On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response

Ligang LIU

A dissertation submitted to
Kochi University of Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Graduate School of Engineering
Kochi University of Technology
Kochi, Japan

September 2009

On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response **Ligang LIU**

On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response

Ligang LIU

A dissertation submitted to
Kochi University of Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Special Course for International Students
(Information Systems Engineering)
Graduate School of Engineering
Kochi University of Technology
Kochi, Japan

September 2009

Abstract

On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response

Ligang LIU

Adaptive filtering algorithms can find applications in many real-world systems. Nowadays, with increase in the demand for higher quality communication, a kind of long adaptive filter is frequently encountered in practical application, such as the network echo cancellation and acoustic echo cancellation. Increase of adaptive filter length from decades to hundreds or thousands causes the conventional adaptive algorithms encounter new challenges. First, the convergence speed of adaptive algorithms is greatly degraded, because their convergence speed is inversely proportional to the adaptive filter length. Second, the computational complexity of adaptive algorithms is increased considerably because it is usually a function of the adaptive filter length. Third, the convergence quality of the adaptive filter is degraded.

Much efforts have been made to find new adaptive algorithms to solve these problems. Recently, a new adaptation paradigm, proportionate adaptation, was developed in a novel perspective. Its main philosophy comes from a fact that most of these long impulse responses are *sparse* in nature, which is dominated by regions where magnitudes are zero or close to zero. Many proportionate adaptive algorithms have been proposed in literature.

In this dissertation, I concentrate on the problems how to improve various proportionate adaptive algorithms, in terms of convergence speed, steady-state misalignment and computational complexity. The main highlights of my research are listed bellow.

- The proportionate adaptation is discussed as a whole framework and various existing proportionate adaptive algorithms are reviewed in detail.
- A measurement of impulse response sparsity is incorporated into the proportionate adaptive algorithms in order to improve their convergence for various impulse responses with different sparsity.
- A derivation for optimal proportionate adaptation step gain is provided in a straightforward and simple way.
- Several approaches are proposed to improve the promising SPNLMS algorithm. A convergence criterion is established and then a refined segment function is proposed to approximate the original mu-law function as exact as possible. Some methods are proposed to reduce the computational complexity of proportionate adaptive algorithms.
- In order to improve convergence quality of proportionate NLMS algorithms, a non-parametric variable step-size approach is proposed. Taking into account negative effect of the disturbance signals, an optimal global step size can be obtained by forcing the *a posteriori* error not to be zero, but to be the disturbance signal. Based on this idea, a variable step-size approach is proposed to cancel the pure *a posteriori* error at each iteration. It has very fast convergence as proportionate adaptive algorithms, as well as very low steady-state misalignment of variable step-size algorithms.
- The non-parametric variable step-size approach is extended into the proportionate affine projection algorithms to improve the estimate accuracy of sparse impulse response for correlated input signals.
- A novel perspective of proportionate adaptation is proposed to determine the proportionate step gain. All existing proportionate adaptive algorithms determine their proportionate step gain according to the shape of the impulse response. Analysis reveals that the proportionate step gain should be computed according to the difference of current estimate of adaptive filter and its optimal value. Based on this result, a novel approach is proposed to determine

proportionate step gain. The proposed algorithms demonstrate faster convergence speed than the counterparts, for both sparse and non-sparse impulse responses.

key words Digital signal processing, adaptive filtering, least-mean-square, proportionate adaptation, sparse impulse response, system identification, echo cancellation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Adaptive algorithms of research in this dissertation are concerned on improvement of the proportionate adaptive algorithms, particularly focusing on identification of sparse impulse response.

1.1 Background of Research

The signal processing technology is widely applied in practical applications to obtain information of interest. Nowadays, adaptive filter is employed extensively in many fields such as communications, radar, sonar, seismology and biomedical engineering [1–4]. An adaptive filter can adjust its transfer function by *itself*. By contrast, a non-adaptive filter is a static filter whose coefficients are invariant during the period of interest. An adaptive algorithm provides a mechanism to change behavior of adaptive filter based on a certain performance criterion. It plays an important role in an adaptive filter.

A lot of efforts have been made to pursuit better adaptive algorithms in terms of:

- faster convergence speed;
- lower misalignment, i.e. higher estimate accuracy;
- lower computation cost;
- and robustness against variation of environment.

There are many factors that influence the performance of adaptive algorithms, such as the structure of adaptive filter, definition of cost function, and optimization method. The adaptive filter can be constructed as finite impulse response (FIR) filter, infinite impulse response (IIR) filter, lattice or non-linear filter. The FIR filter is widely used due to its simplicity. Basically there are two kinds of cost functions: the least-mean-square error that is widely used due to its efficiency and simplicity, and the least-square error. From these two kinds of cost functions, adaptive algorithms can fall into two main categories: stochastic algorithm and deterministic algorithm. This dissertation works on the FIR filter with the least-mean-square error as its cost function.

Various applications of adaptive filtering differ in the manner how the input signal and desired signal are defined. According to the definition, the adaptive filtering applications can be classified into four classes: identification, inverse modeling, prediction and interference canceling [2]. In the identification context, an adaptive filter aims to estimate an unknown impulse response. For example, an echo canceler is designed to identify an unknown echo path impulse response and generate a replica of the echo signal. The echo can be efficiently eliminated by subtracting the echo replica from the near end signal.

1.2 Source of Long Adaptive Filters

With the development of communication technology, such as Voice over IP and hands-free communication system, the demand of higher quality of service and better user experience are increased considerably. A kind of long impulse response has frequently encountered in recent years.

In a network echo cancellation system, the echo signal arises from imperfection of the hybrid, as shown in Fig. 1.1. Traditionally, the impulse response of a hybrid spans at most 12 millisecond (ms) and it typically has approximately 100 coefficients. The impulse response of hybrid 8 defined in ITU-T Recommendation G.168 [5] is illustrated in Fig. 1.2. However, in a

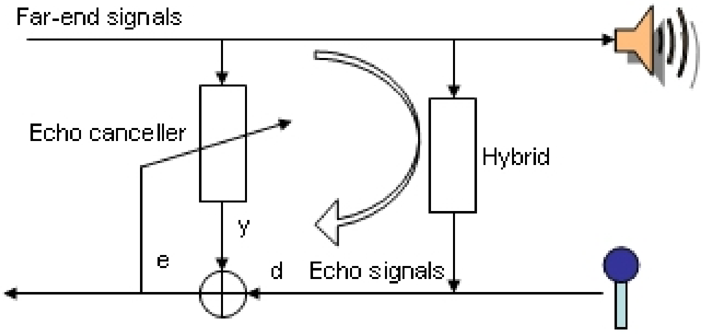

Fig. 1.1 Network echo cancellation.

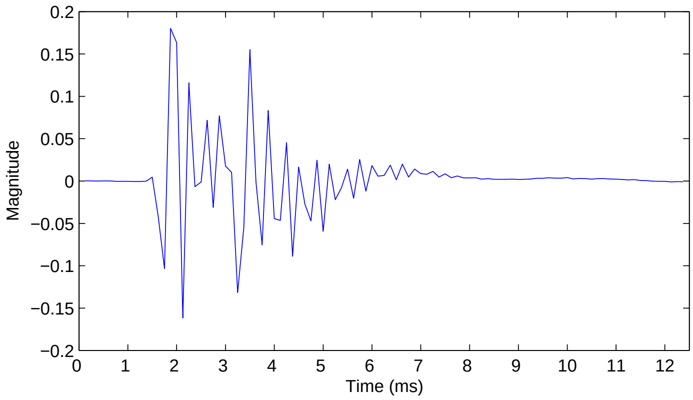

Fig. 1.2 Impulse response of Hybrid 8 defined in ITU-T G.168

voice over IP (VoIP) system, where traditional telephony equipment is connected to the packet switched network, a network echo canceler is forced to cover a large duration in order to model the possible presence of an unknown and potentially large delay. The delay could be caused by

- signal processing, such as encoding and decoding;
- packet switching or network congestion;
- jitter buffer.

Depending on the network condition, the bulk delay can vary significantly in the range between 32 and 128 ms. Two network echo paths including two different delays are illustrated in Fig. 1.3. Consequently, the network echo path is typically of length 64-128 ms [5]. If the sampling rate is 8 kHz, it means an echo canceler has to equip at least 512 tap weights.

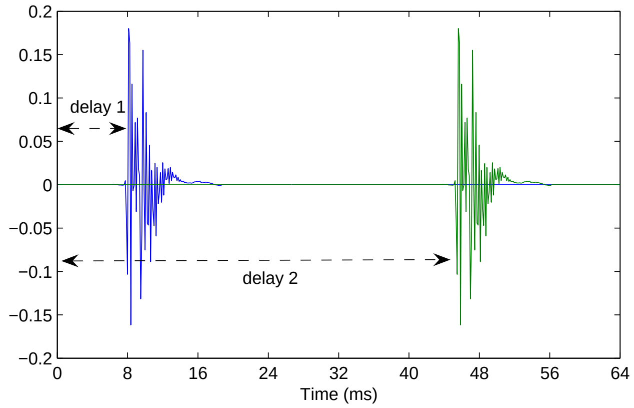

Fig. 1.3 Two network echo paths with different delay time (8k sampling rate).

Other situations include a conference system or a hands-free communication system, where an acoustic echo canceler is required to estimate an acoustic impulse response and then generate a replica of acoustic echo. A block diagram of acoustic echo cancellation is shown in Fig. 1.4. The length of an acoustic impulse response is related to the reverberation time of a enclosed space, while the reverberation time is proportional to the volume of the space and inversely proportional to the absorption area [6]. In order to efficiently cancel the echo signal in a acoustic echo cancellation system, it is required to handle as long as 64-128 ms reverberation time. Another aspect of length of an acoustic echo canceler is the data rate. For an 8k sampling rate system, the echo canceler must have as many as 512 tap weights. With increase of data rate from 8k to 44.1k, the length of acoustic echo canceler increases greatly, even more than 2000 tap weights. Figure 1.5 illustrates an acoustic impulse response at 44.1k sampling rate. The coefficients after 5000 still have noticeable magnitude.

1.3 New Challenges

Generally speaking, with increase of adaptive filter length from decades to hundreds or thousands, the traditional adaptive algorithms encounter new challenges:

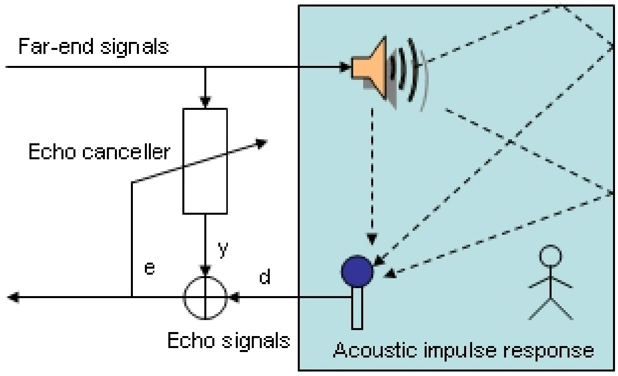

Fig. 1.4 Block diagram of acoustic echo cancellation.

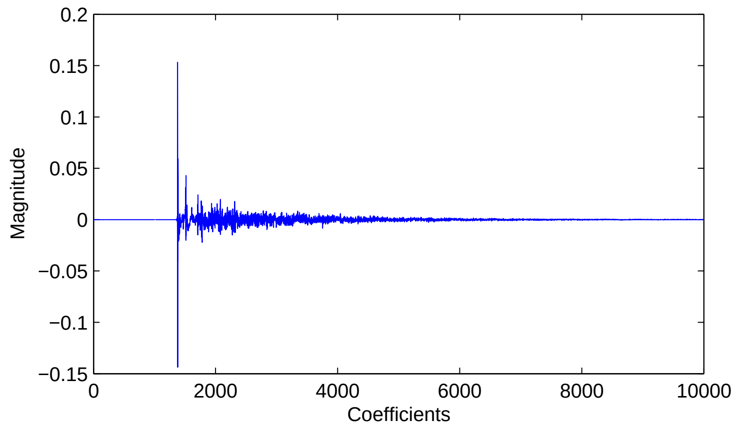

Fig. 1.5 A long acoustic impulse response (44.1k sampling rate).

- Firstly, the convergence speed of adaptive algorithms are greatly degraded, because it is inversely proportional to the adaptive filter length [2, 7, 8]. The convergence speed per sample (in dB) is given by [8]

$$R = \frac{10}{\ln 10} \cdot \frac{1}{N}\alpha(2-\alpha), \tag{1.1}$$

where α is a step-size parameter and N is the length of adaptive filter.

- Secondly, the computational complexity of adaptive algorithms are increased considerably because it is usually a function of the adaptive filter length. For example, the lest mean-square (LMS) algorithm require $2N$ multiplications per sample, and the recursive leastsquares (RLS) algorithm require $O(N^2)$ multiplications per sample.

- Thirdly, the accuracy of the adaptive filter is degraded in a noise environment.

Much efforts have been made to find new adaptive algorithms to solve these problems. Recently, a new adaptation paradigm, proportionate adaptation, was developed to speed up the convergence of adaptive algorithms exploiting the structure information of this kind of long impulse response. Its main philosophy comes from a fact that most of these long impulse responses are *sparse* in nature.

1.4 Sparse Impulse Response

A sparse impulse response is an impulse response that only a small percentage of its components with a significant magnitude while the rest are zero or small. Another definition could be the following: an impulse response is sparse if a large fraction of its energy is concentrated in a small fraction of its duration [9]. For example, in a network impulse response, only about 4-12 ms in a 64 or 128 ms time duration are "active" and the others are zeros. Figure 1.3 illustrates two typical network echo paths. Other sparse impulse responses include marine seismic signals and geophysical seismic impulse response.

Sparsity of these long impulse response arises from the fact that, although an adaptive filter with a large number of coefficients has to be used to synthesize the echo path, only a small fraction of the coefficients is significantly different from zero (active coefficients) and modeling the actual dispersion part. All other coefficients are just zero or unnoticeably small values (inactive coefficients) and needed for representation of the bulk delay. An echo path with typical length of 64-128 ms exhibits an *active* region in the range of 8 – 12 ms duration and consequently, the impulse response is dominated by regions where magnitudes are close to zero, making the network echo path a sparse impulse response.

In a conference system or a hands-free communication system, an acoustic echo path usually is not as sparse as a network echo path impulse response. However, with the increase of user demand, it is necessary to equip an echo canceler of as many as 512 - 2048 tap weights in order to deal with possible propagation delay of sound. It is well known that the reverberation time of an acoustic impulse response is proportional to the volume of the enclosed space and inversely proportional to the absorption area. Hence the reverberation time is varying with various acoustic environments.

1.5 Proportionate Adaptation

The traditional adaptive algorithms assign one same step size to all coefficients regardless of the structure of the target impulse response. For these algorithms, the large coefficients need more iterations to converge to the steady state than the small coefficients. In order to accelerate convergence of the large coefficients, it seems that a large step size parameter should be assigned for them. This results in proportionate adaptation.

Proportionate adaptation is a recently developed adaptation paradigm, which has attracted much attention [9–11]. It speeds up the convergence for sparse impulse response by exploiting the sparse structure of long impulse responses. The idea behind proportionate adaptive algorithms is to update each coefficient individually by assigning a step size proportional to its estimated magnitude. The PNLMS algorithm was firstly proposed in [8] introducing a diagonal step-size control matrix $\mathbf{G}(t)$ in order to associate a step-size parameter for each coefficient. It greatly speeds up the initial convergence of adaptive filter. However, its convergence begins to slow dramatically thereafter. In addition, its convergence speed will degrade for non-sparse impulse response. Many modifications have been proposed to improve the PNLMS algorithm, focusing on a certain specific aspect, such as convergence speed, estimate accuracy, or computational complexity.

1.6 Contributions

In this dissertation, I concentrate on the problems how to improve the proportionate adaptive algorithms, in terms of convergence speed, steady-state misalignment and computational complexity reduction at the same time. The main contributions of this dissertation are listed as following:

- The proportionate adaptation is discussed as a whole framework and various existing proportionate adaptive algorithms are reviewed in detail.
- A measurement of impulse response sparsity is incorporated into the proportionate adaptive algorithms in order to improve their convergence for various impulse responses with different sparsity. The related results have been published in [12, 13].
- A derivation for optimal proportionate adaptation step gain is provided in a straightforward and simple way.
- Several approaches are proposed to improve the promising segment proportionate NLMS (SPNLMS) algorithm. A convergence criterion is established through analyzing convergence characteristics of proportionate adaptive algorithms. Some methods are proposed to reduce the computational complexity of proportionate adaptive algorithms. The related results have been published in [14] and submitted in [15].
- In order to improve convergence quality of the proportionate NLMS algorithm, a non-parametric variable step-size approach is proposed. It has very fast convergence as proportionate adaptive algorithms, as well as very low steady-state misalignment of variable step-size algorithms. The related results have been published in [16].
- The non-parametric variable step-size approach is extended into the proportionate affine projection algorithms to improve the estimate accuracy for correlated input signals. The related results have been published in [17, 18].
- A novel perspective of proportionate adaptation is proposed to determine the proportionate

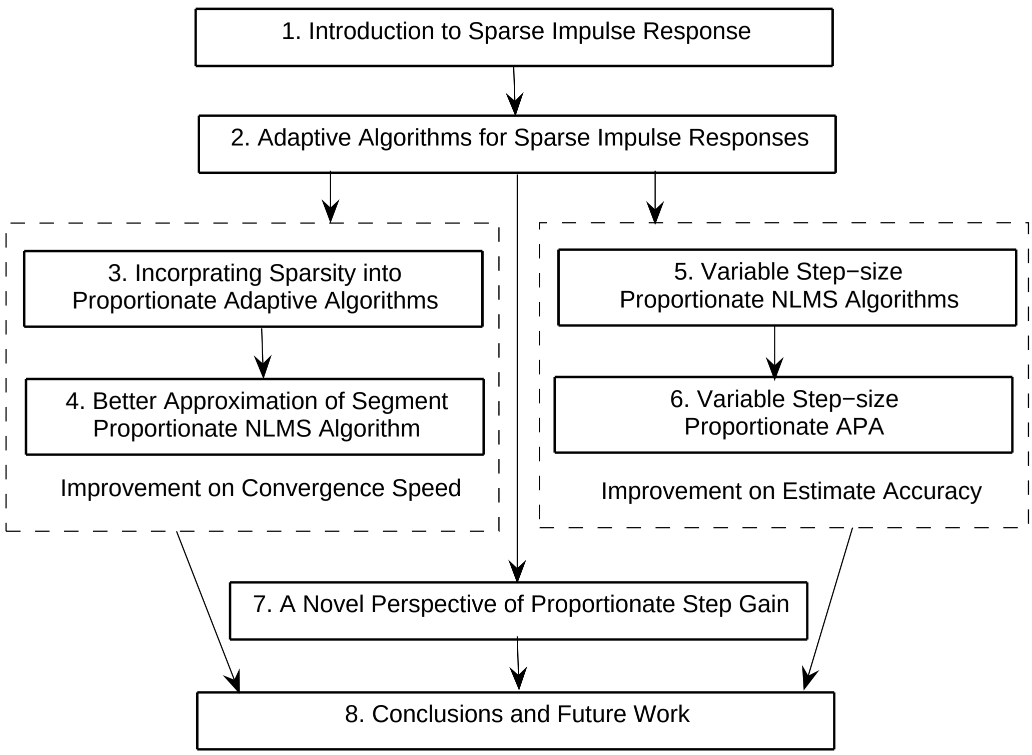

Fig. 1.6 Outline of the dissertation.

step gain. All existing proportionate adaptive algorithms determine their proportionate step gain according to the shape of the impulse response. The analysis reveals that the proportionate step gain should be computed according to the difference of current estimate of adaptive filter and its optimal value. Based on this result, a novel approach is proposed to determine proportionate step gain. The related results have been submitted in [19].

1.7 Outline

The outline of the dissertation is depicted in Fig. 1.6.

Chapter 2 briefly reviews traditional adaptive algorithms in a wide range, and then state-of-art of proportionate adaptive algorithms. The philosophy of the proportionate adaptation paradigm is introduced and various proportionate adaptive algorithms are reviewed in detail.

The advantages and disadvantages of various proportionate adaptive algorithms are discussed. Although proportionate adaptive algorithms are initially developed in order to improve the convergence of long sparse impulse responses, some techniques are introduced so that the computational complexity and estimate accuracy are improved.

In **chapter 3** and **chapter 4** focus on improvement of convergence speed of proportionate adaptive algorithms. **Chapter 3** proposes to incorporate sparsity measurement into the proportionate NLMS algorithms in order to achieve fast convergence speed not only for sparse and impulse response, but also for non-sparse one [12, 13]. In **chapter 4**, some approaches are proposed to improve a promising segment proportionate NLMS (SPNLMS) algorithm [14, 15]. A straightforward and simple derivation of the optimal proportionate step-size control matrix is provided at first in proportionate adaptation framework. Several methods are proposed to improve its convergence speed. Two approaches are proposed to reduce computational complexity.

Chapter 5 and **6** concentrate on how to improve the steady-state misalignment of proportionate adaptive algorithms using variable step-size technique. In **chapter 5** an non-parametric variable step size approach is derived to improve the estimate accuracy of proportionate NLMS algorithms [16]. **Chapter 6** extends this variable step size approach to proportionate APA to improve the performance for colored input signals [17, 18].

In **chapter 7**, a new perspective is proposed to improve performance of proportionate adaptive algorithms [19]. The new approach computes its proportionate step gain according to the difference of adaptive coefficients and its optimal value, in stead of the shape of adaptive filter. The proposed algorithms demonstrate better convergence performance than the existing counterparts. It behaves faster convergence after a fast initial period, and also for non-sparse impulse response.

Chapter 8 concludes this dissertation and provides suggestions for further research.

Chapter 2

Adaptive Algorithms for Sparse Impulse Response

2.1 Introduction

An adaptive filter is a filter that self-adjusts its tap weights based on an optimizing principle. Contrary, a non-adaptive filter has invariant filter coefficients. For some applications, because some parameters of the required properties are not known in advance , an adaptive filter is required to satisfy the requirement of application. For instance, when the properties of input signal or disturbance signal are unknown, it usually employs an adaptive filter to refine the values of the filter coefficients using feedback. This chapter provides a brief introduction of adaptive algorithms and recent advances in adaptive algorithms for long sparse impulse response.

A block diagram of adaptive algorithm is illustrated in Fig. 2.1. The adaptation process

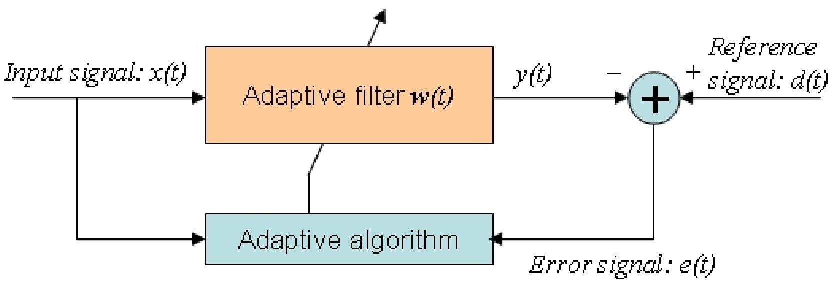

Fig. 2.1 Block diagram of adaptive filter

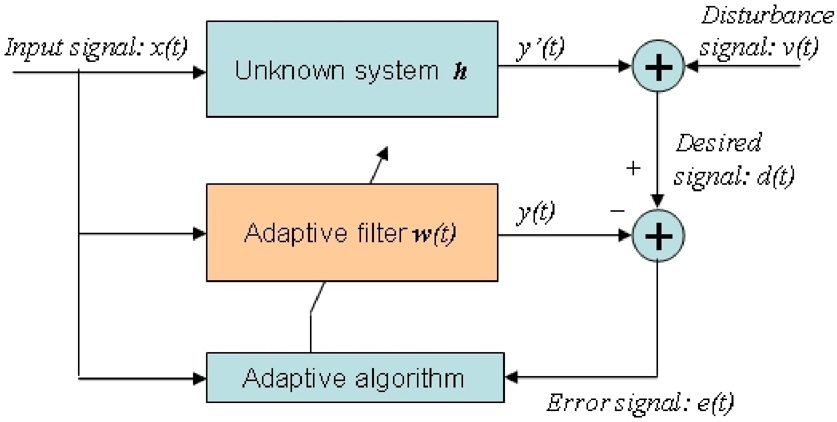

Fig. 2.2 Block diagram of system identification.

generally involves two primary elements: a cost function and adaptation method. The cost function is a criterion for optimum performance of the filter and the adaptation method determines how to modify the filter coefficients in order to minimize the cost function.

Consider a system to identify an unknown impulse response **h**, as shown in Fig. 2.2. Bold lowercase letters indicate vectors and bold uppercase letters denote matrices. All vectors are column vectors, $(.)^T$ indicates transpose, and t is the time index. $E\{\cdot\}$ denotes the mathematical expectation operator. The input signals $x(t)$ pass through **h** and the output is called desired signal $d(t)$. Define the input vector as

$$\mathbf{x}(t) = [x(t)\ x(t-1)\ \cdots\ x(t-N+1)]^T, \tag{2.1}$$

then the output of adaptive filter is

$$y(t) = \mathbf{w}^T(t)\mathbf{x}(t). \tag{2.2}$$

Usually, the desired signal, $d(t)$, is contaminated by background noise $v(t)$, i.e.,

$$d(t) = \mathbf{h}^T\mathbf{x}(t) + v(t). \tag{2.3}$$

An adaptive filter **w** is designed to estimate **h** iteratively. It is presumed that **w** and **h** have same number of coefficients, N, in this dissertation.

Although many cost functions have been proposed for adaptive filtering, the most frequently used cost is the so-called least-mean-squared error, where the error signal $e(t)$ and the cost function $\mathcal{J}(t)$ are defined as

$$e(t) = d(t) - y(t) = d(t) - \mathbf{w}^T(t)\mathbf{x}(t), \tag{2.4}$$

$$\mathcal{J}(t) = E\left\{e^2(t)\right\}. \tag{2.5}$$

An adaptive filter $\mathbf{w}$ is designed to minimize $\mathcal{J}(t)$. The optimal tap weights of an FIR filter can be obtained by solving the Wiener-Hopf equation if the statistics of the underlying signals are available. The solution of Wiener-Hopf equation is given as [2]:

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}, \tag{2.6}$$

where $\mathbf{R}$ is the auto-correlation matrix of input vector, $\mathbf{R} = E\{\mathbf{x}(t)\mathbf{x}^T(t)\}$, and $\mathbf{p}$ is the cross-correlation vector of desired signal and input vector, $\mathbf{p} = E\{d(t)\mathbf{x}(t)\}$.

2.2 Conventional Adaptive Filtering Algorithms

2.2.1 Steepest Descent Method

An alternative method to find $\mathbf{w}_o$ is to use an iterative search algorithm that starts at some arbitrary initial point in the weight vector space and progressively moves towards the optimal point. There are many iterative search algorithms derived to minimize the underlying cost function with the true statistics replaced by their estimates obtained in a certain manner. Gradient-based iterative methods are commonly exploited as steepest descent method, i.e., to adjust tap weights iteratively and move along the error surface towards the optimal value:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\nabla_{\mathbf{w}}\mathcal{J}(t), \tag{2.7}$$

where α is a step size parameter and $\nabla_{\mathbf{w}}\mathcal{J}(t)$ denotes gradient of $\mathcal{J}(\mathbf{w}(t))$ respect to adaptive weight vector $\mathbf{w}(t)$,

$$\nabla_{\mathbf{w}}\mathcal{J}(t) = \frac{\partial\mathcal{J}(t)}{\partial\mathbf{w}(t)}. \tag{2.8}$$

It is a vector pointing in the direction of the change in filter coefficients that will cause the greatest increase in the error signal. Solving $\nabla_{\mathbf{w}}\mathcal{J}(\mathbf{w}(t))$ yields

$$\nabla_{\mathbf{w}}\mathcal{J}(t) = \mathbf{p} - \mathbf{R}\mathbf{w}(t). \tag{2.9}$$

Replacing it into (2.7), the steepest descent algorithm is given by

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\left[\mathbf{p} - \mathbf{R}\mathbf{w}(t)\right], \tag{2.10}$$

$$= \mathbf{w}(t) + \alpha\mathbf{R}\left[\mathbf{h} - \mathbf{w}(t)\right]. \tag{2.11}$$

2.2.2 LMS Algorithm

The steepest descent algorithm is not implementable in practical application, because of lack of statistical information of $\mathbf{R}$ and $\mathbf{p}$ in priori. In addition, the unknown system $\mathbf{h}$ is subject to time-varying. Not only the statistical property of input signals is subject to change, the unknown system $\mathbf{h}$ is also subject to be time-varying.

The LMS algorithm is a stochastic implementation of the steepest descent algorithm. It is obtained by substituting the instantaneous estimates into the steepest descent algorithm, i.e., replacing $\mathbf{R}$ and $\mathbf{p}$ in (2.10) by their instantaneous estimates. $\mathbf{R} \approx \mathbf{x}(t)\mathbf{x}^T(t)$ and $\mathbf{p} \approx d(t)\mathbf{x}(t)$. Its coefficient updating equation is given by

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{x}(t)e(t). \tag{2.12}$$

Due to rough estimates of $\mathbf{R}$ and $\mathbf{p}$, adaptation of the LMS algorithm is quite random. The advantages of the LMS algorithm include its simplicity in implementation, stable and robust performance against different signal conditions. Its main disadvantage is slow convergence for correlated input singnals due to eigenvalue spread.

2.2.3 Normalized LMS (NLMS) Algorithm

The convergence of LMS is affected by the magnitude of the input signals. In order to make its convergence behavior independent of the input energy, the NLMS algorithm was proposed that the filter vector update is normalized by the input energy. The algorithm is given by

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \frac{\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{x}(t) + \delta}, \tag{2.13}$$

where δ is a small constant called regularization parameter in order to avoid dividing by zero when the input signal is zero in a long period.

The NLMS algorithm can be also derived from many methods [2, 7]. It can be viewed as a variable step-size LMS algorithm where an optimal step size parameter is achieved by solving a constrained optimization problem using least perturbation principle. The least perturbation principle says, minimizing the coefficient vector update, subject to a constraint:

$$\begin{aligned} \mathbf{w}(t+1) &= \min_{\mathbf{w}(t+1)} \|\mathbf{w}(t+1) - \mathbf{w}(t)\|_2^2, \\ \text{subject to} \quad & \mathbf{x}^T(t)\mathbf{w}(t+1) = d(t). \end{aligned} \tag{2.14}$$

This problem can be solved by using the method of Lagrange multipliers [2] and the resulting algorithm is NLMS. The NLMS algorithm applied a minimum norm update to the solution vector such that the *a posteriori* error of the most recently added equation is exactly zero.

The NLMS algorithm is widely used in practical application because its simplicity and stability. A step-size parameter in the range of $0 < \alpha < 2$ can assure the NLMS algorithm convergence. The main disadvantage of NLMS is that it suffers from slow convergence for correlated input signals.

2.2.4 Affine Projection Algorithm (APA)

For correlated input signals, especially the speech, the convergence speed of the LMS-type algorithms will depend on the eigenvalues of the input signal's auto-correlation matrix. The APA improves convergence of the adaptive filter by pre-whitening the input signal. APA is an intermediate algorithm between the NLMS algorithm and the RLS algorithm, since it has both a performance and a complexity in between those of NLMS and RLS.

Define the input matrix $\mathbf{X}(t)$ as the P successive past input vectors and the desired vector $\mathbf{d}(t)$ as the P successive past value of $d(t)$, where P is the projection order, as [2]

$$\mathbf{X}(t) = [\mathbf{x}(t)\ \mathbf{x}(t-1)\ \cdots\ \mathbf{x}(t-P+1)], \tag{2.15}$$

$$\mathbf{d}(t) = [d(t)\ d(t-1)\ \cdots\ d(t-P+1)]^T. \tag{2.16}$$

APA solves the P most recent equations exactly, based on a minimum norm weight vector update:

$$\mathbf{w}(t+1) = \min_{\mathbf{w}(t+1)} \|\mathbf{w}(t+1) - \mathbf{w}(t)\|_2^2, \tag{2.17}$$

$$\text{subject to} \quad \mathbf{X}^T(t)\mathbf{w}(t+1) = \mathbf{d}(t). \tag{2.18}$$

The cost function is minimized by taking the partial derivatives for all entries of the coefficient vector and setting the results to zero. The resulting APA can be briefly summarized as [2]:

$$\mathbf{e}(t) = \mathbf{d}(t) - \mathbf{X}^T(t)\mathbf{w}(t), \tag{2.19}$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{X}(t) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(t), \tag{2.20}$$

where α is an global constant step-size parameter, and δ is the regularization parameter and $\mathbf{I}$ is a $P \times P$ identity matrix.

The NLMS algorithm is a special case of the APA with $P = 1$. The APA exploits more information of input signal. Consequently, it obtains a more accurate estimate of gradient. It achieves a faster convergence speed for the correlated input signals than NLMS only with a modest increase of computational complexity.

2.2.5 Recursive Least-Square (RLS) Algorithm

The RLS algorithm is not a stochastic algorithm, but a deterministic one. It is the solution of least-square problem, whose cost function is deterministic. An exponentially weighted cost function is defined as:

$$\mathcal{J}(t) = \sum_{i=0}^{t} \lambda^i |e(t-i)|^2, \tag{2.21}$$

where λ is an exponential weighting factor which effectively limits the number of input samples based on which the cost function is minimized. Generally, λ is close but less than 1. The optimal value for the tap-weight vector is defined by the following normal equations

$$\begin{aligned}
\mathbf{\Phi}(t) &= \sum_{i=1}^{t} \lambda^{t-i}\mathbf{x}(i)\mathbf{x}^T(i) = \lambda\mathbf{\Phi}(t-1) + \mathbf{x}(t)\mathbf{x}^T(t), && (2.22)\\
\mathbf{z}(t) &= \sum_{i=1}^{t} \lambda^{t-i} d(i)\mathbf{x}(i) = \lambda\mathbf{z}(t-1) + d(t)\mathbf{x}(t), && (2.23)\\
\mathbf{\Phi}(t)\mathbf{w}(t) &= \mathbf{z}(t). && (2.24)
\end{aligned}$$

Solving $\mathbf{w}(t)$ yields

$$\mathbf{w}(t) = \mathbf{\Phi}^{-1}(t)\mathbf{z}(t). \tag{2.25}$$

Then, using the matrix inversion lemma to the recursive model of correlation matrix $\mathbf{\Phi}(t)$ to make it possible to invert it recursively [Refer to Appendix A.1]. For exponentially weighted RLS algorithm, its coefficient vector updating equations are depicted by:

$$\begin{aligned}
\mathbf{S}(n) &= \frac{1}{\lambda}\left[\mathbf{S}(n-1) + \frac{\Psi(n)\Psi^T(n)}{\lambda + \Psi^T(n)\Psi(n)}\right], && (2.26)\\
\mathbf{w}(n+1) &= \mathbf{w}(n) + \mathbf{S}(n)e(n), && (2.27)\\
\mathbf{\Psi}(n) &= \mathbf{S}(n-1)\mathbf{X}(n). && (2.28)
\end{aligned}$$

The RLS algorithm utilizes all the information contained in the input signals from the start of the adaptation up to the present. It converges at a much higher speed than the LMS algorithm and the APA.

2.2.6 Comparison of Computational Complexity

The complexity of an adaptive algorithm is determined by the number of multiplications (with divisions counted as multiplications) per iteration.

The LMS algorithm require $2N$ multiplications per iteration: N is for calculation of $\mathbf{x}^T(t)\mathbf{w}(t)$, and N is for coefficient vector updating. The computational complexity of NLMS is approximately same to LMS. For shift-structure input data, calculation of normalization item $\mathbf{x}^T(t)\mathbf{x}(t)$ could be computed in a recursive way:

$$\sigma_x^2(t) = \sigma_x^2(t-1) + x^2(t) - x^2(t-N+1), \tag{2.29}$$

which only needs 3 multiplications. Or, it can be calculated in a way of power estimate:

$$\sigma_x^2(t) = \lambda\sigma_x^2(t-1) + (1-\lambda)x^2(t), \tag{2.30}$$

which needs 3 multiplications too.

The APA requires approximate $(P^2 + 2P)N + O(P^3)$ multiplications per iteration. There exists a fast implementation whose computational complexity is about $2N + 20P$, see [20–23] for detail.

The superior performance of the RLS algorithm is attained at the expense of a large increase in computational complexity. The RLS algorithm requires a total of $N^2 + 5N + 2$ multiplications, which increases as the square of N. For example, when $N = 512$, the RLS algorithm requires 263170 multiplications, whereas the NLMS algorithm requires only 1027 multiplications.

Table 2.1 compares the computational complexity of the related conventional adaptive filtering algorithms.

Table 2.1 Comparison of Computational Complexity of Conventional Algorithms

	$\times/\div$	$+/-$
LMS	$2N+1$	$2N$
NLMS	$2N+4$	$2N+2$
APA	$(P^2+2P)N+P^3+P$	$(P^2+2P)N+P^3+P^2$
FAPA	$2N+20P$	
RLS	N^2+5N+2	N^2+3N

2.3 Proportionate Adaptive Algorithms for Sparse Impulse Response

The conventional adaptive filtering algorithms suffer slow convergence for long adaptive filter with hundreds or thousands tap weights. Fortunately, long impulse responses are sparse in nature, which is dominated by regions where magnitudes are zero or close to zero. There are several methods in literature to improve convergence of the adaptive algorithms for long sparse impulse responses. One method is to estimate the bulk delay [24–26], and one is to detect the active coefficients [27, 28].

The proportionate adaptation is a recently proposed method. It does not explicitly locate the active coefficients a priori, but try to assign every coefficient different step-size parameter. The conventional adaptive algorithms assign one same step size to all coefficients regardless of the structure information. For these algorithms, the large coefficients need more iterations to converge than the small ones. In order to accelerate convergence of the large coefficients, a large step size should be assigned to them. This results in the proportionate adaptation by exploiting the sparse structure [9–11]. Proportionate adaptive algorithms update each coefficient individually by assigning a step size proportional to its estimated magnitude.

In this section, various proportionate adaptive algorithms are reviewed and their main advantages and disadvantages are discussed.

2.3.1 Framework of Proportionate NLMS Algorithm

Applying a diagonal step-size control matrix,

$$\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}, \tag{2.31}$$

into the LMS coefficient vector updating equation in order to associate one step size with each coefficient, it gives a proportionate adaptation (called proportionate LMS):

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{G}(t)\mathbf{x}(t)e(t). \tag{2.32}$$

A normalized version can be obtained by normalizing the updating equation with the Euclidean norm of the input vector in order to eliminate the influence of input signal power. The resulting algorithm has a form described in [8] as:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \frac{\alpha}{\|\mathbf{x}(t)\|_2^2 + \delta}\mathbf{G}(t)\mathbf{x}(t)e(t), \tag{2.33}$$

where δ is a regularization parameter. However, this dominator may cause the algorithm unstable for impulsive excitation signals [29].

The proportionate NLMS algorithm has another form in literature, where the dominator is the weighted Euclidean norm of the input vector, $[\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)]$, in stead of $[\mathbf{x}^T(t)\mathbf{x}(t)]$. It can be described by [30–36, 64]:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \delta}, \tag{2.34}$$

This form is a nature extension of (2.32). Table 2.2 summaries the proportionate NLMS algorithm.

In [37], a different derivation of PNLMS algorithm is provided. The PNLMS algorithm can also be derived by finding an optimal variable step-size parameter in the minimum mean squared error sense as in [35]. The convergence property of proportionate adaptive algorithms

Table 2.2 Proportionate NLMS Algorithms

Initialization:
$\mathbf{w}(1) = \mathbf{0}$
Global step-size parameter α
Regularization parameter δ
For all t:
$e(t) = d(t) - \mathbf{x}^T(t)\mathbf{w}(t)$
Formation of $g_n(t), \quad n = 0, 1, \cdots, N-1$
$\mathbf{G}(t) = \text{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}$
$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \dfrac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \delta}$

has been extensively studied in [8, 11, 38].

Various proportionate NLMS algorithms have been proposed in different way. They mainly differ in the definition of $\mathbf{G}(t)$.

2.3.2 PNLMS Algorithm

The original definition of the diagonal element of matrix $\mathbf{G}(t)$ in the PNLMS algorithm is described as [8]:

$$\gamma_n(t) = \max\{|w_n(t)|, \rho \max\{\delta_\rho, |w_0(t)|, \cdots, |w_{N-1}(t)|\}\}, \tag{2.35}$$

$$g_n(t) = \frac{\gamma_n(t)}{\frac{1}{N}\sum_{i=0}^{N-1} |\gamma_i(t)|}. \tag{2.36}$$

Here, δ_ρ is used to prevent $\mathbf{w}(t)$ from stalling during initialization stage. Its typical value is 0.01. The parameter ρ is used to prevent coefficients from stalling when they are much smaller than the largest one. Its typically value is from $1/N$ to $5/N$. It can be seen that if the current magnitude of a coefficient is large, a large step size parameter will be assigned, for a small coefficient the proportionate step size is small. It emphasizes the large coefficients to speed up their convergence, so it demonstrates very fast initial convergence for sparse impulse response. However, its convergence will slow down dramatically after its fast initial convergence, and its convergence speed degrades greatly if the target impulse response is not sparse enough.

2.3.3 PNLMS++ Algorithm and CPNLMS Algorithm

The PNLMS++ algorithm is proposed in [29] to improve the convergence of PNLMS after its fast initial period. It alternates the PNLMS algorithm and the NLMS algorithm during the adaptation. For odd-numbered time index t, the matrix $\mathbf{G}(t)$ is calculated as (2.35) and (2.36). For even-numbered time index t, $\mathbf{G}(t) = \mathbf{I}$. It converges at least as fast as the NLMS algorithm. However, it could cause degradation of the convergence in the initial period.

The composite PNLMS (CPNLMS) was proposed in [31] to switch the PNLMS adaptation to the ordinary NLMS when slow convergence is detected. The learning curve of PNLMS is the sum of each tap weight, which is so-called composite learning curve. At the beginning of the adaptation, coefficients with greater estimated magnitude control the composite learning curve. Because they have a greater weight in their learning curves, the adaptive algorithm has a fast adaptation in the initial period. As the sum of the large taps learning curves reaches the small taps, the adaptation will slows down because small taps have a smaller value, resulting in a slow composite learning curve. Thus by comparing the error signal to a proper ratio of the out error, the beginning of slow region can be detected and switch to the NLMS algorithm. The CPNLMS algorithm improves convergence of PNLMS after fast initial period. Its main problem is the difficulty to decide a reasonable error threshold when the slow convergence begins just according to the error signal.

2.3.4 Improved PNLMS (IPNLMS) Algorithm

The proportionate adaptive algorithms were originally designed for sparse impulse response. For a non-sparse impulse response, the convergence speed of the PNLMS algorithm degrades greatly, even worse than the NLMS algorithm. The improved PNLMS (IPNLMS) algorithm is proposed in [30] to avoid degradation in the case when the underlying impulse response is non-sparse.

The reason why the PNLMS algorithm converges slower than the NLMS algorithm for dispersive impulse response is because of the brutal choice, the maximum operator. Instead of this brutal choice, the average of the current coefficients is adopted to partly remove the negative effect of inaccurate estimate of $\mathbf{w}(t)$. Hence the IPNLMS algorithm converges as fast as the PNLMS algorithm for sparse impulse response and its performance is not much worse than the NLMS algorithm for dispersive one. The diagonal element of $\mathbf{G}(t)$ in the IPNLMS algorithm can be described as

$$g_n(t) = \frac{1-\beta}{2N} + \frac{(1+\beta)|w_n(t)|}{2\sum_{i=0}^{N-1}|w_i(t)| + \epsilon}. \quad (-1 \leq \beta < 1) \tag{2.37}$$

Here ϵ is a very small positive number to avoid dividing by zero. The adjustable parameter, $\beta \in [-1, 1)$, can change the behavior of IPNLMS between NLMS and PNLMS. It can be seen that IPNLMS is equivalent to NLMS when $\beta = -1$ and for β close to 1 it behaves like PNLMS.

2.3.5 Mu-law Proportionate NLMS Algorithm

The MPNLMS algorithm was proposed to achieve optimal proportionate step size control matrix $\mathbf{G}(t)$ in the proportionate adaptation framework [33, 34].

It seems that PNLMS assigns too much adaptation step gain to the large coefficients but too little to the small ones. This is the reason why convergence speed degrades greatly after the fast initial period. Analysis shows that the coefficient of adaptive algorithm exponentially converges to its optimal value. It is reasonable to know that optimal convergence could be achieved when all coefficients reach a small vicinity of their optimal value after same iterations. To achieve this objective, a non-linear relationship is required. The MPNLMS algorithm therefore proposes to exploit the logarithm of the coefficient magnitude as its proportionate step size, instead of directly using the absolute value. Consequently, both the large and small coefficients converge at the same fast speed, so the overall convergence speed of the adaptive algorithm is greatly accelerated. For the MPNLMS algorithm, it has:

$$F(w_n(t)) = \ln(1 + \mu|w_n(t)|), \tag{2.38}$$

$$\gamma_n(t) = \max\{F(w_n(t)), \rho \max\{\delta_\rho, F(w_n(t)), \cdots, F(w_n(t))\}\}, \tag{2.39}$$

$$g_n(t) = \frac{\gamma_n(t)}{\frac{1}{N}\sum_{i=0}^{N-1}|\gamma_i(t)|}. \tag{2.40}$$

Here μ is an objective convergence criterion, typically $\mu = 1000$. Much of the simulation results, see [39], have proved that the MPNLMS algorithm is one of the fastest proportionate adaptive algorithms.

The main advantage of the MPNLMS algorithm is its very fast convergence speed for sparse impulse response. It is also tested that its fast convergence speed will not degrades much for non-sparse impulse response. The main disadvantage is its computation load is too heavy to be implementable in a real-time application because of the presence of N logarithmic operations in every iteration.

2.3.6 Segment Proportionate NLMS Algorithm

In order to reduce the expensive computational complexity of the MPNLMS algorithm, a segment line is proposed to approximate the mu-law function in (2.38). The whole function range can be divided into two sections: one for very small values and one for the other value. Therefore, two straight lines are used to replace the mu-law function. This results in a computation efficient algorithm, SPNLMS [34], where

$$F(w_n(t)) = \begin{cases} 400|w_n(t)|, & |w_n(t)| < 0.005 \\ 2, & \text{otherwise.} \end{cases} \tag{2.41}$$

The sharpness of the initial slope is critical because it emphasizes the small coefficients.

2.3.7 Proportionate Affine Projection Algorithm (PAPA)

For correlated input signals, especially speech, convergence of the PNLMS algorithm will depend on the eigenvalue spread of the input signal's autocorrelation matrix $\mathbf{R}$. The proportionate affine projection algorithm (PAPA) is a natural extension of proportionate NLMS algorithms. It is expected to present faster convergence for highly correlated input signals at cost of a modest increase in computational complexity.

The coefficient updating equation of PAPA can be described by:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(t). \tag{2.42}$$

When $P = 1$, the PAPA degenerates into the PNLMS algorithm, and when all of the elements of $\mathbf{G}(t)$ are identical, i.e., $g_0(t) = \cdots = g_{N-1}(t) = 1$, the PAPA reduces to the standard APA.

Another way to improve convergence for correlated input signals is transform domain proportionate adaptive algorithms. In [40] and [41], the authors combine the ideas of the PNLMS algorithm and of the transform domain adaptive algorithm in order to improve the convergence speed for sparse impulse responses and to accelerate the convergence for colored input signals. The proposed method employs a wavelet transform to decorrelate the input signal and sparse adaptive filters.

A novel adaptive filtering scheme, the Krylov proportionate NLMS algorithm [42], was recently proposed to exploit the benefits of the proportionate NLMS algorithm, but it is also effective for non-sparse impulse response. It has fast convergence with only linear complexity, and no use of any a priori information.

2.4 Reduction of Computational Complexity

LMS-type FIR adaptive algorithms require at least $2N$ multiplication operations per iteration. Its computational complexity increases with increasing the filter length and therefore such a filter can become prohibitive for certain applications, especially for real-time implementation and high data-rate applications. Updating the entire coefficient vector is costly in terms of power, memory, and computation and is sometimes impractical for resource-limited devices. In this section, some popular computation reduction approaches are reviewed.

2.4.1 Partial-Updating Adaptive Algorithms

The partial-updating adaptive algorithms only update a portion of the coefficients per iteration [43–46]. By this way, the overall computational complexity of adaptive algorithms is less than that of a full-updating adaptive algorithm.

Two types of partial update algorithms are prevalent in the literature and have been described in [43]. The periodic NLMS algorithm updates all the filter coefficients every P iterations instead of every iteration:

$$\mathbf{w}((k+1)P) = \mathbf{w}(kP) + \alpha \frac{\mathbf{x}(kP)e(kP)}{\mathbf{x}^T(kP)\mathbf{x}(kP) + \delta}. \tag{2.43}$$

The coefficient updates for this algorithm are regular, as only N/P coefficients are updated at each iteration.

The sequential NLMS algorithm sequentially updates filter coefficients per iteration:

$$w_i(t+1) = \begin{cases} w_i(t) + \alpha x(t-i+1)e(t), & \text{if } (t+1-i) \mod P = 0 \\ w_i(t), & \text{otherwise.} \end{cases} \tag{2.44}$$

The reduction in the complexity of the algorithm is approximately proportional to the reduction in the number of filter coefficients updated per iteration.

2.4.2 Selective Partial-Updating Adaptive Algorithms

The main disadvantage of partial-updating algorithms is the reduction in convergence speed. The convergence speed of adaptive algorithm is inevitably reduced when only a subset of the filter coefficients are updated per iteration. The reduction in the convergence speed is proportional to the reduction in the number of filter coefficients updated.

The selective partial-updating adaptive algorithms were developed to alleviate this problem [47–50]. The common goal of these algorithms is to update selected blocks of filter coefficients at every iteration using a selection criterion. Partitioning the input vector $\mathbf{x}(t)$ and the coefficient vector $\mathbf{w}(t)$ into M blocks and every block has $K = N/M$ elements (P is an integer):

$$\mathbf{x}(t) = [\mathbf{x}_1^T(t)\ \mathbf{x}_2^T(t) \cdots \mathbf{x}_M^T(t)]^T, \tag{2.45}$$

$$\mathbf{w}(t) = [\mathbf{w}_1^T(t)\ \mathbf{w}_2^T(t) \cdots \mathbf{w}_M^T(t)]^T \tag{2.46}$$

It is assumed that B blocks will be updated at every iteration. The selection of blocks can be carried out by considering the following constrained minimization problem:

$$\begin{aligned} &\min_{I \in S} \min_{\mathbf{w}_I(t+1)} \|\mathbf{w}_I(t+1) - \mathbf{w}_I(t)\|_2^2, \\ &\text{subject to } d(k) = \mathbf{x}^T(t)\mathbf{w}(t+1), \end{aligned} \tag{2.47}$$

where I is B-subset of the set $\{1, 2 \cdots M\}$ and S is the collection of all B-subsets. The resulting selective partial-updating NLMS (SPU-NLMS) algorithm is given by:

$$\begin{aligned} &\mathbf{w}_I(t) = \mathbf{w}_I(t) + \alpha \frac{\mathbf{x}_I(t)e(t)}{\mathbf{x}_I^T(t)\mathbf{x}_I(t) + \delta}, \\ &I = \{i : \|\mathbf{x}_i(t)\|_2^2 \text{ is one of the } B \text{ largest among } \|\mathbf{x}_1(t)\|_2^2, \cdots, \|\mathbf{x}_K(t)\|_2^2\}. \end{aligned} \tag{2.48}$$

This selection criterion ranks the regressors vector blocks according to their squared Euclidean norms and selects those blocks with the B largest norms for updating.

If $B = M$, the SPU-NLMS algorithm becomes identical to the NLMS algorithm. If $P = 1$, i.e. $M = N$, the SPU-NLMS algorithm becomes M-Max PU-NLMS algorithm, where only the M coefficients whose input vector is the M largest ones. The M-Max PU-NSLMS algorithm is shown to have a convergence rate that is closest to the full updating algorithm.

This approach can be naturally extended to APA, where only a fraction of coefficients are updated. The SPU-APA was summarized in [51].

2.4.3 Set-Membership Adaptive Algorithm

Another efficient approach to reduce computational complexity is to employ a set-membership filtering (SMF) approach [52–55]. SMF adaptive algorithms feature reduced computational complexity primarily due to data-selective updating. In set-membership filtering, the adaptive filter $\mathbf{w}(t)$ is designed to achieve a specified bound on the magnitude of the output error.

A constraint set $\mathcal{H}(t)$ contains all vectors $\mathbf{w}$ that result in an output error $e(t)$ bounded by γ.

$$\mathcal{H}(t) = \{\mathbf{w} \in C^N : |d(t) - \mathbf{w}^T\mathbf{x}(t)| \leq \gamma\}. \tag{2.49}$$

SMF adaptive algorithms seek solutions that belong to the exact membership set $\psi(t)$ constructed by input signal and desired signal pairs,

$$\psi(t) = \bigcap_{i=1}^{t} \mathcal{H}(t). \tag{2.50}$$

The idea of set-membership adaptive recursion techniques is to adapt the coefficient vector such that it will always remain within the constraint set. The framework of set-membership filtering is similar to the LMS adaptation process:

$$\mathbf{w}(t+1) = \begin{cases} \mathbf{w}(t) + \left(1 - \dfrac{\gamma}{|e(t)|}\right)\dfrac{\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{x}(t) + \delta}, & \text{if } |e(t)| > \gamma \\ \mathbf{w}(t), & \text{otherwise.} \end{cases} \tag{2.51}$$

Here, γ is a predetermined parameter as a bound on the noise and generally $\gamma = \sqrt{2}\sigma_v$ can get good performance. This adaptation process states that if the previous estimate already belongs to the constraint set, $\mathbf{w}(t) \in \mathcal{H}(t)$, it is a feasible solution and no updating is needed. However, if $\mathbf{w}(t) \notin \mathcal{H}(t)$, coefficients updating is required.

The set-membership filtering approach was smoothly extended to APA in [56, 57].

2.4.4 Reduced-Complexity Proportionate Adaptive Algorithms

In order to reduce the computational complexity of proportionate adaptive algorithms, the aforementioned complexity-reduced algorithms are applied into the proportionate adaptation.

The selective-partial updating proportionate NLMS algorithm was proposed in [58]. It is assumed that B blocks will be updated at every iteration. The selection of blocks can be carried out by considering the following constrained minimization problem:

$$\begin{aligned} &\min_{I\in S} \min_{\mathbf{w}_I(t+1)} \left\{ [\mathbf{w}_I(t+1) - \mathbf{w}_I(t)]^T \, \mathbf{G}^{-1}(t) \, [\mathbf{w}_I(t+1) - \mathbf{w}_I(t)] \right\}, \\ &\text{subject to } d(k) = \mathbf{x}^T(t)\mathbf{w}(t+1), \end{aligned} \tag{2.52}$$

where I is B-subset of the set $\{1, 2 \cdots M\}$ having B elements and S is the collection of all B-subsets.

The resulting selective partial-updating PNLMS algorithm is given by

$$\begin{aligned} \mathbf{w}_I(t) &= \mathbf{w}_I(t) + \alpha \frac{\mathbf{G}_I(t)\mathbf{x}_I(t)e(t)}{\mathbf{x}_I^T(t)\mathbf{G}_I(t)\mathbf{x}_I(t) + \delta}, \\ I &= \left\{ i : \|\mathbf{x}_i(t)\|_G^2 \text{ is one of the } B \text{ largest among } \|\mathbf{x}_1(t)\|_G^2, \cdots, \|\mathbf{x}_M(t)\|_G^2 \right\}. \end{aligned} \tag{2.53}$$

This selection criterion ranks the regressors vector blocks according to their weighted Euclidean norm and selects those blocks with the M largest norms for updating. A method is proposed to reduce the computational complexity further in [59, 60]. There also existing some other methods to reduce the computational complexity using partial updating, such as [41, 61].

The set-membership filtering adaptive algorithms can be also applied into proportionate adaptation process to reduce the computational complexity of proportionate adaptive algorithms, and decrease misalignment at the same time. The set-membership proportionate NLMS algorithms can be depicted by:

$$\mathbf{w}(t+1) = \begin{cases} \mathbf{w}(t) + \left(1 - \dfrac{\gamma}{|e(t)|}\right) \dfrac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \delta}, & \text{if } |e(t)| > \gamma \\ \mathbf{w}(t), & \text{otherwise.} \end{cases} \tag{2.54}$$

The set-membership filtering can be extended to PAPA, whose coefficient updating is described as [62]:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t) + \delta\mathbf{I}\right]^{-1}\boldsymbol{\alpha}(t)\mathbf{e}(t), \tag{2.55}$$

where $\boldsymbol{\alpha}(t) = \mathrm{diag}\{\alpha_0(t)\ \cdots\ \alpha_{P-1}(t)\}$ is a time-varying diagonal step-size matrix and

$$\alpha_p(t) = \begin{cases} 1 - \dfrac{\gamma}{|e_p(t)|} & \text{if } |e_p(t)| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{2.56}$$

Here $e_p(t)$ is the p-th element of error vector $\mathbf{e}(t)$.

2.5 Summary

This chapter introduces adaptive filtering algorithms, including the conventional adaptive algorithms, proportionate adaptive algorithms and their recent advances. The basic principles of adaptive filtering algorithms are introduced at first. Then, conventional adaptive filtering algorithms are briefly reviewed. Proportionate adaption and various proportionate adaptive algorithms are discussed in detail. At last, various adaptive algorithms to reduce the computational complexity of adaptive algorithms are introduced, where combination of these algorithms and proportionate adaptation is included also.

Chapter 3

Incorporating Sparsity into Proportionate Adaptative Algorithms

3.1 Introduction

The PNLMS algorithm greatly speeds up the convergence for sparse impulse response. However, its convergence begins to slow dramatically after a fast initial period. Furthermore, it is effective only when the target impulse response is very sparse. Its performance for non-sparse systems can be relatively poor, even slower than the NLMS algorithm.

In a time-varying environment, the underlying impulse response may vary over a relatively large range that its sparsity could change from sparse to non-sparse. For example, in an acoustic echo cancellation system, sparsity of room transfer function could vary greatly. Two factors influence the sparsity of the underlying room transfer function: distance between the speaking person and the microphone, the characteristics of the acoustic environment [6, 63]. Usually, the room transfer function changes significantly with distance between the person speaking and the microphone picking up speech. If the person is near the microphone, the acoustic echo path will be sparse enough to be applicable for PNLMS algorithm, and vice versa. The acoustic characteristics of environment can be evaluated by the reverberation time, which is proportional to the volume of the enclosed space and inversely proportional to the absorption area. For an outdoor environment, the reverberation time is reduced significantly due to the lack of reflections from the walls. As a consequence, acoustic impulse response of an outdoor environment can be considered to be more sparse than that of an enclosed space. The sparsity variation of acoustic impulse response also arises due to changes in temperature, pressure, acoustic source movements and changes of the acoustic environment.

Therefore, a proportionate adaptive algorithm is required to perform well for both sparse and non-sparse impulse responses. Several approaches have been proposed to solve these problems, such as the PNLMS++ algorithm [29], the IPNLMS algorithm [30], and the IIPNLMS algorithm [32]. Many other variants have been proposed in various applications, see [12, 13, 34, 40, 64].

In this chapter, a new approach is proposed to improve convergence of proportionate adaptive algorithms for non-sparse impulse response. The proposed approach first estimate how sparse is the target impulse response, then automatically adjust the related parameters to assure their convergence.

3.2 Measure of Sparsity

The first problem encountered is how to measure the sparsity of a given impulse response? Many definitions of sparsity have been proposed and used in the literature to quantify how much energy of a vector is packed into only a few components [4] [65]. The sparsity $\zeta(\mathbf{w})$ of a vector $\mathbf{w}$ can be defined using the difference between its L-1 norm and L-2 norm, as

$$\xi(\mathbf{w}) = \frac{N}{N - \sqrt{N}}\left(1 - \frac{||\mathbf{w}||_1}{\sqrt{N}||\mathbf{w}||_2}\right), \tag{3.1}$$

where N is the length of $\mathbf{w}$, $||\mathbf{w}||_1$ and $||\mathbf{w}||_2$ are the L-1 norm and L-2 norm of $\mathbf{w}$, respectively:

$$||\mathbf{w}||_1 = \sum_{n=0}^{N-1} |w_n|, \tag{3.2}$$

$$||\mathbf{w}||_2 = \sqrt{\sum_{n=0}^{N-1} |w_n|^2}. \tag{3.3}$$

This definition possesses the following properties:

1. $0 \leq \xi(\mathbf{w}) \leq 1$;
2. $\xi(c\mathbf{w}) = \xi(\mathbf{w})$, where c is non-zero constant;
3. $\xi(\check{\mathbf{w}}) = \xi(\mathbf{w})$, where $\check{\mathbf{w}}$ is a permutation of $\mathbf{w}$ (up to signs);
4. $\xi(\mathbf{w}) = 1$, if and only if $\mathbf{w}$ contains only **one** single non-zero component;
5. $\xi(\mathbf{w}) = 0$, if and only if **all** components are equal (up to signs).

3.3 Sparsity-Guided IPNLMS

For the IPNLMS algorithm, $g_n(t)$ is approximately proportionate to the absolute magnitude of the target impulse response. Furthermore, by adjusting the parameter β, its convergence behavior is adjusted. For a sparse impulse response, IPNLMS converges faster if β close to 1. And for a dispersive one, a β close to -1 should be assigned to assure that it does not converge slower than NLMS.

For impulse responses with different sparsity, different β should be used to achieve a fast convergence speed. Unfortunately, this parameter is *predetermined* and *constant* during the adaptation process. If this parameter is *smart* enough to know the sparsity of target impulse response, then it can adjust itself to an optimal value, the IPNLMS algorithm will converge at fast as possible.

It is necessary to rewrite the definition of diagonal elements of the IPNLMS algorithm in

(2.37) into another form by simple replacement

$$g_n(t) = (1-\beta) + \frac{\beta\,|w_n(t)|}{\frac{1}{N}\sum_{i=0}^{N-1}|w_i(t)| + \epsilon}, \quad 0 \le \beta < 1. \tag{3.4}$$

Here the range of β is regulated from $[-1, 1)$ to $[0, 1)$. This new equation not only solves the range mismatch of $\beta(\in [-1, 1))$ and $\xi(\mathbf{w})(\in [0, 1])$, but also directly reveals the relationship how the adjustable parameter β influences the structure of the step-size control matrix.

Then a new $g_n(t)$ with a time-varying $\beta(t)$ is proposed in the IPNLMS algorithm,

$$g_n(t) = (1-\beta(t)) + \frac{\beta(t)|w_n(t)|}{\frac{1}{N}\sum_{i=0}^{N-1}|w_i(t)| + \epsilon}, \quad 0 \le \beta(t) < 1. \tag{3.5}$$

Because the target impulse response is unknown, the sparsity of current adaptive filter is calculated as

$$\xi(t) = \frac{N}{N-\sqrt{N}}\left(1 - \frac{\|\mathbf{w}(t)\|_1}{\sqrt{N}\|\mathbf{w}(t)\|_2}\right). \tag{3.6}$$

The estimation of sparsity is then transformed into the parameter domain of $\beta(t)$ using aforementioned function

$$\beta(t) = f(\xi(t)). \tag{3.7}$$

The estimation of sparsity is then transformed into the parameter domain of β using aforementioned function $f(x)$. The initial value of $\xi(0)$ should be a large value that accelerates the convergence of large coefficients. The rest work is same to the IPNLMS algorithm.

3.4 Relation between ξ and β

The next problem is how to determine $\beta(t)$ according to the sparsity of impulse response ξ. Although it is known that there exists a real relationship between convergence speed of IPNLMS and the sparsity of the target impulse response, it seems hopeless to derive a straightforward expression between them. Fortunately, large amount of computer simulations can reveal their relationship.

3.4 Relation between ξ and β

In order to obtain various impulse responses with different sparsity, two methods are used. In the first method, the active coefficients of impulse response have same value 1, and the inactive coefficients are 0. In the second method, the impulse responses of different sparsity are synthetically generated using an $N \times 1$ exponentially decaying window that is defined as [66]

$$\mathbf{e}_{win} = [\overbrace{1/p, \cdots, 1/p}^{q},\ e^{-1/\psi}, \cdots, e^{-(N-q)/\psi}]^T. \tag{3.8}$$

Here q models the bulk delay, p is a positive number and $\psi(> 0)$ is a decay constant. The impulse response is obtained by

$$\mathbf{h} = \mathbf{e}_{win} \star \dot{\mathbf{w}}, \tag{3.9}$$

where $\dot{\mathbf{w}}$ is an $N \times 1$ vector of Gaussian sequence and $\star$ denotes element-wise multiplication of vectors. By adjusting ψ from 0 to 1, different impulse responses are obtained with different sparsity. Figure 3.1 illustrates 3 impulse responses used in computer simulations. It can be seen that this kind of impulse response can approximately represent most of real world transmission channel, such as the room transfer function.

In order to discover relationship between β and ξ, various impulse responses with different sparsity, which were generated using two kinds of impulse responses introduced above, were tested. For each impulse response, various β was tested from 0.0 to 0.95 with incrementation of 0.05. Then by comparing their convergence time, β that make the IPNLMS algorithm converge at the fastest speed is determined with respect to the sparsity ξ of the given impulse response.

The results of a large number of simulations are illustrated in Fig. 3.2. $+$ is for the first kind of impulse responses. $\circ$ is for the second kind of impulse responses. The dashed line is result of cubic cure fitting. It is shown that, even with different kinds of impulse responses, the following consistent conclusions can be drawn:

1. For a sparse impulse response $\mathbf{w}$ whose $\xi(\mathbf{w}) > 0.6$, more sparse it is, larger a β is necessary;
2. For a non-sparse impulse response whose $0.2 < \xi(\mathbf{w}) < 0.6$, β is not sensitive to the

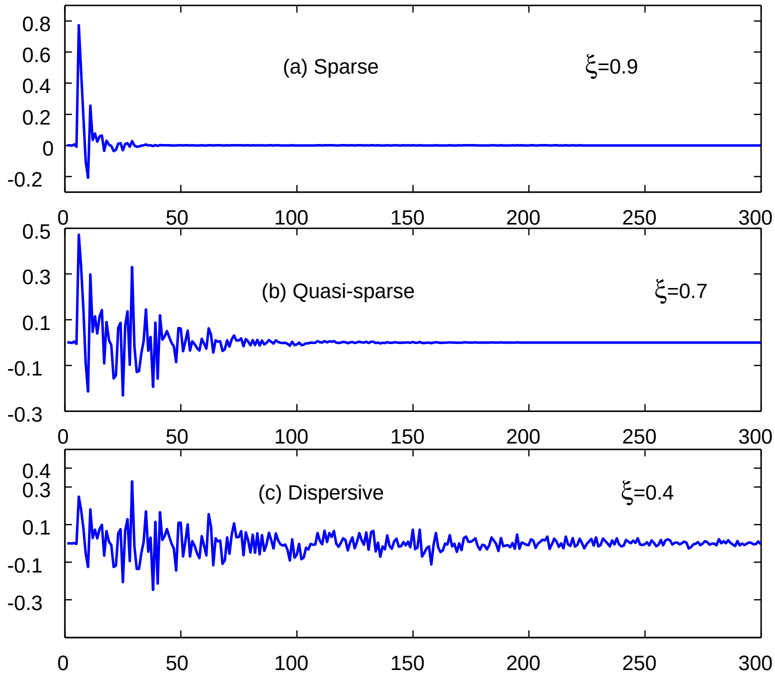

Fig. 3.1 Three impulse responses with different sparsity.

sparsity: for any $\beta \in [0.1, 0.2]$, the IPNLMS algorithm converges at almost same speed;

3. For a dispersive impulse response whose $\xi(\mathbf{w}) < 0.2$, β should be smaller.

Using curve fitting of a cubic function, a simple real function $f(\xi)$ is obtained, as

$$f(\xi) = 3\xi^3 - 3\xi^2 + \xi. \tag{3.10}$$

3.5 Simulation Results of Proposed Approach

To evaluate the performance of the proposed algorithm, many computer simulations were conducted to identify various impulse responses with an adaptive filter. The common conditions for the simulations are described as follows. Both the unknown system $\mathbf{h}$ and the adaptive filter $\mathbf{w}$ have 300 coefficients, $N = 300$. The input signal $x(t)$ and the disturbance $v(t)$ are uncorrelated zero-mean Gaussian signal with variance of 1.0 and 10^{-4}, respectively. The constant step size $\alpha = 0.25$.

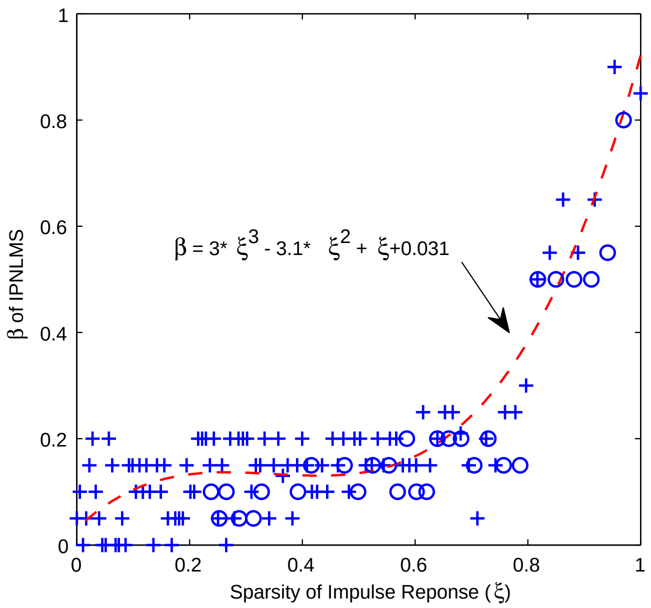

Fig. 3.2 Relationship between sparsity of impulse response ξ and β of IPNLMS.

Then using three impulse responses in Fig. 3.1, the convergence of the related algorithms is compared with the proposed approach. The simulation results are illustrated in Fig. 3.3 - 3.5. The results verify the effectiveness of the proposed algorithm. It can be seen from Fig. 3.3(a) that for a sparse impulse response as in Fig. 3.1(a), the proposed algorithm achieved as fast convergence speed as IPNLMS. For a quasi-sparse impulse response as in Fig. 3.1(b), the proposed algorithm over-performs IPNLMS and NLMS, as shown in Fig. 3.3(b). It can be observed from Fig. 3.3(c) that for a dispersive impulse response as in Fig. 3.1(c), the proposed algorithm retains the fast convergence of NLMS while IPNLMS fails.

The tracking ability of proposed algorithm is shown in Fig. 3.4. The first impulse response to be identified is in Fig. 3.1(a) and then abruptly changes to Fig. 3.1(c) at iteration 10^4. It can be seen from Fig. 3.4 that the proposed algorithm behaves better than IPNLMS in a time-varying environment. The change of $\beta(t)$ and $\xi(t)$ is shown in Fig. 3.5. In the simulations, the calculations of sparsity of adaptive filter were executed once every 10 iterations, and the value of $\beta(t)$ converged soon.

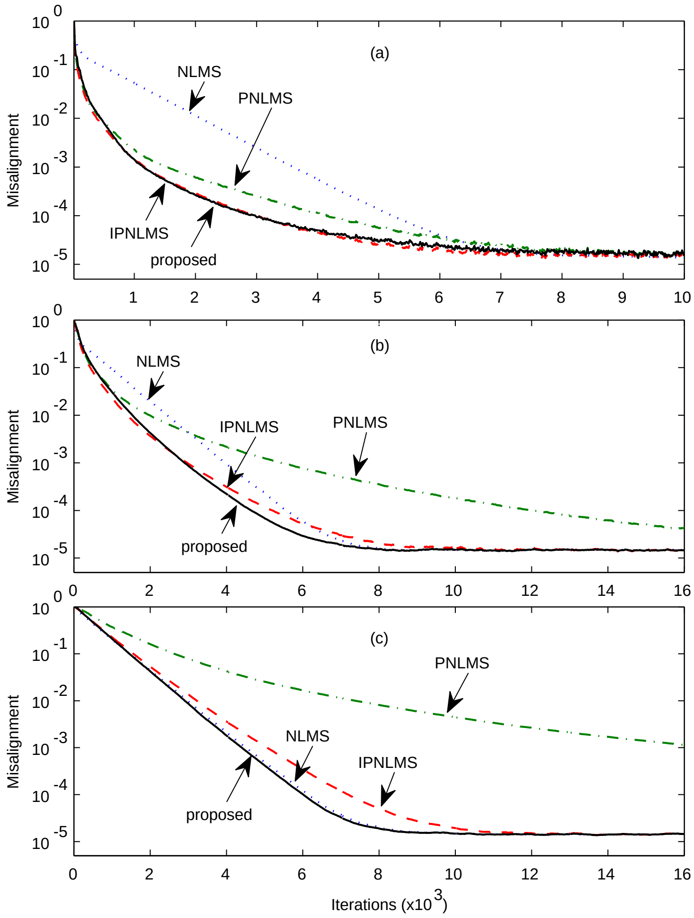

Fig. 3.3 Comparison of convergence speed of Sparsity-Guided IPNLMS. (a) is with the impulse response in Fig. 3.1(a). (b) is with Fig. 3.1(b). (c) is with Fig. 3.1(c).

3.6 Sparsity-Guided MPNLMS Algorithm

The MPNLMS algorithm has a consistent convergence speed for a sparse impulse response, compared to the PNLMS algorithm. In a time-varying environment, the impulse response may vary over a relatively large range that would be non-sparse at times. The behavior of the MPNLMS algorithm can be adjusted between proportionate adaptation and the conventional NLMS algorithm by adjusting the parameters, δ_ρ and ρ. These parameters are constant during the MPNLMS. In the IPNLMS algorithm, for a sparse impulse response, a value of β

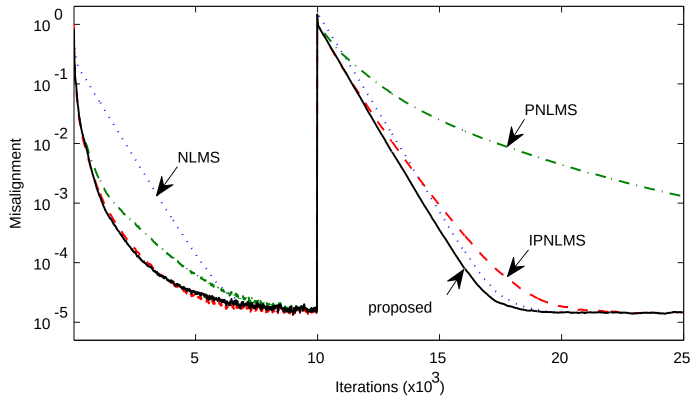

Fig. 3.4 Comparison of tracking abilities, when the unknown impulse response suddenly change at iteration 10^4 from (a) to (c).

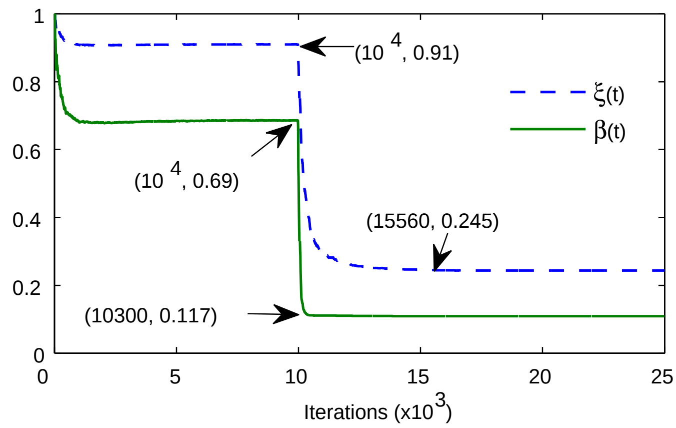

Fig. 3.5 Trace of $\xi(t)$ and $\beta(t)$ in simulation of suddenly change of impulse response at iteration 10^4 from (a) to (c).

close to 1 is used to achieve fast convergence, and for a dispersive channel an β close to -1 is assigned to assure that it does not converge slower than NLMS.

A new algorithm is proposed to incorporate the sparsity into the MPNLMS algorithm so that its performance for non-sparse impulse response is improved. The new algorithm is referred to as the improved MPNLMS (IMPNLMS) algorithm throughout this section.

Now, the diagonal elements of the step-size control matrix $\mathbf{G}(t)$ for the proposed

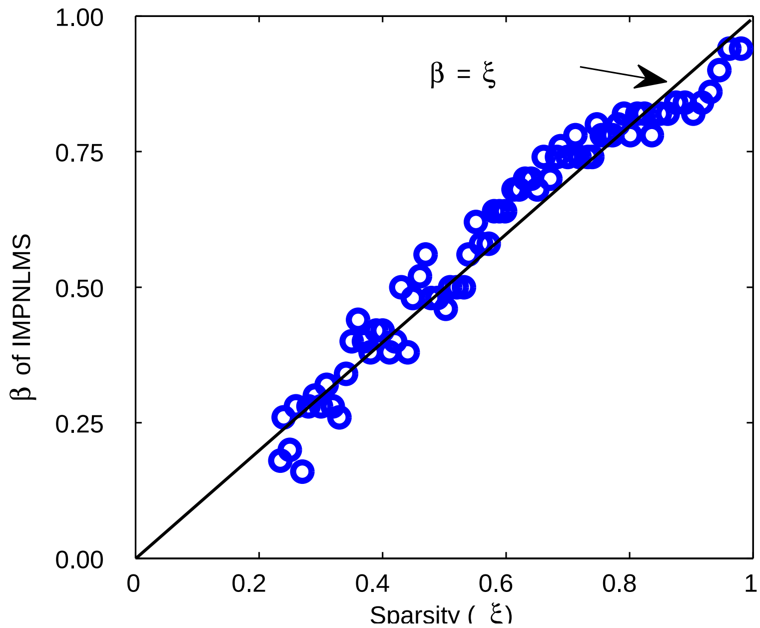

Fig. 3.6 β for different sparsity ξ.

IMPNLMS algorithm is

$$g_n(t) = (1 - \beta(t)) + \frac{\beta(t)\ F(|w_n(t)|)}{\frac{1}{N}\ \|F(|\mathbf{w}(t)|)\|_1 + \epsilon}. \tag{3.11}$$

The estimation of channel sparsity is then transformed into the parameter domain of β in the IPNLMS algorithm with

$$\beta(t) = \xi(t). \tag{3.12}$$

This relation was obtained through numerous simulations using the same approach in Section 3.4. The simulation results are illustrated in Fig. 3.6. It can seen that the good $\beta(t)$ is a linear function of the impulse response sparsity $\xi(t)$.

3.7 Simulation Results of the IMPNLMS Algorithm

To evaluate the performance of the proposed algorithm, many simulations were conducted with four algorithms: NLMS, IPNLMS, MPNLMS, and the proposed IMPNLMS algorithm.

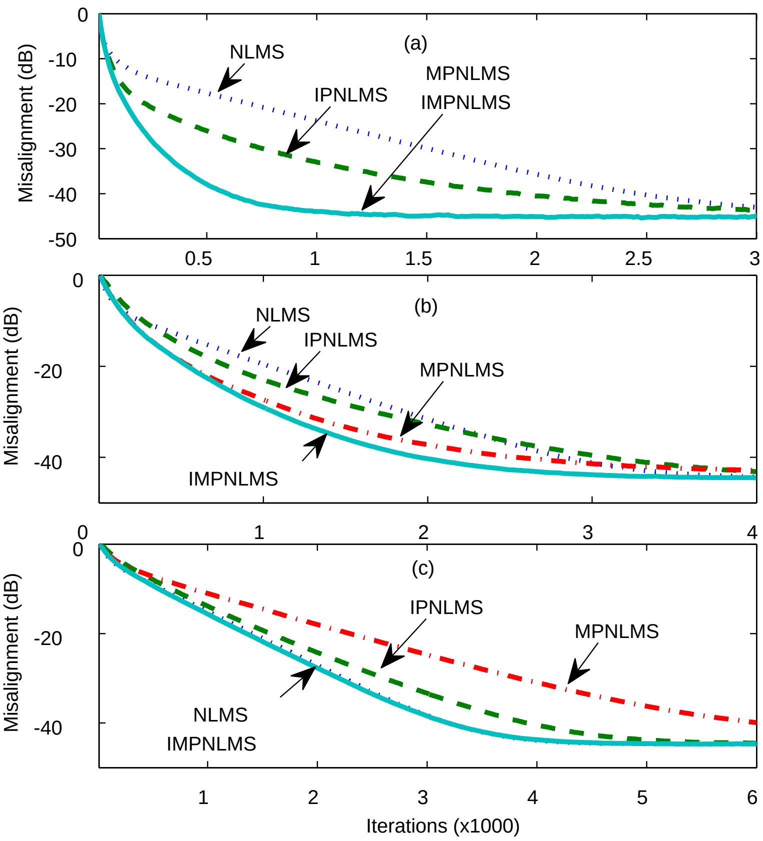

Fig. 3.7 Simulation 1: convergence speed with white input.

The common conditions for the simulations are as follows. It is assumed that the unknown impulse response $\mathbf{h}$ is modeled by a FIR filter with $N = 300$ coefficients and that the adaptive filter $\mathbf{w}$ has the same number of coefficients. The disturbance $v(t)$ is a zero-mean Gaussian signal with a variance of 0.01. A constant step size $\alpha = 0.25$ was used for all algorithms. For IPNLMS, $\beta = 0.25$. The initial value of ξ is a large number, such as 0.96. The results illustrated in the following figures are average of 100 simulations. The performance of the echo path identification is quantified using the mean square deviation defined as $10\log_{10}\|\mathbf{h}-\mathbf{w}(t)\|_2^2/\|\mathbf{h}\|_2^2$.

In Simulation 1, white Gaussian noise with zero-mean and unit variance was used as the input of the simulated system. The convergence speed of the related algorithms with three impulse responses shown in Fig. 3.1 respectively are compared in Fig.3.7. It can be observed

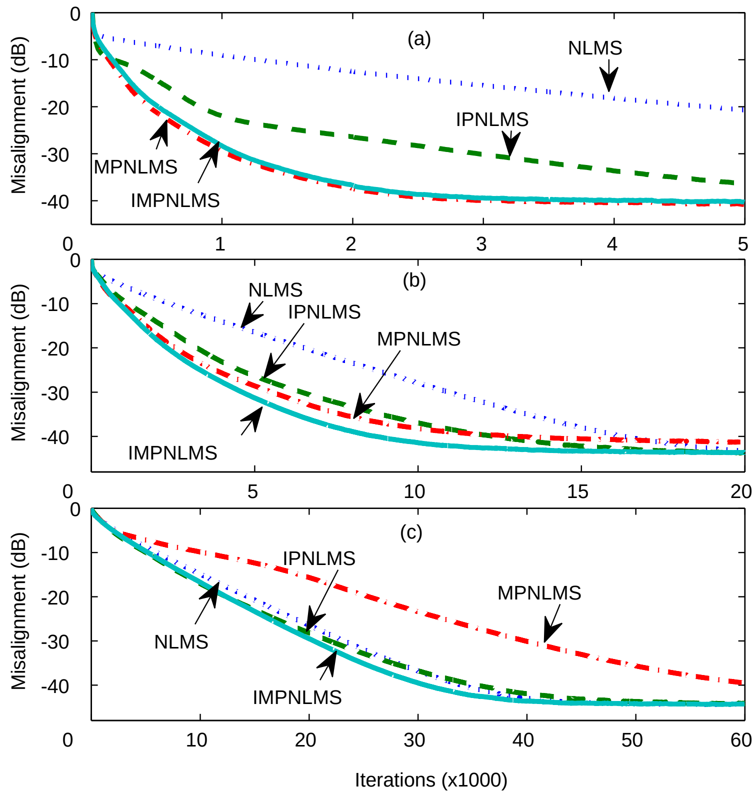

Fig. 3.8 Simulation 2: convergence speed with correlated input.

from the figure that the proposed IMPNLMS algorithm converges as fast as MPNLMS for a sparse channel, as shown in Fig. 3.7(a), and that for a dispersive channel in Fig. 3.1(c), the proposed IMPNLMS algorithm does not convergence slower than the NLMS algorithm while the MPNLMS algorithm relatively converges slow, as shown in Fig. 3.7(c).

In Simulation 2, the performance of the aforementioned algorithms was evaluated with the highly correlated signal as input. The input signals are obtained with the model $x(k) = u(k)/(1 - 0.9z^{-1})$, where $u(k)$ is a discrete white Gaussian signal with zero-mean and unit variance. Their convergence speed is compared in Fig. 3.8. It can be observed that for the colored signals the proposed IMPNLMS algorithm also converges faster than MPNLMS when the impulse response is dispersive. The tracking ability is very important for the real-world application especially in time-varying environment.

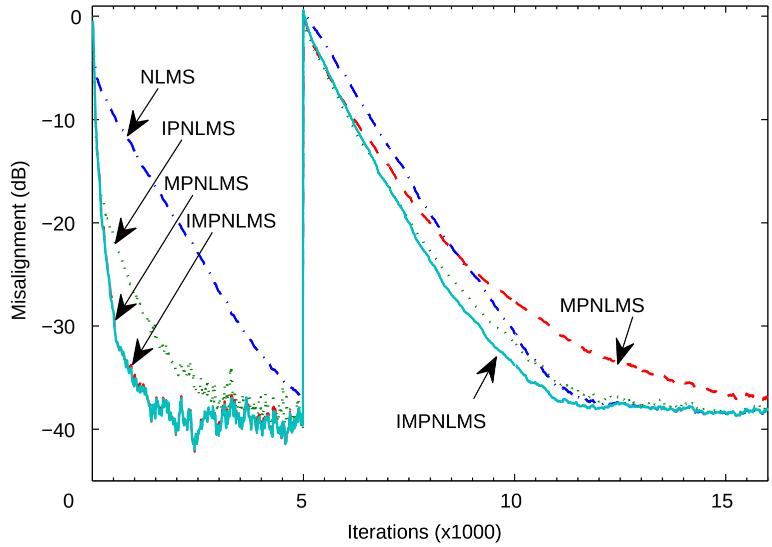

Fig. 3.9 Simulation 3: comparison of tracking ability.

In Simulation 3, the reconvergence performance of the proposed algorithm was evaluated. The first channel to be identified is that in Fig. 3.1(a) and then abruptly changes to a dispersive impulse response in Fig. 3.1(c) at iteration 5000. It can be observed from Fig. 3.9 that the proposed IMPNLMS algorithm behaves better than the MPNLMS algorithm in a time-varying environment.

3.8 Computational Complexity

The additional computational complexity of the proposed sparsity detection method is light. The IPNLMS algorithm need $4N + 1$ multiplication/division in each iteration. The additional computation of the proposed approach is to estimate the sparsity, in (3.1), (3.6), that consumes $N + 5$ multipliers/division, $2N + 4$ additions and 1 square root. It is unnecessary to compute it at each iteration, because it changes relatively slowly. It can be calculated in a large interval varying from 10 to N, thus its computational cost is greatly reduced without loss of performance. So the proposed approach does not significantly increase its computational complexity.

3.9 Conclusions

There exists a deterministic relationship between the given sparsity of target impulse response and the optimal step gain matrix that can guarantee fastest convergence speed. Computer simulations reveal their relationship. Then a new algorithm is proposed to incorporate sparsity of target impulse response into the IPNLMS algorithm and MPNLMS algorithm. The proposed approach exploits the sparsity information of the target system. It adaptively detects the sparsity of the adaptive filter, therefore a suitable β is adjusted accordingly. Simulation results show that the proposed algorithm achieves fastest convergence speed than the original algorithms, whose parameter is predetermined according to rough estimate. The proposed approach is especially suitable to a time-varying impulse response that its sparsity changes.

Chapter 4

Better Approximation of Segment Proportionate NLMS Algorithm

4.1 Introduction

Among various different proportionate adaptive filtering algorithms, the MPNLMS algorithm achieves optimal step-size control matrix $\mathbf{G}(t)$ in the framework of proportionate adaptation [33, 34]. Instead of directly using the current estimated magnitude of a coefficient, it uses the logarithm of the magnitude as the proportionate step gain of each coefficient. The main disadvantage of the MPNLMS algorithm is its heavy computational complexity because it needs N logarithmic operations at each iteration, which is cost in a DSP implementation. A segment function is proposed to replace these logarithmic operations, which is called the segment PNLMS (SPNLMS) algorithm [34]. The computation cost of the SPNLMS algorithm is almost same to that of PNLMS and IPNLMS but its convergence is almost as fast as MPNLMS. Hence the SPNLMS algorithm is favorable for practical application involving a sparse impulse response.

However, there are several problems in MPNLMS and SPNLMS. In this chapter, several approaches are proposed to improve them. The unique contributions in the chapter are as follows. At first, a straightforward derivation is provided to choose the step-size control matrix $\mathbf{G}$ so that the fastest convergence speed is achieved. Then, an approach is proposed to improve their performance by adaptively determining the important parameter μ, which is related to an objective convergence criterion. Thirdly, a refined segment function is proposed for the SPNLMS algorithm to approximate mu-law function as exact as possible, hence fast convergence of MPNLMS is retained in various environments. At last, the computational complexity of proportionate adaptive algorithms is discussed and some methods are proposed to reduce the computational complexity of the proposed adaptive algorithm.

4.2 Segment Proportionate NLMS Algorithm

Applying a diagonal step-size control matrix, $\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}$, in the LMS coefficient update equation in order to associate one step size with each coefficient, results in the proportionate LMS algorithm, which is given by

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{G}(t)\mathbf{x}(t)e(t). \tag{4.1}$$

A normalized version can be obtained by dividing the coefficient updating term in (4.1) with the Euclidean norm of the input vector, i.e., $||\mathbf{x}(t)||_2^2$, in order to eliminate influence of the input signal power. The resulting PNLMS algorithm has a form as in [8]:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{||\mathbf{x}(t)||_2^2 + \delta}, \tag{4.2}$$

where $||\mathbf{x}(t)||_2^2 = \mathbf{x}^T(t)\mathbf{x}(t)$ is the Euclidean norm of $\mathbf{x}(t)$. The main problem of this formula is, this dominator may cause algorithm unstable for impulsive excitation signals [29].

The normalized version of the proportionate LMS algorithm presents another form in the literature, where the dominator is the $\mathbf{G}$ weighted Euclidean norm of $\mathbf{x}(t)$, which is denoted by $||\mathbf{x}(t)||_G^2$,

$$||\mathbf{x}(t)||_G^2 = \mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t). \tag{4.3}$$

There, the normalized version is depicted as [30, 33–35, 58]:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \delta}, \tag{4.4}$$

where δ is an regularization parameter. This formula is a natural extension of (4.1).

This result is a solution of a constrained optimization problem in a way similar to that of NLMS, i.e., to find a variable step-size so that the Euclidean norm of the coefficient vector updating in successive time index is minimized, subject to a constrain that the *a posteriori* error is zero [2]:

$$\mathbf{w}(t+1) = \min_{\mathbf{w}(t+1)} \left\{[\mathbf{w}(t+1) - \mathbf{w}(t)]^T \mathbf{G}^{-1}(t)[\mathbf{w}(t+1) - \mathbf{w}(t)]\right\}, \tag{4.5}$$

$$\text{subject to} \quad \mathbf{x}^T(t)\mathbf{w}(t+1) = d(t). \tag{4.6}$$

This problem can be solved by using the method of Lagrange multipliers [2], whose cost function to minimize is

$$\mathcal{J}(t) = [\mathbf{w}(t+1) - \mathbf{w}(t)]^T\, \mathbf{G}^{-1}(t)\, [\mathbf{w}(t+1) - \mathbf{w}(t)] + 2\lambda \left[d(t) - \mathbf{x}^T(t)\mathbf{w}(t+1)\right], \tag{4.7}$$

where λ is a Lagrange multiplier.

Setting $\dfrac{\partial \mathcal{J}(t)}{\partial \mathbf{w}(t+1)} = \mathbf{0}$ and $\dfrac{\partial \mathcal{J}(t)}{\partial \lambda} = 0$, yields [Refer to Appendix A.2]

$$\mathbf{G}^{-1}(t)[\mathbf{w}(t+1) - \mathbf{w}(t)] - \lambda \mathbf{x}(t) = \mathbf{0}. \tag{4.8}$$

$$d(t) - \mathbf{x}^T(t)\mathbf{w}(t+1) = 0. \tag{4.9}$$

Pre-multiplying both sides of (4.8) by $\mathbf{G}(t)$, and then rearranging it,

$$\mathbf{w}(t+1) - \mathbf{w}(t) = \lambda \mathbf{G}(t)\mathbf{x}(t), \tag{4.10}$$

and then pre-multiplying both sides of (4.10) by $\mathbf{x}^T(t)$, combining with (4.9), remembering $e(t) = d(t) - \mathbf{x}^T(t)\mathbf{w}(t)$, yields

$$e(t) = \lambda \mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t), \tag{4.11}$$

$$\lambda = \frac{e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)}. \tag{4.12}$$

Then substituting λ into (4.10), introducing a global constant step-size parameter α to balance between the convergence and the steady-state misalignment, a popular formula of proportionate NLMS algorithms is obtained as in (4.4).

Different definitions of $\mathbf{G}(t)$ have been proposed, which can be summarized uniformly as:

$$\tilde{g}_n(t) = \max\{F(w_n(t)), \rho \max\{\delta_\rho, F(w_0(t)), \cdots, F(w_{N-1}(t))\}\}, \tag{4.13}$$

$$g_n(t) = \frac{\tilde{g}_n(t)}{\frac{1}{N}\sum_{i=0}^{N-1} \tilde{g}_i(t)}, \tag{4.14}$$

$$\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}. \tag{4.15}$$

Here, $F(w_n(t))$ is a real-valued function to map the current estimate value of coefficient $w_n(t)$ to step-size gain.

The PNLMS algorithm is the first proportionate adaptive algorithm, where $F(w_n(t)) = |w_n(t)|$. For MPNLMS, it has

$$F(w_n(t)) = \ln(1 + \mu |w_n(t)|), \tag{4.16}$$

where μ is an objective convergence criterion and typically $\mu = 1000$. The SPNLMS was proposed to remove the N logarithmic operations of MPNLMS at each iteration, where

$$F(w_n(t)) = \begin{cases} 400|w_n(t)|, & |w_n(t)| < 0.005 \\ 2, & \text{otherwise.} \end{cases} \tag{4.17}$$

The convergence of related adaptive algorithms to identify a sparse impulse response is illustrated in Fig. 4.1. It can seen that the proportionate NLMS algorithms have very fast convergence speed than NLMS, especially MPNLMS and SPNLMS.

In practical application, the SPNLMS algorithm is favorable because of its fast convergence and reasonable computational complexity. However, two problems can be found in the SPNLMS adaptation process. First, the determination of the important parameter μ in (4.16) is objective specifically in network echo cancellation. It is not suitable to a general environment. Second, the segment function in (4.17) is a very coarse approximation of mu-law function in (4.16). It is believed that it can not retain the fast convergence of MPNLMS. As shown in Fig.

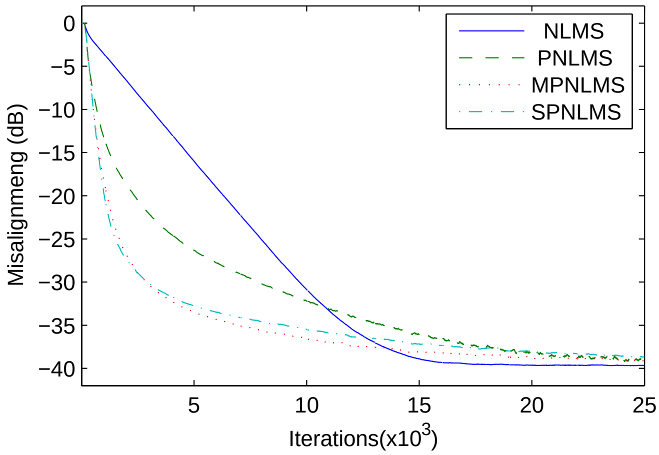

Fig. 4.1 Convergence comparison of proportionate algorithms ($\alpha = 0.2$).

4.1, SPNLMS shows slow convergence than MPNLMS.

4.3 Fastest Step Size Control Matrix G

The PNLMS algorithm was conceived intuitively and it did not provide a quantitative analysis to prove that the proposed $\mathbf{G}$ is optimal. For a sparse impulse response $\mathbf{h}$, many approaches have been proposed to determine $\mathbf{G}$. Among these definitions, there should be one that can provide the fastest convergence speed. In this subsection, a straightforward derivation of $\mathbf{G}$ is provided to achieve the fastest convergence speed. The derivation will follow the similar procedure with that in [34], but this derivation is more straightforward to be understood easily. It is presumed that sequences $x(t)$ and $v(t)$ are zero mean, stationary, jointly normal, and with finite moments, in the following derivation.

Given a sparse impulse response $\mathbf{h}$, among the various definitions of proportionate step size control matrix $\mathbf{G}$, there exists one that can achieve the fastest convergence speed. The following derivation pursuit an optimal $\mathbf{G}$ that achieves the fastest convergence. Starting from steepest descent algorithm with proportionate adaptation, yields [7, 34]

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \mathbf{G}\mathbf{R}\left[\mathbf{h} - \mathbf{w}(t)\right] \tag{4.18}$$

where $\mathbf{R}$ is the auto-correlation matrix of the input vector. Defining the weight error vector as

$$\tilde{\mathbf{w}}(t) = \mathbf{h} - \mathbf{w}(t). \tag{4.19}$$

Following the similar procedure in [34], subtracting $\mathbf{h}$ at both sides of (4.18) and rearranging the terms, results in

$$\begin{aligned}\tilde{\mathbf{w}}(t+1) &= (\mathbf{I} - \alpha\mathbf{GR})\,\tilde{\mathbf{w}}(t)\\ &= (\mathbf{I} - \alpha\mathbf{GR})^t\,\tilde{\mathbf{w}}(0),\end{aligned} \tag{4.20}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix. For Gaussian white input signal with variance σ_x^2, it has $\mathbf{R} = \sigma_x^2\mathbf{I}$. Without loss of generality, it assumes here that the initial value of adaptive filter is zeros, i.e., $\mathbf{w}(0) = \mathbf{0}$, where $\mathbf{0}$ is an $N \times 1$ vector with all elements are zeros. Then it gives $\tilde{\mathbf{w}}(0) = \mathbf{h}$. So it can obtain that

$$\tilde{\mathbf{w}}(t+1) = (\mathbf{I} - \alpha\sigma_x^2\mathbf{G})^t\mathbf{h}, \tag{4.21}$$

For the n-th coefficient, it gives

$$\tilde{w}_n(t+1) = (1 - \alpha\sigma_x^2 g_n)^t h_n. \tag{4.22}$$

When $0 < \alpha\sigma_x^2 g_n < 2$, the adaptive coefficient $w_n(t)$ exponentially converge to h_n. Equation (4.22) implies that the coefficient error of the proportionate algorithms will exponentially converge in the mean sense. In practical applications, which always includes various background noise or/and measurement noise, a coefficient is regarded as convergent after it has reached an ϵ-vicinity of h_n, where ϵ is a objective convergence criterion of adaptive filter.

It is assumed that all coefficients are bigger than ϵ, otherwise it has converged already. Therefore, the n-th coefficient is regarded as convergent when

$$|(1 - \alpha\sigma_x^2 g_n)^t h_n| < \epsilon. \tag{4.23}$$

Define the coefficient convergence time of the n-th coefficient $\tilde{t}_n$ as the number of iterations that it is required to reach the ϵ-vicinity of h_n. By solving this inequality for t, it obtains $\tilde{t}_n$ for the n-th coefficient is

$$\tilde{t}_n = \left\lceil \frac{\log_2 (\epsilon/|h_n|)}{\log_2 \left|1 - \alpha\sigma_x^2 g_n\right|} \right\rceil, \tag{4.24}$$

where $\lceil\cdot\rceil$ is the ceiling function. Assuming that the constant step size α is small enough and $0 < \alpha g_n \sigma_x^2 \ll 1$, the following approximation holds

$$\log_2 \left|1 - \alpha\sigma_x^2 g_n\right| \approx -\frac{\alpha\sigma_x^2 g_n}{\log_2 e}, \tag{4.25}$$

where e is the base of natural logarithms.

Finally, the iteration number of the n-th coefficient to converge is obtained as:

$$\tilde{t}_n \approx \left\lceil \log_2 e \cdot \frac{\log_2 (|h_n|/\epsilon)}{\alpha\sigma_x^2 g_n} \right\rceil. \tag{4.26}$$

This result indicates that the convergence time of the n-th coefficient is proportional to the magnitude of $\log_2 (|h_n|/\epsilon)$ and inversely proportional to g_n.

It seems that the convergence could be improved by increasing g_n. However, the fastest overall convergence speed can be achieved if and only if *all* coefficients reach the ϵ-vicinity of their desired values *at the same time*, otherwise more adaptation has to be wasted for the coefficients that have converged earlier [34].

For convenience sake, defining

$$\mu = \frac{1}{\epsilon}, \tag{4.27}$$

then from (4.26), it is obvious that when

$$g_n = \log_2 (\mu|h_n|), \tag{4.28}$$

all these iteration numbers are identical, i.e.,

$$\tilde{t}_0 = \tilde{t}_1 = \cdots = \tilde{t}_{N-1} = \left\lceil \frac{\log_2 e}{\alpha \sigma_x^2} \right\rceil . \tag{4.29}$$

Consequently, the optimal step-size control matrix $\mathbf{G}$ should be designed according to (4.28) in order to achieve the fastest overall convergence speed.

For the small coefficients that $|h_n| < \epsilon$, (4.28) is not applicable because it will lead to negative g_n. By adding 1 inside the logarithm it can guarantee the proportionate step size g_n positive. It results in

$$g_n = \log_2 (1 + \mu |h_n|) . \tag{4.30}$$

For a relatively large $|h_n|$, this modification will not greatly affect the value of g_n because $\mu|h_n|$ is large enough. For a small $|h_n|$ this will assign g_n a reasonable value to assure these coefficients to convergence and track changes of $\mathbf{h}$.

Actually, $\mathbf{h}$ is unknown or time-varying in real world applications. But the above analysis provides inspiration for design of new proportionate adaptive algorithm. Instead of the real value of $\mathbf{h}$, the magnitude estimate of the adaptive filter can be applied, which is a frequently used method in the adaptive filtering area.

Equation (4.30) becomes a proportionate step size approach, which is similar to that in (4.16):

$$g_n(t) = \log_2 (1 + \mu |w_n(t)|) . \tag{4.31}$$

The above analysis provides an explanation of an optimal step-size control matrix of the proportionate adaptive algorithm. There are two different places with the derivation in [34], which makes this derivation to be understood easily. First, this derivation does not assume $\sum_{n=0}^{N-1} g_n(t) = N$, i.e., it is not necessary to normalize $g_n(t)$ by its average. This modification not only simplifies the derivation, and it also results in a method to save N multiplies, which will be discussed latter. Second, it uses $\log_2$ instead of the natural logarithm $\ln(x)$ in above derivation. This modification will induce a new segment function to better approximate optimal proportionate step size, which will be discussed in Section 4.5.

4.4 Determination of Suitable Convergence Criterion ϵ

It is evident that the convergence of the MPNLMS algorithm partly depends on the choice of convergence criterion ϵ. It is proposed that $\epsilon = 0.001$ is good enough for the most NEC applications [34]. This subsection tries to refine the choice of this parameter.

The mean square deviation (MSD) of the proportionate algorithms has been derived in [8]. The MSD is approximately equal to that of NLMS, as

$$\begin{aligned} \mathrm{MSD} &= \lim_{t\to\infty} E\,\|\mathbf{h} - \mathbf{w}(t)\|_2^2 \\ &= \frac{\sigma_v^2}{\sigma_x^2} \cdot \frac{\alpha}{2-\alpha}. \end{aligned} \tag{4.32}$$

This result is valid for all proportionate NLMS algorithms because it is independent from $\mathbf{G}(t)$.

In a noisy environment, the coefficients of the adaptive filter will randomly walk around its optimal value after the adaptive filter has reached the steady-state. It is well known that $E\{\tilde{w}_n(t)\} = 0, (t \to \infty, \forall n)$. Therefore, (4.32) implied that the variance of $\tilde{w}_n(t)$ is:

$$\sigma_{\tilde{w}_n}^2 = \frac{\sigma_v^2}{\sigma_x^2} \cdot \frac{\alpha}{N(2-\alpha)}. \tag{4.33}$$

This result means that $w_n(t)$ will fluctuate around h_n in the range of

$$\sigma_{\tilde{w}_n} = \sqrt{\frac{\sigma_v^2}{\sigma_x^2} \cdot \frac{\alpha}{N(2-\alpha)}}. \tag{4.34}$$

Naturally, a suitable convergence criterion should be determined according to this value. Define a real number $\tilde{\mu}$

$$\tilde{\mu} = \frac{1}{\sigma_{\tilde{w}_n}} = \sqrt{\frac{\sigma_x^2}{\sigma_v^2} \cdot \frac{N(2-\alpha)}{\alpha}}. \tag{4.35}$$

This result indicates that μ should be adjusted according to α, N and the signal-noise-ratio

(SNR) σ_x^2/σ_v^2. For a similar condition in [34], where $\sigma_x^2/\sigma_v^2 = 10^3$, $\alpha = 0.3$ and N= 512, μ should be 1703 according to (4.35). So $\mu = 1000$ is a relatively suitable choice for it. However, for a long filter in high SNR environment, a much greater μ is necessary. For example, if $\sigma_x^2/\sigma_v^2 = 10^4$, $N = 1024$ and $\alpha = 0.1$, a suitable value for μ should be 31190, which is greatly large than 1000. Many computer simulations in Section 4.7 illustrate how the choice of μ influences the convergence performance in different noise level.

In order to estimate SNR, it is necessary to know σ_x^2 and σ_v^2. The variance of the input signal σ_x^2 can be easily obtained using a exponential window as

$$\sigma_x^2(t) = \lambda\hat{\sigma}_x^2(t-1) + (1-\lambda)x^2(t), \tag{4.36}$$

where λ is a forgetting factor in a range between 0.99 and 0.9999 [2]. The power estimation of the disturbance signal, σ_v^2, can be obtained during the silences in practical application, such as in a network echo cancellation system.

4.5 Better Approximate of Segment Function

The MPNLMS algorithm is not applicable in a real-time implementation, since it additionally requires N logarithmic operations at every iteration. The SPNLMS algorithm demonstrates very fast convergence speed than PNLMS with moderate increase of computation cost. It is a good choice for practical application. However, the segment line function in (4.17) is a very course approximation of mu-law function. It only divides the whole function range into two sections: one for very small value and one for the others. Furthermore, this approximate is based on a specific μ, i.e., $\mu = 1000$. It can not satisfy the requirement of different μ. As discussed in the previous section, the suitable μ should be adjusted based on (4.35).

In this section, a new line function is proposed to approximate μ-law function in (4.31) as exactly as possible. The new segment line function is required to satisfy two requirements: 1) it should adjust the value according to suitable μ determined in previous section; 2) it should approximate (4.31) as exactly as possible. Furthermore, in order to reduce the computational cost of the line function, μ is choose as an integer of power of 2.

The proposed approach consists two steps. Firstly, a suitable μ is obtained based on (4.35), but as power of 2:

$$k = \lfloor \log_2 \tilde{\mu} \rfloor, \tag{4.37}$$

$$\mu = 2^k, \tag{4.38}$$

where $\lfloor \cdot \rfloor$ is the floor function. In this case, multiplying μ by $w_n(t)$ can be implemented digitally very efficiently by means of shift registers, therefore the cost of this multiplication can be ignored.

Secondly, a new segment line function is proposed to approximate $F(w_n(t))$ defined in (4.31). The proposed segment line function can be summarized as

$$\tilde{F}(x) = 2^{k+1-i}x + i - 2 + 2^{1-i}, \ (2^{i-1}-1)\epsilon \leq x < (2^i - 1)\epsilon \tag{4.39}$$

where $i = 1, 2, \cdots, k$. This expression can be expended as:

$$\tilde{F}(x) = \begin{cases} 2^k x, & 0 \leq x < \epsilon \\ 2^{k-1}x + 1/2, & \epsilon \leq x < 3\epsilon \\ 2^{k-2}x + 5/4, & 3\epsilon \leq x < 7\epsilon \\ \cdots & \\ 2x + k - 2 + 2^{1-k}, & 1/2 - \epsilon \leq x < 1 - \epsilon \end{cases} \tag{4.40}$$

The proposed segment line divides the range of coefficient magnitude into multiple segments, in stead of two segments as in (4.17), in order to approximate the corresponding logarithm function in (4.31). It has two advantages to replace the natural logarithmic operation by the logarithm of base 2. First, it is easy to find a unifying formula for a segment function for various μ. In addition, because the proposed μ has the form of 2^k, the calculation of $2^k x$ can

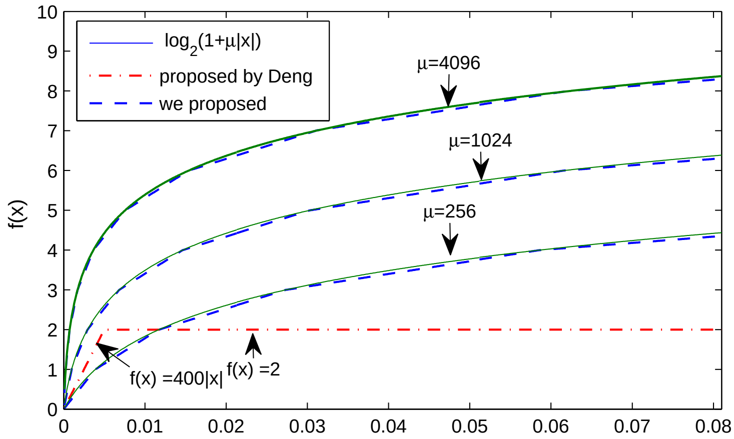

Fig. 4.2   Proposed line segment of different μ.

be simplified by shifting the index of x. This will reduce the computation cost of the SPNLMS algorithm. Figure 4.2 illustrates the proposed function $\tilde{F}(x)$ of different μ. It can be observed that the proposed segment line function can fit $\log_2(1+\mu|x|)$ of various μ better than the original definition in (4.17). It is believed that the new segment line function can retain fast convergence speed of MPNLMS. For convenience sake, the refined SPNLMS algorithm with this definition of $\tilde{F}(x)$ is referred to as the SPNLMS+ algorithm.

4.6   Computational Complexity and Its Reduction

Usually, computational complexity of adaptive algorithms is measured by the number of multiplications (or division). Because the regressors of LMS and NLMS have shift-structure, $||\mathbf{x}(t)||^2$ can be calculated recursively by subtracting off the square of the outgoing sample and adding in the square of the incoming sample, which consumes only 2 multiplications. Consequently, NLMS requires approximate $2N$ multiplications at every iteration: N multiplications for calculation of $\mathbf{w}^T(t)\mathbf{x}(t)$ and N multiplications for coefficient updating. Several methods are proposed in [67] to reduce the computational complexity of the PNLMS algorithms, i.e., updating only active coefficients, halting adaptation when small residual error is detected. Here,

Table 4.1 Computational Complexity of Proportionate Algorithms

Step No.	Operations	Multiplications	Additions
(1)	$e(t) = d(t) - \mathbf{w}^T(t)\mathbf{x}(t)$	N	N
(2)	$\gamma_n(t) = F(\lvert w_n(t)\rvert)$	–	–
(3)	$L_{\max} = \max\{\delta_\rho, \gamma_0(t), \cdots, \gamma_{N-1}(t)\}$	0	0
(4)	$\tilde{g}_n(t) = \max\{\gamma_n(t), \rho L_{\max}\}$	0	0
(5)*	$g_n(t) = \dfrac{\tilde{g}_n(t)}{\frac{1}{N}\sum_{i=0}^{N-1}\tilde{g}_i(t)}$	N	$N-1$
(6)	$\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}$	0	0
(7)	$\mathbf{x}_G(t) = \mathbf{G}(t)\mathbf{x}(t)$	N	0
(8)	$\lVert\mathbf{x}(t)\rVert_G^2 = \mathbf{x}^T(t)\mathbf{x}_G(t)$	N	$N-1$
(9)	$\mathbf{w}(t+1) = \mathbf{w}(t) + \dfrac{\alpha e(t)}{\lVert\mathbf{x}_G(t)\rVert_G^2 + \delta}\mathbf{x}_G(t)$	$N+2$	$N+1$
	Summation	$5N+3$	$4N+1$

additional proposals are discussed.

It is addressed that PNLMS algorithm increases the computational complexity about 50% compared to NLMS [8, 29]. The extra per-tap multiply is needed for calculatoin of $\mathbf{G}(t)\mathbf{x}(t)$, which requires N multiplications. Furthermore, because the proportionate NLMS algorithms generally use the form in (4.4), as a result, additional N multiplies are required to calculate the denominator $\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)$. In brief, for proportionate NLMS adaptation, $5N + 3$ multiplications are required per sample.

Table 4.1 lists the multiplications and additions of proportionate adaptive algorithms, where step (2) is dependent on the specific proportionate algorithms. For PNLMS, this step does not need any multiplication and addtion. For the SPNLMS algorithm, it requires at most additional N multiplies at every iteration to compute $\tilde{F}(|w_n(t)|)$. The proposed new segment line function based on logarithm to base 2 in (4.31) makes calculation of $\tilde{F}(|w_n(t)|)$ efficient by bit-shifting operation because both μ and ϵ are power of 2. Hence, the computational cost of $\tilde{F}(|w_n(t)|)$ could be ignored.

4.6.1 Removal of Normalization

In [8], it is stated that the normalization of $g_n(t)$ by its average is critical, where the dominator has a form of $||\mathbf{x}(t)||_2^2$ in stead of $||\mathbf{x}(t)||_G^2$. The reason is that it is required to removes any sensitivity of misalignment noise to the exact shape of $g_n(t)$.

Normalization of $g_n(t)$ by its average value, step (5) in Tab. 4.1, requires $N+1$ divisions and $N-1$ additions. Here, however, it is proposed to eliminate this step when the denominator is $||\mathbf{x}(t)||_G^2$ because it is not necessary for the proposed SPNLMS+ algorithm anymore.

Note that, when the denominator is $\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)$, any non-zero constant factor of $g_n(t)$ does not greatly change the behavior of the proportionate adaptive algorithms. Defining a step-size control matrix by dropping normalization, a new proportionate step-size matrix is obtained as:

$$\tilde{\mathbf{G}}(t) = \mathrm{diag}\{\tilde{g}_0(t), \tilde{g}_1(t), \cdots, \tilde{g}_{N-1}(t)\}. \tag{4.41}$$

The relation between the new and the original proportionate step size matrix is,

$$\bar{g}(t) = \frac{1}{N}\sum_{i=0}^{N-1} \tilde{g}_i(t), \tag{4.42}$$

$$\tilde{\mathbf{G}}(t) = \bar{g}(t)\mathbf{G}(t). \tag{4.43}$$

Replacing the new definition into the coefficient updating equation results in

$$\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)+\delta} = \frac{\tilde{\mathbf{G}}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\tilde{\mathbf{G}}(t)\mathbf{x}(t)+\bar{g}(t)\delta}. \tag{4.44}$$

This implies that, by dropping the normalization of $g_n(t)$, the coefficient updating equation with $\tilde{\mathbf{G}}(t)$ is equivalent to the original one, which can save N multiplies at every iteration. In stead, the regularization term δ need multiply $\bar{g}(t)$, which only needs 1 multiplication. For SPNLMS with a sparse impulse response, $\bar{g}(t)$ is in the range from 0.5 to 2.0 (see simulation results). Because δ is a small number (in order to avoid divergence when the power of input signal is small), consequently, replacing $\bar{g}(t)\delta$ with δ will not greatly affect the convergence performance.

This makes the calculation of $\bar{g}(t)$ unnecessary.

4.6.2 Partial Updating of $\tilde{\mathbf{G}}(t)$

The computational complexity of SPNLMS can be further reduced by calculating $\tilde{g}_{(}t)$ in every M iterations, instead of in every iteration, because $w_n(t)$ changes significantly only over large numbers of iterations. Using this method, the average computation cost for calculation of $\tilde{\mathbf{G}}(t)$ is reduced from $2N$ to $2N/M$. Simulation results show that even when M is a relatively large number, such as $M = N$, the proposed SPNLMS+ algorithm demonstrates approximately similar convergence with full update of $\tilde{\mathbf{G}}(t)$.

4.6.3 Selective Partial-Updating of coefficient vector $\mathbf{w}(t)$

Another technique to reduce the computational complexity is the selective partial updating, where only L coefficients are updated at every iteration ($L \leq N$). See [51] and [49] for the details and [58] for its application to proportionate NLMS algorithms. Generally, it saves $N - L$ multiplies in each iteration. However, this technique needs to calculate $\mathbf{G}(t)$ at every iteration and it requires to select the L largest coefficients to update, which costs additional sort operation at every iteration.

The proposed approach is summarized in Table 4.2.

4.7 Simulation Results

To evaluate the performance of the proposed approaches, many computer simulations were conducted with related algorithms. The common conditions for the simulations are as follows. The unknown impulse response $\mathbf{h}$ is a sparse impulse response illustrated in Fig. 1.3. The adaptive filter $\mathbf{w}$ has 512 coefficients ($N = 512$).

The relevant algorithms, including NLMS, PNLMS, MPNLMS, SPNLMS, and the pro-

Table 4.2 Summary of SPNLMS+ Algorithm

Initialization:
$\mathbf{w}(1) = \mathbf{0}$,
δ is a small number,
$\tilde{\mu} = \sqrt{\frac{\sigma_x^2}{\sigma_v^2} \frac{N(2-\alpha)}{\alpha}}$,
$k = \lfloor \log_2 \tilde{\mu} \rfloor$,
$u = 2^k$,
For all t:
$e(t) = d(t) - \mathbf{w}^T(t)\mathbf{x}(t)$
if mod $(t, M) == 0$
$g_n(t) = \tilde{F}(\lvert w_n(t) \rvert)$
$\tilde{g}_n(t) = \max \left\{ g_n(t), \rho \max\{\delta_\rho, g_0(t), \cdots, g_{N-1}(t)\} \right\}$
$\mathbf{G}(t) = \mathrm{diag}\,\{\tilde{g}_0(t), \tilde{g}_1(t), \cdots, \tilde{g}_{N-1}(t)\}$
$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha \frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \delta}$

posed SPNLMS+, are tested in various level of noise. The disturbance signal $v(t)$ are zero-mean white Gaussian signal that is uncorrelated to the input signal. Its magnitude is adjusted to get a certain SNR. The regularization parameter $\delta = 1.0$. The parameters of the proportionate algorithms use their typical value, $\delta_\rho = 0.01$, $\rho = 0.01$. For MPNLMS, $\mu = 1000$. The performance of the relevant algorithm is quantified using the normalized misalignment defined as $10 \log_{10}\{\|\mathbf{h} - \mathbf{w}(t)\|_2^2 / \|\mathbf{h}\|_2^2\}$ (in dB). The results illustrated in the following figures are the ensemble average of over 100 independent trials.

4.7.1 Influence of Different μ

Firstly, it examines the effectiveness of the proposed μ for MPNLMS. The input signal $\{x(t)\}$ and the disturbance $\{v(t)\}$, which is assumed to be independent from $\{x(t)\}$, are zero-mean Gaussian sequence. The step-size $\alpha = 0.25$.

Figure 4.3 compares the convergence of MPNLMS with different μ in three level of SNR,

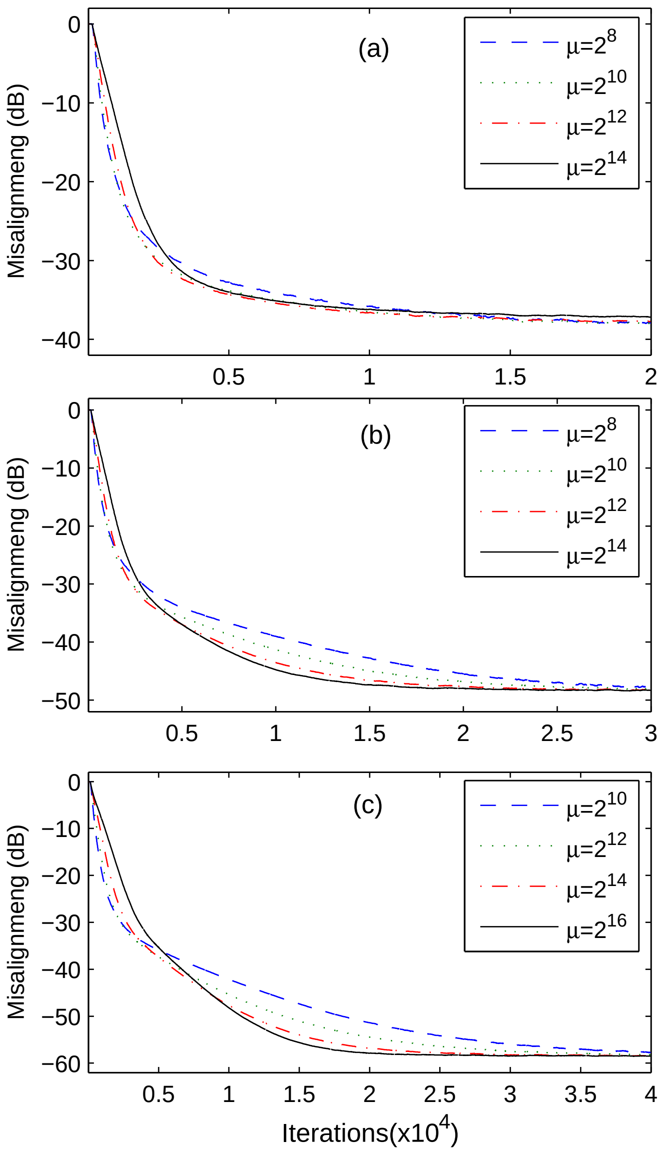

Fig. 4.3 Influence of μ on convergence speed of MPNLMS in various level of SNR. (a) SNR = 10^3, (b) SNR = 10^4, (c) SNR = 10^5.

where subplot (a) is for SNR = 10^3, subplot (b) is for SNR = 10^4, subplot (c) is for SNR = 10^5, respectively. The solid lines indicate the proposed μ in corresponding condition. When SNR is in the similar condition to that in [34], the proposed μ is 2^{10}, which approximately equals to that proposed in [34]. It can be seen from Fig. 4.3(a) that, although MPNLMS of different μ demonstrate almost similar convergence performance in this condition, the proposed one is best among these for candidates. Figure 4.3(b) and Fig. 4.3(c) illustrates the convergence with different μ when SNR is relatively high. It can be seen that the convergence of $\mu \leq 2^{10}$ has relatively faster initial convergence speed. However, their convergence begin slow after its normalized misalignment is less than 30dB. A large μ can improve the convergence thereafter, $\mu = 2^{12}$ for SNR $= 10^4$ and $\mu = 2^{14}$ for SNR $= 10^5$. In these condition, the proposed μ can achieve better over-all convergence than the others. With the increase of SNR, a bigger μ is necessary to drive the algorithm to reach its steady-state faster.

From these simulations, it can draw a conclusion that there is a trade-off in the choice of μ. A small μ, in the range between 2^8 and 2^{12}, can achieve faster convergence at the initial stage. However, a larger μ is preferred thereafter to quickly achieve its steady-state. It can also find that there is critical point in the convergence of MPNLMS algorithm. Before that point, a small μ can lead faster convergence but after that point a big μ can lead to faster convergence. The proposed approach can provide an acceptable μ to balance the fast initial convergence and the over-all convergence speed. In short, the proposed μ can achieve better performance than the others in these three conditions.

4.7.2 Convergence of Proposed SPNLMS+ Algorithm

The performance of the proposed SPNLMS+ algorithm is compared with three others, PNLMS, MPNLMS and original SPNLMS in three level of SNR, where subplot (a) is for SNR $= 10^3$, subplot (b) is for SNR $= 10^4$, subplot (c) is for SNR $= 10^5$, respectively. Figure 4.4 illustrates simulation results. It can be seen from Fig. 4.4(a) that SPNLMS+ shows similar convergence performance with the others when the noise level is relatively high. In this condition, the proposed segment line is based on $\mu = 2^{10}$, which is same to that of MPNLMS. In the other two conditions, the SPNLMS+ algorithm shows faster convergence than MPNLMS. In Fig. 4.4(b) and Fig. 4.4(c), the original SPNLMS demonstrates slow convergence speed than MPNLMS because its segment line function only provides a course approximation of corresponding mu-law function. The proposed SPNLMS+ algorithm, however, shows better conver-

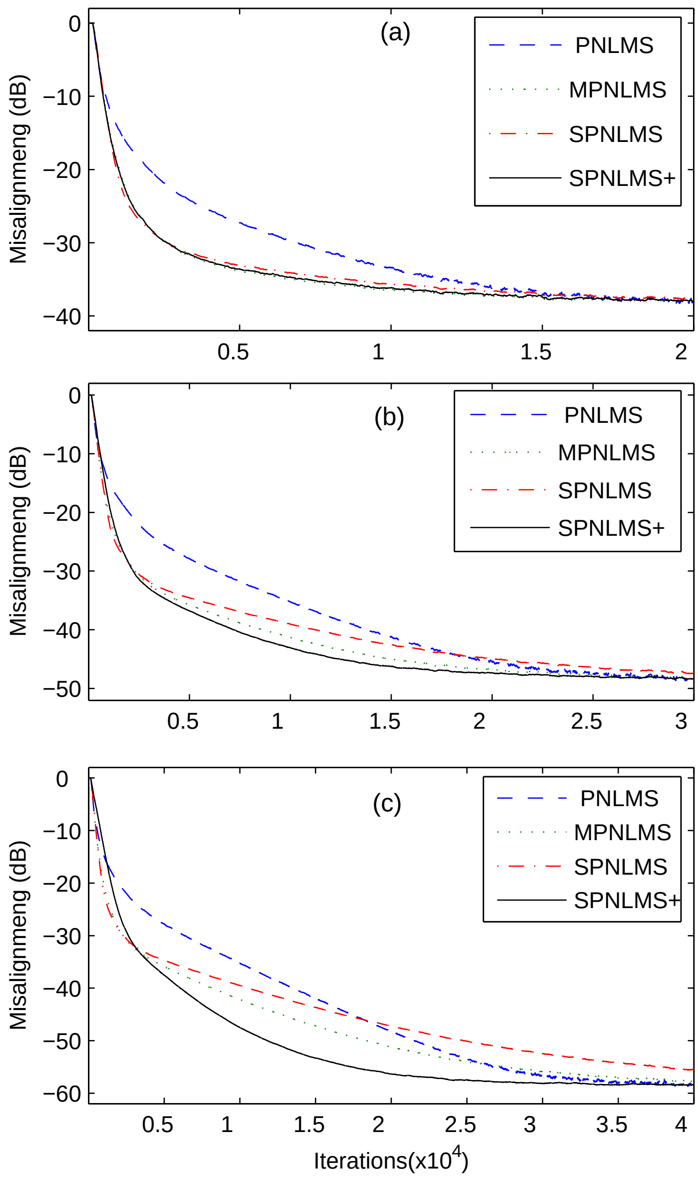

Fig. 4.4 Convergence comparison of SPNLMS+ in various level of SNR. (a) SNR = 10^3, (b) SNR = 10^4, (c) SNR = 10^5.

gence performance than SPNLMS. The reason is the two proposed approaches: a more suitable convergence criterion and the more precise line segment function. Its convergence is improved more in Fig. 4.4(c), where SPNLMS converges slowly after its fast initial convergence while the proposed SPNLMS+ presents faster over-all convergence.

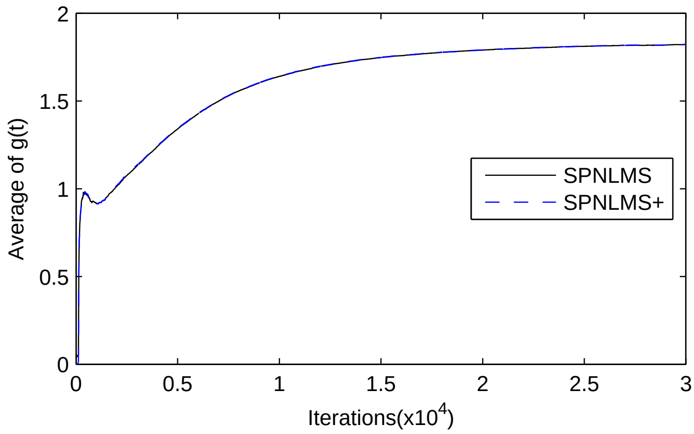

Fig. 4.5 Evolution of $\bar{g}(t)$ in Fig. 4.4(b).

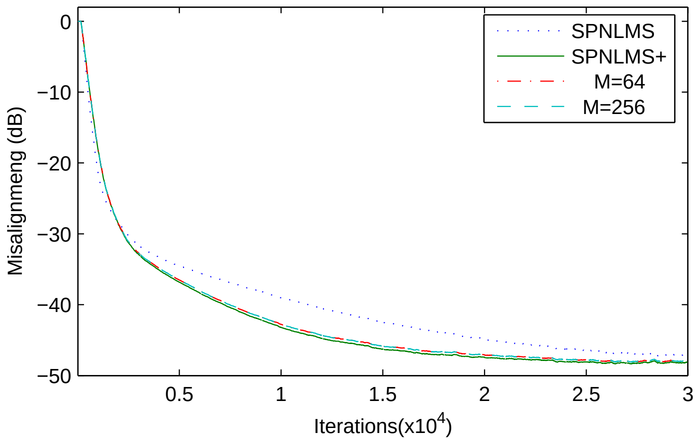

Fig. 4.6 Convergence of SPNLMS+ using computation reduction methods.

4.7.3 Effectiveness of Computation Reduction Methods

Two methods have been proposed to reduce the computational complexity of the SPNLMS+ algorithm. By dropping normalization of $g_n(t)$ by its average $\bar{g}(t)$, N multiplies are saved in every iteration. It further is proposed to drop calculation of $\bar{g}(t)$ because its value is in a small range around 1.0. Figure 4.5 shows the evolution of $\bar{g}(t)$ in the simulation of Fig. 6(b). It can be seen that, although $\bar{g}(t)$ of SPNLMS and SPNLMS+ is small in the first several iterations, its value maintains in a limited range between 0.5 and 2.0. As a result, the convergence of SPNLMS+ without normalization of $g_n(t)$ by $\bar{g}(t)$ is same to that with normalization.

It also is proposed to calculate $\tilde{\mathbf{G}}(t)$ once at every M iterations instead of at every iteration. The convergence of this method with that of full calculation method are compared in Fig. 4.6. The solid lines indicates the SPNLMS+ algorithm with full calculation. $M = 64$ and $M = 256$ indicate the SPNLMS+ algorithm that calculates $\tilde{\mathbf{G}}(t)$ every 64 and 256 iterations respectively. It can be seen that the proposed methods present as fast convergence as full calculation method, hence the computational complexity of SPNLMS+ is greatly reduced.

4.8 Conclusions

Two strategies are presented to improve the performance of the SPNLMS algorithm. A convergence criterion is established at first, then an approach is proposed to improve its convergence in various noise level environments. A refined segment function is proposed to approximate the mu-law function as exact as possible. The computational complexity is analyzed and two approaches are proposed to reduce the computational complexity. The performance improvement of the proposed algorithms is verified by numerous simulations.

Chapter 5

Variable Step-size Proportionate NLMS Algorithms

5.1 Introduction

The steady-state performance is another important aspect of an adaptive filter. Low misalignment is desired in practical application. Unfortunately, the design of the step-size control matrix $\mathbf{G}(t)$ of proportionate adaptive algorithms cannot improve the steady-state misalignment. Actually, the steady-state misalignment of the proportionate algorithms is controlled by global step-size parameter. The steady-state misalignment of proportionate adaptive algorithms is approximately equal to that of their non-proportionate counterpart having the same global step-size parameter [8]. When global step size is adjusted to obtain fast convergence, the steady-state misalignment becomes large, and vice versa. The step-size parameter reflects a trade-off between fast convergence and low steady-state misalignment. One solution to decrease the steady-state misalignment is variable step-size technique. If we adaptively control the step size parameter so that it remains large in the transient state of adaptive filter, fast convergence speed is achieved. If we decrease the step-size parameter to be small as the convergence proceeds, low steady-state misalignment can be obtained.

Different variable step-size approaches have been proposed in the literature. In [68], the squared instantaneous error was used as criteria to change the step size. In [69], an optimal step size was proposed for NLMS algorithm by adaptively minimizing the mean-square deviation at each iteration. In [70], a steepest descend method was proposed to update the step size to minimize the squared error. A non-parametric variable step-size NLMS algorithm was proposed in [71] by adjusting the step size to cancel the *a posteriori* error. Recently, this method was applied in the under-modeling acoustic echo cancellation system [72]. In [73], it was extended further to APA with a new perspective of signal enhancement. In [74], a criterion is proposed to measure how close the adaptive filter is to optimal value by maximizing the decrease of mean-square deviation at each iteration, and then it proposes an variable step-size approach based on this criterion. However, these approaches are only applicable for non-proportionate adaptive algorithms.

In this chapter, a variable step-size approach is proposed for proportionate NLMS algorithms. Because of the presence of a diagonal proportionate step-size matrix, it is complex to analyze the behavior of proportionate NLMS algorithms. It is well known that the principle of least perturbation plays an important role in derivation of the NLMS algorithm. Inspired by this, the principle is exploited into the proportionate LMS algorithm and obtain a derivation of proportionate NLMS algorithm. Note that, by canceling the *a posteriori* error at each iteration, theoretically, this principle can lead to perfect performance in a noise free context: fast convergence speed and zero misalignment. However, with the presence of background noise or disturbance signal, canceling the *a posteriori* error will introduce additional noise into the coefficient update. Taking into account this negative effect of the disturbance signals, the optimal global step size can be obtained by forcing the *a posteriori* error not to be zero, but to be the disturbance signal. Based on this idea, a variable step-size parameter is derived to cancel the pure *a posteriori* error at each iteration. The proposed approach is independent from the proportionate step-size control matrix.

5.2 Inspiration

The steady-state misalignments of proportionate adaptive algorithms are almost not affected by $\mathbf{G}(t)$, but controlled by global step size α. The steady-state misalignment are approximately equal to its non-proportionate version of same global step-size parameter [8]. When α is adjusted to obtain fast convergence, the misalignment becomes large, and vice versa.

It is difficult to analyze the behavior of the proportionate NLMS algorithms because presence of $\mathbf{G}(t)$ introduces nonlinearity, particularly with term $[\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t) + \epsilon]^{-1}$. Nevertheless, it is proved in [8] that its steady-state misalignment is independent with $\mathbf{G}(t)$ and approximately equal to NLMS of the same global step-size parameter. In order to decrease the steady-state misalignment of proportionate adaptive algorithms, it is necessary to develop a global variable step-size parameter.

A non-parametric variable step-size method was proposed in [71] for the NLMS algorithm in a simple and elegant way. Recently, this method is applied into IPNLMS in [75] and results in a VSS-IPNLMS algorithm. It first regards the proportionate step-size parameter of IPNLMS as two parts: the first term is NLMS-type step size and the second term is the proportionate step size. And then it combines the non-parametric variable step size $\alpha(t)$ into the NLMS-type step size, so that

$$g_n(t) = \alpha(t)\frac{1-\beta}{2N} + \frac{(1+\beta)|w_n(t)|}{2\sum_{i=0}^{N-1}|w_i(t)| + \delta}. \tag{5.1}$$

However, as discussed above, this method cannot decrease the steady-state misalignment by only designing a variable proportionate step-size parameter $g_n(t)$. In this section, based on the result in [75], an efficient variable global step-size parameter is obtained to improve the steady-state misalignment of proportionate adaptive algorithms. The resulting approach is a natural extension of the non-parametric variable step-size method to the proportionate adaptive algorithms rather than a simple combination.

The proportionate NLMS algorithms can be viewed as a normalized version of *proportion-*

ate LMS algorithm, which means the LMS algorithm with proportionate adaptation. Inspired by this fact, beginning with the proportionate LMS algorithm, a derivation of proportionate NLMS algorithm is obtained. At last, a variable global step size can be achieved in the proportionate adaptation framework. It is well known that the NLMS algorithm can be derived from the LMS algorithm using the principle of least perturbation, that is to maintain the next coefficient vector as close as possible to current estimate, while forcing the *a posteriori* error to be zeros [2, 3]. The *a posteriori* error $\bar{e}(t)$ is defined as

$$\begin{aligned}\bar{e}(t) &= d(t) - \mathbf{x}^T(t)\mathbf{w}(t+1) \\ &= \mathbf{x}^T(t)\tilde{\mathbf{w}}(t+1) + v(t),\end{aligned} \tag{5.2}$$

where $\tilde{\mathbf{w}}(t) = \mathbf{h} - \mathbf{w}(t)$ is the coefficient error vector.

Apply this principle to proportionate LMS algorithm, a derivation of proportionate NLMS algorithm can be achieved as bellow. Starting from the proportionate LMS algorithm, with a global variable step size parameter $\alpha(t)$,

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)\mathbf{G}(t)\mathbf{x}(t)e(t). \tag{5.3}$$

Subtract $\mathbf{h}$ at both sides and rearrange the terms

$$\tilde{\mathbf{w}}(t+1) = \tilde{\mathbf{w}}(t) - \alpha(t)\mathbf{G}(t)\mathbf{x}(t)e(t). \tag{5.4}$$

By premultiplying $\mathbf{x}^T(t)$ at both sides, a relation between the *a priori* error and the *a posterior* error is obtained as

$$\bar{e}(t) = \left[1 - \alpha(t)\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)\right]e(t). \tag{5.5}$$

In a noise free system, applying the principle of least perturbation to force $\bar{e}(t) = 0$, solving $\alpha(t)$ in (5.5), yields

$$\alpha(t) = \frac{1}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)} \tag{5.6}$$

Then replacing $\alpha(t)$ into (5.3), the normalized proportionate LMS algorithm of $\alpha = 1$ is obtained.

Both NLMS and proportionate NLMS can satisfy the principle of least perturbation in a noise free system. It has the fastest convergence speed and zero misalignment by canceling $\bar{e}(t)$ at each iteration. The optimal step size is 1 in this case. However, in practical applications, the disturbance signal $v(t)$ is inevitable. Consequently, adaptive algorithm cannot achieve zero misalignment. The reason could be explained by the fact that in the presence of $v(t)$, forcing the *a posteriori* error $\bar{e}(t)$ to be zero will force the adaptive filter to adapt the disturbance signal, which will introduce noise to the adaptation process. Actually, what is really wanted is to force the *pure a posteriori* error to be zero, i.e.

$$\mathbf{x}^T(t)\tilde{\mathbf{w}}(t+1) = 0. \tag{5.7}$$

By combining (5.7) with (5.2), it implies that in a noisy environment the coefficients should be updated to make the *a posteriori* error not to be zero, but to be the disturbance signal:

$$\bar{e}(t) = v(t). \tag{5.8}$$

In practical application, although the disturbance signal $v(t)$ is not available, its power level can be approximately estimated. For this reason the optimal step-size parameter can be found in such a way that

$$E\{\bar{e}^2(t)\} = E\{v^2(t)\}. \tag{5.9}$$

5.3 Proposed Variable Step-size

Based on this idea, a variable global step-size proportionate NLMS algorithm can be derived as following. Ignoring the regularization term ϵ and rewriting proportionate coefficient updating equation with a time-varying global step size $\alpha(t)$,

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)}. \tag{5.10}$$

Following the similar procedure as above, an expression between the *a priori* error and the *a posterior* error for proportionate NLMS algorithm is obtained as,

$$\bar{e}(t) = [1 - \alpha(t)]\, e(t). \tag{5.11}$$

In a noise free system, $v(t) = 0$, forcing $\bar{e}(t) = 0$, yields $\alpha(t) = 1$. This result coincides the previous analysis.

When $v(t) \neq 0$, squaring and taking mathematical expectation at both sides of (5.11), combining it with (5.9), solving $\alpha(t)$, the time-varying global step-size is obtained as

$$\alpha(t) = 1 - \sqrt{\frac{E\{v^2(t)\}}{E\{e^2(t)\}}}. \tag{5.12}$$

Note that this result has a similar formula with the VSS-IPNLMS algorithm. But it is applied to a global step-size parameter in stead of the NLMS part of the proportionate step size $g_n(t)$ of IPNLMS as in [75]. The simulation results in the next section show that this approach can greatly improve the steady-state misalignment of proportionate adaptive algorithms.

It can be found that the resulting variable global step-size parameter is independent of $\mathbf{G}(t)$. Although the proportionate algorithms have complicated behavior because of presence of $\mathbf{G}(t)$, this expression provides a simple, straightforward approach to control its global step-size parameter. In the transient state of the adaptive filter, $E\{e^2(t)\}$ will be large, hence $\alpha(t)$ is also large. As a consequence, fast convergence speed can be expected. After the adaptive filter reaches a certain degree convergence, $E\{e^2(t)\}$ becomes small, hence $\alpha(t)$ decreases. Consequently, low misalignment can be observed.

5.4 Practical Considerations

There are several practical considerations related to this expression. First, $E\{e^2(t)\}$ and $E\{v^2(t)\}$ must be estimated as $\hat{\sigma}_e^2(t)$ and $\hat{\sigma}_v^2(t)$, respectively. The quantity of $\hat{\sigma}_e^2(t)$ can be estimated using an exponential window as:

$$\hat{\sigma}_e^2(t) = \lambda\hat{\sigma}_e^2(t-1) + (1-\lambda)e^2(t), \tag{5.13}$$

where $\lambda = 1 - 1/(KN)$. A large K can obtain smooth estimate of $\sigma_e^2(t)$ but it will reduce the tracking ability of the adaptive filter. In order to avoid being divided by zero, $\hat{\sigma}_e^2(0)$ should be a small positive number. The power estimation of the disturbance signal, $\hat{\sigma}_v^2(t)$, can be obtained during the silences in a practical application, such as in a network echo cancellation system.

Second, in order to assure the algorithm stable, $\alpha(t)$ has to be limited in a suitable range. The estimate of $\hat{\sigma}_e^2(t)$ and $\hat{\sigma}_v^2(t)$ could slightly deviate from their theoretical values, which could result in a negative step size or large one to drive the adaptive algorithm to diverge. It is necessary to restrict $\alpha(t)$ in a range that the stability of the adaptive algorithm is guaranteed, $0 \leq \alpha_{\min} \leq \alpha(t) \leq \alpha_{\max} \leq 2$. Suitable choice of $\alpha_{\min}$ and $\alpha_{\max}$ can make the proposed algorithm robust to an inaccurate estimate of $\hat{\sigma}_v^2$. When $\alpha_{\min} = \alpha_{\max} = \alpha$, it degrades to the standard constant step-size proportionate NLMS algorithm.

Third, $\mathbf{G}(t)$ should be determined by a specific approach. It can be calculated using the MPNLMS algorithm described by MPNLMS. The related variable step-size algorithm is referred to as the VSS-MPNLMS algorithm. It is preferable to apply the segment proportionate version described by SPNLMS because it has excellent performance and light computation load. The resulting algorithm with this definition of $\mathbf{G}(t)$ will be referred to as the VSS-SPNLMS algorithm for convenience sake. The VSS-SPNLMS algorithm demonstrates very high convergence speed and low steady-state misalignment than the existing algorithms.

Another issue is related to the accuracy of $\sigma_v^2(t)$. The proposed algorithm can achieve optimal $\alpha(t)$ if an accurate estimate of $\sigma_v^2(t)$ is available. If it is inaccurate, convergence performance of the proposed algorithm will deteriorate. In the case of $\hat{\sigma}_v^2(t) \gg E\{v^2(t)\}$, $\alpha(t)$ will be smaller than the value defined in (5.12). The convergence speed of the proposed algorithm will be slowed down because $\alpha(t)$ reaches $\alpha_{\min}$ soon. However the low misalignment can be achieved after more iterations. In the case of $\hat{\sigma}_v^2(t) \ll E\{v^2(t)\}$, $\alpha(t)$ will be greater than its optimal value. The initial fast convergence speed will be observed but the misalignment will be increased because $\alpha(t)$ is close to $\alpha_{\max}$. For a modest inaccurate estimate of $\hat{\sigma}_v^2(t)$, it is believed that the performance of the proposed algorithm behaves better than that of the NPVSS-NLMS because it benefits from the fast proportionate adaptive algorithms. The simulation results in Section 4 show that the proposed algorithm can tolerate a large range of estimate error of $\hat{\sigma}_v^2(t)$.

The proposed variable step-size segment proportionate NLMS algorithm (VSS-SPNLMS) is summaried in Table 5.1.

5.5 Computational Complexity

The additional computational amount of the proposed VSS-SPNLMS is composed of four parts compared to the standard NLMS algorithm. First, the calculation of $\mathbf{G}(t)$ costs at most $(2N{+}2)$ multiplications, N additions and $2N + 1$ comparisons. Second, $\mathbf{x}_g(t) = \mathbf{G}(t)\mathbf{x}(t)$ costs N multiplications and $\mathbf{x}^T(t)\mathbf{x}_g(t)$ costs N multiplications. Third, the estimate of $\hat{\sigma}_e^2(t)$ and calculation of $\alpha(t)$ will cost 4 multiplications, 1 square-root operation, and 4 additions. The remaining operations are common with the NLMS algorithm, i.e. calculation of $e(t)$ (N multiplications) and coefficients update (N multiplications). In summary, compared to the NLMS algorithm, the dominant additional computation cost of the proposed VSS-SPNLMS algorithm is $(4N{+}4)$ multiplications and 1 square-root operation. Compared to the SPNLMS algorithm, the additional computation cost is only 4 multiplications and 1 square-root operation. For practical application, such as network echo cancellation, the computational complexity of the proposed

Table 5.1 The proposed VSS-SPNLMS algorithm

Initialization:

$\mathbf{w}(0) = \mathbf{0}$; $\hat{\sigma}_e^2(0) = 0.01$;

$\epsilon = M\sigma_x^2$, $(M \in Z^+, M \geq 1)$

$\lambda = 1 - 1/(KN), (K \in Z^+,\ K \geq 1)$

$0 \leq \alpha_{\min} < \alpha_{\max} \leq 1$

For all t:

$$\bar{g}_n(t) = \begin{cases} 400|w_n(t)|, & |w_n(t)| < 0.005 \\ 2, & \text{otherwise.} \end{cases}$$

$\gamma_n(t) = \max\{\bar{g}_n(t), \rho \max\{\delta_\rho, \bar{g}_0(t), \cdots, \bar{g}_{N-1}(t)\}\}$

$g_n(t) = \frac{\gamma_n(t)}{\frac{1}{N}\sum_{i=0}^{N-1}\gamma_i(t)}$

$\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}$

$e(t) = d(t) - \mathbf{x}^T(t)\mathbf{w}(t)$

$\hat{\sigma}_e^2(t) = (1-\lambda)\hat{\sigma}_e^2(t-1) + \lambda e^2(t)$

$\alpha(t) = 1 - \sqrt{\hat{\sigma}_v^2(t)/\hat{\sigma}_e^2(t)}$

if $\alpha(t) < \alpha_{\min}$, $\quad \alpha(t) = \alpha_{\min}$,

if $\alpha(t) > \alpha_{\max}$, $\quad \alpha(t) = \alpha_{\max}$,

$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha(t)\frac{\mathbf{G}(t)\mathbf{x}(t)e(t)}{\mathbf{x}^T(t)\mathbf{G}(t)\mathbf{x}(t)+\delta}$

Table 5.2 Comparison of Computational Complexity

	$\times/\div$	$+/-$	$\sqrt{\ }$	Comparison
NLMS	2N+4	2N+1	0	0
PNLMS	5N+3	4N+1	0	N
SPNLMS	6N+4	4N+1	0	2N+1
NPVSS-NLMS	2N+8	2N+3	1	1
VSS-SPNLMS	6N+8	4N+3	1	2N+3

algorithm is moderate. The computational complexity of the related algorithms are compared in Table 5.2.

Note that the computational complexity can be further reduced using selective partial updating algorithms [60], where only a part of coefficients are updated at every iteration. It can save $(N - M)$ multiplications, where M is the number of updated coefficients.

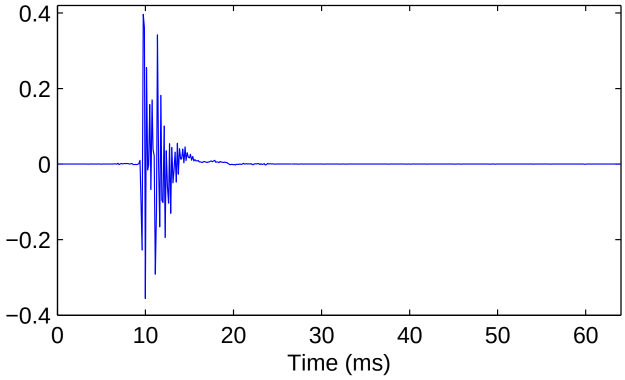

Fig. 5.1 A typical sparse network echo path.

5.6 Simulation Results

To evaluate the performance of the proposed algorithm, many computer simulations were conducted in the context of system identification. The unknown system $\mathbf{h}$ is from a sparse network echo path illustrated in Fig. 5.1. The adaptive filter $\mathbf{w}$ have the same number of coefficients with $\mathbf{h}$, $N = 512$. The related algorithms are tested in numerous simulations, NLMS, MPNLMS, SPNLMS, NPVSS-NLMS [71], and the proposed VSS-MPNLMS and VSS-SPNLMS. Three kinds of performance are compared: convergence speed, steady-state misalignment and tracking ability, using different input signals. The disturbance signal $v(t)$ are zero-mean white Gaussian signal that is uncorrelated to the input signal. Its magnitude is adjusted to get a certain signal-to-noise ratio (SNR) in dB. The regularization parameter $\epsilon = 10\sigma_x^2$. For the proportionate algorithms, $\delta_\rho = 0.01$, $\rho = 2/N$. Three kinds of signals are input into the adaptive algorithms: (1) white Gaussian signal, (2) highly colored signals and (3) speech signals. The performance is evaluated using the normalized misalignment (in dB) defined by $10\log_{10}\|\tilde{\mathbf{w}}(t)\|_2^2/\|\,\mathbf{h}\,\|_2^2$. Two scenarios are assumed in the following simulations: (1) real value of $\hat{\sigma}_v^2(t)$ is known, (2) $\hat{\sigma}_v^2(t)$ is inaccurate.

Table 5.3 Parameters for different input signals

Parameters	White input	Colored input	Speech input
$\alpha_{\min}$	0.008	0.008	0.05
$\alpha_{\max}$	1.0	1.0	1.0
K	1	2	4
ϵ	σ_x^2	$5\sigma_x^2$	$10\sigma_x^2$

5.6.1 Simulations using accurate estimate of $\sigma_v^2(t)$

Firstly, performance of the proposed approach is compared with the others using real value of $\sigma_v^2(t)$. In the 1st set of simulations, the input signal is zero-mean white Gaussian signal with $\sigma_x^2 = 1$, and $v(t)$ is adjusted to get 15dB SNR. Figure 5.2 shows the result: subplot (a) compares convergence speed in the first 10^4 iterations of the related algorithms and subplot (b) illustrates the steady-state misalignment of the related algorithms and their tracking abilities. It can be observed that the SPNLMS algorithm converges faster than the conventional NLMS algorithm with same step size. Also, they have almost the same steady-state misalignment. The proposed VSS-MPNLMS and VSS-SPNLMS present almost same fast initial convergence speed, compared with the SPNLMS ($\alpha = 1$) algorithm. However, they can obtain lower steady-state misalignment; about 18dB improvement can be observed in 3×10^4 iterations, as shown in Fig. 5.2(b). To achieve this low level misalignment, the SPNLMS requires a very small step size. In this case, $\alpha = 0.008$, its convergence speed is greatly degraded. Although the NPVSS-NLMS algorithm can achieve this low level misalignment, its convergence speed is obviously slower than the proposed algorithms. It can be seen from Fig. 5.2(a) that the proposed VSS-MPNLMS algorithm reaches -15dB misalignment in approximate 500 iterations but the NPVSS-NLMS algorithm reaches this level in around 1200 iterations. It can be also observed that the VSS-SPNLMS algorithm presents almost same performance with the VSS-MPNLMS algorithm, even though it is only a simplified variant of the VSS-MPNLMS algorithm.

The tracking ability of adaptive algorithm is important in a non-stationary environment

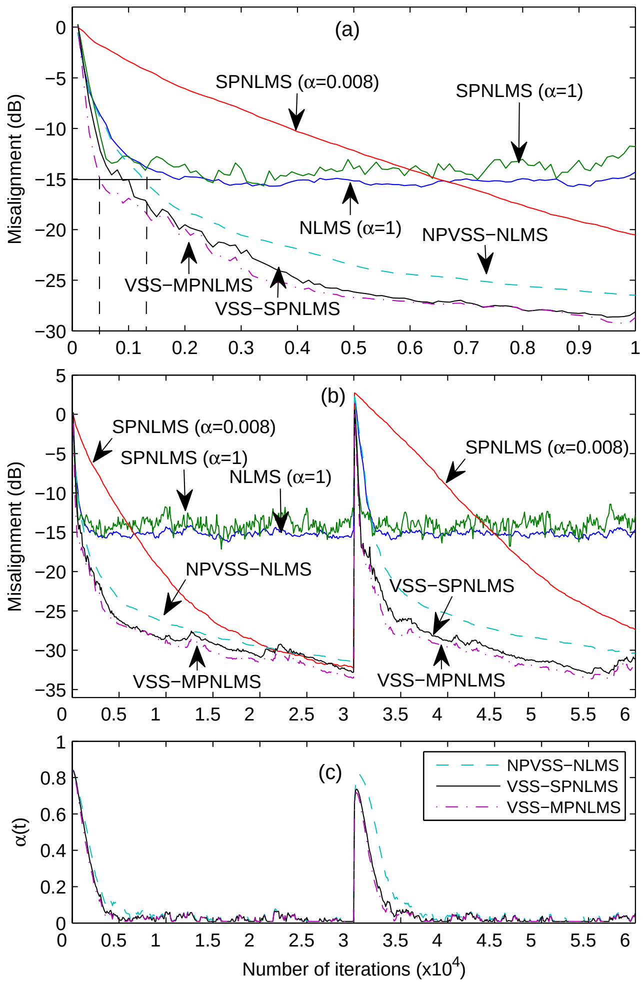

Fig. 5.2 Comparison of VSS-SPNLMS to the related algorithms using white Gaussian signal. SNR=15dB. (a) Convergence speed of the first 10^4 iterations. (b) Comparison of steady-state misalignment and tracking ability. (c) Evolution of $\alpha(t)$.

where the unknown impulse response may suddenly change. For a network echo canceler, for example, the echo path is apt to shift backward or forward as a result of the delay jitter. The tracking ability of relevant algorithms is also illustrated in Fig. 5.2(b). The unknown impulse response was suddenly shifted to the right by 12 samples when $t = 3 \times 10^4$. It can be seen from this figure that the proposed two algorithms present good tracking performance when an unpredicted change of unknown impulse response happens. Furthermore, they outperform the counterparts in low steady-state misalignment after reconvergence. Evolution of $\alpha(t)$ of three variable step-size algorithms is shown in Fig. 5.2(c). It can be observed that the variable step size $\alpha(t)$ of the three VSS algorithms present almost same trend: large enough for fast convergence and small enough for low steady-state misalignment. Once the unknown system changes, $\alpha(t)$ can quickly detect this change and increase to a large value in order to reconverge as fast as possible. Although their $\alpha(t)$ present same trend, both VSS-MPNLMS and VSS-SPNLMS show faster convergence speed than the non-proportionate algorithms. It can be explained by the fact that the proportionate algorithms benefit from the information of sparse structure while the conventional algorithms not.

In the 2nd set of simulations, the input signal is highly colored signals generated by filtering a zero-mean, unit variance, white Gaussian signal through a first-order system $G(z) = 1/(1 - 0.9z^{-1})$. The magnitude of disturbance signals is adjusted to get 30dB SNR. Figure 5.3 illustrates the result: subplot (a) compares convergence speed of the first 10^4 iterations and subplot (b) compares their steady-state misalignments and tracking abilities. It can be observed from Fig. 5.3(a) that the SPNLMS ($\alpha = 1.0$) algorithm can achieve fast initial convergence speed over the conventional NLMS algorithm, even in the case of this highly colored input signal. However, its convergence becomes slow thereafter. Furthermore, in this case, the steady-sate misalignment of the SPNLMS algorithm is greater than that of the NLMS algorithm, approximately -22dB vs -32dB. The proposed VSS-MPNLMS algorithm and the VSS-SNLMS algorithm achieve almost same fast initial convergence speed with SPNLMS ($\alpha = 1.0$), but it can reach lower steady-state misalignment. Comparing to the SPNLMS ($\alpha = 1.0$) algorithm, about 18dB improvement can be observed in 10^5 iterations. In order to decrease its steady-state, a small step-size is necessary for the SPNLMS algorithm, $\alpha = 0.008$ in this case, whose convergence speed is greatly slowed down. In this case, the NPVSS-NLMS algorithm failed. Not only its convergence speed is slower than the conventional NLMS ($\alpha = 1.0$) algorithm, but also its steady-state misalignment can not be improved. Figure 5.3(b) illustrates their tracking abilities.

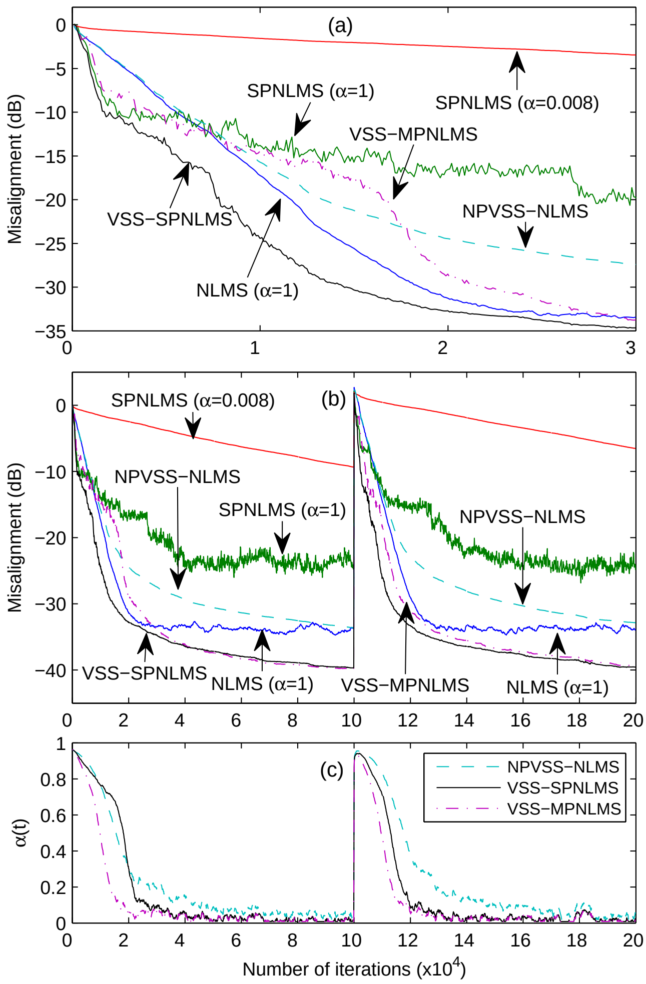

Fig. 5.3 Comparison of VSS-SPNLMS to the related algorithms using highly colored signal. SNR=30dB. (a) Convergence speed of the first 10^4 iterations. (b) Comparison of steady-state misalignment and tracking ability. (c) Evolution of $\alpha(t)$.

The unknown impulse response was suddenly shifted to the right by 12 samples when $t = 10^5$. It can be seen that the proposed two algorithms present good tracking performance. Evolution of $\alpha(t)$ of three variable step-size algorithms is illustrated in Fig. 5.2(c). It can be observed that the variable step size $\alpha(t)$ of the three VSS algorithms present approximately same trend.

In the 3rd set of simulations, the input is from a speech of 8k sampling rate. The disturbance signal is added to get 30dB SNR. The unknown impulse response was suddenly shifted

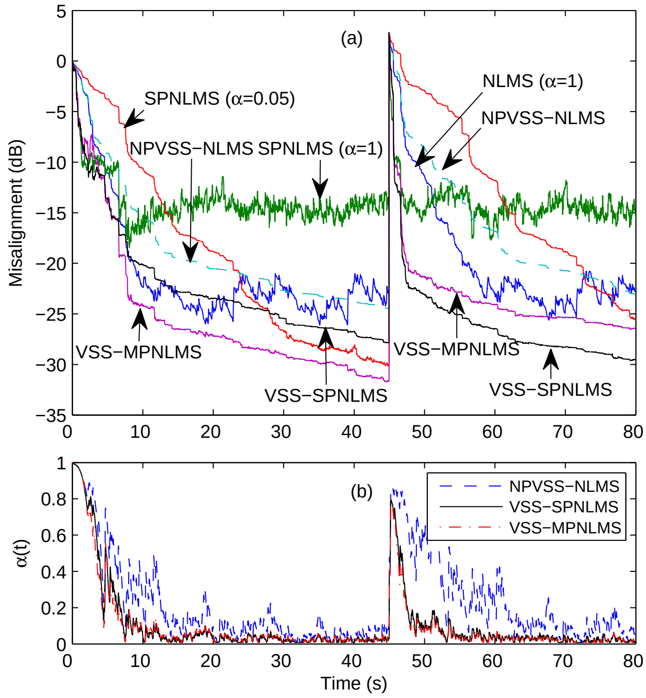

Fig. 5.4 Comparison of VSS-SPNLMS to the related algorithms using speech signal. SNR=30dB. (a) Comparison of misalignment and tracking ability. (b) Evolution of $\alpha(t)$

to the right by 12 samples after time is 45 second. Figure 5.4 illustrates the result: subplot (a) compares their convergence speed, steady-state misalignment and tracking ability, and subplot (b) shows $\alpha(t)$ of three VSS algorithms. It can be observed that the proposed VSS-MPNLMS and VSS-SPNLMS present excellent performance in the concern performance criteria: fast convergence speed, low steady-state misalignment, and tracking ability. In this case, the NPVSS-NLMS algorithm is outperformed.

It should note that the convergence performance of the proposed VSS-MPNLMS and VSS-SPNLMS algorithm present more smooth and consistent performance after the unknown impulse response suddenly change, while their performance before this change are lack of this kind of smoothness and consistency. This phenomenon can be observed in Fig. 5.2 - Fig. 5.4. It is because of the initialization method: $\mathbf{w}(0)$ is $\mathbf{0}$, $\hat{\sigma}_e^2(0)$ is a small value that influences optimal $\alpha(t)$ of the first hundreds iterations. In practical application, once the algorithm is started, it works in tracking mode unless the algorithm is forcedly reset. So the excellent tracking performance of the proposed algorithms are preferable in practical application. Especially the VSS-SPNLMS algorithm even behaves better than the VSS-MPNLMS algorithm for the highly colored input signals and speech signals, while its computational complexity is modest in real-time application.

5.6.2 Simulations with inaccurate estimate of $\sigma_v^2(t)$

As discussed in the previous section, the estimate accuracy of $\hat{\sigma}_v^2(t)$ influences the performance of variable step-size algorithms. In this subsection, the performance of the proposed VSS-SPNLMS algorithms is tested in three cases: (1) the estimate is exactly accurate $\hat{\sigma}_v^2(t) = E\{v^2(t)\}$; (2) the estimate is smaller than its real value, $\hat{\sigma}_v^2(t) = 0.8E\{v^2(t)\}$; and (3) the estimate is greater than its real value, $\hat{\sigma}_v^2(t) = 1.5E\{v^2(t)\}$. The NPVSS-NLMS is also illustrated for comparison, but it use an accurate $\hat{\sigma}_v^2(t)$.

The simulation results using white Gaussian signals are illustrated in Fig. 5.5. It can be seen that the convergence performance with an accurate $\sigma_v^2(t)$ is the best among these three cases. In case (2), a small $\sigma_v^2(t)$ causes the steady-state misalignment larger than in case (1). The reason is, when $\hat{\sigma}_v^2(t)$ is excessively smaller than its real value, $\alpha(t)$ remains large in the steady state - around 0.25 in Fig. 5.5(b). In case (3), a large $\hat{\sigma}_v^2(t)$ will cause $\alpha(t)$ small and it reaches $\alpha_{\min}$ faster than in case (1), as shown in Fig. 5.5(b). In this case, $\alpha_{\min}$ warrants VSS-SPNLMS to converge at relatively high speed and to achieve a low level misalignment, although it requires more iterations. In brief, in any case, VSS-SPNLMS can maintain fast convergence speed and achieve low steady-state misalignment. It does not behave worse than SPNLMS even with a large estimate error of $\hat{\sigma}_v^2(t)$. In the simulations, it can tolerate +50% and -20% estimate error of $\hat{\sigma}_v^2(t)$. So the proposed VSS-SPNLMS is robust to a relatively inaccurate estimate of $\hat{\sigma}_v^2(t)$.

The simulation results with inaccurate estimate of $\hat{\sigma}_v^2(t)$ using highly colored AR(1) signals

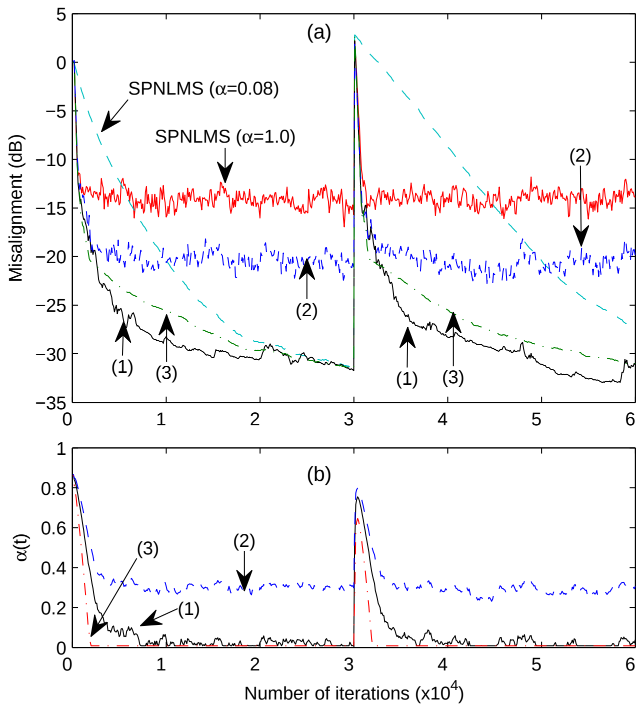

Fig. 5.5 Comparison of VSS-SPNLMS to the related algorithms with inaccurate estimate of $\hat{\sigma}_v^2(t)$ using white Gaussian signal. SNR=15dB. (a) Comparison of misalignment and tracking ability. (b) Evolution of $\alpha(t)$.

are illustrated in Fig. 5.6. In this condition, the VSS-SPNLMS algorithm with accurate estimate of $\hat{\sigma}_v^2(t)$ demonstrates best convergence performance. It can be seen from subplot (b) that with the convergence of the algorithm, $\alpha(t)$ decreases so that the steady-state misalignment achieves a low level. In case (2), a small $\sigma_v^2(t)$ causes $\alpha(t)$ remains large in the steady state - around 0.4 so its misalignment is relatively large but its convergence is fast. In case (3), although $\alpha(t)$ becomes $\alpha_{\min}$ soon, the proposed VSS-SPNLMS even shows better performance than NPVSS-NLMS.

The simulation results with inaccurate estimate of $\hat{\sigma}_v^2(t)$ using speech signals are illustrated in Fig. 5.7. It can be seen that for speech signals, the proposed algorithm demonstrates very fast initial convergence speed compared to NPVSS-NLMS. It can draw the same conclusion that VSS-SPNLMS with an accurate $\sigma_v^2(t)$ presents best performance. If $\sigma_v^2(t)$ is smaller than its real value, $\alpha(t)$ is larger than in case (1). If $\hat{\sigma}_v^2(t)$ is larger than its real value, $\alpha(t)$ will be

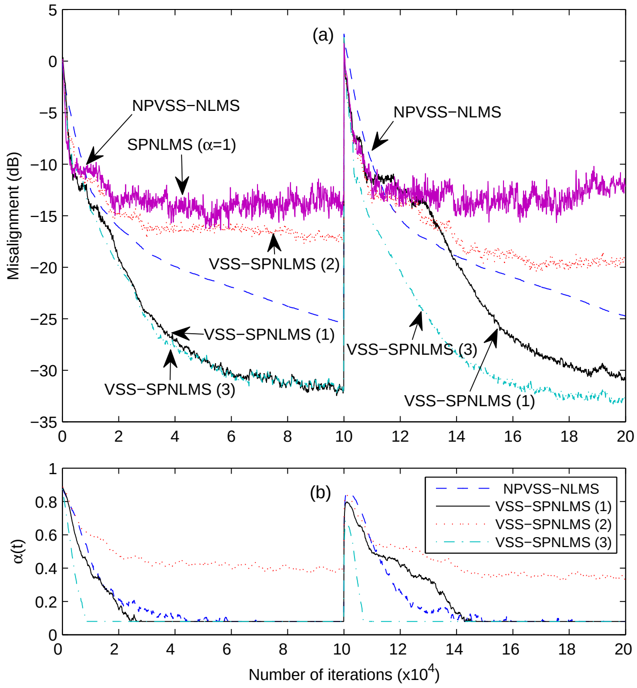

Fig. 5.6 Comparison of VSS-SPNLMS to the related algorithms with inaccurate estimate of $\hat{\sigma}_v^2(t)$ using highly colored signals. SNR=30dB. (a) Comparison of misalignment and tracking ability. (b) Evolution of $\alpha(t)$.

small and it reaches $\alpha_{\min}$ soon. A reasonable $\alpha_{\min}$ warrants VSS-SPNLMS to demonstrate good performance.

5.6.3 Comparison with other proportionate algorithms

In this subsection, the proposed VSS-SPNLMS algorithm is compared with three proportionate algorithms in [59, 62, 75]. In [59], O. Hoshuyama et al. proposed an approach to determine $\mathbf{G}(t)$ based on the gradient approximated by the difference between the current coefficient and an averaged filter coefficient with delay. In [62], S. Werner et al. combined proportionate adaptation with the set-membership filtering, where $\alpha(t)$ is replaced by

$$\alpha_S(t) = \begin{cases} 1 - \gamma/|e(t)|, & \text{if } |e(t)| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

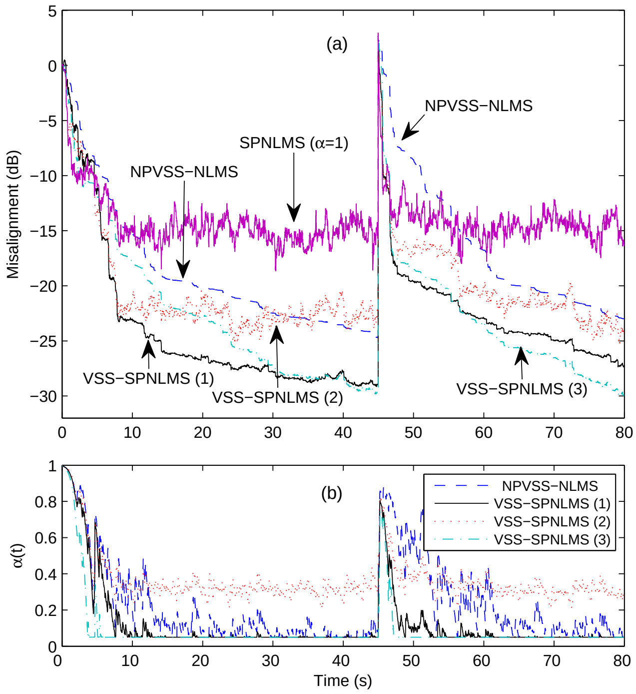

Fig. 5.7 Comparison of VSS-SPNLMS to the related algorithms with inaccurate estimate of $\hat{\sigma}_v^2(t)$ using speech signals. SNR=30dB. (a) Comparison of misalignment and tracking ability. (b) Evolution of $\alpha(t)$.

Here, γ is a predetermined parameter as a bound on the noise and generally $\gamma = \sqrt{2}\sigma_v$ can get good performance [62].

Their convergence and tracking ability using white Gaussian signal when SNR=30dB are shown in Fig. 5.8. The NPVSS-NLMS algorithm is also illustrated for comparison sake. It can be seen that the proposed VSS-SPNLMS algorithm demonstrates the best performance, in the convergence speed, steady-state misalignment and tracking. The VSS-IPNLMS algorithm in [75] presents a relatively higher convergence speed and tracking than the NPVSS-NLMS algorithm. However, its steady-state misalignment did not decrease much. Although the methods in [59] and [62] demonstrate almost the same initial convergence speed with VSS-SPNLMS, their steady-state misalignment cannot achieve the same low level as VSS-SPNLMS. The main disadvantage of the method in [59] is its computational complexity. It requires $4N$ multiplications

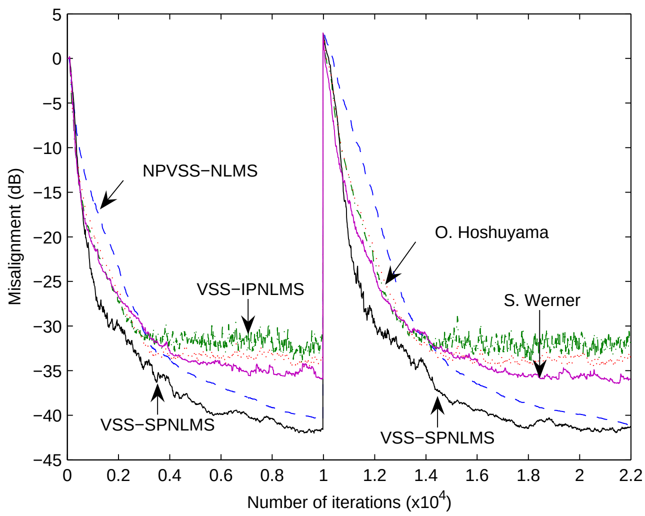

Fig. 5.8 Comparison of VSS-SPNLMS to the other counterparts. SNR=30dB. For VSS-IPNLMS, constant step-size parameter $\alpha = 1.0$, $\beta = -0.5$. For Gordy, constant step-size parameter $\alpha = 1.0$. For SM-PNLMS, $\gamma = \sqrt{2}\sigma_v$.

to calculate $\mathbf{G}(t)$ in each iteration while VSS-SPNLMS only requires at most $2N$ multiplications. Furthermore, it has many parameters that is difficult to choose in practical application. The main advantage of the method in [62] is its relatively low computational complexity because it halts the adaptation process when $|e(t)| \leq \gamma$. However, as to the convergence speed and steady-state misalignment, it cannot behave better than VSS-SPNLMS as shown in Fig. 5.8. Because it uses $|e(t)|$, instead of $\sigma_e(t)$ as in the proposed approach, and $|e(t)|$ demonstrates stochastic property in steady state, so α_s changes randomly and the steady-state misalignment cannot be improved greatly.

5.7 Conclusions

A method is proposed to combine the variable step-size approach into the proportionate NLMS algorithms. The proposed approach provides both fast convergence speed and low steady-state misalignment. Simulation results have shown that the proposed VSS-SPNLMS algorithm outperforms the classical adaptive algorithm for sparse impulse response. It has very fast convergence speed, tracking performance and low misalignment at the same time. Its additional computation load is very little. The main advantage of this algorithm is, it only requires an power estimate of the disturbance signals, $\hat{\sigma}_v^2(t)$, which can be easily obtained in practical application. Furthermore, it can tolerate a relatively large estimate error of $\hat{\sigma}_v^2(t)$. This makes it very preferable in practical application.

Chapter 6

Variable Step-size Proportionate Affine Projection Algorithms

6.1 Introduction

The optimal step-size control matrix $\mathbf{G}(t)$ of the MPNLMS algorithm was derived on the assumption that the input is white Gaussian signals. For highly correlated input signals, especially speech, convergence speed of proportionate NLMS algorithms will depend on the eigenvalues spread of the input signal's autocorrelation matrix $\mathbf{R}$. The proportionate affine projection algorithm (PAPA) is the natural extension of the PNLMS algorithm. It is expected to present faster convergence for highly correlated input signals at the cost of a modest increase in computational complexity. In this chapter, the previous variable step-size approach is extended to the PAPAs for sparse impulse response identification.

6.2 Proportionate Affine Projection Algorithm

The APA achieves a faster convergence speed for correlated input signals than the NLMS algorithm with only a modest increase in computational complexity. It exploits more information from the input signal, not only the current input vector but also the most recent P input vectors. The proportionate APA (PAPA) is expected to converge faster than the proportionate NLMS algorithms for colored input signals.

The PAPA can be briefly summarized as follows:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \alpha\mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(t), \tag{6.1}$$

where α is a global constant step size, δ is the regularization parameter and $\mathbf{I}$ is a $P \times P$ identity matrix.

The PAPA is a solution of a constrained optimization problem in a way similar to that of APA. That is to minimize the coefficient vector update in successive time index, subject to the P constraints that the *a posteriori* error is zero [2]:

$$\mathbf{w}(t+1) = \min_{\mathbf{w}(t+1)}\left\{[\mathbf{w}(t+1) - \mathbf{w}(t)]^T\mathbf{G}^{-1}(t)[\mathbf{w}(t+1) - \mathbf{w}(t)]\right\}, \tag{6.2}$$

$$\text{subject to} \quad \mathbf{X}^T(t)\mathbf{w}(t+1) = \mathbf{d}(t). \tag{6.3}$$

This problem can be solved by using the method of Lagrange multipliers [2], whose cost function to minimize is

$$\mathcal{J}(t) = [\mathbf{w}(t+1) - \mathbf{w}(t)]^T\,\mathbf{G}^{-1}(t)\,[\mathbf{w}(t+1) - \mathbf{w}(t)] + 2\boldsymbol{\lambda}^T\left[\mathbf{d}(t) - \mathbf{X}^T(t)\mathbf{w}(t+1)\right], \tag{6.4}$$

where $\boldsymbol{\lambda} = [\lambda_0\ \lambda_1\ \cdots\ \lambda_{P-1}]^T$ is a Lagrange multiplier vector. Setting $\dfrac{\partial\mathcal{J}(t)}{\partial\mathbf{w}(t+1)} = \mathbf{0}$ and $\dfrac{\partial\mathcal{J}(t)}{\partial\lambda_p} = 0$, $(p = 0\cdots P-1)$, it gives [Refer to Appendix 1.2]

$$\mathbf{G}^{-1}(t)\,[\mathbf{w}(t+1) - \mathbf{w}(t)] = \mathbf{X}(t)\boldsymbol{\lambda}, \tag{6.5}$$

$$\mathbf{X}^T(t)\mathbf{w}(t+1) = \mathbf{d}(t). \tag{6.6}$$

Pre-multiplying both sides of (6.5) by $\mathbf{G}(t)$, and then rearranging it,

$$\mathbf{w}(t+1) - \mathbf{w}(t) = \mathbf{G}(t)\mathbf{X}(t)\boldsymbol{\lambda}, \tag{6.7}$$

and then pre-multiplying both sides of (6.7) by $\mathbf{X}^T(t)$, combining with (6.6), remembering $\mathbf{e}(t) =$

$\mathbf{d}(t) - \mathbf{X}^T(t)\mathbf{w}(t)$, yields a relation between the *a priori* error and input matrix:

$$\mathbf{e}(t) = \mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t)\boldsymbol{\lambda}. \tag{6.8}$$

Solving $\boldsymbol{\lambda}$ from this equation

$$\boldsymbol{\lambda} = \left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t)\right]^{-1} \mathbf{e}(t), \tag{6.9}$$

and substituting into (6.7), introducing a global constant step-size parameter α and a regularization parameter δ, yields the resulting proportionate APA as in (6.1).

The step-size control matrix defined by MPNLMS was derived on the assumption that the input is white. The mu-law PAPA (MPAPA) is expected to achieve faster convergence speed than MPNLMS for colored input signals. Its computation efficient version, SPAPA, is favorable for real-world application because of its implementable low computational complexity.

6.3 Formulation of VSS Proportionate Affine Projection Algorithms

The objective is to find a variable step-size approach that is applicable to PAPAs. Unfortunately, because of the presence of $\mathbf{G}(t)$, it is very difficult to analyze the transient performance of PAPAs. In this section, a variable step size parameter is proposed for PAPA.

The APA can be derived from the principle of least perturbation, that is to maintain the next coefficient vector as close as possible to the current estimate, while forcing the *a posteriori* output estimation error to be zeros [2, 3, 7]. The *a posteriori* output estimation error vector $\mathbf{r}(t)$ is defined as [7]

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{d}(t) - \mathbf{X}^T(t)\mathbf{w}(t+1) \\ &= \mathbf{X}^T(t)\tilde{\mathbf{w}}(t+1) + \mathbf{v}(t), \end{aligned} \tag{6.10}$$

$$\tilde{\mathbf{w}}(t) = \mathbf{h} - \mathbf{w}(t), \tag{6.11}$$

where $\tilde{\mathbf{w}}(t)$ is the coefficient error vector and $\mathbf{v}(t) = [v(t)\ v(t-1)\ \cdots\ v(t-P+1)]^T$ is the disturbance signal vector. Compared to $\mathbf{r}(t)$, the error $\mathbf{e}(t)$ plays the role of the *a priori* output estimation error vector.

The APA can satisfy the principle of least perturbation in a noise-free system. It has the fastest convergence speed and zero misalignment by canceling $\mathbf{r}(t)$ at each iteration. The optimal step size is *one* in this case. However, in practical application, a disturbance signal is inevitable. Therefore the adaptive algorithm cannot achieve zero misalignment. This could be explained by the fact that in the presence of $v(t)$, attempts to force $\mathbf{r}(t)$ to be zero will introduce noise to the adaptive filter update [71]. Actually, what we would like is to force the *a posteriori* estimation error to be zero. That is

$$\mathbf{X}^T(t)\tilde{\mathbf{w}}(t+1) = \mathbf{0}, \tag{6.12}$$

where $\mathbf{0}$ is a $P \times 1$ column vector whose elements are all zeros. Combining (6.10) with (6.12) implies that, in a noisy environment, we should update the coefficients to make the *a posteriori* error not to be zero, but to be the disturbance signal: $\mathbf{v}(t)$,

$$\mathbf{r}(t) = \mathbf{v}(t). \tag{6.13}$$

In practical application, although the disturbance signal $v(t)$ is not available, its power level can be estimated. For this reason the optimal step-size parameter can be found in such a way that

$$E\{r_p^2(t)\} = E\{v_p^2(t)\}, \quad p = 0 \cdots P-1 \tag{6.14}$$

where $r_p(t)$ is the p-th element of $\mathbf{r}(t)$, and $v_p(t)$ is the p-th element of $\mathbf{v}(t)$. Note that $v_p(t) = v(t-p)$.

Based on above notion, a VSS-PAPA can be derived as follows. Rewrite (6.1) with a $P \times P$

time-varying step-size diagonal matrix $\boldsymbol{\alpha}(t)$, ignoring the regularization term $\delta\mathbf{I}$,

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t)\right]^{-1}\boldsymbol{\alpha}(t)\mathbf{e}(t). \tag{6.15}$$

Subtracting $\mathbf{h}$ at both sides and rearranging the terms, yields

$$\tilde{\mathbf{w}}(t+1) = \tilde{\mathbf{w}}(t) - \mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t)\right]^{-1}\boldsymbol{\alpha}(t)\mathbf{e}(t). \tag{6.16}$$

Pre-multiplying $\mathbf{X}^T(t)$ at both sides yields a relation between the *a posteriori* estimation error and the *a priori* output estimation error

$$r_p(t) = \left[1 - \alpha_p(t)\right] e_p(t), \tag{6.17}$$

where $\alpha_p(t)$ is the p-th diagonal element of $\boldsymbol{\alpha}(t)$, and $e_p(t)$ is the p-th element of $\mathbf{e}(t)$. This result is very interesting in relation to PAPA. It can be observed that the *a posteriori* estimation error $\mathbf{r}(t)$ is determined by the step size parameter $\boldsymbol{\alpha}(t)$ and error vector $\mathbf{e}(t)$ and is independent from $\mathbf{G}(t)$. Consequently, a simple variable step-size approach is expected for PAPA from this relation, following a similar procedure to [73].

Squaring and taking mathematical expectation at both sides of (6.17), and combining it with (6.14), gives

$$E\{r_p^2(t)\} = \left[1 - \alpha_p(t)\right]^2 E\{e_p^2(t)\} = E\{v^2(t-p)\}. \tag{6.18}$$

Solving the equation, the p-th time-varying step-size $\alpha_p(t)$ is obtained with a simple expression as

$$\alpha_p(t) = 1 - \sqrt{\frac{\sigma_v^2(t-p)}{\sigma_{e_p}^2(t)}}, \tag{6.19}$$

where $\sigma_v^2(t-p) = E\{v^2(t-p)\}$ is the variance of $v(t-p)$ and $\sigma_{e_p}^2(t) = E\{e_p^2(t)\}$ is the variance of $e_p(t)$.

In the transient state of the adaptive filter, $\sigma_{e_p}^2(t)$ will be large, hence $\alpha_p(t)$ is also large. Consequently, fast convergence speed can be expected. After the adaptive filter reaches to within the immediate vicinity of its optimal value, $\sigma^2_{e_p}(t)$ becomes small, hence $\alpha_p(t)$ decreases. As a result, low misalignment can be observed.

There are some practical considerations related to this expression, which is similar with those described in the previous chapter. The first is the estimations of $\sigma^2_{e_p}(t)$ and $\sigma^2_v(t)$. The quantity of $\sigma^2_{e_p}$ can be estimated using an exponential window as:

$$\hat{\sigma}^2_{e_p}(t) = \lambda_1 \hat{\sigma}^2_{e_p}(t-1) + (1-\lambda_1)e^2_p(t), \tag{6.20}$$

where

$$\lambda_1 = 1 - 1/(K_1 N), \quad (K_1 \in Z^+, K_1 \geq 1) \tag{6.21}$$

A large K_1 can obtain a smooth estimate of $\sigma^2_{e_p}(t)$ but it will reduce the tracking ability of the adaptive filter. In practical application, power estimation of the disturbance signal, $\hat{\sigma}^2_v(t)$, can be obtained during the silences in a network echo cancellation system. An estimate of the disturbance signal, $\hat{v}(t)$, can even be obtained using an additional adaptive filter, as proposed in [76]. Therefore, by using the same method with $\hat{\sigma}^2_{e_p}(t)$, $\hat{\sigma}^2_v(t)$ can be obtained by

$$\hat{\sigma}^2_v(t) = \lambda_1 \hat{\sigma}^2_v(t-1) + (1-\lambda_1)\hat{v}^2(t). \tag{6.22}$$

The second issue is stability. These estimates could lead to minor deviations from their theoretical values, which may result in a negative step size or a large one and force the adaptive algorithm to diverge. It is necessary to restrict $\alpha_p(t)$ in range so that the stability of the adaptive algorithm is guaranteed, $0 \leq \alpha_{\min} \leq \alpha_p(t) \leq \alpha_{\max} \leq 2$. Suitable choice of $\alpha_{\min}$ and $\alpha_{\max}$ can make the proposed algorithm robust to an inaccurate estimate of $\hat{\sigma}^2_v$. More detailed discussions on this issue will be presented in the next subsection.

The third issue is the determination of $\mathbf{G}(t)$. Although it can be determined by any proportionate adaptive algorithm, it is preferable to adopt the segment proportionate version described by SPNLMS because it has excellent performance and light computation load. This definition is used throughout this chapter. The proposed algorithm with this definition of $\mathbf{G}(t)$ will be referred as VSS-SPAPA for convenience sake. Note that when $\mathbf{G}(t)$ is an identity matrix the proposed VSS-SPAPA degrades into the VSS-APA in [73].

It seems that the proposed variable step-size PAPA is similar to the set-membership PAPA (SM-PAPA) proposed in [62], where $\alpha_p(t)$ is replaced by

$$\alpha_{sm,p}(t) = \begin{cases} 1 - \gamma/|e_p(t)|, & \text{if } |e_p(t)| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{6.23}$$

Here, γ is a predetermined parameter as a bound on the noise. Since there is no averaging on, it is obvious that a lower misalignment cannot be expected. The proposed variable step size described in (6.19) provides an optimal criteria of γ for the SM-PAPA, i.e. γ should choose according to σ_v.

6.4 Adaptive Estimate of $\sigma_v^2(t)$ and Its Influence

The proposed VSS-SPAPA can obtain optimal $\alpha_p(t)$ if an accurate estimate of $\sigma_v^2(t)$ is available. As explained in the previous subsection, a relatively accurate estimate can be easily obtained during silence in a network echo cancellation system, since there are many pauses in natural speech. However, if the power of the disturbance signals changes between two consecutive estimations, its new estimate will not be available immediately. A method was proposed to adaptively estimate $\sigma_v^2(t)$ only using the signals available in the system, which can be described by [73]

$$\hat{\sigma}_v^2(t) = \max\{0,\ \hat{\sigma}_d^2(t) - \hat{\sigma}_{\hat{y}}^2(t)\}, \tag{6.24}$$

$$\hat{\sigma}_d^2(t) = \lambda_2 \hat{\sigma}_d^2(t-1) + (1-\lambda_2) d^2(t), \tag{6.25}$$

$$\hat{\sigma}_{\hat{y}}^2(t) = \lambda_2 \hat{\sigma}_{\hat{y}}^2(t-1) + (1-\lambda_2) \hat{y}^2(t), \tag{6.26}$$

where $\hat{y}(t) = \mathbf{x}^T(t)\mathbf{w}(t)$ is the output of the adaptive filter, and

$$\lambda_2 = 1 - 1/(K_2 N), \quad (K_2 \in Z^+, K_2 \geq K_1). \tag{6.27}$$

This estimate is approximately accurate after the adaptive filter has reached its steady state, because

$$E\{v^2(t)\} \approx E\{d^2(t)\} - E\{\hat{y}^2(t)\}. \tag{6.28}$$

The method provides a sub-optimal solution of $\sigma_v^2(t)$ if it is unavailable in the given practical application. However, after the adaptive filter has converged to within the immediate vicinity of its optimal value, it can also be found that [72]

$$E\{e^2(t)\} \approx E\{d^2(t)\} - E\{\hat{y}^2(t)\}. \tag{6.29}$$

Therefore, the difference between $\hat{\sigma}_{e_p}^2(t)$ and $\hat{\sigma}_v^2(t)$ will be insignificant in the steady state of the adaptive filter. As a result $\alpha_p(t)$ will be very small, and low steady-state misalignment observed. However, this approach will lead to a slow convergence speed if this approach is applied during the transient period of the adaptive filter. For example, if the unknown impulse response changes suddenly, this approach is believed to present poor tracking ability because $\alpha_p(t)$ remains relatively small. To improve the adaptive filter's tracking ability in this scenario, the value of $\alpha_{\min}$ should not be too small, and the steady-state misalignment will be increased as a consequence. An alternative is to introduce a impulse response variation detector in the algorithm, see [73] and reference therein.

Let us discuss the influence of an inaccurate estimate of $\hat{\sigma}_{\hat{v}}^2(t)$ on the convergence performance of the adaptive algorithm. In the case of $\hat{\sigma}_v^2(t) \gg \sigma_v^2(t)$, $\alpha_p(t)$ will be smaller than its optimal value defined in (6.19). The convergence speed of the proposed algorithm will be slowed because $\alpha_p(t)$ reaches $\alpha_{\min}$ soon. However low misalignment will be achieved using more iterations. In the case of $\hat{\sigma}_v^2(t) \ll \sigma_v^2(t)$, $\alpha_p(t)$ will be greater than its optimal value. Fast initial convergence speed will be observed but the misalignment will increase because $\alpha_p(t)$ is close to $\alpha_{\max}$. For a modest inaccurate estimate of $\hat{\sigma}_v^2(t)$, the performance of the proposed algorithm is better than that of the VSS-APA because it benefits from the fast proportionate adaptive

Table 6.1 The proposed VSS-SPAPA

Initialization:
$\mathbf{w}(0) = \mathbf{0};\ \hat{\sigma}_v^2(0) = 0;\ \hat{\sigma}_{\hat{y}}^2(0) = 0; \hat{\sigma}_d^2(0) = 0;$
$\delta = PM\sigma_x^2,\ (M \in Z^+, M \geq 1)$
$\lambda_1 = 1 - 1/(K_1 N), (K_1 \in Z^+,\ K_1 \geq 1)$
$\lambda_2 = 1 - 1/(K_2 N), (K_2 \in Z^+,\ K_2 \geq K_1)$
for $p = 0, 1, \cdots, P-1$
$\quad \hat{\sigma}_{e_p}^2(0) = 1;$
For all t:
$\bar{g}_n(t) = \begin{cases} 400\|w_n(t)\|, & \|w_n(t)\| < 0.005 \\ 2, & \text{otherwise.} \end{cases}$
$L_{\max} = \max\{\delta_\rho, \bar{g}_0(t), \cdots, \bar{g}_{N-1}(t)\}$
$\gamma_n(t) = \max\{\bar{g}_n(t), \rho L_{\max}\}$
$g_n(t) = \gamma_n(t) / \left[\frac{1}{N}\sum_{i=0}^{N-1}\gamma_i(t)\right]$
$\mathbf{G}(t) = \mathrm{diag}\{g_0(t), g_1(t), \cdots, g_{N-1}(t)\}$
$\hat{\mathbf{y}}(t) = \mathbf{X}^T(t)\mathbf{w}(t)$
$\mathbf{e}(t) = \mathbf{d}(t) - \hat{\mathbf{y}}(t)$
if $\hat{\sigma}_v^2(t)$ is not available,
$\quad \hat{\sigma}_d^2(t) = \lambda_2\hat{\sigma}_d^2(t-1) + (1-\lambda_2)d^2(t)$
$\quad \hat{\sigma}_{\hat{y}}^2(t) = \lambda_2\hat{\sigma}_{\hat{y}}^2(t-1) + (1-\lambda_2)\hat{y}^2(t)$
$\quad \hat{\sigma}_v^2(t) = \max\{0,\ \hat{\sigma}_d^2(t) - \hat{\sigma}_{\hat{y}}^2(t)\}$
for $p = 0, 1, \cdots, P-1$
$\quad \hat{\sigma}_{e_p}^2(t) = (1-\lambda_1)\hat{\sigma}_{e_p}^2(t-1) + \lambda_1\hat{e}_p^2(t)$
$\quad \alpha_p(t) = 1 - \sqrt{\hat{\sigma}_v^2(t-p)/\hat{\sigma}_{e_p}^2(t)}$
$\quad$ if $\alpha_p(t) < \alpha_{\min}, \quad \alpha_p(t) = \alpha_{\min},$
$\quad$ if $\alpha_p(t) > \alpha_{\max}, \quad \alpha_p(t) = \alpha_{\max},$
$\boldsymbol{\alpha}(t) = \mathrm{diag}\{\alpha_0(t), \cdots, \alpha_{P-1}(t)\}$
$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{G}(t)\mathbf{X}(t)\left[\mathbf{X}^T(t)\mathbf{G}(t)\mathbf{X}(t) + \delta\mathbf{I}\right]^{-1}\boldsymbol{\alpha}(t)\mathbf{e}(t)$

algorithms. The simulation results in Section 4 show that the proposed algorithm can tolerate a large $\hat{\sigma}_v^2(t)$ estimate error.

The proposed variable step-size segment proportionate affine projection algorithm (VSS-SPAPA) is summarized in Table 6.1.

6.5 Computational Complexity

Compared to the standard APA, the additional computation load of the proposed VSS-SPAPA is composed of four parts. First, the calculation of $\mathbf{G}(t)$ costs $2N + 2$ multiplications or divisions, N additions and $3N$ comparisons. Second, the calculation of $\mathbf{G}(t)\mathbf{X}(t)$ costs PN multiplications. Third, the estimate of $\hat{\sigma}^2_{e_p}(t)$ and calculation of $\boldsymbol{\alpha}(t)$ will cost $4P + 6$ multiplications or divisions, P square-root operations, and $2P + 2$ additions or subtractions. Finally, the calculation of $\boldsymbol{\alpha}(t)\mathbf{e}(t)$ costs P multiplication. The remaining operations are common with APA. In summary, the dominant additional computation cost of the proposed VSS-SPAPA is $(P + 2)N + 5P + 8$ multiplications or divisions operations and P square-root operations. For practical applications, such as network echo cancellation, the value of projection order P is usually in the range of 2 to 8. Therefore the computational complexity of the proposed VSS-SPAPA is moderate. The additional computation load is worth the considerable performance improvement in sparse impulse response, and illustrate this in the next section.

6.6 Simulation Results

To evaluate the performance of the proposed algorithm, many computer simulations were conducted in the context of system identification. Four algorithms are compared in numerous simulations, APA, SPAPA, VSS-APA [73] and the proposed VSS-SPAPA. The unknown system $\mathbf{h}$ is taken from a network echo path illustrated in Fig. 6.1(a). Both $\mathbf{h}$ and the adaptive filter $\mathbf{w}$ have same length, $N = 512$. For the proportionate algorithms, $\delta_\rho = 0.01$, $\rho = 1/N$. For VSS-SPAPA, $\lambda_1 = 1 - 1/2N$, $\lambda_2 = 1 - 1/4N$, $\alpha_{\min} = 0.005$. $\alpha_{\max} = 1.0$ is assigned because a large step size for SPAPA does not considerably improve its convergence speed but results in higher misalignment. The regularization parameters are chosen by $\epsilon = 10P\sigma_x^2$ for all the algorithms. The disturbance signal $v(t)$ is an independent white Gaussian noise. The convergence performance is evaluated using the normalized misalignment (in dB) defined by

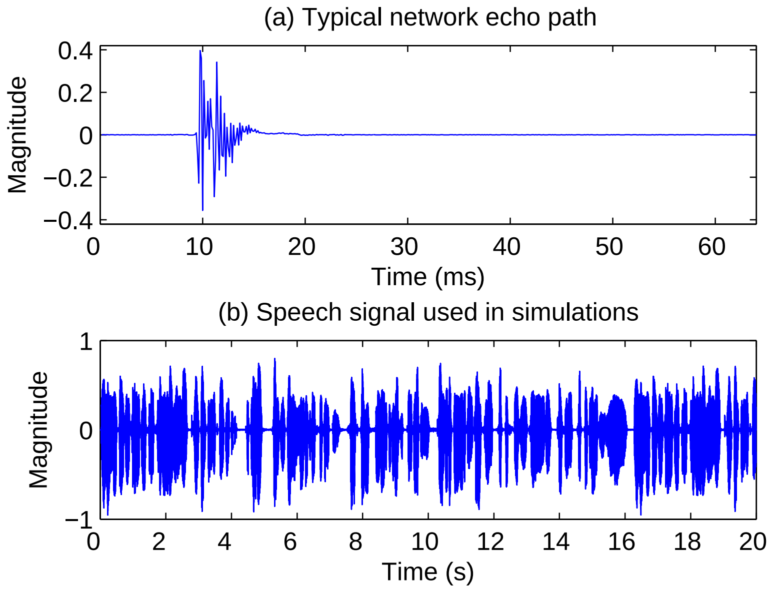

Fig. 6.1 (a) A typical sparse network echo path used in the simulations. (b) A segment of speech signal with 8k sampling rate used in the simulations.

$10\log_{10}\left(\|\mathbf{h}-\mathbf{w}(t)\|_2^2/\|\mathbf{h}\|_2^2\right)$.

6.6.1 Simulations with real value of $\sigma_v^2(t)$

The performance of the proposed VSS-SPAPA is compared with the others using the real value of $\sigma_v^2(t)$. The signal-to-noise rate (SNR) is adjusted to 20dB for the simulation results illustrated. The misalignment of the four algorithms is compared with three kinds of input signal: (a) white Gaussian noise signal, (b) highly colored signals generated by a AR(1) process, and (c) speech signals illustrated in Fig. 6.1(b).

In the 1st set of simulations, the input sequence was zero-mean white Gaussian noise signal with $\sigma_x^2 = 1$. For APA and SPAPA, a constant step $\alpha = 1$ is assigned. Convergence speed in the first 15000 iterations of the related algorithms is compared in Fig. 6.2(a). The projection order $P = 1$. Therefore the related algorithms degrade into their corresponding NLMS versions. It can be seen that the SPNLMS algorithm converges faster than the conventional NLMS algorithm with same step size. It can also be found that they have almost the same steady-state

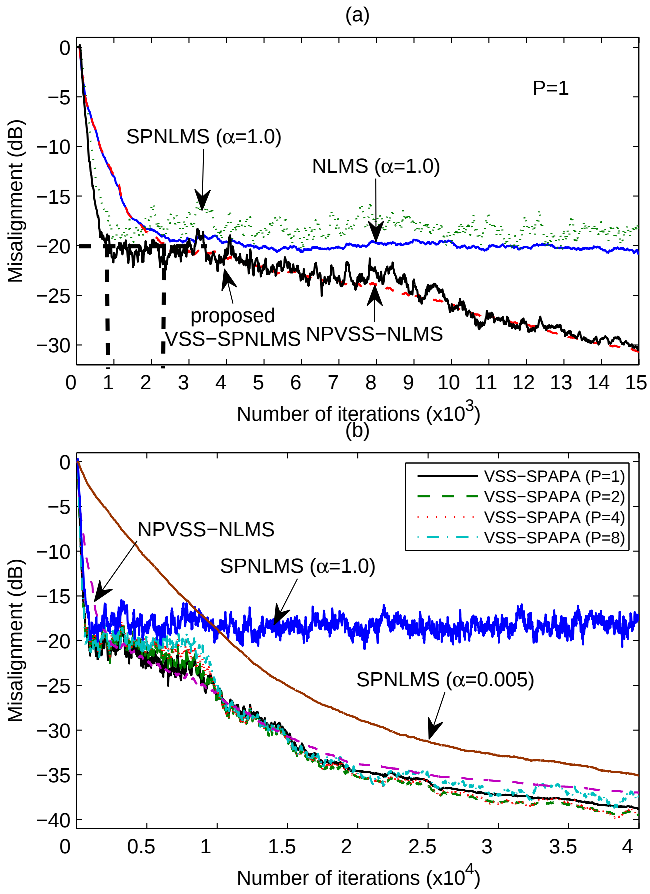

Fig. 6.2 Comparison of VSS-SPAPA to the related algorithms with white Gaussian noise input. (a) P=1. (b) Comparison of different projection order, P=1,2,4,8. SPNLMS and NPVSS-NLMS [71] are also illustrated.

misalignment. The proposed VSS-SPNLMS algorithm achieves almost the same fast initial convergence as the SPNLMS algorithm. However, it can obtain lower steady-state misalignment; about 18dB improvement can be observed in 4×10^4 iterations, as shown in Fig. 6.2(b). To achieve this low level misalignment, the SPNLMS requires a very small step size. α is 0.005 in this case, whose convergence speed is greatly degraded. Although the NPVSS-NLMS algorithm in [71] can achieve almost the same low misalignment, its convergence speed is significantly lower than the proposed algorithm. It can be seen from Fig. 6.2(a) that the proposed VSS-SPNLMS algorithm reaches −20dB misalignment in approximately 900 iterations but the NPVSS-NLMS algorithm reaches that level in about 2200 iterations. Figure 6.2(b) illustrates

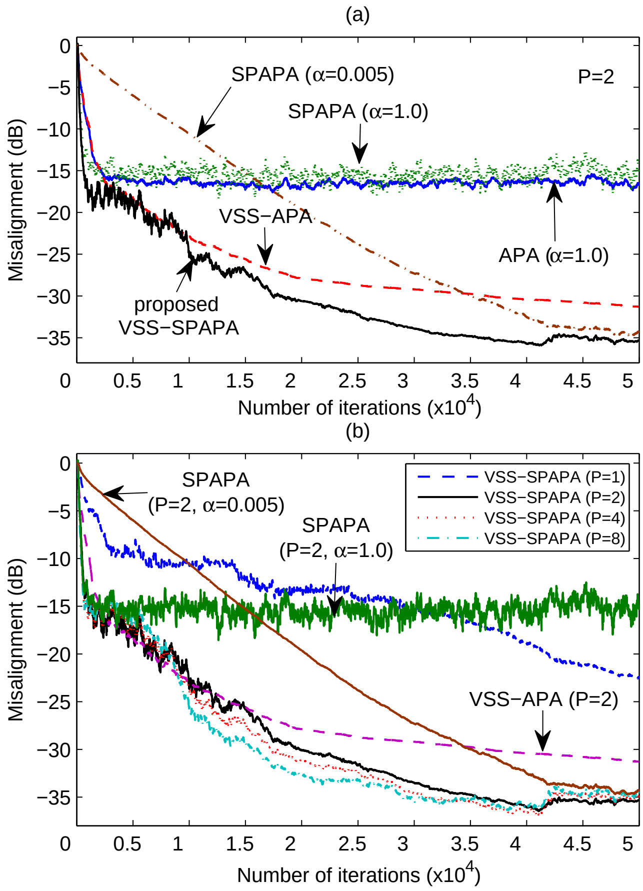

Fig. 6.3 Comparison of VSS-SPAPA to the related algorithms using highly colored input signal generated by $G(z)$. (a) P=2. (b) Comparison with different projection order, P=1,2,4,8. SPAPA and VSS-APA are also illustrated.

the results of different projection order. It can be seen that in the case of white input signal, the increase in the projection order from 2 to 8 does not considerably improve the convergence speed of the proposed VSS-SPNLMS. This suggests that the control matrix $\mathbf{G}(t)$ determined by SPNLMS is nearly optimal for white input signal. The proposed VSS-SPNLMS is preferred to obtain optimal convergence speed and lower misalignment with least computation cost.

In the 2nd set of simulations, the input sequence $\{x(t)\}$ is an AR(1) process generated by filtering a zero-mean white Gaussian signal through a first-order system $G(z) = 1/(1 - 0.9z^{-1})$.

The simulation results with projection order $P = 2$ are illustrated in Fig. 6.3(a). The proposed VSS-SPAPA achieves almost the same initial convergence speed with SPAPA, but it can reach a much lower steady-state misalignment - approximately 20dB improvement can be observed in 5×10^4 iterations. Although VSS-APA can almost achieve this level of misalignment, its convergence speed is lower than the proposed VSS-SPAPA. Figure 6.3(b) shows the misalignment of the proposed algorithm of different projection order. In this case, when P=1, all of the related algorithms present very low convergence speed. Their performance can be greatly improved by increasing P= 2. However, with the increase in the projection order from 4 to 8, the convergence speed of VSS-SPAPA does not increase further, while the convergence speed of APA and VSS-APA will improve with increased projection order from 1 to 8. The VSS-SPAPA with P= 2 is preferable in this case to obtain the best performance with a modest increase in computational complexity.

In the 3rd set of simulations, the input is from a speech segment illustrated in Fig. 6.1(b). The disturbance signal $\{v(t)\}$ is uncorrelated zero-mean white Gaussian noise with SNR= 20dB. Figure 6.4(a) illustrates the result with projection order $P = 8$. The proposed VSS-SPAPA achieves almost the same fast initial convergence compared to the SPAPA, but it can achieve lower steady-state misalignment - about 15dB improvement can be observed in 15 seconds. In this case, VSS-APA cannot achieve this misalignment level (or it needs many more minutes to achieve it). The VSS-SPAPA outperforms VSS-APA both on convergence speed (approximately twice as fast) and on low misalignment (approximately 2dB lower). Figure 6.4(b) compares the convergence of the proposed algorithm at different projection orders. In the case of speech signal input, with an increase in projection order from 1 to 8, the convergence speed of all the related algorithms was improved. Taking into account computational complexity, the performance of P= 4 is good enough in the case of speech signal input with a modest increase in computation load.

The tracking ability of the adaptive algorithms is important in a non-stationary environment where the unknown impulse response may suddenly change. In a network echo cancellation system, the echo path is subject to shift backward or forward as a result of delay jitter. Figure

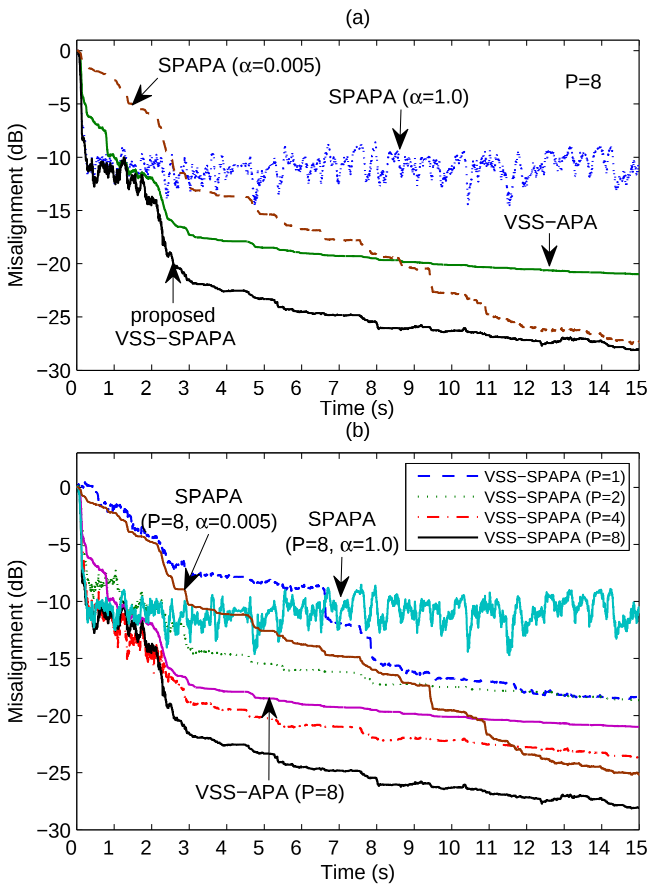

Fig. 6.4 Comparison of VSS-SPAPA to the related algorithms using speech signal. (a) P=8. (b) Comparison with different projection order, P=1,2,4,8. SPAPA and VSS-APA are also illustrated.

6.5 illustrates the result of the tracking ability of the relevant algorithms with speech signal input and P=4. The unknown impulse response suddenly shifted to the right by 12 samples. It can be seen from this figure that the proposed algorithm presents good tracking performance after the unpredicted change in the unknown impulse response. Furthermore, it outperforms its counterparts in low steady-state misalignment after reconvergence.

6.6.2 Simulations with inaccurate estimate of $\sigma_v^2(t)$

As discussed in the previous section, the estimated accuracy of $\hat{\sigma}_v^2(t)$ influences the convergence of VSS-APA and VSS-SPAPA. Figure 6.6 illustrates the simulation results of the relevant

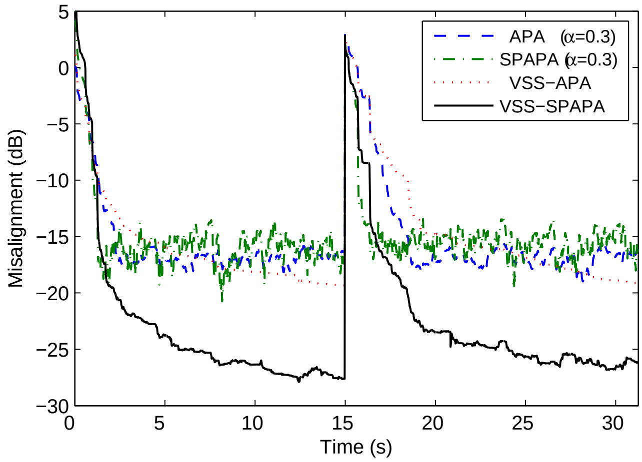

Fig. 6.5 Comparison of tracking ability of VSS-SPAPA to the relevant algorithms using speech signals, P=4. The unknown impulse response changes after 15 second.

algorithms with: AR(1) input signals in subplot (a), and speech input signals in subplot(b), respectively. It can seen that VSS-SPAPA with an accuracy of $\sigma_v^2(t)$ is best among the three cases: (1) the power estimate of the disturbance signal is greater than its real value, $\hat{\sigma}_v^2(t) = 1.5\sigma_v^2(t)$; (2) $\hat{\sigma}_v^2(t) = \sigma_v^2(t)$ and (3) $\hat{\sigma}_v^2(t)$ is smaller than its real value, $\hat{\sigma}_v^2(t) = 0.8\sigma_v^2(t)$. In any case VSS-SPAPA can maintain fast convergence with both AR(1) signals and speech signals. In case (1), a large estimate $\hat{\sigma}_v^2(t)$ will cause $\alpha_p(t)$ to be small and reach $\alpha_{\min}$ faster than in case (2). In any case, $\alpha_{\min}$ warrants VSS-SPAPA to achieve low misalignment. It is obvious that this requires more iterations but is still faster than VSS-APA, as shown in the figure. However, if $\hat{\sigma}_v^2(t)$ is excessively smaller than its real value, the steady-state misalignment of VSS-SPAPA is relatively higher, as shown in the figure, because $\alpha_p(t)$ remains large in the steady state - around 0.25 in case (3). The VSS-SPAPA does not behave worse than the SPAPA even with a large estimate error of $\hat{\sigma}_v^2(t)$. It can tolerate more than +50% estimate error and about -20% estimate error of $\hat{\sigma}_v^2(t)$. The proposed VSS-SPAPA is robust to a relatively inaccurate estimate of $\hat{\sigma}_v^2(t)$.

If an estimate of $\sigma_v^2(t)$ is not available in practical application, it can be adaptively estimated according to (6.28). Figure 6.7 illustrates the results of the proposed VSS-SPAPA with this estimate method included. As discussed in the previous section, the problem with this

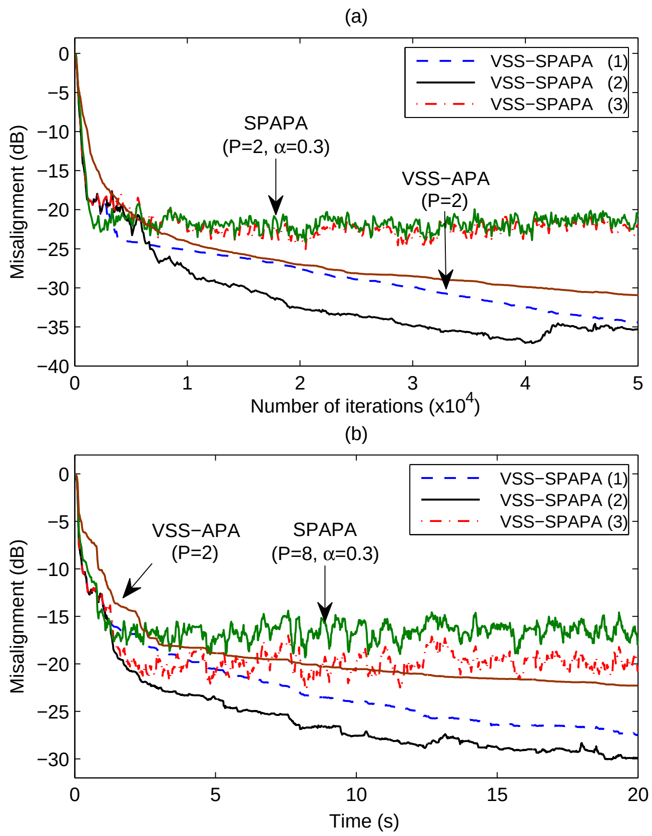

Fig. 6.6 Comparison of VSS-SPAPA to the relevant algorithms with inaccurate estimate of $\hat{\sigma}_v^2(t)$. (a) With highly colored input signals generated by $G(z)$. P= 2. (b) With speech input signals, P=8. (1) $\hat{\sigma}_v^2(t) = 1.5\sigma_v^2(t)$, (2) $\hat{\sigma}_v^2(t) = \sigma_v^2(t)$, and (3) $\hat{\sigma}_v^2(t) = 0.8\sigma_v^2(t)$.

adaptive estimate method is that the estimate is only effective when the adaptive filter has converged. Otherwise, the estimate will be very inaccurate and cause $\alpha_p(t)$ to be very small. Hence the tracking ability of the proposed algorithms will be considerably worsened. So the relevant algorithms are tested on the assumption that the unknown impulse response suddenly changes by shifting to the right 12 samples. The subplot (a) is with the highly colored AR(1) input (P=2) and subplot (b) is with speech input signals (P=8). For SPAPA and APA, the constant step size is $\alpha = 0.3$, which is suitable for practical application.

It can be seen that VSS-SPAPA can also quickly track the change in unknown impulse response and then achieve a lower misalignment after it has reconverged. It outperforms both

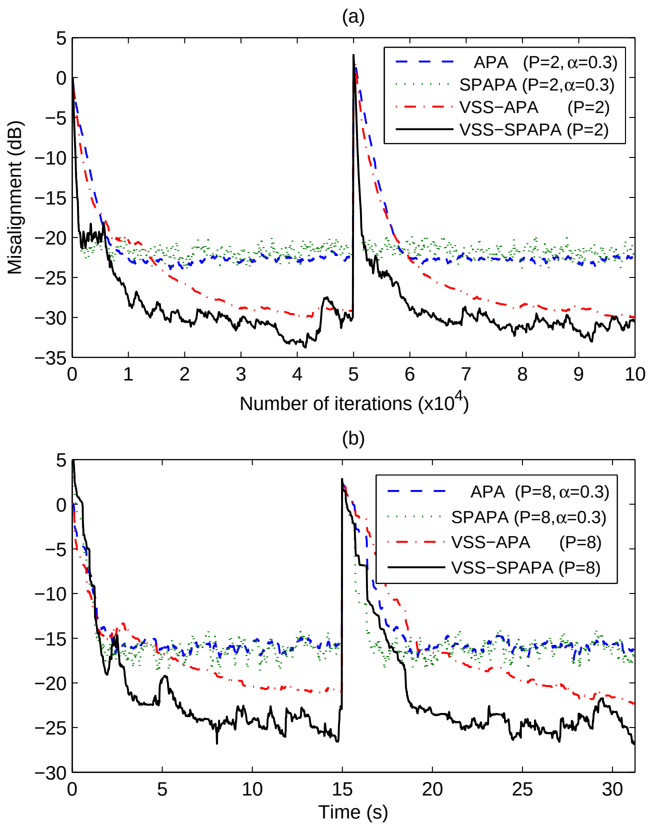

Fig. 6.7 Comparison of VSS-SPAPA to the relevant algorithms with adaptive estimate of $\sigma_v^2(t)$. The unknown impulse response changes at 5×10^4 iteration. (a) With highly colored input generated by $G(z)$. P=2. (b) With speech input signal, P=8.

the non-proportionate counterparts and the constant step-size algorithms. However, compared to the result with real value of $\hat{\sigma}_v^2(t)$, the steady-state misalignment of VSS-SPAPA with adaptive estimate of $\sigma_v^2(t)$ is worse than that with real value of $\sigma_v^2(t)$. For example, the misalignment of VSS-SPAPA in Fig. 6.3 reaches −34dB in 3×10^4 iterations, but in Fig. 6.7(a) it only reaches about −30dB. Figure 6.7(b) shows the result when the input is speech signal. The proposed VSS-SPAPA can achieve a lower steady-state misalignment than SPAPA, approximate 10dB improvement was achieved. The steady-state misalignment of VSS-APA was the same as that of VSS-SPAPA, but it needs more time to reach that level. Compared to the scenario where the accurate estimate of $\sigma_v^2(t)$ is known, as shown in Fig. 6.5(b), the steady-state misalignment in this scenario reaches only approximately −25dB in 15 seconds. As expected, the reconvergence performance of both VSS-APA and VSS-SPAPA are slower than their non-VSS counterparts during the period from 15s to 20s. Nevertheless, the VSS-SPAPA outperforms VSS-APA in convergence speed with almost the same steady-state misalignment.

In brief, the proposed VSS-SPAPA achieves faster convergence and lower misalignment than the conventional algorithms for the identification of sparse impulse response in the tested cases of different input signals.

6.7 Conclusions

A method is proposed to introduce a variable step-size approach into the proportionate affine projection algorithm for identification of the sparse impulse response. The proposed algorithm can achieve not only very fast convergence but also relatively low misalignment. It is particularly efficient for highly colored input signals, such as speech. It does not require many parameter adjustment so it is easy to use in practical application. It only requires an estimate of $\sigma_v^2(t)$, the power level of the disturbance signals. If this estimate is not available, an adaptive estimate method is applicable with only a little performance loss. Simulations show that the proposed VSS-SPAPA outperforms the conventional adaptive algorithms for the identification of sparse impulse response.

Chapter 7

A Novel Perspective of Proportionate Step Size

7.1 Introduction

Generally speaking, information is distributed unevenly in practical application. Network echo path, for instance, has hundreds or thousands coefficients but only a small fraction of coefficients have prominent value and the others are zeros or very small. Room impulse response, although not as sparse as network echo path, has a small fraction of big coefficients distributed in a narrow time period and the others are relatively small. Two impulse responses frequently dealt with in practical application are illustrated in Fig. 7.1. By assigning large coefficients relatively large step gain, the overall convergence of adaptive filter can be improved. The resulting proportionate adaptive algorithms introduce a diagonal step-size control matrix $\mathbf{G}(t)$ in order to individually assign each coefficient a step size parameter proportional to its magnitude estimate at time t.

The disadvantages of the PNLMS algorithm are twofolds. On the one hand, the convergence becomes slow after a fast initial period. On the other hand, they are generally inefficient for a non-sparse impulse response. Some modifications have been proposed to solve these problems. As to the first problem, there are two algorithms. The PNLMS++ algorithm can improve the convergence of PNLMS after its fast initial period [29]. It alternates PNLMS and NLMS during the adaptation. It converges at least as fast as the NLMS algorithm. However, it also

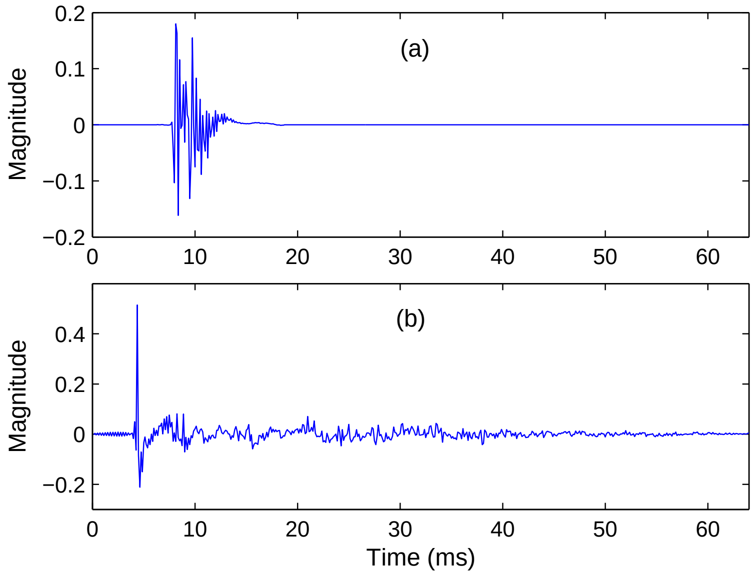

Fig. 7.1 Two impulse responses frequently encountered. (a) Network echo path (sparse); (b) Room impulse response (non-sparse).

make the convergence in the initial period degrade. The CPNLMS algorithm was proposed to switch PNLMS adaptation to NLMS when slow convergence is detected [31]. Its main problem is there are many parameters have to choose in priori. It is also difficult to decide when it should switch to NLMS just according to the error signal. As to the second problem, the IPNLMS algorithm is proposed in [30] to avoid degradation in the case when the underlying impulse response is non-sparse.

The reason why PNLMS converges slower than NLMS for non-sparse impulse response is because of the brutal choice of the maximum operator. Instead, the average of the current coefficients is adopted to partly remove the negative effects of the inaccurate estimate. The IPNLMS algorithm converges as fast as the PNLMS algorithm for sparse impulse responses and its performance is not worse than the NLMS algorithm for non-sparse impulse responses. It is interesting that IPNLMS shows better convergence performance than PNLMS after the fast initial period. The reason is because it assigns the small coefficients reasonable step gain, which is usually larger than that of the PNLMS algorithm.

All existing proportionate adaptive algorithms exploit the shape of the target impulse response to decide the proportionate step gain for every coefficient. This is not always suitable and it is the reason why the existing proportionate adaptive algorithms exhibit slow convergence after the fast initial period and for non-sparse impulse response.

In this chapter, a novel method is proposed to determine the proportionate step gain. Actually, the proportionate step-size parameter should be determined according to the distance between the current estimate and its optimal value. Based on this idea, a novel approach for proportionate step-size is proposed. This approach distribute the large adaptation gains not only to taps that are currently large, but also to taps that tend to be large. Simulation results verify the effectiveness of the proposed algorithm.

7.2 Optimal Proportionate Step Gain

Proportionate adaptive algorithms determine its step gain for each coefficient according to the shape of impulse response so that large coefficients get large step gain. However, this method is not alway suitable. Actually, convergence of proportionate adaptive algorithms can be divided into two periods:

- fast initial period,
- the second period thereafter.

At the initial period, PNLMS assigns large step gain for large coefficients hence the adaptive filter converges at high speed. After large coefficients have reached a small vicinity of its optimal value, PNLMS enters its second period. In the second period, convergence of adaptive filter is dominated by the relatively small coefficients. Because PNLMS assigns small step gain for these small coefficients, it converges at very low speed in this period. This is the reason why PNLMS converges slowly during the second period.

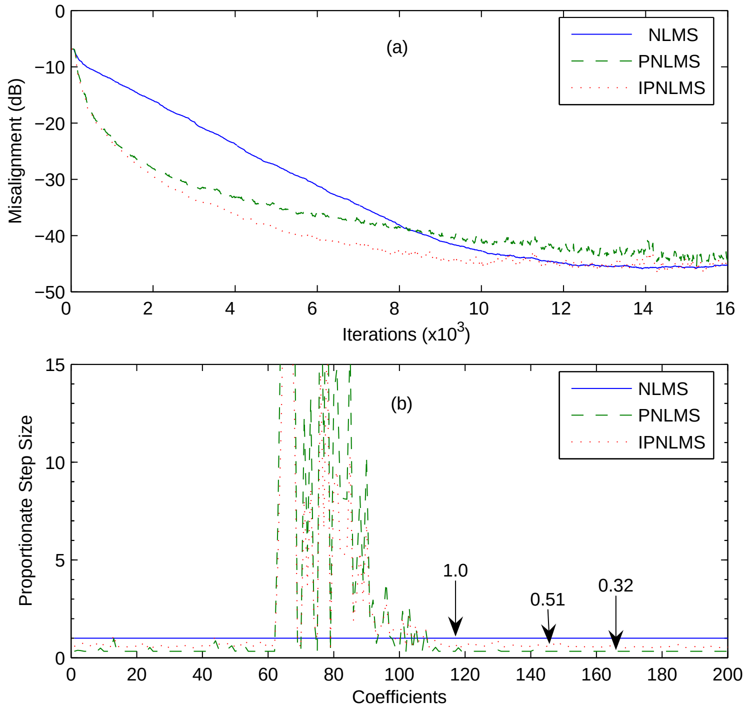

Fig. 7.2 (a) Convergence of PNLMS and IPNLMS. (b) Proportionate step gain of PNLMS and IPNLMS when t = 2000 (only show the first 200 coefficients).

Figure 7.2(a) compares the convergence of PNLMS and IPNLMS using the sparse network echo path illustrated in Fig 7.1(a). Figure 7.2(b) illustrates g_n when $t = 2000$, where generally the proportionate adaptive algorithms enter the second period. It can be seen that after $t = 2000$, the convergence speed of both PNLMS and IPNLMS become slow than NLMS. The proportionate step gain is mainly distributed among the large coefficients, hence the convergence during the second period is slow. It can be seen that the proportionate step gain of PNLMS and IPNLMS for the small coefficients are 0.32 and 0.51, respectively. This value is smaller than that of NLMS, since most step gain is distributed to the large coefficients. Because IPNLMS assign the small coefficients a relatively large proportionate step gain than PNLMS, it shows faster convergence speed than PNLMS during the second period.

An ideal approach is to automatically distribute the proportionate step gain evenly to give the small coefficients fair chance to converge as fast as possible during the second period. The

CPNLMS algorithm can switch PNLMS to NLMS by explicitly detecting the second period. However, it is difficult to exactly detect it by comparing the output error of adaptive filter with a priori threshold.

The objective of this section is to find a new approach to adaptively determine proportionate step gain. It seems that it is not enough to adjust the proportionate step size $\mathbf{G}(t)$ only according to the shape of impulse response. $\mathbf{G}(t)$ should be automatically adjusted in the second period so that it can distribute approximately the same step gain to all coefficients.

Given any impulse response $\mathbf{h}$, let us investigate the convergence of proportionate adaptive algorithms in order to find a suitable approach for $\mathbf{G}(t)$ in different period. It is assumed that the proportionate step gain changes only in a relatively long period, such as M iterations, which is valid in practical application. So it can be investigated in a way of block by block, where block size is M. Defining the coefficient error vector as $\tilde{\mathbf{w}}(t) = \mathbf{h} - \mathbf{w}(t)$. Consider evolution of $\tilde{\mathbf{w}}(t)$ during every M iterations separately. At the k-th block, adaptive filter begins with $\mathbf{w}(kM)$ as its initial value and it finally reaches $\mathbf{w}((k+1)M)$. The objective is to find a suitable proportionate step gain $\mathbf{G}_k$ for k-th block. Starting from steepest descent algorithm, with proportionate adaptation gain $\mathbf{G}_k$, gives [7]

$$\begin{aligned} \mathbf{w}(kM+i+1) &= \mathbf{w}(kM+i) + \alpha \mathbf{G}_k \mathbf{R} \left[\mathbf{h} - \mathbf{w}(kM+i)\right], \\ &\qquad (k = 0, 1, \cdots; i = 0, \cdots, M-1). \end{aligned} \tag{7.1}$$

where $\mathbf{R}$ is the auto-correlation matrix of input signal. For Gaussian white input signal, $\mathbf{R} = \sigma_x^2 \mathbf{I}$, where σ_x^2 is the variance of $\{x(t)\}$ and $\mathbf{I}$ is an $N \times N$ identity matrix. Subtracting $\mathbf{h}$ at both sides of (7.1) and rearranging the terms, it gives

$$\tilde{\mathbf{w}}(kM+i+1) = (\mathbf{I} - \alpha \mathbf{G}_k \mathbf{R})\, \tilde{\mathbf{w}}(kM+i). \tag{7.2}$$

Extending this serious, it obtains that

$$\tilde{\mathbf{w}}((k+1)M) = (\mathbf{I} - \alpha \mathbf{G}_k \mathbf{R})^M\, \tilde{\mathbf{w}}(kM). \tag{7.3}$$

For the n-th coefficient, it gives

$$\tilde{w}_n((k+1)M) = (1 - \alpha\sigma_x^2 g_{k,n})^M [h_n - w_n(kM)]. \tag{7.4}$$

When $0 < \alpha\sigma_x^2 g_{k,n} < 2$, $\tilde{w}_n(t)$ exponentially decreases to 0. Without loss of generality, assume that the initial value of adaptive filter is zeros, $w_n(0) = 0$. In the initial period, when $k = 0$, it gives

$$\tilde{w}_n(M) = (1 - \alpha\sigma_x^2 g_{0,n})^M h_n. \tag{7.5}$$

For a large coefficient h_n, it is obvious that a large $g_{0,n}$ will lead $\tilde{w}_n(M)$ decreases at higher speed so that the adaptive filter $\mathbf{w}$ converges faster. It implies that in the initial period, it is reasonable to determine step gain according to the shape of $\mathbf{h}$ so that the adaptive filter converges at high speed.

After several blocks, large coefficients have converged. As a result, $[h_n - w_n(kM)]$ becomes small and $[\mathbf{h} - \mathbf{w}(kM)]$ is not such a sparse impulse response as $\mathbf{h}$. It implies that in the second period, the step gain of each coefficient should not be determined according to the shape of $\mathbf{h}$ anymore, but according to the difference of $\mathbf{h}$ and $\mathbf{w}(kM)$. Assigning large coefficients too large step gain in the second period has two disadvantages. First, it makes large coefficients produce large estimate error. Second, it causes that the small coefficients cannot obtain reasonable step gain. The proportionate step gain should accordingly adjusted to assign every coefficient fair chance to converge as fast as possible. It is obvious that if $[h_n - w_n(kM)]$ is used to determine the proportionate step gain, it is expected to improve the convergence of proportionate adaptive algorithms in the second period.

Note that it seems that $g_n(t)$ could be determined based on $|w_n(t) - w_n(t - M)|$. This form has an obvious explanation: the coefficients that have changed noticeably during the last M iterations are likely to change in the future. However, it is difficult to implement in practical application because it requires $M \times N$ memory unit to store last M coefficient vectors $\mathbf{w}(t)$.

The proposed approach is a simple approach with similar behavior and it can be implemented efficiently because it only stores additional 1 coefficient vector.

7.3 Proposed Approach of Proportionate Step Gain

In practical application, $\mathbf{h}$ is unknown so $[\mathbf{h} - \mathbf{w}(kM)]$ is unavailable. But above analysis implies that a suitable proportionate step gain can be achieved by using difference of adaptive filter at different time. Inspired by this, in this paper, it is proposed to use the differential of $\mathbf{w}$ at time $(kM{+}i)$ and (kM) to determine the proportionate step gain, i.e., replacing $|w_n(t)|$ in PNLMS and IPNLMS with $|w_n(kM+i) - w_n(kM)|$. Applying this method to PNLMS, a novel PNLMS algorithm can be obtained, which is referred to as DPNLMS in this article for convenience sake. Its diagonal element can be obtained using:

$$L_{\max} = \max\{\gamma, |w_n(kM+i) - w_n(kM)|, \cdots, |w_n(kM+i) - w_n(kM)|\}, \tag{7.6}$$

$$g_n(kM+i) = \max\left\{|w_n(kM+i) - w_n(kM)|, \rho L_{\max}\right\}. \tag{7.7}$$

Applying this approach to IPNLMS, the diagonal element of the resulting algorithm (referred to as DIPNLMS) is:

$$g_n(kM+i) = \beta + \frac{(1-\beta)|w_n(kM+i) - w_n(kM)|}{\frac{1}{N}\sum_{j=0}^{N-1}|w_j(kM+i) - w_j(kM)| + \epsilon}. \tag{7.8}$$

In the initial period when $k = 0, (i = 0 \cdots M-1)$,

$$|w_n(kM+i) - w_n(kM)| = |w_n(i)|, \tag{7.9}$$

so the proposed algorithms behave like PNLMS and IPNLMS. The large coefficient can obtain large step gain hence fast initial convergence can be observed. After several blocks, these large coefficients have converged, therefore $|w_n(kM+i) - w_n(kM)|$ becomes small. As a result, the proposed algorithms will make $g_n(t)$ have almost similar values hence all coefficients get similar

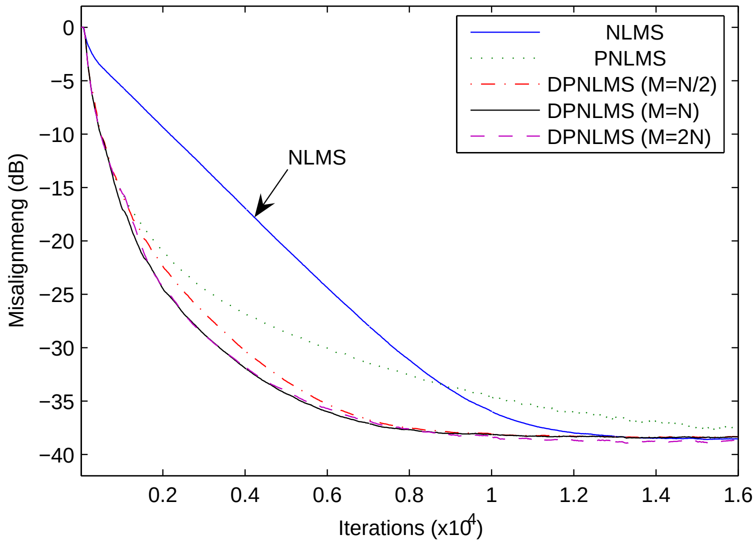

Fig. 7.3 Convergence of DPNLMS using sparse network echo path.

step gain in order to improve their convergence in the second period. Actually, they behave more like NLMS in this period. It is interesting that the proposed approach show a good property that it distributes the large adaptation gains to coefficients that tend to be large. At the beginning of every block when $i = 0$, $|w_n(kM+i) - w_n(kM)| = 0$ so all $g_n(t)$ is equal. With i increases, $|w_n(kM+i) - w_n(kM)|$ indicates the direction of $w_n(t)$ growing up. So the proportionate step gain is proportional to its increase in the past i iterations. The choice of M should as large as that to let $|w_n(kM+i) - w_n(kM)|$ has large value that embodies the increase of coefficient. Simulation results show that $M = N$ or $M = 2N$ presents good performance.

The computational complexity of the proposed algorithms are almost same to their original version: only additional N subtraction is required. Simulation results in the next section show that the proposed algorithms have better convergence performance than the original ones.

7.4 Simulation Results

This section compares convergence speed of the proposed algorithms with NLMS, PNLMS and IPNLMS. Two impulse responses used in simulations are illustrated in Fig. 7.1. The length

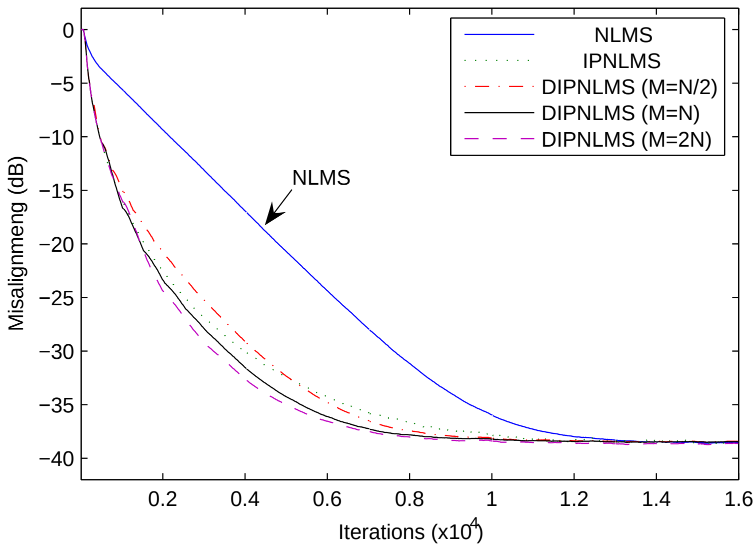

Fig. 7.4 Convergence of DIPNLMS using sparse network echo path.

of adaptive filter $\mathbf{w}$ is $N = 512$. The input signal $x(t)$ is zero-mean white Gaussian noise with unit variance and the disturbance signal $v(t)$ are zero-mean white Gaussian signal that is uncorrelated to $x(t)$. Its magnitude is adjusted to get 30dB signal-noise-ratio (SNR). The global step size parameter $\alpha = 0.25$, and the regularization parameter $\epsilon = 10$. For proportionate algorithms, ρ and γ use their typical values: $\rho = 0.01$, $\gamma = 0.01$. For IPNLMS and DIPNLMS, $\beta = 0.5$. The performance is quantified using the normalized misalignment (in dB) defined as $10\log_{10}\left(||\mathbf{h} - \mathbf{w}(t)||_2^2/||\mathbf{h}||_2^2\right)$. The illustrated results are ensemble average of over 100 independent trials.

Figure 7.3 compares the convergence speed of the proposed DPNLMS ($M = N/2$, $M = N$ and $M = 2N$, respectively) with NLMS and PNLMS using the sparse network echo path in Fig. 7.1(a). It can be seen that DPNLMS considerably improved the convergence speed of PNLMS. When $M = N$ or $M = 2N$, DPNLMS presents good convergence but when $M = N/2$, its performance improvement is not as big as the other two. Figure 7.4 illustrates the convergence of the proposed DIPNLMS comparing to NLMS and IPNLMS. The proposed DIPNLMS algorithm demonstrates better convergence than the IPNLMS algorithm. It can be seen that $M = N/2$ can

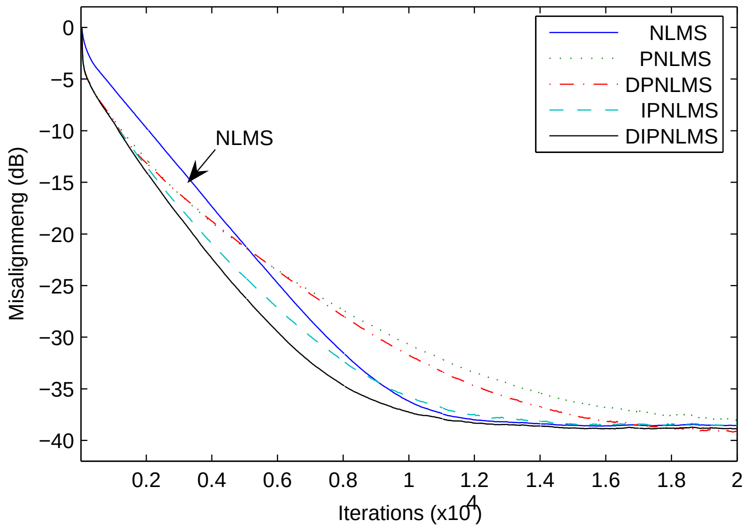

Fig. 7.5 Convergence of proposed approach using non-sparse room impulse response.

not obtain good performance while $M = N$ or $M = 2N$ is a good choice in these situations.

Figure 7.5 compares the convergence speed of the proposed algorithms using a non-sparse room impulse response illustrated in Fig. 7.1(b). For the proposed DPNMS and DIPNLMS, $M = 2N$. It can be seen that PNLMS shows lower convergence speed than NLMS in this case. However, DPNLMS presents faster convergence than PNLMS. For non-sparse impulse response, IPNLMS demonstrates faster convergence than NLMS and the proposed DIPNLMS algorithm also further improves the convergence speed of IPNLMS. In summary, the proposed approach demonstrates as fast initial convergence speed as the original approach, but it can improve the overall convergence speed for both sparse impulse response and non-sparse impulse response.

7.5 Conclusions

By assigning a large proportionate step gain for large coefficient, proportionate adaptive algorithms greatly improve their convergence speed for sparse impulse response. However, this method also causes low convergence speed of small coefficients. In this chapter, a novel method is proposed to determine proportionate step gain according to the difference of current estimate of adaptive filter and its optimal value. This approach distributes large adaptation gain to these coefficients that tend to be large. Furthermore, it automatically adjusts the proportionate step gain of small coefficients after large coefficients have converged so that the convergence in the second period is improved. Simulation results have verified the effectiveness of the proposed approach.

Chapter 8

Conclusions and Future Directions

8.1 Conclusions of This Dissertation

With increase in the demand for higher quality communication, a kind of long adaptive filters are frequently encountered in practical applications, such as the network echo cancellation and acoustical echo cancellation. Conventional adaptive algorithms severely suffer from 1) low convergence speed, 2) increase of computational complexity and 3) degrade of estimate accuracy, with this kind of long adaptive filters, especially for correlated input signals. Much efforts have been made to design new adaptive algorithms to improve the performance of the adaptive filter with hundreds or thousands of coefficients.

Proportionate adaptive algorithms are recently received considerable attention in adaptive filtering field. This new paradigm is based on the fact that most of these long impulse responses are *sparse* in nature, i.e., only a small fraction of coefficients are active and most of the others are zeros. Conventional adaptive algorithms assign same step-size parameter to all coefficients. As a result, the large coefficients require more iterations to converge than the small ones. To accelerate the convergence of large coefficients, larger step-size should be assigned to these large coefficients, which results in proportionate adaptation. The idea behind the PNLMS algorithm is to update each coefficient of the filter individually by assigning each coefficient a step-size parameter proportional to its estimated magnitude. Various modifications have been proposed to improve it.

In this dissertation, I concentrate on improvement of various proportionate algorithms, in order to accelerate their convergence speed, decrease the steady-state misalignment and to reduce their computational complexity.

Chapter 1 introduces the background of this research and the objective of this dissertation.

In Chapter 2, the proportionate adaptation is discussed in a whole framework and various existing proportionate adaptive algorithms are reviewed in detail.

Chapter 3 and chapter 4 concentrate on convergence speed improvement of proportionate adaptive algorithms. In chapter 3, measurement of impulse response sparsity is incorporated into proportionate algorithms, IPNLMS and MPNLMS, to improve their convergence for various impulse responses with different sparsity. The proposed approach adaptively detects the sparsity of target impulse response, and then changes the performance of proportionate adaptive algorithms by adjusting the parameter β to achieve fast convergence with respect to the sparsity of impulse response. In chapter 4, several approaches are proposed to improve the favorable SPNLMS algorithm. At first, a derivation of optimal proportionate adaptation gain is provided in a straightforward and simple way. In the original derivation, it is assumed that $\sum_{i=0}^{N-1} g_i(t) = N$. This is unnecessary and the new derivation withdraws this assumption so that the derivation becomes simple and easy to understand. Then, a convergence criterion is established after the analysis of the convergence characteristics, then a method is proposed to determine the important parameter μ. In order to retain the fast convergence speed, a refined segment function is proposed to approximate the original mu-law function as exact as possible. At last, some methods are proposed to reduce the computational complexity of proportionate NLMS algorithms.

Chapter 5 and chapter 6 concentrate on improvement of convergence quality of proportionate adaptive algorithms using variable step-size technique. In Chapter 5, a non-parametric variable step-size approach is proposed to improve convergence quality of the proportionate NLMS algorithms. Its basic idea comes from a fact that, an optimal global step size can be obtained by forcing the *a posteriori* error not to be zero as in the conventional methods, but to be the disturbance signal, because canceling the *a posteriori* error will introduce additional noise into the coefficient updating when the background noise, or disturbance signal, is presence. The proposed approach provides both fast convergence speed and low steady-state misalignment. This approach is extended into the proportionate APA in chapter 6 to improve the estimate accuracy for correlated input signals. In order to apply the non-parametric variable step-size approach for the PAPAs, a diagonal global step-size matrix $\boldsymbol{\alpha}(t)$ is introduced into. Based on the similar inspiration to VSS-SPNLMS in chapter 5: forcing the *a posteriori* error vector not to be zero as but to be the disturbance signal vector, a non-parametric variable step-size approach is obtained, which is independent with the proportionate step-size control matrix $\mathbf{G}(t)$. Simulation results have shown that the proposed VSS-SPNLMS algorithm and the VSS-SPAPA outperform conventional adaptive algorithms for sparse impulse response. The main advantage of this non-parametric variable step-size approach is, it only requires an power estimate of the disturbance signals, which can be easily obtained in practical application. Furthermore, it can tolerate a relatively large estimate error of power estimate of the disturbance signals. This makes it very preferable in practical application.

Chapter 7 proposes a novel perspective for proportionate adaptive algorithms. All existing proportionate adaptive algorithms exploit the shape of the target impulse response to decide the proportionate step gain for every coefficient. This is not always suitable and it is the reason why existing proportionate adaptive algorithms exhibit slow convergence after the fast initial period and for non-sparse impulse response. Using a block-style analysis method, it reveals that, the optimal proportionate step gain should be determined according to the difference between the current estimate and its optimal value of each coefficient, $|\mathbf{h} - \mathbf{w}(t)|$. Inspired by this, it is proposed to calculate the proportionate step gain according to the difference between the current estimate of adaptive filter and its certain past estimate $|\mathbf{w}(t) - \mathbf{w}(t-M)|$, instead of according to the shape of the target impulse response $|\mathbf{h}|$. This approach shows a interesting property that it assigns large proportionate step gain for the coefficients that tend to be large. It is obvious that in the initial period this method has similar behavior with the PNLMS algorithm or the IPNLMS algorithm. After that, it behaves more like the NLMS algorithm. Consequently, it has faster convergence than the original algorithms in the second period. Therefore, its convergence speed is accelerated considerably. Furthermore, this method can improve the convergence of proportionate NLMS algorithms after the fast initial period, since in the second period the small coefficients obtain relatively fair step gain. The proposed approach is also effective for non-sparse impulse response. Because for a non-sparse impulse response, the proposed approach can automatically assign relatively same proportionate step gain for all coefficients.

8.2 Possible Directions for Future Research

Proportionate adaptive algorithms demonstrate favorable performance than conventional adaptive algorithms for sparse impulse response. In the future, the following issues can be proposed:

First, analysis of convergence characteristics. Although the behavior of proportionate adaptive algorithms is complicated, a elegant analysis was provided in [8] for white Gaussian input signals. In [8], it is stated that a global step-size parameter α less than $2/3$ is necessary to assure the PNLMS algorithm stable. However, many simulations for sparse impulse response have shown that even with a relatively large global step-size parameter, $\alpha = 1.0$ for example, the PNLMS algorithm can converge. The analysis in [8] exploits the so-called "slow adaptation" assumption, that is to assume a very small global step-size parameter, $0 < \alpha \ll 1$, so that the relevant signals or components can be regarded as independent each other in order to simplify the analysis. Although the result is useful for an ordinary α, the underlying behavior is different from the analysis process under "slow adaptation" assumption. This phenomena could be studied. Furthermore, with some reasonable assumptions, it is possible to analyze the performance of proportionate adaptive algorithms for non-Gaussian signals.

Second, transform domain proportionate adaptive algorithms. Although this dissertation concentrates on time domain adaptive algorithms, it is possible to apply the philosophy of proportionate adaptation into transform domain adaptive algorithms. For the correlated input signals, the convergence of proportionate adaptive algorithms can be improved by whitening, or so-called uncorrelation, the input signal, i.e., to make the eigenvalue spread of the input signal's auto-correlation matrix smaller. The affine projection algorithm partly uncorrelates the input using the past P input vectors. Another method is to perform a unitary transform on the input signal. The typical transforms include discrete cosine transform (DCT), discrete Fourier transform (DFT) and wavelet transform. Not all proportionate adaptive algorithms can be smoothly extended into transform domain. A sparse impulse response is not sparse anymore after some unitary transforms. This will be the main concern of the extension. Using block adaptation and DFT, the frequency domain proportionate adaptive algorithms are expected to improve convergence for correlated input, especially for speech. The main advantages of frequency domain adaptive algorithms include: saving computation, exploiting spectral information of correlated input signals to improve performance. Its main disadvantage is introducing delay into the adaptation process. It is expected that applying proportionate adaptation into frequency domain adaptive algorithms can achieve faster convergence with lower computational complexity.

Third, computational complexity reduction methods based on variable step-size technique. Several computational complexity reduction methods have been proposed in this dissertation. They aim to remove redundant computation at each iteration, or partly update coefficient vector without degradation of convergence speed using selective partial-updating approach. Another approach, that is to halt the adaptation process totally when the output error is in a bounded range, as the set-membership filtering algorithms do, is promising to considerably reduce computational complexity. However, the set-membership filtering adaptive algorithms show limited steady-state misalignment improvement and relatively slow convergence. The reason is, $|e(t)|$ is not a good convergence criterion. In the non-parametric variable step-size approach, an estimate of power of $e(t)$, $\sigma_e^2(t)$, is exploited. This estimate is more accurate than $|e(t)|$ to measure how close the adaptive filter $\mathbf{w}(t)$ is near to its optimal value $\mathbf{h}$. It is expected that this method can present better performance than the set-membership filtering adaptive algorithms. There are two approaches can be proposed to halt the adaptation process in order to reduce computational complexity greatly. The first one is to replace $|e(t)|$ in the set-membership filtering adaptive algorithms with $\sigma_e(t)$. The second one is to halt the adaptation process in the non-parametric variable step-size proportionate when $\alpha(t)$ is less than a predetermined threshold α_s. Along this direction, more works is required to verify the effectiveness of these two methods, and to evaluate their performance on computation reduction.

Fourth, study on the proposed novel proportionate step gain. The optimal proportionate step gain should be determined according to the difference between the current estimate and its optimal value of each coefficient. Because the optimal value of each coefficient is not available, it is proposed in Chapter 7 to approximately calculate the proportionate step gain according to the difference between the current estimate of adaptive filter and its certain past estimate. Simulation results have shown the effectiveness of this method. In the future, it is proposed to investigate it in detail. It is obvious that, more accurate an approximation of $|\mathbf{h} - \mathbf{w}(t)|$ is, better performance improvement could be achieved. Although the approximation $|\mathbf{w}(t) - \mathbf{w}(t-M)|$ proposed in Chapter 7 has demonstrated good performance, it is possible to find more approaches to do this work. Since this method is even effective for non-sparse impulse response, it is proposed to investigate the performance of this method into some other applications, where the underlying impulse response generally is not as sparse as the network echo path impulse response, such as in wireless channel equalization, active noise control, and so on.

Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I want to thank the Kochi University of Technology (KUT) for giving me permission to commence this dissertation as a Special Scholarship Program (SSP) student, to do the necessary research work and to use many research resources. I am deeply indebted to my supervisor, Prof. Masahiro FUKUMOTO, whose help, stimulating suggestions and encouragement helped me in all the time of research and writing of this dissertation. At the same time, I want to appreciate Prof. Shiyong ZHANG and Prof. Yipong ZHONG in Fudan University, China, for their persistent warmhearted support and help in the passed four years.

I am grateful to all others who helped me to revise parts of this thesis, particularly at the end of my dissertation: Prof. Kazunori SHIMAMURA , Prof. Makoto IWATA, Prof. Masanori HAMAMURA and Dr. Shinichi YOSHIDA, as member of my supervision group. Thanks to all of the member of Department of Information Systems Engineering in KUT and the guests of final defense committee, thanks for their many valuable comments and advice. I'm also very grateful for the guidance of Prof. Takao NISHITANI and Prof. Yoshimasa KIMURA.

My colleagues in Fukumoto Lab. supported me in my research work. I want to thank them for all their help, support, interest and valuable hints. Especially I am obliged to Dr. Sachio SAIKI, Dr. Eiji FUKUTOMI, Mr. Kouichirou KANAI and all the warmhearted friends.

I am grateful to the members at International Relationship Center of KUT, Prof. Mikiko BAN, Mr. Motoi YOSHIDA, Ms. Kimi KIYOOKA, Ms. Mariko KUBO, for their persistent support on my life and study in Japan. I am grateful to the secretaries in the Department of Information Systems Engineering, and librarians in KUT, for helping the departments to run smoothly and for assisting me in many different ways. I wish to thank all of my best friends in KUT, Dr. Miao SONG, Dr. Peiqian LIU, Dr. Yongkui LI, and so on, for helping me get through the difficult time, and for all the emotional support, entertainment, and caring they provided.

Finally, I wish to express my love and gratitude to all my family. I'd particularly like to thank my mother for never advising me to quit this project. They had more faith in me than could ever be justified by logical argument. Last but not least, I thank my dearest wife - you told me the truth about love. Your patient love enabled me to complete this work.

List of Symbols

Bold lowercase letter: column vector.

Bold uppercase letter: matrix.

$E\{\cdot\}$	:	Expectation operation.
$\{\cdot\}^T$	:	Transposition operation.
t	:	time index.
N	:	Number of adaptive filter coefficients.
$\mathbf{w}(t)$	:	Adaptive filter coefficient vector, $\mathbf{w}(t) = [w_0(t),\ w_1(t),\ \cdots,\ w_{N-1}(t)]^T$.
$\mathbf{h}$	:	Unknown impulse response, $\mathbf{h} = [h_0,\ h_1,\ \cdots,\ h_{N-1}]^T$.
$\tilde{\mathbf{w}}(t)$	:	Adaptive filter coefficient error vector, $\tilde{\mathbf{w}}(t) = \mathbf{h} - \mathbf{w}(t)$.
$x(t)$	:	Input signal.
$\mathbf{x}(t)$	:	Input vector, $\mathbf{x}(t) = [x(t)\ x(t-1)\ \cdots\ x(t-N+1)]^T$.
$\mathbf{R}(t)$	:	Input correlation matrix, $\mathbf{R}(t) = E\{\mathbf{x}(t)\mathbf{x}^T(t)\}$.
$y(t)$	:	Output of adaptive filter, $y(t) = \mathbf{x}^T(t)\mathbf{w}(t)$.
$d(t)$	:	Desired signal of adaptive filter.
$v(t)$	:	Disturbance signal.
$e(t)$	:	Error signal of an adaptive filter, $e(t) = d(t) - y(t)$.
P	:	Projection order of APA.
p	:	p-th element of projection order P. $p = 0,\ \cdots\ P-1$.
α	:	Constant global step-size parameter.
$\alpha(t)$	:	Time-varying global step-size parameter.
$\boldsymbol{\alpha}(t)$	:	Time-varying global step-size control matrix. $\boldsymbol{\alpha}(t) = \mathrm{diag}\{\alpha_0(t),\ \cdots,\ \alpha_{P-1}(t)\}$.
$\mathbf{G}(t)$	:	Proportionate step-size control matrix, $\mathbf{G}(t) = \mathrm{diag}\{g_0(t),\ g_1(t),\ \cdots,\ g_{N-1}(t)\}$.

$\mathbf{X}(t)$	:	Input matrix: $\mathbf{X}(t) = [\mathbf{x}(t)\ \mathbf{x}(t-1) \cdots \mathbf{x}(t-P+1)]$.
$\mathbf{d}(t)$	:	Vector with recent P desired signal samples, $\mathbf{d}(t) = [d(t)\ d(t-1) \cdots d(t-P+1)]^T$.
$\mathbf{v}(t)$	:	Vector with recent P disturbance samples, $\mathbf{v}(t) = [v(t)\ v(t-1) \cdots v(t-P+1)]^T$.
θ	:	Regularization parameter.
ϵ	:	A small positive number, or an objective convergence criterion of MPNLMS.
λ	:	Forgetting factor, or Lagrange multiplier.
$\boldsymbol{\lambda}$	:	Diagonal matrix of Lagrange multiplier. $\boldsymbol{\lambda} = \text{diag}\{\lambda_0, \cdots, \lambda_{P-1}\}$.
$\|\|\mathbf{x}\|\|_2^2$	:	Euclidean norm of vector $\mathbf{x}$, $\|\|\mathbf{x}\|\|_2^2 = \mathbf{x}^T\mathbf{x}$.
$\|\|\mathbf{x}\|\|_G^2$	:	Weighted Euclidean norm of vector $\mathbf{x}$, $\|\|\mathbf{x}\|\|_G^2 = \mathbf{x}^T\mathbf{G}\mathbf{x}$.
β	:	Adjustable parameter of the IPNLMS algorithm.
$\beta(t)$	:	Time-varying adjustable parameter of IPNLMS and IMPNLMS.

Appendix A

Related Mathematical Results

A.1 Matrix Inversion Lemma

If $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{A}+\mathbf{BCD}$ are full rank matrix, then

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}. \tag{A.1}$$

According to the matrix inversion lemma, replacing $\mathbf{A}=\mathbf{R}$, $\mathbf{B}=\mathbf{x}$, $\mathbf{D}=\mathbf{x}^T$ and $\mathbf{C}=1$, yields

$$(\mathbf{R}+\mathbf{xx}^T)^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1}\mathbf{xx}^T\mathbf{R}^{-1}}{1+\mathbf{x}^T\mathbf{R}^{-1}\mathbf{x}}. \tag{A.2}$$

This result is used in derivation of the RLS algorithm.

A.2 Gradient with Respect to Vector

Let $\mathbf{w} = [w_1,\ w_2,\ \cdots, w_N]^T$ is a column vector, $\mathcal{J}(\mathbf{w})$ is a real function of $\mathbf{w}$. The gradient is a vector defined as:

$$\nabla_{\mathbf{w}}\mathcal{J}(\mathbf{w}) = \nabla_{\mathbf{w}}\mathcal{J}(w_1, w_2, \cdots, w_N) = \left[\frac{\partial \mathcal{J}}{\partial w_1}, \frac{\partial \mathcal{J}}{\partial w_2}, \cdots, \frac{\partial \mathcal{J}}{\partial w_N}\right]^T. \tag{A.3}$$

Let $\mathbf{x}$ is an $N\times 1$ column vector, $\mathbf{A}$ is $N \times N$ matrix, according the definition above, the following identities hold:

$$\nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{x}) = \mathbf{x}, \tag{A.4}$$

$$\nabla_{\mathbf{w}}(\mathbf{x}^T\mathbf{w}) = \mathbf{x}^T, \tag{A.5}$$

$$\nabla_{\mathbf{w}}(\mathbf{w}^T\mathbf{A}\mathbf{w}) = 2\mathbf{A}\mathbf{w}. \tag{A.6}$$

This results are widely used in derivation of normalized proportionate adaptive algorithms and proportionate affine projection algorithms.

References

[1] B. Widrow and S.D. Stearns, Adaptive signal processing, Prentice-Hall, Upper Saddle River, NJ, 1985.

[2] S. Haykin, Adaptive Filter Theory, 4th ed, Prentice-Hall, Upper Saddle River, NJ, 2003.

[3] P.S.R. Dinz, Adaptive filtering: algorithms and practical implementation, Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[4] Y. Huang, J. Benesty, and J. Chen, Acoustic MIMO Signal Processing (Signals and Communication Technology), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] ITU-T, "ITU-T recommendation G.168: Digital network echo cancellers," 2007.

[6] P. Loganathan, A.W.H. Khong, and P.A.A. Naylor, "Sparseness controlled proportionate algorithm for acoustic echo cancellation," Proc. European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, pp.1–5, Aug 2008.

[7] A.H. Sayed, Fundamentals of Adaptive Filtering, Wiley, New York, 2003.

[8] D.L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," IEEE Trans. Speech Audio Processing, vol.8, no.5, pp.508–518, Sept. 2000.

[9] Y. Huang, J. Benesty, and J. Chen, Sparse Adaptive Filters, ch. 4, Springer-Verlag New York Inc., Secaucus, NJ, USA, 2005.

[10] R. Martin, W. Sethares, R. Williamson, and J. Johnson, C.R., "Exploiting sparsity in adaptive filters," IEEE Trans. Signal Processing, vol.50, no.8, pp.1883–1894, Aug. 2002.

[11] Z. Chen, S. Haykin, and S.L. Gay, Proportionate Adaptation: New Paradigms in Adaptive Filters, ch. 8, Wiley, New York, 2005.

[12] L. Liu, M. Fukumoto, and S. Saiki, "An improved mu-law proportionate NLMS algorithm," Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP'08), Las Vegas, USA, pp.3797–3800, Mar 2008.

[13] L. Liu, M. Fukumoto, and S. Zhang, "A variable parameter improved proportionate normalized lms algorithm," Proc. IEEE Asia Pacific Conference On Circuits And Systems (APCCAS'08), Macao, China, pp.201–204, Nov 2008.

[14] L. Liu, M. Fukumoto, and S. Zhang, "A variable step-size proportionate NLMS algorithm," Proc. IEEE International Symposium on Circuits and Systems (ISCAS'09), Taipei, Taiwan, pp.2045–2048, May 2009.

[15] L. Liu, M. Fukumoto, and S. Saiki, "Better approximation of segment proportionate normalized LMS algorithm," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., Submitted for publication.

[16] L. Liu, M. Fukumoto, S. Saiki, and S. Zhang, "A variable step-size proportionate NLMS algorithm for identification of sparse impulse response," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., Conditional accepted.

[17] L. Liu, M. Fukumoto, S. Saiki, and S. Zhang, "A variable step-size proportionate affine projection algorithm for network echo cancellation," Proc. 16th International Conference on Digital Signal Processing (DSP2009), Santorini, Greece, pp.1–6, Jul. 2009.

[18] L. Liu, M. Fukumoto, S. Saiki, and S. Zhang, "A variable step-size proportionate affine projection algorithm for identification of sparse impulse response," EURASIP J. Advances in Signal Process., vol.2009, no.10, pp.1–12, 2009.

[19] L. Liu, M. Fukumoto, and S. Saiki, "A novel proportionate normalized least mean-square algorithm," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., Submitted for publication.

[20] S. Gay and S. Tavathia, "The fast affine projection algorithm," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '95), Detroit, Michigan, pp.3023–3026, May 1995.

[21] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "Fast projection algorithm and its step size control," Proc. of IEEE International Conference on Acoustics, Speech, and Signal

Processing, (ICASSP'95), Washington, DC, USA, pp.945–948, IEEE Computer Society, 1995.

[22] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "A fast projection algorithm for adaptive filtering," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol.E78-A, no.10, pp.1355–1361, Oct. 1995.

[23] S.L. Gay, The fast affine projection algorithm, pp.23–46, Kluwer Academic Publishers, Hingham, MA, USA, 2000.

[24] D.L. Duttweiler, "Subsampling to estimate delay with application to echo canceling," IEEE Trans. Acoust., Speech, Signal Processing, vol.31, no.5, pp.1090–1099, Oct 1983.

[25] Y. Cheng and D.M. Etter, "Analysis of an adaptive technique for modeling sparse systems," IEEE Trans. Acoust., Speech, Signal Processing, vol.37, no.2, pp.254–263, Feb 1989.

[26] O. Tanrikulu, "Method and apparatus for fast network bulk delay and span estimation for line echo cancellation," August 2004.

[27] J. Homer, "Detection guided NLMS estimation of sparsely parametrized channels," IEEE Trans. Circuits Syst. II, vol.47, no.12, pp.1437 – 1442, Dec. 2000.

[28] J. Homer, I. Mareels, and C. Hoang, "Enhanced detection-guided NLMS estimation of sparse FIR-modeled signal channels," IEEE Trans. Circuits Syst. I, vol.53, no.8, pp.1783–1791, Aug. 2006.

[29] S. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," Proc. of the 32nd Asilomar Conference on Signals, Systems and Computers, (ACSSC'98), Pacific Grove, Calif, USA, pp.394–398, Nov 1998.

[30] J. Benesty and S.L. Gay, "An improved PNLMS algorithm," Proc. IEEE International Conference on Acoustic, Speech and Signal Processing, (ICASSP'02), Orlando, USA, pp.1881–1884, May 2002.

[31] M. Nekuii and M. Atarodi, "A fast converging algorithm for network echo cancellation," IEEE Signal Processing Lett., vol.11, no.4, pp.427–430, April 2004.

[32] J. Cui, P. Naylor, and D. Brown, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04), Montreal, Quebec, Canada, pp.141–144, May 2004.

[33] H. Deng and M. Doroslovački, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," IEEE Signal Processing Lett., vol.12, no.3, pp.181– 184, March 2005.

[34] H. Deng and M. Doroslovachi, "Proportionate adaptive algorithms for network echo cancellation," IEEE Trans. Signal Processing, vol.54, no.5, pp.1794 – 1803, May 2006.

[35] B. Jelfs and D. Mandic, "A unifying approach to the derivation of the class of PNLMS algorithms," Proc. 15th International Conference on Digital Signal Processing (DSP'07), London, UK, pp.35–38, July 2007.

[36] P. Mahale, "SVS-PNLMS algorithm for acoustic echo cancellation," Proc. of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008., pp.1–6, Apr 2008.

[37] S. Gay and S. Douglas, "Normalized natural gradient adaptive filtering for sparse and non-sparse systems," Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '02), Orlando, USA, pp.1405–1408, May 2002.

[38] M. Doroslovaki and H. Deng, "On convergence of proportionate-type NLMS adaptive algorithms," Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'06), Toulouse, Franc, pp.105–108, Mar 2006.

[39] H. Deng, Adaptive algorithms of sparse impulse response identification, d.sc dissertation, Dept. of Electrical and Computer Engineering, George Washington University, Washington, DC, Mar 2005.

[40] H. Deng and M. Doroslovački, "Wavelet-based mpnlms adaptive algorithm for network echo cancellation," EURASIP J. Audio Speech Music Process., vol.2007, no.1, pp.16–16, Jan. 2007.

[41] M.R. Petraglia and G. Barboza, “Improved PNLMS algorithm employing wavelet transform and sparse filters,” Porc. 16th European Signal Processing Conference (EUSIPCO2008), Lausanne, Switzerland, pp.1–5, Aug 2008.

[42] M. Yukawa, “Krylov-proportionate adaptive filtering techniques not limited to sparse systems.,” IEEE Trans. Signal Processing, vol.57, no.3, pp.927–943, 2009.

[43] S. Douglas, “Adaptive filters employing partial updates,” IEEE Trans. Circuits Syst. II, vol.44, no.3, pp.209–216, Mar 1997.

[44] T. Aboulnasr and K. Mayya, “Selective coefficient update of gradient-based adaptive algorithms,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’97), Munich, Germany, pp.1929–1932, Apr. 1997.

[45] O. Tanrkulu, B. Baykal, A.G. Constantinides, and J.A. Chambers, “Residual echo signal in critically sampled subband acoustic echo cancelers based on IIR and FIR filterbanks,” IEEE Signal Processing Lett., vol.45, pp.901–912, 1997.

[46] T. Schertler, “Selective block update of NLMS type algorithms,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’98), Seattle, WA, pp.1717–1720, May 1998.

[47] T. Aboulnasr and K. Mayyas, “Complexity reduction of the NLMS algorithm via selective coefficient update,” IEEE Trans. Signal Processing, vol.47, no.5, pp.1421–1427, May 1999.

[48] K. Dogancay and O. Tanrkulu, “Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’00), Istanbul, Turkey, pp.448–451, June 2000.

[49] K. Dogancay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial updates,” IEEE Trans. Circuits Syst. II, vol.48, no.8, pp.762–769, Aug 2001.

[50] A. Tandon, M.N.S. Swamy, and M.O. Ahmad, “Partial-update L_∞-norm based algorithms,” IEEE Trans. Circuits Syst. I, vol.54, no.2, pp.411–419, Feb 2007.

[51] K. Dogancay and O. Tanrikulu, “Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP’00), pp.448–451, May 2000.

[52] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.F. Huang, “Set-membership adaptive equalization and updator-shared implementation for multiple channel communications systems,” IEEE Trans. Signal Processing, vol.46, no.9, pp.2372–2384, Sep 1998.

[53] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.F. Huang, “Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size,” IEEE Signal Processing Lett., vol.5, no.5, pp.111–114, May 1998.

[54] S. Werner, M.L.R. de Campos, and J.A. Apolinario, “Data-selective constrained affine projection algorithm,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’01), Salt Lake City, Uta, pp.3745–3748, May 2001.

[55] P.S.R. Diniz and S. Werner, “Set-membership binormalized data-reusing algorithms,” IEEE Trans. Signal Processing, vol.51, no.1, pp.124–134, Jan 2003.

[56] S. Werner and P.S.R. Diniz, “Set-membership affine projection algorithm,” IEEE Signal Processing Lett., vol.8, no.8, pp.231–235, Aug 2001.

[57] S. Werner, J.A.A. Jr., M.L.R. de Campos, and P.S.R. Diniz, “Low-complexity constrained affine projection algorithms,” IEEE Trans. Signal Processing, vol.53, no.12, pp.4545–4555, Dec 2005.

[58] O. Tanrikulu and K. Dogancay, “Selective-partial-update proportionate normalized least-mean-squares algorithm for network echo cancellation,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP’02), Orlando, FL, USA, pp.1889–1892, May 2002.

[59] O. Hoshuyama, R. Goubran, and A. Sugiyama, “A generalized proportionate variable step-size algorithm for fast changing acoustic environments,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’04), Montreal, Quebec,

Canada, pp.161–164, May 2004.

[60] J.D. Gordy, T. Aboulnasr, and M. Bouchard, “Reduced-complexity proportionate NLMS employing block-based selective coefficient updates,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’08), Las Vegas, USA, pp.233–236, Mar. 2008.

[61] H. Deng and M. Doroslovacki, “New sparse adaptive algorithms using partial update,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP ’04), pp.845–8, May 2004.

[62] S. Werner, J.A. Apolinario, and P.S.R. Diniz, “Set-membership proportionate affine projection algorithms,” EURASIP J. Audio, Speech, and Music Process., vol.2007, no.1, pp.1–10, 2007.

[63] A.W.H. Khong and P.A.A. Naylor, “Efficient use of sparse adaptive filters,” Proc. 40th Asilomar Conf. on Signals, Systems and Computers (ACSSC ’06), Pacific Grove, Calif, USA, pp.1375–1379, Oct-Nov 2006.

[64] T. Gansler, J. Benesty, S.L. Gay, and M.M. Sondhi, “A robust proportionate affine projection algorithm for network echo cancellation,” Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP’00), pp.793–796, May 2000.

[65] P.O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” J. Mach. Learn. Res., vol.5, pp.1457–1469, 2004.

[66] A.W.H. Khong, P.A. Naylor, and J. Benesty, “A low delay and fast converging improved proportionate algorithm for sparse system identification,” EURASIP J. Audio Speech Music Process., vol.2007, no.1, pp.13–20, 2007.

[67] T. Gansler, J. Benesty, M.M. Sondhi, and S.L. Gay, “Dynamic resource allocation for network echo cancellation,” Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP’01), Salt Lake City, UT, USA, pp.3233–3236, May 2001.

[68] R. Kwong and E. Johnston, “A variable step size LMS algorithm,” IEEE Trans. Signal

Processing, vol.40, no.7, pp.1633–1642, July 1992.

[69] A. Mader, H. Puder, and G.U. Schmidt, “Step-size control for acoustic echo cancellation filter - an overview,” Signal Processing, vol.80, no.9, pp.1697–1719, Sept. 2000.

[70] A.I. Sulyman and A. Zerguine, “Convergence and steady-state analysis of a variable step-size NLMS algorithm,” Signal Processing, vol.83, no.6, pp.1255–1273, 2003.

[71] J. Benesty, H. Rey, L.R. Vega, and S. Tressens, “A nonparametric VSS NLMS algorithm,” IEEE Signal Processing Lett., vol.13, no.10, pp.581–584, Oct 2006.

[72] C. Paleologu, S. Ciochina, and J. Benesty, “Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation,” IEEE Signal Processing Lett., vol.15, no.10, pp.5–8, Jan. 2008.

[73] C. Paleologu, J. Benesty, and S. Ciochina, “A variable step-size affine projection algorithm designed for acoustic echo cancellation,” IEEE Trans. Acoust., Speech, Signal Processing, vol.16, no.8, pp.1466–1478, Nov. 2008.

[74] H.C. Shin, A.H. Sayed, and W.J. Song, “Variable step-size NLMS and affine projection algorithms,” IEEE Signal Processing Lett., vol.11, no.2, pp.132–135, Feb. 2004.

[75] S. Saiki and M. Fukumoto, “Improvement of the IPNLMS algorithm exploiting variable step size,” IEICE Tech. Rep., vol.107, no.180, pp.25–28, Aug 2007.

[76] J. Tanpreeyachaya, I. Takumi, and M. Hata, “Performance improvement of variable step-size NLMS,” IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol.E78-A, no.8, pp.905–914, Aug. 1995.

Appendix B

Related Publications

B.1 Articles in or submitted to refereed journals

1. Ligang Liu, Masahiro Fukumoto, Sachio Saiki, and Shiyong Zhang. A Variable Step-size Proportionate Affine Projection Algorithm for Identification of Sparse Impulse Response. *EURASIP J. on Advances in Signal Process.*, To appear soon.
2. Ligang Liu, Masahiro Fukumoto, Sachio Saiki, and Shiyong Zhang. A Variable Step-size Proportionate NLMS Algorithm for Identification of Sparse Impulse Response. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Conditionally Accepted.
3. Ligang Liu, Masahiro Fukumoto, and Sachio Saiki. A Novel Proportionate Normalized Least Mean Square Algorithm. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Submitted for publication.
4. Ligang Liu, Masahiro Fukumoto, and Sachio Saiki. Better Approximation of Segment Proportionate NLMS Algorithm. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Submitted for publication.

B.2 Articles in full paper refereed international conference proceedings

1. Ligang Liu, Masahiro Fukumoto and Shiyong Zhang. A variable step-size proportionate affine projection algorithm for network echo cancellation. In *Proc. the 16th Interna-*

tional Conference on Digital Signal Processing (DSP2009), pages 1-6, Aegean island of Santorini, Greece, Jul. 2009

2. Ligang Liu, Masahiro Fukumoto, and Shiyong Zhang. Improvement of the Mu-law Proportionate NLMS Algorithm. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS2009)*, pages 2045-2048, Taipei, Taiwan, May. 2009
3. Ligang Liu, Masahiro Fukumoto, and Shiyong Zhang. A variable parameter improved proportionate normalized LMS Algorithm. In *Proc. IEEE International Conference on Circuts and Systems (APCCAS2008)*, pages 201-204, Macau, China, Dec. 2008
4. Ligang Liu, Masahiro Fukumoto, and Sachio Saiki. An improved mu-law proportionate NLMS algorithm. In *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2008)*, pages 2987-3000, Las Vegas, USA, Mar. 2008
5. Ligang Liu, Masahiro Fukumoto, and Sachio Saiki. A New Structure for Sound Reproduction System. In *Proc. IEEE International Symposium on Circuits and Systems (ISCAS2008)*, pages 1144 - 1147, Seattle, USA, May. 2008

B.3 Articles in local conference proceedings

1. Ligang Liu, Masahiro Fukumoto, and Sachio Saiki. Proportionate adaptative algorithms guided by estimation of channel sparsity. In *Proc. of the 22nd SIP SYMPOSIUM (SIP'07)*, pages 587–591, Sendai, Japan, Nov. 2007
2. Sachio Saiki, Masahiro Fukumoto, and Ligang Liu. An improved VSS-IPNLMS algorithm exploiting adaptive parameter adjustment. In *Proc. of the 23nd SIP SYMPOSIUM (SIP'08)*, pages 207–211, Kanazawa, Japan, Nov. 2008
3. Ligang Liu, Masahiro Fukumoto, Sachio Saiki, Eiji Fukutomi and Shiyong Zhang. A variable parameter IPNLMS algorithm, In *Proc. of the International Conference on Next Era Information Networking, (NEINE'08)*, pages 139-144, Kochi, Japan, Dec. 2008

B.3 Articles in local conference proceedings

4. Sachio Saiki, Ligang Liu, Masahiro Fukumoto. An Improved VSS-IPNLMS Algorithm, In *Proc. of the International Conference on Next Era Information Networking, (NEINE'08)*, pages 151-156, Kochi, Japan, Dec. 2008
5. Kouichirou Kanai and Masahiro Fukumoto and Sachio Saiki and Eiji Fukutomi and Ligang Liu. Movement Detection of the Audient in Sound Field Reproduction, In *Proc. of the International Conference on Next Era Information Networking, (NEINE'08)*, pages 324-326, Kochi, Japan, Dec. 2008
6. Ligang Liu, Masahiro Fukumoto, Sachio Saiki, and Shiyong Zhang. Adaptive inverse filtering algorithms for sound reproduction without model of channel. In *Proc. of the International Conference on Next Era Information Networking, (NEINE'07)*, pages 269–274, Shanghai, China, Sep. 2007
7. Sachio Saiki, Masahiro Fukumoto, and Ligang Liu. A convergence evaluation of the IPNLMS algorithm exploiting variable step-size. In *Proc. of the International Conference on Next Era Information Networking (NEINE'07)*, pages 313–317, Shanghai, China, Sep. 2007
8. Eiji Fukutomi, Ligang Liu, Masahiro Fukumoto, and Akihiro Shimizu. Security evaluation of cryptosystem exploiting variable function. In *Proc. of the International Conference on Next Era Information Networking (NEINE'07)*, pages 256–261, Shanghai, China, Sep. 2007
9. Kouichirou Kanai, Masahiro Fukumoto, Sachio Saiki, and Ligang Liu. Receive the particular zone sound using the correlation. In *Proc. of the International Conference on Next Era Information Networking (NEINE'07)*, pages 493–495, Shanghai, China, Sep. 2007
10. Ligang Liu, Sachio Saiki, Masahiro Fukumoto, and Shiyong Zhang. Cross-talk component cancellation using adaptive inverse filter in sound field reproduction system. In *Proc. of the International Conference on Next Era Information Networking (NEINE'06)*, pages 114–118, Kochi, Japan, Sep. 2006

11. Sachio Saiki, Masahiro Fukumoto, and Ligang Liu. Improvement of convergence accuracy for BCGM-OR algorithm. In *Proc. of the International Conference on Next Era Information Networking (NEINE'06)*, pages 119–124, Kochi, Japan, Sep. 2006

12. Eiji Fukutomi, Ligang Liu, Masahiro Fukumoto, and Akihiro Shimizu. Security of cryptosystems using variable function. In *Proc. of the International Conference on Next Era Information Networking (NEINE'06)*, pages 74–79, Kochi, Japan, Sep. 2006