國 立 交 通 大 學

電信工程學系

碩 士 論 文

兩段式非線性回音消除之收斂性分析與最佳步伐調適

Two-Staged Nonlinear Acoustic Echo Cancellation: Convergence Analysis and Step-size Control

研 究 生： 李克強

指導教授： 謝世福 博士

中 華 民 國 九十六 年 八 月

兩段式非線性回音消除之收斂性分析與最佳步伐調適

學生：李克強　　指導教授：謝世福

國立交通大學電信工程學系碩士班

摘要

免持聽筒或是視訊會議系統常驅動功率放大器和喇叭操作在飽和非線性區，導致一般使用的線性回音消除器性能降低。在本篇論文中，我們將對非線性回音消除建構出一個無記憶式片段線性(PWL)處理器串聯一個線性濾波器的模型，它能有效降低運算量。另外，爲了克服在串聯式系統的發散問題，我們採取兩段式調適的方式，首先只有線性濾波器開始更新，接著 PWL 和線性濾波器的係數同時進行調適。更進一步，推導出它的收斂分析及穩定條件，並在電腦模擬得到驗證。最後，我們發展出 LMS 演算法在不同時間、不同時間及不同閥(tap)的最佳步伐調適來加強收斂速度，同時提供他們實用的實現方式和電腦模擬。

Two-Staged Nonlinear Acoustic Echo Cancellation: Convergence Analysis and Step-size Control

Student : K. C. Li **Advisor : S. F. Hsieh**

Department of Communication Engineering

National Chiao Tung University

Abstract

Hand-free phone or teleconferencing system drives the power-amplification and loudspeaker commonly into saturated nonlinear region, leading to that the performance of conventional acoustic echo cancellation (AEC) reduced. In this thesis, we will build a cascade model which consists of a memoryless piece-wise linear (PWL) processor and a linear filter for AEC. It is beneficial to reduce the computation. Besides, in order to overcome the divergence problem in a cascade system, we will adopt the two-stage adaptation that starts with a linear filter, and then joint adaptation of PWL and linear coefficients follows. Further, the convergence analysis and stability criterion will be derived and the computer simulation will justify our analysis. Finally, LMS algorithms with optimum time-variant and time-&tap- variant step-sizes are developed to improve convergence rate. Their practical implementations and computer simulations are also provided.

Acknowledgments

In the process of writing this thesis, I received a lot of assistance and encouragement from many persons. First of all, I would like to express my profound gratitude to my thesis advisor Dr. S.F. Hsieh. Throughout the composition of this thesis, Dr. Hsieh provided me with many enlightening viewpoints and insightful suggestions. My special thanks go to all my lab-mates for their inspiration and help. Finally, Last but not least, I owe my deep appreciation to my family and my girlfriend for their endless support and encouragement.

Contents

中文摘要……i

English Abstract……ii

Acknowledgement……iii

Contents……iv

List of Figures……vii

List of Tables……xii

1 Introduction……1

2 Adaptive Nonlinear AEC……6

2.1 Adaptive nonlinear LMS AEC using PWL structure……6

2.2 Computation of adaptive nonlinear LMS AEC……11

2.3 Partial update of adaptive nonlinear LMS AEC……13

2.3.1 Periodic partial update LMS algorithm……14

2.3.2 Sequential partial update LMS algorithm……14

2.3.3 Stochastic partial update LMS algorithm……15

2.3.4 Variant-period partial update LMS algorithm……16

2.3.5 Located partial update LMS algorithm……17

2.4 Summary……18

3 Two-staged adaptation and its convergence analysis……19

3.1 Two-staged adaptation……19

3.2 Convergence and stability analysis of linear adaptation……22

3.2.1 Mean bias of linear coefficient weight error……22

3.2.2 Second moment of linear coefficient weight error……24

3.3 Convergence analysis of joint adaptation of linear and PWL coefficients....29

3.4 Summary……..……..32

4 Step-size control for Nonlinear AEC……..……..…..33

4.1 Derivation of optimum time- variant step-size LMS (OTLMS) algorithm…34

4.2 Practical implementations of OTLMS algorithm……...36

4.3 Derivation of optimum time-&tap-variant step-size LMS (OTTLMS) algorithm……39

4.4 Practical implementation of OTTLMS algorithm……..43

4.5 Optimal step-size of NLMS algorithm……...45

4.6 Summary……46

5 Computer Simulations……47

5.1 Simulation parameters and system performance measures……47

5.2 Nonlinear AEC based on LMS algorithm……..51

5.2.1 Performance comparison of PWL, polynomial and linear AECs……51

5.2.2 Comparison of partial update LMS algorithms…….……..56

5.2.3 Partition number of PWL processor……59

5.3 Two-staged adaptation and its convergence analysis…….61

5.3.1 Convergence and stability analysis of the first stage…….....61

5.3.2 Convergence analysis of the second stage……..71

5.3.3 Switching point of two-staged adaptation……..….75

5.3.4 Experiments of the two-staged adaptation……..77

5.4 Control of step-size……81

5.4.1 OTLMS algorithm……...……81

5.4.2 Practical implements of OTLMS algorithm……..……..83

5.4.3 OTTLMS algorithm……90

5.4.4 Practical implement of OTTLMS algorithm……...……91

5.4.5 OTLMS and OTTLMS algorithms in the two-staged adaptation……96

5.4.6 Experiments on the step-size controls……98

6 Conclusion……104

Appendix……110

Bibliography……112

List of Figures

1.1 Diagram of hands-free telephone system……...……..1

1.2 Nonlinear acoustic echo cancellation system...……...…….2

2.1 Nonlinear acoustic echo canceller based on piecewise linear structure…..…7

2.2 (a) A CPWL curve with three segments (b)~(d) Associated canonical curves……..……...……..…….8

2.3 The block diagram of a PWL processor…….……...……..……..9

3.1 Cascade model of system (right hand part) and mirror adaptive system (left hand part)……...…….……20

3.2 Performance comparison of the two update schemes with two different step-sizes……..……21

4.1 True nonlinear I/O mapping curve (solid), CPWL function (dot)…….... 38

4.2 RIR decay envelop……...…….…….…….44

5.1.1 Typical nonlinear I/O mapping curve using a raised-cosine function.....50

5.1.2 Room impulse response……..…….……50

5.2.1 Nonlinear I/O mapping curve of PWL and polynomial models……..52

5.2.2 Modeling errors of PWL and polynomial models in fitting to a raised cosine function…….…….……...53

5.2.3 Modeling errors of PWL and orthogonal polynomial models in fitting a raised-cosine function……...……53

5.2.4 ERLE comparison……55

5.2.5 Linear coefficients misalignments……...55

5.2.6 Nonlinear curve misalignments…….56

5.2.7 Saturated curve for nonlinear I/O mapping……..……57

5.2.8 ERLE of partial update schemes for a raised-cosine mapping…….58

5.2.9 ERLE of partial update schemes for a saturated mapping……...58

5.2.10 Raised cosine curve and the corresponding PWL curves with two partition number: 2(b) and 4(c)…….……..59

5.2.11 ERLE for raised-cosine mapping using various partition numbers…….60

5.2.12 ERLE under saturated curve using various partition numbers…….60

5.3.1 Residual error power during the first stage with μ_h=0.01 …….62

5.3.2 Linear coefficients misalignment during the first stage with μ_h=0.01 ……...62

5.3.3 Diverged residual error power during the first stage with over-sized μ_h=0.013 ……..……63

5.3.4 Diverged linear coefficients misalignment during the first stage with over-sized μ_h=0.013 ……63

5.3.5 Linear coefficients misalignment during the first stage under different SNRs…….65

5.3.6 Residual error power during the first stage under different SNRs……66

5.3.7 Linear coefficient misalignment during the first stage with different step-sizes……..66

5.3.8 Residual error power during the first stage under different step-sizes……67

5.3.9 Linear coefficients misalignment during the first stage with different powers of the output of PWL processor……67

5.3.10 Residual error power during the first stage with different powers of the output of PWL processor…….68

5.3.11 Linear coefficient misalignment during the first stage using different γ_d …..68

5.3.12 Residual error power during the first stage using different γ_d ……69

5.3.13 Different Pdfs of the PWL processor output……69

5.3.14 Linear coefficients misalignment during the first stage using different Pdfs...70

5.3.15 Residual error power during the first stage using different Pdfs……70

5.3.16 Misalignment of combined coefficient weight error during the second stage.72

5.3.17 Residual error power during the second stage……...73

5.3.18 Misalignment of combined coefficient weight error under skipping during the 2^{nd} stage……73

5.3.19 Residual error power under skipping during the second stage……74

5.3.20 Misalignment of combined coefficient weight error under decoupling during the 2^{nd} stage……74

5.3.21 Residual error power under decoupling during 2^{nd} stage……75

5.3.22 Misalignment of combined coefficient weight error of the two-staged adaptation……76

5.3.23 Residual error power of the two-staged adaptation……...76

5.3.24 Residual error power with different switching points……77

5.3.25 ERLE with a two level speech for linear AEC (dash line), nonlinear AEC based on PWL structure (solid line) and Polynomial one (dot line)……78

5.3.26 ERLE with the speech of a woman with tone in mandarin……80

5.3.27 ERLE with the speech of a man with tone in English……80

5.4.1 Step-size of OTLMS algorithm……....82

5.4.2 Performance comparison between optimal time-variant and fixed step sizes..82

5.4.3 Nonlinear I/O mapping curves with the modeling……84

5.4.4 Time-variant step-size with different mapping curves……85

5.4.5 Residual error power with different mapping curves…...……85

5.4.6 Time-variant step-size with different $\hat{\beta}$ ……86

5.4.7 Residual error power with different $\hat{\beta}$ ……86

5.4.8 Time-variant step-size using the first by using first-order recursive procedure with different $\widehat{\beta}$ …….....88

5.4.9 Residual error power using the first by using first-order recursive procedure with different $\widehat{\beta}$ …….……88

5.4.10 Residual error power using the two different procedures with β_o ……89

5.4.11 Residual error power using the two different procedures with one-half β_o 89

5.4.12 Residual error power using the two different procedures with two times β_o .90

5.4.13 Performance comparison of convergence rate for OTLMS, OTTLMS and fixed step-sizes…….91

5.4.14 Performance comparison of convergence rate for OTLMS, OTTLMS with three different mapping curves……..…….93

5.4.15 Model of RIR as an exponential decay envelope……93

5.4.16 Performance comparison of convergence rate for OTLMS and OTTLMS with three different mapping curves…….……94

5.4.17 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.96$ with three different mapping curve…….……94

5.4.18 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.98$ with three different mapping curve…….……95

5.4.19 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.94$ with three different mapping curve……95

5.4.20 Residual error power of OTTLMS, OTLMS, fixed step-size LMS for the two-staged adaptation with fixed switching point…….97

5.4.21 Residual error power of OTTLMS, OTLMS, fixed step-size LMS for the two-staged adaptation with detection of state-stay linear filter……..97

5.4.22 ERLE of practical OTTLMS, practical OTLMS, fixed step-size LMS algorithm with the first part of speech and pseudo nonlinear echo……..99

5.4.24 ERLE of practical OTTLMS, practical OTLMS, fixed step-size LMS algorithm with the second part of speech and a pseudo nonlinear echo……100

5.4.25 Step-size of practical OTLMS algorithm with the second part of speech and a pseudo nonlinear echo……100

5.4.26 ERLE of practical OTLMS, fixed step-size LMS algorithm with the first part of speech and a true nonlinear echo……..…….101

5.4.27 Step-size of practical OTLMS algorithm with the first part of speech and a true nonlinear echo……102

5.4.28 ERLE of practical OTLMS, fixed step-size LMS algorithm with second part of speech for true nonlinear echo……102

5.4.29 Step-size of practical OTLMS algorithm with the second part of speech for a true nonlinear echo……...……..103

List of Tables

Table 2.1 Look-up table of a PWL processor……...10

Table 2.2 Look-up table of decomposition introduced by Heredia [20]……10

Table 2.3 Comparison of computational cost, no. multiplication per sample……..12

Chapter 1
Introduction

In these years, hands-free telephone and teleconference systems are widely used. Acoustic echo cancellation (AEC) is a major concern in telecommunications, where echo delay is particularly annoying for speakers. The problem occurs 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 follows.

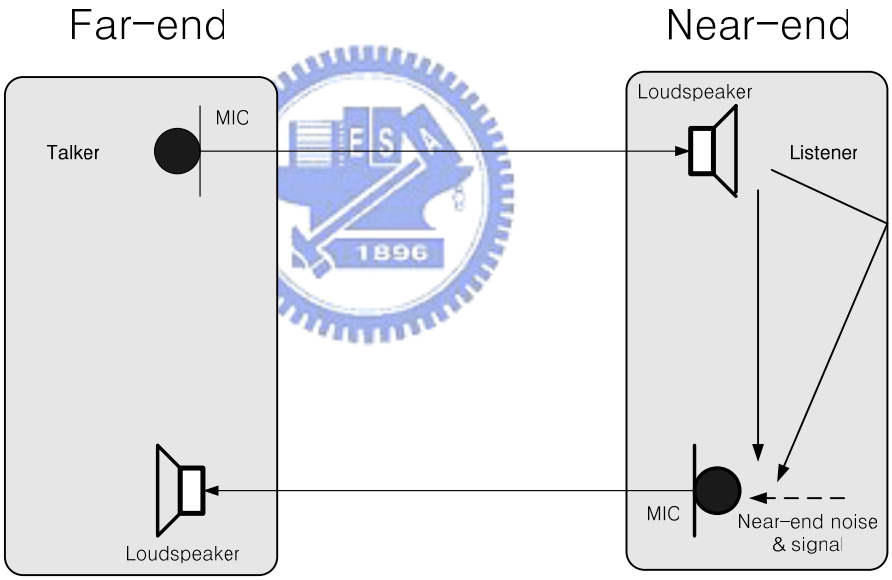

Fig. 1.1 Diagram of hands-free telephone system

A simplified diagram of hands-free telephone system is shown in Fig. 1.1. Assume that a talker in the far-end uses microphone to communicate to the listener in near-end, the far-end speech will be transmitted back to the far-end through the loudspeaker and room impulse response. The main object of acoustic echo cancellation (AEC) is to estimate the unknown echo path and subtract the estimated echo components from the microphone output.

In the past, based on the gradient theory, the acoustic echo that is linearly dependent on the loudspeaker can be cancelled effectively [1]-[3]. However, more and more telecommunications areas use heads-free devices to improve customer comfort. These devices drive higher power-amplification and power loudspeaker commonly driven into saturation region [4]. This issue leads to a nonlinear filtering problem that cannot be solved by conventional linear AEC.

In this thesis, the nonlinear AEC system is shown in Fig.1.2. The signal from the far end is passing through the nonlinear loudspeaker and the room impulse response and then is picked up by the microphone. The nonlinear AEC is supposed to cancel the nonlinear echo. The nonlinear echo can be cancelled perfectly if the nonlinear AEC filter is identical to the nonlinear loudspeaker and room impulse response.

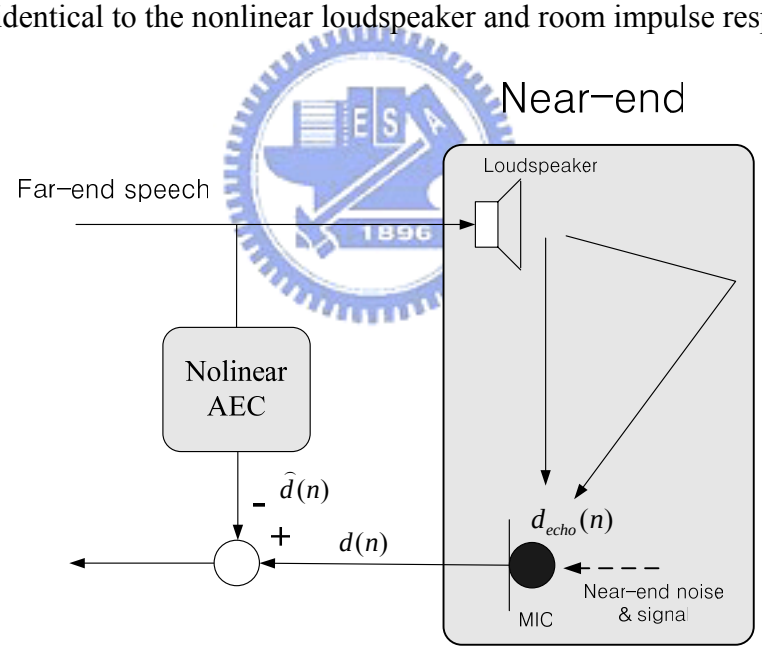

Fig.1.2 Nonlinear acoustic echo cancellation system

The design and analysis of nonlinear adaptive filtering is difficult. The popular method is via polynomial functions, i.e. truncated Volterra [5], Wiener [6] and Hammerstein models [7]-[9]. Wiener and Hammerstein model are special cases of Volterra one with worse fitting but much less coefficients accompanied less

computations. Although Hammerstein model has the least complexity, it still has large computations.

In the thesis, we will employ the piecewise linear (PWL) [10] method to lessen this issue in nonlinear AEC. These are also other reasons which suggest that it may be worthwhile to investigate the simplicity of PWL implementation, as well as its theoretical analysis. The former is due to the fact that digital controllers based on such systems can be built easily using "if $p(x)$ then $f(x)$ else… ". The latter own itself to the maturity of linear algebra.

Historically, a closed form of the canonical PWL (CPWL) was presented by Chua [10]. Lin used the least-mean-square (LMS) algorithm for CPWL [11]. The identification of Hammerstein model using two-segment nonlinearity, different polynomial functions on positive and negative regions, was demonstrated in [12]. Later on, Vörös demonstrated multi-segment PWL characteristics, the same idea as CPWL, with recursive-least-square (RLS) [13] in a special case of, high SNR, a specific set of initials and parameters. In this thesis, we will employ it for nonlinear AEC in various cases and discuss.

Although the PWL processor has benefit on the computation, it still is proportional to the linear filter length M. In the past, many types of selective update schemes for the adaptive linear filter have been described in [14]-[16]. We will extend their concepts and take advantages of the particular PWL structure to develop PWL coefficient selective update schemes.

After demonstrating our system flow, we now focus on the stability issue. The transient behavior of linear filter and PWL processor can account for the divergence problem encountered in cascade system. Convergence cannot be guaranteed, since each filter (linear filter or PWL processor) behaves to compensate the other one's

misalignment, which can lead to a perpetual oscillating system.

In order to overcome this difficulty, Guérin [7] points out that the linear filter has to adapt continuously so as to react to any change in the acoustic path, and the PWL filter must not adapt until the linear filter has sufficiently converged. The two-staged strategy starts with a linear filter and then joint adaptation of PWL and linear coefficients follows once the linear filter has sufficiently converged in the first stage. Moreover, in this thesis we will derive the theoretical convergence analysis and stability criterion of the two-staged algorithm.

Next, we will focus on the step-size. We know a large step size gives a faster convergence but also large small residual error power. Therefore, various methods employing varying step-size have be examined by the other researchers, including time-varying [17], tap-varying [18] or both time- & tap- varying [19]. We will use the convergence analysis of the first stage to develop the optimum time-variant and optimum time-&tap-variant step-size LMS algorithm.

In Chapter 2, we introduce our system model and its update scheme. In Chapter 3, a two staged algorithm is used and its performance analysis of a PWL structure is proved analytically. In Chapter 4 The optimal step size with a nonlinear modeling error is derived. Simulations support our works in Chapter 5 and conclusion will be given in Chapter 6.

The main efforts in this thesis are:

1. We introduce a PWL structure of adaptive nonlinear AEC with lower complexity and develop its joint LMS adaptive algorithm.
2. We develop five types of PWL coefficients selective update schemes
3. For joint LMS adaptive algorithm, we derive and verify convergence analyses of the two-staged adaptation with a stability condition of linear step-size during the

first stage.

4. Further, we discuss the convergence behavior during the first stage with different factors, nonlinear effect, power and Pdf. of the far-end signal, step-size and SNR.
5. Optimum time-variant and time-&tap- variant step-size LMS algorithm of linear adaptive filter with a nonlinear modeling error are derived and verified.
6. The corresponding practical forms are also proposed and discussed.

Chapter 2
Nonlinear Adaptive filter

In Chapter 2 we will build a system model for nonlinear AEC and develop its adaptive algorithm. Moreover, we will discuss the issue of the computation cost. Further, five types of partial update schemes will be presented to reduce the computational load.

In Section 2.1, the nonlinear AEC is first introduced as a cascade of nonlinear processor and linear adaptive filter, where a PWL function is used to model the nonlinear loudspeaker. Moreover, we will develop the joint LMS-type algorithm to update both PWL and linear coefficients. In Section 2.2, the computation complexity of the PWL-based nonlinear adaptive filter is compared to that of polynomial-based model and linear AEC. In Section 2.3, several partial update schemes are proposed to further reduce the computation at the cost of degraded convergence rate.

2.1 Adaptive nonlinear LMS AEC using PWL structure

In order to separate the identification of the nonlinear loudspeaker parameters and the tracking of the linear acoustic path changes, Fig. 2.1 shows a typical cascade nonlinear AEC [8], also known as a Hammerstein model. The far end signal $x(n)$ is fed into the PWL processor *f(x)* that approximates the nonlinear mapping function with one or more linear equations. It has been exploited to compensate for the effect of nonlinear echo.

The output $s(n)$ of the PWL processor passes through a linear filter to form a pseudo nonlinear echo $\hat{d}(n)$. Here we assume the nonlinear distortion is caused only by the loudspeaker and the let $d(n)$ denote the desired signal.

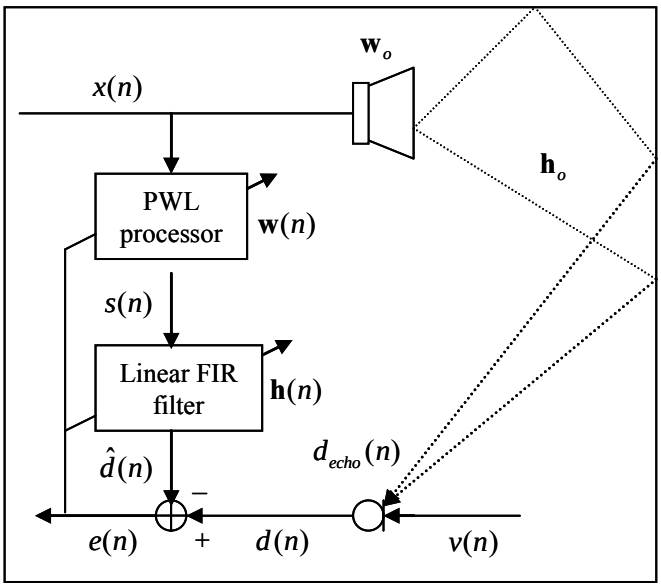

Fig. 2.1 Nonlinear acoustic echo canceller based on piecewise linear structure

The PWL function *f(x)* for the speech input range [-1 1] is assumed to be symmetric and its prototype is given by:

$$f(x)=\begin{cases} m_1 x & ,\alpha_1 \le |x| < \alpha_2 \\ m_2(x-\alpha_2)+m_1\alpha_2 & ,\alpha_2 \le |x| < \alpha_3 \\ \vdots & \\ m_N(x-\alpha_N)+m_{N-1}(\alpha_N-\alpha_{N-1})+\cdots m_1\alpha_1, & \alpha_N \le |x| < \alpha_{N+1}, \end{cases} \tag{2.1.1}$$

where m_j and α_j account for the slope and partition parameters of the linear subregion, respectively, with $\alpha_1 = 0$ and $\alpha_{N+1} = 1$.

The prototype of PWL function in Eq. (2.1.1) consists of a series of linear subfunctions which are properly partitioned into subregions of the nonlinear curve. Here we adopt a canonical piece-wise linear (CPWL) function [10], which is an analytic formula with several absolute-value operators. Its memoryless form with zero offset is given by

$$f(x) = w_1 x + \sum_{j=2}^{N} w_j \left(\left| x - \alpha_j \right| \right),$$

where the CPWL coefficient w_j is a function of some slopes of the sublinear

functions and N is the CPWL tap order. Extending to the symmetrical function and associated to Eq. (2.1.1), we can get a modified form as follows

$$f(x)=\sum_{j=1}^{N} w_j f_j(x), \tag{2.1.2}$$

where

$$f_j(x)=\left(\frac{1}{2}|x-\alpha_j|-\frac{1}{2}|x+\alpha_j|+x\right)$$

and $w_1=m_1, w_i=(m_i-m_{i-1}),\ \forall i=2\sim N$.

We use an example to demonstrate how the CPWL function works. Consider a 3-segment CPWL function in Fig. 2.2 with partition paraments $\alpha_2=0.4$ and $\alpha_3=0.7$ and slopes $w_1=1, w_2=-0.4$ and $w_3=-0.6$. In Fig. 2.2, we can observe it performs canonically with every breakpoint α_i.

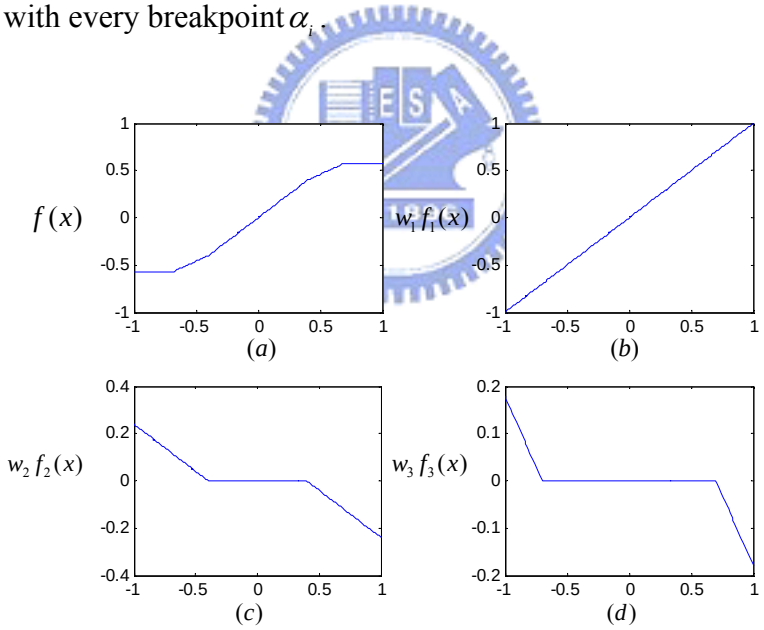

Fig. 2.2 (a) A CPWL curve with three segments (b)~(d) Associated canonical curves

From now on, all PWL functions are of canonical form. The block diagram of a PWL processor is shown in Fig. 2.3, where the far end signal is decomposed to N analytic signals on the block f_j, then multiplied by its associated coefficient w_j, and finally all of them are synthesized together to approximate the nonlinearly

distorted loudspeaker output. Here the *j*-th output of the block $f_j(x)$ will be null when the far end signal x is smaller than the *j*-th partition area value. It will be beneficial to reduce computation load, which will be dealt with later.

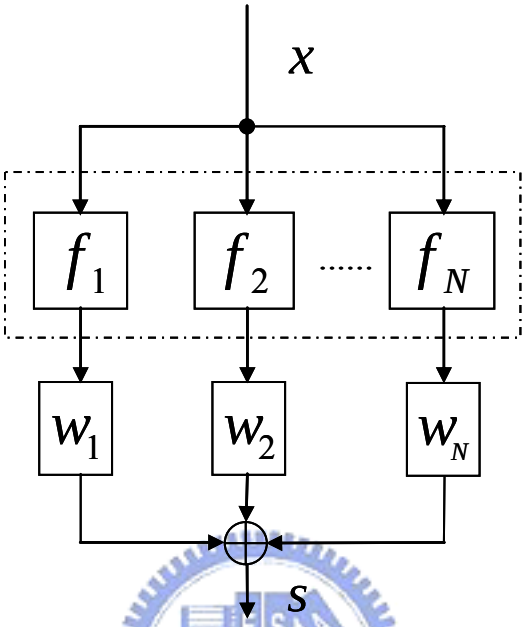

Fig. 2.3 The block diagram of a PWL processor

The overall nonlinear AEC can be represented as a vector form. The output of the nonlinear processor $s(n)$ is given by

$$s = \mathbf{w}^T \cdot \mathbf{f}\,, \tag{2.1.3}$$

where $\mathbf{w} = [w_1 \;\; w_2 \;\; \ldots \; w_N]^T$ and

$$\mathbf{f} = [f_1(x) \;\; f_2(x) \;\; \ldots \;\; f_N(x)]^T$$

resembles a decomposition that maps real numbers into vectors using a set of predefined partition parameters $\{0,\ \alpha_2, \cdots \alpha_N, 1\}$. For example, if the input x is $0.8, N = 4$, and the partition parameters are $\{0,\ 0.4,\ 0.7,\ 0.9,\ 1\}$, then the vector $\mathbf{f}$ will be $[0.8,\ \ 0.4,\ \ 0.1,\ \ 0]^T$. In Table 2.1, we show the look-up table for a symmetric partition $\{\alpha_1,\ \alpha_2, \cdots \alpha_N, \alpha_{N+1}\}$ without a mixed part that contains both

positive and negative elements. [20] has also proposed a similar decomposition shown in Table 2.2. But our modified CPWL function is an explicit equation to achieve the decomposition and our procedure can be easily extended to that, by $g_1(x) = f_1(x)$ and $g_j(x) = f_j(x) - f_{j-1}(x), \forall j = 2 \sim N$.

Table 2.1 Look-up table of a PWL processor

$f_j(x)$	$\lvert x\rvert \geq \alpha_j$	$\lvert x\rvert \leq \alpha_j$
$x \geq 0$	$x - \alpha_j$	0
$x < 0$	$x + \alpha_j$	0

Table 2.2 Look-up table of decomposition introduced by Heredia [20]

$g_j(x)$	$\lvert x\rvert > \alpha_{j+1}$	$\lvert x\rvert \in (\alpha_{j+1}, \alpha_j)$	$\lvert x\rvert \leq \alpha_j$
$x \geq 0$	$\alpha_{j+1} - \alpha_j$	$x - \alpha_j$	0
$x < 0$	$\alpha_j - \alpha_{j+1}$	$\alpha_j + x(n)$	0

After discussing the decomposition of the vector $\mathbf{f}$, we now come back to focus on the system flow in Fig. 2.1. From Eq. (2.1.3), the delay tap form of PWL processor $\mathbf{s}(n) = [s(n)\ s(n-1)\ \ldots\ s(n-M+1)]^T$ can be expressed as

$$\mathbf{s}(n) = \mathbf{F}(n) \cdot \mathbf{w}, \tag{2.1.4}$$

where $\mathbf{F}(n) = [\mathbf{f}(n)\ \ \mathbf{f}(n-1)\ \cdots\ \ \mathbf{f}(n-M+1)]^T$ (2.1.5)

is the delayed tap mapping matrix. Therefore, the nonlinear AEC output signal $\hat{d}(n)$ can be written as

$$\hat{d}(n) = \mathbf{s}^T(n) \cdot \mathbf{h},$$

where $\mathbf{h} = [h_0 \; h_1 \; \ldots \; h_{M-1}]^T$ represents the estimated coefficients vector of the linear FIR filter with M being the length of the filter. The estimated error is

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n) \\ &= d(n) - \mathbf{s}^T(n) \cdot \mathbf{h} - v(n) \end{aligned}$$

If the coefficients vectors are updated with step size μ_h and μ_w, a joint LMS-type adaptive algorithm according to the gradient of the cost function, $J(n) = e^2(n)$, is given by

$$\begin{aligned} \mathbf{h}(n+1) &= \mathbf{h}(n) + \mu_h \mathbf{w}(n) \cdot \mathbf{F}(n) e(n) \\ &= \mathbf{h}(n) + \mu_h \mathbf{s}(n) e(n) \end{aligned} \tag{2.1.6}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w \mathbf{F}^T(n) \cdot \mathbf{h}(n) e(n). \tag{2.1.7}$$

Now that we have developed a nonlinear adaptive filter algorithm for a PWL structure in (2.1.6) and (2.1.7). Similarly, in case of a polynomial structure, we can simply modify the delayed tap mapping matrix $\mathbf{F}(n)$ in Eq. (2.1.6) and (2.1.7) by setting it as follows:

$$\mathbf{F}(n) = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M+1) \\ x^2(n) & x^2(n-1) & \cdots & x^2(n-M+1) \\ & & \vdots & \\ x^N(n) & x^N(n-1) & \cdots & x^N(n-M+1) \end{bmatrix}.$$

2.2 Computation of adaptive nonlinear LMS AEC

The following discussion with respect to computational complexity is based on the number of real multiplications that is required by different structures. In Eq. (2.1.6), the matrix-vector product of $\mathbf{w}(n) \cdot \mathbf{F}(n)$ is simply $\mathbf{s}(n)$, which is readily available as shown in Eq.(2.1.4) and Fig. 2.3. However, as opposed to the linear adaptive AEC algorithm, the nonlinear algorithm needs $O(MN + N) \approx O(MN)$ more

computation due to the matrix-vector product of $\mathbf{F}^T(n)\cdot\mathbf{h}(n)$ in the vital equation (2.1.7). We note that the polynomial structure also has the same complexity [21].

We also note that the output of the operation $f_j(x)$ in matrix $\mathbf{F}(n)$ would be null when the far end signal x at iteration n is smaller than the partition parameter α_{j+1}. Due to the zero output of the block f_j, the average number of non-zero entries of $\mathbf{f}(n)$, the column vector of the matrix $\mathbf{F}^T(n)$, will be $O(\frac{N}{2})$ for a uniformly distributed far end signal x. Besides, the computation cost will almost reduces to $O(M\frac{N}{2})$. The computational cost is listed in Table 2.3. We can see the PWL structure has the lower computation than that of polynomial.

Table 2.3 Comparison of computational cost, no. multiplication per sample

Complexity of computation	No. of multiplication (approx.) per sample
Linear AEC	$2M$
Polynomial NAEC ($N=3$)	$2M+MN$
PWL NAEC ($N=3$)	$2M+\frac{MN}{2}$

2.3 Partial update of adaptive nonlinear LMS AEC

However, the PWL structure has reduced the computation on the matrix multiplication $\mathbf{F}^T(n)\cdot\mathbf{h}(n)$ with $O(\frac{N+1}{2}M)$ multiplications. It still is proportional to the linear filter length M . In acoustic echo cancellation, adaptive linear filter often require a large number M of coefficients to model the acoustic echo path with sufficient accuracy. It means that for long linear filter the adaptation task can become more prohibitively expensive.

Partial updating of the LMS adaptive linear filter has been proposed to reduce computational costs and power consumption [22], which is quite attractive in the area of mobile computing and communications that requires the adaptive linear filter to have a very large number of coefficients. Updating the entire coefficient vector of the adaptive linear filter is costly in terms of power, memory, and computations and is sometimes impractical for mobile devices.

In the past, many types of selective update schemes for the adaptive linear filter have been described in [14]-[16]. In this section, after introducing these selective update schemes, we will extend their concepts to the three types of PWL coefficient selective update schemes and take advantages of the particular PWL structure to develop the two types of ones in which only one PWL coefficient are adjusted at each sample time in order to reduce the matrix multiplication $\mathbf{F}^T(n)\cdot\mathbf{h}(n)$ down to N multiplications.

2.3.1 Periodic partial update LMS algorithm

The most prevalent type in the literature of selective update scheme is referred to as the periodic LMS algorithm [14]. To reduce computation needed during the update part of the adaptive filter by a factor of N, the periodic LMS algorithm updates all the filter coefficients every N iterations instead of every iteration. In addition, the coefficient updates for this algorithm are regular, as only one coefficient is changed at one iteration. With this concept, the PWL coefficient update is given by

$$w_j(n+1)=\begin{cases} w_j(n)+\mu e(l)\left[\mathbf{F}^T(l)\cdot\mathbf{h}(l)\right]_j, if\ j=\left(n\ \bmod\ N\right)+1,\ l=N\lfloor n/N\rfloor \\ \qquad w_j(n) \qquad\qquad\qquad ,\text{otherwise.} \end{cases} \tag{2.3.1}$$

where $\lfloor\cdot\rfloor$ denotes the truncation operation, n mod N denotes iteration n modulo N. By considering N iterations of the updates in Eq. (2.3.1), it can be shown that this algorithm is equivalent to the following N-fold coefficient vector update:

$$\mathbf{w}(n+N)=\mathbf{w}(n)+\mu e(n)\mathbf{F}^T(n)\cdot\mathbf{h}(n). \tag{2.3.2}$$

It describes a modified version of the LMS adaptive algorithm that uses every Nth instantaneous gradient to update the filter coefficients.

2.3.2 Sequential partial update LMS algorithm

Like the periodic LMS algorithm, for the sequential LMS algorithm [15] the update coefficient is chosen in a predetermined fashion, a regular pattern, but uses sequential gradient vector signal respect to $w_j(n)$. Extending to the PWL coefficients update, it is given by

$$w_j(n+1)=\begin{cases} w_j(n)+\mu e(n)\left[\mathbf{F}^T(n)\cdot\mathbf{h}(n)\right]_j, \text{ if } j=\left(n\ \bmod\ N\right)+1, \\ \qquad w_j(n) \qquad\qquad\qquad ,\text{otherwise.} \end{cases} \tag{2.3.3}$$

Define Ψ_j by filling *1* on the *j*-th diagonal entry of the zero matrix and the above update equation can be written in a more compact form

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\Psi_{n\%N+1}\mathbf{F}^T(n)\cdot\mathbf{h}(n). \tag{2.3.4}$$

2.3.3 Stochastic partial update LMS algorithm

Being similar to the sequential LMS algorithm in the sense that also uses data-independent updating scheme, the stochastic partial update LMS [16] algorithm performs sequential instant gradient respect to $w_j(n)$. The difference is as follows. At a given iteration *k*, the sequential LMS processes a regular processing strategy to select which one coefficient is to be updated, whereas for the stochastic partial update LMS, one of the coefficient is chosen at random from $\{1, 2\cdots N\}$ with probability $1/N$ and subsequently the update is performed i.e.,

$$w_j(n+1) = \begin{cases} w_j(n) + \mu e(n)\left[\mathbf{F}^T(n)\cdot\mathbf{h}(n)\right]_j, & \text{if } j \text{ is chosen at random} \\ w_j(n), & \text{otherwise} \end{cases} \tag{2.3.5}$$

and the coefficient vector update can be expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\Psi_n\mathbf{F}^T(n)\cdot\mathbf{h}(n), \tag{2.3.6}$$

where Ψ_n now is a random matrix chosen at random from $\Psi_j, j = 1\cdots N$ with probability $1/N$ (recall that Ψ_j by filling one on the *j*-th diagonal entry of the zero matrix).

2.3.4 Variant-periodic partial update LMS algorithm

Unlike previous 3 well-known partial update schemes which are applicable to general LMS-type algorithms, we will take advantages of the particular PWL structure to develop a variant periodic partial update scheme.

As noted earlier, when the far end signal $x(n)$ is smaller than the partition parameter α_{j+1}, the output of the operation $\mathrm{f}_j\left(x(n)\right)$ would be null. As a result, the periodic LMS algorithm would be inefficient. In order to solve this issue, we propose a variant periodic LMS algorithm that takes advantages of the located partition area of far end signal $x(n)$ to avoid an inefficient update term.

For example, if the located partition area of far end signal $x(n)$ is the 2nd one, the variant periodic LMS algorithm updates its first coefficient $c_1(n)$ at time n and second coefficient $c_2(n)$ at time $n+1$. If the located partition area of far end signal $x(n+2)$ is the 3rd one, it updates the first coefficient $c_1(n)$ at time $n+2$, second coefficient $c_2(n)$ at time $n+3$ and third coefficient $c_3(n)$ at time $n+4$. Here we denoted $Q(n)$ is the number of located partition area of far-end signal $x(n)$ at time n. For example the partition is $\{0,\ 0.4,\ 0.7,\ 0.9,\ 1\}$ and far end signal $x(n)$ is 0.8, then $Q(l)$ is 3. The following non-period LMS algorithm is give by:

$$w_j(n+1)=\begin{cases} w_j(n)+\mu e(l)\left[\mathbf{F}^T(l)\cdot\mathbf{h}(l)\right]_j, & if\ j=\left(n\ \bmod\ Q(l)\right)+1,\ l=Q(l)\lfloor n/Q(l)\rfloor \\ w_j(n) & ,\text{otherwise} \end{cases} \tag{2.3.7}$$

and the $Q(n)$ -fold coefficient vector update can be expressed as

$$\mathbf{w}(n+Q(l))=\mathbf{w}(n)+\mu e(n)\mathbf{F}^T(n)\cdot\mathbf{h}(n),\ l=Q(l)\lfloor n/Q(l)\rfloor \tag{2.3.8}$$

2.3.5 Located partial update LMS algorithm

Moreover, we will again utilize the particular PWL structure to develop a located partial update scheme.

As we know, the PWL processor would decompose the far end signal $x(n)$ into *N* analytic signals. The resulting analytic signal may be zero, depending on the far end signal $x(n)$. If far end signal $x(n)$ falls into the partition parameter α_{j+1}, the corresponding *j*-th analytic signal will be null. This characteristic is also the main idea of the variant periodic LMS algorithm in Section 2.3.4. We also know the nonlinearity mostly happens for a high level input. Therefore, the *(j+1)*th entry of PWL coefficient vector has higher priority than the *j*-th entry.

By combining these characteristics, we propose a new partial update LMS algorithm, located LMS, for PWL coefficient as follows.

$$w_j(n+1) = \begin{cases} w_j(n) + \mu e(n)\left[\mathbf{F}^T(n)\cdot\mathbf{h}(n)\right]_j, & j = Q(n) \\ w_j(n), & \text{otherwise.} \end{cases} \tag{2.3.9}$$

where $Q(n)$ is the number of located partition area of far end signal $x(n)$ at time n. The update strategy is to choose the PWL coefficient which far end signal $x(n)$ falls into. The benefit is when the power loudspeaker commonly driven with saturation region, mostly high level far end signal, the nonlinearity information is sufficiently used on the PWL coefficient. On the contrary, if the nonlinearity effect was insignificant in case of low level far end signal, it maintains the linear part of PWL coefficients. In the same way, the above update equation can be written in a more compact form as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\Psi_{Q(n)}\mathbf{F}^T(n)\cdot\mathbf{h}(n). \tag{2.3.10}$$

Computer simulations in Chapter 5 will compare the performance of the above 5

partial update and full LMS algorithm.

2.4 Summary

In this chapter, we performed an adaptive nonlinear AEC based on a PWL type function and developed its joint LMS algorithm in section 2.1. Moreover, the comparison of computation complexity of three structures, linear AEC, nonlinear AEC in a case of PWL and polynomial one was discussed in section 2.2. The PWL one has the lower computation than that of polynomial and has just about $2M$ more multiplications than that of linear AEC. Finally, we presented 3 well-known and 2 proposed partial update of LMS algorithm in section 2.3. They all keep the computation on the matrix multiplication $\mathbf{F}^T(n)\cdot\mathbf{h}(n)$ with M. The computation simulations in chapter 5 will compare the above performances explicitly.

Chapter 3
Two-Staged Adaptation and Its Convergence Analysis

In Chapter 2, we have derived the joint adaptation of the nonlinear PWL AEC. However, each filter (or processor) behaves to compensate the other one's misalignment. This can result in a perpetual oscillating system.

Therefore, in this chapter we will adopt two-staged strategy [7] to overcome this difficulty. This strategy is to start with a linear filter update in the first stage, and then joint update of both PWL and linear coefficients follows in the second stage.

In Section 4.1 the two-staged adaptation is introduced. In the first stage, the convergence analysis and stability criterion will be derived in Section 4.2. After that, we will derive the convergence analysis of the second stage in Section 4.3.

3.1 Two-staged adaptation

For simplicity, Fig. 2.1 is redrawn in Fig. 3.1. Here we assume the nonlinear loudspeaker and linear room impulse response are time-invariant; the near end signal $v(n)$ only contains a white Gaussian noise (WGN) and the nonlinear echo

$$d(n) = \mathbf{s}_o^T(n) \cdot \mathbf{h}_o + v(n),$$

where $s_o(n)$ is the optimal PWL processor output with the optimal PWL coefficients $\mathbf{w}_o$ and $\mathbf{h}_o$ is the optimal linear coefficients.

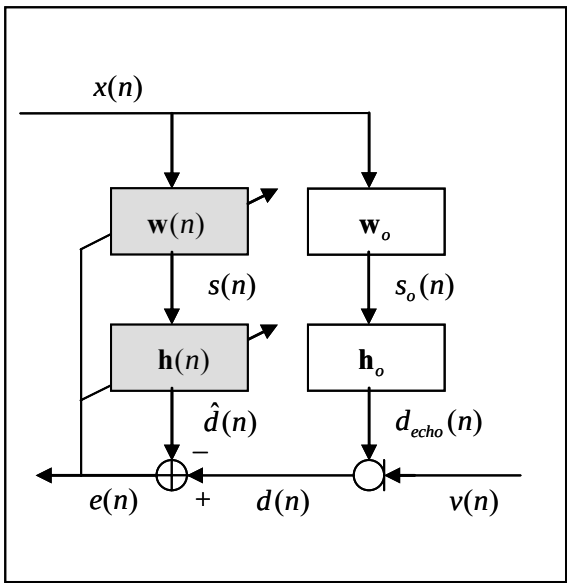

Fig. 3.1 Cascade model of system (right hand part) and mirror adaptive system (left hand part)

In the cascade structure of Fig. 3.1, if the joint updates of both nonlinear PWL coefficients $\mathbf{w}(n)$ and linear coefficients $\mathbf{h}(n)$ are performed simultaneously as given in Eq. (2.1.6) and (2.1.7), the danger of divergence can happen. To illustrate this tendency of divergence, Fig. 3.2 includes two desperate divergent curves of joint-updating schemes with two step sizes $\mu_h = \mu_w = 0.002$ and 0.003 and SNR=20dB.

The transient behavior of linear filter and PWL processor can account for the divergence problem encountered in cascade system. When both $\mathbf{w}(n)$ and $\mathbf{h}(n)$ are far away from their optimum coefficients $\mathbf{w}_o$ and $\mathbf{h}_o$, respectively, in the early transient stage, the resulting residual error $e(n)$ does not push either coefficients towards their optimum points. As a result, convergence cannot be guaranteed, since each filter (linear filter or PWL processor) behaves to compensate the other one's misalignment, which can lead to a perpetual oscillating system. An analytical stability criterion for joint adaptation of the nonlinear AEC can be very strenuous.

In order to overcome this difficulty, Guérin [5] points out that the linear filter has to adapt continuously so as to react to any change in the acoustic path, and the PWL filter must not adapt until the linear filter has sufficiently converged. The two-staged strategy starts with a linear filter, and then joint adaptation of PWL and linear coefficients follows once the linear filter has sufficiently converged in the first stage. Fig. 3.2 shows the significant improvements in residual error power using the two-staged algorithm.

In the past the convergence analysis of a cascade system was done under a perfect information linear or nonlinear part [21]. Next, we will proceed to derive the theoretical convergence analysis of the two-staged algorithm.

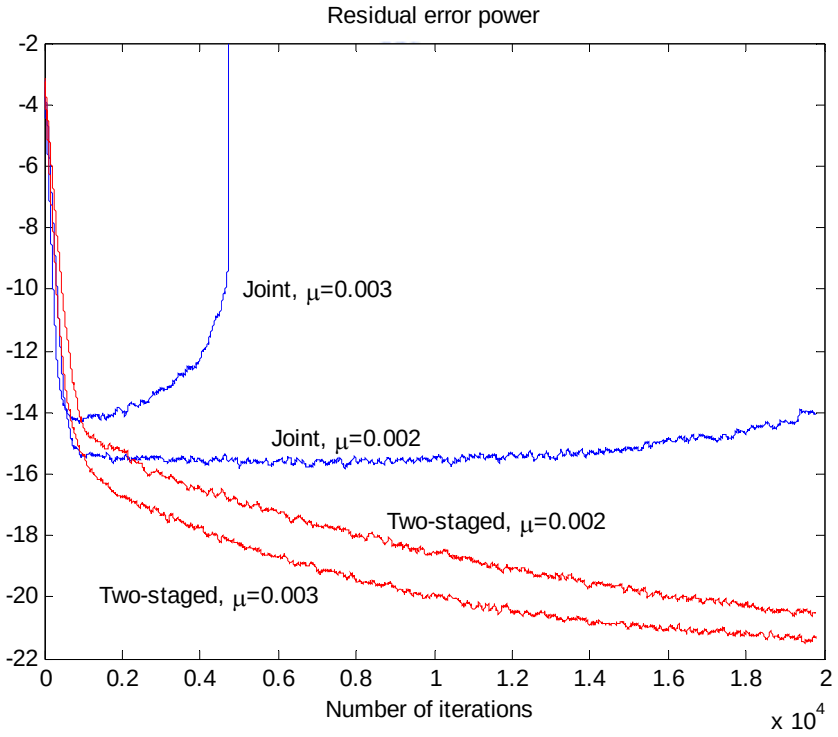

Fig. 3.2 Performance comparison of the two update schemes with two different step-sizes

3.2 Convergence and stability analysis of linear adaptation

In the first stage, only linear coefficients update under a fixed PWL coefficients. We denote the linear filter weight-error vector by

$$\boldsymbol{\varepsilon}_h(n) = \mathbf{h}(n) - \mathbf{h}_o . \tag{3.2.1}$$

where $\mathbf{h}_o$ is the optimal linear filter. The estimation error produced by nonlinear AEC filter can be expressed as

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n) \\ &= \mathbf{s}_o^T(n)\cdot\mathbf{h}_o + v(n) - \mathbf{s}^T(n)\cdot\mathbf{h}(n) \\ &= \mathbf{s}_o^T(n)\cdot\mathbf{h}_o + v(n) - \left(\mathbf{s}_e(n)+\mathbf{s}_o(n)\right)^T\left(\mathbf{h}_o+\boldsymbol{\varepsilon}_h(n)\right) \\ &= v(n) - \mathbf{s}_e^T(n)\cdot\mathbf{h}_o - \mathbf{s}^T(n)\cdot\boldsymbol{\varepsilon}_{\mathbf{h}}(n) \end{aligned} \tag{3.2.2}$$

where $\mathbf{s}_e(n) = \mathbf{s}(n) - \mathbf{s}_o(n)$ is the error of PWL processor. Using Eq. (2.1.6) and (3.2.2), we may rewrite $\boldsymbol{\varepsilon}_h(n+1)$ as

$$\begin{aligned} \boldsymbol{\varepsilon}_h(n+1) &= \mathbf{h}(n) + \mu_h\mathbf{s}(n)e(n) - \mathbf{h}_o \\ &= \boldsymbol{\varepsilon}_h(n) + \mu_h\mathbf{s}(n)\left(v(n) - \mathbf{s}_e^T(n)\cdot\mathbf{h}_o - \mathbf{s}^T(n)\cdot\boldsymbol{\varepsilon}_{\mathbf{h}}(n)\right) \\ &= \left[\mathbf{I} - \mu_h\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]\cdot\boldsymbol{\varepsilon}_h(n) + \mu_h\left[v(n)\mathbf{s}(n) - \mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\cdot\mathbf{h}_o\right] \end{aligned} \tag{3.2.3}$$

3.2.1 Mean bias of linear coefficient weight error

Taking the expectation on Eq. (3.2.3) and assuming the variation of $\boldsymbol{\varepsilon}_h(n)$ is slow compared with that of $\mathbf{s}(n)$, the first moment of $\boldsymbol{\varepsilon}_h(n)$ is given by

$$E\left\{\boldsymbol{\varepsilon}_h(n+1)\right\} = \left(\mathbf{I}_M - \mu_h\mathbf{R}_s\right)\cdot E\left\{\boldsymbol{\varepsilon}_h(n)\right\} + \mathrm{E}\left\{\mathbf{f}_h(n)\right\}, \tag{3.2.4}$$

where $\mathbf{f}_h(n) = \mu_h\left[v(n)\mathbf{s}(n) - \mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right]$ and $\mathbf{R}_s$ is the correlation matrix of $\mathbf{s}(n)$ by applying the unitary similarity transformation. We can diagonalize $\mathbf{R}_s$ as follows: $\mathbf{Q}_s^T\mathbf{R}_s\mathbf{Q}_s = \mathbf{D}_s$, where $\mathbf{Q}_s$ is an unitary matrix and $\mathbf{D}_s$ is a diagonal

matrix consisting of the eigenvalues $\lambda_{s,i}$ of $\mathbf{R}_s$. Let $\mathbf{k}_h(n) = \mathbf{Q}_s^T \boldsymbol{\varepsilon}_h(n)$, then we may transform Eq. (4.2.4) into the form

$$E\{\mathbf{k}_h(n+1)\} = (\mathbf{I}_M - \mu_h \mathbf{D}_s) \cdot E\{\mathbf{k}_h(n)\} - \mathbf{Q}_s^T \mathbf{R}_{s,s_e} \mathbf{h}_o , \tag{3.2.5}$$

where $\mathbf{R}_{s,s_e}$ is the cross-correlation matrix of $\mathbf{s}(n)$ and $\mathbf{s}_e(n)$. We may go on to express Eq. (4.2.5) as

$$E\{\mathbf{k}_h(n)\} = -\mathbf{D}_s^{-1}\mathbf{Q}_s^T \mathbf{R}_{s,s_e} \mathbf{h}_o + \left(\mathbf{k}_h(0) + \mathbf{D}_s^{-1}\mathbf{Q}_s^T \mathbf{R}_{s,s_e} \mathbf{h}_o\right)(\mathbf{I}_M - \mu_h \mathbf{D}_s)^n . \tag{3.2.6}$$

It makes sense that the linear coefficients weight error $\boldsymbol{\varepsilon}_h(n)$ convergences to a biased estimate due to the effect of nonlinear bias. This linear bias part also agrees with the optimal linear weight error [5], denoted by $\boldsymbol{\varepsilon}_{h,MMSE} = \mathbf{h}_{MMSE} - \mathbf{h}_o$, in the minimum mean square error (MMSE) sense. We will prove it as follows. First, $\mathbf{h}_{MMSE}$ satisfies the equation $\nabla_{\mathbf{h}}\{J(n)\}_{\mathbf{h}=\mathbf{h}_{MMSE}} = 0$, which depends on the correlation matrix $\mathbf{R}_s$ and the cross-correlation matrix $\mathbf{R}_{s,s_e}$. Using now the quasistationarity hypothesis, the optimal filter is then defined by the following expression $\mathbf{h}_{MMSE} = \mathbf{R}_s^{-1} \cdot \mathbf{R}_{s,s_e} \cdot \mathbf{h}_o$. Hence, it can yield $\mathbf{k}_{h,MMSE} = -\mathbf{D}_s^{-1}\mathbf{Q}_s^T \mathbf{R}_{s,s_e} \mathbf{h}_o$.

If the far end signal $x(n)$ is a white noise, we can find that the output of PWL processor $\mathbf{s}(n)$ is also white due to the symmetric PWL function. The white property of $\mathbf{s}(n)$ can be explained as follows. Applying the PWL function, we can rewrite the correlation function of $\mathbf{s}(n)$ as $E\{s(n)s(n+m)\} = E\{f(x(n))\, f(x(n+m))\}$. When $n \neq m$, the expectation can be taken apart as $E\{f(x(n))\}E\{f(x(n+m))\}$ in case of the far end white signal $x(n)$. With an odd PWL function and an even Pdf. of the far end signal $x(n)$, the expectation of $\mathbf{s}(n)$ is zero, so is the correlation function of $\mathbf{s}(n)$ when $n \neq m$. That means $\mathbf{s}(n)$ has white property; so does $\mathbf{s}_e(n)$. Therefore,

we have the matrix $\mathbf{Q}_s$ is an identity matrix, $\mathbf{R}_s = \sigma_s^2 \mathbf{I}$ and $\mathbf{R}_{s,s_e} = \sigma_{s,s_e}^2 \mathbf{I}$, where σ_s^2 is the variance of $s(n)$ and σ_{s,s_e}^2 is the covariance of $\mathbf{s}(n)$ and $\mathbf{s}_e(n)$.

Now we may simplify Eq. (3.2.6) as

$$E\{\boldsymbol{\varepsilon}_h(n+1)\} = (1-\mu_h\sigma_s^2)E\{\boldsymbol{\varepsilon}_h(n)\} - \mu_h\sigma_{s,s_e}^2\mathbf{h}_o. \tag{3.2.7}$$

We note that $\mathbf{k}_h(n)$ is equal to $\boldsymbol{\varepsilon}_h(n)$ when the far-end signal is white.

Solving the recursive equation in Eq. (3.2.7), we may get the solution

$$E\{\boldsymbol{\varepsilon}_h(n)\} = -\frac{\sigma_{s,s_e}^2}{\sigma_s^2}\mathbf{h}_o + \left(\boldsymbol{\varepsilon}_h(0) + \frac{\sigma_{s,s_e}^2}{\sigma_s^2}\mathbf{h}_o\right)(1-\mu_h\sigma_s^2)^n. \tag{3.2.8}$$

We can see the steady state bias of $\boldsymbol{\varepsilon}_h(n)$ is a fraction of the optimal linear filter $\mathbf{h}_o$.

3.2.2 Second moment of linear coefficient weight error

With the same uncorrelated assumption of $\boldsymbol{\varepsilon}_h(n)$ and $\mathbf{s}(n)$, the second moment of $\boldsymbol{\varepsilon}_h(n)$ from Eq. (3.2.4) is given by

$$\begin{aligned} &E\left\{\left\|\boldsymbol{\varepsilon}_h(n+1)\right\|_2^2\right\} \\ &= E\left\{\boldsymbol{\varepsilon}_h^T(n)\left(1-2\mu_h\mathbf{s}(n)\cdot\mathbf{s}^T(n)+\mu_h^2E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}\right)\boldsymbol{\varepsilon}_h(n)\right\} \\ &\quad +2E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot\left(\mathrm{E}\left\{\mathbf{f}_h(n)\right\}-\mu_h\mathrm{E}\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{f}_h(n)\right\}\right)+E\left\{\left\|\mathbf{f}_h(n)\right\|_2^2\right\} \end{aligned} \tag{3.2.9}$$

From Eq. (A.8) in Appendix, the term $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}$ in Eq. (3.2.9) can be expressed as

$$E\left\{\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}\right\} = \left[\sigma_s^4 M + \left(m_{s^4} - \sigma_s^4\right)\right]\mathbf{I}$$

where $m_{s^4} = E\left\{s^4(n)\right\}$. Next, by assuming the 4th moment m_{s^4} of $s(n)$ is comparable to the power σ_s^4 of the 2nd moment of $s(n)$ and the length of the FIR filter M is sufficiently large, we can approximate it as follows:

$$E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\} \approx M\sigma_s^4\mathbf{I}. \tag{3.2.10}$$

Similarly, from Eq. (A.9) and (A.10), we have

$$\mathrm{E}\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{f}_h(n)\right\} = -\mu_h E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}\mathbf{h}_o$$

$$= -\mu_h\left(M\sigma_s^2\sigma_{s,s_e}^2 + \left(m_{s^3,s_e} - \sigma_s^2\sigma_{s,s_e}^2\right)\right)\mathbf{h}_o,$$

$$E\left\{\left\|\mathbf{f}_h(n)\right\|_2^2\right\} = \mu_h^2\left[E\left\{v(n)\mathbf{s}^T(n)\cdot\mathbf{s}(n)v(n)\right\} + \mathbf{h}_o^T\cdot E\left\{\mathbf{s}_e(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e(n)\right\}\cdot\mathbf{h}_o\right]$$

$$= \mu_h^2\left[M\sigma_s^2\sigma_v^2 + \left(M\sigma_s^2\sigma_{s_e}^2 + \left(m_{s^2s_e^2} - \sigma_s^2\sigma_{s_e}^2\right)\right)\left\|\mathbf{h}_o\right\|_2^2\right],$$

where $\sigma_{s_e}^2$ is the variance of $s_e(n)$, $m_{s^3,s_e} = E\left\{s_k^3 s_{e,k}\right\}$ and $m_{s^2,s_e^2} = E\left\{s_k^2 s_{e,k}^2\right\}$. With the approximation, m_{s^3,s_e} and m_{s^2,s_e^2} is comparable to $\sigma_s^2\sigma_{s,s_e}^2$ and $\sigma_s^2\sigma_{s_e}^2$, respectively, and $M \gg 1$, we have

$$\mathrm{E}\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{f}_h(n)\right\} \approx -\mu_h M\sigma_s^2\sigma_{s,s_e}^2\mathbf{h}_o \tag{3.2.11}$$

$$E\left\{\left\|\mathbf{f}_h(n)\right\|_2^2\right\} \approx \mu_h^2\left(M\sigma_s^2\sigma_v^2 + M\sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right), \tag{3.2.12}$$

Therefore, substituting Eq. (3.2.10), (3.2.11) and (3.2.12) into (3.2.9), we get

$$E\left\{\left\|\boldsymbol{\varepsilon}_h(n+1)\right\|_2^2\right\} \approx \left(1-2\mu_h\sigma_s^2+\mu_h^2M\sigma_s^4\right)E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} + 2E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot \mathbf{h}_o^T\mu_h\sigma_{s_e}^2\left(-1+\mu_hM\sigma_s^2\right) + \mu_h^2M\sigma_s^2\left(\sigma_v^2+\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right) \tag{3.2.13}$$

The stability of the recursion (3.2.13) is guaranteed if $\left|1-2\mu_h\sigma_s^2+\mu_h^2M\sigma_s^4\right| < 1$, from which we can get the upper bound of the step-size μ_h as

$$\mu_h < \frac{2}{M\sigma_s^2}. \tag{3.2.14}$$

This equation is the stability criterion for the linear LMS adaptive algorithm in the first stage.

We can use the linear algebra method to solve the coupled recursive Eq. (3.2.7) and (3.2.13). By cascading $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\}$ and $E\left\{\boldsymbol{\varepsilon}_h(n)\right\}$ to form a new vector, we have the recursive vector equation as

$$\boldsymbol{\theta}(n+1) = \mathbf{A}\cdot\boldsymbol{\theta}(n) + \mathbf{b}, \tag{3.2.15}$$

where

$$\mathbf{b} = \left[\mu_h^2 M \sigma_s^2 \left(\sigma_v^2 + \sigma_{s_e}^2 \left\| \mathbf{h}_o \right\|_2^2 \right) \quad \mu_h \sigma_{s,s_e}^2 \mathbf{h}_o^T \right]^T,$$

$$\boldsymbol{\theta}(n) = \left[E\left\{ \left\| \boldsymbol{\varepsilon}_h(n) \right\|_2^2 \right\} \quad E\left\{ \boldsymbol{\varepsilon}_h(n) \right\} \right]^T$$

and

$$\mathbf{A} = \begin{bmatrix} \left(1 - 2\mu_h \sigma_s^2 + \mu_h^2 M \sigma_s^4\right) & 2\mathbf{h}_o^T \mu_h \sigma_{s_e}^2 \left(-1 + \mu_h M \sigma_s^2\right) \\ \mathbf{0} & (1 - \mu_h \sigma_s^2)\mathbf{I}_M \end{bmatrix}.$$

The solution of Eq. (3.2.14) is given by

$$\boldsymbol{\theta}(n) = \left(\mathbf{I} - \mathbf{A}\right)^{-1} \cdot \mathbf{b} + \mathbf{A}^n \cdot \left(\boldsymbol{\theta}(0) - \left(\mathbf{I} - \mathbf{A}\right)^{-1} \cdot \mathbf{b} \right) \tag{3.2.16}$$

where we have assumed that $\left(\mathbf{I} - \mathbf{A}\right)$ is invertible. This assumption of invertible $\left(\mathbf{I} - \mathbf{A}\right)$ can be justified by noting that the diagonal entries of $\left(\mathbf{I} - \mathbf{A}\right)$ are all positive so long as the stability criterion in Eq. (3.2.14) is met.

The steady state $\boldsymbol{\theta}(n)$ is equal to $\left(\mathbf{I} - \mathbf{A}\right)^{-1} \cdot \mathbf{b}$, from which the steady state of $E\left\{ \left\| \boldsymbol{\varepsilon}_h(n) \right\|_2^2 \right\}$ is given by

$$\lim_{x \to \infty} E\left\{ \left\| \boldsymbol{\varepsilon}_h(n) \right\|_2^2 \right\} = \frac{\mu_h M \left(\sigma_v^2 + \sigma_{s_e}^2 \left\| \mathbf{h}_o \right\|_2^2 \right) + 2 \dfrac{\sigma_{s,s_e}^2 \sigma_{s_e}^2}{\sigma_s^4} \left\| \mathbf{h}_o \right\|_2^2 \left(1 - \mu_h M \sigma_s^2\right)}{2 - \mu_h M \sigma_s^2}. \tag{3.2.17}$$

By examining the matrix $\mathbf{A}$ in Eq. (3.2.16), we can see that the convergence rate is a monotonic increasing function of the step-size μ_h and the variance σ_s^2 of PWL processor output. Moreover, in Eq. (3.2.17) the steady-state of the 2nd moment of $\boldsymbol{\varepsilon}_h(n)$ increases with an increase of the near-end noise variance σ_v^2, the step-size μ_h, the PWL output power σ_s^2, and the nonlinearity factor σ_{s,s_e}^2 and $\sigma_{s_e}^2$. We will simulate this property in Chapter 5.

We can also deduce the compact form of the second moment of $\boldsymbol{\varepsilon}_h(n)$

iteratively. By plugging Eq. (3.2.7) into (3.2.13) and denoting $K_1 = \left(1-2\mu_h\sigma_s^2+\mu_h^2 M\sigma_s^4\right)$, $K_2 = 2\left(-\mu_h\sigma_{s_e}^2+\mu_h^2 M\sigma_s^2\sigma_{s_e}^2\right)$ and $K_3 = \mu_h^2\left(M\sigma_s^2\sigma_v^2+M\sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right)$, we have

$$
\begin{aligned}
E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} &\approx K_1 E\left\{\left\|\boldsymbol{\varepsilon}_h(n-1)\right\|_2^2\right\}+K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(n-1)\right\}\cdot+K_3 \\
&= K_1\left[E\left\{\left\|\boldsymbol{\varepsilon}_h(n-2)\right\|_2^2\right\}+K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(n-2)\right\}+K_3\right] \\
&\quad +K_2\mathbf{h}_o^T\left[(1-\mu_h\sigma_s^2)E\left\{\boldsymbol{\varepsilon}_h(n-2)\right\}-\mu_h\sigma_{s,s_e}^2\mathbf{h}_o\right]+K_3 \\
&= K_1^{n-1}\left[K_1 E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\}+K_2\mathbf{h}_o^T\cdot E\left\{\boldsymbol{\varepsilon}_h(n-2)\right\}+K_3\right] \\
&\quad +\left[\cdots\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]\cdots(1-\mu_h\sigma_s^2)+K_1^{n-2}\right]K_2\mathbf{h}_o^T\cdot \\
&\quad \left[(1-\mu_h\sigma_s^2)E\left\{\boldsymbol{\varepsilon}_h(0)\right\}-\mu_h\sigma_{s,s_e}^2\mathbf{h}_o\right]+\left[1+K_1+K_1^2+\cdots+K_1^{n-2}\right]K_3 \\
&\quad -\left\{1+\left[(1-\mu_h\sigma_s^2)+K_1\right]+\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]+\cdots+\right. \\
&\quad \left.\left[\cdots\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]\cdots(1-\mu_h\sigma_s^2)+K_1^{n-3}\right]\right\}\mu_h\sigma_{s,s_e}^2K_2\left\|\mathbf{h}_o\right\|_2^2
\end{aligned}
$$

$$
\begin{aligned}
&= K_1^n E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\}+\left[\cdots\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]\cdots(1-\mu_h\sigma_s^2)+K_1^{n-1}\right] \\
&\quad K_2\mathbf{h}_o^T\cdot\left[(1-\mu_h\sigma_s^2)E\left\{\boldsymbol{\varepsilon}_h(0)\right\}-\mu_h\sigma_{s,s_e}^2\mathbf{h}_o\right]+K_3\left[1+K_1+K_1^2+\cdots+K_1^{n-2}\right] \\
&\quad -\left\{1+\left[(1-\mu_h\sigma_s^2)+K_1\right]+\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]+\cdots+\right. \\
&\quad \left.\left[\cdots\left[\left[(1-\mu_h\sigma_s^2)+K_1\right](1-\mu_h\sigma_s^2)+K_1^2\right]\cdots(1-\mu_h\sigma_s^2)+K_1^{n-2}\right]\right\}\mu_h\sigma_{s,s_e}^2K_2\left\|\mathbf{h}_o\right\|_2^2
\end{aligned}
$$

Using the geometric series formula, it can be simplified as:

$$
\begin{aligned}
E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} &\approx K_1^n E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\}+\frac{(1-\mu_h\sigma_s^2)^n-K_1^n}{(1-\mu_h\sigma_s^2)-K_1}K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(0)\right\}-\left[1+\frac{(1-\mu_h\sigma_s^2)^2-K_1^2}{(1-\mu_h\sigma_s^2)-K_1}\right. \\
&\quad \left.+\frac{(1-\mu_h\sigma_s^2)^3-K_1^3}{(1-\mu_h\sigma_s^2)-K_1}+\cdots+\frac{(1-\mu_h\sigma_s^2)^{n-1}-K_1^{n-1}}{(1-\mu_h\sigma_s^2)-K_1}\right]\mu_h\sigma_{s,s_e}^2K_2\left\|\mathbf{h}_o\right\|_2^2+\frac{1-K_1^n}{1-K_1}K_3 \\
&= K_1^n E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\}+\frac{(1-\mu_h\sigma_s^2)^n-K_1^n}{(1-\mu_h\sigma_s^2)-K_1}K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(0)\right\}-\frac{1}{(1-\mu_h\sigma_s^2)-K_1} \\
&\quad \left[\frac{1-(1-\mu_h\sigma_s^2)^n}{1-(1-\mu_h\sigma_s^2)}-\frac{1-K_1^n}{1-K_1}\right]\mu_h\sigma_{s,s_e}^2K_2\left\|\mathbf{h}_o\right\|_2^2+\frac{1-K_1^n}{1-K_1}K_3
\end{aligned}
$$

$$E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} \approx \frac{K_3}{1-K_1} - \frac{1}{(1-\mu_h\sigma_s^2)-K_1}\left[\frac{1}{\mu_h\sigma_s^2} - \frac{1}{1-K_1}\right]\mu_h\sigma_{s,s_e}^2 K_2 \left\|\mathbf{h}_o\right\|_2^2$$
$$+ (1-\mu_h\sigma_s^2)^n\left[\frac{K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(0)\right\}}{(1-\mu_h\sigma_s^2)-K_1} + \frac{\mu_h\sigma_{s,s_e}^2 K_2\left\|\mathbf{h}_o\right\|_2^2}{\mu_h\sigma_s^2\left[(1-\mu_h\sigma_s^2)-K_1\right]}\right] +$$
$$+ K_1^n\left[E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\} - \frac{K_2\mathbf{h}_o^T E\left\{\boldsymbol{\varepsilon}_h(0)\right\}}{(1-\mu_h\sigma_s^2)-K_1} - \frac{\mu_h\sigma_{s,s_e}^2 K_2\left\|\mathbf{h}_o\right\|_2^2}{(1-K_1)\left[(1-\mu_h\sigma_s^2)-K_1\right]} - \frac{K_3}{1-K_1}\right] \tag{3.2.18}$$

The expression of the 2nd moment of linear coefficient weight error in Eq. (3.2.18) appears to be tedious, as compared to the compact vector form in Eq. (3.2.16).

Let us consider the special case of perfect PWL coefficients. The second moment of $\boldsymbol{\varepsilon}_{h,i}(n)$ can be easily obtained by setting the nonlinear coefficient weight error $\boldsymbol{\varepsilon}_{\mathbf{w}} = 0$ so that $A_2 = A_3 = 0$ i.e., $K_1 = \dfrac{\mu_h M\sigma_v^2}{2-\mu_h M\sigma_s^2}$ and $K_2 = 0$. The Eq. (3.2.18) reduces to

$$E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} \approx \frac{K_3}{1-K_1} + K_1^n\left[E\left\{\left\|\boldsymbol{\varepsilon}_h(0)\right\|_2^2\right\} - \frac{K_3}{1-K_1}\right]$$

which is a well known result [3].

Finally, after derivation of first and second moments of the linear coefficient weight error, we turn our attention to the residual output power. From Eq. (3.2.2), the mean square error (i.e., residual error) is given by

$$\begin{aligned} J_h(n) &= E\left\{\left|e(n)\right|^2\right\} \\ &= \sigma_v^2 + \mathbf{h}_o^T \mathrm{E}\left\{\mathbf{s}_e(n)\cdot\mathbf{s}_e^T(n)\right\}\mathbf{h}_o + E\left\{\boldsymbol{\varepsilon}_h^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\boldsymbol{\varepsilon}_h(n)\right\} \\ &\quad + 2\mathbf{h}_o^T\cdot\mathrm{E}\left\{\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}\cdot E\left\{\boldsymbol{\varepsilon}_{\mathbf{h}}(n)\right\}. \end{aligned} \tag{3.2.19}$$

Because the variation of $\boldsymbol{\varepsilon}_h(n)$ is slow compared to $\mathbf{s}(n)$, hence

$$E\left\{\boldsymbol{\varepsilon}_h^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\boldsymbol{\varepsilon}_h(n)\right\} = \sigma_s^2 E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} \tag{3.2.20}$$

From Eq. (3.2.19) and (3.2.20), the mean square error can be written as

$$J_h(n) = \sigma_v^2 + \sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2 + \sigma_s^2 E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} + 2\sigma_{s,s_e}^2 \mathbf{h}_o^T \cdot E\left\{\boldsymbol{\varepsilon}_\mathbf{h}(n)\right\}. \quad (3.2.21)$$

which depends on the $E\left\{\boldsymbol{\varepsilon}_\mathbf{h}(n)\right\}$ and the . $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\}$ that are derived earlier in Eq. (3.2.7) and (3.2.18) or the compact Eq. (3.2.16).

3.3 Convergence analysis of joint adaptation of linear and PWL coefficients

After the convergence of linear coefficients, the nonlinear adaptive filter switches to the 2nd stage in which linear and PWL coefficients will be updated jointly. Now, the residual error is given by

$$e(n) = v(n) - \mathbf{w}_o^T \cdot \mathbf{F}^T(n) \cdot \boldsymbol{\varepsilon}_h(n) - \boldsymbol{\varepsilon}_w^T(n) \cdot \mathbf{F}^T(n) \cdot \mathbf{h}_o - \boldsymbol{\varepsilon}_w^T(n)\mathbf{F}^T(n)\boldsymbol{\varepsilon}_h(n) \quad (3.3.1)$$

The coupled linear and nonlinear weight error in the fourth term of Eq. (3.3.1) renders difficulty in convergence analysis. However, with wide band signal like speech, loudspeaker nonlinearities are much less dominant than the linear components in general. Therefore, we can assume initial PWL weight error is much smaller than optimal PWL coefficients. Moreover, the converged linear coefficients would be approximately to optimal linear filter, namely,

$$\boldsymbol{\varepsilon}_h(n) << \mathbf{h}_o\,, \quad \boldsymbol{\varepsilon}_w(n) << \mathbf{w}_o\,, \quad (3.3.2)$$

where $\boldsymbol{\varepsilon}_w(n) = \mathbf{w}(n) - \mathbf{w}_o$. With sufficiently small perturbation errors in linear and nonlinear coefficients in Eq. (3.3.2), the 2nd-order perturbation term can be discarded and the estimation error becomes

$$e(n) \approx v(n) - \mathbf{w}_o^T \cdot \mathbf{F}^T(n) \cdot \boldsymbol{\varepsilon}_h(n) - \boldsymbol{\varepsilon}_w^T(n) \cdot \mathbf{F}^T(n) \cdot \mathbf{h}_o \quad (3.3.3)$$

Denote the combined linear and PWL coefficient weight error as $\boldsymbol{\varepsilon}(n) = \left[\boldsymbol{\varepsilon}_h(n) \quad \boldsymbol{\varepsilon}_w(n)\right]^T$ and let

$$\mathbf{G}(n) = \left[\mathbf{w}_o^T \cdot \mathbf{F}^T(n) \quad \mathbf{F}^T(n) \cdot \mathbf{h}_o \right]^T, \tag{3.3.4}$$

then Eq. (3.3.3) can be expressed as follows:

$$e(n) \approx v(n) - \mathbf{G}^T(n) \cdot \boldsymbol{\varepsilon}(n). \tag{3.3.5}$$

By substituting by (2.1.6), (2.1.7) and (3.1.5) into the combined coefficient weight error

$$\boldsymbol{\varepsilon}(n+1) = \begin{bmatrix} \mathbf{h}(n+1) \\ \mathbf{w}(n+1) \end{bmatrix} - \begin{bmatrix} \mathbf{h}_o \\ \mathbf{w}_o \end{bmatrix},$$

we have

$$\boldsymbol{\varepsilon}(n+1) = \boldsymbol{\varepsilon}(n) + \mathbf{T} \cdot \begin{bmatrix} \mathbf{F}(n) \cdot \left(\boldsymbol{\varepsilon}_w(n) - \mathbf{w}_o \right) \\ \mathbf{F}^T(n) \cdot \left(\mathbf{h}_o - \boldsymbol{\varepsilon}_h(n) \right) \end{bmatrix} \left(v(n) - \mathbf{G}^T(n) \cdot \boldsymbol{\varepsilon}(n) \right)$$

where $\mathbf{T} = \begin{bmatrix} \mu_h \mathbf{I}_M & \mathbf{O} \\ \mathbf{O} & \mu_w \mathbf{I}_N \end{bmatrix}$. According to the assumption in (3.3.2), it can be approximated as

$$\boldsymbol{\varepsilon}(n+1) \approx \left(\mathbf{I} - \mathbf{T} \cdot \mathbf{G}(n) \cdot \mathbf{G}^T(n) \right) \cdot \boldsymbol{\varepsilon}(n) - v(n) \mathbf{T} \cdot \mathbf{G}(n) \tag{3.3.6}$$

Due to the independent assumption of near end noise $v(n)$, we can apply the same procedure in Section 4.2 to derive the solution of first moment of $\boldsymbol{\varepsilon}(n+1)$ from Eq. (3.3.6) as follows:

$$E\{\boldsymbol{\varepsilon}(n)\} \approx \left(\mathbf{I} - \mathbf{T} \cdot \mathbf{R}_{\mathbf{G}} \right)^N \cdot E\{\boldsymbol{\varepsilon}(0)\},$$

where $\mathbf{R}_{\mathbf{G}}$ is the correlation matrix of $\mathbf{G}(n)$. With a suitable step-size, the magnitude of this geometric ratio must be less than unity for all *n*, we can see that the estimate is unbiased.

Similarly, the second moment of $\boldsymbol{\varepsilon}(n)$ is given by

$$E\left\{ \left\| \boldsymbol{\varepsilon}(n+1) \right\|_2^2 \right\} = E\left\{ \boldsymbol{\varepsilon}_h^T(n) \left(1 - 2\mathbf{T} \cdot \mathbf{R}_{\mathbf{G}} + \mathbf{T}^2 E\left\{ \mathbf{G}(n) \cdot \mathbf{G}^T(n) \cdot \mathbf{G}(n) \cdot \mathbf{G}^T(n) \right\} \right) \boldsymbol{\varepsilon}_h(n) \right\}$$
$$+ E\left\{ \left\| v(n) \mathbf{T} \cdot \mathbf{G}(n) \right\|_2^2 \right\}.$$

Here, we assume the term $E\left\{\mathbf{G}(n)\cdot\mathbf{G}^T(n)\cdot\mathbf{G}(n)\cdot\mathbf{G}^T(n)\right\}$ can be approximated as $\mathbf{R}_G^2$. By applying the unitary similarity transformation, $\mathbf{R}_G$ is transformed into a simpler form: $\mathbf{Q}^T\cdot\mathbf{R}_G\cdot\mathbf{Q}=\mathbf{D}$, where $\mathbf{Q}$ is an unitary matrix and $\mathbf{D}$ is a diagonal matrix consisting of the eigenvalues λ_i of $\mathbf{R}_G$. Letting $\mathbf{K}(n)=\mathbf{Q}^T\cdot\boldsymbol{\varepsilon}(n)$, we can deduce the second moment of $\boldsymbol{\varepsilon}(n)$ and mean square error $J(n)$ as follows:

$$E\left\{\left\|\boldsymbol{\varepsilon}(n)\right\|_2^2\right\}=\sum_{i=1}^{M+N}\frac{\sigma_v^2T_i\lambda_i}{2-T_i\lambda_i}+\left[\left|k_i(0)\right|^2-\frac{T_i\sigma_v^2}{2-T_i\lambda_i}\right](1-T_i\lambda_i)^{2n}, \tag{3.3.7}$$

$$J(n)=\sigma_v^2+\sum_{i=1}^{M+N}\frac{\sigma_v^2T_i\lambda_i}{2-T_i\lambda_i}+\lambda_i\left[\left|k_i(0)\right|^2-\frac{T_i\sigma_v^2}{2-T_i\lambda_i}\right](1-T_i\lambda_i)^{2n}, \tag{3.3.8}$$

where T_i is the *i*-th diagonal entry of $\mathbf{T}$ and $k_i(0)$ is the initial value of *i*-th entry of $\mathbf{K}(n)$.

Now that we have derived the theoretical convergence analysis of the two-staged PWL algorithm. Similar results also hold in case of a polynomial structure simply by setting the delayed tap mapping matrix $\mathbf{F}(n)$ as

$$\mathbf{F}(n)=\begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M+1)\\ x^2(n) & x^2(n-1) & \cdots & x^2(n-M+1)\\ & & \vdots & \\ x^N(n) & x^N(n-1) & \cdots & x^N(n-M+1)\end{bmatrix}.$$

However, with a different delayed tap mapping matrix $\mathbf{F}(n)$, the convergence behavior of the nonlinear AEC is altered. In the case of a polynomial structure, polynomial output power σ_s^2 and nonlinearity factors, σ_{s,s_e}^2 and $\sigma_{s_e}^2$, in Eq. (3.2.15), (3.2.20), and the eigenvalues λ_i of the correlation matrix $\mathbf{R_G}$ in Eq. (3.3.6) and (3.3.7) will be different from those of a PWL structure.

3.4 Summary

Because of the divergence problem encountered in cascade system, we adopt the two-staged algorithm to overcome this problem in Section 3.1. Moreover, we derived the convergence analysis of the two-staged algorithm.

In Section 3.2 we perform the convergence analysis and stability criterion of the first stage. By examining the convergence analysis, we indeed discuss how the factors: the near-end noise variance σ_v^2, the step-size μ_h, the PWL output power σ_s^2, and the nonlinearity factor σ_{s,s_e}^2 and $\sigma_{s_e}^2$ do affect it. After that, we derived the convergence analysis of the second stage in Section 3.3. The above convergence analysis also can be extended to a case of polynomial based Nonlinear AEC.

In the next Chapter, we will use the convergence analysis of the first stage to develop the optimum time-variant and optimum time-&tap-variant step-size.

Chapter 4
Step-size Control for Nonlinear AEC

In Chapter 3, theoretical analysis indicates that its transient residual error power depends on (1) the step-size parameter (2) the eigenvalues of the corresponding correlation matrix, (3) the initial tap coefficient setting, (4) the number of iterations of the algorithm.

In this Chapter, we will focus on the step-size. Due to the tradeoff between fast convergence rate and small residual error power, we cannot find an optimal step-size which provides the best performances of both. That is a large step size gives a faster convergence but also large small residual error power. Therefore, various methods employing varying step-size have be examined by the other researchers, including time-varying [17], tap-varying [18] or both time- & tap- varying [19].

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.

For comparison with the convention studies, our work is under a nonlinear system, memoryless PWL processor cascading a linear FIR filter. Due to the existing error on PWL processor, the influence of nonlinearity, caused by the loudspeaker, is an factor of the step size of the FIR linear filter.

In the following section, we will derive the optimal time-varying step size LMS (OTLMS) algorithm in Section 4.1. Its practical implement with the monotonously decreasing slope model of nonlinear I/O curve will be discussed in Section 4.2. Further, we will also derive the optimal time- & tap-varying step size LMS (OTTLMS)

and give the general iterative equation for the tap coefficients error variance in Section 4.3. In the same way, not only the monotonously decreasing slope model of nonlinear I/O curve but also the model of the room impulse response will be used to accomplish the practical implement in Section 4.4.

4.1 Derivation of optimum time- variant step-size LMS (OTLMS) algorithm

Instead of using a constant step-size, the time varying one, all taps used identical step-size which is time-variant, is based on using large step-size when the linear FIR filter is far from the Wiener solution, thus speeding up the convergence rate and when it is near the optimum, small step size is used to achieve lower MSE. Overall mechanism will obtain a better performance than a constant step-size one.

Under a nonlinear system, memoryless PWL processor cascading a linear FIR filter, the influence of nonlinearity, caused by the loudspeaker, will let the linear FIR filter converge to a bias estimate due to the existing error on PWL processor. Therefore, the cost function i.e., residual error power, can not simplify to the norm of the linear FIR coefficient error. As well as Eq. (3.2.21), it contains the term with the first order and second order of the FIR coefficient error. Here we rewrite the first moment of linear coefficient weight error Eq. (3.2.7) and the second of one (3.2.13) by using a time varying step-size as follows:

$$E\{\boldsymbol{\varepsilon}_h(n+1)\} = \left(1-\mu_h(n)\sigma_s^2\right)E\{\boldsymbol{\varepsilon}_h(n)\} - \mu_h(n)\sigma_{s,s_e}^2\mathbf{h}_o \tag{4.1.1}$$

$$E\left\{\left\|\boldsymbol{\varepsilon}_h(n+1)\right\|_2^2\right\} \approx \left(1-2\mu_h(n)\sigma_s^2+\mu_h^2(n)M\sigma_s^4\right)E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} + 2E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot \mathbf{h}_0\left(-\mu_h(n)\sigma_{s_e}^2+\mu_h^2(n)M\sigma_s^2\sigma_{s_e}^2\right)+\mu_h^2(n)\left(M\sigma_s^2\sigma_v^2+M\sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right). \tag{4.1.2}$$

Substituting Eq. (4.1.1) and (4.1.2) into Eq. (3.2.20), we get $J_h(n+1)$ as

$$J_h(n+1) = \sigma_v^2 + \sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2 + 2\sigma_{s,s_e}^2 \mathbf{h}_o^T \cdot E\left\{\boldsymbol{\varepsilon}_\mathbf{h}(n+1)\right\} + \sigma_s^2 E\left\{\left\|\boldsymbol{\varepsilon}_h(n+1)\right\|_2^2\right\}$$

$$= \sigma_v^2 + \sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2 + 2\sigma_{s,s_e}^2 \mathbf{h}_o^T \cdot \left[\left(1-\mu_h(n)\sigma_s^2\right) E\left\{\boldsymbol{\varepsilon}_h(n)\right\} - \mu_h(n)\sigma_{s,s_e}^2 \mathbf{h}_o\right]$$
$$+ \sigma_s^2 \left[\left(1-2\mu_h(n)\sigma_s^2 + \mu_h^2(n)M\sigma_s^4\right) E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} + 2E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\right.$$
$$\left. \cdot \mathbf{h}_0 \left(-\mu_h(n)\sigma_{s_e}^2 + \mu_h^2(n)M\sigma_s^2\sigma_{s_e}^2\right) + \mu_h^2(n)\left(M\sigma_s^2\sigma_v^2 + M\sigma_s^2\sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2\right)\right]$$

Indeed, rewrite it in terms of the residual error power $J_h(n)$ as

$$J_h(n+1) = J_h(n) + 2\mu_h(n)\left[-2\sigma_{s,s_e}^2\sigma_s^2\mathbf{h}_o^T \cdot \left\{\boldsymbol{\varepsilon}_h(n)\right\} - \sigma_s^4 E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} - \sigma_{s,s_e}^4 \left\|\mathbf{h}_o\right\|_2^2\right]$$
$$+ \mu_h^2(n)\left[2M\sigma_{s,s_e}^4\sigma_s^2\mathbf{h}_o^T \cdot \left\{\boldsymbol{\varepsilon}_h(n)\right\} + M\sigma_s^6 E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} + M\left(\sigma_s^2\sigma_v^2 + \sigma_s^2\sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2\right)\right]$$

$$= J_h(n) + \mu_h(n)\left[-\sigma_s^2 J_h(n) + \sigma_s^2\sigma_v^2 + \sigma_{s,s_e}^2\sigma_s^2 \left\|\mathbf{h}_o\right\|_2^2 - \sigma_{s,s_e}^4 \left\|\mathbf{h}_o\right\|_2^2\right]$$
$$+ \mu_h^2(n)M\sigma_s^4 J_h(n) \qquad (4.1.3)$$

The optimal time varying step-size can be obtained by taking derivative of Eq. (4.1.3) with respect to $\mu_h(n)$ and setting the result equal to zero.

$$\frac{\partial J(n)}{\partial \mu_h(n)} = 0$$

Thus we can get the optimal time varying step-size

$$\mu_{h,OLMS}(n) = \frac{\sigma_s^2 J_h(n) - \sigma_s^2\sigma_v^2 - \left(\sigma_s^2\sigma_{s_e}^2 - \sigma_{s,s_e}^4\right)\left\|\mathbf{h}_o\right\|_2^2}{M\sigma_s^4 J_h(n)} \qquad (4.1.4)$$

To simply the form of optimal step size residual error power, we substitute Eq. (4.1.4) into (4.1.3), we can get

$$J_h(n+1) = \left(1-\mu_{h,OLMS}(n)\sigma_s^2\right) J_h(n) + \mu_{h,OLMS}(n)\left[\sigma_s^2\sigma_v^2 + \left(\sigma_s^2\sigma_{s_e}^2 - \sigma_{s,s_e}^4\right)\left\|\mathbf{h}_o\right\|_2^2\right]. \quad (4.1.5)$$

Letting $\beta = \sigma_v^2 + \left(\sigma_{s_e}^2 - \frac{\sigma_{s,s_e}^4}{\sigma_s^2}\right)\left\|\mathbf{h}_o\right\|_2^2$, the OTLMS algorithm for the first stage of two-staged adaptation is summarized by the following equations:

$$e(n) = d(n) - \mathbf{w}^T \cdot \mathbf{F}^T(n) \cdot \mathbf{h}(n)$$

$$\mu_{h,OLMS}(n) = \frac{1}{M\sigma_s^2} - \frac{\beta}{M\sigma_s^2 J_h(n)} \qquad (4.1.6)$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu_h \mathbf{s}(n)e(n)$$

$$J_h(n+1) = \left(1 - \mu_{h,OLMS}(n)\sigma_s^2\right) J(n) + \mu_{h,OLMS}(n)\sigma_s^2 \beta \qquad (4.1.7)$$

The step-size adjustment is based on (2) and (4). If null nonlinearity, the parameter β can be modified as $\beta = \sigma_v^2$.

4.2 Practical implementations of OTLMS algorithm

We have derived Optimum time variant step-size for LMS algorithm in Eq. (4.1.4). But it is useless since we require prior statistics knowledge $\sigma_s^2, \sigma_{s,s_e}^2$ and σ_{s,s_e}^2 of the nonlinearity the second norm $\|\mathbf{h}_o\|_2^2$ of the exponential decay model of RIR $\mathbf{h}_o$. We can use the monotonously decreasing slope model of nonlinear I/O curve and assume $\|\mathbf{h}_o\|_2^2 = 1$ on β for practical implementation. Moreover, the regression of expectation of residual error power in Eq. (4.1.5) also can be replaced by time mean estimate with first-order recursive filtering

$$\hat{J}_h(n+1) = (1-\lambda)\hat{J}(n) + \lambda e^2(n), \qquad (4.2.1)$$

where λ is the forgetting parameter close to 0. We will discuss these two recursive Eq. (4.1.5) and (4.2.1) of residual error power for practical implementation in the chapter 5

Here we will explain the monotonously decreasing slope model of nonlinear I/O mapping curve explicitly in the following discussion. As we know, the PWL function approximating the nonlinearity by using several linear affine descriptions is a very old and often used method. By doing this, the basic problem is transformed from a single nonlinear equation into several linear equations. Approximation of nonlinear I/O curve can be described simply and flexibly by several slopes. Therefore, we can use the idea to model the nonlinearity of loudspeaker and get its prior statistics

knowledge σ_s^2, σ_{s,s_e}^2 and σ_{s,s_e}^2. Furthermore, the harmonic distortion introduced by loudspeaker and their amplifiers was proportioned to the power of input [7] i.e., the harmonic distortion is larger with higher volume of input. In the nonlinear I/O curve, we can found the higher volume area will saturate much seriously, i.e., the corresponding slope is smaller. It means the slope of each area is monotonically decreasing to the power of input. The property of nonlinear I/O mapping curve is monotonously decreasing slope.

From pervious discussion in chapter 2, the CPWL function exploits the difference between the adjacent slopes to approximate the nonlinear curve. For simplicity, we assume the difference of the adjacent slopes is the same with a uniform partition. Although it is a special case, the assumption can lessen the CPWL coefficients to be only two ones: initial slope γ_0 (the first area) and difference of the adjacent slopes γ_d. Hence, the optimal CPWL coefficients $w_{o,1}, w_{o,2} \cdots w_{o,N}$ turn into $\gamma_o, \gamma_d \cdots \gamma_d$ where $\gamma_o \approx w_{o,1}$ and γ_d should be negative due to the monotonously decreasing property of slope

Here, we use an example to demonstrate the model. The nonlinear I/O mapping curve is a nonlinear transformation of raised-cosine function as given below [23]

$$f(x) = \begin{cases} -1 & , x < -\dfrac{1+\xi}{2T} \\ Tx - \dfrac{1-\xi}{2} + \dfrac{\pi}{\xi}\cos\left(\dfrac{2Tx\pi + \pi}{2\xi}\right) & , -\dfrac{1+\xi}{2T} \le x \le -\dfrac{1-\xi}{2T} \\ 2Tx & , -\dfrac{1-\xi}{2T} \le x \le \dfrac{1-\xi}{2T} \\ Tx + \dfrac{1-\xi}{2} - \dfrac{\pi}{\xi}\cos\left(\dfrac{2Tx\pi - \pi}{2\xi}\right) & , \dfrac{1-\xi}{2T} \le x \le \dfrac{1+\xi}{2T} \\ 1 & , x > \dfrac{1+\xi}{2T}. \end{cases}$$

We use a soft-clipping type $\xi = 1, T = 1$ which is similar to sigmoid function [24].

Further, nonlinear transformation of raised-cosine function can be simplified as follow:

$$f(x)=\begin{cases} -1 & ,x<-1 \\ x+\pi\cos\left(\dfrac{2x\pi+\pi}{2}\right) & ,0\le x\le -1 \\ x-\pi\cos\left(\dfrac{2x\pi-\pi}{2}\right) & ,0\le x\le 1 \\ 1 & ,x<-1. \end{cases}$$

In the CPWL function with a uniform partition on [0 0.33 0.66 1] we choose the initial slope $\gamma_o = 1.9$ and the difference of the adjacent slopes $\gamma_d = -0.4$. Further, we plot both of them in figure 4.1. W can see the model by CPWL function is fitting well to the true nonlinear I/O mapping curve with suitable parameters γ_o and γ_d.

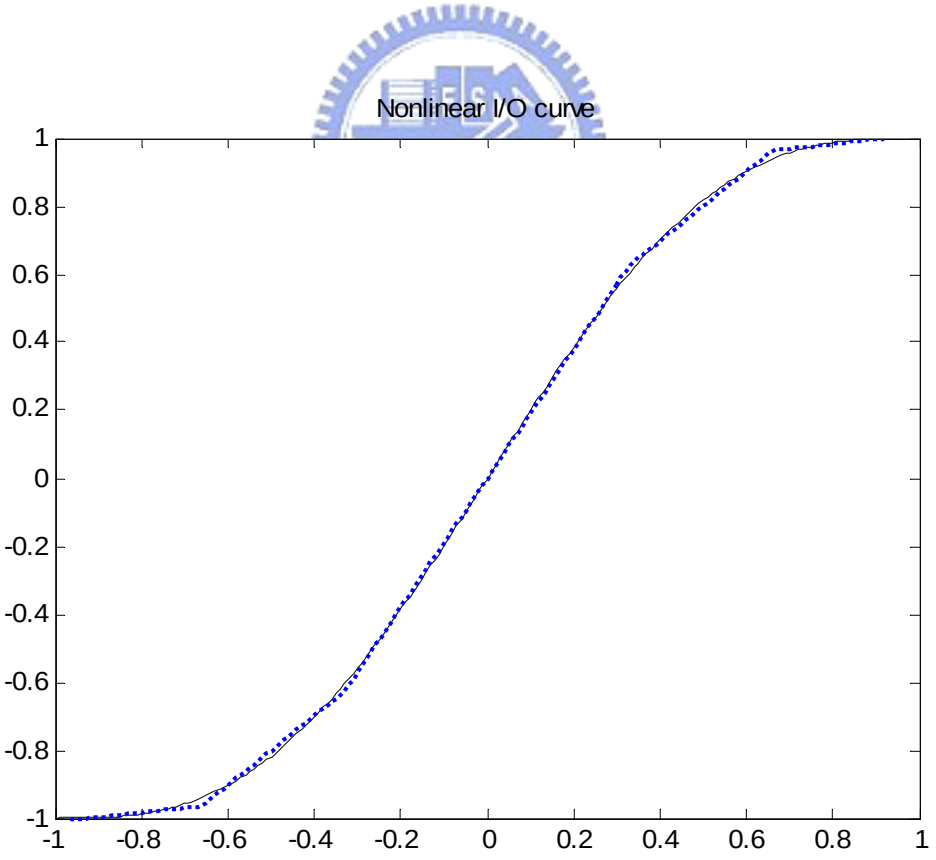

Fig. 4.1 True nonlinear I/O mapping curve (solid), CPWL function (dot)

4.3 Derivation of optimum time-&tap-variant step-size LMS (OTTLMS) algorithm

In section 4.1, it uses identical step-size for all taps. However, we know the fact that the expected variation of a room impulse response becomes progressively smaller along the series by the same exponential ratio as the impulse response energy decay. As a result, the algorithm adjusts taps with large errors in large steps, and taps with small errors in small steps. In this section, we want to derive the optimal step-size on individual tap of linear filter that minimize the residual error power for each iteration step.

Here, we cannot use the convergence analysis in chapter 3, the second norm of linear weight error, to accomplish our purpose due to individual tap. We have to derive the each tap coefficient error variance. Hence, we use a diagonal matrix $\mathbf{U}(n)$, with diagonal entry $\mu_{h,k}(n),\ \forall k = 0 \sim M\text{-}1$, to replace the step size and its corresponding LMS algorithm can be rewritten as

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mathbf{U}(n)\mathbf{s}(n)e(n)$$

$$e(n) = d(n) - \mathbf{s}^T(n)\cdot\mathbf{h}\ .$$

The following discussion will follow the uncorrelated assumption of $\boldsymbol{\varepsilon}_h(n)$ and $\mathbf{s}(n)$ and assume the far-end signal is white. Applying the same procedure as Section 3.2, the linear filter weight error can be given by

$$\boldsymbol{\varepsilon}_h(n+1) = \left[\mathbf{I} - \mathbf{U}(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]\cdot\boldsymbol{\varepsilon}_h(n) + \mathbf{U}(n)\left[v(n)\mathbf{s}(n) - \mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right] \quad (4.3.1)$$

With the same assumption, $\boldsymbol{\varepsilon}_h(n)$ is uncorrelated to $\mathbf{s}(n)$, we can get the correlation matrix of the linear filter weight error

$$
\begin{aligned}
&\mathbf{R}_h(n+1) = E\left\{\boldsymbol{\varepsilon}_h(n+1)\boldsymbol{\varepsilon}_h^T(n+1)\right\} \\
&= E\left\{\left[\mathbf{I}-\mathbf{U}(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]\cdot E\left\{\boldsymbol{\varepsilon}_h(n)\cdot\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot\left[\mathbf{I}-\mathbf{U}(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]^T\right\} \\
&\quad + E\left\{\left[\mathbf{I}-\mathbf{U}(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]\cdot E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{U}(n)\cdot\left[v(n)\mathbf{s}(n)-\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right]^T\right\} \\
&\quad + E\left\{\mathbf{U}(n)\cdot\left[v(n)\mathbf{s}(n)-\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right]\cdot E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot\left[\mathbf{I}-\mathbf{U}(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right]^T\right\} \\
&\quad + E\left\{\mathbf{U}^2(n)\cdot\left[v(n)\mathbf{s}(n)-\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right]\cdot\left[v(n)\mathbf{s}(n)-\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\right]^T\right\}
\end{aligned}
$$

Because of the independence to $\mathbf{s}(n)$ and zero mean of the assumption of the noise $v(n)$, we have

$$
\begin{aligned}
\mathbf{R}_h(n+1) &= \mathbf{R}_h(n) - 2\mathbf{U}(n)\cdot\mathbf{R}_s\cdot\mathbf{R}_h(n) + \mathbf{U}^2(n)\cdot E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{R}_h(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\} \\
&\quad - E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\cdot\mathbf{R}_{s,s_e} + \mathbf{U}^2(n)\cdot E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\} \\
&\quad - \mathbf{R}_{s,s_e}^T\cdot\mathbf{h}_o\cdot E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\} + \mathbf{U}^2(n)\cdot E\left\{\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\cdot\mathbf{h}_o\cdot E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\} \\
&\quad + \sigma_v^2\mathbf{U}^2(n)\cdot\mathbf{R}_s + \mathbf{U}^2(n)E\left\{\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\cdot\mathbf{h}_o^T\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}
\end{aligned}
\tag{4.3.2}
$$

From Appendix Eq. (A.2), (A.5) and (A.6), we can derive Eq. (4.3.2) in the terms of its diagonal entry $g_k(n)$ as follows:

$$
\begin{aligned}
g_k(n+1) &= \left[1-2\mu_{h,k}(n)\sigma_s^2 + \mu_{h,k}^2(n)\left(m_{s,4}-\sigma_s^4\right)\right]g_k(n) + \mu_{h,k}^2(n)\sigma_s^4 trace\left(\mathbf{R}_h(n)\right) \\
&\quad + 2\left[-\mu_{h,k}(n)\sigma_{s,s_e}^2 E\left\{\varepsilon_{h,k}(n)\right\}h_{o,k} + \mu_{h,k}^2(n)\left(m_{s^3,s_e}-\sigma_s^2\sigma_{s,s_e}^2\right)E\left\{\varepsilon_{h,k}(n)\right\}h_{o,k}\right. \\
&\quad \left.+\sigma_s^2\sigma_{s,s_e}^2 E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o\right] + \mu_{h,k}^2(n)\left[\sigma_v^2\sigma_s^2 + \left(m_{s^2s_e^2}-\sigma_s^2\sigma_{s_e}^2\right)h_{o,k}^2 + \sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right],
\end{aligned}
\tag{4.3.3}
$$

where $\mu_{h,k}(n)$ is the *k*-th diagonal entry of $\mathbf{U}(n)$, the step-size of the *k*-th tap.

Substituting Eq. (4.3.1) and $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} = \sum_{k=0}^{M-1} g_k(n)$ with Eq. (4.3.3) into Eq. (3.2.20), we have

$$
J_h(n+1) = \sigma_v^2 + \sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2 + 2\sigma_{s,s_e}^2\sum_{k=0}^{M-1}h_{o,k}\cdot E\left\{\varepsilon_{h,k}(n+1)\right\} + \sigma_s^2\sum_{k=0}^{M-1}g_k(n+1)
$$

$$= \sigma_v^2 + \sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2 + 2\sigma_{s,s_e}^2 \sum_{k=0}^{M-1} h_{o,k} \cdot \left(1 - \mu_{h,k}(n)\sigma_s^2\right) E\left\{\varepsilon_{h,k}(n)\right\} - \mu_{h,k}(n)\sigma_{s,s_e}^2 h_{o,k}^2$$
$$+ \sigma_s^2 \sum_{k=0}^{M-1} \left[1 - 2\mu_{h,k}(n)\sigma_s^2 + \mu_{h,k}^2(n)\left(m_{s^4} - \sigma_s^4\right)\right] g_k(n) + \mu_{h,k}^2(n)\sigma_s^4 trace\left(\mathbf{R}_h(n)\right)$$
$$+ 2\left[-\mu_{h,k}(n)\sigma_{s,s_e}^2 E\left\{\varepsilon_{h,k}(n)\right\} h_{o,k} + \mu_{h,k}^2(n)\left(m_{s^3,s_e} - \sigma_s^2\sigma_{s,s_e}^2\right) E\left\{\varepsilon_{h,k}(n)\right\} h_{o,k}\right.$$
$$\left. + \sigma_s^2\sigma_{s,s_e}^2 E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o\right] + \mu_{h,k}^2(n)\left[\sigma_v^2\sigma_s^2 + \left(m_{s^2s_e^2} - \sigma_s^2\sigma_{s_e}^2\right) h_{o,k}^2 + \sigma_s^2\sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2\right]$$

$$= J_h(n) + 2\sigma_{s,s_e}^2 \sum_{k=0}^{M-1} h_{o,k} \cdot \left(-\mu_{h,k}(n)\sigma_s^2\right) E\left\{\varepsilon_{h,k}(n)\right\} - \mu_{h,k}(n)\sigma_{s,s_e}^2 h_{o,k}^2$$
$$+ \sigma_s^2 \sum_{k=0}^{M-1} \left[-2\mu_{h,k}(n)\sigma_s^2 + \mu_{h,k}^2(n)\left(m_{s^4} - \sigma_s^4\right)\right] g_k(n) + \mu_{h,k}^2(n)\sigma_s^4 trace\left(\mathbf{R}_h(n)\right)$$
$$+ 2\left[-\mu_{h,k}(n)\sigma_{s,s_e}^2 E\left\{\varepsilon_{h,k}(n)\right\} h_{o,k} + \mu_{h,k}^2(n)\left(m_{s^3,s_e} - \sigma_s^2\sigma_{s,s_e}^2\right) E\left\{\varepsilon_{h,k}(n)\right\} h_{o,k}\right.$$
$$\left. + \sigma_s^2\sigma_{s,s_e}^2 E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o\right] + \mu_{h,k}^2(n)\left[\sigma_v^2\sigma_s^2 + \left(m_{s^2s_e^2} - \sigma_s^2\sigma_{s_e}^2\right) h_{o,k}^2 + \sigma_s^2\sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2\right].$$

(4.3.4)

Applying the same method in Section 4.1, taking derivative of Eq. (4.3.4) with respect to $\mu_{h,k}(n)$ and setting the result equal to zero, we can get

$$\mu_{h,k,OLMS}(n) = \frac{2\sigma_{s,s_e}^2 h_{o,k} \cdot \left\{\varepsilon_{h,k}(n)\right\} + \sigma_s^2 g_k(n) + \dfrac{\sigma_{s,s_e}^4}{\sigma_s^2} h_{o,k}^2}{\left(m_{s^4} - \sigma_s^4\right) g_k(n) + \sigma_s^4 trace\left(\mathbf{R}_h(n)\right) + \left[\sigma_v^2\sigma_s^2 + \left(m_{s^2s_e^2} - \sigma_s^2\sigma_{s_e}^2\right) h_{o,k}^2 + \sigma_s^2\sigma_{s_e}^2 \left\|\mathbf{h}_o\right\|_2^2\right]}$$
$$2\left[\left(m_{s^3,s_e} - \sigma_s^2\sigma_{s,s_e}^2\right) E\left\{\varepsilon_{h,k}(n)\right\} h_{o,k} + \sigma_s^2\sigma_{s,s_e}^2 E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o\right]$$

(4.3.5)

Substitute Eq.(4.3.5) into Eq.(4.3.3), a recursive formula for the tap coefficient error variance $g_i(n)$ is given by:

$$g_k(n+1) = \left(1 - \mu_{h,k,OLMS}(n)\sigma_s^2\right) g_k(n) + \mu_{h,k,OLMS} \frac{\sigma_{s,s_e}^4}{\sigma_s^2} h_{o,k}^2.$$

With null nonlinearity i.e., $\sigma_{s,s_e}^2 = \sigma_{s_e}^2 = m_{s^4} = m_{s^2s_e^2} m_{s^3,s_e} = 0$, we can rewrite the OTTLMS algorithm

$$\mu_{h,k,OLMS}(n)=\frac{\sigma_s^2 g_k(n)}{\left(m_{s^4}-\sigma_s^4\right)g_k(n)+\sigma_s^4 trace\left(\mathbf{R}_h(n)\right)+\sigma_v^2\sigma_s^2}$$

$$g_k(n+1)=\left(1-\mu_{h,k,OLMS}(n)\sigma_s^2\right)g_k(n).$$

$$E\left\{\varepsilon_{h,k}(n+1)\right\}=\left(1-\mu_{h,k,OLMS}(n)\sigma_s^2\right)E\left\{\varepsilon_{h,k}(n)\right\}$$

We found the results fit the works on tradition AEC [25].

Although the time-& tap- step-size is optimal, it must take very larger computations. For this reason, we observe some term in Eq. (4.3.5) can be neglected. That is when m_{s^4}, m_{s^3,s_e} and m_{s^2,s_e^2} is comparable to σ_s^4, $\sigma_s^2\sigma_{s,s_e}^2$ and $\sigma_s^2\sigma_{s_e}^2$,respectively, and $M\gg 1$, we can get $\left(m_{s^4}-\sigma_s^4\right)g_k(n)\ll\sigma_s^4 trace\left(\mathbf{R}_h(n)\right)$, $\left(m_{s^2s_e^2}-\sigma_s^2\sigma_{s_e}^2\right)h_{o,k}^2\ll\sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2$ and $\left(m_{s^3,s_e}-\sigma_s^2\sigma_{s,s_e}^2\right)E\left\{\varepsilon_{h,k}(n)\right\}h_{o,k}\ll\sigma_s^2\sigma_{s,s_e}^2E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o$.

Therefore, the approximated OTTLMS algorithm during the first stage is summarized by the following equations:

$$e(n)=d(n)-\mathbf{w}^T\cdot\mathbf{F}^T(n)\cdot\mathbf{h}(n)$$

$$\mathbf{U}_{OLMS}(n)=\begin{pmatrix}\mu_{h,0,OLMS}(n) & & 0\\ & \ddots & \\ 0 & & \mu_{h,M-1,OLMS}(n)\end{pmatrix}, \text{ where}$$

$$\mu_{h,k,OLMS}(n)=\frac{2\sigma_{s,s_e}^2h_{o,k}\cdot\left\{\varepsilon_{h,k}(n)\right\}+\sigma_s^2g_k(n)+\frac{\sigma_{s,s_e}^4}{\sigma_s^2}h_{o,k}^2}{\sigma_s^4trace\left(\mathbf{R}_h(n)\right)+2\sigma_s^2\sigma_{s,s_e}^2E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\mathbf{h}_o+\sigma_v^2\sigma_s^2+\sigma_s^2\sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2} \quad (4.3.6)$$

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mathbf{U}(n)\cdot\mathbf{s}(n)e(n)$$

$$g_k(n+1)=\left(1-\sigma_s^2\mu_{h,k,OLMS}(n)\right)g_k(n)+\mu_{h,k,OLMS}\frac{\sigma_{s,s_e}^4}{\sigma_s^2}h_{o,k}^2 \quad (4.3.7)$$

$$E\left\{\varepsilon_{h,k}(n+1)\right\}=\left(1-\sigma_s^2\mu_{h,k,OLMS}(n)\right)E\left\{\varepsilon_{h,k}(n)\right\}-\mu_{h,k,OLMS}\sigma_s^2\sigma_{s,s_e}^2h_{o,k}^2 \quad (4.3.8)$$

The step-size adjustment is based on Eq. (4.3.6), (4.3.7) and (4.3.8).

In the following discussion, we will use the approximated OTTLMS algorithm to discuss.

4.4 Practical implementation of OTTLMS algorithm

In Section 4.2, we have demonstrated the monotonously decreasing slope model of nonlinear I/O curve to let optimal time-varying step-size LMS (OTLMS) algorithm practical. Here, the optimal time-&tap- varying step-size LMS (OTTLMS) algorithm not only needs prior statistics knowledge $\sigma_s^2, \sigma_{s,s_e}^2$ and σ_{s,s_e}^2 of the nonlinearity output but also the prior knowledge of room impulse response (RIR) $\mathbf{h}_o$. However, we can use the exponential decay model of RIR [18] and use the iteration form of $g_k(n)$ and $E\{\varepsilon_{h,k}(n)\}$ in Eq. (4.3.8) and (4.3.9) for practical implementation.

Assume the RIR $\mathbf{h}_o$ can be modeled as an exponential decay envelope shown in Figure 4.2. Let decay envelop function be:

$$\gamma_{h,k} = \gamma_{h,0}\gamma_h^{\,i} \quad \text{for} \quad k = 1 \sim M-1 \tag{4.4.1}$$

where γ_h is the room exponential decay factor. Here we fix the second norm of RIR as *1*. Therefore the initial parameter $\gamma_{h,0}$ of decay envelop function can be expressed as

$$\gamma_{h,0} = \sqrt{1-\gamma_h^2+\gamma_h^{2M}}$$

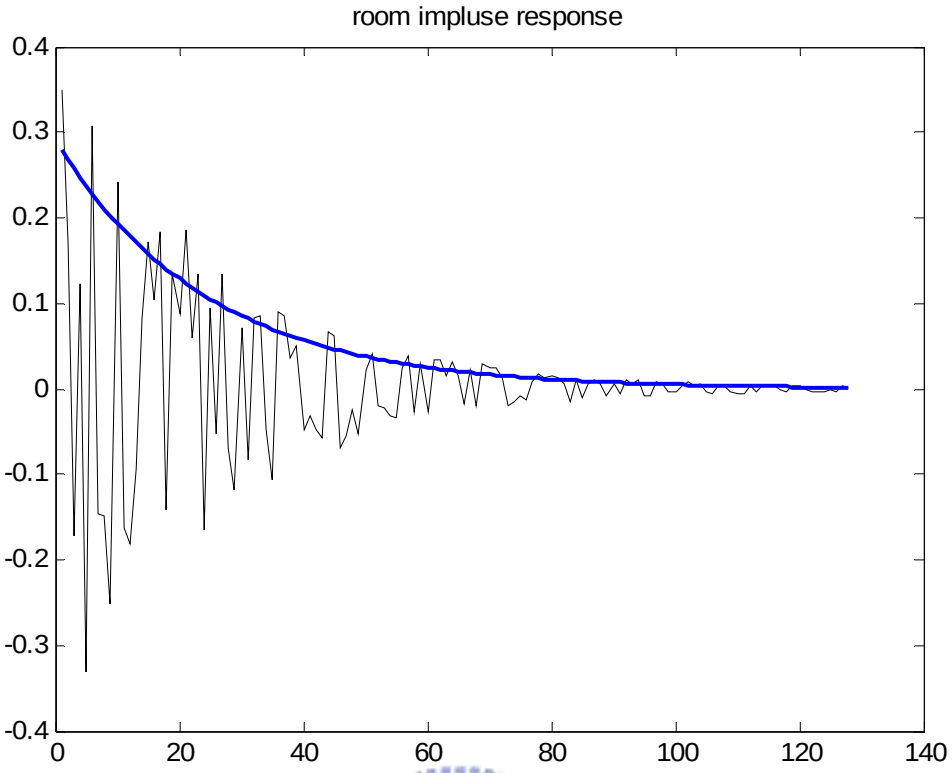

Fig. 4.2 RIR decay envelop

The diagonal element of tap coefficient error variance is $g_k(n) = E[(h_k(n) - h_{o,k})^2],\ k = 0,\ldots,M\text{-}1$. We let the initial filter tap coefficients to be zero i.e., $h_k(0) = 0$ so that $g_k(0) = E[(h_k(0) - h_{o,k})^2] = h_{o,k}^2 \approx \gamma_{h,k}^2$ and $E\{\varepsilon_{h,k}(0)\} = E\{(h_k(0) - h_{o,k})\} = -h_{o,k} \approx -\gamma_{h,k}$. We substitute $g_k(0)$ into Eq. (4.3.7) to get $\mu_{h,k,OLMS}(0)$. With $\mu_{h,k,OLMS}(0)$ plugged into Eq. (4.3.8) and (4.3.9) we can get $g_k(1)$ and $E\{\varepsilon_{h,k}(1)\}$ so forth i.e., $g_k(0), E\{\varepsilon_{h,k}(0)\} \rightarrow \mu_{h,k,OLMS}(0) \rightarrow g_k(1), E\{\varepsilon_{h,k}(1)\} \rightarrow \mu_{h,k,OLMS}(1) \rightarrow \ldots$. Thus, we can obtain $\mu_{h,k,OLMS}(n)$ for $i = 0,\ldots,M\text{-}1$ at any iteration step. The practical OTTLMS algorithm with an exponential RIR model can be described as follows:

1. Measure room exponential decay factor γ_h to get $\gamma_{h,0} = \sqrt{1 - \gamma_h^2 + \gamma_h^{2M}}$ and

$\gamma_{h,k} = \gamma_{h,0}\gamma_h^{\,i}$ for $k = 1,\ldots,M\text{-}1$.

2. Set up initial value $g_k(0) \approx \gamma_{h,k}^2$ for $k = 0,\ldots,M\text{-}1$.

3. $\mathbf{U}_{LMS}(n) = \begin{pmatrix} \mu_{h,0,OLMS}(n) & & 0 \\ & \ddots & \\ 0 & & \mu_{h,M-1,OLMS}(n) \end{pmatrix}$ where

$$\mu_{h,k,OLMS}(n) = \frac{2\sigma_{s,s_e}^2 \gamma_{h,k} \cdot \{\varepsilon_{h,k}(n)\} + \sigma_s^2 g_k(n) + \frac{\sigma_{s,s_e}^4}{\sigma_s^2}\gamma_{h,k}^2}{\sigma_s^4 trace(\mathbf{R}_h(n)) + 2\sigma_s^2\sigma_{s,s_e}^2 E\{\boldsymbol{\varepsilon}_h^T(n)\}\mathbf{h}_o + \sigma_v^2\sigma_s^2 + \sigma_s^2\sigma_{s_e}^2 \|\mathbf{h}_o\|_2^2}.$$

4. $e(n) = d(n) - \mathbf{w}^T \cdot \mathbf{F}^T(n) \cdot \mathbf{h}(n)$

5. $\mathbf{h}(n+1) = \mathbf{h}(n) + \mathbf{U}(n) \cdot \mathbf{s}(n)e(n)$

6. $g_k(n+1) = \left(1 - \sigma_s^2 \mu_{h,k,OLMS}(n)\right) g_k(n) + \mu_{h,k,OLMS} \frac{\sigma_{s,s_e}^4}{\sigma_s^2}\gamma_{h,k}^2$

7. $E\{\varepsilon_{h,k}(n+1)\} = \left(1 - \sigma_s^2 \mu_{h,k,OLMS}(n)\right) E\{\varepsilon_{h,k}(n)\} - \mu_{h,k,OLMS}\sigma_s^2\sigma_{s,s_e}^2\gamma_{h,k}^2$

By using the monotonously decreasing slope model of nonlinear I/O curve and the exponential RIR model, we can implement the OTTLMS algorithm.

4.5. OTNLMS and OTTNLMS algorithm

The above discussions are based on LMS algorithm. However, the adjustment is directly proportional to the input signal. Therefore, when the input signal is large, the LMS filter suffers from a gradient noise amplification problem. To overcome this difficulty, we may use the normalized LMS (NLMS) filter. The time-variant and time-&tap- variant step-size on the NLMS algorithm by assuming the input signal $x(n)$ is WSS with zero mean, approximated $\mathbf{s}^T(n)\mathbf{s}(n)$ by $M\sigma_s^2$ [3] i.e., $\mu_{h,OTTNLMS}(n) = M\sigma_s^2\mu_{h,OTTLMS}(n)$ and $\mu_{h,k,OTNLMS}(n) = M\sigma_s^2\mu_{h,k,OTLMS}(n)$, can be obtained. Therefore, The OTNLMS algorithm during the first stage is summarized by the following equations:

$$e(n) = d(n) - \mathbf{w}^T \cdot \mathbf{F}^T(n) \cdot \mathbf{h}(n)$$

$$\mu_{h,ONLMS}(n) = 1 - \frac{\beta}{J_h(n)}$$

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu_h \mathbf{s}(n) e(n)}{\|\mathbf{s}(n)\|_2^2 + \delta}$$

$$J_h(n+1) = \left(1 - \frac{\mu_{h,ONLMS}(n)}{M}\right) J(n) + \frac{\mu_{h,OMLMS}(n)}{M} \beta$$

Similarly, the OTTNLMS algorithm during the first stage is summarized by the following equations:

$$e(n) = d(n) - \mathbf{w}^T \cdot \mathbf{F}^T(n) \cdot \mathbf{h}(n)$$

$$\mu_{h,k,ONLMS}(n) = \frac{M\left[2\sigma_{s,s_e}^2 \sigma_s^2 h_{o,i} \cdot \{\varepsilon_{h,i}(n)\} + \sigma_s^4 g_i(n) + \sigma_{s,s_e}^4 h_{o,i}^2\right]}{\sigma_s^4 trace(\mathbf{R}_h(n)) + 2\sigma_s^2 \sigma_{s,s_e}^2 E\{\boldsymbol{\varepsilon}_h^T(n)\} \mathbf{h}_o + \sigma_v^2 \sigma_s^2 + \sigma_s^2 \sigma_{s_e}^2 \|\mathbf{h}_o\|_2^2}.$$

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

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mathbf{U}(n) \cdot \mathbf{s}(n) e(n)}{\|\mathbf{s}(n)\|_2^2 + \delta}$$

$$g_k(n+1) = \left(1 - \frac{\mu_{h,k,ONLMS}(n)}{M}\right) g_k(n) + \mu_{h,k,ONLMS} \frac{\sigma_{s,s_e}^4}{M\sigma_s^4} h_{o,i}^2.$$

$$E\{\varepsilon_{h,k}(n+1)\} = \left(1 - \frac{\mu_{h,k,ONLMS}(n)}{M}\right) E\{\varepsilon_{h,k}(n)\} - \mu_{h,k,ONLMS} \frac{\sigma_{s,s_e}^2}{M\sigma_s^2} h_{o,k}^2.$$

4.6 Summary

In this chapter, we have derived the OTLMS in Eq. (4.1.6) and (4.1.7) the OTTLMS in Eq. (4.3.6), (4.3.7), (4.3.8). Their practical implementations are provided in Section 4.2 and Section 4.4. We also develop the corresponding NLMS-type algorithm in section 4.5. The overall of discussion will be verified in Chapter 5.

Chapter 5
Computer simulations

In this chapter, we will demonstrate the simulation results of the nonlinear AEC with PWL structure to compare adaptive algorithms in Chapter 2, and to verify previous convergence analysis in Chapter 3 and optimal variable step-size, OTLMS and OTTLMS in Chapter 4.

In Section 5.1, we will define some parameters used in following simulations. In Section 5.2, we will compare the performance of nonlinear AEC with difference structures and linear AEC. Moreover, the comparison of simulation results and theoretical analyses will be presented in Section 5.3. At the same time, we will make several experiments on a real nonlinear echo with different structures of nonlinear AEC. In Section 5.4, a series of simulations and experiments on the optimum varying step-size, OTLMS and OTTLMS, will be discussed.

5.1 Simulation parameters and system performance measures

In our simulations, unless otherwise stated, the far-end signal is a uniformly distributed white noise. The nonlinear I/O mapping curve is a raised-cosine function (See Section 4.2[23]) as given below

$$f(x)=\begin{cases} -1 & ,x<-\dfrac{1+\xi}{2T} \\ Tx-\dfrac{1-\xi}{2}+\dfrac{\pi}{\xi}\cos\left(\dfrac{2Tx\pi+\pi}{2\xi}\right) & ,-\dfrac{1+\xi}{2T}\le x\le -\dfrac{1-\xi}{2T} \\ 2Tx & ,-\dfrac{1-\xi}{2T}\le x\le \dfrac{1-\xi}{2T} \\ Tx+\dfrac{1-\xi}{2}-\dfrac{\pi}{\xi}\cos\left(\dfrac{2Tx\pi-\pi}{2\xi}\right) & ,\dfrac{1-\xi}{2T}\le x\le \dfrac{1+\xi}{2T} \\ 1 & ,x>\dfrac{1+\xi}{2T}. \end{cases}$$

We use a soft-clipping type $\xi=1, T=1$ which is similar to a sigmoid function. After simplification, this raised-cosine function becomes

$$f(x)=\begin{cases} -1 & ,x<-1 \\ x+\pi\cos\left(\dfrac{2x\pi+\pi}{2}\right) & ,0\le x\le -1 \\ x-\pi\cos\left(\dfrac{2x\pi-\pi}{2}\right) & ,0\le x\le 1 \\ 1 & ,x<-1. \end{cases}.$$

Fig 5.1.1 shows the curve of the raised-cosine function.

The room impulse response in Fig 5.1.2 is generated by a random number generator with an exponential damping factor. The length of the room impulse response is set to be 128.

The signal to noise ratio at the microphone is defined as

$$SNR=10\log_{10}\frac{P_{echo}}{P_v}$$

where P_{echo} is the power of the nonlinear echo $d_{echo}(n)$ and P_v is the power of the background noise. In Fig. 3.1, the nonlinear echo $d_o(n)=\mathbf{s}_o^T(n)\cdot\mathbf{h}_o$ is produced by the loudspeaker output passing through the room impulse response. Typically SNR is set to be 30dB, unless otherwise stated.

As to the part of nonlinear AEC, the step sizes are $\mu_h=\mu_w=0.006$; the length of the linear filter is set to be 128, which is identical to the length of the room impulse

response; the number of nonlinear weights is *3*, i.e., only odd order for polynomial equal to 5^{th} (excluding the even order), and the partition vector of the PWL processor is $\begin{bmatrix} 0 & 0.33 & 0.66 & 1 \end{bmatrix}$.

To evaluate system performance, residual error power, ERLE, and coefficient misalignment are major system performance measures for comparison purposes. With the assumption of high SNR, the performance measure of echo return loss enhancement (ERLE) can be formulated as

$$ERLE(dB) = 10 \log_{10} \frac{E\{d^2(n)\}}{E\{e^2(n)\}}.$$

The linear coefficients misalignment is defined as the normalized norm of the linear coefficients error

$$\frac{\boldsymbol{\varepsilon}_h(n)}{\|\mathbf{h}_o\|_2} \triangleq \frac{\|\mathbf{h}(n)-\mathbf{h}_o\|_2}{\|\mathbf{h}_o\|_2}.$$

Moreover, the fitting of each nonlinear processor is a factor of performance. We want to evaluate which structure is better for nonlinear AEC. The most direct method is to calculate the squared difference of the nonlinear curve between PWL/polynomial type and raised-cosine curve. Due to the symmetry and zero offset, we only need to get the positive part of nonlinear I/O mapping curve. Therefore, we denote the nonlinear curve misalignment as follows:

$$\triangle f(n) \triangleq \frac{1}{N_a} \|\mathbf{f}_{\mathbf{a}}(n)-\mathbf{f}_{\mathbf{a},o}\|_2^2,$$

where $f_{\mathbf{a}}(n)$ is the PWL/polynomial output vector of nonlinear I/O mapping curve at the *n*th iteration due to the input vector $\mathbf{a}$ of dimension N_a, set by [0.1 0.2 0.3 …1], and $f_{\mathbf{a},o}$ is the desired raised-cosine output vector.

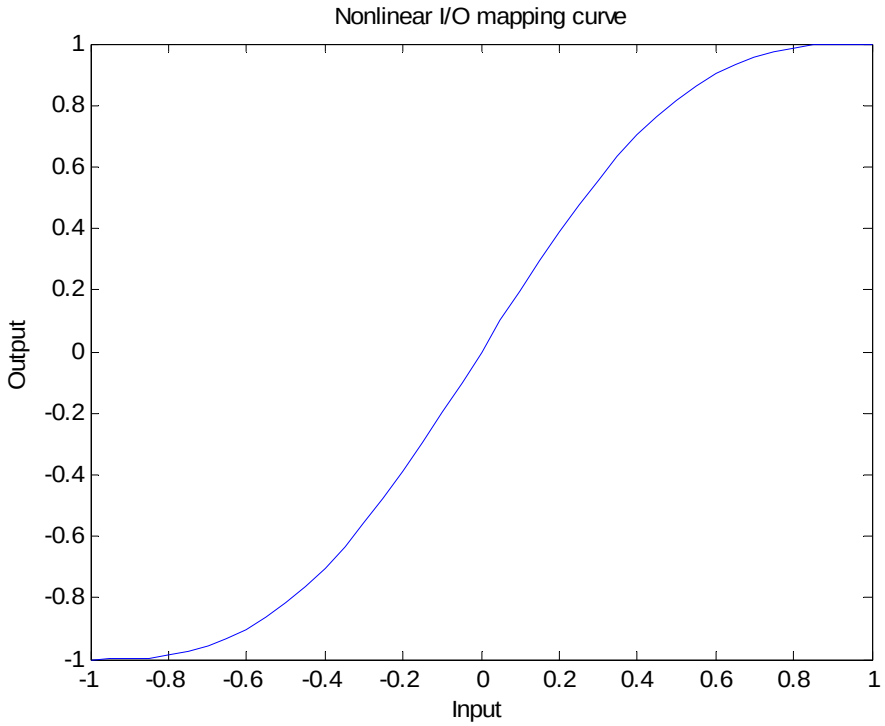

Fig. 5.1.1 Typical nonlinear I/O mapping curve using a raised-cosine function

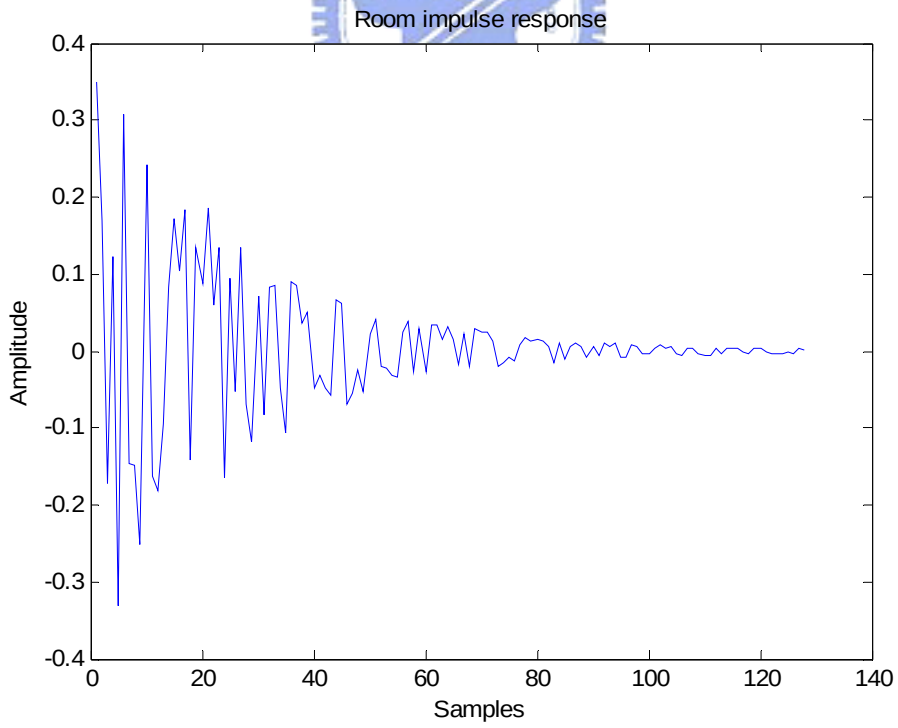

Fig. 5.1.2 Room impulse response

5.2 Nonlinear AEC based on LMS algorithm

In this section, we will perform simulations of the nonlinear LMS AEC filter. The performance of each structure will be compared in Section 5.2.1. In Section 5.2.2 we will simulate various partial update LMS algorithms. In Section 5.2.3 the issue of partition number will be discussed.

5.2.1 Performance comparison of PWL, polynomial and linear AECs

In this section, we will compare nonlinear PWL and polynomial AECs The most important issue is the nonlinear modeling. The PWL coefficients are found by fitting the raised-cosine curve based on LMS algorithm. When the order N of PWL processor is 3 with a uniform partition being [0 0.33 0.66 1], the PWL coefficients corresponding to $f_1(n)$~$f_3(n)$ can be shown to be

$$w_1 = 1.9050 \quad w_2 = -0.8886 \quad w_3 = -0.8875 .$$

Similarly, if the nonlinear AEC uses the polynomial structure with only odd order, the polynomial coefficients would be

$$a_1 = 1.8384 \quad a_3 = -0.8734 \quad a_5 = -0.0252 ,$$

i.e., $f(x) = a_1 x + a_3 x^3 + a_5 x^5$. The corresponding curves, raised-cosine, PWL and polynomial one, are plotted in Fig 5.2.1, these curves have similar behavior. To examine their differences in details, we plot the squared error between PWL/polynomial models and raised-cosine mapping in Fig 5.2.2. We found the PWL model performs better than the polynomial one in case of fitting a raised cosine curve with degree of freedom being 3.

Further, in order to avoid the correlations of the power series, we use an orthogonal polynomial to instead it. For a uniform distribution, the orthogonal pair

would be $x(n)$, $x^3(n)-\frac{3}{5}x(n)$ and $x^5(n)-1.11x^3(n)+0.24x(n)$. We can plot the squared error curves of PWL and orthogonal polynomial models in Fig 5.2.3. The result shows the orthogonal polynomial structure can improve the fitting even better than PWL structure. In the following discussion, the orthogonal polynomial structure would be used for comparison.

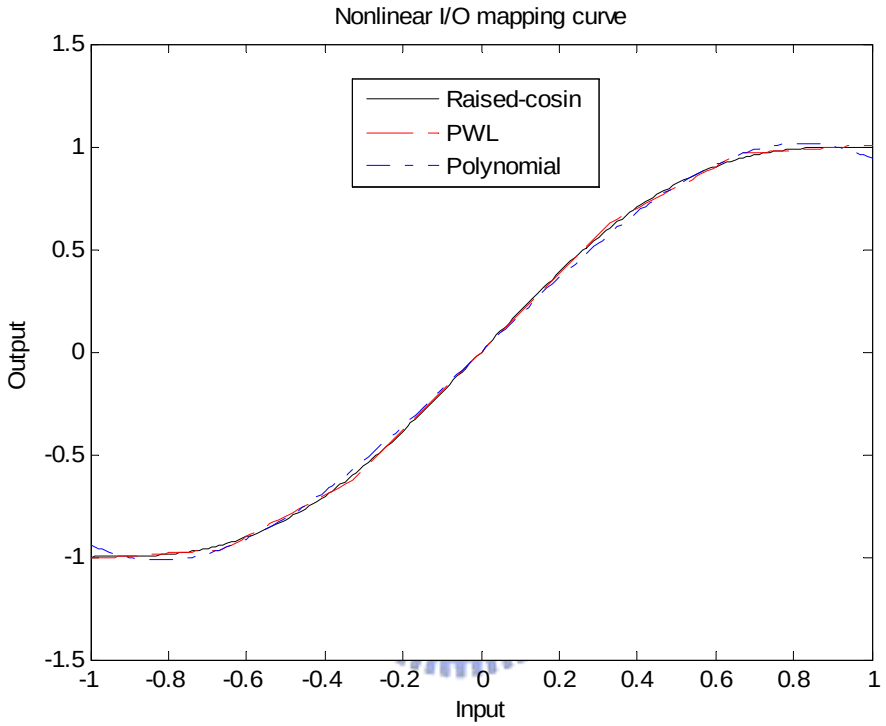

Fig. 5.2.1 Nonlinear I/O mapping curve of PWL and polynomial models

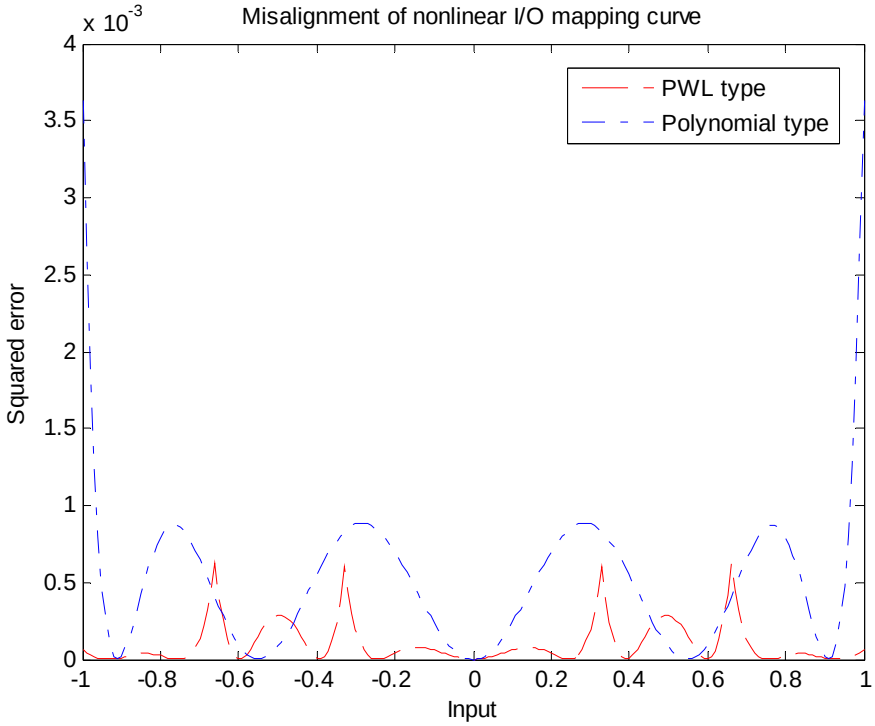

Fig. 5.2.2 Modeling errors of PWL and polynomial models in fitting to a raised-cosine function

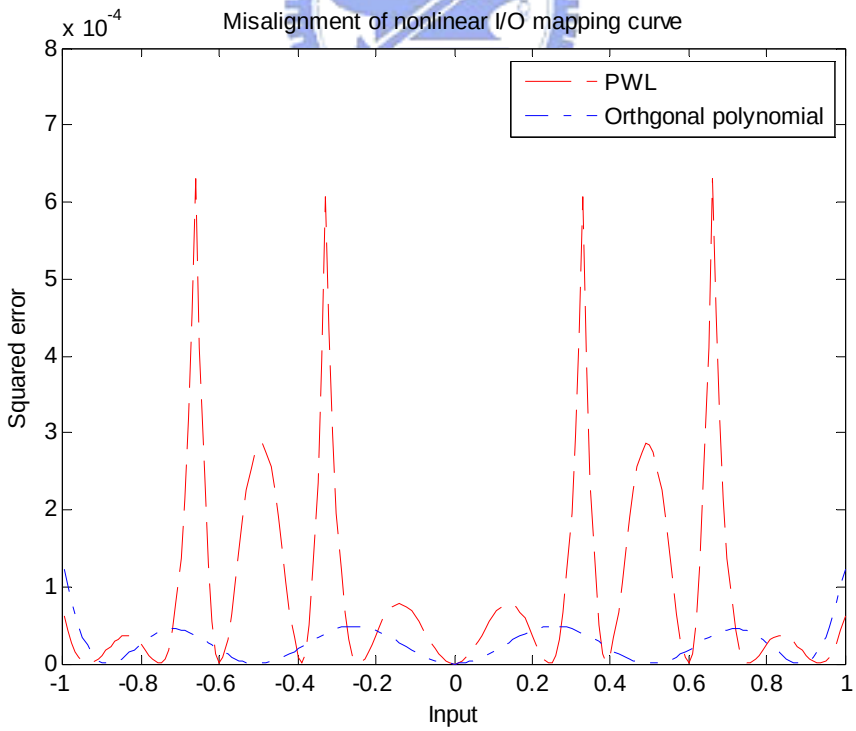

Fig. 5.2.3 Modeling errors of PWL and orthogonal polynomial models in fitting a raised-cosine function

In Chapter 2, we have derived the joint adaptation of the nonlinear PWL AEC. However, each filter (or processor) behaves to compensate the other one's misalignment. This can result in a perpetual oscillating system. Therefore, we will adopt two-staged strategy [7] with a switching point at 2000-th iteration to proceed.

The ERLE, linear coefficients misalignment and nonlinear curve misalignment will be used in our simulations to compare the performances of PWL, polynomial, and linear AECs, as plotted in Fig 5.2.4~5.2.6, respectively. The step-size for nonlinear polynomial AEC is chosen as 0.01 so that its converged ERLE is identical to that of PWL-type for a fair comparison. We can find the PWL structure has better ERLE than polynomial one due to its faster convergence in Fig 5.2.4 and Fig. 5.2.6. Moreover, the PWL structure has slightly lower steady-state linear coefficients misalignment in Fig 5.2.5. The result comes from that the error of nonlinear curve also affects the steady-state linear coefficients misalignment due to the cascade model. Therefore, the results demonstrate the PWL structure is the most effective structure than the others in the scenario of low-order fitting of a raised cosine function.

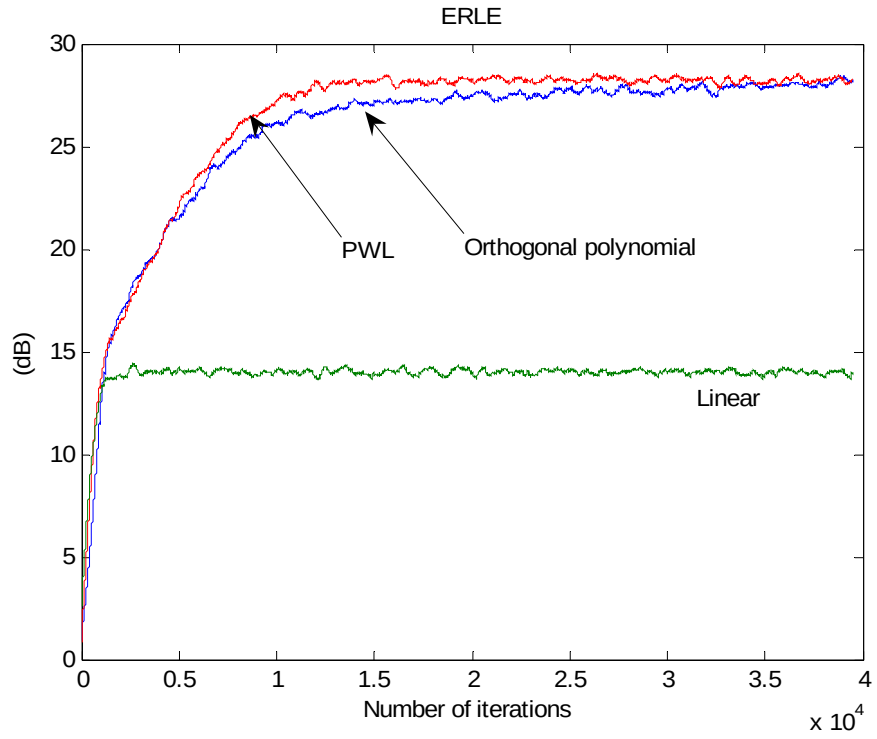

Fig. 5.2.4 ERLE comparison

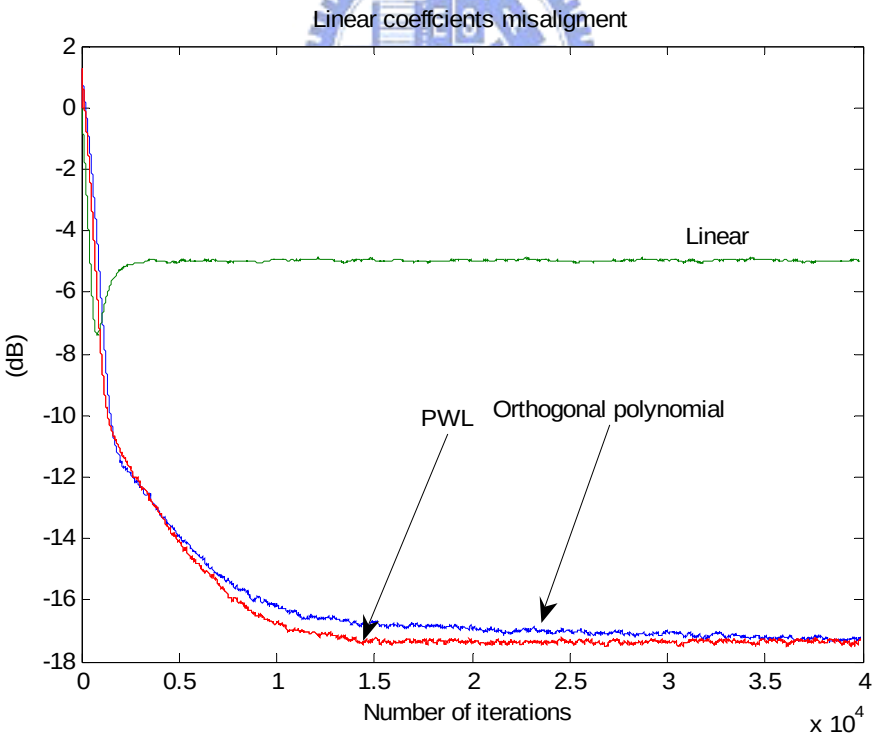

Fig. 5.2.5 Linear coefficients misalignments

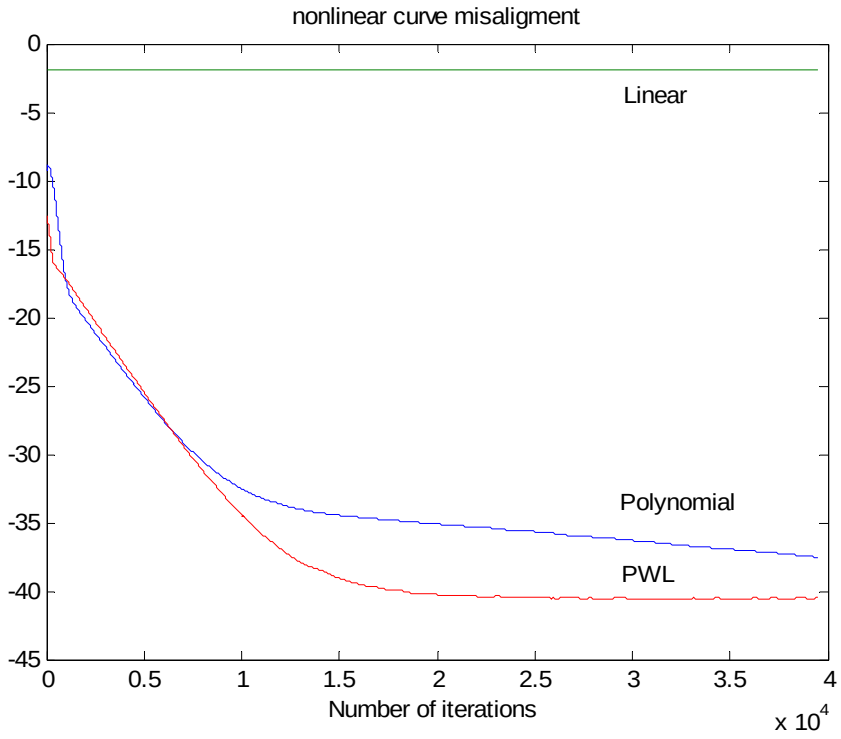

Fig. 5.2.6 Nonlinear curve misalignments

5.2.2 Comparison of partial update LMS algorithms

In this section, we will simulate various partial update schemes discussed in Section 2.3. In addition to the raised-cosine curve, a saturated curve

$$f(x) = 2.5967x + \text{-}3.3283x^3 + 1.7833x^5,$$

plotted in Fig 5.2.7, is also used to account for highly nonlinear I/O mapping curve of the loudspeaker system.

Fig. 5.2.8 shows the ERLE comparison in case of a raised-cosine curve. We can find all of the partial-update LMS algorithms converge and the order of convergence rate being, Random > Located > Periodic > Variant periodic > Sequential LMS. The Random, Located and Periodic LMS algorithm take about two times of iterations than the original LMS algorithm to reach steady-state ERLE but only one-third computation of the original one. The others, Variant periodic and Sequential schemes do not benefit the computational efficiency in this case.

As to the case of a saturated curve, we can see from Fig. 5.2.9 that the saturation affects the convergences of these partial-update LMS algorithms enormously except for the Variant periodic one. The convergence speed of these partial-update LMS algorithms rank as follows, Variant periodic > Random ≈ Periodic ≈ Sequential > Located LMS.

To sum up, the Random, Periodic and Variant periodic schemes are good candidates for partial update scheme considering the ERLE convergence performance in cases of a raised cosine and highly saturated mapping functions.

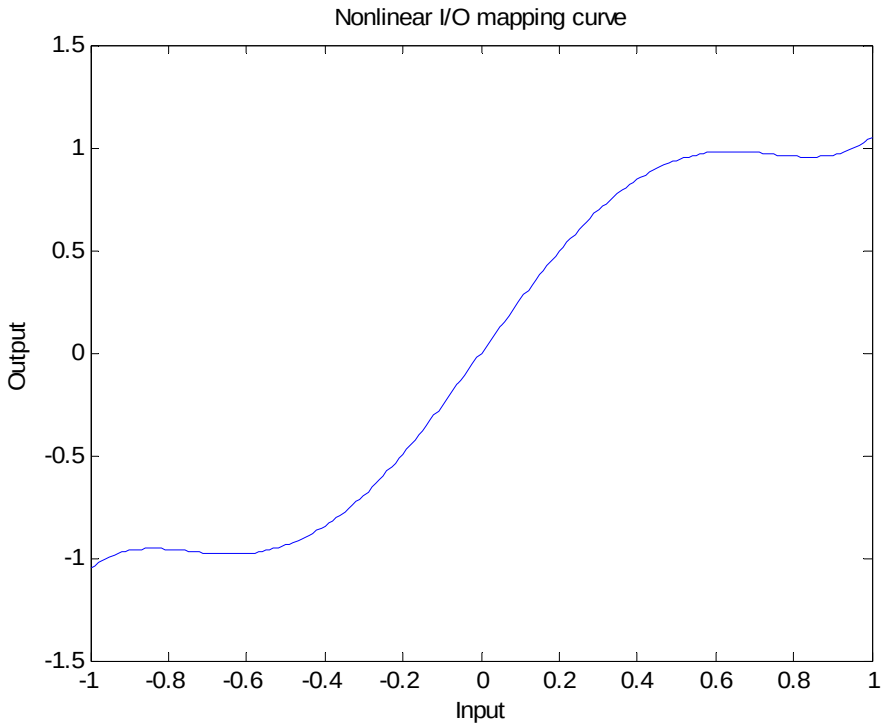

Fig. 5.2.7 Saturated curve for nonlinear I/O mapping

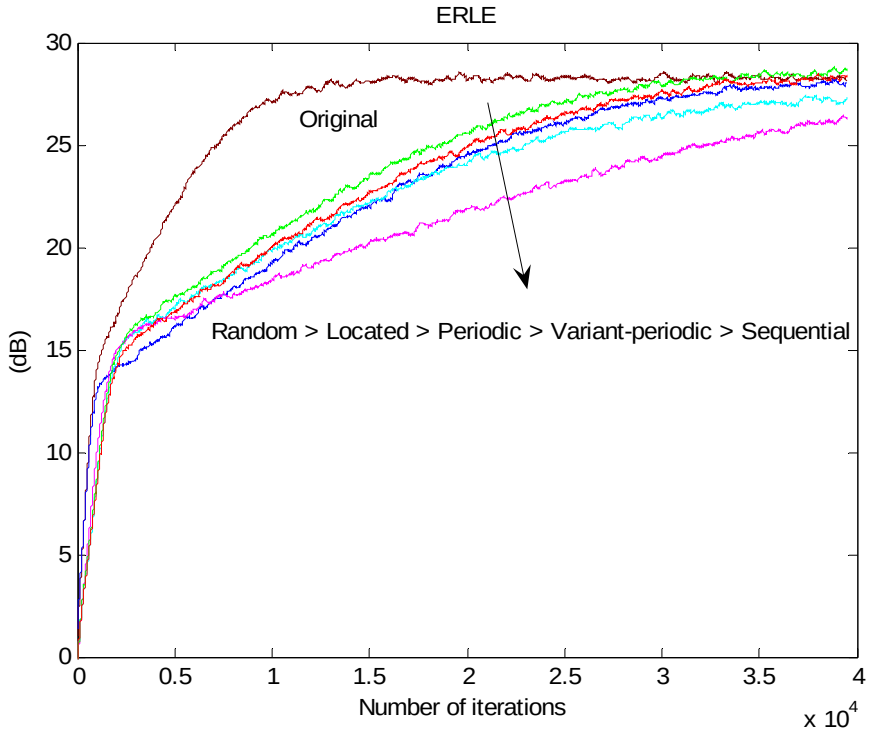

Fig. 5.2.8 ERLE of partial update schemes for a raised-cosine mapping

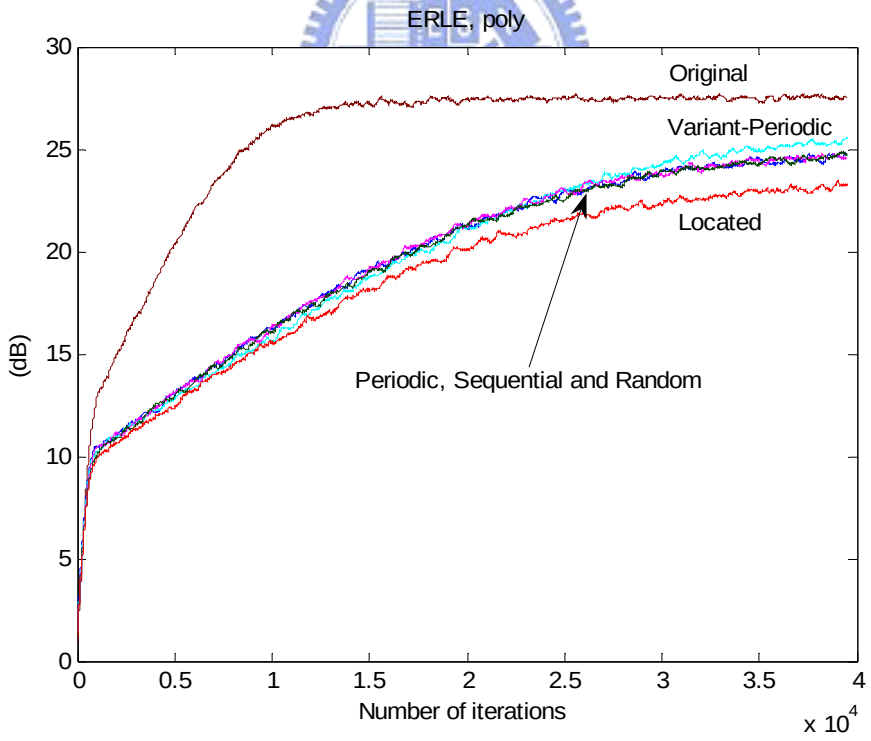

Fig 5.2.9 ERLE of partial update schemes for a saturated mapping

5.2.3 Partition number of PWL processor

Partition is an important issue of PWL function. In figure 5.2.10, if we get more resources in partition, the fitting can be better. We also want to demonstrate this property on nonlinear PWL AEC With the assumption of uniform partitioning, there will be simulations with different partition numbers under two nonlinear I/O mapping curves, raised-cosine and saturated one.

In Fig 5.2.10, we present the ERLE curves under raised-cosine mapping with the partition number being *4, 8, 16, 32, 64, 256,* respectively. We find that as the partition number becomes larger, the performance will be better due to a better fitting PWL curve. Similar results also hold in case of a saturated mapping curve, as can be seen in Fig 5.2.11. Another phenomenon is that in case of higher nonlinearity, the performance would be worse. We will also verify it in Section 5.3.

(a) (b) (c)

Fig. 5.2.10 Raised cosine curve and the corresponding PWL curves with two partition number: 2(b) and 4(c)

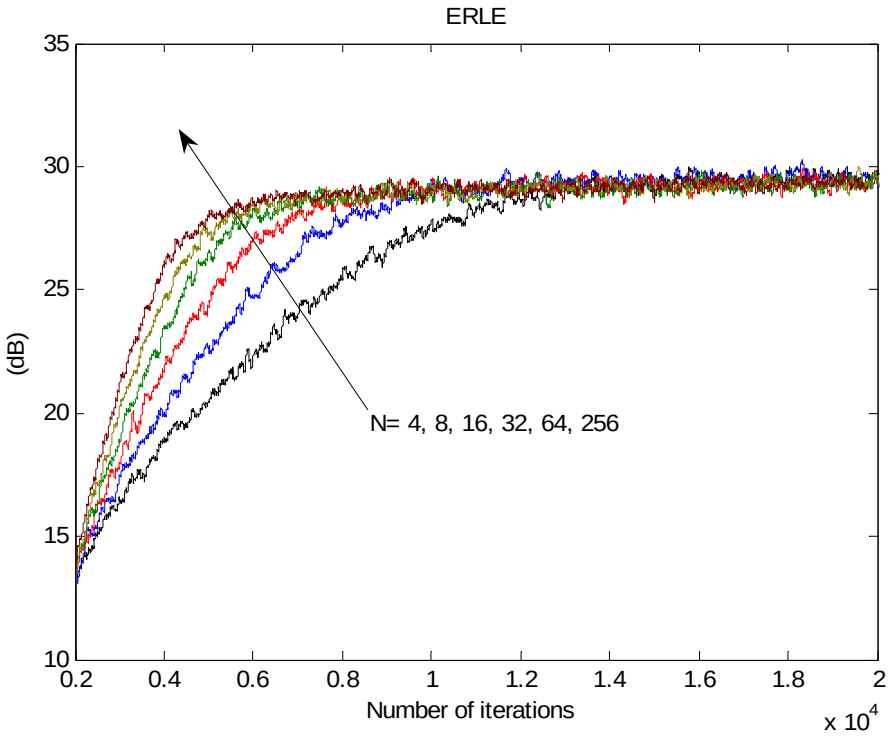

Fig. 5.2.11 ERLE for raised-cosine mapping using various partition numbers

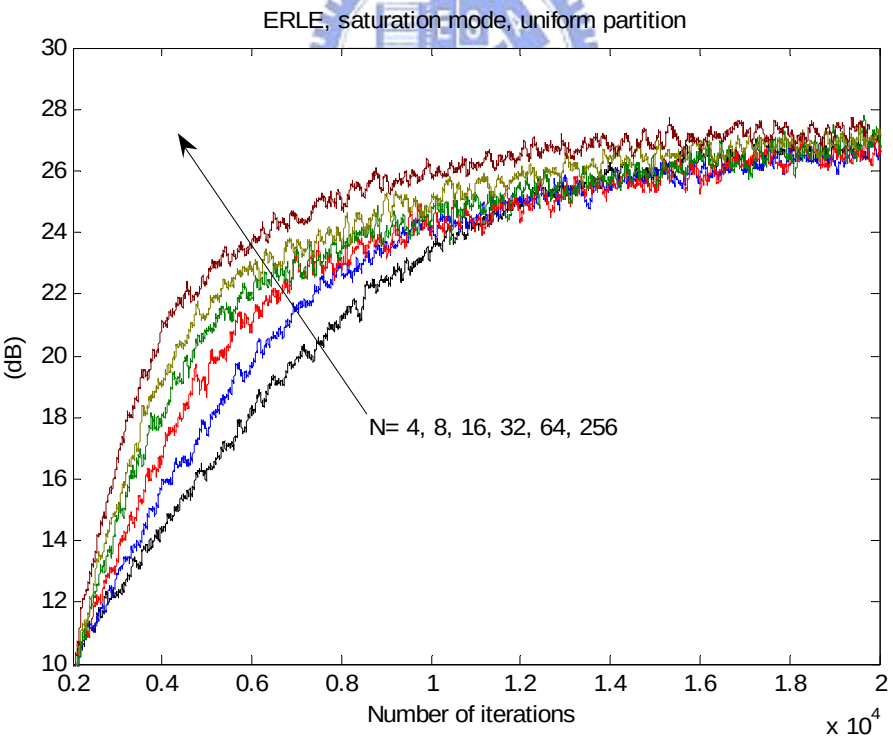

Fig. 5.2.12 ERLE under saturated curve using various partition numbers

5.3 Two-staged adaptation and its convergence analysis

In this section, we will simulate two-staged adaptation. Due to the our main part : the first stage, we not only test and verify the convergence analysis and stability bound, but also simulate how the important factors do influence the convergence behavior in Section 5.3.1. The second stage will also be verified in Section 5.3.2 with two procedures that keep off the effect of zero eigen-value. In Section 5.3.3, we will consider the switching point of the two-staged adaptation. The experiments will be presented in Section 5.3.4.

5.3.1 Convergence and stability analysis of the first stage

To verify our convergence analysis based on PWL structure, we show the simulated and theoretical curves on two stages with uniformly distributed input over ±1. In the first stage, only updates linear filter, we let the step size $\mu_h = 0.01$ under the stability bound in Eq. (3.2.14), SNR=30 dB, the length of the room impulse response is set to be 128, which is identical to number of taps of the linear filter, the nonlinear filter order is 3 with a uniform partition on [0 0.33 0.66 1], the PWL coefficients of the loudspeaker is $w_{o,1} = 1.9050$ $w_{o,2} = -0.8886$ $w_{o,3} = -0.8875$ to approach a raised-cosine curve and the initial ones of the PWL processor is $w_1 = 1.905$ $w_2 = 0$ $w_3 = 0$ equivalent to a straight line with slope 1.905. As shown the linear coefficients misalignment and residual error power in Fig 5.3.1 and 5.3.2, the theoretical curves are plotted from (3.2.16) and (3.2.21), respectively, the simulation results agree well with the theoretical curves. If we set the step size $\mu_h = 0.013$ over the stability bound in Eq. (3.2.14), the results in Fig 5.3.3 and 5.3.4 show it would not converge with time. That means the stability bound is work.

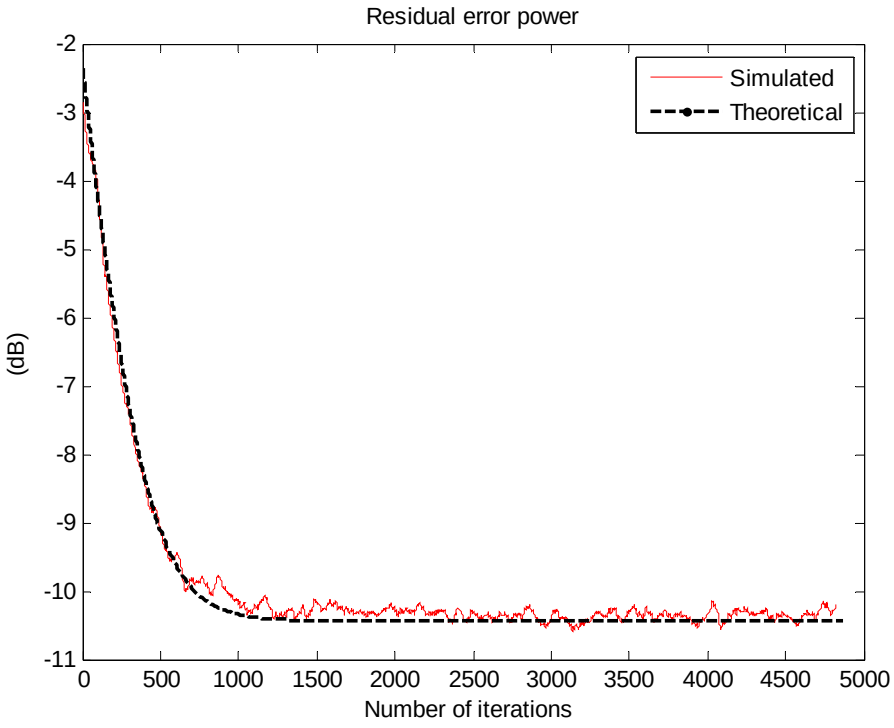

Fig. 5.3.1 Residual error power during the first stage with μ_h=0.01.

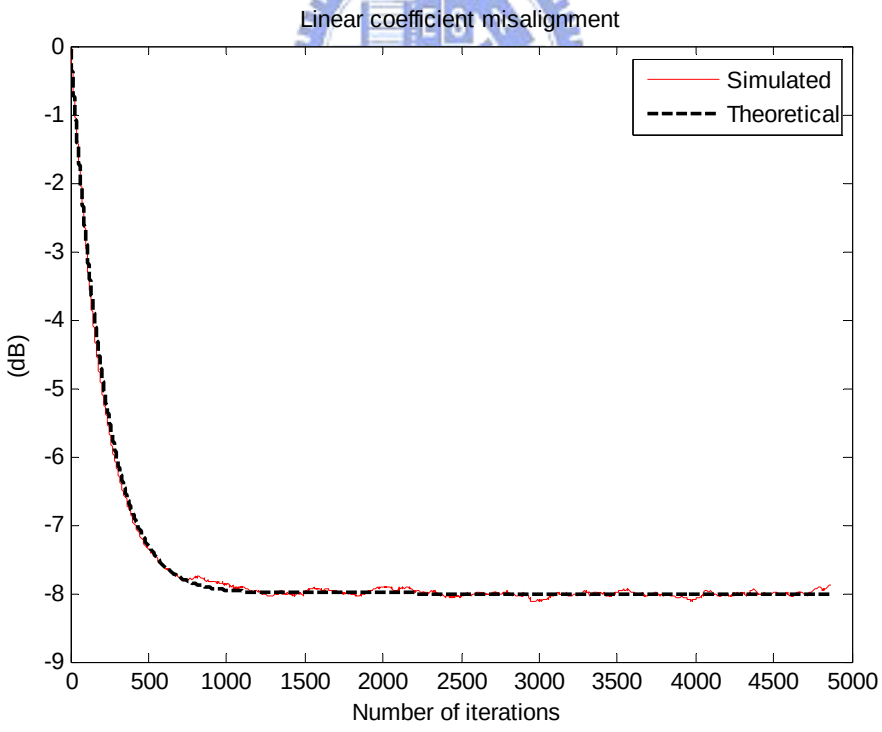

Fig. 5.3.2 Linear coefficients misalignment during the first stage with μ_h=0.01

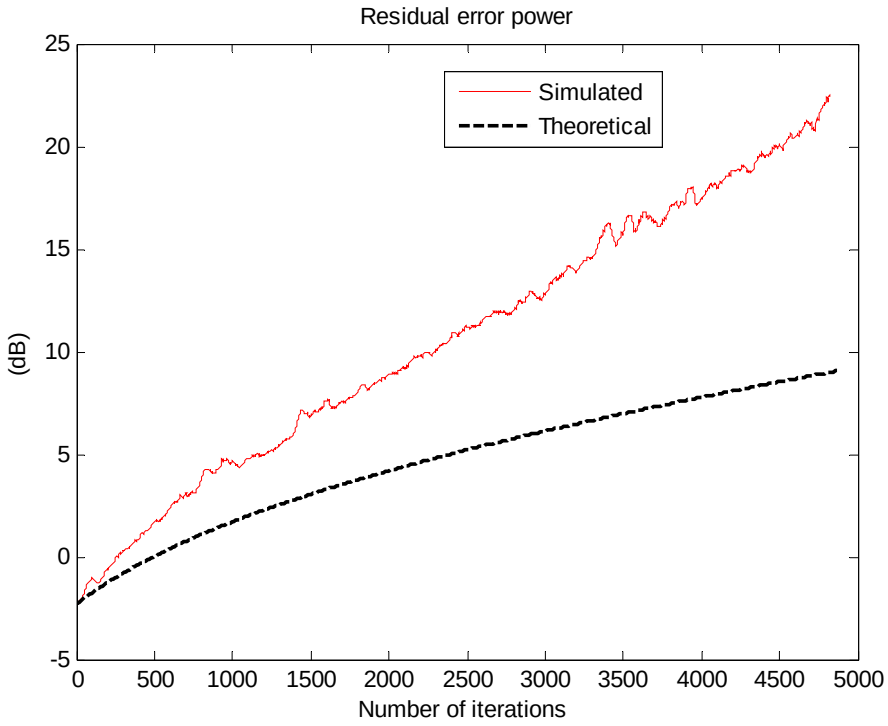

Fig.5.3.3 Diverged residual error power during the first stage with over-sized μ_h=0.013

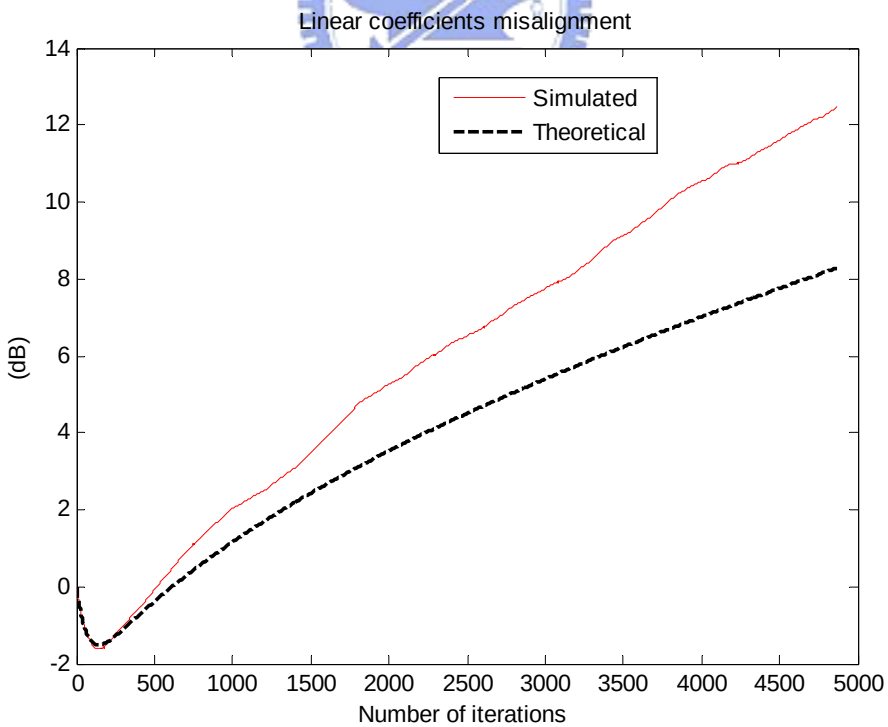

Fig 5.3.4 Diverged linear coefficients misalignment during the first stage with over-sized μ_h=0.013

Further, we want to find the factors that influence the convergence behavior of the first stage in Eq. 3.2.16, 3.2.17, 3.2.21. In the following simulations, unless otherwise stated, μ_h is set to 0.001.

The first trial is the factor of the SNR, we set the different SNR 10dB and 30dB to simulate. In Fig. 5.3.5 and 5.3.6, the factor of SNR will influence the steady-state of linear coefficient misalignment and residual error power. However, both of them have similar convergence speed as what we say in Chapter 3.

Next, we want to make sure how the step-size μ_h affects the convergence behavior. In order to observe that, difference μ_h 0.001, 0.006 and 0.01 is set. In Fig. 5.3.7 and 5.3.8 we found that as the step-size μ_h increases, the steady-state also enlarges but it doesn't happen on the convergence speed. When the step-size changes form 0.006 to 0.001, the convergence speed becomes slow. It is unlike the changes of 0.001 to 0.006. As well as, an optimal time-varaint step-size for the convergence speed must exist. We will process the step-size control to get the best performance, convergence speed and the steady-state. The issues will be discussed explicitly in the Section 5.4.

Moreover, the variance of PWL processor output σ_s^2 will result in the convergence speed. We set three choices of amplitude of far-end signal and theirs corresponding σ_s^2 are *1.2, 0.9* and *0.6*, respectively. In Fig. 5.3.9 and 5.3.10, we found the large power has better convergence speed than small one but worst steady-state due to the nonlinearity effect.

The most important factor is the nonlinearity of loudspeaker. If the nonlinearity is more conspicuous the covariance σ_{s,s_e}^2 of $\mathbf{s}(n)$ and $\mathbf{s}_e(n)$ and the variance $\sigma_{s_e}^2$ of $s_e(n)$ are larger. In Eq. (3.2.16) and (3.2.17), the convergence speed of linear

coefficient weight error won't change but the steady-state will enlarge. In order to verify this, we use the monotonously decreasing slope model of nonlinear I/O mapping curve with by setting different γ_d. The value γ_d can be regards as the nonlinearity. With a lot nonlinearity, then γ_d is much negative. In Fig 5.3.11 and 5.3.12 we present the simulation by setting different $\gamma_d = -0.9, -0.5$ and 0.2. The simulation results show that the convergence rate is similar; steady-state error is a monotonically increasing function of nonlinear factors.

Finally, we show the simulations that the far-end signals use different Pdfs. For a fair comparison, we fix the corresponding σ_s^2 as 0.4 and show its corresponding Pdfs in Fig 5.3.13. Fig. 5.3.14 and 5.3.15, we found all of them keep the same convergence speed. However, the Uniform one has smaller steady-state than others due to the nonlinearity factor σ_{s,s_e}^2 and $\sigma_{s_e}^2$.

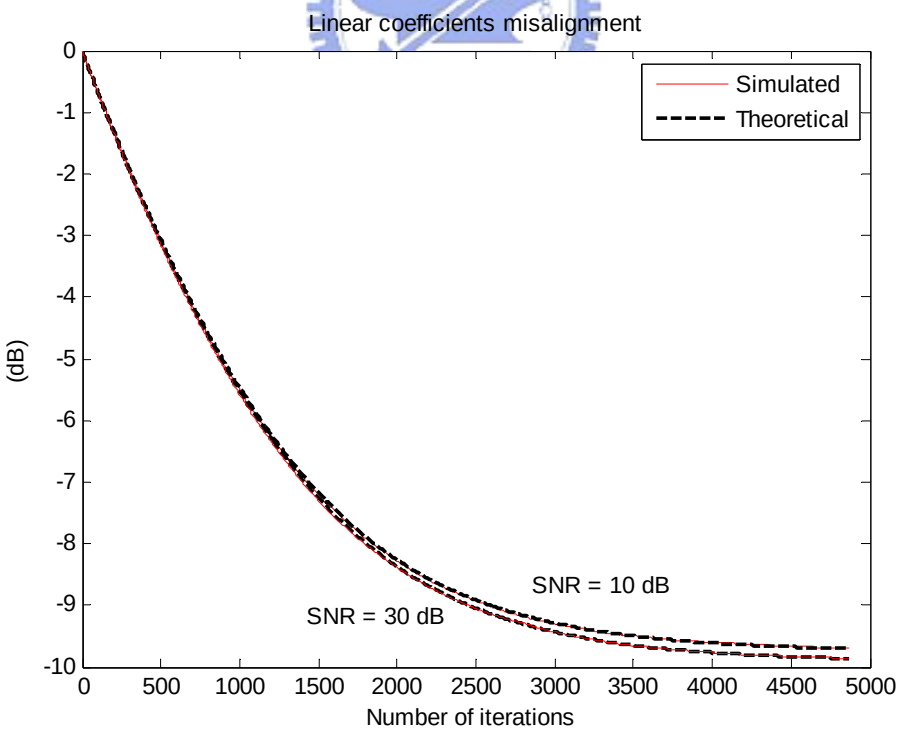

Fig. 5.3.5 Linear coefficients misalignment during the first stage under different SNRs

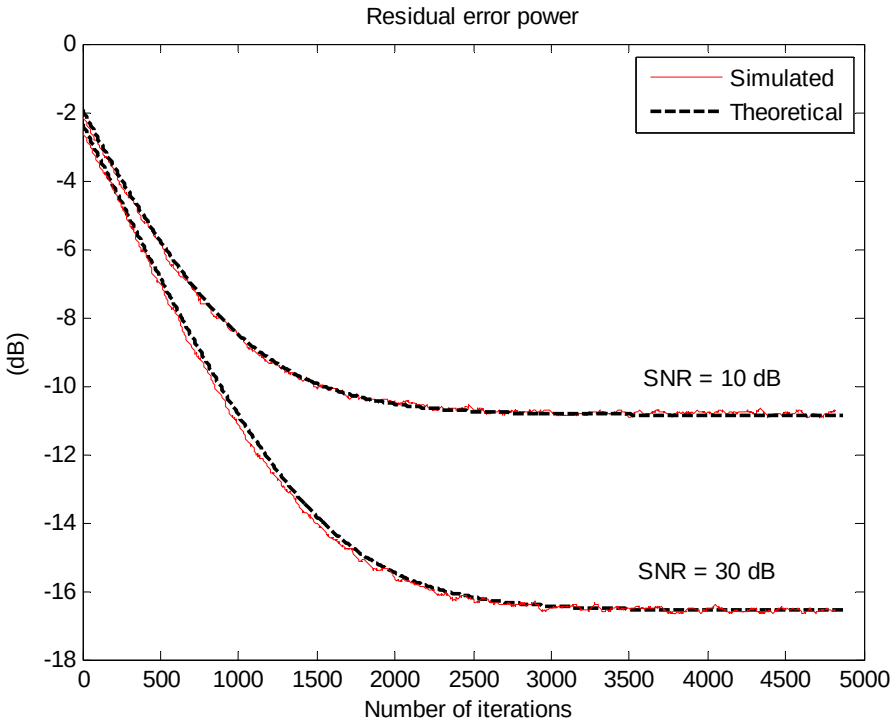

Fig. 5.3.6 Residual error power during the first stage under different SNRs

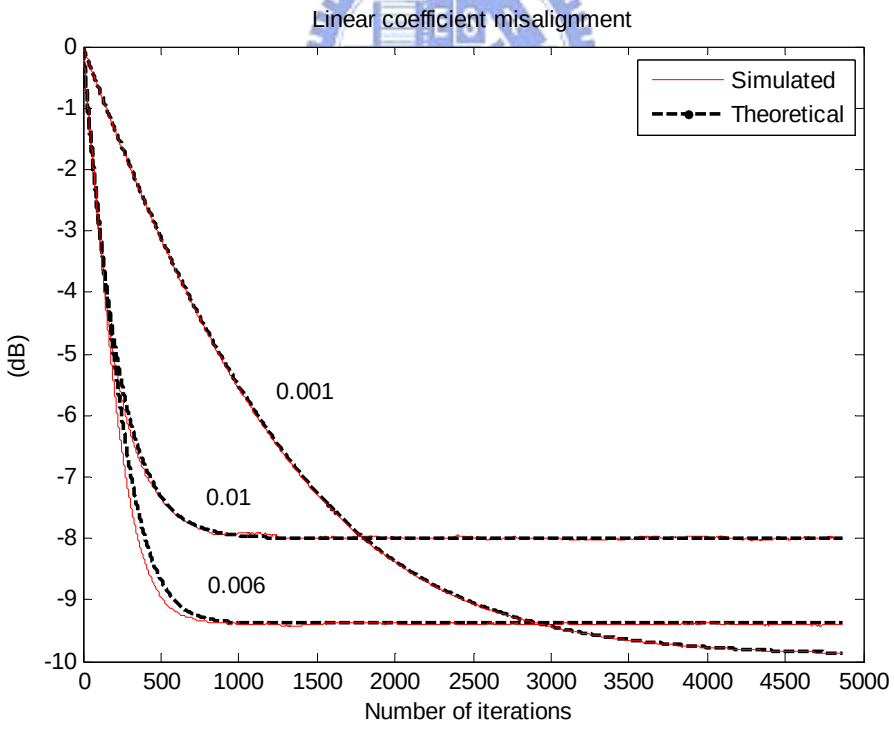

Fig. 5.3.7 Linear coefficient misalignment during the first stage with different step-sizes

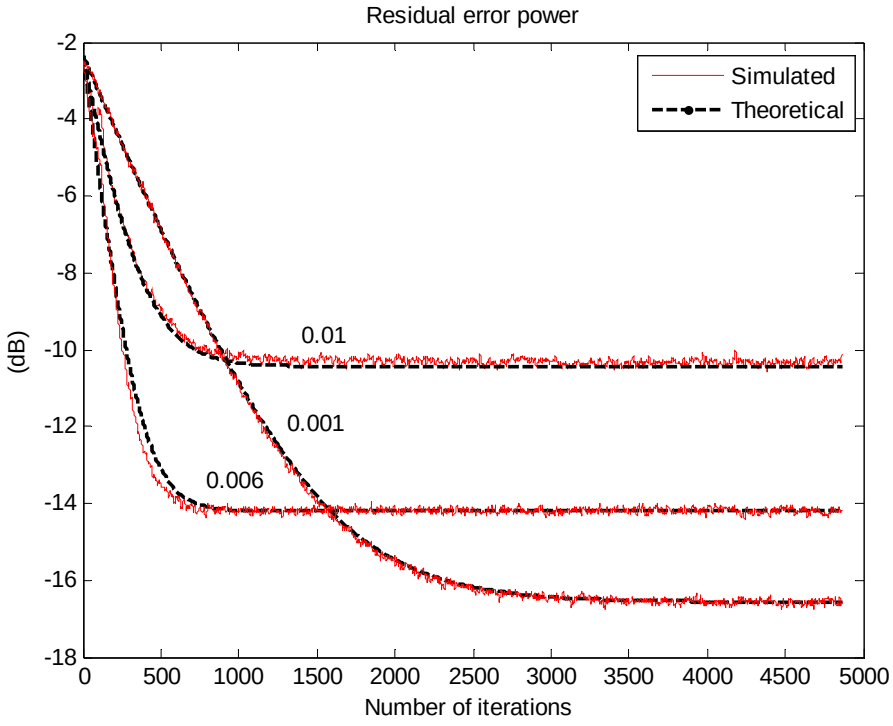

Fig. 5.3.8 Residual error power during the first stage under different step-sizes

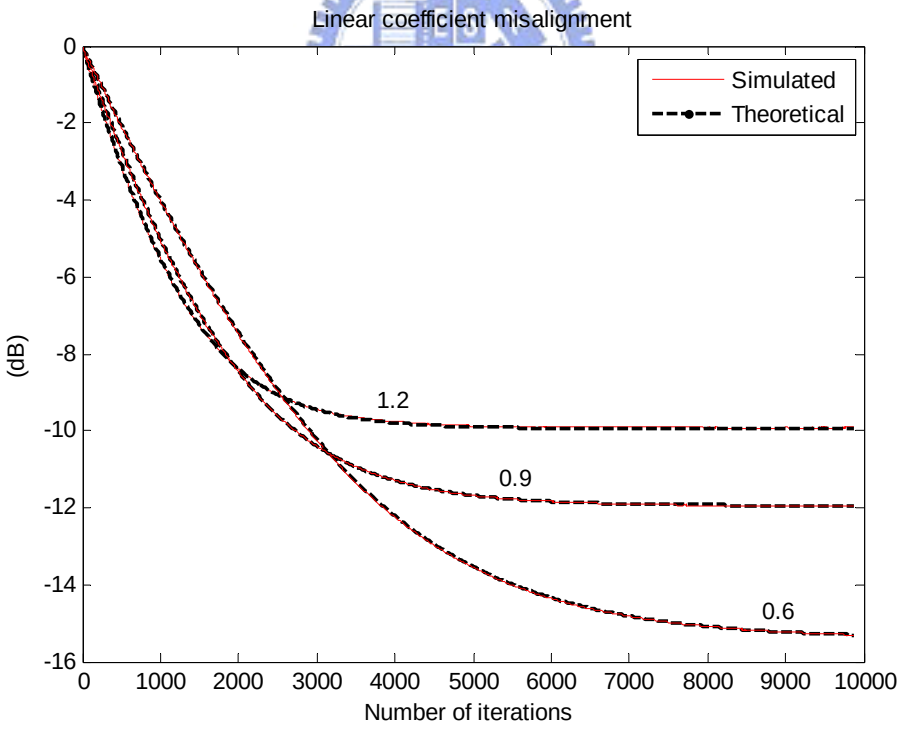

Fig. 5.3.9 Linear coefficients misalignment during the first stage with different powers of the output of PWL processor

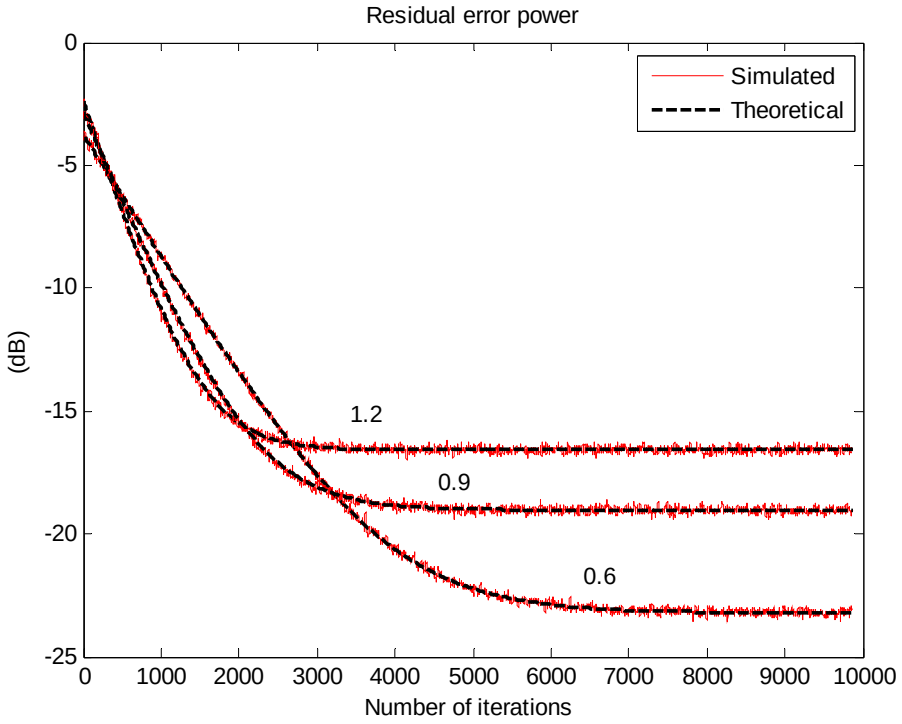

Fig. 5.3.10 Residual error power during the first stage with different powers of the output of PWL processor

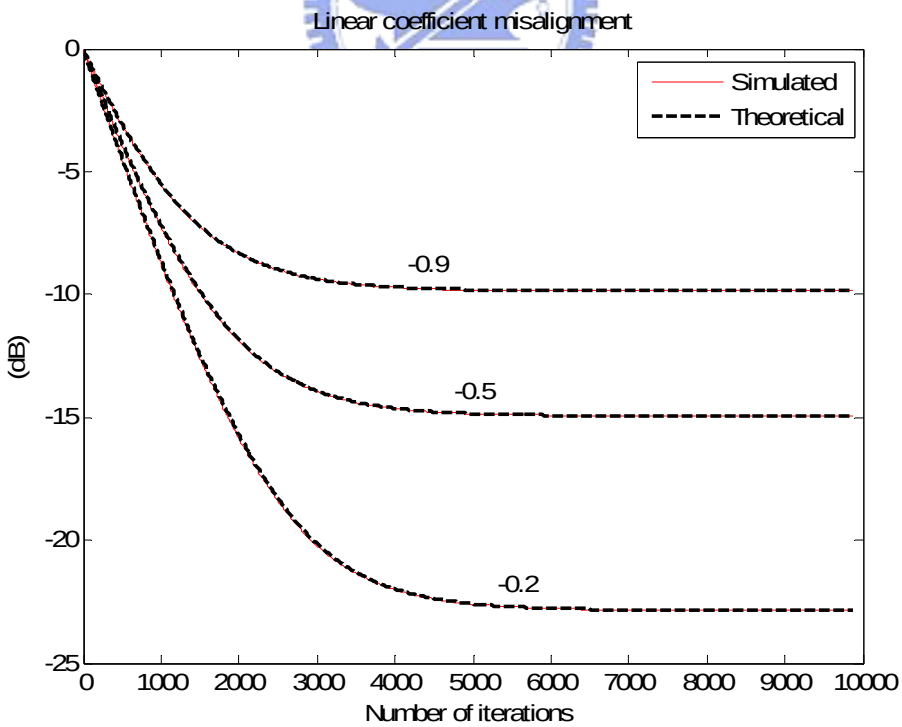

Fig. 5.3.11 Linear coefficients misalignment during the first stage using γ_d

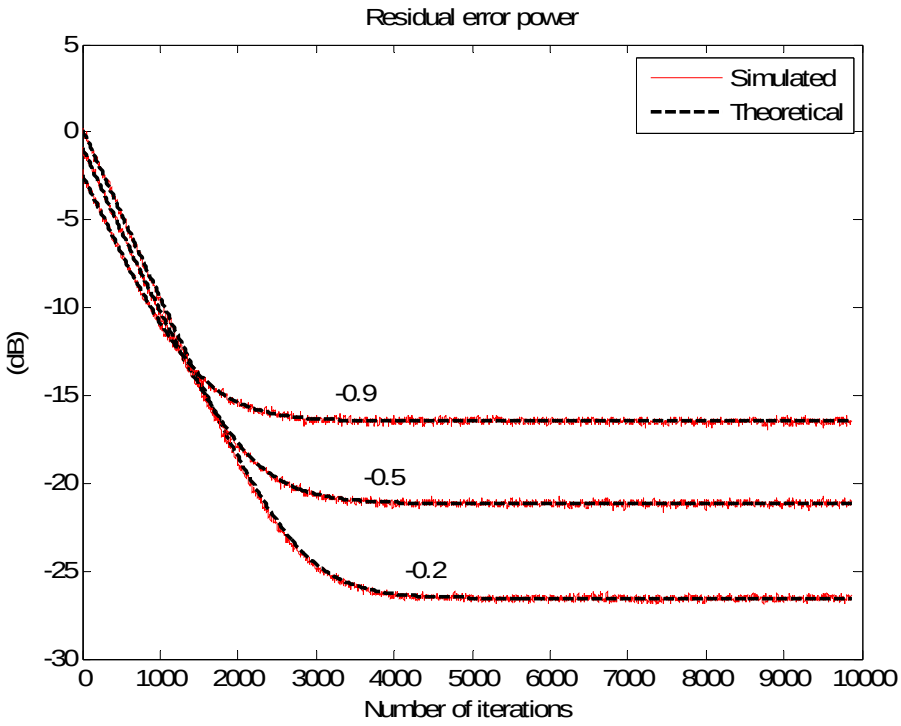

Fig. 5.3.12 Residual error power during the first stage using γ_d

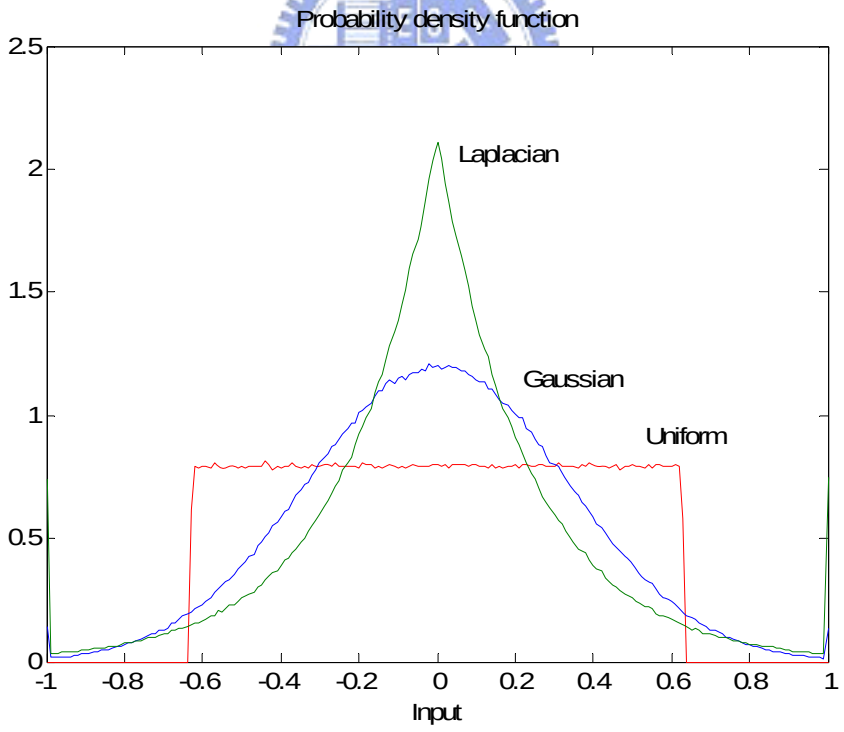

Fig 5.3.13 Different Pdfs of far-end signal

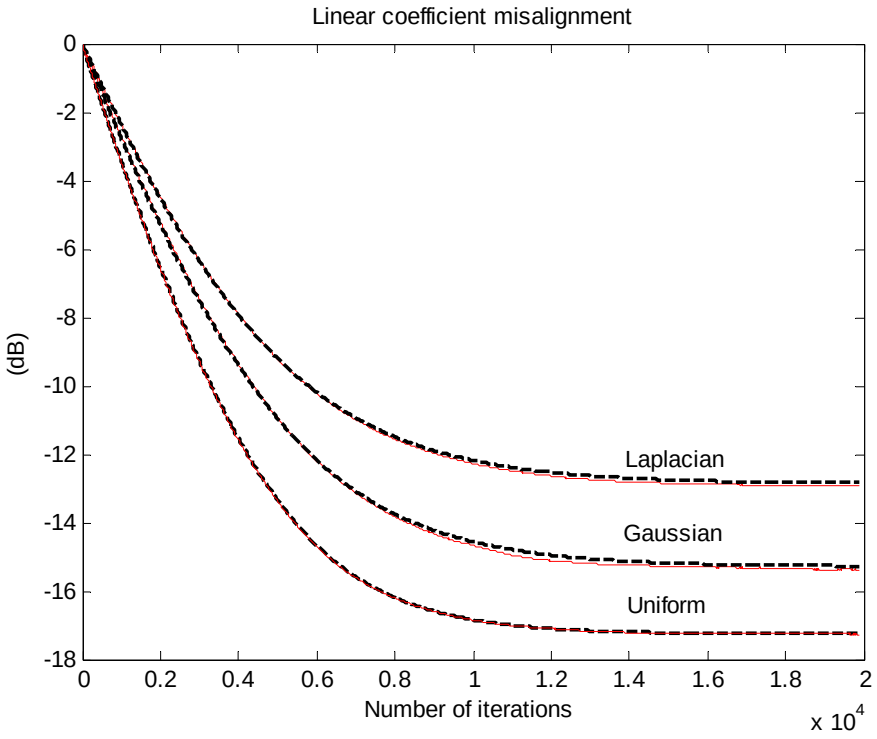

Fig. 5.3.14 Linear coefficients misalignment during the first stage using different Pdfs

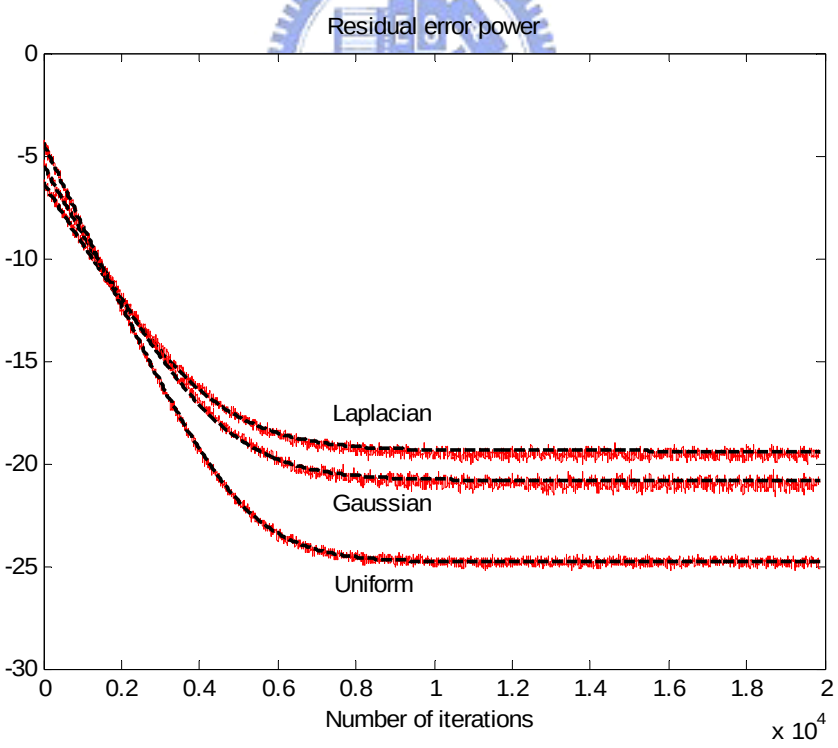

Fig. 5.3.15 Residual error power during the first stage using different Pdfs

5.3.2 Convergence analysis of the second stage

After the first stage, we continue to analysis the convergence behavior of the second stage. We keep using all of information of the stage 1. That means the initial linear FRI filter of the second stage would be the end one of the first stage. Here, in order to avoid a perpetual oscillation, we modify the norm of the initial linear FRI filter to 1 and fix it in every iteration. Beside, the PWL processor must also be modified. The misalignment of combined linear and PWL coefficient weight error $\boldsymbol{\varepsilon}(n) = \left[\boldsymbol{\varepsilon}_h(n) \quad \boldsymbol{\varepsilon}_w(n)\right]^T$ is defined as

$$\frac{\boldsymbol{\varepsilon}(n)}{\left\|\left[\mathbf{h}_o \quad \mathbf{w}_o\right]\right\|_2} \triangleq \frac{\left\|\left[\mathbf{h}_o \quad \mathbf{w}_o\right] - \left[\mathbf{h}(n) \quad \mathbf{w}(n)\right]\right\|_2}{\left\|\left[\mathbf{h}_o \quad \mathbf{w}_o\right]\right\|_2}.$$

In Fig. 5.3.16 and 5.3.17, we simulate the misalignment of combined linear and PWL coefficient weight error and residual error power and plot the theoretical curve by Eq. (3.3.7) and (3.3.8), respectively. The misalignment of the simulated and theoretical curves was purely mismatching after 5000 iteration. But the residual error power has similar behavior on simulated and theoretical curves. It just exists with a little high mismatch at start due to the approximation in Eq. (4.3.2). Therefore, we go to find out what the reason bring about this results. We found the correlation matrix $\mathbf{R}_G$ has an approximately zero eigen-value. Above the Eq. (4.3.7), the zero eigen-value will not make the corresponding term of misalignment reduce by iterating as time. As a result, the zero eigen-value will bound the convergence of misalignment, but not affect the residual error power in Eq (4.3.8).

In order to solve this problem, we make a trick that skips the corresponding term with the zero eigen-value and denote it as skipping. Under the skipping, the results between simulated curve and theoretical curve are similar with a little mismatch In Fig 5.3.18 and 5.3.19.

Besides, we know that in the second stage the weight error of both, linear and PWL one, are small enough. Here, we assume they wouldn't react to each other, that is the correlation matrix $\mathbf{R}_{\mathbf{G}}$ can be simplified as

$$\mathbf{R}_{\mathbf{G}} = \begin{bmatrix} E\left\{\mathbf{s}_o(n)\cdot\mathbf{s}_o^T(n)\right\} & \mathbf{0} \\ \mathbf{0} & E\left\{\mathbf{P}^T(n)\cdot\mathbf{h}_o\cdot\mathbf{h}_o^T\cdot\mathbf{P}(n)\right\} \end{bmatrix}.$$

We denote the procedure as decoupling. In Fig. 5.3.20 and Fig. 5.3.21, the theoretical curves are almost fitting to the simulated curves.

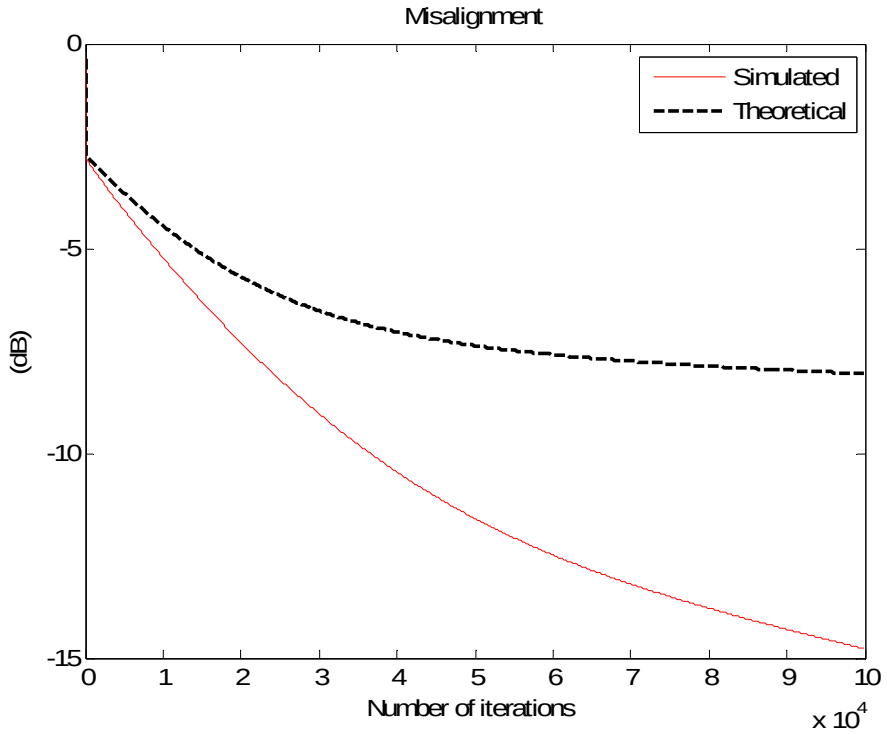

Fig. 5.3.16 Misalignment of combined coefficient weight error during the second stage

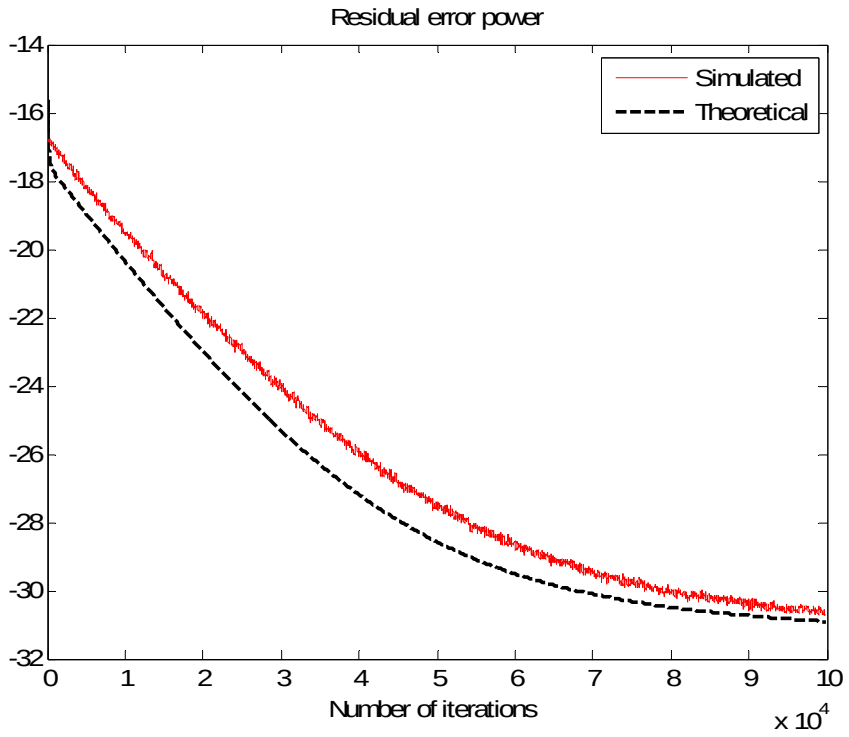

Fig 5.3.17 Residual error power during the second stage

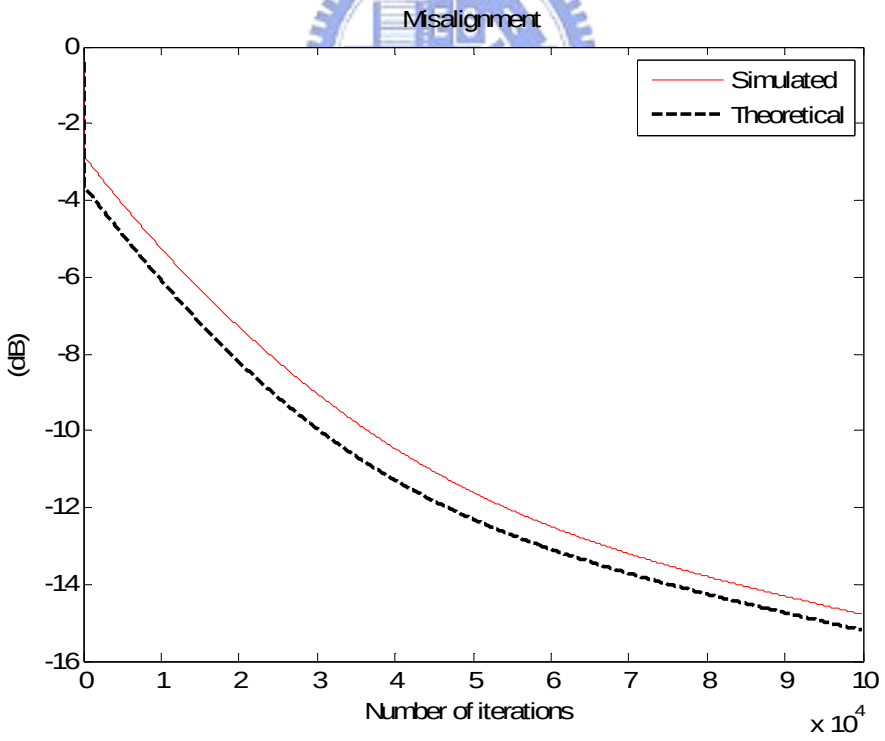

Fig. 5.3.18 Misalignment of combined coefficient weight error under skipping during the second stage

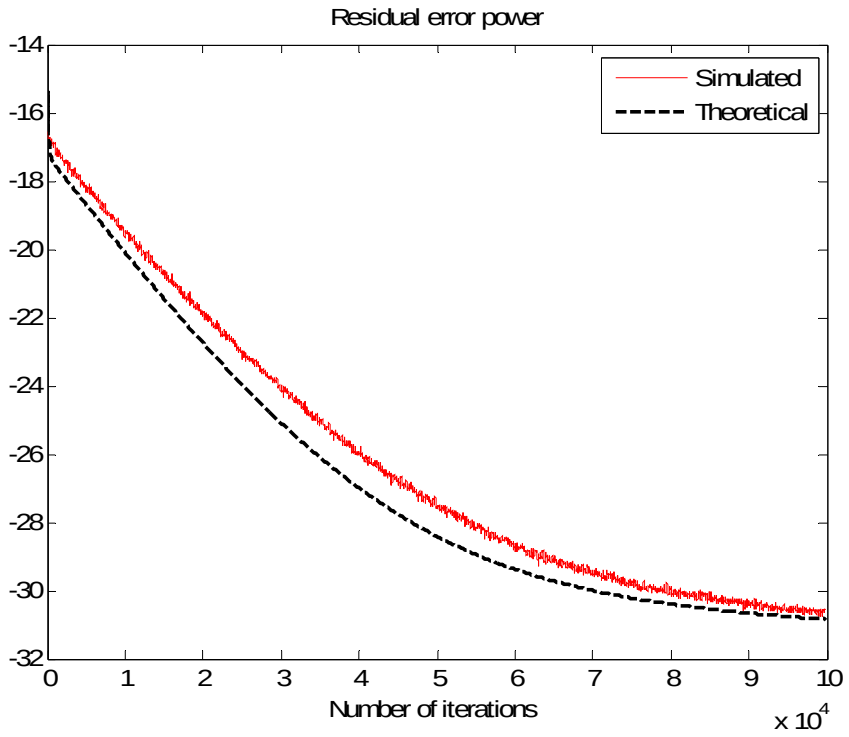

Fig. 5.3.19 Residual error power under skipping during the second stage

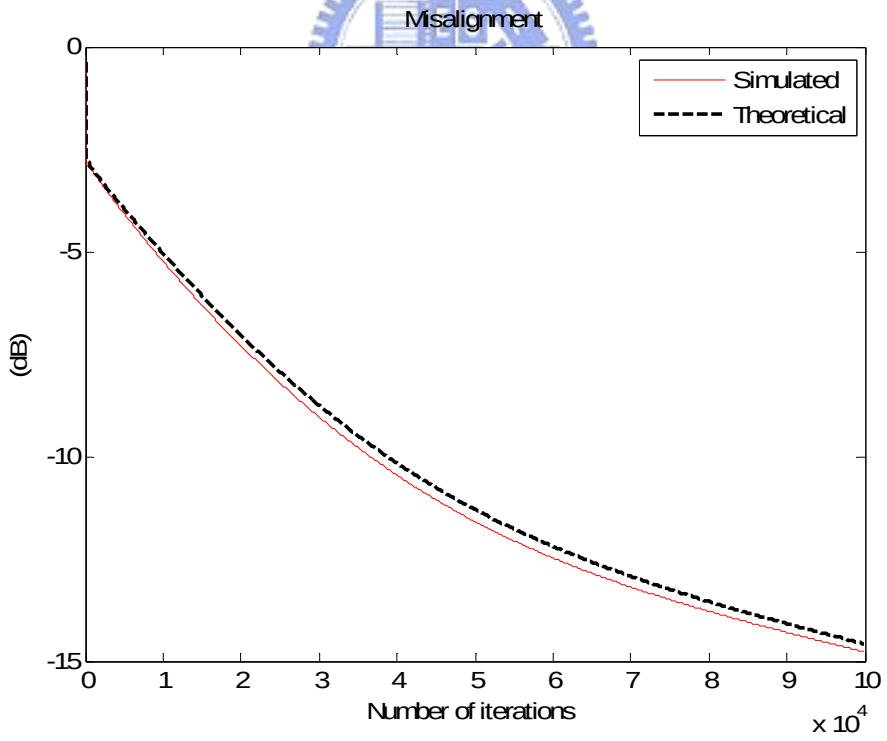

Fig. 5.3.20 Misalignment of combined coefficient weight error under decoupling during the second stage

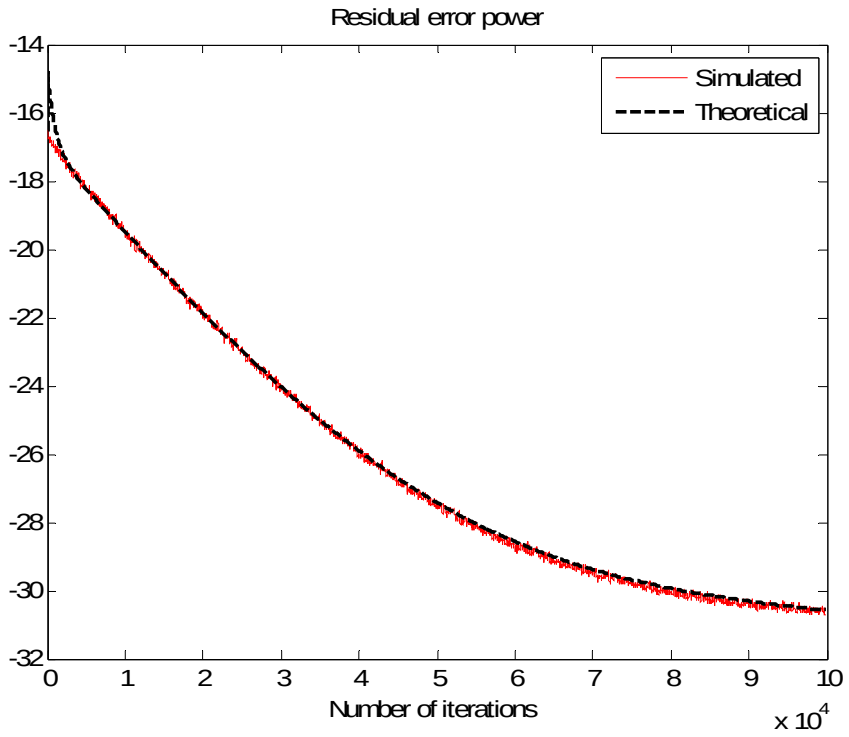

Fig. 5.3.21 Residual error power under decoupling during the second stage

5.3.3 Switching point of two-staged adaptation

Guérin point out that the PWL filter must not adapt until the linear filter has sufficiently converged. Therefore, the pervious simulations were done while the PWL processor was started at most with a steady-state linear filter. In this section, we want to discuss the convergence behavior when the PWL processor was operated with a transient-state linear filter and what the switching point, number of iteration, is a better setting? First, we demonstrate simulated and theoretical curves of misalignment and residual error power of two-staged adaptation with a small step-size $\mu_h = \mu_w$=0.001 in Fig 5.3.22 and 5.3.23, respectively. We found the number of iteration that linear filter converged in first stage is about 3000. In order to observe the issue of switching point, we change it as 3000, 1000 and 300. In Fig. 5.3.24, The two-staged adaptation is robust to the switching point, even the number of iteration 1000 still get better performance than 3000 one. However, if we start to update the

PWL coefficients too fast as the case 300, it will cause a perpetual oscillation at first and worse performance than the others.

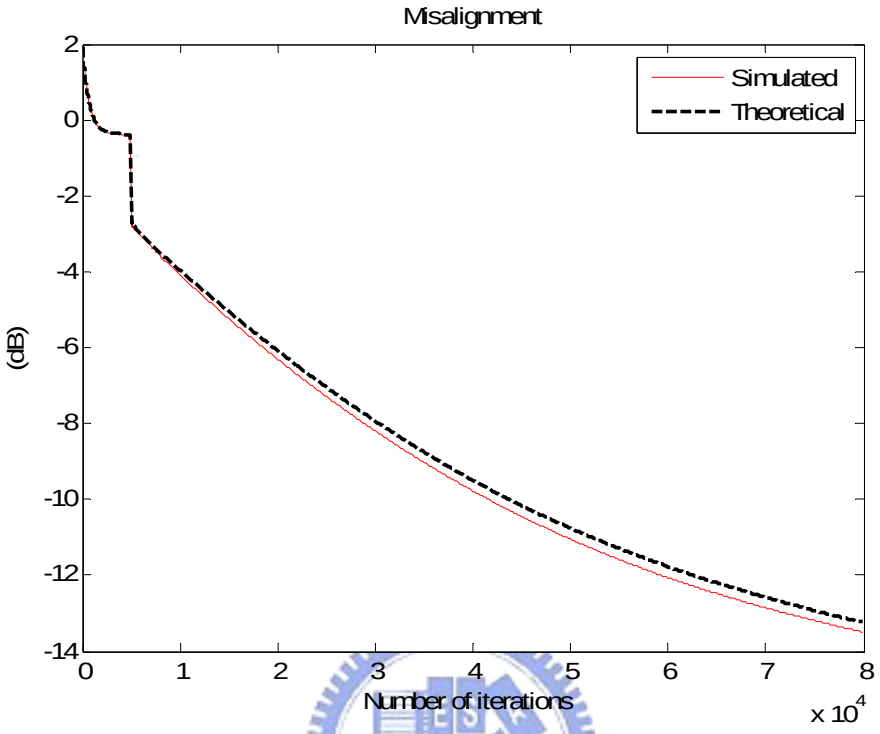

Fig. 5.3.22 Misalignment of combined coefficient weight error of the two-staged adaptation

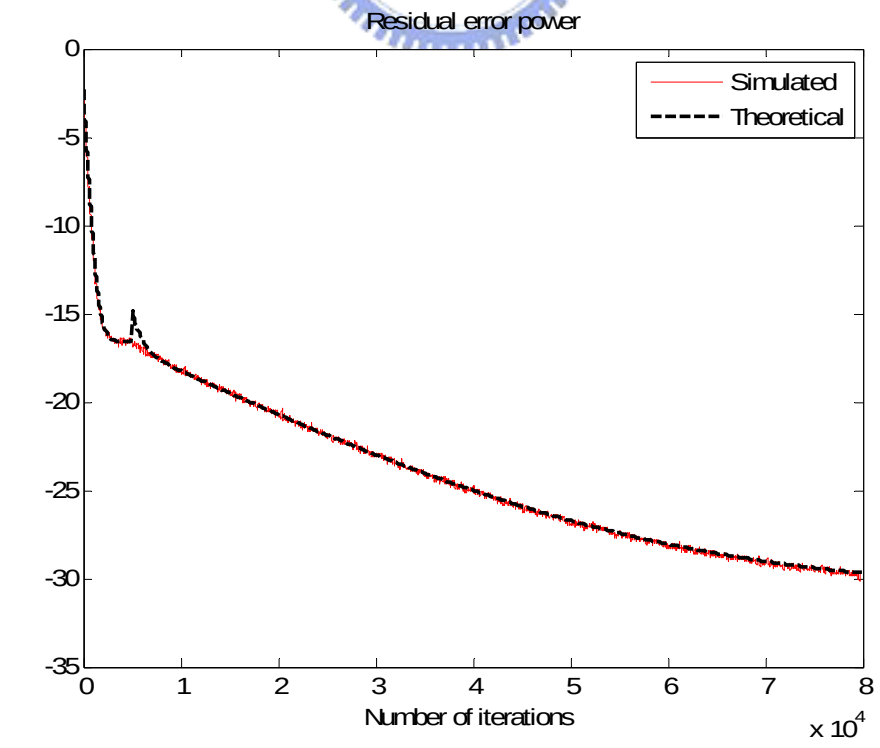

Fig. 5.3.23 Residual error power of the two-staged adaptation

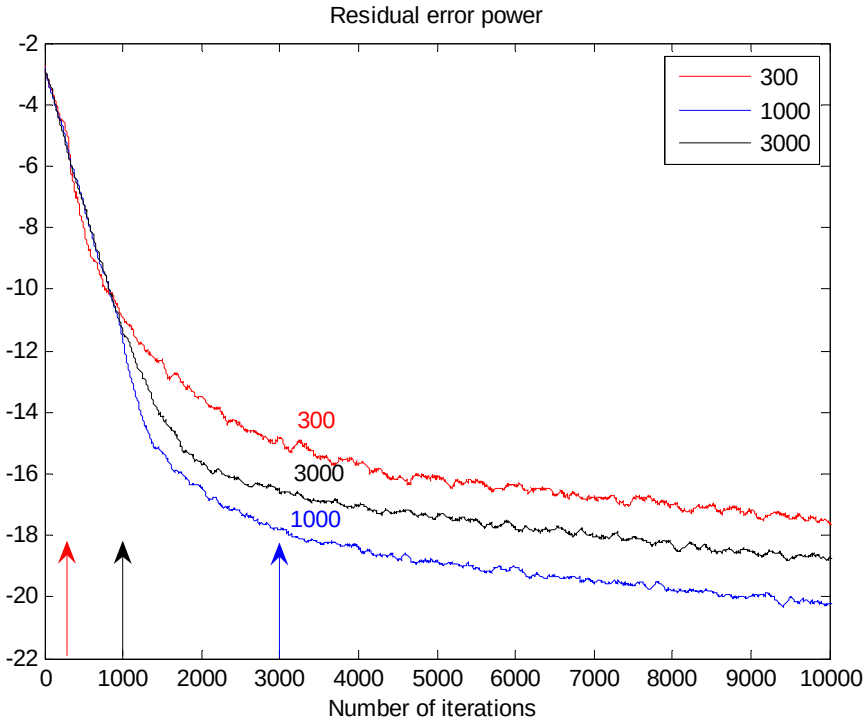

Fig 5.3.24 Residual error power with different switching points

5.3.4 Experiments of the two-staged adaptation

In this section, we preset two experiments of nonlinear acoustic echo canceller. In the first part, to evaluate the performance of proposed approach, PWL structure, we present simulation results obtained for nonlinear acoustic echo to compare the performance with only linear AEC and nonlinear AEC based on polynomial structure. The far end signal, consists of two segments of different volume speech sampled by 8kHz, is represented in bottom of Fig. 5.3.22. The first segment keeps under 0.6 magnitudes and after about 20000 iterations the second one reaches maximum height with time. This signal drives the desktop loudspeaker, low-cost 2.5 inch diameter, and catches the microphone, Creative-MC1000, above the loudspeaker about 6 inches to imitate the cell phone.

Except the joint LMS-type adaptive algorithm in chapter 2, we refer to [7] for adaptation strategy, the same idea in Chapter 3. However, due to the unexpected real

data, it compares standard deviation of linear filter with a set of thresholds to achieve the detection of linear filter state-stay. Moreover, the nonlinearity mostly happens on a high volume input. A simpler detector is based on the power of far end signal is implemented and is compared to a fixed threshold. Therefore, we adjust those thresholds to obtain the optimum Echo Return Loss Enhancement (ERLE) of nonlinear AEC. Fig. 5.3.25 shows the ERLE resulting from Linear AEC and Nonlinear AEC based on PWL structure and polynomial one for different volumes.

First, during first segment the nonlinearity effect is not available, the ERLE of nonlinear AEC are comparable to that obtained with the linear AEC. Second, due to the existing nonlinearity in second segment, the gap in performance is obvious: the mean difference in ERLE is about 3 dB and 2 dB to linear AEC and polynomial structure, respectively. This shows the PWL structure has better performance than the others.

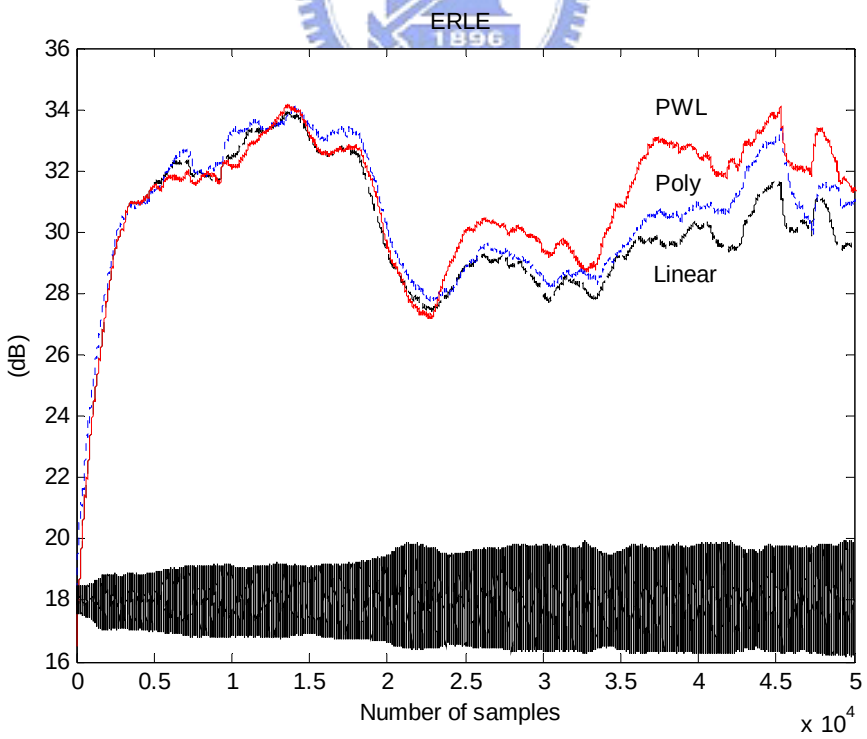

Fig. 5.3.25 ERLE with a two level speech for linear AEC (dash line), nonlinear AEC based on PWL structure (solid line) and Polynomial one (dot line)

In the second experiment, we process two kinds of MTK speech, a woman in mandarin tone and a man in English tone, from a real mechanism of cellular phone with two-staged adaptation. Here, the nonlinear effect only caused by loudspeaker and avoided from analog amplifier. A detector based on the linear filter standard of deviation and the power of far end signal is used. We compare the performance between linear AEC and nonlinear AEC based on PWL structure. In Fig 5.3.26, we demonstrate the speech of a woman in mandarin tone. We found that the ERLE of the PWL structure has a peak 3dB better than one of the Linear AEC around 31K samples and an average 2 dB at high volume than one of the Linear AEC. However, at 2.2k samples the PWL structure is 2dB worse than the linear AEC.

Next, we show the experiment with the speech of a man in English tone. In Fig. 3.3.27, the result with the speech of a man has a more consistent performance at high volume than one of a woman. Moreover, the Peak and average ERLE of PWL structure is 6 dB at 2.3k iterations and 2.5 dB respectively than the linear AEC. We deduce the results that the performance of PWL structure with the speech of a man is better is owing to the aliasing effect.

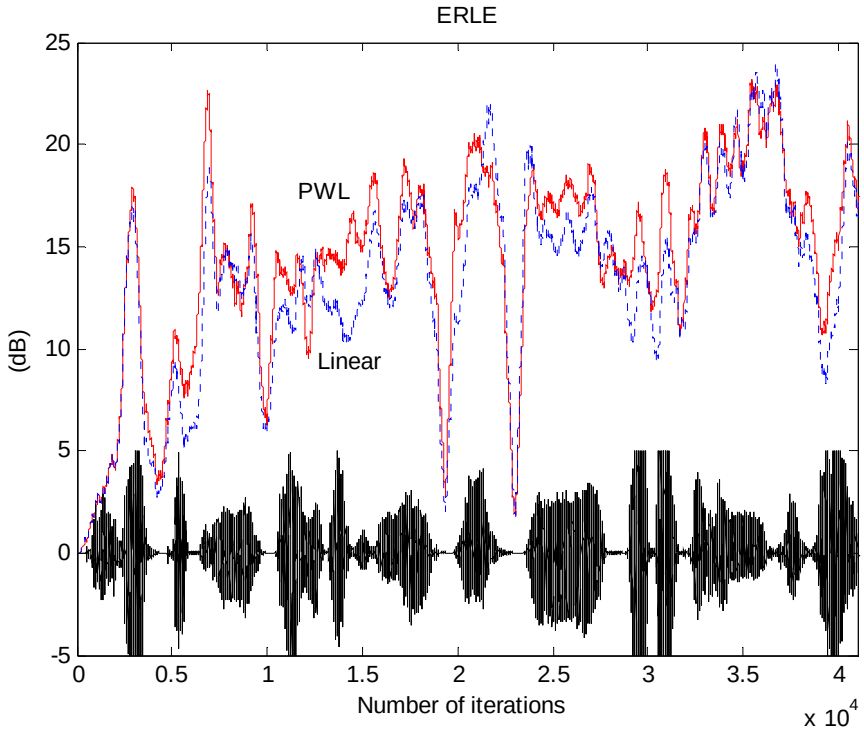

Fig 5.3.26 ERLE with the speech of a woman with tone in mandarin

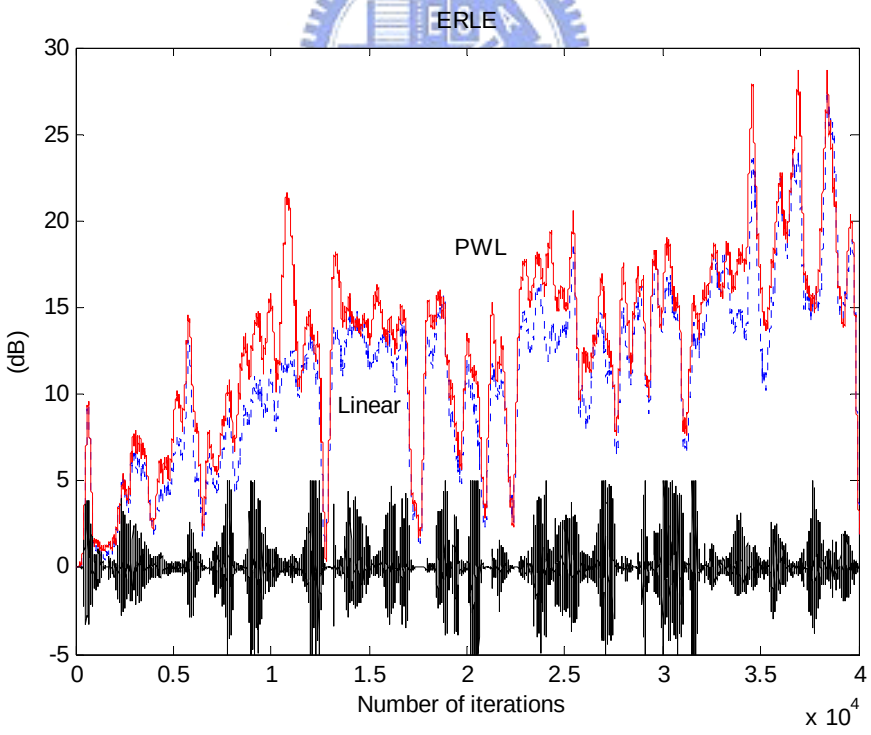

Fig 5.3.27 ERLE with the speech of a man with tone in English

5.4 Controls of step-size

In this section, computer simulations are used to verify the algorithm discussed in Chapter 4. First, we will compare the convergence rate of OTLMS and fixed step-size LMS algorithm in Section 5.4.1. Next, Section 5.4.2 will show the practical OTLMS algorithm and discuss it with difference choices parameter and nonlinear modeling. Moreover, the OTTLMS will be discussed in Section 5.4.3 and its practical type with nonlinear and room impulse response modeling will also be shown in Section 5.4.4. In order to the compact of two-staged adaptation, we apply the OTLMS and OTTLMS of the first stage to the second stage in Section 5.4.5. Finally, a series of experiment will be presented in Section 5.4.6.

5.4.1 OTLMS algorithm

In this section, we want to demonstrate the OTLMS algorithm does work and has better performance than the fixed step-sizes. Further, the theoretical equation in Eq. (4.1.5) also can be verified in the simulation. Fig 5.4.1 shows the step-size of OTLMS was changed form about 0.006 to almost zero. Therefore, we choose the fixed step-size as 0.006 and 0.001 to make a fair comparison. Fig. 5.4.2 show the convergence curve of optimal time-variant and fix step-size LMS algorithm. We found the OTLMS has benefits not only on convergence speed but also steady-state of residual error power.

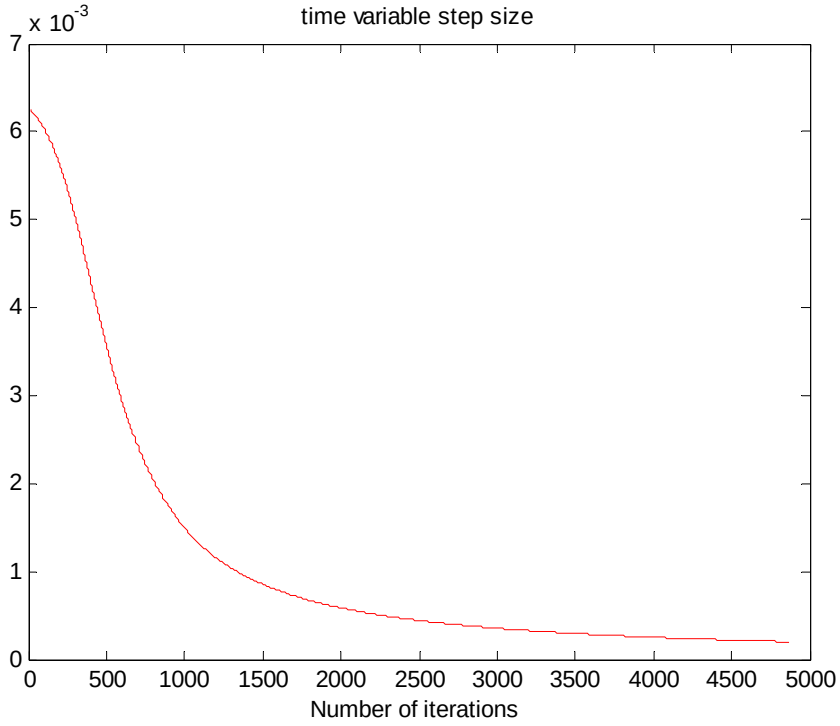

Fig. 5.4.1 Step-size of OTLMS algorithm

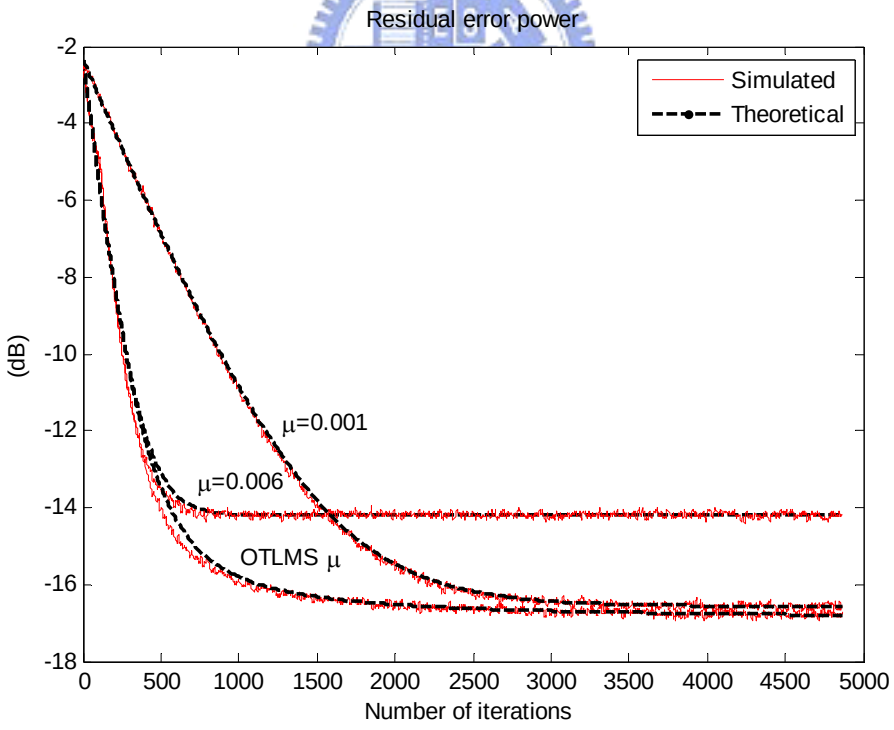

Fig. 5.4.2 Performance comparison between optimal time-variant and fixed step sizes

5.4.2 Practical implements of OTLMS algorithm

We have shown thus far that the use of step-size adjustment improves the convergence rate of the LMS algorithm and reduces the steady-state of residual error power. In this Section, we will go a step to process and discuss the practical OTLMS algorithm.

The most important issue is the parameter $\beta_o = \sigma_v^2 + \left(\sigma_{s_e}^2 - \frac{\sigma_{s,s_e}^4}{\sigma_s^2} \right) \|\mathbf{h}_o\|_2^2$. In order to approximate it, we focus on the nonlinearity. With a mismatch nonlinear modeling, it will result on the prior statistics knowledge $\sigma_s^2, \sigma_{s,s_e}^2$ and σ_{s,s_e}^2. Fro the reason, we want to show does the practical OTLMS is robust to them. We set the optimal PWL coefficients as $c_1 = 1.9050$ $c_2 = -0.8886$ $c_3 = -0.8875$. We use the monotonously decreasing slope model of nonlinear I/O mapping curve A, B and C

The corresponding curves were presented in Fig 5.4.3. Fig 5.4.4 shows the time-variant step-size of these three types nonlinear mapping curve A, B and C. The figures of residual error power were plotted in Fig. 5.4.5. We can see that with B and C, the performances are similar to optimum one. That means the practical OTLMS algorithm is robust to the mismatch modeling. However, if we use mapping curve A (ie, a linear curve), the convergence speed of it is slower than the fixed step–size $\mu_h = 0.006$. As a result, we deduce the modeling is necessary and effective.

Next, with an approximated error on $\hat{\beta}$ from near-end noise or nonlinear modeling error, it might influence the performance of practical OTLMS algorithm. Therefore, we use two different choices of parameter $\hat{\beta}$ (i.e., the five times and one fifth the optimum β_o) to discuss. In Fig. 5.4.6 and 5.4.7 are the corresponding time-variant step-size and residual error power, respectively. Fig 5.4.6 shows that with a larger β, the initial of step-size would be smaller but each time-variant step-size

takes the same time to convergence. The corresponding performances in Fig 5.4.7 have apparent difference that five times β_o has worse convergence speed at start than one-tenth one and even the fixed step-size $\mu_h = 0.006$. However, after that the different choices of OTLMS have comparable performance.

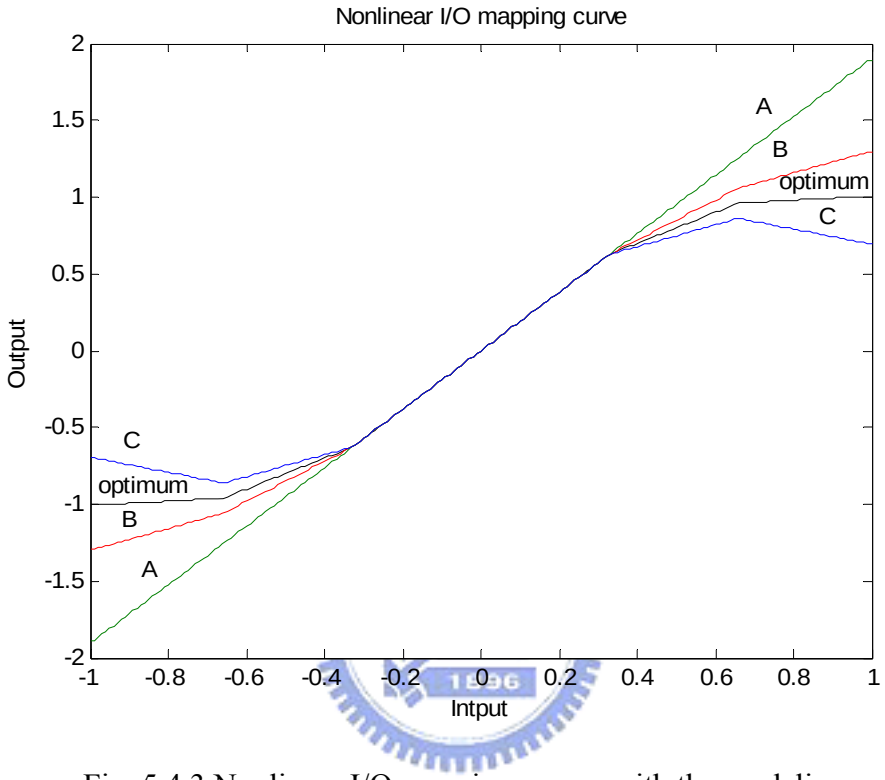

Fig. 5.4.3 Nonlinear I/O mapping curves with the modeling

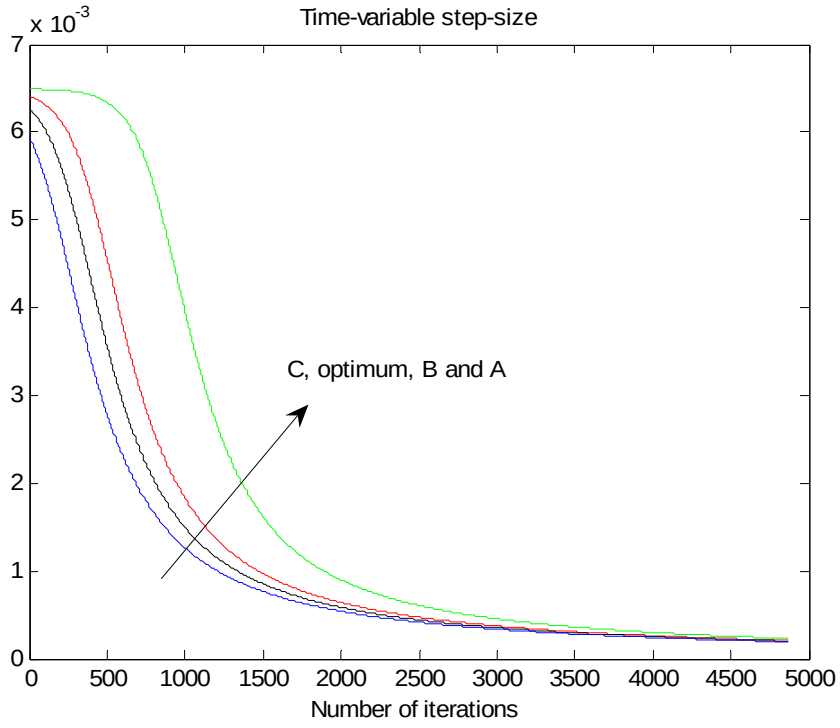

Fig. 5.4.4 Time-variant step-size with different mapping curves

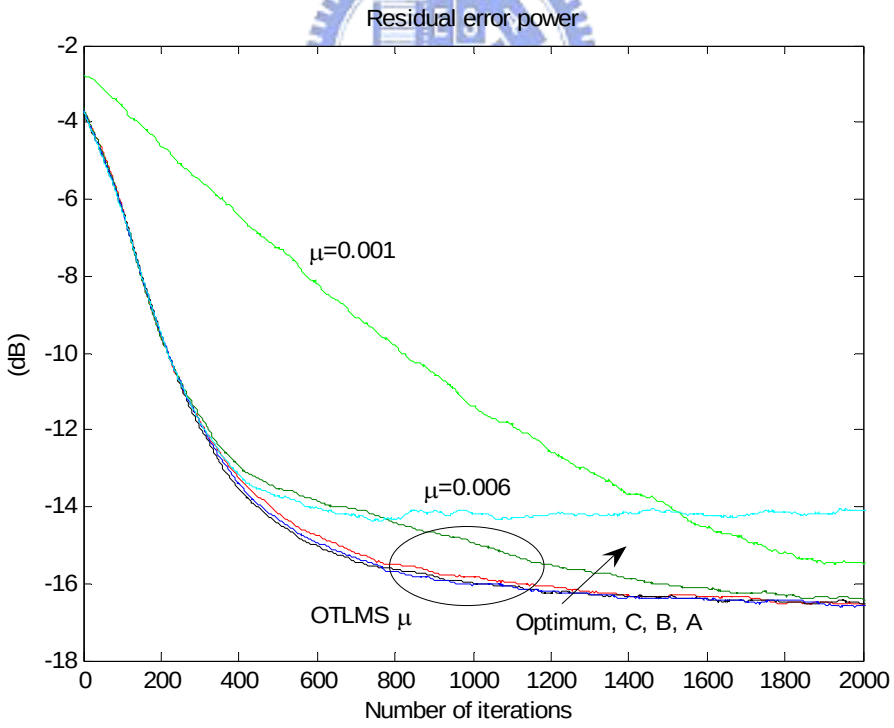

Fig. 5.4.5 Residual error power with different mapping curves

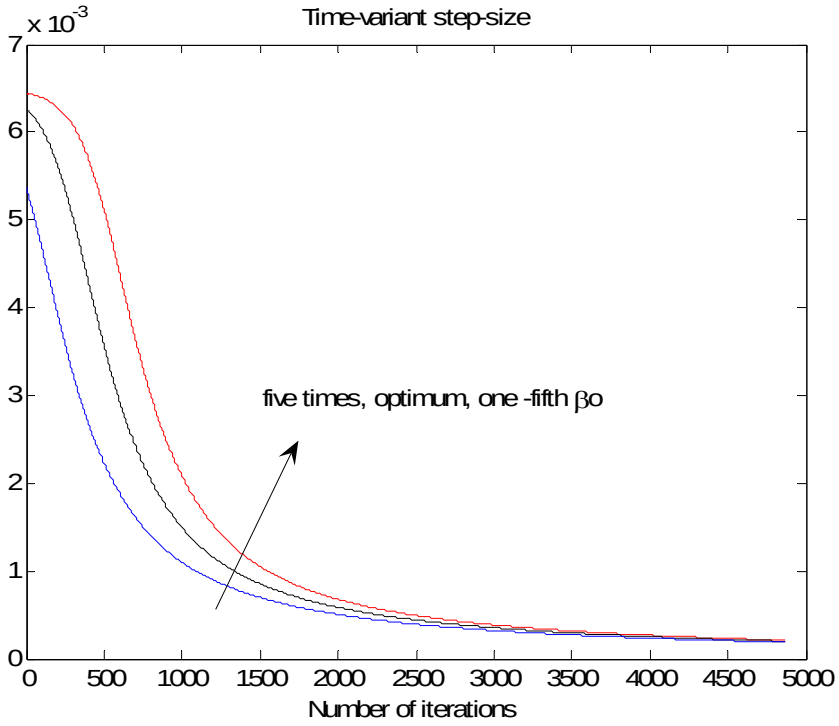

Fig. 5.4.6 Time-variant step-size with different $\hat{\beta}$

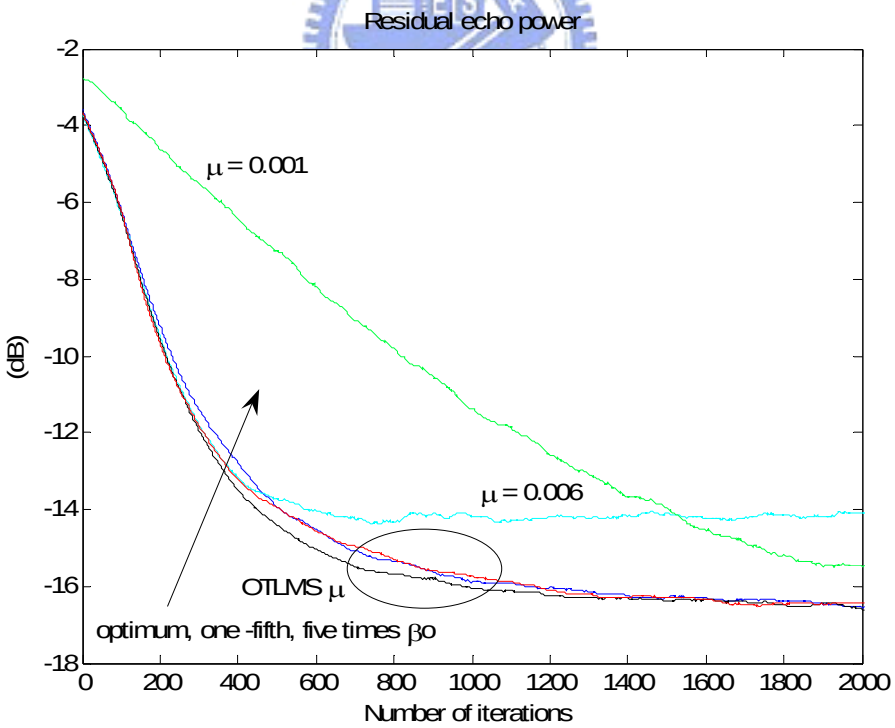

Fig. 5.4.7 Residual error power with different $\hat{\beta}$

Moreover, another issue is the residual error power. Because the fluctuation of speech signal and the changing of room impulse response, the statistics of residual

error power are varying with time. However, the regression of expectation of residual error power in Eq. (4.1.5) also can be replaced by time mean estimate with first-order recursive filtering as follows:

$$\hat{J}_h(n+1) = (1-\lambda)\hat{J}(n) + \lambda e^2(n)$$

where λ is the forgetting parameter close to 0. Here, we continue to use difference choice of the parameters β_o and discuss it with first-order recursive procedure of residual error power. The overall settings are the same as before, the far-end signal is white with uniform distribution, etc…. In Fig. 5.4.8 and Fig 5.4.9 show the time-variant step-size and residual error power, respectively. With the case, one-second β_o, the residual error power can not be reduced in advance because of an unchanged step-size. The robustness of a smaller $\hat{\beta}$ wasn't happened on this procedure. However, two times β_o keeps the performance and has a lower steady-state than optimum β_o. Therefore, using first-order recursive procedure of residual error power, we suggest using a larger $\hat{\beta}$ for implementation.

Finally, we compare two produces, theoretical and first-order recursive, with the same $\hat{\beta}$ in Fig. 5.4.10, 5.4.11 and 5.4.12. the results show the theoretical procedure is outperform to the first-order recursive one

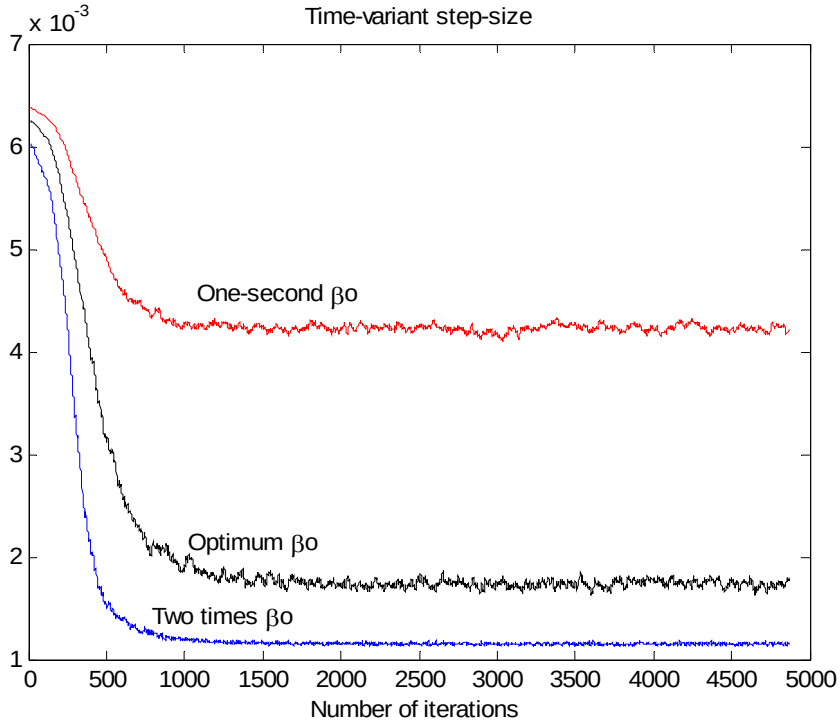

Fig. 5.4.8 Time-variant step-size using the first by using first-order recursive procedure with different $\hat{\beta}$

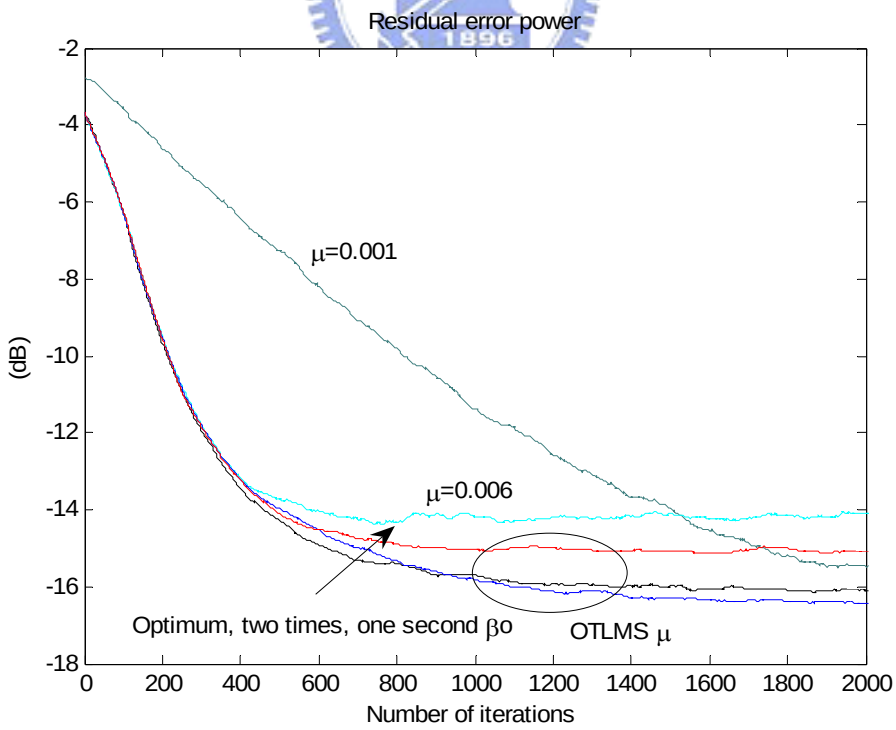

Fig. 5.4.9 Residual error power using the first by using first-order recursive procedure with different $\hat{\beta}$

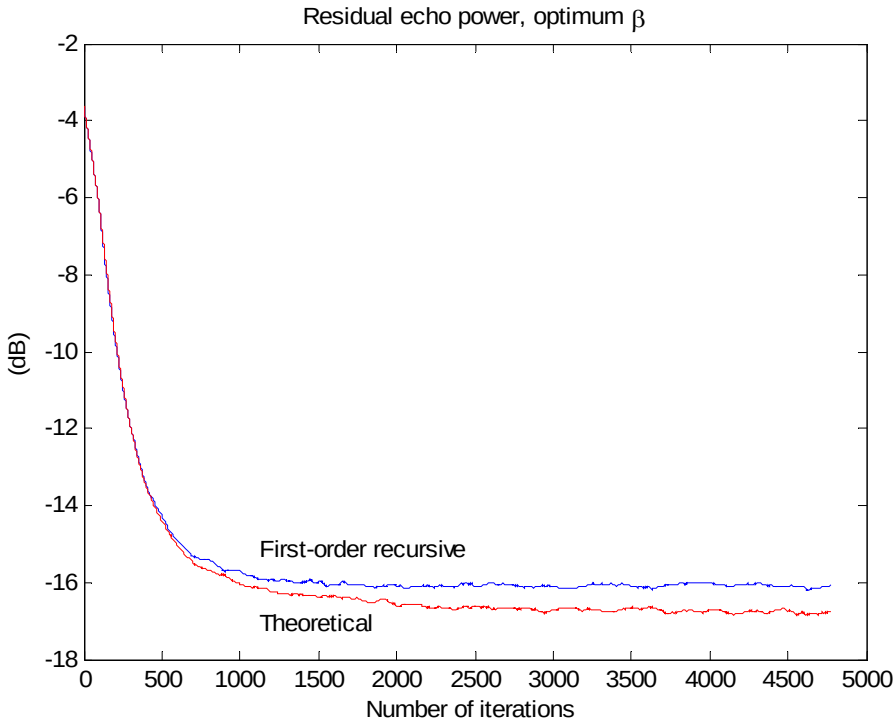

Fig. 5.4.10 Residual error power using the two different procedures with the optimum β_o

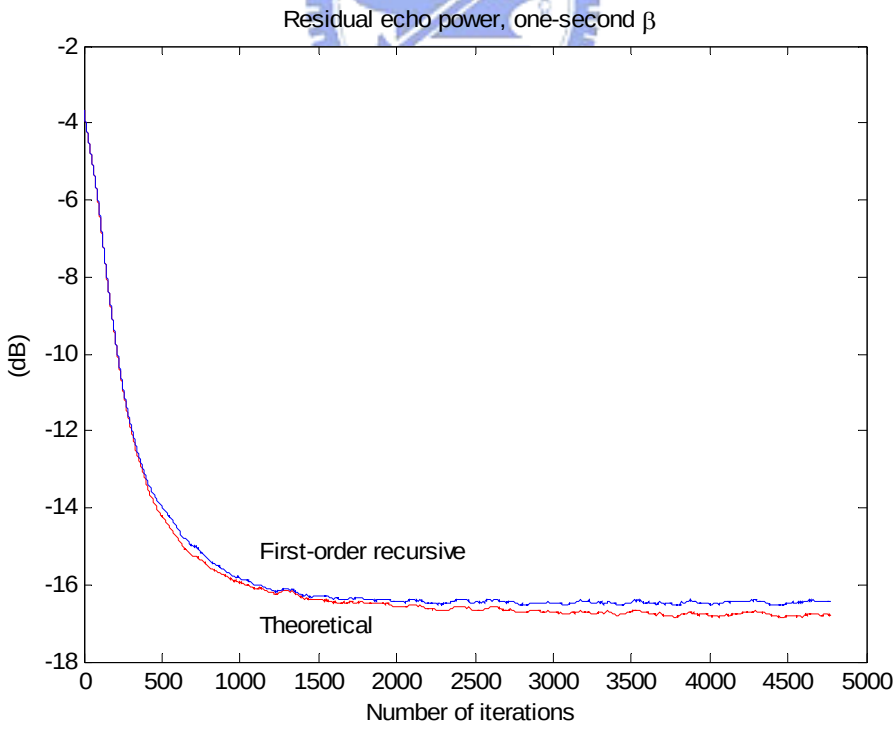

Fig. 5.4.11 Residual error power using the two different procedures with one half of the optimum β_o

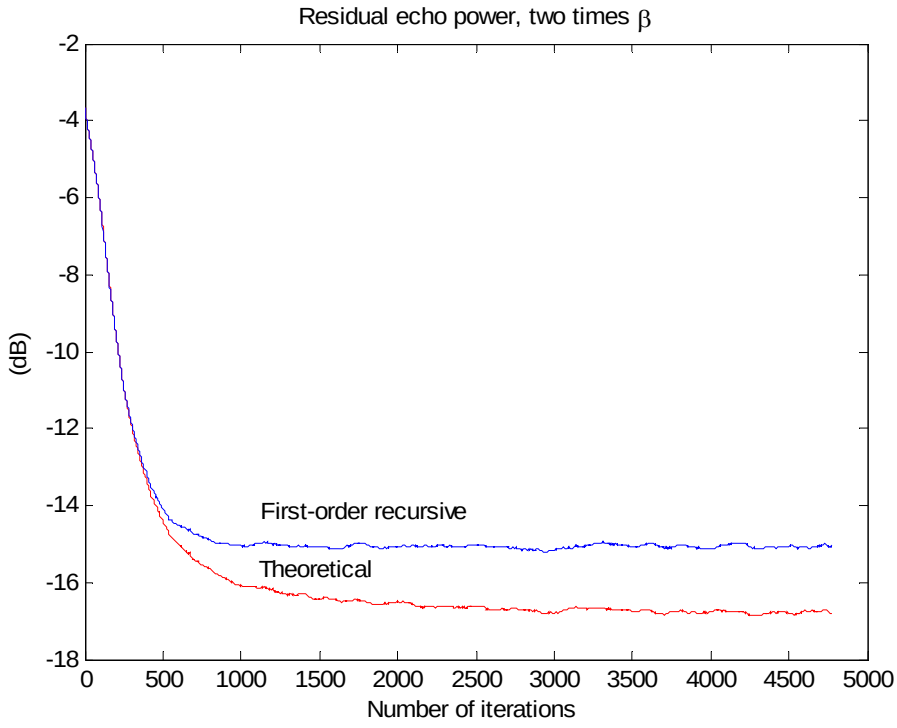

Fig. 5.4.12 Residual error power using the two different procedures with two times of the optimum β_o

5.4.3 OTTLMS algorithm

We have demonstrated that the time-variant step-size adjustment improves the convergence rate and steady-state of the LMS algorithm. In this Section, we will focus on the proposed algorithms using optimum-time & tap-variant step-size. Here, the OTTLMS step-size was used a compact form in Eq. (4.3.5). Figure 5.4.13 illustrates the convergence rate of OTLMS, OTTLMS and fixed step-size as $\mu_h = 0.006, 0.001$. The theoretical curve of OTTLMS was plotted from the regression Eq. (4.3.4) and matched to the simulated curve. We found The OTTLMS algorithm indeed works and has convergence speed about two times than OTLMS one.

Further, we test the approximated form of OTTLMS step-size in Eq. (4.3.6). The performance of it is similar to original one. Therefore, in the following discussion the

approximate form of OTTLMS step-size will be used.

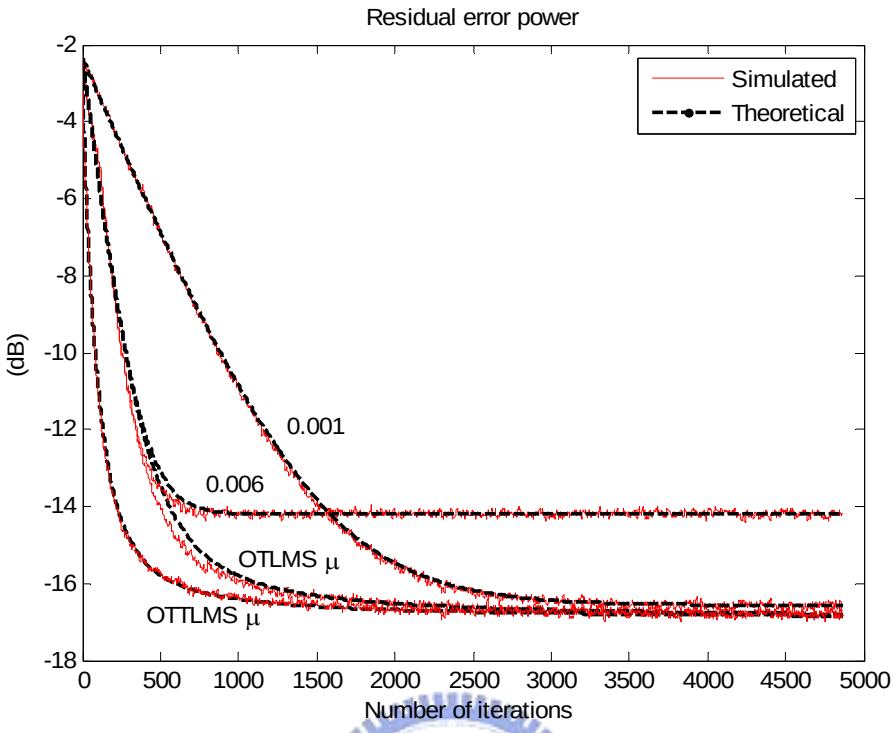

Fig. 5.4.13 Performance comparison of convergence rate for OTLMS, OTTLMS and fixed step-sizes

5.4.4 Practical implement of OTTLMS algorithm

In order to implement the OTTLMS algorithm, we model the nonlinear I/O mapping curve and the room impulse response. In this section, we will exhibit the performance of each mismatching model and discuss it.

First, we discuss the issue of nonlinear effect. According to the monotonously decreasing slope model of the above, we set three different choices of nonlinear mapping curve A, B, C in Fig 5.4.3. Fig 5.4.14 shows that B and C has similar performance to OTLMS algorithm. That means the practical OTTLMS algorithm is robust to the mismatch of nonlinear modeling. However, if we use mapping curve A (i.e., a linear curve), the convergence speed of it is slower than the practical OTLMS. Finally, we found that on the issue of nonlinearity, the results of practical OTTLMS

correspond with ones of practical OTLMS.

Second, we focus on the issue of room impulse response. In section 4.4, we have modeled the RIR $\mathbf{h}_o$ as an exponential decay envelope. The room exponential decay factor γ_h is chosen as three values 0.98, 0.96 and 0.94. We plot the corresponding model in Fig 5.4.15. Fig 5.4.16 presents the performance of different values γ_h and shows that curve B has the best convergence rate due to the modeling. As a more similar envelope, the convergence rate will be faster.

Finally, we combine both of the models to discuss. In Fig. 5.4.17, 5.4.18 and 5.4.19, we use γ_h as 0.96, 0.98 and 0.94 to compare the nonlinearity, respectively. Furthermore, the nonlinear mapping curve is used as A, B and C the same as before. We found the appearance of things that the combined modeling error will make the performance worse from Fig. 5.4.17 to Fig. 5.4.19. In the comparison to nonlinear mapping curves, it presented the robustness the same as before. However, in the case $\gamma_h = 0.94$, the practical OTTLMS with mapping curve C was failed, slower convergence speed than OTLMS. The reason is that a saturated model of nonlinear I/O mapping curve C cascading an exponential decay envelope with larger decay $\gamma_h = 0.94$ will induce a more mismatching model error to original one. Therefore, the performance is poor.

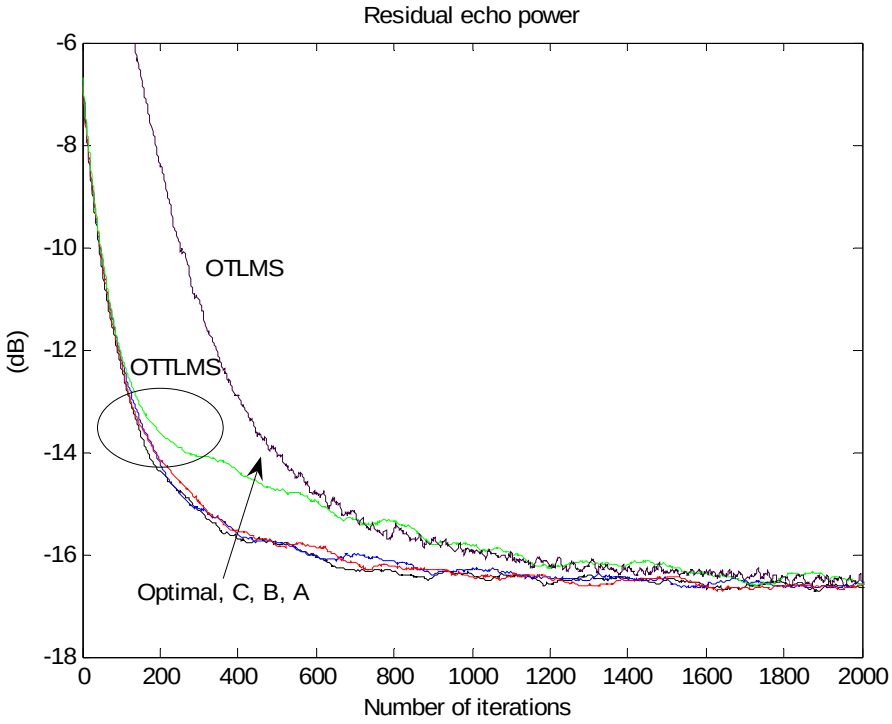

Fig. 5.4.14 Performance comparison of convergence rate for OTLMS, OTTLMS with three different mapping curves

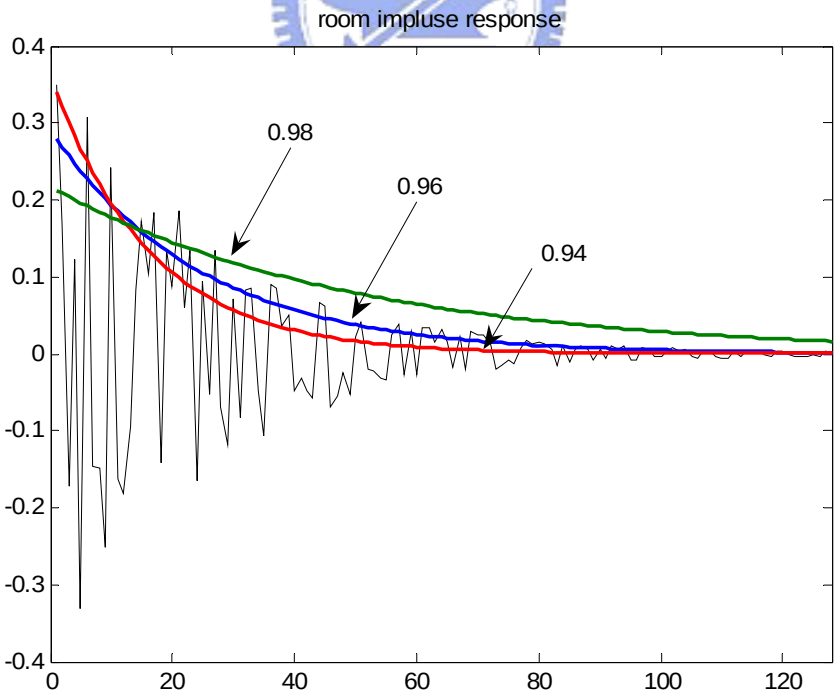

Fig 5.4.15 Model of RIR as an exponential decay envelope

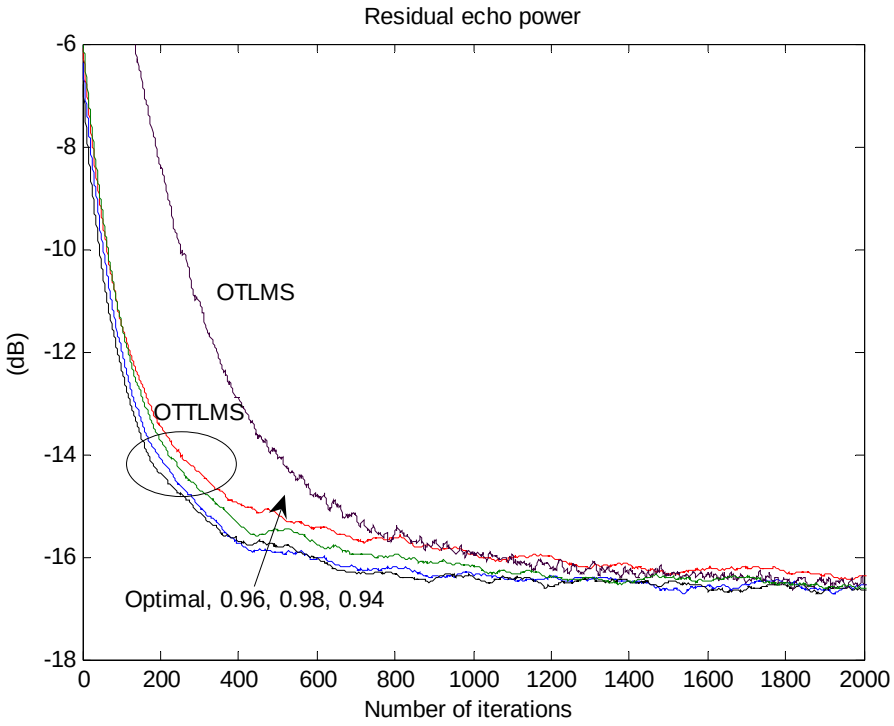

Fig 5.4.16 Performance comparison of convergence rate for OTLMS and OTTLMS with three different choices $\gamma_h = 0.98$, 0.96 and 0.94

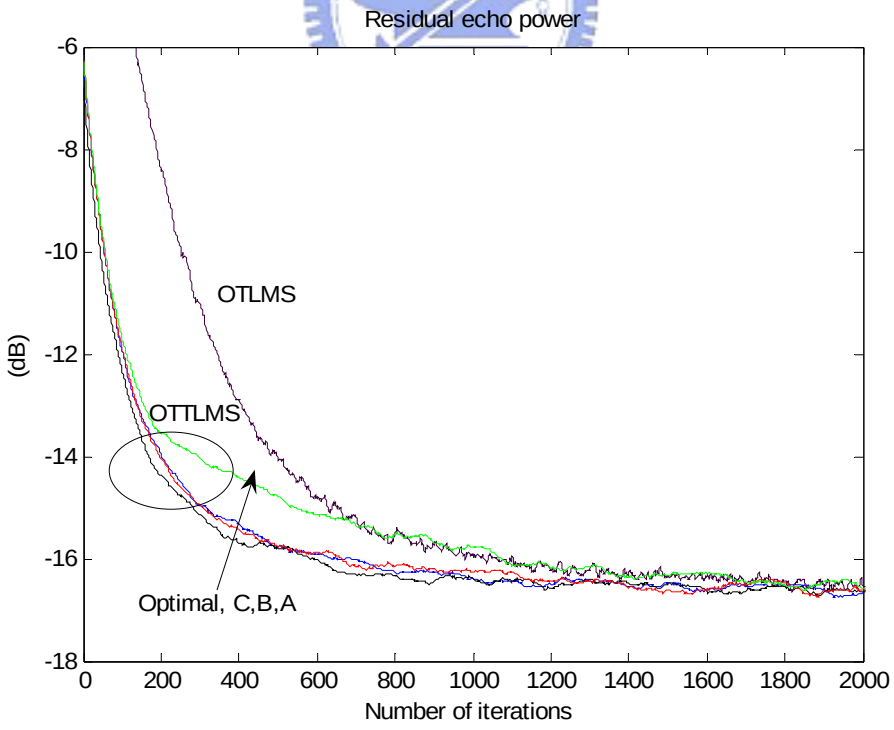

Fig. 5.4.17 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.96$ with three different mapping curves

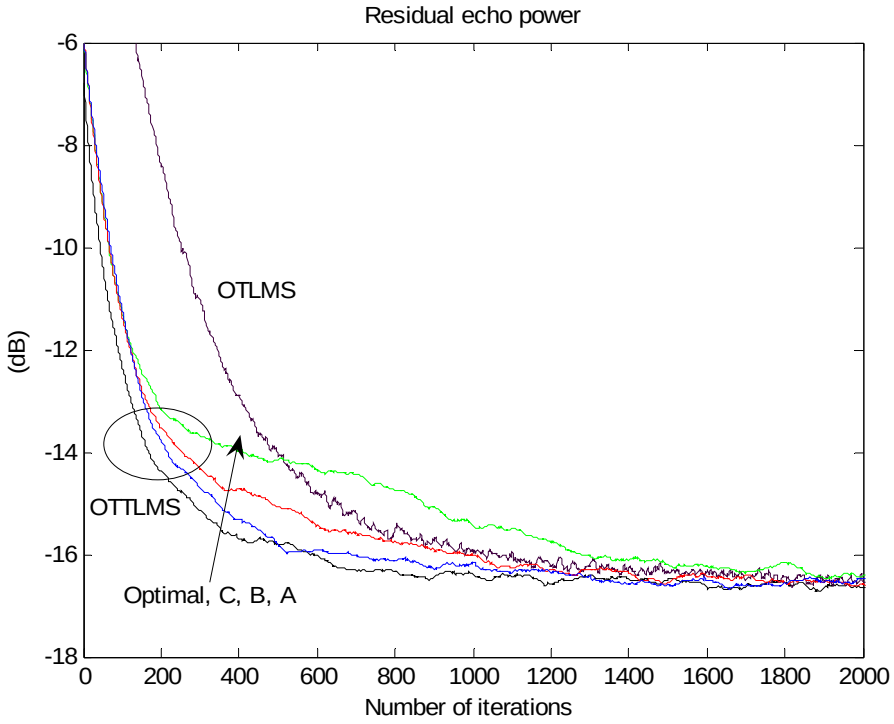

Fig. 5.4.18 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.98$ with three different mapping curves

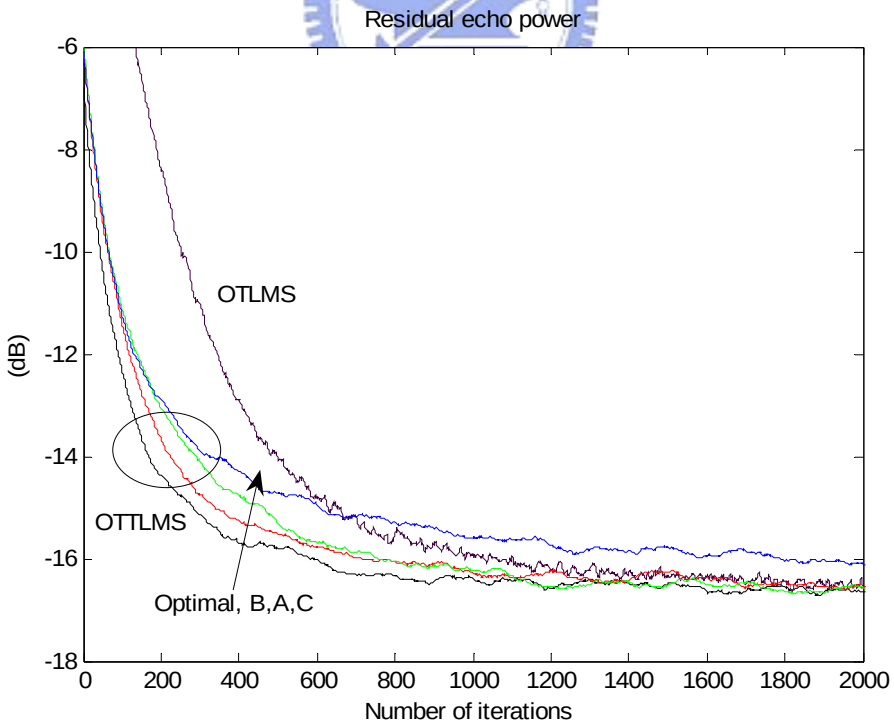

Fig. 5.4.19 Performance comparison of convergence rate for OTLMS and OTTLMS at $\gamma_h = 0.94$ with three different mapping curves

5.4.5 OTLMS and OTTLMS algorithms in the two-staged adaptation

We have showed the benefit of time-variant and time-&tap- variant step-size of the first stage. In the second stage, the both of step-sizes, linear and PWL one are not easy to derive. Moreover, the residual error should be small enough and the linear FIR filter is near the optimum, the variable step-size should also keep small. Therefore, the variant step-size of the second stage is unnecessary. In this section, we continue to use the variant step-size of the first stage on the second stage, and compare to fixed step-size 0.001 and 0.006. the simulation condition are the same as before, a white uniform distributed far end signal, a raise-cosine likely nonlinear I/O mapping curve, exponential decay room impulse response, also set the PWL step-size of all to 0.006 and after 5000 iteration, the nonlinear processor operates. In figure 5.4.20, we found that the variable-step size has comparable performance to the fixed ones during the second stage. The results of the first stage are the same as before.

Next, we apply the procedure by comparing linear filter standard of deviation with a set of thresholds to achieve the detection of linear filter state-stay. Figure 5.4.21 shows OTTLMS has the best performance due to the fastest convergence of linear filter and the OTLMS also get the benefits on transient behavior.

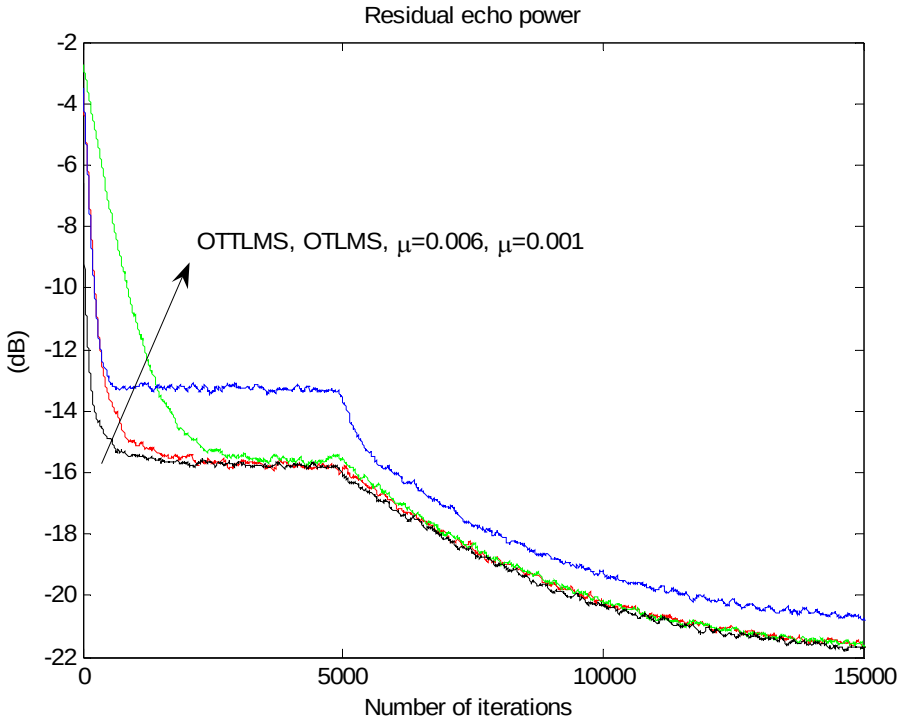

Fig. 5.4.20 Residual error power of OTTLMS, OTLMS, fixed step-size LMS for the two-staged adaptation with fixed switching point

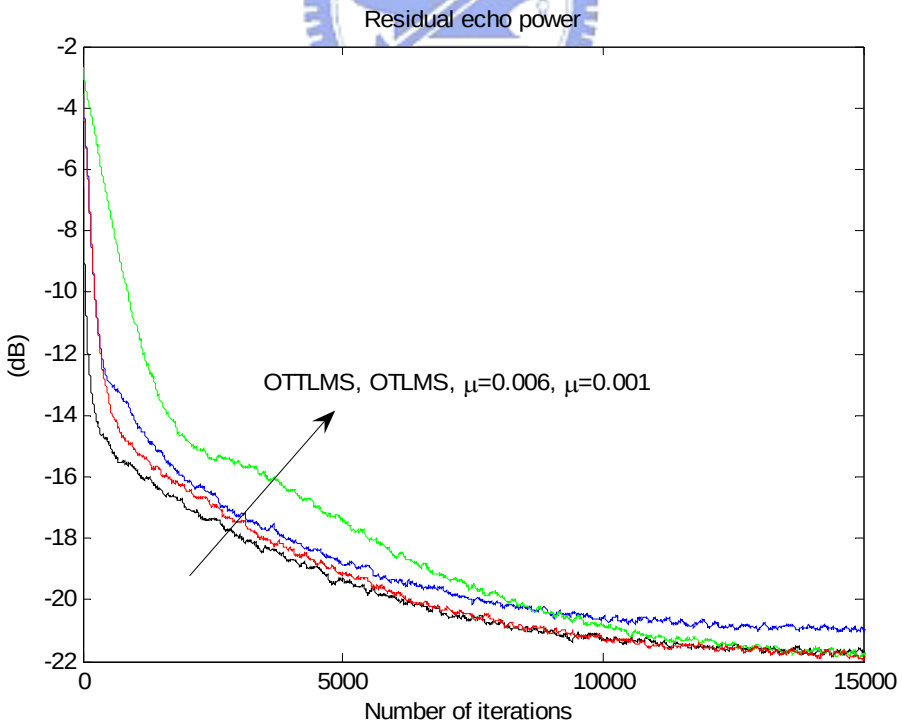

Fig. 5.4.21 Residual error power of OTTLMS, OTLMS, fixed step-size LMS for the two-staged adaptation with detection of state-stay linear filter

5.4.6 Experiments on the step-size controls

For simplicity, we only consider that the nonlinear AEC with an artificial room impulse response and nonlinear loudspeaker. The environment is setting by a raise-cosine likely nonlinear I/O mapping curve and exponential decay room impulse response. We use speech as input signal for verification. In order to avoid fluctuation, the amplitude of speech is almost fixed. Therefore we separate the speech in bottom of Fig. 5.3.22 into two parts with the breakpoint 20000 samples and resample with one-fourth time the original sample rate to approximate a full band signal. We also plot these two speeches in the bottom o Fig 5.4.18 and 5.4.20, respectively

In the following experiment, we will use the NLMS type to discuss. The practical OTNLMS use the first-order recursive procedure of residual error power to implement and the practical OTTNLMS was discussed in section 4.5. Fig 5.4.22 and Fig 5.4.24 show that the practical OTNLMS algorithm do work and has faster convergence rate and better ERLE than fixed step-size $\mu_h = 0.2$ and 1 but OTTNLMS failed. Fig. 5.4.19 and Fig 5.4.21 demonstrated the corresponding time-variant step-size of practical OTNLMS, respectively.

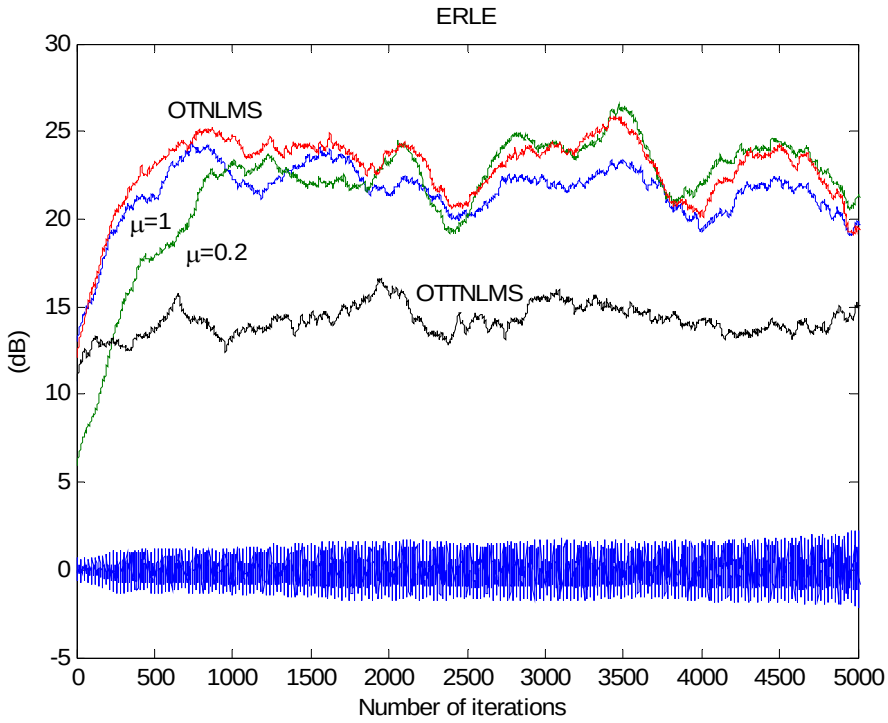

Fig. 5.4.22 ERLE of practical OTTLMS, practical OTLMS, fixed step-size LMS algorithm with the first part of speech and pseudo nonlinear echo

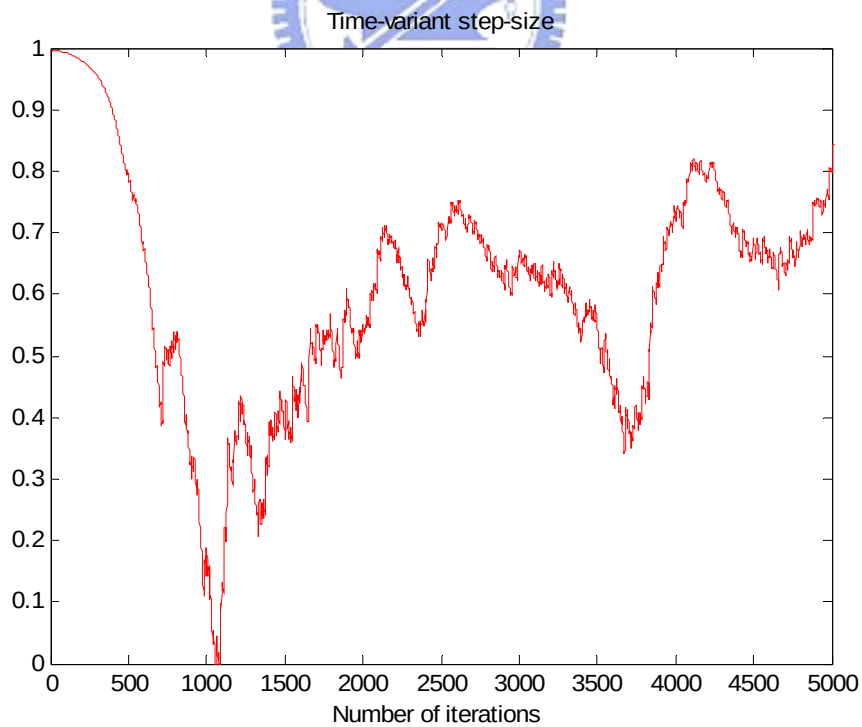

Fig. 5.4.23 Step-size of practical OTLMS algorithm with the first part of speech and a pseudo nonlinear echo

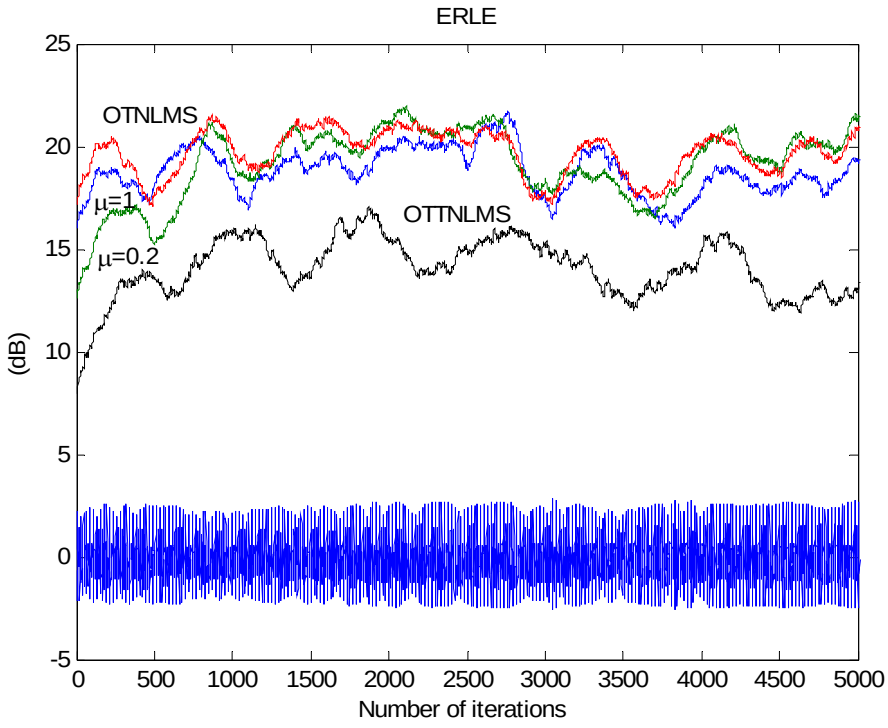

Fig. 5.4.24 ERLE of practical OTTLMS, practical OTLMS, fixed step-size LMS algorithm with the second part of speech and a pseudo nonlinear echo

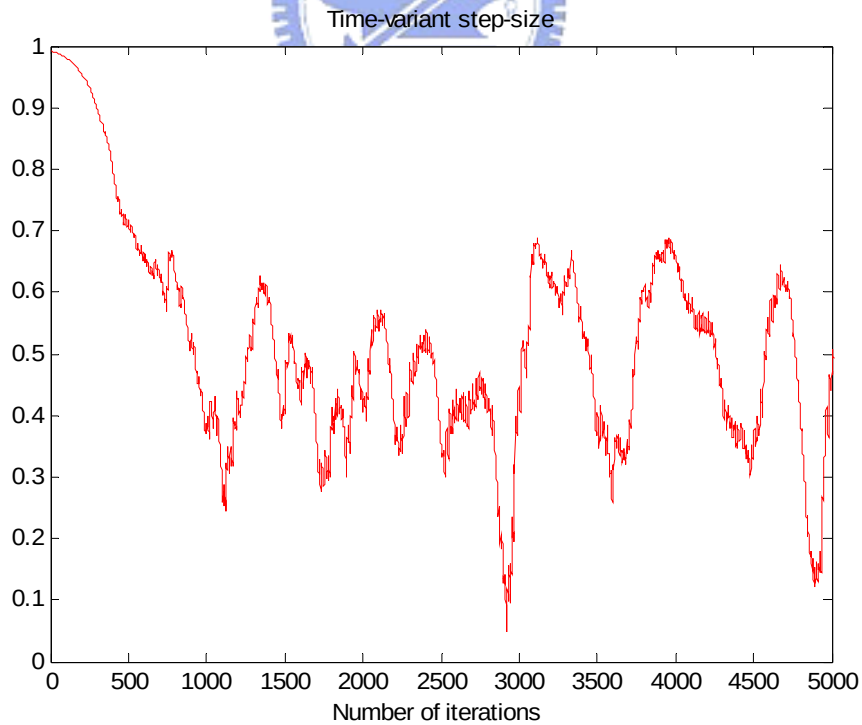

Fig. 5.4.25 Step-size of practical OTLMS algorithm with the second part of speech and a pseudo nonlinear echo

In advance, we use the step-size control on the true echo, the same near-end signal in Section 5.3.4. The separated and decimated procedures were also used on the nonlinear echo. Because the practical OTTNLMS failed, we only do the comparison on practical OTNLMS and fixed step-size NLMS algorithm. In Fig 5.4.26 and Fig 5.4.28, the results show the practical OTNLMS has comparable performance to fixed step-size $\mu_h = 1$. At the same time, the case $\mu_h = 0.2$ always keeps worse performance. That means it doesn't converge due to the varying room impulse response. It can explain why the performance of practical OTNLMS does not outperform the case $\mu_h = 1$.

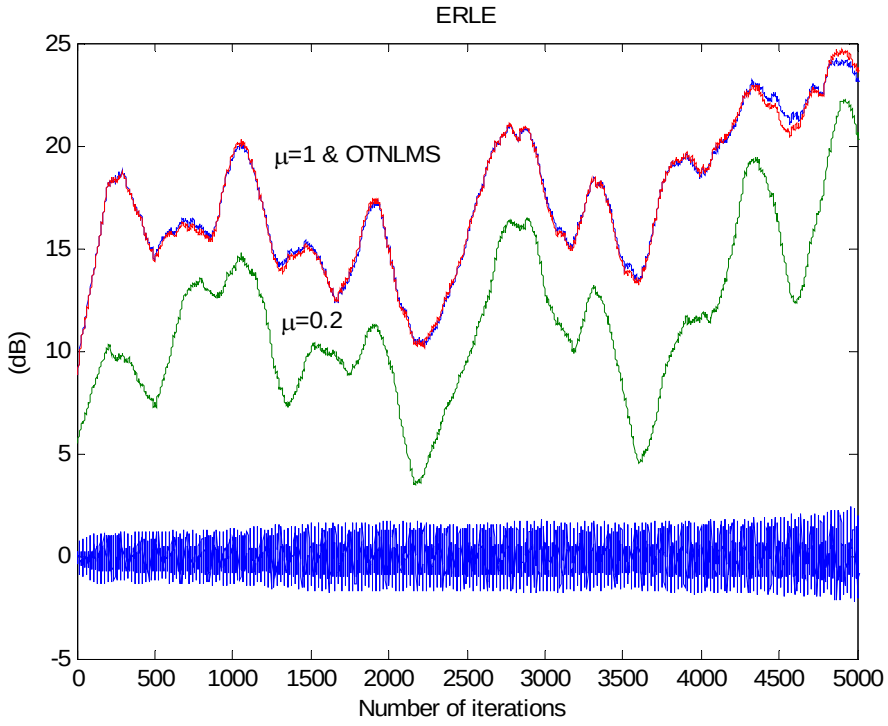

Fig. 5.4.26 ERLE of practical OTLMS, fixed step-size LMS algorithm with the first part of speech and a true nonlinear echo

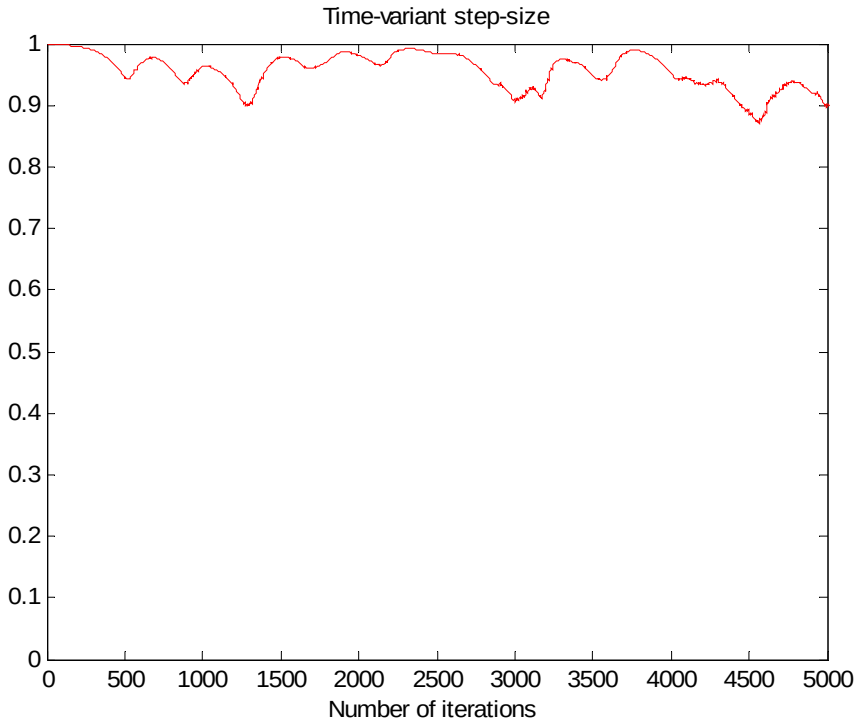

Fig. 5.4.27 Step-size of practical OTLMS algorithm with the first part of speech and a true nonlinear echo

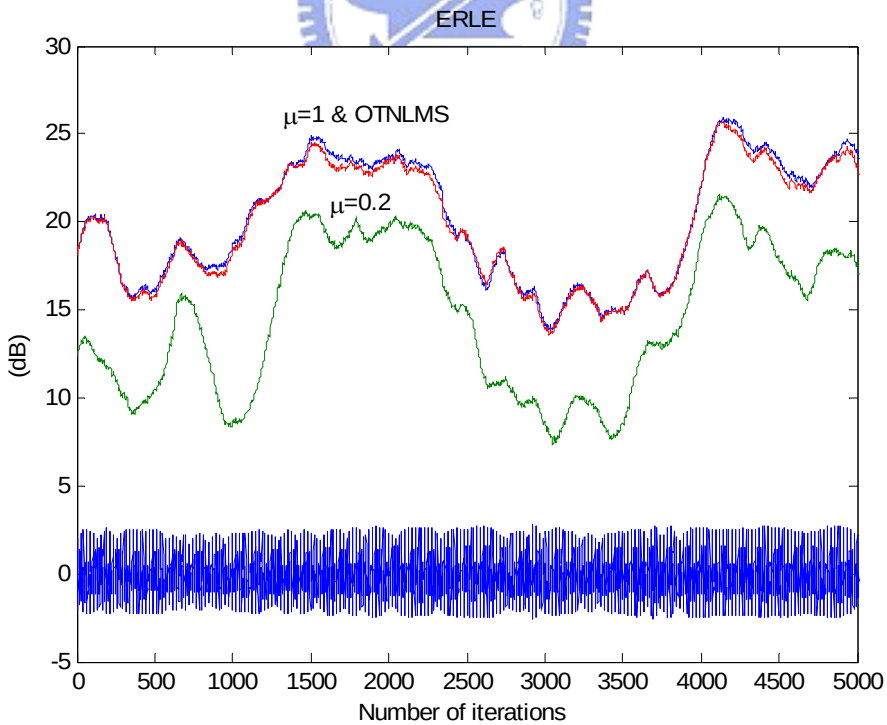

Fig. 5.4.28 ERLE of practical OTLMS, fixed step-size LMS algorithm with second part of speech for true nonlinear echo

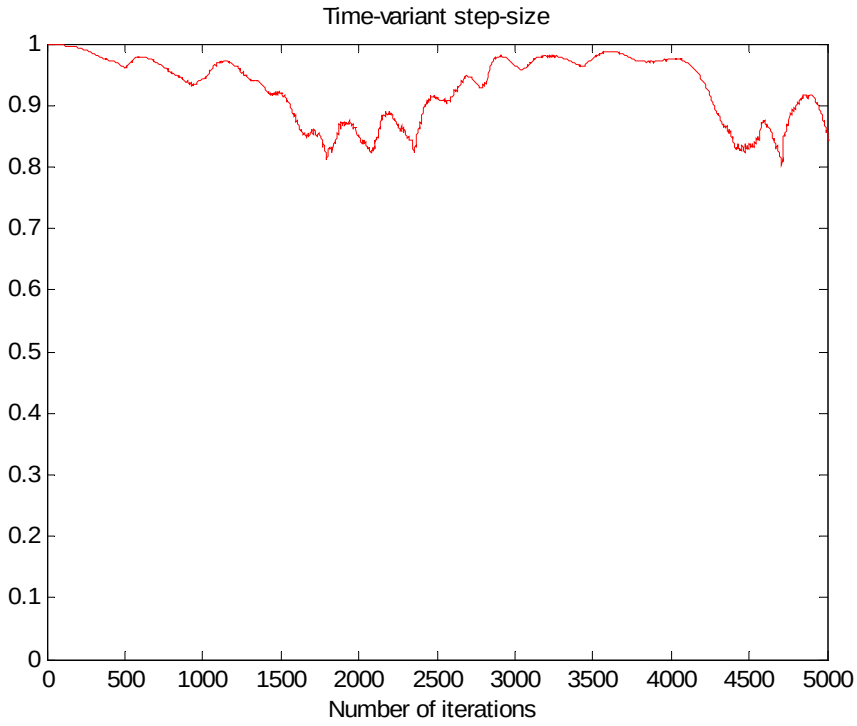

Fig. 5.4.29 Step-size of practical OTLMS algorithm with the second part of speech for a true nonlinear echo

Chapter 6
Conclusions

We have developed the nonlinear AEC system where the scheme is a cascade model in a memoryless PWL processor and linear FIR filter. Its joint LMS adaptive algorithm had lower computation than conventional nonlinear AEC based on polynomial function. Simulation results also have showed that the PWL structure has better performance than polynomial in the case of a raised-cosine I/O mapping curve. For computational cost, we have developed five types of PWL coefficients selective update schemes. They all kept the computation on the matrix multiplication $\mathbf{F}^T(n)\cdot\mathbf{h}(n)$ with M. In the simulation, they all have worked and have benefited the computational efficiency in this case of a raised-cosine I/O mapping curve. We suggested that the Random, Periodic and Variant periodic schemes are good candidates for partial update scheme considering the ERLE convergence performance in cases of a raised cosine and highly saturated mapping functions. The experiment of a real environment, we have show that the PWL processor structure has better performance than a polynomial one and exceed about 2~3 dB more than the linear AEC with different speech.

Since each filter (linear filter or PWL processor) behaves to compensate the other one's misalignment, which can lead to a perpetual oscillating system. In order to overcome this difficulty, we have adopted two-staged algorithm, starting with a linear filter, and then joint PWL and linear coefficients update follow with a steady state linear filter and derived its convergence analysis with two lemma.

In the first stage, only linear coefficients update under a fixed PWL coefficients. We make up the *Lemma 1* with Eq. (3.2.14), (3.2.16), (3.2.17) and (3.2.21) as follow:

Lemma 1: If far end signal $x(n)$ is white, modeling of nonlinear I/O mapping is symmetric in a cascade model and only linear filter coefficients update, then we have the stability criterion of step-size μ_h, the coupled vector $\boldsymbol{\theta}(n)$ of $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\}$ and $E\left\{\boldsymbol{\varepsilon}_h(n)\right\}$, the steady state of $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\}$ and the mean square error $J_h(n)$ as follows:

$$\mu_h < \frac{2}{M\sigma_s^2},$$

$$\boldsymbol{\theta}(n) = \left(\mathbf{I}-\mathbf{A}\right)^{-1}\cdot\mathbf{b} + \mathbf{A}^n\cdot\left(\boldsymbol{\theta}(0) - \left(\mathbf{I}-\mathbf{A}\right)^{-1}\cdot\mathbf{b}\right),$$

$$\lim_{x\to\infty} E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} = \frac{\mu_h M\left(\sigma_v^2 + \sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right) + 2\dfrac{\sigma_{s,s_e}^2\sigma_{s_e}^2}{\sigma_s^4}\left\|\mathbf{h}_o\right\|_2^2\left(1-\mu_h M\sigma_s^2\right)}{2-\mu_h M\sigma_s^2}$$

and

$$J_h(n) = \sigma_v^2 + \sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2 + \mathbf{C}^T\boldsymbol{\theta}(n),$$

where σ_s^2, $\sigma_{s_e}^2$ and σ_v^2 are the variance of $s(n)$, $s_e(n)$ and near-end noise $v(n)$, respectively, σ_{s,s_e}^2 is the covariance of $s(n)$ and $s_e(n)$, $\boldsymbol{\theta}(n) = \left[E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\} \quad E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\right]^T$, $\mathbf{b} = \left[\mu_h^2 M\sigma_s^2\left(\sigma_v^2 + \sigma_{s_e}^2\left\|\mathbf{h}_o\right\|_2^2\right) \quad \mu_h\sigma_{s,s_e}^2\mathbf{h}_o^T\right]^T$

$\mathbf{A} = \begin{bmatrix} \left(1-2\mu_h\sigma_s^2+\mu_h^2M\sigma_s^4\right) & 2\mathbf{h}_o^T\mu_h\sigma_{s_e}^2\left(-1+\mu_h M\sigma_s^2\right) \\ \mathbf{0} & (1-\mu_h\sigma_s^2)\mathbf{I}_M \end{bmatrix}$ and $\mathbf{C} = \left[\sigma_s^2 \quad 2\sigma_{s,s_e}^2\mathbf{h}_o^T\right]^T$.

By examining the matrix $\mathbf{A}$, we can see that the convergence rate is a monotonic increasing function of the step-size μ_h and the variance σ_s^2 of PWL processor output. The steady-state of $E\left\{\left\|\boldsymbol{\varepsilon}_h(n)\right\|_2^2\right\}$ increases with an increase of the near-end noise variance σ_v^2, the step-size μ_h, the PWL output power σ_s^2, and the

nonlinearity factor σ^2_{s,s_e} and $\sigma^2_{s_e}$. The simulation results also have fitted our observations.

In the second stage, linear and PWL coefficients are updated jointly. We make up the *Lemma 2* with Eq. (3.3.2), (3.3.7) and (3.3.8) as follow:

Lemma 2: If far end signal $x(n)$ is white, modeling of nonlinear I/O mapping is symmetric in a cascade model and linear and PWL coefficients are updated jointly with the assumption

$$\boldsymbol{\varepsilon}_h(n) << \mathbf{h}_o\,,\ \ \boldsymbol{\varepsilon}_w(n) << \mathbf{w}_o\,,$$

then we have the second moment of $\boldsymbol{\varepsilon}(n)$ and mean square error $J(n)$ as follows:

$$E\left\{\left\|\boldsymbol{\varepsilon}(n)\right\|_2^2\right\} = \sum_{i=1}^{M+N} \frac{\sigma_v^2 T_i \lambda_i}{2 - T_i\lambda_i} + \left[\left|k_i(0)\right|^2 - \frac{T_i\sigma_v^2}{2-T_i\lambda_i}\right](1-T_i\lambda_i)^{2n}$$

and

$$J(n) = \sigma_v^2 + \sum_{i=1}^{M+N} \frac{\sigma_v^2 T_i \lambda_i}{2 - T_i\lambda_i} + \lambda_i\left[\left|k_i(0)\right|^2 - \frac{T_i\sigma_v^2}{2-T_i\lambda_i}\right](1-T_i\lambda_i)^{2n}.$$

where T_i is the *i*-th diagonal entry of $\mathbf{T} = \begin{bmatrix} \mu_h \mathbf{I}_M & \mathbf{O} \\ \mathbf{O} & \mu_w \mathbf{I}_N \end{bmatrix}$, $k_i(0)$ is the initial value of *i*-th entry of $\mathbf{K}(n) = \mathbf{Q}^T \cdot \boldsymbol{\varepsilon}(n)$ and λ_i is the eigenvalues of $\mathbf{R}_G$.

In the simulation due to an approximately zero eigen-value, the misalignment of the simulated and theoretical curves was purely mismatching. In order to solve this problem, we make two procedures denoted as skipping and decoupling. After that, the theoretical curves are almost fitting to the simulated curves.

The overall analysis also can be extend to a polynomial structure simply by setting the delayed tap mapping matrix $\mathbf{F}(n)$ as

$$\mathbf{F}(n)=\begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M+1) \\ x^2(n) & x^2(n-1) & \cdots & x^2(n-M+1) \\ & & \vdots & \\ x^N(n) & x^N(n-1) & \cdots & x^N(n-M+1) \end{bmatrix}.$$

After we have derived the convergence analysis, we have used the theoretical equation of the first stage to derive the optimum time-variant step-size LMS (OTLMS) and time-&tap-variant step-size LMS (OTTLMS) algorithm due to speed up the convergence rate. Here, we make up two lemmas as follow:

Lemma 3 If far end signal $x(n)$ is white, modeling of nonlinear I/O mapping is symmetric in a cascade model and only linear filter coefficients update, we have the OTLMS algorithm for the first stage of two-staged adaptation as follows:

(1) $e(n)=d(n)-\mathbf{w}^T\cdot\mathbf{F}^T(n)\cdot\mathbf{h}(n)$,

(2) $\mu_{h,OLMS}(n)=\dfrac{1}{M\sigma_s^2}-\dfrac{\beta}{M\sigma_s^2 J_h(n)}$,

(3) $\mathbf{h}(n+1)=\mathbf{h}(n)+\mu_h\mathbf{s}(n)e(n)$,

(4) $J_h(n+1)=\left(1-\mu_{h,OLMS}(n)\sigma_s^2\right)J(n)+\mu_{h,OLMS}(n)\sigma_s^2\beta$,

where $\beta=\sigma_v^2+\left(\sigma_{s_e}^2-\dfrac{\sigma_{s,s_e}^4}{\sigma_s^2}\right)\left\|\mathbf{h}_o\right\|_2^2.\sigma_s^2$, $\sigma_{s_e}^2$ and σ_v^2 are the variance of $s(n)$, $s_e(n)$ and near-end noise $v(n)$, respectively, σ_{s,s_e}^2 is the covariance of $\mathbf{s}(n)$.

Simulation results have shows the OTLMS has benefits not only on convergence speed but also steady-state of residual error power. We also have developed its practical form for implementation. In simulation, we have found it is robust to a smaller $\widehat{\beta}$ and the choosing on $\widehat{\beta}$ would be smaller in order to keep the performance. However, the regression of expectation of residual error power in Eq. (4.1.5) also can be replaced by time mean estimate with first-order recursive filtering as follows:

$$\widehat{J}_h(n+1) = (1-\lambda)\widehat{J}(n) + \lambda e^2(n)$$

The robustness of a smaller $\hat{\beta}$ wasn't happened on this procedure and we suggest using a larger $\widehat{\beta}$ for implementation.

Next, we focus on the OTTLMS algorithm.

Lemma 4 If far end signal $x(n)$ is white, modeling of nonlinear I/O mapping is symmetric in a cascade model and only linear filter coefficients update, we have the OTTLMS algorithm for the first stage of two-staged adaptation as follows:

(1) $e(n) = d(n) - \mathbf{w}^T \cdot \mathbf{F}^T(n) \cdot \mathbf{h}(n)$,

(2) $\mathbf{U}_{OLMS}(n) = \begin{pmatrix} \mu_{h,0,OLMS}(n) & & 0 \\ & \ddots & \\ 0 & & \mu_{h,M-1,OLMS}(n) \end{pmatrix}$, where

$$\mu_{h,k,OLMS}(n) = \frac{2\sigma^2_{s,s_e} h_{o,k} \cdot \{\varepsilon_{h,k}(n)\} + \sigma_s^2 g_k(n) + \frac{\sigma^4_{s,s_e}}{\sigma_s^2} h^2_{o,k}}{\sigma_s^4 trace(\mathbf{R}_h(n)) + 2\sigma_s^2\sigma^2_{s,s_e} E\{\boldsymbol{\varepsilon}_h^T(n)\}\mathbf{h}_o + \sigma_v^2\sigma_s^2 + \sigma_s^2\sigma^2_{s_e}\|\mathbf{h}_o\|_2^2}$$

(3) $\mathbf{h}(n+1) = \mathbf{h}(n) + \mathbf{U}(n) \cdot \mathbf{s}(n) e(n)$,

(4) $g_k(n+1) = (1 - \sigma_s^2 \mu_{h,k,OLMS}(n)) g_k(n) + \mu_{h,k,OLMS} \frac{\sigma^4_{s,s_e}}{\sigma_s^2} h^2_{o,k}$,

(5) $E\{\varepsilon_{h,k}(n+1)\} = (1 - \sigma_s^2 \mu_{h,k,OLMS}(n)) E\{\varepsilon_{h,k}(n)\} - \mu_{h,k,OLMS}\sigma_s^2\sigma^2_{s,s_e} h^2_{o,k}$.

where, $\mu_{h,k}(n)$ is the k-th individual tap step-size, σ_s^2, $\sigma^2_{s_e}$ and σ_v^2 are the variance of $s(n)$, $s_e(n)$ and near-end noise $v(n)$, respectively, σ^2_{s,s_e} is the covariance of $s(n)$ and $g_k(n)$ is the k-th linear coefficient error variance.

In simulation, We have found The OTTLMS algorithm indeed works and has convergence speed about two times than OTLMS one. Moreover, its practical form also have work and been robust to the mismatch of nonlinear modeling and the RIR.

Finally, we also showed a series experiments for practical implementation. First,

we only consider that the nonlinear AEC with an artificial room impulse response and nonlinear loudspeaker. The practical OTNLMS have had faster convergence rate and lower ERLE than fixed step-size but OTTNLMS failed. In advance, we use the step-size control on the true echo. However, the performance did not benefit from the variant step-size.

The future work can be: (1) find the optimum switching point by using the theoretical analysis, (2) solve an approximately zero eigen-value of convergence analysis in second stage, (3) the optimum time-variant and time-&tap-variant step-sizes of linear filter and PWL processor for the second stage (4) use our algorithms to process a real echo and work.

Appendix

In this appendix, we are trying to simplify the third term, the 5-th, 7-th and 9-th in Eq.(4.3.2) by using the white property of PWL processor output $\mathbf{s}(n)$. Further, we will use the results to deduce the three terms $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}$, $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}$ and $E\left\{\mathbf{s}_e(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e(n)\right\}$

First, we denote the third term as

$$\mathbf{B} = E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{R}_h(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\} \tag{A.1}$$

With the assumption, $\boldsymbol{\varepsilon}_h(n)$ is uncorrelated to $\mathbf{s}(n)$, kl-th term in Eq.(1) is given by

$$b_{kl} = \sum_{p=0}^{M-1}\sum_{q=0}^{M-1} a_{pq}E(s_k s_p s_q s_l)$$

where a_{pq} is the pq-th entry of the correlation matrix $\mathbf{R}_h(n)$ and s_k is the k-th entry of $\mathbf{s}(n)$. With white property of PWL processor output $\mathbf{s}(n)$, for $k = l$ we have

$$\begin{aligned} b_{kk} &= \sum_{p=0}^{M-1}\sum_{q=0}^{M-1} a_{pq}E\left\{s_k^2 s_p s_q\right\} \\ &= a_{kk}E\left\{s_k^4\right\} + \sum_{p=0}^{M-1} a_{pp}E\left\{s_k^2\right\}E\left\{s_p^2\right\} \\ &= a_{kk}m_{s^4} + \sum_{p=0}^{M-1} a_{pp}\sigma_s^4 \end{aligned}$$

and for $k \neq l$, we have

$$\begin{aligned} b_{kl} &= \sum_{p=0}^{M-1}\sum_{q=0}^{M-1} a_{pq}E\left\{s_k s_p s_q s_l\right\} \\ &= a_{kl}E\left\{s_k^2\right\}E\left\{s_l^2\right\} + a_{lk}E\left\{s_l^2\right\}E\left\{s_k^2\right\} \\ &= 2a_{kl}\sigma_s^4 \end{aligned}$$

Therefore, we can obtain the matrix form of $\mathbf{B}$ as

$$\mathbf{B} = \sigma_s^4\left[2\mathbf{R}_h(n) - 3D\left(\mathbf{R}_h(n)\right) + trace\left(\mathbf{R}_h(n)\right)\cdot\mathbf{I}\right] + m_{s^4}D\left(\mathbf{R}_h(n)\right) \tag{A.2}$$

where operator D is to keep the diagonal entry and zero the others on the matrix.

Similarly, the 5-th, 7-th and 9-th term in Eq. (4.3.2) can be also denoted as

$$\mathbf{C} = E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}, \tag{A.3}$$

$$\mathbf{D} = E\left\{\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\cdot\mathbf{h}_o\cdot E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\} = \mathbf{C}^T,$$

$$\mathbf{E} = E\left\{\mathbf{s}(n)\cdot\mathbf{s}_e(n)^T\cdot\mathbf{h}_o\cdot\mathbf{h}_o^T\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}, \tag{A.4}$$

respectively. Applying the same procedure, we have

$$\begin{aligned}\mathbf{C} = \mathbf{D}^T = \sigma_s^2 m_{s,s_e}\left[2E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T - 3D\left(E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\right) + trace\left(E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\right)\cdot\mathbf{I}\right] \\ + m_{s^3,s_e} D\left(E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T\right),\end{aligned}$$

$$\mathbf{C}+\mathbf{D} = \mathbf{U}^2(n)\cdot\left[\sigma_s^2 m_{s,s_e}\left[2\mathbf{J} - 3D\left(\mathbf{J}\right) + trace\left(\mathbf{J}\right)\cdot\mathbf{I}\right] + m_{s^3,s_e} D\left(\mathbf{J}\right)\right], \tag{A.5}$$

$$\mathbf{E} = \sigma_s^2\sigma_{s_e}^2\left[2\mathbf{h}_o\cdot\mathbf{h}_o^T - 3D\left(\mathbf{h}_o\cdot\mathbf{h}_o^T\right) + \left\|\mathbf{h}_o\right\|_2^2\cdot\mathbf{I}\right] + m_{s^2,s_e^2} D\left(\mathbf{h}_o\cdot\mathbf{h}_o^T\right), \tag{A.6}$$

where $\mathbf{J} = E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T + \mathbf{h}_o\cdot E\left\{\boldsymbol{\varepsilon}_h^T(n)\right\}$, $\sigma_{s,s_e}^2 = E\left\{s_k s_{e,k}\right\}$, $m_{s^3,s_e} = E\left\{s_k^3 s_{e,k}\right\}$ and $m_{s^2,s_e^2} = E\left\{s_k^2 s_{e,k}^2\right\}$.

Next, we use the above results to simplify three terms $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}$, $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e^T(n)\right\}$ and $E\left\{\mathbf{s}_e(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}_e(n)\right\}$. First, by setting $\mathbf{R}_h(n)$ as an identity matrix, the term $E\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\cdot\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}$ is equal to the Eq. (A.1). With the modification, we have

$$E\left\{\left\{\mathbf{s}(n)\cdot\mathbf{s}^T(n)\mathbf{s}(n)\cdot\mathbf{s}^T(n)\right\}\right\} = \left[\sigma_s^4 M + \left(m_{s^4} - \sigma_s^4\right)\right]\mathbf{I} \tag{A.7}$$

Similarly, by setting $E\left\{\boldsymbol{\varepsilon}_h(n)\right\}\cdot\mathbf{h}_o^T$ and $\mathbf{h}_o\cdot\mathbf{h}_o^T$ as an identity matrix in Eq (A.3) and (A.4), we have

$$E\left\{s(n)\cdot s^T(n)\cdot s(n)\cdot s_e^T(n)\right\} = M\sigma_s^2 m_{s,s_e} + \left(m_{s^3,s_e} - \sigma_s^2 m_{s,s_e}\right), \tag{A.8}$$

$$E\left\{s_e(n)\cdot s^T(n)\cdot s(n)\cdot s_e(n)\right\} = M\sigma_s^2\sigma_{s_e}^2 + \left(m_{s^2,s_e^2} - \sigma_s^2\sigma_{s_e}^2\right), \tag{A.9}$$

respectively.

Bibliography

[1] W. Hsu, F. Chui, and D. A. Hodges, "An acoustic echo canceler," *IEEE Journal Solid State Circuits* vol. 24, no. 6, pp. 1639-1646, Dec. 1989.

[2] S. M. Kuo, Y. C. Huang, and Z. Pan, "Acoustic noise and echo cancellation microphone system for videoconferencing," *IEEE Trans. Consumer Electronics* vol. 41, no. 4, pp. 1150-1158, Nov. 1995.

[3] S. Haykin, *Adaptive filter theory*, 4th ed. Prentice-Hall, 2002.

[4] A. N. Birkett and R. A. Goubran, "Limitations of handsfree acoustic echo cancellers due to nonlinear loudspeaker distortion and enclosure vibration effects," *ICASSP*, pp. 103-106, Oct. 1995.

[5] **V.** J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.,* pp. 10-26, July 1991.

[6] M. Schetzen, "*The Volterra and Wiener Theories of Nonlinear Systems,*" New **York** Wiley, 1980.

[7] A. Guerin, G.. Faucon and R. Le Bouquin-Jeannes, "Nonlinear acoustic echo cancellation based on Volterra filters,"*IEEE Trans. Speech and Audio Processing*, vol. 11, issue 6, pp.672 - 683. Nov. 2003.

[8] J. P. Costa, A. Lagrange, and A. Arliaud, " Acoustic echo cancellation using nonlinear cascade filters," *ICASSP*, pp.v389-v392, 2003.

[9] K. Narendra and P. Gallman, "An iterative method for the identification of nonlinear systems using a Hammerstein model," *IEEE Trans. Automatic Control*, vol. 11, issue 13, pp. 546-550. Jul. 1966.

[10] L.O. Chua and S. M. Kang, "Section-wise piecewise-linear functions: Canonical representation, properties, and applications," *Proceedings of the IEEE,* vol. 65,

issue 6, pp.915-929. Jun. 1977.

[11] J. N. Lin and R. Unbehauen, “Adaptive nonlinear digital filter with canonical piecewise-linear structure,” *IEEE Tran. Circuits and Systems,* vol. 37, issue 3, pp. 347-353. Mar. 1990.

[12] J. Voros, “Iterative algorithm for parameter identification of Hammerstein systems with two-segment nonlinearities,” *IEEE Tran. Automatic Control,* vol. 44, issue 11, pp.2145-2149. Nov. 1999.

[13] V. Vörös, “Modeling and parameter identification of systems with multisegment piecewise-linear characteristics,” *IEEE Trans. Automatic Control,* vol.47, issue1, pp. 184-188. Jan. 2002.

[14] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*. New York: Wiley-Intersci., 1987.

[15] S.C. Douglas, “Adaptive filters employing partial updates,” *IEEE Trans. Circuits Syst.*, vol. CAS-II, pp. 209–216, Mar. 1997.

[16] Godavarti, M. and Hero, A.O., "Partial update LMS algorithms," *IEEE Trans. Signal Processing*, vol. 53, issue 7, pp. 2382-2399, July 2005

[17] R. H. Kwong and E. W. Johnston, “A variable step size LMS algorithm,” *IEEE Trans. Signal Processing,* vol. 40, pp. 1633–1642, July 1992.

[18] 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, pp. 101-108, January 1993.

[19] 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.

[20] Heredia, E.A. and Arce, G.R. "Piecewise Volterra filters bands on the threshold decomposition operator," *IEEE ICASSP*, vol. 3, pp. 1593-1596, May 1996.

[21] G. Y. Jiang and S. F. Hsieh, “Nonlinear acoustic echo cancellation using orthogonal polynomial,” *ICASSP*, vol.5, pp. 273-276. 2006.

[22] M. Godavarti, “Implementation of a G.165 line echo canceller on Texas Instruments’ TMS320C3x and TMS320C54x chips,” in *Proc. ICSPAT Conf.*, pp. 65–69, Sep. 1997.

[23] H. Dai and W. P. Zhu, “An acoustic echo cancellation scheme using raised-cosine function for nonlinear compensation,” *TENCOM*, vol. A, pp.131-134, Nov. 2004.

[24] A. N. Birkett, and R. A. Goubran, “Nonlinear loudspeaker compensation for hands free acoustic echo cancellation,” *Electronics Letters* vol. 32, issue 12, pp. 1063-1064, Jun. 1996.

[25] J. M. Chen, “Step-Size Adjustments of NLMS Algorithm for Acoustic Echo Cancellation,” *NCTU Master Thesis*, June 2005.