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

Université d'Ottawa · University of Ottawa

ALGORITHMS FOR ACOUSTIC ECHO CANCELLATION IN THE PRESENCE OF DOUBLE TALK

by

Timothy J. Creasy, P.Eng.

B.A.Sc., University of Waterloo, 1992

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

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Ottawa, Ontario, Canada
November 1999

Abstract

Acoustic echo cancellation is an attractive way to enhance speech quality and provide a full-duplex communication channel for hands-free telephony. Among the many implementation challenges, the problem of "double talk" is considered one of the toughest. When near-end speech and the acoustic echo of the far-end speech are simultaneously present in the microphone signal, it is very difficult to separate these two components and provide satisfactory echo attenuation.

After providing a thorough overview of the application and the major adaptive filtering algorithms used in echo cancellation, this thesis presents an in-depth study of the published techniques for dealing with double talk. We classify these existing approaches according to their underlying methodology, such as signal level comparison, signal correlation, weight progression monitoring, and dual filter architecture. The strengths and weaknesses of a number of representative algorithms are discussed in some detail. We argue that detection of echo path changes yields a more robust echo canceller than direct double talk detection.

The major contribution of this thesis is the development of a novel variable step size approach that uses a measure of the correlation between successive gradient estimates to control the system's adaptation rate. Two main algorithms, denoted GC-VSS3 and PC-VSS, are derived by applying this approach to the Least Mean Squares and Affine Projection

algorithms, respectively. We demonstrate the superior performance of the proposed algorithms in a double talk environment. This is done through a series of simulation experiments using white noise, coloured noise, and recorded speech as the input signals. We also show how an auxiliary double talk detector, based on the proposed variable step size technique, may be employed to control the peripheral echo suppression components that are usually required in a hands-free telephone set. Finally, computational complexity is analyzed and shown to be reasonable for practical implementation when certain beneficial properties of the adaptive filter structure are exploited.

Acknowledgments

My sincere thanks are due to my research supervisor, Dr. Tyseer Aboulnasr, for providing such insightful and constructive advice, and especially for her constant encouragement in seeing this work to its completion.

I am grateful to the Natural Sciences and Engineering Research Council (NSERC) and the University of Ottawa for their generous financial support of my graduate studies program in the form of several scholarships.

I wish to acknowledge Heping Ding and his colleagues in the Advanced Terminals group at Nortel Networks for inspiring the topic of this thesis, and for alerting me to the existence of the Rohrs-Younce patent [Roh90], which contains some parallels with my independent approach to the problem at hand.

My deep appreciation is expressed to my wonderful family and friends, who have been so kind and supportive in many different ways.

I am forever indebted to my dear wife, who has shown enormous patience and made many personal sacrifices to support me throughout this exercise. Thank you from the bottom of my heart, Sondra, for your unwavering love and understanding.

Table of Contents

List of Acronyms

AEC	Acoustic echo cancellation
AP	Affine projection
codec	Coder/decoder to convert signals between analog and digital domains
CPU	Central processing unit
DC	Direct current (i.e. 0 Hz)
DSP	Digital signal processor / processing
DT	Double talk
DTD	Double talk detection / detector
EEC	Electrical echo cancellation
EERLE	Excess echo return loss enhancement
EPC	Echo path change
ERL	Echo return loss
ERLE	Echo return loss enhancement
FAP	Fast affine projection
FFT	Fast Fourier transform
FIR	Finite impulse response
FTF	Fast transversal filter
GC-VSS	Gradient correlation – variable step size
IIR	Infinite impulse response
ITU	International Telecommunication Union
LEC	Line echo cancellation
LMS	Least mean square
LS	Least squares
MAC	Multiply-accumulate operation
MIPS	Million instructions per second

MMSE	Minimum mean-squared error
MOS	Mean opinion score
MSE	Mean-squared error
NEC	Network echo cancellation
NLMS	Normalized least mean square
NLP	Non-linear processor
PC-VSS	Projection correlation – variable step size
PCS	Personal communications system / service
RLS	Recursive least squares
RMS	Root mean square
ST	Single talk
TIP/TP	Total impulse power to tail power ratio
VSS	Variable step size

List of Symbols

The convention followed in this thesis is to use boldface lower case for vectors and boldface upper case for matrices. Scalar quantities are indicated by italics.

B	block size in block algorithms
$c(n)$	correlation value of instantaneous and average gradient estimate vectors
$\bar{c}(n)$	time-averaged value of $c(n)$
$cc(n)$	correlation coefficient of instantaneous and average gradient estimate vectors
$d(n)$	primary signal, containing the echo, near-end talker's speech, and background noise picked up by the microphone
$\mathbf{d}(n)$	vector of recent primary input samples, used in AP algorithms
$e(n)$	estimation error; also the signal to be sent to the far end
$\mathbf{e}(n)$	estimation error vector produced by AP algorithms
$E\{\cdot\}$	statistical expectation operator
$\hat{E}\{\cdot\}$	estimate of expectation obtained via time averaging
f_s	system sampling rate
$\mathbf{g}(n)$	negative gradient estimate
$\bar{\mathbf{g}}(n)$	time-average of the negative gradient estimate
$\tilde{\mathbf{g}}(n)$	short-term average gradient
$\{h_i\}$	impulse response of the acoustic echo path
$\mathbf{h}(n)$	impulse response vector
$\mathbf{h}'(n)$	impulse response vector after an echo path change
$\mathbf{I}$	identity matrix
J_{MSE}	mean-squared error cost function

J_{LS}	least squares cost function
k	block index
K	block size for averaging the gradient correlation $c(n)$
L	length of the acoustic echo path in samples
m_0	number of consecutive sign changes to decrease step size in Harris algorithm
m_1	number of consecutive identical signs to increase step size in Harris algorithm
M	window size for time-averaged estimates
n	discrete time index
N	number of taps or weights in the adaptive filter
$O(\cdot)$	operator indicating order of computational complexity
$p(n)$	time-averaged correlation estimate, an intermediate variable in GC-VSS3
$p_x(n)$	power in the tap-input vector
P	projection order
$q(n)$	generic measure on which to base a variable-step-size algorithm
R	consecutive time intervals over which transfer criteria must be satisfied in Ochiai algorithm
$s(n)$	sign of gradient correlation
$\bar{s}(n)$	time-averaged value of $s(n)$
T	system sampling period
T_C	period of checking transfer conditions in Ochiai algorithm
T_D	one-way end-to-end delay in the telephone network
T_{HANG}	hangover time for double talk detection
T_{HOLD}	hold-off time for double talk detection
T_{IC}	time for initial convergence to a specified level of echo cancellation
T_{RDT}	time for recovery after double talk
T_{RPV}	time for recovery after an echo path variation
$u(n)$	near-end speech signal
$v(n)$	ambient noise component of the microphone signal at the near end
$\{w_i\}$	adaptive filter's weights or coefficients
$\mathbf{w}(n)$	weight vector
$\Delta\mathbf{w}(n)$	weight update vector
$\mathbf{w}^*$	optimum weight vector
$x(n)$	reference signal, containing speech and noise from the far end

$\mathbf{x}(n)$	tap-input vector
$\mathbf{X}(n)$	excitation signal matrix, used in AP algorithms
$y(n)$	echo signal
$\hat{y}(n)$	adaptive filter output, providing an estimate of the echo signal
z	unit sample advance operator in z-transform domain
α	constant smoothing parameter
β	scaling factor, used in different ways in various algorithms
γ	scaling factor, used in different ways in various algorithms
Γ	threshold value (subscripts indicate specific thresholds for various algorithms)
δ	regularization parameter used in power-normalized algorithms
$\boldsymbol{\varepsilon}(n)$	normalized residual echo vector, in AP algorithms
$\eta(n)$	white noise sequence
λ	exponential forgetting factor
$\{\lambda_i\}$	eigenvalues of the input autocorrelation matrix
λ_{max}	the largest eigenvalue in the input autocorrelation matrix
μ	fixed step size
$\mu(n)$	variable step size
μ_{min}	lower limit for variable step size
μ_{max}	upper limit for variable step size
τ_d	time dispersion of the echo
$\boldsymbol{\Phi}(n)$	time-averaged autocorrelation matrix of the reference input
$\chi_b(n)$	autocorrelation of the reference input for the b^{th} lag
$\boldsymbol{\Omega}(n)$	weight error
∂	partial derivative operator
∇	gradient operator

1 INTRODUCTION

A journey of a thousand miles begins with a single step.
— Chinese proverb

As we approach the new millennium, one prediction we can make with confidence is that personal communications technology will continue to become more pervasive and usable than ever. The vision of the telecommunications industry is that people will be able to exchange high-quality voice and high-speed data messages with ease — any time, anywhere [Rab95]. The end-user devices or "terminals" that enable this will have two important characteristics: they will be untethered and hands-free. We have already witnessed the beginning of this revolution with the explosion in the use of mobile phones and speaker phones during the past decade. The next generation of terminals will be even more portable and powerful, driven by advances in semiconductors and signal processing. Indeed, the advent of a *Star Trek*-like chest-pin communicator for mass consumer use may not be very distant.

In the interim, several outstanding technical challenges must be addressed. One area of intense study is acoustic echo cancellation (AEC). This is a signal processing technique that attempts to remove the undesirable local echo picked up by the microphone in a hands-

free terminal. Successful application of AEC can lead to a great improvement in the quality of voice communications, allowing more natural two-way conversations to take place over a speaker phone.

Much of the research into acoustic echo cancellation today involves improving on the way current algorithms behave in their normal mode of operation — that is, with speech coming only from the far end of the telephone line. The aim is to obtain ever better performance during single talk, meaning a higher degree of echo cancellation and faster response to changes in the acoustic environment.

In this thesis, we take a different tack, focussing instead on AEC operation in the presence of double talk. When the local talker's speech is present in conjunction with the acoustic echo of the far-end signal, it becomes very difficult to cancel the echo portion alone, so performance is generally degraded. This is a very practical problem that has not yet been satisfactorily resolved, despite various attempts. The aim of our research has been to investigate the existing techniques for dealing with double talk, then try to use this insight to arrive at an improved solution.

The outline of the remainder of this thesis is as follows:

- Chapter 2 describes in detail the subject of acoustic echo cancellation for hands-free telephony, including the problem of double talk. This provides the motivation for the balance of the thesis.
- Chapter 3 reviews the theory of adaptive filtering, to provide a foundation for the results and analysis that follow.

- Chapter 4 is an extensive survey of the published body of research that attempts to address the double talk problem. The limitations of these existing techniques are explained, highlighting the need for a more robust solution.
- Chapter 5 presents a novel approach for dealing with double talk: a variable-step-size adaptive filtering algorithm based on gradient correlation, GC-VSS. Its good performance in a number of different operating environments is demonstrated via simulation. For improved convergence speed with coloured input signals, the variable step size approach is combined with the Affine Projection, resulting in a projection correlation algorithm denoted PC-VSS. Furthermore, an auxiliary double talk detection technique is introduced to control the residual echo suppression circuitry. Lastly, the computational complexity of the proposed algorithms is analyzed and shown to be reasonable for real-time implementation.
- Chapter 6 provides the overall conclusions of the thesis, summarizes its contributions, and offers some recommendations for future research work in this area.

2 THE APPLICATION: HANDS-FREE TELEPHONY

> *Providing means for a comfortable hands-free telephone conversation is a long-standing problem... It may be considered as one of the most challenging problems currently under consideration in digital signal processing.*
> — Hänsler (1992)

Several examples of hands-free telephony are quite commonplace today. Many desktop phones in an office environment have a loudspeaking capability. Corporate conference rooms are often equipped with more sophisticated (and more expensive) versions of these speakerphones. At the high end of this spectrum comes video conference equipment. All of these communication devices are distinguished from conventional telephones by the fact that the loudspeaker and microphone are located at some distance from the user's ear and mouth, respectively. These critical components are together called *electroacoustic transducers*; that is, they convert between electrical and acoustic signals. It should be noted that some hands-free terminals incorporate multiple loudspeakers, multiple microphones, or both. For example, video conference systems often provide a stereophonic communications channel [Son95, Ben97]. In this thesis, we restrict ourselves to the single loudspeaker, single microphone case.

One obvious candidate for hands-free telephony is the car phone. Today, too many people try to drive with one hand on the steering wheel, while clutching a cell phone to their ear in the other. The popular media has found a hot issue in the dangers of "dialling and driving" [Pri95]. Road safety would no doubt be improved if people could carry on a conversation while keeping their hands on the steering wheel and their attention on the traffic around them. To achieve this, the phone's receiver circuit could possibly be connected into the car's existing loudspeaker system, while a microphone array could be built into the dashboard. Extending the hands-free paradigm a little further, the operations of dialling outgoing calls and picking up incoming ones could be activated by voice commands. The automobile is a challenging environment for hands-free telephony, in part due to the highly reverberant enclosure and ambient noise level [Cha97]. Nevertheless, our vehicles are destined to see widespread application of this technology in the near future.

2.1 Acoustic Echoes

An *echo* is the distorted and delayed version of a sound or signal, reflected back to its source. Acoustic echoes are the bane of hands-free telephony. They result from the acoustic coupling between the loudspeaker and microphone in a hands-free terminal.

To explain this phenomenon in more detail, consider the scenario shown in Figure 2.1. Neil is having a point-to-point telephone conversation with Farrah. Neil is using a very basic desktop speakerphone in his office. Farrah is holding a regular handset to her ear. Between them is the telephone network, which provides a delay T_D in each direction. When Farrah speaks, her voice is broadcast from the loudspeaker in Neil's phone into his office space. The

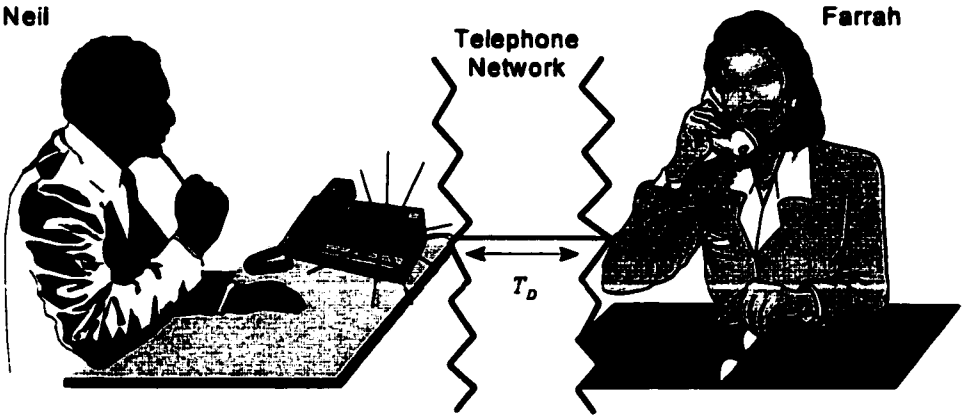

Figure 2.1 Example of a telephone conversation employing a hands-free phone at one end. Neil is at the near end, while Farrah is at the far end.

sound waves bounce off the walls and other obstacles, and some of the energy returns to the microphone, where it is amplified and transmitted back to Farrah. She perceives this as an annoying echo, with a round-trip delay of $2T_D$.

On the other hand, Neil does not notice such an effect. Farrah's handset has minimal acoustic coupling between the receiver at her ear and the mouthpiece, typically -45 dB [Bre99]. Furthermore, less amplification is required in a conventional phone, due to the proximity of the transducers to the ear and mouth. Thus the feedback path to Neil is negligible. It may seem ironic that it is Farrah who has to endure degraded reception quality due to the limitations of Neil's hands-free phone. But in reality both users will be affected if the natural flow of conversation is disrupted. Ultimately their business relationship may suffer as a result.

The amount of acoustic echo picked up by the hands-free terminal's microphone depends on a number of factors:

- the size and dimensions of the room;
- the position of the electroacoustic transducers with respect to the walls, ceiling, floor, and objects in the room;
- the degree to which sound waves reflect off these various surfaces, which depends on their shape and material composition:
- the volume setting of the loudspeaker, which is usually adjustable; and
- the amount of direct coupling between the loudspeaker and microphone within the speakerphone housing.

The *perception* of the echo by the far-end user depends on the additional factor of the round-trip delay. If the delay were zero, the echo would sound about the same to Farrah as if she were physically present in Neil's office. This natural type of echo is known as *sidetone,* and is actually beneficial — without it, deaf people often have difficulty speaking clearly. But subjective listening studies have demonstrated that as the echo delay increases, the perceived voice quality degrades, assuming a fixed echo level [Yas91]. This delay can come from several sources, both within the network and the terminal equipment, as summarized in Table 2.1. Any call routed via geostationary satellite experiences a significant delay due to the long propagation distance. In mobile phone systems, some delay is inherent in the air interface protocol so as to provide time diversity over the fading radio channel; low bit rate speech coders introduce additional delay. In video conferencing, the audio is intentionally delayed to synchronize with the video signal, which experiences delays due to video coding. A delay of less than 10 ms is almost negligible and requires much less echo suppression than one of 500 ms.

Delay source	Round-trip delay (ms)
geostationary satellite link	500 - 600
undersea cable circuit	50 - 150
terrestrial switching equipment	10 - 80
cellular/PCS baseband processing	80 - 180
video coding	~ 800

Table 2.1 Typical delays present in telephone communications [Son80, Lew93, Gil94].

2.2 Acoustic Echo Control

The natural first step in acoustic echo control is to reduce the amount of acoustic coupling between the loudspeaker and microphone. In a high-end audio- or video-conferencing system, these can be separate components, spaced metres apart. This luxury cannot be afforded in a desktop speakerphone, where for practical reasons the distance between transducers is limited by the size of the set. But other passive measures can be taken in the design of the enclosure. The materials (usually plastics), dimensions, and overall structure affect the way that acoustic waves propagate, so consideration must be given to these factors. One solution may be to insert sound insulation into the housing. Vibration due to resonances within the set is also a key concern [Bir95]. Damping mechanisms for moving parts like the touch-tone keys may be needed to stop them from rattling.

Then more sophisticated transducers and arrays can be considered. For example, instead of an omnidirectional microphone, a directional microphone can be used. The loudspeaker

can be strategically placed at the null of the directional microphone's reception pattern [Bau95]. Again, the feasibility and extent of this type of approach will vary with the application.

Once the acoustic aspects have been adequately dealt with, it is possible to consider applying electronic techniques to enhance the echo control [Han92, Han94]. The most crude technique is to insert switches in the send and receive paths and activate only one path at a time. This provides *half-duplex* communication, and is generally unsatisfactory. An enhancement on this is to use variable losses instead of switches, and adjust the attenuation in proportion to the echo level [Lew93]. The current generation of inexpensive speakerphones ordinarily employ one of these basic approaches.

2.3 Acoustic Echo Cancellation

A more sophisticated technique is to attempt to *cancel* the echo, rather than just suppress it. This leads to the possibility of true full-duplex communication, which permits more natural conversations. The basic principle of acoustic echo cancellation (AEC) is to generate a replica of the actual echo and subtract it from the microphone signal. Echo estimation is generally accomplished by using an adaptive filter to model the impulse response of the acoustic echo path. The filter weights are adjusted by an adaptive algorithm that minimizes the estimation error in some way. We will defer discussing the details about such algorithms until Chapter 3.

Figure 2.2 shows the AEC configuration for a typical desktop speakerphone. The notation of the diagram will be maintained throughout this thesis and is summarized below:

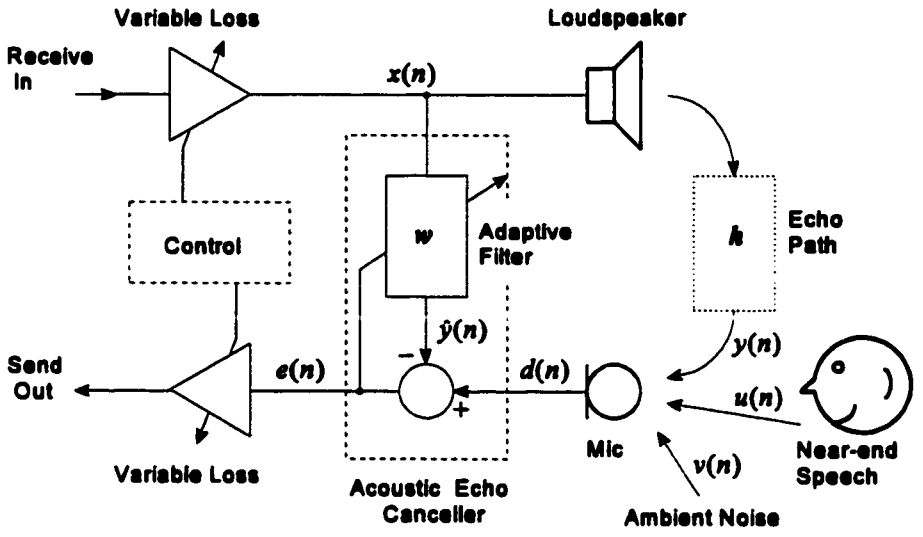

Figure 2.2 Typical configuration of an acoustic echo canceller (AEC).

$\{h\}$ the impulse response of the acoustic echo path;

$\{w\}$ the adaptive filter's weights or coefficients;

$x(n)$ the *reference* signal, containing speech and noise from the far end;

$y(n)$ the echo signal;

$u(n)$ the near-end speech signal;

$v(n)$ the ambient noise at the near end;

$d(n)$ the *primary* signal, containing the echo, near-end talker's speech, and background noise picked up by the microphone;

$\hat{y}(n)$ the estimated echo signal generated by the adaptive filter;

$e(n)$ the estimation error, also the signal to be sent to the far end.

Note that AEC alone is often unable to reduce the echo to an acceptable level. Therefore most designs incorporate some variable losses in the send and receive paths to further

attenuate the residual echo. This is vital to maintain the stability of the feedback loop, preventing an undesirable effect called *howling*. Another common approach is to apply *centre clipping*, also known as a non-linear processor (NLP), in the send path. This unit effectively cuts out small amplitude signals while passing stronger ones, using a non-linear transfer function [Mur90, Mak97].

The configuration shown in Figure 2.2 is not unique to this application. Indeed, the concept has been borrowed from another type of echo canceller used in telephone communications. It is therefore prudent to consider now the electrical echo canceller (EEC) and see how knowledge gained from its development might be applied to AEC.

2.4 Comparison with Electrical Echo Cancellation

The first experimental echo cancellers were designed in the late 1960s [Son66, Son67], to permit full-duplex voice communication over long distances, especially over satellite links. Surprisingly, it took more than a decade for them to see commercial application, partly due to implementation cost [Wei77]. They are now commonly used on all long distance connections. Figure 2.3 shows the basic configuration of an EEC as used in the telephone network. A *hybrid* transformer is used at the connection point between the two-wire subscriber loop and a four-wire long-haul link. A fixed impedance is used to balance the hybrid, but any mismatch results in electrical echoes being reflected back into the four-wire path. The EEC attempts to cancel such echoes. Depending on the point in the system where they are deployed, EECs are sometimes referred to as network echo cancellers or line echo cancellers.

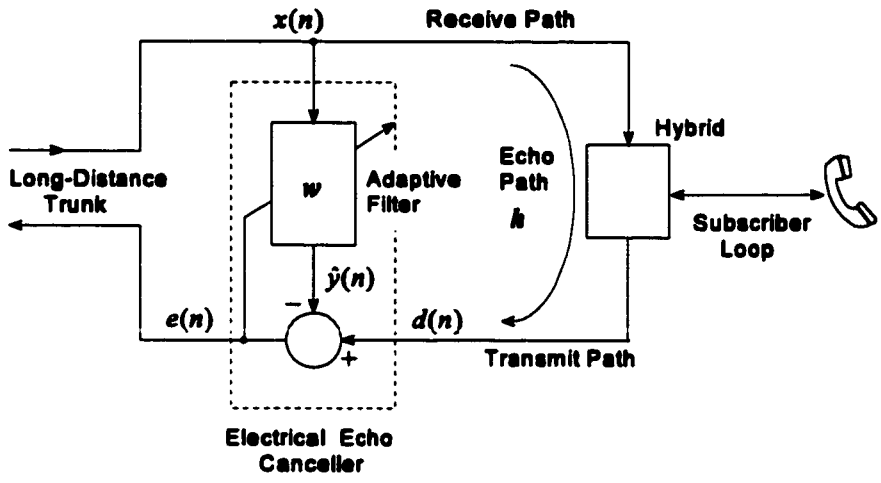

Figure 2.3 Basic configuration of an electrical echo canceller.

AEC is clearly very similar to EEC at a block-diagram level. However, there are some significant differences in the nature of acoustic and electrical echo paths that make AEC much more challenging in practice:

- a larger number of adaptive weights is required to form an adequate model;
- the echo level before cancellation is generally much higher; and
- the impulse response varies more dramatically over the duration of a call.

Each of these issues is considered in more detail in the following paragraphs.

The adaptive filter length required to fully model an echo path may be determined as the product of the time dispersion of the echo and the system sampling rate: $L = \tau_d \cdot f_s$. The standard sampling rate used in telephony is 8 kHz. For EEC, the dispersive part of the impulse response from a single hybrid lasts about 6 ms, in which case 48 filter weights would be required. In practice, this may need to be increased to a few hundred weights to account for additional bulk delay in the network. A much larger number of weights is required to model an acoustic echo path, typically $N \geq 1000$. This may be attributed to the fact that sound

waves propagate much more slowly than electrical signals. The speed of sound in air is about 343 m/s[1]. As an example, a wall located 10 m from the speakerphone will produce a direct echo with a delay of 58 ms. Multiple reflections off this wall and other objects can produce much longer delays. The impulse response generally has an exponentially decaying envelope, since these reflections die out over time [Mak93].

In EEC, there is a guaranteed minimum 6 dB of attenuation through the passive hybrid. More typical values for the *echo return loss* are 11 to 15 dB [Gou70, Gri84]. However, in AEC the magnitude of the echo $y(n)$ is usually larger than the desired near-end signal $u(n)$, due to the proximity of the loudspeaker to the microphone. In this case, there is usually a *gain* of 12 dB or more [Gil94] between receive out and send in paths. This is possible due to the presence of amplifiers in the echo path. Therefore, there is a greater potential for instability of the loop, and a larger amount of echo cancellation is required.

In EEC, the echo path is often different from one telephone call to the next, due to the different impedances of the subscriber loops, but changes very little during the course of a call. In AEC, variations in the room impulse response can be significant and must be tracked for the duration of the call, not just when the connection is first established.

Other differences include the degree of loss-switching, and the level and type of howling protection [ITU97].

---

[1] measured in dry air at 20°C and a pressure of 1.0 atm [Hal86]

2.5 Limitations of AEC

A number of factors have been identified as limiting the theoretical performance of acoustic echo cancellers.

Undermodelling exists when the number of parameters in the adaptive filter is less than that of the actual impulse response. This means there are certain modes of the system that are not represented by the model; thus it will not be possible to cancel the response of these modes. For example, in using a Finite Impulse Response (FIR) adaptive filter the *tail* of the actual response, or portion after the last FIR tap, is not modelled. The achievable cancellation has been shown to be a ratio of the total impulse power to the tail power (TIP/TP) [Kna94].

Interestingly, **overmodelling** can also degrade performance. This is partly due to the fact that any excess parameters will not converge to exactly zero, so will contribute to some error in the adaptive filter output. Furthermore, the convergence time of most adaptive algorithms is proportional to the number of parameters, so by using too many we will slow down convergence and impair tracking capability. Selection of an appropriate number of adaptive parameters is critical.

The **dynamic** nature of the acoustic echo path limits cancellation ability. As people move themselves and other objects around the room, the impulse response will change, often dramatically. Just as it takes some time for the adaptive filter to converge on an initial solution, there will be some lag in tracking the echo path variations. Thus the dynamic performance of the echo canceller, in terms of reconvergence speed and tracking error, becomes an important design criterion [Yua94].

Nonlinearity of the acoustic echo path presents another problem. The major source is the loudspeaker, specifically the nonlinear relationship between force and displacement in the cone suspension system. This soft clipping manifests itself as odd harmonics that may represent up to 10 percent of the output level of a small loudspeaker operating at high volume [Bir95]. Possible solutions include pre-distorting the loudspeaker signal and using neural networks in the echo canceller [Bir96]. Under very large signal conditions, hard clipping may occur due to overloading of amplifiers or codecs.

Noise in the input signals to the AEC will also degrade performance. This is generally dominated by the background acoustic noise picked up by the microphone along with the echo. The primary sources are fans in the ventilation system or nearby equipment, hum from fluorescent lighting, and chatter from adjoining work areas. While usually much less significant, electronic noise from amplifiers, quantization noise from codecs, and finite precision arithmetic can be lumped into this category. It is interesting to note that if the near-end ambient noise is high, the talker will naturally increase his speech level so that the speech signal is always louder than the noise [Hei93]. *Active noise cancellation* may be employed to improve the signal-to-noise ratio [Wid75, Ell93], but this subject is beyond the scope of this thesis.

In the next section, we consider another critical source of interference: the speech of the near-end user.

2.6 Double Talk

The speech signals transmitted over a telephone network are almost always part of a conversation. It is a characteristic of human dialogue that once in a while both people speak simultaneously. This is referred to as "double talking" [Gou70], or now more commonly just *double talk.*

Evidence that it occurs is provided by the observation that people find half-duplex communications quite unnatural. For example, when using a walkie-talkie, it is standard practice to add the word "Over" at the end of a transmission to indicate that it is the other user's turn to speak. It is essential for comprehension to enforce single talk in this manner, but it tends to chop up the conversation. This is not desirable on a telephone call, where people are accustomed to having a full-duplex connection.

Double talk is especially noticeable when the conversing parties are having a heated discussion or argument. But it can also take place amidst longer periods of single talk when the current listener interjects a comment or sound of acknowledgment, or at the overlap between the two sides of the dialogue. The frequency of double talk occurrence in a given conversation depends on many factors, including the social relationship and ethnic backgrounds of the participants [Lew93]. One study of telephone conversations showed that when one person is speaking (i.e. a single talk situation), 20% of the time the other party interrupts before the first one is finished, creating double talk [Bra68].

In a nutshell, the problem with double talk for an AEC system is that the near-end speech interferes with the adaptation process. It gets mixed in with the echo, and thus acts like a measurement noise at the primary input. In the absence of intervention, this leads to

degradation of the echo path model, so more echo will pass through uncancelled. Moreover, it will take some time for the adaptive filter to re-tune itself once the double talk stops.

The usual way to try to solve this is to add another signal processing block called a *double talk detector* (DTD). The need for a reliable double talk detector in AEC is actually two-fold:

- to freeze the filter weights to prevent them from being corrupted, and
- to set the attenuation levels of the variable losses in the send and receive signal paths [Mee98].

We now make a key observation. Any significant increase in the residual error after echo cancellation could be due to either double talk or a change in the echo path. In the former case the best thing to do would be to freeze the filter weights at their values just before the near-end speech started, whereas in the latter it is desirable to track the change as quickly as possible [Car96]. These are conflicting outcomes, so we can see that the error power alone is not a useful criterion for double talk detection.

Another point to consider is that AEC algorithms with rapid convergence will generally exhibit comparably fast divergence at the onset of double talk. Therefore, as the single-talk performance of echo cancellers is improved through more and more sophisticated techniques, the problem of detecting double talk, or at least mitigating its effect, becomes more critical and more challenging.

A presentation and discussion of existing techniques for dealing with double talk is the subject of Chapter 4.

2.7 Performance Evaluation

Once an algorithm for AEC has been designed and implemented, we need some way to evaluate its performance. In other words we want to determine how well it achieves its objective: making duplex hands-free communication clear and echo-free to the far-end listener.

The most direct way to evaluate performance is to ask the far-end user. Ultimately this is the person who has to endure any speech degradation produced by the system. Subjective testing can take into account psychoacoustic effects such as the perception of echo that are not otherwise directly evident. Typically a *mean opinion score* (MOS) is obtained from a live test [Son80]. Participants are asked to rate communications quality on a scale of 1 to 5, where 1 means poor and 5 means excellent. Then raw results are averaged. However, such tests have their drawbacks: they are expensive to conduct, their results are hard to quantify, and they may not be very repeatable.

Some automated means of testing that produces consistent, quantifiable results is desirable to enable direct comparisons between different algorithms or implementations [Nay94]. Ideally there would be a way to correlate such objective test results with the subjective ones, but this remains an area of ongoing research [Gil94, ITU97].

The two most significant objective performance measures in evaluating an AEC system are (i) the amount of echo cancellation achievable in the steady state, after convergence; and (ii) the convergence time required to reach this steady state. Thus the principal performance measures we will use in the balance of this thesis are as follows:

- Echo Return Loss Enhancement (ERLE), the attenuation of the echo signal due to the acoustic echo canceller:

$$ERLE = 10 \log \frac{\hat{E}\{d^2(n)\}}{\hat{E}\{e^2(n)\}} \quad \text{(dB)} \tag{2.1}$$

where the operator $\hat{E}\{\cdot\}$ represents an estimate of the expected value, obtained via a simple time averaging process.

- T_{IC}: the initial convergence time required to achieve a pre-defined ERLE level.
- T_{RDT}: the recovery time to achieve the same ERLE after a double talk period.
- T_{RPV}: the recovery time to achieve the same ERLE after an echo path variation.

Additionally, there are two useful performance measures that may be made in a simulation environment, where we have full control and knowledge of the actual echo path response and the additive noise. These are not readily applicable in real-time measurements of an AEC implementation.

- Excess Echo Return Loss Enhancement (EERLE), i.e. the ERLE that would exist if the near-end speech $u(n)$ and background noise $v(n)$ were not present in the microphone signal. During double talk, this is a more useful performance measure than ERLE because it is a ratio of the actual echo to the residual echo after cancellation:

$$EERLE = 10 \log \frac{\hat{E}\{[d(n) - u(n) - v(n)]^2\}}{\hat{E}\{[e(n) - u(n) - v(n)]^2\}} \quad \text{(dB)} \tag{2.2}$$

- Weight error, defined as the normalized Euclidean distance between the current weight vector and the actual impulse response (truncated to N samples):

$$\Omega(n) = 20 \log \frac{\| \mathbf{w}(n) - \mathbf{h}(n) \|}{\| \mathbf{h}(n) \|} \quad \text{(dB)} \tag{2.3}$$

The weight error gives an indication of the degree of echo cancellation achievable, behaving much like the inverse of the EERLE when the input signal has a flat frequency spectrum. However, the similarity is not so evident when a speech signal is used. The spectrum of a short portion of speech, especially a vowel sound, has very distinct concentrations of frequencies. If the AEC models the echo path especially well in these frequency regions, the EERLE will be high, even if the overall weight error is large due to poor modelling out-of-band. Therefore this measure must be used with caution.

ITU Recommendation G.167 specifies target values for objective AEC performance [ITU93b]. Instead of ERLE, a related quantity known as terminal coupling loss (TCL) is quoted. This is defined as the overall attenuation around the loop, from the receive-in port to the send-out port, so it includes the inherent ERL of the echo path, the ERLE, and any additional contribution from the variables losses and NLP. The requirement is for a TCL of 45 dB for single talk and 30 dB during double talk. The corresponding ERLE depends on the other losses. Clearly the better the ERLE, the less additional loss is needed to achieve the specification. Among other things, G.167 also requires that 20 dB of cancellation be achieved within one second after any of the following events: (i) the start of operation, (ii) an echo path change, or (iii) the end of double talk. These specified values may be relaxed under certain operating conditions, such as small round-trip delay. Furthermore, it is known that background noise psychoacoustically masks echo [Gil94], so the amount of echo cancellation necessary in a noisy environment like a motor vehicle is less than in a conference room.

2.8 Implementation Issues

Signal processing functions such as echo cancellers may be implemented in dedicated hardware; alternatively, they can be programmed into a general-purpose digital signal processor (DSP) via software. Ultimately the approach chosen will depend on cost, which itself depends on such factors as flexibility of customization, production volume, and time to market. Examples of both types of implementation are provided in Table 2.2.

Year	Description	# Weights	Notes	Reference
1988	Sakai *et al* rack-mounted AEC unit	2000	uses 22 DSP chips (MB86232)	[Mur90]
1996	OKI Semiconductor MSM7602 chip	216	cascadable; for EEC or AEC; 30dB ERLE (typ.)	[Sar96]
1996	Texas Instruments TMS320C8x software	800	28% CPU loading at 50 MHZ	[Qi96]
1998	Dialogic Corp. EEC in Motorola DSP56K software	128	4% CPU loading at 80 MHZ	[Gup98]
1998	Texas Instruments TMS320C6x software	512	3% CPU loading at 200 MHZ	[Zha98]
1999	Lucent Technologies T8534 chip	64	4 channel EEC; $2.29 per channel	[Luc99]

Table 2.2 Representative echo canceller implementations in either dedicated hardware or DSP software.

This table clearly shows how continuing advances in microelectronics have improved the ability to realize practical AEC systems. A decade ago, one cumbersome prototype required 22 DSP chips to implement an AEC. The current state-of-the-art enables a single

powerful (but still expensive) DSP to implement multiple such functions in parallel, possibly for multi-channel operation. Today's inexpensive DSPs can handle a single channel adequately, and are thus more likely choices for use in a desktop speakerphone [Bre99].

Programmable DSP chips are designed to perform multiplications very efficiently, often one per clock cycle. Usually they can do an accumulation of the product at no extra cost — jointly called a multiply-accumulate (MAC) operation. New processor architectures such as the TMS320C6x make no distinction in the order of the addition and multiplication, since they may be executed in parallel using distinct arithmetic units. Divisions may be performed iteratively over several cycles, with the execution time directly related to the required accuracy of the result. Ultimately, the execution time or corresponding count of million-instructions-per-second (MIPS) required for a given algorithm is very processor dependent. To achieve optimal performance, algorithms must often be tailored to a particular DSP. In general, AEC implementations are limited by the number of multiplications required, so we use this as the primary figure of merit when comparing the computational complexity of various algorithms.

It should be noted that in a real-time implementation, the peak processing power required is a more important criterion than the average MIPS. It is desirable to have an algorithm that uses a steady amount of computation per input sample, thereby achieving the most productive use of the signal processor. In contrast, an algorithm whose processing requirements are bursty in nature is not as efficient to implement; the processor has to be more powerful to handle the bursts of activity, but will be relatively idle in between.

One way to smooth out the peaks in a bursty algorithm would be to delay the signals to be processed. But as we have already noted, the perception of any residual echo after

cancellation is sensitive to the round-trip delay. Because of this, the International Telecommunication Union has recommended limits on the amount of additional delay introduced in each of the send and receive paths by the AEC processing. The maximum processing delay for tethered hands-free telephones is specified as 2 ms [ITU93b]. This puts a constraint on the nature of the signal processing blocks inserted into the signal paths, as we will see in the next chapter.

Another important factor to consider in an AEC implementation is the amount of memory required by the algorithm. This may affect the choice of DSP chip used, and has a bearing on the cost of the overall system.

3 FUNDAMENTALS OF ADAPTIVE FILTERING

> *The use of adaptive filters does require a different mind-set on the part of the designer... Designing with adaptive filters is rather like listing good things to do when bringing up children, instead of trying to define all the qualities of good adults.*
>
> — Lewis (1993)

Adaptive filters have parameters that can change value over time. They are called for in any situation where a fixed filter is not adequate. This need arises when the characteristics of a system to be modelled are time-varying or simply not known in advance. As we have already seen, acoustic echo cancellation is a prime example. AEC is classified as a system identification problem, where the acoustic echo path is the unknown plant. Adaptive filters can also be employed in other configurations, including inverse modelling (as used for channel equalization) and prediction (e.g. speech coding) [Joh95].

This chapter presents a concise overview of adaptive filtering theory. We introduce the relevant terminology, describe the major algorithms and their variants, and highlight important results from research that spans four decades. The intent is to provide a theoretical foundation for the balance of this thesis. For a more detailed treatment, the reader is directed to one of the excellent reference works on this subject [Cla93, Hay96, Wid99, Gle99].

3.1 Structures

While analog versions are possible, adaptive filters are usually realized in discrete time using digital hardware or software. This affords the designer better control of implementation precision and therefore system behaviour. Digital realizations are also a more natural match to the most basic internal architecture for an adaptive filter, the transversal structure shown in Figure 3.1 [Opp89].

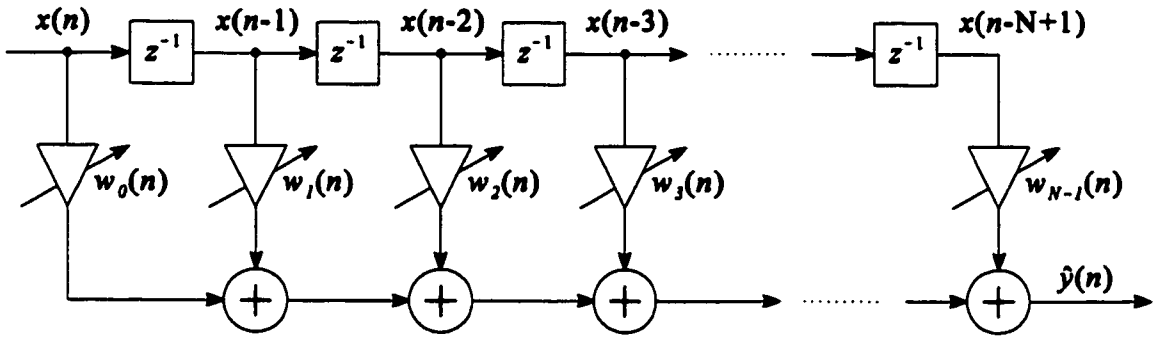

Figure 3.1 Finite Impulse Response transversal filter structure. In its adaptive form, each of the weights is variable, as indicated.

The input signal x is fed into a delay line tapped off at N equally spaced time intervals T. In a discrete-time system with sampling rate $1/T$, this translates into a unit delay between taps, denoted z^{-1}. Each tap input $x(n-i)$ is weighted by a factor w_i. Then all these are summed to form the filter output:

$$\hat{y}(n) = \sum_{i=0}^{N-1} w_i(n)\, x(n-i) = x(n) * w(n) \qquad (3.1)$$

This convolution sum shows that the output is a linear combination of the delayed inputs. The sequence of tap weights $\{w_i\}$ represents the impulse response of the filter, which is clearly

of finite duration; the index n indicates its time-varying nature. The output equation can be written more compactly in vector notation[2]:

$$\hat{y}(n) = \mathbf{w}^T(n)\,\mathbf{x}(n) \tag{3.2}$$

where $\mathbf{x}(n) = [x(n) \quad x(n-1) \quad x(n-2) \ \ldots \ x(n-N+1)]^T$ is known as the tap input vector consisting of the N latest input samples, and $\mathbf{w}(n) = [w_0(n),\ w_1(n),\ \ldots\ ,\ w_{N-1}(n)]^T$ is the weight vector.

Other adaptive filter structures exist. For example, another type of finite impulse response (FIR) structure is the lattice filter, often used in linear prediction [Hug93]. All FIR filters are inherently stable, since they do not contain a feedback mechanism; they have an all-zero transfer function. Infinite impulse response (IIR) filters have also been experimented with for echo cancellation [Shy89]. We might presume that with the introduction of a few adaptive poles in place of zeroes, a better approximation of an acoustic echo path could be generated. However, a recent paper [Lia98] has cast doubt on this notion, concluding that it takes a similar number of parameters to model an acoustic echo path, whether the transfer function of the model is rational (IIR) or polynomial (FIR). Given that IIR approaches are plagued by problems of instability, slow convergence speed, and getting trapped in local minima, the vast majority of the work on acoustic echo cancellation to date has focussed on FIR structures. We will assume a linear discrete-time transversal filter for the balance of this thesis.

---

[2] The notation $(\cdot)^T$ indicates the transposition operator.

3.2 Weight Optimization

One critical detail remains: how to determine appropriate values for the filter weights at a given time. Ultimately, we would like to achieve a performance level that is in some sense optimal. The obvious way to judge an adaptive filter's performance is by how well its output matches the desired response. The difference between the two is known as the *estimation error*:

$$e(n) = d(n) - \mathbf{w}^T(n)\,\mathbf{x}(n) \tag{3.3}$$

Clearly, it is desirable to minimize this error in some way. We cannot expect an exact match since the filter operates in an environment of random signals. But if we can define a *cost function* relating some measure of the error to the filter weights, the optimum weights will be those values that minimize this function.

The choice of a cost function is influenced by mathematical tractability. It turns out that a good candidate is the mean-squared error (MSE). We write this cost function as $J_{MSE} = E\{e^2(n)\}$, where $E\{\cdot\}$ denotes the statistical expectation operator. By substituting Eq. (3.3), we find that J_{MSE} is a quadratic function of the filter weights. This can be viewed as a surface in $N+1$ dimensions, shaped like a bowl with a unique minimum at the bottom. It helps to visualize this in three-dimensional space, as shown in Figure 3.2. The two dimensions of the horizontal plane represent two of the adaptive filter weights, while the MSE is on the vertical axis. The coordinates of the bottom of the bowl-shaped surface, $\mathbf{w}^* = (w_1^*, w_2^*)$, represent the optimum solution in the minimum MSE (MMSE) sense.

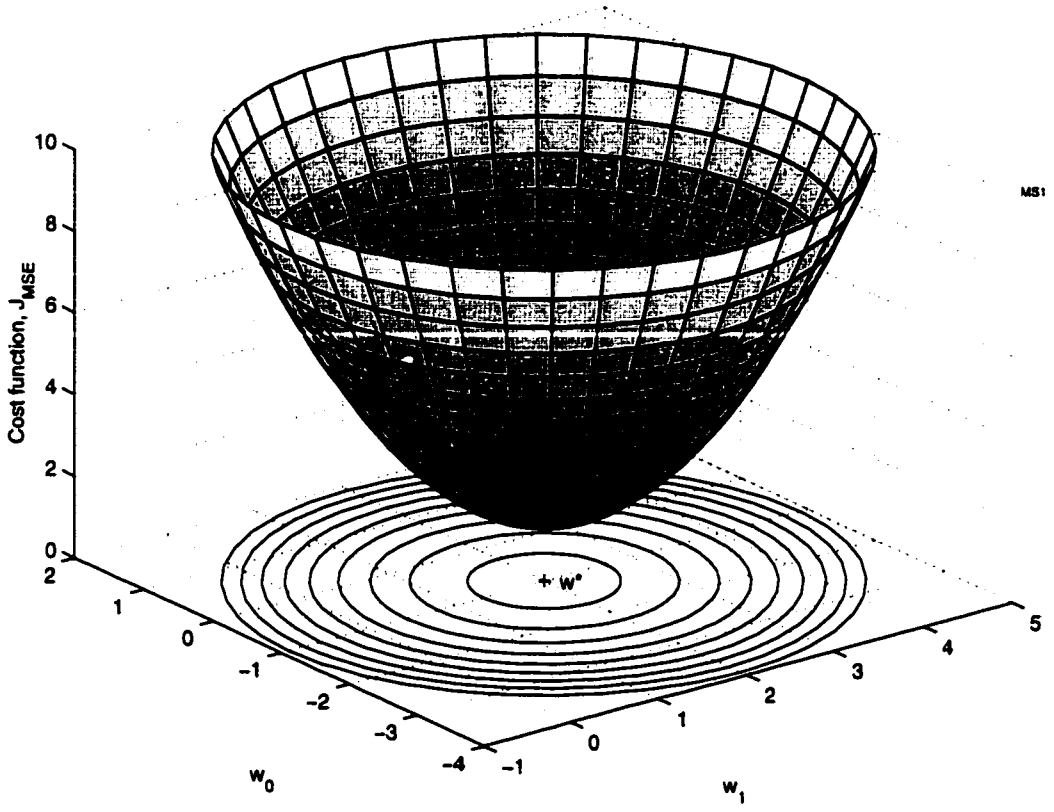

Figure 3.2 Visualization of error performance surface for $N = 2$, $\mathbf{w}^* = (-1, 2)$, and a white (spectrally flat) reference input.

Immediate calculation of the optimum would require advance knowledge of the input statistics and the response of the system being modelled. Since this is not usually available, iterative techniques are used to search the MSE surface for the minimum. Starting from an arbitrary point on the surface of the bowl, the optimum may be reached by following the path of steepest descent all the way to the bottom. In mathematical terms we determine the *gradient* of the cost function at time n, by taking its partial derivatives with respect to the individual weights. This yields [Hay96]:

$$\nabla J_{MSE}(n) = \frac{\partial J_{MSE}(n)}{\partial \mathbf{w}(n)} = -2E\{e(n)\mathbf{x}(n)\} \tag{3.4}$$

Thus the *method of steepest descent* is to apply a weight update at each iteration in the direction of the negative gradient:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \Delta\mathbf{w}(n) \tag{3.5}$$

where

$$\begin{aligned} \Delta\mathbf{w}(n) &\propto -\nabla J(n) \\ \Delta\mathbf{w}(n) &\propto E\{e(n)\mathbf{x}(n)\} \end{aligned} \tag{3.6}$$

The *principle of orthogonality* states that at the optimum (and nowhere else) the estimation error is orthogonal to the tap input samples:

$$E\{e(n)\mathbf{x}(n)\} = \mathbf{0} \tag{3.7}$$

This makes intuitive sense, because at the bottom of the bowl the gradient is zero; the weights are at their optimum, so the weight update should also vanish to the zero vector.

3.3 Least Mean Square Algorithm

The Least Mean Square (LMS) algorithm, developed by Widrow and Hoff almost 40 years ago [Wid60], has become the real workhorse of adaptive filtering. Its popularity stems from its simplicity and robustness. It may be derived directly from the steepest descent algorithm by removing the expectation operation on the gradient term. Thus the instantaneous value of the gradient estimate, which varies randomly from one step to the next, is used in the update vector. As such, the LMS is classified as a *stochastic gradient* algorithm [Hay99].

The weight update of the LMS algorithm is calculated at each iteration as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{3.8}$$

where the parameter μ is known as the *step size*. Its value is quite critical, affecting the stability, convergence speed, and steady state MSE of the algorithm.

After a large number of iterations, the weight vector $\mathbf{w}(n)$ will approach the optimum value $\mathbf{w}^*$, but will never quite reach it exactly. This is because the LMS algorithm relies on a noisy estimate of the gradient, not its exact value as in steepest descent. Thus the steady state MSE exceeds the minimum MSE by an amount called the excess MSE. *Misadjustment* is defined as the ratio of excess MSE to minimum MSE; thus it is a measure of the distance between the LMS steady-state solution and the optimal solution [Hay96]. The misadjustment is directly proportional to the step size, so to achieve the best steady-state solution, μ should be as *small* as possible. However, by taking very small steps, it will take a long time to converge. In fact, the convergence speed is roughly proportional to the step size, so it would be nice to make the step size μ as *large* as possible. These conflicting demands highlight one of the weaknesses of the LMS algorithm.

To guarantee LMS convergence in the mean square it is required that $0 < \mu < \lambda_{max}^{-1}$, where λ_{max} is the maximum of the eigenvalues $\{\lambda_i\}$, $i = 1...N$ of the input autocorrelation matrix. Furthermore, the convergence time is inversely related to the *eigenvalue spread*, the ratio of largest to smallest eigenvalues [Hay96]. Thus the fastest convergence is achieved with an input whose eigenvalues are all equal, i.e. white noise. In contrast, human speech tends to exhibit large eigenvalue spreads, leading to slower convergence.

Of all the adaptive algorithms considered here, the basic LMS is the simplest in both concept and implementation. Computational complexity is about $2N$ operations per input sample. The LMS has seen widespread application in many fields, and this has demonstrated its robust behaviour over a broad range of signal environments.

3.4 Normalized LMS

If the reference input signal level varies with time, as is the case with speech, the basic LMS algorithm is subject to a *gradient noise amplification* effect [Hay96]. This is because the weight update at each step is directly proportional to the input vector $\mathbf{x}(n)$. So when $\mathbf{x}(n)$ is large the weights jump around with large instantaneous offsets, effectively undoing any fine tuning achieved when the input was smaller.

The Normalized Least Mean Square (NLMS) algorithm overcomes this problem [Nag67, Alb67]. Here the weight update is normalized with respect to the energy in the input vector at each iteration:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu\, e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta} \tag{3.9}$$

The *regularization parameter* δ is included in the denominator to prevent the update term from blowing up when the input energy is very small. Its value is chosen to be much smaller than the typical input energy, so that it does not affect performance under normal operating conditions. The NLMS algorithm is known to be stable for step sizes that satisfy $0 < \mu < 2$ [Hsi83].

3.5 Variable Step Size LMS

Variable step size (VSS) algorithms attempt to address the tradeoff between convergence rate and misadjustment inherent in the LMS and NLMS algorithms. They do this by dynamically changing the step size μ according to the state of convergence. A large step size value is used when the adaptive filter weights are far from the optimum, to accelerate

convergence. This is gradually reduced as the weights approach the optimum, so that a lower misadjustment is eventually obtained [Eva93]. In theory, at least, they provide the best of both worlds.

Many VSS algorithms have been proposed, each differing in the way that the system state is determined and the step size is updated. Examples may be found in [Har86, Sha88, Kwo92, Mat93, Sug94, Cas95, Miy95, May95a, Hei97]. In some cases each filter weight has its own variable step size, but more often there is one time-varying step size applied to all weights.

3.6 Block LMS

Block adaptive filtering processes data in sequential blocks, rather than one sample at a time. The block LMS algorithm updates the filter weights only at the end of each block:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{b=0}^{B-1} e(kB+b)\,\mathbf{x}(kB+b) \tag{3.10}$$

where k is the block number and B is the block length. The cancellation error and the cumulative weight update term on the right side of Eq. (3.10) are still calculated at each sample using the latest block weights [Cla82]. The effect of the summation can be seen as an averaging of the instantaneous gradient vector, which improves the estimate of the true gradient as B is increased. Unfortunately, this does not mean increased convergence speed; in fact the bound on the step size is tighter for the block LMS algorithm, which may result in slower adaptation [Hay96]. The main advantage of such block algorithms is their reduced computational complexity.

A new class of *block-exact* algorithms has been identified recently. These offer significant computational savings without sacrificing performance — they produce the same error signal as their step-by-step counterparts. The downside is that they introduce in the signal path a delay equal to the block length, though this can be considerably shorter than the adaptive filter length [Gle99]. The fast exact LMS algorithm is one example of this class [Ben92].

3.7 Frequency Domain and Subband Adaptive Filtering

Filtering in the frequency domain involves multiplication, which is simpler to perform than the time-domain convolution of Eq. (3.1) [Shy92]. Dentino *et al* [Den78] have suggested collecting sequential blocks of data for both the input and desired output and performing a Fast Fourier Transform (FFT) on each. The components of the input spectrum are multiplied by independent weights, one value per spectral bin. The filter output spectrum is compared to the desired spectrum, bin by bin, and the error in each is used to update the corresponding weight via the standard LMS algorithm. To generate the time-domain output an inverse FFT is also required. Similar techniques based on other transforms [Nar83] and using more flexible structures [Pra94] have been proposed.

In subband techniques the filtering is actually performed in the time domain, but the one full-band FIR filter is replaced with several smaller filters running at a lower rate [Gil92, Rya92, Cou98, Lu99] — this is appealing for AEC where N is so large. A set of analysis filters is used to split both the input and desired signals into several frequency bands, which can then be downsampled. Adaptive filtering is performed, generating a subband error signal

for each band. These are upsampled, passed through a set of synthesis filters, and added together to produce the full-band error. Computational savings are achieved due to the decimation in time.

A consequence of splitting the input into frequency bins or bands is that the power spectral density over each narrow band is flatter or more white than over the full band [Bea95]. This will speed up convergence of an LMS-based algorithm. These techniques also permit using independent step sizes for each bin or band, normalized to the energy therein; this too leads to faster overall convergence to the MMSE solution.

However, there are some drawbacks associated with both types of frequency-dependent filtering. The block transform techniques introduce a delay in the signal path equal to the time to fill the input buffers and calculate the transforms. There is also a delay in the subband paradigm, due to the inherent group delay of the analysis and synthesis filters. We know that additional delay makes the perceived echo worse, so this works against us. This delay and the less frequent weight updates may result in degradation of tracking performance. Some researchers have introduced delayless versions of these techniques to address this [Mor95, Lar98]. Another difficulty is that at the edge of the subbands, undesirable aliasing effects may occur due to non-ideal filter responses.

3.8 Least Squares Algorithms

We now turn to Least Squares (LS) adaptive filtering, which differs from the gradient-based techniques in the optimization criterion used to adapt the filter weights. Here we minimize the cumulative squared error at each iteration [Hay96]:

$$J_{LS}(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \tag{3.11}$$

where $0 < \lambda \leq 1$ is an exponential weighting factor than emphasizes recent performance and gradually "forgets" the past estimation error; this improves tracking ability.

Least Squares algorithms use all the data acquired up to time n to calculate a new optimal weight vector on that iteration. Thus they are designated *deterministic*. The LMS step-size parameter μ is effectively replaced by $\Phi^{-1}(n)$, the inverse of the time-averaged autocorrelation matrix of the input vector $\mathbf{x}(n)$. The major advantage of LS is that the rate of convergence does not depend on the eigenvalue spread of the input autocorrelation matrix. Convergence is typically an order of magnitude faster than the LMS algorithm. Also, the steady-state MSE approaches the minimum MSE in the case $\lambda = 1$.

Direct calculation of the matrix inverse is very costly. By using an iterative approach, the Recursive Least Squares (RLS) algorithm reduces the computational complexity to $O(N^2)$ [Lju83], but this is still prohibitive for applications like AEC with large N. However, several clever enhancements have been made, such as the Fast Transversal Filter (FTF) algorithm [Cio84], which further reduce the burden to the range of $8N$ [Mak97]. This then can be regarded as an upper limit on what competing algorithms of similar or inferior performance should cost. Unfortunately, problems regarding stability have been observed for some fast RLS variants, especially in fixed-point implementations. This explains why most real-world designs today stay clear of these intricate approaches and use the more reliable LMS algorithm or its variants instead.

3.9 Affine Projection Algorithm

A generalized form of the NLMS algorithm was developed some time ago [Oze84] but received little attention until recently. Known as the affine projection (AP) algorithm, it offers an intermediate solution between the basic NLMS algorithm and Least Squares approaches. It calculates the error as the performance of the current weight vector based upon the previous P input vectors, where P is the order of the algorithm. The AP algorithm is executed as follows:

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\,\mathbf{w}(n) \tag{3.12}$$

$$\boldsymbol{\varepsilon}(n) = [\,\mathbf{X}^T(n)\mathbf{X}(n) + \delta\mathbf{I}\,]^{-1}\mathbf{e}(n) \tag{3.13}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\,\mathbf{X}(n)\,\boldsymbol{\varepsilon}(n) \tag{3.14}$$

where $\mathbf{d}(n) = [d(n)\ \ d(n-1)\ \ d(n-2)\ \ \ldots\ \ d(n-P+1)]^T$ is a P-by-1 vector of the previous primary input samples, $\mathbf{X}(n) = [\mathbf{x}(n)\ \ \mathbf{x}(n-1)\ \ \mathbf{x}(n-2)\ \ \ldots\ \ \mathbf{x}(n-P+1)]$ is the N-by-P excitation signal matrix, and δ is the regularization parameter that keeps the matrix inversion well behaved. The selection of P represents a tradeoff between cost and performance. In the special case $P = 1$, AP reduces to the NLMS algorithm, which may therefore be viewed as a first order affine projection. Significant improvement over NLMS may be achieved with P as low as 2 to 5, especially with a coloured input signal such as speech [Ree98]. In a direct implementation, the computational complexity is still relatively high: $O(2PN + 7P^2)$.

A fast AP algorithm (FAP) has been proposed [Gay95, Tan95] and is especially appealing for AEC due to its low complexity for long filters. In its stabilized form, the FAP requires $2N+30P$ multiplications per input sample. For large N and $P \le 10$, say, this

represents only a small increase in complexity over the NLMS algorithm's $2N$ operations. More recently, a block-exact FAP has been developed [Tan99].

3.10 Summary and Discussion

The catalog of adaptive filtering algorithms available to the signal processing engineer today is large and growing rapidly. Only a select few have been mentioned in this section, those most important for the application at hand. The elegantly simple LMS (or NLMS) remains the benchmark by which other algorithms are judged. But newer techniques offer the potential to overcome the slow convergence associated with the very long filter and speech input present in AEC.

4 DEALING WITH DOUBLE TALK: A SURVEY OF EXISTING TECHNIQUES

The design of a good double talk detector is difficult and much more of an art than the design of the adaptive filter itself.
— Weinstein (1977)

In section 2.6 we discussed the presence of double talk in telephone conversation and explained the problem this creates for echo cancellers. A survey of the open literature reveals an abundance of remedies for this, although the majority of these are conceived with EEC in mind. Their applicability to AEC must therefore be examined. We may classify these solutions into the following categories:

- signal level comparison
- signal correlation
- weight progression monitoring
- dual filter architectures

After a brief review of some fundamentals in section 4.1, the following four sections of this chapter are devoted to investigating these categories, one at a time. In each case we present and analyze one representative algorithm in detail, then follow up with a short description of related approaches. Finally in section 4.6 we briefly capture other techniques that do not fall into the main categories identified above.

Note that in many cases the notation used in the source has been modified here to maintain consistency throughout this thesis with the notation introduced in Figure 2.2.

4.1 Fundamentals of Double Talk Detection

As mentioned earlier, the standard way of addressing the problem at hand is to use a double talk detector. This is typically a separate signal processing block that monitors the loudspeaker and microphone signals, and possibly others. Under certain conditions the DTD will decide that double talk is present. This sends a control signal to the echo canceller algorithm to cease adaptation until the double talk condition stops. It also adjusts the amount of variable loss in the send and receive signal paths, and it controls the non-linear processor.

If the filter weights are allowed to adapt when near-end speech is present, the excess MSE will increase and the weights will diverge from the optimum. This suggests it is important to adopt a *conservative* control strategy that enables adaptation under only the most favourable signal conditions. On the other hand, fast tracking is essential to maintain a good level of echo cancellation when the echo path varies. This suggests it is important to adopt an *aggressive* control strategy that always enables adaptation except under the most unfavourable signal conditions. These contradictory requirements make the control problem a very difficult one, so some tradeoffs are inevitable.

There are two types of errors that a double talk detector can make:

- a *false hit* occurs if the DTD declares double talk when none is present;
- a *miss* occurs if the DTD does not detect double talk when it is present.

The seriousness of these errors depends on the cancellation state at the time. If a false hit occurs when the filter weights are far from the optimum the residual echo will be significant, but if the filter had already converged the user may notice no difference. Conversely a miss has more serious consequences when the filter is in a converged state, because the degradation of performance will be greater. A common way to reduce misses at the tail end of double talk is to delay adaptation for a fixed period of time after the double talk condition ends; this also prevents excessive switching in and out of double talk mode between speech syllables. This delay is known as the *hangover time*, denoted T_{HANG}. A value of around 100 ms is typically employed [Och77, Qi96]. Similarly a *hold-off time*, T_{HOLD}, may be used to delay freezing adaptation for a brief period after the initial detection of a double talk condition, thereby minimizing the number of false hits.

There are other ways of dealing with the double talk issue besides straight detection. Some researchers have tried to design a *robust* adaptive algorithm that is insensitive to double talk. These are generally variable-step-size approaches that reduce the step size in the presence of double talk so that any deviation of the filter weights from the optimum is minimal. Such techniques are also included in the survey that follows.

4.2 Signal Level Comparison

The basic double talk detection method used in electrical echo cancellers is a signal level comparison. In early EEC implementations, the instantaneous magnitude of the microphone signal is compared with the largest magnitude component of the reference input vector. Double talk is declared if:

$$|d(n)| > \frac{1}{2} \max \{ |x(n)|, |x(n-1)|, \ldots, |x(n-N+1)| \} \quad (4.1)$$

since if $d(n)$ were solely echo, it would be at least 6 dB down from the input signal due to the attenuation in the hybrid [Son80].

More sophisticated techniques generalize this comparison to use signal power estimates, rather than instantaneous values:

$$p_d(n) > \Gamma_{DT}\, p_x(n) \quad (4.2)$$

where Γ_{DT} is the double talk detection threshold, and $p_s(n)$ indicates a short-term power estimate of a signal $s(n)$ at time n. Equation (4.2) implies that if the ratio of the send-in signal power to the received signal power exceeds a certain level, double talk is indicated. There are several ways of calculating the power estimates, including a sliding window:

$$p_s(n) = \sum_{j=0}^{M-1} |s(n-j)| \quad (4.3)$$

where M is the window size, i.e. the number of samples included in the estimate. In this case it is not necessary to make this comparison for every input sample; it can be done at regular intervals. This involves a tradeoff of detection latency and probability of false hits.

It is worth repeating that these conventional EEC double talk detectors rely on a certain amount of loss in the echo path. This allows them to distinguish strong near-end speech from echo by the relative levels of the microphone and loudspeaker signals. Unfortunately this will not work in most AEC situations, where the echo return path has a net gain due to amplification. Here, echo is usually the dominant component in the microphone signal, even when near-end speech is present. Thus other methods must be used to detect double talk.

4.3 Signal Correlation

A more promising technique is to monitor the correlation between the error and the reference input. We know from the principle of orthogonality that the expected value of this correlation is zero when the weights have converged to the optimum. It is instructive to expand the correlation expression to see its components:

$$\begin{aligned} E\{e(n)\mathbf{x}(n)\} &= E\{[d(n) - \hat{y}(n)]\mathbf{x}(n)\} \\ &= E\{[y(n) + u(n) + v(n) - \hat{y}(n)]\mathbf{x}(n)\} \\ &= E\{[y(n) - \hat{y}(n)]\mathbf{x}(n)\} + E\{u(n)\mathbf{x}(n)\} + E\{v(n)\mathbf{x}(n)\} \end{aligned} \tag{4.4}$$

The first term will clearly vanish to zero if the adaptive filter is able to generate an echo replica that matches the true echo — this is essentially a restatement of the principle of orthogonality. (Of course, there are limitations to how well this can be achieved in practice, as discussed in section 2.5.) A subsequent change in the echo path will result in imperfect cancellation, and the first term will diverge away from zero. However, the second and third terms will always be zero, assuming that any near-end speech and background noise are uncorrelated with the far-end speech. In other words, double talk will not affect the correlation measure. It appears then that this correlation could be used to distinguish between echo path change and double talk.

4.3.1 Ye-Wu Algorithm

Ye and Wu were among the first to make this observation and try to exploit it [Ye91]. They calculate a running estimate of the cross-correlation between $e(n)$ and $\mathbf{x}(n)$. To arrive at this measure, they first calculate the individual correlation terms:

$$\phi_i(n) = \frac{p_{ei}(n)}{\sqrt{p_e(n)\,p_i(n)}} \tag{4.5}$$

where the following recursive relations with exponential weighting factor λ are used:

$$p_e(n) = \lambda p_e(n-1) + (1-\lambda)\,e^2(n) \tag{4.6}$$

$$p_i(n) = \lambda p_i(n-1) + (1-\lambda)\,x^2(n-i) \tag{4.7}$$

$$p_{ei}(n) = \lambda p_{ei}(n-1) + (1-\lambda)\,e(n)\,x(n-i) \tag{4.8}$$

Then they combine these individual correlation terms into a single measure called the average cross-correlation:

$$\Phi(n) = \frac{1}{N}\sum_{i=0}^{N-1} |\phi_i(n)| \tag{4.9}$$

The final step is to compare the average cross-correlation at each iteration with a predetermined threshold, Γ_Φ. If the correlation measure exceeds the threshold, this is taken to mean that the adaptive filter has not converged, so its weights are adjusted using the basic NLMS algorithm. On the other hand, whenever $\Phi(n)$ falls below the threshold, the weights are frozen.

Notwithstanding the title of their paper: "A new double talk detection algorithm based on the orthogonality theorem", Ye and Wu's approach does not really detect double talk at all. It would be more apt to call it an *echo path change detector*. The algorithm works in a defensive way. Rather than sensing the onset of double talk and then freezing adaptation, the Ye-Wu algorithm will freeze adaptation whenever it is able to, independent of whether double talk is actually occurring. As a result, the proportion of time during which the system is actively adapting will be reduced compared to one using a conventional double talk detector.

One of the biggest problems with such an algorithm is determining an appropriate value for the threshold, especially if the measure being compared to it is not clearly bimodal. If the threshold is set too high, adaptation will be frozen prematurely, and may switch on and off several times before the system converges properly. It will also take longer for the algorithm to unfreeze and track changes in the echo path. On the other hand, if the threshold is set too low the system will become much more vulnerable to double talk, even in its fully converged state.

There is a significant overhead in complexity to implement this algorithm, especially to perform the N divisions and N square root operations at each iteration, as evidenced in Eq. (4.5). This is a severe burden for AEC, where N is required to be large.

4.3.2 Variations

One improvement we suggest for this algorithm is to use different thresholds for turning adaptation on and off. By providing some hysteresis in the trigger levels, it is more likely to avoid multiple transitions and reduce the likelihood of unfreezing during double talk. On the downside, this leaves us with the problem of setting two levels, not just one.

Shan and Kailath propose using the correlation between the output error and the average input $\bar{x}(n)$ as an "automatic gain control" (i.e. variable step size) in the NLMS weight update [Sha88]:

$$\mu(n) = \alpha \hat{E}\{\bar{x}(n)e(n)\} \tag{4.10}$$

where α is a constant parameter. They apparently overlooked two important facts: (i) the correlation estimate can be negative, which would cause the algorithm to diverge, and (ii) the average of $x(n)$ is generally zero when the input is speech.

Casco *et al* propose a slightly different variable step size algorithm [Cas95]. They use the correlation between the residual error and the adaptive filter's output, which is of course a linear combination of the recent inputs. The step size at each iteration can be seen as:

$$\mu(n) = \alpha \hat{E}\{ e(n)\hat{y}(n) \} = \alpha \hat{E}\{ e(n)\mathbf{w}^T(n)\mathbf{x}(n) \} \tag{4.11}$$

where α is again a constant parameter. The authors claim improved ERLE and robustness to double talk. The simplicity of the approach is appealing. Unfortunately, our experiments have shown that the step size can go to zero even when the filter weights are far from the optimum. A trivial case is when $\mathbf{w}(n) = \mathbf{0}$, which is typically used as the initialization value for the AEC weights. The correlation measure can also become negative at times when a large positive step size is desired. Thus it appears this algorithm is fundamentally flawed.

A technique intended to protect against short-term double talk hazards, before a conventional DTD can respond, is presented in [Hay83]. Here, adaptation is prevented on any iteration where the magnitude of the residual error sample $|e(n)|$ exceeds a threshold value Γ'_e. The threshold itself is updated on a block basis according to the short-term correlation between $e(n)$ and $x(n)$, so that it will become larger when the echo path changes:

$$\Gamma'_e(k) = \max\{ \Gamma_{\min}, \Gamma_e(k) \} \tag{4.12}$$

where

$$\Gamma_e(k) = \lambda \Gamma_e(k-1) + \gamma \frac{\sum_i \left| \sum_n e(n)\, x(n-i) \right|}{\sum_i x^2(n-i)} \tag{4.13}$$

and $\Gamma_{\min}$, λ, γ are parameters whose recommended values are not specified in this paper.

4.4 Weight Progression Monitoring

A geometrically intuitive way to distinguish double talk from echo path change is based on the observation that the adaptive filter weights behave differently under the two conditions. When the adaptive filter is in an unconverged state, the trajectory of the weights at time n is in the same general direction that the weights moved in the past. Upon convergence, the current and previous directions of the weights should be unrelated, since the weight vector "wanders" around the optimum. This wandering continues during double talk, only with larger steps. This distinction is illustrated for the two-dimensional case in Figure 4.1.

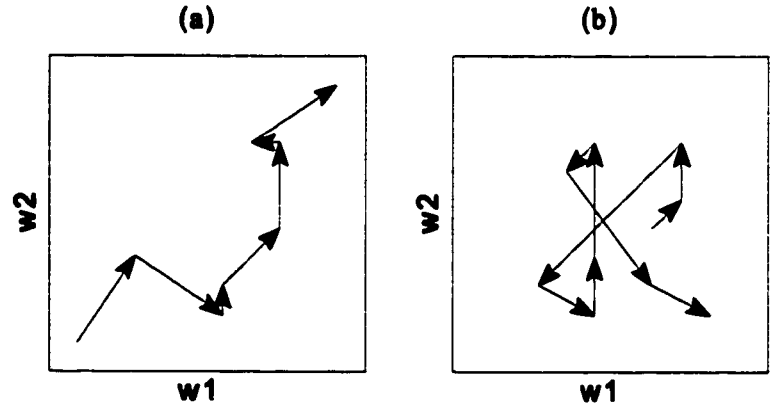

Figure 4.1 Typical trajectories of adaptive filter weights for $N = 2$ (a) after echo path change (b) during double talk.

4.4.1 Rohrs-Younce Algorithm

One method that exploits this weight progression concept has been published by Rohrs and Younce in the form of a United States' patent [Roh90]. As such, its description is couched in legal terminology, without much hard analysis or explanation of how it works. But we will find it instructive to study this algorithm, especially since the patent was assigned to Tellabs Inc., a leading manufacturer of network echo cancellation equipment.

For each new input sample, the first step of the Rohrs-Younce algorithm is to determine the current LMS weight update in the usual way:

$$\Delta \mathbf{w}(n) = \mu e(n)\mathbf{x}(n) \tag{4.14}$$

Note that a fixed step size is used. Then a time average of this weight update vector is obtained, using a forgetting factor α:

$$\overline{\Delta \mathbf{w}}(n) = \alpha \overline{\Delta \mathbf{w}}(n-1) + \Delta \mathbf{w}(n) \tag{4.15}$$

The instantaneous weight update at time n is correlated with the previous time average by calculating the dot product of these vectors and then taking the sign of the result:

$$s(n) = \text{sign}[\Delta \mathbf{w}(n) \cdot \overline{\Delta \mathbf{w}}(n-1)] \tag{4.16}$$

This value will be +1 if the instantaneous update vector is pointing in the same general direction as the trend of the recent past, and -1 otherwise. The correlation measure is smoothed using a simple low pass filter with exponential weighting factor λ:

$$\bar{s}(n) = \lambda \bar{s}(n-1) + (1-\lambda)s(n) \tag{4.17}$$

Finally, the weights are updated under the condition that the smoothed correlation value exceeds a predetermined threshold, Γ_S:

$$\mathbf{w}(n+1) = \begin{cases} \mathbf{w}(n) + \Delta \mathbf{w}(n), & \bar{s}(n) > \Gamma_S \\ \mathbf{w}(n), & \text{otherwise} \end{cases} \tag{4.18}$$

Unfortunately the text of the patent gives no guidance as to the selection of the parameters α, λ, Γ_S, or μ, nor any performance analysis.

An additional detail, not shown in the preceding equation, is that the weight update is forced to occur whenever the current ERLE is within 6 dB of its recent maximum value $ERLE_{MAX}$. This can be interpreted as allowing 6 dB of degradation in ERLE before freezing

the filter weights. $ERLE_{MAX}$ is reset to zero after a double talk period once the smoothed correlation value exceeds its threshold.

One drawback of this algorithm that is immediately evident is its computational complexity. Performing the vector dot product requires N operations per input sample, and the vector updates in Eqs. (4.15) and (4.18) use an additional $3N$. Thus there is at least a 200% increase in complexity over the basic LMS algorithm. Another problem that we have mentioned elsewhere is the difficulty in selecting a good threshold level, Γ_S.

4.4.2 Variations

The origins of this concept of monitoring the weight progression can be traced back to work on *stochastic approximation*, which predates the LMS algorithm. Here a quantity θ is estimated iteratively by a sequence $\{\epsilon_n\}$. In this one-dimensional case, Kesten remarked that frequent changes in the sign of $\epsilon_n - \epsilon_{n-1}$ indicate that the estimate is close to θ, while few sign fluctuations indicate that the approximation procedure has not yet converged [Kes58]. He proposed to accelerate convergence by starting with a large step size and reducing it gradually according to the cumulative number of sign changes up to time n. Of course, this had nothing to do with double talk!

Harris *et al* present a variable step size LMS algorithm based on extending this concept to N dimensions, and allowing the step sizes to increase if necessary [Har86]. This represents a special class of VSS in which each weight has its own step size. The Harris algorithm monitors the signs of all the individual weight updates at each iteration. If any sign stays the same for m_l consecutive samples, it is presumed that the corresponding weight is far from the optimum, so its step size is increased to accelerate convergence. Conversely, if the sign

alternates for m_0 consecutive samples, this is an indication that the weight is near the optimum, and the step size is decreased. While this may be reasonable for a hardware implementation, which can exploit massive parallelism, it would be costly to implement in software.

4.5 Dual Filter Architecture

The previous classes of double talk algorithms have been confined by the notion of having only a single adaptive filter. Ochiai and colleagues did some lateral thinking and appear to have been the first to suggest that two filters might be used, in a foreground-background configuration [Och77]. This general scheme is shown in Figure 4.2.

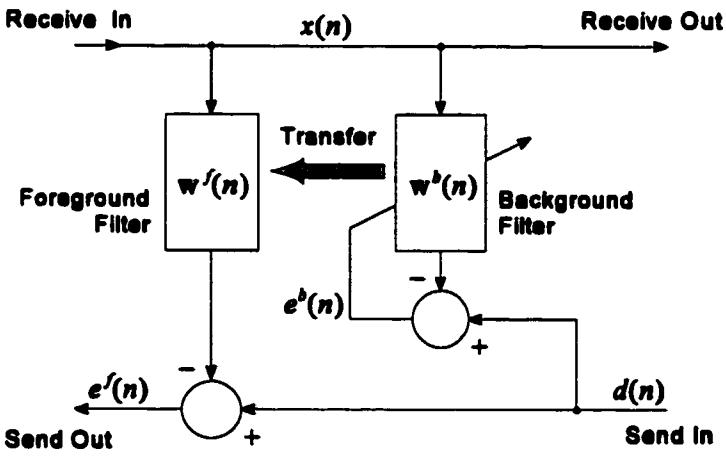

Figure 4.2 Foreground/background dual filter architecture.

Here the foreground filter provides the echo replica used for cancellation, but its weights are not adapted in the conventional sense. Meanwhile the background filter is allowed to adapt

aggressively. The weight values are "transferred" (actually copied) from the background filter to the foreground filter whenever they represent an improved model. The relative improvement is judged according to some appropriate set of criteria.

4.5.1 Ochiai Algorithm

The transfer conditions used by Ochiai *et al* for their EEC system are:

(i) the estimation error from the background filter is smaller than that from the foreground filter by a factor of at least β:

$$p_e^b(k) < \beta p_e^f(k) \tag{4.19}$$

(ii) the primary signal (mainly echo) is being cancelled by the background filter by a factor of at least γ:

$$p_e^b(k) < \gamma\, p_d(k) \tag{4.20}$$

(iii) the microphone signal is smaller than the far-end signal:

$$p_d(k) < p_x(k) \tag{4.21}$$

This third case prevents transfer during a clear double talk condition (compare to Eq. (4.2) with $\Gamma_{DT} = 1$). All three transfer criteria are checked periodically, every T_C seconds, hence the block index k as opposed to sample index n for the power estimates above. For extra protection against false transfer, conditions (i) to (iii) must all be satisfied over R consecutive time intervals before a transfer will occur.

In their implementation, Ochiai *et al* used the NLMS algorithm to adapt the background filter, although nothing precludes the use of other adaptive algorithms in conjunction with a dual filter architecture. The two filters were of length $N = 324$. After some

experimentation, the following parameter values were reported to be suitable: $M = 128$, $T_C = 16$ ms, $T_{HANG} = 128$ ms, $\mu = 1$, $\beta = 0.875$, $\gamma = 0.125$, $R = 3$.

The cost of this foreground/background architecture is a 50% increase in both the memory requirements and the computational load versus a single filter implementation. The latter is due to the fact that a second echo replica must be formed. The transfer of weights from background to foreground can be done in parallel in a hardware implementation. But in software, this would have to be done serially, which would add a computational penalty. To meet real-time constraints, the processor may have to forego updating the background filter weights for one cycle while the transfer is performed, for instance.

This method results in an aggressive update of the background filter, and a conservative update of the foreground one. Overall performance is reported to be very good. The authors conducted an experiment where they applied recorded speech at both ends of the line under controlled conditions. They counted the number of double talk intervals that occurred, and the number of "false transfers" that occurred during this double talk. They report a very low rate of false transfers: 7×10^{-4} for the case where the near-end signal level was 20 dB lower than the far-end speech. In other words, 99.93% of the time the foreground weights were correctly frozen during double talk.

A subjective evaluation also found that the EEC system with two echo path models was rated significantly higher than one with a single echo path model plus conventional double talk detector over a wide range of ERL levels and loop delays.

One caveat is that if the far-end and near-end signals are correlated at all during double talk, the background filter may have a smaller residual error even if its weights are actually

further from the optimum than the foreground filter's. This may cause an unwanted coefficient transfer [Wan95]. Another drawback is that after the double talk ceases, the background filter will take some time to reconverge to the optimum, so it will not be immediately useful for further updates.

4.5.2 Variations

A refinement of the dual filter approach is described in [Lew93]. Here, the foreground weights are copied back to the background filter after a period of double talk divergence. By effectively resetting the background weights to their state before double talk started, this "downdate" procedure enables immediate tracking of a changing echo path, without waiting for the background filter to reconverge.

Wang *et al* recommend using a faster algorithm such as FTF for the background filter, while keeping NLMS for the foreground filter [Wan95]. Only one filter is adapted at a given time, depending on the state of the canceller. The FTF provides fast convergence initially and after an echo path change. Periodic intervention by the NLMS helps to keep the system stable. The problem with this idea is that a real-time implementation is bounded by the worst case performance; average performance means little. Even though the FTF may not be operating all the time, the system must be able to process $8N$ operations, a big increase over Ochiai's original proposal.

Another variation of the foreground/background architecture is proposed by Chang *et al* [Cha86]. They describe a complicated procedure of switching back and forth between two filters of different lengths. The longer one is used for coarse modelling, usually in the background, to find the appropriate position for the weights in the shorter filter, which

converges faster. The main benefits of this approach apply only to EEC, since in AEC the entire echo path model is dispersive.

Amano *et al* suggest a dual filter approach with a different twist [Ama95]. They actually have a subband filter structure in the foreground. For double talk detection, they employ a single filter in the background. It only covers the band from DC to 1000 Hz, where most speech is concentrated, and thus can be operated at a reduced rate. The background ERLE is compared to a pair of thresholds to discriminate between double talk and echo path changes. This asymmetric structure precludes the possibility of transferring weights from background to foreground in this case, it is simply used for detecting changes in the system state. The main similarity with the Ochiai algorithm is that the background filter continuously adapts, while the foreground one can be frozen whenever double talk is suspected.

4.6 An Assortment of Other Techniques

The remaining published remedies for the problem of double talk in echo cancellation do not readily fall into one of the categories discussed above. In some cases, they are hybrid approaches. For completeness, some of these alternatives are briefly reviewed in the following paragraphs.

Gänsler *et al* measure the similarity between the reference and primary input signals. They do this by evaluating the coherence function on a block-by-block basis in the frequency domain [Gan97]. They do not discuss AEC in their paper, only EEC.

Another suggestion is to exploit the statistical distribution of double talk duration [Min85]. Long periods of continuous double talk are known to be rare. Thus, upon sensing

an increase in the estimation error, Minami's algorithm holds off adapting the filter weights until enough time has passed that the sustained error is almost certainly due to an echo path change, not double talk. This is a very simple approach, but has a major downside: the convergence time is increased by the required hold-off time, found to be 3 seconds in practice. This conservative approach was intended to be applied to EEC, where echo path variations after initial convergence are much less common than AEC.

Also for EEC, Chao and Tsujii propose a three-port configuration that uses the signal from the near-end two-wire circuit (beyond the hybrid) as an additional input [Cha89]. This is not directly applicable to AEC, since we have no direct access to a comparable third port. Conceptually, we would have to have another microphone as a sensor in the acoustic environment to mirror the EEC situation, but this would introduce many complications. Incidentally, these authors claim their system is able to track echo path changes during double talk.

Yoo and Cho propose a DTD that includes two lattice predictors, driven by signals $x(n)$ and $d(n)$, respectively. A control block monitors the rate of change of the reflection coefficients, and indicates double talk when the near-end speech is changing much faster than the far-end speech [Yoo97]. They claim this technique greatly reduces the detection delay. However, it seems to be based on the assumption that the amplitude of the near-end speech is larger than the echo, a false hypothesis for a desktop speakerphone set-up.

Heitkämper *et al* have developed a complicated DTD procedure that includes aspects of both the signal level comparison and correlation techniques [Hei95, Hei97]. They calculate a time-varying average coupling factor between the reference and error signals to represent

the typical echo return loss. This factor is updated only when the correlation between the reference and primary input signals is high, implying that only far-end speech is present. The DTD declares double talk whenever the ratio of the actual signal levels $e(n)$ and $x(n)$ exceeds the average coupling factor by a certain margin. An apparent problem with this is that these ratios are likely to be highly dependent on the frequency content of the input signal, which can vary dramatically in the case of speech.

Doherty and Porayath present a spread-spectrum technique to estimate the room impulse response even during double talk periods. They superimpose a pseudo-noise training signal on the loudspeaker signal, and correlate this known sequence with the portion received at the microphone, to provide an estimate of the room transfer function. They claim the training signal is "innocuous" and unaffected by the presence of near-end speech, since these signals are theoretically uncorrelated [Por96, Doh97]. In practice some residual correlation is present, which prevents perfect cancellation.

Asharif *et al* introduce the Correlation LMS algorithm, based on using the autocorrelation function of $x(n)$ as the reference input and the cross-correlation of $x(n)$ with $d(n)$ as the primary input in an NLMS structure [Ash99]. Thus the adaptive filtering is done on the correlation terms, but the resultant weights are copied to a fixed foreground filter, which estimates the echo in the conventional sense. By making the assumption that the correlation between the near-end and far-end speech is zero, they claim that the proposed algorithm is insensitive to double talk. However, this is yet to be demonstrated for real speech signals.

4.7 Summary and Discussion

This chapter has reviewed a multitude of suggestions for combatting the problem of AEC divergence during double talk. The fact that so many fundamentally different proposals exist is a testament to both the difficulty of the problem and the creativity of the signal processing community in trying to address it. We have chosen to group these algorithms according to their inherent signal processing techniques, in order to emphasize their similarities and differences. Another way to classify them is into two types: (i) those that detect double talk and then freeze adaptation, and (ii) those that detect echo path changes and then enable adaptation.

The problem with direct double talk detection is that it does not work instantaneously. It is impossible to distinguish double talk and an echo path change based on a single sample; many iterations are required to obtain good estimates of the system state via time averaging. This means there is an inevitable delay between the onset of double talk and its detection. During this time, if the step size is large, the adaptive filter weights will rapidly move away from the optimum. Then when the DTD activates, the weights will be frozen at these inferior values for the remainder of the double talk. No matter how accurate a DTD is, any significant detection delay may render it useless. Fujii and Ohga suggest compensating for this detection delay by applying a corresponding delay to the adaptation process [Fuj93]. In this way double talk can be detected before the filter weights are adversely affected. This seems to be a band-aid solution, and has the unfortunate effect of delaying any convergence or tracking by an equivalent amount of time.

Echo path change detection appears to be a more intelligent strategy, in that the filter weights are only adapted when necessary. For example, by taking the conservative stance of freezing the adaptive filter weights whenever the system has converged, Ye and Wu avoid double talk problems before they happen [Ye91]. However, their tactic of using a fixed step size for the adaptation has some serious weaknesses:

- if an echo path change is falsely detected during double talk, divergence will be rapid;
- to minimize such false hits, they must set their correlation threshold quite high; as a result, adaptation tends to switch on and off during the convergence phase, leading to slower overall convergence;
- also, there is a higher final misadjustment.

These points underline the dependence on the critical threshold parameter. An improvement would be adjust the adaptation rate so that it is small during double talk and large when required to track a change in the echo path.

In the next chapter we contribute a new approach for dealing with double talk, based partly on what has been learned from this study of the relevant published work.

5 PROPOSED VARIABLE-STEP-SIZE ALGORITHMS

> *More attention should be paid on how to design properly control mechanisms associated with adaptive filtering algorithms, which provide adequate overall acoustic echo attenuation while permitting satisfactory double talk operation.*
>
> — Gilloire and Hänsler (1994)

The preceding chapters have established the critical need to distinguish between double talk and echo path changes in the operation of an acoustic echo canceller. Existing techniques that attempt to achieve this goal are not fully satisfactory. In this chapter we propose a new approach that draws on some elements of previous research, but is fundamentally different in its underlying concept.

The discussion at the end of Chapter 4 leads us to develop a set of algorithms in which the step size itself can adapt according to the state of the system. Ideally this $\mu(n)$ should be large when far from the optimum, and small when near the optimum *or in the presence of double talk.* The proposed algorithms will be shown to offer:

- fast convergence and low misadjustment during single talk;
- resistance to disturbance from near-end speech;

- a means of controlling the auxiliary echo suppression circuitry via a double talk detection signal;
- reasonably low computational complexity.

All of these important attributes are highlighted through simulations and analysis in due course, but first we explain the algorithms' development.

5.1 Appeal of a Variable Step Size Approach

Consider the situation where the AEC system has converged and is stable near the optimum for some time. The step size will have been reduced to its minimum value, perhaps zero (i.e. the coefficients are frozen), and the ERLE will be at its maximum. There are two possible scenarios that can change the state of the system; either:

(i) a change in the echo path occurs, or

(ii) near-end speech begins.

In case (i) we want the AEC to adapt as quickly as possible to the new room impulse response, which implies increasing the step size to a large value initially, then decreasing it gradually as the new optimum is reached. However, in case (ii) we want to keep the step size very small to prevent divergence of the adaptive filter coefficients due to the near-end speech. Conceptually, we would like to set μ at each step based on the probability that we are in double talk or converged versus unconverged. Since it is not possible to distinguish the two cases on the first sample, the prudent thing to do is to keep the step size small until it is clear that case (i) has occurred. It may be possible to compromise and increase the step size gradually when it is quite likely that (i) has occurred, and then continue to increase it more

quickly as that likelihood increases. Similarly, we should reduce the step size as it becomes more likely that (ii) has occurred, reducing it to the minimum when this becomes almost certain. In other words, the aggressiveness of the adaptation should be proportional to the confidence that such adaptation would be beneficial. By having a continuously variable range of step sizes with some smoothing, we might expect to benefit from the following advantages:

- lower misadjustment in the converged state;
- continuous convergence, without false stopping and restarting; and
- some margin for error if the wrong state is estimated for a short period of time (since the step size does not change very rapidly).

The primary issue in designing a variable step size algorithm is to decide on what time varying measure $q(n)$ to base the step size. The step-size update is usually of the form:

$$\mu(n) = \alpha\mu(n-1) + f(q(n)) \tag{5.1}$$

where α is a smoothing factor and $f(\cdot)$ denotes some function of the driving force. Other researchers have used a variety of measures to drive the step size. For example, Kwong uses the mean square error itself [Kwo92]:

$$q(n) = e^2(n) \tag{5.2}$$

This helps to accelerate convergence since the squared error is large when far from the optimum, and small when near the optimum. However, this is only true in the absence of a near-end signal. After convergence, $e(n)$ is dominated by any near-end speech or noise that is present. When this gets large, the step size gets large too, and so performance is very poor during double talk, unless a separate DTD is used, as well as in noisy environments.

Mayyas realized this limitation and proposed another VSS that is much more robust to near-end interference [May95a]. With

$$q(n) = e(n)e(n-1) \tag{5.3}$$

the step size update now depends on the autocorrelation of $e(n)$ with a unit lag. This works well when the near-end signal $u(n)$ is white, since it contains no correlation from one sample to the next. In contrast, the echo is coloured by the room transfer function, providing a relatively strong autocorrelation of $e(n)$ if residual echo is present. The consequence is that the step size changes in accordance with the amount of echo, but is not affected by white noise interference. However, in a telephony application the near-end signal contains speech, which is highly coloured and non-stationary, so performance of this VSS will degrade. In other words, the Mayyas VSS algorithm is not robust to realistic double talk in AEC.

We also note here for future reference that Mathews and Xie [Mat93] derive a step size update that attempts to minimize the squared estimation error on each iteration using a stochastic gradient technique:

$$q(n) = \frac{1}{2}\frac{\partial e^2(n)}{\partial \mu(n-1)} = e(n)e(n-1)\mathbf{x}^T(n)\mathbf{x}(n-1) \tag{5.4}$$

5.2 Inspiration for a Robust VSS Algorithm: Gradient Correlation

Given the proven advantages of VSS approaches, we will follow this general direction. However, we want to use some measure of the convergence state and double talk condition to drive the step size update. As first noted in section 4.4, a compelling way to distinguish between double talk and echo path change is to monitor the progression of the adaptive filter

weights over time. If we think of the weight vector as representing a point in N-space, the "motion" of this point as the filter adapts can provide an indication of the system state. If the weights are continually moving with a consistent trend in one direction, they must be far from the optimum, whereas if they are oscillating randomly around a fixed point for some time, the system has probably converged. The magnitude of such oscillations is related to the noise level in the adaptation process. Note that for the purpose of system state determination, we are more interested in the *relative* movement of the weights than their absolute position. Specifically, we want to know whether there is any similarity or *correlation* between the direction the weights are moving currently and their general trajectory from the recent past. In an LMS approach, the weights are updated according to the gradient of the error performance surface. To exploit the key observation made above, we therefore propose to use the correlation between the instantaneous and time-averaged gradient vectors as the driving force in the step size update[3]:

$$q(n) = -\nabla_{\mathbf{w}}(n) \cdot -\bar{\nabla}_{\mathbf{w}}(n-1) \tag{5.5}$$

Note that the time-averaged gradient intentionally excludes the current estimate, otherwise there would always be some correlation.

It is instructive at this point to study the evolution of the quantity in Eq. (5.5) in some more detail. Let $\mathbf{g}(n) = -\nabla_{\mathbf{w}}(n) = e(n)\mathbf{x}(n)$ represent the negative gradient estimate[4] at time n. Figure 5.1 shows the results of a simple experiment. For a system with $N = 2$, the

---

[3] The correlation between two vectors is obtained by taking their dot product.

[4] To avoid cumbersome language, we usually omit the word "negative" when referring to the gradient or its estimates. Strictly speaking, it is always implied.

adaptive filter weights are placed at the optimum, $\mathbf{w} = \mathbf{h} = (1, 1)$. This represents the ideal condition after convergence, where the step size has been reduced to a very small value so that adaptation is essentially frozen. White noise is applied at the reference input. From this starting point we consider what happens under the two different conditions that have to be distinguished. In the case of double talk, we see that the instantaneous gradient vectors point in all directions, with no apparent bias for a particular angle. In contrast, after an abrupt echo path change the gradient vectors are confined to the lower right half plane, which contains the new optimum $\mathbf{h}' = (1.8, 0.2)$. The directions are uniformly distributed over half the unit circle. This is because for a given input vector $\mathbf{x}(n)$ and a scalar error $e(n)$, there are only two possible *directions* for the instantaneous gradient estimate: either $+\mathbf{x}(n)$ or $-\mathbf{x}(n)$, the polarity being dictated by the sign of $e(n)$.

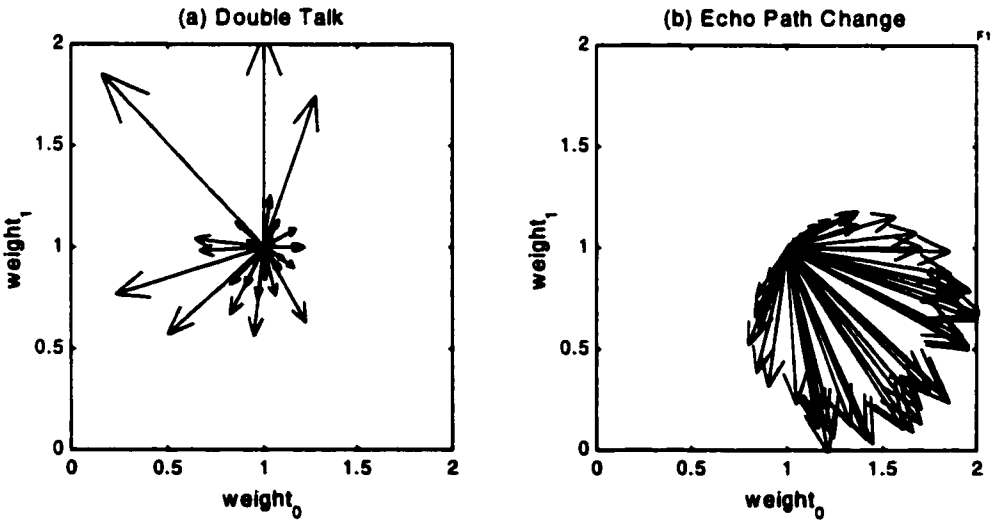

Figure 5.1 Comparison of a series of instantaneous gradient estimates after initial convergence to $\mathbf{w} = \mathbf{h} = (1, 1)$ for $N = 2$: (a) during double talk, and (b) after an echo path change to $\mathbf{h}' = (1.8, 0.2)$.

It should be obvious from Figure 5.1 that it would be impossible to distinguish double talk from an echo path change on the basis of a single gradient estimate. Even in this trivial case it takes several samples before a clear pattern can be identified. In a real application, either of these conditions can begin at any point in time, and the transition from one state to another will be gradual.

Now consider extending this situation to N dimensions. The coordinate axes divide an N-space into 2^N regions, each of which can be associated with a different combination of polarities of the components; for example, 2-space has 4 quadrants, 3-space has 8 octants. However, independent of N, there are still only two possible directions for the gradient estimate vector at a given time, either $+\mathbf{x}(n)$ or $-\mathbf{x}(n)$. In other words it is confined to point into one of two out of the 2^N possible regions. The actual optimum lies in one of the 2^N possible regions. But since $\mathbf{x}(n)$ is assumed to be uncorrelated with the position of the optimum, the likelihood that $\mathbf{g}(n)$ and the actual optimum are in the same direction is very low when N is large. This shows the price paid by the LMS algorithm for using instantaneous gradient estimates, which are inherently noisy.

We recall from Chapter 3 that the steepest descent algorithm updates the filter weights using the expected value of the gradient, which points directly down the contours of the MSE bowl. Strictly speaking, this requires averaging the estimates at a given time over the ensemble of all possible experiments. Since we do not have access to more than one realization, we must be content with the next best thing: a time average of the gradient estimates. Averaging of the gradient estimates over several consecutive iterations, as in the block LMS algorithm (with block size $B > 1$), reduces the variation in the estimates and

increases the likelihood that the resulting vector is pointing in the desired direction, i.e. towards the optimum. Figure 5.2 demonstrates this. In double talk, the block averaging has reduced the magnitude of the gradient estimates, since the instantaneous vectors add destructively. However, after an echo path change the gradient estimates add constructively, so the block averaged version is better aligned with the optimum.

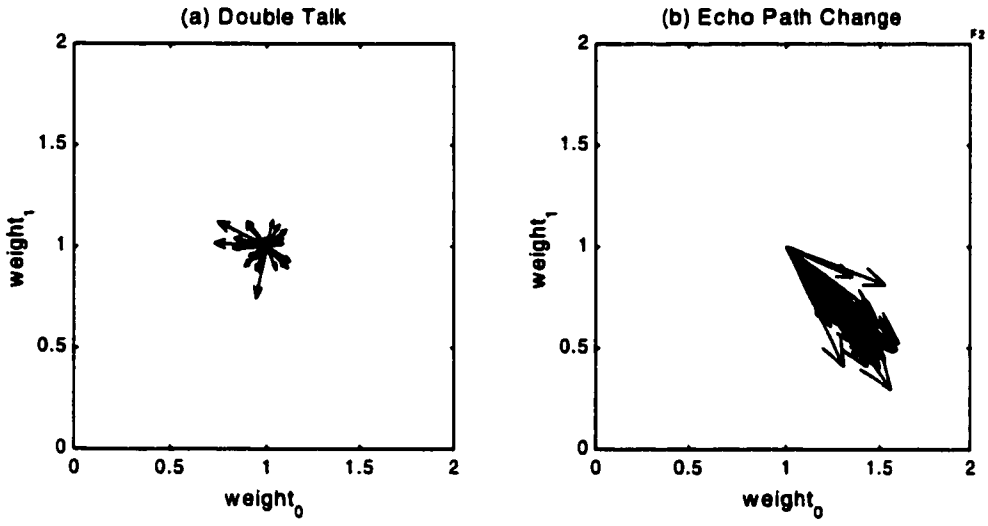

Figure 5.2 Comparison of block-averaged gradient estimates for $N = 2$ and a block size of 10 samples: (a) during double talk, and (b) after an echo path change.

Of course, it is not possible to plot the gradient vectors for a much larger dimension, but we can look at the distribution of the angle between the block-averaged gradient estimate and the optimum update vector. This is done in Figure 5.3, comparing the distribution of the angle for $N = 2$, 16 and 1024. Plots (a) and (b) agree with the previous results for $N = 2$ shown in Figures 5.1 and 5.2. When N is increased to 16, we see that the instantaneous gradient is seldom within 45 degrees of the optimum. By averaging, we can reduce the directional error somewhat, but the distribution is not nearly as close to the ideal zero degrees

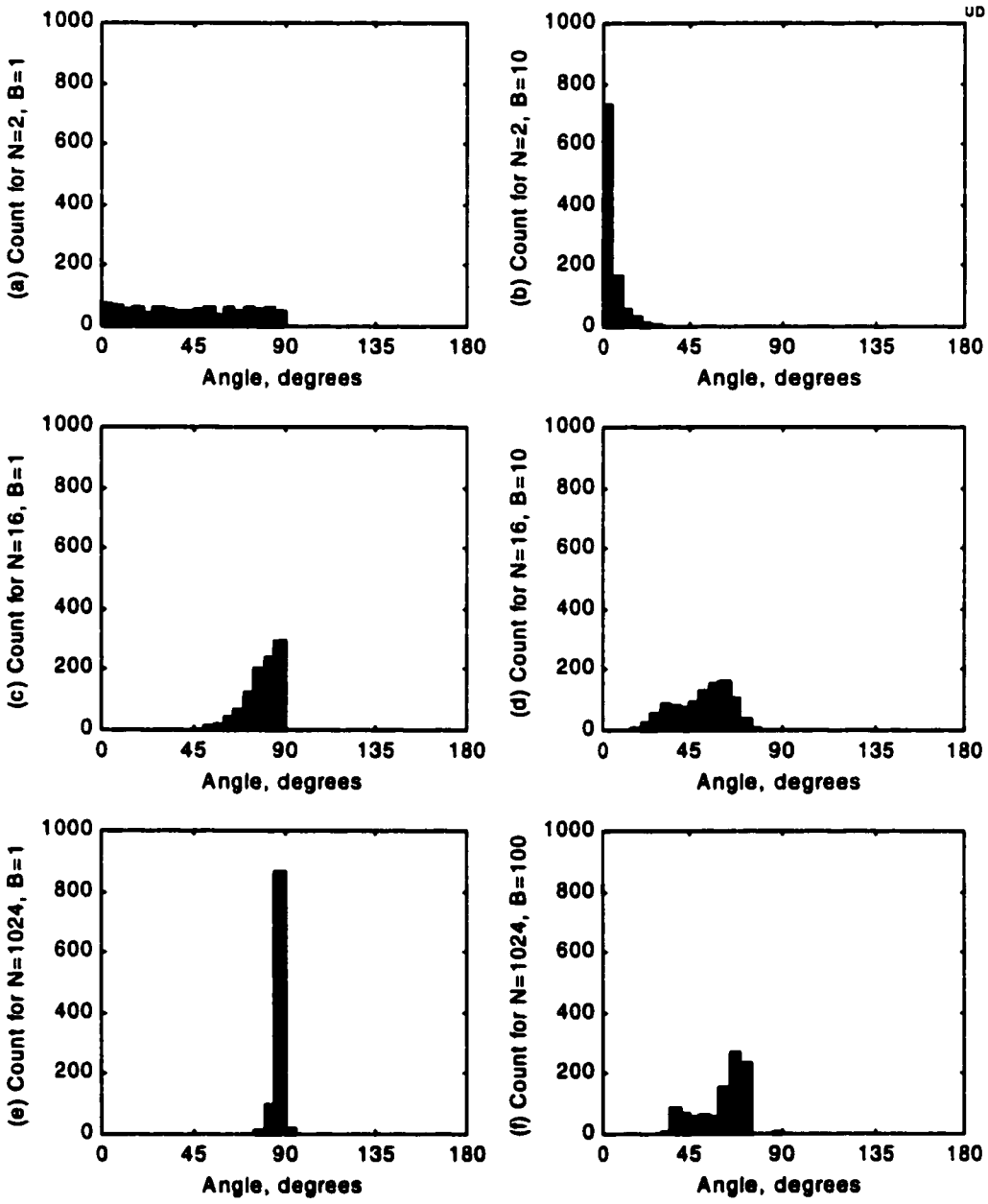

Figure 5.3 Histograms showing the distribution of angles between the true gradient and its block-averaged estimate for different values of N and B. The results for each plot come from a Monte Carlo simulation with 1000 runs. The width of each bin is 5 degrees.

as for $N = 2$. Finally, for a very long adaptive filter with $N = 1024$, we see in plot (e) that the vast majority of instantaneous gradients are almost orthogonal to the ideal update direction. It takes a considerable amount of averaging just to bring the average into partial alignment with the optimum, as shown in plot (f) with $B = 100$.

We can conclude that the amount of averaging required to obtain a good estimate of the past weight trajectory depends strongly on the adaptive filter length.

5.3 Development of Proposed GC-VSS Algorithm

The details of our proposed algorithm are introduced in several stages in this section. The overall approach is referred to as Gradient Correlation - Variable Step Size (GC-VSS). First we start with a basic version and then show how it has evolved through a number of modifications in an effort to improve performance. The incremental improvements are demonstrated qualitatively, but we save detailed experiments for section 5.4.

5.3.1 Basic GC-VSS Algorithm Formulation

The instantaneous estimate of the MSE gradient, as used directly by the LMS algorithm, is given by:

$$\mathbf{g}(n) = e(n)\mathbf{x}(n) \tag{5.6}$$

Strictly speaking this is the negative direction of the gradient estimate, since the gradient points up the slope of the MSE bowl. The desired direction of the LMS update is down the bowl's surface. To obtain a better estimate of the true gradient, the time average of this gradient vector is calculated, using an exponential forgetting factor λ:

$$\bar{\mathbf{g}}(n) = \lambda \bar{\mathbf{g}}(n-1) + (1-\lambda)\mathbf{g}(n) \tag{5.7}$$

This is similar to the update used in [Roh90] except there they do not weight the current gradient estimate (see Eq. (4.15)), which would appear to lead to an unstable recursion. An alternate average is the finite windowed one:

$$\bar{\mathbf{g}}(n) = \sum_{b=0}^{B-1} \mathbf{g}(n-b) \tag{5.8}$$

We have omitted the constant $1/B$ factor, as would be expected of an average in the true sense, because we are interested in the direction of this vector, not its absolute magnitude. An equivalent recursive calculation is:

$$\bar{\mathbf{g}}(n) = \bar{\mathbf{g}}(n-1) + \mathbf{g}(n) - \mathbf{g}(n-B) \tag{5.9}$$

We will use this instead, for reasons that will become clear in section 5.7. Note that Eq. (5.9) can lead to numerical instability in a floating point implementation, but is safe when using fixed point arithmetic. Next, the instantaneous gradient estimate at time n is correlated with the old time average by calculating the dot product of these vectors:

$$c(n) = \mathbf{g}(n) \cdot \bar{\mathbf{g}}(n-1) \tag{5.10}$$

The step size is updated based on the correlation at each iteration according to:

$$\mu'(n) = \alpha \mu(n-1) + \gamma c(n) \tag{5.11}$$

where $0 < \alpha, \gamma < 1$. The idea behind this update is that the step size should increase if $c(n)$ is strongly positive and decrease if $c(n)$ is zero or negative. The parameter α helps to smooth the step size, and since it is less than 1, causes the step size to decrease when $E\{\ c(n)\ \}$ is zero.

Often VSS algorithms limit the minimum step size to be a small positive value to ensure a minimal tracking capability after initial convergence [May95b]. However, our algorithm is designed to automatically increase the step size whenever tracking is required, so the

minimum can be safely set to zero. This has the advantage of allowing complete freezing during double talk. It also means there is one less parameter whose value has to be selected to balance different aspects of a performance tradeoff. Thus in the proposed algorithm the variable step size is limited to a range $[0, \mu_{MAX}]$, where the upper limit is primarily intended to ensure stability:

$$\mu(n) = \begin{cases} 0, & \mu'(n) < 0 \\ \mu_{MAX}, & \mu'(n) > \mu_{MAX} \\ \mu'(n), & \text{otherwise} \end{cases} \tag{5.12}$$

Finally the weights may be updated as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu(n)e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta} \tag{5.13}$$

The only difference from the NLMS weight update is the time-dependence of the step size.

5.3.2 Initial Results for Basic GC-VSS

In Figure 5.4 we show typical algorithm performance over three modes of operation: initial convergence, double talk, and echo path change[5]. The baseline for comparison is the NLMS algorithm *without* any double talk detection. Parameter values for both algorithms are provided in the figure's caption. The near-end and far-end signals used in this initial stage of validation are both white Gaussian noise.

We observe that the new GC-VSS algorithm's curves track the NLMS at the very beginning, when the step sizes are essentially the same. However, by the time the weight error has been decreased to about -20 dB the step size starts to shrink, and thus the convergence

---

[5] Details of the simulation environment used, particularly the system configuration, will be provided in section 5.4.1.

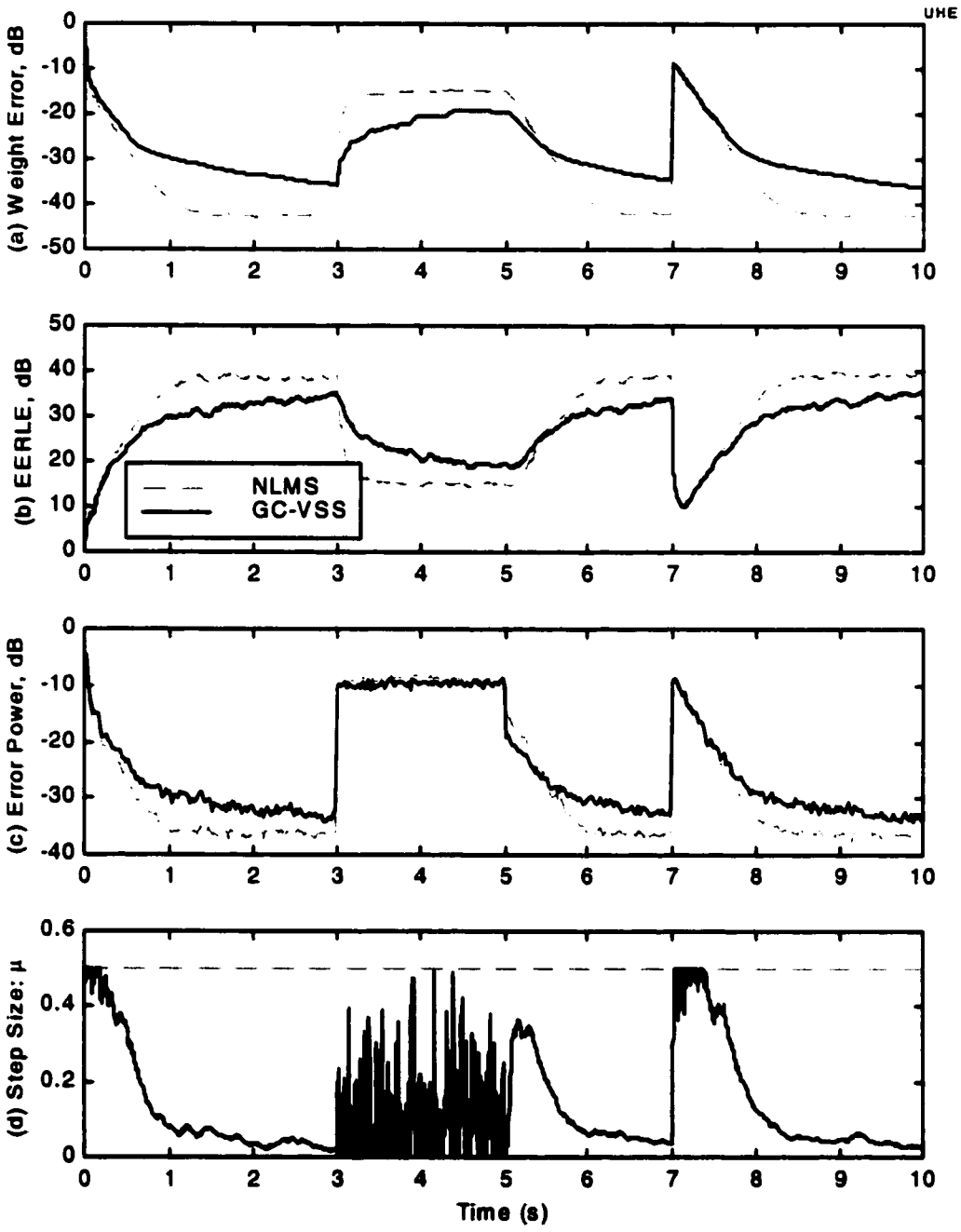

Figure 5.4 Typical performance of basic GC-VSS algorithm. Parameter values are $\alpha = 0.999$, $\gamma = 0.0002$, $B = 500$, $\mu_{MAX} = 0.5$. Adaptive filter length $N = 1024$. Double talk is applied between 3 and 5 seconds, and an abrupt echo path change occurs at 7 seconds. Performance using NLMS ($\mu = 0.5$) without any double talk detection is shown for comparison.

becomes much slower. This is due to the smaller correlation term $c(n)$ as the error $e(n)$, and thus the gradient magnitudes, become smaller.

When double talk is applied at the 3-second mark, the NLMS diverges rapidly due to the gradient noise amplification effect. GC-VSS also diverges, although at a slower rate because the step size is less. However, we see that the step size is very noisy, even reaching its maximum value for brief instants during double talk, again due to the large variance in the correlation term (even though its *mean* is theoretically zero). By the time double talk stops at $t = 5$ seconds, the excess ERLE has dropped by 14 dB, and is only about 5 dB better than the unprotected NLMS. We note that as soon as the disturbance stops, the GC-VSS step size increases quickly since the algorithm detects that there is a significant weight error. Reconvergence performance here and after the abrupt echo path change at $t = 7$ seconds mirror the initial convergence for both algorithms.

The main conclusion of this first experiment is that the step size update experiences a gradient noise amplification effect of its own. The step size becomes too small too quickly during initial convergence and too large during double talk, depending on the amplitude of the gradient terms in the correlation. This illustrates the need to *normalize* the step size update term, in a similar way that the normalized LMS was developed as an improvement on the basic LMS for cases where there are changes in the input signal level. This observation leads us to the first modification to GC-VSS.

5.3.3 Modification 1: Normalization

We could use the *correlation coefficient* of the instantaneous and average gradients as a normalized step size update term:

$$cc(n) = \frac{\mathbf{g}(n) \cdot \bar{\mathbf{g}}(n)}{\|\mathbf{g}(n)\| \, \|\bar{\mathbf{g}}(n)\|} \tag{5.14}$$

This expression is equal to the cosine of the angle between the two vectors, and has a range $[-1, +1]$. However, its calculation requires a square root operation in the denominator as well as the obvious division. Instead, we can simply use the sign of the correlation, similar to the concept of the sign LMS algorithm. The step size recursion is then written:

$$\mu'(n) = \alpha \mu(n-1) + \gamma \, \text{sign}[c(n)] \tag{5.15}$$

This means the update term takes on values of γ, 0, or $-\gamma$, which can be seen as an extreme type of normalization.

The simulation results provided in Figure 5.5 clearly show the improvement in performance versus the original case. The solid line represents Modification 1, labelled GC-VSS1 for short. It must be noted that we have increased the adaptation gain γ by a factor of 10, to compensate for the sign operation's inherent gain reduction at moderate gradient amplitudes. Other parameter values remain unchanged from the previous experiment.

Note how the step size stays larger for longer now, allowing the algorithm to converge to a weight error of -45 dB after 3 seconds have elapsed — this is a lower level of misadjustment than the previous NLMS example. When double talk begins, the GC-VSS step size has a value around 0.1, and this is reduced significantly within 100 ms as the algorithm "senses" double talk. During this reaction time, the weight error diverges quite rapidly to about -30 dB, where it almost plateaus but continues to erode slightly over the next two seconds. By the end of the double talk, a 10 dB improvement in echo cancellation is achieved compared to the basic GC-VSS, and this translates into a much quicker recovery after $t = 5$ s.

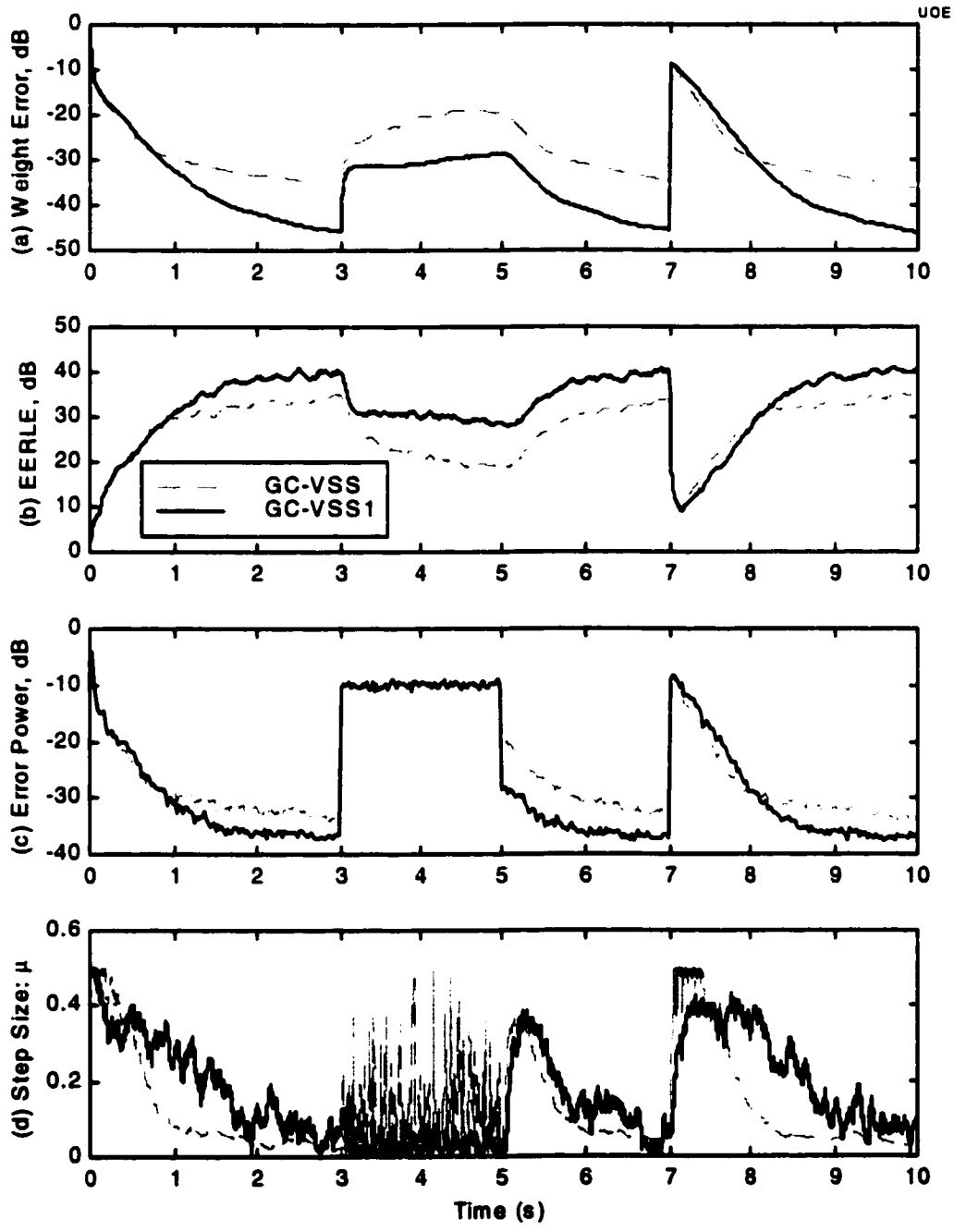

Figure 5.5 Improvement in performance of GC-VSS algorithm by using Modification 1. Parameter values remain unchanged, except $\gamma = 0.002$.

One final point to note is that while the reconvergence after the echo path change is similar to the initial adaptation, a slight start-up delay is observed. This reason for this is revealed in the step size plot (d), where it takes about 200 ms for the step size to reach its new peak. This represents this algorithm's echo path change detection delay. The fact that the peak step size is slightly below the maximum level of 0.5 could be alleviated by increasing the adaptation gain γ. However, this will also increase the step size during double talk, leading to greater divergence, so there is a tradeoff here.

5.3.4 Modification 2: Smoothing

One apparent way to improve performance further would be to somehow de-noise the *instantaneous* gradient before correlation. In this case we could think of having a short-term average gradient from the recent past $\tilde{\mathbf{g}}(n)$, and comparing its direction with the longer-term average from the more distant past $\bar{\mathbf{g}}(n)$. We presented this approach in [Cre98], where the weights and step sizes were updated on a block basis every K samples, K being the block size. However, we now note that the correlation measure so obtained is given by:

$$\begin{aligned} \tilde{c}(n) &= \tilde{\mathbf{g}}(n) \cdot \bar{\mathbf{g}}(n-K) \\ &= \left(\sum_{k=0}^{K-1} \mathbf{g}(n-k) \right) \cdot \bar{\mathbf{g}}(n-K) \\ &= \sum_{k=0}^{K-1} \left(\mathbf{g}(n-k) \cdot \bar{\mathbf{g}}(n-K) \right) \end{aligned} \tag{5.16}$$

Compare this to a correlation measure formed by summing the instantaneous correlations using the basic proposed GC-VSS algorithm:

$$\begin{aligned} \bar{c}(n) &= \sum_{k=0}^{K-1} c(n-k) \\ &= \sum_{k=0}^{K-1} \left(\mathbf{g}(n-k) \cdot \bar{\mathbf{g}}(n-k-1) \right) \end{aligned} \tag{5.17}$$

In the latter case the long term average is more recent, so can be considered a better estimate of the trend in the direction of travel, especially in a non-stationary environment. Moreover, this way we obtain a new correlation measure on every iteration, rather than once per block, meaning we have more information on which to base the step size and we should be able to react faster to a change in the operating environment. The update equation becomes:

$$\mu'(n) = \alpha \mu(n-1) + \gamma \operatorname{sign}[\bar{c}(n)] \tag{5.18}$$

where the only difference from Modification 1 is that we smooth the gradient correlation before taking its sign. By de-noising the instantaneous correlation in this way, we should reduce the likelihood of obtaining sign values of -1 when there really is some underlying trend in the gradient estimates.

The effect of this modification is evident in Figure 5.6. Compared to the previous GC-VSS1, the step size has become larger when the misadjustment is high, reaching its maximum after the echo path change. This leads to faster and deeper convergence. The improvement in convergence behaviour comes despite lowering the gain factor γ to 0.0005. This in turn produces a smaller typical step size during double talk, which results in a 2 dB improvement in EERLE during double talk. In other words, Modification 2 has indirectly given us a way to reduce the noise in the step size updates, which as we have previously seen is beneficial in limiting the amount of EERLE degradation during double talk.

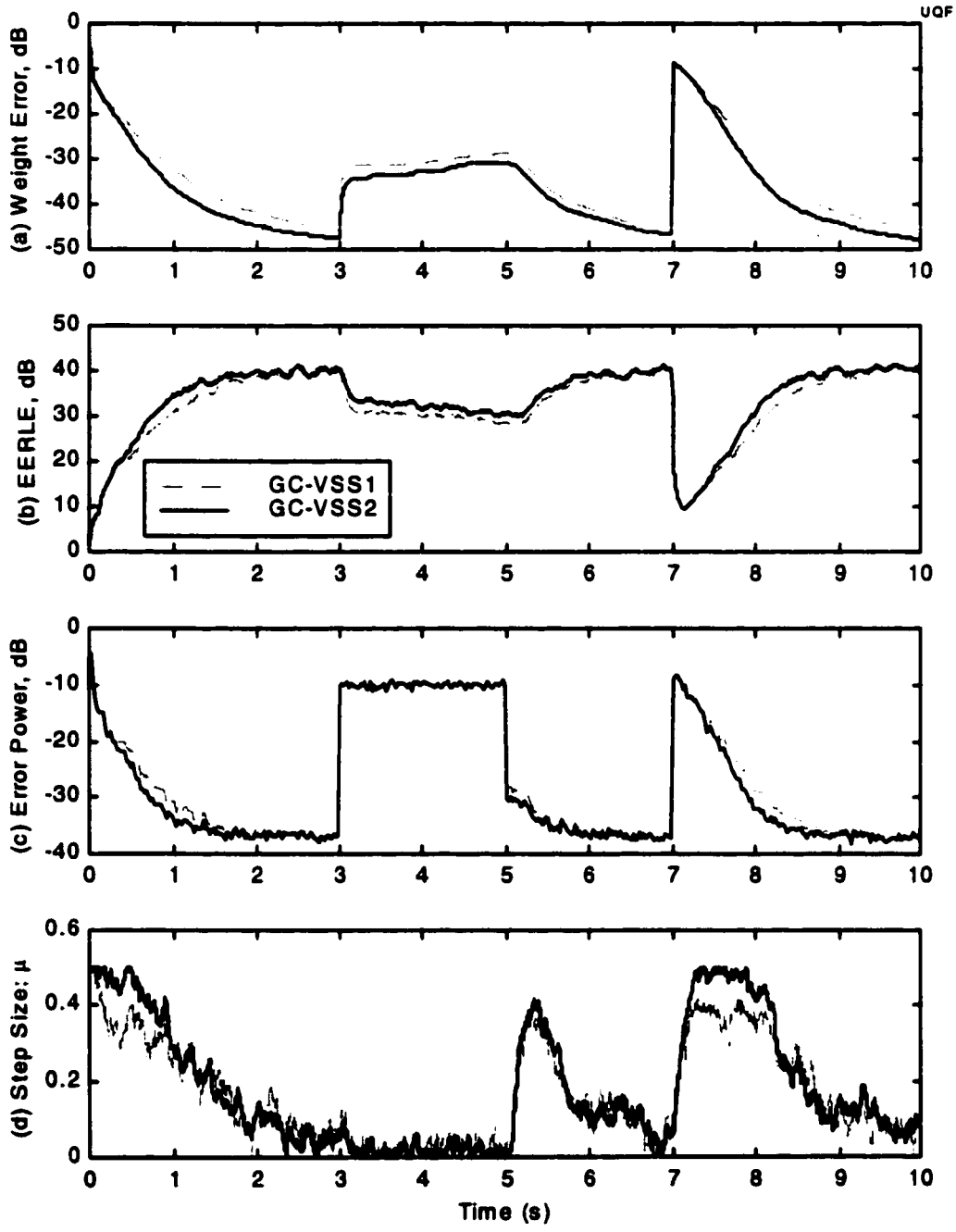

Figure 5.6 Further improvement in performance of GC-VSS algorithm by using Modification 2, with new values $\alpha = 0.9995$, $\gamma = 0.0005$, and additional parameter $K = 10$.

5.3.5 Modification 3: Dual Loop

As a final refinement, we now introduce another stage in the step-size update procedure. Here an intermediate variable $p(n)$ maintains the time-average of the gradient correlation, and it is this estimate that actually drives the step size:

$$p(n) = \beta\, p(n-1) + (1-\beta)\,\mathrm{sign}[\bar{c}(n)] \tag{5.19}$$

$$\mu'(n) = \alpha\,\mu(n-1) + \gamma\,\mathrm{sign}[p(n)]\,p^2(n) \tag{5.20}$$

where β is a new correlation smoothing parameter, providing an additional degree of freedom. This dual-loop approach is inspired by Mayyas's VSS procedure, where decoupling of the state estimation and step size update was found to provide more independent control of convergence speed and final MSE [May95a]. Furthermore, by squaring the estimate $p(n)$ in the step size update, the effect of strong correlations are amplified, and weaker correlations correspondingly have less influence. Note that taking the square removes the sign from the correlation estimate. We have to add this polarity back in by multiplying by sign[$p(n)$], because we want a negative correlation to decrease the step size, not increase it. Compared to Modification 2, this latest variant allows the instantaneous adjustments of the step size to lie over a wide dynamic range, rather than the fixed positive or negative increments, $\pm\gamma$. Finally, we note that $p(n)$ is implicitly bounded by [-1, +1] due to the sign operation in Eq. (5.19), so the step size update remains normalized.

Simulation results for this algorithm are shown in Figure 5.7. The EERLE curve is almost identical to the previous one, except during double talk when an additional 6 dB of echo cancellation is maintained. A related observation is that the EERLE of GC-VSS3 maintains a noticeable advantage over GC-VSS2 for almost 1 second after double talk ceases.

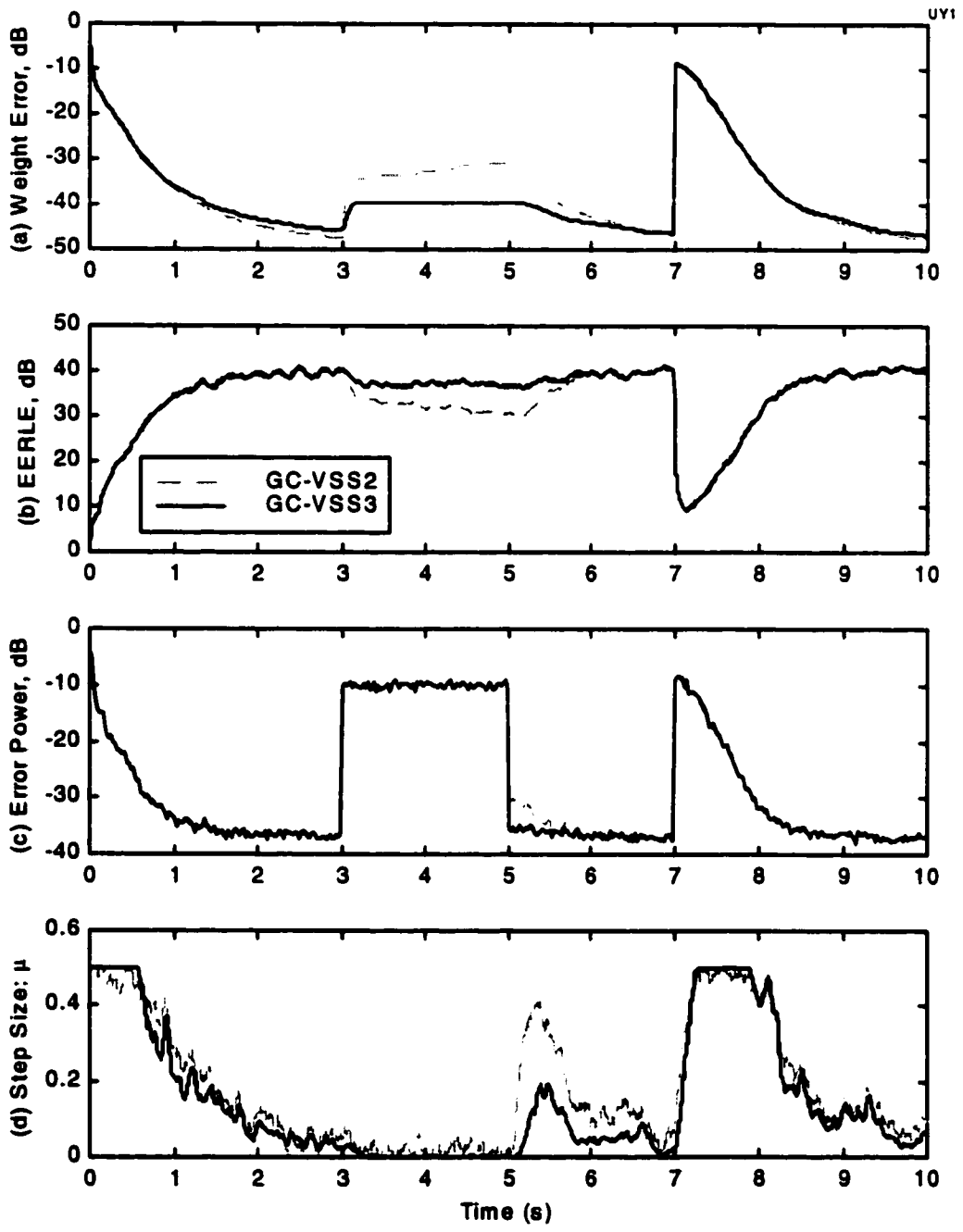

Figure 5.7 Further improvement in performance of GC-VSS algorithm by using Modification 3, with new values $\alpha = 0.99$, $\gamma = 0.02$, and additional parameter $\beta = 0.9995$.

The origin of this performance gain can be seen in the step size plot. The step size drops to essentially zero within 200 ms of the onset of double talk and stays there for its duration. This can be attributed to the use of the intermediate variable $p(n)$, which is allowed to have negative excursions even though the step size is clipped at zero. The largest separation in the two step size curves occurs between 5 and 6 seconds. This is simply due to the fact that with the proposed GC-VSS3 algorithm less correction of the filter weights is required after double talk ceases, so the step size does not need to grow very large before reconvergence.

5.3.6 Summary of Proposed GC-VSS3 Algorithm

We will refer to this latest algorithm as GC-VSS3 from now on. For convenience the steps, including all the incremental modifications, are presented in Table 5.1.

Some explanation is in order regarding the choice of initial conditions. The weight vector is initialized to all zeroes, since zero is the best blind estimate of each sample of the impulse response, which is equally likely to be positive or negative. In a real system, it may be possible to initialize the weights at the start of a phone call with the last known good set, i.e. the weights at the end of the previous connection, to get a "head start". The tap input vector is also initialized to zero. Not only does this make sense, as there is no far-end speech before a call is connected, but it helps to accelerate initial convergence — the normalized step size will be large when the input power is small. The step size μ and the smoothed correlation variable p are each set to their maximum value at the start, since it is assumed the weights are far from the optimum and we want to converge as quickly as possible. Double talk at the beginning of a conversation is rare, but even if it did occur then, there is little to lose at this point.

Table 5.1 GC-VSS3 Algorithm

Initialization: Set $\mathbf{w}(1) = \mathbf{0}$, $\mathbf{x}(0) = \mathbf{0}$, $\bar{\mathbf{g}}(0) = \mathbf{0}$, $\bar{c}(0) = 0$, $p(0) = 1$, $\mu(0) = \mu_{MAX}$

Computation: At each iteration $n = 1, 2, 3 \ldots$, perform the following steps.

Step 1: $e(n) = d(n) - \mathbf{w}^T(n)\,\mathbf{x}(n)$

Step 2: $\mathbf{g}(n) = e(n)\mathbf{x}(n)$

Step 3: $\bar{\mathbf{g}}(n) = \bar{\mathbf{g}}(n-1) + \mathbf{g}(n) - \mathbf{g}(n-B)$

Step 4: $c(n) = \mathbf{g}(n) \cdot \bar{\mathbf{g}}(n-1)$

Step 5: $\bar{c}(n) = \bar{c}(n-1) + c(n) - c(n-K)$

Step 6: $p(n) = \beta\, p(n-1) + (1-\beta)\,\mathrm{sign}[\bar{c}(n)]$

Step 7: $\mu'(n) = \alpha\,\mu(n-1) + \gamma\,\mathrm{sign}[p(n)]\,p^2(n)$

Step 8:
$$\mu(n) = \begin{cases} 0, & \mu'(n) < 0 \\ \mu_{MAX}, & \mu'(n) > \mu_{MAX} \\ \mu'(n), & \text{otherwise} \end{cases}$$

Step 9:
$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta}\,\mathbf{g}(n)$$

5.4 Performance Analysis of GC-VSS3 via Simulation

We have already shown some initial simulation results to illustrate the evolution of the proposed gradient correlation approach. In this section we investigate performance more thoroughly and methodically, using quantitative measurements to make comparisons with other algorithms and optimize parameter values.

5.4.1 General Simulation Configuration

We have developed a MATLAB-based simulation platform to allow us to experiment with different AEC algorithms, adjust their parameters, and measure performance in various environments. To make comparisons meaningful, the following system set up is used throughout this thesis, unless specifically stated otherwise.

- The system sampling rate is 8 kHz, and the simulation is run for 10 seconds, or 80,000 samples.
- The initial echo path **h** is characterized by the finite impulse response shown in the Appendix, with a length of $L = 1200$ samples. This response was recorded in a real conference room using an inexpensive speaker phone. It has been normalized to give an Echo Return Loss of 0 dB for a white noise input.
- The adaptive filter has an FIR transversal structure with $N = 1024$ weights. By using $N < L$, we incorporate some undermodelling into the simulation, making it more realistic.
- The weight vector and tap input vector are both initialized with zeroes: $\mathbf{w}(1) = \mathbf{0}$, $\mathbf{x}(0) = \mathbf{0}$.

- Double talk is simulated during the interval $t = [3, 5]$ seconds by applying a near-end signal at a level of 10 dB below the uncancelled echo. This is the maximum level expected for near-end speech in many AEC applications, and represents the worst case in terms of its potential to cause the weights to diverge.
- An abrupt echo path variation is simulated at time $t = 7$ seconds by switching to an alternate echo path, $\mathbf{h}'$, also shown in the Appendix. This modified echo path has been generated by applying an independent random offset to each of the samples in the original impulse response. The same $\mathbf{h}'$ is used in every simulation.

Having defined the system configuration, the remaining items to specify are the algorithm to use and the signals to apply in a given simulation. Later on, we will use recorded speech data to emulate the real operating environment for AEC. But initially we apply white noise for the far- and near-end signals. This provides a more consistent basis for the comparison of simulation results and in some ways yields better insight into algorithmic behaviour. So for the first set of simulations, the following conditions are used:

- The far-end signal $x(n)$ is white Gaussian noise, with zero mean and unit variance.
- The near-end signal $u(n)$ is white Gaussian noise, with zero mean and variance of 0.316 (i.e. 10 dB below the uncancelled echo).
- The background noise $v(n)$ added into the microphone signal is white Gaussian noise, with zero mean and variance of 0.01 (i.e. 40 dB below the uncancelled echo).

In order to have a reliable means of comparing performance, we calculate a number of the objective measures first defined in section 2.7. Specifically, the measures to be extracted from the simulation results are:

- $EERLE_{ST}$: the excess ERLE in single talk mode after initial convergence, measured from $t = 2$ to 3 seconds.
- $EERLE_{DT}$: the excess ERLE in double talk mode after any initial divergence, measured from $t = 4$ to 5 seconds.
- T_{IC} : time required for initial convergence to a weight error of -30 dB.
- T_{RPV} : time required for recovery after the abrupt echo path variation, again to a weight error level of -30 dB.
- T_{RDT} : time required for recovery after double talk ceases, again to a weight error level of -30 dB.

The EERLE is calculated *after* averaging the ratio components over a full window of one second. Note that weight error can be used as a measure of convergence since we have a white noise input, so all modes of the echo path system are excited. It is a smoother curve than the EERLE itself, so gives a more repeatable result from one simulation run to the next.

While we are primarily interested here in improving the performance during double talk, we must also ensure that the initial convergence and subsequent tracking of a changing echo path are not adversely affected. In a real AEC system, it be may be acceptable to allow ERLE to degrade a certain amount during DT, especially if improvements can be made in other areas, such as tracking ability. This can be tolerated, because as far as the far-end listener

is concerned, the near-end speech will mask part of the echo. Therefore, $EERLE_{DT}$ is not the sole criterion on which performance should be judged.

5.4.2 Experiment 1: Performance Comparison in White Noise

This experiment compares the quantitative performance of the proposed GC-VSS3 algorithm to the NLMS without double talk protection and to the patented Rohrs-Younce algorithm, which claims double talk resistance. The weight error is plotted versus time for each case in Figure 5.8. This measure of misadjustment provides the clearest way to compare a large set of simulation results on one plot, and as explained in section 2.7, is directly related to EERLE in this white noise environment.

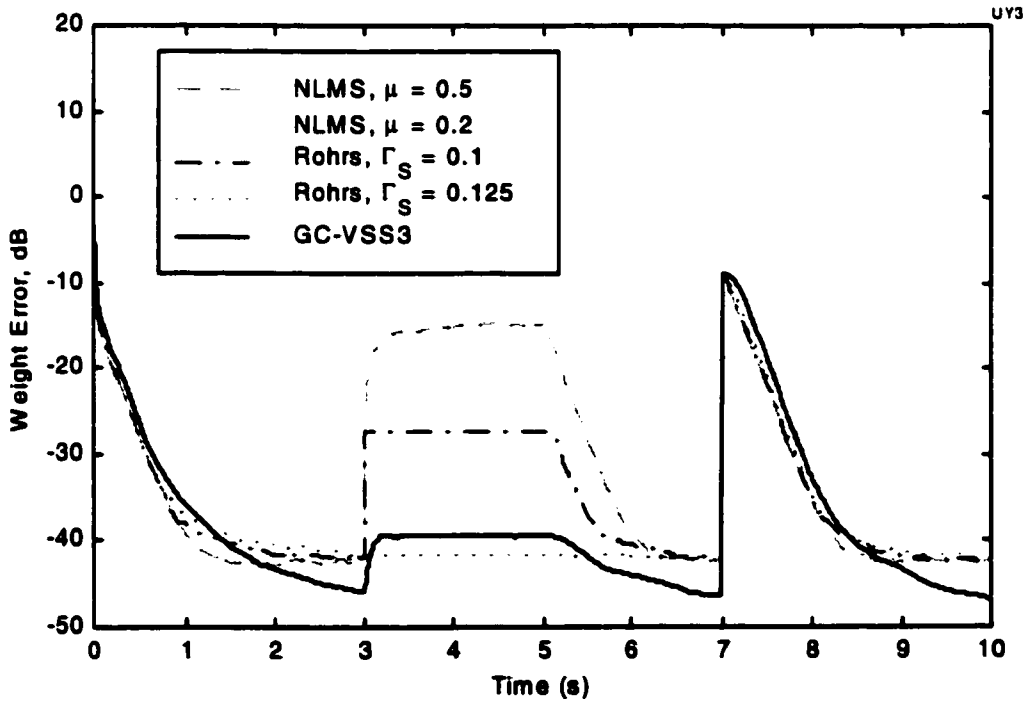

Figure 5.8 GC-VSS3 performance compared to NLMS and Rohrs-Younce algorithms, with selected parameter values.

The numerical results of Experiment 1 are listed in Table 5.2. For completeness, the performance numbers for the gradient correlation-based predecessors to GC-VSS3 are also provided in the table. These measures are extracted from the curves presented in section 5.3 and represent near-optimal performance for each algorithm variant, given a block size fixed at $B = 500$.

Algorithm	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RDT} (ms)	T_{RPV} (ms)
NLMS, $\mu = 0.5$	38.6	15.1	566	572	767
NLMS, $\mu = 0.2$	34.6	29.6	1917	42	935
Rohrs, $\Gamma_S = 0.125$	38.0	38.0	566	0	808
Rohrs, $\Gamma_S = 0.1$	38.4	27.3	566	202	777
GC-VSS	33.8	19.8	1052	777	1058
GC-VSS1	39.3	29.4	862	156	1040
GC-VSS2	39.9	31.3	666	0	867
GC-VSS3	39.5	37.2	633	0	885

Table 5.2 Performance of various algorithms with white noise inputs.

Two values for the fixed NLMS step size are tried. The value $\mu = 0.5$, equal to the μ_{MAX} used for GC-VSS, represents the case of the fastest convergence, but also the fastest divergence during double talk. In contrast, $\mu = 0.2$ represents the smallest step size that can still achieve -40 dB misadjustment at $t = 3$ seconds, just before the double talk starts. Using any smaller step size would give less meaningful results regarding the divergence during double talk. It may be surprising that the tabulated $EERLE_{ST}$ is better for $\mu = 0.5$ than $\mu = 0.2$,

but recall that this measure has been averaged over the interval from $t = 2$ to 3 seconds, during which time the $\mu = 0.2$ case was still converging. If more time had elapsed before the measurement, the steady state echo cancellation would indeed have been greater with $\mu = 0.2$ than $\mu = 0.5$.

These unprotected NLMS results provide a useful performance baseline. They represent the worst case in terms of double talk performance; in other words, this corresponds to the situation when a double talk detector misses the double talk, or is absent altogether. But at the same time the T_{IC} and T_{RPV} results indicate the *best case* performance in terms of tracking, since this algorithm will react immediately to an echo path change.

The parameters to be used for fair comparison of the Rohrs-Younce algorithm have been determined via a separate series of simulations (not shown here). The best parameter values for this particular set up, assuming a fixed step size of $\mu = 0.5$, were found to be: $\lambda = 0.998$, $\beta = 0.999$, and $\Gamma_S = 0.125$. The plot shows that in this case the Rohrs-Younce algorithm is successful in freezing the adaptive filter weights for the duration of the double talk. However, it has been found that this attractive behaviour is very sensitive to the choice of parameter values. To demonstrate this, another case was run using $\Gamma_S = 0.1$, and the results are shown in the table and figure for comparison. This minor change in the threshold level resulted in a huge difference in performance: a degradation of 10.7 dB in EERLE during double talk. This divergence occurred immediately at the onset of double talk, since the state estimation variable $\bar{s}(n)$ happened to be above the threshold at this time. This is a graphic example of what can take place with a non-robust fixed step size approach to AEC in the presence of double talk.

It is interesting to note that in almost all cases, the time for reconvergence after an echo path change is longer than for the initial convergence, even though the weight error at $t = 7$ seconds is smaller than that at 0 seconds. This can be explained by the input power normalization inherent in the NLMS algorithm. At the start of the simulation (which roughly corresponds to the start of a phone call) the tap input vector is pre-loaded with zero, so the normalization factor is small leading to large steps and fast convergence. This effect is amplified for large N, since it takes some time for the tap input vector to become filled with non-zero values. In contrast, when the echo path change occurs in the middle of the simulation the tap input power is large, so the actual normalized step size is much smaller.

The winner in terms of overall performance is the particular case of the Rohrs-Younce algorithm with $\Gamma_S = 0.125$. However, the GC-VSS3 is within 1 dB of the best $EERLE_{DT}$ performance, and only 70 to 80 ms slower to converge and reconverge — due to its inherent echo path change detection delay. Such a small degradation is probably negligible in a practical AEC implementation. What may be a more important differentiator in the real world is robustness to parameter variation. In the next section we demonstrate that the proposed GC-VSS3 algorithm does not exhibit the dramatic variation in performance seen with Rohrs-Younce when its parameter values are altered.

5.4.3 Experiment 2: Parameter Sensitivity

It may have been noticed that the GC-VSS3 algorithm has a relatively large number of parameters: B, K, α, γ, β, μ_{MAX}, and δ. Setting these to appropriate values to obtain achieve good performance is a concern. This section demonstrates the sensitivity of the GC-

VSS3 algorithm to each of its parameters via simulation. Using this information, some guidelines for selecting values are provided.

A multi-part experiment has been constructed to obtain insight into the algorithm's behaviour. The default parameter values used in the GC-VSS3 algorithm are given in Table 5.3. This parameter combination has been found to give good results in a white noise environment. In Experiment 2 we vary each of these one at a time, while keeping the remaining parameters fixed at their default values. The only exception to this is the regularization parameter which is maintained at 10 throughout. This value represents 1% of the typical power in the reference input signal. Besides its use in normalization, δ is also used as a threshold in a simple near-end speech detector: when the tap input power falls below this level, adaptation is frozen.

Description	Symbol	Value
block size for gradient average	B	500
window size for correlation	K	10
step-size smoothing factor	α	0.99
step size update gain factor	γ	0.02
correlation smoothing factor	β	0.9995
maximum (initial) step size	μ_{MAX}	0.5
regularization parameter	δ	10

Table 5.3 Default GC-VSS3 parameter values for Experiment 2.

Note that T_{RDT}, the recovery time after double talk, is not recorded here, since in almost all cases the GC-VSS3 algorithm prevents significant divergence during double talk. Using the -30 dB weight error level as the recovery target, effectively no time is required to recover.

In the first part of the experiment, the block size B is varied, with the results shown in Table 5.4. We observe that performance generally improves as the block size is increased. This is to be expected, as the block-averaged gradient estimate will contain less noise, and thus will be better aligned with the optimum, increasing the gradient correlation term. Of course, implementation cost[6] will also increase with B, so the optimum value will depend on finding a good balance between this cost and the performance benefits. In the case of Experiment 2(b) the gains achieved beyond $B = 500$ are minimal, so this is a reasonable block size to use. Based on additional experience from using other adaptive filter lengths, we make the empirical claim that a value of $B \approx N/2$ represents a good choice. However, it is comforting to note that the sacrifice in performance is quite small if the block size is reduced below this recommended level, which may be expedient to conserve MIPS and memory.

The sensitivity to variations in window size K parallels that of block size B in many ways. As K increases, the EERLE in single talk rises and the convergence times are reduced, although negligible gains are observed beyond $K = 20$. The most interesting observation from Table 5.5 is that the EERLE during double talk reaches a peak for $K = 10$, then subsequently degrades. The explanation for this is that using a longer window introduces a larger echo path change detection delay. The step size, while small, remains above zero for longer, so the

---

[6] This matter will be discussed in more detail in section 5.7.

B	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
100	36.8	32.4	1195	1252
300	39.2	36.5	675	940
500	39.5	37.2	634	886
700	39.7	37.0	631	879
1000	39.8	38.3	631	873

Table 5.4 Performance sensitivity to block size B for Experiment 2(a).

weights have more time to diverge before they are effectively frozen. The bulk of this divergence happens within 100 ms of the onset of double talk.

The $K = 1$ case represents the situation *without* any time averaging of the correlation before taking its sign. The results show that roughly 4 dB of improvement in the EERLE measures and roughly 50% reduction in convergence time are obtained by applying the appropriate amount of averaging (as introduced in Modification 2).

K	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
1	35.0	33.6	1395	1508
5	39.0	36.8	697	1034
10	39.5	37.2	634	886
20	39.8	35.0	631	856
30	39.8	33.4	631	841

Table 5.5 Performance sensitivity to window size K for Experiment 2(b).

In Experiment 2(c) we find the performance versus step-size smoothing factor α follows the same trend as for K. The one measure that does not improve asymptotically with larger α is the EERLE during double talk. A reasonable balance is obtained around $\alpha = 0.99$, though performance remains decent over a considerable range of this parameter. With γ and other parameters constant, a larger α leads to a larger step size, which speeds up convergence thereby improving the EERLE measured from 2 to 3 seconds, but also causes faster divergence during double talk. In the latter case, the step size cannot be reduced quite as quickly due to the memory built into it by a large α.

α	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
0.98	39.0	36.8	775	1100
0.985	39.3	37.3	685	985
0.99	39.5	37.2	634	886
0.993	39.7	36.7	631	869
0.995	39.8	35.9	631	856

Table 5.6 Performance sensitivity to step-size smoothing factor α for Experiment 2(c).

In general, optimal values for this parameter will depend on convergence speed, which in turn depends on the adaptive filter size N. For example, with a much smaller $N = 10$, convergence could be expected within 100 iterations, so α should be reduced to allow the step size to become small faster. A rule of thumb is to set

$$\alpha = 1 - (10N)^{-0.5} \tag{5.21}$$

For $N = 1024$ this formula yields $\alpha = 0.99$, while for $N = 10$ the recommended value is $\alpha \approx 0.9$.

Increasing the step-size update gain factor in Experiment 2(d) generally yields an improvement in single talk EERLE and convergence times. But $EERLE_{DT}$ drops off slightly after peaking at $\gamma = 0.015$. Once again this is related to the higher steady-state step size, which leads to faster initial divergence at the onset of double talk. Ideally the gain factor should be set as large as is possible without seriously degrading double talk performance due to false detection of echo path change. We see that $\gamma = 0.02$ represents a good tradeoff under these simulation conditions.

γ	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
0.010	39.0	36.8	773	1100
0.015	39.3	37.4	656	943
0.020	39.5	37.2	634	886
0.025	39.6	36.9	631	873
0.030	39.7	36.5	631	861

Table 5.7 Performance sensitivity to step-size update gain factor γ for Experiment 2(d).

Table 5.8 shows that as the correlation smoothing factor β is increased, performance during initial convergence and double talk is improved. The downside is that recovery time after echo path variation actually degrades. The reason behind the latter effect is that with a longer correlation memory, it takes more time for the algorithm to react to the strong

gradient correlation that results from the echo path change. Note that $\beta = 1$ would represent the infinite memory case, which might be fine for a static situation, but becomes a handicap in a non-stationary environment — incidentally this value would also create overflow problems in a real implementation. A practical guideline for setting this parameter is:

$$\beta = 1 - 1/N \tag{5.22}$$

though this may be tweaked for better performance in specific situations.

β	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
0.998	39.4	35.5	695	826
0.999	39.4	36.7	664	836
0.9993	39.5	36.9	645	858
0.9995	39.5	37.2	634	886
0.9998	39.9	37.5	631	1038

Table 5.8 Performance sensitivity to correlation smoothing factor β for Experiment 2(e).

The final algorithm parameter to be varied in Experiment 2 is the maximum step size. The results in Table 5.9 reveal that the main difference in performance is in the initial convergence time. This is to be expected, since we always set the initial step size to the maximum, and in typical operation the step size remains at this upper limit until the weights have at least partially converged. When the limit is higher, the weights should converge faster (up to a point), which matches what we observe. In this particular experiment, any $\mu_{MAX} > 0.5$ provides minimal improvement in recovery time after path variation. This is

because the system is already reconverging by the time the rising step size reaches this level, so it immediately begins to shrink again. Unless there are much larger path variations in a given speakerphone set-up, we can safely say that the choice of μ_{MAX} is primarily driven by initial convergence speed. There is less than 0.5 dB difference in $EERLE_{ST}$ performance over the range of μ_{MAX} tested. The greater variation in cancellation level during double talk may be simply due to a higher level of the step size just as the double talk began.

μ_{MAX}	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RPV} (ms)
0.4	39.4	36.7	869	1023
0.5	39.5	37.2	634	886
0.6	39.7	37.3	491	850
0.8	39.8	35.7	404	846
1.0	39.8	35.3	414	844

Table 5.9 Performance sensitivity to maximum step size μ_{MAX} for Experiment 2(f). Note that the initial step size $\mu_0 = \mu_{MAX}$ in each case.

5.4.4 Experiment 3(a): Performance in Coloured Noise

As a step towards using real speech for the input signals, we next conduct an experiment where the far-end input remains stationary, but is quite highly autocorrelated. Coloured noise is obtained by applying a zero-mean white Gaussian noise $\eta(n)$ as the input to the simple one-pole IIR filter described by[7]:

[7] This is similar to the colouring filter used by [May95a, Kwo92], but with a higher low-pass cutoff frequency.

$$x(n) = 0.7x(n-1) + \eta(n) \tag{5.23}$$

This type of input is known to produce a non-uniform MSE surface with very elongated elliptical contours. This can cause convergence problems for gradient-based algorithms, since they can spend a lot of time moving perpendicular to the contours without making much progress towards the optimum.

Indeed the results shown in Figure 5.9 are rather disappointing. The GC-VSS algorithm only reaches an EERLE of about 13 dB before its step size rapidly drops to zero. Further analysis has revealed that at this time the smoothed correlation $c(n)$ is actually strongly negative.

If there is any consolation, it may be that the Rohrs-Younce algorithm fared even worse in this experiment, reaching an echo cancellation level of only 7 dB. Numerical results for both these algorithms, plus the LMS benchmark, are shown in Table 5.11 a few pages ahead.

This poor performance should probably not come as a big surprise. NLMS performance is notoriously affected by correlated inputs. This is usually addressed in practice by using a more sophisticated technique, which is how we will proceed in the next section.

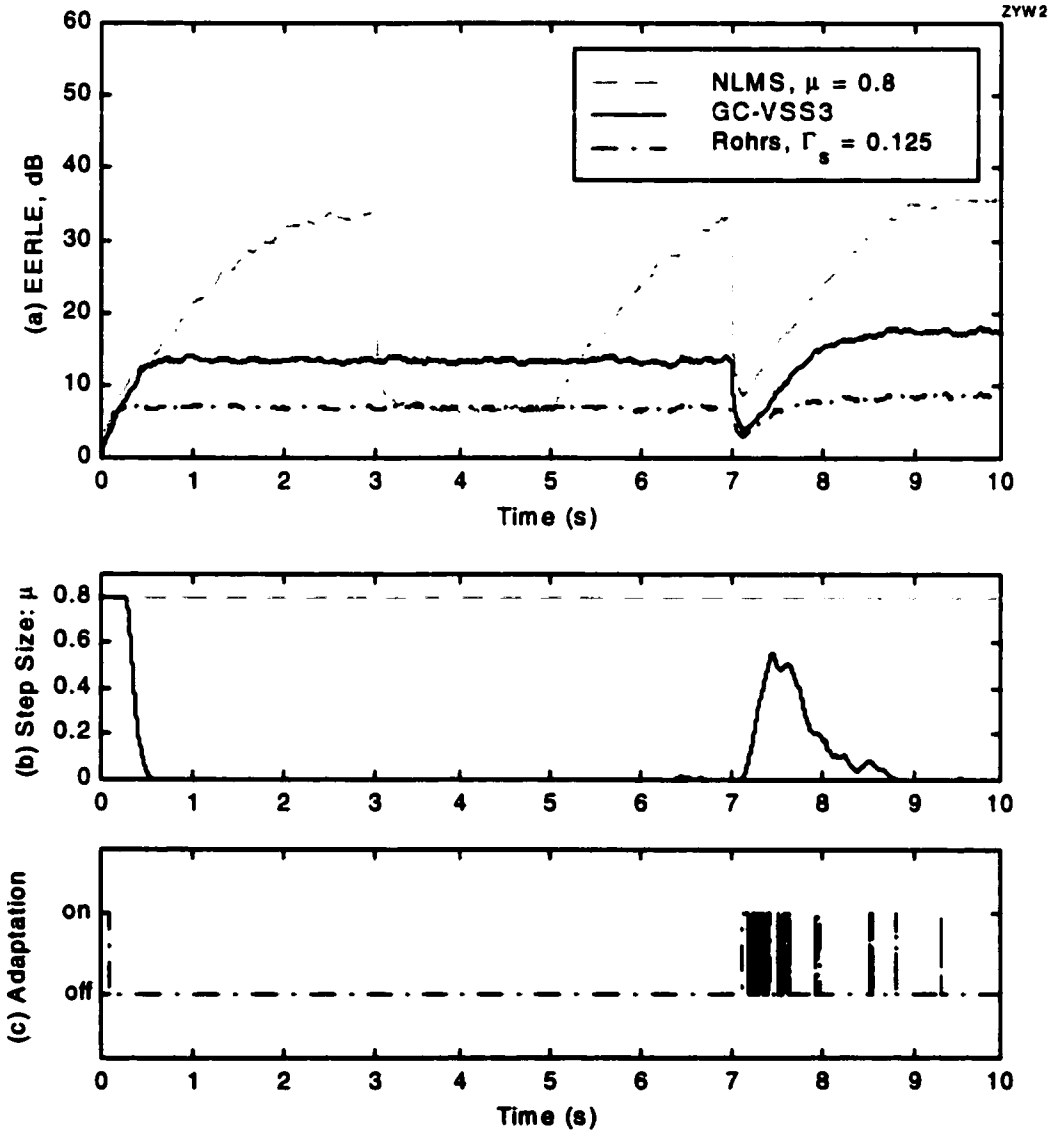

Figure 5.9 GC-VSS3 performance with a coloured noise input, compared to NLMS (μ = 0.8) and Rohrs-Younce (Γ_S = 0.125). Plot (c) shows how adaptation switches on and off for the Rohrs-Younce case.

5.5 Projection Correlation VSS Algorithm

There is no reason why the proposed gradient correlation VSS approach should be limited in application to an NLMS algorithm. We choose now to investigate how it could work with the Affine Projection algorithm. Since the weights updates in this case do not strictly follow the gradient itself, but a projection thereof, it is more appropriate to call it a *Projection* Correlation VSS algorithm, or PC-VSS for short.

5.5.1 PC-VSS Formulation

The proposed PC-VSS algorithm is outlined in Table 5.10. The main difference from the GC-VSS3 algorithm lies in the first three steps, which embody the Affine Projection algorithm as described in section 3.9. We retain the notation $\mathbf{g}(n)$ but now it denotes the projection, not the gradient. It is effectively normalized in step 2, so we no longer see the normalization in the weight update equation, step 10. The initialization is essentially the same as for the NLMS-based version.

5.5.2 Experiment 3(b): Performance in Coloured Noise

We are now in a position to rerun Experiment 3 with the new algorithm. Results are provided in Figure 5.10 for the proposed PC-VSS with parameter values α = 0.995, β = 0.9998, γ = 0.005, B = 1000, K = 20, and μ_{MAX} = 0.5. This is compared to the AP with a fixed step size of 0.2, chosen as a reasonable tradeoff between convergence speed and misadjustment during double talk. Both algorithms are of projection order P = 5. Overall results for Experiment 3 including part (a), are presented in Table 5.11. Note that in this table and the one that follows, the criterion for convergence in measuring the adaptation speed and

Table 5.10 PC-VSS Algorithm

Initialization: Set $\mathbf{w}(1) = \mathbf{0}$, $\mathbf{X}(0) = \mathbf{0}$, $\bar{\mathbf{g}}(0) = \mathbf{0}$, $\bar{c}(0) = 0$, $p(0) = 1$, $\mu(0) = \mu_{MAX}$

Computation: At each iteration $n = 1, 2, 3 \ldots$, perform the following steps:

Step 1: $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\,\mathbf{w}(n)$

Step 2: $\boldsymbol{\varepsilon}(n) = [\,\mathbf{X}^T(n)\mathbf{X}(n) + \delta\mathbf{I}\,]^{-1}\mathbf{e}(n)$

Step 3: $\mathbf{g}(n) = \mathbf{X}(n)\,\boldsymbol{\varepsilon}(n)$

Step 4: $\bar{\mathbf{g}}(n) = \bar{\mathbf{g}}(n-1) + \mathbf{g}(n) - \mathbf{g}(n-B)$

Step 5: $c(n) = \mathbf{g}(n) \cdot \bar{\mathbf{g}}(n-1)$

Step 6: $\bar{c}(n) = \bar{c}(n-1) + c(n) - c(n-K)$

Step 7: $p(n) = \beta\, p(n-1) + (1-\beta)\,\mathrm{sign}[\bar{c}(n)]$

Step 8: $\mu'(n) = \alpha\,\mu(n-1) + \gamma\,\mathrm{sign}[p(n)]\,p^2(n)$

Step 9:
$$\mu(n) = \begin{cases} 0, & \mu'(n) < 0 \\ \mu_{MAX}, & \mu'(n) > \mu_{MAX} \\ \mu'(n), & \text{otherwise} \end{cases}$$

Step 10: $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)\,\mathbf{g}(n)$

recovery times is now set at 25 dB smoothed EERLE (instead of -30 dB weight error). This is because (i) weight error is less meaningful now than in a white noise environment, and (ii) a level of 30 dB EERLE is a lot harder to obtain in the non-white situation, so this is relaxed slightly.

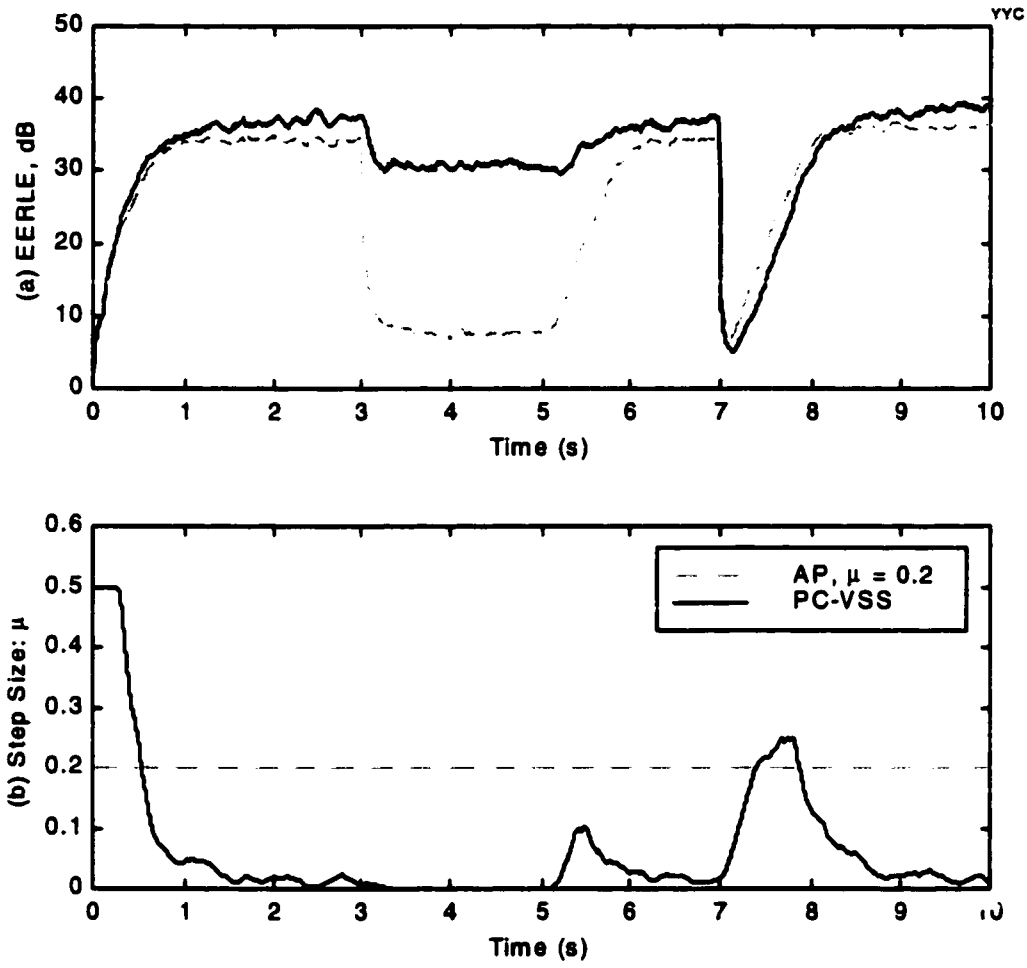

Figure 5.10 Improved performance in coloured noise by using projection algorithms (both of order $P = 5$). Fixed step size AP ($\mu = 0.2$) and proposed PC-VSS are shown to give similar performance in single talk, but the latter fares much better during double talk.

We see that the PC-VSS is able to overcome the convergence problems of GC-VSS3 in this type of signal environment. It outperforms AP in the initial convergence phase because its initial step size is higher than the AP's fixed step size. However, reconvergence after the echo path change is a little slower than AP due to the delay in ramping up the variable step size. Single talk EERLE is 3 dB better than AP, and this improvement margin grows to 23 dB

Algorithm	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RDT} (ms)	T_{RPV} (ms)
NLMS, $\mu = 0.8$	33.1	6.6	1430	1148	1188
Rohrs, $\Gamma_S = 0.125$	6.9	6.9	—	—	—
GC-VSS3	13.4	13.3	—	—	—
AP, $P = 5, \mu = 0.2$	34.2	7.7	569	795	812
PC-VSS	37.2	30.8	522	0	958

Table 5.11 Performance of various algorithms with coloured noise inputs.

during double talk. These results are encouraging as we approach the most important experiment of all.

5.5.3 Experiment 4: Performance with Real Speech

Now we are ready to apply input signals that are more realistic for acoustic echo cancellation: human speech that has been digitally recorded. The signals used in this experiment are displayed and characterized in the Appendix. Recordings from two different voices are used for the near- and far-end signals. Their RMS amplitude levels have been normalized to unity, to make the comparison with the noise inputs more valid. The performance of the various candidate algorithms is compared in Figures 5.11 and 5.12, while numerical results are given in Table 5.12.

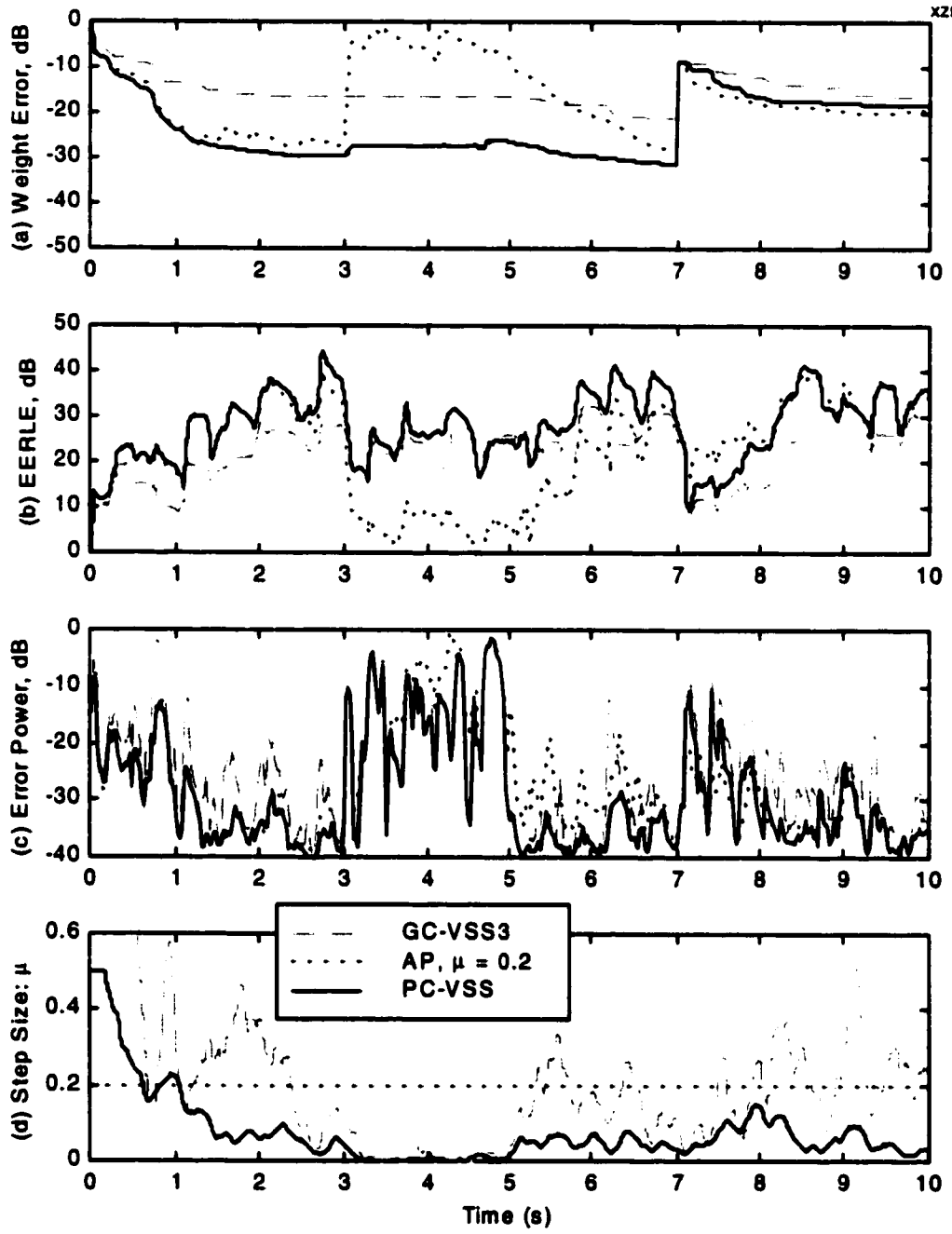

Figure 5.11 Comparison of different algorithms' performance with speech signals, showing the superiority of proposed PC-VSS. Both projection algorithms are of order $P = 5$. The AP case is for a fixed step size $\mu = 0.2$.

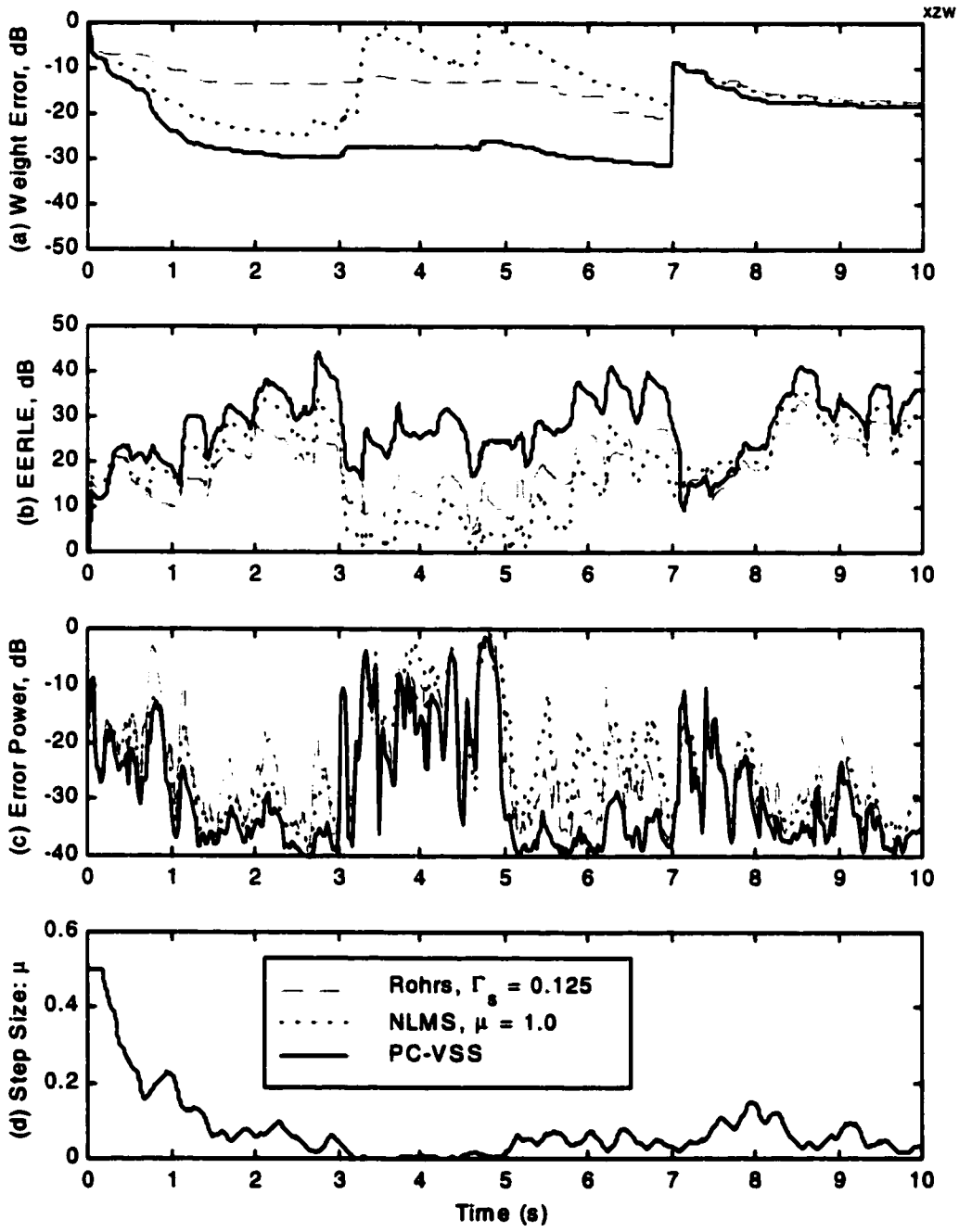

Figure 5.12 Performance of NLMS and Rohrs-Younce algorithms with speech signals. PC-VSS is shown again as a reference.

Algorithm	$EERLE_{ST}$ (dB)	$EERLE_{DT}$ (dB)	T_{IC} (ms)	T_{RDT} (ms)	T_{RPV} (ms)
NLMS, $\mu = 1.0$	30.5	5.7	1625	1228	1258
Rohrs, $\Gamma_S = 0.125$	24.7	12.3	1987	854	1268
GC-VSS3	26.7	23.8	1969	0	1444
AP, $P = 5, \mu = 0.2$	34.6	5.9	1145	1201	782
PC-VSS	36.4	26.6	1134	352	1106

Table 5.12 Performance of various algorithms with speech inputs.

PC-VSS clearly outshines the others in all categories, except one: the AP provides faster recovery after echo path change because it has a constantly high step size and thus reacts very quickly to changes in the impulse response. GC-VSS3 may appear from the numerical results to be better at recovering after double talk, but this is simply an anomaly in the data. Looking at the EERLE curve in Figure 5.11, we see that the PC-VSS is almost always superior to the GC-VSS3, except for a short window around 5 seconds where this measurement was made.

The degradation of 10 dB in EERLE from single talk to double talk seems high compared to the results seen previously for a stationary environment. However, this is still a big improvement on the competing Rohrs-Younce algorithm, or versus the AP with no double talk protection. Interestingly, the GC-VSS3 algorithm performs reasonably well in a speech environment, despite its poor performance seen in the coloured noise case.

5.5.4 Experiment 5: Simultaneous Double Talk and Echo Path Change

The preceding experiments have studied algorithm performance in situations of double talk and echo path change, but treated these as distinct events. In a real system, we cannot

expect the modes of operation to be so well behaved. Thus we now consider the special case where double talk and echo path change occur at the same time. Specifically, we move the echo path change of the previous simulations back in time from $t = 7$ seconds to $t = 4$ seconds, right in the middle of the double talk period. Otherwise, the experimental conditions are identical to Experiment 4. The resultant performance curves for the PC-VSS algorithm are shown in Figure 5.13. For comparison, the case without any double talk or echo path change is also shown in the background.

The results are identical to those of Experiment 4 up until $t = 4$ seconds, as would be expected. The step size is essentially zero at this time. Then when the echo path change occurs during the double talk, the weight error jumps, reflecting the increased distance between the weights and the new optimum. Rather than stay frozen at this value though, we see a reconvergence taking place immediately. In other words, the PC-VSS algorithm permits *beneficial* adaptation during double talk if the conditions are right for this. Note that the step size starts increasing slightly at 4 seconds, but then decreases once the new steady state misadjustment level is reached. Once the double talk ceases at $t = 5$ seconds the step size grows larger still, as the conditions are now right for further convergence.

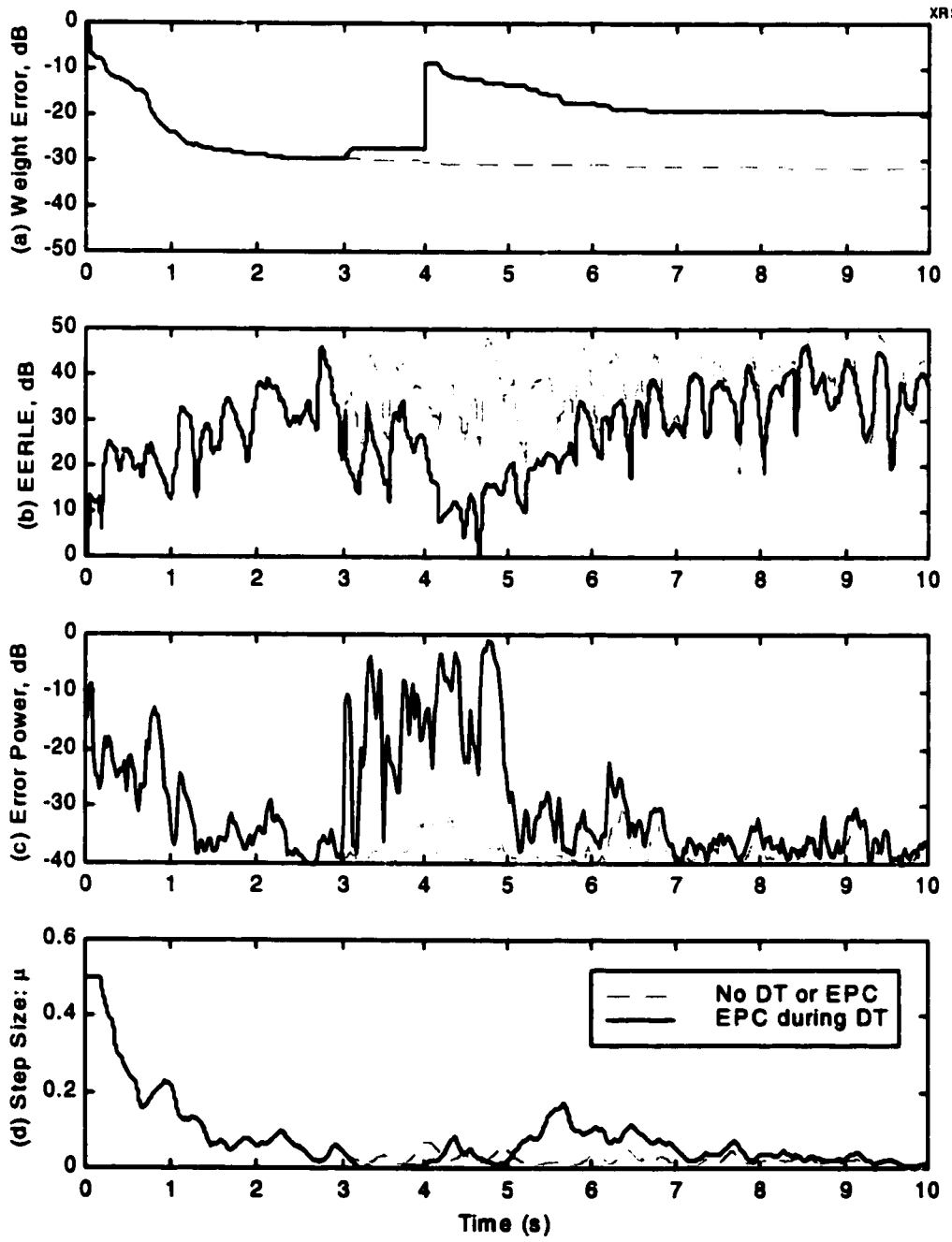

Figure 5.13 Simulation of an echo path change (at 4 seconds) in the middle of double talk (from 3 to 5 seconds) with the proposed PC-VSS algorithm. The case with only single talk and a static echo path is presented for comparison.

5.6 Auxiliary Double Talk Detection

Echo path change detection, as employed in the proposed GC-VSS algorithms, has been shown to offer a clear performance advantage over direct double talk detection. However, considering the overall acoustic echo control system, there is a potential weakness in this approach: the lack of a mechanism to control the variable losses and non-linear processor. These components will probably still be necessary, despite the improvement in echo cancellation performance, due to the high degree of echo suppression required in a hands-free telephone. Uniike the echo canceller itself, these signal processing blocks *do* need a double talk detector to activate them correctly.

We have stressed throughout this thesis that double talk detection is a difficult problem for AEC. It is important to note, though, that DTD for residual echo suppression control has different design constraints than DTD for suspension of echo cancellation. Most significantly, a longer detection delay can be tolerated, but the acceptability of false hits is reduced, since they cause unwanted modulation of the signal and background noise level heard by the far-end user.

Careful inspection of Figure 5.11 provides the following observable condition for the presence of double talk: large MSE *and* small step size, simultaneously. In order to use this for DTD, we must somehow convert this qualitative description into a mathematical expression. A simple hard-decision DTD could use two thresholds, declaring DT when both of these conditions are satisfied:

$$p_e(n) > \Gamma_e \quad \text{and} \quad \mu(n) < \Gamma_\mu \tag{5.24}$$

Note that large MSE alone could be an indication of an echo path change, but in such a case the step size will increase. This will negate the second part of the condition in Eq. (5.24), so DT will not be signalled. A small hold-off time T_{HOLD} is necessary to avoid false DTD initially after an echo path variation, before the step size has had time to grow (i.e. due to the echo path change detection delay). Furthermore, a hangover time T_{HANG} is desirable to avoid unnecessary switching in and out of DT mode when the MSE goes low, such as the short silent periods between syllables of speech.

To demonstrate this technique, we apply it to the data obtained from Experiment 4. Figure 5.14 shows the logic used to make the decision about whether double talk is present. In plot (c) the *raw* DTD signal represents the logical AND of the two binary components above it. Finally, the *conditioned* DTD signal shows the improvement once a holdoff time of 50 ms and a hangover time of 100 ms are factored in. Recalling that we applied double talk in the interval $t = [3, 5]$ seconds, we note that the detection is successful in this case. The DTD reacted within 150 ms of the actual onset of double talk, and turned off about 100 ms after the near-end speech disappeared. No unwanted switching is observed over the 10 seconds of the simulation.

All the experiments up to this point have fixed the level of the near-end speech at 10 dB below the uncancelled echo. As mentioned previously, this represents highest level that can be expected in an inexpensive speakerphone, so it is the worst case in terms of disturbing the adaptive filter weights. Ironically this -10 dB level is the *best* case for double talk detection. To show that the auxiliary DTD presented here can still work well under less optimal conditions, the near-end speech level is now dropped by another 10 dB (i.e. 20 dB below the

uncancelled echo), with the results shown in Figure 5.15. The algorithm parameters are unchanged from the previous case. Note that the conditioned DTD signal responds when it should for the most part, but the detector does switch off twice during the interval of interest. These gaps correspond to momentary quiet spots between syllables in the near-end speech. One potential way to eliminate them would be to adjust the threshold levels slightly, but excessive reliance on tweaking is undesirable. A more robust approach would be to increase the hangover time by a few hundred milliseconds. This can be done without much detrimental effect, since the DTD is no longer used to freeze the AEC adaptation.

In conclusion, the use of the PC-VSS algorithm's state variable, namely its step size, proves to be an invaluable means of determining whether double talk is present. A more sophisticated technique could provide a gradually variable control of the losses and NLP, depending on the amount of double talk. Application of fuzzy logic methods comes to mind, but is beyond the scope of this thesis.

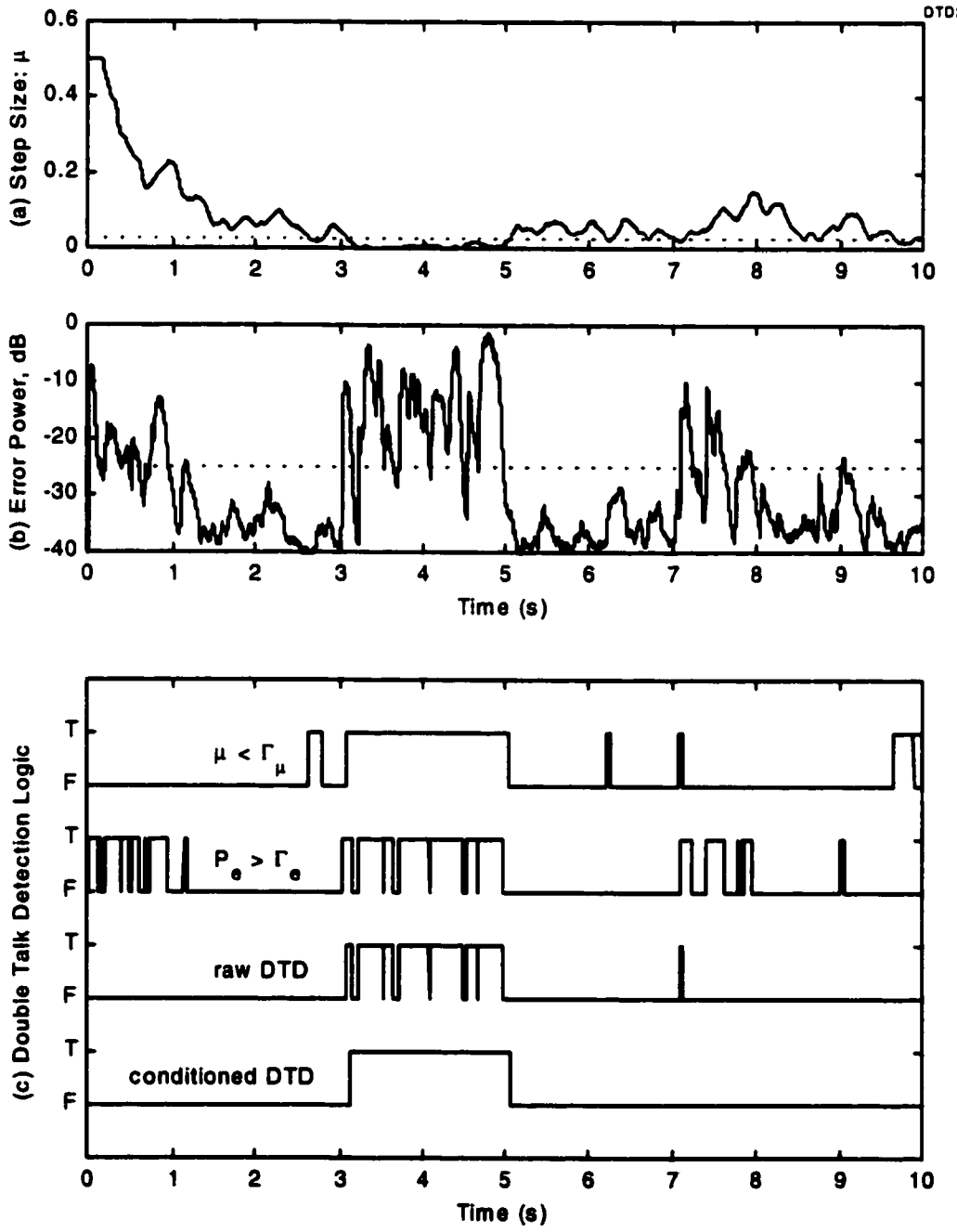

Figure 5.14 Auxiliary double talk detection for Experiment 4 with parameters $\Gamma_\mu = 0.025$, $\Gamma_e = -25$ dB, $T_{HOLD} = 50$ ms, $T_{HANG} = 100$ ms. The dotted lines indicate the respective threshold levels.

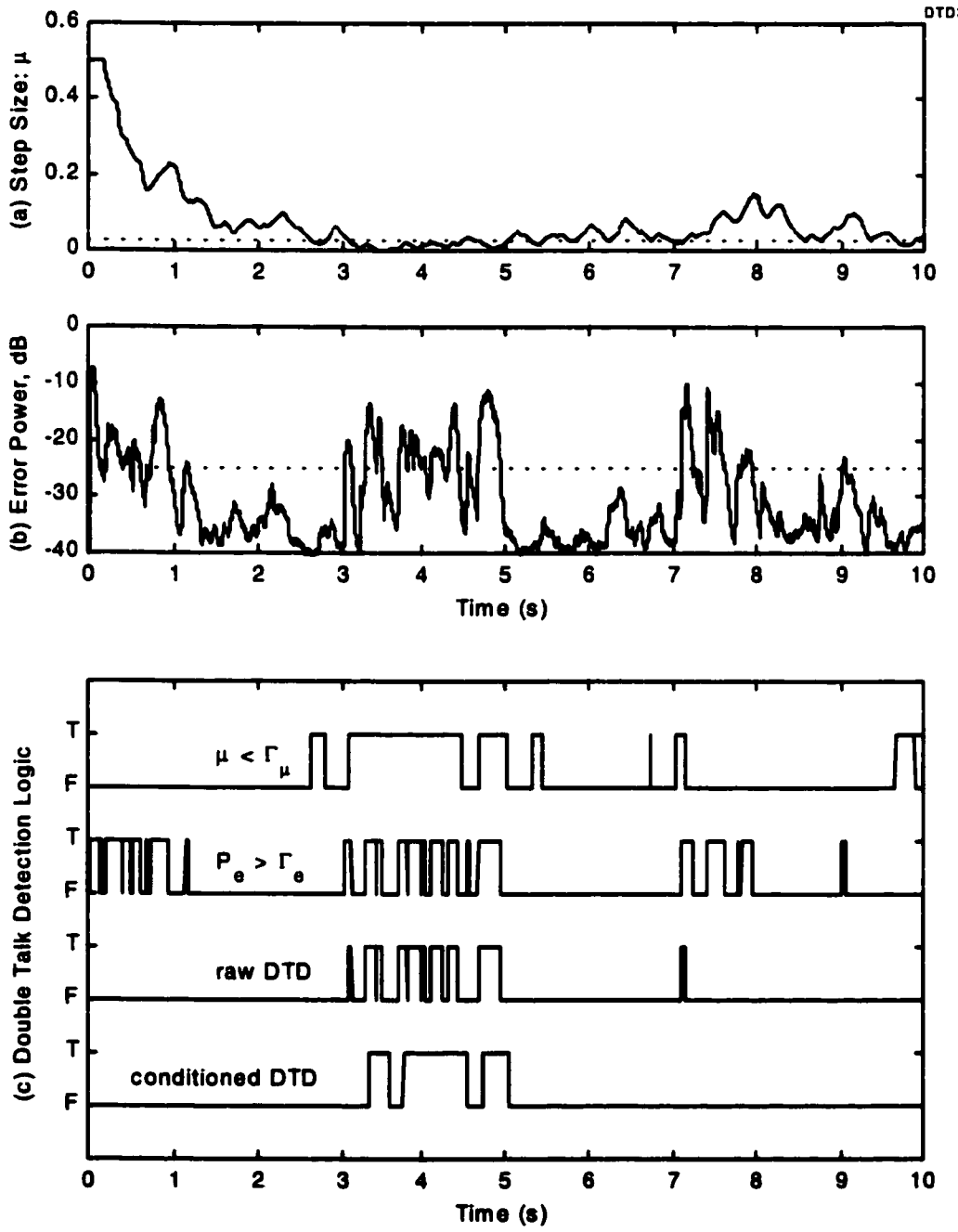

Figure 5.15 Auxiliary double talk detection with near-end speech level reduced by 10 dB compared to the previous case.

5.7 Computational Complexity

The LMS complexity of roughly $2N$ operations per input sample is the benchmark by which other algorithms are usually judged. In general, variable step size algorithms add a small amount of overhead, but this is usually insignificant when N is large. It must be admitted, however, that the VSS approach presented here is much more complicated than most. Calculation of the vector dot product alone would take N operations. In this section we show how to reduce the computational complexity of the proposed GC-VSS3 algorithm to a reasonable level, by exploiting the redundancy in consecutive tap input vectors.

The key to the savings lies in simplifying the correlation between the instantaneous and average gradient estimates as follows:

$$\begin{aligned}
\mathbf{g}(n)\cdot\bar{\mathbf{g}}(n-1) &= \left[e(n)\,\mathbf{x}(n)\right]\cdot\left[\sum_{b=0}^{B-1} e(n-b-1)\,\mathbf{x}(n-b-1)\right] \\
&= \left[e(n)\,\mathbf{x}(n)\right]\cdot\left[\sum_{b=1}^{B} e(n-b)\,\mathbf{x}(n-b)\right] \\
&= \sum_{i=0}^{N-1} e(n)\,x(n-i)\sum_{b=1}^{B} e(n-b)\,x(n-b-i) \\
&= e(n)\sum_{b=1}^{B} e(n-b)\sum_{i=0}^{N-1} x(n-i)\,x(n-b-i) \\
&= e(n)\sum_{b=1}^{B} e(n-b)\,\chi_b(n)
\end{aligned} \tag{5.25}$$

where $\chi_b(n)$ is the input autocorrelation for the b^{th} lag, windowed over the length of the adaptive filter. Each of these scalar variables can be calculated recursively as:

$$\chi_b(n) = \chi_b(n-1) + x(n)\,x(n-b) - x(n-N)\,x(n-N-b) \tag{5.26}$$

for $b = 1, 2, \ldots B$. There are two options for obtaining the last term on the right:

Option (i). Since this product has already been calculated N iterations previously (see the second term), it could be stored then and later retrieved. This option requires a memory of NB such terms, which could be excessive for the large N, B in AEC. However, this may be quite reasonable for other applications.

Option (ii). The product could be recalculated from stored values of its components. Then memory of only $N+B$ terms of $x(n)$ is required. We already store the first N for the tap input vector $\mathbf{x}(n)$, so this means an additional B memory locations are needed.

Thus overall calculation of $\mathbf{g}(n)\cdot\tilde{\mathbf{g}}(n-1)$ requires $2B+1$ multiplications for option (i) or $3B+1$ multiplications for option (ii). Choice of one of the two options represents a trade-off between computational complexity and storage space.

Note that it is conceivable to have a compromise option somewhere between these two extremes, where the product terms are stored for some values of $1 < b < B$, and the remainder are recalculated. This might be of interest in an implementation with tight memory and real-time constraints, where neither option (i) or (ii) is feasible for a desired combination of N, B. However this mixed method would increase the implementational complexity.

This simplified approach suggested in Eqs. (5.25) and (5.26) clearly avoids the N multiplications and additions required to calculate the full dot product $\mathbf{g}(n)\cdot\tilde{\mathbf{g}}(n)$. But the majority of the computational savings come from the fact that we do not need to calculate either $\mathbf{g}(n)$ or $\tilde{\mathbf{g}}(n)$ directly on each iteration. This saves at least $3N$ additional operations per iteration.

We note that the power in the tap input vector, used to normalize the step size, is given by the recursive autocorrelation expression of Eq. (5.26) with a lag of *zero*:

$$\mathbf{x}^T(n)\mathbf{x}(n) = \chi_0(n) = \chi_0(n-1) + x^2(n) - x^2(n-N) \tag{5.27}$$

Thus for the weight update equation we need to calculate the following scalar only once per iteration before multiplying by $\mathbf{x}(n)$:

$$\frac{\mu(n)e(n)}{\chi_0(n) + \delta} \tag{5.28}$$

In fact, by initializing $\chi_0(0) = \delta$ at the beginning, we can build the regularization parameter into the normalization factor and avoid doing the constant addition in the denominator on every iteration.

This *fast exact* version of the GC-VSS3 algorithm is summarized in Table 5.13. The overall complexity is shown to be $2N+3B+26$ DSP operations per input sample, where the worst case option (ii) is assumed for calculating step 2. We count the sign operations and the comparisons in step 7 as one operation each. The division in step 8 is counted as equivalent to ten operations, which is believed to be adequate given the low precision required of the normalized step size. Using representative parameter values of $N = 1024$, $B = 500$ and a sampling rate $f_s = 8$ kHz, this works out to a total of 28.6 MIPS, which is well within the capabilities of today's inexpensive programmable DSP chips.

We can show that as an incremental addition to the basic Affine Projection, the PC-VSS does not greatly increase the computational complexity. Referring back to Table 5.10, steps 1 to 3 and 10 represent the underlying AP algorithm, which was earlier quoted as costing $2PN+7P^2$ calculations per input sample. The additional steps 4 to 9 used to generate the

variable step size require an extra $3N+12$ DSP operations per sample. For a typical case of $N = 1024$ and $P = 5$, this represents only a 30% increase in complexity.

Table 5.13 Fast Exact GC-VSS3 Algorithm

Initialization: Set $\mathbf{w}(1) = \mathbf{0}$, $\mathbf{x}(0) = \mathbf{0}$, $\bar{c}(0) = 0$, $p(0) = 1$, $\mu(0) = \mu_{MAX}$, $\chi_0(0) = \delta$, and $\chi_b(0) = 0$ for $b = 1 \ldots B$

Computation: At each iteration $n = 1, 2, 3 \ldots$, perform the following steps:

		# of Operations
Step 1:	$e(n) = d(n) - \mathbf{w}^T(n)\,\mathbf{x}(n)$	$\{N\}$
Step 2:	for $b = 0, 1, 2 \ldots B$:	
	$\chi_b(n) = \chi_b(n-1) + x(n)\,x(n-b) - x(n-N)\,x(n-N-b)$	$\{2B+2\}$
Step 3:	$c(n) = e(n) \sum_{b=1}^{B} e(n-b)\,\chi_b(n)$	$\{B+1\}$
Step 4:	$\bar{c}(n) = \bar{c}(n-1) + c(n) - c(n-K)$	$\{2\}$
Step 5:	$p(n) = \beta\,p(n-1) + (1-\beta)\,\mathrm{sign}[\bar{c}(n)]$	$\{3\}$
Step 6:	$\mu'(n) = \alpha\,\mu(n-1) + \gamma\,\mathrm{sign}[p(n)]\,p^2(n)$	$\{5\}$
Step 7:	$\mu(n) = \begin{cases} 0, & \mu'(n) < 0 \\ \mu_{MAX}, & \mu'(n) > \mu_{MAX} \\ \mu'(n), & \text{otherwise} \end{cases}$	$\{2\}$
Step 8:	$\mathbf{w}(n+1) = \mathbf{w}(n) + \dfrac{\mu(n)\,e(n)}{\chi_0(n)}\,\mathbf{x}(n)$	$\{N+11\}$

Unfortunately it is not apparent how a reduction in complexity similar to that given above for the GC-VSS could be obtained with the PC-VSS algorithm. Nor does the Fast Affine Projection seem to lend itself to efficient calculation of the projection correlation. If it is not possible to implement PC-VSS directly in real time, one alternative might be to perform the step size update portion on a block basis. These considerations are beyond the scope of this thesis.

Another important implementation issue is memory size. Table 5.14 summarizes the storage requirements of the proposed algorithms, with NLMS as a benchmark. Note that scalar variables such as step size are neglected for the purposes of comparison. This table shows that for GC-VSS3 an option (ii) implementation would require approximately $3B+K$ more memory locations than NLMS. This should not be a serious problem. However, as suggested earlier, option (i) is dominated by the NB term, which for large N and B may be an unrealistic storage requirement using current technology. Two examples are given in the table, the first for a typical EEC application, and the second for AEC. The resulting calculations show that option (ii) requires considerably less storage space.

There is also an NB term in the total for the PC-VSS algorithm. This comes from the way the long term average projection is calculated using an FIR window (see step 4 of Table 5.10). If this were replaced by an exponential IIR window as shown in Eq. (5.7), the storage requirement for this variable would be reduced to N words. The numbers in parentheses for the examples in Table 5.14 are based on using this alternative.

Note that DSP memory is usually quoted in bytes. So depending on whether 16-bit or 32-bit words are used for storage, a multiplier of 2 or 4 may be needed to convert the quantities given in the table into bytes.

Variable	NLMS	Fast Exact GC-VSS3 option (i)	Fast Exact GC-VSS3 option (ii)	PC-VSS
$\mathbf{w}$	N	N	N	N
$\mathbf{x}$	N	N	$N+B$	—
$\mathbf{X}$	—	—	—	NP
e	—	B	B	P
χ	—	$(N+1)(B+1)$	B	—
c	—	K	K	K
$\boldsymbol{\varepsilon}$	—	—	—	P
$\mathbf{g}$	—	—	—	$N(B+1)$
$\bar{\mathbf{g}}$	—	—	—	N
Totals	$2N$	$NB+3N+2B+K$	$2N+3B+K$	$NB+NP+3N+2P+K$
Example 1 $N=64$ $B=32$ $K=10$ $P=5$	128	2314	234	2580 (532)
Example 2 $N=1024$ $B=500$ $K=10$ $P=5$	2048	516,082	3558	520,212 (8212)

Table 5.14 Memory storage requirements for various algorithms, in units of DSP words.

5.8 Discussion

We have shown that by monitoring the trajectory of the gradient estimates, we can reliably distinguish between echo path and double talk. The concept introduced in section 5.2 has been proven valid.

The proposed algorithms may be viewed as generalizations of several different existing algorithms. For example, the basic GC-VSS reverts to the Mathews-Xie VSS if we set $B = 1$ and $\alpha = 1$. In other words their algorithm lacks the time averaging of the previous gradients, which we found to be very necessary in arriving at a reliable correlation value, especially for a large adaptive filter length N. The Rohrs-Younce algorithm is very similar to GC-VSS1, but the latter has the advantage of a variable step size. There are also some similarities with the Harris algorithm for the trivial case $N = 1$ where use of the m_0, m_1 run count variables can be thought of as somewhat equivalent to the averaging performed by using B blocks.

6 CONCLUSIONS

The echo always has the last word.
— German proverb

6.1 Summary of Contributions

Chapter 4 provided an extensive review of existing double talk mitigation techniques. This is believed to be the one of the most comprehensive surveys on this topic to date, collecting a wide variety of creative suggestions for dealing with the double talk phenomenon in echo cancellation in a single place.

The most significant contribution is the new variable step size approach based on the correlation of instantaneous and long-term gradient estimates. We offered useful insight into how this correlation behaves under different conditions, especially large adaptive filter length N, and used this to come up with enhancements to the basic GC-VSS algorithm. Furthermore we demonstrated the algorithm's effectiveness against double talk via simulation.

A performance comparison was made between the new algorithms and the patented Rohrs-Younce technique, showing the general superiority of the former, including a reduced sensitivity to parameter variation.

We developed an efficient "fast exact" version of the GC-VSS3 algorithm, by observing that the redundancy in the tap input vector could be exploited. Options were noted to permit a tradeoff between memory and computational requirements.

We pointed out that the central VSS approach is not restricted in application to the NLMS weight update, but can be combined with other iterative algorithms. As an example, the Affine Projection algorithm was transformed into a new Projection Correlation VSS algorithm. This was shown to provide enhanced performance when operating with coloured signals such as speech.

Lastly, we addressed the issue of controlling the variable losses or NLP used to suppress the residual echo after cancellation. A combination of the proposed algorithm's step size and the error power gives a good indication of whether double talk is occurring or not, even when the near-end speech is 20 dB below the level of the uncancelled echo.

6.2 Recommendations for Future Work

During the course of this research, several ideas appeared as potential avenues to pursue, but were not followed since they were beyond the immediate scope of this thesis. Some suggestions for future directions are as follows:

1. Proceed to real-time implementation and lab testing of the proposed algorithms. Simulations can only go so far to investigate algorithmic behaviour. There is no test like the real world, especially for subtle problems like the issue of double talk.

2. Undertake a theoretical analysis of the proposed algorithms, the goal being to generate a mathematical model that can be used to accurately predict performance under certain conditions and to determine the optimal parameter values and their relationships.

3. Attempt to integrate the proposed PC-VSS approach with the Fast Affine Projection algorithm. If a means can be found of maintaining the performance benefits of the PC-VSS algorithm in conjunction with the relatively low complexity of the FAP, it will make this approach all the more attractive for commercial implementation.

4. Investigate how the proposed approach could be applied to stereophonic (or multi-channel) AEC. For background reading in this area, some good references are [Son95, Ben97, Jon98].

5. Extend the proposed auxiliary DTD from the simple dual threshold method to use a fuzzy logic approach, or at least one that provides a continuously variable output level based on the strength of the double talk and confidence in the detector's decision.

6. Experiment with other variations of $f(\mathbf{g}(n)\cdot\bar{\mathbf{g}}(n))$ in the step size update.

7. Derive a frequency domain or subband version of the proposed VSS algorithm. Presumably these could be expected to provide better performance for coloured inputs at reduced cost, due to their implicit pre-whitening of the reference signal.

Hopefully reading this thesis will have stirred some interest in these or other related topics. There is still much room for performance improvement in this area.

APPENDIX

A.1 Acoustic Echo Paths used in Simulations

The acoustic echo path used in the simulation experiments of Chapter 5 is characterized in Figure A1. Plot (a) shows the time-domain impulse response **h**. This was recorded in room Minto 2014 at Carleton University, Ottawa. This is a furnished conference room measuring approximately 12 metres by 5.5 metres. To make this measurement, bandlimited white noise was applied at the loudspeaker, and the microphone signal was recorded to digital audio tape. Then these were used as the reference and primary inputs to train a 1200-tap NLMS adaptive filter off-line. The impulse response was obtained as a time average of the filter weights after convergence [Web95].

Plot (b) shows the envelope of the impulse response, displayed as the magnitude on a logarithmic scale. This shows the exponential decay of the response with time quite clearly.

Plot (c) graphs the Total Impulse Power – Tail Power ratio versus the number of taps in the adaptive filter. This demonstrates that with an adaptive filter of length $N = 1024$, the maximum attainable level of echo cancellation is approximately 40 dB.

Plot (d) is the magnitude response of the acoustic echo path versus frequency, treating the echo path as a filter. Note that the response is bandlimited within the telephony band, but relatively flat between the low and high frequency cutoffs.

To simulate a change in the acoustic echo path, the recorded one is modified by applying a randomly generated scaling factor to each of the samples in **h**. The resultant modified echo path **h**′ is characterized in Figure A2. Note that the overall shape of the response remains similar to the original despite the modification.

A.2 Recorded Speech used in Simulations

The later experiments in Chapter 5 used the recorded speech signals characterized in Figure A3. The far-end speech is a female voice uttering the concatenated series of standard test phrases: "It's easy to tell the depth of a well. Four hours of steady work faced us. A large size in stockings is hard to sell. The juice of lemons makes fine punch. It's easy to tell the depth of a well." The near-end speech is a male voice uttering: "He punched viciously at the ball." The duration of the near-end speech is much shorter, because it is only used for 2 seconds to simulate the double talk mode. It is displayed as 10 dB lower than the level of the far-end speech to illustrate the relative magnitudes of these signals as seen by the echo canceller.

The power spectral densities of both speech signals are shown in plots (c) and (d). Note that the male voice at the near end has more of a low frequency content.

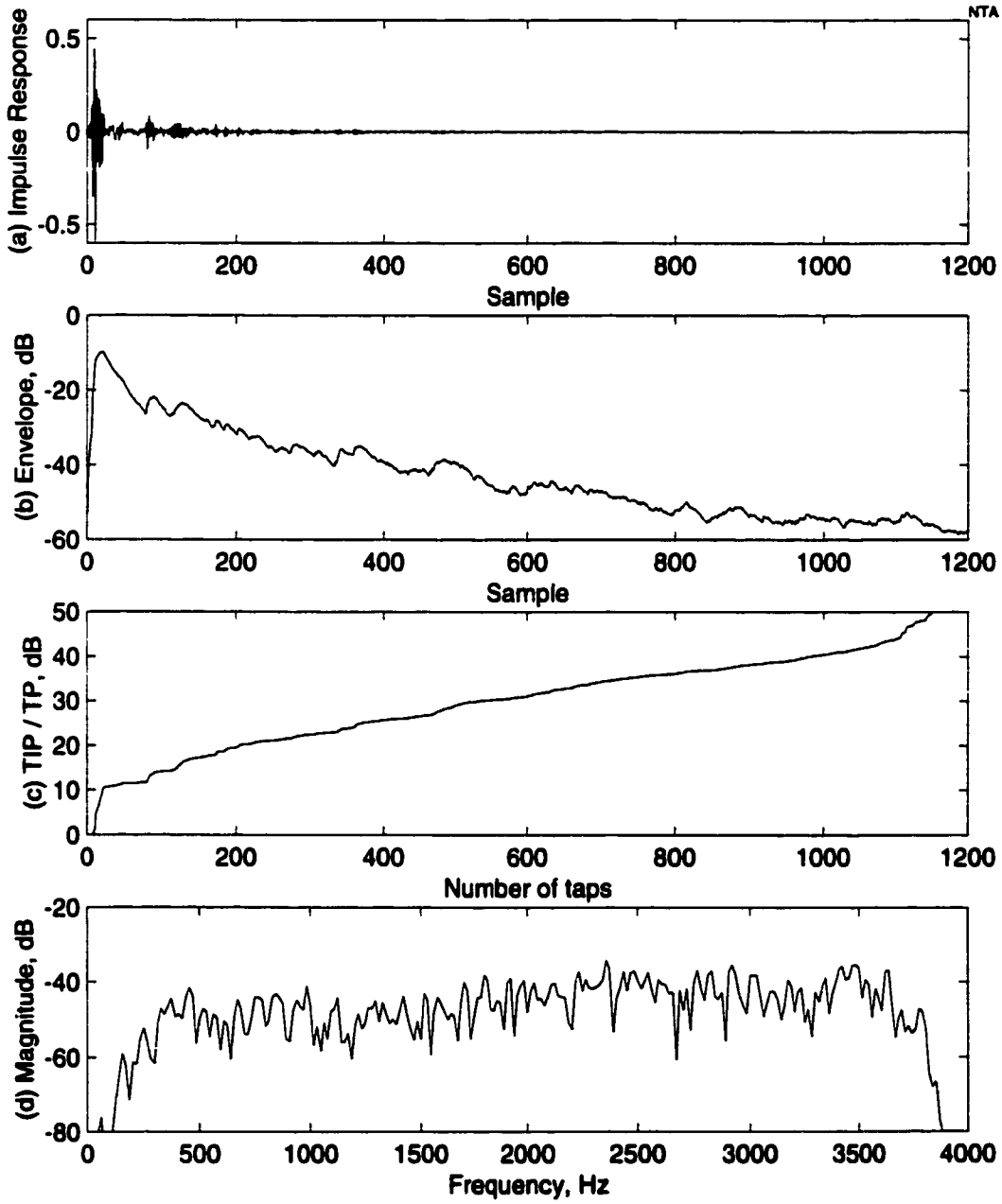

Figure A1 Characterization of acoustic echo path **h** used in simulations.

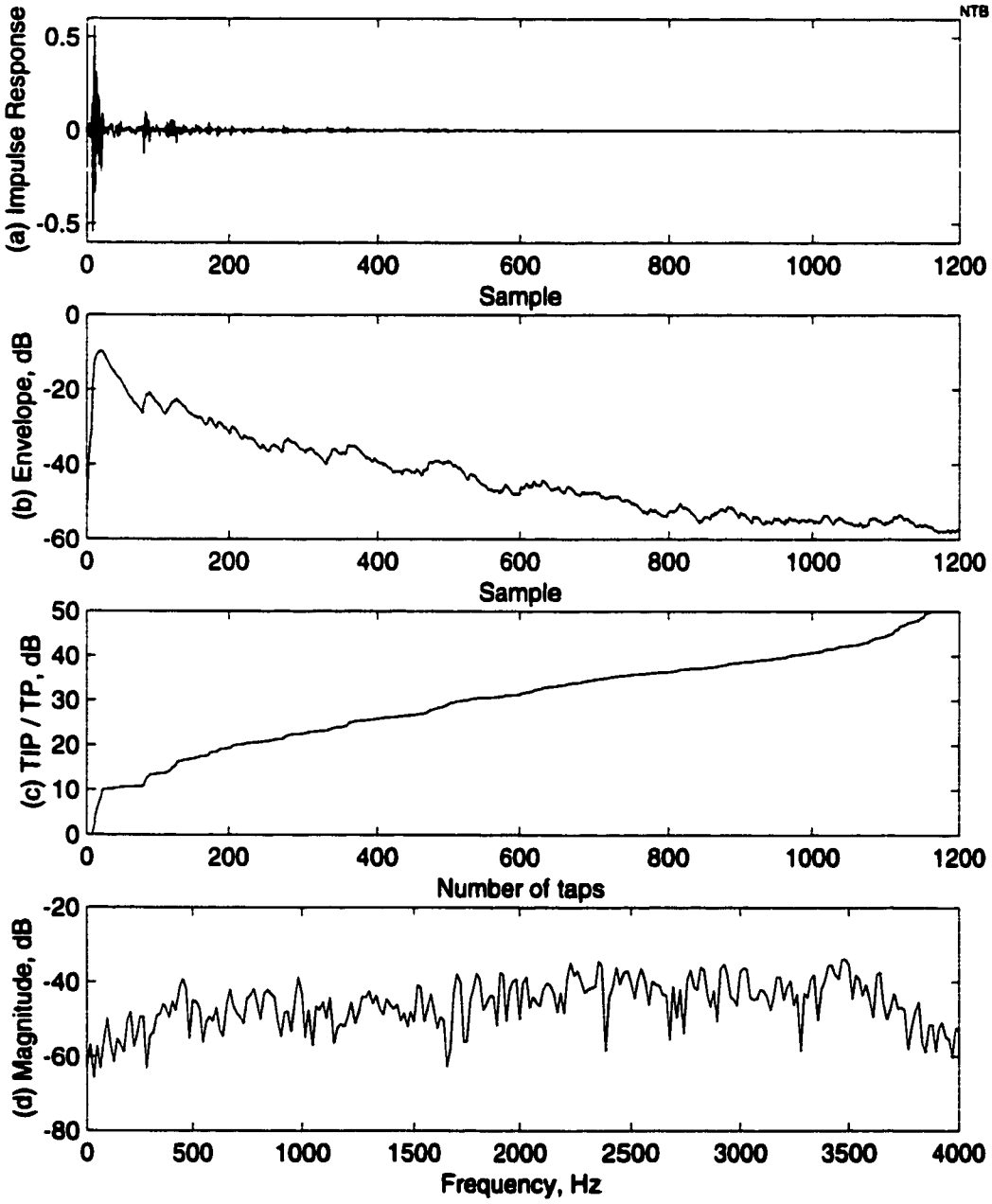

Figure A2 Characterization of modified acoustic echo path $\mathbf{h}'$, used to simulate an echo path change.

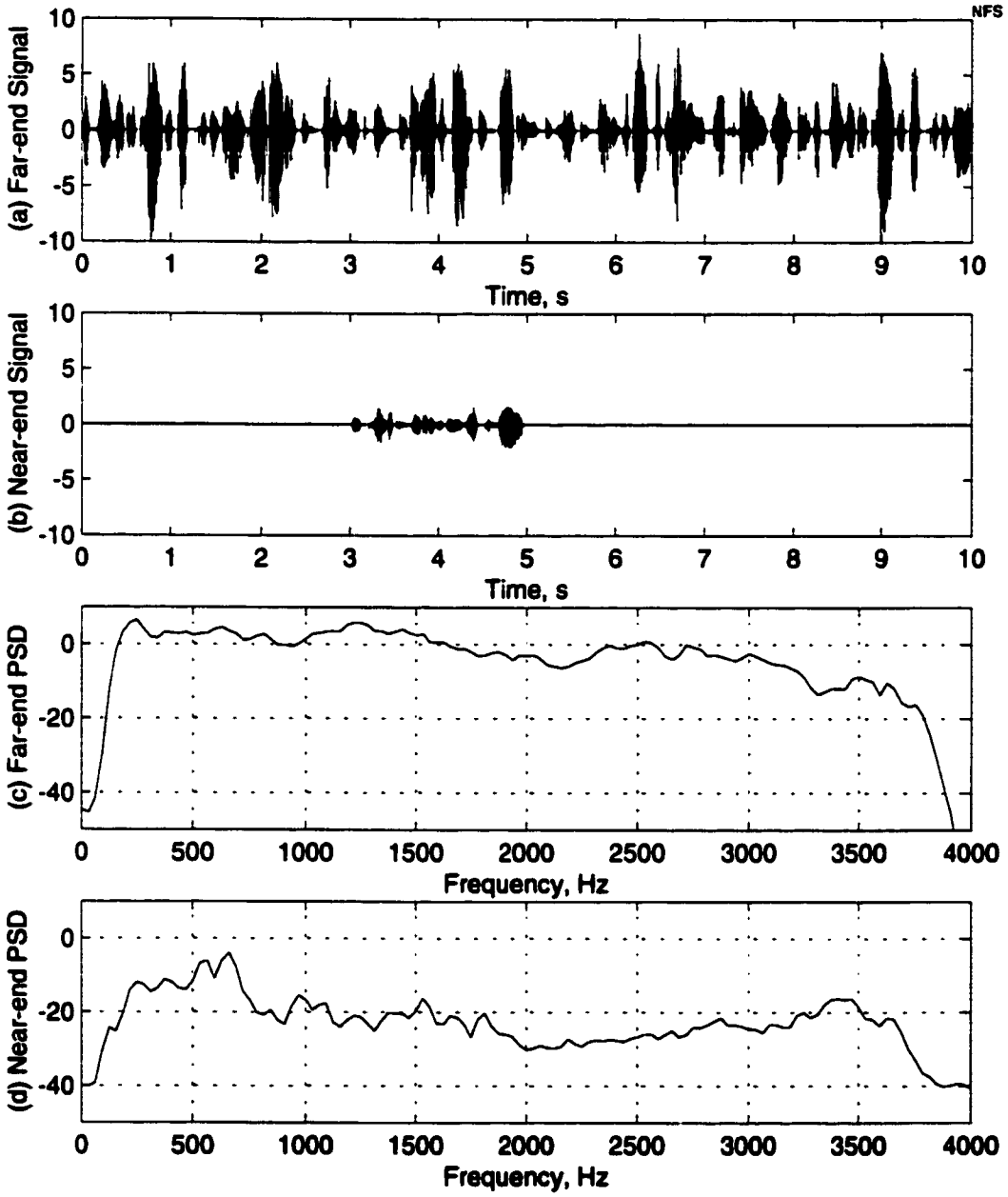

Figure A3 Recorded speech signals and their power spectral densities.

BIBLIOGRAPHY

Abbreviations

ASSP	Acoustics, Speech, and Signal Processing
ICASSP	IEEE International Conference on Acoustics, Speech, and Signal Processing
ICC	IEEE International Conference on Communications
ICSPAT	International Conference on Signal Processing Applications and Technology
IEE	Institute of Electrical Engineers (U.K.)
IEEE	Institute of Electrical and Electronics Engineers
IEICE	Institute of Electronics, Information, and Communications Engineers (Japan)
IRE	Institute of Radio Engineers
ISCAS	IEEE International Symposium on Circuits and Systems
J.	Journal
Proc.	Proceedings of
Trans.	Transactions on
Univ.	University

Alb67 ALBERT, A.E. and L.S. GARDNER. *Stochastic Approximation and Nonlinear Regression*, MIT Press, Cambridge, MA, 1967.

Ama95 AMANO, F., H.P. MEANA, A. DE LUCA and G. DUCHEN. "A multirate acoustic echo canceler structure", *IEEE Trans. Communications*, vol. 43, pp. 2172-2176, July 1995.

Ash99 ASHARIF, M.R., T. HAYASHI and K. YAMASHITA. "Correlation LMS algorithm and its application to double-talk echo cancelling", *Electronics Letters*, vol. 35, pp. 194-195, February 1999.

Bau95 BAUMHAUER, J.C. Jr. *et al.* "Audio technology used in AT&T's terminal equipment", *AT&T Technical J.*, vol. 74, pp. 57-70, March/April 1995.

Bea95 BEAUFAYS, F. "Transform-domain adaptive filters: an analytical approach", *IEEE Trans. Signal Processing*, vol. 43, pp. 422-431, February 1995.

Ben92 BENESTY, J. and P. DUHAMEL. "A fast exact LMS adaptive algorithm", *IEEE Trans. Signal Processing*, vol. 40, pp. 2904-2920, December 1992.

Ben97 BENESTY, J., D.R. MORGAN and M.M. SONDHI. "A better understanding and an improved solution to the problems of stereophonic acoustic echo cancellation", *ICASSP*, vol. 1, pp. 303-306, 1997.

Bir95 BIRKETT, A.N. and R.A. GOUBRAN. "Limitations of handsfree acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects", *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 103-106, 1995.

Bir96 BIRKETT, A.N. and R.A. GOUBRAN. "Nonlinear loudspeaker compensation for hands free acoustic echo cancellation", *Electronics Letters*, vol. 32, pp. 1063-1064, June 1996.

Bra68 BRADY, P.T. "A statistical analysis of on-off patterns in sixteen conversations", *Bell System Technical J.*, vol. 47, pp. 73-92, 1968.

Bre99 BREINING, C. *et al.* "Acoustic echo control — an application of very-high-order adaptive filters", *IEEE Signal Processing Magazine*, vol. 16, pp. 42-69, July 1999.

Car96 CARLEMALM, C., F. GUSTAFSSON and B. WAHLBERG. "On the problem of detection and discrimination of double talk and change in the echo path", *ICASSP*, pp. 2742-2745, 1996.

Cas95 CASCO, F., H. PEREZ, M. NAKANO and M. LOPEZ. "A variable step size (VSS-CC) NLMS algorithm", *IEICE Trans. Fundamentals*, vol. E78-A, pp. 1004-1009, August 1995.

Cha86 CHANG, H. and B.P. AGRAWAL. "A DSP-based echo canceller with two adaptive filters", *ICC*, pp. 1674-1678, 1986.

Cha89 CHAO, J. and S. TSUJII. "A new configuration for echo canceller adaptable during double talk periods", *IEEE Trans. Communications*, vol. 37, pp. 969-974, September 1989.

Cha97 CHAOUI, J. and S. DE GREGORIO. "A robust echo cancellation system especially suited for wireless car handsfree environment", *IEE Colloquium on Adaptive Signal Processing for Mobile Communication Systems*, pp. 3/1-3/6, 1997.

Cio84 CIOFFI, J.M. and T. KAILATH. "Fast, recursive least squares transversal filters for adaptive filtering", *IEEE Trans. Acoustics, Speech & Signal Processing*, vol. 32, pp. 304-337, April 1984.

Cla82 CLARK, G.A., S.K. MITRA and S.R. PARKER. "Block implementation of adaptive digital filters", *IEEE Trans. Acoustics, Speech & Signal Processing*, vol. 29, pp. 744-752, June 1982.

Cla93 CLARKSON, P.M. *Optimal and Adaptive Signal Processing*, CRC Press, Boca Raton, FL, 1993.

Cou98 DE COURVILLE, M. and P. DUHAMEL. "Adaptive filtering in subbands using a weighted criterion", *IEEE Trans. Signal Processing*, vol. 46, pp. 2359-2371, September 1998.

Cre98 CREASY, T. and T. ABOULNASR. "A robust adaptive filtering algorithm for acoustic echo cancellation", *IEEE DSP Workshop*, Bryce Canyon, Utah, 1998.

Den78 DENTINO, M., J. MCCOOL and B. WIDROW. "Adaptive filtering in the frequency domain", *Proc. IEEE*, vol. 66, pp. 1658-1659, December 1978.

Doh97 DOHERTY, J.F. and R. PORAYATH. "A robust echo canceler for acoustic environments", *IEEE Trans. Circuits & Systems II*, vol. 44, pp. 389-396, May 1997.

Ell93 ELLIOT, S.J. and P.A. NELSON. "Active noise control", *IEEE Signal Processing Magazine*, vol. 10, pp. 12-35, October 1993.

Eva93 EVANS, J.B., P. XUE and B. LIU. "Analysis and implementation of variable step size adaptive algorithms", *IEEE Trans. Signal Processing*, vol. 41, pp. 2517-2535, August 1993.

Fuj93 FUJII, K. and J. OHGA. "Compensation for the double-talk detection delay in echo canceller systems", *IEICE Trans. Fundamentals*, vol. E76-A, pp. 1143-1146, July 1993.

Gan97 GÄNSLER, T. "A robust frequency-domain echo canceller", *ICASSP*, vol. 3, pp. 2317-2320, 1997.

Gay95 GAY, S.L. and S. TAVATHIA. "The fast affine projection algorithm", *ICASSP*, vol. 5, pp. 3023-3026, 1995.

Gil92 GILLOIRE, A. and M. VETTERLI. "Adaptive filtering in subbands with critical sampling: analysis, experiments, and application to acoustic echo cancellation", *IEEE Trans. Signal Processing*, vol. 40, pp. 1862-1875, August 1992.

Gil94 GILLOIRE, A. "Performance evaluation of acoustic echo control: required values and measurement procedures", *Annales Télécommunications*, vol. 49, pp. 368-372, July/August 1994.

Gil94b GILLOIRE, A. and E. HÄNSLER. "Acoustic echo control" (editorial), *Annales Télécommunications*, vol. 49, p. 359, July/August 1994.

Gle99 GLENTIS, G.-O., K. BERBERIDIS and S. THEODORIDIS. "Efficient LS adaptive algorithms for FIR transversal filtering: a unified view", *IEEE Signal Processing Magazine*, vol. 16, pp. 13-41, July 1999.

Gou70 GOULD, R.G. and G.K. HELDER. "Transmission delay and echo suppression", *IEEE Spectrum*, pp. 47-54, April 1970.

Gri84 GRITTON, C.W.K. and D.W. LIN. "Echo cancellation algorithms", *IEEE ASSP Magazine*, pp. 30-37, April 1984.

Gup98 GUPTA, P. *et al.* "Improved echo canceller design and implementation", *ICSPAT*, 1998.

Hal86 HALLIDAY, D. and R. RESNICK. *Fundamentals of Physics, 2nd ed.*, John Wiley & Sons, New York, 1986.

Han92 HÄNSLER, E. "The hands-free telephone problem – an annotated bibliography", *Signal Processing*, vol. 27, pp. 259-271, June 1992.

Han94 HÄNSLER, E. "The hands-free telephone problem: an annotated bibliography update", *Annales Télécommunications*, vol. 49, pp. 360-367, July/August 1994.

Har86 HARRIS, R.W., D.M. CHABRIES and F.A. BISHOP. "A variable step (VS) adaptive filter algorithm", *IEEE Trans. ASSP*, vol. 34, pp. 309-316, April 1986.

Hay83 HAYASHI, T., S. UNAGAMI, M. KOSHIKAWA and K. MURANO. "Echo canceller with effective double talk control", *GLOBECOM*, pp. 1389-1393, 1983.

Hay96 HAYKIN, S. *Adaptive Filter Theory*, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 1996.

Hay99 HAYKIN, S. "Adaptive filters", in "Recent developments in the core of digital signal processing", *IEEE Signal Processing Magazine*, vol. 16, pp. 20-22, January 1999.

Hei93 HEITKÄMPER, P. and M. WALKER. "Adaptive gain control and echo cancellation for hands-free telephone systems", *ISCAS*, vol. 1, pp. 455-458, 1993.

Hei95 HEITKÄMPER, P. "Optimization of an acoustic echo canceller combined with adaptive gain control", *ICASSP*, pp. 3047-3050, 1995.

Hei97 HEITKÄMPER, P. "An adaptation control for acoustic echo cancellers", *IEEE Signal Processing Letters*, vol. 4, pp. 170-172, June 1997.

Hsi83 HSIA, T.C. "Convergence analysis of LMS and NLMS adaptive algorithms", *ICASSP*, pp. 667-670, 1983.

Hug93 HUGHES, P.J., S.F.A. IP and J. COOK. "Adaptive Filters - A Review of Techniques", Chapter 3 of *Digital Signal Processing in Telecommunications*, edited by F.A. Westall and S.F.A. Ip, Chapman & Hall, London, 1993.

ITU93a ITU-T RECOMMENDATION G.165, *Echo Cancellers*, International Telecommunication Union, 1993.

ITU93b ITU-T RECOMMENDATION G.167, *Acoustic Echo Cancellers*, International Telecommunication Union, 1993.

ITU97 ITU-T RECOMMENDATION G.168, *Digital Network Echo Cancellers*, International Telecommunication Union, 1997.

Joh95 JOHNSON, C.R. Jr. "On the interaction of adaptive filtering, identification, and control", *IEEE Signal Processing Magazine*, vol. 12, pp. 22-37, March 1995.

Kes58 KESTEN, H. "Accelerated stochastic approximation", *Ann. Math. Stat.*, vol. 29, pp. 41-59, March 1958.

Kna92 KNAPPE, M.E. *Acoustic Echo Cancellation: Performance and Structures*, M.Eng. Thesis, Carleton Univ., 1992.

Kna94 KNAPPE, M.E. and R.A. GOUBRAN. "Steady-state performance limitations of full-band echo cancellers", *ICASSP*, vol. 2, pp. 73-76, 1994.

Kuo94 KUO, S.M. and Z. PAN. "Development and analysis of distributed acoustic echo cancellation microphone system", *Signal Processing*, vol. 37, pp. 333-344, 1994.

Kwo92 KWONG, R.H. and E.W. JOHNSTON. "A variable step size LMS algorithm", *IEEE Trans. Signal Processing*, vol. 40, pp. 1633-1642, July 1992.

Lar98 LARIVIERE, J. and R. GOUBRAN. "GMDFα with adaptive reconstruction filters and zero throughput delay", *ICASSP*, vol. 6, pp. 3553-3556, 1998.

Lew93 LEWIS, A. "Adaptive Filtering – Applications in Telephony", Chapter 4 of *Digital Signal Processing in Telecommunications*, edited by F.A. Westall and S.F.A. Ip, Chapman & Hall, London, 1993.

Lia98 LIAVAS, A.P. and P.A. REGALIA. "Acoustic echo cancellation: Do IIR models offer better modeling capabilities than their FIR counterparts?", *IEEE Trans. Signal Processing*, vol. 46, pp. 2499-2504, September 1998.

Lu99 LU, Y. and J.M. MORRIS. "Gabor expansion for acoustic echo cancellation", *IEEE Signal Processing Magazine*, vol. 16, pp. 68-80, March 1999.

Luc99 LUCENT TECHNOLOGIES INC., "Lucent Technologies introduces system-on-a-chip echo canceler with tenfold performance improvement", Company Press Release, April 20 1999.

Mak93 MAKINO, S., Y. KANEDA and N. KOIZUMI. "Exponentially weighted step size NLMS adaptive filter based on the statistics of a room response", *IEEE Trans. Speech & Audio Processing*, vol. 1, pp. 101-108, January 1993.

Mak97 MAKINO, S. "Acoustic echo cancellation", in "The past, present, and future of audio signal processing", *IEEE Signal Processing Magazine*, vol. 14, pp. 39-41, September 1997.

Mat90 MATHEWS, V.J. and Z. XIE. "Stochastic gradient adaptive filters with gradient adaptive step sizes", *ICASSP*, pp. 1385-1388, 1990.

Mat93 MATHEWS, V.J. and Z. XIE. "A stochastic gradient adaptive filter with gradient adaptive step size", *IEEE Trans. Signal Processing*, vol. 41, pp. 2075-2087, June 1993.

May95a MAYYAS, K. and T. ABOULNASR. "A robust variable step size LMS-type algorithm: analysis and simulations", *ICASSP*, pp. 1408-1411, 1995.

May95b MAYYAS, K. *Gradient Adaptive Digital Filtering: Problems and Solutions*, Ph.D. Thesis, Univ. Ottawa, 1995.

Mes84 MESSERSCHMITT, D.G. "Echo cancellation in speech and data transmission", *IEEE J. Selected Areas in Communications*, vol. 2, pp. 283-297, 1984.

Min85 MINAMI, S. and T. KAWASAKI. "A double-talk detection method for an echo canceller", *ICC*, pp. 1492-1497, 1985.

Miy95 MIYATAKE, M.N. *et al.* "A time-varying step size normalized LMS algorithm for adaptive echo canceler structures", *IEICE Trans. Fundamentals*, vol. E78-A, pp. 254-258, February 1995.

Mor95 MORGAN, D.R. and J.C. THI. "A delayless subband adaptive filter architecture", *IEEE Trans. Signal Processing*, vol. 43, pp. 1819-1830, August 1995.

Mur90 MURANO, K., S. UNAGAMI and F. AMANO. "Echo cancellation and applications", *IEEE Communications Magazine*, vol. 28, pp. 49-55, January 1990.

Nag67 NAGUMO, J.I. and A. NODA. "A learning method for system identification", *IEEE Trans. Automatic Control*, vol. 12, pp. 282-287, June 1967.

Nar83 NARAYAN, S.S., A.M. PETERSON and M.J. NARASHIMA. "Transform domain LMS algorithm", *IEEE Trans. Acoustics, Speech & Signal Processing*, vol. 31, pp. 609-615, June 1983.

Nay94 NAYLOR, P., J. ALCAZAR, J. BOUDY and Y. GRENIER. "Enhancement of hands-free telecommunications", *Annales Télécommunications*, vol. 49, pp. 373-379, July/August 1994.

Och77 OCHIAI, K., T. ARASEKI and T. OGIHARA. "Echo canceler with two echo path models", *IEEE Trans. Communications*, vol. 25, pp. 589-595, June 1977.

Opp89 OPPENHEIM, A.V. and R.W. SCHAFER. *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.

Oze84 OZEKI, K. and T. UMEDA. "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties", *Electronics and Communications in Japan*, vol. 67-A, pp. 19-27, 1984.

Por96 PORAYATH, R., J.F. DOHERTY and S.F. RUSSELL. "An adaptive acoustic echo canceler for hands-free teleconferencing", *Digital Signal Processing*, vol. 6, pp. 29-36, January 1996.

Pra94 PRADO, J. and E. MOULINES. "Frequency-domain adaptive filtering with applications to acoustic echo cancellation", *Annales Télécommunications*, vol. 49, pp. 414-428, July/August 1994.

Pri95 PRICE, L. "The hazards of driving and dialing", Cable News Network, http://www.cnn.com/HEALTH/9510/cell_phone, October 20, 1995.

Qi96 QI, D. "Acoustic Echo Cancellation: Algorithms and Implementation on the TMS320C8X", *Application Report, Texas Instruments*, 1996.

Rab95 RABINER, L.R. "Towards Vision 2001: voice and audio processing considerations", *AT&T Technical J.*, vol. 74, pp. 4-13, March/April 1995.

Ree98 REED, M.J. and M.O.J. HAWKSFORD. "Acoustic echo cancellation with the fast affine projection", *IEE Colloquium on Audio and Music Technology*, pp. 16/1-16/8, 1998.

Roh90 ROHRS, C.E. and R.C. YOUNCE. "Double talk detector for echo canceller and method", *U.S. patent #4,918,727*, April 17, 1990.

Rya92 RYAN, J.G. *Subband Adaptive Filters*, M.Eng. Thesis, Carleton Univ., 1992.

Sar96 SARMA, J. "Echo canceller design ensures hands-free high fidelity telephony", *EDN Magazine*, vol. 41, pp. 131-133, August 1996.

Sha88 SHAN, T.J. and T. KAILATH. "Adaptive algorithms with an automatic gain control feature", *IEEE Trans. Circuits & Systems*, vol. 35, pp. 122-127, January 1988.

Shy89 SHYNK, J.J. "Adaptive IIR filtering", *IEEE ASSP Magazine*, vol. 6, pp. 4-21, April 1989.

Shy92 SHYNK, J.J. "Frequency-domain and multirate adaptive filtering", *IEEE Signal Processing Magazine*, vol. 9, pp. 14-37, January 1992.

Son66 SONDHI, M.M. and A.J. PRESTI. "A self-adaptive echo canceller", *Bell System Technical J.*, vol. 45, pp. 1851-1854, December 1966.

Son67 SONDHI, M.M. "An adaptive echo canceller", *Bell System Technical J.*, vol. 46, pp. 497-511, March 1967.

Son80 SONDHI, M.M. and D.A. BERKELEY. "Silencing echoes on the telephone network", *Proc. IEEE*, vol. 68, pp. 948-963, August 1980.

Son95 SONDHI, M.M., D.R. MORGAN and J.L. HALL. "Stereophonic acoustic echo cancellation – an overview of the fundamental problem", *IEEE Signal Processing Letters*, vol. 2, pp. 148-151, August 1995.

Sug94 SUGIYAMA, A. "Stochastic gradient algorithms with a gradient adaptive and limited step size", *IEICE Trans. Fundamentals*, vol. E77-A, pp. 534-538, March 1994.

Tan95 TANAKA, M., Y. KANEDA, S. MAKINO and J. KOJIMA. "Fast projection algorithm and its step size control", *ICASSP*, pp. 3023-3026, 1995.

Tan99 TANAKA, M., S. MAKINO and J. KOJIMA. "A block exact fast affine projection algorithm", *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 79-86, January 1999.

Vuk96 VUKADINOVIC, M.O. *Acoustic Echo Cancellation Structures Based on Perceptual Hearing Criteria*, M.A.Sc. Thesis, Univ. of Ottawa, 1996.

Wan95 WANG, Y., K. NAKAYAMA and Z. MA. "A new structure for noise and echo cancelers based on a combined fast adaptive filter algorithm", *IEICE Trans. Fundamentals*, vol. E78-A, pp. 845-853, July 1995.

Web95 WEBSTER, T.G.D. *Tracking Performance of Acoustic Echo Cancellers*, M.Eng. Thesis, Carleton Univ., 1995.

Wei77 WEINSTEIN, S.B. "Echo cancellation in the telephone network", *IEEE Communications Society Magazine*, pp. 8-15, January 1977.

Wid60 WIDROW, B. and M.E. HOFF. "Adaptive switching circuits", *IRE WESCON*, vol. 4, pp. 96-104, 1960.

Wid75 WIDROW, B. *et al.* "Adaptive noise canceling: principles and applications", *Proc. IEEE*, vol. 63, pp. 1692-1716, December 1975.

Wid99 WIDROW, B. and S.D. STEARNS. *Adaptive Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 1999.

Yas91 YASUKAWA, H. *et al.* "Echo return loss required for an acoustic echo controller based on a subjective assessment", *IEICE Trans.*, vol. E74, pp. 692-705, 1991.

Ye91 YE, H. and B. WU. "A new double-talk detection algorithm based on the orthogonality theorem", *IEEE Trans. Communications*, vol. 39, pp. 1542-1545, November 1991.

Yoo97 YOO, J.H. and S.H. CHO. "A new double talk detector using the lattice predictors for an acoustic echo canceller", *IEEE Region 10 Annual Conference (TENCON)*, vol. 2, pp. 483-486, 1997.

Yua94 YUAN, H. *Dynamic Behaviour of Acoustic Echo Cancellation*, M.Eng. Thesis, Carleton Univ., 1994.

Zha98 ZHANG, Z. and G. SCHMER. *Performance Analysis of Line Echo Cancellation Implementation using TMS320C6201*, Application Report SPRA421, Texas Instruments, 1998.