Adaptive Filter Design for Sparse Signal Estimation

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jie Yang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Gerald E. Sobelman, Advisor

December, 2011

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to my advisor, Professor Gerald E. Sobelman, for his support, guidance and encouragement throughout my Ph.D. study at the University of Minnesota. Without him, this dissertation would not have been possible. I would also like to thank Professor Keshab K. Parhi, Professor David Frank and Professor Mostafa Kaveh for their support as my committee members. Their careful reviews and helpful feedback improve the quality of this work greatly.

Additionally, I would like to thank the Department of Electrical & Computer Engineering for the continuous financial support for my Ph.D. study. The things I learnt through the Ph.D. program will continue to inspire me in the future.

I would also like to thank my supervisor Zigang Yang, my manager Hardik Gandhi, my Buddy Raul Blazquez and all the group members during my intern in Texas Instruments for their numerous helpful discussions contributing to this dissertation.

Finally, I want to give special thanks to my family, for their love, care, encouragement and support all the time. Thank them for always being together with me through happy or hard time.

Abstract

Recently, sparse signal estimation has become an increasingly important research area in signal processing due to its wide range of applications. Efficient adaptive algorithms have been developed for estimation of various sparse signals, and the approaches developed are usually application-specific. In this dissertation, we investigate the algorithm and system design for sparse signal estimation of several applications of practical interest, specifically echo cancellation, compressive sensing, and power amplifier pre-distortion.

For echo cancellation, different approaches are considered to find the optimal solution. A series of algorithms are proposed to improve the performance and reduce the cost. Specifically, we describe novel adaptive tap algorithms with selective update criteria, a μ-law proportionate technique incorporated with efficient memorized proportionate Affine Projection Algorithms, and a new class of proportionate algorithms with gradient-controlled individual step sizes which can be implemented either in the time domain or the frequency domain.

For compressive sensing algorithms with the l_0 norm constraint, a sparse LMS algorithm with segment zero attractors is introduced. It can achieve significant convergence and error performance improvements while providing reduced computational cost, especially for large sparse systems with colored inputs. Such filters can also be combined with cascade or multistage realizations, thereby yielding even more efficient implementations.

We also describe new results for the non-linear signal estimation problem in power amplifier (PA) pre-distortion with dynamic nonlinearities, where the signal can be represented using a Volterra series with sparse coefficients. An efficient solution using a power-indexed look-up table (LUT) based digital pre-distortion (DPD) is proposed to address the current challenge of poor dynamic performance of a PA operating with wideband signals. Experimental results obtained using a 2 GHz power amplifier driven by a 2-carrier WCDMA signal demonstrate very robust and stable performance for the PA in dynamic environments.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Along with developments in digital signal processors, adaptive filters have become more common and are routinely used in devices such as cell phones, digital cameras and medical instruments. Compared with non-adaptive filters, the advantages of an adaptive filter are its self-adjusting ability based on real-time data input, and its iterative solution according to a certain optimization algorithm, such as Least Mean Square (LMS) and Normalized LMS (NLMS) [1-3]. Sparse signal estimation has recently received much attention as an active research area with many potential applications. However, for applications where the system is relatively sparse, LMS and NLMS are not favored due to the high power consumption and low convergence speed. This motivates the design of a more efficient adaptive filter with great convergence performance specially designed for sparse signal recovery problem.

1.1 Sparse Signal Definition & Applications

A sparse signal is one in which most of its weight is distributed on a small number of the total resource. Taking network echo [4-5] as an example, Fig. 1.1 shows a typical

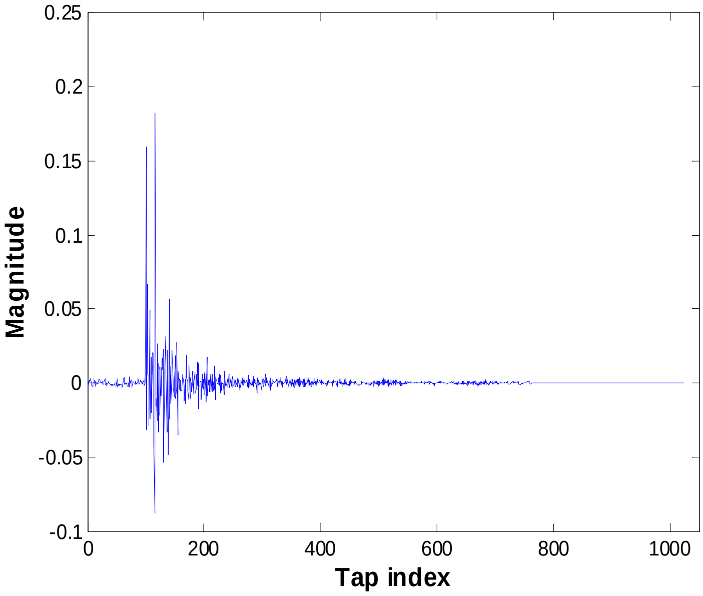


Fig. 1.1: A typical sparse network echo impulse response sampled at 8 KHz.

network echo path impulse response sampled at 8 kHz. The network echo results from the impedance mismatch of the hybrid connected between a four-wire central network circuit and a two-wire local network circuit of a telephone network. The duration of the echo path impulse response is usually 64 to 128 ms, and the active region where the impulse response coefficients have large magnitude, is relatively small. This property makes the network echo a sparse signal.

Acoustic echo [4-5] is another type of echo in voice communication systems. It arises when the terminal's microphone picks up the output of a loudspeaker in the same room, which exists in many communication scenarios such as hands-free car phone systems, cell phones in speaker mode, and conference phone systems. As the network echo shown in Fig. 1.1, acoustic echo is also sparse, but is more dispersive and varies based on multiple external factors like positions and movements of subjects relative to the headset.

Many other real world channels of practical interest exhibit sparse characteristics, such as multipath wireless channels dominated by a relatively small number of significant paths [6-7], frequency selective channels with large delay spread and most energy localized to small regions in delay [8-9], and multicarrier underwater acoustic channels with significant Doppler effects [10]. Recently, compressive sensing [11-13] has emerged as an important area of signal processing, where the signal is highly sparse with only a small number of non-zero elements in the transform domain. Sparse signal processing of compressive sensing has the potential of revealing significant reduction in the sampling rate and processing manipulations which can reduce the implementation cost.

1.2 Background Overview

1.2.1 Echo Cancellation

Echo cancellation [4-5] has been studied for several decades and excellent performance has been achieved by current echo cancellers. However, recent development of hands-free telephone systems, Internet phones, and teleconference systems require further improvements to the technique to increase speech quality and to enhance listening experience.

Adaptive filtering is a commonly used mechanism to suppress echoes in either Network Echo Cancellation (NEC) of the telephone network or Acoustic Echo

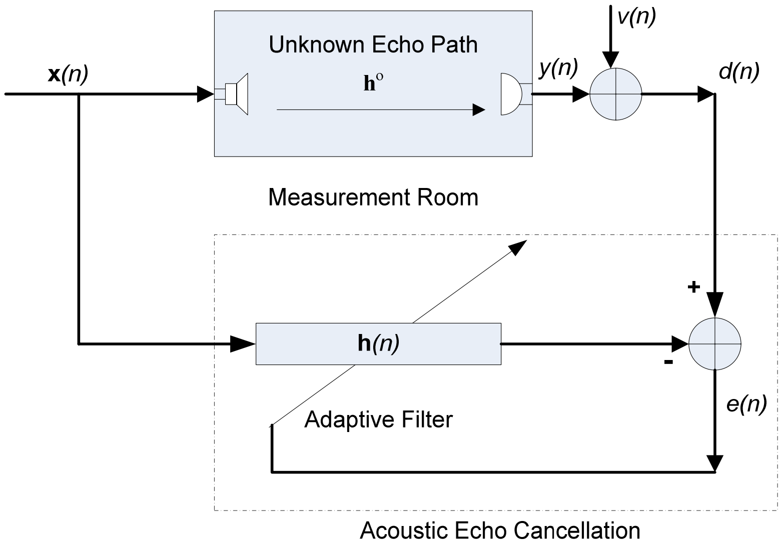


Fig. 1.2: System model for AEC.

Cancellation (AEC) in hands-free audio terminals. Fig. 1.2 shows a general AEC configuration, where an adaptive filter is used to identify the unknown echo path by adaptively adjusting its coefficients. The estimated coefficients are then used to provide a replica of the echoes which can be subtracted from the target signal to achieve cancellation.

Normalized Least Mean Square (NLMS) [1-3] is the most widely used adaptive filter algorithm due to its computational simplicity. Consider the AEC system in Fig. 1.2, and let the input vector at iteration *n*, the unknown echo path impulse response and its estimate at iteration *n* be defined as: $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^H$, $\mathbf{h^o} = [h_0, h_1, \ldots, h_{L-1}]^T$, and $\mathbf{h}(n) = [h_0(n), h_1(n), \ldots, h_{L-1}(n)]^T$, respectively, where L is the length of the adaptive filter. The unknown system output is:

$$y(n) = \mathbf{x}^H(n)\mathbf{h}^o \,. \tag{1.1}$$

With $v(n)$ representing the measurement noise added to $y(n)$, the desired response $d(n)$ of the AEC model is given by:

$$d(n) = y(n) + v(n) = \mathbf{x}^H(n)\mathbf{h}^o + v(n) \,. \tag{1.2}$$

The error signal $e(n)$ is calculated based on the estimated coefficients of the previous iteration $\mathbf{h}(n-1)$:

$$e(n) = d(n) - \mathbf{x}^H(n)\mathbf{h}(n-1) \,. \tag{1.3}$$

The update scheme for NLMS is:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \frac{\mu \mathbf{x}(n)e(n)}{\mathbf{x}^H(n)\mathbf{x}(n) + \delta} \,, \tag{1.4}$$

with step size μ and regularization factor δ.

NLMS is a stochastic gradient descent adaptive algorithm with adaptation based on only the error signal at current iteration. For identification of a long, sparse and rapidly changing acoustic echo response, it has the disadvantages of slow convergence speed and poor tracking ability. This motivated the development of various algorithms that exploit the sparse nature of the signal to achieve better performance. Recent improvements in two important performance indicators of echo cancellers, namely convergence speed and computational complexity, have been achieved by proportionate adaptive algorithms [14]-[18] and a partial update mechanism [19]-[22], respectively. The proportionate algorithms can speed up the convergence rate by individually assigning different step sizes to each filter tap, whereas partial update adaptive filter can reduce power and area consumption by updating only a subset of the filter taps every

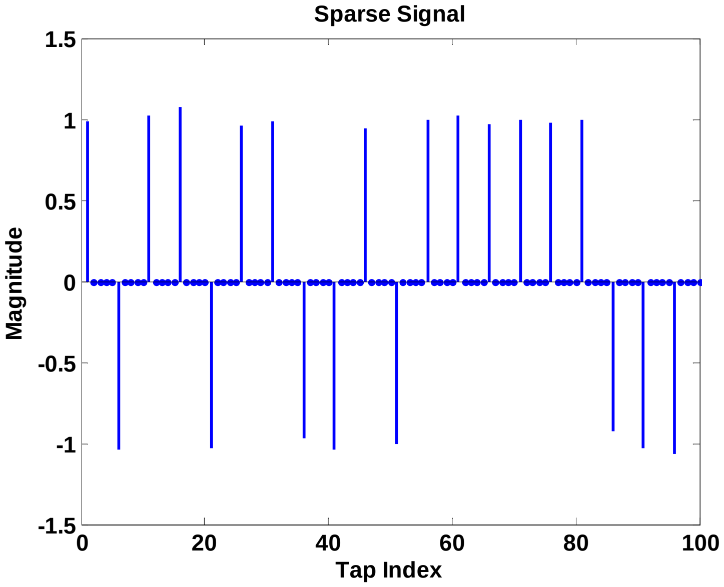


Fig. 1.3: Typical sparse signal in compressive sensing for image reconstruction.

iteration while still maintaining good error performance. These two current techniques will be reviewed in greater detail in the following chapter.

1.2.2 Compressive Sensing

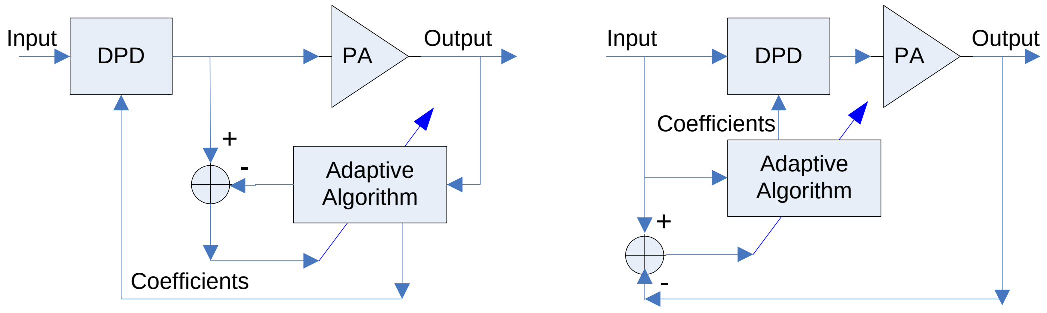
The problem of sparse signal estimation from random measurements has gained increasing interest due to the recently introduced framework of Compressive Sensing [11-13]. It has many promising applications in wireless communications and image signal processing, where highly sparse signals contain sufficient information for approximation or exact recovery. Fig. 1.3 shows a typical signal to recover in an image reconstruction problem, which has only a small number of nonzero coefficients. Generally, identification of these sparse signals requires the minimization of the l_0 norm of the candidate solution **h** as part of the cost function:

$$\min_{\mathbf{h}} \|\mathbf{h}\|_0 . \qquad (1.5)$$

$\|.\|_0$ is the l_0 norm defined as the number of non-zero elements of a vector, which results in zero attractions for all coefficients when minimizing the cost functions. Unfortunately, this optimization problem is often quite difficult to solve because there is a need for an intractable combinatorial search. Alternative approaches have been proposed using good approximations of the l_0 norm, such as the l_1 norm, and various weighted l_1 norm functions, as these are more tractable as sparsity-promoting factors in the optimization process.

1.2.3 Digital Pre-Distortion for Power Amplifiers

Power Amplifiers (PAs) in radio transmitters of wireless communications systems have only a limited operating region [23]. The nonlinearity introduced by the PA can cause severe nonlinear distortion to the output signal and can interference on adjacent radio channel frequencies, especially for a PA operating with wideband signals like those in 3G and LTE systems. To compensate for this non-ideal behavior of the PA and to improve the PA efficiency, various analog linearization techniques such as RF feedback and RF feed-forward have been developed, as described in [23-25]. However, the analog solutions usually lead to considerable increase in cost due to the additional analog network. As an attractive alternative, baseband digital pre-distortion (DPD) [26-28] has been proven to be more efficient and cost effective compared to the analog solutions. Most of the current DPD implementations use amplitude and phase look-up tables (AM-to-AM and AM-to-PM) to accommodate arbitrary input signals [26] to compensate for the instantaneous nonlinear distortion introduced by the PA. However, for communications where the PA has to support wideband input signals, those memory-less models are highly inadequate for achieving good linearization



(a) (b)

Fig. 1.4: DPD structure with (a) indirect learning and (b) direct learning.

performance of the DPD. With the memory effect [29] considered, the Volterra Series [27] and various simplified nonlinear models based on the Volterra Series as presented in [28] have been used for the modeling of PA behavior.

Fig. 1.4 shows a general DPD architecture with indirect learning and direct learning structures. The inverse nonlinear characters of the PA are estimated with an adaptive filter, which are then used to pre-distort the PA input to produce an overall linear system response. In this thesis, our research is based on the recently proposed nonlinear model of a Generalized Memory Polynomial (GMP) [28] which achieves the best performance to date, as demonstrated in [28]. The GMP can be summarized as follows:

$$\begin{aligned} y_{GMP}(n) = \sum_{k=0}^{K_a-1}\sum_{l=0}^{L_a-1} a_{kl}x(n-l)\left|x(n-l)\right|^k + \sum_{k=1}^{K_b}\sum_{l=0}^{L_b-1}\sum_{m=1}^{M_b} b_{klm}x(n-l)\left|x(n-l-m)\right|^k \\ + \sum_{k=1}^{K_c}\sum_{l=0}^{L_c-1}\sum_{m=1}^{M_c} c_{klm}x(n-l)\left|x(n-l-m)\right|^k \end{aligned} \tag{1.6}$$

where k is a nonlinear order index and m is a memory index. The first, second and third terms represent coefficients for aligned signal and envelope, signal and lagging envelope, and signal and leading envelope at time *n*, respectively. First, collect the coefficients, e.g., a_{kl} , b_{klm}, c_{klm} in (1.6) into one $J \times 1$ vector, denoted as $\mathbf{w}$, where J is the total number of coefficients. Next, assemble all the product terms associated with the coefficients, e.g., the products of the aligned signal with envelope, the signal with lagging envelope, and the signal with leading envelope over some period N into a $N \times J$ matrix **X**. Then the model output can be expressed as:

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}\,, \tag{1.7}$$

where $\hat{\mathbf{y}}$ is an $N \times 1$ vector estimate of actual output vector $\mathbf{y}$.

For the inverse modeling method used in Fig. 1.4 (a), we similarly have:

$$\hat{\mathbf{x}} = \mathbf{Y}\mathbf{w}\,. \tag{1.8}$$

Y is generated in the same way as **X**. With reference to Fig. 1.4 (a), the error vector is then given by:

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}\,. \tag{1.9}$$

The least square solution which minimizes $\left\|\mathbf{e}\right\|^2$ can be calculated as follows:

$$\mathbf{w} = \left(\mathbf{Y^H Y}\right)^{\mathbf{-1}} \mathbf{Y^H x}\,. \tag{1.10}$$

In this way, the nonlinear system identification problem can be formulated as a linear system identification problem as described above, and various adaptive algorithms can be used for the coefficient training of the inverse PA model. Fig. 1.5 shows a plot of the converged pre-distorter coefficients for a typical class AB RF PA operating with wideband signals. Since the higher order nonlinearities are always small compared with linear coefficients in the model, the set of coefficients for DPD is also a sparse system, as can be seen in Fig. 1.5.

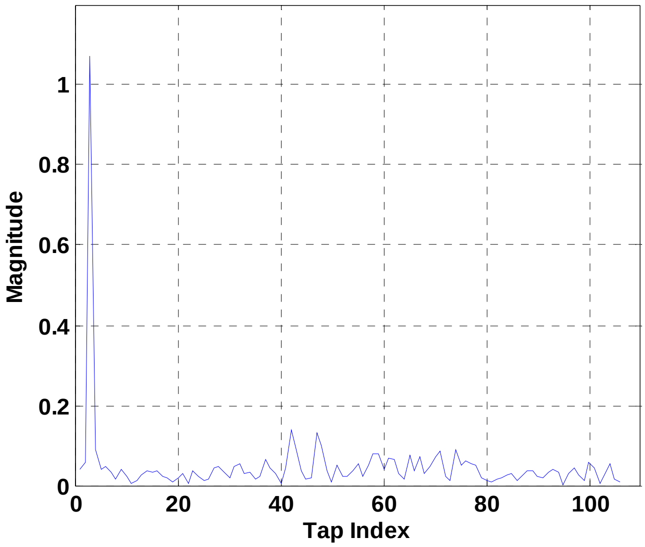


Fig. 1.5: A plot of converged pre-distorter coefficients.

1.3 Summary of Contributions

In this work, a variety of efficient adaptive algorithms are developed for the sparse signal estimation problem as introduced above. The major part of the thesis focuses on echo cancellation, as it was the problem motivating this work and because of its wide range of practical applications. In addition, research has also been conducted in the areas of compressive sensing and nonlinear system identification. All of proposed algorithms can be extended to other scenarios where sparse signal identification is utilized. The specific contributions of this thesis are as follows:

Chapter 2 and 3: Novel adaptive tap partial update adaptive algorithms for network echo cancellation are proposed in both the time domain and the frequency domain. As the echo path channel is typically long and sparse, it is unnecessary and inefficient to update all of the taps. Although partial update algorithms can be used to solve this problem, it is difficult to predetermine a fixed number of partial-update taps without a priori knowledge of the channel. Controlled by a simple sparseness measurement of the filter coefficients, the new algorithms adapt not only the filter coefficients but also the number of taps to be updated, iteration by iteration. Simulation results show that, compared with fully updated algorithms, the proposed partial update approach achieves both faster initial convergence and lower computational complexity while achieving almost the same steady-state error.

Chapter 4: For echo cancellation, the proportionate Affine Projection Algorithm (APA) has been proposed to take advantage of the sparseness of the echo path. In this chapter, we apply the µ-law proportionate technique to APA combined with a recently proposed structure having a memory of the proportionate step-size control matrices. Simulation results show that this new algorithm, termed μ-law Memorized Improved Proportionate APA (MMIPAPA), as well as its segmented implementation (SMIPAPA), yields a further performance improvement while maintaining a low computational complexity.

Chapter 5: In Acoustic Echo Cancellation (AEC), the echo paths are often long, sparse and rapidly changing. By maximizing the decrease in the mean square deviation of estimated coefficients between adjacent iterations, we obtain the optimum step size for each filter tap at each iteration. A time-averaging gradient estimate is used as the gain distribution vector to assign individual step sizes so that the filter tap update is in proportion to the magnitude of the gradient vector. Based on this new proportionate technique, the Gradient-Controlled Proportionate Affine Projection Algorithm (GC-

PAPA), with the Gradient-Controlled Proportionate Normalized Least Mean Square (GC-PNLMS) algorithm as a special case when the projection order is one, are proposed in this chapter. We also propose the Gradient-Controlled Proportionate Multi-Delay Filter (GC-PMDF) in the frequency domain. Simulation results demonstrate significant improvements in the convergence rate of the proposed algorithms compared with traditional proportionate algorithms.

Chapter 6: For sparse filter design in compressive sensing, the zero-attracting LMS (ZA-LMS) incorporates the l_1 norm penalty function into the quadratic LMS cost function to promote the sparseness during the adaptation process. The reweighted ZA-LMS (RZA-LMS) is developed using reweighted zero attractors with better performance. In this chapter, we propose two new sparse LMS algorithms with segment zero attractors, referred to as Segment RZA-LMS and Discrete Segment RZA-LMS. The Segment RZA-LMS outperforms RZA-LMS by using a piece-wise approximation of the reciprocal term in the iterative algorithm of RZA-LMS. The Discrete Segment RZA-LMS is further developed to achieve faster convergence speed and better steady state error performance than Segment RZA-LMS.

Chapter 7: Baseband digital pre-distortion (DPD) has been proposed as an efficient and low-cost solution which can reduce the nonlinearity introduced by radio-frequency power amplifiers operating with wideband signals. Although recent improvements to the technique have achieved good static performance, the DPD performance degrades for power amplifiers exhibiting dynamic nonlinearities within a short time. In this chapter, we propose the power-indexed look-up table (LUT) design for DPD, which provides pre-distorter coefficients adaptively and in real-time by storing, switching, and interpolating multiple DPD solutions in hardware. As demonstrated in experimental results obtained using an actual 2-GHz power amplifier, the proposed design achieves very stable and robust linearization performance for power amplifiers having dynamic nonlinearities.

Chapter 2

Time Domain Adaptive Algorithms for Echo Cancellation

2.1 Introduction

Network echo cancellation (NEC) has been an active research area for several decades. As shown in Fig. 2.1, the echo path impulse response is caused by the impedance mismatch between four-wire and two-wire circuits connected via the network hybrid [4-5]. In NEC, the echo path impulse response is typically of duration 64 to 128 ms. For a sampling frequency of 8 kHz, the total number of taps will be at least 512-1024 as shown in Fig. 1.1. Various adaptive filtering algorithms such as Least Mean Square (LMS) and Normalized LMS (NLMS) have been employed to cancel the echo resulting from the echo path or to identify the impulse response of the echo path [1-3].

The disadvantages of using traditional adaptive algorithms for network echo cancellation are high power consumption and slow convergence speed. It has been observed that the impulse response of a typical echo path is usually long and sparse. The active region, which is defined as the region where the impulse response

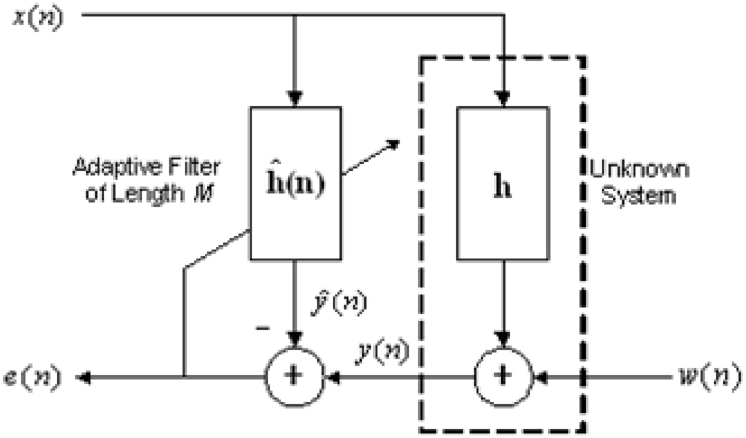


Fig. 2.1: System structure of NEC.

coefficients have large magnitude, is relatively small. By taking advantage of this property, a partial update algorithm adapts only a small subset of the filter taps in each iteration. When applied to the network echo canceller system, this can dramatically reduce the power consumption while retaining nearly the same performance as traditional adaptive algorithms.

One of the most popular partial update adaptive schemes is MMax [20-22], in which only a fixed number M' of taps are adapted each iteration. The filter taps selected for update are based on two possible criteria. The first one updates the filter coefficients corresponding to the M' largest output values of the adaptive filter (MMax1), while the second updates the filter coefficients corresponding to the M' largest input values to the adaptive filter (MMax2). Simulation results show that MMax1 has a fast initial convergence speed, but the steady-state convergence error with this criterion is much larger than for NLMS. MMax2 has small convergence error (almost the same as fully updated NLMS). However, the initial convergence speed is slow. Combination of the two criteria has also been proposed in [22], but the way in which the fixed value of M' is chosen still remains an issue. The method for determining M' plays a key role in the success of a partial update adaptive filter, since lower power consumption requires M' to be reduced, while maintaining fast convergence and good error performance require a large value of M'. However, without a priori knowledge of the network echo channel, it is difficult to determine the optimal value of M' in the previously proposed algorithms.

Although we can reduce power consumption by using a partial update algorithm, it is also important to improve the convergence speed. Several approaches have been proposed to address this problem, including the Proportionate NLMS (PNLMS) [14] and improved PNLMS [15], and modified versions such as improved IPNLMS (IIPNLMS) and sparseness controlled IPNLMS (SC-IPNLMS) [30-33]. All of these algorithms have better convergence speed than NLMS. However, they are all fully updated adaptive algorithms which have large computational complexity and power consumption.

In this chapter, a new adaptive filter design is presented in [34], which can adaptively change the number of partial-update taps and achieve the same or better convergence speed than the aforementioned adaptive algorithms, with a lower computational complexity as possible. Based on the characteristics of the channel, it is natural to assign small M' to sparse channels and large M' to non-sparse channels. Using this idea, the proposed algorithm, named as Sparseness-Controlled Adaptive Tap IPNLMS-MMax (SC-AT IPNLMS-MMax), adaptively changes M' during the iterations according to the real-time sparseness measure of the channel. Simulation results show that it can achieve the same or even better convergence performance as fully updated adaptive algorithms while only updating a small number of taps, which reduces both computational complexity and power consumption.

The remainder of this chapter is organized as follows. Section 2.2 briefly reviews the PNLMS and IPNLMS algorithms, and the MMax partial update algorithm. In Section 2.3, we introduce a novel scheme to adaptively change the number of partial-update taps during the iterations and our proposed SC-AT IPNLMS-MMax algorithm is described. Simulation results are presented in Section 2.4 to compare the performance of SC-AT IPNLMS-MMax with previous fixed-tap partial update algorithms and also several fully updated adaptive algorithms. Section 2.5 gives our conclusions.

2.2 Background of Proportionate Algorithms & Partial Update Adaptive Filter

2.2.1 PNLMS & IPNLMS

For the network echo canceller, a traditional adaptive algorithm such as NLMS performs relatively poorly because of the long and sparse echo path impulse response. To improve the convergence speed, proportionate adaptive algorithms, such as PNLMS and several improved versions of PNLMS, have gained increased attention recently. PNLMS introduces a step-size control matrix so that each filter coefficient is updated using an independent step-size. To review the PNLMS algorithm, we define the actual system’s impulse response, estimated impulse response and the input vector of the system in Fig. 2.1 as follows:

$$\mathbf{h} = \left[h_0,\ h_1,....h_{L-1}\right]^T,$$

$$\hat{\mathbf{h}}(n) = \left[\hat{h}_0(n),\ \hat{h}_1(n),....\hat{h}_{L-1}(n)\right]^T,$$

$$\mathbf{x}(n) = \left[x(n),\ x(n-1),....x(n-L+1)\right]^T,$$

where L is the length of the adaptive filter. The desired response and output are given by:

$$d(n) = \mathbf{h}^T\mathbf{x}(n) + w(n), \tag{2.1}$$

where $w(n)$ is additive Gaussian white noise with zero mean and variance σ_w^2.

The update scheme for the proportionate adaptive algorithm is generalized as follows:

$$e(n) = d(n) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n), \tag{2.2}$$

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\mathbf{x}^{\mathbf{T}}(n)\mathbf{Q}(n)\mathbf{x}(n)+\delta}, \tag{2.3}$$

$$\mathbf{Q}(n) = diag\left\{q_0(n), q_1(n),...., q_{L-1}(n)\right\}, \tag{2.4}$$

where δ is the regularization parameter and for PNLMS $\delta = \frac{\delta_{NLMS}}{L}$ to ensure the same steady state normalized misalignment as NLMS. For the $\mathbf{Q}(n)$, each $q_l(n)$ is defined as:

$$q_l(n) = \kappa_l(n) \Big/ \sum_{i=0}^{L-1}\kappa_i(n), \tag{2.5}$$

$$\kappa_l(n) = \max\left\{\left|\hat{h}_l(n)\right|,\ \rho\max\left\{\gamma, \left|\hat{h}_0(n)\right|,...., \left|\hat{h}_{L-1}(n)\right|\right\}\right\},\ l = 0, 1, ..., L-1, \tag{2.6}$$

where γ is a small value (typically 0.01) used to prevent $\hat{h}_l(n)$ from stalling during the initialization stage, and ρ (typically $1/L$) is also designed to prevent $\hat{h}_l(n)$ from stalling when the its magnitude is small.

PNLMS introduces the $\mathbf{Q}(n)$ parameter into the NLMS algorithm, which operates as a step-size control matrix. However, when it is applied to a non-sparse impulse response, performance degrades dramatically and is worse than NLMS. This situation is improved in the IPNLMS algorithm which can be regarded as a combination of NLMS and PNLMS. By removing the inaccurate choice of $\kappa_l(n)$, $q_l(n)$ is re-defined in IPNLMS as:

$$q_l(n) = \frac{1-\alpha}{2L} + \frac{(1+\alpha)\left|\hat{h}_l(n)\right|}{2\left\|\hat{\mathbf{h}}(n)\right\| + \varepsilon}, \; l = 0, 1, ..., L-1, \tag{2.7}$$

where ε is a small value used to prevent dividing by zero, and δ in (2.3) is set to $\frac{1-\alpha}{2L}\delta_{NLMS}$. The adjustable parameter $\alpha \in [-1, 1]$ determines the behavior of IPNLMS, as it ranges between NLMS ($\alpha = -1$) and PNLMS ($\alpha = 1$). IPNLMS works well for both non-sparse and sparse impulse responses, and it gives faster convergence speed than PNLMS while still maintaining the same mean square error (MSE) performance. Results from [15] show that good choices of α are 0, -0.5 and -0.75 (We will use $\alpha = -0.75$ in our proposed algorithm in this chapter).

2.2.2 Partial Update Criteria

Partial update adaptive filters have been of great interest due to their ability to save power consumption. The two MMax partial update criteria are widely used because of their simplicity, comparable performance and convergence speed compared with fully updated adaptive algorithms, which can be described as follows:

1) MMax1-NLMS

$$\hat{h}_l(n+1)=\begin{cases}\hat{h}_l(n)+\dfrac{\mu e(n)x(n-l)}{\|\mathbf{x}(n)\|^2+\delta}, & \begin{pmatrix}\text{if } l \text{ corresponds to one of the } M\,' \\ \text{biggest } \left|x(n-l)\hat{h}_l(n)\right|, l\text{=0,....,}L\text{-1}\end{pmatrix} \\ \hat{h}_l(n), & (else)\end{cases}. \qquad (2.8)$$

2) *MMax2-NLMS*

$$\hat{h}_l(n+1)=\begin{cases}\hat{h}_l(n)+\dfrac{\mu e(n)x(n-l)}{\|\mathbf{x}(n)\|^2+\delta}, & \begin{pmatrix}\text{if } l \text{ corresponds to one of the } M\,' \\ \text{biggest } \left|x(n-l)\right|, l\text{=0,....,}L\text{-1}\end{pmatrix} \\ \hat{h}_l(n), & (else)\end{cases}. \qquad (2.9)$$

The MMax1 criterion chooses to update the filter coefficients corresponding to the $M\,'$ largest outputs using the NLMS algorithm, which takes into account the amplitude of the coefficients [22]. Simulation results have shown that it can achieve a fast initial convergence speed. However, after the coefficients are close to their optimal values, this strategy produces a large steady-state mean square error. On the other hand, MMax2 updates the filter coefficients corresponding to the $M\,'$ largest input values using the NLMS algorithm [22]. Theoretical and simulation results show that, with appropriate choice of $M\,'$, this algorithm can achieve very good performance close to the fully updated NLMS algorithm, but its initial convergence speed is slower than MMax1. In addition, how to choose an appropriate $M\,'$ still remains an open question for almost all partial update algorithms. This motivates us to design a technique for an adaptive tap partial update adaptive filter that is controlled by the sparseness of the estimated filter coefficients.

It should also be mentioned that although the MMax1 and MMax2 criteria have been used in conjunction with the NLMS algorithm, they can also be applied to other adaptive algorithms such as PNLMS and IPNLMS.

2.3 Proposed Sparseness-Controlled Adaptive Tap Partial Update Algorithms

2.3.1 Definition of Sparseness

A sparseness measure of a channel vector $\mathbf{h}(n)$ has been defined as [35]:

$$\xi(n) = \frac{L}{L-\sqrt{L}}\left[1-\frac{\|\mathbf{h}(n)\|_1}{\sqrt{L}\,\|\mathbf{h}(n)\|_2}\right], \tag{2.10}$$

where L is the length of the vector $\mathbf{h}(n)$. Notice that ξ has the following properties:

- For a delta channel or delayed delta channel, $\xi = 1$, regardless of the length of the channel.
- For a constant (or moving average) channel, $\xi = 0$.
- $0 \le \xi \le 1$ for any channel vector $\mathbf{h}$.

We calculate the sparseness measure $\hat{\xi}(n)$ of the estimated filter coefficients $\hat{\mathbf{h}}(n)$ during each iteration. As the estimated filter coefficients $\hat{\mathbf{h}}(n)$ gradually converge to their optimal values, the sparseness measurement $\hat{\xi}(n)$ will also converge to its optimal value, which is the sparseness of the echo path impulse response. Indeed, $\hat{\xi}(n)$ is expected to converge even faster than $\hat{\mathbf{h}}(n)$ does, as it is less sensitive to the fluctuation of $\hat{\mathbf{h}}(n)$ around its optimal value due to the stochastic gradient characteristic. Here, we propose a new sparseness measure:

Table 2.1: Comparisons between the two sparseness measures

Sparseness Measures	$\hat{\xi}(n)$	$s\hat{p}(n)$
Range	$0 \le \hat{\xi}(n) \le 1$	$0 < s\hat{p}(n) < 1$
Delta Channel	$\hat{\xi}(n) = 1$ (the most sparse)	$s\hat{p}(n) \approx \left(1 - \frac{1}{\sqrt{L}}\right)^P \to 1$ (depending on *L*, the most sparse if $L \to \infty$)
Constant Channel (Moving Average)	$\hat{\xi}(n) = 0$ (the least sparse)	$s\hat{p}(n) \to 0$ (the least sparse)
Adjustable	No	Yes, the degree of sparseness can be adjusted by power *P*.

$$s\hat{p}(n) = \left[1 - \frac{\left\|\hat{\mathbf{h}}(n)\right\|_1}{\sqrt{L}\left\|\hat{\mathbf{h}}(n)\right\|_2 + \varsigma}\right]^P, \tag{2.11}$$

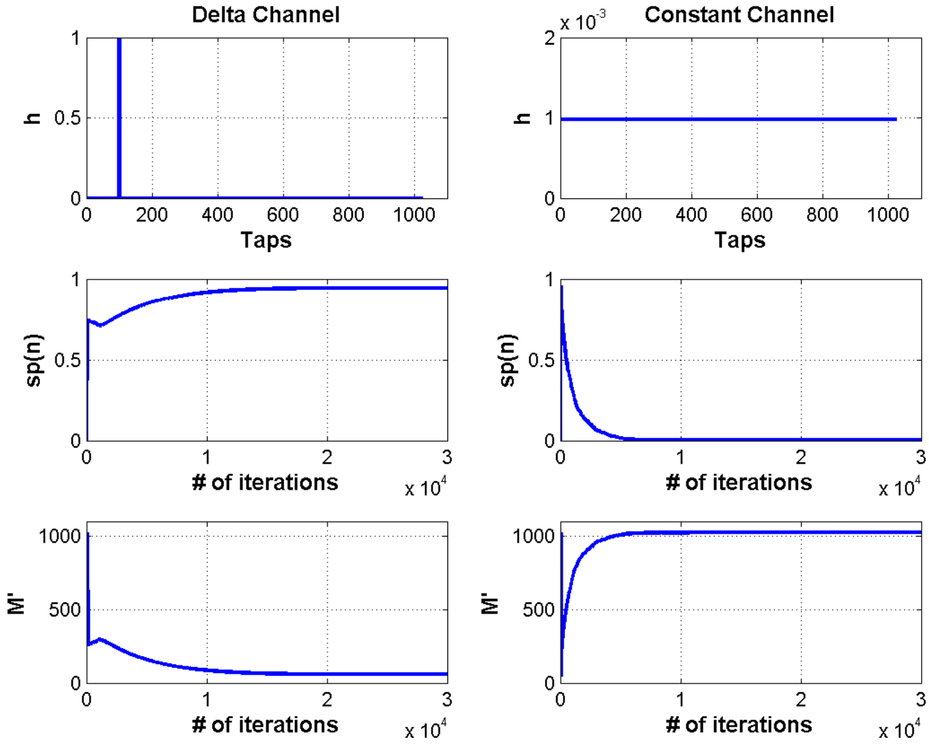
which differentiates delta channels having different numbers of taps, *e.g.* the sparseness of a 2-tap delta channel should be smaller than that of a 1024-tap delta channel. In addition, the degree of sparseness measurement can be adjusted by the power *P* which gives more freedom in trading off between performance and power consumption. ς is a small value used to prevent from dividing by zero in the initial iteration. The comparison between $\hat{\xi}(n)$ and $s\hat{p}(n)$ is shown in Table 2.1.

2.3.2 Sparseness Controlled Adaptive Tap Strategy

We propose a new adaptive tap strategy, which makes use of the sparseness measure $s\hat{p}(n)$ to adaptively control the number of taps that need to be updated in each iteration:

$$M'(n) = \left(1 - s\hat{p}(n)\right)L, \tag{2.12}$$

where L is the total number of filter taps and M' is the number of taps that need to be



(a) (b)

Fig. 2.2: Adaptive tap partial update technique for (a) a delta channel, and (b) a constant channel.

updated. The larger the sparseness measure value, the smaller the number of partial-update taps.

Fig. 2.2 shows the adaptation of M' as the sparseness measure $s\hat{p}(n)$ converges for both the delta channel (the sparsest channel) and the constant channel (the least sparse channel). Here, L is 1024 and P is set to 1 in Equation (2.11). For the sparse channel, the number of updated taps converges quickly to a small number (58 out of 1024), while for the non-sparse channel, our adaptive tap strategy is actually equivalent to a full tap algorithm (1024 out of 1024). These results demonstrate that our sparseness-controlled adaptive tap partial update strategy can work well over a wide range of impulse responses.

We also propose to divide the convergence process of the estimated coefficients into two stages, *i.e.* before and after the convergence of the sparseness measure, which is determined by the value of $|s\hat{p}(n)-s\hat{p}(n-1)|$. If $|s\hat{p}(n)-s\hat{p}(n-1)| \geq \varepsilon$, the sparseness measure has not yet converged, while if $|s\hat{p}(n)-s\hat{p}(n-1)| < \varepsilon$, the sparseness measure can be viewed as having converged. In the first stage, extra attention is needed for the coefficients having large magnitude to ensure a fast initial convergence speed, where the MMax1 criterion is used. In the second stage, it is no longer necessary to focus on the large coefficients as all of the estimated coefficients are already fluctuating around their optimal values. Therefore, the MMax2 criterion is adopted to yield a small convergence error. In simulations, we have found that a value of $\varepsilon = 0.0001$ gives good results.

In the next section, the sparseness-controlled adaptive tap partial update algorithm is presented and the simulation results show very good performance, even when compared with the fully updated IPNLMS algorithm.

2.3.3 Proposed Sparseness Controlled Adaptive Tap IPNLMS-MMax Partial Update Algorithm

Based on the ideas that have been described in Section 2.3.2, we propose a new sparseness-controlled adaptive tap partial update algorithm, called SC-AT IPNLMS-MMax, which is derived using the MMax partial update criteria, along with IPNLMS and NLMS algorithms. In each iteration, only M' out of L taps are adaptively updated, where the value of M' is also adaptively changing according to equation (2.12). The convergence process is regarded as a combination of two stages, as presented in Section 2.3.2, and different partial update criteria are used as follows:

1) *1st stage:* $|\hat{sp}(n) - \hat{sp}(n-1)| \geq \varepsilon$

$$q_l(n) = \begin{cases} \dfrac{1-\alpha}{2L} + \dfrac{(1+\alpha)\left|\hat{h}_l(n)\right|}{2\left\|\hat{\mathbf{h}}(n)\right\| + \varepsilon}, & \begin{pmatrix} \text{if } l \text{ corresponds to one of the } M' \text{biggest} \\ \left|x(n-l)\hat{h}_l(n)\right|,\ l = 0,1,....,L-1 \end{pmatrix}, \\ 0, & (\text{else}) \end{cases} \quad (2.13)$$

$$\hat{h}_l(n+1) = \begin{cases} \hat{h}_l(n) + \dfrac{\mu\, q_l(n) e(n) x(n-l)}{\mathbf{x}^{\mathrm{T}}(n)\mathbf{Q}(n)\mathbf{x}(n) + \delta}, & (\text{if } l \text{ corresponds to one of the } M' \\ \mathbf{Q}(n) = diag\left\{q_0(n),\ ...,q_{L-1}(n)\right\}, & \text{biggest} \left|x(n-l)\hat{h}_l(n)\right|,\ l = 0,...,L-1) \, . \\ \hat{h}_l(n), & (\text{else}) \end{cases}$$

(2.14)

2) *2nd stage:* $|\hat{sp}(n) - \hat{sp}(n-1)| < \varepsilon$

$$\hat{h}_l(n+1) = \begin{cases} \hat{h}_l(n) + \dfrac{\mu\, e(n) x(n-l)}{\left\|\mathbf{x}(n)\right\|^2 + \delta}, & \begin{pmatrix} \text{if } l \text{ corresponds to one of the } M' \\ \text{biggest } \left|x(n-l)\right|, l\text{=0,....,}L\text{-1} \end{pmatrix} \\ \hat{h}_l(n), & (else) \end{cases} . \quad (2.15)$$

In the first stage, IPNLMS is used rather than NLMS. IPNLMS can achieve faster convergence speed than NLMS because of its use of the step-size control matrix $\mathbf{Q}(n)$ so that each filter coefficient is updated proportionally to its magnitude. Since we are also using MMax1, where the magnitude of the filter coefficients are considered in the sorting process, coefficients of large magnitude have more chances to be updated compared with those having an optimal value of zero. Now, the coefficients with large magnitude are not only updated with bigger step size, but also updated more frequently since we are selecting M' taps based on the coefficient magnitudes. This can significantly speed up the initial convergence rate as the estimated coefficients approach their optimal values. Then, in the second stage when the sparseness $s\hat{p}(n)$ reaches a steady value, all filter coefficients fluctuate around their optimal values. Thus, a simple MMax2 adaptive tap partial update criterion with the NLMS adaptive algorithm is used, which is sufficient to achieve good performance. Note that because we use an adaptive tap partial update strategy, this leads to significantly reduced computational complexity compared with the fully updated IPNLMS.

2.4 Simulations & Results

The performance of our adaptive tap partial update algorithm is presented in this section using the normalized misalignment (in dB) as the objective function:

$$\eta(n) = 20\log_{10}\left(\frac{\left\|\mathbf{h} - \hat{\mathbf{h}}(n)\right\|_2}{\left\|\mathbf{h}\right\|_2}\right). \tag{2.16}$$

The setup for the simulations is based on Fig. 1.1, where the length of the adaptive filter is chosen to be 1024, the input signal is generated according to a Gaussian distribution with zero mean and variance 1, and $w(n)$ is additive Gaussian white noise with zero mean and variance 0.001 such that the SNR is 30 dB. The same step size $\mu = 0.3$ is used.

Fig. 2.3 shows the misalignment comparison among 1024-tap NLMS, 1024-tap IPNLMS, 1024-tap SC-IPNLMS [33], and our proposed partial update SC-AT IPNLMS-MMax algorithm with the parameter *P*=2. It can be seen that SC-AT IPNLMS-MMax has the fastest convergence speed among all the algorithms with just a

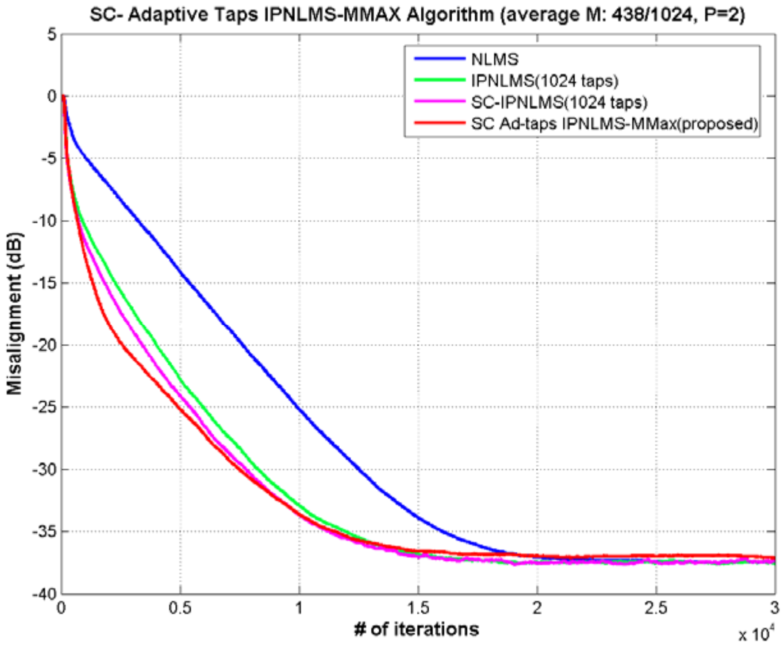


Fig. 2.3: Misalignment of 1024-tap NLMS, 1024-tap IPNLMS, 1024-tap SC-IPNLMS and partial update SC-AT IPNLMS-MMax with *P*=2.

small number of taps updated in each iteration. The average value of M' for *P*=2 is only 438. Fig. 2.4 shows the further improvement in steady-state error achieved by using SC-AT IPNLMS-MMax with *P*=3. The average value of M' increases to 580 for *P*=3, and the same steady-state misalignment as in the fully adaptive algorithms is achieved by using SC-AT IPNLMS-MMax algorithm.

The misalignment of the SC-AT IPNLMS-MMax algorithm in Fig. 2.4 is slightly smaller than that in Fig. 2.3 because as *P* increases, the sparseness measure decreases, resulting in a larger value of M' each iteration. The larger the number of partially updated taps, the smaller the misalignment will be. This demonstrates the power of adjusting the degree of sparseness, which gives us more freedom to control the power consumption (or, equivalently, the number of multiplications) during iterations

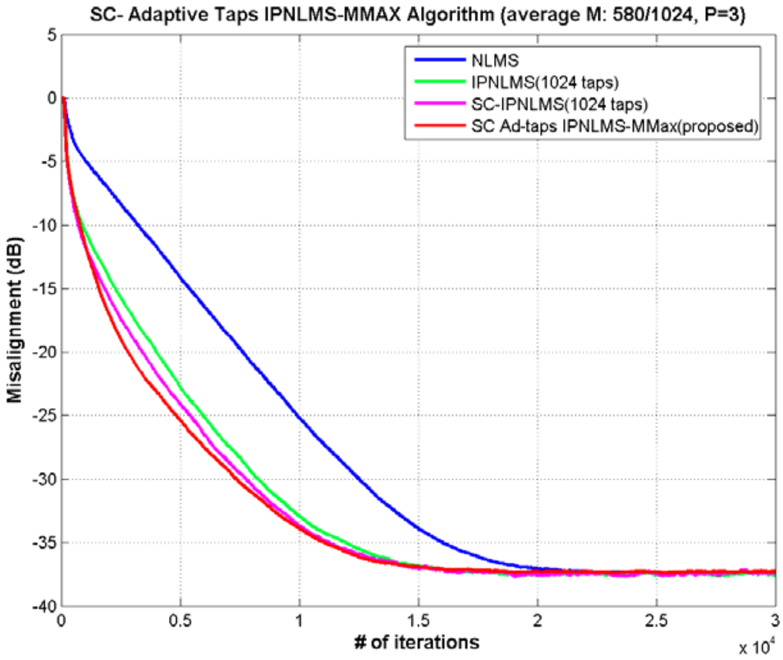


Fig. 2.4: Misalignment of 1024-tap NLMS, 1024-tap IPNLMS, 1024-tap SC-IPNLMS and partial update SC-AT IPNLMS-MMax with *P*=3.

according to our different misalignment objectives. These results indicate that we can choose a small value of *P* to reduce the number of multiplications without significantly impacting the convergence error.

2.5 Conclusions

In this chapter, we have proposed a novel strategy for adaptively changing the number of partially updated taps using a new sparseness measure of an adaptive filter. The convergence process has been analyzed and the method of having two convergence stages with different partial update criteria have been used to construct a novel adaptive tap partial update algorithm, SC-AT IPNLMS-MMax. The simulation results show that for SC-AT IPNLMS-MMax, faster convergence speed is obtained with less computation than with the fully updated IPNLMS and SC-IPNLMS. Our future research in this area will focus on computationally efficient sparseness representations as well as on the development of low power architectures for the proposed algorithm.

Chapter 3

Frequency Domain Adaptive Algorithms for Echo Cancellation

3.1 Introduction

As an attractive alternative to time domain adaptive algorithms, frequency domain adaptive algorithms have gained attention recently because of their efficient implementations and delay performance improvements. For example, the fast-LMS (FLMS) [36] method implements an efficient block-LMS updating strategy through the use of the FFT. However, FLMS has the drawback of introducing a long delay between the input and the output. To address this delay problem, a flexible MDF structure was proposed [37], where the L-tap adaptive filter is partitioned into an arbitrary number K of sub-filters having a smaller block size N. To further improve the convergence speed, [38] incorporates the proportionality control of the IPNLMS algorithm together with the frequency domain MDF structure, and proposes the improved proportionate MDF (IPMDF). It can be observed from simulation results shown in [38] that IPMDF achieves both an improvement in the convergence performance as well as a better time-varying tracking ability compared with MDF and IPNLMS. The primary limitation of IPMDF is its computational complexity.

Also, a partial update MDF structure, SPMMax-MDF was designed in [39] by applying the SPMMax partial update criterion to MDF. However, without a priori knowledge of the network echo path, it is difficult to determine the optimal subset of taps to update in order to guarantee convergence. In Chapter 2, we have described our proposed time-domain adaptive tap partial update filter for IPNLMS inspired by the convergence behavior of a sparseness quantity, which can adaptively change the number of filter taps being updated according to a real-time sparseness value. In this chapter, we present a low complexity, fast converging and efficient frequency-domain partial update algorithm using the adaptive tap strategy as we proposed in [40]. We also divide the convergence process of the estimated filter coefficients into two stages, before and after the convergence of the sparseness measure, using the MMax-IPMDF and SPMMax-MDF structures, respectively. Simulation results show that our proposed frequency domain adaptive algorithm achieves the same convergence performance as fully updated IPMDF while only updating a small number of taps, which reduces both computational complexity and power consumption.

3.2 Background of Frequency Domain Adaptive Filters

3.2.1 MDF & IPMDF

MDF was proposed in order to address the inherent delay problem of previous frequency domain adaptive algorithms [37]. This partitions the adaptive filter of length

L into K smaller delay sub-filters having block size N so that $L = KN$. It reduces the delay of FLMS by a factor of K. We first define m as the frame index, k as the sub-filter index, and the following time domain quantities

$$\mathbf{x}(mN) = [x(mN),...,x(mN - L + 1)]^T,$$

$$\mathbf{X}(m) = [\mathbf{x}(mN),...,\mathbf{x}(mN + N - 1)]^T,$$

$$\mathbf{y}(m) = [y(mN),...,y(mN + N - 1)]^T,$$

$$\hat{\mathbf{h}}(m) = [\hat{\mathbf{h}}_{\mathbf{0}}{}^T(m),...,\hat{\mathbf{h}}_{\mathbf{K-1}}{}^T(m)]^T,$$

$$\hat{\mathbf{h}}_k(m) = [\hat{h}_{kN}(m),...,\hat{h}_{kN+N-1}(m)]^T,\ k = 0,...,K-1,$$

$$\mathbf{e}(m) = \mathbf{d}(m) - \mathbf{X}^T(m)\mathbf{h}(m-1) = [e(mN),...,e(mN + N - 1)]^T. \qquad (3.1)$$

To describe the MDF algorithm, let $\mathbf{F}$ and $\mathbf{F}^{-1}$ be the $2N \times 2N$ Fourier Matrix and inverse Fourier Matrix, respectively. Due to the structure of the MDF, the m^{th} block input vector to the k^{th} sub-filter in the frequency domain is the same as the $(m+1)^{\text{th}}$ block input vector to the $(k+1)^{\text{th}}$ sub-filter. Therefore, only one $2N$-point FFT of the overlapped vector is needed to obtain the frequency domain input per block iteration.

Define $\underline{\mathbf{X}}(m-k) =$ diag$\left\{\mathbf{F}\left[x\left((m-k)N - N\right),...,x\left((m-k)N + N - 1\right)\right]^T\right\}$ to be the m^{th} block input to the k^{th} sub-filter, as well as the following frequency domain quantities:

$$\underline{\mathbf{d}}(m) = \mathbf{F}\begin{bmatrix}\mathbf{0}_{N\times1}\\ \mathbf{d}(m)\end{bmatrix},\ \underline{\hat{\mathbf{h}}}_{\mathbf{k}}(m) = \mathbf{F}\begin{bmatrix}\hat{\mathbf{h}}_{\mathbf{k}}(m)\\ \mathbf{0}_{N\times1}\end{bmatrix},\ \underline{\mathbf{e}}(m) = \mathbf{F}\begin{bmatrix}\mathbf{0}_{N\times1}\\ \mathbf{e}(m)\end{bmatrix},$$

$$\mathbf{G}^{01} = \mathbf{F}\,\mathbf{W}^{01}\,\mathbf{F}^{-1},\ \mathbf{G}^{10} = \mathbf{F}\,\mathbf{W}^{10}\,\mathbf{F}^{-1}, \qquad (3.2)$$

where $\mathbf{W}^{01} = \begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{I}_{N\times N}\end{bmatrix}$, $\mathbf{W}^{10} = \begin{bmatrix}\mathbf{I}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\end{bmatrix}$.

Then the MFD frequency domain adaptive algorithm is as in [38]:

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{d}}(m) - \mathbf{G}^{01} \sum_{k=0}^{K-1} \underline{\mathbf{X}}(m-k)\,\hat{\underline{\mathbf{h}}}_{\mathbf{k}}(m-1), \tag{3.3}$$

$$\mathbf{S}(m) = \lambda\,\mathbf{S}(m-1) + (1-\lambda)\,\underline{\mathbf{X}}^{*}(m)\underline{\mathbf{X}}(m), \tag{3.4}$$

$$\hat{\underline{\mathbf{h}}}_{k}(m) = \hat{\underline{\mathbf{h}}}_{k}(m-1) + \mu\,\mathbf{G}^{10}\underline{\mathbf{X}}^{*}(m-k)\left[\mathbf{S}(m)+\delta\right]^{-1}\underline{\mathbf{e}}(m). \tag{3.5}$$

$0 << \lambda \le 1$ is the forgetting factor and $\mu = \beta(1-\lambda)$ is the step size with $0 < \beta \le 1$ [38]. $\mathbf{S}(m)$ and δ are regularization parameters.

As an enhancement of MDF, IPMDF [38] incorporates the idea of proportionate step-size control of IPNLMS [15]. It introduces a time-domain step-size control matrix $\mathbf{Q}_{\mathbf{k}}(m)$ to each sub-filter:

$$\mathbf{Q}_{k}(m) = diag\left\{\mathbf{q}_{k}(m)\right\},\ \mathbf{q}_{k}(m) = [q_{kN}(m),\ldots,q_{kN+N-1}(m)], \tag{3.6}$$

$$q_{kN+j}(m) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{h}_{kN+j}(m)\right|}{2\left\|\hat{\mathbf{h}}\right\|_{1} + \varepsilon}, \tag{3.7}$$

with a small number ε used to prevent dividing by zero, and $\alpha \in [-1, 1]$ is an adjustable parameter similar to that used in IPNLMS. The update scheme for IPMDF is now generalized as:

$$\hat{\underline{\mathbf{h}}}_{k}(m) = \hat{\underline{\mathbf{h}}}_{k}(m-1) + L\mu\,\mathbf{G}_{1}\,\mathbf{Q}_{k}(m)\,\mathbf{G}_{2}\,\underline{\mathbf{X}}^{*}(m-k)\left[\mathbf{S}(m)+\delta\right]^{-1}\underline{\mathbf{e}}(m), \tag{3.8}$$

with $\mathbf{G}^{10}$ in (3.2) replaced by the product of $\mathbf{G}_{1}\mathbf{G}_{2}$ where $\mathbf{G}_{1} = \mathbf{F}\begin{bmatrix}\mathbf{I}_{N\times N}\\ \mathbf{0}_{N\times N}\end{bmatrix}$ and $\mathbf{G}_{2} = \begin{bmatrix}\mathbf{I}_{N\times N} & \mathbf{0}_{N\times N}\end{bmatrix}\mathbf{F}^{-1}$. To achieve the same steady-state error, the regularization parameters $\mathbf{S}(0)$ in (3.4) and δ in (3.5) need to be scaled by $\frac{1-\alpha}{2}$ for IPMDF.

Simulations [38] show significant convergence improvement of IPMDF over MDF.

3.2.2 MMax and SPMMax Partial Update Criteria

As introduced in Chapter 2, MMax partial update scheme can achieve significant computational cost reduction compared with fully updated adaptive algorithms. Recently SPMMax partial update criterion combining MMax1 and MMax2 is proposed in [22] to improve the convergence performance while inheriting the computational simplicity. For notation convenience, let us rewrite the partial update NLMS as follows:

$$\hat{\mathbf{h}}(m) = \hat{\mathbf{h}}(m-1) + \mu \frac{\mathbf{P}(m)\mathbf{x}(m)e(m)}{\left\|\mathbf{P}(m)\mathbf{x}(m)\right\|_2^2 + \delta}, \tag{3.9}$$

where $\mathbf{P}(m) = diag\left\{p_0(m),\dots,p_{L-1}(m)\right\}$ determines the portion of taps being updated. For the MMax1 scheme:

$$p_l(m) = \begin{cases} 1, & \begin{pmatrix} if\ l\ corresponds\ to\ M_1 \text{ maxima } of \\ \left|x(m-l+1)h_l(m)\right|,\ \ l = 0,\dots,L-1 \end{pmatrix} \\ 0, & (else) \end{cases}. \tag{3.10}$$

For the MMax2 scheme:

$$p_l(m) = \begin{cases} 1, & \begin{pmatrix} if\ l\ corresponds\ to\ M_2 \text{ maxima} \\ of\left|x(m-l+1)\right|,\ \ l = 0,\dots,L-1 \end{pmatrix} \\ 0, & (else) \end{cases}. \tag{3.11}$$

To achieve faster convergence, the SPMMax criterion incorporates the idea of updating filter coefficients corresponding to the largest M_1 products of the input and the filter coefficients with updating coefficients corresponding to the largest M_2 input values, as shown below:

if mod$(m,T) \neq 0$: $p_l(m)$ updates as (3.10).

if mod$(m,T) = 0$: $p_l(m)$ updates as (3.11).

The quantities M_1, M_2, T are arbitrary numbers; how to select their optimum values remains an open question for almost all partial update algorithms. This motivates us to design a technique for an adaptive tap partial update filter that is controlled by the sparseness of the estimated filter coefficients.

3.3 Proposed Frequency Domain Adaptive Tap Partial Update Algorithms

3.3.1 Two-Stage Adaptive Tap Strategy

We have discussed in Chapter 2 that, for sparse network echo cancellation, as the estimated filter coefficients $\hat{\mathbf{h}}(m)$ gradually converge to their optimal values, the sparseness measurement $\hat{\xi}(m)$ of the estimated filter coefficients will also converge to its optimal value, which is the sparseness of the echo path impulse response. Indeed, $\hat{\xi}(m)$ is expected to converge even faster than $\hat{\mathbf{h}}(m)$ does, as it is less sensitive to the fluctuation of $\hat{\mathbf{h}}(m)$ around its optimal value due to the stochastic gradient characteristic. Here, we use $s\hat{p}(m)$ as defined in (2.11) in last chapter as the sparseness measure of the time domain estimated filter coefficients to adaptively control the number of taps that need updating in the frequency domain in each iteration:

$$M'(m) = 2L \times (1 - s\hat{p}(m)) . \qquad (3.12)$$

Note that the total number of corresponding frequency domain taps is $2L$ for an L-tap time domain adaptive filter. It has been shown in Chapter 2 that, with P set to 1, $M'(m)$ converges quickly to a small number for the delta channel, while for the constant channel our adaptive strategy reduces to a full tap algorithm. The sparseness-controlled adaptive tap strategy works well over a wide range of impulse responses. The larger the sparseness measure value, the smaller the number of partial update taps.

The convergence process of the estimated coefficients can be divided into two stages, i.e. before and after the convergence of the sparseness measure. If $\left|s\hat{p}(m) - s\hat{p}(m-1)\right| \geq \gamma$, then $s\hat{p}(m)$ has not yet converged, and extra attention is needed for the coefficients having large magnitude to ensure a fast initial convergence speed. Otherwise, $s\hat{p}(m)$ can be viewed as having converged. Here γ is a small value used to differentiate the two stages and we have found that a value of 0.0001 gives good results.

3.3.2 Proposed Adaptive Tap Partial Update IPMDF

Based on the ideas that have been described in Sections 3.3.1, we propose a new frequency domain adaptive tap partial update algorithm. The convergence process is regarded as a combination of two stages, where different frequency domain adaptive algorithms and different partial update criteria are used. First let us define the $2N \times 1$ vectors $\hat{\underline{\mathbf{h}}}(m)$ and $\underline{\boldsymbol{\chi}}(m)$ in the frequency domain:

$$\hat{\underline{\mathbf{h}}}(m) = [\hat{\underline{\mathbf{h}}}_0{}^T(m), \ldots, \hat{\underline{\mathbf{h}}}_{K-1}{}^T(m)]^T = [\hat{\underline{h}}_0(m), \ldots, \hat{\underline{h}}_{2L-1}(m)]^T,$$

$$\underline{\boldsymbol{\chi}}(m) = [\underline{\boldsymbol{\chi}}_0{}^T(m), \ldots, \underline{\boldsymbol{\chi}}_{K-1}{}^T(m)]^T = [\underline{\chi}_0(m), \ldots, \underline{\chi}_{2L-1}(m)]^T.$$

Our adaptive tap partial update IPMDF algorithm is as described in Table 3.1. In the 1st stage, the MMax-IPMDF algorithm is used for faster initial convergence with filter coefficients considered in the sorting process ($M'(m)$ maxima of $\left|\underline{\chi}_l(m)\hat{\underline{h}}_l(m)\right|$

Table 3.1: Proposed frequency domain partial update adaptive algorithm

1) $1st\ Stage: |s\hat{p}(m) - s\hat{p}(m-1)| \geq \gamma$

$$M'(m) = 2L \times (1 - s\hat{p}(m))$$

$$p_l(m) = \begin{cases} 1, & \left(\begin{array}{c} if\ l\ corresponds\ to\ M'(m)\ \text{maxima} \\ of \left| \underline{\chi}_l(m)\underline{\hat{h}}_l(m) \right|, l = 0, \ldots, 2L-1 \end{array} \right) \\ 0, & (else) \end{cases}$$

$$\mathbf{P}_k(m) = diag\left\{ p_{kN}(m), \ldots, p_{kN+N-1}(m) \right\}$$

$$\underline{\tilde{\mathbf{X}}}(m-k) = \mathbf{P}_k(m)\underline{\mathbf{X}}(m-k)$$

$$\underline{\hat{\mathbf{h}}}_k(m) = \underline{\hat{\mathbf{h}}}_k(m-1) + L\mu\,\mathbf{G}_1\,\mathbf{Q}_k(m)\,\mathbf{G}_2\,\underline{\tilde{\mathbf{X}}}^*(m-k) \times \left[\mathbf{S}(m) + \delta \right]^{-1} \underline{\mathbf{e}}(m)$$

2) $2nd\ Stage: |s\hat{p}(m) - s\hat{p}(m-1)| < \gamma$

$$M'(m)\ remain\ the\ same\ as\ last\ M'(m)\ of\ the\ 1st\ stage$$

$$if\ \text{mod}(m,T) \neq 0: \quad p_l(m) = \begin{cases} 1, & \left(\begin{array}{c} if\ l\ corresponds\ to\ M'(m)\ \text{maxima} \\ of \left| \underline{\chi}_l(m)\underline{\hat{h}}_l(m) \right|, l = 0, \ldots, 2L-1 \end{array} \right) \\ 0, & (else) \end{cases}$$

$$if\ \text{mod}(m,T) = 0: \quad p_l(m) = \begin{cases} 1, & \left(\begin{array}{c} if\ l\ corresponds\ to\ M'(m)\ \text{maxima} \\ of \left| \underline{\chi}_l(m) \right|, l = 0, \ldots, 2L-1 \end{array} \right) \\ 0, & (else) \end{cases}$$

$$\mathbf{P}_k(m) = diag\left\{ p_{kN}(m), \ldots, p_{kN+N-1}(m) \right\}$$

$$\underline{\tilde{\mathbf{X}}}(m-k) = \mathbf{P}_k(m)\underline{\mathbf{X}}(m-k)$$

$$\underline{\hat{\mathbf{h}}}_k(m) = \underline{\hat{\mathbf{h}}}_k(m-1) + \mu\,\mathbf{G}^{10}\,\underline{\tilde{\mathbf{X}}}^*(m-k)\left[\mathbf{S}(m) + \delta \right]^{-1} \underline{\mathbf{e}}(m)$$

instead of $\left|\underline{\chi}_l(m)\right|$). In the 2nd stage, when the sparseness measure reaches a steady state value, all filter coefficients fluctuate around their optimal values. Then, the SPMMax-MDF algorithm [39] is sufficient to achieve good performance. Since $s\hat{p}(m)$ rarely changes in the 2nd stage, calculation of $s\hat{p}(m)$ is unnecessary and the number of taps updating in the 2nd stage remains the same as the last $M'(m)$ of the 1st stage. Note that the diagonal matrix $\underline{\tilde{\mathbf{X}}}(m-k)$ in Table 3.1 contain $2L-M'$ null elements across all k, which significantly reduces the computational complexity compared with full-tap IPMDF.

3.4 Simulations & Results

The performance is evaluated using the normalized misalignment:

$$\eta(m) = 20\log_{10}\left(\frac{\left\|\mathbf{h}-\hat{\mathbf{h}}(m)\right\|_2}{\left\|\mathbf{h}\right\|_2}\right) \text{ (dB).} \tag{3.13}$$

The echo to identify is a 512-tap truncated sparse echo path impulse response as shown in Fig. 3.1. A white Gaussian signal with zero mean and variance 1 is used as the

system input $x(n)$, and $w(n)$ is a white Gaussian noise signal with zero mean and variance 0.001 such that the SNR is 30 dB. As in [38], simulation is setup with $L=512$, $N=64$, and the following parameters are used : $\alpha=-0.75$, $T=8$,

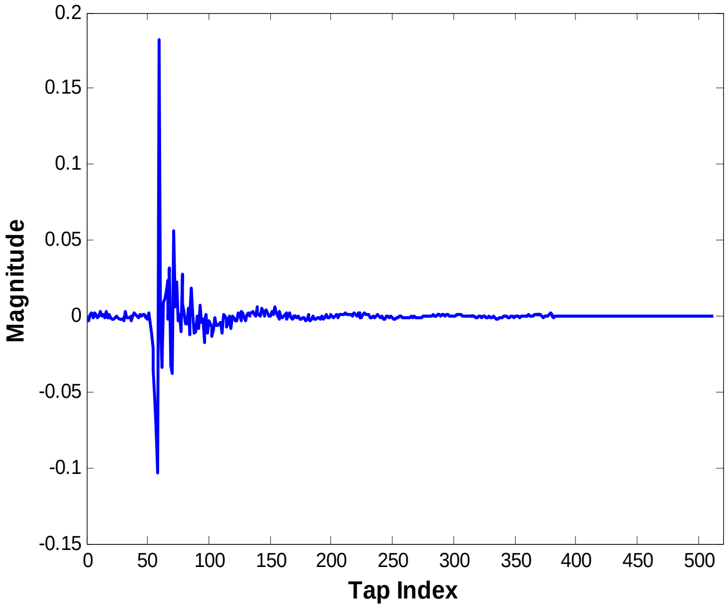


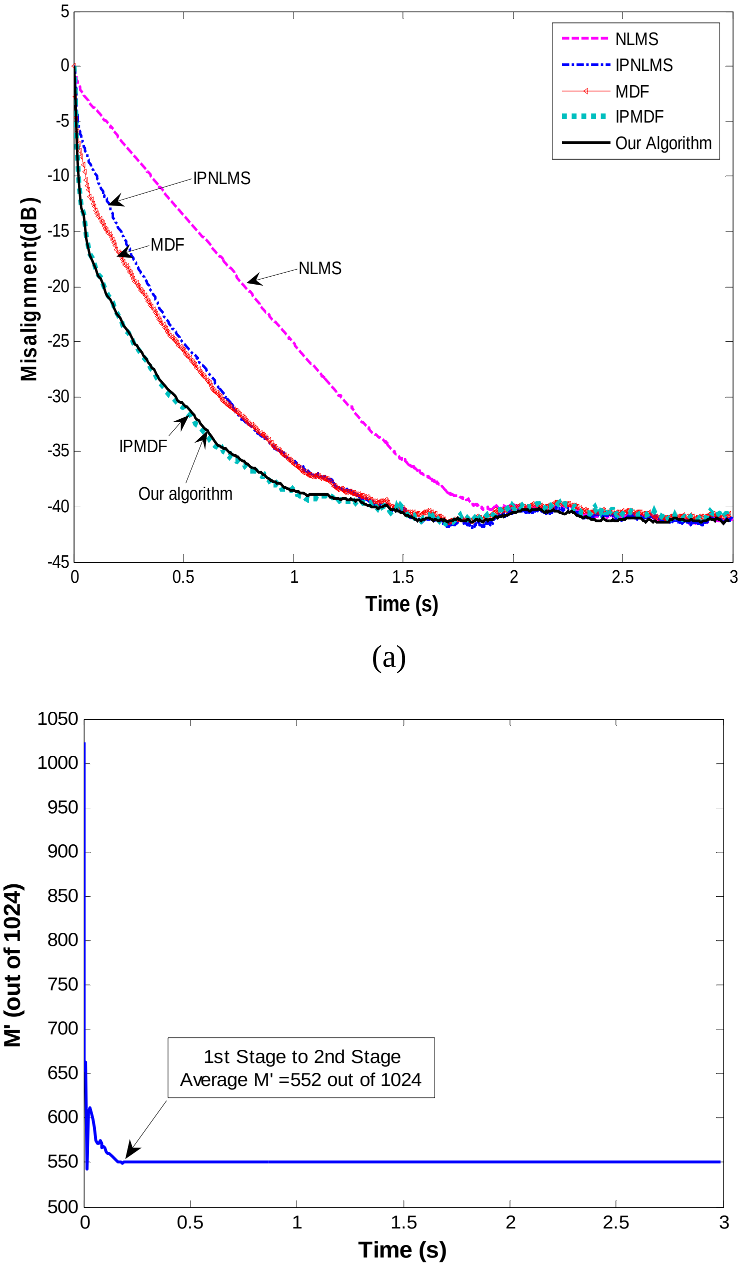
Fig. 3.1: 512-tap truncated sparse echo path impulse response.

$$\lambda = \left[1 - \frac{1}{3L}\right]^N, \ \mu = 1 - \lambda, \ \delta_{MDF} = \frac{\sigma_x{}^2 20N}{L},$$

$$\delta_{IPMDF} = \frac{1-\alpha}{2}\delta_{MDF}, \ \mathbf{S}_{MDF}(0) = \frac{\sigma_x{}^2}{100},$$

$$\delta_{IPMDF} = \frac{1-\alpha}{2}\delta_{MDF}, \ \mathbf{S}_{IPMDF}(0) = \frac{1-\alpha}{2}\mathbf{S}_{MDF}(0).$$

Fig. 3.2 shows the misalignment comparisons among fully updated NLMS, IPNLMS, MDF, IPMDF and our proposed partial-IPMDF, where the step size of NLMS and IPNLMS is set to be 0.15 for the same steady-state performance. A plot of the number of taps which are being updating during each iteration is also given; this rapidly converges to a small number, at which point the algorithm enters into its 2nd stage. With $P = 3$ in (2.11), the average number of taps that are updating for our



(a)

(b)

Fig. 3.2: (a) Misalignment of NLMS, PNLMS, MDF, PMDF and our proposed algorithm. (b) The number of filter taps being updated of our proposed algorithm.

algorithm is 552 (about $0.5 \times 2L$) out of 1024 ($2L$). However, it still achieves the same convergence speed as fully updated IPMDF (and is thus faster than SPMMax-MDF).

Moreover, the trade-off between convergence performance and computational complexity may be adjusted by choosing different values for P .

3.5 Conclusions

In this chapter, we have proposed a novel frequency-domain partial update algorithm which adapts both the number of taps which are updated as well as the filter coefficients. The convergence process is implemented as a two-stage process using the frequency-domain MMax-IPMDF and SPMMax-MDF procedures, respectively, together with our adaptive tap strategy. The simulation results show that, compared with fully updated IPMDF, our algorithm achieves the same convergence speed with a significantly reduced computational complexity.

Chapter 4

The µ-Law Proportionate Adaptive Algorithms for Echo Cancellation

4.1 Introduction

As discussed in Chapter 2 and Chapter 3, Proportionate Normalized Least Mean Square (PNLMS) algorithm [14] has been proposed to speed up the convergence rate by exploiting the sparsity with a step size control matrix to update each filter coefficient in proportion to the magnitude of its estimate. To further improve the performance, µ-law PNLMS (MPNLMS) was proposed in [16] using a step size control matrix proportional to the logarithm of coefficient magnitudes rather than using coefficient magnitudes directly.

The proportionate Affine Projection Algorithm (APA) was developed for echo cancellation due to its fast convergence speed and tracking ability [41, 17]. Later, the Improved Proportionate APA (IPAPA) [18] provided more robustness in non-sparse channels. Recently, a new proportionate APA having a memory of step size control matrices was proposed in [42], named Memorized Improved Proportionate APA

(MIPAPA). In MIPAPA, the proportionate factors in the previous iterations are used for the current update of filter coefficients, which leads to a dramatic reduction of the complexity while maintaining the same performance as conventional proportionate APA. Based on the efficient recursive implementation of MIPAPA, this chapter presents two new algorithms, μ-law MIPAPA (MMIPAPA) and Segment MIPAPA (SMIPAPA), proposed by us in [43], which achieve a significant improvement in the convergence speed with a low computational complexity by incorporating the μ-law proportionate technique into MIPAPA.

4.2 Background of the μ-Law Proportionate Technique

We have discussed NLMS in the beginning of this thesis. The Affine Projection Algorithm proposed by Ozeki in [41] can be viewed as a generalization of the NLMS algorithm. Define $\mathbf{X}(n)=[\mathbf{x}(n),\mathbf{x}(n-1),\ ...,\mathbf{x}(n-P+1)]$ as the input vector with projection order *P*, with $\mathbf{x}(n)$ defined as in Section 1.2.1. The error vector $\mathbf{e}(n)=[e(n),e(n-1),\ ...,e(n-P+1)]^T$ is then given by $\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\hat{\mathbf{h}}(n-1)$, where $\mathbf{d}(n)=[d(n),d(n-1),\ ...,d(n-P+1)]^T$ is the desired response. The following update equation is used in APA, where $\mathbf{I}_P$ is a *P* × *P* identity matrix.:

$$\hat{\mathbf{h}}(n)=\hat{\mathbf{h}}(n-1)+\beta\mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{X}(n)+\delta\mathbf{I}_P\right]^{-1}\mathbf{e}(n)\,. \tag{4.1}$$

The proportionate idea of IPNLMS as described in Section 2.2.1 can be applied to APA directly to derive IPAPA [18]:

$$\hat{\mathbf{h}}(n)=\hat{\mathbf{h}}(n-1)+\beta\mathbf{Q}(n-1)\mathbf{X}(n)\left[\mathbf{X}^T(n)\mathbf{Q}(n-1)\mathbf{X}(n)+\delta\mathbf{I}_P\right]^{-1}\mathbf{e}(n)\,. \tag{4.2}$$

$\mathbf{Q}(n)$ is as defined as it is for IPNLMS in (2.7). With *P* equal to 1, IPAPA is equivalent to IPNLMS.

The μ-law proportionate technique was first introduced in MPNLMS [16] using the logarithm of the coefficient magnitude as the step size instead of using its absolute value directly. We incorporate the logarithmic proportionate updating scheme to APA through the use of a modified diagonal step size control matrix $\mathbf{Q}'(n)$ with its elements defined as

$$q'_l(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{F(\hat{h}_l(n))}{2\left\|F(\hat{\mathbf{h}}(n))\right\|_1},\ l = 0, 1, ..., L-1, \tag{4.3}$$

$$F(\hat{h}_l(n)) = \ln(1+\mu\left|\hat{h}_l(n)\right|). \tag{4.4}$$

Typically, a value of μ=1000 is used. ||.||$_1$ in (4.3) denotes the l_1 norm, which is the sum of the absolute value of each vector element.

4.3 Proposed Efficient μ-Law Improved Proportionate Affine Projection Algorithms

In this chapter, we propose an efficient low-cost μ-Law Improved Proportionate Affine Projection Algorithms by adopting the recursive implementation with a memory of previous step size control matrices in MIPAPA [42]. Let us rewrite (4.2) as:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \beta\mathbf{P}(n)\left[\mathbf{X}^T(n)\mathbf{P}(n) + \delta\mathbf{I}_P\right]^{-1}\mathbf{e}(n), \tag{4.5}$$

$$\mathbf{P}(n) = \mathbf{Q}'(n-1)\mathbf{X}(n) = [\mathbf{Q}'(n-1)\mathbf{x}(n), \mathbf{Q}'(n-1)\mathbf{x}(n-1), ..., \mathbf{Q}'(n-1)\mathbf{x}(n-P+1)], \tag{4.6}$$

$$\mathbf{Q}'(n) = \text{diag}\left\{q'_0(n), q'_1(n), ..., q'_{L-1}(n)\right\}. \tag{4.7}$$

Taking account of previous step size control matrices $\mathbf{Q}'(n-2), \mathbf{Q}'(n-3), ..., \mathbf{Q}'(n-P)$, a different $\mathbf{P}(n)$ can be used to update $\hat{\mathbf{h}}(n)$, denoted as $\mathbf{P}'(n)$:

$$\mathbf{P}'(n) = [\mathbf{Q}'(n-1)\mathbf{x}(n), \mathbf{Q}'(n-2)\mathbf{x}(n-1), ..., \mathbf{Q}'(n-P)\mathbf{x}(n-P+1)]. \tag{4.8}$$

It can be observed that $\mathbf{P}'(n)$ can be implemented recursively as follows:

$$\mathbf{P}'(n) = [\mathbf{Q}'(n-1)\mathbf{x}(n), \mathbf{P}'_{-1}(n-1)], \tag{4.9}$$

$$\mathbf{P}'_{-1}(n-1) = [\mathbf{Q}'(n-2)\mathbf{x}(n-1), ..., \mathbf{Q}'(n-P)\mathbf{x}(n-P+1)]. \tag{4.10}$$

In this way, the computations required for the evaluation of $\mathbf{P}'(n)$ are dramatically reduced, from *P* × *L* multiplications down to *L* multiplications. With $q'_l(n)$ defined in (4.3) and (4.4), and $\mathbf{P}'(n)$ defined in (4.9) and (4.10), this new algorithm, termed µ-law Memorized IPAPA (MIPAPA), can be rewritten as follows:

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \beta\mathbf{P}'(n)\left[\mathbf{X}^T(n)\mathbf{P}'(n) + \delta\mathbf{I}_P\right]^{-1}\mathbf{e}(n). \tag{4.11}$$

Due to the computational expense of the logarithmic operations, a piece-wise segment approximation is usually favored in real time applications. Thus, we also propose the Segment Memorized IPAPA (SMIPAPA), in which the following empirical function is used in (4.3), which has been found to give the best approximation of (4.4):

$$F(\hat{h}_l(n)) = \begin{cases} 380\left|\hat{h}_l(n)\right|, & \left|\hat{h}_l(n)\right| < 0.005 \\ 1.9, & \text{otherwise} \end{cases}. \tag{4.12}$$

4.4 Simulations & Results

The setup for the simulations is based on the echo cancellation for a typical network echo path with sparseness *ξ*=0.8 [35]. The length *L* of the adaptive filter needed for

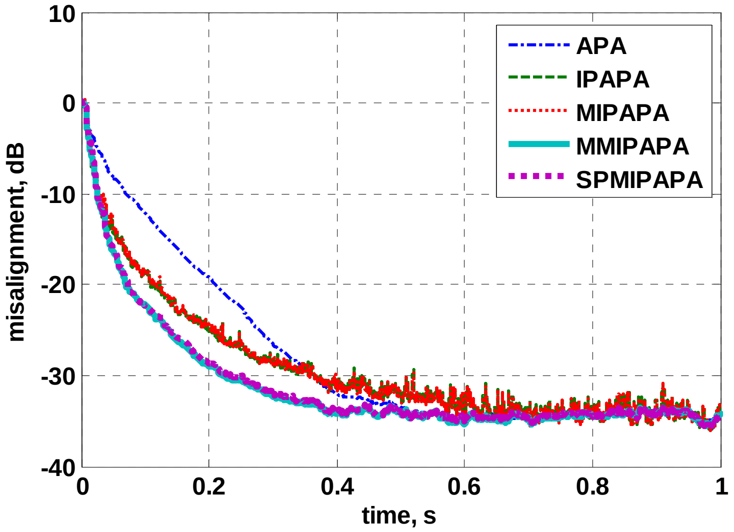
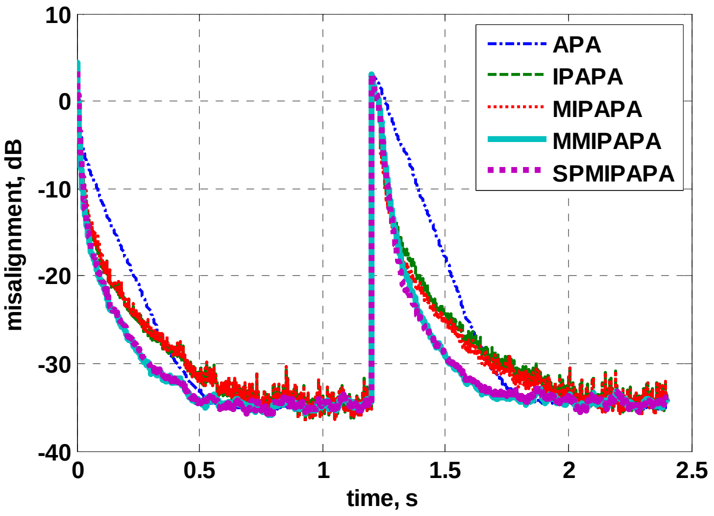


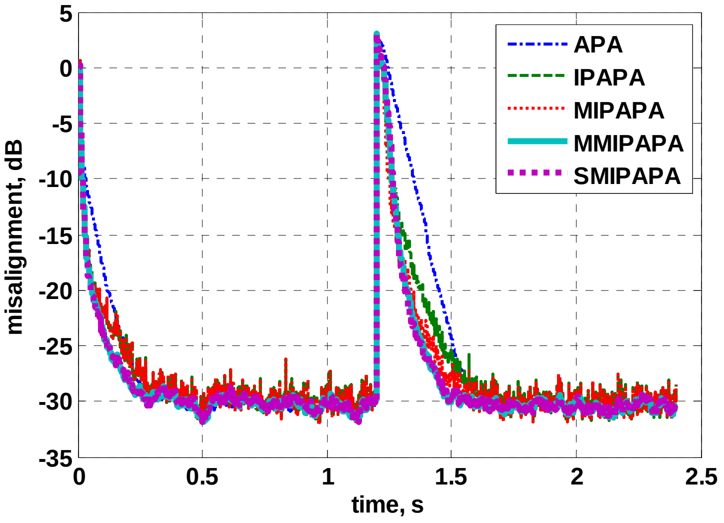
Fig. 4.1: Misalignment performance of adaptive algorithms with projection order *P*=2 for identification of a typical network echo path.

effective echo cancellation is 512. The input signal is generated according to a Gaussian distribution with zero mean and variance 1, and an additive Gaussian white noise is added such that the Signal to Noise Ratio (SNR) is 30 dB. The step size β=0.3 and the adjustable parameter α=0.6 are used for comparisons among IPAPA, MIPAPA, MMIPAPA and SMIPAPA.

Fig. 4.1 compares the convergence performance of APA, IPAPA, MIPAPA with the proposed MMIPAPA and SMIPAPA with projection order *P*=2 for the same steady state misalignment. It can be seen that MMIPAPA achieves improvement in the convergence rate compared with MIPAPA and IPAPA, and SMIPAPA closely follows the behavior of MMIPAPA. Fig. 4.2 further compares the tracking ability of the algorithms with projection orders *P*=2 and *P*=8 for a time varying channel impulse. At time 1.2s, there is an abrupt time shift of 20 samples in echo path. For a high projection



(a)



(b)

Fig. 4.2: Tracking ability of adaptive algorithms for identification of a typical network echo path: (a) *P*=2. (b) *P*=8.

order of *P*=8, MIPAPA converges faster than traditional IPAPA for tracking the time varying echo path, at the expense of additional memory requirements. Both of the proposed algorithms, MMIPAPA and SMIPAPA, perform better than MIPAPA, but the improvement over MIPAPA for *P*=8 is not as large as for *P*=2. While MMIPAPA requires additional L logarithmic functions, L multiplications and L additions per iteration, the extra cost of SMIPAPA is less than L multiplications and comparisons per iteration, compared with MIPAPA.

4.5 Conclusions

By incorporating the μ-law proportionate technique into MIPAPA, we have proposed the MMIPAPA algorithm and its more computationally efficient implementation, SMIPAPA. Simulation results show improved convergence speed for both of these new algorithms. Among affine algorithms of low projection order, SMIPAPA has been shown to be a low cost, efficient and robust method for echo cancellation.

Chapter 5

A Class of Gradient-Controlled Proportionate Algorithms

5.1 Introduction

The Proportionate NLMS (PNLMS) in [14] and the improved version of the PNLMS (IPNLMS) in [15], originally proposed for network echo cancellation, is also applicable to the acoustic echo identification, as well as the frequency domain Improved Proportionate MDF (IPMDF) algorithm which incorporates the proportionate update of IPNLMS with the frequency domain MDF structure. Using concepts based on PNLMS and IPNLMS, numerous proportionate algorithms [16-18, 30-34, 38-40, 42-46] have been developed for echo cancellation. The use of individual step sizes proportional to the magnitude of each filter coefficient leads to faster convergence compared with non-proportionate algorithms. The gain distribution vector used for step size assignments among different filter coefficients is derived using the following intuition: Tap weights that are currently estimated as being far from zero get significantly more update energy than those currently estimated as being close to zero [14]. Although statistical analysis of the steady-state performance of PNLMS is provided in [14], a theoretical derivation of the step size gain distribution vector has been lacking. This is also the case for other proportionate algorithms based on PNLMS.

In this chapter, we formulate an optimization problem with the goal of decreasing the mean square deviation of coefficient estimates between adjacent iterations. By maximizing the goal function with regard to the individual step sizes for each filter tap, we obtain the optimum value of each step size for each filter tap at each iteration. In addition, using a time-averaging gradient vector to approximate the optimum step size gain distribution vector, we proposed a new class of proportionate algorithms for AEC in both the time domain and the frequency domain in [47], referred to as the Gradient-Controlled Proportionate Affine Projection Algorithm (GC-PAPA), the Gradient-Controlled Proportionate Normalized Least Mean Square (GC-PNLMS) algorithm, and the Gradient-Controlled Proportionate Multi-delay Filter (GC-PMDF). Simulation results demonstrate that our proposed algorithms achieve significant improvement in the convergence rate while maintaining the same steady-state error performance, as compared to traditional proportionate algorithms.

The remainder of this chapter is organized as follows. In Section 5.2 we state the optimization problem based on APA and solve for the optimum individual step size for each filter tap, which leads to the proposed GC-PAPA, with GC-PNLMS as a special case when the projection order equals 1. The proposed frequency domain adaptive algorithm GC-PMDF using the new proportionate technique is presented in Section 5.3. Section 5.4 compares the performance of the proposed algorithms with the traditional proportionate algorithms for AEC. Finally, in Section 5.5, the results of our work are summarized and conclusions are drawn.

5.2 Proposed Gradient-Controlled Proportionate Affine Projection Algorithms & Gradient-Controlled Proportionate Normalized Least Mean Square in Time Domain

5.2.1 Optimum Individual Step Size

We have introduced the APA in Chapter 4, which is very effective for coefficient adaptation with colored input signals such as speech. It updates the filter tap weights on the basis of multiple input vectors, and can be viewed as a generalization of the NLMS algorithm. Instead of using step sizes in proportion to the coefficients magnitude as the conventional proportionate APA, we solve for the optimum individual step size for each filter tap at iteration *n* of the training process. Define $\mathbf{X}(n)$ as the input matrix for APA with projection order of *P*: $\mathbf{X}(n) = [\mathbf{x}(n), \mathbf{x}(n-1), ..., \mathbf{x}(n-P+1)]$. The system output, the desired signal and the error signal at iteration *n* are given by:

$$\mathbf{y}(n) = \mathbf{X}^H(n)\mathbf{h}^o, \tag{5.1}$$

$$\mathbf{d}(n) = \mathbf{y}(n) + \mathbf{v}(n) = \mathbf{X}^H(n)\mathbf{h}^o + \mathbf{v}(n), \tag{5.2}$$

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^H(n)\mathbf{h}(n-1), \tag{5.3}$$

where $\mathbf{y}(n)$, $\mathbf{d}(n)$, $\mathbf{v}(n)$ and $\mathbf{e}(n)$ are $P \times 1$ column vectors:

$$\mathbf{y}(n) = [y(n), y(n-1), ..., y(n-P+1)]^T ,$$

$$\mathbf{d}(n) = [d(n), d(n-1), ..., d(n-P+1)]^T ,$$

$$\mathbf{v}(n) = [v(n), v(n-1), ..., v(n-P+1)]^T ,$$

$$\mathbf{e}(n) = [e(n), e(n-1), ..., e(n-P+1)]^T .$$

The update recursion for APA is as follows:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n) . \tag{5.4}$$

The μ in (5.4) is a scalar step size. With the projection order P set to be 1, APA is equivalent to the NLMS algorithm.

To find the optimum step size for each filter tap at iteration *i*, we introduce an $L \times L$ diagonal step size matrix $\mathbf{M}(n) = \mathrm{diag}\left\{\mu_0(n), \mu_1(n), ..., \mu_{L-1}(n)\right\}$:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{M}(n)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n) . \tag{5.5}$$

Equation (5.5) can be expressed in terms of the weight error vector $\tilde{\mathbf{h}}(n) = \mathbf{h}^o - \mathbf{h}(n)$ as:

$$\tilde{\mathbf{h}}(n) = \tilde{\mathbf{h}}(n-1) - \mathbf{M}(n)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n) . \tag{5.6}$$

By squaring both sides of (5.6) and taking expectations, we obtain:

$$\begin{aligned}\mathrm{E}\left\|\tilde{\mathbf{h}}(n)\right\|^2 = \mathrm{E}\left\|\tilde{\mathbf{h}}(n-1)\right\|^2 & -2\Re\left(\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{X}^H(n)\mathbf{M}(n)\tilde{\mathbf{h}}(n-1)\right]\right) \\ & +\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{X}^H(n)\mathbf{M}^2(n)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n)\right] \\ & \triangleq \mathrm{E}\left\|\tilde{\mathbf{h}}(n-1)\right\|^2 - \mathbb{F}(\mu_0(n), \mu_1(n), ..., \mu_{L-1}(n))\end{aligned} \tag{5.7}$$

where $\mathbf{M}^2(n)$ denotes an $L \times L$ diagonal matrix with elements of $\mu_0{}^2(n), \mu_1{}^2(n), ..., \mu_{L-1}{}^2(n)$. With the assumption of $\mu_0(n), \mu_1(n), ..., \mu_{L-1}(n)$ being independent from each other, the optimum individual step size values can be found by maximizing the decrease in the mean square deviation from iteration (*n*-1) to iteration *n* with respect to the step size at each filter tap $\mu_l(n)$, $l = 0, 1, ..., L-1$. It can be formulated as a series of optimization problems with goal functions $\mathbb{F}\left(\mu_l(n)\right)$, $l = 0, 1, ..., L-1$, respectively:

$$\max_{\mu_l(n)} \left\{ \mathrm{E}\left\|\tilde{\mathbf{h}}(n-1)\right\|^2 - \mathrm{E}\left\|\tilde{\mathbf{h}}(n)\right\|^2 \right\} = \max_{\mu_l(n)} \left\{ \mathbb{F}\left(\mu_l(n)\right) \right\}, \quad l = 0, 1, ..., L-1. \tag{5.8}$$

If we assume that the effect of the difference among individual step sizes on the second term in (5.7) is negligible when the error signal is sufficiently small, then the goal function becomes:

$$\mathbb{F}\left(\mu_l(n)\right) = 2\mu_l(n)\Re\left(\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{x}_l(n)\tilde{h}_l(n-1)\right]\right)$$
$$-\mu_l^2(n)\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n)\right] + C. \tag{5.9}$$

The $\mathbf{x}_l(n)$ in (5.9) denotes the l^{th} column vector of $\mathbf{X}^H(n)$, and C represents the terms which are independent of $\mu_l(n)$. Taking the derivative of the goal function with respect to $\mu_l(n)$ leads to the optimum solution for $\mu_l(n)$:

$$\frac{\partial\mathbb{F}\left(\mu_l(n)\right)}{\partial\mu_l(n)} = 2\Re\left(\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{x}_l(n)\tilde{h}_l(n-1)\right]\right)$$
$$-2\mu_l(n)\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n)\right] = 0 \tag{5.10}$$

$$\mu_l^o(n) = \frac{\Re\left(\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{x}_l(n)\tilde{h}_l(n-1)\right]\right)}{\mathrm{E}\left[\mathbf{e}^H(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n)\right]}. \tag{5.11}$$

Assume that the measurement noise is identically and independently distributed, statistically independent of the input $\mathbf{X}(n)$ and sufficiently small. The optimum individual step size can be approximated by replacing $\mathbf{e}(n)$ in (5.11) using (5.3) as:

$$\mu_l^o(i) \approx \frac{\Re\left(\mathrm{E}\left[\tilde{\mathbf{h}}^H(n-1)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{x}_l(n)\tilde{h}_l(n-1)\right]\right)}{\mathrm{E}\left[\tilde{\mathbf{h}}^H(n-1)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{X}^H(n)\tilde{\mathbf{h}}(n-1)\right]+V}, \tag{5.12}$$

where V is defined as $V=\sigma_v{}^2\mathrm{Tr}\left\{\mathrm{E}\left[\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\right]\right\}$. According to (5.4), the gradient vector of APA at iteration n is

$$\mathbf{g}(n)=\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n). \tag{5.13}$$

It can be seen that, when $\mathbf{v}(n)$ is assumed to be zero, equation (5.13) becomes:

$$\mathbf{g}(n)=\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{X}^H(n)\tilde{\mathbf{h}}(n-1). \tag{5.14}$$

The term $\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{X}^H(n)$ can be viewed as a projection matrix onto $\mathbb{R}\left(\mathbf{X}(n)\right)$, which is the range space of $\mathbf{X}(n)$. Thus, $\mathbf{g}(n)$ can be regarded as a projection of $\tilde{\mathbf{h}}(n-1)$ onto $\mathbb{R}\left(\mathbf{X}(n)\right)$, and (5.12) can be rewritten as

$$\mu_l^o(n) \approx \frac{\Re\left(\mathrm{E}\left[g_l^*(n)\tilde{h}_l(n-1)\right]\right)}{\mathrm{E}\left\|\mathbf{g}(n)\right\|^2+V}, \tag{5.15}$$

where $g_l(n)$ denotes the l^{th} element of the gradient vector $\mathbf{g}(n)$. This quantity is thus the optimum individual step size for the l^{th} filter tap at iteration n.

5.2.2 Proposed Gradient-Controlled Proportionate APA & NLMS

The problem in calculating $\mu_l^o(n)$ in (5.15) is that $\tilde{h}_l(n-1)$ is unknown during the convergence process. Since our objective is to find the gain distribution vector to assign the individual step size for each filter tap, we are only interested in the optimum step size ratio among different filter taps at each iteration. It can be observed from (19) that the optimum individual step size is related to the l^{th} element of the expectation of $\mathbf{g}(n)$, during both the initial adaptation stage when $\tilde{\mathbf{h}}(n)$ is large, when $\mathbf{g}(n)$ can be regarded as a projection of $\tilde{\mathbf{h}}(n-1)$, and after the initial adaptation stage when there is less difference in $\tilde{h}_l(n)$ over different filter taps as the adaptive filter converges. Therefore, we propose a new proportionate technique to distribute the adaptation energy during each iteration, in which the individual step size for the next update of the l^{th} filter tap is proportional to the magnitude of the l^{th} element of the expectation of the gradient vector. The expectation of $\mathbf{g}(n)$ can be estimated by averaging $\mathbf{g}(n)$ over time as:

$$\hat{\mathbf{g}}(n) = \beta\hat{\mathbf{g}}(n-1) + (1-\beta)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n). \tag{5.16}$$

where β is used as a forgetting factor with the range $\beta \in (0,1)$. Using this approximation, we propose a new class of proportionate APA, called Gradient-controlled Proportionate APA (GC-PAPA), where the filter taps are updated recursively as follows:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mu\mathbf{Q}(n)\mathbf{X}(n)\left(\mathbf{X}^H(n)\mathbf{Q}(n)\mathbf{X}(n)\right)^{-1}\mathbf{e}(n), \tag{5.17}$$

$$\mathbf{Q}(n) = \operatorname{diag}\{\mathbf{q}(n)\},\ \mathbf{q}(n) = \left[q_0(n), q_1(n), \ldots, q_{L-1}(n)\right]^T, \tag{5.18}$$

$$q_l(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{g}_l(n-1)\right|}{2\left\|\hat{\mathbf{g}}(n-1)\right\|_1 + \varepsilon},\ \ l = 0, 1, \ldots, L-1. \tag{5.19}$$

$\mathbf{Q}(n)$ in (5.17) is an $L \times L$ diagonal matrix which assigns individual step sizes for each filter tap in each iteration. ε in (5.19) is a small value to prevent dividing by zero, and $\hat{\mathbf{g}}(n)$ is calculated as in (5.16). The adjustable combination of both non-proportionate and proportionate terms via the parameter $\alpha \in [-1,+1]$ is used to improve the robustness for signal identification with various degrees of sparseness. We can see that for GC-PAPA, if $\alpha = 1$ the adaptation energy for every update is distributed over the filter taps in proportion to the magnitude of the corresponding elements of the averaged gradient vector $\hat{\mathbf{g}}(n)$.

With projection order $P = 1$, we derive the Gradient-Controlled Proportionate NLMS (GC-PNLMS) algorithm as a special case of GC-PAPA, which can be summarized as follows:

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \frac{\mu \mathbf{Q}(n)\mathbf{x}(n)e(n)}{\mathbf{x}^H(n)\mathbf{Q}(n)\mathbf{x}(n)}, \tag{5.20}$$

$$\mathbf{Q}(n) = \mathrm{diag}\left\{\mathbf{q}(n)\right\},\ \mathbf{q}(n) = \left[q_0(n), q_1(n), \ldots, q_{L-1}(n)\right]^T, \tag{5.21}$$

$$q_l(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{g}_l(n-1)\right|}{2\left\|\hat{\mathbf{g}}(n-1)\right\|_1 + \varepsilon},\ \ l = 0, 1, \ldots, L-1, \tag{5.22}$$

$$\hat{\mathbf{g}}(n) = \beta\hat{\mathbf{g}}(n-1) + (1-\beta)\mathbf{x}(n)\left(\mathbf{x}^H(n)\mathbf{x}(n)\right)^{-1} e(n). \tag{5.23}$$

The relation between the proposed gradient controlled proportionate technique and the traditional proportionate technique can be seen by comparing the step size control matrix $\mathbf{Q}(n)$ of GC-PNLMS and IPNLMS. IPNLMS uses $\left|\mathbf{h}(n)\right|$ as the gain distribution vector to control the individual step sizes over different filter taps for the next coefficient updates, while GC-PNLMS uses $\left|\hat{\mathbf{g}}(n)\right|$ as the gain distribution vector. With the initial value $\mathbf{h}(0)$ and $\hat{\mathbf{g}}(0)$ set to $\mathbf{0}$ and α set to 1 in (2.7) and (5.22), (2.3) becomes:

$$\mathbf{h}(n) = \mu \sum_{i=1}^{n} \mathbf{Q}(i)\mathbf{x}(i)\left(\mathbf{x}^{H}(i)\mathbf{Q}(i)\mathbf{x}(i)\right)^{-1} e(i), \tag{5.24}$$

and (5.23) becomes:

$$\hat{\mathbf{g}}(n) = (1-\beta)\sum_{i=1}^{n} \beta^{n-i}\mathbf{x}(i)\left(\mathbf{x}^{H}(i)\mathbf{x}(i)\right)^{-1} e(i). \tag{5.25}$$

By comparing (5.24) and (5.25), it can be observed that the IPNLMS uses a weighted average of the gradient vector as the gain distribution vector, while the proposed GC-PNLMS introduces a forgetting factor *β* to average the gradient vector as the gain distribution vector. Simulation results presented in Section 5.4 will show that the proposed GC-PNLMS outperforms IPNLMS, especially after the initial convergence period due to the introduction of *β*.

5.3 Proposed Gradient-Controlled Proportionate Multi-Delay Filter in Frequency Domain

The frequency domain adaptive filter is another attractive choice for AEC due to its efficient implementation and good performance for colored input signals. In Chapter 3, we have reviewed the Multi-Delay Filter (MDF) where the *L*-tap frequency domain adaptive filter is partitioned into an arbitrary number *K* of sub-filters with each having a smaller block size *N* so that $L = KN$. The inherent delay problem of frequency domain adaptive filters thus have been addressed by MDF, and simulations in [37] show that MDF achieves faster convergence compared to the time domain NLMS algorithm. To enhance its performance, IPMDF was developed as we have discussed in Chapter 3. In this chapter, using the theoretical results we have derived in Section 5.2, the Gradient-Controlled Proportionate MDF (GC-PMDF) is proposed to further improve the convergence speed of frequency domain adaptive filters.

Continued with the introduction of MDF in Section 3.2.1, let us rewrite the MDF update equations here again:

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{d}}(m) - \mathbf{G}^{01} \sum_{k=0}^{K-1} \underline{\mathbf{X}}(m-k)\, \underline{\mathbf{h}}_k(m-1), \tag{5.26}$$

$$\mathbf{S}(m) = \lambda\, \mathbf{S}(m-1) + (1-\lambda)\, \underline{\mathbf{X}}^*(m)\underline{\mathbf{X}}(m), \tag{5.27}$$

$$\underline{\mathbf{h}}_k(m) = \underline{\mathbf{h}}_k(m-1) + \mu \mathbf{G}^{10} \underline{\mathbf{X}}^*(m-k)\left[\mathbf{S}(m) + \delta\right]^{-1} \underline{\mathbf{e}}(m),\ k = 0, 1, ..., K-1. \tag{5.28}$$

m is the block iteration index, k is the sub-filter index, and $0 \le \lambda \le 1$ is the smoothing factor. δ and $\mathbf{S}(m)$ are regularization parameters. $\mu = \beta(1-\lambda)$ is the step size for MDF with $0 < \beta \le 1$ as in. The update scheme in (5.28) can be written as

$$\underline{\mathbf{h}}_k(m) = \underline{\mathbf{h}}_k(m-1) + \mu \mathbf{G}_1 \mathbf{G}_2 \underline{\mathbf{X}}^*(m-k)\left[\mathbf{S}(m) + \delta\right]^{-1} \underline{\mathbf{e}}(m), \tag{5.29}$$

where $\mathbf{G}_1 = \mathbf{F}\begin{bmatrix} \mathbf{I}_{N\times N} \\ \mathbf{0}_{N\times N} \end{bmatrix}$ and $\mathbf{G}_2 = \begin{bmatrix} \mathbf{I}_{N\times N} & \mathbf{0}_{N\times N} \end{bmatrix} \mathbf{F}^{-1}$ so that the product of $\mathbf{G}_1\mathbf{G}_2$ equals to $\mathbf{G}^{10}$ in (5.28). To distribute the adaptation energy over different filter taps at each block iteration, the following step size control matrix of $\mathbf{Q}_k(m)$, $k = 0, 1, ..., K-1$, is introduced to update each sub-filter in GC-PMDF:

$$\underline{\mathbf{h}}_k(m) = \underline{\mathbf{h}}_k(m-1) + L\mu \mathbf{G}_1 \mathbf{Q}_k(m) \mathbf{G}_2 \underline{\mathbf{X}}^*(m-k)\left[\mathbf{S}(m) + \delta\right]^{-1} \underline{\mathbf{e}}(m), \tag{5.30}$$

$$\mathbf{Q}_k(m) = diag\left\{\mathbf{q}_k(m)\right\},\ \mathbf{q}_k(m) = [q_{kN}(m), ..., q_{kN+N-1}(m)], \tag{5.31}$$

$$q_{kN+n}(m) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{g}_{kN+n}(m-1)\right|}{2\left\|\hat{\mathbf{g}}(m-1)\right\|_1 + \varepsilon},\ n = 0, 1, ..., N-1,\ \ k = 0, 1, ..., K-1. \tag{5.32}$$

ε is a small value to prevent dividing by zero, and $\hat{\mathbf{g}}(m)$ is a time averaging estimate of the expectation of the gradient vector:

$$\hat{\mathbf{g}}(m) = \left[\hat{\mathbf{g}}_0(m), ..., \hat{\mathbf{g}}_{K-1}(m)\right], \tag{5.33}$$

$$\hat{\mathbf{g}}_k(m) = \beta \hat{\mathbf{g}}_k(m-1) + (1-\beta)\mathbf{G}_2 \underline{\mathbf{X}}^*(m-k)\left[\mathbf{S}(m) + \delta\right]^{-1} \underline{\mathbf{e}}(m),\ k = 0, 1, ..., K-1, \tag{5.34}$$

with $\beta \in (0,1)$ used as a forgetting factor, and δ and $\mathbf{S}(m)$ as the regularization parameters. Similar to GC-PAPA, we use the combination of both non-proportionate and proportionate updates for each sub-filter with a controllable parameter $\alpha \in [-1, +1]$. With $\alpha = -1$, GC-PMDF is equivalent to MDF, while $\alpha = 1$ leads to the proportionate update where each individual step size is assigned to be proportional to the magnitude of the corresponding element of $\hat{\mathbf{g}}(m)$. To achieve the same steady state error, the regularization parameters $\mathbf{S}(0)$ and δ in (5.30) and (5.34) need to be scaled by ${}^{(1-\alpha)}\!/_{2}$ compared to MDF. As will be described in the following section, simulations show significant improvements in the convergence speed of GC-PMDF over MDF and IPMDF for both white input and colored inputs in the AEC application.

5.4 Simulations & Results

The simulations are performed within the context of AEC considering the single talk scenario shown in Fig. 1.2. With a sampling rate of 8 kHz, the measured acoustic echo impulse response is truncated to the first 512 coefficients, as plotted in Fig. 5.1. Our

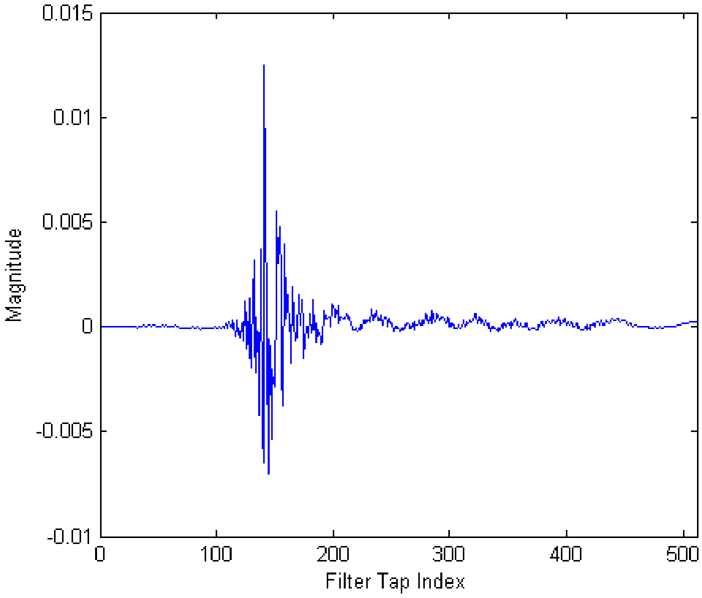
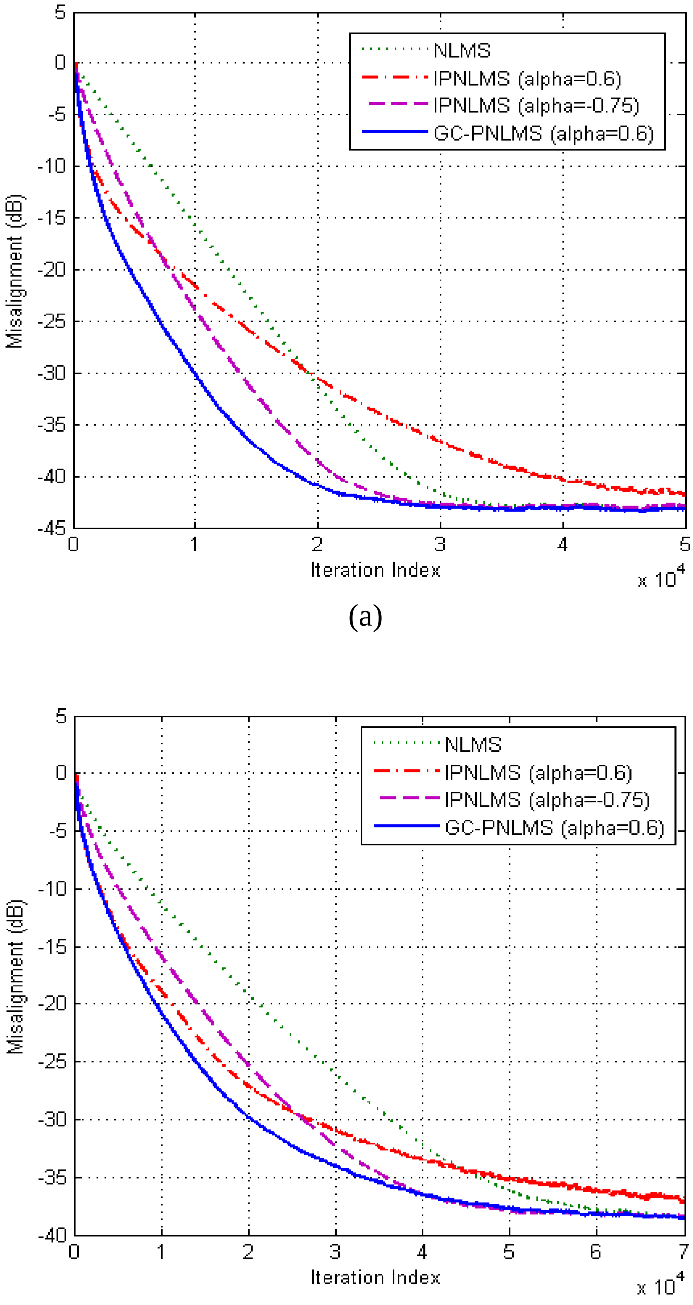


Fig. 5.1: A typical room acoustic echo path impulse response truncated to the first 512 coefficients.

proposed GC-PAPA/GC-PNLMS and GC-PMDF are compared with the traditional proportionate algorithms in the time domain and in the frequency domain, respectively. Their performances are again evaluated in terms of the normalized misalignment, which is defined as in (2.16). Each learning curve is produced by ensemble averaging over 10 independent trials. Two signals are considered as possible inputs, a white zero-mean Gaussian signal and a colored signal which is generated by filtering a white zero-mean Gaussian random sequence through a first-order system having a transfer function $H(z) = \frac{1}{1-0.8z^{-1}}$. An independent white Gaussian measurement noise $v(i)$ is added to the output so that the Signal-to-Noise Ratio (SNR), as calculated below, is 30 dB:

$$\text{SNR} = 10\log_{10}\left(\frac{\text{E}\left[y^2(i)\right]}{\text{E}\left[v^2(i)\right]}\right) \text{ (dB)}. \tag{5.35}$$



(a)

(b)

Fig. 5.2: Misalignment for NLMS, IPNLMS and GC-PNLMS. (a) White input, μ=0.1, β=0.999. (b) Colored input, μ=0.3, β=0.9999.

5.4.1 Performance of Time Domain Proportionate Algorithms

For the time domain adaptive filters, the filter length L is set to be 512 for the identification of the unknown echo path in Fig. 5.1. First, the simulation is performed for a white zero-mean Gaussian input signal with variance 1. The same step size $\mu = 0.1$ is used for comparisons among different algorithms, and the regulation parameters are set to achieve the same steady-state misalignment. With the projection order $P = 1$, the performance of the proposed GC-PNLMS is compared with the NLMS and the traditional proportionate algorithm IPNLMS in Fig. 5.2 (a). For IPNLMS, the choice of the controllable parameter α is limited to small values to ensure the overall convergence. Simulations in [15] show that $\alpha = -0.75$ produces the best trade-off between the initial convergence rate and the overall convergence. It can be seen from Fig. 5.2 (a) that when a larger value $\alpha = 0.6$ is used for IPNLMS, a faster initial convergence rate is achieved, but the performance degrades after the initial stage and the overall convergence is even worse than for NLMS. With the use of a forgetting factor of $\beta = 0.999$ in (5.23), our proposed GC-PNLMS allows larger values of α and thus can obtain a significant improvement in the convergence rate by as much as 5 dB compared to IPNLMS with $\alpha = -0.75$, and 15 dB compared to NLMS through the entire adaptation process. Similar results can be seen in Fig. 5.2 (b), where the simulation is performed using the colored input signal. In the case of a highly correlated input, the step size is adjusted to be $\mu = 0.3$ to speed up the simulations, and $\beta = 0.9999$ is used in (5.23) to achieve the same misalignment as NLMS. It can be seen from Fig. 5.2 (b) that the proposed GC-PNLMS with $\alpha = 0.6$ maintains the fastest convergence rate during

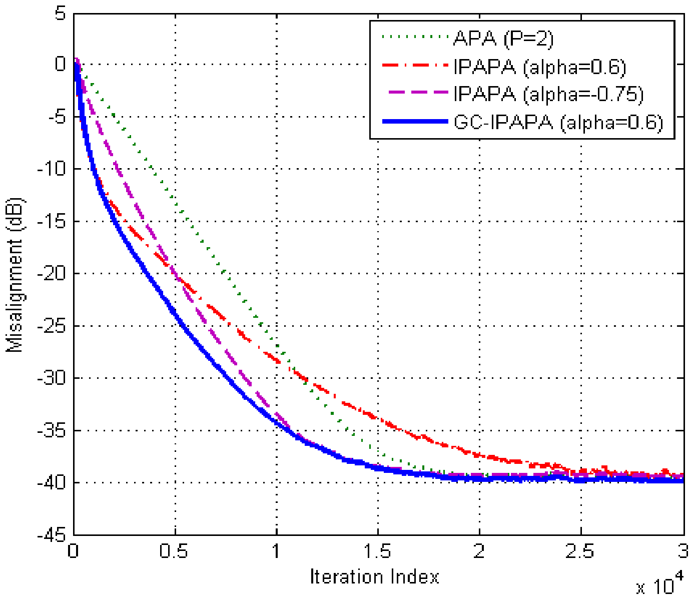


Fig. 5.3: Misalignment for APA, IPAPA and GC-PAPA with white input. *P*=2, μ=0.1, β=0.999.

the entire convergence process for the colored input signal as well, while IPNLMS with $\alpha = 0.6$ quickly degrades and performs poorly as compared with NLMS.

Simulation results in Figs. 5.3-5.5 demonstrate the performance of the proposed GC-PAPA and the traditional IPAPA with projection orders $P = 1, 2, 4, 8$ with a white zero-mean Gaussian input. The step size is set to be $\mu = 0.1$, and the misalignment curve is plotted for APA, IPAPA with $\alpha = 0.6$ and $\alpha = -0.75$, and for the proposed GC-PAPA with $\alpha = 0.6$ and $\beta = 0.999$. It is found that GC-PAPA achieves the fastest convergence rate among these algorithms with the same steady-state misalignment, although only a slight improvement is observed as compared to IPAPA with $\alpha = -0.75$ when the projection order P is higher than 4.

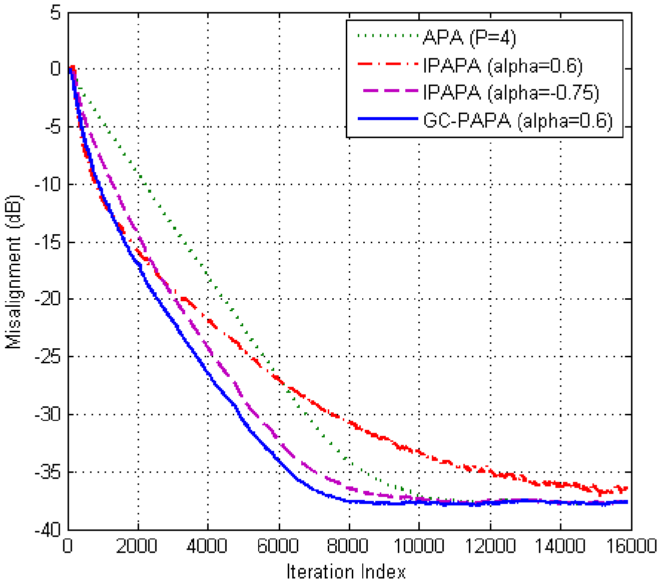


Fig. 5.4: Misalignment for APA, IPAPA and GC-PAPA with white input. *P*=4, μ=0.1, β=0.999.

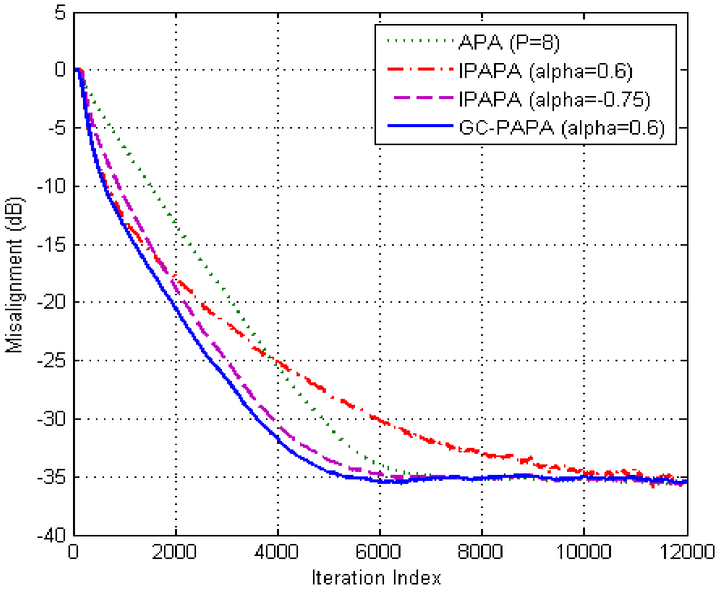


Fig. 5.5: Misalignment for APA, IPAPA and GC-PAPA with white input. *P*=8, μ=0.1, β=0.999.

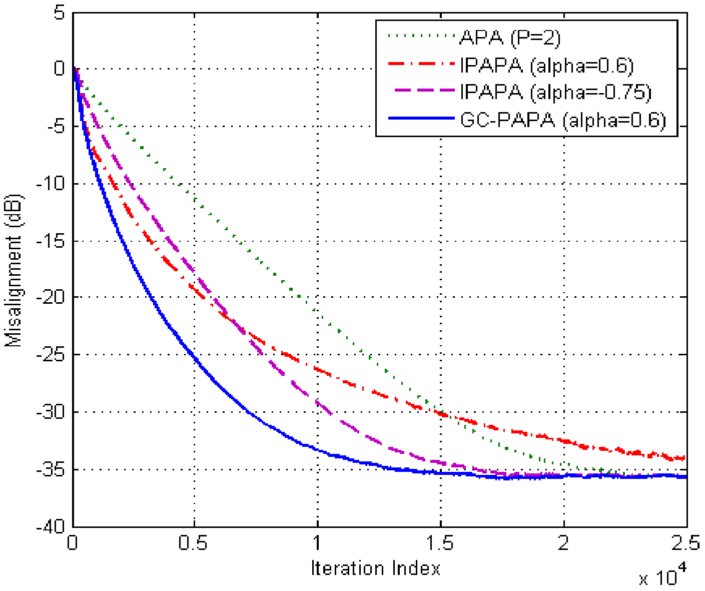


Fig. 5.6: Misalignment for APA, IPAPA and GC-PAPA with colored input. P=2, μ=0.15, β=0.9999.

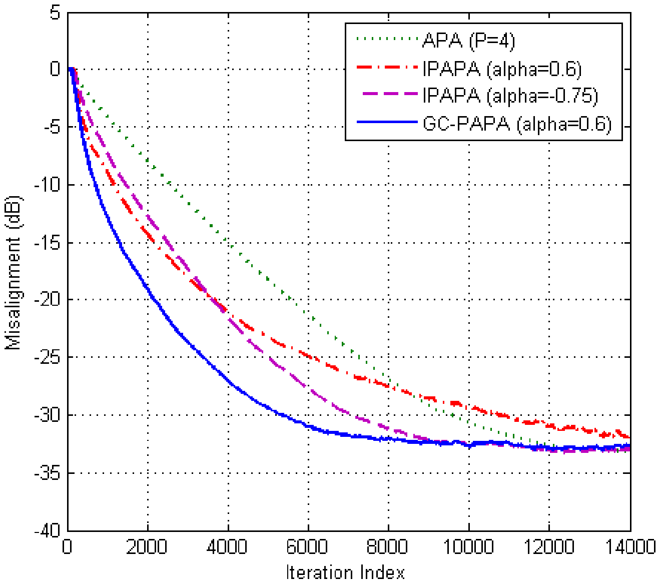


Fig. 5.7: Misalignment for APA, IPAPA and GC-PAPA with colored input. P=4, μ=0.15, β=0.9999.

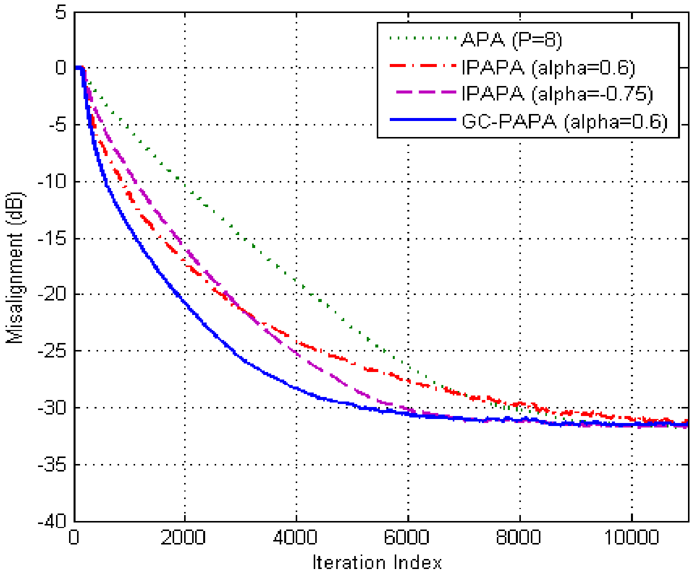


Fig. 5.8: Misalignment for APA, IPAPA and GC-PAPA with colored input. P=8, μ=0.15, β=0.9999.

In the case of the colored input, the advantage of GC-PAPA is greater than that for the white input, as shown in Figs. 5.6-5.8. A step size of $\mu = 0.15$ is used for all of the algorithms and the forgetting factor is set to be $\beta = 0.9999$ for GC-PAPA. As seen in the simulation results, with the same steady-state misalignment, significant improvements in convergence speed are achieved by the proposed GC-PAPA as compared with IPAPA for projection orders as high as $P = 8$, which makes GC-PAPA a superior choice for AEC for which the input is usually a highly correlated speech signal.

5.4.2 Performance of Frequency Domain Proportionate Algorithms

In addition to the evaluation of the time domain GC-PAPA, simulations are also performed to evaluate the performance of the proposed frequency domain GC-PMDF as compared with MDF and IPMDF. The filter length L is set to be 512. Although a smaller sub-filter block size N leads to a faster convergence rate (since the coefficients of the adaptive filter update more frequently as N becomes smaller), it has been shown in [38] that when $N < 64$ additional decreases in N produce little improvement in the convergence rate. Thus, in our simulations the sub-filter block size N is set to 64, and the number K of sub-filters in the frequency domain is set to 8. The following parameters are used in the simulations to achieve the same misalignment among different algorithms as in [38]:

$$\lambda = \left[1 - \frac{1}{3L}\right]^N, \ \mu = 1 - \lambda, \ \beta = 0.9, \ \delta_{MDF} = \frac{\sigma_x{}^2 20N}{L}, \ \delta_{GC-PMDF} = \delta_{IPMDF} = \frac{1-\alpha}{2}\delta_{MDF},$$

$$\mathbf{S}_{MDF}(0) = \frac{\sigma_x{}^2}{100}, \ \mathbf{S}_{GC-PMDF}(0) = \mathbf{S}_{IPMDF}(0) = \frac{1-\alpha}{2}\mathbf{S}_{MDF}(0).$$

The same white and colored inputs used for the time domain adaptive filters are applied as inputs to the frequency domain adaptive filters, and the simulation results are shown in Figs. 5.9 and 5.10, respectively. For IPMDF, $\alpha = -0.75$ is the choice that achieve the greatest overall performance. When a larger α is used the IPMDF algorithm becomes unstable. Thus, the misalignment curves are plotted for IPMDF with $\alpha = -0.75$ and $\alpha = -0.2$ for comparison. Additional simulations not shown here indicate that $\alpha = 0.4$ is the best choice for GC-PMDF, although only slight performance differences are observed when using other positive values of α. The proposed GC-PMDF accelerates the convergence rate by allowing a greater proportionate update in (5.32), while obtaining the same final misalignment for both the white input and the colored

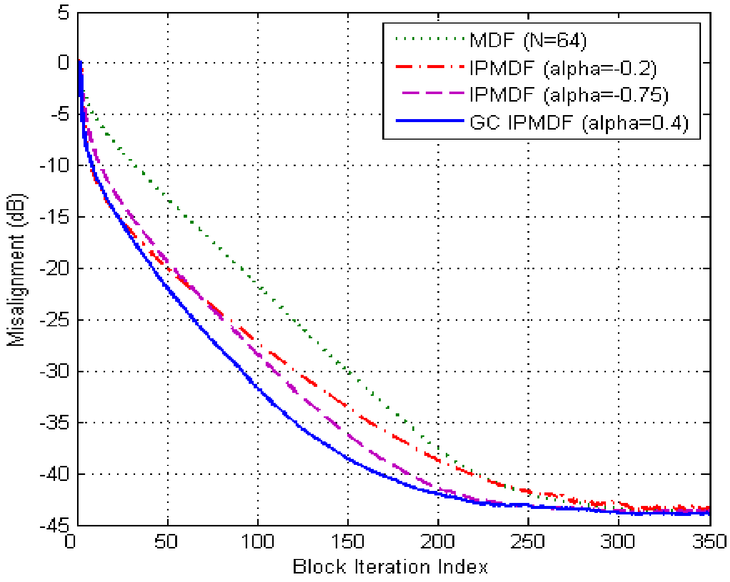


Fig. 5.9: Misalignment for MDF, IPMDF, and GC-PMDF with white input. N=64, K=8, β=0.99.

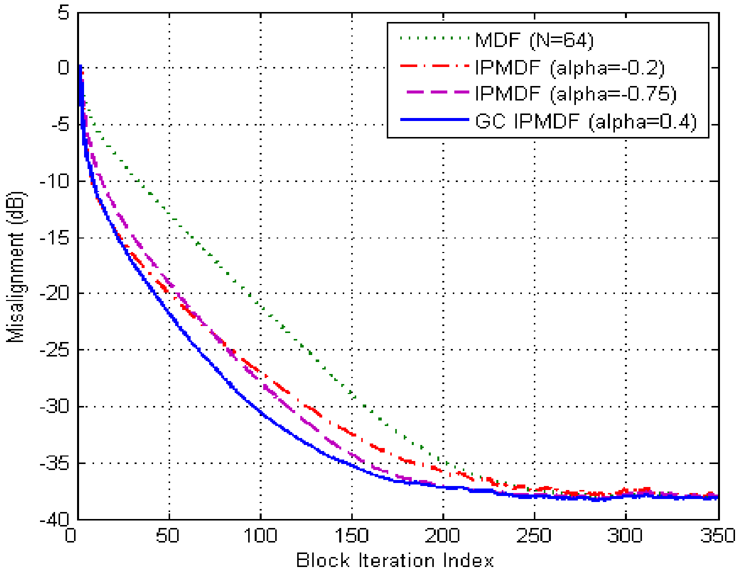


Fig. 5.10: Misalignment for MDF, IPMDF, and GC-PMDF with colored input. N=64, K=8, β=0.99.

input. Therefore, the proposed GC-PMDF is more robust and faster than IPMDF for AEC.

5.5 Conclusions

In this chapter, the optimum individual step sizes have been derived for each filter tap in the AEC scenario. Based on these theoretical results, a new class of proportionate algorithms has been proposed in both the time domain and the frequency domain. Using a time-averaging gradient vector as the gain distribution vector, individual adaptation energies are assigned to the filter taps at each iteration. The proposed GC-PAPA, GC-PNLMS and GC-PMDF allow the use of a larger value of α, and thus achieve significant improvements in the convergence rate compared with traditional IPAPA, IPNLMS, and IPMDF, as demonstrated in the simulation results. This new class of proportionate algorithms is more robust and converges faster than traditional proportionate algorithms, especially when the input is highly correlated, as in AEC. Other algorithms that are based on the traditional proportionate update of PNLMS and IPNLMS can also potentially be improved by incorporating this new gradient-controlled proportionate technique.

Chapter 6

Sparse LMS with Segment Zero Attractors for Sparse Signal Recovery

6.1 Introduction

Different from the sparse echo path with a small number of coefficients having large magnitude, the signals to be recovered in compressive sensing usually have only a small number of nonzero values. Many practical channels exhibit this sparse character, where conventional LMS and NLMS cannot produce acceptable performance. This motivates the design of efficient adaptive filters exploiting the sparse nature of the signal to be estimated.

Inspired by the recent research on the least absolute shrinkage and selection operator (LASSO) [11] and compressive sensing [12-13], Ref. [48-49] propose a modified LMS by incorporating the l_1 norm into the quadratic cost function of the conventional LMS. It results in a zero attractor for all filter coefficients during the updating process, and is named zero-attracting LMS (ZA-LMS). A reweighted zero-attracting LMS (RZA-LMS) is also proposed in [48]. With different reweighted zero attractors for different filter taps, RZA-LMS selectively induces filter taps with small magnitude to zero rather than uniformly attract all filter taps to zero. This leads to a better performance compared with ZA-LMS. In this chapter, based on RZA-LMS, we propose two new types of sparse LMS with segment zero attractors as in [50]. The first one is referred to as Segment RZA-LMS (S-RZA-LMS), where a piece-wise approximation is used instead of the reciprocal in the iterative updating equation of RZA-LMS. It allows for more freedom when choosing different zero-attractors for filter taps and thus achieves a faster convergence rate and a lower steady state error floor. Furthermore, a Discrete Segment RZA-LMS (DS-RZA-LMS) is developed using an inverse slope segment function of the piece-wise approximation for filter taps with small magnitude. Simulation results show further improvement of DS-RZA-LMS over S-RZA-LMS. Our two proposed algorithms outperform the conventional RZA-LMS for both white inputs and correlated inputs, and they also eliminate the additional division operations found in the RZA-LMS algorithm.

The remainder of this chapter is organized as follows. Section 6.2 briefly reviews the ZA-LMS and RZA-LMS algorithms. In Section 6.3, our proposed algorithms are presented and numerical simulation results are provided in Section 6.4, including comparisons among the different algorithms. Section 6.5 gives the conclusions.

6.2 Background of the Zero Attracting Algorithms

6.2.1 Review of the Conventional LMS

To briefly review the conventional LMS algorithm, consider a linear system with the actual system response $\mathbf{w}_{\text{opt}}$ and the input vector $\mathbf{x}(n)$ defined as follows:

$$\mathbf{w}_{\text{opt}} = [w_0, w_1, \ldots, w_{L-1}]^T,$$

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T,$$

where n is the time index and L is the length of the adaptive filter. The desired response $d(n)$ is given by:

$$d(n) = \mathbf{w}_{\text{opt}}{}^T \mathbf{x}(n) + v(n), \tag{6.1}$$

with $v(n)$ denoting the additive Gaussian white noise. The adaptive filter is used to estimate the unknown system response $\mathbf{w}_{\text{opt}}$ based on the input signal $\mathbf{x}(n)$ and the desired signal $d(n)$. Let $\mathbf{w}(n)$ be the estimated coefficients at time n, $\mathbf{w}(n) = [w_0(n), w_1(n), \ldots, w_{L-1}(n)]^T$. The conventional LMS algorithm is derived by minimizing a cost function defined as:

$$J(n) = \frac{1}{2} e^2(n), \tag{6.2}$$

where $e(n)$ is the instantaneous error:

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n). \tag{6.3}$$

Using the gradient descent updating, the filter coefficients are updated as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{\partial J(n)}{\partial \mathbf{w}(n)} = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n). \tag{6.4}$$

μ is the step size controlling the trade off between convergence rate and steady-state behaviour of the algorithm.

6.2.2 ZA-LMS & RZA-LMS

Although LMS is the most commonly used algorithm for adaptive signal recovery, it is not well suited for the case when the signal is relative sparse. ZA-LMS exploits the sparseness by introducing the l_1 norm of filter coefficients in the quadratic cost function of LMS as shown below:

$$J_1(n) = \frac{1}{2}e^2(n) + \gamma \left\| \mathbf{w}(n) \right\|_1 . \tag{6.5}$$

γ is a controllable parameter that determines the degree of zero attraction of the l_1 norm for the filter coefficients. Following the gradient descent updating, the update equation is then modified to be:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{\partial J_1(n)}{\partial \mathbf{w}(n)} = \mathbf{w}(n) - \rho\,\mathrm{sgn}(\mathbf{w}(n)) + \mu e(n)\mathbf{x}(n) , \tag{6.6}$$

where $\rho = \mu\gamma$ and sgn(.) is a component-wise sign function defined as:

$$\mathrm{sgn}(x) = \begin{cases} x/|x| & x \neq 0 \\ 0 & x = 0 \end{cases} . \tag{6.7}$$

Compared (6.6) with the update equation (6.4), it can be observed that the additional term $-\rho\,\mathrm{sgn}(\mathbf{w}(n))$ of ZA-LMS, named zero attractor in [48], tends to always shrink the filter taps to zero. It leads to faster initial convergence rate when the actual system response is mainly zero.

While in ZA-LMS the same zero attractor is uniformly used to update all filter taps, RZA-LMS uses individual zero attractors for different filter taps. The reweighted zero attractors are due to the use of a log-sum penalty function in the quadratic cost function, which behaves more like the l_0 norm than the l_1 norm:

$$J_2(n) = \frac{1}{2}e^2(n) + \gamma' \sum_{i=1}^{L} \log(1 + \frac{|w_i|}{\varepsilon'}) . \tag{6.8}$$

The ε' is used as a threshold value that RZA-LMS uses to selectively shrink filter taps with small magnitudes. The zero attractors only affect the taps that are comparable to ε', and there is little zero attraction for the taps with $|w_i| >> \varepsilon'$. According to the gradient descent updating, the coefficients are updated by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{\partial J_2(n)}{\partial \mathbf{w}(n)} = \mathbf{w}(n) - \frac{\rho \operatorname{sgn}(\mathbf{w}(n))}{1+\varepsilon|\mathbf{w}(n)|} + \mu e(n)\mathbf{x}(n) \tag{6.9}$$

with $\rho = {}^{\mu\gamma'}\!/_{\varepsilon'}$ and $\varepsilon = {}^{1}\!/_{\varepsilon'}$.

6.3 Proposed Algorithms with Segment Zero Attractors

Based on the iterative updating equation (6.9) of RZA-LMS algorithm, we propose Segment RZA-LMS (S-RZA-LMS) which uses a piece-wise approximation of the reciprocal term $\frac{1}{1+\varepsilon|\mathbf{w}(n)|}$ to make more use of the sparseness. As shown in Fig. 1, the following piece-wise linear function $f_1(w_i)$ can be a fairly close resemblance of the reciprocal term in (6.9):

$$f_1(w_i) = \begin{cases} -4|w_i| + 1 & \text{if } w_i < 0.22 \\ 0.12 & \text{otherwise} \end{cases}, \tag{6.10}$$

for $i = 1, 2, ..., L$. The filter coefficients are then updated as follows:

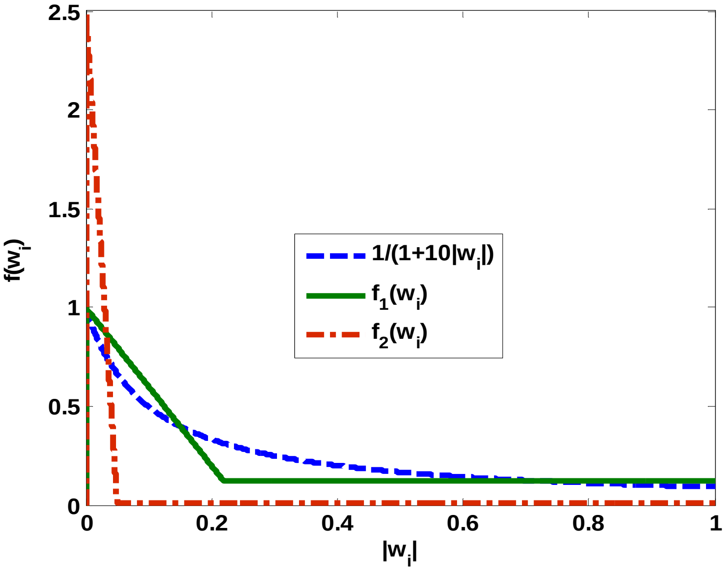


Fig. 6.1: Piece-wise functions used for Segment RZA-LMS.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - f_1(\mathbf{w}(n))\rho\,\mathrm{sgn}(\mathbf{w}(n)) + \mu e(n)\mathbf{x}(n)\,, \tag{6.11}$$

with a segment zero attractor vector $-f_1(\mathbf{w}(n))\rho\,\mathrm{sgn}(\mathbf{w}(n))$. The simulation results in Section 6.4 show that by using this approximation, the adaptation process of S-RZA-LMS follows close enough to the convergence of the RZA-LMS.

The S-RZA-LMS allows for more freedom when assigning different zero attractors for filter taps. Its performance can be further improved when the piece-wise function in (6.11) is carefully chosen. Fig. 6.1 gives another choice of $f_2(w_i)$, which is defined as:

$$f_2(w_i) = \begin{cases} -50|w_i| + 2.51 & \text{if } w_i < 0.05 \\ 0.01 & \text{otherwise} \end{cases}, \tag{6.12}$$

for $i = 1, 2, ..., L$. Using $f_2(w_i)$ in (6.11) instead of $f_1(w_i)$ can lead to a series of zero attractors which exert more force on the small value filter taps to zero while hardly affecting coefficients with large value. In this way S-RZA-LMS can achieve faster convergence rate and lower steady state error than RZA-LMS.

Furthermore, we propose a modified Discrete Segment RZA-LMS (DS-RZA-LMS), which uses an inverse slope segment for filter taps with small magnitude in the piece-wise function as follows:

$$f_3(w_i) = \begin{cases} 2500|w_i| & \text{if } w_i < 0.05 \\ 0.01 & \text{otherwise} \end{cases}, \tag{6.13}$$

for $i = 1, 2, ..., L$. The positive slope permits larger slope value and therefore outperforms S-RZA-LMS when the signal to be recovered is sparse. The simulation results of different algorithms are shown in the next section. Note our proposed algorithm also eliminates the additional logic caused by the division term in RZA-LMS algorithm. Instead of L multiplications and L divisions to compute $\dfrac{1}{1+\varepsilon|\mathbf{w}(n)|}$, the segment term in our algorithms only needs less than L multiplications and comparisons.

6.4 Simulations & Results

In this section the performance of our proposed algorithms are presented for both white input signal and correlated input signal. For comparison purposes, the performance is evaluated using the Mean Square Deviation (MSD) defined as follows:

$$\text{MSD}(n) = 10\log_{10} \text{E}\left\{ \ \left\|\mathbf{w}_{\text{opt}} - \mathbf{w}(n)\right\|_2^2 \right\} \text{ (dB)}. \tag{6.14}$$

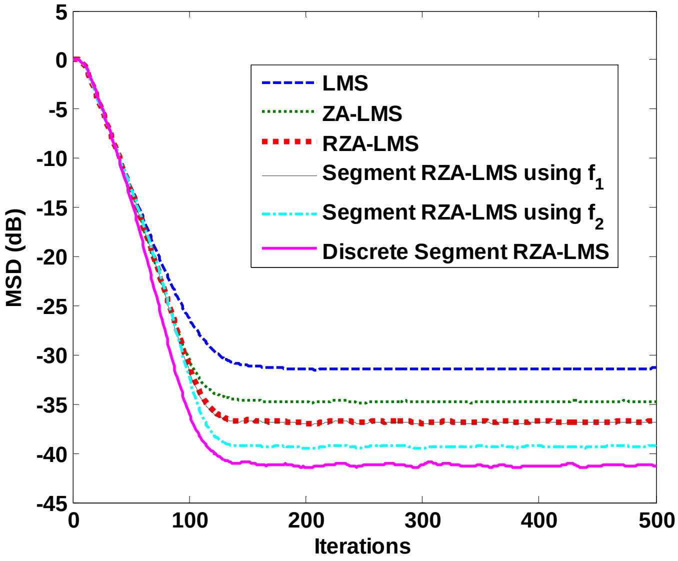


Fig. 6.2: Convergence performance and steady state behavior for different filters, with white input.

The simulation setup of [48] is adopted for comparisons. The channel response consists of 16 coefficients $L = 16$ and initially the 5th tap is set to 1 and others to zero. This leads to a channel with sparseness equal to 1/16. An additive Gaussian white noise is added such that the SNR is 30 dB. The performance of the six filters, LMS, ZA-LMS, RZA-LMS, our proposed S-RZA-LMS using $f_1(\mathbf{w})$, S-RZA-LMS using $f_2(\mathbf{w})$, and DS-RZA-LMS using $f_3(\mathbf{w})$, are compared in Fig. 6.2 for white input signal. The simulation uses the same step size $\mu = 0.05$ for all the algorithms. Other parameters are set to be $\rho = 5\times10^{-4}$ and $\varepsilon = 10$. It can be seen from Fig. 6.2 that the convergence of our proposed S-RZA-LMS using $f_1(\mathbf{w})$ strictly follows that of the RZA-LMS. With better choice of the segment function $f_2(\mathbf{w})$, S-RZA-LMS can achieve lower error floor after convergence. The DS-RZA-LMS further speeds up the convergence rate, and behaves

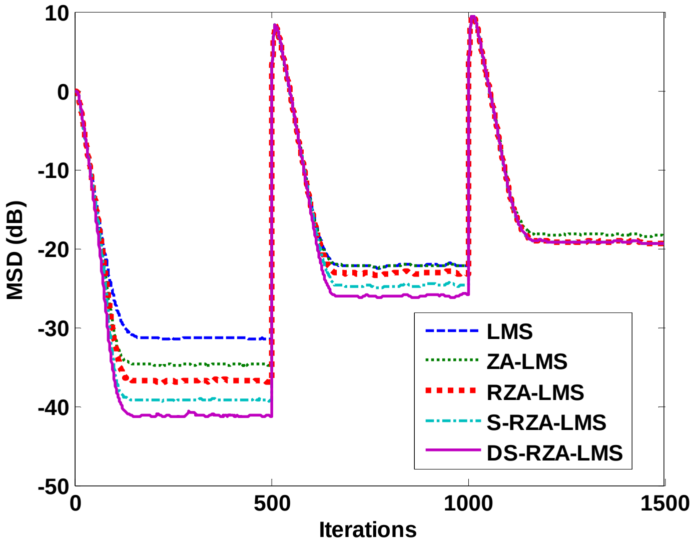


Fig. 6.3: Convergence performance, steady state behavior, and tracking ability for different filters and different sparse channels, with white input.

the best among the six filters for the steady state error through the whole convergence process.

The second simulation in Fig. 6.3 is set up for the time varying tracking ability of different algorithms and also for the case when the channel is less sparse. For clarity the S-ZA-LMS using $f_1(\mathbf{w})$ is not included since it strictly follows the convergence of RZA-LMS. The initial set up is the same as the first simulation. After 500 iterations, all the odd taps are set to be of value 1 with the even taps remaining zero, which makes the sparseness of the channel change to 8/16. Then after 1000 iterations, all the even taps are set to -1 with the odd taps remaining at 1, which is a completely non-sparse channel. The parameters used for simulation remain the same and the input is still white. As seen in Fig. 6.3, DS-RZA-LMS maintains the best performance among the five filters. When the number of non-zero taps increases to 8 out of 16, S-RZA-LMS using $f_2(\mathbf{w})$ and DS-

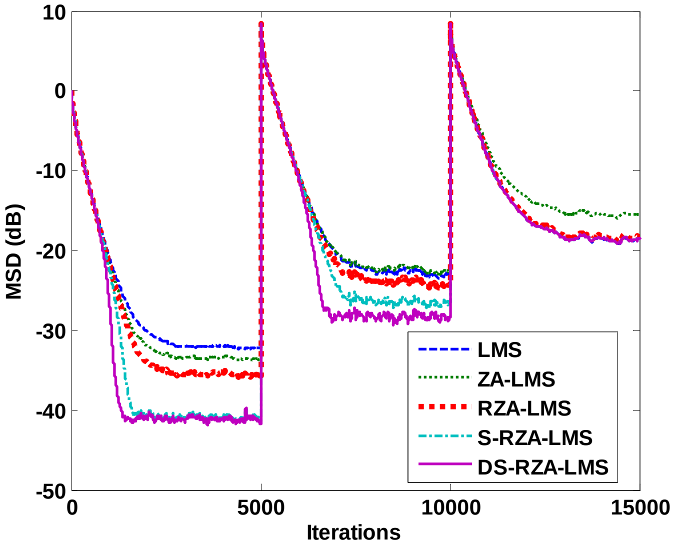


Fig. 6.4: Convergence performance, steady state behavior, and tracking ability for different filters and different sparse channels, with correlated input.

RZA-LMS still have 3-5 dB gain for the MSD performance compared with LMS, while ZA-LMS and RZA-LMS obtain little improvement. For a completely non-sparse channel, ZA-LMS works even worse than the conventional LMS algorithm. RZA-LMS and our proposed algorithms perform comparably to the conventional LMS.

The third simulation is set up for correlated input. The correlated input signal $\mathbf{x}(n)$ is generated by passing a white Gaussian random signal through a first-order filter with system response $H(z)=\dfrac{1}{1-0.8z^{-1}}$. The step size for all filters is set to $\mu=0.015$ and $\rho=3\times10^{-5}$, $\varepsilon=10$. The channel setup is the same as the second simulation, with switching times adjusted to 5000th iteration and 10000th iteration, respectively. The MSD performance is shown in Fig. 6.4. It shows similar trend for the correlated input compared with that for the white input. Note that for the case of correlated input, the performance improvements gained by our proposed S-RZA-LMS using $f_2(\mathbf{w})$ and DS-RZA-LMS are even larger than that for white input signal, which is about 5-7 dB compared with RZA-LMS. This even holds for the less sparse channel, when the sparseness changes at the 5000th iteration. The convergence rate is also increased, especially for DS-RZA-LMS. For non-sparse channel after 10000th iteration, the RZA-LMS, S-RZA-LMS and DS-RZA-LMS behave the same as conventional LMS, while the performance ZA-LMS degrades to unacceptable level. It can be observed that our proposed S-RZA-LMS and DS-RZA-LMS outperform the recently proposed ZA-LMS and RZA-LMS, for both white input signal and correlated input signal.

6.5 Conclusions

In this chapter, two new types of Segment RZA-LMS and Discrete Segment RZA-LMS algorithms have been proposed based on the iterative representation of the RZA-LMS algorithm. With more freedom when choosing the individual zero attractors for different filter taps, our proposed algorithms behave better than RZA-LMS with faster convergence and lower steady state error. The introduction of a linear segment function in zero attractors also eliminates the additional division operations in RZA-LMS. Simulation results demonstrate that the proposed S-RZA-LMS and DS-RZA-LMS are superior to previous ZA-LMS and RZA-LMS for both a highly sparse channel and a less sparse channel, and robust performance is obtained for a non-sparse channel. The improvement gained by our algorithms is even better when the input is a correlated signal.

Chapter 7

Digital Pre-Distortion for Power Amplifiers with Dynamic Nonlinearities Using Power-Indexed Look-Up Tables

7.1 Introduction

A conventional radio Power Amplifier (PA) operating with non-constant envelope wideband signals such as those used in 3G or other air interface standards, can cause nonlinear distortion in the output signals and interference on adjacent radio channel frequencies [23]. Operating the PA with a large back-off power level to control the nonlinear distortion would significantly reduce PA efficiency and therefore result in a higher operating cost. To address this problem, various linearization techniques, such as RF feedback, RF feed-forward, pre-distortion and post-distortion, have been developed in [23-28]. Among these different solutions, the baseband digital pre-distortion (DPD) [26-28] technique has proven to be the most efficient and cost effective solution. Although recent improvements have included PA memory effects [29] and have achieved excellent static performance [28, 51-57], current DPD approaches have relatively poor performance for PAs with dynamic nonlinearities. Advanced PA topologies, like asymmetric or N-way Doherty, and slow or fast drain modulation, exhibit very different nonlinear characteristics for signals having different power levels [58]. For applications with multi-carrier communications, large power variations can be seen over a short time interval. Taking an LTE signal as an example, the carrier power may vary >6dB on adjacent 1ms time slots. It is quite difficult for the DPD to re-adapt to a new set of coefficients within such a short time using a reasonable amount of adaptation hardware. Experiments show a significant degradation of DPD performance in this case, and there have been no robust solutions discussed in the literature that address the problem.

In this chapter a novel system design proposed in [60] is described, which improves the dynamic performance of DPD by using power-indexed LUTs, interpolation methods, a LUT update strategy, and an adaptive gain adjustment mechanism. Experimental results obtained using a 2-GHz PA with a 2-carrier WCDMA signal demonstrate that the proposed design can achieve very robust and stable pre-distortion performance for PA with dynamic nonlinearities.

7.2 Proposed Power-Indexed LUT Technique of DPD

7.2.1 System Overview

The generic DPD architecture is illustrated in Fig. 1.4, where the inverse nonlinear characteristics of the PA are estimated using either indirect learning (a) or direct learning (b), which are then used to pre-distort the PA input to compensate for the non-ideal behavior of the PA. The Volterra series and a variety of simplified models based on the Volterra series have been used for PA nonlinear modeling. The adapted pre-distorter coefficients, also referred to as the DPD solutions, are regularly updated through adaptive algorithms.

Fig. 7.1 shows the block diagram of our proposed system to improve the dynamic performance of DPD when there are abrupt signal power variations over a short time. Multiple DPD solutions associated with their power indices are stored and updated regularly to the LUT after the coefficients have been well adapted in the training process. A running average power meter is used to monitor the signal power level change, and if the detected power change is beyond the preset range, a different DPD solution for the current power level will be reloaded into the pre-distorter in the DPD data path. This new DPD solution can be one of the previously stored solutions in the power-indexed LUT, or it can be an interpolated solution when the current power level lies between two power indices in the LUT. Based on measurements and analyses of different nonlinear characteristics of a PA operating at various power levels, a set of nonlinear interpolation methods and a LUT update strategy are proposed in [60] as described in the next section.

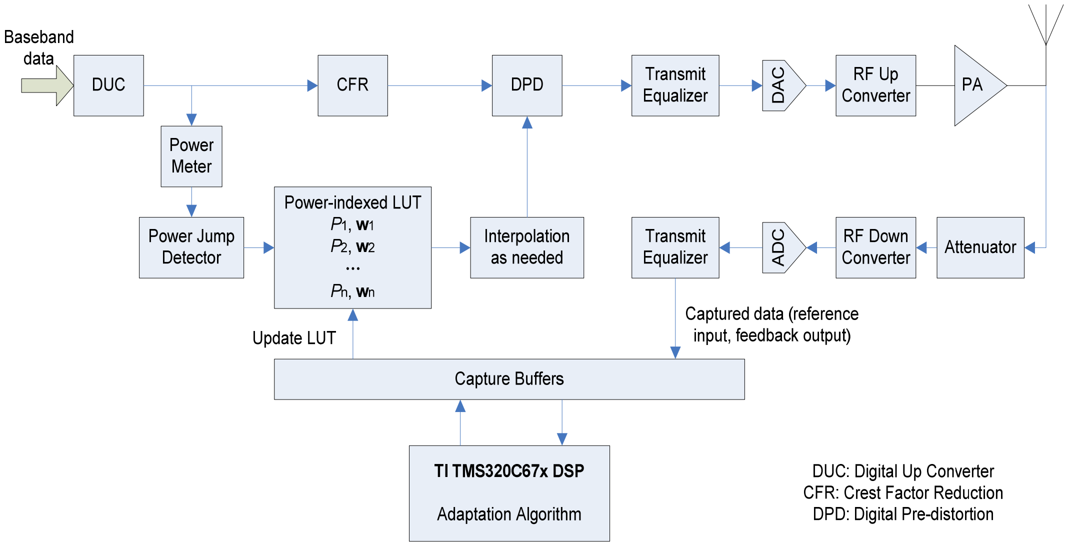


Fig. 7.1: System diagram of DPD with the proposed power-indexed LUT.

With its efficient implementation in hardware, the proposed design permits more frequent updates of pre-distorter coefficients from the LUT, and thus results in a more stable linearization performance of the DPD especially for rapidly changing real-time wideband signals. In addition, the system can be flexibly set by different parameters to accommodate various types of PAs or signals with different types of air interfaces, by adjusting the threshold of power hopping, the active LUT size, the interpolation methods, or the update operations.

7.2.2 Interpolation Methods

The power-indexed LUT stores multiple DPD solutions for a PA operating at various power levels. If the current operating power level is close to a previously stored power index, its associated pre-distortion coefficients in the LUT can be directly used as the DPD solution. Otherwise, interpolation is used to generate an intermediate DPD solution based on the available DPD solutions stored in the LUT.

Interpolation functions that can be applied to the parameters of nonlinear models like Volterra series are not as straightforward as for linear models. The interpolated DPD solution needs to be accurate enough to perform the linearization for signals with a new power level, while still having an efficient implementation as well as providing real-time control. From measurements, we have observed that the DPD solution for a PA operating at a higher power level produces more robust linearization results than using the DPD solution for a lower power level. Therefore, we propose a set of nonlinear interpolation schemes for DPDs as part of our solution. When the current power level P lies between two indices of the power-indexed LUT, $P \in (P_{low}, P_{high})$, and where $\mathbf{w}_\mathrm{l}$ and $\mathbf{w}_\mathrm{h}$ are the corresponding DPD solutions, the desired DPD solution $\mathbf{w}$ for power level P is computed as:

$$\mathbf{w} = f(\alpha)\cdot\mathbf{w}_h + (1 - f(\alpha))\cdot\mathbf{w}_l\,, \tag{7.1}$$

$$f(\alpha) = 1 - (1-\alpha)^n,\ n = 1, 2, 3, \ldots, \tag{7.2}$$

$$\alpha = \frac{10\log_{10}(P/P_{low})}{10\log_{10}(P_{high}/P_{low})},\ P \in (P_{low}, P_{high})\,. \tag{7.3}$$

Note that α has a range of (0, 1), and its value is determined by the region in (P_{low}, P_{high}) where P lies (in dB), which defines the nonlinear nature of the proposed interpolation methods. The value of $f(\alpha)$ reflects the dependency of $\mathbf{w}$ on the DPD solution $\mathbf{w}_\mathrm{h}$ for the higher power index P_{high}, and n is an adjustable parameter controlling the degree of nonlinearity of the interpolation for different operating power regions of the PA or for different PAs. Experimental results in Section 7.3 demonstrate great performance using this interpolation scheme.

7.2.3 Update Strategy

Updating the power-indexed LUT with a newly adapted DPD solution is one of the primary operations in this configuration. Once a new set of coefficients have converged for the current power level, both the power index and the DPD solution are updated to the LUT. Note that the power indices or the distribution of the power indices is not preset. Rather, they are updated based on the real-time signal power level to ensure the accuracy of the DPD solutions provided either by switching between or by interpolating between DPD solutions stored for prior power levels.

The update scheme can be summarized as follows:

1) If the power index P_{new} for the newly adapted DPD solution $\mathbf{w}_{\text{new}}$ is available in the LUT, overwrite its solution for P_{new}.
2) Otherwise, if the LUT is currently not full, update both P_{new} and $\mathbf{w}_{\text{new}}$ to the LUT.
3) If P_{new} is not available in the LUT and the LUT is full, first sort the gaps between adjacent power level indices in the dB domain (taking P_{new} in account) and then find the one which is the lower bound of the smallest gap, denoted as P_{low}. If P_{new} is not P_{low}, replace P_{low} and its solution with P_{new} and $\mathbf{w}_{\text{new}}$ in the LUT.

Flexibility can be achieved by using separate power-indexed LUTs for different power region, with a larger LUT size when the signal power is high. Moreover, if the PA is not stable over time, an expiration time can be associated with each DPD solution in the LUT to provide robust performance.

7.2.4 Adaptive Gain Compensation

Due to the time interval between the training process and the performance of predistortion using the trained solution, the linearization may be suboptimal when the input

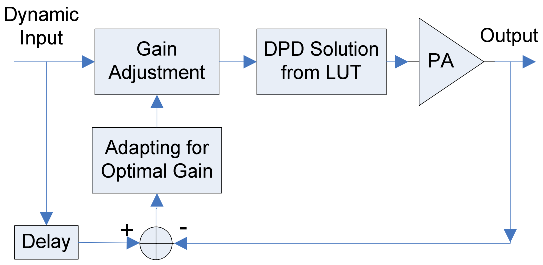
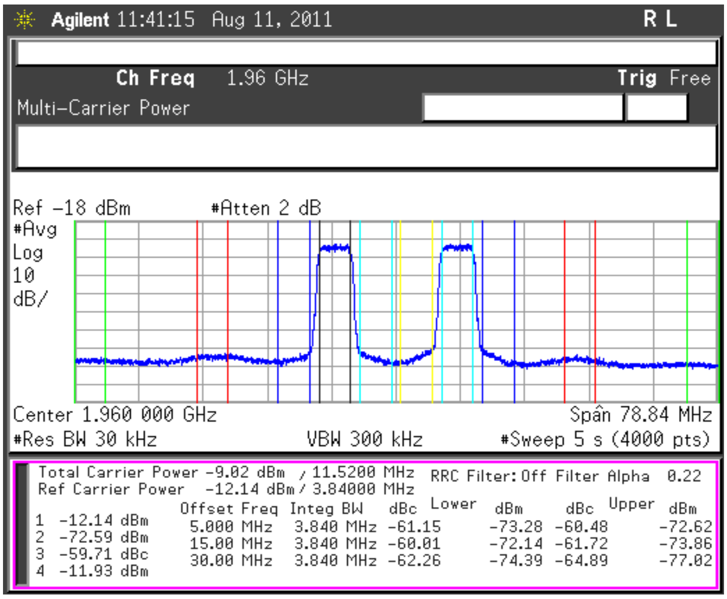


Fig. 7.2: Structure of adaptive gain compensation.

pattern is experiencing a rapid change. To enhance its performance, a fast adaptive gain compensation before the DPD can be used, as shown in Fig. 7.2. Based on the real-time error feedback, simple sample-by-sample adaptive algorithms such as Least Mean Square (LMS) can be used to adapt the gain to be applied ahead of the DPD, which leads to better DPD performance in such situations.

7.3 Experimental Results

We have tested the performance of our power-indexed LUT design using the current generation TI GC5330 evaluation platform [57] and the performance is demonstrated in [60]. The device under test is two-way symmetric LDMOS Doherty with 51dBm $P_{1\mathrm{dB}}$ and 47% efficiency at 1.96 GHz, driven by a 2-carrier 20-MHz WCDMA input signal with 6.5 dB PAR after Crest Factor Reduction. The DPD sampling rate is 245.76 MSPS, and a direct learning architecture with a block-based least squares adaptation algorithm has been used to adapt for new DPD solutions based on a reduced Volterra nonlinear DPD model that was described in [59]. Experiments were run at a PA output power range from 30 dBm to 46 dBm, and an Agilent spectrum analyzer was used to monitor

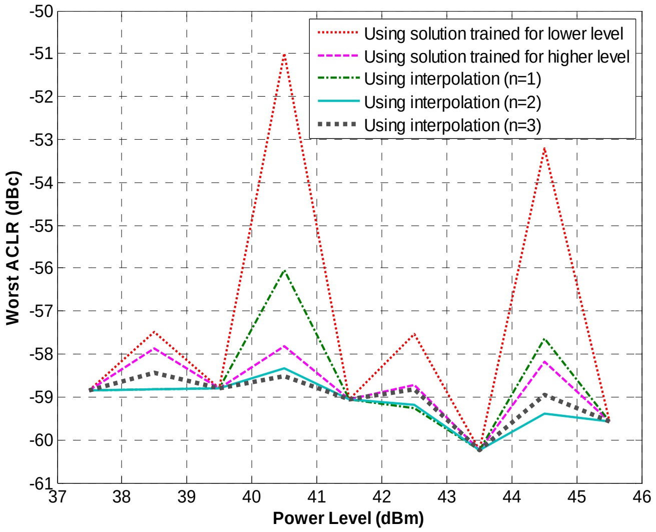


```
Agilent 11:41:15  Aug 11, 2011                         R L
            Ch Freq     1.96 GHz                         Trig Free
Multi-Carrier Power

Ref -18 dBm         #Atten 2 dB
#Avg
Log
10
dB/

Center 1.960 000 GHz                                  Span 78.84 MHz
#Res BW 30 kHz              VBW 300 kHz          #Sweep 5 s (4000 pts)

  Total Carrier Power -9.02 dBm  / 11.5200 MHz  RRC Filter: Off Filter Alpha  0.22
  Ref Carrier Power   -12.14 dBm / 3.84000 MHz
                     Offset Freq  Integ BW     dBc  Lower  dBm      dBc  Upper  dBm
 1  -12.14 dBm        5.000 MHz   3.840 MHz  -61.15       -73.28   -60.48      -72.62
 2  -72.59 dBm        15.00 MHz   3.840 MHz  -60.01       -72.14   -61.72      -73.86
 3  -59.71 dBc        30.00 MHz   3.840 MHz  -62.26       -74.39   -64.89      -77.02
 4  -11.93 dBm
```

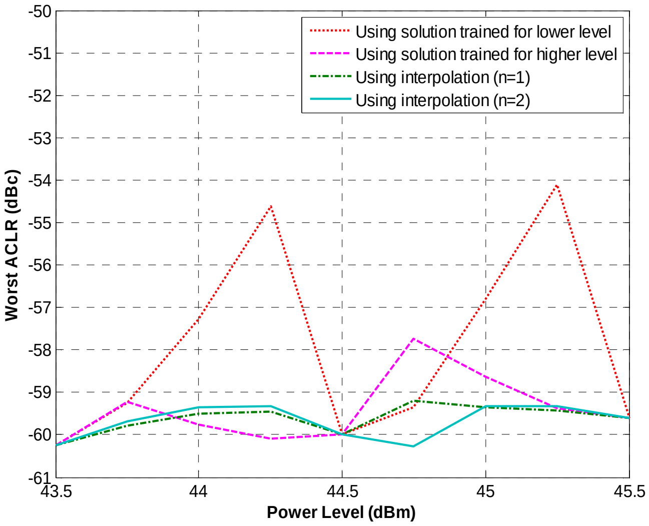
Fig. 7.3: PA output spectrum with power-indexed LUT of DPD.

and record the PA's output nonlinearities.

For the current PA under test, significant degradation of DPD performance of up to 8 dB Adjacent Channel Leakage Ratio (ACLR) increase can be observed during the test when there is abrupt power level change of 1 dB within the training time interval. As shown in the captured shot of the PA output spectrum in Fig. 7.3, with our proposed power-indexed LUT, the DPD is able to maintain a consistently good linearization performance for dynamic input signals. The worst-case ACLR of PA output spectrum using interpolated DPD solutions from the LUT are shown in Fig. 7.4 (a) and (b) based on experimental results. As can be seen, when the gap between two adjacent power indexes is large, interpolation with n>1 outperforms that with n=1, and when it is smaller than 1 dB, interpolation with n=1 or direct switching to the DPD solution trained for the higher power level can also work reasonably well. The methodology also



(a)



(b)

Fig. 7.4 (a) Performance of LUT interpolations: available DPD solutions at power level of 37.5, 39.5 41.5 43.5 and 45.5 dBm. Interpolation used for DPD solutions at power level of 38.5, 40.5, 42.5, 44.5 dBm. (b) Performance of LUT interpolations: available DPD solutions at power level of 43.5, 44.5, and 45.5 dBm. Interpolation used for DPD solutions at power level of 43.75, 44, 44.25, 44.75, 45, and 45.25 dBm.

offers the option to have multiple LUTs separately for low and high power regions of DPD, having different LUT sizes and different values of *n*.

7.4 Conclusion

In this chapter, we have described a novel solution to improve the dynamic performance of PA linearization with wideband input signals using DPD with power-indexed LUTs. Measurement results have shown the proposed design can achieve robust and stable linearization performance for dynamic applications. With a flexible system setup, it can support a variety of PA topologies and modulation standards, and significantly improve PA efficiency and save substantial costs for next generation base stations.

Chapter 8

Conclusions & Future Work

Sparsity is a feature present in many practical signals and systems. Efficient signal processing techniques which can exploit the sparseness property are very attractive from an implementation perspective. In this dissertation, we have proposed adaptive algorithms and system designs for sparse signal estimation in three widely used digital signal processing applications, specifically echo cancellation, compressive sensing, and power amplifier pre-distortion.

For network echo cancellation, inspired by the goal of reducing computational complexity, we have developed the adaptive tap partial update proportionate adaptive filter with maximal use of sparsity and great Mean Square Error (MSE). Simulation results have shown that, by adaptively choosing the partially updated taps based on the sparseness of the estimated coefficients, the same convergence speed and error performance can be achieved with more than 50% reduced computational complexity compared with conventional proportionate adaptive filters. Additionally, the μ-law proportionate technique is incorporated to a recently proposed memorized improved proportionate APA, giving a further improvement in the convergence performance while maintaining low computational complexity. In future work, hardware implementation of the adaptive tap partial update algorithms can be considered. A periodic update of the sparseness measurement can be adopted with little performance degradation. Moreover, effort can be devoted to the simplification of real-time sparseness calculation to achieve further reduction of the computational cost.

For acoustic echo cancellation where the enclosure environment is challenging, our main focus is to improve the convergence rate of echo cancellers. With the use of a time-averaging gradient vector as the step size gain distribution vector, we proposed a new class of gradient-controlled proportionate algorithms based on the optimum individual step size derived from a theoretical approach. It can be incorporated into both the time domain Affine Projection algorithms and the frequency domain Multi-Delay filter. Simulations demonstrate significant improvement in the convergence speed as compared to the traditional proportionate algorithms based on IPNLMS, especially for highly correlated signal inputs. This new proportionate technique can be extended to many other proportionate algorithms in future work with the expectation of faster convergence speed and more robust performance for colored inputs.

For the sparse signal recovery problem in compressive sensing where the signal has only a small number of non-zero values, the low-cost segment reweighted zero attracting algorithms are developed. With the piece-wise linear functions assigning different zero attractors during the convergence process, the sparse segment zero-attracting LMS outperforms the recently proposed ZA-LMS and RZA-LMS in both error performance and tracking ability. At the same time the computational cost is significantly reduced due to the elimination of additional logic caused by the division term in RZA-LMS. Future work can include a study of various applicable sparse signals to improve the system flexibility, as well as the development of an efficient hardware implementation.

Finally we have also discussed the current challenges of the nonlinear sparse system estimation problem in DPD. A novel power-indexed LUT-based DPD solution is proposed to compensate for the dynamic nonlinearities introduced by the PA operating with wideband signals. The complete solution, including power-indexed LUTs, interpolation methods, a LUT update strategy, and an adaptive gain adjustment mechanism, have been tested in the lab with a 2 GHz PA and very robust and stable results have been demonstrated. In future work, elements of this solution can be integrated to wideband transmit-receive digital signal processors, which can have significant practical benefits.

Bibliography

[1] S. Haykin, *Adaptive Filter Theory, Fourth Edition*, Prentice Hall, 2002.

[2] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Willey & Sons Inc., 2003.

[3] B. Farhang-Boroujeny, *Adaptive Filters*, John Willey & Sons Inc., 1998.

[4] S.L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, Kluwer Academic Publishers, 2001.

[5] J. Benesty, *Advances in Network and Acoustic Echo Cancellation*, Springer, 2001.

[6] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Select. Areas Commun.*, pp. 128-137, Feb. 1987.

[7] A. F. Molisch, "Ultrawideband propagation channels-Theory, measurement, and modeling," *IEEE Trans. Veh. Technol*., pp. 1528-1545, Sep. 2005.

[8] S. F. Cotter and B. D. Ral, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun*., pp. 374-377, Mar. 2002.

[9] C. Carbonelli, S. Vedantam, and U. Mitra, "Sparse channel estimation with zero tap detection," *IEEE Trans. Wireless Commun*., pp. 1743-1753, May 2007.

[10] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Trans. Signal Processing*, pp. 1708-1721, Mar. 2010.

[11] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of Royal Statistical Society*, series B, vol. 58, pp. 267-288, 1996.

[12] D. Donoho, "Compressive sensing," *IEEE Trans. Information Theory*, vol. 52, pp. 1289-1306, Apr. 2006.

[13] W. U. Bajwa, J. Haupt, G. Raz and R. Nowak, “Compressed channel sensing,” *Proc. 42nd Annu. Conf. Information Sciences and Systems*, pp. 5-10, 2008.

[14] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” *IEEE Trans. Speed and Audio Processing*, vol. 8, no. 5, pp. 508-518, Sep. 2000.

[15] J. Benesty and S. L. Gay, “An improved PNLMS algorithm,” *IEEE International Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1881-1884, 2002.

[16] H. Deng and M. Doroslovacki, “Proportionate adaptive algorithms for network echo cancellation,” *IEEE Trans. Signal Processing*, vol. 54, issue 5, pp. 1794-1803, May 2006.

[17] T. Gansler, J. Benesty, S. L. Gay, and M. Sondhi, “A robust proportionate affine projection algorithm for network echo cancellation,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 793-796, 2000.

[18] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, “A generalized proportionate variable step-size algorithm for fast changing acoustic environments,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 161-164, 2004.

[19] S. C. Douglas, “Adaptive filter employing partial updates,” *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209-216, Mar. 1997.

[20] T. Aboulnasr and K. Mayyas, “Complexity reduction of the NLMS algorithm via selective coefficient update,” *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1421-1424, May 1999.

[21] P. A. Naylor and W. Sheriliker, “A short-sort M-Max NLMS partial-update adaptive filter with applications to echo cancellation,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 373-376, 2003.

[22] H. Deng and M. Doroslovacki, “New sparse adaptive algorithms using partial update,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 845-848, 2004.

[23] C. Cripps, *RF Power Amplifiers for Wireless Communications*. Norwood, MA: Artech House, 1999.

[24] P. B. Kenington, *High-Linearity RF Amplifier Design*. Boston, MA: Artech House, 2000.

[25] S. I. Mann, M. A. Beach, and K. A. Morris, “Digital baseband Cartesian loop transmitter,” *IET Electronics Letters*, vol. 37, pp. 1360-1361, Oct. 2001.

[26] J. K. Cavers, “Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements,” *IEEE Trans. Veh. Technol.*, vol. 39, pp. 374-382, Nov. 1990.

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

[28] D. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, “A generalized memory polynomial model for digital predistortion of RF power amplifiers,” *IEEE Trans. Sig. Proc.*, vol. 54, no. 10, pp. 3852-3860, 2006.

[29] W. Bösch and G. Gatti, “Measurement and simulation of memory effects in predistortion linearizers,” *IEEE Trans. Microw. Theory Tec*h., vol. 37, pp. 1885-1890, Dec. 1989.

[30] O. Tanrikulu and K. Dogancay, “Selective-partial-update proportionate normalized least-mean-squares algorithm for network echo cancellation,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol.2, pp.1889-1892, 2002.

[31] A. W. H. Khong and P. A. Nalor, “Efficient use of sparse adaptive filters,” *Asilomar Conf. on Signals, Systems and Computers*, pp. 1375-1379, Nov. 2006.

[32] J. Cui, P. A. Naylor, and D. T. Brown, “An improved IPNLMS algorithm for echo cancellation in packet-switched networks,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, pp. iv-141-iv-144, May 2004.

[33] A. W. H. Khong and P. A. Naylor, “Selective-Tap Adaptive Filtering With Performance Analysis for Identification of Time-Varying Systems,” *IEEE Trans Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1681-1695, July 2007.

[34] J. Yang, X. Zhu, G. E. Sobelman, and K.K. Parhi, “Sparseness-controlled adaptive tap algorithms for partial update adaptive filters,” *7th Int. Conf. on Information, Communications and Signal Processing*, Dec 2009.

[35] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457-1469, Nov. 2004.

[36] E. R. Ferrara, “Fast implementations of LMS adaptive filters,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 474-475, 1980.

[37] J. S. Soo and K. K. Pang, “Multi-delay block frequency domain adaptive filter,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 373-376, Feb. 1990.

[38] A. W. H. Khong, J. Benesty, and P. A. Naylor, “A low delay and fast converging improved proportionate algorithm for sparse system identification,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, 2007.

[39] A.W.H. Khong, X. Lin, M. Doroslovacki and P.A. Naylor, “Frequency domain selective tap adaptive algorithms for sparse system identification,” *IEEE International Conf. on Acoustics, Speech, and Signal Processing*, pp.229-232, 2008.

[40] J. Yang and G. E. Sobelman, “Frequency domain adaptive tap partial update adaptive algorithm for network echo cancellation,” *IEEE International Conf. on Acoustics, Speech, and Signal Processing*, pp. 4046-4049, 2010.

[41] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electron. Commun. Jpn.*, vol. 67-A, no. 5, pp. 19-27, May 1984.

[42] C. Paleologu, S. Ciochina, and J. Benesty, “An efficient proportionate affine projection algorithm for echo cancellation,” *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 165-168, Feb 2010.

[43] J. Yang and G. E. Sobelman, “Efficient μ-law improved proportionate affine projection algorithm for echo cancellation,” *IET Electronics Letters*, vol. 47, issue 2, pp. 73-74, 2011.

[44] B. N. M. Laska, R. A. Goubran, and M. Bolic, “Improved proportionate subband NLMS for acoustic echo cancellation in changing environments,” *IEEE Trans. Signal Processing*, vol. 15, pp.337-340, 2008.

[45] P. Loganathan, A. W. H. Khong, and P. A. Naylor, “A class of sparseness-controlled algorithms for echo cancellation,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1591-1601, Nov. 2009.

[46] F. C. de Souza, O. J. Tobias, R. Seara, and D. R. Morgan, “A PNLMS algorithm with individual activation factors,” *IEEE Trans. Signal Process*., vol. 58, no. 4, pp. 2036-2047, Apr. 2010.

[47] J. Yang and G. E. Sobelman, “A class of gradient-controlled proportionate algorithms for acoustic echo cancellation,” currently under review.

[48] Y. Chen, Y. Gu, and A. O. Hero, “Sparse LMS for system identification,” *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing*, pp. 3125-3128, Taipei, Taiwan, Apr. 2009.

[49] K. Shi and X. Ma, “Transform domain LMS algorithms for sparse system identification,” *IEEE International Conf. on Acoustics, Speech and Signal Processing*, pp. 3714-3717, 2010.

[50] J. Yang and G. E. Sobelman, “Sparse LMS with segment zero attractors for adaptive estimation of sparse signals,” *IEEE Asia Pacific Conf. on Circuits and Systems*, pp. 422-425, Dec. 2010.

[51] J. Kim and K. Konstantinou, “Digital predistortion of wideband signals based on power amplifier model with memory,” *IET Electronics Letters*, vol. 37, pp. 1417-1418, Nov. 2001.

[52] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, “A robust digital baseband predistorter constructed using memory polynomials,” *IEEE Trans. Commun*., vol. 52, no. 1, pp. 159-165, Jan. 2004.

[53] P. Celka, N. J. Bershad, and J. M. Vesin, “Stochastic gradient identification of polynomial Wiener systems: analysis and application,” *IEEE Trans. Signal Processing*, vol. 49, no.2, pp. 301-313, Feb. 2001.

[54] A. E. Nordsjö and L. H. Zetterberg, “Identification of certain time-varing nonlinear Wiener and Hammerstein systems,” *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 577-592, Mar. 2001.

[55] N. J. Bershad, P. Celka, and S. McLaughlin, “Analysis of stochastic gradient identification of Wiener-Hammerstein systems for nonlinearities with Hermite polynomial expansions,” *IEEE Trans. Signal Processing*, vol. 49, no. 5, pp. 1060-1072, May 2001.

[56] L. Ding, Z. Ma, D. R. Morgan, and M. Zierdt, “A least-squares/Newton method for digital pre-distortion of wideband signals,” *IEEE Trans. Commun*., vol. 54, no. 5, pp. 833-840, May 2006.

[57] H. Gandhi and W. Abbott, “A digital signal processing solution for PA linearization and RF impairment correction for multi-standard wireless transceiver systems,” *European Microwave Conference (EuMC)*, pp. 719-722, Sep. 2010.

[58] B. Kim, I. Kim, and J. Kim, “Advanced Doherty architecture,” *IEEE Microw. Mag.*, vol. 11, no. 5, pp. 72-86, Aug. 2010.

[59] H. Gandhi, “A flexible Volterra-based adaptive digital pre-distortion solution for wideband RF power amplifier linearization,” *Microwaves and RF Magazine*, Jul. & Aug. 2008.

[60] J. Yang, Z. Yang, H. Gandhi, and G. E. Sobelman, “Digital pre-distortion for power amplifiers with dynamic nonlinearities using power-indexed look-up tables,” currently under review.