INFORMATION TO USERS

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

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

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

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

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

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

UMI[®]

Effects of Delay Jitter and Clock Slippage on Network Echo Cancellation

submitted by

Xin Xu, B.Eng.

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

Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario, Canada K1S 5B6

© Copyright 2000. Xin Xu

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

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

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

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

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

0-612-48477-7

Canada

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

Effects of Delay Jitter and Clock Slippage on Network Echo Cancellation

submitted by

Xin Xu, B.Eng.

in partial fulfillment of the requirements for the degree of Master's of Engineering

Dr. Rafik. A. Goubran

Thesis Supervisor:

Dr. Rafik A. Goubran

Chair, Department of Systems and Computer Engineering

Carleton University

January, 2000

ABSTRACT

This thesis studies the effects of delay jitter and clock slippage on network echo cancellation.

It evaluates the performance of sparse-tap algorithms, such as STWQ (Scrub Taps Waiting in a Queue), and TDE (Time Delay Estimation) based NLMS (Normalized Least Mean Square), as well as the Full-tap NLMS for comparison.

This thesis introduces a new sparse-tap echo cancellation algorithm, SEER (Squared Error Estimation and Realignment). The performance of the proposed algorithm is identical to the conventional TDE in the absence of delay jitter and clock slippage. Its performance surpasses most conventional algorithms in the presence of delay jitter and clock slippage. The new algorithm can achieve up to 20dB enhancement in ERLE (Echo Return Loss Enhancement) level compared with other algorithms.

ACKNOWLEDGEMENTS

First of all, I'd like to thank Professor Rafik. A. Goubran, who is an intelligent researcher, a wise department leader, and a diligent guide for my master's study. Thank you for the encouragement you gave me and the endless support and understanding.

Thanks, also, to Mr. Dieter Schulz from Mitel Corporation, for helping me with the problem specification and providing the related technical documents.

I would also like to thank Danny Lemay, Darlene, Judy and all other technical staff and non-technical staff in the department, who provided me with lab facilities and a friendly environment for both studying and researching.

My friend, Fang Wang, you are always so generous with your precious time with me, and have comforted me with friendship.

My special little guy, you're seven months old now. You're a little examiner when sitting in your highchair and staring at mom's thesis, a little typist when the keyboard's in front of you, and a big bad boy when you won't go to sleep late at night. Thank you for your cooperation and all the joys you gave me during the construction of this thesis.

Thank you, Dad, Mom, and my little sister. Your unconditional love and trust will always be with me and support me no matter what kind of difficulty I encounter.

Also, thanks to my husband, for the advice you gave me and baby-sitting whenever needed without complaining.

Finally, thanks to the Department of Systems and Computer Engineering, Carleton University, CITO, Mitel Corporation and Ontario Graduate Scholarship for the financial support.

For my dear parents

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
TABLE OF CONTENTS.....	vi
LIST OF FIGURES.....	viii
LIST OF ABBREVIATION.....	x
Chapter 1. INTRODUCTION	1
Chapter 2. PROBLEM DEFINITION AND SIMULATION MODEL	4
2.1 Problem Description.....	4
2.1.1 Echoes	4
2.1.2 Some Fundamental Requirements of an Echo Canceller [G168]	6
2.2 Problem Statement	7
2.3 Simulation Model.....	7
2.3.1 Echo Channel Impulse Response.....	8
2.3.2 Timing Jitter Model.....	9
2.3.3 Network Delay Jitter Model	9
Chapter 3. BACKGROUND REVIEW.....	10
3.1 Historical Review of Echo Cancellers	10
3.1.1 Echo Suppressor	10
3.1.2 Echo Canceller	10
3.2 Adaptive Filters and Algorithms	12
3.2.1 Filter Structure	12
3.2.2 Filtering Algorithms.....	13
3.3 Echo Cancellation Methods Using Sparse-tap Filters	16
3.3.1 STWQ (Scrub Taps Waiting in a Queue)	16
3.3.2 Variations of STWQ.....	19
3.3.3 TDE (Time Delay Estimation) Based Sparse-tap NLMS	21
Chapter 4. Evaluation of Different Algorithms	26
4.1 The Interface.....	26
4.2 Static Echo Channel.....	27
4.2.1 General Performance	28
4.2.2 The Selection of Filter Parameters	29
4.3 Effect of Delay Jitters on Echo Cancellation	34
4.3.1 The Detection of Delay Jitters.....	34
4.3.2 Simulation Results	35
4.4 Effect of Timing Jitter	39
4.4.1 One Sample Loss or Shortage.....	40
4.4.2 Multiple-sample Losses	53
4.4.3 Re-sampling	64
Chapter 5. PROPOSED ALGORITHM.....	70
5.1 Introduction	70
5.2 The Idea of SEER	71
5.3 Theoretical Foundation.....	72
5.3.1 Error Analysis	72

5.3.2 The Statistical Tests.....	72
5.4 Algorithm Implementation.....	78
5.4.1 Thresholds Determination.....	79
5.4.2 Center Recalculation.....	80
5.4.3 Flow Chart	81
5.5 Simulation Results.....	81
5.5.1 Delay Jitters.....	81
5.5.2 Sample Loss	85
Chapter 6. CONCLUSION.....	88
6.1 Conclusion	88
6.2 Future Work	89
References.....	91
Appendix A.....	95
Appendix B.....	98
Appendix C.....	101

LIST OF FIGURES

Figure 1-1 Internet-supported IP Gateway for Voice.....	2
Figure 2-1 A Hybrid in a Telephone Network	4
Figure 2-2 Echo Channel Model.....	8
Figure 3-1 A Simple Echo Canceller Configuration	11
Figure 3-2 Structure of a Transversal Filter	13
Figure 3-3 Dispersive Regions of a Typical Echo Channel Impulse Response	16
Figure 3-4 STWQ (Scrub Taps Waiting in a Queue)	17
Figure 3-5 Grouping Method	21
Figure 3-6 Time Delay Estimation Signal Sequences	22
Figure 3-7 The TDE Method Echo Canceller Structure.....	24
Figure 4-1 Simulation Interface.....	27
Figure 4-2 Speed of Convergence	28
Figure 4-3 Effect of Constraint in TDE	30
Figure 4-4 Effect of The Number of Short Filters.....	31
Figure 4-5 STWQ with Elimination rate = 1	32
Figure 4-6 STWQ with Different Elimination Rate (a)=10, (b)=100, (c)=100,150,200.....	34
Figure 4-7 TDENCR for Delay Jitter	36
Figure 4-8 TDECR for Delay Jitter	36
Figure 4-9 STWQ for Delay Jitter.....	37
Figure 4-10 NLMS for Delay Jitter	37
Figure 4-11 Sample Loss, One Sample at a Time	40
Figure 4-12 Sample Shortage, One Sample at a Time	41
Figure 4-13 TDENCR for SS Scenario One	42
Figure 4-14 TDECR for SS Scenario One.....	43
Figure 4-15 STWQ for SS Scenario One.....	44
Figure 4-16 NLMS for SS Scenario One.....	45
Figure 4-17 Sample Loss, One Sample at a Time	46
Figure 4-18 Sample Shortage, One Sample at a Time	47
Figure 4-19 TDENCR for SS Scenario Two	48
Figure 4-20 TDECR for SS Scenario Two	49
Figure 4-21 STWQ for SS Scenario Two.....	50
Figure 4-22 NLMS for SS Scenario Two	51
Figure 4-23 Multiple Sample Losses Scenario One.....	53
Figure 4-24 TDENCR MS Scenario One	54
Figure 4-25 TDECR MS Scenario One	55
Figure 4-26 STWQ MS Scenario One.....	55
Figure 4-27 NLMS MS Scenario One	56
Figure 4-28 Multiple Sample Losses Scenario Two, Case One	57
Figure 4-29 TDENCR MS: Scenario Two, Case One.....	58
Figure 4-30 TDECR MS, Scenario Two, Case One.....	59
Figure 4-31 STWQ MS, Scenario Two, Case One	60
Figure 4-32 NLMS MS, Scenario Two, Case One.....	60
Figure 4-33 Multiple Sample Losses Scenario Two, Case Two	61

Figure 4-35 TDECR MS, Scenario Two, Case Two.....	62
Figure 4-36 STWQ MS, Scenario Two, Case Two	63
Figure 4-37 NLMS MS, Scenario Two, Case Two.....	63
Figure 4-38 An Analog Signal Sampled at Different Rates.....	65
Figure 4-39 Re-sample with Sample Loss.....	66
Figure 4-40 TDENCR with Resample	67
Figure 4-41 TDECR with Resample.....	67
Figure 4-42 STWQ for Resample.....	68
Figure 4-43 Full-tap NLMS for Resample	68
Figure 5-1 <i>pdf</i> of $\tilde{J}_n$ (red) and J_n (blue).....	73
Figure 5-2 P (<i>incorrect rejection</i>) versus μ	77
Figure 5-3 P (<i>incorrect acceptance</i>) versus μ and ρ	78
Figure 5-4 Delay Jitter Effect of SEER and TDECR (1).....	83
Figure 5-5 Delay Jitter with SEER and TDECR (2)	84
Figure 5-6 Sample Loss Scenario One SEER and TDECR	85
Figure 5-7 Sample Loss Scenario Two SEER and TDECR	86

LIST OF ABBREVIATION

AGWN	Additive Gaussian White Noise
ATM	Asynchronous Transfer Mode
DC	Direct Cross-correlation
DPLL	Digital Phase Locked Loop
ERL	Echo Return Loss
ERLE	Echo Return Loss Enhancement
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
IP	Internet Protocol
LS	Least Square
NLMS	Normalized Least Mean Square
PBX	Private Branch eXchange
pdf	probability density function
PSTN	Public Switched Telephone Network
SEER	Squared Error Estimation and Realignment
SNR	Signal to Noise Ratio
STWQ	Scrub Taps Waiting in a Queue
TDE	Time Delay Estimation
TDECR	Time Delay Estimation with Center Recalculation
TDENCR	Time Delay Estimation without Center Recalculation

Chapter 1. INTRODUCTION

Traditional telephone networks have been one of the most commonly used communication media for more than a hundred years. Although telephone networks were originally designed to transmit analog speech signals, they are capable of transmitting digital signals at a reasonable rate [HENN93]. The increased use of packet networks has resulted in increased delays and packet losses, which has affected the operation of the network echo cancellers.

Network echoes, also referred to as line echoes, are the reflections of signals transmitted back to the sender due to imperfections of the network. The presence of echoes makes full-duplex transmission impossible. The echo channel in networks that employ conventional switching methods is static, namely having a time-invariant impulse response. An echo channel can also be time variant, which is caused by the dynamic variation in network characteristics. When a packet network such as ATM (Asynchronous Transfer Mode) or IP (Internet Protocol) is used, the echo channel has a time-variant impulse response due to network traffic, load distribution, and other network issues [CEFR98] [FALC85].

Echo cancellation techniques are well established for conventional networks. Those algorithms, however, are not capable of handling echoes in the current network circumstances with packet switching methods. When using conventional switching methods, the locations of the hybrids are known and fixed, so the delays of echoes are fixed as well. Modern switching methods such as ATM and IP, in contrast, produce echoes with unknown delays, which introduce the time-varying aspect. At the same time, timing jitter caused by network synchronization problem will also influence the echo cancellation procedure. These factors

may cause changes to the echo channel's impulse response that are difficult for the conventional echo cancellers to track.

New echo cancellers are required as telephone networks evolve. This thesis evaluates the performance of previously known echo cancellation algorithms and proposes a new algorithm to deal with network echoes corrupted by delay jitters caused by packet switching networks as shown in Figure 1-1 and timing jitters caused by clock differences.

Figure 1-1 Internet-supported IP Gateway for Voice

The proposed algorithm, SEER (Squared Error Estimation and Realignment), uses a statistical approach to detect the occurrence of delay jitter and estimate the delays of the echoes, once a delay jitter or clock slippage occurs. In most cases, it is demonstrated that,

when SEER is implemented, the overall performance of the echo canceller can improve by up to 20dB in terms of ERLE (Echo Return Loss Enhancement) level, compared with previous methods.

This thesis is organized as follows:

Chapter 2, “PROBLEM DEFINITION AND SIMULATION MODEL”, defines the problem encountered in the network that will be dealt with by the new canceller and describes the echo channel model that will be used in the simulation.

Chapter 3, “BACKGROUND REVIEW”, introduces the fundamentals of adaptive filtering and related algorithms, which are the bases of most echo cancellation algorithms.

Chapter 4, “EVALUATION OF DEFFERENT ALGORITHMS”, compares the previously known algorithms and evaluates their performances in current channel model.

Chapter 5, “PROPOSED ALGORITHM”, introduces a new algorithm and compares it with the existing algorithms.

Chapter 6, “CONCLUSION”, summarizes the study and concludes the thesis.

Chapter 2. PROBLEM DEFINITION AND SIMULATION MODEL

2.1 Problem Description

2.1.1 Echoes

2.1.1.1 The Origin of The Echoes

All transmission lines suffer from impedance mismatches. These mismatches cause some of the energy from voice and data signals transmitted to reflect back to the originator.

The hybrids in the network are the main source of echoes in today's telephone lines. A hybrid is a transformer, which transforms data or voice signals from a two-wire local loop into a four-wire transmission line, as shown in Figure 2-1.

Figure 2-1 A Hybrid in a Telephone Network

Whenever a transmitted signal encounters a hybrid, an echo is generated and sent back. The echo channel's impulse response of the conventional telephone line is known because the hybrid locations are fixed.

2.1.1.2 Impulse Response of The Echo Channel in a Telephone Line

Unlike the case of acoustic echoes, echoes in telephone lines are sparse. The impulse response of a sparse system is characterized as one in which a large part of the impulse response is zero.

The impulse response of a system is the response of the system when taking an impulse as input.

If an impulse is transmitted to the telephone network, an echo will be returned to the sender only when the impulse encounters a hybrid, so when it is transmitting between two hybrids, no echo will be generated. Obviously the number of echoes the sender would receive is directly related to the number of hybrids along the transmission line. Between any two of the echoes, there is just silence. So the telephone network is sparse in nature.

However, the further the hybrid is from the sender, the less energy the echo will have. Thus, the farthest echoes will not have enough energy to be detected. Consequently, the number of echoes is expected to be less than the number of hybrids in the telephone network.

The telephone network can be viewed as a fully physically connected network when using conventional switching methods. The delays of the echoes are known because of the predetermined locations of hybrids. The conventional echo canceller will, therefore, be successful under this condition.

In recent years, digital packet networks have been replacing conventional switched networks. These networks will not establish physical connections between terminals, but virtual connections instead. This virtual connection causes most of the problems that challenge the conventional echo canceller, such as long delays, delay jitters and packet losses.

In this thesis, a solution is found for the problems caused by delay jitters and timing jitters in network echo cancellation.

2.1.1.3 Timing Jitters and Delay Jitters

Timing jitter (Clock Slippage)

Timing jitter is a type of analog communication line distortion caused by the variation of a signal from its reference timing positions. This distortion can cause data transmission errors, particularly at high speeds, and can also influence the accuracy of an echo canceller. The effect of timing jitter can be reflected by the residual mean-squared error of the signal entering the receiver and its relation to the adaptation, timing recovery and channel parameters.

Timing jitter is most commonly caused by a clock rate difference between the sender and receiver, namely the clock slippage. If the clock rate on the receiver side is faster, samples of the signal from the echo channel will be lost. On the other hand, if the clock rate on the sender side is faster, sample shortage will happen to the signal from the echo channel, which will result in a duplicated sample.

Refer to Chapter 4 for a detailed explanation and examples.

Delay jitters

Delay jitter can be caused by the unpredictability of traffic, routing methods and adaptive jitter buffers in the network. Delay jitter causes the delays of the echoes to change, so the impulse response of the echo channel changes correspondingly.

Also refer to Chapter 4 for a detailed explanation and examples.

2.1.2 Some Fundamental Requirements of an Echo Canceller [G168]

- 1) Large steady state ERLE (Echo Return Loss Enhancement)
- 2) Rapid convergence
- 3) Low returned echo level during single talk
- 4) Low divergence during double talk
- 5) Assured double talk detection
- 6) Proper operation during facsimile and low speed (<9.6kb/s) voice-band data transmissions

2.2 *Problem Statement*

By evaluating the performance of sparse-tap algorithms, such as STWQ (Scrub Taps Waiting in a Queue), and TDE (Time Delay Estimation) based NLMS (Normalized Least Mean Square), as well as the Full-tap NLMS for comparison, this thesis studies the effects of delay jitter and clock slippage on network echo cancellation.

And also this thesis introduces a new sparse-tap echo cancellation algorithm SEER (Squared Error Estimation and Realignment). The performance of the proposed algorithm is identical to the conventional TDE in the absence of delay jitter and clock slippage. Its performance surpasses most conventional algorithms in the presence of delay jitter and clock slippage.

2.3 *Simulation Model*

In order to design a simulation model that addresses the problem specified, it is essential to determine the impulse response of the desired channel model. According to the problem definition, a sparse model with long echo path delay is used.

2.3.1 Echo Channel Impulse Response

The echo channel, as previously stated in section 2.1, has a sparse impulse response, meaning that most of the coefficients are zero. An echo channel model is designed to fulfill some typical condition, such as being able to cancel echoes with long delays, and the co-existence of multiple echoes in the network.

The model is shown in Figure 2-2. The sampling rate is assumed to be 8 kHz, which is the sampling rate of speech signals in current networks. Twenty-five hundred taps are used to simulate the echoes with maximum 312ms delay. Three echoes are designed with duration of 0.625 ms or five taps each, in accordance with specifications found in literary references and protocols. [G168] [MARG93]

Figure 2-2 Echo Channel Model

The locations of the three echoes are centered at: 1) delay 60 ms, at tap 480, 2) delay 125 ms, at tap 1000, and 3) delay 187ms, at tap 1500.

Now that the skeleton of the echo channel model has been established, the next section will examine the presence of delay jitters and timing jitters.

2.3.2 Timing Jitter Model

Due to the asynchronous nature of the networks, a slight clock rate difference between the transmitting and receiving ends is expected. This results in a slight difference between the sampling rate of the reference signal sent into the echo canceller and the echo signal from the echo channel. When DPLL (Digital Phase Locked Loop) can't compensate the phase differences, a sample loss will occur if the clock on the receiver side is faster. Otherwise, there will be a sample deficit, and for this case, the substitution strategy could be either by the last sample prior to the sample loss or simply by zeroes. In this thesis, the previous sample is chosen.

2.3.3 Network Delay Jitter Model

Dynamic network features such as routing methods and traffic conditions in the network cause the delays of echoes to be always variant. Accordingly, the impulse response of the echo channel undergoes changes that severely degrade the performance of the echo canceller.

In the simulation, the delay jitters are designed to occur randomly such that at each time instance, the probability of the occurrence of a delay jitter is in the range of 1/5000-1/10000.

Chapter 3. BACKGROUND REVIEW

This chapter provides necessary background information for the understanding of this thesis and discusses previous research, where the echo path has a sparse type impulse response.

The history of echo cancellers will first be briefly reviewed, including an introduction to the evolution of echo cancellation and the echo canceller's FIR (Finite Impulse Response) filter structure. Following this some typical algorithms suitable for sparse channel models will be discussed.

3.1 *Historical Review of Echo Cancellers*

3.1.1 Echo Suppressor

Echo suppressors were the first approach to eliminate the effect of echoes on the telephone lines. The main idea behind echo suppressors is that when a person speaks, all signals from the opposite direction are attenuated significantly, preventing the audibility of echoes. But this brute-force solution would, apparently, suppress the incoming signal from the opposite direction as well, which makes full-duplex transmission impossible. Additionally, when the echo suppressor changes directions, the initial segment is clipped off, resulting in an awkward and disjointed conversation.

3.1.2 Echo Canceller

Once adaptive filtering theories were developed in the field of digital signal processing, their applications were quickly extended to echo cancellation [SOND66]. An echo canceller

is an adaptive filter that can adaptively model the echo channel's impulse response and duplicate the echo signals produced by the echo channel. When the duplicated signal is subtracted from the signal received from the channel, the echoes are eliminated.

Figure 3-1 A Simple Echo Cancellation Configuration

Figure 3-1 shows a typical echo canceller configuration, where $c(n)$ is the impulse response of the echo channel, $w(n)$ are the coefficients of the echo canceller, $x(n)$ is the reference signal from the talker at the near end, $d(n)$ is the desired signal from the listener at the far end, and $e(n)$ is the error signal (the signal from the far end $d'(n)$, minus the output of the echo canceller $x(n)*w(n)$). The output signal of the echo canceller is used to duplicate the echo signal. When it is subtracted from the signals originating from the far end, in principle, the talker at the near end will only receive the signal originally from the listener at the far end. ERL (Echo Return Loss) is used to measure the effect of the echo channel to the input signal $x(n)$ and ERLE (Echo Return Loss Enhancement) is used to evaluate the performance of the echo canceller on attenuating the echo signal.

The development of echo cancellers is a big leap from echo suppressors, because echoes can be eliminated without attenuating the incoming signal from the listener at the far end and full-duplex transmission is then realized. There are a good variety of methods used for echo cancellation, all of which, however, utilize adaptive filtering algorithms.

3.2 Adaptive Filters and Algorithms

The term “adaptive filter” refers to filters whose coefficients could be optimized according to certain criteria. The most commonly used criteria are LMS (Least Mean Square) and LS (Least Square).

3.2.1 Filter Structure

There are many types of adaptive filters, but for the purpose of this thesis, FIR filters are used instead of IIR (Infinite Impulse Response) filters, and only the most popular one, the transversal filter, is used because it is one of the most efficient and has the simplest format.

The transversal filter has a simple FIR filter configuration that is composed of delay, addition and multiplication, as shown in Figure 3-2. Signal $x(n)$, $n = 0, 1, \dots$, passes through a filter h_n , $n = 0, 1, \dots, M-1$, where M is the order of the filter, and:

h_n : the coefficients of the filter

z^{-1} : the unit delay

Σ : symbol of summation

$x(n)$: input signal

$y(n)$: the output signal of $x(n)$ through the filter h_n , which has been defined as the convolution of $x(n)$ and h_n .

$$y(n) = x(n) * h_n$$

$$= \sum_{k=0}^{M-1} x(k)h_{n-k} \quad n = 0, 1, \dots$$

Where $*$ denotes convolution.

Figure 3-2 Structure of a Transversal Filter

This filter structure is considered to be one of the best linear filter structures for echo cancellers, and is used in this thesis. [G165][G168][HAYK96]

3.2.2 Filtering Algorithms

In this section, a review of the adaptive algorithms used in this thesis is presented. All of these algorithms are variations of the LMS algorithm.

The following algorithms are listed in complex form, although the simulations in this thesis only used real data.

3.2.2.1 Full-tap NLMS

The standard Normalized Least-Mean Squares (NLMS) algorithm is referred to as Full-tap NLMS in this thesis to differentiate it from the sparse-tap algorithms mentioned hereafter.

Assume that the echo channel can be approximated by an N-order transversal filter, using the notations in Figure 3-1: the input signal $x(n)$ formed an N by 1 vector at time n, $\mathbf{X}(n)$, and the corresponding adaptation functions are:

$$y(n) = \mathbf{W}^H(n) \mathbf{X}(n) \quad (3-1)$$

$$e(n) = d(n) + \mathbf{C}^H \mathbf{X}(n) - y(n) \quad (3-2)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu \mathbf{X}(n) e^*(n)}{|\mathbf{X}(n)|^2} \quad (3-3)$$

Where the symbols in the algorithm represent the following quantities:

N: filter order

n: adaptation time

$d(n)$: the desired signal at time n

$y(n)$: filter output at time n

$e(n)$: the estimation error at time n

μ : the step size, where $0 < \mu < 2$

$$\mathbf{W}(n) = \begin{bmatrix} w(0) \\ w(1) \\ w(2) \\ \vdots \\ \vdots \\ w(N-1) \end{bmatrix} : \text{the filter's coefficients vector after nth iteration}$$

$$\mathbf{C}(n) = \begin{bmatrix} c(0) \\ c(1) \\ c(2) \\ \vdots \\ c(N-1) \end{bmatrix} : \text{the impulse response vector of the echo channel}$$

$$\mathbf{X}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \vdots \\ x(n-N+1) \end{bmatrix} : \text{the input signal vector to the echo channel and the echo canceller}$$

See also Haykin's book [HAYK96] for reference.

3.2.2.2 Sparse-tap NLMS

Cancellation of multiple echoes requires an adaptive filter with a large number of coefficients enabling it to span long delays. The number of coefficients can be reduced significantly by allocating a limited number of coefficients to the taps for each echo response, and the coefficients of the remaining taps are set to zero. NLMS is used for the adaptation of taps with non-zero coefficients.

Some sparse-tap filter concepts are introduced below, while concepts associated with certain algorithms are listed in the corresponding sections.

1) Dispersive region: In an echo path with multiple echoes, there are several versions of the same input signal, each one from different "impedance irregularities". In other words, the overall echo channel consists of several small echo paths. The equivalent impulse response

of the echo path can be shown as Figure 3-3, and the duration of the echoes is called dispersive region.

Figure 3-3 Dispersive Regions of a Typical Echo Channel Impulse Response

2) Constraint: When using a sparse-tap filter to simulate an echo channel, an arbitrary number must be used to estimate the dispersive region; and therefore, determine the size of the filter. This arbitrary number is called the constraint.

3.3 Echo Cancellation Methods Using Sparse-tap Filters

3.3.1 STWQ (Scrub Taps Waiting in a Queue)

The characteristics of the algorithm [KAWA86] are summarized as follows:

- 1) All taps are kept in two different queues: one active queue and one scrub queue.
- 2) Only the coefficients of the taps in the active queue will be updated after each iteration and used to calculate the replica of the echo.
- 3) After certain iterations, several taps with the smallest coefficients will be placed into the inactive queue and a corresponding number of taps will be selected from the inactive queue and made active.

See Figure 3-4 for reference.

Figure 3-4 STWQ (Scrub Taps Waiting in a Queue)

N : total number of taps required to cancel the echo

M : total taps required that have non-zero coefficients

$A(n)$: active queue $n = 1, 2, 3, \dots, M$

$S(n)$: scrub queue $n = 1, 2, 3, \dots, N-M$

Before the details of this algorithm are discussed, some concepts must first be explained.

1) Active queue: This queue contains all the taps that are used to duplicate the echo channel. The length of this queue should be selected at least to cover the sum of all the dispersive regions.

2) Scrub queue: This queue keeps all the taps with insignificant coefficients. When the canceller is operating, the coefficients of these taps are set to zero.

3) Elimination rate: The number of taps that are eliminated from the active queue to the scrub queue when required. If this number is too small, the algorithm will converge slowly; if it is too large, this may either increase computational expense or even cause divergence.

4) Significant taps: Those taps that have non-zero coefficients after convergence are considered to be significant taps.

Based on the concepts mentioned above, the detailed scenario is introduced as follows:

- a) Calculate the total number of taps needed, N , according to the maximum echo path delay.
- b) Choose M taps from the N taps according to the dispersive region to make an active queue. Put the remaining $(N-M)$ taps into a scrub queue. The M taps can either be selected randomly from the N total taps or according to some preset criteria. Both will yield similar results.
- c) Iterate for a certain time, usually until the filter with active taps converges, i.e. approximately two to five times of the filter's length. Denote this number as Q .
- d) If the elimination rate is m , choose m taps from the active queue with the least significant coefficients, make them inactive and put them into the scrub queue. Select m new taps from the scrub queue, usually according to the FIFO (First In First Out) order and activate them.
- e) Repeat steps c and d.

The procedure used to determine the size of the active queue is worth noting, because the elimination process will never cease. In other words, as in the above explanation, there is not a criterion to stop steps c and d. Choosing the proper size of the active queue has to take into account the elimination rate. If the size of the active queue just covers all the dispersive regions, it is possible that significant taps will be put into the scrub queue. Because of the large N , a good number of iterations might be required to update them into the active queue again. And even when the echo canceller has converged, where converged means all significant taps are in the active queue, after certain time, m taps which are significant taps will still be removed and placed into the scrub queue. To solve this problem, the length of the active queue should be at least equal to the sum of the dispersive region and the elimination

rate. This solution will prevent the algorithm from diverging if a big elimination rate has been chosen, but the algorithm will suffer from unnecessary computation load.

3.3.2 Variations of STWQ

There are several variations of STWQ that may perform better under certain situations. The Two-stage STWQ and the Fast STWQ algorithms are the most popular examples.

Two-stage STWQ

A Two-stage STWQ [IKED94] usually consists of a flat-delay estimation and a tap-position control with a constraint.

a) Flat-delay estimation

M active taps are selected out of N total taps, and the rest are kept in a scrub queue. After K iterations, a detection rate for each tap (the maximal absolute coefficient) is calculated according to the latest iterations.

Active taps are shifted with one tap and the above procedure is repeated until the maximum detection rate reaches a threshold. The tap with the maximal absolute value is denoted $I_{\max}$.

b) Tap-position control with a constraint

Active taps are updated using STWQ. The only difference between this method and the conventional STWQ is that only the taps that fall into the range $I_{\max} \pm L$ will be updated, where L is the constraint, and it must be large enough to cover the echo.

When compared with conventional STWQ, Two-stage STWQ does have some advantages in canceling single echoes. Two-stage STWQ converges fast with less computational complexity. However, its elaborate updating rule is not designed to handle multiple echoes.

Fast STWQ (grouping method)

This algorithm was first proposed by Shigeji Ikeda and Akihiko Sugiyama [IKED94] for the sparse-tap adaptive FIR filter to cancel an unknown number of echoes in the telephone line. Some concepts associated with this algorithm are described below.

- 1) Hopping order: A number used to mark the sub-groups, with the most significant sub-group having a hopping order of 1.
- 2) Staying time: The iteration time for each sub-group according to the hopping order; the smaller the hopping order, the longer the staying time.

N total taps are divided into S sub-groups, with each sub-group having its own scrub queue. The hopping order is set the same as the order of the sub-group and the staying time for each group T_s is set as T_0 initially.

M active taps are selected from group 1 for the first time and the first active group is set to group 1 and iterated T_0 times. The active taps will still be kept in the active queue, but the group is made inactive. Each of the groups is selected and made active sequentially according to the hopping order until all have been selected once.

The hopping orders are then modified according to the sum of the coefficients in each sub-group and the staying times are also calculated according to certain rules. These procedures are repeated until the stop criterion (such as total iteration time, ERLE, etc.) is satisfied. See Figure 3-5 for reference.

Figure 3-5 Grouping Method

3.3.3 TDE (Time Delay Estimation) Based Sparse-tap NLMS

Because the impulse response of a network echo-channel is sparse in nature, only the portions of impulse response with non-zero values contribute to the generation of echoes. If calculating the cross-correlation of the input signal to the echo channel and the output signal of the echo channel, the positions of the echoes in the echo channel's impulse response can be determined [YIP87]. There are several publications that state the principles of this method

[CHEN89] [HIRA96] [MARQ97], but basically the cross-correlation reflects the relationship between the input signal and the echo signal. This is shown in Figure 3-6.

Figure 3-6 Time Delay Estimation Signal Sequences

The positions of the echoes can be located according to the peaks of the cross-correlation between the input signal and the echo signal, if the channel is noise free or the noise is uncorrelated to the input signal $x(n)$.

Using the notations in Figure 3-6, the cross-correlation of $x(n)$ and $y(n)$ is:

$$\begin{aligned}
 R_{xy}(\tau) &= E[x(k+\tau)y(k)] \\
 &= E[x(k+\tau)(x(k)*c(k)+v(k))] \\
 &= E[x(k+\tau)x(k)*c(k)] + E[x(k+\tau)v(k)] \\
 &= R_{xx}(\tau)*c(k)
 \end{aligned} \tag{3-4}$$

where $E[x]$ denotes the expected value of x . When assuming the input signal $x(n)$ is white (which is valid for data and approximately valid for voice signals), we have:

$$\begin{aligned}
 R_{xx}(\tau) &= \sigma_x^2 & \tau = 0 \\
 &= 0 & \tau \neq 0
 \end{aligned} \tag{3-5}$$

where σ_x^2 is the energy of $x(n)$, also the variance of $x(n)$.

Then from (3-5), (3-4) can be derived to:

$$R_{xy}(\tau) = \sigma_x^2 C(\tau) \tag{3-6}$$

Therefore, the locations of the peaks of the cross-correlation are the same as the positions of the echoes. This relationship holds for the noise-free and with practical SNR cases. This is the underlying principle of the TDE method. Based on equation (3-6), the TDE method first calculates the positions (delays) of the echoes, then uses multiple small filters centered at the calculated delays to form the echo canceller.

The structure of the echo canceller for TDE is shown in Figure 3-7.

Figure 3-7 The TDE Method Echo Canceller Structure

For implementation efficiency, several approaches for cross-correlation calculation have been proposed to identify the delays of the echoes [JACO93] (see Appendix A) and also a comparison between difference time delay estimators has been performed [FERT86]. In this thesis, DC (Direct Cross-Correlation) is used, because it is very straightforward compared to other estimators [HAHN96] [JEBA96].

When using the TDE methods, two different strategies can be employed. One uses center recalculation and the other does not.

1) TDE method without center recalculation:

a) Assume there are n echoes in total that need to be eliminated. Choose an arbitrary constraint L , which is usually determined according to statistical results about the largest possible dispersive region of an echo. If the dispersive region is l , L must at least be equal to l . In this thesis, L can be set manually during simulation and in most cases, L is chosen to be equal to $2 \cdot l$.

b) Estimate the centers of echoes by calculating the cross-correlation of the input signal and the echo signal and finding the largest n positions, which are the delays of the echoes.

c) Center short filters with length L at the calculated delays.

d) Update the coefficients of the short filters.

2) TDE method with center recalculation:

The first four parts of this method are identical to those of the TDE method without center recalculation, and an additional part e is added:

e) Whenever an ERLE drop is detected, either caused by a delay jitter or a sample loss, the centers of the delay will be recalculated.

All these methods were developed for conventional networks. The performances of these methods are severely inhibited when timing jitters and delay jitters occur during the echo cancellation procedure (as will be demonstrated in the next chapter, along with the evaluation of these methods). A new method must be developed to solve the echo cancellation problem in current telephone networks. This could be achieved either by a modification of these known methods, or by using an entirely new approach. For this reason, the known methods were evaluated and analyzed as the first step in this research.

Chapter 4. Evaluation of Different Algorithms

Simulations were used to evaluate well-established methods in order to study the performances of the existing echo cancellation methods in the networks and develop new adaptation algorithms to suit the current conditions. The echo cancellation methods that were used in the simulations are: Full-tap NLMS (NLMS), TDE without Center Recalculation (TDENCR), TDE with Center Recalculation (TDECR), and Scrub Taps Waiting in a Queue (STWQ).

When speaking of the performance of a certain algorithm, we usually refer to its speed of convergence, its tracking ability, the ERLE level at convergence, and its computational complexity. [G168][HAYK96]

In this chapter, the four methods are first studied under the condition of a static echo channel, without timing jitters, delay jitters, or any fast changes. Then, they are evaluated in the presence of delay jitters or timing jitters respectively.

Before the presentation of the simulation results, the interface used to set the simulation parameters will be introduced briefly in section 4.1.

4.1 *The Interface*

The simulation interface provides a user-friendly graphic window with general instructions, which makes it not only useful for this project, but also valuable for other people. This is shown in Figure 4-1.

4.2.1 General Performance

The ERLE-iteration plots for NLMS, STWQ, and TDEs are shown in Figure 4-2. Since in the simulation, the echo channel is assumed to be static, no center recalculation is required, and TDECR performs the same as TDENCR.

Figure 4-2 Speed of Convergence

All the parameters used to obtain the simulation results in Figure 4-2 are identical except that Full-tap NLMS uses full filter taps for adaptation (2501 taps) and STWQ and TDEs use a smaller number of taps (30 out of 2501).

The plots show that all the methods converge to a similar ERLE level and it is obvious that the NLMS method has a significantly slower speed of convergence compared with the sparse filtering methods. It is also worth noting that the TDENCR and TDECR methods, as a

result of accumulating samples to compute cross-correlation, start adaptation later than other methods and therefore achieve convergence later than STWQ. Since the cross-correlation is only computed once in this case, neither the advantages nor the severe side-effects of TDECR are shown.

As for computational expenses, Full-tap NLMS needs more computations than other methods per iteration. As a result of the center recalculation strategy of TDECR, STWQ becomes the most economical method in terms of computation, and TDENCR is the second best for the static echo channel.

4.2.2 The Selection of Filter Parameters

The selection of filter parameters has a remarkable effect on the performances of these echo cancellation methods. Here we focus on the following most important parameters: the constraint and the number of small filters in TDENCR and TDECR (see 3.3.3), and the elimination rate (see 3.3.1) in STWQ.

4.2.2.1 Choosing Constraint

The constraint is used to estimate the range of the dispersive region for each echo. It should, therefore, be small to reduce the computational complexity but big enough to cover the full range of the echo.

If the constraint is set to be twice the desired value, it will double the number of computations. But the number of active taps (constraint times the number of small filters) is very small, usually in the order of 20-30 taps. As a result, the computation will not increase significantly even when the constraint is doubled. On the other hand, if the constraint is chosen to be smaller than the desired one, it will decrease the computation cost as expected,

but it may also cause divergence. Accordingly, this number is always chosen to be bigger than the range of the echoes, which could be obtained through statistical analysis. [G168][YIP87]

The effects of choosing different numbers of active taps are shown in Figure 4-3.

Figure 4-3 Effect of Constraint in TDE

According to the echo channel impulse response model designed for this project (Figure 2-2), the desired number of active taps is 15 (5 taps multiplied by 3 echoes). The number of small filters is three in above simulations, so when constraint = 5, the number of active taps is 15. When constraint is equal to 10, the result doesn't show much difference, but when constraint is chosen to be 2, it converges at a very low ERLE level.

4.2.2.2 Choosing the Number of Short Filters in TDE

This number is used to estimate the number of echoes in the entire echo path. This number has to be set before the establishment of the echo canceller.

Compared to the size of the constraint, this aspect is relatively less important. Even if this number is chosen a little smaller than desired, only the smallest and furthest echoes will be ignored. If this number is chosen too big, there are several imaginary echoes, which will only introduce redundant computation. In both cases, the performance of the echo canceller will not be influenced tremendously. However, if the number is selected too small, which means significant echoes will be neglected, the performance of the echo canceller will be degraded.

The effect of choosing different numbers of short filters is shown in Figure 4-4.

Figure 4-4 Effect of The Number of Short Filters

In the above simulation, a small echo with longer delay is added to the designed model to achieve an effect whereby, even if it is neglected, the performance won't be influenced severely. In the figure above, when using four and three small filters to estimate the echoes, similar results are achieved. However, when using two small filters, the echo canceller failed to perform properly, as in this instance, a large echo is ignored.

4.2.2.3 Choosing Elimination Rate

In the STWQ algorithm, the elimination rate is a factor directly related to its convergent and also tracking aspects.

Figure 4-5 and Figure 4-6 show the performance of STWQ under different elimination rates. Clearly, the convergence speed is increased along with the increasing of the elimination rate. And as a side effect, the computational expense also goes up.

Figure 4-5 STWQ with Elimination rate = 1

(a)

(b)

(c)

Figure 4-6 STWQ with Different Elimination Rate (a)=10, (b)=100, (c)=100,150,200

The optimal choice could be determined. But ascertaining this optimum number is beyond the scope of this thesis.

4.3 Effect of Delay Jitters on Echo Cancellation

This section explores the performance of the four echo cancellation methods in the presence of delay jitters. The detection of the jitters' occurrence is a crucial issue for TDECR.

4.3.1 The Detection of Delay Jitters

A mechanism for jitter detection is incorporated into all kinds of TDECR methods. One example is the Center of Gravity [MARG93]. Unfortunately, this mechanism is only capable

of detecting jitters after the algorithm has converged. The recalculation procedure of TDECR is frozen before the convergence, and therefore it performs identically to TDENCR.

Assuming the detection is reasonably accurate, the performance of the echo canceller before the convergence is very unpredictable. If the occurrence of delay jitter is relatively frequent, the algorithm may never converge to a desired level, and thus the detecting method may never have the chance to play a role. This is shown in the simulation results.

A statistical approach has been developed in this project that is capable of detecting channel changes at any time during the adaptation process with only a small probability of error. This will be discussed, in more detail, in Chapter 5.

This approach is imported into TDECR to improve its performance. In all simulations associated with delay jitters in this thesis, the TDECR method includes this detecting method.

4.3.2 Simulation Results

The ERLE-iterations plots for the four methods when delay jitters are present in the channel are shown in Figure 4-7 to Figure 4-10.

Figure 4-7 TDENCR for Delay Jitter

Figure 4-8 TDECR for Delay Jitter

Figure 4-9 STWQ for Delay Jitter

Figure 4-10 NLMS for Delay Jitter

In comparing the four plots, the following observations can be made:

- 1) The speed of convergence (the slope of the plot) for each method is inherently the same as when the echoes are static. All four methods experience an instantaneous drop of ERLE when a jitter occurs. The recovery curves for ERLE are similar for NLMS, TDENCR, and STWQ. But for the TDECR method, it depends on when the next jitter occurs. More specifically, if a subsequent jitter arrives at a time instant such that the center recalculation is not affected, then the correct re-determination of delays will bring the ERLE level back up to the level prior to the previous jitter. In such a situation, the number of iterations required is at least the duration of the channel's impulse response. On the other hand, if the next jitter arrives at a time instant close to the end of the sample accumulation for the delay recalculation, the echo centers will be incorrectly determined. As a result, TDECR will not perform any better than TDENCR.
- 2) The tracking abilities for the four methods, in the presence of delay jitters, are all rather poor. Though TDECR could be better than the other algorithms, it is not always true. This can be explained by the fact that rapid channel variations such as delay jitters are indeed beyond the tracking capacities of these methods. The plots show that all methods, except TDECR, restart the adaptation, upon receiving jitters, almost from the initial ERLE level. Therefore the tracking abilities for the three methods depend almost completely on the speed of convergence. For TDECR, the recalculation of delays serves as a tracking mechanism. When jitters arrive at a very low frequency, it can track the channel change after accumulating enough samples to calculate the cross-correlation. If the jitters arrive at relatively higher frequency, this tracking mechanism fails completely.

- 3) The ERLE levels that the four methods can achieve, again, depend on their speeds of convergence and the frequency of jitters. In the situations where all the methods fail to track the channel changes, the duration of the adaptive filters' operation at very low ERLE is set by the frequency of jitters, and the highest ERLE level a method can achieve depends on its inherent speed of convergence. The TDECR method has better tracking ability, which renders it a higher achievable ERLE level. This ERLE level depends on the longest interval where there are no two jitters very close to each other. It can be further deduced that for jitters arriving at higher frequency, in which case the center recalculation fails to track the jitters, the TDECR method can reach an ERLE level no greater than that of TDENCR.
- 4) The computational complexities for STWQ, TDENCR and NLMS are the same as those in the absence of channel variations. But TDECR requires more computations for the center recalculation when delay jitters are detected.

4.4 Effect of Timing Jitter

As stated in the previous chapter, timing jitter may cause sample loss or sample shortage when the echo signal from the echo channel enters the echo canceller because of the sampling rate difference of the two sides. In general, if the sampling rate is faster on the side in which the echo channel is located, sample loss will happen, otherwise there will be sample shortage.

The performances of the four echo cancellation methods will be studied under three categories of sample loss. They are: Single Sample Loss (one sample is lost at one time, SSL in short), Multiple Sample Losses (several samples are lost at the same time, MSL in short),

and re-sampling problem (signal is re-sampled from an analog signal, RS in short). In the first two categories, two scenarios are simulated.

Scenario one: In this scenario there is no clock difference between the echo canceller and the entry port of the channel. The entry port is the part of the network model prior to the echo channel, while the exit port (which will be discussed later) is the part of the network model posterior to the echo channel.

Scenario two: In this scenario, there is a clock difference between the echo canceller and the entry port of the channel.

Detailed descriptions and simulation results are shown in sequence.

4.4.1 One Sample Loss or Shortage

Scenario one: There is no clock difference between the echo canceller and the entry port of the echo path, but there is a clock difference between the exit port of the echo channel and the echo canceller.

Figure 4-11 Sample Loss, One Sample at a Time

In Figure 4-11, the echo canceller operates at rate F_s , but the echo signal is at rate F_s' , and $F_s < F_s'$. Signal X_n enters the echo path and the echo canceller at the same rate F_s .

X_n passing through the echo path, the echo Y_n is generated and enters the echo canceller. But the echo canceller is functioning at a lower rate, so a sample will be lost every certain number of iterations depending on the difference between F_s and F_s' . In Figure 4-11, Y_5 and Y_{10} are lost, assuming $F_s = 4$ and $F_s' = 5$.

Figure 4-12 Sample Shortage, One Sample at a Time

If $F_s > F_s'$, there will be a sample shortage occurring over time depending on the difference between F_s and F_s' . This is shown in Figure 4-12. Y_4 is used twice because Y_5 is not available at the time required (assuming $F_s = 5$ and $F_s' = 4$).

Simulation results are shown in Figure 4-13 to Figure 4-16. On each page, the upper figure represents sample loss and the lower one represents sample shortage.

2501 taps TDENCR, $\mu = 0.27$, 1 sample is lost every 10000 iterations

2501 taps TDENCR, $\mu = 0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-13 TDENCR for SS Scenario One

2501 taps TDECR, $\mu = 0.27$, 1 sample is lost every 10000 iterations

2501 taps TDECR, $\mu = 0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-14 TDECR for SS Scenario One

2501 taps STWQ, $\mu = 0.27$, 1 sample is lost every 10000 iterations

2501 taps STWQ, $\mu = 0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-15 STWQ for SS Scenario One

Figure 4-16 NLMS for SS Scenario One

These plots show that in this situation the four methods perform similarly to those in the case of delay jitters. This is not surprising, since the effect of sample loss to the echo canceller's impulse response is precisely equivalent to that of a delay jitter. The only difference is that one occurs regularly and the other occurs at random. Therefore we omit the discussion on the performances of these methods and only point out that the regularity of this kind of sample loss in fact permits a deterministic and quantitative analysis of the performances. For example, the condition under which TDECR fails to track a channel change can be determined by the two sampling frequencies.

To conclude this section, the following question is put forth:

Knowing such sample losses correspond to a shift of the channel's impulse response by ± 1 tap, is it necessary to recalculate the centers as done in TDECR and result in long period of waiting time?

Scenario two: There is a clock difference between both the echo canceller and the entry port of the echo path and also the exit port of the echo channel and the echo canceller.

Figure 4-17 Sample Loss, One Sample at a Time

In Figure 4-17, the echo canceller is functioning at rate F_s , while the sampling rate of the echo channel is $F_{s'}$, and $F_s < F_{s'}$. When the input signal X_n enters the echo channel, there is a clock difference, and because $F_{s'} > F_s$, after some time, one previous sample has to be reused due to sample shortage. In the above figure, Y_4 and Y_8 are used twice, assuming $F_{s'} = 5$ and $F_s = 4$. When the echo signal Y_n from the echo path enters the echo canceller, because the latter is at a lower rate, after a certain time interval, a sample loss occurs. In the example above, Y_5 is lost when assuming $F_{s'} = 5$ and $F_s = 4$.

Figure 4-18 Sample Shortage, One Sample at a Time

Conversely, if the clock on the echo path side is slower, $F_s > F_{s'}$, when the input signal X_n enters the echo channel, a sample loss will occur after some time. In Figure 4-18, for example, Y_5 and Y_{10} are lost, when assuming $F_s = 5$ and $F_{s'} = 4$. When the echo signal Y_n from the echo channel enters the echo canceller, a sample reuse will occur after some time due to the fact that F_s is faster than $F_{s'}$. In the above example, Y_4 is reused when assuming $F_s = 5$ and $F_{s'} = 4$.

2501 taps TDENCR, $\mu = 0.27$, 1 sample is lost every 10000 iterations

2501 taps TDENCR, $\mu = 0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-19 TDENCR for SS Scenario Two

2501 taps TDECR, $\mu = 0.27$, 1 sample is lost every 10000 iterations

2501 taps TDECR, $\mu = 0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-20 TDECR for SS Scenario Two

Figure 4-21 STWQ for SS Scenario Two

ERLE for 2501 taps NLMS, $\mu=0.27$, 1 sample is lost every 10000 iterations

2501 taps NLMS, $\mu=0.27$, previous sample is reused 1 time every 10000 iterations

Figure 4-22 NLMS for SS Scenario Two

The following conclusions can be drawn from the simulation results in Figure 4-19 to Figure 4-22. Again the upper figure shows sample loss, and the lower figure shows sample shortage:

1) All four algorithms perform similarly in terms of converging and tracking, though NLMS is slower than the others. For each occurrence of a sample loss or sample shortage, the performance of the echo canceller degrades, as indicated by the drop of the ERLE level. After the drop, the curve rises first relatively quickly then more slowly to the same level as that before the drop and adaptation continues normally.

2) NLMS achieves a lower ERLE level after the same iteration time as other algorithms.

3) TDECR and TDENCR show not much difference, because the centers of delays didn't change (details as follows). Though TDECR attempts to recalculate the centers, the extra computational expense will nullify any gains.

Take the sample loss case as an example to explain the performance of the echo canceller.

Due to the clock differences between the echo canceller and the echo channel, there is a reused sample of the input signal that enters the echo path but not the echo canceller every certain time period. This sample will have an impact on the echo canceller until it exits the echo channel. Thus it will change not only the delays of the equivalent channel, but also the channel's impulse response.

Meanwhile, a sample of the echo signal from the echo channel will be lost periodically. This loss could be treated as a channel shift by the echo canceller. Combined with the previous channel shift, the overall effect of the sample loss will not have any impact on the echo delays. But the sample of the input signal that enters the echo path but not the echo canceller will affect the echo canceller's performance throughout the entire echo channel's

In Figure 4-23, for example, if assuming $F_s = 4$ and $F_s' = 5$, Y_5 should be lost when the echo signal enters the echo canceller. But in this case, Y_5 is saved and as a consequence, every sample is buffered until Y_9 arrives and is lost together with Y_{10} .

The effect of multiple losses is shown in Figure 4-24 to Figure 4-27.

Figure 4-24 TDENCR MS Scenario One

Figure 4-25 TDECR MS Scenario One

Figure 4-26 STWQ MS Scenario One

Figure 4-27 NLMS MS Scenario One

From Figure 4-24 to Figure 4-27, the following are observed:

- 1) For TDENCR, STWQ and NLMS, the inherent speed of convergence and tracking ability do not change. The performance of the echo canceller, however, is improved because it is able to achieve a higher ERLE level, by allowing these methods to have longer adaptation time between sample losses.
- 2) For TDECR, it doesn't show much difference when compared to the result of single sample loss, because accumulating sample losses is equivalent to shifting the channel by ± 2 , ± 3 , or more, depending on the sample loss scheme. The only advantage is that by reducing the number of center recalculations, the computational expense is decreased.

Though for the other three methods except TDECR, this innovation does provide them with more adaptation time between each loss, and the performance is improved, their tracking

abilities are still poor, even when compared with TDECR. While the computational expense is reduced in TDECR, it still needs a period of time to accumulate samples to calculate the cross-correlation, and during this interval, the ERLE level will still remain low, as before.

Scenario two: Refer to the single sample loss section for description.

Case one: The clock rate on the echo canceller side is slower than that of the echo path side.

Figure 4-28 Multiple Sample Losses Scenario Two, Case One

In Figure 4-28, because $F_s < F_{s'}$, a sample will be reused periodically when the input signal enters the echo channel. Assuming $F_s = 4$ and $F_{s'} = 5$, X_4 is reused. Y_5 should be lost when the echo signal enters the echo canceller, but in this case, Y_5 is saved and as a consequence, every sample after it is buffered until Y_9 arrives and is lost together with Y_{10} .

The simulation results of different algorithms are shown in Figure 4-29 to Figure 4-32.

Figure 4-29 TDENCR MS: Scenario Two, Case One

From the observation of these figures, several points can be observed:

- 1) The performances of the four algorithms are similar, though the convergence and tracking ability of NLMS are still worse than the other algorithms.
- 2) From the figures, two ERLE drops can be observed. Unlike scenario one, there is a sample reuse at the entry port of the echo channel. This reuse can not be treated the same as the sample losses because it cannot be accumulated. That is the reason for the first drop. The second reused sample at the entry port of the echo channel and the two sample losses at the echo canceller cause the second drop.

- 3) Although TDECR can achieve a higher ERLE level after a certain time interval than all the other algorithms, its tracking ability is still poor in this case and the effort used to recalculate the centers does not gain a corresponding improvement.

Figure 4-30 TDECR MS, Scenario Two, Case One

TDECR does have some advantages over the other algorithms in this situation, because the occurrence of a sample reuse when the input signal enters the echo path is involved in an equivalent echo channel change. When the delays of the echoes are calculated correctly and small filters are realigned, the ERLE level can be recovered. It cannot return to the same level as that before the reuse, for the reason that not only the delay of the channel changed but also the impulse response of the channel, in terms of the relationship between the input and output signal of the echo channel. However, when the second reuse occurs and simultaneously, two samples from the echo channel are lost, the channel shifts back to the

Figure 4-31 STWQ MS, Scenario Two, Case One

Figure 4-32 NLMS MS, Scenario Two, Case One

original position. Nevertheless, it is not possible for any algorithm to return directly to the ERLE level before the occurrence of the sample losses, because the amplitude of echo channel's impulse response changes.

Case two: The clock rate on the echo canceller side is faster than that of the echo path side.

Figure 4-33 Multiple Sample Losses Scenario Two, Case Two

In Figure 4-33, because $F_s > F_{s'}$, when the input signal X_n enters the echo channel, a sample will be lost periodically. For the same reason as mentioned at the beginning of this section, this sample could be saved. When assuming $F_s = 5$ and $F_{s'} = 4$, as in this example, X_5 is a sample that should be lost, but it is buffered, as well as the samples following it until X_9 and X_{10} are lost. When the echo signal from the echo channel enters the echo canceller, a sample will be reused when it is required. In the above example, Y_4 is reused.

The simulation results of this case are shown in Figure 4-34 to Figure 4-37.

Figure 4-34 TDENCR MS, Scenario Two, Case Two

Figure 4-35 TDECR MS, Scenario Two, Case Two

Figure 4-36 STWQ MS, Scenario Two, Case Two

Figure 4-37 NLMS MS, Scenario Two, Case Two

The observations and the explanation for the results of case two are similar to that of case one. Unfortunately, there still has been no significant progress made in applying the multiple-sample-loss strategy to the sample loss problem.

4.4.3 Re-sampling

Besides the scenarios discussed in the previous two sections, there is another situation also under consideration.

It is one of the cases that there is only sampling rate differences between the exit port of the echo channel and the entry port of the echo canceller. Because the mechanism of this sample loss or sample reuse is different from the ones in the previous two sections, it is explained separately.

Re-sampling is the term used to describe the procedure when a digitized signal is sampled at a different frequency, which includes down-sampling meaning that the new sampling frequency is slower than the previous one, and up-sampling meaning that the new sampling frequency is faster than the previous one. Refer to Figure 4-38 for an example of re-sampling.

In Figure 4-38, in the upper figure, a sine wave of one period is shown and sampled at 20Hz. In order to obtain a down-sampled version of this signal at 18.2Hz, the digitized signal is first converted into an analog signal and then re-sampled at 18.2Hz, as shown in the lower figure.

Figure 4-38 An Analog Signal Sampled at Different Rates

Another reason this situation is discussed separately from other cases is that this kind of sample loss or reuse will not degrade the performance of the echo canceller; the influence is positive. This can be shown in the following discussion and simulation by taking the sample loss case as an example.

In Figure 4-39, assuming the sampling rate in the near end is F_s , the sampling rate in the far end is F_s' , and $F_s' > F_s$.

Figure 4-39 Re-sample with Sample Loss

When the signal from the near end sampled at rate F_s enters the echo channel, it encounters some A/D-D/A operations, and does something unknown. Then, when the echo signal comes out of the echo channel, it is sampled by F_s' from an analog signal.

Because the echo canceller is operating at rate F_s and the echo signal is sampled at F_s' , after a certain time interval, a sample will be lost.

In the simulation, it is not possible to get an analog signal, so the signal is re-sampled in the digital domain.

The simulation results are shown below.

Figure 4-40 TDENCR with Resample

Figure 4-41 TDECR with Resample

Figure 4-42 STWQ for Resample

Figure 4-43 Full-tap NLMS for Resample

In Figure 4-40 to Figure 4-43, the following can be observed.

- 1) All four algorithms perform similarly.
- 2) The ERLE level drops gradually before the occurrence of a sample loss and after the sample loss, the ERLE level goes up correspondingly.
- 3) Full-tap NLMS achieves a lower ERLE level when converged than the other algorithms.
- 4) There is no difference between TDENCR and TDECR, because the impulse response of the echo channel does not change in this case; there is no center recalculation involved.

This kind of sample loss won't degrade the performance of the echo canceller for the reason that the echo signal has been re-sampled at a different frequency, so the samples are all slightly different from the samples that precede the resample procedure. Furthermore, this difference accumulates with time. When a sample is lost, this difference is eliminated correspondingly, so the proper reference signal and echo signal enter the echo canceller and continue the adaptation.

Having evaluated all the algorithms with a static channel, in the presence of delay jitters, and the presence of timing jitters, the existing problem is revealed explicitly. All the algorithms suffer from not having sufficient tracking ability to follow the quick channel changes. In light of this, a new method is proposed with a better tracking ability.

Chapter 5. PROPOSED ALGORITHM

5.1 *Introduction*

Simulations have shown that the TDECR is more promising than other existing algorithms in dealing with multiple echoes of long delay with the presence of delay jitters and sample losses. However, the performance of this method will be degraded severely by relatively frequent occurrences of delay jitters or sample losses. This can be understood from the following two aspects.

Firstly, since the echoes are of long delay, the recalculation of echo centers requires a correspondingly long sequence of samples. The frequent arrival of channel variations (sample loss or delay jitters) makes it quite probable that the sequence of samples for the center recalculation contains contradictory information in terms of the locations of the echo centers. As a result, the probability of an erroneous determination of echo center increases as the frequency of channel variation increases.

Secondly, as another effect of the echoes with long delays, the echo canceller implementing TDECR, upon detecting a jitter or sample loss, has to wait for a correspondingly long period to obtain enough samples (if no more channel variations occur in this period) to recalculate the centers. In this period, the echo canceller either freezes on the pervious configuration or keeps on adapting all over again. In both cases, the transmission will suffer from poor echo cancellation in this “waiting” period. From this analysis, it is natural to conclude that the recalculation of echo centers is neither reliable nor efficient for echoes of long delay coexisting with relatively frequent delay jitters or sample losses.

It has been observed that, in most communication networks, such channel variations are usually small in terms of the relative changes of the echoes' location with respect to the absolute delay of the echoes. This extra information leads to a new echo cancellation algorithm, which will be presented in this chapter, with a significantly improved performance upon the previously known methods.

This chapter will proceed as follows. First, the idea of the proposed algorithm will be introduced. Next some statistical analysis underlying the algorithm will be discussed. After that is the elucidation of the simulation using the algorithm in detail, followed by some results where the comparison between the new algorithm and the other algorithms will be performed.

5.2 The Idea of SEER

SEER, representing Squared Error Estimation and Realignment, employs a statistical approach to delay jitter detection and center re-determination.

Assume that the variation of the echo channel is within $[-\omega T, \omega T]$ and it has an impulse response $h(kT)$ with duration MT , where ω , k and M are positive integers, and T is the sampling interval.

Assume that the centers of the echoes in $h(kT)$ are at $C=[C_1T, C_2T, \dots, C_LT]$, where L is the number of echoes, and in the echo canceller's impulse response, the centers of the echoes are at $B=[B_1T, B_2T, \dots, B_LT]$. When there is no variation of the echo channel, $B=C$ at any time $t=kT$. When the echo channel varies, knowing $\omega < M$, the echo canceller works as follows:

- 1) When the canceller is adapting to simulate $h(kT)$, at any time instant $t=kT$, a statistical test T_1 is performed with the null hypothesis $H_1: \mathbf{B}=\mathbf{C}$. Upon obtaining the error $e(kT)$, H_1 is either rejected or accepted.
- 2) If H_1 is accepted, the canceller continues adapting and goes to 1)
- 3) If H_1 is rejected, store $\mathbf{B}$ in $\mathbf{B}_0$, go to 4)
- 4) Let $\Delta = \delta \mathbf{T}[1,1,\dots,1]$, a $1 \times L$ vector, where $\delta \in [-\omega, \omega]$ and $\mathbf{B}=\mathbf{B}_0 + \Delta$. Then a statistical test T_2 is performed with the null hypothesis $H_2: \mathbf{B}=\mathbf{C}$. Upon receiving the next $e(kT)$, H_2 is either rejected or accepted.
- 5) If H_2 is accepted, the canceller continues adapting and goes to 1)
- 6) If H_2 is rejected, choose another $\delta \in [-\omega, \omega]$ and go to 4)

The most essential problem is how to set up the rejection or acceptance criterion for H_1 and H_2 . It will become clear in the following sections.

5.3 Theoretical Foundation

5.3.1 Error Analysis

As noted above, the critical problem is how to determine the acceptance criterion for H_1 and H_2 . This will be elucidated in this section. All the results are derived based on the assumption that both signal and channel are real, though for the complex case, a similar conclusion can be achieved easily.

For detailed error analysis and the notations used in this section, refer to Appendix C.

5.3.2 The Statistical Tests

Having explored the statistics of J_n and $\tilde{J}_n$, we are ready to set up the statistical tests T_1 and T_2 .

5.3.2.1 T_1 Test

T_1 test serves to determine whether a delay jitter has occurred at time instant $t=nT$.

Assume the statistics of J_{n-1} can be obtained at the previous time instant $t=(n-1)T$, then the statistics of J_n and $\tilde{J}_n$ can be decided from equation (C-16) (C-17) or equation (C-19), (C-20) and equation (C-26) (C-27) respectively. The *pdf* of J_n and $\tilde{J}_n$ are sketched in Figure 5-1. A threshold Th_1 is to be selected such that, upon receiving the square error $\hat{J}_n$ at $t=nT$, where $\hat{J}_n$ denotes the observation of J_n or $\tilde{J}_n$. If $\hat{J}_n > Th_1$, the canceller claims a detection of channel variation, and H_1 is rejected. If $\hat{J}_n < Th_1$, the canceller claims no channel variation has occurred, and H_1 is accepted.

Figure 5-1 pdf of $\tilde{J}_n$ (red) and J_n (blue)

Associated with the test based on Th_1 , there are four conditional probabilities:

$$P_{T_1}(\text{correct rejection}) \triangleq P_{T_1}(\text{reject } H_1 | B \neq C) = \int_{Th_1}^{\infty} f_{\tilde{J}_n}(x) dx$$

$$P_{T_1}(\text{incorrect rejection}) \triangleq P_{T_1}(\text{reject } H_1 | B = C) = \int_{Th_1}^{\infty} f_{J_n}(x) dx$$

$$P_{T_1}(\text{correct acceptance}) \triangleq P_{T_1}(\text{accept } H_1 | B = C) = \int_0^{Th_1} f_{J_n}(x) dx$$

$$P_{T_1}(\text{incorrect acceptance}) \triangleq P_{T_1}(\text{accept } H_1 | B \neq C) = \int_0^{Th_1} f_{\tilde{J}_n}(x) dx$$

Among the four probabilities, the most critical one is $P_{T_1}(\text{incorrect acceptance})$, since an incorrect acceptance of H_1 means that the canceller has failed to detect a channel variation that has just occurred, resulting in a severe disturbance of the adaptation of the canceller. A

lower Th_1 is required to increase the sensitivity of the detection. Conversely, decreasing Th_1 will increase $P_{T_1}(\text{incorrect rejection})$, meaning that the canceller is more likely to claim a channel variation although there has not been one. This will increase the number of unnecessary computations. Thus, a compromise has to be made between $P_{T_1}(\text{incorrect acceptance})$ and $P_{T_1}(\text{incorrect rejection})$ when selecting the threshold Th_1 .

5.3.2.2 T_2 Test

When the echo canceller detects a channel variation, it stores the current coefficients as a backup and knowing the variation is small, it shifts its impulse response by δ taps, where $|\delta|$ is a small non-zero integer. It then performs test T_2 to check whether its impulse response is aligned to the impulse response of the channel (i.e. testing H_2). If H_2 is accepted, meaning it considers the two impulse responses to be aligned, the adaptation procedure is resumed; if H_2 is rejected, it selects a new δ and continues the search for the alignment until H_2 is accepted.

Similarly, T_2 is also performed based on the received value $\hat{J}_n$ for the square error random variable. In this case, the statistics of J_n and $\tilde{J}_n$ are obtained from the statistics of J_{n-1} prior to the detected channel variation, and this is fixed during the search for the proper alignment. A threshold Th_2 is selected for the acceptance criterion. Upon receiving $\hat{J}_n$, the echo canceller claims the alignment is found (H_2 is accepted) if $\hat{J}_n < Th_2$, and claims the alignment is not found (H_2 is rejected) if $\hat{J}_n \geq Th_2$. The four condition probabilities associated with test T_2 are:

$$P_{T_2}(\text{correct rejection}) \triangleq P_{T_2}(\text{reject } H_2 | B \neq C) = \int_{Th_2}^{\infty} f_{J_n}(x) dx$$

$$P_{T_2}(\text{incorrect rejection}) \triangleq P_{T_2}(\text{reject } H_2 | B = C) = \int_{Th_2}^{\infty} f_{J_n}(x) dx$$

$$P_{T_2}(\text{correct acceptance}) \triangleq P_{T_2}(\text{accept } H_2 | B = C) = \int_0^{Th_2} f_{J_n}(x) dx$$

$$P_{T_2}(\text{incorrect acceptance}) \triangleq P_{T_2}(\text{accept } H_2 | B \neq C) = \int_0^{Th_2} f_{J_n}(x) dx$$

A compromise between $P_{T_2}(\text{incorrect acceptance})$ and $P_{T_2}(\text{incorrect rejection})$ is made by selecting Th_2 .

Since tests T_1 and T_2 are indeed identical in nature, given that the statistics of J_n and $\tilde{J}_n$ are the same for T_1 and T_2 , the selections of Th_1 and Th_2 , in principle, can be made identical. However, in this algorithm, when performing test T_2 , the canceller's impulse response prior to detecting the channel variation has been stored. In this case, there is less risk for an incorrect acceptance of H_2 . In the worst case, the canceller can reset its coefficients to those prior to the detected channel variation and restart the alignment search. Therefore the performance of the canceller can only degrade to as low as that prior to the delay jitter or sample loss' occurrence. Further, if the canceller makes an incorrect acceptance of H_2 , with high probability, it will be detected by test T_1 at the next time instant. Furthermore, if Th_2 is low, the incorrect rejection of H_2 can cause the canceller to miss the correct alignment and have to wait until the same alignment is tested in the next round. For these reasons, the selection of Th_2 can be made more in favor of computational efficiency, i.e., by making Th_2 higher.

5.3.2.3 Further Discussion

To reduce the probability of incorrect acceptance or incorrect rejection, a large $E[\tilde{J}_n]/E[J_n]$ ratio is desired. Let ρ denote this quantity and Th denote the threshold for the test. Let $\mu = \frac{Th}{E[J_n]}$.

The probability of incorrect rejection is only a function of μ , invariant of ρ , or, more precisely

$$P(\text{wrong rejection}) = \int_{\mu}^{\infty} \frac{1}{\sqrt{2x\pi}} e^{-\frac{x}{2}} dx$$

However, the probability of incorrect acceptance, which is of more concern in this algorithm, is a function of both ρ and μ . More precisely:

$$P(\text{wrong acceptance}) = \int_0^{\mu} \frac{1}{\sqrt{2x\pi\rho}} e^{-\frac{x}{2\rho}} dx$$

A plot of $P(\text{incorrect rejection})$ versus μ is shown in Figure 5-2 The plot of $P(\text{incorrect acceptance})$ versus μ and ρ is shown in Figure 5-3.

Figure 5-2 P (incorrect rejection) versus μ

From the discussion in previous sections, we have shown that, at time of convergence, ρ has the same order as SNR. Then we can interpret Figure 5-3 roughly as the relationship between $P(\text{incorrect acceptance})$, SNR, and the threshold.

It is apparent from Figure 5-3 that increasing SNR can significantly reduce $P(\text{incorrect acceptance})$.

Figure 5-3 P (incorrect acceptance) versus μ and ρ

Another observation from equation (C-28) is that ρ will be relatively small before the canceller converges, due to the large value of $E[d_n^2(k)]$. Since we hope the exponential decay of $E[d_n^2(k)]$ rapidly decreases ρ , at the time before the canceller converges, a more conservative strategy in selecting the thresholds should be employed. This will result in a reduced probability of missing channel variations, at the cost of increased computational complexity.

5.4 Algorithm Implementation

As previously described, there are several key issues involved in the realization of the algorithm. First is the selection of the two thresholds Th_1 and Th_2 . Second is the utilization of TDE center recalculation.

5.4.1 Thresholds Determination

Two thresholds are used to trigger on/off the center realignment process. Unlike sample losses, the occurrences of delay jitters are not predictable for the echo canceller. Consequently, the detection of delay jitters relies on the first threshold Th_1 .

The determination of Th_1 becomes a compromise between $P(\text{incorrect acceptance})$ and $P(\text{incorrect rejection})$. More precisely, if it is selected too small, $P(\text{incorrect rejection})$ will increase, i.e., most probably, a certain number of non-delay jitters will be recognized as delay jitters. On the other hand, if it is too large, $P(\text{incorrect acceptance})$ will increase, and a considerable number of delay jitters will be missed. Wrongly detected delay jitters will only cause an increase of computational expense, while missed delay jitters may result in tremendous degradation of the echo canceller's performance. Therefore, choosing a small Th_1 is preferred to choosing a large one.

Another concern arises when examining equation (C-30 a). The second part of the dominant decreases with the procedure of adaptation, so $E[\tilde{J}_n]/E[J_n]$ increases accordingly. By intuition, after convergence, even choosing a larger Th_1 (compared with the Th_1 used when converging), we will not sacrifice $P(\text{incorrect acceptance})$ but obtain the decrease of $P(\text{incorrect rejection})$.

These aspects were considered in the design of the simulations:

- 1) Initially, μ is set to be 9, which will result in $P(\text{incorrect rejection})$ in the order of 10^{-3} and $P(\text{incorrect acceptance})$ in the order of 10^{-3} , when ρ is at the order of 10^{-2} .
- 2) With increasing iterations, μ is increased.
- 3) According to Figure 5-2, when μ is larger than 20, $P(\text{incorrect rejection})$ will not decrease, so it is not worthwhile to increase μ after this point. $P(\text{incorrect acceptance})$ will continuously decrease along with the adaptation procedure.

As for the determination of Th_2 , we prefer a high $P(\text{incorrect acceptance})$ to a high $P(\text{incorrect rejection})$, which will offer the echo canceller a chance to adapt rather than searching for the center forever. So Th_2 is chosen to be relatively larger than Th_1 , which is 15. This leads to a little higher $P(\text{incorrect acceptance})$ and a lower $P(\text{incorrect rejection})$ than that of $Th_1=9$ at the beginning of adaptation

5.4.2 Center Recalculation

Though $P(\text{incorrect acceptance})$ could be made very low by choosing μ properly, incorrect acceptances always exist. This is especially true for H_2 , because μ is set to be relatively high, and the probability of incorrect acceptance is even greater. Detecting this as a delay jitter and triggering on the center realignment procedure in the next few iterations could compensate this effect. Alternatively, if this incorrect acceptance cannot be recognized until enough samples have been accumulated for cross-correlation calculation, center recalculation is used to check the correctness of the center realignment. If the two results are consistent,

the delays and the echo canceller's coefficients from the center reallocation are formally accepted and all the backups before the occurrence of the delay jitter are purged. Otherwise, the delays are centered at the locations obtained from the center recalculation procedure, and the echo canceller's coefficients are reset to the backup from before the occurrence of the delay jitter.

5.4.3 Flow Chart

The flow chart on the next page is a brief summary of the implementation of SEER. Without mentioning any details of the chart, several notations must be clarified.

In the chart, Th_1 and Th_2 denote the two thresholds used to trigger on/off T_1 and T_2 tests; w is the size of the window used to calculate local variance, within which the variance can be considered to be time invariant, because w is chosen to be small. *Need* is a flag to indicate whether recalculation of centers of delays is needed.

5.5 *Simulation Results*

This section includes simulation results using the SEER method and a comparison to the TDECR method, because it is considered, among all the existing algorithms, to perform the best.

5.5.1 Delay Jitters

The problem of delay jitters can be handled more efficiently by applying the new detecting method and the center realignment procedure. Despite the fast convergence resulting from the quick center realignment, it may also achieve a higher ERLE level when converged, depending on the arrival frequency of the delay jitters. Which can be shown in Figure 5-4.

2501 taps SEER and TDECR, $\mu = 1$, 1 jitter happened with probability 1/8000

Figure 5-4 Delay Jitter Effect of SEER and TDECR (1)

Figure 5-4 shows that the speed of convergence for SEER and TDECR are the same in the absence of jitters. However, when jitters happen and are detected, SEER can return to the same ERLE level as prior to the occurrence of the jitter, while TDECR has to stay low until the centers are calculated. Only if the recalculation is correct, TDECR can return to the same ERLE level as before. The realignment process results in the difference of ERLE level between SEER and TDECR, 20 dB on average without any computational penalty.

When SEER and TDECR were evaluated in the presence of delay jitters, a very interesting result happened to be captured. The following explanation refers to Figure 5-5.

In TDECR, at iteration 10000, a channel change occurred, but during the monitoring of the simulation, it is found that a change is wrongly detected at iteration 8620. After enough

samples are accumulated, the cross-correlation is calculated at iteration 11122. Because most of the samples used to calculate it contain the information before the channel change, the centers of the delays are wrongly determined. After several iterations, the jitter detector finds out the previous calculation is wrong, and recalculate it. When the centers of delays are correctly calculated at iteration 15092, the adaptation has been proceeding for some time, so this correctness even causes ERLE to drop. Any wrongly detected channel changes will influence the performance of TDECR tremendously.

Figure 5-5 Delay Jitter with SEER and TDECR (2)

Next, examine the result of SEER. The detecting procedure is the same. However, when a jitter is detected at iteration 8620 although there is none), the center realignment process is not triggered on because $Th2 > Th1$. When a delay jitter happens at iteration 10000, the realignment process is triggered on until the correct centers (which are different from those

before the delay jitter) are located. After enough samples are accumulated, the cross-correlation is calculated at iteration 11122, and the centers of delays are calculated and checked with the one obtained from the realignment process. From the flow chart, we know that if the two centers are not consistent, the coefficients of the echo canceller will be reset to those existing before the jitter was detected (whether a jitter occurs or not). The coefficients are not reset, so the speed of convergence increases.

5.5.2 Sample Loss

This section will be organized similarly to chapter 4, except that MSL (Multiple Sample Losses) will not be discussed because MSL is an attempt to improve the performance of the echo canceller when sample losses are present. Also different scenarios in SS (Single Sample Loss) will be illustrated sequentially in comparison with TDECR.

5.5.2.1 Sample Loss for Scenario One

Figure 5-6 Sample Loss Scenario One SEER and TDECR

From Figure 5-6 above, the sample losses are revealed by observing ERLE drops and the effect is compensated for very quickly as shown by the recovery of the ERLE level. As for the sample shortage case, the situation is the same. SEER performs better in both cases.

As we can see, the speed of convergence and tracking ability of SEER are both better than TDECR. After convergence, the ERLE is increased by 20 dB in average with no increment in computation.

5.5.2.2 Sample Loss for Scenario Two

2501 taps SEER and TDECR, $\mu = 0.27$, 1 sample is lost every 10000 iterations

Figure 5-7 Sample Loss Scenario Two SEER and TDECR

Figure 5-7 shows that the speed of convergence and tracking ability of SEER are both better than those of TDECR. After convergence, the proposed method can achieve an ERLE level of 60 dB while TDECR can only achieve 40 dB. Furthermore, whenever a sample loss

happens, SEER recovers faster than TDECR. At the same time, no additional computation is required.

In this chapter, the principle of the proposed algorithm SEER, and its theoretical analysis are stated and also the simulation results are shown including the comparison with TDECR. When the new delay jitter detection method and the center realignment process are combined, a delay jitter or a sample loss may have virtually no effect on the echo canceller's performance.

Chapter 6. CONCLUSION

This thesis has presented a new statistical method to determine the occurrence of delay jitter and an adaptation method to compensate for the effect of delay jitter and/or sample loss on echo cancellation. The new method is named SEER, which stands for Squared Error Estimation and Realignment.

6.1 Conclusion

When SEER is compared with other existing algorithms, such as TDECR, which shows better performance than the rest, the following conclusions can be drawn.

Firstly, when using this new detecting method, the occurrence of delay jitter can be captured with a low probability of incorrect acceptance and incorrect rejection. Consequently, if a delay jitter were present, it would, in all likelihood, be detected quickly. On the other hand, most of the time, it won't recognize a normal variation of echo cancellation procedure as a delay jitter.

Secondly, when a sample loss or delay jitter occurs and is detected, the new center realignment procedure offers the echo canceller a precious opportunity to operate with the desired ERLE level, instead of waiting for the center recalculation and subsequent sample accumulation.

Thirdly, the SEER algorithm not only has a faster convergence speed and a better tracking ability, but also in some cases, for example, sample loss scenario two, it can achieve a higher ERLE level after convergence than other algorithms, which is shown in Figure 5-7. It

suggests that, in the situation with the presence of delay jitters and clock slippage in the network, using this method for echo cancellation is more effective.

Fourthly, this new method offers many benefits with only a slight increase in computational expenses.

It should be noted that the local variance estimation procedure only introduces one multiplication and two additions in each iteration. Thus, the complexity brought by the calculation of local variance can be neglected when compared to the adaptation of the echo canceller's coefficients.

6.2 *Future Work*

Note that the center recalculation procedure is error prone and the probability of a wrong determination increases along with the increasing frequency of channel change. On the other hand, the center realignment is quite reliable. Therefore, it should be possible to freeze the center recalculation process in SEER under certain conditions.

References:

- [BELL95] John J.C. Bellamy, "Digital Network Synchronization", IEEE Com. Mag. April, 1995.
- [CEFR98] Stefano Bregni Cefriel, "A Historical Perspective on Telecommunications Network Synchronization", IEEE Comm. Mag. June 1998.
- [GUST96] Catharina Carlemalm, Fredrik Gustafsson, and Bo Wahlberg, "On the Problem of Detection and Discrimination of Double Talk and Change in the Echo Path", Proc. of ICASSP'96, pp. 1295-1298.
- [CHEN89] Yung-Fu Cheng and Delores M. Etter., "Analysis of an Adaptive Technique for Modeling Sparse Systems", IEEE, Trans. Acoust., Speech. Signal Processing, Vol. 37, No. 2, Feb. 1989, P254-264.
- [HENN93] M. S. El-Hennawey, K. A. Hamied, M. Y. Zaidan and M. Atiquzzaman, "A DSP-based Microcomputer system for echo measurement in telephone networks", INT. J. ELECTRONICS, 1993, VOL. 75, NO. 3, 407-424.
- [FALC85] David D. Falconer, "Timing Jitter Effects On Digital Subscriber Loop Echo Cancellers", Trans. Com, Vol. COM-33, No. 8, Aug., 1985.
- [FERT86] Antoni Fertner and Anders Sjolund, "comparison of Various Time Delay Estimation Methods by Computer Simulation", IEEE, Trans. Acoust., Speech. Signal Processing, Vol. ASSP-34, No. 5, Oct. 1986, P1329-1330.
- [G165] ITU T Recommendation for Echo cancellers
- [G168] ITU-T Recommendation for Echo cancellers

- [HAHN96] Peter J. Hahn, V. John Mathews and Thao D. Tran, "Adaptive Realization of A Maximum Likelihood Time Delay Estimator", Proc. of ICASSP'96, pp. 3121-3124.
- [HAYK96] Simon Haykin, "Adaptive Filter Theory", Third Edition, Prentice Hall, New Jersey, 1996.
- [HIRA96] Akihiro Hirano, Akihiko Sugiyama and Shigeji Ikeda , "DSP Implementation And Performance Evaluation Of Sparse-tap Adaptive FIR Filters With Tap-position Control", IEEE Proc. of ICASSP '96, pp. 1295-1298.
- [HOME98] John Homer and Robert R. Bitmead, "Quantifying the Effects of Dimension on the Convergence Rate of the LMS Adaptive FIR Estimator", , IEEE, Trans. Acoust., Speech. Signal Processing, Vo. 46, No. 10, Oct. 1998, P2611-2615.
- [IKED94] Shigeji Ikeda and Akihiko Sugiyama , "A Fast Convergence Algorithm For Sparse-tap Adaptive FIR Filters For An Unknown Number Of Multiple Echoes", Proc. of ICASSP'94, pp. 41-44.
- [JACO93] Giovanni Jacovitti and Gaetano Scarano, "Discrete Time Techniques For Time Delay Estimation", IEEE, Trans. Acoust., Speech. Signal Processing, vol. 41, No. 2, Feb. 1993.
- [JEBA96] S. Ben Jebara, and M. Jaidane-Saidane, "Best Input for Optimal Tracking Randomly Time-varying Systems: Justification of Adaptive Predictive Structure", Proc. of ICASSP'96, pp. 1957-1960.
- [KART91] Peter Kartaschoff, "Synchronization in Digital communications Networks". Proc. of the IEEE, Vol. 79, No. 7, July, 1991.

- [KAWA86] Shin-ichi Kawamura and Mitsutoshi Hatori , “A Tap Selection Algorithm for Adaptive Filters”, Proc. of ICASSP'86, pp. 2979-2982
- [LEON94] Alberto Leon-Garcia, “Probability and Random Processes for Electrical Engineering”, Second Edition, Addison-Wesley, New York, 1994.
- [LIAV98] Athanasios P. Liavas and Phillip A. Regalia, “Acoustic Echo Cancellation: Do IIR Models Offer Better Modeling Capabilities than Their FIR Counterparts?”, IEEE, Trans. Acoust., Speech. Signal Processing, Vol. 46, No. 9, Sept. 1998, P2499-2504.
- [MA96] Xinyu Ma and Chrysostomos L. Nikias, “Robust Time Delay Estimation Using Lower-order Statistics”, Proc. of ICASSP'96, pp. 3125-3128.
- [MARG93] V. A. Margo , D. M. Etter, and N. C. Carlson , “Multiple Short-Length Adaptive Filters For Time-Varying Echo Cancellation”, Proc. of ICASSP'93 , pp. 1161-1164, 1993.
- [MARQ97] Paulo A. C. Marques, Fernando M. G. Sousa and Jose M. N. Leitaó, “A DSP Based Long Distance Echo Canceler Using Short Length Centered Adaptive Filters”, Proc. of ICASSP'97 , pp. 1885-1888.
- [MESS86] David G. Messerschmitt, “Asynchronous and Timing Jitter Insensitive Data Echo Cancellation”, Trans. Com. Vol. COM-34, No. 12, Dec., 1986.
- [NIKO97] Nikolic “Line Echo Canceler”, Implementation of block update and NLMS algorithms using TMS320C54X, 1997.
- [RUPP98] Markus Rupp, “A Family of Adaptive Filter Algorithms with Decorrelating Properties”, , IEEE, Trans. Acoust., Speech. Signal Processing, Vol. 46, No. 3, March 1998, P771-775.

- [SOND66] M.M Sondhi and A.J. Presti, "A self adaptive echo canceler", Bell Syst. Tech. J., Dec, 1966, P1851-1854.
- [TIEN99] Po L. Tien and Maria C. Yuang, "Intelligent Voice Smoother for Silence-Suppressed Voice over Internet", IEEE, Journal on Selected Areas in Communication, Vol. 17, No. 1, Jan. 1999, P29-41.
- [YIP87] Philip C. Yip and Delores M. Etter, "An Adaptive Technique For Multiple Echo Cancellation In Telephone Networks", Proc. of ICASSP'87, pp. 2133-2136.

Appendix A – Time Delay Estimator

The conventional TDE technique consists of estimating the cross-correlation between the original signal and its filtered version and identifying the maximum value. This has been designated as DC (Direct Correlation).

Consider the following signals:

$$X(n) = S(n) + v_1(n) \quad (\text{A-1})$$

$$Y(n) = aS(n) + v_2(n) \quad (\text{A-2})$$

where $X(n)$ and $Y(n)$ are two differently scaled and delayed signals, and $v_1(n), v_2(n)$ are random Gaussian noises uncorrelated with each other and with $S(n)$. In another words, $X(n)$ is the input signal of a channel and $Y(n)$ is a filtered and delayed version of $X(n)$.

Assuming that the signals are ergodic, the time average is equal to the assemble average, then the expected value for the signal can be estimated by the time average over an interval, where the interval is big enough.

Applying the above assumption, the estimation of $R_{xy}(\tau)$ can be written as:

$$\hat{R}_{xy}(\tau) = \frac{1}{M - \tau} \sum_{k=1}^{N-\tau} x(k) y(k + \tau) \quad (\text{A-3})$$

where M is the length of signal $X(n)$ and $Y(n)$, which is also the time period over which the average is taken on.

If we assume $S(n)$ is white noise, and $\sigma_s^2, \sigma_{v_1}^2$ are the variances of $S(n)$ and $v_1(n)$ respectively, then the auto-correlation of $X(n)$ is :

$$\begin{aligned} R_{xx}(\tau) &= \sigma_s^2 + \sigma_{v_1}^2 & \tau &= 0 \\ &= 0 & \tau &\neq 0 \end{aligned} \quad (\text{A-4})$$

Therefore, the estimation can be expressed as:

$$\hat{R}_{xy}(\tau) = \sum_{j=0}^{N-1} c_j \hat{R}_{xx}(\tau - j) = c_\tau (\sigma_s^2 + \sigma_{v_1}^2) \quad (\text{A-5})$$

where $c_j, j = 1, 2, \dots, N-1$ is the impulse response of the echo channel.

Because one sample function of $\hat{R}_{xx}(\tau)$ may not give an accurate estimation, a long-term average of $\hat{R}_{xx}(\tau)$ is used to estimate the expected value of $\hat{R}_{xx}(\tau)$, which can be expressed as:

$$E[\hat{R}_{xy}(\tau)] = \sum_{j=0}^{N-1} c_j E[\hat{R}_{xx}(\tau - j)] \quad (\text{A-6})$$

$$\text{Where } E[\hat{R}_{xx}(\tau)] = \frac{1}{K} \sum_{i=1}^K \hat{R}_{xx}^i(\tau)$$

Since c_j is proportional to $\hat{R}_{xy}(\tau)$, the location of the peaks of $\hat{R}_{xy}(\tau)$ are the locations of c_j , i.e.,

$$\hat{D}_{DC} = \arg \max \hat{R}_{DC}(\tau) = \arg \max \hat{R}_{xy}(\tau) \quad (\text{A-7})$$

where $\hat{D}_{DC}$ denotes the estimated delay using DC.

From an implementation point of view, DC is computationally expensive, i.e., calculation in real-time is difficult. Other TDE methods such as the ASDF (Average Square Difference Function) estimator, the AMDF (Average Magnitude Difference Function) estimator, and relay correlation or HS (Hybrid Sign) correlation estimator tend to replace this conventional DC estimator.

Herein, three kinds of TDE method will be introduced:

- 1) DC (Direct Cross-correlation)

This method has been described above. By calculating the cross-correlation of the input signal $x(n)$ and the output signal $y(n)$ directly, it employs multiplication.

2) AMDF (Average Magnitude Difference Function):

$$\hat{R}_{AMDF}(\tau) = \frac{1}{M - \tau} \sum_{k=1}^{M-\tau} |x(k) - y(k + \tau)| \quad (\text{A-8})$$

$$\hat{D}_{AMDF} = \arg \min \hat{R}_{AMDF}(\tau) \quad (\text{A-9})$$

Obviously, AMDF is multiplication free, which means it is a relatively less expensive and faster estimator. There is a complex demonstration of how this works, but it is beyond the scope of this thesis.

$$3) \text{ ASDF: } \hat{R}_{ASDF}(\tau) = \frac{1}{M - \tau} \sum [x(k) - y(k + \tau)]^2 \quad (\text{A-10})$$

$$\hat{D}_{ASDF} = \arg \min \hat{R}_{ASDF}(\tau) \quad (\text{A-11})$$

In this method, only squaring is involved, which is well-known to be more easily implemented than the multiplication that's involved in DC.

Appendix B – The User Interface

1) The requirements to run this interface:

Windows 95/NT

Matlab 5.0 and up.

Echo channel Model files which contain a 1byM vector, Model_c, channel impulse response, where M is the max delay of the channel.

2) Uicontrols explanation

List Box:

Model file list: all the files with extension .mat in the selected directory.

Edit field:

Model file directory: the directory containing all the Model files.

Number of iterations: the number of iterations in the simulation.

Number of taps: number of taps chosen for the filter.

pdf: the pdf of delay jitter's occurrence.

Pop-up menu:

Plot color: the color used to plot the simulation results.

Input filter: the filters which could be applied to the input signal(LPF, HPF, BPF or none).

Input signal: the type of input signal (white Gaussian or speech), the speech option is not available at this point.

Display: the way to show the simulation results(ERLE, squared error or both).

CheckBox:

fromFile: if it is checked, the input and echo signal will be obtained from a file “signal.mat”. Otherwise, signals are generated and saved into “signal.mat”.

Delay jitter: if it is checked, delay jitters will occur during the simulation.

Noise: white noise will be added to the echo signal according to the SNR if checked.

Grid: figures will be ‘grid on’ if checked.

Hold: figures will be ‘hold on’ if checked.

Radio-button:

Sample loss: sample loss will happen if checked at the rate specified under “Loss rate”.

Text Field:

State indicator: idle or busy or error (at the right lower corner of the window).

Status bar: shows the status of the simulation (at the bottom of the window).

Slider:

number of run: the number of individual runs to estimate assemble average.

number of losses: the number of samples lost at each time.

Loss rate: the frequency of sample losses.

S/N ratio: SNR (dB).

number of small filters: the number of small filters used in TDE.

length of small filters: the constraint used in TDE.

step size: the step size used in the NLMS update.

Button:

Load Model: load the selected Model file from the file list.

Plot Model: plot the channel model in the time domain.

Model frequency: plot the channel model in the frequency domain.

Plot FIR: plot the simulated FIR filter and the channel model in time domain.

FIR frequency: plot the simulated FIR filter and the channel model in frequency domain.

TDE-NLMS: TDE algorithm.

STWQ: STWQ algorithm.

Run NLMS: Full-tap NLMS algorithm.

SEER: the proposed algorithm.

Exit: close all the windows associated with the control window including itself.

3) To start

Copy all the .m and .mat files into one directory.

In Matlab command window, type simuwin then follow the instruction at the bottom of the control window.

4) Troubleshooting tips

- a) Always load a Model file before beginning any simulation.
- b) The total taps number chosen for the echo canceller must be bigger than M.
- c) Use Load Model button to reset the 'state indicator'.

Appendix C – Error Analysis

The subscript n is used to denote $t=nT$, and k to index the samples in a sequence of either an input or an impulse response. More precisely, at any time instant $t=nT$, $x_n(k)$ denotes the input sequence, $h_n(k)$ denotes the channel's impulse response, and $g_n(k)$ denotes the echo canceller's coefficients. Let y_n denote the output from the channel, z_n denote the output from the echo canceller, e_n denote the error, and η_n denote the AWGN (Additive White Gaussian Noise) sequence in the channel. Since the impulse response of the channel has a duration MT , k can only take on values from $\{0,1,2,\dots,M-1\}$.

Then we have:

$$y_n = \sum_k x_n(M-1-k)h_n(k) + \eta_n \triangleq \mathbf{X}^H \mathbf{H} + \eta_n \quad (\text{C-1})$$

$$z_n = \sum_k x_n(M-1-k)g_n(k) \triangleq \mathbf{X}^H \mathbf{G} \quad (\text{C-2})$$

$$e_n = y_n - z_n \quad (\text{C-3})$$

$x_n(k)$ forms a column vector $\mathbf{X}$, and $g_n(k)$ and $h_n(k)$ form column vectors $\mathbf{G}$ and $\mathbf{H}$ respectively. H denotes transpose conjugation. When assuming the input $x_n(k)$ is a zero-mean stationary white Gaussian sequence, it then follows that its auto-correlation is

$$R_{x_n}(k) = E[x_n(m+k)x_n(m)] = \begin{cases} \sigma_x^2, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (\text{C-4})$$

From (C-1), (C-2), and (C-3), we have

$$e_n = \sum_k [h_n(k) - g_n(k)]x_n(M-1-k) + \eta_n \quad (\text{C-5})$$

For all $k \in \{0,1,\dots,M-1\}$, n indexes the number of iterations, let $d_n(k) = h_n(k) - g_n(k)$.

Then e_n can be expressed alternatively as:

$$e_n = \sum_k x_n(M-1-k)d_n(k) + \eta_n \quad (\text{C-6})$$

Due to the assumption that $x_n(k)$ and η_n are zero-mean, it follows immediately that:

$$E[e_n] = 0 \quad (\text{C-7})$$

Since e_n is not a stationary (not even a wide-sense stationary) random process with respect to time n , its auto-correlation function is complicated by two variables. However, we are only interested in the variance of e_n at any time instant $t=nT$. It can be obtained as follows:

$$\begin{aligned} E[e_n^2] &= E\left[\left(\sum_k x_n(M-1-k)d_n(k) + \eta_n\right)^2\right] \\ &= E\left[\left(\sum_k x_n(M-1-k)d_n(k)\right)^2\right] + 2E\left[\eta_n \sum_k x_n(M-1-k)d_n(k)\right] + E[\eta_n^2] \end{aligned}$$

Since η_n is assumed to be zero-mean and with variance $\sigma_{\eta_n}^2$, or simply σ_{η}^2 (since it is time-invariant), we have

$$E[e_n^2] = \sigma_{\eta}^2 + E\left[\sum_k x_n^2(M-1-k)d_n^2(k)\right] + 2E\left[\eta_n \sum_i \sum_{j \neq i} x_n(M-1-i)x_n(M-1-j)d_n(i)d_n(j)\right]$$

Because $x_n(k)$ is assumed to be a white sequence, the last term in the expression above is zero. Then we have:

$$E[e_n^2] = \sigma_{\eta}^2 + E\left[\sum_k x_n^2(M-1-k)d_n^2(k)\right]$$

Since $d_n(k)$ only depends on $x_{n-1}(k)$, and $x_{n-1}(k)$ and $x_n(k)$ are not correlated, we have:

$$\begin{aligned} E[e_n^2] &= \sigma_{\eta}^2 + E\left[\sum_k x_n^2(M-1-k)d_n^2(k)\right] \\ &= \sigma_{\eta}^2 + \sigma_x^2 \sum_k E[d_n^2(k)] = \sigma_{\eta}^2 + M\sigma_x^2 E[d_n^2(k)] \end{aligned} \quad (\text{C-8})$$

where σ_x^2 is the variance of $x_n(k)$.

From equation (C-6), it can be observed that e_n is a linear combination of $x_n(k)$ and η_n . Under the assumption that $x_n(k)$ and η_n are Gaussian, the claim can be made that e_n is also a Gaussian random variable for any n , and from equation (C-7) and equation (C-8), we have obtained its mean and variance.

If, at this point, we consider e_n^2 as random variable J_n or $J_n = e_n^2$, then from equation (C-8), we have determined

$$E[J_n] = \sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)]$$

Denote this quantity $\sigma_{e_n}^2$, i.e.,

$$E[J_n] = \sigma_{e_n}^2 = \sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)] \quad (C-9)$$

Since at any time instant, e_n is a zero-mean Gaussian random variable with variance $\sigma_{e_n}^2$, it follows $J_n = e_n^2$ is a Chi-square random variable with degree of freedom 1. Its pdf (Probability Density Function) is:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma_{e_n}^2 x}} e^{-\frac{x}{2\sigma_{e_n}^2}}, x > 0 \quad (C-10)$$

The square-error random variable J_n has a mean $\sigma_{e_n}^2$, as shown in equation (C-9), and we know for such a random variable its variance [LEON94] is:

$$\text{VAR}[J_n] = 2\sigma_{e_n}^4 \quad (C-11)$$

The focus is now on the Chi-square random variable J_n . At any time instant $t=nT$ during the adaptive process, the algorithm performs a statistical test on the received value of this random variable. This will be illustrated later.

Notice from equation (C-9) that

$$E[J_n] = \sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)]$$

To further study the behavior of J_n , we have to take a closer look at the term $d_n(k)$, the difference between the impulse response of the channel and that of the canceller.

In general, $d_n(k)$ can be expressed in a recursive form as a function of $d_{n-1}(k)$. However, depending on the updating algorithm for the filter coefficients such as LMS or LS, it will be somewhat different, although indeed, in general, it behaves similarly among different algorithms. Here we will use the LMS criterion as an example to study its characteristics.

Using the notations for LMS algorithm in Haykin's book, 1996 [HAYK96]:

$$d_n(k) = d_{n-1}(k) \left[1 - \mu \sum_k x_{n-1}^2(k) \right] + \mu x_{n-1}(k) \eta_n \quad (\text{C-12})$$

where μ is the step size of the LMS algorithm. When the input sequence is white, we can obtain the expected value of $d_n(k)$ in the recursive form:

$$E[d_n(k)] = E[d_{n-1}(k)](1 - \mu M \sigma_x^2) \quad (\text{C-13})$$

When μ is selected such that $|1 - \mu M \sigma_x^2| < 1$,

$$E[d_n(k)] \rightarrow 0, n \rightarrow \infty$$

This is also known as the phenomenon of unbiased convergence. The variance of $d_n(k)$ has the recursive form:

$$E[d_n^2(k)] = E[d_{n-1}^2(k)](1 - \mu M \sigma_x^2)^2 + \mu^2 \sigma_x^2 \sigma_\eta^2 \quad (\text{C-14})$$

It follows $E[d_n^2(k)] \rightarrow \mu^2 \sigma_x^2 \sigma_\eta^2, n \rightarrow \infty$

Now, let us assume there is no channel variation at time $t=nT$. Then consider:

$$\begin{aligned}\frac{E[J_n]}{E[J_{n-1}]} &= \frac{\sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)]}{\sigma_\eta^2 + M\sigma_x^2 E[d_{n-1}^2(k)]} \\ &= \frac{\sigma_\eta^2 + M\sigma_x^2 \left(E[d_{n-1}^2(k)(1 - \mu M\sigma_x^2)^2 + \mu^2 \sigma_x^2 \sigma_\eta^2] \right)}{\sigma_\eta^2 + M\sigma_x^2 E[d_{n-1}^2(k)]}\end{aligned}$$

Let $Q = \mu M\sigma_x^2$. Since $|1 - \mu M\sigma_x^2| < 1$, it follows that $0 < Q < 2$. Knowing the signal to noise ratio $SNR = \frac{\sigma_x^2}{\sigma_\eta^2}$, the above expression can be simplified as

$$\begin{aligned}\frac{E[J_n]}{E[J_{n-1}]} &= 1 - \frac{(SNR)ME[d_{n-1}^2(k)](2Q - Q^2) - \frac{Q^2}{M}}{1 + (SNR)ME[d_{n-1}^2(k)]} \\ &\approx 1 - \frac{(2Q - Q^2)(SNR)ME[d_{n-1}^2(k)]}{1 + (SNR)ME[d_{n-1}^2(k)]}\end{aligned}\tag{C-15}$$

Equation (C-15) is a fairly informative expression.

First, at time n when the canceller has not converged, $(SNR)ME[d_{n-1}^2(k)] \gg 1$, we have:

$$\frac{E[J_n]}{E[J_{n-1}]} = 1 - 2Q + Q^2 = (1 - Q)^2$$

Or alternatively,

$$E[J_n] = (1 - Q)^2 E[J_{n-1}]\tag{C-16}$$

Since J_n is a Chi-square random variable, its variance is known as:

$$VAR[J_n] = 2(1 - Q)^4 E^2[J_{n-1}]\tag{C-17}$$

Secondly, when the canceller approaches convergence, from equation (C-14):

$$E[d_n^2(k)] = \mu^2 \sigma_x^2 \sigma_\eta^2 = \frac{Q^2}{M^2(SNR)}\tag{C-18}$$

Substituting equation (C-18) into equation (C-15), we have

$$\frac{E[J_n]}{E[J_{n-1}]} = 1 - \frac{(2Q - Q^2) \frac{Q^2}{M}}{1 + \frac{Q^2}{M}} \approx 1$$

or

$$E[J_n] = E[J_{n-1}] \quad (\text{C-19})$$

and similarly, its variance is

$$\text{VAR}[J_n] = 2E^2[J_{n-1}] \quad (\text{C-20})$$

Thus, the statistics of the random variable J_n have been obtained in terms of the statistics of the random variable J_{n-1} , given that no channel variation has occurred at time $t=nT$.

To set up the acceptance criteria for the statistical tests in the algorithm, it is also necessary, as illustrated in the later section, to explore the statistics of J_n when a channel variation does occur at $t=nT$ time instant. In this situation, we denote this random variable $\tilde{J}_n$.

Assume at $t=nT$, the channel's impulse response shifts by δT , where δ is some small integer in $\{\pm 1, \pm 2, \pm 3, \dots, \pm \omega\}$. This can model both delay jitters and sample losses.

Following similar procedures, we can easily obtain:

$$E[\tilde{J}_n(\delta)] = \sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)] \quad (\text{C-21})$$

$$\text{Where } \tilde{d}_n(k) = h_n(k + \delta) - g_n(k)$$

And

$$\begin{aligned} \tilde{d}_n(k) &= h_n(k + \delta) - h_n(k) + h_n(k) - g_n(k) \\ &= h_n(k + \delta) - h_n(k) + d_n(k) \end{aligned} \quad (\text{C-22})$$

Then:

$$\begin{aligned} E\left[\tilde{d}_n^2(k)\right] &= E\left[\left(h_n(k+\delta) - h_n(k) + d_n(k)\right)^2\right] \\ &= E\left[\left(h_n(k+\delta) - h_n(k)\right)^2\right] + 2E\left[\left(h_n(k+\delta) - h_n(k)\right)d_n(k)\right] + E\left[d_n^2(k)\right] \end{aligned}$$

With an assumption $d_n(k)$ and $h_n(k)$ are independent, we have:

$$\begin{aligned} E\left[\tilde{d}_n^2(k)\right] &= E\left[\left(h_n(k+\delta) - h_n(k)\right)^2\right] + 2E\left[\left(h_n(k+\delta) - h_n(k)\right)\right]E\left[d_n(k)\right] + E\left[d_n^2(k)\right] \\ &= 2R_{h_n}(0) - 2R_{h_n}(\delta) + E\left[d_n^2(k)\right] \end{aligned} \tag{C-23}$$

where $R_{h_n}(k)$ is the auto-correlation function of $h_n(k)$.

Substituting equation (C-23) into equation (C-21), and combining it with equation (C-9),

we obtain:

$$E\left[\tilde{J}_n(\delta)\right] = E\left[J_n\right] + 2M\sigma_x^2\left(R_{h_n}(0) - R_{h_n}(\delta)\right) \tag{C-24}$$

When the input is white, equation (C-24) can be expressed alternatively as

$$E\left[\tilde{J}_n(\delta)\right] = E\left[J_n\right] + 2M\left(R_y(0) - R_y(\delta) - \sigma_\eta^2\right) \tag{C-25}$$

Assume δ takes on values in $\{\pm 1, \pm 2, \pm 3, \dots, \pm \omega\}$ with equal probability $1/2\omega$, then:

$$\begin{aligned} E\left[\tilde{J}_n\right] &= \sum E\left[\tilde{J}_n(\delta)\right] \frac{1}{2\omega} \\ &= E\left[J_n\right] + 2M\left(R_y(0) - \sigma_\eta^2\right) - \frac{M}{\omega}R_y(\delta) \end{aligned} \tag{C-26}$$

This is the expected value of the square-error random variable $\tilde{J}_n$, when a channel variation has occurred at time $t=nT$. Its variance follows immediately:

$$\text{VAR}[\tilde{J}_n] = 2E^2[\tilde{J}_n] \quad (\text{C-27})$$

To compare $E[\tilde{J}_n]$ and $E[J_n]$, consider:

$$\begin{aligned} \frac{E[\tilde{J}_n]}{E[J_n]} &= \frac{E[J_n] + 2\frac{M}{\omega}\sigma_x^2 \sum_{\delta} (R_h(0) - R_h(\delta))}{E[J_n]} \\ &= 1 + \frac{2\frac{M}{\omega}\sigma_x^2 \sum_{\delta} (R_h(0) - R_h(\delta))}{\sigma_\eta^2 + M\sigma_x^2 E[d_n^2(k)]} \end{aligned} \quad (\text{C-28})$$

From equation (C-14)

$$E[d_n^2(k)] = E[d_{n-1}^2(k)](1-Q)^2 + \mu^2\sigma_x^2\sigma_\eta^2$$

and $E[d_0^2(k)] = E[h_0^2(k)] = R_h(0)$, if there hasn't been any channel variation before $t=nT$.

We can still safely obtain:

$$E[d_n^2(k)] = R_h(0)(1-Q)^{2m} + \mu^2\sigma_x^2\sigma_\eta^2 \quad (\text{C-29})$$

even if there has been some channel variation. In equation (C-29), m is some integer smaller than n and increasing with n .

Substituting equation (C-29) into equation (C-28), we have:

$$\begin{aligned} \frac{E[\tilde{J}_n]}{E[J_n]} &= 1 + \frac{2\frac{M}{\omega}\sigma_x^2 \sum_{\delta} (R_h(0) - R_h(\delta))}{\sigma_\eta^2(1 + M\mu^2\sigma_x^4) + M\sigma_x^2 R_h(0)(1-Q)^{2m}} \\ &= 1 + \frac{2M(\text{SNR}) \sum_{\delta} (R_h(0) - R_h(\delta))}{\omega \left(1 + \frac{Q^2}{M}\right) + \omega M(\text{SNR}) R_h(0)(1-Q)^{2m}} \end{aligned} \quad (\text{C-30 a})$$

If the channel variations that occur before $t=nT$ have been detected and handled properly, the last term of the denominator approaches zero exponentially, which allows us to simplify the expression as:

$$\begin{aligned} \frac{E[\tilde{J}_n]}{E[J_n]} &\approx 1 + \frac{2M(SNR) \sum_{\delta} (R_h(0) - R_h(\delta))}{\omega \left(1 + \frac{Q^2}{M}\right)} \\ &\approx 1 + \frac{2M}{\omega} (SNR) \sum_{\delta} (R_h(0) - R_h(\delta)) \end{aligned} \quad (\text{C-30 b})$$

Knowing $M \gg \omega$, given a practically acceptable SNR, it usually follows:

$$\frac{E[\tilde{J}_n]}{E[J_n]} \gg 1 \quad (\text{C-31})$$