)$	near-end speech signal
$\hat{s}(n)$	estimate of the near-end speech signal
$f(n)$	far-end speech signal
$\mathbf{X}_f$	far-end excitation vector
$\mathbf{w}_f$	far-end linear predictive coefficients
$\mathbf{X}_e$	echo excitation vector

w_e	echo linear predictive coefficients
$\hat{X}_e$	estimate of the echo excitation vector
$\hat{w}_e$	estimate of the echo linear predictive coefficients
$\hat{r}_e$	estimate of the echo autocorrelation vector
h	loudspeaker-room-microphone impulse response vector
L	size of loudspeaker-room-microphone impulse response
M	“order” of the linear predictive model, i.e. number of LP coefficients
P	block size used to compute the LP coefficients
α_f	delay of the long-term predictor filter, far-end signal model
α_e	delay of the long-term predictor filter, echo signal model
G_f	gain of the long-term predictor filter, far-end signal model
G_e	gain of the long-term predictor filter, echo signal model
$P(z)$	inverse of the long-term synthesis filter, Z-domain
$A(z)$	inverse of the short-term synthesis filter, Z-domain
$\hat{X}_e(z)$	estimate of the echo excitation sequence, Z-domain
$\hat{X}_{es}(z)$	estimate of the short-term excitation sequence, Z-domain
$\hat{X}_{el}(z)$	estimate of the long-term excitation sequence, Z-domain
$err(n)$	error between the echo and the estimate of the echo
$J(n)$	mean-square-error cost function
$\nabla J(n)$	gradient of the cost function
$der(n, k)$	partial derivative of the estimate of the echo excitation with respect to the impulse response taps

$ERLE(n)$ echo return loss enhancement as a function of the time index, n

$D(n)$ system distance as a function of the time index, n

List of Abbreviations

AEC	Acoustic echo cancellation
ASR	Automatic speech recognition
DFT	Discrete Fourier transform
EC	Echo cancellation
ERLE	Echo-return loss enhancement
FAP	Fast affine projection
GMDF	Generalized MDF
HMM	Hidden Markov model
LMS	Least mean square
LPC	Linear predictive coding
MARF	Multiple adaptive reconstruction filters
MDF	Multi-delay frequency domain adaptive filter
LRM	Loudspeaker-room-microphone
NLMS	Normalized LMS
NR	Noise reduction
SARF	Single adaptive reconstruction filter
SD	System distance

Chapter 1

Introduction

Hands-free telephony continues to become an increasingly desirable feature of modern telecommunications, whether in a conference room or mobile setting. Fundamental to the user acceptability of these systems is the performance of algorithms for acoustic echo cancellation and noise reduction. Acoustic echo cancellation (AEC) allows hands-free talkers to communicate in a natural manner, with far-end speech and near-end speech partially (or wholly) overlapping in time, i.e. in a full-duplex mode. Without the use of an AEC, the far-end speech, broadcast through the speaker in a telephone set enclosure, will be picked up by the set's microphone and transmitted back to the far-end talker. (See figure 1). Perceptually,

Figure 1 Acoustic echo, noise and near-end speech return to the far-end listener.

this is annoying to the users and if extreme will contribute to the result of loudspeaker “howling”.

Initial hands-free terminals achieved echo-free conversations by introducing manual or automatic switched-loss functions in the transmit and receive paths [39]. This came at the price of naturalness, i.e. talkers were forced to communicate in a half-duplex mode, where near-end and far-end speech can only occur non-overlapping in time.

In addition to the problem of acoustic echo, most environments contribute undesirable non-stationary noise. In a conference room setting, these noise sources may be computer fan noise, air-conditioning noise or general office noise (other telephone conversations, office equipment, etc.). In a mobile setting, the noise sources may include engine noise, fan or air-conditioning noise, tire noise, etc. At higher speeds wind noise is usually predominant and without a noise-reduction algorithm component, may reduce the signal to noise ratio to the point of far-end listener incomprehension [15]. In less extreme conditions the noise power may still be sufficient to cause listener fatigue - once again contributing to a decline in user acceptance.

Problem Statement

Throughout the history of AEC implementations, the least-mean-square algorithm (LMS) or Normalized LMS algorithm (NLMS) [42] has often been the method of choice, due to its simplicity and low computational requirements. In recent years, as available processing power has increased, algorithms which offer better performance, albeit at a higher computational cost, have become desirable. For a number of reasons, one such algorithm which has received considerable interest recently is the Generalized Multi-Delay Frequency (GMDF) domain adaptive filter [20,21,25,32,33] (described in Chapter 2). Although the GMDF has been found to perform significantly better than the NLMS algorithm, especially with long impulse responses and speech as input, it still has some drawbacks. Since the algorithm operates in the frequency domain, some block processing is necessary before filtering can take place. This block processing introduces throughput delay, which is undesirable, especially in situations where the communication link is already introducing delay. All other factors remaining constant, an increase in echo time-delay will increase the human perception of the echo [18, 19]. Hence, algorithms which increase throughput delay must achieve higher levels of echo cancellation in order to

obtain an equivalent amount of perceived echo cancellation.

Reducing the throughput delay of the GMDF algorithm is one of the problems addressed by this thesis.

A second drawback to the GMDF is the number of discrete Fourier transforms (DFTs) that must be performed during each iteration of the filter. Some of these DFTs are needed for correcting aliasing problems during the filter update stage, (specifically, during the calculation of the correlation between the filter error and the filter input). Two versions of the algorithm have therefore emerged, the *constrained* GMDF, where the aliasing problem is corrected and the *unconstrained* GMDF, where the aliasing problem is not corrected, but the number of DFTs is reduced. Most practical implementations of the algorithm use the unconstrained version, obtaining a reduced computational load, but at the expense of performance loss associated with the uncorrected aliasing problem.

Reducing this performance loss with only minor (if any) computational increases is the second major problem addressed by this thesis.

In recognition of the need for noise reduction in hands-free telephony, work has been done recently in the area of combining AEC and noise reduction algorithms in an attempt to reduce overall computations and improve system performance. Structures have been examined where a noise reduction filter has been placed before the AEC filter, after the AEC structure, or both [5,13,14,24]. Optimal filtering has been examined, as well as psycho-acoustic based noise reduction filtering [2,3,7,4,11]. The third major problem addressed by this thesis is the addition of a noise-reduction component within the GMDF structure which offers good performance but requires minimal additional computations.

In the area of mobile communications, most speech coding methods are based on some variant of Linear Predictive Coding (LPC), the main purpose of which is to reduce the amount of bits sent across a channel [9,10,16,17,26,30,34,38]. Operating an AEC on the LPC parameters at the receiver, before the decoding stage, offers some important advantages, described in Chapter 3, but summarized here as:

- The coding process produces a noise-like “innovation sequence” which, if used as an input to an NLMS algorithm, will speed up the convergence rate.
- The AEC and the LPC can share some of the computation processing load, since the domain transformation (from time to LPC parameters) is now part of the coding stage. (This is compared to frequency domain filtering where a separate domain transformation stage is needed.)
- An echo code book may be used to store the necessary innovation sequence for the echo cancellation process, reducing the adaptive filtering process to a simple table lookup procedure.

Previous researchers have recognized some of these advantages [28;41] but have not fully explained the relationship between LPC parameters before and after a room transfer function. This is the fourth major problem addressed by this thesis.

An outline of the remainder of this thesis is as follows: The GMDF algorithm is reviewed in Chapter 2. Low-delay and delay-less versions of the algorithm are introduced and simulations in a variety of environments are reported. Adaptive reconstruction filters are introduced and simulation results are reported. A noise-reduced GMDF algorithm is introduced and simulation results are reported. The noise-reduction method is tested as a front-end to a commercial automatic speech recognition (ASR) engine. In Chapter 3 the LPC-based AEC is introduced and relationships between LPC parameters before and after a room’s transfer function are

derived. A simplified structure is introduced and simulation results are reported. Reduced computation methods are examined and simulations comparing these results to earlier work are reported. Chapter 4 summarizes the work and discusses the appropriateness of GMDF and LPC echo cancellers for various system operating specifications.

Contributions

- A method of obtaining adjustable throughput delay, down to zero delay, was developed for the GMDF adaptive filter algorithm.
- The depth and rate of convergence of the GMDF algorithm was improved through the idea of allowing the reconstruction filters to become adaptive.
- A noise reduction algorithm was developed and integrated into the GMDF algorithm, allowing improved ERLE with only minor additional computations.
- Exact relationships between LPC parameters before and after a loudspeaker-room-microphone transfer function were derived for stationary and non-stationary signals.
- A computationally simplified LPC-AEC structure was developed and shown to have considerable improvement in depth and rate of convergence over traditional methods.

Chapter 2

Generalized Multi-Delay Frequency Domain Adaptive Filter (GMDF)

The generalized multi-delay frequency domain adaptive filter (GMDF) has received considerable attention in echo cancellation recently due to several advantages of the algorithm, namely, quick convergence, high levels of cancellation and low delay [20,21,25,32,33]. Systems with long impulse responses, such as loudspeaker-room-microphone (LRM) transfer functions, are particularly well suited to GMDF, since the algorithm subdivides the estimated system impulse response into several smaller impulse responses (resulting in low delay). Also, by performing the filter coefficient adaptation in the frequency domain (and using normalized step sizes), quick convergence is achieved regardless of the condition number (i.e., eigenvalue spread) of the correlation matrix of the input data [27,31]. The algorithm is therefore suitable for speech signals,

where large condition numbers are typical.

This chapter introduces the GMDF algorithm, and describes three extensions to the algorithm proposed by the author; delayless GMDF, GMDF with adaptive reconstruction filters and noise-reduced GMDF. In addition, it compares the performance of the GMDF algorithm with that of the Affine Projection Algorithm and the NLMS algorithm.

Description of Algorithm

Generation of Filter Output Samples

The generalized multi-delay frequency domain filter is an extension to the multi-delay frequency domain filter (MDF), first proposed by Soo and Pang [32] and independently by Xu and Grenier [37] and Sommen [33]. The “multi-delay” aspect of the algorithm refers to the subdividing of a filter into multiple delayed subfilters. For example, the linear convolution of a filter with 12 taps is depicted in figure 2. At each iteration of the filter, one output value is produced (in our example, y_n) consisting of the inner product of the filter weight vector and the current input vector, i.e.

$$y_n = \mathbf{w}^T \mathbf{x}_n \quad (1)$$

where superscript T indicates transpose. Here the filter coefficients are defined as

$$\mathbf{w} = [w_0 \ w_1 \ \dots \ w_{N-1}]^T \quad (2)$$

and the current input block is

$$\mathbf{x}_n = [x_n \ x_{n-1} \ \dots \ x_{n-N+1}]^T \quad (3)$$

Figure 2 Depiction of FIR filter convolution.

Figure 3 shows that an equivalent output will be obtained by subdividing w into a number of smaller filters, each of which operates on a delayed (and smaller) input block.

Figure 3 Subdivision of filter into 3 smaller filters gives equivalent results.

The “multi-delayed” version of the output is therefore

$$y_n = \sum_{k=0}^{K-1} \mathbf{w}_k^T \mathbf{x}_{n-kR} \quad (4)$$

where the original filter of size N is divided into K subfilters of size R , where

$$N = KR \quad (5)$$

and

$$\mathbf{w}_k = [w_{kR} \ w_{kR+1} \ \dots \ w_{(k+1)R-1}]^T \quad (6)$$

and the input vector is redefined as

$$\mathbf{x}_i = [x_i \ x_{i-1} \ \dots \ x_{i-R+1}]^T \quad (7)$$

In the example shown in figure 3, $N=12$, $K=3$, $R=4$.

A frequency domain version of this filter can be realized with the overlap-save or overlap-add algorithm [36], in which each subfilter and its associated input block is transformed to the frequency domain, multiplied together, then inverse transformed. Applying this to the example of figure 3, we can express the operation schematically as in figure 4. Each input block is extended to a length of M , where $M=2R=8$ in order to avoid circular aliasing.

Figure 4 Frequency domain “overlap-save” version of subdivided filter. Each iteration of the filter would move the blocks R spaces to the right and produces R output values. Here, $R=4$.

Here, the input blocks are transformed to the frequency domain through the relation

$$X_k = F_M x_k^{(M)}, \quad 0 \leq k \leq K-1 \quad (8)$$

where F_M is the Discrete Fourier Transform operator of size M by M , with coefficients defined as

$$[F_M]_{n,l} = \left\{ e^{-j \frac{2\pi}{M} nl} \right\}, \quad 0 \leq n, l \leq M-1 \quad (9)$$

and $x_k^{(M)}$ is the M -extended input block, defined as

$$x_k^{(M)} = \begin{bmatrix} x_{n-(k+1)R+1} \\ x_{n-(k+1)R+2} \\ \vdots \\ x_{n-(k+1)R+M-1} \\ 0 \end{bmatrix}, \quad 0 \leq k \leq K-1 \quad (10)$$

Similarly, filter coefficients are transformed to the frequency domain via

$$W_k = F_M w_k^{(M)}, \quad 0 \leq k \leq K-1 \quad (11)$$

where $w_k^{(M)}$ is the zero-extended block of filter coefficients defined as

$$w_k^{(M)} = \begin{bmatrix} w_{kR} \\ w_{kR+1} \\ \vdots \\ w_{(k+1)R-1} \\ \mathbf{0}_{M-R,1} \end{bmatrix}, \quad 0 \leq k \leq K-1 \quad (12)$$

where $\mathbf{0}_{n,k}$ is a zero filled matrix of size n by k .

Representing the inverse Fourier transform operator as F_M^{-1} and the element by element multiplication of matrices or vectors by $\otimes$, the transformed input and weight vectors produces the circularly convolved output vector

$$\tilde{y}_k = F_M^{-1}(X_k \otimes W_k), \quad 0 \leq k \leq K-1 \quad (13)$$

which must be stripped of the first $R-1$ elements and last element to achieve linear convolution, i.e.

$$\hat{y}_k = \begin{bmatrix} \mathbf{0}_{R-1,R-1} & \mathbf{0}_{R-1,R} & \mathbf{0}_{2R-1,1} \\ \mathbf{0}_{R,R-1} & I_R & \\ \mathbf{0}_{1,2R-1} & & 0 \end{bmatrix} \tilde{y}_k, \quad 0 \leq k \leq K-1 \quad (14)$$

The final output is then a sum of the K blocks, i.e.

$$\hat{y} = \sum_{k=0}^{K-1} \hat{y}_k \quad (15)$$

Since the inverse transform is a linear operator, it can be performed after the summation, reducing the number of DFTs required by $K-1$, resulting in

$$\hat{y} = \begin{bmatrix} \mathbf{0}_{R-1,R-1} & \mathbf{0}_{R-1,R} & \mathbf{0}_{2R-1,1} \\ \mathbf{0}_{R,R-1} & I_R & \\ \mathbf{0}_{1,2R-1} & & 0 \end{bmatrix} F_M^{-1} \left(\sum_{k=0}^{K-1} X_k \otimes W_k \right) \quad (16)$$

As indicated in Figure 4 and suggested by equation 16, $\hat{y}$ is effectively truncated to R elements. Each iteration of the filter will therefore shift the input and output blocks by $P=R$ samples. Note that this results in computational savings, since the DFT of the

input blocks can be re-used in the next iteration of the filter, i.e. X_k at iteration s equals X_{k+1} at iteration $s+1$, $0 \leq k \leq K-2$.

The *generalized* multi-delay filter removes the restriction of shifting R samples at each iteration. The amount of shifting, P , is controlled by the parameter α , where

$$P = \frac{R}{\alpha}, \quad P, \alpha \in \mathbb{I} \quad (17)$$

A number of points should be mentioned; firstly, $\alpha = 1$ corresponds to regular multi-delay filtering and results in the least computations per output sample. $\alpha > 1$ implies that more than one estimate of each output sample will be produced and hence there is a need for a *reconstruction filter* to combine the estimates. Also, at each iteration, the immediately previous iteration's DFT transformed input blocks cannot be used, rather, transformed input blocks from α iterations previous must be used. This means that the computational savings from the reuse of transformed input blocks can still be achieved, but the memory requirements increase α -fold. As will be shown, the advantage of $\alpha > 1$ is in terms of increased convergence speed and higher overall levels of convergence. The improved performance is generally attributed to the ability to update the filter weights more often than every R^{th} output sample as will be discussed below.

Figure 5 schematically shows the effect of α on the generation of the output samples for $R=4$. Formally, let

$$\mathbf{g} = [g_0 \ g_1 \ \dots \ g_{\alpha-1}]^T \quad (18)$$

be the vector of reconstruction weights, with the constraint

$$\sum_{i=0}^{G-1} g_i = 1 \quad (19)$$

then the final output samples are generated by

$$y_n = \sum_{i=0}^{G-1} g_i \hat{y}_{(s-i); (j+iP)} \quad (20)$$

where $\hat{y}_{m;n}$ represents the n^{th} *element* of the m^{th} iteration output vector, $\hat{y}_m$ and $j = \text{remainder}((n+1)/(sP))$ where s is the current iteration and P is defined as in (17).

Figure 5 Example of output sample generation for $R=4$, $\alpha=1,2$ and 4. Output samples require reconstruction from α estimates. $\hat{y}_s$ is the block output for the s^{th} iteration and is truncated to size R as suggested by equation (16).

Updating of Filter Coefficients

The time domain LMS update is well known as

$$\mathbf{w}_{s+1} = \mathbf{w}_s + \mu \mathbf{x}_s e_s \quad (21)$$

where μ is the step size and e_s is the error between the current output, y_s and the desired output, d_s i.e.

$$e_s = y_s - d_s = \mathbf{w}_s^T \mathbf{x}_s - d_s \quad (22)$$

In (21) the estimate of the gradient vector is comprised of one error value, multiplied by the current input vector, i.e.

$$\hat{\nabla}_s = \mathbf{x}_s e_s \quad (23)$$

The frequency domain version of the LMS is a block-based LMS. The equivalent time domain version will have a vector of errors, not a scalar error, with each element of the error vector associated with a corresponding input vector. The result is that the estimate of the gradient vector is averaged across the error vector as shown by

$$\hat{\nabla}_s = \chi_s e_s \quad (24)$$

where

$$\chi_s = \begin{bmatrix} x_{sP} & x_{sP+1} & \cdots & x_{sP+R-1} \\ x_{sP-1} & x_{sP} & \cdots & x_{sP+R-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{sP-R+1} & x_{sP-R+2} & \cdots & x_{sP} \end{bmatrix} \quad (25)$$

and

$$\mathbf{e}_s = \begin{bmatrix} y_{sP} & \cdots & y_{sP+R-1} \end{bmatrix}^T - \begin{bmatrix} d_{sP} & \cdots & d_{sP+R-1} \end{bmatrix}^T \quad (26)$$

From (24) we can write the j^{th} element of the gradient estimate as

$$\hat{\nabla}_{s,j} = \sum_{i=0}^{R-1} \mathbf{e}_{s,i} x_{sP-j+i} \quad (27)$$

which is the cross-correlation between the error vector and an input vector. Linear correlation, like linear convolution, may be performed in the frequency domain with appropriate zero-padding and element stripping [36]. Specifically, by expressing the frequency domain version of the error as

$$\mathbf{E}_s = \mathbf{F}_M \begin{bmatrix} \mathbf{0}_{M-R,1} \\ \mathbf{e}_s \end{bmatrix} \quad (28)$$

and

$$\mathbf{X}_s = \mathbf{F}_M \begin{bmatrix} x_{sP-R-1} \\ \vdots \\ x_{sP+R-1} \\ 0 \end{bmatrix} \quad (29)$$

we can write the circular correlation as

$$\tilde{\nabla} = \mathbf{F}_M^{-1} (\mathbf{X}_s^* \otimes \mathbf{E}_s) \quad (30)$$

where $\mathbf{X}_s^*$ is the complex conjugate of $\mathbf{X}_s$ and subsequently, the linear correlation as

$$\hat{\nabla}_s = \begin{bmatrix} \mathbf{I}_R & \mathbf{0}_{R,R} \\ \mathbf{0}_{R,R} & \mathbf{0}_{R,R} \end{bmatrix} \tilde{\nabla}_s \quad (31)$$

The difference between linear and circular correlation has led to two versions of the GMDF, known as constrained and unconstrained GMDF. It has been found that the unconstrained version suffers some performance degradation but is often worth the significant computational savings resulting from the 2K less DFTs required per iteration [21, 25]. Writing the constraint operator as

$$\mathbf{C} = \mathbf{F}_M \begin{bmatrix} \mathbf{I}_R & \mathbf{0}_{R,R} \\ \mathbf{0}_{R,R} & \mathbf{0}_{R,R} \end{bmatrix} \mathbf{F}_M^{-1} \quad (32)$$

we can then write the constrained frequency domain weight update equation as

$$\mathbf{W}_{s+1} = \mathbf{W}_s + \mu C(\mathbf{X}_s^* \otimes \mathbf{E}_s) \quad (33)$$

and the unconstrained version as

$$\mathbf{W}_{s+1} = \mathbf{W}_s + \mu (\mathbf{X}_s^* \otimes \mathbf{E}_s) \quad (34)$$

One of the advantages of frequency domain filtering is the ability to normalize the step size based on the power levels in each frequency bin. If we let

$$\mathbf{T}_s = \begin{bmatrix} P_{s,1}^{-1} \\ \vdots \\ P_{s,M}^{-1} \end{bmatrix} \quad (35)$$

where $P_{s,i}$ is an estimate of the power level in the i^{th} bin at iteration s , we can then write the normalized constrained and unconstrained weight update equations as

$$\mathbf{W}_{s+1} = \mathbf{W}_s + \mu C(\mathbf{T}_s \otimes \mathbf{X}_s^* \otimes \mathbf{E}_s) \quad (36)$$

and

$$\mathbf{W}_{s+1} = \mathbf{W}_s + \mu (\mathbf{T}_s \otimes \mathbf{X}_s^* \otimes \mathbf{E}_s) \quad (37)$$

respectively.

It should be noted that these equations correspond to a single subfilter and would need to be repeated K times for each iteration.

Asymptotic Computational Complexity

A valid definition of computational complexity is difficult given the varied structures of DSP processors available today, however, some insight is gained by examining the number of real multiplies and real additions required by the GMDF algorithm and comparing them to those required by NLMS. Table 1 lists the approximate number of operations required by NLMS and unconstrained GMDF, where N is the size of the impulse response, K is the number of subfilters and R is the subfilter block size. As well, the DFT size is $M=2R$. For the fast affine projection (FAP) algorithm, N is the filter length and P is the projection order.

Algorithm	Real Mults	Real Adds
NLMS	$2N$	$2N$
FAP	$2N+20P$	$N+7P$
GMDF	$\alpha(24K+8\log_2 R)$	$\alpha(16K+12\log_2 R)$

Table 1 Real mults and adds for NLMS, FAP and GMDF calculated per output sample. Note: $N=KR$.

As an example, we can compare the real multiplication operations required for each type of filter given an impulse response of 640 taps, projection order of 10 (for the FAP), with a standard configuration of α equal to half the blocksize, i.e. $\alpha=R/2$ (for the GMDF). In this case, for NLMS we require approximately 1280 operations. For the FAP we require approximately 2190 operations. For GMDF, we require $7,680 + 4R\log_2 R$ operations. Good simulation results have been obtained with $R=16$, in which case the number of operations would be approximately 7936, or 6 times the requirement for NLMS (or 3.6 times the requirement for FAP). For this additional cost, the improvement

in performance is quite significant, as will be shown below (see figure 6 on page 24). Note that with a sampling frequency of 8kHz, with $R=16$ (and a DFT blocksize of $2R$ i.e. 32) the lowest frequency that can be represented will have a period of $32/8000$ seconds,

$$f_{\min} = 8000 / 32 = 250\text{Hz} \quad (38)$$

that is

This will also be the frequency resolution of the DFT block, i.e. the frequency bins represent increments of 250Hz. Constrained by the desire to use power of 2 block sizes, if we increase our block size to 64, $f_{\min}$ will be 125 Hz, which is unnecessarily low, given that the input signal is telephone bandwidth limited speech. (It will allow better frequency resolution however.) Alternatively, if we use a smaller block size, e.g. 16, $f_{\min}$ (and the frequency resolution) becomes 500 Hz, which may be too high to represent the low-frequency content of the signal or the finer details of the signal in the frequency domain. Hence, a block size of 32 ($R=16$) is not purely an arbitrary choice but represents a trade-off between computational complexity and frequency resolution.

Low-delay and Delayless GMDF

The perceptual effect of throughput delay exacerbates the acoustic echo cancellation problem [18,19]. Conventional subband filtering with delay being introduced through the bandpass filters, and block transform filtering with large blocks, often suffer from this disadvantage [22]. The GMDF algorithm allows large impulse responses to be achieved with block processing, but where the size of each subfilter block (and hence the throughput delay) can be reduced, at the expense of an increase in the number of subfilters.

For example, referring back to figure 4 on page 11, the generation of output sample y_n involves future samples of x , specifically, x_{n-1} , x_{n+2} and x_{n+3} . By reducing the block size, the delay would be reduced. The delay is therefore determined by

$$D = (R-1)/f_s$$

where f_s is the sampling frequency.

Unfortunately, the block size cannot be reduced beyond a certain point without a substantial decline in performance due to poor frequency resolution. To illustrate this, figure 6 shows the average ERLE for a system operating on 2.3 seconds of synthetic speech as recorded in a real conference room. The filter has the equivalent of $N=640$ taps, and block sizes are $R = 32, 16, 8, 4, 2$ and 1 , with corresponding subfilters numbering $K = 20, 40, 80, 160, 320$ and 640 (such that $N=KR$). The average ERLE for NLMS is also plotted, as a reference performance level, along with the average ERLE for the Fast Affine Projection algorithm (projection order = 10).

Figure 6 Average ERLE vs. throughput delay for NLMS, FAP (10th order) and GMDF with different block sizes (indicated by R).

There are two aspects that contribute to the delay in the GMDF algorithm, filtering and coefficient updating. Delays in filtering are due to the use of future samples of the input sequence (reference signal) during the blocking operation. Delays from coefficient updating are due to the use of future samples of the desired sequence (primary signal) as well as the input sequence.

Examination of equation (10) on page 12 shows that $\mathbf{x}_k^{(M)}$ involves input sequence samples greater than n only when $k=0$, i.e. only the first subfilter makes use of future samples. This suggests the *filtering* operation can be made delayless by operating the first block in the time-domain (all other blocks continue to operate in the frequency domain).

The coefficient update aspect is more problematic, since every subfilter's adaptation is based on the same error vector, defined as

$$\mathbf{e}_s = [\mathbf{y}_{sP} \ \cdots \ \mathbf{y}_{sP+R-1}]^T - [\mathbf{d}_{sP} \ \cdots \ \mathbf{d}_{sP+R-1}]^T \quad (41)$$

Figure 7 is a schematic representation of the use of the error vector in relation to the generation of the current output sample with $R=4$ and $\alpha=R$. Let us examine the generation of y_n . The filter coefficients used to produce $\hat{y}_s$ are updated using the error vector $\mathbf{e}_{s-J}$. Some of the elements of this vector involve future samples, as indicated by placement of the vertical dotted line, representing the present time. Only by basing the filter coefficient update on the error vector of α iterations in the past, do we obtain coefficient updating with no dependence on future samples. Hence, delayless filtering requires delayed filter coefficient updating. Unfortunately, this has a direct effect on the rate of convergence, since the step size must be decreased for stability [35]. (The effect of “incorrect” gradient estimates cannot be immediately recognized in the next iteration.)

Figure 7 suggests an alternative method of achieving *low delay* GMDF processing, without decreasing the block size, that is, perform the filtering operation for the first subfilter block in the time domain and choose an appropriate coefficient update delay based on a tolerable decline in the rate of convergence. In other words, the more delay that can be tolerated in the coefficient updates, the less throughput delay will result. By delaying the coefficient update by a full α iterations, completely delayless throughput will be obtained. To illustrate the effect on convergence of this method, Figure 8 shows 25 seconds of ERLE for 8 different delay amounts. The signal in this case was filtered white noise (identical samples for each trial were used), with a simulated conference room impulse response and a block size of 8. Two observations are noted - first the rate of convergence is slowed by decreasing the throughput delay, and second, the steady-state maximum ERLE is decreased. The amount of steady-state ERLE reduction is not linearly related to the delay however, and delays of less than 1ms eventually converge to approximately 80 dB.

Figure 7 Schematic representation of the relationship in time between the error vector, e_s , and the output sample y_n , with $R=4$ and $\alpha=R$.

Figure 8 ERLE for different throughput delay amounts using $R=16$, $\alpha=8$ for filtered white noise and a simulated conference room impulse response of 640 taps.

Figure 9 compares the performance of the new method of obtaining delayless GMDF and NLMS for synthetic speech recorded in a real conference room. The new method clearly outperforms the small block method and averages approximately 2 dB/sample better ERLE than the NLMS method for the full 2.3 seconds, or approximately 3 dB/sample for the first second. Notice that the NLMS outperforms the new method for only the first 150ms. The major dips in the ERLE curves are due solely to speech power dropping between words and do not indicate performance degradation.

Figure 9 ERLE for delayless GMDF and NLMS

GMDF with Adaptive Reconstruction Filters

Further performance gains can be achieved by examining the role of the reconstruction filter. Traditionally, this filter has been restricted to being a moving average (MA) filter, combining the estimates of each output sample to produce a final output sample.

At each iteration of the GMDF's main filter, an estimate of the output at a block of discrete time locations is made, i.e. $\hat{\mathbf{y}}_s$. An examination of the error variance of these estimates shows that the error variance is higher at the beginning and end of the block of estimates. This is especially evident with the *unconstrained* GMDF algorithm, where circular correlation is used instead of linear correlation. Figure 10 plots the error variance as a function of the intra block index for synthetic speech recorded in a conference room, using differing amounts of overlap ($\alpha=1,2,4,8$, $R=16$).

Examining figure 10, it is evident that since the output estimates have different error variance, a moving average filter, where each element of the reconstruction vector (defined below) is weighted by the same amount may not, therefore, produce the best final output sample. Intuitively, it would be better to weight more heavily those estimates with the least error variance.

Figure 10 Error variance vs. intra-block index for unconstrained GMDF.

Note that figure 10 is the result for a single speaker. Figure 11 shows the error variance for eight different speakers, using the same overlap. (Overlap=44%, or $\alpha=2$) Here we see that there is sufficient difference amongst the speakers to warrant an adaptive reconstruction filter approach, as opposed to a fixed reconstruction filter of appropriate weighting factors. That is, we know that a moving average filter is not appropriate. The question then becomes, is there an optimal fixed weight reconstruction filter? Based on figure 11, it would appear not; there is sufficient variation between speakers to warrant an adaptive approach.

Figure 11 Error variance vs. intra-block index for eight speakers (Males shown black, females shown red.)

Mathematically, we can express the output of the reconstruction filter as in equation (20), repeated here for convenience:

$$y_n = \sum_{i=0}^{\alpha-1} g_i \hat{y}_{(s-i); (j+iP)} \quad (42)$$

where $j = \text{remainder}((n+1)/(sP))$ or more conveniently, in matrix notation as

$$y_n = \mathbf{g}^T \mathbf{v}_j^{(s)} \quad (43)$$

with $\mathbf{v}_j^{(s)}$ defined as a vector of past estimates corresponding to the same time location, i.e. the reconstruction vector:

$$\mathbf{v}_j^{(s)} = \begin{bmatrix} \hat{y}_{s;j} \\ \hat{y}_{(s-1); (j+P)} \\ \hat{y}_{(s-2); (j+2P)} \\ \vdots \\ \hat{y}_{(s-\alpha+1); (j+(\alpha-1)P)} \end{bmatrix} \quad (44)$$

(Figure 12 shows an example of $\mathbf{v}_j$ and its relation to $\hat{\mathbf{y}}_s$.) Representing the reconstruction filter at iteration s as $\mathbf{g}^{(s)}$, the NLMS update equations are well known as:

$$\mathbf{g}^{(s+1)} = \mathbf{g}^{(s)} + \frac{\mu_R \mathbf{v}_j^{(s)} e_n}{\mathbf{v}_j^{(s)T} \mathbf{v}_j^{(s)}} \quad (45)$$

where μ_R is the reconstruction filter step-size and e_n is the reconstruction error,

defined as

$$e_n = y_n - \mathbf{g}^{(s)T} \mathbf{v}_j^{(s)} \quad (46)$$

In addition to this, when the α parameter is chosen to be smaller than maximum, i.e. smaller than the block size R , the result is that more than one output sample is produced during each iteration. That is, P samples are produced where

$$P = \frac{R}{\alpha}, \quad P, \alpha \in \mathbb{I} \quad (47)$$

and R is given by equation (5).

The P samples are generated from reconstruction vectors $\mathbf{v}_j^{(s)}$, $j=1\dots P$ and equation (45) should be modified to reflect the fact that more than one reconstruction vector is available to calculate the gradient estimate. There are (at least) two possible modifications that can be used. Firstly, the gradient can be averaged over the P vectors using a standard *block* LMS approach, e.g.

$$\mathbf{g}^{(s+1)} = \mathbf{g}^{(s)} + \frac{\mu_R \mathbf{V}^{(s)} \mathbf{e}}{\text{tr}[\mathbf{V}^{(s)T} \mathbf{V}^{(s)}]} \quad (48)$$

where $\mathbf{V}^{(s)}$ is defined as

$$\mathbf{V}^{(s)} = \begin{bmatrix} \mathbf{v}_1^{(s)} & \mathbf{v}_2^{(s)} & \dots & \mathbf{v}_P^{(s)} \end{bmatrix} \quad (49)$$

and the error vector, $\mathbf{e}$ is defined as $\mathbf{e} = [e_n \ e_{n+1} \ \dots \ e_{n+P-1}]^T$ and $\text{tr}[\cdot]$ is the trace of the matrix. This approach is subsequently referred to as the Single Adaptive

$\alpha=4, R=8$

Figure 12 Example of the relationship between $\hat{y}_s$, $\hat{y}_{s;i}$ and $v_j^{(s)}$ for the case of $R=8, \alpha=4$.

Reconstruction Filter (SARF). The second approach is to have a separate adaptive filter for each of the P reconstruction vectors. This method is referred to as the Multiple Adaptive Reconstruction Filters method (MARF) and is described mathematically in the following manner:

Let the matrix

$$\mathbf{G}^{(s)} = \begin{bmatrix} \mathbf{g}_1^{(s)} & \mathbf{g}_2^{(s)} & \dots & \mathbf{g}_P^{(s)} \end{bmatrix} \quad (50)$$

represent the P adaptive reconstruction filters, where the $\mathbf{g}_j^{(s)}$ are defined as

$$\mathbf{g}_j^{(s)} = \begin{bmatrix} g_{j;0}^{(s)} & g_{j;1}^{(s)} & \dots & g_{j;\alpha-1}^{(s)} \end{bmatrix}^T \quad (51)$$

The P output values generated at each iteration are then defined as

$$\begin{bmatrix} y_n \\ y_{n+1} \\ \vdots \\ y_{n+P-1} \end{bmatrix} = \text{diag} \left[\mathbf{G}^{(s)T} \mathbf{V}^{(s)} \right] \quad (52)$$

and the update equations are given by

$$\mathbf{g}_j^{(s+1)} = \mathbf{g}_j^{(s)} + \frac{\mu_R \mathbf{v}_j^{(s)} e_{n+j-1}}{\mathbf{v}_j^{(s)T} \mathbf{v}_j^{(s)}}, \quad j=1 \dots P \quad (53)$$

Figure 13 compares the performance for two delayless GMDF systems for 2.3 seconds of synthetic speech recorded in a real conference room. The effective length of the main filter's impulse response was $N=640$, which was divided into $K=40$ subfilters of size $R=16$ blocksize. In figure 13, the two trials both used the maximum α value possible, i.e. $\alpha=16$. Note that in this case, the SARF method is identical to the MARF method, since $P=1$. Using an adaptive reconstruction filter results in an average of 2.4 dB/sample improvement in ERLE.

Figure 13 ERLE for 2.3 seconds of synthetic speech recorded in a real conference room for a delayless GMDF system using a moving average reconstruction filter and an adaptive reconstruction filter.

As was shown in Table 1, the number of computations involved with the GMDF algorithm is directly proportional to α . Reducing α is therefore desirable, from a complexity point of view. Unfortunately, there will be a corresponding drop in performance when α is reduced. Empirically, a good compromise seemed to be $\alpha=R/2$. (With $R=16$, this represents an 8-fold decrease in the number of computations.)

Figure 14 was plotted with the same axis as figure 13 in order to show the overall reduction in performance resulting from $\alpha=R/2$. In this figure, the moving average reconstruction filter is plotted alongside the SARF and the MARF methods. As can be seen, the single adaptive filter performs only slightly better than the moving average method (0.08 dB/sample improvement), whereas the multiple adaptive filter method averages a performance improvement of 2.25 dB/sample over the single adaptive filter method and 2.34 dB/sample over the moving average method.

Figure 14 ERLE for 2.3 seconds of synthetic speech recorded in a real conference room for moving average, SARF and MARF.

Figure 15 compares the performance for a moving average reconstruction filter vs. a multiple adaptive reconstruction filter, using three versions of α , based on the block size. In all cases, the step size was set to 0.03125. A low step size value is needed, since the input to the adaptive reconstruction filter is almost DC (resulting in a steep error surface). As can be seen, the adaptive reconstruction method performs better than the moving average methods in almost all cases. These trials were conducted with synthetic speech recorded in a real conference room. The synthetic speech wave forms were comprised of phonetically balanced sentences representing female speech.

Frequency Response and Adaptation over Time of ARF Coefficients

Figure 16 shows the adaptation of adaptive reconstruction filter (ARF) coefficients over one second. In this experiment, two new output samples are produced during each iteration of the filter - requiring two reconstruction filters. (In this situation, $P=2$ in equation 50). The first filter's coefficients are shown in black, the second filter's coefficients in red. Comparing this to the error variance (see figure 11) we see that in general, intra block error values vary inversely with coefficient values. This is expected, as the filter adapts towards minimizing values that are contributing to greater amounts of error. The most error occurs with the largest intra block index, which during the reconstruction filter operation is multiplied in this example, with coefficient number eight. As can be seen, coefficient number 8 is rapidly decreased from its initial value of $1/\alpha$.

Additional insight may be gained by examining the frequency response of the reconstruction filters, as they adapt over time. Figure 17 shows the response during one experiment, where a reconstruction filter is initialized to a moving average filter (all coefficients equal $1/\alpha$). As can be seen, the filter quickly converges, and remains stable, at a new low-pass configuration. A low-pass response is expected since ideally, all the

Figure 15 Average ERLE vs. block size for multiple adaptive reconstruction filters (red) and moving average reconstruction filters (black) for $\alpha = \text{block size}$, block size/2 and block size/4.

estimates (the input to the reconstruction filters) will be identical. This is due to the fact that all of the estimates are estimating the output for an identical moment in time. Therefore, taken as an ensemble of estimates, one would ideally expect $\propto$ identical estimates, which in the frequency domain corresponds to a DC signal.

By examining the magnitude of the frequency components of the input signal and comparing them to the magnitude of the frequency response of the reconstruction filter, another explanation emerges for the new low-pass configuration of the reconstruction filter. Figure 18 shows a normalized histogram for the magnitudes of the input frequency components and the ARF frequency response over one second of adaptation. As can be seen, the ARF steers nulls towards the high frequency (error) components of the input signal.

Figure 16 Adaptation over time of ARF coefficients. First ARF filter shown in black, second ARF filter shown in red.

Figure 17 Adaptation over time of the magnitude of the normalized frequency response of an ARF.

Figure 18 Normalized histogram of one second of the adaptive reconstruction filter's frequency response (shown in red) and the input signal's frequency components (shown in black).

Noise Reduced GMDF

Generation of Filter Output Samples

In order to emphasize that the output from a GMDF-based AEC is an estimate of the echo, we reformulate the R -length vector of output sample estimates at each iteration, s , as $\hat{\mathbf{e}}_s = [\hat{e}_{s;0} \quad \hat{e}_{s;1} \quad \cdots \quad \hat{e}_{s;R-1}]^T$. Each vector element represents an output estimate for a specific point in time. As mentioned above, since the GMDF algorithm allows for an overlap to exist between successive iteration output blocks, there may be more than one estimate for each point in time, requiring a reconstruction filter to combine them to determine the final (single) output value.

Generation of NR Filter Output Samples

In the context of acoustic echo cancellation, the GMDF output is subtracted from a microphone signal to produce a system output $z = m - \hat{e}$. With a noise-reduced microphone signal we can define a noise-reduced system output as $z_{nr} = m_{nr} - \hat{e}$.

The noise-reduced microphone signal, m_{nr} is obtained from the inverse DFT of its frequency domain counterpart, i.e.

$$m_{nr}(n) = 1/N \sum_{k=0}^{N-1} M_{nr}(k) e^{j2\pi kn/N} \quad (54)$$

The task of estimating the magnitude and phase of M_{nr} is accomplished by the spectral subtraction technique, where

$$|M_{nr}(k)| = \left(|M(k)|^\lambda - \beta \mathbf{E}[|S(k)|^\lambda] \right)^{\frac{1}{\lambda}}, \quad k = 0, 1, \dots, N-1 \quad (55)$$

and

$$\angle M_{nr}(k) = \angle M(k) = \frac{M(k)}{|M(k)|}, \quad k = 0, 1, \dots, N-1 \quad (56)$$

where $\mathbf{E}[|S(k)|]$ is the expectation of the magnitude of the spectral coefficients of the noise. As described in [29], variants of the technique can be accounted for by different choices for λ and β . As discussed in [1] and [40], perceptual tolerances phase errors allow us to simply use the phase of the noisy signal as an estimate of the noise-reduced signal's phase.

With $\lambda = \beta = 1$, we may write

$$M_{nr}(k) = M(k) - \mathbf{E}[|S(k)|] \frac{M(k)}{|M(k)|}, \quad k = 0, 1, \dots, N-1 \quad (57)$$

so that in the time domain as well, we may separate the microphone signal from the noise estimate as

$$m_{nr}(n) = m(n) - \frac{1}{N} \sum_{k=0}^{N-1} NR(k) e^{\frac{j2\pi kn}{N}} \quad (58)$$

where the frequency-domain noise-reduction component is defined as

$$NR(k) = \mathbf{E}[\|S(k)\|] \frac{M(k)}{|M(k)|}, \quad k = 0, 1, \dots, N-1 \quad (59)$$

The GMDF algorithm requires the calculation of the spectral coefficients of the microphone signal in order to calculate the frequency domain error. This suggests a simple adaptive filter mechanism for estimating $NR(k)$, where the input to the filter is $M(k)/|M(k)|$ and the filter weights converge to approach $\mathbf{E}[\|S(k)\|]$, given an appropriate error criterion. (See below.)

Updating of NR Filter Weights

Fundamental to the NR adaptation equations is the definition of adaptation error and the sequence order of both the adaptation error and the weight update calculations. (See figures 19 and 20.) Let us define an AEC-error, $AECerr(k) = M(k) - \hat{E}(k)$. This exclusion of the NR filter effects results in the AEC filter being independent of the NR filter. The NR error is then defined as $NRerr(k) = AECerr(k) - NR(k)$. With a mean-square-error cost function of

$$J(k) = \mathbf{E}[NRerr^2(k)] \quad (60)$$

and the NR filter weights defined as $NRtaps(k)$, following [31] we can then write the one-tap, complex LMS update equation as

$$NRtaps_{s+1}(k) = NRtaps_s(k) + NR\mu \left(\frac{M(k)}{|M(k)|} NRerr^*(k) \right) \quad (61)$$

where $*$ indicates complex conjugation and $NR\mu$ is the step size. Note that there are N single tap filters, one for each frequency bin. Also, since the NR filter error is complex, the NR tap weights will become complex, even though the NR tap weight is estimating a real value. This is due to phase estimation errors of $\hat{E}(k)$ which propagate phase

Figure 19 Schematic of Noise-Reduced GMDF.

Figure 20 Details of NR filter for N spectral components

errors to $NRerr$. Restricting the tap weights to real values results in a decrease in noise-reduction performance, as shown below. What is also of interest is that the NLMS and LMS algorithms will have the same update equation, since it is always the case that $\|M(k)/\|M(k)\| = 1$.

Convergence of the Noise Reduction Filter

The cost function is given as

$$\begin{aligned}
 J(k) &= E[NRerr(k)NRerr^*(k)] \\
 &= E[|AECerr(k)|^2] - NRtaps(k)E\left[AECerr(k)\frac{M^*(k)}{|M(k)|}\right] \\
 &\quad - NRtaps^*(k)E\left[AECerr^*(k)\frac{M(k)}{|M(k)|}\right] \\
 &\quad + NRtaps^*(k)E\left[\frac{M(k)M^*(k)}{|M(k)|^2}\right]NRtaps(k)
 \end{aligned} \tag{62}$$

Taking the partial derivative with respect to the tap weights and equating to zero, we obtain

$$\begin{aligned}
 \frac{\partial J(k)}{\partial NRtaps(k)} &= -2p^* + 2NRtaps(k) = 0 \\
 \Rightarrow NRtaps(k) &= p^*
 \end{aligned} \tag{63}$$

where the cross correlation p is given by

$$p = E\left[AECerr^*(k)\frac{M(k)}{|M(k)|}\right] \tag{64}$$

Allowing the filter to adapt only when no near-end or far-end speech is present, $AEC_{err}(k) = M(k) = S(k)$, and the optimum tap weights will therefore be

$$NRtaps_{opt}(k) = E \left[\frac{S(k)S^*(k)}{|S(k)|} \right] = E[|S(k)|] \quad (65)$$

which is the desired outcome.

Performance Evaluation

Two sets of trials were conducted, the first set examined the performance of the algorithm based on ERLE and system distance. The second set examined the effect of the algorithm on the error rates of a state-of-the-art commercial speech recognition engine.

ERLE and System Distance Trials

These trials were conducted using 4 male and 4 female talkers. The speech was convolved with a known loudspeaker-room-microphone impulse response truncated to 640 taps, to produce an echo signal. Four types of near-end noise were added to the echo at average SNRs from -5 to 15dB. Mean ERLE results for the first 1.75 seconds of adaptation are plotted in figure 21 with the results averaged over the 8 talkers. These results show an increase of approximately 9 dB at low SNR (-5 dB) (where regular GMDF fails, i.e. < 0 dB mean ERLE) and 6 dB at higher SNR (15 dB). For the case where the NR tap weights are restricted to real values, there is a decrease in the ERLE by approximately 3 dB.

The additional computations over the regular GMDF are $11N$ real multiplies and $8N$ real additions per P output samples (based on an arctan table lookup for the calculation

of the angle of the input spectral coefficients in equation 59). For example, typical values of N and P of 64 and 8 respectively (using an estimated impulse response of 640 taps, divided into 10 subfilters with 8 output samples per iteration) require 88 multiplies and 64 additions per output sample. In the same context, regular GMDF requires 2440 real multiplies and 2039 real additions per output sample, therefore the noise reduction enhancement requires less than 4% additional computations.

Figure 21 Mean ERLE vs SNR, with NR complex-valued tap weights (solid lines), with NR real-valued tap weights (dotted lines) and without NR (dashed lines) for car noise (circles), office noise (diamonds), street noise (stars) and air-conditioning noise (squares).

Speech Recognition Engine Trials

In order to evaluate the ability of the noise-reduction algorithm to improve existing speech recognition performance, trials were conducted using a corpus of cellular phone recordings from the Oregon Graduate Institute of Technology's Centre for Spoken Language Understanding (CSLU). The database consisted of 500 utterance for each digit from zero to nine, as recorded by volunteers phoning the CSLU automatic database collection program, for a total of approximately 5000 utterances. To this set of utterances, car noise was added at SNRs of 0, 5, 10 and 15dB to form four new sets of utterances. The four sets were then passed through the NR-GMDF algorithm to produce an additional four sets for a total of eight sets of utterances. The eight sets were then submitted to the NuanceTM speech recognition engine and scored for correctness in the determination of the classification of each utterance. The results are shown in figure 22.

Figure 22 Error percent vs. SNR for utterances with and without NR-GMDF processing

Chapter 3

Linear Predictive Coding Based Acoustic Echo Cancellation

The idea of decorrelating an adaptive filter's input signal through the use of linear prediction in order to improve echo cancellation performance was first discussed in [41] where the problem of *line* echo cancellation was examined. In their case, the small impulse response of the unknown system's transfer function allowed for simplifications in the derivations. Subsequently, in [28], the idea was extended to acoustic echo cancellation combined with linear predictive coding, but here again, simplifications were introduced in the assumption of non-changing prediction coefficients and the lack of a long-term prediction filter. In contrast, the work presented here makes no simplifying

assumptions with regards to the prediction coefficients and includes a long-term prediction filter, as is commonly found in actual coding systems. The coding process was based on the GSM enhanced full-rate coder [8] where a 10th order LP model is used, but with a 1-tap long-term predictor instead of an adaptive code book.

The remainder of this chapter is organized as follows: The next section describes a general structure and a simplified structure for LPC-AEC without a long-term predictor, for the stationary and non-stationary case. Basic relationships are derived. Following this, the long-term predictor is introduced into the simplified structure and the new relationships between the LPC parameters, before and after the room's transfer function, are derived. The adaptation equations for controlling a filter which estimates those relationships is then derived. Following this, the simulations are described and the results are compared with the traditional method of AEC and with results obtained using simplifications introduced by previous authors. Finally, the computational complexity of such an approach is examined and compared with the traditional method.

General & Simplified Structure for LPC-AEC (without long-term predictor)

The general structure is given in figure 23, where the loudspeaker-room-microphone transfer function produces an echo $e(n)$ and a near-end speech signal $s(n)$ from the far-end speech, $f(n)$ and an unknown source respectively. In the LPC decoder, we have available three signals of interest: $\mathbf{X}_f$, the (short-term) far-end excitation vector, $\mathbf{w}_f$, the far-end LP coefficients and $f(n)$. The problem is to estimate $\mathbf{X}_e$ and $\mathbf{w}_e$ (the short-term echo excitation vector and the echo LP coefficients) as some function of the three available signals. H1 and H2 represent these (as yet unknown) functions.

Figure 23 General structure for LPC-AEC.

Note that generation of $f(n)$ (if needed) can be incorporated into H1 and H2, but it is computationally more efficient to generate it only once. In contrast to the general structure, the simplified structure is given in figure 24. Here, $\mathbf{w}_f$ is directly used as an estimate for $\mathbf{w}_e$.

There are two primary cases of interest for each structure - when $f(n)$ is stationary (hence $\mathbf{w}_f$ doesn't change over time) and when $f(n)$ is non-stationary ($\mathbf{w}_f$ changes every 5ms). For each case, a relationship is derived for $\hat{\mathbf{X}}_e$ and $\hat{\mathbf{r}}_e$, the short-term excitation vector estimate and autocorrelation vector estimate respectively. From the autocorrelation vector, the estimate of the LP coefficients, $\hat{\mathbf{w}}_e$ is obtained through Levinson-Durbin recursion.

Case 1. Stationary Far-end Speech, General Structure

- Estimation of $r_e(\eta)$:

By definition:

$$r_e(\eta) = E[e(n)e(n-\eta)] \quad (66)$$

where $E[\cdot]$ is the expectation. Since $e(n)$ is the convolution of $f(n)$ with the room's impulse response, we may write

$$\begin{aligned} r_e(\eta) &= E \left[\sum_{j=0}^{L-1} h(j)f(n-j) \sum_{k=0}^{L-1} h(k)f(n-\eta-k) \right] \\ &= \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} h(j)h(k)r_f(\eta+k-j) \end{aligned} \quad (67)$$

where $h(j)$, $j = 0, 1, \dots, L-1$ is the loudspeaker-room-microphone impulse response.

With a little manipulation, this may be viewed as a filter operating on $r_f(\eta)$. That is, since $h(j) = 0$, $j < 0$, $j > L$, we may write

$$r_e(\eta) = \sum_{k=0}^{L-1} h(k) \sum_{j=k-L+1}^{k+L-1} h(j)r_f(\eta+k-j) \quad (68)$$

Figure 24 Simplified structure for LPC-AEC.

Let $i = j - k$. We may then write

$$r_e(\eta) = \sum_{i=-L+1}^{L-1} \sum_{k=0}^{L-1} h(k)h(i+k)r_f(\eta-i) \quad (69)$$

Defining the “autocorrelation” of $h(k)$ as

$$r_h(i) = \sum_{k=0}^{L-1} h(k)h(i+k) \quad (70)$$

we may then write the filter operation as

$$r_e(\eta) = \sum_{i=-L+1}^{L-1} r_h(i)r_f(\eta-i) \quad (71)$$

- Estimation of $X_e(n)$:

If we examine the filtering of $f(n)$ to obtain $e(n)$, we have

$$e(n) = \sum_{j=0}^{L-1} h(j)f(n-j) \quad (72)$$

$f(n)$ can be expressed as an LPC model as follows

$$f(n) = \sum_{i=1}^M w_f(i)f(n-i) + X_f(n) \quad (73)$$

and substituting this into equation (72) gives us

$$\begin{aligned} e(n) &= \sum_{j=0}^{L-1} h(j) \left[\sum_{i=1}^M w_f(i)f(n-j-i) + X_f(n-j) \right] \\ &= \sum_{j=0}^{L-1} h(j) \sum_{i=1}^M w_f(i)f(n-j-i) + \sum_{j=0}^{L-1} h(j)X_f(n-j) \end{aligned} \quad (74)$$

$e(n)$ may also be expressed as an LPC model. That is,

$$e(n) = \sum_{i=1}^M w_e(i)e(n-i) + X_e(n) \quad (75)$$

Combining equations (74) and (75) and solving for $X_e(n)$ we therefore obtain

$$\begin{aligned} X_e(n) &= \sum_{j=0}^{L-1} h(j) \sum_{i=1}^M w_f(i) f(n-j-i) + \sum_{j=0}^{L-1} h(j) X_f(n-j) - \sum_{i=1}^M w_e(i) e(n-i) \\ &= \sum_{j=0}^{L-1} h(j) \left[X_f(n-j) + \sum_{i=1}^M (w_f(i) - w_e(i)) f(n-j-i) \right] \end{aligned} \quad (76)$$

Case 2. Non-stationary Far-end Speech, General Structure

- Estimation of $r_e(\eta)$:

Rewriting equation (66) for the short term case gives us

$$r_e(\eta) = \sum_{n=\eta}^{P-1} e(n)v(n)e(n-\eta)v(n-\eta) \quad (77)$$

where $v(n)$ is a window function and is zero outside the range of $0 \leq n \leq P-1$.

Substituting for $e(n)$ we obtain

$$r_e(\eta) = \sum_{n=\eta}^{P-1} \sum_{j=0}^{L-1} h(j) f(n-j) v(n) \sum_{k=0}^{L-1} h(k) f(n-\eta-k) v(n-\eta) \quad (78)$$

or

$$r_e(\eta) = \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} h(j) h(k) \sum_{n=\eta}^{P-1} f(n-j) v(n) f(n-\eta-k) v(n-\eta) \quad (79)$$

Unfortunately, the right-most summation term does not simplify to the short-term autocorrelation of $f(n)$. Further simplification is only possible by using the *cross-correlation* between differently windowed versions of $f(n)$. Let us define a new

function consisting of $f(n)$ multiplied by a window shifted by $-j$.

Referring to this new function as $f_j(n)$, we then have

$$f_j(n) = \begin{cases} f(n)v(n+j), & -j \leq n \leq P-1-j \\ 0 & , \quad \text{otherwise} \end{cases} \quad (80)$$

and therefore

$$f_j(n-j) = \begin{cases} f(n-j)v(n), & 0 \leq n \leq P-1 \\ 0 & , \quad \text{otherwise} \end{cases} \quad (81)$$

Writing the cross-correlation between $f_j(n)$ and $f_k(n)$ we have

$$r_{f_j f_k}(\eta) = \sum_{n=-\infty}^{\infty} f_j(n) f_k(n-\eta) \quad (82)$$

and

$$\begin{aligned} r_{f_j f_k}(\eta + k - j) &= \sum_{n=-\infty}^{\infty} f_j(n-j) f_k(n-\eta-k) \\ &= \sum_{n=\eta}^{P-1} f_j(n-j) f_k(n-\eta-k) \end{aligned} \quad (83)$$

Finally, substituting into (79) gives us

$$r_e(\eta) = \sum_{j=0}^{L-1} \sum_{k=0}^{L-1} h(j) h(k) r_{f_j f_k}(\eta + k - j) \quad (84)$$

- Estimation of $X_e(n)$:

The substitution for $f(n-j)$ that occurs between (73) and (74) is

$$f(n-j) = \sum_{i=1}^M w_f(i) f(n-i-j) + X_f(n-j) \quad (85)$$

The assumption implicit in (85) is the same $\mathbf{w}_f$ is used for all previous blocks, i.e. the stationary case. In the non-stationary case, $\mathbf{w}_f$ changes at each new block, and bigger

“ j ”s correspond to “ $\mathbf{w}_f$ ”s from previous blocks in time. We can adjust for this in the following manner:

Let $\mathbf{w}_f$ represent the current block’s far-end LP coefficients.

Let $\mathbf{w}_f^{(-s)}$ represent the far-end LP coefficients, s blocks in the past.

With a block size of P and n from 0 to $P-1$ we want to rewrite equation (72) to explicitly show the blocks from which each $f(n-j)$ is derived.

Let the total number of previous blocks required be S , where

$$S = \left\lfloor \frac{L-1}{P} \right\rfloor \quad (86)$$

Therefore, in general, with $L > 2P$ (i.e. $S > 0$) we have

$$\begin{aligned} e(n) = & \sum_{j=0}^n h(j)f(n-j) \\ & + \sum_{j=SP+n+1}^{j=L-1} h(j)f(n-j) \\ & + \sum_{s=0}^{s=S-1} \sum_{j=sP+n+1}^{j=(s+1)P+n} h(j)f(n-j) \end{aligned} \quad (87)$$

where the first summation represents values from the current block, the second summation represents values from the most distant block (the S th block) and the third summation represents blocks between the current and the S th block.

Using the LP model for $f(n)$, with the coefficients explicitly shown on a block basis, we obtain

$$\begin{aligned}
e(n) = & \sum_{j=0}^n h(j) \left[\sum_{i=1}^M w_f(i) f(n-i-j) + X_f(n-j) \right] \\
& + \sum_{j=SP+n+1}^{L-1} h(j) \left[\sum_{i=1}^M w_f^{(-S-1)}(i) f(n-i-j) + X_f(n-j) \right] \\
& + \sum_{s=0}^{S-1} \sum_{j=(s+1)P+n}^{(s+1)P+n} h(j) \left[\sum_{i=1}^M w_f^{(-s-1)}(i) f(n-i-j) + X_f(n-j) \right]
\end{aligned} \tag{88}$$

In order to express (88) in terms of (75) we first add, then subtract all the coefficients corresponding to the current block's LP coefficients, i.e.

$$\begin{aligned}
e(n) = & \sum_{j=0}^n h(j) \sum_{i=1}^M w_f(i) f(n-i-j) + \\
& \sum_{j=SP+n+1}^{L-1} h(j) \sum_{i=1}^M w_f(i) f(n-i-j) + \\
& \sum_{s=0}^{S-1} \sum_{j=(s+1)P+n}^{(s+1)P+n} h(j) \sum_{i=1}^M w_f(i) f(n-i-j) + \\
& \sum_{j=SP+n+1}^{L-1} h(j) \sum_{i=1}^M \left(w_f^{(-S-1)}(i) - w_f(i) \right) f(n-i-j) + \\
& \sum_{s=0}^{S-1} \sum_{j=(s+1)P+n}^{(s+1)P+n} h(j) \sum_{i=1}^M \left(w_f^{(-s-1)}(i) - w_f(i) \right) f(n-i-j) + \\
& \sum_{j=0}^{L-1} h(j) X_f(n-j)
\end{aligned} \tag{89}$$

Combining the first three lines we obtain

$$e(n) = \sum_{i=1}^M w_f(i) \sum_{j=0}^{L-1} h(j) f(n-i-j) + Y(n) + \sum_{j=0}^{L-1} h(j) X_f(n-j) \tag{90}$$

or,

$$e(n) = \sum_{i=1}^M w_f(i) e(n-i) + Y(n) + \sum_{j=0}^{L-1} h(j) X_f(n-j) \quad (91)$$

Comparing this to (75) and solving for $X_e(n)$ we obtain

$$X_e(n) = Y(n) + \sum_{j=0}^{L-1} h(j) X_f(n-j) + \sum_{i=1}^M (w_f(i) - w_e(i)) e(n-i) \quad (92)$$

where

$$Y(n) = \sum_{j=SP+n+1}^{L-1} h(j) \sum_{i=1}^M (w_f^{-S-1}(i) - w_f(i)) f(n-i-j) + \sum_{s=0}^{S-1} \sum_{j=sP+n+1}^{(s+1)P+n} h(j) \sum_{i=1}^M (w_f^{-s-1}(i) - w_f(i)) f(n-i-j) \quad (93)$$

Note that if most of the energy of the loudspeaker-room-microphone impulse response is located within the first P samples, (93) may be approximated by

$$Y(n) \approx \sum_{i=1}^M (w_f^{(-1)}(i) - w_f(i)) \sum_{j=n+1}^{L-1} h(j) f(n-i-j) \quad (94)$$

Towards a Simplified Structure

A comparison of the LP coefficients before and after a speaker-room-microphone transfer function was undertaken, as a first step towards a simplified structure. Three impulse responses were truncated from 1200 taps down to 100, 256, 512 and 640 taps for a total of 12 impulse responses. Five seconds of female speech was filtered using the 12 impulse responses. Using a blocksize of 40 samples and a 10th order LPC model, a normalized LPC distortion measure was constructed as follows:

$$D_{f-e}(s) = \sum_{i=1}^{10} \frac{(w_f^{(s)}(i) - w_e^{(s)}(i))^2}{w_f^{(s)}(i)^2} \quad (95)$$

Averaging over all the blocks gives the mean normalized distance measure,

$$\overline{D}_{f-e} = \frac{1}{S} \sum_{s=1}^S D_{f-e}(s) \quad (96)$$

where S is the total number of blocks in the 5 second speech signal.

By plotting the average normalized distortion vs the impulse response length, we can see that in general, the distortion or difference between the LP coefficients before and after a room, increases as the impulse response increases. See figure 25.

Figure 25 Average normalized distortion vs. impulse response truncation point for 3 impulse responses, truncated at 100, 256, 512 and 640 taps.

Simplified Structure for LPC-AEC (with long-term predictor)

In the previous sections, only short term prediction was included in the analysis. However, as pointed out by Steele [30], long term prediction should be included to remove the periodicity from the excitation vector (which is related to the pitch period of the original speech).

In the following, α_f is the delay of the long-term filter, and G_f is the gain of the long-term filter. The simplified structure (see figure 24) is adopted where the echo signal is forced to conform to an LPC model where the LP coefficients are taken from the far-end speech, i.e.,

$$\mathbf{w}_e = \mathbf{w}_f, \quad \alpha_e = \alpha_f, \quad G_e = G_f \quad (97)$$

This means that only $\mathbf{X}_e$, the long-term excitation vector of the echo signal, needs to be estimated. This is discussed below for both cases of interest: i.e., when $f(n)$ is *stationary* (hence $\mathbf{w}_f, \alpha_f, G_f$ don't change over time) and when $f(n)$ is *non-stationary* ($\mathbf{w}_f, \alpha_f, G_f$ change every 5ms). For both cases we use the inverse-LPC synthesis filter structure as shown in figure 26.

Figure 26 Synthesis filters used to estimate the echo.

Here, $1/P(z)$ is the long-term synthesis filter and is defined by

$$P(z) = 1 - G_f z^{-\alpha_f} \quad (98)$$

and $1/A(z)$ is the short-term synthesis filter and is defined by

$$A(z) = 1 - \sum_{i=1}^M w_f(i) z^{-i} \quad (99)$$

From the short-term excitation, in the Z-domain we have

$$\hat{X}_{es}(z) = \hat{E}(z) \left[1 - \sum_{i=1}^M w_f(i) z^{-i} \right] \quad (100)$$

and in the time domain

$$\hat{X}_{es}(n) = \hat{e}(n) - \sum_{i=1}^M w_f(i) \hat{e}(n-i) \quad (101)$$

From the long-term excitation, in the Z-domain we have

$$\hat{X}_e(z) = \hat{X}_{es}(z) \left[1 - G_f z^{-\alpha_f} \right] \quad (102)$$

Combined with (101), in the time-domain we then have

$$\hat{e}(n) = \sum_{i=1}^M w_f(i) \hat{e}(n-i) + G_f \left[\hat{e}(n - \alpha_f) - \sum_{i=1}^M w_f^{(-c-l)}(i) \hat{e}(n-i - \alpha_f) \right] + \hat{X}_{e,l}(n) \quad (103)$$

where c is given by

$$c = \left\lfloor \frac{\alpha_f - n}{P} \right\rfloor \quad (104)$$

Case 1. Estimate of $\hat{X}_e(n)$ for Stationary Signal

Following directly from (103)

$$\hat{X}_e(n) = \hat{e}(n) - \sum_{i=1}^M w_f(i) \hat{e}(n-i) - G_f \left[\hat{e}(n - \alpha_f) - \sum_{i=1}^M w_f(i) \hat{e}(n-i - \alpha_f) \right] \quad (105)$$

and substituting each $\hat{e}(n)$ with the convolution of $f(n)$ and the estimate of the room impulse response, we obtain

$$\begin{aligned} \hat{X}_e(n) &= \sum_{j=0}^{L-1} h(j) f(n-j) \\ &\quad - G_f e(n - \alpha_f) - \sum_{i=1}^M w_f(i) [e(n-i) - G_f e(n-i - \alpha_f)] \\ &= \sum_{j=0}^{L-1} h(j) \left\{ G_f f(n-j - \alpha_f) + \sum_{i=1}^M w_f(i) [f(n-i-j) - G_f f(n-i - \alpha_f - j)] + X_f(n-j) \right\} \\ &\quad - G_f e(n - \alpha_f) - \sum_{i=1}^M w_f(i) [e(n-i) - G_f e(n-i - \alpha_f)] \\ &= \sum_{j=0}^{L-1} h(j) \left\{ G_f f(n-j - \alpha_f) + \sum_{i=1}^M w_f(i) [f(n-i-j) - G_f f(n-i - \alpha_f - j)] \right. \\ &\quad \left. - G_f f(n-j - \alpha_f) - \sum_{i=1}^M w_f(i) [f(n-i-j) - G_f f(n-i - \alpha_f - j)] \right\} \\ &\quad + \sum_{j=0}^{L-1} h(j) X_f(n-j) \\ &= \sum_{j=0}^{L-1} h(j) X_f(n-j) \end{aligned} \quad (106)$$

which confirms the equivalence of the two filter structures shown in figure 27.

This means that the excitation vector of the echo signal can be recovered by simply convolving the excitation vector of the far end speech with the room impulse response and that it is not related to the LP coefficients of the signal anymore. This result provides a significant simplification over the general structure discussed above.

Figure 27 Equivalent filter structures.

Case 2. Estimate of $\hat{X}_e(n)$ for Non-stationary Signal

Unfortunately, the simplification in the derivation of the echo excitation vector (see equation 106) is only valid for the stationary case. Writing all $\hat{e}(n)$ in equation (103) in terms of convolution of the far end speech with the room impulse response as in (106) we obtain

$$\begin{aligned} \sum_{j=0}^{L-1} h(j)f(n-j) &= G_f \sum_{j=0}^{L-1} h(j)f(n-j-\alpha_f) + \sum_{i=1}^M w_f(i) \sum_{j=0}^{L-1} h(j)f(n-j-i) \\ &\quad - \sum_{i=1}^M w_f^{(-c-1)} G_f \sum_{j=0}^{L-1} h(j)f(n-i-j-\alpha_f) + \hat{X}_e(n) \end{aligned} \quad (107)$$

Solving for $\hat{X}_e(n)$ we have

$$\begin{aligned} \hat{X}_e(n) &= \sum_{j=0}^{L-1} h(j)f(n-j) - G_f \sum_{j=0}^{L-1} h(j)f(n-j-\alpha_f) \\ &\quad - \sum_{i=1}^M w_f(i) \sum_{j=0}^{L-1} h(j)f(n-i-j) + \sum_{i=1}^M w_f^{(-c-1)} G_f \sum_{j=0}^{L-1} h(j)f(n-i-j-\alpha_f) \quad (108) \\ &= \sum_{j=0}^{L-1} h(j) \frac{\partial \hat{X}_e(n)}{\partial h(j)} \end{aligned}$$

where

$$\begin{aligned} \frac{\partial \hat{X}_e(n)}{\partial h(j)} &= f(n-j) - G_f f(n-j-\alpha_f) \\ &\quad - \sum_{i=1}^M w_f(i) f(n-i-j) + \sum_{i=1}^M w_f^{(-c-1)} G_f f(n-i-j-\alpha_f) \end{aligned} \quad (109)$$

Derivation of Adaptation Equations

The gradient is based on an error and there are two methods of defining the error, the

first based on $e(n)$, the second based on $X_e(n)$.

Method 1:

The error in the structure is defined as

$$err(n) = e(n) - \hat{e}(n) \quad (110)$$

and the cost function is J where

$$J(n) = E[err(n)^2] \quad (111)$$

The gradient of J is then given by

$$\nabla J(n) = \begin{bmatrix} \frac{\partial J(n)}{\partial h_0} \\ \frac{\partial J(n)}{\partial h_1} \\ \vdots \\ \frac{\partial J(n)}{\partial h_{L-1}} \end{bmatrix} \quad (112)$$

If we look at the k^{th} element of the gradient, we have

$$\begin{aligned} \nabla_k J(n) &= \frac{\partial}{\partial h_k} E[err(n)^2] \\ &= 2E \left[err(n) \frac{\partial}{\partial h_k} err(n) \right] \end{aligned} \quad (113)$$

and the derivative of the error with respect to h_k is

$$\begin{aligned}
\frac{\partial}{\partial h_k} err(n) &= \frac{\partial}{\partial h_k} (e(n) - \hat{e}(n)) \\
&= \frac{\partial}{\partial h_k} \left(e(n) - \sum_{j=0}^{L-1} h_j f(n-j) \right) \\
&= -f(n-k)
\end{aligned} \tag{114}$$

We can then write the *estimate* of the gradient for the k th element as

$$\hat{\nabla}_k J(n) = -2 \left[(e(n) - \hat{e}(n)) f(n-k) \right] \tag{115}$$

Here we see that this method results in the equivalent gradient to the traditional LMS case. That is

$$\hat{\nabla}_k J(n) = -2 (err(n) f(n-k)) \tag{116}$$

Method 2:(the proposed method)

The error in the structure is defined as

$$\begin{aligned}
err(n) &= X_e(n) - \hat{X}_e(n) \\
&= X_e(n) - \sum_{j=0}^{L-1} h(j) \frac{\partial \hat{X}_e(n)}{\partial h(j)}
\end{aligned} \tag{117}$$

The derivative with respect to h_k is

$$\frac{\partial}{\partial h_k} err(n) = -der(n, k) \tag{118}$$

where

$$der(n, k) = \frac{\partial \hat{X}_e(n)}{\partial h(k)} \quad (119)$$

Note that in this method, the derivative of the error is $der(n, k)$ instead of $f(n-k)$ as in the traditional method. Using this error in the adaptation scheme of the NLMS filter has certain advantages over the traditional method. First der is now being filtered instead of the speech signal f . Since der is whiter than f the convergence properties of the filter should be better. Second, the tap input vector in this method at time n is

$$der(n) = [der(n,1) \ der(n,2) \ \ der(n, M)] \quad (120)$$

at time $(n+1)$ the tap vector is

$$der(n+1) = [der(n+1,1) \ der(n+1,2) \ \ der(n+1, M)] \quad (121)$$

which means that the tap input vectors are not as statistically dependent as the tap input vectors in the traditional method.

This is an advantage, since the convergence rate using the LMS algorithm is inversely proportional to the condition number of the input correlation matrix. Figure 28 shows the condition number vs. time, for an LPC-AEC system and a traditional LMS system. As can be seen, the LPC-AEC system is expected to achieve quicker convergence in almost all iterations.

Figure 28 Condition number estimate vs time for LPC-AEC system and NLMS system (traditional AEC).

Computer Simulations

In order to evaluate the performance of the LPC-AEC algorithm and compare it against the traditional method of AEC, simulations were performed using real speech from 4 male and 4 female talkers. The final configuration of the LPC-AEC system is shown in figure 29. Echo for each trial was obtained by convolving the speech with a known conference room impulse response, truncated to 256 taps. Two measures were obtained for each trial, Echo Return Loss Enhancement (ERLE) (shown in figure 30), and System Distance (D) (shown in figure 31), defined below.

$$ERLE(n) = 10 \log_{10} \frac{E[e(n)]}{E[e(n) - \hat{e}(n)]} \quad (122)$$

$$D(n) = 10 \log_{10} \frac{\|\mathbf{h} - \hat{\mathbf{h}}(n)\|^2}{\|\mathbf{h}\|^2} \quad (123)$$

The System Distance is preferred over ERLE since it is not dependent on the input speech characteristics, and therefore gives a more accurate measure of convergence rate of the canceller. Figure 31 shows the System Distance for both methods (LPC-AEC and traditional) averaged over the 8 talkers. As is evident in the figure, the LPC-AEC method results in an improvement in rate and depth of convergence, achieving an improvement of greater than 13 dB over 1.5 seconds.

Figure 29 Proposed structure for LPC-AEC.

Figure 30 ERLE vs time for NLMS (traditional AEC) and LPC-AEC.

Figure 31 System Distance vs time for NLMS and LPC-AEC.

If stationarity of the far-end signal is assumed, the calculation of $\hat{X}_e(n)$ can be accomplished as in equation (106). We will refer to this as X_f based LPC-AEC. In contrast, for the non-stationary case the calculation of $\hat{X}_e(n)$ is accomplished as in equation (108) and is referred to as simply LPC-AEC. Figures 32 and 33 show the system distance for 2 trials. Each trial is averaged over 8 talkers (4 male, 4 female) and show the effect of the stationarity assumption when the impulse response is quite small (64 taps) compared to a larger impulse response (256 taps). As can be seen, the use of the “correct” calculation of $\hat{X}_e(n)$, i.e. equation (108) results in improved convergence only with larger impulse responses.

Figure 32 System Distance vs time for NLMS, Xf based LPC-AEC and LPC-AEC with true impulse response of 64 taps.

Figure 33 System Distance vs time for NLMS, Xf based LPC-AEC and LPC-AEC with true impulse response of 256 taps.

Chapter 4

Conclusions

Acoustic echo and ambient noise continue to be problematic for the hands-free interface environment. Transform domain processing holds advantages over traditional time domain processing, most notably in the speed and depth of convergence of adaptive filters whose function is to reduce the echo and noise components. Of the alternative domains, the most common is the frequency domain and the GMDF algorithm, with its generalized approach to subdividing the impulse response and allowing arbitrary amounts of block overlap, is well suited to handling typical conference room echo and speech signals. This thesis reports on three major improvements to the algorithm;

- reducing the throughput delay,

The throughput delay can be reduced or eliminated entirely by processing the first subfilter in the time domain and delaying the coefficient updating by α iterations. There is a corresponding decrease in cancellation depth, the larger the coefficient update delay.

- improving the convergence with only minor computational increases,

The error variance within each iteration's output block necessitates a reconstruction filter that is not a simple moving average filter. Due to observed differences in different speakers' error variance "profile", allowing the reconstruction filter to be adaptive increases the performance of the filter. The error variance problem (i.e. the error variance is not flat across the output block), is due to a computational "shortcut" referred to as the *unconstrained* GMDF.

In cases where more than one reconstruction filter is required for each iteration of the filter, the choice of allowing two independent adaptive reconstruction filters improves the performance over the case of restricting the algorithm to a single adaptive (or non-adaptive) reconstruction filter, (approximately 2.25 dB/sample increase over the single adaptive filter method and 2.34 dB/sample over the moving average method). In some cases, this means the performance can be maintained at an average ERLE amount and the number of computations can be reduced by half.

- combining a new noise-reduction method within the algorithm.

The noise-reduction component has been shown to improve the ERLE in simulations with a number of noise types. These include car noise, office noise, street noise and air-conditioning noise. In terms of improving the performance of an automatic speech recognizer, preprocessing with the noise-reduction component decreased the error rate significantly. (8.2% improvement at 0 dB SNR, based on a state-of-the-art

commercial automatic speech recognition engine¹.)

For mobile applications, the LPC domain is well suited, since most of the speech coding methods use some variant of LPC and some of the processing steps can therefore be shared between the coder and the echo canceller. The other main advantage of the LPC domain is an improvement in convergence rate due to the use of a noise-like innovation sequence, which spectrally, is flatter than the original speech sequence. It was also shown that:

- Using the exact equations governing the relationship between the echo excitation sequence and the far-end signal's model parameters allowed for more than 10 dB improvement in system distance, when the impulse response was at least 256 taps in length. Shorter impulse responses, e.g. 64 taps, showed only minor improvements.
- Shorter impulse responses would however, offer the ability of using an echo code book to store the necessary innovation sequences for the echo cancellation process, which reduces the adaptation process to a table lookup process, reducing algorithmic computational complexity.

Given a choice of which algorithm to use for a particular application, a number of factors must be considered. Firstly, there is the application's environment, specifically, the size of the impulse response being estimated. For larger impulse responses, as found in conference rooms, the GMDF algorithm is recommended, due to its ability to subdivide the impulse response, without incurring large block delays, common to block LMS schemes. For mobile environments or environments with smaller impulse responses, (or both, i.e. car interiors), the LPC method, especially the simplified structure, would be preferred for two reasons. Firstly, the simplification to the general LPC structure, as proposed here, assumes we can replace the LP coefficients of the echo

¹ NuanceTM Speech Recognition System, version 6.2, Nuance Communications.

signal with the LP coefficients of the far-end signal. As shown in chapter 3, the amount of distortion introduced by this simplification decreases, given a decrease in the length of the impulse response. Secondly, a smaller impulse response will mean that fewer past blocks of LP coefficient's values will need to be stored (and processed).

Future Work

Extensions to this work could involve modifying the combined AEC and noise reduced GMDF to exploit psychoacoustic properties of the human auditory processing system. This is a relatively new area of noise processing and offers the potential of eliminating the most perceptually annoying artifact of spectral-subtraction-based noise reduction algorithms - residual "musical" noise.

On the LPC side, additional work could be done in the extension of the idea to network echo cancellation, where the involved transfer functions are usually more stable and their associated impulse responses usually consist of sparse clusters of weighting coefficients. It is in this area where there is the most potential for echo code-books which could be used to reduce the cancellation procedure to a simple table-lookup process.

References

- [1] J. Lariviere, R. Goubran, "GMDF for noise reduction and echo cancellation," , *IEEE Signal Processing Letters*, vol 7, no. 8, pp. 230-232, August 2000.
- [2] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 126 -137, March 1999.
- [3] S. Gustafsson, P. Jax, A. Kamphausen, P. Vary, "A postfilter for echo and noise reduction avoiding the problem of musical tones," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.
- [4] C. Beaugeant, V. Turbin, P. Scalart, A. Gilloire, "New optimal filtering approaches for hands-free telecommunication terminals," *Signal Processing*, vol. 64, pp. 33-47, 1998.
- [5] S. Gustafsson, R. Martin, P. Vary, "Combined acoustic echo control and noise reduction for hands-free telephony," *Signal Processing*, vol. 64, pp. 21-32, 1998.
- [6] J. Lariviere, R. Goubran, "GMDF α with adaptive reconstruction filters and zero throughput delay," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3553-3557, 1998.
- [7] S. Gustafsson, R. Martin, P. Vary, "A novel psychoacoustically motivated audio enhancement algorithm preserving background noise characteristics," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [8] K. Jarvinen, J. Vainio, P. Kapanen, T. Honkanen, P. Haavisto, "GSM enhanced full rate codec," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 771-774, 1997.
- [9] A. Kumar, A. Gersho, "LD-CELP speech coding with nonlinear prediction," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 89-91, Apr. 1997.
- [10] S. McClellan, J.D. Gibson, "Variable-rate CELP based on subband flatness,"

- IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 2, pp. 120-130, Mar. 1997.
- [11] D.E. Tsoukalas, J. Mourjopoulos, G. Kokkinakis, "Perceptual filters for audio signal enhancement," *J. Audio Eng. Soc.*, vol. 45, no. 1/2, pp. 22-35, Jan./Feb. 1997.
 - [12] Y. Guelou, A. Benamar, P. Scalart, "Analysis of two structures for combined acoustic echo cancellation and noise reduction," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 637-640, 1996.
 - [13] R. Le Bouquin Jeannes, "Optimization of a cascaded structure for noise and echo cancellation using a noise filtering preprocessing," *Annals des Telecommunications*, vol. 51, no. 11-12, pp. 579-584, 1996.
 - [14] P. Scalart, A. Benamar, "A system for speech enhancement in the context of hands-free radiotelephony with combined noise reduction and acoustic echo cancellation," *Speech Communications*, vol. 20, no. 3, pp. 203-214, 1996.
 - [15] R. Le Bouquin, "Enhancement of noisy speech signals: Application to mobile radio communications," *Speech Communication*, vol 18, pp. 3-19, 1996.
 - [16] A. Kataoka, S. Hayashi, T. Moriya, S. Kurihara, K. Mano, "Basic algorithm of conjugate-structure algebraic CELP (CS-ACELP) speech coder," *NTT Review*, vol. 8, no. 4, pp. 24-29, Jul. 1996.
 - [17] N. Kitawaki, "An 8-kbit/s speech coding method (CS-ACELP) standardized by ITU," *NTT Review*, vol. 8, no. 4, pp. 16-23, Jul. 1996.
 - [18] *Recommendation ETR 250, Transmission and Multiplexing (TM); Speech communication quality from mouth to ear for 3.1 kHz handset telephony across networks*, European Telecommunications Standards Institute, July 1996.
 - [19] *Recommendation ITU-T G.131, Control of talker echo*, International Telecommunication Union, August 1996.
 - [20] E. Moulines, O.A. Amrane, Y. Grenier, "The generalized multidelay adaptive filter: structure and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp 14-28, Jan. 1995.

- [21] A. Baganne, A. Gilloire, E. Martin, P. Martineau, J.P. Thomas, "Implementation methods of an acoustic echo cancellation algorithm GMDF," *IEEE International Workshop on Acoustic Echo and Noise Control*, pp. 83-86, Jun. 1995.
- [22] D.R. Morgan, J.C. Thi, "A delayless subband adaptive filter architecture," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 1819-1830, Aug. 1995.
- [23] S. Gay, S. Tavathia, "The fast affine projection algorithm," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1995.
- [24] R. Martin, J. Altenhoner, "Coupled adaptive filters for acoustic echo control and noise reduction," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1995.
- [25] J. Prado, E. Moulines, "Frequency-domain adaptive filtering with applications to acoustic echo cancellation," *Ann. Telecommunications*, vol. 49, no. 7-8, pp. 414-428, 1994.
- [26] S. Hayashi, A. Kataoika, T. Moriya, "8 kbit/s short and medium delay speech codecs based on CELP coding," *European Transactions on Telecommunications and Related Technologies*, vol. 5, no. 5, pp. 583-590, 1994.
- [27] J.J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, pp. 14-37, Jan. 1992.
- [28] C. Acker, P. Vary, "Combined implementation of predictive speech coding and acoustic echo cancellation," *Signal Processing VI: Theories and Applications*, Elsevier Science Publishers B.V. pp. 1641-1644, 1992.
- [29] Y. Ephraim, "Statistical-model-based speech enhancement systems," *Proceedings of the IEEE*, vol. 80, no.10, pp. 1526-1555, Oct. 1992.
- [30] R. Steele (Ed), *Mobile Radio Communications*, New York, IEEE Press, 1992.
- [31] S.S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, N.J. 1991.
- [32] J-S. Soo, K.K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-38, no. 2, pp. 373-376, Feb. 1990.

- [33] P.C.W. Sommen, "On the convergence properties of a Partitioned Block Frequency Domain Adaptive Filter (PBFDAF)," *Signal Processing V: Theories and Applications*, L. Torres, E. Masgrau, and M.A. Lagunas (eds.) Elsevier Science Publishers B.V., 1990.
- [34] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journal of the Acoustic Society of America*, vol. 87, no. 4, pp. 1738-1752, Apr. 1990.
- [35] G. Long, F. Ling, J.G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 9, pp. 1397-1405, Sep. 1989.
- [36] A.V. Oppenheim, R.W. Schaffer, *Discrete-time signal processing*, Prentice-Hall Inc. Englewood Cliffs, N.J., pp. 553-560, 1989.
- [37] M. Xu, Y. Grenier, "Acoustic channel identification," *Proc. EUSIPCO-88*, pp. 1401-1404, 1988.
- [38] M.R. Schroeder, B.S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 937-940, 1985.
- [39] C.W.K. Gritton, D.W. Lin, "Echo cancellation algorithms," *IEEE Acoustic Speech and Signal Processing Magazine*, pp. 30-37, April 1984.
- [40] D.L. Wang, J.S. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, pp. 679-681, Aug. 1982.
- [41] S. Yamamoto, S. Kitayama, J. Tamura, H. Ishigami, "An adaptive echo canceller with linear predictor," *Transaction of the IECE of Japan*, vol. E-62, no. 12, pp. 851-857, Dec 1979.
- [42] B. Widrow, J.R. Glover, Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong, Jr., R.C. Goodlin, "Adaptive noise cancelling: principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, Dec. 1975.

Appendix A: Recording Room Characteristics

All simulations described in this thesis were processed with impulse responses obtained from a conference room of approximate dimensions 6 meters by 16 meters by 4 meters in height. Standard fiberglass tiles covered the ceiling, in a framework of metal support “T” bars. The floor was covered with carpeting and the walls were standard painted drywall. A large conference table was centered in the room and padded high backed chairs were positioned around the table (20 chairs). The impulse response of the room was obtained by playing white noise from a conference phone (Nortel VISTA 350 speaker phone) and recorded to DAT machine (TEAC model DA-P20). An NLMS algorithm was allowed to adapt for 20 minutes using a 1024 tap filter. The converged filter taps were used as the impulse response of the room.