

# NLMS 演算法收斂步伐調整之回音消除

學生：陳俊銘

指導教授：謝世福

國立交通大學電信工程學系碩士班



基於簡單性和實用性，回音消除最常用的演算法就是 NLMS 演算法。然而，NLMS 演算法最大的缺點就是收斂速度太慢，因為其使用固定的收斂步伐 (step-size)。我們提出一種調整收斂步伐的新方法稱為理想步伐 NLMS 演算法。所謂的『理想』是指這種收斂步伐在每次疊代時可以提供最小閾係數誤差的均方誤差 (MSE)，以致於和其他演算法比較起來有最快的收斂速度。每一個閾 (tap) 有自己的、隨閾係數誤差改變而調整的收斂步伐。在偵測雙邊對話方面，我們分析了 J. C. Liu 所提出來的的方法。它是利用計數器的概念在殘響回音突然變大時，用一段短時間內去記錄事件發生的次數。回音衰減量 (ERLE) 被用來表示使用不同門限 (threshold) 的成本。最後，電腦模擬將用來驗證支持之前的分析。

# **Step-Size Adjustments of NLMS Algorithm for Acoustic Echo Cancellation**

**Student : J. M. Chen**

**Advisor : S. F. Hsieh**

**Department of Communication Engineering**

**National Chiao Tung University**



The most often used algorithm for acoustic echo cancellation is normalized least mean square (NLMS) algorithm due to its simplicity and robustness. However, the major drawback of NLMS algorithm is its slow convergence rate due to its constant step-size. We propose a new approach of adjusting step-size called optimum step-size NLMS. The word “optimum” means the step-size can provide the minimum tap coefficient error mean square error at each iteration step which leads the fastest convergence rate than other algorithms. Each tap has its individual time-variant step-size which adjusts with tap coefficient error variance. We analyze double talk detector proposed by J. C. Liu. It uses the concept of counter to record number of abrupt events happened during a short period when an abrupt change in residual echo. Echo return loss enhancement (ERLE) is used to indicate the cost of using different threshold. Finally, computer simulations will be presented to support the analysis.

# Acknowledgement

I would like to express my deepest gratitude to my advisor, Dr. S. F. Hsieh, for his enthusiastic guidance and great patience, especially the autonomy in research. I also appreciate all my lab-mates J. S. Chen, T. Y. Wu and B. X. Wu for their inspiration and help. Finally, I would like to show my thanks to my parents for their unceasing encouragement and love. I also want to thank that I have the chance to accomplish this thesis.



# Contents

Chinese Abstract	i
English Abstract	ii
Acknowledgement	iii
Contents	iv
List of Tables	vi
List of Figures	vii
1 Introduction	1
2 Adaptive Algorithms for Acoustic Echo Canceller	5
2.1 Configuration of an acoustic echo canceller	6
2.2 Least mean square algorithm	7
2.3 Step-size adjustment	8
2.3.1 Variable step-size NLMS algorithm	9
2.3.2 Exponentially weighted step-size NLMS algorithm	9
2.3.3 Proportionate NLMS algorithm	10
3 Optimum Step-size for NLMS Algorithm	13
3.1 Optimum step-size sequence of Slock	14
3.2 Notations	16
3.3 Derivation of optimum time- & tap-variant step-size LMS algorithm	17
3.4 Practical ONLMS algorithm	20
3.5 Simplified ONLMS algorithm	22
3.6 Comparison of convergence	24
3.7 Computational complexity	26

4	Double Talk Detection	28
4.1	Double talk in echo canceller .....	29
4.2	A scheme to distinguish EPC and DT .....	31
4.3	Analysis of correct decision probability .....	33
4.3.1	Assume $ e_a(n) $ as Gaussian random variable .....	34
4.3.2	Decision sample .....	36
5	Computer Simulations	40
5.1	Simulation parameters and room impulse response .....	41
5.2	Performance comparison of convergence rate for different algorithms.....	44
5.3	ONLMS algorithm .....	47
5.4	Tap coefficient error variance .....	51
5.5	Nonlinear distortion .....	53
5.6	The correct probability of system decision using different thresholds.....	55
5.7	Average ERLE using different threshold .....	57
5.8	AEC with DTD .....	58
6	Conclusions	62
	Appendix	64
	Bibliography	66

# List of Tables

3.1	Computational requirements for different algorithms.....	27
5.1	Performance of convergence rate for different algorithms under SNR = 30 dB.	45
5.2	Performance of convergence rate for different algorithms under SNR = 10 dB.	46
5.3	Performance for different algorithms under SNR = 30 dB.....	47
5.4	Performance for different algorithms under SNR = 10 dB.....	47
5.5	Performance of using different step-size adjustment under SNR = 30 dB.....	48
5.6	Performance of using different step-size adjustment under SNR = 10 dB.....	49



# List of Figures

1.1	Diagram of hands-free communication system.....	1
1.2	Hands-free communication system with AEC.....	2
2.1	Configuration of an acoustic echo canceller.....	6
3.1	RIR decay envelop.....	21
3.2	Piecewise time-variant and tap-variant step-size for different state.....	23
4.1	Block diagram of the AEC with DTD.....	29
4.2	Liu's scheme to distinguish DT or EPC.....	31
4.3	Exhibition of the schemes reaction when detected DT.....	34
4.4	Each sample $ e_a(n) $ is assumed to be a Gaussian random variable within the period $n_d$ samples.....	35
4.5	Tap coefficient misalignments with different step-sizes under DT situation....	35
4.6	Exponential distribution of random variable $K$ .....	37
5.1.1	Room impulse response.....	42
5.1.2	Room impulse response in the same room.....	43
5.1.3	Speech signal with sampling rate 8 kHz.....	43
5.2.1	Convergence curve of different algorithms under $SNR = 30dB$ .....	45
5.2.2	Convergence curve of different algorithms under $SNR = 10dB$ .....	46
5.3.1	Comparison of Slock's algorithm and ONLMS under high SNR.....	48
5.3.2	Comparison of Slock's algorithm and ONLMS under low SNR.....	49
5.3.3	Convergence curves using piecewise time-variant and tap-variant step-size.....	50
5.4.1	$G_{large}(n)$ and $G_{small}(n)$ for ESNLMS algorithm.....	52
5.4.2	$G_{large}(n)$ and $G_{small}(n)$ for PNLMS algorithm.....	52
5.4.3	$g_i(n)$ for ONLMS algorithm where $i = 10, 70, 170, 270$ .....	53
5.5.1	Sigmoid function.....	54
5.5.2	Different $\alpha$ effect on ERLE under different SNR.....	55
5.6.1	Probability density function of $k$ with $N_r = 50$ and $200$ under DT.....	56

5.6.2 The correct probability of system decision using different  $N_T$  .....57

5.7.1 The average ERLE using different  $N_T$  .....58

5.8.1 Misalignment under DT and EPC with  $N_T = 50$  .....59

5.8.2 Misalignment under DT and EPC with  $N_T = 150$  .....60

5.8.3 Misalignment under DT and EPC with  $N_T = 250$  .....60

5.8.4 Misalignment under DT and EPC using speech with  $N_T = 150$  .....61





# Chapter 1

## Introduction

In these years, hands-free telephone and teleconference systems are widely used [8], [18]. The main problem of those systems is acoustic echo which make listener annoyed. The problem comes about as a result of the reflections of the signal from the loudspeaker back to the microphone. We will introduce the fundamental problem and techniques of acoustic echo cancellation as below.

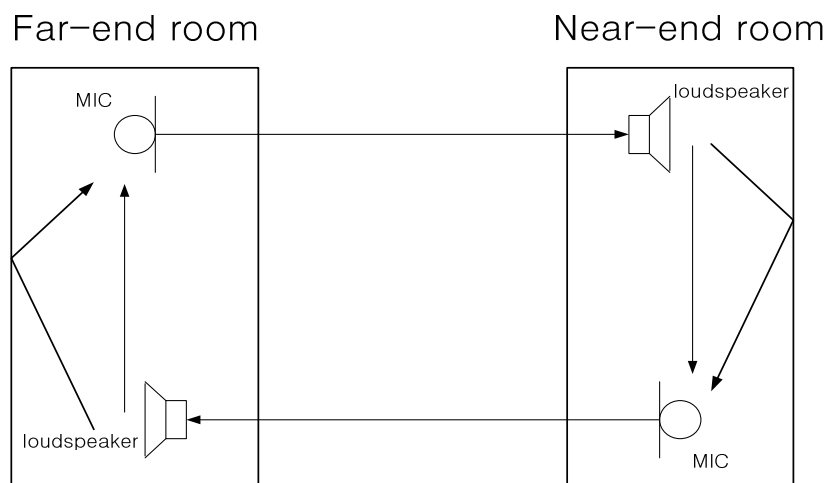


Figure 1.1 Diagram of hands-free communication system

Figure 1.1 is a simplified diagram of a hands-free communication system, which provides an idea of how the acoustic echo causes inconvenience to the listeners in the rooms. Assume that a speaker in the far-end room is in communication using microphone with another speaker in the near-end room, the far-end speech will be transmitted back to the far-end room as a result of the coupling of the loudspeaker and microphone. That is, the far-end speakers will be annoyed by listening to his/her own speech delayed by the round-trip time of system.

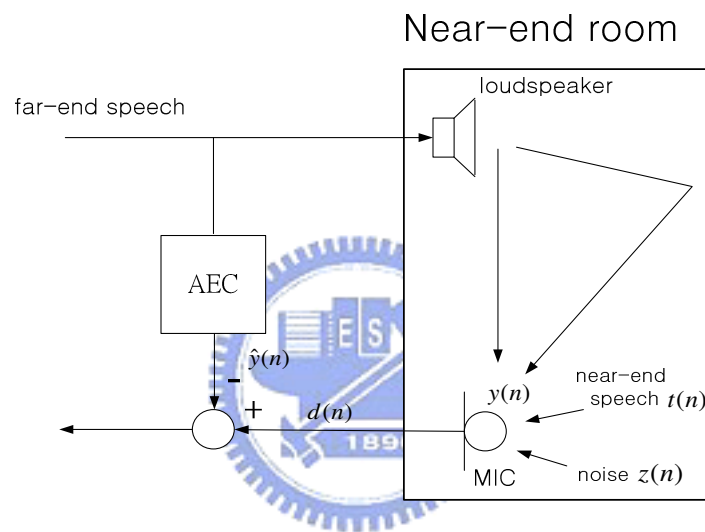


Figure 1.2 Hands-free communication system with AEC

A hands-free communication system with acoustic echo canceller (AEC) is shown in Figure 1.2 to overcome acoustic echo and provide satisfactory speech quality. The AEC estimates the impulse response of the room and generates an estimate of the echo. The estimated echo is then subtracted from the near-end microphone signal which includes the real echo.

The most often used algorithm for AEC is least mean square (LMS) due to its simplicity, leading to its implementation in many applications [8]. However, the major drawback of LMS algorithm is its slow convergence rate. Due to a constant step-size, the inherent limitation of the LMS forces a compromise between the opposing

fundamental requirements of fast convergence rate and small misalignment. Many researches ([2], [4], [6]) have focused on using variable step-sizes to improve the convergence rate. Three of these step-size adjustments will be introduced in Chapter 2 and a new approach (optimum time- & tap-variant step-size NLMS) which has the fastest convergence rate will be proposed in Chapter 3. The word “optimum” means the step-size can provide the minimum tap coefficient error mean square error (MSE) at each iteration step. We use individual time-variant step-size for each tap which results in optimum step-size NLMS (ONLMS) algorithm. Since it needs prior knowledge of room impulse response (RIR), we use an exponential decay room impulse response model for practical implementation in Section 3.4. We also provide a simplified piecewise ONLMS algorithm to reduce its computation load in Section 3.5. We find that tap coefficient error variance is an important issue of step-size adjustment. When the tap coefficient error variance is large, we use a larger step-size and vice versa, so the step-size can vary with tap coefficient error variance. We will discuss this in Section 3.6.

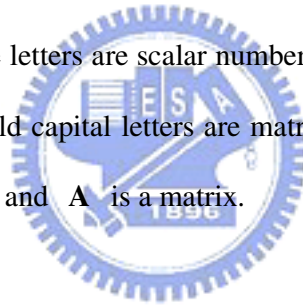
Double-talk (DT) is another serious problem in the acoustic echo cancellation. This situation occurs when simultaneous talks occurs for both near-end and far-end speakers. It makes the AEC filter fails to estimate room impulse response and filter coefficient diverges. In Chapter 4, we use the simple DT detection scheme proposed in [14]. It uses a counter to record the number of abrupt events happened during a short period when an abrupt change is detected in error signal. After a short time delay, if the value of counter is larger than threshold, then the system decides the abrupt change is caused by DT. Otherwise, it is by echo path change.

In Chapter 5, computer simulations follow to verify the results of our analysis in Chapter 3 and Chapter 4.

The main efforts in this thesis are:

- (1) Proposed ONLMS algorithm which can provide the fastest convergence rate compare to other step-size adjustment algorithms e.g. variable step-size NLMS (VSNLMS) [2], exponentially weighted step-size NLMS (ESNLMS) [6], proportionate NLMS (PNLMS) [4].
- (2) Provide simplified piecewise ONLMS algorithm to reduce the complexity.
- (3) Discuss why ESNLMS and PNLMS have two different slopes of convergence curve.
- (4) Analysis of Liu's algorithm based on the system correct decision probability.
- (5) Give a practical choice of threshold according to average ERLE.

In this thesis, lower case letters are scalar numbers, lower case letters with under line are vector forms, and bold capital letters are matrix forms. For example,  $a$  is a scalar number,  $\underline{a}$  is a vector and  $\mathbf{A}$  is a matrix.



# Chapter2

## Adaptive Algorithms for Acoustic Echo Cancellation



We will introduce different adaptive algorithms for acoustic echo canceller (AEC) which belong to the category of stochastic gradient algorithms in this Chapter. The difference of these algorithms is the adjustment of step-size. Because of the inherent limitation of the least mean square (LMS) algorithm (tradeoff between fast convergence rate and small misalignment), researchers have been constantly looking for alternative approach to improve its performance. Many approaches use varying step-size including time-varying [2], tap-varying [6] or both time- & tap- varying [4] step-size, which will be introduced in Section 2.3.

Throughout this thesis, the word “time-variant” represents all taps use identical step-size which is time-variant. The word “tap-variant” means each tap has individual

time-invariant step-size, and the word “time- & tap-variant” means each tap has its individual time-variant step-size.

## 2.1 Configuration of an acoustic echo canceller

Figure 2.1 shows the configuration of an acoustic echo canceller. The echo canceller identifies the impulse response  $\underline{w}(n)$  between the loudspeaker and the microphone. An adaptive filter  $\hat{w}(n)$  is used to identify  $\underline{w}(n)$  since the impulse response  $\underline{w}(n)$  can vary with time. For easy implementation and stability purpose,  $\hat{w}(n)$  is usually implemented using an FIR filter.

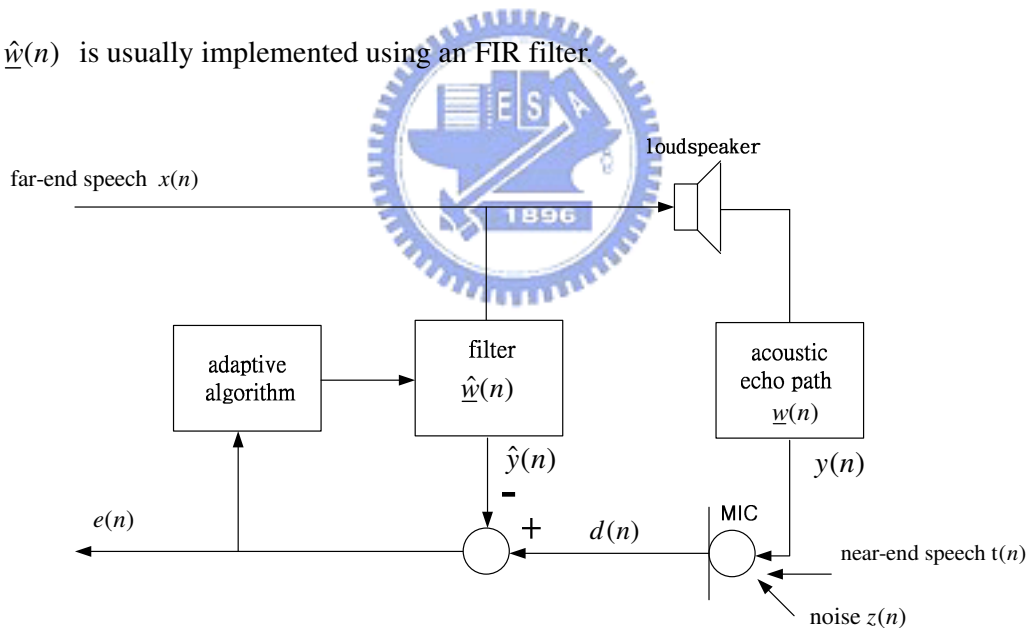


Figure 2.1 Configuration of an acoustic echo canceller

If  $\hat{w}(n)$  is equal to  $\underline{w}(n)$ , then the echo-estimate  $\hat{y}(n)$  will be equal to the echo  $y(n)$  and the echo can be cancelled perfectly under noise free and no double talk situation. Adaptive filtering is required to obtain a good replica of the echo, since the

echo path is usually unknown and the time varying. The adaptive algorithm should provide:

1. Real-time operation.
2. Fast convergence speed.
3. High echo return loss enhancement (ERLE),

where ERLE is defined as the ratio of the real echo power to the residual echo power:

$$\text{ERLE(dB)} = 10 \log_{10} \frac{E[y^2(n)]}{E[(y(n) - \hat{y}(n))^2]} \quad (2.1)$$

## 2.2 Least mean square algorithm

The least mean square (LMS) algorithm is an iterative algorithm to adjust the estimated impulse response so as to minimize the mean square error. The adaptive identification algorithm for the AEC is described as follows [1]:

$$\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n) + ue(n)\underline{x}(n) \quad (2.2)$$

$$e(n) = d(n) - \hat{\underline{w}}^T(n)\underline{x}(n) \quad (2.3)$$

$u$  : the scalar step-size

$e(n)$  : error signal (return signal)

$d(n)$  : the signal received by the microphone

$\underline{x}(n) = [x(n), x(n-1), \dots, x(n-L+1)]$  : the far-end signal with length  $L$

$L$  : the number of tap weights in the adaptive filter  $\hat{\underline{w}}(n)$

$\underline{w}(n) = [w_0(n), \dots, w_{L-1}(n)]^T$ , true echo path vector (room impulse response)

$\hat{\underline{w}}(n) = [\hat{w}_0(n), \dots, \hat{w}_{L-1}(n)]^T$ , estimate echo path vector

To make the LMS algorithm insensitive to changes of the level of input signal, the step-size is normalized, resulting in the NLMS algorithm described as:

$$\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n) + \frac{u}{\|\underline{x}(n)\|^2} e(n)\underline{x}(n) \quad (2.4)$$

Convergence of the mean squared error (MSE) is guaranteed [1] when  $0 < u < 2$ , and is fastest at  $u = 1$  for white noise.

The NLMS algorithm has been the focus of much study due to its simplicity, leading to its implementation in many applications. Its transient mean square error (MSE) depends on (1) the step-size parameter  $u$  in the NLMS algorithm, (2)  $\lambda_i, i = 0, \dots, L-1$ , the eigenvalues of the input correlation matrix, (3) the initial tap coefficient setting, (4) the number of iterations of the algorithm. Given these expressions, we are wondering: does there exist an optimum value for  $u$  which yields rapid convergence and small MSE? It is clearly a tradeoff. The final excess MSE is directly proportional to the step-size of the NLMS while the convergence time increases as the step-size decreases. A constant step-size parameter controls both the convergence rate of the filter coefficients and the final MSE from the Wiener solution. This disadvantage turns up a thought: adjusting step-size. Three popular approaches of adjusting step-size will be introduced in the following Section.

## 2.3 Step-size adjustment

Instead of using a constant step-size, time-variant, tap-variant and time- & tap-variant step-size are used in [2], [6], [4], respectively. In this Section, we will introduce these three typical approaches of step-size adjustment.



### 2.3.1 Variable step-size NLMS algorithm

One popular approach is to employ a time-varying (time-variant) step-size for coefficient update recursion. This is based on using large step-size when the algorithm is far from the optimal solution, thus speeding up the convergence rate. When the AEC filter coefficient is near the optimum, small step-size is used to achieve lower MSE, thus achieving better overall performance.

In [2], the step-size adjustment of variable step-size NLMS (VSNLMS) algorithm is controlled by the power of the error signal:

$$u'(n+1) = \alpha u(n) + \beta e^2(n) \quad \text{with } 0 < \alpha < 1, \beta > 0 \quad (2.5)$$

and

$$u(n+1) = \begin{cases} u_{\max} & \text{if } u'(n+1) > u_{\max} \\ u_{\min} & \text{if } u'(n+1) < u_{\min} \\ u'(n+1) & \text{otherwise} \end{cases} \quad (2.6)$$

The motivation is that a large error signal i.e.,  $e(n)$  will cause the step-size to increase to provide faster tracking while a small error signal will result in a decrease in the step-size to yield smaller misalignment. The constant  $u_{\max}$  is chosen to ensure that the MSE remains bounded and  $u_{\min}$  is chosen to provide a minimum level of tracking ability. The VSNLMS algorithm has reduced the tradeoff between misalignment and convergence rate of the fixed step-size NLMS algorithm.

### 2.3.2 Exponentially weighted step-size NLMS algorithm

Knowledge of the room impulse response (RIR) is rarely used in conventional algorithms. An adaptive algorithm taking into account the variation of an acoustic echo path is expected to improve convergence.

The exponentially weighted step-size NLMS (ESNLMS) algorithm [6] uses a

different step-size (tap-variant) for each tap of an adaptive filter. These step-sizes are time-invariant and weighted proportional to the expected variation of a RIR. The algorithm is based on the fact that the expected variation of a RIR becomes progressively smaller at the same exponential rate as the impulse response energy decays. As a result, the algorithm adjusts coefficients with large echo path variation in large steps, and coefficients with small echo path variation in small steps. The ESNLMS algorithm is expressed as:

$$\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n) + \mathbf{U} \frac{e(n)}{\|\underline{x}(n)\|^2} \underline{x}(n) \quad (2.7)$$

and a diagonal step-size matrix  $\mathbf{U}$  with diagonal form is introduced to account for the tap-variant step-sizes:

$$\mathbf{U} = \begin{bmatrix} u_0 & & & 0 \\ & u_1 & & \\ & & \ddots & \\ 0 & & & u_{L-1} \end{bmatrix} \quad (2.8)$$


where  $u_i = u_0 \gamma^i$  for  $i = 0, \dots, L-1$  and  $\gamma$  is the room exponential attenuation factor ( $0 < \gamma < 1$ ).

The scalar step-size  $u$  in Eq.(2.5) is replaced by a step-size matrix  $\mathbf{U}$  in Eq.(2.8). Elements  $u_i$  are time-invariant and decrease exponentially from  $u_0$  to  $u_{L-1}$  with the same ratio  $\gamma$  as the real room impulse response  $\underline{w}(n)$ .

### 2.3.3 Proportionate NLMS algorithm

For acoustic echo cancellation, it is reasonable to assume that the echo path is sparse, i.e., many coefficients are zeros, therefore only the nonzero active coefficients

need to be identified (updated). This is the idea behind the proportionate NLMS (PNLMS) [4] algorithm. It exploits the sparseness of such impulse response to achieve significantly faster adaptation than NLMS.

The PNLMS algorithm updates each coefficient of the filter independently of the others by adjusting the adaptation step-size in proportion to the estimated filter coefficient. Specific equations are

$$u_{i,PNLMS}(n) = \max \left\{ \rho \max \left\{ \delta, |\hat{w}_0(n)|, \dots, |\hat{w}_{L-1}(n)| \right\}, |\hat{w}_i(n)| \right\} \quad (2.9)$$

$$\bar{u}(n) = \frac{1}{N} \sum_{i=0}^{N-1} u_{i,PNLMS}(n) \quad (2.10)$$

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \frac{\beta u_{i,PNLMS}(n) e(n) x(n-i)}{\bar{u}(n) \|x(n)\|^2}$$

where  $\beta$  : scalar

$\rho$  : to avoid filter taps being stuck at zeros forever

$\delta$  : for the case when coefficients are zero

The step-sizes are calculated from the last estimate of the filter coefficients so that a large coefficient receives a large step-size, thus increasing the convergence rate. The concept is similar to ESNLMS algorithm, but the advantage of this technique compared to ESNLMS is that less a priori information is needed e.g.,  $w(n)$ . From Eq.(2.9), the active coefficients with large step-sizes are adjusted faster than non active coefficients (i.e., small or zero coefficients). Hence, PNLMS converges much faster than NLMS for sparse impulse response (i.e., responses for which only a few coefficients are significant).

From Eq.(2.8) and Eq.(2.9), we observe that both ESNLMS and PNLMS algorithms rely on the concept of using large step-size for large tap. It is quite intuitive that large tap will produce large estimate tap coefficient error and should use

large step-size for fast tracking. This is appropriate at the stage of initial adaptation. However, this intuition is no longer sustained after initial period and this is why these two algorithms have two distinct slopes on convergence curve.

From Eq.(2.5), the VSNLMS algorithm uses the power of error signal to adjust the step-size which is different from ESNLMS or PNLMS at first look. However, the same idea of large step-size used for large error (whether individuals or amount all taps), since the error signal is the sum of all taps coefficient error. This makes VNLMS use identical (time-variant) step-size.

Computer simulation will compare standard NLMS algorithm and other variable step-size algorithms described above in Chapter 5.



# Chapter 3

## Optimum Step-size for NLMS

### Algorithm

Most existing work on step-size optimization has considered a constant step-size [9], [10]. Two fairly easy approaches to optimize such a constant step-size are either to minimize the maximum of the absolute value of the eigenvalues of input signal correlation matrix or to minimize the steady state MSE. The latter approach is used most often in practice since after the initial convergence, which is a temporary phenomenon, one has to live with the steady state MSE. This consideration leads to a small step-size and hence slow convergence. Bershad [9] has considered choosing the step-size to minimize the MSE at the end of the observation interval via analysis and numerical evaluation. This is a quite meaningful criterion in applications such as channel equalization for data communications. Chen and Wang [10] derived a closed form formula for the optimum selection of the step-size for a desired MSE. The only attempt to derive an optimum variable step-size (time-variant) sequence was by Slock [7]. Slock determined the optimum step-size (time-variant) to provide the minimum MSE at each iteration step. A closed form solution for an optimum step-size sequence was derived for NLMS algorithm in case of discrete-valued white input signal by

Slock.

We are trying to find out optimum step-size to provide the minimum MSE at each iteration step which is similar to Slock. Slock has assumed that tap coefficient error variances are identical for all taps. This assumption results in time-variant step-size. Instead, we use individual time-variant step-size for each tap due to tap coefficient error variance may not be identical for each tap. The optimum step-size can be obtained by setting the derivative of tap coefficient error variance formula with respect to  $u_i(n)$  equal to zero. Hence, the optimum step-size can provide minimum MSE at each iteration step. Slock also uses discrete-valued white signal as input signal which is different from the continuous-valued white Gaussian signal that we use. We use the RIR exponential decay model for practical implementation. This derivation is based on an LMS algorithm.

We will introduce the optimum step-size sequence of Slock in Section 3.1. In Section 3.3, we derived optimum time- & tap-variant step-size LMS (OLMS) algorithm and the similar method based on the NLMS algorithm can obtain ONLMS algorithm. We also give the general iteration formula for the tap coefficient error variance. Practical implementation and simplified piece step-size of ONLMS algorithm will be proposed in Section 3.4 and Section 3.5, respectively. The comparison of convergence rate of ONLMS and other algorithms will be discussed in Section 3.6. Finally, comparison of computational complexity will be shown in Section 3.7.

### **3.1 Optimum step-size sequence of Slock**

Slock has proposed a time varying (time-variant) step-size sequence [7] that

provided the minimum MSE at each iteration step. A closed form solution for an optimum step-size sequence was derived for NLMS algorithm in case of discrete white input signal described below:

$$u(n) = u(n-1) \frac{1 - \frac{u(n-1)}{L}}{1 - \frac{u^2(n-1)}{L}} \quad (3.1)$$

and

$$u(0) = 1 - \frac{J_{\min}}{\sigma_d^2} \quad (3.2)$$

where  $\sigma_d^2 = E[d^2(0)]$  and  $d(n)$  is the signal received by the microphone . When  $J_{\min} = 0$ ,  $u(0) = 1$  which leads to  $u(n) \equiv 1$ . It means that one should use maximum convergence speed in the noiseless case. On the other hand, if  $J_{\min} = \sigma_d^2$ ,  $u(0) = 0$  and hence  $u(n) \equiv 0$  which also shows: the equality of the noise to the variance of the desired signals means no improvement is possible by adapting the filter. The step-size curve of Eq.(3.1) decays with time which agrees the intuition that large step-size in initial state for fast convergence rate and small step-size in steady state for small misalignment. Different initial values  $u(0)$  will merely correspond to a shift in time of the convergence process.

Slock assumes that tap coefficient error variances are identical for all taps in his analysis. This assumption simplifies the analysis and results only in case of time-variant step-size. However, it may not be hold. This makes Slock's algorithm converges slower then the proposed optimum step-size NLMS algorithm. In next Section, we derive time- & tap-variant optimum step-size without the assumption and further discuss this issue in Section 3.6.

### 3.2 Notations

Before derivation and descriptions, the following notations are used [see also Fig

2.1]:

$$\sigma_z^2 = E[z(n)^2], \text{ noise power}$$

$$\sigma_x^2 = \text{far-end input signal variance}$$

$$\underline{v}(n) = \hat{w}(n) - \underline{w}(n) = [v_0(n), \dots, v_{L-1}(n)]^T, \text{ coefficient error vector}$$

$$\mathbf{U}(n) = \begin{bmatrix} u_0(n) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & u_{L-1}(n) \end{bmatrix} \text{ time- \& tap-variant step-size diagonal matrix}$$

$$\mathbf{R}_x(n) = E[\underline{x}(n)\underline{x}^T(n)] = \mathbf{R}_x, \text{ correlation matrix of the far-end signal}$$

$$\mathbf{R}_v(n) = E[\underline{v}(n)\underline{v}^T(n)] = \begin{bmatrix} g_0(n) & & \\ & \ddots & \\ & & g_{L-1}(n) \end{bmatrix}, \text{ correlation matrix of tap coefficient error}$$

Different from Slock, we assume the input signal  $\underline{x}(n)$  is continuous-valued white Gaussian noise (WGN) in the following Sections. For simplicity, the assumption of echo path is time invariant (i.e.,  $\underline{w}(n) = \underline{w}$ ) and near-end speech  $t(n) = 0$  are also made.



### 3.3 Derivation of optimum time- & tap-variant step-size

#### LMS algorithm

We want to find out the step-size which can minimize each tap coefficient error variance i.e., MSE for each iteration step. Hence, we use step-size diagonal matrix  $\mathbf{U}(n)$  for LMS algorithm of Eq.(2.2) and (2.3) which can be rewritten as:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{U}(n)e(n)\underline{\mathbf{x}}(n) \quad (3.3)$$

$$e(n) = d(n) - \hat{\mathbf{w}}^T(n)\underline{\mathbf{x}}(n) \quad (3.4)$$

We plug  $\hat{\mathbf{w}}(n) = \underline{\mathbf{v}}(n) + \underline{\mathbf{w}}$  into Eq.(3.4) so that

$$e(n) = d(n) - (\underline{\mathbf{v}}(n) + \underline{\mathbf{w}})^T \underline{\mathbf{x}}(n) \quad (3.5)$$

Substitute Eq.(3.5) into Eq.(3.3) to obtain

$$\begin{aligned} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \mathbf{U}(n)\underline{\mathbf{x}}(n)[d(n) - (\underline{\mathbf{v}}(n) + \underline{\mathbf{w}})^T \underline{\mathbf{x}}(n)] \\ &= \hat{\mathbf{w}}(n) + \mathbf{U}(n)\underline{\mathbf{x}}(n)d(n) - \mathbf{U}(n)\underline{\mathbf{x}}(n)(\underline{\mathbf{v}}(n) + \underline{\mathbf{w}})^T \underline{\mathbf{x}}(n) \end{aligned} \quad (3.6)$$

and the coefficient error vector  $\underline{\mathbf{v}}(n) = \hat{\mathbf{w}}(n) - \underline{\mathbf{w}}$  at time  $n+1$  becomes

$$\begin{aligned} \underline{\mathbf{v}}(n+1) &= \hat{\mathbf{w}}(n+1) - \underline{\mathbf{w}} \\ &= \hat{\mathbf{w}}(n) + \mathbf{U}(n)\underline{\mathbf{x}}(n)d(n) - \mathbf{U}(n)\underline{\mathbf{x}}(n)(\underline{\mathbf{v}}(n) + \underline{\mathbf{w}})^T \underline{\mathbf{x}}(n) - \underline{\mathbf{w}} \\ &= \underline{\mathbf{v}}(n) - \mathbf{U}(n)\underline{\mathbf{x}}(n)\underline{\mathbf{v}}^T(n)\underline{\mathbf{x}}(n) + \mathbf{U}(n)\underline{\mathbf{x}}(n)[d(n) - \underline{\mathbf{w}}^T \underline{\mathbf{x}}(n)] \\ &= [I - \mathbf{U}(n)\underline{\mathbf{x}}(n)\underline{\mathbf{x}}^T(n)]\underline{\mathbf{v}}(n) + \mathbf{U}(n)\underline{\mathbf{x}}(n)z(n) \end{aligned} \quad (3.7)$$

We may now express the correlation matrix of the coefficient error as follows:

$$\begin{aligned} \mathbf{R}_v(n+1) &= \mathbf{R}_v(n) - 2\mathbf{U}(n)E[\underline{\mathbf{x}}(n)\underline{\mathbf{x}}^T(n)\underline{\mathbf{v}}(n)\underline{\mathbf{v}}^T(n)] + \mathbf{U}^2(n)E[\underline{\mathbf{x}}(n)\underline{\mathbf{x}}^T(n)\underline{\mathbf{v}}(n)\underline{\mathbf{v}}^T(n)\underline{\mathbf{x}}(n)\underline{\mathbf{x}}^T(n)] \\ &\quad + \sigma_z^2\mathbf{U}^2(n)\mathbf{R}_x \end{aligned} \quad (3.8)$$

where the cross-products terms  $[I - \mathbf{U}(n)\underline{\mathbf{x}}(n)\underline{\mathbf{x}}^T(n)]\underline{\mathbf{v}}(n)$  and  $\mathbf{U}(n)\underline{\mathbf{x}}(n)z(n)$  in

Eq.(3.7) disappear because of the independent and zero mean assumptions of the noise  $z(n)$ .

We assume the step size is small, so that the LMS filter acts as a low-pass filter with a low cutoff frequency [1]. With this assumption, the variation of the coefficient error vector  $\underline{v}(n)$  with time is slow compared with that of the input vector  $\underline{x}(n)$ . In the direct-averaging method [1], we may replace the product  $\underline{x}(n)\underline{x}^T(n)$  by its expected value and Eq.(3.8) becomes:

$$\begin{aligned} \mathbf{R}_v(n+1) = & \mathbf{R}_v(n) - 2\mathbf{U}(n)\mathbf{R}_x\mathbf{R}_v(n) + \mathbf{U}^2(n)E[\underline{x}(n)\underline{x}^T(n)\underline{v}(n)\underline{v}^T(n)\underline{x}(n)\underline{x}^T(n)] \\ & + \sigma_z^2\mathbf{U}^2(n)\mathbf{R}_x \end{aligned} \quad (3.9)$$

The Gaussian assumption (see in the Appendix) is used next to simplify the third term in Eq.(3.9), then we can obtain

$$\mathbf{R}_v(n+1) = (\mathbf{I} - 2\mathbf{U}(n)\mathbf{R}_x)\mathbf{R}_v(n) + \sigma_x^4\mathbf{U}^2(n)[2\mathbf{R}_v(n) + \text{trace}(\mathbf{R}_v(n))\cdot\mathbf{I}] + \sigma_z^2\mathbf{U}^2(n)\mathbf{R}_x \quad (3.10)$$

The diagonal terms of Eq.(3.10) become

$$g_i(n+1) = (1 - 2u_i(n)\sigma_x^2 + 2u_i^2(n)\sigma_x^4)g_i(n) + u_i^2(n)\sigma_x^4\sum_{j=0}^{L-1}g_j(n) + u_i^2(n)\sigma_x^2\sigma_z^2 \quad (3.11)$$

Eq.(3.11) is the general iteration form of tap coefficient error variance. If we assume tap coefficient error variance are identical i.e.,  $g_0(n) = \dots = g_{L-1}(n)$ , then the fourth term of Eq.(3.11) will become  $Lu^2(n)\sigma_x^4g(n)$  and Eq.(3.11) will be independent of  $i$ , which finally results in optimum time-variant step-size. However, this assumption used by Slock may not hold. We use identical (time-variant) step-size for individual tap coefficient error variance to verify it. Based on NLMS algorithm, Eq.(3.11) can be rewritten as

$$g_i(n+1) = (1 - 2\frac{u(n)}{L}\sigma_x^2 + 2\frac{u^2(n)}{L^2}\sigma_x^4)g_i(n) + \frac{u^2(n)\sigma_x^2}{L^2}(\sigma_x^2\sum_{j=0}^{L-1}g_j(n) + \sigma_z^2) \quad (3.12)$$

Due to different initial value  $g_i(0) = E[(\hat{w}(0) - w_i)^2] = w_i^2$ , coefficient error variance for different tap will not be identical for any  $n$ . Thus, using identical step-size for every tap converges slower than using individual step-size i.e., Slock's algorithm has slower convergence rate than the ONLMS algorithm.

The optimum time-& tap-variant step-size can be obtained by taking derivative of Eq.(3.11) with respect to  $u_i(n)$  and setting the result equal to zero.

$$\frac{dg_i(n+1)}{du_i(n)} = -2\sigma_x^2g_i(n) + 4u_i(n)\sigma_x^4g_i(n) + 2u_i(n)\sigma_x^4\sum_{j=0}^{L-1}g_j(n) + 2u_i(n)\sigma_x^2\sigma_z^2 = 0 \quad (3.13)$$

Thus we can get the optimum time- & tap-variant step-size

$$u_{i,OLMS}(n) = \frac{g_i(n)}{\sigma_x^2(2g_i(n) + \sum_{j=0}^{L-1}g_j(n) + \sigma_z^2)} \quad [\text{LMS}] \quad (3.14)$$

for  $i = 0, \dots, L-1$ . The similar method based on the NLMS algorithm by assuming the input signal  $\underline{x}(n)$  is WSS with zero mean can be obtained as follows:

$$u_{i,ONLMS}(n) = \frac{Lg_i(n)}{(2g_i(n) + \sum_{j=0}^{L-1}g_j(n) + \sigma_z^2)} \quad [\text{NLMS}] \quad (3.15)$$

for  $i = 0, \dots, L-1$ . Difference between Eq.(3.14) and Eq.(3.15) is due to the denominator in Eq.(2.4), note that  $\|\underline{x}(n)\|^2 \approx L\sigma_x^2$ .

The step-size obtained from this approach is optimum in the sense that, if it is used from  $n=0$  to  $n=k$ , the tap coefficients error at the time instant  $n=k$  is minimum.

Substitute Eq.(3.14) into Eq.(3.11), we can get a recursive formula for the tap coefficient error variance  $g_i(n)$  as follows:

$$g_i(n+1) = g_i(n)(1 - \sigma_x^2 u_{i,ONLMS}(n)) \quad [\text{LMS}] \quad (3.16)$$

for  $i = 0, \dots, L-1$  and similarly

$$g_i(n+1) = g_i(n) \left(1 - \frac{u_{i,ONLMS}(n)}{L}\right) \quad [\text{NLMS}] \quad (3.17)$$

The ONLMS algorithm is summarized by the following equations:

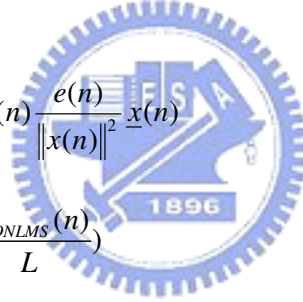
$$1. \quad e(n) = d(n) - \hat{\mathbf{w}}^T(n) \mathbf{x}(n)$$

$$2. \quad \mathbf{U}_{ONLMS}(n) = \frac{L \mathbf{R}_v(n)}{(2\mathbf{R}_{v,ii}(n) + \text{tr}(\mathbf{R}_v(n)) + \sigma_z^2 \mathbf{I})} \quad \text{where}$$

$$\mathbf{U}_{ONLMS}(n) = \begin{pmatrix} u_{0,ONLMS}(n) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & u_{L-1,ONLMS}(n) \end{pmatrix}$$

$$3. \quad \hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{U}_{ONLMS}(n) \frac{e(n)}{\|\mathbf{x}(n)\|^2} \mathbf{x}(n)$$

$$4. \quad \mathbf{R}_v(n+1) = \mathbf{R}_v(n) \left(\mathbf{I} - \frac{\mathbf{U}_{ONLMS}(n)}{L}\right)$$



The step-size adjustment is based on (2) and (4).

We will use ONLMS algorithm in the following discussion.

### 3.4 Practical ONLMS algorithm

We have derived Optimum time- & tap-variant step-size for NLMS algorithm in Eq.(3.15). But it is useless since it requires prior knowledge of RIR  $w_i$  due to  $g_i(n) = E[(\hat{w}(n) - w_i)^2]$ . However, we can use the exponential decay model of RIR and use the iteration form of  $g_i(n)$  in Eq.(3.17) to for practical implementation.

Assume the RIR  $w_i$  can be modeled as an exponential decay envelope  $h_i$  shown in Figure 3.1. Let decay envelop function be:

$$h_i = h_0 \gamma^i \quad \text{for } i = 1, \dots, L-1 \quad (3.18)$$

where  $\gamma$  is the room exponential decay factor.

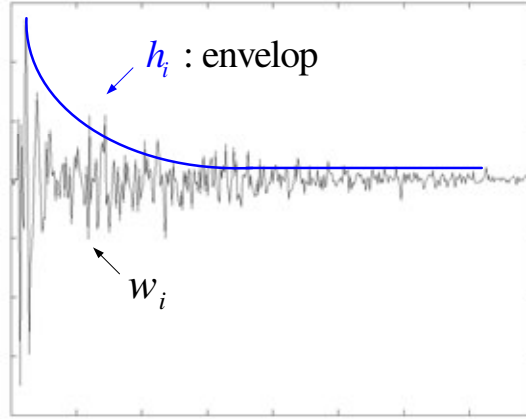


Figure 3.1 RIR decay envelop

The diagonal element of tap coefficient error variance matrix  $\mathbf{R}_v(n)$  is  $g_i(n) = E[(\hat{w}_i(n) - w_i)^2]$ ,  $i = 0, \dots, L-1$ . We let the initial filter tap coefficients to be zero i.e.  $\hat{w}_i(0) = 0$  so that  $g_i(0) = E[(\hat{w}_i(0) - w_i)^2] = w_i^2 \approx h_i^2$ . We substitute  $g_i(0)$  into Eq.(3.15) to get  $u_i(0)$ . With  $u_i(0)$  plugged into Eq.(3.17) we can get  $g_i(1)$  and so forth i.e.,  $g_i(0) \rightarrow u_i(0) \rightarrow g_i(1) \rightarrow u_i(1) \rightarrow \dots$ . Thus, we can obtain  $u_i(n)$  for  $i = 0, \dots, L-1$  at any iteration step. The practical ONLMS algorithm with an exponential RIR model can be described as follows:

1. Measure room exponential decay factor  $\gamma$  to get  $h_i = h_0 \gamma^i$
2. Set up initial value  $g_i(0) \approx h_i^2$  for  $i = 0, \dots, L-1$

$$3. \mathbf{U}_{ONLMS}(n) = \frac{L\mathbf{R}_v(n)}{(2\mathbf{R}_{v,ii}(n) + tr(\mathbf{R}_v(n)) + \sigma_z^2)\mathbf{I}} \quad \text{where}$$

$$\mathbf{U}_{ONLMS}(n) = \begin{pmatrix} u_{0,ONLMS}(n) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & u_{L-1,ONLMS}(n) \end{pmatrix}$$

$$4. e(n) = d(n) - \hat{\mathbf{w}}^T(n)\underline{\mathbf{x}}(n)$$

$$5. \hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{U}_{ONLMS}(n)e(n)\underline{\mathbf{x}}(n)$$

$$6. \mathbf{R}_v(n+1) = \mathbf{R}_v(n)\left(\mathbf{I} - \frac{\mathbf{U}_{ONLMS}(n)}{L}\right)$$

By using the exponential RIR model, we can implement the ONLMS algorithm.

### 3.5 Simplified ONLMS algorithm

Using optimum time- & tap-variant step-size indeed increases convergence rate. However, this algorithm has a higher computational cost than the conventional LMS algorithms.

In order to reduce the computational load, we propose another simplified version of ONLMS algorithm which uses piecewise time-variant and tap-variant step-size. The method of choosing the step-size is described as follows:

1.  $0 \leq n \leq n_0$  (initial state): At  $n = 0$ , from Eq.(3.15) we obtain

$$u_{i,ONLMS}(0) = \frac{Lg_i(0)}{(2g_i(0) + \sum_{j=0}^{L-1} g_j(0) + \sigma_z^2)} \cong \alpha g_i(0) = \alpha w_i^2 \approx \alpha h_i^2 \quad (3.20)$$

so  $u_i$  can be set proportional to  $g_i(0) \approx h_i^2$  i.e.,

$$u_i \triangleq \alpha h_i^2 \quad \text{for } i = 0, \dots, L-1 \quad (3.21)$$

Eq.(3.21) satisfied Eq.(3.15) only at  $n=0$ . Convergence rate is fast only at the initial state and then slows down when  $n > n_0$ .

2.  $n > n_1$  (steady state): We substitute Eq.(3.15) into Eq.(3.17) to obtain

$$g_i(n+1) = g_i(n) \left( \frac{g_i(n) + \sum_{j=0}^{L-1} g_j(n) + \sigma_z^2}{2g_i(n) + \sum_{j=0}^{L-1} g_j(n) + \sigma_z^2} \right) \quad (3.22)$$

Since  $g_i(n) \ll \sum_{j=0}^{L-1} g_j(n) + \sigma_z^2$ , the fractional term of Eq.(3.22) will be close to but

less than unity. After  $n_1$  iterations, the difference among  $g_i(n)$  for  $i=0, \dots, L-1$  can be regarded as zero. Thus for  $n > n_1$ ,  $g_0(n) \approx \dots \approx g_{L-1}(n)$ , so  $u_i$  can be set proportional to  $g_i^0(n) = 1$  (constant) i.e.,

$$u_i \triangleq \alpha' \quad \forall i \quad (3.23)$$

This turns into NLMS algorithm. Eq.(3.23) satisfied Eq.(3.15) when  $n > n_1$ . Convergence rate is fast only at steady state.

3. For transient state  $n_0 \leq n \leq n_1$ , we set  $u_i$  proportional to  $g_i^{1/2}(0)$  i.e.,

$$u_i \triangleq \alpha'' h_i \quad \text{for } i=0, \dots, L-1 \quad (3.24)$$

This turns out to be the ESNLMS algorithm.

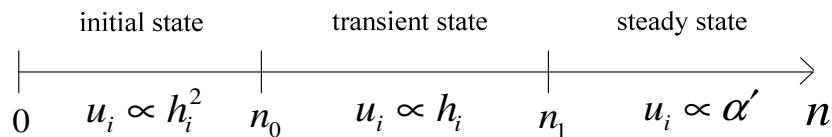


Figure 3.2 Piecewise time-variant and tap-variant step-size for different state

In conclusion, Figure 3.2 illustrates three appropriate choices of step-size in different state. This results in piecewise time-variant and tap-variant step-size which reduces the computational load. However, simplified ONLMS does not perform as well as ONLMS.

### 3.6 Comparison of convergence

In this Section, we are trying to find out the reason that ONLMS has fastest convergence rate than other algorithms and why PNLMS and ESNLMS have two distinct slopes of convergence curve from the point of view of the tap coefficient error variance. The average power of error signal  $e(n)$  can be treated as convergence gain energy which is composed of all tap coefficient error variance, and can be expressed as [1]:

$$J(n) = E[|e(n)|^2] = \sigma_x^2 \sum_{i=0}^{L-1} g_i(n) + \sigma_z^2 \quad (3.25)$$

We can find out that both of the ESNLMS and PNLMS algorithms use large step-size for large taps and small step-size for small taps from Eq.(2.8) and Eq.(2.9), respectively. This means large taps have large  $g_i(n)$  for these two algorithms at initial state.

At the stage of initial adaptation, the power of error signal is dominated by the contribution from large taps (large  $g_i(n)$ ). Convergence rate is very fast because large taps use large step-size. Eventually, however, the amount  $g_i(n)$  at these taps reaches a point where it is becoming comparable to the amount  $g_i(n)$  from the



others small taps, the amount  $g_i(n)$  from small taps starts to dominate the power of error signal. The convergence rate then slows down because small taps use small step-size. This phenomenon makes both ESNLMS and PNLMS algorithms have two distinct slopes of convergence curve. For PNLMS algorithm, initial rapidly period through experiment [4] will last until ERLE achieved  $ERLE_{\text{BREAK}} = -10\log_{10}(\rho)$  where  $\rho$  is defined in Eq.(2.9). Once  $ERLE_{\text{BREAK}}$  has been achieved, PNLMS is no longer offering any advantage over NLMS with a constant step-size.

Instead of just using large step-size, the ONLMS algorithm sets step-size directly proportionate to  $g_i(n)$  which is similar to PNLMS and ESNLMS algorithms at initial state. Large  $g_i(n)$  uses large step-size to increase the convergence rate. After the initial period,  $g_i(n)$  becomes identical for every tap, which means the  $J(n)$  is composed of every  $g_i(n)$  uniformly. From Eq.(3.15), identical  $g_i(n)$  results in identical  $u_i(n)$  thus convergence rate will not slow down.

In conclusion,  $u_i(n)$  should be set proportional to  $g_i(n)$  for fast convergence rate. Both PNLMS and ESNLMS algorithms use large step-size for large taps. This is appropriate since large taps indeed have large  $g_i(n)$  at initial stage. However, when the values of  $g_i(n)$  for large taps are less than those for small taps, the convergence rate then slows down. On the other hand,  $g_i(n)$  for every tap will become identical for ONLMS algorithm after initial period. This results in identical  $u_i(n)$  and will not slow down the convergence rate.

### 3.7 Computational complexity

We have already discussed several approaches of adjusting step-sizes which increase the convergence rate. However, using time variant or tap-variant step-size has high computational complexity compared with the constant step-size.

The usual way to measure algorithm complexity is to compute required numbers of multiplications. Under this measure, LMS algorithm has a complexity of order  $2L$  since every sample every one of  $L$  impulse response estimated  $\hat{w}_i(n)$  must be multiplied by a delayed far-end signal to perform the convolution equation and these same delayed far-end signal samples must also be multiplied by a scaling of  $e(n)$  to perform the correlation needed in the update equation. NLMS algorithm has a complexity of order  $3L$  because in addition to the multiplies required before, squares of each of the  $x(n-i)$  are needed to form  $\|x(n)\|^2$ .

In practice, NLMS is usually implemented as an order  $2L$  algorithm with the simple trick of maintaining a running  $\|x(n)\|^2$  by subtracting off the square of the outgoing sample. We assume this trick to be used by other algorithms as NLMS and the computational complexities of various algorithms are shown in Table 3.1.

We can find that the cost of increased convergence rate is the increased computation load. Traditional step-size adjustment approaches need  $L$  multiplications more compared to NLMS at every adaptation. ONLMS with has the fastest convergence rate needs additional  $2L$  multiplications.

Algorithm	Multiplications/sample
LMS	2L
NLMS	2L
VSNLMS	3L
ESNLMS	3L
PNLMS	3L
ONLMS	4L
Simplified ONLMS	3L

Table 3.1 Computational requirements for different algorithms

# Chapter 4

## Double Talk Detection

In chapter 3, we introduced different adaptive algorithms for acoustic echo canceller. These adaptive algorithms always perform well in the single talk (ST) state. However, in the double talk (DT) state, we can see from Figure 2.1 that microphone input signal includes both near-end talker signal  $t(n)$  and echo signal  $y(n)$ , the filter tap coefficients can fluctuate greatly or diverge to misestimate the echo path which means near-end signal acts as a large interference to the adaptive filter. Hence AEC should stop the filter adaptation during the double talk period. Therefore, double talk detectors are used in AEC to avoid DT situation.

Double talk detection (DTD) plays a very important part in AEC. The basic requirement for DTD is that it is able to detect DT quickly and accurately. Besides, it should also have the ability to distinguish the DT and room echo path change (EPC). In the past, there have been lots of researches on DTD. The conventional DTD algorithms are classified into three categories: (1) Level comparison type [11] is used to detect DT by comparing the error signal level with the primary input signal level e.g. Geigel DTD. (2) CLMS algorithm [12] is used to distinguish DT from varying

echo path and expanded CLMS has better performance than CLMS at the cost of higher computational complexity. (3) Cross-correlation type [13] can detect DT by making use of the orthogonal principle. In this Chapter, we adopt the level comparison to detect the abrupt change in error signal (signals sent to the far-end loudspeaker). In Section 4.1, the DT issue in echo canceller will be introduced and discussed. In Section 4.2, we will introduce a control scheme [14] which can discriminate DT and EPC after a short period  $n_d$ . We are trying to find out the correct probability of system decision under DT or EPC situation. Further analysis will be in Section 4.3.

## 4.1 Double talk in echo canceller

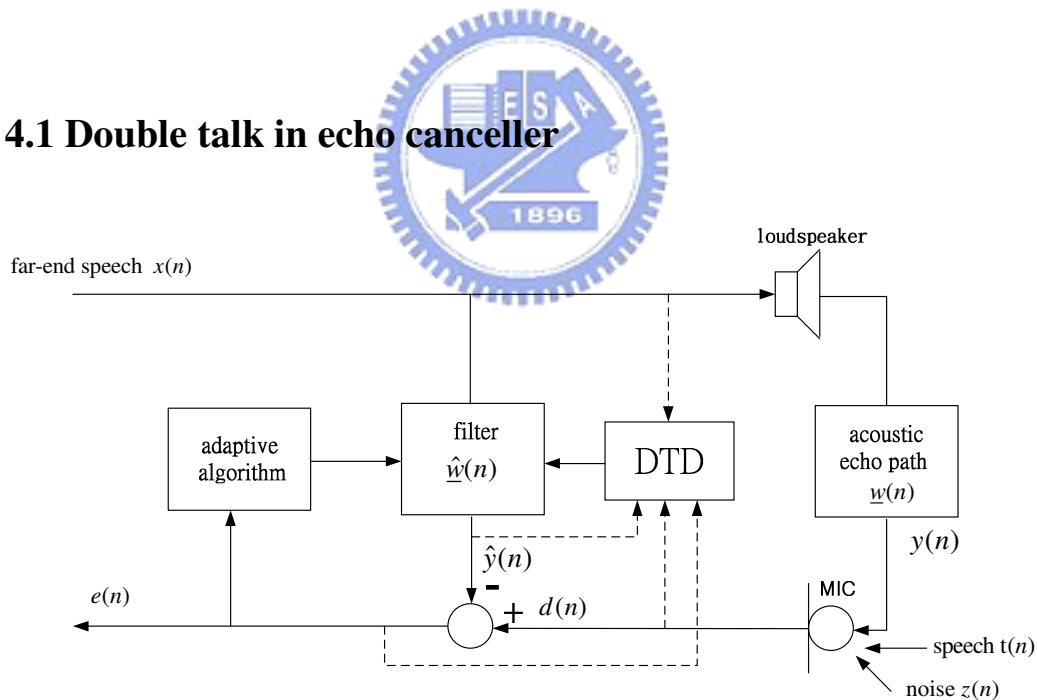


Figure 4.1 Block diagram of the AEC with DTD.

One problem of the AEC is that the performance deteriorates drastically during the DT periods in which signals from both near-end and far-end speakers coexist. In a

common telephone call, the double talk is found to occupy up to 20 percent of the whole period [3]. During the DT periods, the echo is mixed into a large and strongly correlated near-end speech  $t(n)$  and results in a large interference component in the error  $e(n)$ . Consequently, the AEC will quickly diverge from its converged state if the adaptive filter continues to update its coefficients during the DT periods.

We use the DTD to avoid the situation discussed above. According to the decision of the DTD, the adaptive filter updates its coefficients during ST periods and freezes adaptation during DT periods to avoid divergence. The basic DTD block diagram is shown in Figure 4.1. A simple and efficient way of detecting DT is to compare the magnitude of the far-end and near-end signals and declare DT if the near-end magnitude becomes large than a value set by the far-end signals. A proven algorithm that has been in commercial use for many years is the Geigel DTD [15]. In this algorithm, DT is declared if

$$|d(n)| \geq \beta \max\{|x(n)|, |x(n-1)|, \dots, |x(n-L+1)|\} \quad (4.1)$$

The detector threshold  $\beta$  is set to 0.5 if the room attenuation is assumed to be 6 dB and to 0.71 if the attenuation is assumed to be 3 dB. The Geigel DTD is simple and fast. However, when the magnitude of  $d(n)$  is -6dB lower than  $\beta \max\{|x(n)|, |x(n-1)|, \dots, |x(n-L+1)|\}$  during DT, the Geigel DTD fails to detect the DT. We can use another level comparison method to avoid this situation:

$$|x(n)|^2 < C \cdot |e(n)|^2 \quad (4.2)$$

where  $C$  is a constant larger than 1. If Eq.(4.2) takes place, then an abrupt change happened in error signal  $e(n)$ . It will cause large  $e(n)$  when DT happened, regardless of the magnitude of  $d(n)$ .

## 4.2 A scheme to distinguish EPC and DT

To distinguish EPC and DT is important for DTD because the adaptive filter coefficients should be continuously updated during EPC but should not during double talk periods. An adaptation architecture proposed in [14] illustrated in Figure 4.2 to achieve this goal. This scheme uses a counter to record the number of events happened within a short period  $n_d$  when an abrupt change in the error signal. After  $n_d$ , the system can decide whether DT or EPC causes the abrupt change by compare the number of events with some threshold .

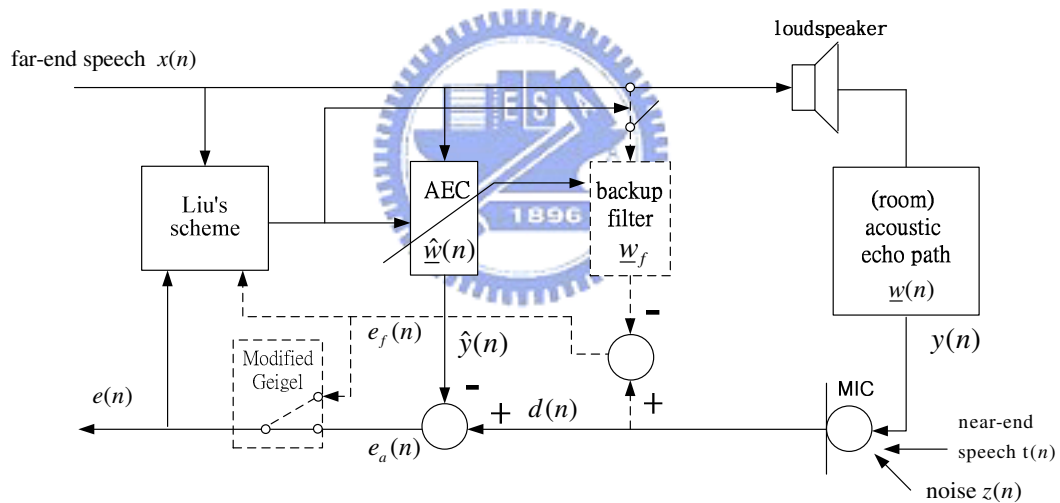


Figure 4.2 Liu's scheme to distinguish DT or EPC.

We assume the convergence state is reached at beginning, then use Eq.(4.2) to detect an abrupt change. In Figure 4.2, whenever an abrupt change between  $e(n)$  and  $x(n)$  is detected, an additional backup filter  $\underline{w}_f$  and the modified Geigel detector is initiated. Within a short time  $n_d$ , the modified Geigel detector compares

$|e_a(n)|$  and  $|e_f(n)|$  then outputs the smaller one to the far-end loudspeaker (error signal). This scheme is also used for detection of DT or EPC, while  $\underline{w}_f$  is used to memorize the filter coefficients  $\hat{w}(n)$  when the abrupt change occurs.

Suppose an abrupt change is detected. The filter  $\underline{w}_f$  and its corresponding control logic as denoted by the dashed line in Figure 4.2 are activated immediately and the system yields

$$\begin{aligned}\underline{w}_f &= \hat{w}(n_f) \\ e_a(n) &= d(n) - \hat{w}^T(n) \cdot \underline{x}(n) \\ e_f(n) &= d(n) - \underline{w}_f^T \cdot \underline{x}(n)\end{aligned}\quad (4.3)$$

The output of error signal  $e(n)$  in the architecture during  $n_d$  is controlled by the modified Geigel as follows (assume room attenuation 6 dB)

$$e(n) = \begin{cases} e_a(n), & |d(n)| < 0.5 \cdot \max\{|x(n)|, |x(n-1)|, \dots, |x(n-L+1)|\} \\ e_f(n), & |d(n)| \geq 0.5 \cdot \max\{|x(n)|, |x(n-1)|, \dots, |x(n-L+1)|\} \end{cases}\quad (4.4)$$

After time  $n_d$ , the modified Geigel detector will be disabled.

Let  $N$  denote the number of abrupt events when  $|e_a(n)|$  is greater than  $|e_f(n)|$  within time period  $n_d$ . If the increase of error signal is due to DT, then  $|e_a(n)|$  using adaptive AEC filter  $\hat{w}(n)$  should be higher than  $|e_f(n)|$  using the frozen backup filter  $\underline{w}_f$ . Otherwise, if the error signal increase is due to EPC,  $|e_f(n)|$  should be higher than  $|e_a(n)|$ . Hence, the control logic after  $n_d$  can decide whether DT or EPC causes the abrupt change according to  $N$  as follows:

$$\text{If } N = \begin{cases} < N_T & \text{echo path change} \\ \geq N_T & \text{double talk} \end{cases}\quad (4.5)$$

where  $N_T$  is the decision threshold for discriminating EPC from DT. The adaptive



filter coefficients after  $n_d$  performs as follows

$$\hat{\underline{w}}(n+1) = \begin{cases} \hat{\underline{w}}(n) + \frac{u}{\|\underline{x}(n)\|^2} e(n) & \text{echo path change} \\ \underline{w}(n_f) & \text{double talk} \end{cases} \quad (4.6)$$

The backup filter  $\underline{w}_f$  with its corresponding control mechanism, disappears after  $n_d$ , and the above logic continues until  $|x(n)|^2 \geq C \cdot |e(n)|^2$ .

Below is a brief summary of the general process of the approach: If an abrupt change is detected at steady (converged) state, the backup filter coupled with the original AEC filter and its corresponding control logic are initialed. Error signal is the output according to the modified Geigel algorithm within a short delay  $n_d$  before new convergence. After  $n_d$ , the system then determines DT or EPC according to Eq.(4.5). If the cause of the abrupt change in the error signal is DT, the adaptive filter  $\hat{\underline{w}}(n)$  is substituted with  $\underline{w}_f$ , and the algorithm stops the adaptation process ( $\hat{\underline{w}}(n+1) = \hat{\underline{w}}(n)$ ), which prohibits coefficient update. Otherwise, the algorithm continues to update.

### 4.3 Analysis of correct decision probability

Although Liu has proposed a scheme to distinguish DT or EPC, it requires further analysis on correct decision probability. In this Section, we will focus on the correct decision probability (or error probability) under DT situation using Liu's scheme.

Figure 4.3 illustrates the curve of misalignment  $\left\| \frac{\underline{w} - \hat{\underline{w}}(n)}{\underline{w}} \right\|$ .

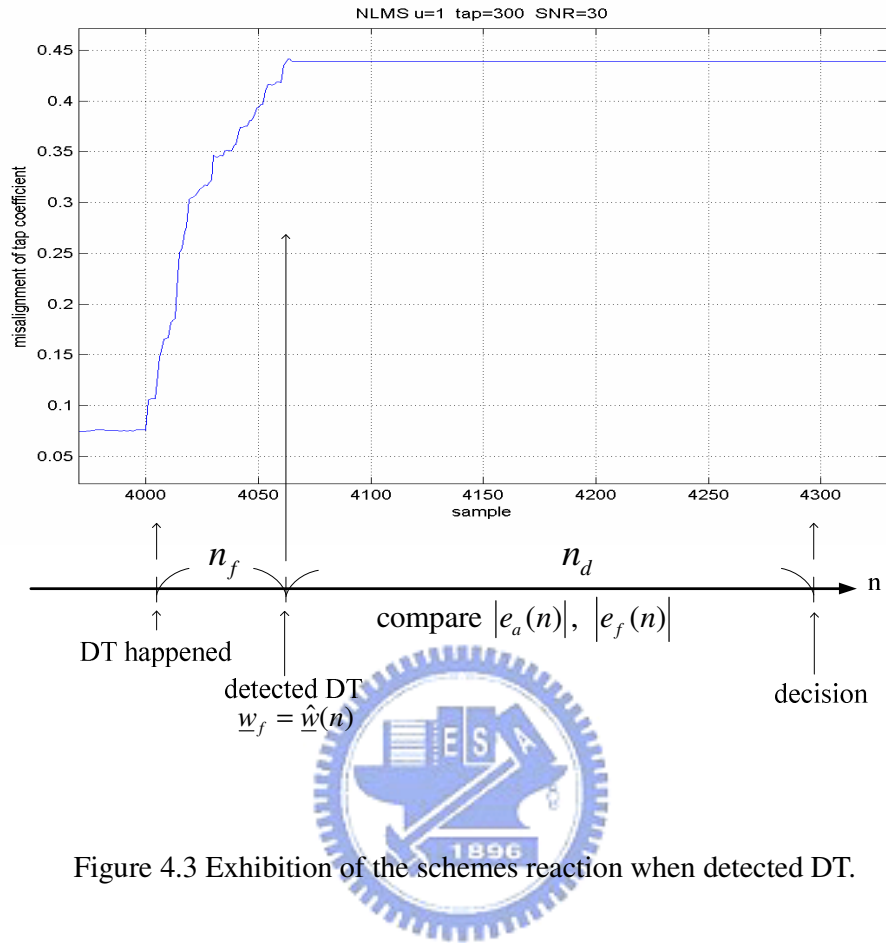


Figure 4.3 Exhibition of the schemes reaction when detected DT.

DT happens at  $n = 4000$  and is detected at  $n = 4065$ . The system starts to compare  $e_a(n)$  and  $e_f(n)$  then decides the cause of abrupt change in error signal after  $n_d$ .

#### 4.3.1 Assume $|e_a(n)|$ as Gaussian random variable

The correct decision of DT situation means when DT occurs,  $N \geq N_T$  after  $n_d$ .

In other words, we want to find out the probability (correct probability) that  $|e_a(n)|$

has at least  $N_T$  times greater than  $|e_f(n)|$  within  $n_d$  sample period. We assume

$|e_a(n)|$  during for  $n_d$  time (sample), denoted as  $x_1, x_2, \dots, x_{n_d}$ , are Gaussian random variables with same variance  $\sigma^2$  and different means  $a_1, \dots, a_{n_d}$  i.e.,  $x_i \sim N(a_i, \sigma_x^2)$  illustrated in Figure 4.4. We use  $a_0$  to represent  $|e_f(n)|$  which is assumed to be a constant. It is actually related to the parameter  $C$  in Eq.(4.2).

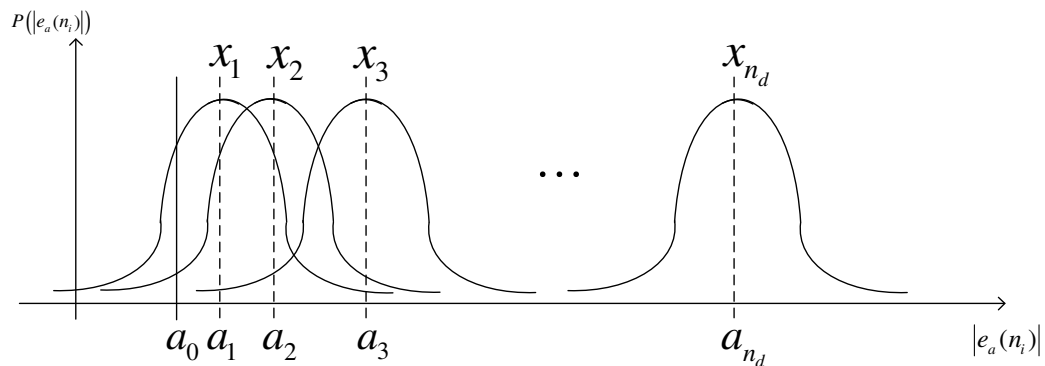


Figure 4.4 Each sample  $|e_a(n)|$  is assumed to be a Gaussian random variable within

the period  $n_d$  samples

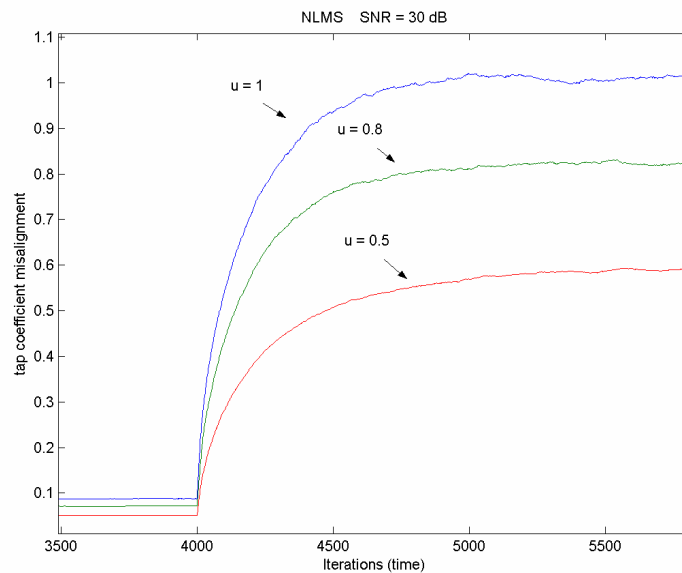


Figure 4.5 Tap coefficient misalignments with different step-sizes under DT situation

Figure 4.5 averages fifty times experiments of the tap coefficient misalignment when DT happens at  $n = 4000$ . From Figure 4.5, we find out that tap coefficient misalignment increases exponentially when continuing the adaptation under DT situation. Since tap coefficient misalignment is another expression of error signal  $|e(n)|$ , we can assume  $a_i = a_1 \cdot e^{(i-1)r}$  for  $i = 2, \dots, n_d$ . The exponential factor  $\gamma$  is related with the step-size.

We use  $A_i$  to represent the abrupt event  $x_i > a_0$  (i.e.,  $|e_a(n_i)| > |e_f(n)|$ ) under DT situation. The probability of  $A_i$  can be expressed as

$$P(A_i) = P(x_i > a_0) = \int_{a_0}^{\infty} f(x_i) dx_i \quad (4.7)$$

where

$$f(x_i) = \frac{1}{\sqrt{2\pi}\sigma_x} \cdot \exp\left(-\frac{(x_i - a_i)^2}{2\sigma_x^2}\right) \quad (4.8)$$

It's hard to express the probability of detect DT correctly since  $P(A_i)$  is different for  $i = 1, \dots, n_d$ . For example, if  $n_d = 3, N_T = 2$  the probability of the system can correctly detect DT situation is written as

$$P(\text{correct}) = P(A_1)P(A_2)(1 - P(A_3)) + P(A_1)P(A_3)(1 - P(A_2)) + P(A_2)P(A_3)(1 - P(A_1))$$

It is even more complicate if  $n_d$  and  $N_T$  are even larger, so we try another way to discuss this issue.

### 4.3.2 Decision sample

We use the random variable  $K$  called "decision sample" to describe the number of needed samples before the number of abrupt events  $x_i > a_0$  (i.e.,  $|e_a(n_i)| > |e_f(n)|$ ) reaches  $N_T$  under DT situation. We can see from

Figure 4.5 that tap coefficient misalignment curves rise sharply at the initial stage when DT happened, which means the number of abrupt events  $x_i > a_0$  will increase closely and rapidly in the meantime. Thus,  $K$  can be modeled as an exponential distribution illustrated as below

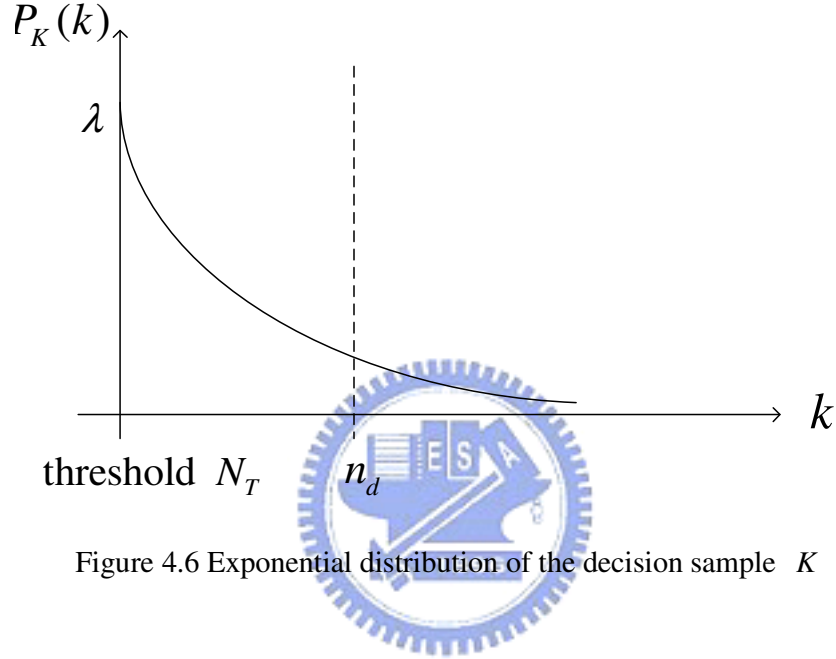


Figure 4.6 Exponential distribution of the decision sample  $K$

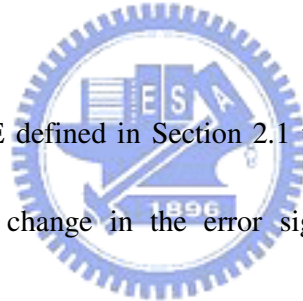
$$P_K(k) = \lambda e^{-\lambda(k-N_T)} \quad (4.9)$$

If  $K$  exceeds  $n_d$ , which means there are not enough abrupt events  $x_i > a_0$  within  $n_d$  period, therefore the system would declare EPC after  $n_d$  under DT situation. Right side of the dash line in Figure 4.6 indicates the probability that the system is unable to detect DT after  $n_d$  under DT situation, so the correct probability can be written as follows:

$$\begin{aligned} P(K \leq n_d) &= 1 - P(K > n_d) \\ &= 1 - e^{-\lambda n_d} \end{aligned} \quad (4.10)$$

The factor  $\lambda$  is a function of  $N_T$  and it will be increased while using smaller  $N_T$ .

Smaller  $N_T$  can raises up the correct probability of system decision under DT situation. However, smaller  $N_T$  is not only sensitive to the back ground noise but also increases the error probability of system decision under EPC situation. It is obviously a tradeoff for system decision of an abrupt change in the error signal. So  $N_T$  plays an important role for system decision of an abrupt change in error signal. The relation between  $\lambda$  and  $N_T$  is roughly inversely proportional from experiments. We will demonstrate this phenomenon in chapter 5. The correct probability of system decision will increase when  $n_d$  is larger, but it also increases the detection time of the system.



We then use the ERLE defined in Section 2.1 to represent the cost of system failing to detect an abrupt change in the error signal. Different  $N_T$  results in different average ERLE for both DT and EPC situation, which is defined as

$$\text{ERLE}(N_T) = P(\text{error})\text{ERLE}_{\text{error}} + P(\text{correct})\text{ERLE}_{\text{correct}} \quad (4.11)$$

where  $\text{ERLE}_{\text{error}}$  is the cost ERLE of a wrong decision and  $\text{ERLE}_{\text{correct}}$  means a correct decision. We assume an abrupt change can be detected instantly, i.e.,  $\underline{w}_f \approx \underline{w}_{ss}$  where  $\underline{w}_{ss}$  is the steady state estimated echo path. In other word, this ERLE can be treated as  $\text{ERLE}_{\text{ST}}$  when system decides the abrupt change is caused by DT and stops adaptation. Hence, Eq.(4.11) can be rewritten as

$$\text{ERLE}_{DT}(N_T) = P(\text{EPC} | \text{DT})\text{ERLE}_{DT} + P(\text{DT} | \text{DT})\text{ERLE}_{ST} \quad (4.12)$$

$$\text{ERLE}_{EPC}(N_T) = P(\text{DT} | \text{EPC})\text{ERLE}_{ST} + P(\text{EPC} | \text{EPC})\text{ERLE}_{ST} \quad (4.13)$$

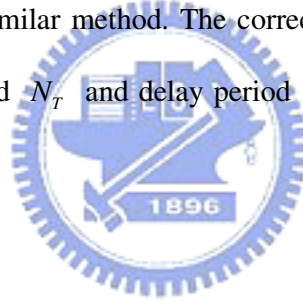
From [17], ERLE under ST (single talk) and DT is given as follows:

$$\text{ERLE}_{ST} = 10 \log_{10} \frac{(2-u)P_y}{uP_z} \quad (4.14)$$

$$\text{ERLE}_{DT} = 10 \log_{10} \frac{(2-u)P_y}{u(P_t + P_z)} \quad (4.15)$$

where  $P_y$ ,  $P_z$  and  $P_t$  are the power of  $y(n)$ ,  $z(n)$  and  $t(n)$ , respectively.

In this Section, we analyzed Liu's scheme under DT situation. The EPC situation can be analyzed using the similar method. The correct decision probability of Liu's scheme is related to threshold  $N_T$  and delay period  $n_d$ . Computer simulations will be shown in Chapter 5.



# Chapter 5

## Computer Simulations

In this Chapter, computer simulations are used to verify the algorithm discussed in Chapter 3 and Chapter 4. First, we define some parameters to measure the convergence time in Section 5.1, and compare the convergence rate of different algorithms in Section 5.2. The performance of the proposed algorithm: ONLMS, its practical approach, simplified ONLMS with tap-variant step-size will be shown in Section 5.3. The issue of tap coefficients error will also be discussed in Section 5.4, and we will focus on the relation between coefficients error variance and step-size for different algorithms.

Nonlinear effect is a practical problem for AEC and we will simulate its effect to our algorithm in Section 5.5. Through Section 5.6-5.8, we give the cost of ERLE and find out its error probability of system decision under abrupt change (DT or EPC) at error signal using different threshold  $N_T$ . We use white Gaussian noise as input signal without any additional statement in this Chapter and speech signals will also be used for verification in our results.



## 5.1 Simulation parameters and room impulse response

The room impulse response  $\underline{w}$ , shown in Figure 5.1.1, is measured and down-sampled from the real environment in our laboratory [16] and an object change in room makes another impulse response shown in Figure 5.1.2. Another impulse response is use to simulate EPC situation. Note that the echo impulse response is measured by Maximum Length Sequence (MLS) [16] method, the sampling rate is 8kHz and down-sampling by a factor  $M = 66$ . Figure 5.1.3 illustrates the speech signal with sampling rate 8 kHz. In our simulations, we use the normalized tap coefficients error (misalignment) factor  $\varepsilon(n)$  defined by the following equation to compare the performance of those algorithms.

$$\varepsilon(n) \triangleq \frac{\|\underline{w} - \hat{\underline{w}}(n)\|}{\|\underline{w}\|} \quad (5.1)$$

The factor  $\varepsilon(n)$  represents system tracking ability of AEC for room impulse response (RIR). The signal to noise ratio is defined as

$$SNR \triangleq 10 \log_{10} \frac{p_y}{p_z} \quad (5.2)$$

where  $p_y$  and  $p_z$  are the average power of the real echo signal and noise.

We are interested in the convergence rate of those conventional algorithms discussed in Chapter 2 and ONLMS algorithm discussed in Chapter 3, so we defined three parameters to measure it as follows:

- (1)  $\varepsilon_{ss}(n)$  : The steady state  $\varepsilon(n)$  that  $|\varepsilon(n) - \varepsilon(n-1)| < 0.001$ .
- (2)  $T_{0.4}$  : The time that misalignment  $\varepsilon(n)$  reaches 0.4.
- (3)  $T_s$  : The time that misalignment  $\varepsilon(n)$  reaches its steady state  $\varepsilon_{ss}(n)$

We also assume both numbers of filter tap and length of RIR equal to 300.

Coefficients of the filter in echo canceller were initialized to zero and the adaptation was disabled for the first 200 samples.

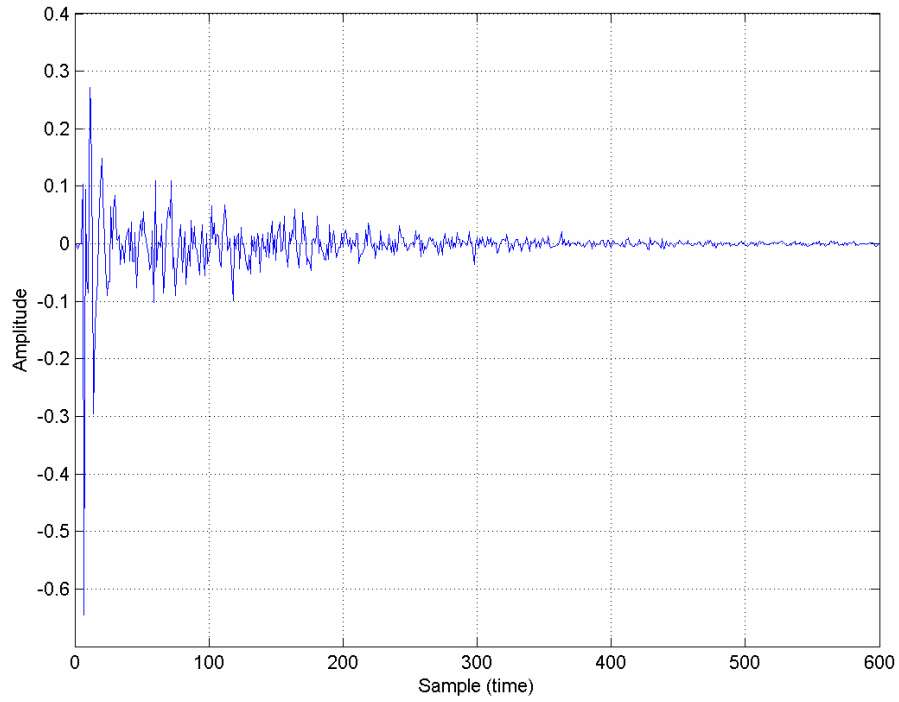


Figure 5.1.1 Room impulse response



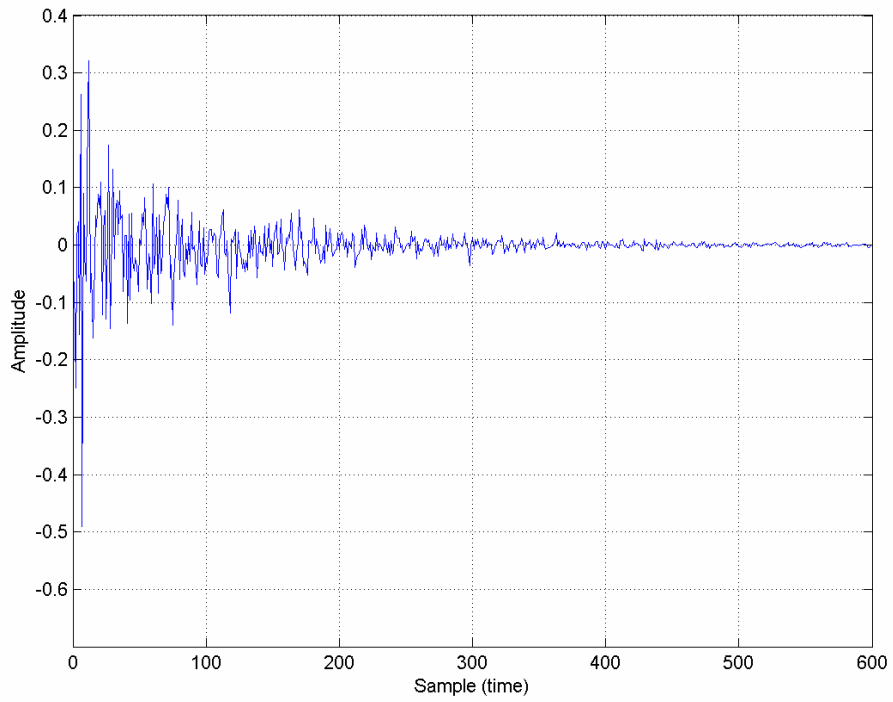


Figure 5.1.2 Room impulse response in the same room

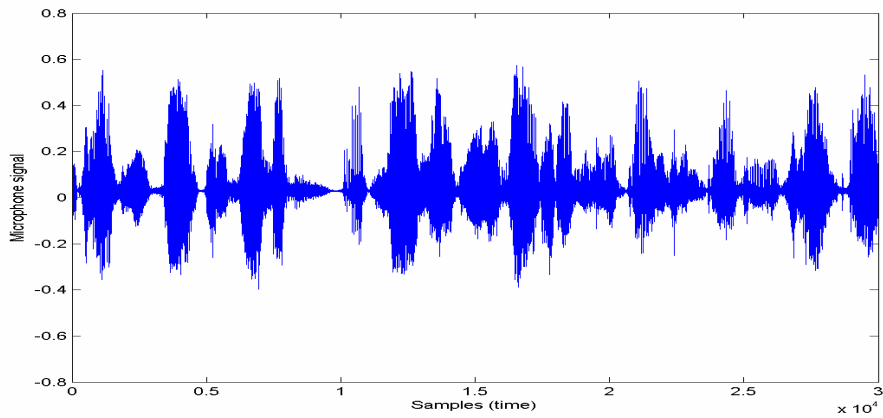


Figure 5.1.3 Speech signal with sampling rate 8 kHz

## 5.2 Performance comparison of convergence rate for different algorithms

There are lots of algorithms using variant step-size perform (i.e., convergence rate) better than using constant step-size such as VSNLMS (time-variant step-size), ESNLMS (tap-variant step-size) and PNLMS (time- & tap-variant step-size) which have been discussed in Chapter 2. In this Section, we compare their convergence rate under different SNR. All algorithms are tuned to achieve the same misalignment ( $\varepsilon_{ss}(n)$ ) for a fair comparison.

Figure 5.2.1 shows the convergence curve of different algorithms. At initial state, NLMS (dash line) has the slowest convergence rate compared to others using variant step-size algorithms. However, PNLMS and ESNLMS converge slower than NLMS after the initial state. We can check Table 5.1 that  $T_{0.4}$  and  $T_s$  also indicate this situation. Two distinct slopes in the convergence curves of PNLMS and ESNLMS are readily apparent in Figure 5.2.1 as discussed in Section 3.6. ESNLMS using tap-variant step-size converges fast than NLMS using constant step-size at first but becomes the slowest at steady state. The VSNLMS using time-variant step-size performs better than NLMS which indeed reduced the tradeoff between misalignment and convergence rate. The proposed ONLMS algorithm using optimum time & tap variant step-size has the fastest convergence rate whether in initial or steady state.

The parameters settings chosen for Figure 5.2.1 and Figure 5.2.2 are as follows:

- $L = 300$ ,  $SNR = 10$  (Figure 5.2.1), 30 dB (Figure 5.2.2)
- NLMS :  $u = 0.35$
- VSNLMS :  $\alpha=0.99$ ,  $\beta=0.04$

- ESNLMS :  $\gamma = 0.015$
- PNLMS :  $\rho = 0.01, \delta = 0.01$
- ONLMS :  $\gamma = 0.015$

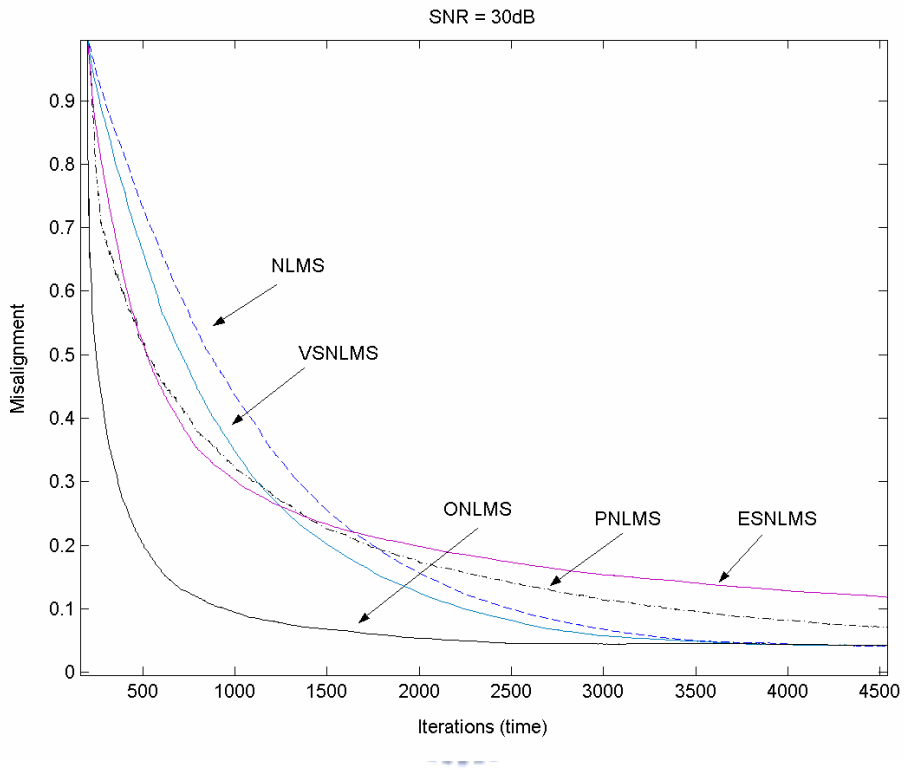


Figure 5.2.1 Convergence curve of different algorithms under SNR = 30dB

algorithms convergence time	NLMS	VSNLMS	ESNLMS	PNLMS	ONLMS
$T_{0.4}$	1089	884	690	755	293
$T_s$	4057	3817	20000	7738	2131

Table 5.1 Performance of convergence rate for different algorithms under SNR = 30dB

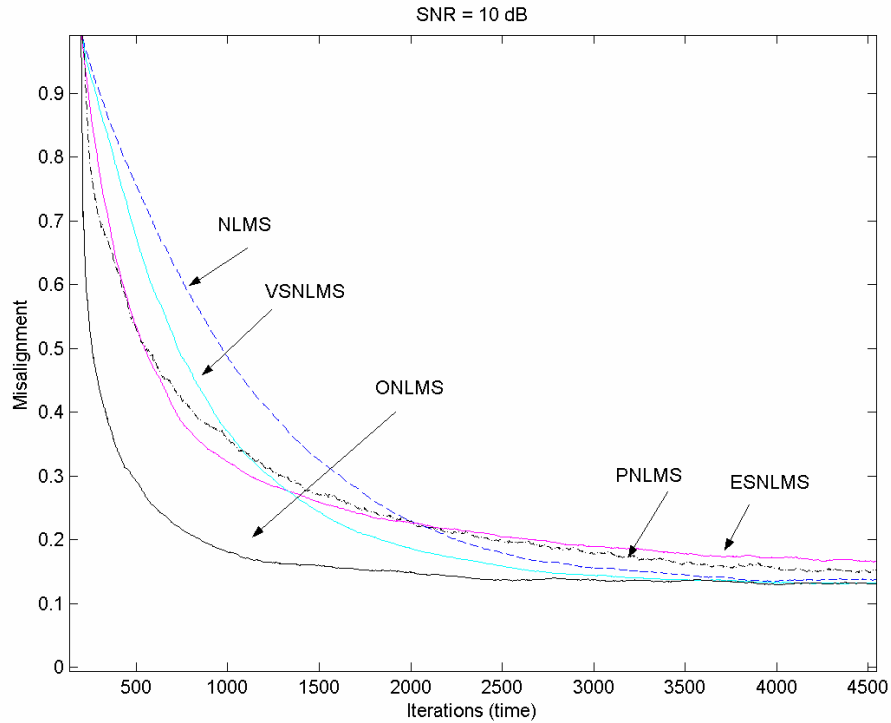


Figure 5.2.2 Convergence curve of different algorithms SNR = 10dB

algorithms convergence time	NLMS	VSNLMS	ESNLMS	PNLMS	ONLMS
$T_{0.4}$	1220	991	706	793	334
$T_s$	4090	2917	20000	7114	1667

Table 5.2 Performance of convergence rate for different algorithms under SNR = 10dB

We can see that algorithms converge fast at initial state and slow at steady state under high SNR, and the opposite situation happened at low SNR. Besides, there is no significant difference between Figure 5.2.1 and Figure 5.2.2 except steady state misalignment which is expected for different SNR [1].

Then we use speech as input signal for verification. Because the fluctuation of

steady state misalignment for speech signal, we redefine  $\varepsilon_{ss}(n)$ : The average of the last 100  $\varepsilon(n)$  samples. The parameters are set the same as Figure 5.2.1

algorithms time & misalignment	NLMS	VSNLMS	ESNLMS	PNLMS	ONLMS
$T_{0.4}$	3586	3580	3522	4077	2021
$\varepsilon_{ss}(n)$	0.078	0.079	0.0942	0.0927	0.0701

Table 5.3 Performance of different algorithms under SNR = 30dB

algorithms time & misalignment	NLMS	VSNLMS	ESNLMS	PNLMS	ONLMS
$T_{0.4}$	3863	3927	3794	5916	3543
$\varepsilon_{ss}(n)$	0.2672	0.2868	0.2443	0.2868	0.2394

Table 5.4 Performance of different algorithms under SNR = 10dB

From Table 5.3 and Table 5.4, ONLMS algorithm indeed has fast convergence rate than other algorithm using speech input.

### 5.3 ONLMS algorithm

We have shown thus far that the use of step-size adjustment improves the convergence rate of the NLMS algorithm. In this Section, we will focus on the proposed algorithms using optimum-time & tap-variant step-size.

The algorithm proposed by Slock using time-variant step-size by Eq.(3.1) which

assumes  $g_0(n) = g_1(n) = \dots = g_{L-1}(n)$ . We drop this assumption to obtain time- & tap-variant step-size, and compare these two algorithms to see the influence of using this assumption.

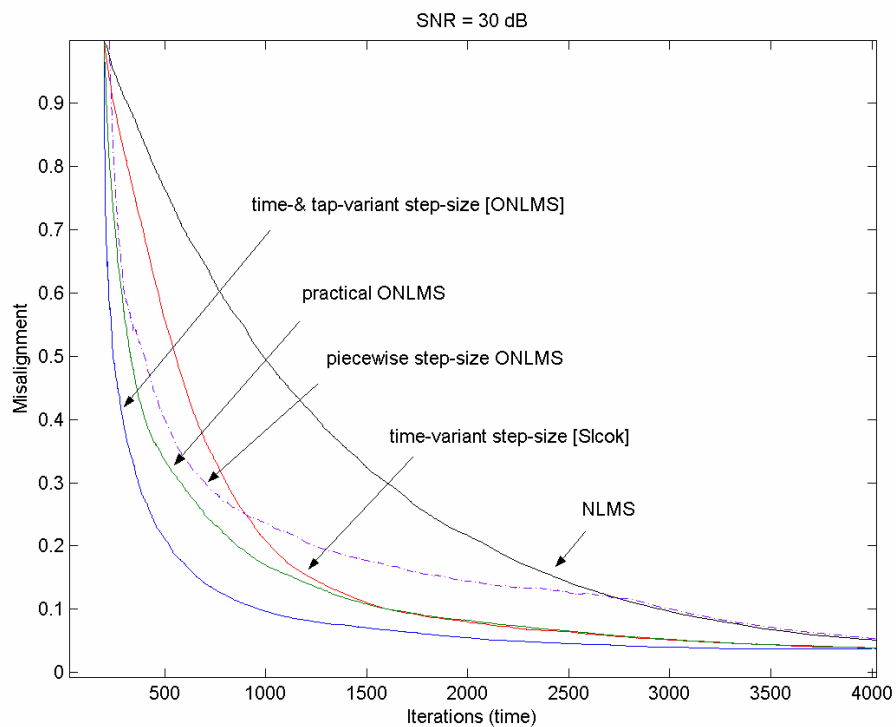


Figure 5.3.1 Comparison of Slock's algorithm and ONLMS under high SNR

step-size adjustment convergence time	NLMS	Slock	Simplified ONLMS	practical ONLMS	ONLMS
$T_{0.4}$	1257	658	502	467	294
$T_s$	5201	3756	3667	3667	2971

Table 5.5 Performance of using different step-size adjustment under SNR = 30dB

Figure 5.3.1 illustrates the convergence rate of ONLMS, practical ONLMS and



Slock algorithm under  $SNR = 30dB$ , and Figure 5.3.2 illustrates  $SNR = 10dB$ . As we discussed in Section 3.6, the  $g_i(n)$  for  $i=0, \dots, L-1$  of Slock's algorithm will not be identical, but it use identical step-size for all taps. This is the reason that Slock's algorithm converges slower than ONLMS shown in Figure 5.3.1 and Figure 5.3.2. The practical ONLMS algorithm which uses exponential decay model to simulate real RIR has similar performance to ONLMS algorithm even in low SNR.

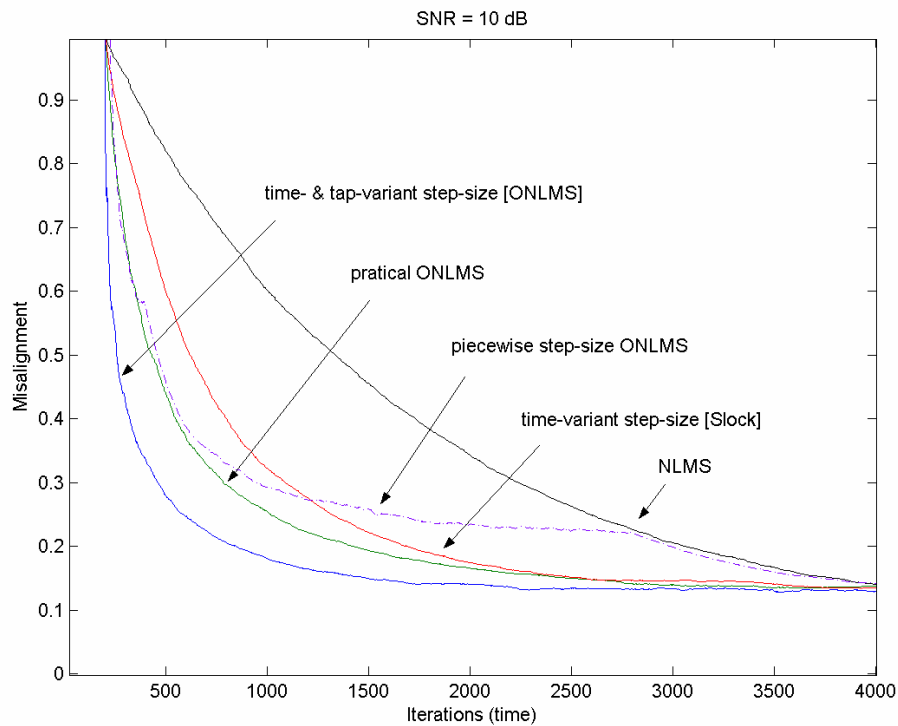


Figure 5.3.2 Comparison of Slock's algorithm and ONLMS under low SNR

step-size adjustment	NLMS	Slock	Simplified ONLMS	practical ONLMS	ONLMS
convergence time					
$T_{0.4}$	1735	779	576	526	323
$T_s$	4995	3666	4995	2905	1968

Table 5.6 Performance of using different step-size adjustment under  $SNR = 10dB$

From the simulation above, ONLMS indeed increases the converge rate. But it also increases the computational load. We use the simplified version of ONLMS which uses piecewise time-variant and tap-variant step-size discussed in Section 3.5.

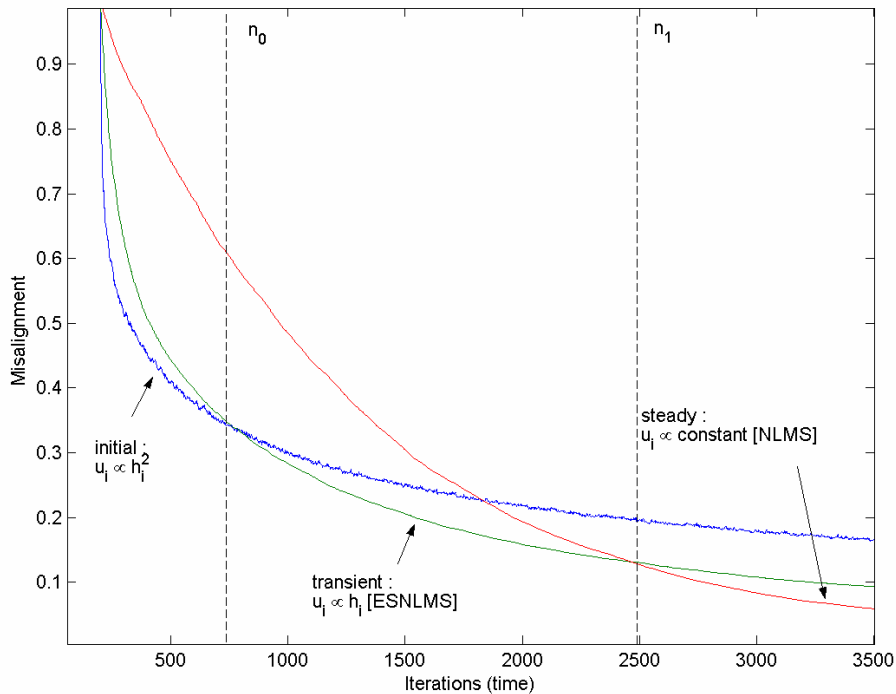


Figure 5.3.3 Convergence curves using piecewise time-variant and tap-variant step-size

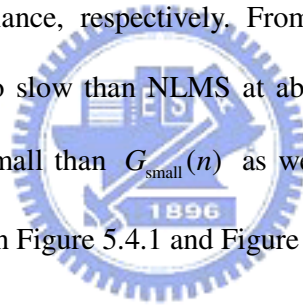
Figure 5.3.3 illustrates three different choices of tap-variant step-size which is based on the discussion in Section 3.6. The left and right dash line indicate  $n_0$  and  $n_1$ , respectively. Three different curves have distinct slopes at different adaptation state. We choose the fastest part of them, respectively, which results in piecewise time-variant and tap-variant step-size. The combined curve performs better than NLMS but not as good as the original ONLMS.

## 5.4 Tap coefficient error variance

The convergence gain energy is composed of tap coefficient error variance  $g_i(n)$  in Eq.(3.25). The variation of  $g_i(n)$  decides the convergence rate that we have discussed in Section 3.6. In this Section, we use computer simulation to verify the discussion.

Since the RIR is exponentially decay, we define first 1/3 echo path of RIR as “large taps” and last 2/3 echo path as “small taps”. The sum of large taps’  $g_i(n)$  represented as  $G_{\text{large}}(n)$  and sum of small taps’ represented as  $G_{\text{small}}(n)$ .

Figure 5.4.1 and Figure 5.4.2 illustrate ESNLMS and PNLMS algorithm’s sums of tap coefficient error variance, respectively. From Figure 5.2.1, ESNLMS and PNLMS convergence start to slow than NLMS at about  $n = 2000$ . This is because  $G_{\text{large}}(n)$  starts to become small than  $G_{\text{small}}(n)$  as we discussed in Section 3.6. We can verify this phenomenon in Figure 5.4.1 and Figure 5.4.2.



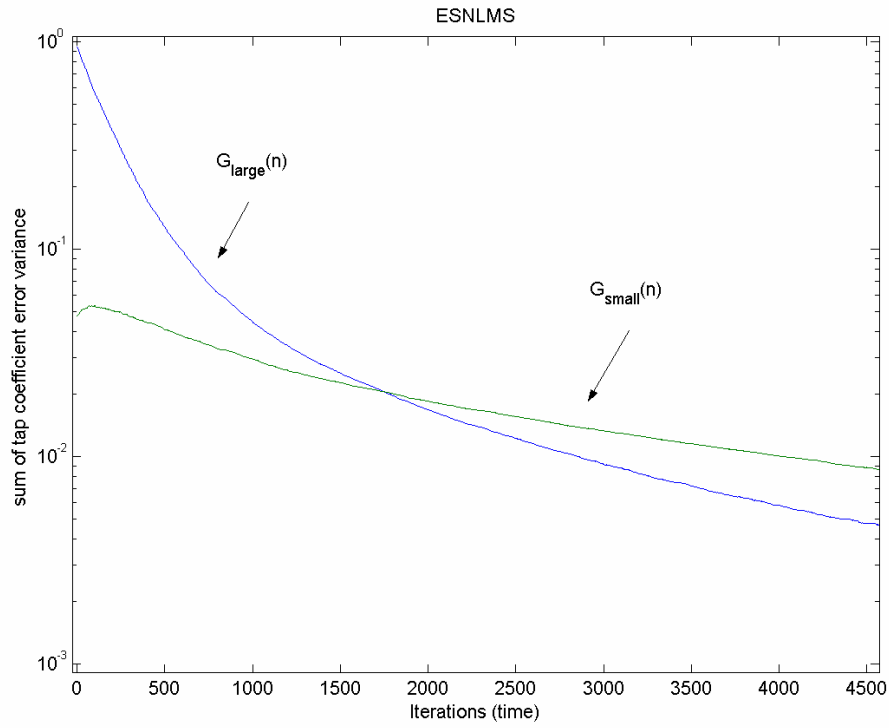


Figure 5.4.1  $G_{\text{large}}(n)$  and  $G_{\text{small}}(n)$  for ESNLMS

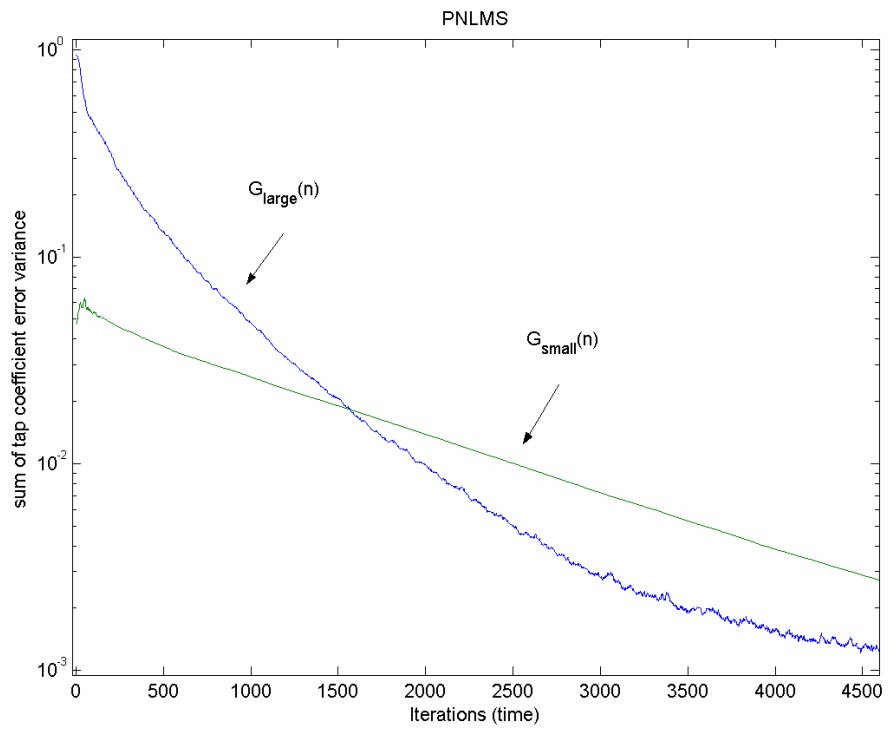


Figure 5.4.2  $G_{\text{large}}(n)$  and  $G_{\text{small}}(n)$  for PNLMS

Figure 5.4.3 illustrates  $g_i(n)$  for ONLMS algorithm  $g_i(n)$  with  $i=10, 70, 170, 270$  representing of large and small taps. After the initial state,  $g_{10}(n) = g_{70}(n) = g_{170}(n) = g_{270}(n), \forall n$ , which means these taps use same step-size. Hence, convergence is still fast after the initial state.

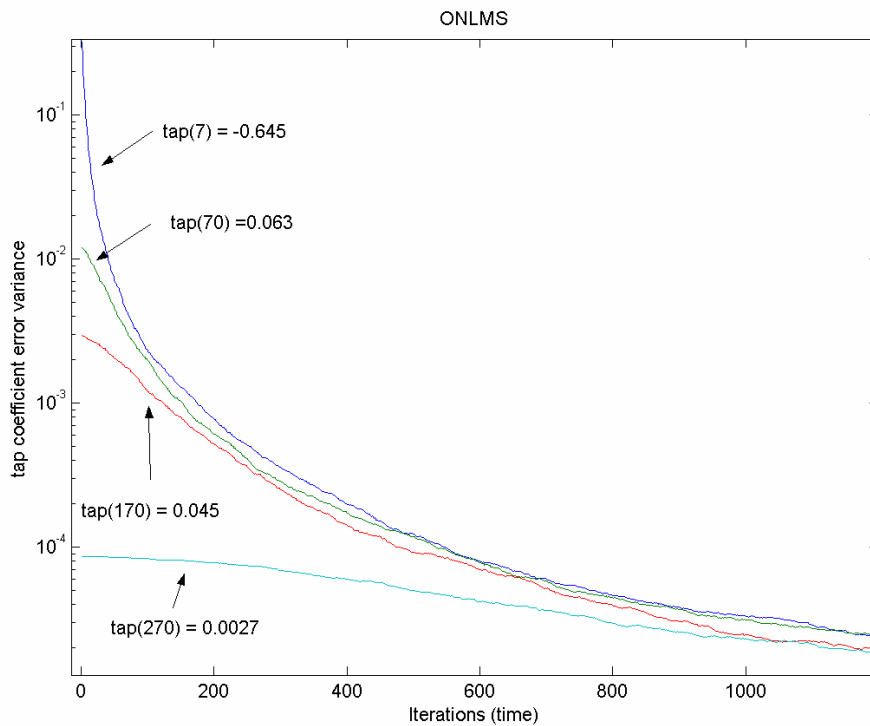


Figure 5.4.3  $g_i(n)$  for ONLMS algorithm where  $i = 10, 70, 170, 270$

## 5.5 Nonlinear distortion

AEC in microphones or video conferencing systems rely on the assumption of a linear echo path. However, low-cost audio equipment or constrains of communication systems cause nonlinear distortion, which limit the ERLE achievable by linear adaptation schemes. This means that an annoying nonlinearly distorted echo (“nonlinear echo”) will be transmitted back to the far-end subscriber. In this Section,

we use saturation as nonlinear distortion on loudspeaker signals and find out the effect upon ONLMS algorithm. White Gaussian noise with zero means and unity variance is used as far-end signals.

A commonly used function for modeling saturation is a sigmoid function (Figure 5.5.1 for  $\alpha = 1, 2$  and  $5$ ) defined as follow:

$$f(x) = \left( \frac{2}{1 - \exp(-\alpha x)} - 1 \right) \beta \quad (5.3)$$

Figure 5.5.2 exhibits how nonlinear distortion effect ERLE under different SNR. Line with circle mark represents no distortion on loudspeaker signals. Compare to this line, others has lower ERLE due to nonlinear effect. The ERLE descend with increasing  $\alpha$ . Hence, an annoying nonlinearly distorted echo transmitted back to the far-end user.

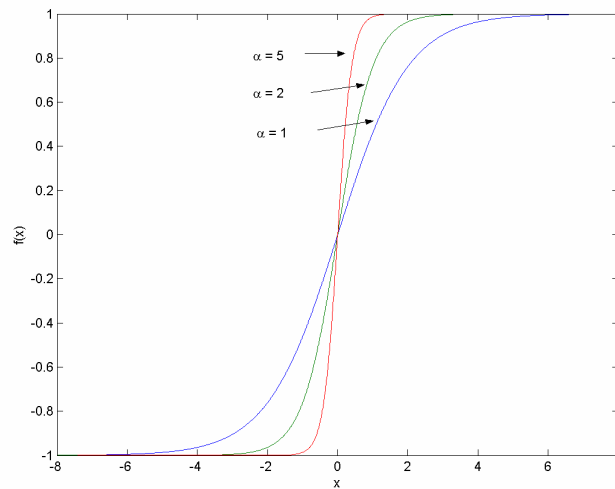


Figure 5.5.1 Sigmoid function

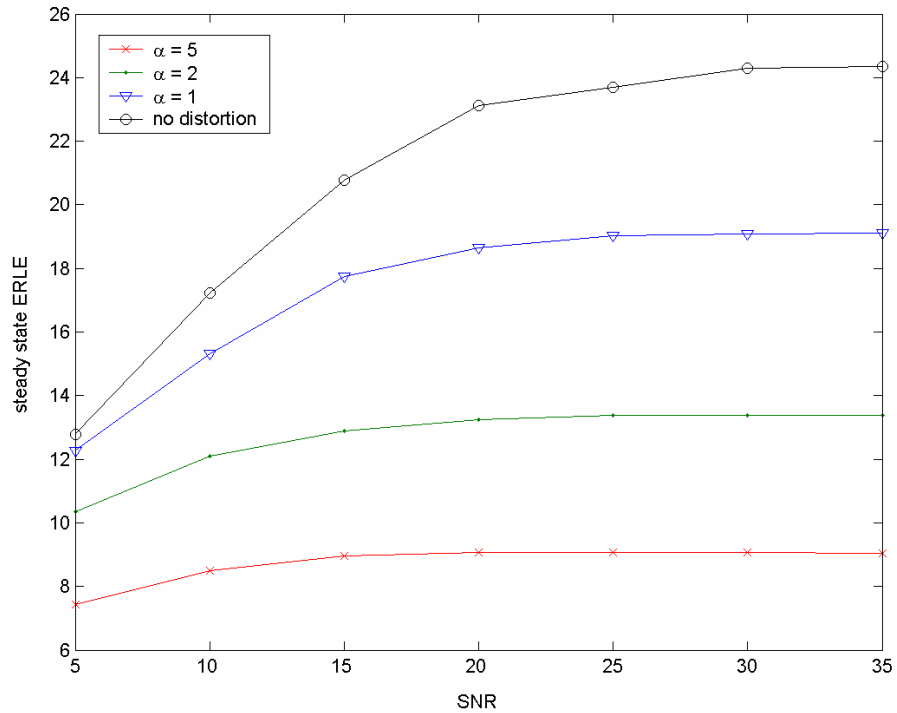


Figure 5.5.2 Different  $\alpha$  effect on ERLE under different SNR.



## 5.6 The correct probability of system decision using different thresholds

We have discussed the architecture to distinguish DT and EPC in Section 4.2 and analyzed it in Section 4.3. We also defined “needed time (simple) for correct decision” as random variable  $K$ . The correct probability decision is given in Eq.(4.10).

Figure 5.6.1 illustrates p.d.f. of  $K$  with  $N_T = 50, 200$  under DT situation. The  $K$  is an exponential distribution random variable which factor  $\lambda$  is inverse proportionate to  $N_T$ . Right side of the dash line indicates the probability of error

decision. The curve of  $N_T = 50$  all lie on left side of the dash line which means using smaller  $N_T$  enhances system decision ability under DT situation. Larger  $N_T$  makes p.d.f. curve more flat (i.e., smaller  $\lambda$ ) and shift toward right side which increase the error probability.

Figure 5.6.2 illustrates the correct decision probability using different  $N_T$  under DT and EPC, respectively. In Figure 5.6.2, we set  $n_d = 300$ ,  $n_s = 40$ ,  $SNR = 30dB$ . Circle mark line represents decision correct probability when DT happens, and asterisk mark represents EPC happens. As shown in Figure 5.6.2, we have high correct decision probability when using low  $N_T$  under DT which is the same conclusion as Figure 5.6.1.

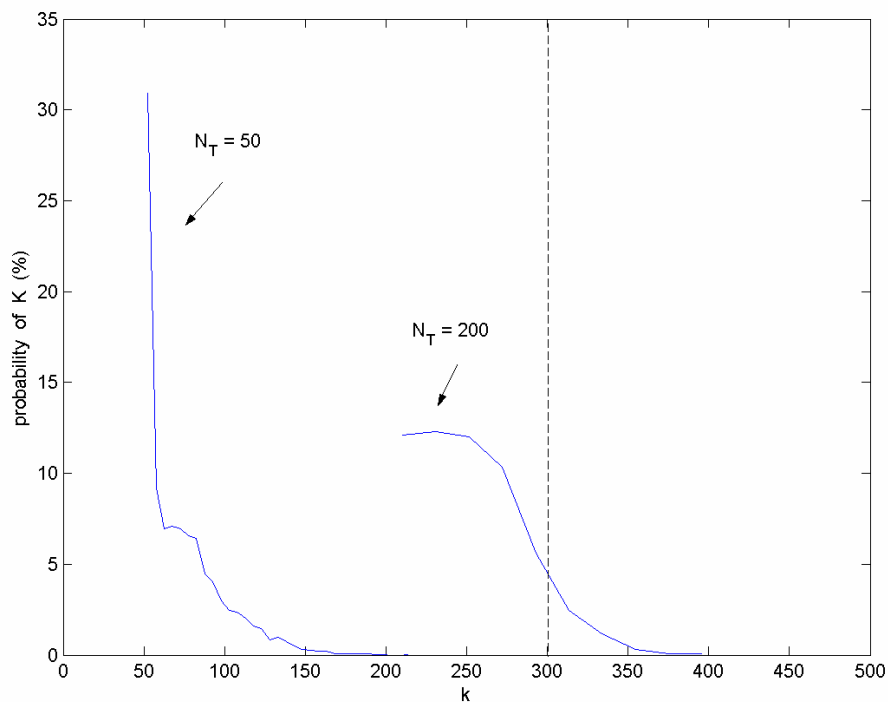


Figure 5.6.1 Probability density function of  $k$  with  $N_T = 50$  and 200 under DT.



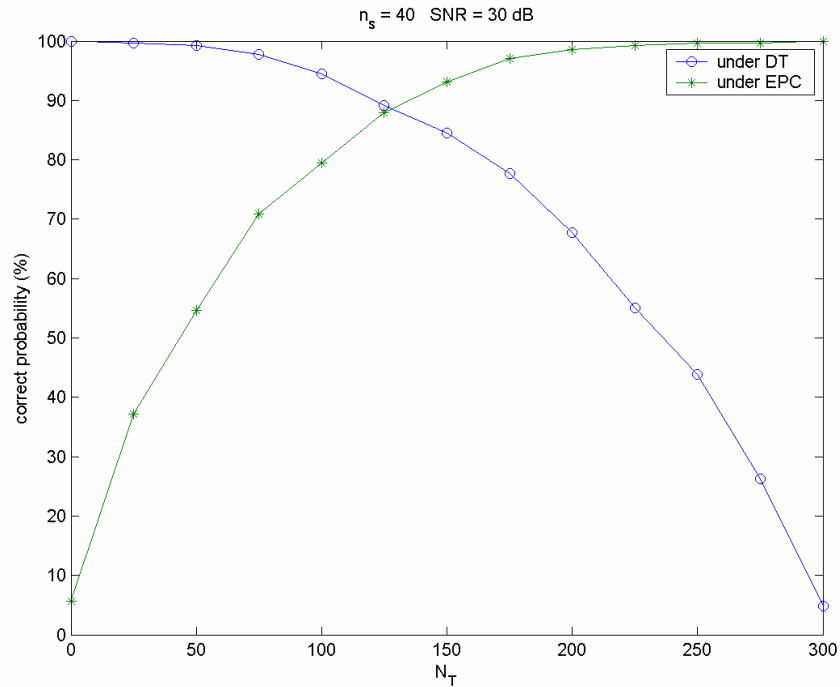


Figure 5.6.2 The correct probability of system decision using different  $N_T$ .

The cross point between these two lines means system under DT and EPC case will have the same correct decision probability, and  $N_T$  is close to  $n_s/2$ . The cross point will be smaller with low SNR.

## 5.7 Average ERLE using different threshold

The average ERLE is defined in Eq.(4.11). Small  $N_T$  increases the error probability of system decision under EPC which results in smaller ERLE. For DT, large  $N_T$  increases the error probability which results in smaller ERLE as shown in Figure 5.7.1.

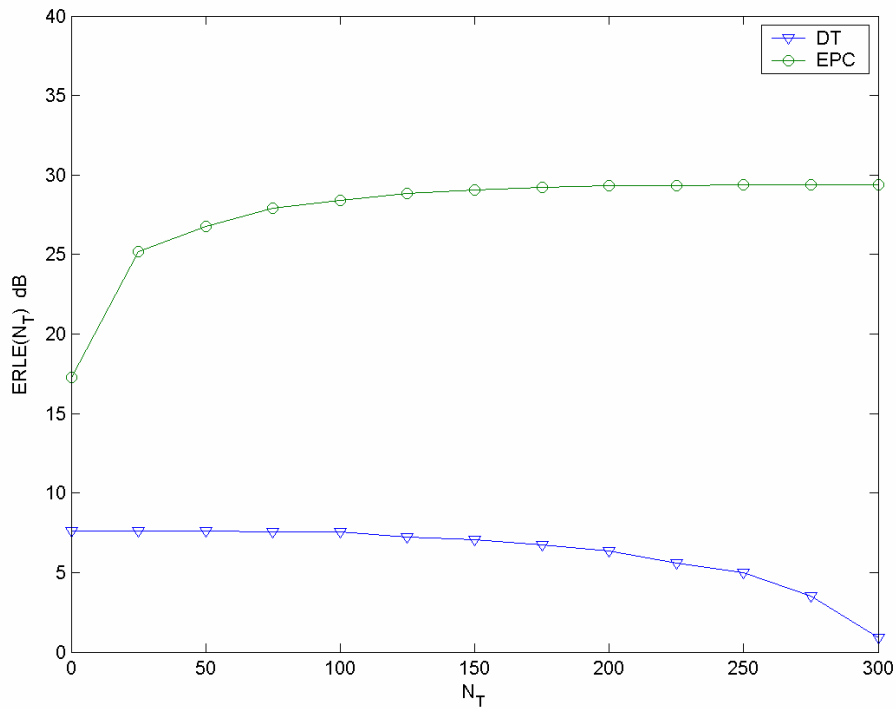


Figure 5.7.1 The average ERLE using different  $N_T$



## 5.8 AEC with DTD

We use the ONLMS algorithm for AEC proposed in Chapter 3 combined with the DTD discussed in Chapter 4 for implementation a complete system.

Figure 5.8.1-5.8.3 illustrate misalignment under DT and EPC using WGN as input signal with  $N_T = 50, 150, 250$  respectively. The parameters settings chosen for the Section are as follows:

- $C = 5$
- $n_d = 300$
- $N_T = 50, 150, 250$
- SNR = 30 dB
- DT :  $n = 30000 \sim 34000$
- EPC :  $n = 70000$

Figure 5.8.1 shows that system make correct decision under DT but error decision under EPC with  $N_T = 50$  (too small). Under EPC, small  $N_T$  makes system decide that abrupt change is caused by DT and stop the adaptation.

When using  $N_T = 150$  ( $n_d/2$ ), system can have correct decision under both DT and EPC shown in Figure 5.8.2. When using  $N_T = 250$  (too large), system makes correct decision under EPC but error decision under DT shown in Figure 5.8.3. The filter coefficients diverge at DT period due to system make error decision.

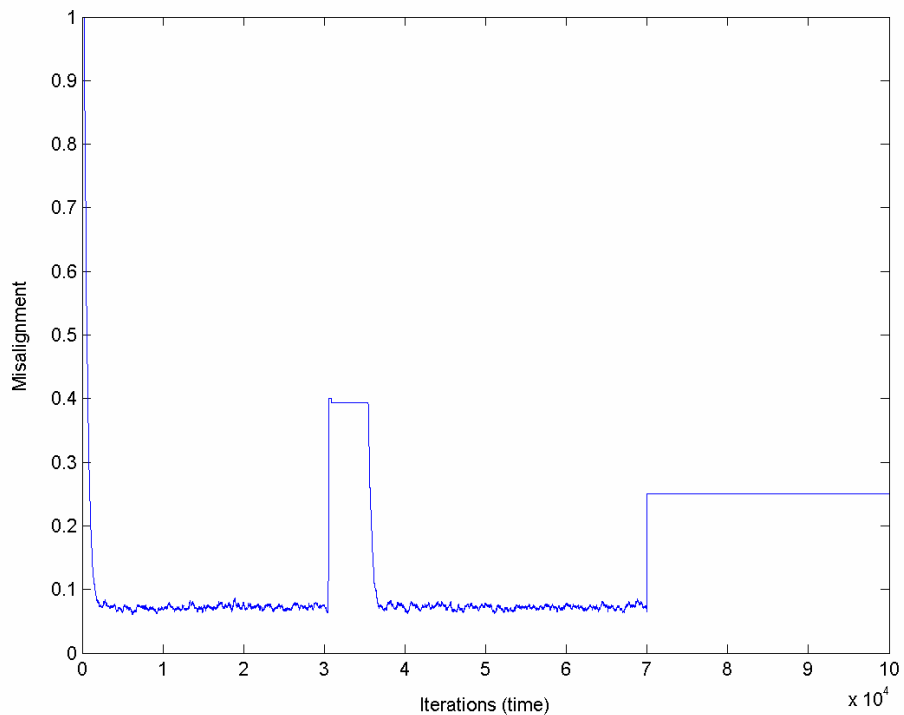


Figure 5.8.1 Misalignment under DT and EPC with  $N_T = 50$

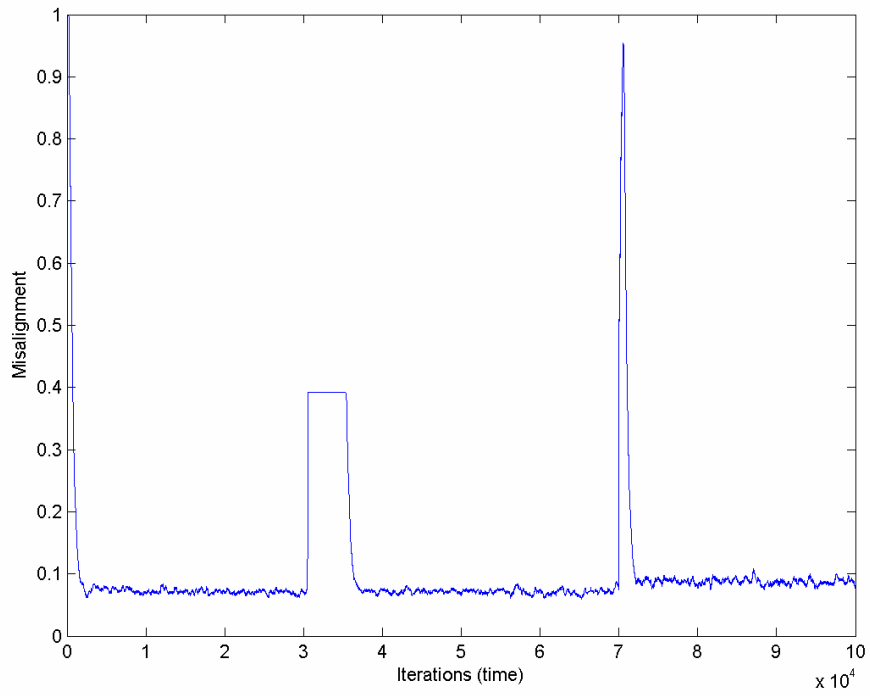


Figure 5.8.2 Misalignment under DT and EPC with  $N_T = 150$

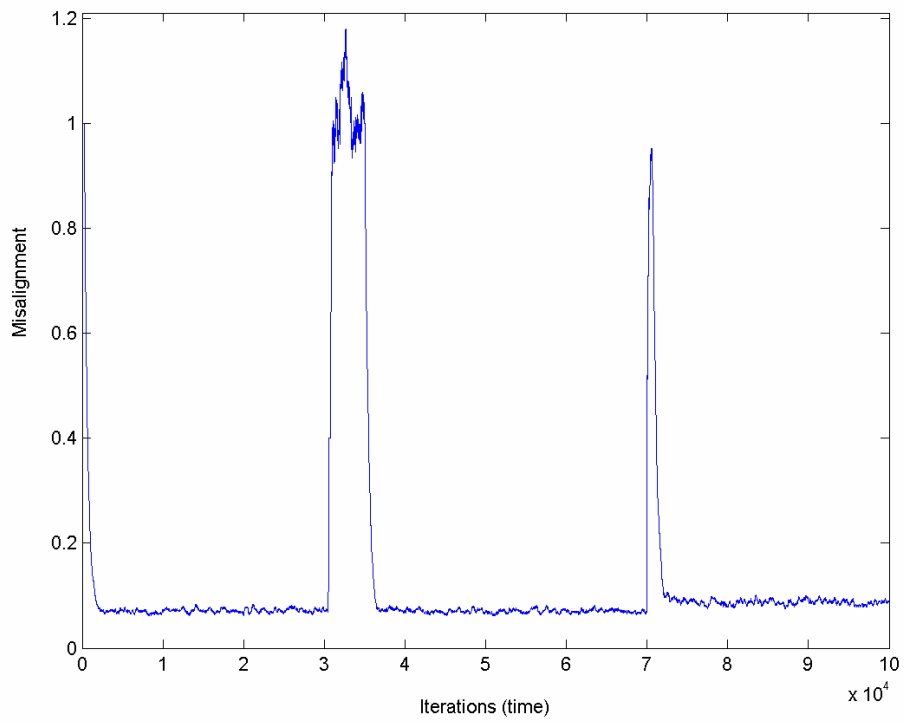


Figure 5.8.3 Misalignment under DT and EPC with  $N_T = 250$

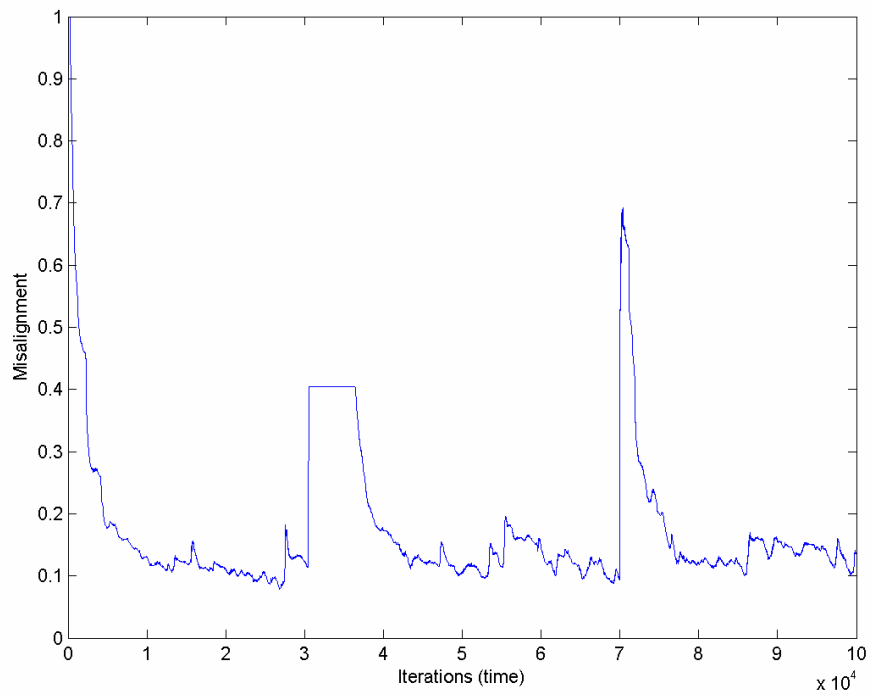
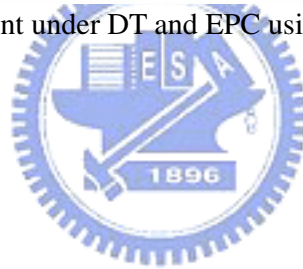
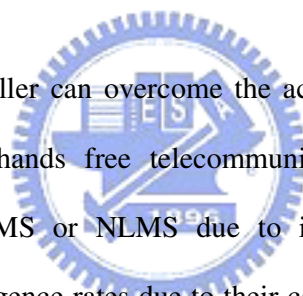


Figure 5.8.4 Misalignment under DT and EPC using speech with  $N_T = 150$



# Chapter 6

## Conclusions



An acoustic echo canceller can overcome the acoustic feedback that interferes with teleconferencing and hands free telecommunication. The most often used algorithms for AEC are LMS or NLMS due to its simplicity. However, these algorithms have slow convergence rates due to their constant step-size. Instead, other algorithms ([2], [3], [6]) use variant step-size to increase the convergence rate.

In Chapter 2 and Chapter 3, we introduced these algorithms and derived ONLMS using time- & tap-variant step-size that provides the minimum MSE at each iteration step which has the fastest convergence rate. The RIR exponential decay model is used to implement ONLMS algorithm in Section 3.4. In order to reduce its computation load, we proposed a simplified piecewise step-size ONLMS algorithm in Section 3.5.

We analyze a simple DTD scheme which can distinguish DT or EPC when an abrupt change in Chapter 4 and use a probability model to explain the relation between threshold factor  $N_r$  and correct decision probability.

Computer simulations shown in Chapter 5 demonstrate that (i) ONLMS

converges fastest than other algorithms whether in initial or steady state, (ii) practical and simplified ONLMS, (iii) sum of tap coefficient error variance for large and small taps of ESNLMS and PNLMS algorithms, (iv) nonlinear effects on the ERLE of ONLMS, (v) average ERLE with different decision thresholds  $N_T$ .

The future work can be: (i) compare ONLMS with another class of algorithm such as RLS, (ii) the selection of parameters  $n_0$  and  $n_1$  for the simplified piecewise step-size ONLMS algorithm, (iii) convergence condition of ONLMS algorithm, (iv) analytical formula for correct decision probability.



# Appendix

## Gaussian Assumption

In this appendix, we are trying to simplify the third term in Eq.(3.9) which is showing below using Gaussian assumption.

$$\mathbf{B} = E[\underline{x}(n)\underline{x}^T(n)\mathbf{A}\underline{x}(n)\underline{x}^T(n)] \quad (1)$$

where  $\mathbf{A}$  represents the symmetric matrix  $\underline{v}(n)\underline{v}^T(n)$ .

The Gaussian assumption: If  $z_1, z_2, z_3, z_4$  are real zero-mean, Gaussian random variables then

$$E[z_1 z_2 z_3 z_4] = E[z_1 z_2]E[z_3 z_4] + E[z_1 z_3]E[z_2 z_4] + E[z_1 z_4]E[z_2 z_3] \quad (2)$$

Using subscripts to denote the components of the vectors  $\underline{x}(n)$  and  $\underline{v}(n)$ , where the dependency upon  $(n)$  is momentarily suppressed, we compute the  $kl$  th term in

Eq.(1):

$$\begin{aligned} b_{kl} &= \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} E(x_k x_p a_{pq} x_q x_l) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} E[a_{pq}] [E(x_k x_p)E(x_q x_l) \\ &\quad + E(x_k x_q)E(x_p x_l) + E(x_k x_l)E(x_p x_q)] \\ &= E[a_{kl}] \sigma_x^4 + E[a_{lk}] \sigma_x^4 + \delta[k-l] \sum_{p=0}^{L-1} E(a_{pp}) \sigma_x^4 \end{aligned} \quad (3)$$



where  $\sigma_x^2 = E[x_k^2]$ . Note that  $x_k$  and  $a_{pq}$  are assumed to be independent, and

$E(x_p x_q)$  is zero for  $p \neq q$  since  $\underline{x}(n)$  is WGN. We can obtain the matrix form of

Eq.(3) :

$$\mathbf{B} = \sigma_x^4 [2\mathbf{R}_v(n) + \text{trace}(\mathbf{R}_v(n)) \cdot \mathbf{I}] \quad (4)$$



# Bibliography

- [1] Simon Haykin, *Adaptive Filter Theory*, four edition, Prentice-Hall, 2002.
- [2] R. H. Kwong and E. W. Johnston, “A variable step size LMS algorithm,” *IEEE Trans. Signal Processing*, vol. 40, pp. 1633–1642, July 1992.
- [3] J. F. Liu, “Robust line echo cancellation in complicated phone call environment,” *IEEE SMC 2001 Conference Proceedings, IEEE International Conference*, vol. 1, pp. 310-315, 2001
- [4] D. L. Duttweiler, “Proportionate normalized least mean square adaptation in echo cancellers,” *IEEE Trans. Speech and Audio Processing*, vol. 8, pp. 508-518, September 2000.
- [5] H. H. Shih, “Implementation of adaptive echo cancellation with double talk detection on DSP,” *NCTU Master Thesis*, June 1997.
- [6] S. Makino, Y. Kaneda and N. Koizumi, “Exponentially weighted stepsize NLMS adaptive filter based on the statistics of a room impulse response,” *IEEE Trans. Speech and Audio Processing*, vol. 1, NO. 1, p.p. 101-108, January 1993.



- [7] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. Signal Processing*, vol. 41, NO. 9, p.p. 2811 – 2825, September 1993.
- [8] S. L. Gay and J. Benesty, "Acoustic signal processing for telecommunication," Kluwer Academic, 2000.
- [9] N. J. Bershad, "On the optimum gain parameter in LMS adaptation," *IEEE Trans. Speech and Audio Processing*, vol. ASSP-35, NO. 7, p.p 1065-1068, July 1987.
- [10] Chen, R. Y. and Wang, C. L., "On the optimum step size for the adaptive sign and LMS algorithms," *IEEE Trans. Circuits Syst.*, vol. 7, NO. 6, p.p. 836-840, June 1990.
- [11] P. C. Yip and D. M. Etter, "An adaptive multiple echo canceller for slowly time-varying echo paths," *IEEE Trans. On Communications*, vol. 38, NO. 10, p.p. 1693-1698, October 1990.
- [12] M.R. Asharif, A. Shimabukuro, T. Hayashi, K. Yamashita, "Expanded CLMS algorithm for double talk echo canceling," *IEEE SMC '99 Conference Proceedings, 1999, IEEE International Conference*, vol. 1, p.p. 998-1001, 1999.
- [13] H. Ye and X. Wu, "A new double talk detection algorithm based on the orthogonality theorem," *IEEE Trans. On Communications*, vol. 39, p.p. 1542-1545, issue 11, November 1991.

- [14] J. C. Liu, "A novel adaptation scheme in the NLMS algorithm for echo cancellation," *IEEE Trans. Signal Processing Letters*, vol. 8, NO. 1, January 2001.
- [15] D. L. Duttweiler, "A twelve-channel digital echo canceller," *IEEE Trans. On Communications*, vol. COM-26, NO. 5, p.p. 647-653, May 1997.
- [16] J. C. Jeng, "Application of the FIR filter to acoustic echo cancellation," *NCTU Master Thesis*, June 1995.
- [17] D. S. Yeh, "Study on adaptive acoustic echo cancellation," *NCTU Master Thesis*, June 1995.
- [18] K. Mark and B. Karlheinz, "Applications of digital signal processing to audio and acoustics," Kluwer Academic 1998.

