Sparsity-aware Adaptive Filtering Algorithms and Application to System Identification

Ran Meng

Department of Electronics

University of York

A thesis submitted for the degree of

Master by Research (MSc)

December 2011

Abstract

In this thesis, low-complexity adaptive filtering algorithms that exploit the sparsity of signals and systems are derived and investigated. Specifically, sparsity-aware normalized least-mean square and affine projection algorithms are developed based on the l_1-norm incorporated to their cost function, which we term zero-attracting NLMS (ZA-NLMS) and zero-attracting APA (ZA-APA). These algorithms are analyzed and applied to the identification of sparse systems. To further improve the filtering performance, the reweighted ZA-NLMS (RZA-NLMS) and reweighted ZA-APA (RZA-APA) are also proposed, which employs reweighted step sizes of the zero attractor for different taps, inducing the attractor to selectively promote zero taps rather than uniformly promote zeros on all the taps. We also develop zero-forcing techniques to further improve their performance when the system has a significantly degree of sparsity, i.e., a very small number of non-zero coefficients. Simulation results show that the proposed algorithms outperform the standard NLMS and APA algorithms in both convergence rate and steady-state performance for sparse systems.

Contents

List of Figures

List of Tables

Acknowledgements

Firstly, I would like to express my most sincere gratitude to my supervisor, Dr. Rodrigo C. de Lamare, for his valuable help and supervision with my research, without which much of this work would not have been possible.

Further thanks to my second supervisor Dr. Vitor H. Nascimento, for his help and support throughout this year of my research.

I also want to say thanks to all the members of the Communications Research Group, for their help and support throughout this year of research.

Finally, my deep gratitude goes to my parents for their unconditional support, end-less love and encouragement.

Declaration

Some of the research presented in this thesis has resulted in some publications. These Publications are listed at the end of Chapter 1.

All work presented in this thesis as original is so, to the best knowledge of the author. References and acknowledgements to other researchers have been given as appropriate.

Nomenclature

APA Affine Projection Algorithm

AR Autoregressive

EMSE Excess Mean Square Error

FAP Fast Affine Projection

FIR Finite-duration Impulse Response

FRLS Fast Recursive Least Square

i.i.d. Independent and Identically Distributed

IIR Infinite-duration Impulse Response

LHS Left-Hand Side

LMS Least Mean Square

MSE Mean Square Error

NLMS Normalized Least Mean Square

PNLMS Proportionate Normalized Least Mean Square

RHS Right-Hand Side

RLS Recursive Least Square

RZA Reweighted Zero-Attracting

SNR Signal-to Noise Ratio

ZA Zero-Attracting

ZF Zero-Forcing

Chapter 1

Introduction

In this chapter, we give a general introduction to adaptive filters and their applications and discuss the motivation and contributions of our work.

1.1 Adaptive Filters

In the last thirty years, the field of digital signal processing has developed in a very fast way. The tremendous growth and development in the digital signal processing area has turned some of its specialized topics into whole fields themselves. One example of a digital signal processing system is the adaptive filter [1–3]. The objective of filtering is to process a signal in order to manipulate the information contained in it. A digital filter is one that processes signals represented in digital format [4]. When dealing with signals whose statistical properties are fixed, the designer can easily choose the most appropriate algorithm to process the signal. However, fixed algorithms cannot process a signal efficiently if its properties are unknown. The solution is to use an adaptive filter which can change its characteristics automatically [5, 6].

As previously discussed, the design of digital filters with fixed coefficients requires well defined prescribed specifications. However, there are situations where the specifications are not available, or are time-varying. An adaptive filter is required when either the fixed specifications are unknown or the specifications cannot be satisfied by time-invariant filters. The ability of an adaptive filter to

operate satisfactorily in an unknown environment and track time variations of the statistics makes the adaptive filter a powerful device for signal processing and control applications. There are many different applications in which adaptive techniques can be used. Some examples are system identification, echo cancelation, equalization of dispersive channels, adaptive beamforming and adaptive control. As the power of digital signal processors has increased, adaptive filters have become much more common and are now routinely used in devices such as mobile phones and other communication devices [2].

1.2 Motivation

In many scenarios, the impulse response of unknown systems can be assumed to be sparse, containing only a few large coefficients interspersed among many negligible ones. Using such sparse prior information can improve the filtering/estimation performance [7]. However, standard adaptive filters do not exploit such information. In the past years, many algorithms exploiting sparsity were based on applying a subset selection scheme during the filtering process, which was implemented via statistical detection of active taps or sequential partial updating [8–10]. Other variants assign proportional step sizes to different taps according to their magnitudes, such as the proportionate normalized least-mean square (PNLMS) [11–14] algorithm and its variants [15].

Motivated by recent progress in compressive sensing [16–22], several authors have considered using the l_1-norm penalty to exploit sparsity [8, 9, 23–26]. The basic idea is to introduce a penalty that favors sparsity in the cost function. In this thesis we propose an alternative approach to identifying sparse systems using affine projection algorithm (APA). A particular form of this APA is the normalized least-mean-square (NLMS) algorithm [5, 6].

1.3 Contribution

In this thesis, we firstly incorporate an l_1-norm penalty on the coefficients into the quadratic cost function of the standard NLMS. This results in a modified NLMS update with a zero attractor for all the taps, which we term ZA-NLMS. Then

we apply the same strategy to the APA, and obtain an updated ZA-APA which can exploit the sparsity in the impulse response of linear systems. To further improve the filtering performance, the reweighted ZA-NLMS (RZA-NLMS) and reweighted ZA-APA (RZA-APA) are also proposed, which employ reweighted step sizes of the zero attractor for different taps, inducing the attractor to selectively promote zero taps rather than uniformly promote zeros on all the taps. We demonstrate via simulations and analytically that the ZA-APA and RZA-APA achieve better steady-state performance than that of the standard APA for sparse models.

Secondly, to further improve the performance of ZA-NLMS and ZA-APA in a sparse systems with a significant degree of sparsity, i.e., a small number of non-zero coefficients, we introduce a new zero-forcing strategy, whose idea is to force the small tap-weight coefficients to zero. By using this strategy, we obtain two new algorithms called zero-forcing NLMS (ZF-NLMS) and zero-forcing APA (ZF-APA).

Simulation results illustrate that the proposed algorithms outperform standard NLMS and APA in both convergence rate and steady-state performance for sparse systems; and the reweighted ones outperform the ordinary zero-attracting algorithms. Furthermore, RZA-NLMS and RZA-APA show robustness when the number of non-zero taps increases, with little loss in performance with respect to the standard ones in non-sparse situations.

1.4 Thesis Outline

The structure of the thesis is as follows.

- In Chapter 2, a review of adaptive filtering is given, and some of its applications are introduced.
- In Chapter 3, the proposed sparsity-aware algorithms are detailed, including zero-attracting algorithms and zero-forcing algorithms.
- In Chapter 4, we analyze the proposed algorithms, including an estimation of their computational complexity, and we carry out the MSE analysis and

steady-state performance.

- In Chapter 5, conclusions and a discussion on possibilities for future work are presented.

1.5 List of Publications

Some of the research presented in this thesis has been published, or will be submitted to some publications at the time of submission of this thesis.

Journal Papers

1. R. Meng, R. C. de Lamare and V. H. Nascimento, "Sparsity-Aware Affine Projection Adaptive Algorithms for System Identification," (under preparation)

Conference Papers

1. R. Meng, R. C. de Lamare and V. H. Nascimento, "Sparsity-Aware Affine Projection Adaptive Algorithms for System Identification," Proc. Sensor Signal Processing for Defence Conference, London, UK, 2011.

Chapter 2

Review of Adaptive Filtering

In this chapter, a review of adaptive filtering is given. Firstly, we introduce the motivation for using adaptive filters. Then we discuss some popular applications of adaptive filters. Finally, some of the most commonly used adaptive algorithms are introduced.

2.1 Objective of Adaptive Filters

The basic objective of an adaptive filter is to set its parameters, in such a way that its output tries to minimize a meaningful objective function involving a reference signal. In many scenarios, the objective function F is a function of the input, the reference and the adaptive filter output signal. We can consider that an adaptive algorithm is composed of three basic items: definition of the minimization algorithm, definition of the objective function and definition of the error signal. The error signal is usually defined as the difference between the filter output and a desired response. The optimal filter parameters are found through minimization of a cost function of the error signal. A useful approach is based on minimizing the mean-square value of the error signal [1].

The basic objective of an adaptive filter is to set its parameters, in such a way that its output tries to minimize a meaningful objective function involving a reference signal.

The general setup of an adaptive filtering system is illustrated in Fig. 2.1.

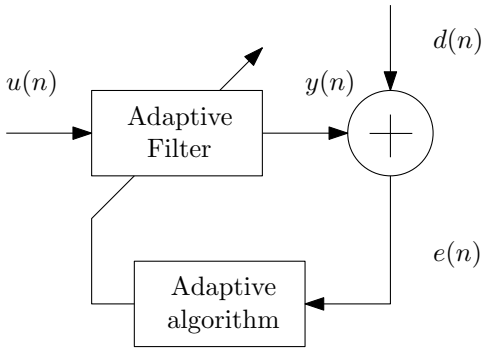

Figure 2.1: General adaptive filter configuration.

Here, $x(n)$ denotes the input signal, $y(n)$ is the adaptive filter output signal, and $d(n)$ denotes the desired signal at time instant n. The error signal $e(n)$ is calculated as $e(n) = d(n) - y(n)$. The error signal is then used to form a performance function that is iteratively minimized by the adaptation algorithm in order to determine the appropriate updating of the filter coefficients. The minimization of the objective function implies that the adaptive filter output signal matches the desired signal in some sense [5].

Basically, there are two major classes of adaptive digital filters, distinguished by the form of the impulse response, namely the finite-duration impulse response (FIR) filter and the infinite-duration impulse response (IIR) filter [4, 27]. FIR filters are implemented with nonrecursive structures, whereas IIR filters utilize recursive structures. Adaptive FIR filters are the most popular ones due to their stability.

The most widely used adaptive FIR filter structure is the transversal filter. The structure of the FIR filter is shown in Fig. 2.2. Here, we can define the complex-valued tap-weight vector with M coefficients as

$$\hat{\mathbf{w}}(n) = [w_0(n), w_1(n), ..., w_{M-1}(n)]^T, \tag{2.1}$$

where $[\,\cdot\,]^T$ is the transpose of a vector or a matrix. With the M-length tap-weight vector shown as above, the complex-valued input signal can be defined as

$$\mathbf{u}(n) = [u(n), u(n-1), ..., u(n-M+1)]^T. \tag{2.2}$$

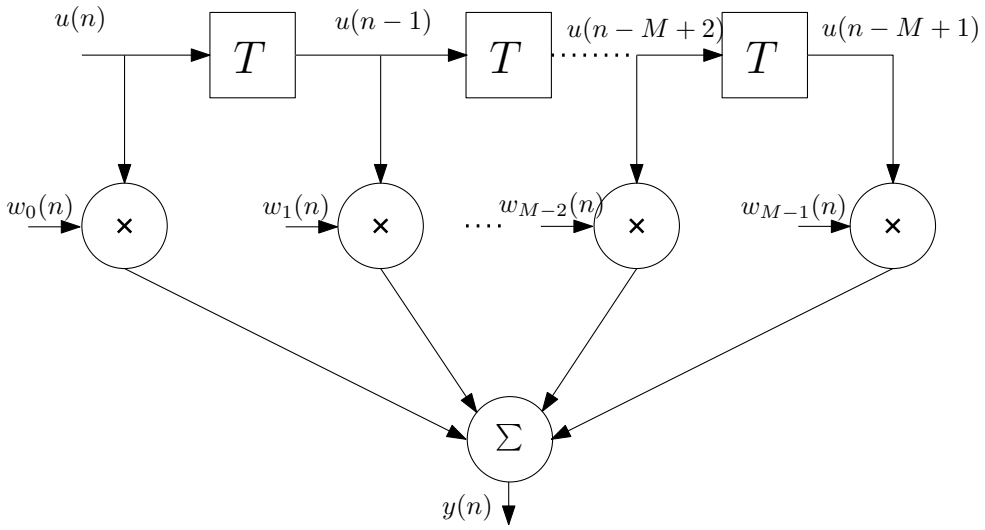

Figure 2.2: FIR Filter: Time-shifted structure of the input signal.

Then the output of the filter is

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n). \tag{2.3}$$

where $[\,\cdot\,]^H$ is the Hermitian transpose of a vector or a matrix.

2.2 Applications

The ability of an adaptive filter to operate satisfactorily in an unknown environment and track time variations of input statistics makes the adaptive filter a powerful device for signal processing and control applications. Although the applications of adaptive filters are quite different in nature, they have one common feature: an input vector and a desired response are used to compute an estimation error, which is in turn used to control the values of a set of adjustable coefficients [5]. However, the essential difference between the various applications arises in the way which the desired response is extracted. From this point of view, we may distinguish four basic classes of adaptive filtering applications, as shown in Fig. 2.3. The notation used in this figure is:

u = input applied to the adaptive filter,

Table 2.1: Applications of adaptive filter

Class of adaptive filtering	**Application**
I: Identification	System identification Layered earth modeling
II: Inverse modeling	Predictive deconvolution Adaptive equalization
III: Prediction	Linear predictive coding Signal detection
IV: Interference canceling	Echo Cancelation Adaptive beamforming

$$y = \text{output of the adaptive filter},$$
$$d = \text{desired response},$$
$$e = d - y = \text{estimation error}.$$

In Table 2.1 we listed some applications that are illustrative of the four basic classes of adaptive filtering applications [1].

Here we make a brief introduction on some of these applications such as system identification, echo cancelation and adaptive beamforming.

2.2.1 System Identification

System identification is the experimental approach to the modeling of a process or a plant. It involves the following steps: experimental planning, the selection of a model structure, parameter estimation, and model validation. Here we discuss briefly the idea of adaptive filtering algorithms for estimating the parameters of an unknown plant modeled as a transversal filter [1]. Suppose we have an unknown plant that is linear and time varying. This plant is characterized by a set of discrete-time measurements that describe the variation of the plant output in response to a known input. The requirement is to develop an on-line transversal filter model for this plant, as illustrated in Fig 2.3(a). The model consists of a finite number of unit-delay elements and a corresponding set of adjustable parameters. When there is no available reference signal, we call the problem blind system identification [28, 29].

Let the available input signal at time n be denoted by the set of samples: $u(n)$,

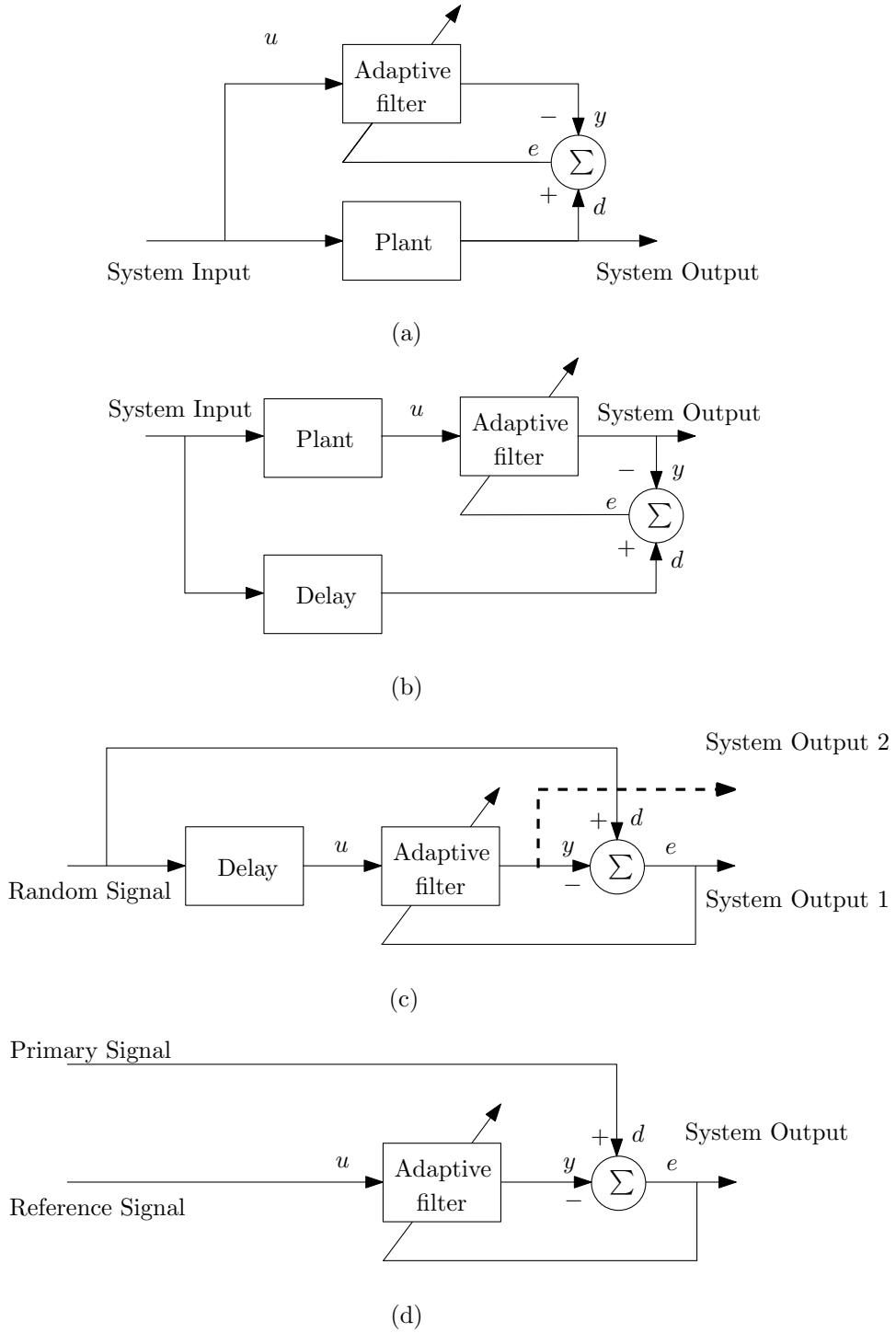

Figure 2.3: Four basic classes of adaptive filtering applications: (a) class I: identification; (b) class II: inverse modeling; (c) class III: prediction; (d) class IV: interference canceling.

$u(n-1)$, ... , $u(n-M+1)$, where M is the number of adjustable parameters in the model. This input signal is applied simultaneously to the plant and the model. Let their respective outputs be denoted by $d(n)$ and $y(n)$. The plant output $d(n)$ serves the purpose of a desired response for the adaptive filtering algorithm employed to adjust the model parameters. The model output is given by

$$y(n) = \sum_{k=1}^{M} \hat{w}_k(n)u(n-k), \tag{2.4}$$

where $\hat{w}_1(n)$, $\hat{w}_2(n)$, ... , and $\hat{w}_M(n)$ are the estimated model parameters. The model output $y(n)$ is compared with the plant output $d(n)$. The difference between them $e(n) = d(n) - y(n)$, defines the estimation error.

When the plant is time varying, the plant output is nonstationary, and so is the desired response presented to the adaptive filtering algorithm. In such a situation, the adaptive filtering algorithm has the task of not only keeping the modeling error small but also continually tracking the time variations in the dynamics of the plant.

2.2.2 Echo Cancelation

Almost all conversations are conducted in the presence of echoes. An echo may be unnoticeably distinct, depending on the time delay involved. If the delay between the speech and the echo is short, the echo is not noticeable but perceived as a form of spectral distortion or reverberation. Generally speaking, the longer the echo delay, the more it must be attenuated before it becomes noticeable [1, 15, 30, 31].

The basic principle of echo cancelation is illustrated in Fig. 2.4 for only one direction of transmission (from speaker A on the far left of the hybrid to speaker B on the right). The adaptive canceler is placed in the four-wire path near the origin of the echo. The synthetic echo, denoted by $\hat{r}(n)$, is generated by passing the speech signal from speaker A through an adaptive filter that ideally matches the transfer function of the echo path. The reference signal, passing through the hybrid, results in the echo signal $r(n)$. This echo, together with the far-end speaker signal $x(n)$ constitutes the desired response for the adaptive canceler.

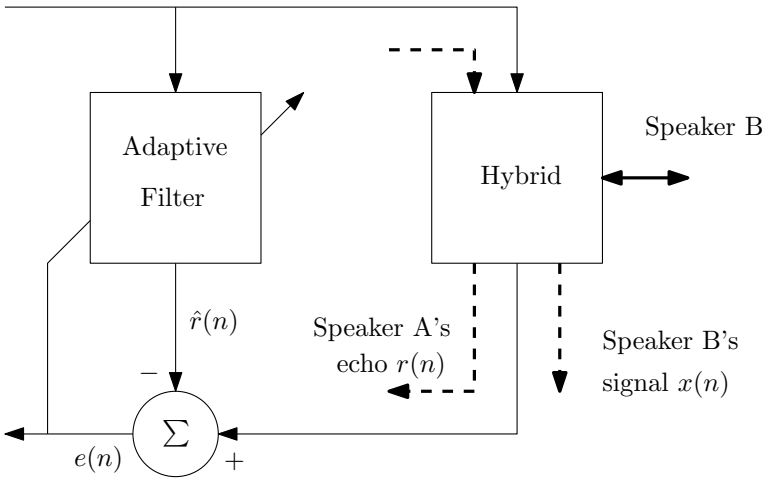

Figure 2.4: Block diagram of an echo cancelation scheme.

The synthetic echo $\hat{r}(n)$ is subtracted from the desired response $r(n) + x(n)$ to yield the canceler error signal for only one direction of transmission (from speaker A on the far left of the hybrid to speaker B on the right).

$$e(n) = r(n) - \hat{r}(n) + x(n). \tag{2.5}$$

Note that the error signal $e(n)$ also contains the far-end speaker signal $x(n)$, In any event, the error signal $e(n)$ is used to control the adjustments made in the coefficients of the adaptive filter.

2.2.3 Adaptive Beamforming

Adaptive beamforming is widely used in radar, sonar, communications, geophysical exploration and biomedical signal processing [15, 24, 32]. In radar systems, the sensors consist of antenna elements that respond to incident electromagnetic waves. In sonar, the sensors consist of hydrophones designed to respond to acoustic waves. In any event, beamforming is used in these systems to distinguish between the spatial properties of signal and noise. [2, 33–35]

In a primitive type of spatial filtering, known as the delay-and-sum beamformer, the various sensor outputs are delayed and then summed. Thus for a

single target, the average power at the output of the beamformer is maximized when it is steered toward the target. A major limitation of the delay-and-sum beamformer is that is has no provisions for dealing with the sources of interference. In order to enable a beamformer to respond to an unknown interference environment, it has to be made adaptive in such a way that is places nulls in the directions of the source of interference automatically and in real time. By so doing, the output signal-to-noise ratio (SNR) of the system is increased , and the directional response of the system is thereby improved. [1]

Adaptive beamforming takes advantage of the interference to change the directionality of the array. When transmitting, the phase and relative amplitude of the signal can be controlled by a beamformer at each transmitter, in order to create a pattern of constructive and destructive interference in the wavefront. When receiving, information from different sensors is combined in a way where the expected pattern of radiation is favorably observed. In communications, adaptive beamforming is used to point an antenna at the signal source to reduce interference and improve the transmission quality.

2.3 Adaptive filtering algorithms

Adaptive filters can be based on various basic algorithms, of which the two most known are the least mean square (LMS) and the recursive least squares (RLS) [5,6]. The LMS algorithm is an extremely simple technique from a computational complexity point of view, however, it may have a poor performance with colored signals. The RLS algorithm has often a high performance, however, it is often too complex to implement in real time [36–38]. This is one of the reasons why designers seek solutions with an improved performance as compared with the LMS and with a significantly lower complexity than the RLS for applications with large filters. As the required adaptive filter lengths grow, the conventional LMS algorithm exhibits a slower convergence rate.

In this section, the LMS, the normalized least-mean-square (NLMS) and the affine projection algorithm are introduced.

2.3.1 The Least Mean Square (LMS) Algorithm

The major advantage of the LMS algorithm is its simplicity and this feature makes the LMS a standard against other linear adaptive algorithms. The cost function of the LMS algorithms is:

$$J(n) = |e(n)|^2, \tag{2.6}$$

where $|\cdot|$ is the Euclidean norm and the $e(n)$ is the error signal, that is equal to the difference between the desired signal and the filter output signal,

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n), \tag{2.7}$$

where $\hat{\mathbf{w}}(n)$ is the filter represented by a M-by-1 tap-weight vector and $\mathbf{u}$(n) is the M-by-1 input signal. Then the gradient vector of $J(n)$ can be expressed as:

$$\frac{\partial J}{\partial \hat{\mathbf{w}}^*} = -\mathbf{p} + \mathbf{R}\hat{\mathbf{w}}(n), \tag{2.8}$$

where $\mathbf{R}$ is the correlation matrix of the received signal and $\mathbf{p}$ is the cross-correlation vector between the received signal and the desired signal. The optimum solution of such a linear filter is known as the Wiener solution that is given by

$$\hat{\mathbf{w}}_0 = \mathbf{R}^{-1}\mathbf{p}. \tag{2.9}$$

One possible solution is to estimate the gradient vector by employing instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$ as follows:

$$\begin{aligned} \mathbf{R} &= \mathbf{u}(n)\mathbf{u}^H(n), \\ \mathbf{p} &= \mathbf{u}(n)d^*(n), \end{aligned} \tag{2.10}$$

then the gradient vector is given by

$$\frac{\partial J}{\partial \hat{\mathbf{w}}^*} = -\mathbf{u}(n)d^*(n) + \mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n). \tag{2.11}$$

So the filter coefficient vector is then updated by

$$\begin{aligned}\hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) - \mu \frac{\partial J}{\partial \hat{\mathbf{w}}^*} \\ &= \hat{\mathbf{w}}(n) + \mu\, \mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \\ &= \hat{\mathbf{w}}(n) + \mu\, \mathbf{u}(n)e^*(n),\end{aligned} \tag{2.12}$$

where μ is the step-size parameter controlling the convergence and the steady-state behavior of the LMS. An approximate condition for the convergence is:

$$0 < \mu < \frac{2}{MS_{max}}, \tag{2.13}$$

where M is the length of the filter and S_{max} is the maximum value of the power spectral density of the received vector [1].

2.3.2 The Normalized LMS (NLMS) Algorithm

The normalized least mean square (NLMS) algorithm is the companion to the ordinary LMS algorithm. We may formulate the NLMS algorithm as a natural modification of the ordinary LMS. The NLMS algorithm usually converges faster than the LMS algorithm, since it utilizes a variable convergence factor aiming at the minimization of the instantaneous output error [5, 39, 40].

Denote a plant modelled by an FIR filter with M coefficients. Given the tap-input vector $\mathbf{u}(n)$ and the desired response $d(n)$, determine the tap-weight vector $\hat{\mathbf{w}}(n+1)$ so as to minimize the squared Euclidean norm of the change

$$\delta\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n), \tag{2.14}$$

in the tap-weight vector $\hat{\mathbf{w}}(n+1)$ with respect to its old value $\hat{\mathbf{w}}(n)$, subject to the constraint

$$\hat{\mathbf{w}}^H(n+1)\mathbf{u}(n) = d(n). \tag{2.15}$$

We can use the method of Lagrange multipiers to solve this optimization problem. The cost function of this problem can be expressed as

$$J(n) = \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 + Re\{\lambda^*[d(n) - \hat{\mathbf{w}}^H(n+1)\mathbf{u}(n)]\}, \tag{2.16}$$

where λ denotes the Lagrange multipliers. By using the method of Lagrange multipliers, the solution of this optimization problem is the following filter coefficient update equation [1]

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n). \tag{2.17}$$

Equation (2.17) clearly shows the reason for using the term normalized so the NLMS algorithm can be viewed as an LMS algorithm with a time-variant step-size parameter. That is the intuitive reason why the NLMS algorithm exhibits a faster rate of convergence than the conventional LMS algorithm.

2.3.3 Proportionate NLMS (PNLMS) Algorithm

The proportionate algorithm is one technique that belongs to a class of algorithms in which each filter coefficient is updated proportionally to its magnitude, resulting in higher convergence speed when the optimal weight vector is sparse [11]. The PNLMS algorithm differs from the NLMS algorithm in that the available adaptation energy is distributed unevenly over the taps [12–14]. The PNLMS algorithm is specified in Table 2.2, where M is the length of the adaptive filter, μ is the step-size parameter, δ is a small positive number used to avoid overflowing, and $\hat{w}_k(n)$ is the kth coefficient of $\hat{\mathbf{w}}$ at time n. The constant δ is important when all the coefficients are zero (in the beginning) and, together with ρ, prevent the very small coefficients from stalling.

The PNLMS algorithm has very fast initial convergence speed, which is favorable for applications such as network echo cancelation. However, after the initial period, it begins to slow down dramatically, even becoming slower than the NLMS algorithm. To solve this problem, some variations such as the improved PNLMS (IPNLMS) and μ-law PNLMS (MPNLMS) [13, 14] are also developed to keep the fast initial convergence speed during the whole adaptation process until the adaptive filter reaches its steady-state.

Table 2.2: Summary of the PNLMS algorithm

Initialization:
$\hat{\mathbf{w}}(0) = [0, 0, ..., 0]^T$
Update for each time instant: $n > 0$
$l_\infty(n+1) = \rho \max\{\delta, \|\hat{w}_1(n)\|, \|\hat{w}_2(n)\|, ..., \|\hat{w}_M(n)\|\}$
$l_k(n+1) = \max\{l_\infty(n+1), \|\hat{w}_k(n)\|\}$
$g_l(n+1) = \frac{l_k(n+1)}{\frac{1}{M}\sum_{i=1}^{M} l_i(n+1)} \quad 1 \leq i \leq L$
$G(n+1) = \text{diag}\{g_1(n+1), g_2(n+1), ..., g_M(n+1)\}$
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu G(n+1)\mathbf{u}(n)e^*(n)}{\mathbf{u}^T(n)G(n+1)\mathbf{u}(n)}$
$n = n + 1$

2.3.4 Affine Projection Algorithm (APA)

Data-reusing algorithms are considered an alternative to increasing the rate of convergence in adaptive filtering algorithms in situations where the input signal is correlated. But the data reusing will increase the misadjustment of these algorithms [2]. The APA and its variations [30, 41–45] is a popular method in adaptive filtering applications, with complexity and performance intermediary between those of LMS and of RLS. Its applications include echo cancellation, channel equalization, interference cancellation, and so forth.

Let us assume that the last N input signal vectors are organized in a M-by-N matrix as follows

$$\mathbf{U}(n) = [\mathbf{u}(n), \mathbf{u}(n-1), ..., \mathbf{u}(n-N+1)], \tag{2.18}$$

where the $\boldsymbol{u}(n)$ denotes the vector of the input signal at time n, and N denotes the APA order. We can also define some vectors representing the filter output $\mathbf{y}(n)$, the desired signal $\mathbf{d}(n)$ and the error $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$ vectors. These vectors are, respectively given by

$$\mathbf{y}(n) = [y(n), y(n-1), ..., y(n-N+1)]^T, \tag{2.19}$$

$$\mathbf{d}(n) = [d(n), d(n-1), ..., d(n-N+1)]^T, \tag{2.20}$$

From (2.18) - (2.20), we can obtain the following equation

$$\mathbf{y}(n) = \mathbf{U}^H(n)\hat{\mathbf{w}}(n). \tag{2.21}$$

For the APA, the tap-weight vector variation is defined as $\delta\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)$. The objective of the APA is to minimize

$$\begin{aligned} &\|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 \\ \text{subject to } &\mathbf{d}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n+1) = 0. \end{aligned} \tag{2.22}$$

Here again the method of Lagrange multiplier can be used to find the solution that minimizes the cost function $J(n)$.

$$J(n) = \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 + Re\{[\mathbf{d}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n+1)]^{\mathrm{H}}\boldsymbol{\lambda}\}, \tag{2.23}$$

where $\boldsymbol{\lambda} = [\lambda(0), \lambda(1), ..., \lambda(N-1)]^T$ denotes the vector of Lagrange multipliers. Then the solution of this optimization problem would be the following filter coefficient update equation

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\,\mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n) \tag{2.24}$$

with $\mu = 1$, and $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$. In general, a step-size $\mu < 1$ is used to control convergence and the steady-state behavior of the APA.

The APA is a generalization of the NLMS adaptive filtering algorithm. When the AP order N is set to one, the equation (2.24) will reduce to the familiar NLMS algorithm. As the AP order increases, so does the convergence speed of the tap weight vector. But unfortunately, the computational complexity of the algorithm will also increase significantly. So the fast affine projection (FAP) [31, 46, 47] has been developed by using a sliding windowed fast recursive-least-square (FRLS) to compute the inverse in a fast manner.

Chapter 3

Proposed Sparsity-Aware Algorithm

In this chapter, the problem of sparse system identification is briefly introduced. The proposed sparsity-aware NLMS and APA techniques that employ zero-attracting and reweighted zero-attracting strategies are then derived. A zero-forcing strategy which can improve the performance of the proposed sparsity-aware algorithms is then detailed . Finally, some experimental results are shown to illustrate their performance.

3.1 Sparse System Identification

In many scenarios of system identification, impulse responses of unknown systems can be assumed to be sparse, containing only a few large coefficients interspersed among many negligible ones. The basic idea of sparse system identification is to try to incorporate those sparse prior information to improve the filtering/estimation performance. In the past years, many algorithms exploiting sparsity were based on applying a subset selection scheme during the filtering process, which was implemented via statistical detection of active taps or sequential partial updating [8–10].

In a sparse system identification application, the desired signal is the output of an unknown sparse system when excited by an input signal. The input signal

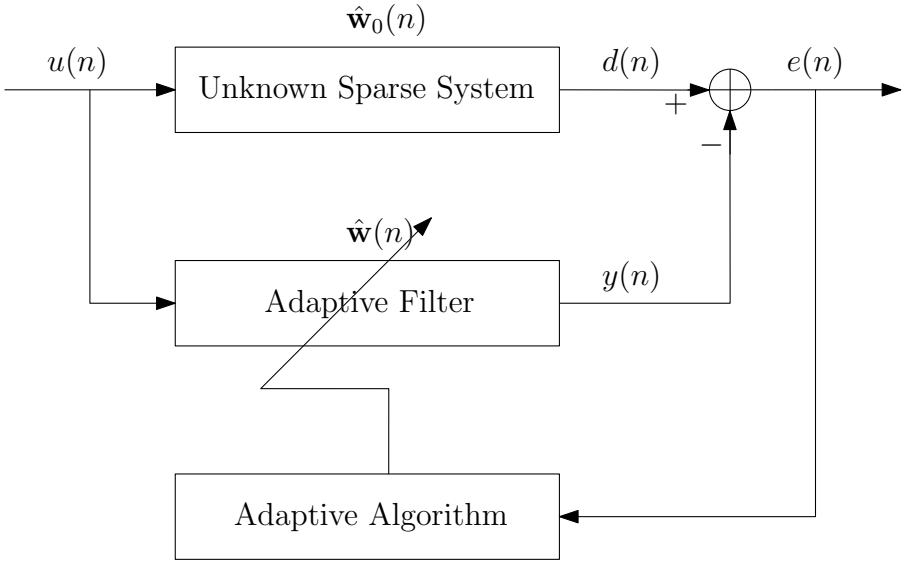

Figure 3.1: Block diagram of sparse system identification using an adaptive algorithm.

is also used as an input for an adaptive filter $\hat{\mathbf{w}}(n)$ with M coefficients to produce an output estimate $y(n)$ which is compared to the reference signal $d(n)$. The error signal $e(n)$ consists of the difference between the desired signal $d(n)$ and the output of the sparse adaptive filter $y(n)$. When the output error $e(n)$ is minimized, the adaptive filter represents a model for the unknown sparse system. The block diagram of sparse system identification is shown in Fig. 3.1. Here, $\mathbf{u}(n)$ is the input signal with M samples that is applied to the unknown sparse system, and the response signal $d(n)$ is the reference signal. The problem we are interested in solving is how to identify the unknown sparse system using an adaptive algorithm that is able to identify and exploit the sparse nature of the system.

3.2 Zero-Attracting NLMS (ZA-NLMS) Algorithm

To exploit the sparsity of the system, we incorporate the l_1-norm optimization strategy with the conventional NLMS algorithm. Here, the l_1-norm penalty on the coefficients is combined into the conventional NLMS cost function (2.14) and (2.15). The objective of the ZA-NLMS is to minimize

$$\begin{aligned} &J_1(n) = \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 + \|\hat{\mathbf{w}}(n)\|_1 \\ &\text{subject to } \; d(n) - \hat{\mathbf{w}}^H(n+1)\mathbf{u}(n) = 0. \end{aligned} \tag{3.1}$$

To solve this constrained optimization problem, we may use the method of Lagrange multipliers. By using this method, the cost function can be expressed as

$$J_1(n) = \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 + Re\{\lambda^*[d(n) - \hat{\mathbf{w}}^H(n+1)\mathbf{u}(n)]\} + \|\hat{\mathbf{w}}(n+1)\|_1. \tag{3.2}$$

We can compute the partial derivative of $J_2(n)$ with respect to $\hat{\mathbf{w}}^*(n+1)$

$$\frac{\partial J_1(n)}{\partial \hat{\mathbf{w}}^*(n+1)} = \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) - \mathbf{u}(n)\lambda^* + \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n+1)], \tag{3.3}$$

where $\mathrm{sgn}[\,\cdot\,]$ is a function that returns the sign of the arguments. Denote a complex number $z = a + bj$, the $\mathrm{sgn}[\,\cdot\,]$ is defined as

$$\mathrm{sgn}(z) = \begin{cases} 1+j & \text{if } a > 0, b > 0 \\ 1-j & \text{if } a > 0, b < 0 \\ j & \text{if } a = 0, b > 0 \\ 1 & \text{if } a > 0, b = 0 \\ 0 & \text{if } a = 0, b = 0 \\ -1 & \text{if } a < 0, b = 0 \\ -j & \text{if } a = 0, b < 0 \\ -1+j & \text{if } a < 0, b > 0 \\ -1-j & \text{if } a < 0, b < 0 \end{cases} \tag{3.4}$$

By equating (3.3) to zero, then we can get

$$\lambda = \frac{e(n)}{\|\mathbf{u}(n)\|^2} + \frac{\alpha}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n+1)]u(n). \tag{3.5}$$

Assuming that $\mathrm{sgn}[\hat{\mathbf{w}}(n+1)] \approx \mathrm{sgn}[\hat{\mathbf{w}}(n)]$, then combining (3.5) with (3.3), we can get the filter coefficient update equation of ZA-NLMS:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n) + \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]. \tag{3.6}$$

Comparing the ZA-NLMS update equation (3.6) with the conventional NLMS function (2.17), we can see an additional term $\alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]$, which attracts the tap coefficients to zero. We call this the zero attractor feature, whose strength is controlled by α. Intuitively, the zero attractor will speed-up convergence when the majority of coefficients of $\hat{\mathbf{w}}$ are zero, i.e., the system is sparse. Table 3.1 summarizes the ZA-NLMS algorithm.

Table 3.1: Summary of the ZA-NLMS algorithm

Initialization:
$\hat{\mathbf{w}}(0) = [0, 0, ..., 0]^T$
Update for each time instant: $n > 0$
$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$
$e(n) = d(n) - y(n)$
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n) + \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]$
$n = n + 1$

3.3 Zero-Attracting Affine Projection Algorithm (ZA-APA)

For conventional APA, we can also apply the same strategy to get a new cost function $J_2(n)$ by combining the instantaneous square error with the l_1-norm

penalty of the coefficient vector. The new cost function is shown as below

$$J_2(n) = \|\hat{\mathbf{w}}(n+1)-\hat{\mathbf{w}}(n)\|^2+Re\{[\mathbf{d}(n)-\mathbf{U}^H(n)\hat{\mathbf{w}}(n+1)]\boldsymbol{\lambda}\}+\alpha\,\|\hat{\mathbf{w}}(n+1)\|_1. \quad (3.7)$$

To minimize the cost function, we can compute the partial derivative of $J_2(n)$ with respect to $\hat{\mathbf{w}}^*(n+1)$

$$\frac{\partial J_2(n)}{\partial \hat{\mathbf{w}}^*(n+1)} = \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) - \mathbf{U}(n)\boldsymbol{\lambda} + \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n+1)]. \quad (3.8)$$

By equating (3.8) to zero, we get

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{U}^H(n)\boldsymbol{\lambda} - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n+1)]. \quad (3.9)$$

Multiplying both sides by $\mathbf{U}^H(n)$ from the left, we obtain

$$\mathbf{d}(n) = \mathbf{U}^H(n)\hat{\mathbf{w}}(n) + \mathbf{U}^H(n)\mathbf{U}(n)\boldsymbol{\lambda} - \alpha\,\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n+1)]. \quad (3.10)$$

Because $\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n)$ we can solve for $\boldsymbol{\lambda}$. Assuming that $\mathrm{sgn}[\hat{\mathbf{w}}(n+1)]\approx\mathrm{sgn}[\hat{\mathbf{w}}(n)]$, with further manipulations, we can obtain the new filter coefficient update equation

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n)+\mu\,\mathbf{U}^+(n)\mathbf{e}(n)+\alpha\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)]-\alpha\mathrm{sgn}[\hat{\mathbf{w}}(n)], \quad (3.11)$$

where $\mathbf{U}^+(n) = \mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}$.

Comparing the ZA-APA update (3.11) to the standard APA update (2.24), the ZA-APA has two additional terms, which attract the tap coefficients to zero. In addition, if we set the AP order N to one, (3.11) reduces to the update formula for the Zero-Attracting NLMS (ZA-NLMS) algorithm. A summary of the ZA-APA is shown in Table 3.2.

Table 3.2: Summary of the ZA-APA

Initialization:
$\hat{\mathbf{w}}(0) = [0, 0, ..., 0]^T$
Update for each time instant: $n > 0$
$\mathbf{y}(n) = \mathbf{U}^H(n)\hat{\mathbf{w}}(n)$
$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$
$\mathbf{U}^+(n) = \mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}$
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\,\mathbf{U}^+(n)\mathbf{e}(n) + \alpha\,\mathbf{U}^+(n)\mathbf{U}^H(n)\text{sgn}[\hat{\mathbf{w}}(n)] - \alpha\text{sgn}[\hat{\mathbf{w}}(n)]$
$n = n + 1$

3.4 Reweighted Zero-Attracting Affine Projection Algorithm (RZA-APA)

Unfortunately, the ZA-APA does not distinguish between zero taps and non-zero taps. Since all the taps are forced to zero uniformly, the performance of ZA-APA can be deteriorated when applied to less sparse systems. In order to solve this problem, we adopt a heuristic approach [8, 9, 26] to reinforce the zero attractor called the reweighted zero-attracting affine projection algorithm (RZA-APA). For the RZA-APA, we use a new l_1-norm penalty to minimize the cost function

$$\begin{aligned} J_3(n) &= \|\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)\|^2 + \alpha \sum_{i=1}^{M} \log(1 + \frac{|\hat{w}_i|}{\varepsilon}) \\ &\text{subject to } \mathbf{d}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n+1) = 0. \end{aligned} \tag{3.12}$$

Here, we may use the method of Lagrange multipliers to solve this constrained optimization problem,

$$\frac{\partial J_3(n)}{\partial \hat{\mathbf{w}}^*(n+1)} = \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) - \mathbf{U}(n)\boldsymbol{\lambda} + \alpha \frac{\text{sgn}(\hat{\mathbf{w}}(n+1))}{1 + \varepsilon|\hat{\mathbf{w}}(n+1)|}, \tag{3.13}$$

By equating (3.13) to zero, we get the new filter coefficient update equation

of RZA-APA

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\,\mathbf{U}^{+}(n)\mathbf{e}(n) + \alpha\,\mathbf{U}^{+}(n)\mathbf{U}^{H}(n)\mathbf{S}(n) - \alpha\mathbf{S}(n), \tag{3.14}$$

where

$$\mathbf{U}^{+}(n) = \mathbf{U}(n)[\mathbf{U}^{H}(n)\mathbf{U}(n)]^{-1},$$
$$S_k(n) = \frac{\mathrm{sgn}[\hat{\mathbf{w}}_k(n)]}{1+\varepsilon|\hat{\mathbf{w}}_k(n)|},\ \text{for } 0{\leq}k{\leq}M,$$

where ε is the shrinkage magnitude, $\hat{\mathbf{w}}_k(n)$ is the kth coefficient of $\hat{\mathbf{w}}$ at time instant n, and $S_k(n)$ is the kth coefficient of $\mathbf{S}$ at time instant n.

The RZA-APA is more sensitive to taps with small magnitudes. The reweighted zero attractor takes more shrinkage exerted on those taps for which magnitudes are comparable to $1/\varepsilon$; and takes less effort on the taps whose $|\hat{\mathbf{w}}(n)|{\gg}1/\varepsilon$. In this way, the bias of the RZA-APA can be reduced. We show a summary of the RZA-APA in Table 3.3.

Table 3.3: Summary of the RZA-APA

Initialization:
$\hat{\mathbf{w}}(0) = [0, 0, ..., 0]^T$
Update for each time instant: $n > 0$
$\mathbf{y}(n) = \mathbf{U}(n)\hat{\mathbf{w}}(n)$
$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n)$
$\mathbf{U}^{+}(n) = \mathbf{U}(n)[\mathbf{U}^{H}(n)\mathbf{U}(n)]^{-1}$
for k=1,2,...,M
{
$S_k(n) = \frac{\mathrm{sgn}[\hat{\mathbf{w}}_k(n)]}{1+\varepsilon\lvert\hat{\mathbf{w}}_k(n)\rvert}$
}
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\,\mathbf{U}^{+}(n)\mathbf{e}(n) + \alpha\,\mathbf{U}^{+}(n)\mathbf{U}^{H}(n)\mathbf{S}(n) - \alpha\mathbf{S}(n)$
$n = n + 1$

The update equation for RZA-NLMS can be obtained from the RZA-APA by

setting N equal to zero and is described by

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)e^*(n) - \alpha\,\frac{\mathrm{sgn}[\hat{\mathbf{w}}(n)]}{1+\varepsilon|\hat{\mathbf{w}}(n)|}. \tag{3.15}$$

3.5 Zero-Forcing Technique

In many scenarios of sparse system identification, some of the tap-weight coefficients are so small that they can be ignored. By ignoring those taps, the computational complexity of the adaptive algorithm could also be reduced. In addition, we can also get some performance gain if those taps that are significantly small and are associated with zero coefficients of the impulse response of the system can be forced to zero. From this point of view, we set up a new idea forcing those small tap-weight coefficients to zero after some iterations. This technique can be incorporated into any sparsity-aware adaptive algorithm.

We incorporate the zero-forcing technique into the ZA-NLMS algorithm as an example to show how it works. In the normal procedure of ZA-NLMS, we will calculate the sign of every tap-weight coefficient in every iteration. When the algorithm is about to converge after the Lth iteration, if some of the coefficients are smaller than the zero-attractor procedure α, they will be forced to change their sign after every iteration. These calculations are not necessary. Therefore, we can set up a threshold η to identify if there are any coefficients that are small enough so that we could force them to zero to reduce the computational complexity. By using this zero-forcing technique, we can get a new algorithm called zero-forcing NLMS(ZF-NLMS). Here, we often choose the time instant $L = 2M$ which is 2 times of the filter length to start the zero-forcing procedure.

Denote $\hat{w}_k(n)$ the kth coefficient of $\hat{\mathbf{w}}$ at time n, the summary of the ZF-NLMS is shown in Table 3.4:

We can see that the difference between ZF-NLMS and ZA-NLMS is just the zero-forcing procedure. This technique can also be incorporate to other algorithms such as ZA-APA, which in turn get the ZF-APA. In the next section, experiments will be designed to test the performance of these proposed algorithms.

Table 3.4: Summary of the ZF-NLMS algorithm

Initialization:

$\hat{\mathbf{w}}(0) = [0, 0, ..., 0]^T$

Update for each time instant: $n > 0$

Step 1: $0 < n < 2M$

$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$

$e(n) = d(n) - y(n)$

$\mu^{'} = \frac{\mu}{\|\mathbf{u}(n)\|^2}$

$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu^{'}\mathbf{u}(n)e^*(n) + \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]$

$n = n + 1$

Step 2: $n > 2M$

For $k = 1, 2, ..., M-1$

{

If $\hat{w}_k(n) < \eta$

$\hat{w}_k^{'}(n) = 0$

$k = k + 1;$

}

$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$

$e(n) = d(n) - y(n)$

$\mu^{'} = \frac{\mu}{\|\mathbf{u}(n)\|^2}$

$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu^{'}\mathbf{u}(n)e^*(n) + \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]$

$n = n + 1$

3.6 Simulations

In this section, simulation results are given to show the performance of the proposed sparsity-aware algorithms in stationary scenarios.

Firstly, we set up a simulation example to compare the performance of proposed sparsity-aware NLMS algorithms compared with the PNLMS and IPNLMS algorithms, which are also improvements of the conventional NLMS algorithm, also designed specified for sparse systems. Then four simulations are set to analyze the performance of zero-attracting algorithms, including the comparison between NLMS, ZA-NLMS, RZA-NLMS, APA, ZA-APA and RZA-APA. The last two simulations are for the zero-forcing algorithm, including the ZF-NLMS, ZF-APA, RZF-NLMS and RZF-APA. The input signals we use in these simulations

are independent and identically distributed (i.i.d.) signals, and the measurement noise here are i.i.d Gaussian white noise as well. In these simulations, the SNR of the system is set to 30dB, the affine projection order N is set to 4, and other parameters of the simulation system are shown in Table 3.5

Table 3.5: Table of Parameters

Exp.	Algorithm	Step-size μ	Zero-attractor α	Reweighted ε
No. 1	NLMS	0.5	5×10^{-4}	100
No. 2	APA	0.5	5×10^{-4}	100
	NLMS	0.5	5×10^{-4}	100
No. 3	APA	0.5	1×10^{-3}	100
	NLMS	0.5	1×10^{-3}	100
No. 4	APA	0.5	1×10^{-3}	100
	NLMS	0.5	1×10^{-3}	100
No. 5	APA	1	1×10^{-3}	100
	NLMS	1	1×10^{-3}	100
No. 6	NLMS	1	1×10^{-3}	100
No. 7	APA	1	1×10^{-3}	100

In the first experiment, we compared the proposed sparsity-aware NLMS algorithms with the conventional NLMS, PNLMS and IPNLMS algorithms. In this experiment, we introduce a 32-tap system with only 2 non-zero coefficients, which is a significantly sparse system. The simulation result describing the MSE against the number of iterations is shown in Fig. 3.2.

As we can see from the MSE results, the PNLMS algorithm has a slightly faster initial convergence rate at the start, but begins to slow down after the initial period. The IPNLMS, which is a improved variation of the PNLMS, has a better convergence rate in the initial period than the PNLMS, but it still does not achieve a better steady-state performance than our proposed sparsity-aware algorithms. Both of our proposed algorithm get the best steady-state performance, especially the RZA-NLMS. So we will focus on our proposed sparsity-aware algorithms in the following experiments.

A system with 32 tap-weight coefficients is set in the next three experiments. Note that the number of non-zero taps for each experiment are 2, 16 and 32. In the experiment 2, we set the 4th and $6th$ tap with value 1 and the others to zero, so that it is a really sparse system with sparsity factor 2/32. In the experiment

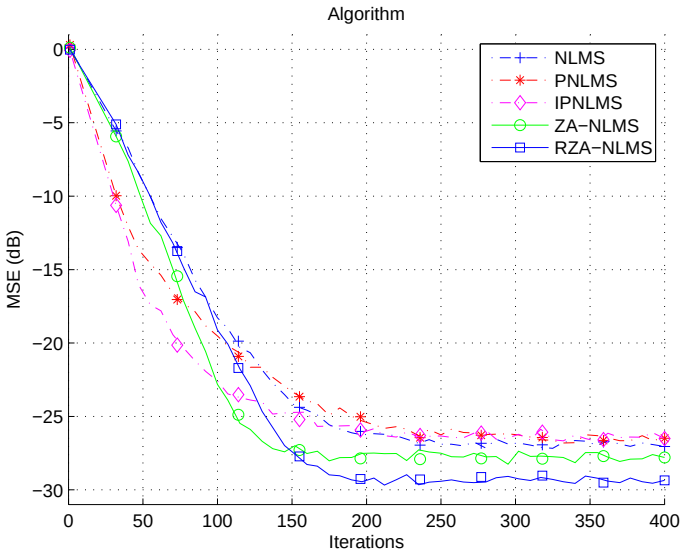

Figure 3.2: Simulated MSE for experiment 1.

3, all the odd taps are set to 1, while all the even taps remain equal to zero. For the experiment 4, all the taps are set to 1, which means it is a totally non-sparse system. In these three experiments, we choose the step-size $\mu = 0.5$ to achieve balance between convergence rate and steady-state performance.

For the experiment 2, the average estimate of mean square error (MSE) is shown in Fig. 3.3. As we can see from the MSE results, both the zero-attracting algorithms and reweight zero-attracting algorithms achieve faster convergence rate and better steady-state performance than the conventional NLMS and APA. The convergence rate of APA is also much faster than the NLMS algorithms. We can also see that when the system is significantly sparse, the convergence rate of the zero-attracting algorithms are faster than the reweighted ones.

As we can see from Fig. 3.4, when the number of non-zero taps increases to 16, the performance of ZA-NLMS and ZA-APA deteriorates, while the RZA-NLMS and RZA-APA still have the best convergence rate. Again, the APA algorithms achieve better convergence rate than the NLMS algorithms.

In the 4th experiment, we introduce a non-sparse system to test if these proposed algorithms can still work in this bad environment. As it can be seen from

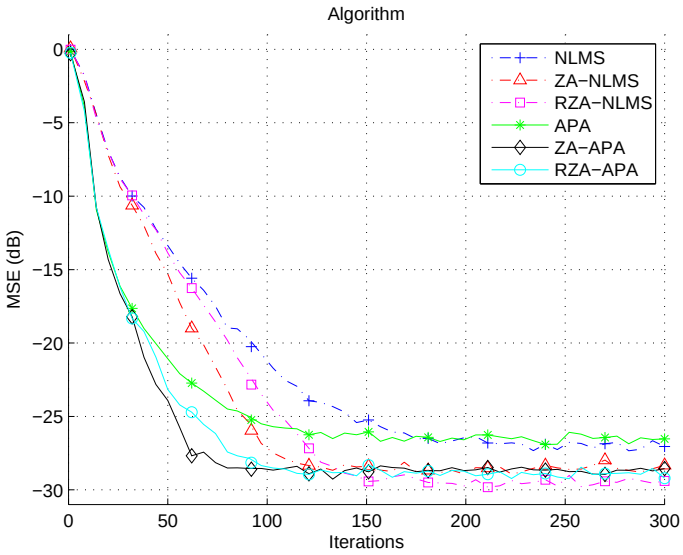

Figure 3.3: Simulated MSE for experiment 2.

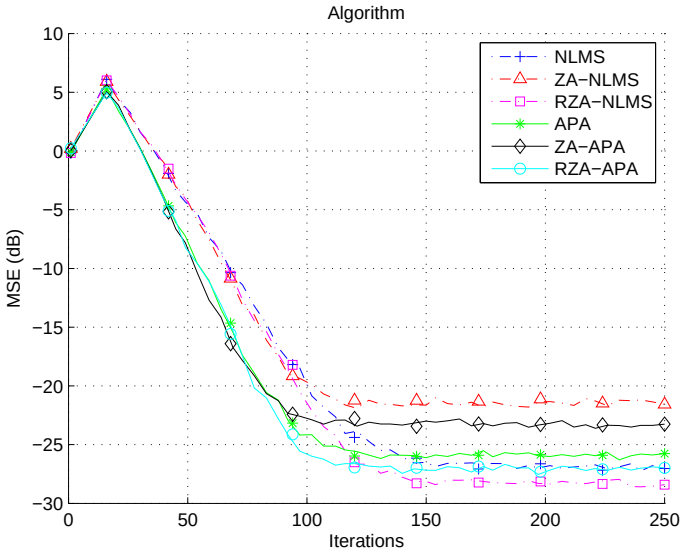

Figure 3.4: Simulated MSE for experiment 3.

the Fig. 3.5, RZA-NLMS and RZA-APA show a robust performance when the system is non-sparse. Due to the uniform zero-attractor parameter, the perfor-

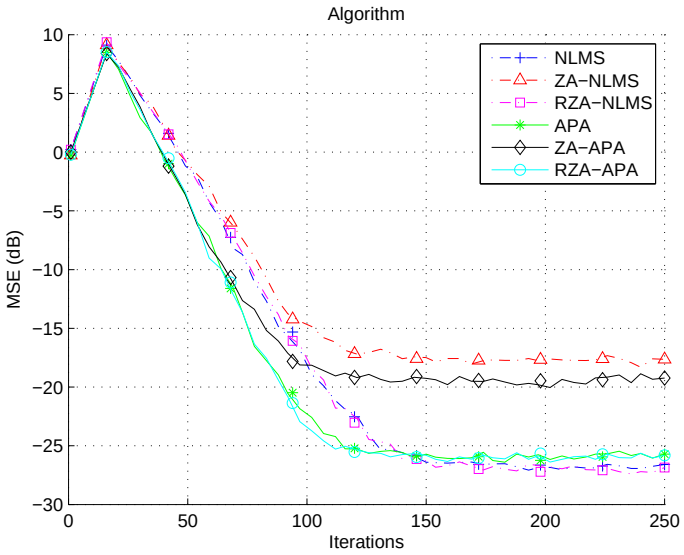

Figure 3.5: Simulated MSE for experiment 4.

mance of ZA-NLMS and ZA-APA is not robust in the non-sparse system.

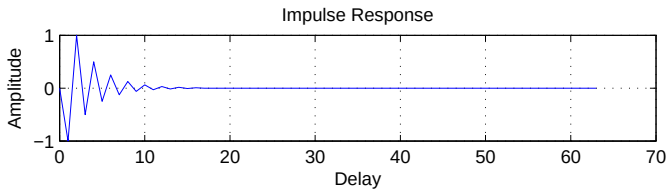

Figure 3.6: The impulse response of the system in the experiment 5.

In the 5th experiment, we introduce a 64-tap system with 16 non-zero coefficients. The impulse response of the system is show in Fig. 3.6. Since this is a large system, it will be much slower in convergence rate, so we choose the step-size $\mu = 1$ to achieve a balance between the steady-state performance and the convergence rate. From Fig. 3.7, for this large sparse system, the reweighted algorithms significantly outperform the conventional NLMS and APA in steady-state MSE. The RZA-APA still has a faster convergence rate compared to the RZA-NLMS. On the other hand, both of them can achieve a good steady-state MSE almost at the same level.

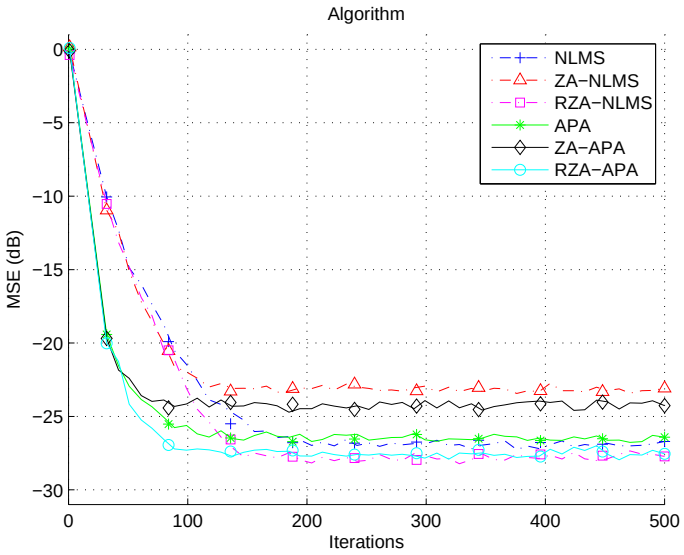

Figure 3.7: Simulated MSE for experiment 5.

In the next two simulations, we introduce a 32-tap sparse system with only 2 non-zero coefficients, to show the improvement that the zero-forcing technique can provide if the system is significantly sparse. The first experiment is the comparison between NLMS algorithms, and the second one is for the APA algorithms.

In experiment 6, the zero-forcing threshold $\eta = 0.005$, which is 5 times higher than the zero-attracting parameter $\alpha = 10^{-3}$. We can see from Fig. 3.8 that when the system is very sparse, the zero-forcing technique can help the ZA-NLMS and RZA-NLMS to get a better result both in steady-state MSE and convergence rate by making full use of the sparse prior information.

For the 7th experiment, as we can see from Fig. 3.9, the zero-forcing technique can also slightly improve the steady-state performance and convergence rate of the sparsity-aware APA.

3.7 Summary

In this chapter, a set of sparsity-aware zero-attracting adaptive algorithms have been developed by incorporating the l_1-norm penalty on the coefficients with the

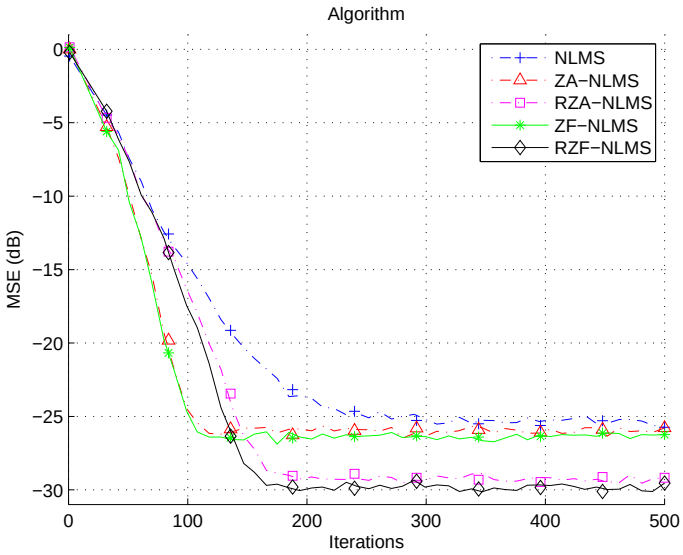

Figure 3.8: Simulated MSE for experiment 6.

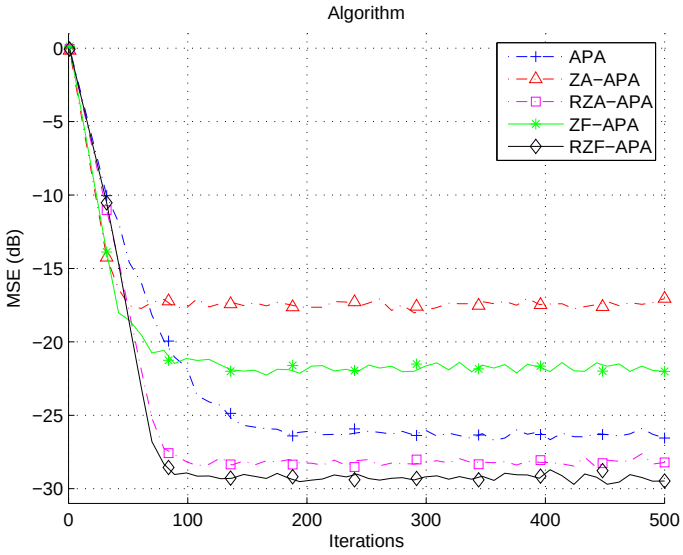

Figure 3.9: Simulated MSE for experiment 7.

conventional NLMS and APA. We have also developed a zero-forcing technique to further improve the performance of the proposed zero-attracting algorithms.

Simulation results have been carried out to compare the proposed algorithms with the conventional ones. As we can see from the results, the proposed algorithms possess faster convergence-rate and better steady-state performance. They can be applied to a number of applications in which the signals and systems under consideration have a sparse nature, resulting in a better performance than the conventional NLMS and APA. Specifically, we have considered the proposed algorithms to identification of sparse systems that occur in applications of control and echo cancellation. We will further analyze the proposed algorithms in the next chapter.

Chapter 4

Analysis of the Proposed Sparsity-Aware Algorithms

In this chapter, an analysis of the proposed algorithms is carried out and includes an assessment of the computational complexity, a convergence analysis, and a steady-state analysis. Simulation results are also provided to illustrate the effectiveness of the analytical expressions for predicting the mean-square error performance of the proposed algorithms.

4.1 Computational Complexity

In this section, we discuss the computational complexity of the proposed sparsity-aware adaptive algorithms and compare them with the complexity of conventional adaptive algorithms. The complexity considered here is the arithmetic complexity, which includes additions and multiplications. We assume that there are only Q non-zero taps in a sparse system modelled as an FIR filter with M coefficients, and the order of the APA is N. For data without a time-shifting structure, we detail the computational complexity of the algorithms in terms of additions and multiplications as shown in Table 4.1.

We set up two experiments to analyze the computational complexity of the NLMS, ZA-NLMS, RZA-NLMS, AP, ZA-AP and RZA-AP algorithms, in which the computational complexity of the proposed sparsity-aware algorithms are shown

Table 4.1: Computational Complexity

Algorithm	Additions	Multiplications	Divisions
NLMS	$3M$	$3M+1$	1
ZA-NLMS	$M+3Q$	$M+3Q+1$	1
RZA-NLMS	$M+4Q$	$M+4Q+1$	$M+1$
APA	N^2M+MN $+Q-M+O(N^3)$	N^2M+N^2+MN $+N+O(N^3)$	M
ZA-APA	N^2M+N^2+3MN $-2N-2M+3Q+O(N^3)$	N^2M+2N^2+3MN $+2N+Q+O(N^3)$	M
RZA-APA	N^2M+N^2+3MN $-2N-2M+4Q+O(N^3)$	N^2M+2N^2+3MN $+N+M+2Q+O(N^3)$	$2M$

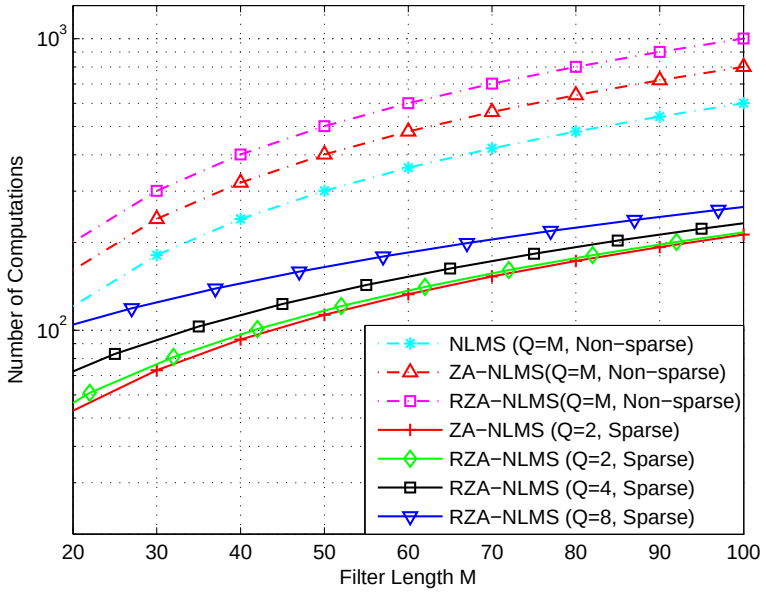

Figure 4.1: The computational complexity of the NLMS algorithms.

as a function of M, and include both additions and multiplications. From Fig. 4.1, we can see that the complexity of ZA-NLMS and RZA-NLMS is a little higher than the conventional NLMS algorithms. In addition, the sparsity of the system can also help reduce the computational complexity of the proposed algorithms. We can also see in Fig. 4.2 that the proposed ZA-APA and RZA-APA have the same effect as the NLMS algorithm.

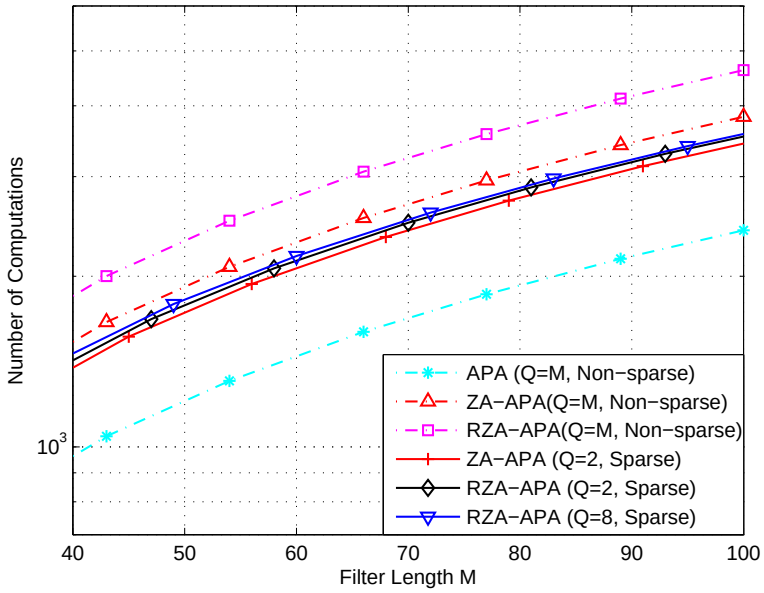

Figure 4.2: The computational complexity of the APA (Affine projection order $N = 4$).

4.2 Analysis of the Proposed Algorithms

In this section, we investigate several characteristics of the proposed ZA-NLMS and ZA-AP algorithms and carry out an analysis of them. Firstly, a sufficient condition for the convergence of the mean weight vector is obtained. Then, steady-state mean-square error expressions are derived.

4.2.1 ZA-NLMS Algorithm

Firstly, we analyze the convergence behavior of the ZA-NLMS algorithm. We use the energy-conservation approach [30, 48–50] to derive theoretical expressions for the excess mean-square error (EMSE) of the ZA-NLMS algorithm. Let us consider the reference data $d(n)$ that arise from the linear model

$$d(n) = \hat{\mathbf{w}}_0^H \mathbf{u}(n) + v(n), \tag{4.1}$$

where $\hat{\mathbf{w}}_0$ is a tap-weight vector of the system that we wish to estimate, $v(n)$ accounts for measurement noise at time instant n, and $\mathbf{u}(n)$ denotes the input signal at time instant n. Our objective is to evaluate the steady-state MSE performance of the ZA-NLMS algorithm. The steady-state MSE can be defined as

$$\text{MSE} \triangleq \lim_{n\to\infty} E|e(n)|^2, \tag{4.2}$$

where

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \tag{4.3}$$

is the output estimation error at time n.

Combining (4.1) with (4.3), we can get

$$\begin{aligned} e(n) &= \hat{\mathbf{w}}_0^H(n)\mathbf{u}(n) + v(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \\ &= [\hat{\mathbf{w}}_0 - \hat{\mathbf{w}}(n)]^H\mathbf{u}(n) + v(n). \end{aligned} \tag{4.4}$$

Combining the update equation of the ZA-NLMS algorithm (3.6) with (4.4), we get

$$\begin{aligned} \delta\hat{\mathbf{w}}(n+1) \triangleq& \hat{\mathbf{w}}_0 - \hat{\mathbf{w}}(n+1) \\ =& \hat{\mathbf{w}}_0 - \left\{ \hat{\mathbf{w}}(n) + \frac{\mu\mathbf{u}(n)e^*(n)}{\|\mathbf{u}(n)\|^2} + \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\text{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\text{sgn}[\hat{\mathbf{w}}(n)] \right\} \\ =& \left[I - \frac{\mu\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2} \right] \delta\hat{\mathbf{w}}(n) - \frac{\mu}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)v^*(n) \\ & - \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\text{sgn}[\hat{\mathbf{w}}(n)] + \alpha\,\text{sgn}[\hat{\mathbf{w}}(n)]. \end{aligned} \tag{4.5}$$

Because the measurement noise $v(n)$ is statistically independent of the input signal $\mathbf{u}(n)$, we have $E[\mathbf{u}(n)v^*(n)] = \mathbf{0}$. Taking expectations of both sides of (4.5), we get

$$\begin{aligned} E\left[\delta\hat{\mathbf{w}}(n+1)\right] =& E\left[I - \frac{\mu\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2} \right] E\left[\delta\hat{\mathbf{w}}(n)\right] \\ & - \alpha E\left[\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2} \right] E\left\{\text{sgn}[\hat{\mathbf{w}}(n)]\right\} + \alpha\, E\left\{\text{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.6}$$

If the SNR is large, and α is small, we can assume that in the steady-state

$$E\left\{\text{sgn}[\hat{\mathbf{w}}(n)]\right\} \approx \text{sgn}(\hat{\mathbf{w}}_0). \tag{4.7}$$

In addition, ref. [51, 52] show that, for long filters ($M \gg 1$), expected value related to $\mathbf{u}(n)$ can be approximated as

$$E\left[\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\right] \approx \frac{E[\mathbf{u}(n)\mathbf{u}^H(n)]}{E[\|\mathbf{u}(n)\|^2]} = \frac{\mathbf{R}}{\text{Tr}(\mathbf{R})} = \frac{\mathbf{R}}{M\sigma_u^2}, \tag{4.8}$$

where $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$, and σ_u^2 is the power of the input signal $\mathbf{u}(n)$. Then (4.6) can be rewritten as

$$E\left[\delta\hat{\mathbf{w}}(n+1)\right] = \left[\mathbf{I} - \frac{\mu\mathbf{R}}{M\sigma_u^2}\right] E\left[\delta\hat{\mathbf{w}}(n)\right] - \frac{\alpha\mathbf{R}}{M\sigma_u^2}\text{sgn}(\hat{\mathbf{w}}_0) + \alpha\,\text{sgn}(\hat{\mathbf{w}}_0). \tag{4.9}$$

Note that the matrix $\alpha\,\text{sgn}(\hat{\mathbf{w}}_0)$ is bounded between $-\alpha\mathbf{I}$ and $\alpha\mathbf{I}$. Therefore, $E[\delta\hat{\mathbf{w}}(n)]$ remains bounded if $[\mathbf{I} - \frac{\mu\mathbf{R}}{M\sigma_u^2}]$ is less than 1, which is satisfied by

$$0 < \mu < \frac{M\sigma_u^2}{\lambda_{\max}}, \tag{4.10}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix of $\mathbf{u}(n)$. We can see that the stability condition of the ZA-NLMS algorithm is independent of the zero-attractor parameter α.

For $n \to \infty$, we assume that the mean coefficient vector $E[\hat{\mathbf{w}}(n)]$ converges. From (4.9), we obtain

$$E\left[\delta\hat{\mathbf{w}}(\infty)\right] = \left[\mathbf{I} - \frac{\mu\mathbf{R}}{M\sigma_u^2}\right] E\left[\delta\hat{\mathbf{w}}(\infty)\right] - \frac{\alpha\mathbf{R}}{M\sigma_u^2}\text{sgn}(\hat{\mathbf{w}}_0) + \alpha\,\text{sgn}(\hat{\mathbf{w}}_0), \tag{4.11}$$

which can be rearranged as

$$E\left[\delta\hat{\mathbf{w}}(\infty)\right] = -\frac{\alpha}{\mu}\text{sgn}(\hat{\mathbf{w}}_0) + \frac{\alpha M\sigma_u^2}{\mu}\mathbf{R}^{-1}\text{sgn}(\hat{\mathbf{w}}_0). \tag{4.12}$$

Then we obtain

$$E\left[\hat{\mathbf{w}}(\infty)\right] = \hat{\mathbf{w}}_0 - \frac{\alpha}{\mu}\mathrm{sgn}(\hat{\mathbf{w}}_0) + \frac{\alpha M \sigma_u^2}{\mu}\mathbf{R}^{-1}\mathrm{sgn}(\hat{\mathbf{w}}_0). \tag{4.13}$$

Note that (4.13) implies that the optimum solution of the ZA-NLMS algorithm is biased, as was also shown for ZA-LMS in [8].

We then proceed to derive the steady-state MSE expression for the ZA-NLMS algorithm. Multiplying both sides of (3.6) by $\mathbf{u}(n)$ from the right, we obtain

$$\begin{aligned}
\hat{\mathbf{w}}^H(n+1)\mathbf{u}(n) =& \hat{\mathbf{w}}^H(n)\mathbf{u}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2}e(n)\mathbf{u}^H(n)\mathbf{u}(n) \\
&+ \frac{\alpha}{\|\mathbf{u}(n)\|^2}\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{u}(n) - \alpha\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\mathbf{u}(n) \\
\hat{\mathbf{w}}^H(n+1)\mathbf{u}(n) =& \hat{\mathbf{w}}^H(n)\mathbf{u}(n) + \mu e(n).
\end{aligned} \tag{4.14}$$

Introducing the a posteriori and a priori error $e_p(n)$ and $e_a(n)$, we have

$$\begin{aligned}
e_p(n) &= \hat{\mathbf{w}}_0^H\mathbf{u}(n) - \hat{\mathbf{w}}^H(n+1)\mathbf{u}(n), \\
e_a(n) &= \hat{\mathbf{w}}_0^H\mathbf{u}(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n).
\end{aligned} \tag{4.15}$$

Then, from (4.14), it holds that

$$e_p(n) = e_a(n) - \mu e(n). \tag{4.16}$$

Combining (4.3) and (4.15), we obtain

$$\begin{aligned}
e(n) =& d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \\
=& \hat{\mathbf{w}}_0^H\mathbf{u}(n) + v(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \\
=& e_a(n) + v(n).
\end{aligned} \tag{4.17}$$

Then (4.16) can be rewritten as

$$\begin{aligned}
e_p(n) =& e_a(n) - \mu e(n) \\
=& (1-\mu)e(n) - v(n).
\end{aligned} \tag{4.18}$$

We can also use (4.16) to express $e(n)$ as follows,

$$e(n) = \frac{1}{\mu}[e_a(n) - e_p(n)]. \tag{4.19}$$

Substituting the above into (3.6), we get

$$\begin{aligned}\hat{\mathbf{w}}(n+1) =& \hat{\mathbf{w}}(n) + \frac{1}{\|\mathbf{u}(n)\|^2}\mathbf{u}(n)[e_a(n) - e_p(n)]^* \\ &+ \frac{\alpha\,\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\,\mathrm{sgn}[\hat{\mathbf{w}}(n)].\end{aligned} \tag{4.20}$$

In the steady-state condition, $E[\|\hat{\mathbf{w}}(n+1)\|^2] \approx E[\|\hat{\mathbf{w}}(n)\|^2]$ when $n \to \infty$, and assuming that $e_a(n)$ and $\hat{\mathbf{w}}(n)$ are independent of $u(n)$ in steady-state. By evaluating the energies of both sides of the above equation, we obtain

$$\begin{aligned}E\left[e_p(n)\frac{\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\frac{\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2}e_p^*(n)\right] =\; & E\left[e_a(n)\frac{\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\frac{\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2}e_a^*(n)\right] \\ & + E\left\{\alpha\, e_a(n)\frac{\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ & - E\left\{\alpha\, e_a(n)\frac{\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ & + E\left\{\alpha\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathbf{u}(n)e_a^*(n)\right\} \\ & + E\left\{\alpha^2\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ & - E\left\{\alpha^2\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ & - E\left\{\alpha\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2}e_a^*(n)\right\} \\ & - E\left\{\alpha^2\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ & + E\left\{\alpha^2\,\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\},\end{aligned} \tag{4.21}$$

with $\frac{\mathbf{u}^H(n)\mathbf{u}(n)}{\|\mathbf{u}(n)\|^2} = 1$, (4.21) can be reduced to

$$\begin{aligned} E\left[e_p(n)\frac{1}{\|\mathbf{u}(n)\|^2}e_p^*(n)\right] &= E\left[e_a(n)\frac{1}{\|\mathbf{u}(n)\|^2}e_a^*(n)\right] \\ &\quad - \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ &\quad + \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ &= E\left[e_a(n)\frac{1}{\|\mathbf{u}(n)\|^2}e_a^*(n)\right] \\ &\quad + \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left[\mathbf{I} - \frac{\mathbf{u}(n)\mathbf{u}^H(n)}{\|\mathbf{u}(n)\|^2}\right]\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.22}$$

With the assumption (4.8), we obtain

$$\begin{aligned} \frac{1}{M\sigma_u^2}E\left[e_p(n)e_p^*(n)\right] = &\quad \frac{1}{M\sigma_u^2}E\left[e_a(n)e_a^*(n)\right] \\ &+ \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(\mathbf{I} - \frac{\mathbf{R}}{M\sigma_u^2}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}, \\ E\left[e_p(n)e_p^*(n)\right] = &\quad E\left[e_a(n)e_a^*(n)\right] \\ &+ \alpha^2 E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(M\sigma_u^2\mathbf{I} - \mathbf{R}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}, \end{aligned} \tag{4.23}$$

where σ_u^2 is the power of the input signal $\mathbf{u}(n)$.

Substituting (4.18) into the left-hand side (LHS) of (4.23), we get

$$\begin{aligned} \mathrm{LHS} =& (1-\mu)^2 E\left[e(n)e^*(n)\right] - (1-\mu)E\left[e(n)v^*(n)\right] \\ &- (1-\mu)E\left[v(n)e^*(n)\right] + E\left[v(n)v^*(n)\right]. \end{aligned} \tag{4.24}$$

Substituting (4.17) into the right-hand side (RHS) of (4.23), we obtain

$$\begin{aligned} \mathrm{RHS} =& E[e(n)e^*(n)] - E[e(n)v^*(n)] - E[v(n)e^*(n)] + E[v(n)v^*(n)] \\ &+ \alpha^2 E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(M\sigma_u^2\mathbf{I} - \mathbf{R}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.25}$$

Combining (4.24) and (4.25), we obtain

$$\begin{aligned} E[e(n)e^*(n)] =& \frac{1}{2-\mu} E[e(n)v^*(n) + v(n)e^*(n)] \\ & - \frac{\alpha^2}{2\mu-\mu^2} E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(M\sigma_u^2\mathbf{I}-\mathbf{R}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.26}$$

With the assumption that the noise $\mathbf{v}(n)$ is statistically independent of the input signal $\mathbf{u}(n)$, i.e., $E[\mathbf{u}(n)v(n)] = 0$, and substituting (4.4) into (4.26), we obtain

$$E|e(n)|^2 = \frac{2}{2-\mu} E|v(n)|^2 - \frac{\alpha^2}{2\mu-\mu^2} E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(M\sigma_u^2\mathbf{I}-\mathbf{R}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \tag{4.27}$$

With the assumption (4.7), then we can arrive at

$$\begin{aligned} \mathrm{MSE} &= \lim_{n\to\infty} E|e(n)|^2 \\ &= \frac{2}{2-\mu}\sigma_v^2 - \frac{\alpha^2}{2\mu-\mu^2} E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\left(M\sigma_u^2\mathbf{I}-\mathbf{R}\right)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \\ &= \frac{2}{2-\mu}\sigma_v^2 - \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\left(M\sigma_u^2\mathbf{I}-\mathbf{R}\right)\mathrm{sgn}(\hat{\mathbf{w}}_0), \end{aligned} \tag{4.28}$$

where σ_v^2 is the power of the observation noise $v(n)$. Finally, the excess MSE can be written as

$$\begin{aligned} \mathrm{EMSE} &= \mathrm{MSE} - \sigma_v^2 \\ &= \frac{\mu}{2-\mu}\sigma_v^2 - \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\left(M\sigma_u^2\mathbf{I}-\mathbf{R}\right)\mathrm{sgn}(\hat{\mathbf{w}}_0). \end{aligned} \tag{4.29}$$

Note that the first part of the expression for the EMSE is a function proportional to $\frac{\mu}{2-\mu}$. In Section 4.3, simulation results will evaluate the expressions obtained for the EMSE in detail.

4.2.2 ZA-APA

The ZA-APA is an improvement of the conventional APA. We can also use the energy-conservation approach [30, 48–50] to derive the theoretical expressions of the EMSE of the ZA-APA. Firstly, multiplying both sides of (3.11) by $\mathbf{U}^H(n)$ from the left, we can find

$$\begin{aligned}\mathbf{U}^H(n)\hat{\mathbf{w}}(n+1) =&\mathbf{U}^H(n)\hat{\mathbf{w}}(n) + \mu\,\mathbf{U}^H(n)\mathbf{U}^+(n)\mathbf{e}(n)\\ &+ \alpha\,\mathbf{U}^H(n)\mathbf{U}^+(n)\mathbf{U}^H(n)\text{sgn}[\hat{\mathbf{w}}(n)] - \alpha\mathbf{U}^H(n)\text{sgn}[\hat{\mathbf{w}}(n)]\\ \mathbf{U}^H(n)\hat{\mathbf{w}}(n+1) =&\mathbf{U}^H(n)\hat{\mathbf{w}}(n) + \mu\,\mathbf{e}(n),\end{aligned} \tag{4.30}$$

where we used the fact that, from the definition of $\mathbf{U}^H(n)\mathbf{U}^+(n) = \mathbf{I}$.

Introducing the a posteriori and a priori error vectors $\mathbf{e}_p(n)$ and $\mathbf{e}_a(n)$, we have

$$\begin{aligned}\mathbf{e}_p(n) &= \mathbf{U}^H(n)\hat{\mathbf{w}}_0 - \mathbf{U}^H(n)\hat{\mathbf{w}}(n+1),\\ \mathbf{e}_a(n) &= \mathbf{U}^H(n)\hat{\mathbf{w}}_0 - \mathbf{U}^H(n)\hat{\mathbf{w}}(n).\end{aligned} \tag{4.31}$$

Then, from (4.30), it holds that

$$-\mathbf{e}_p(n) = -\mathbf{e}_a(n) + \mu\mathbf{e}(n). \tag{4.32}$$

In addition

$$\begin{aligned}\mathbf{e}(n) =&\mathbf{d}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n)\\ =&\mathbf{U}^H(n)\hat{\mathbf{w}}_0 + \mathbf{v}(n) - \mathbf{U}^H(n)\hat{\mathbf{w}}(n)\\ =&\mathbf{e}_a(n) + \mathbf{v}(n).\end{aligned} \tag{4.33}$$

Then (4.32) can be rewritten as

$$\begin{aligned}\mathbf{e}_p(n) =&\mathbf{e}_a(n) - \mu\mathbf{e}(n)\\ =&(1-\mu)\mathbf{e}(n) - \mathbf{v}(n).\end{aligned} \tag{4.34}$$

We can also use (4.32) to express $e(n)$ as follows,

$$\mathbf{e}(n) = \frac{1}{\mu}[\mathbf{e}_a(n) - \mathbf{e}_p(n)]. \tag{4.35}$$

Substituting the above into (3.11), we get

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{U}^+(n)[\mathbf{e}_a(n) - \mathbf{e}_p(n)] + \alpha\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)] - \alpha\mathrm{sgn}[\hat{\mathbf{w}}(n)]. \tag{4.36}$$

Note that $\mathbf{U}^H(n)\mathbf{U}^+(n) = \mathbf{U}^H(n)\mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1} = \mathbf{I}$. By evaluating the energies of both sides of the above equation, and using the steady-state condition $E[\|\hat{\mathbf{w}}(n+1)\|^2] \approx E[\|\hat{\mathbf{w}}(n)\|^2]$ when $n \to \infty$, we obtain

$$\begin{aligned} E\left\{\mathbf{e}_p^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}_p(n)\right\} =& E\left\{\mathbf{e}_a^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}_a(n)\right\} \\ &- \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ &+ \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.37}$$

Substituting (4.34) into the LHS of (4.37), we get

$$\begin{aligned} \text{LHS} =& (1-\mu)^2 E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} \\ &- (1-\mu) E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\} \\ &- (1-\mu) E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} \\ &+ E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\}. \end{aligned} \tag{4.38}$$

Moreover, substituting (4.33) into the RHS of (4.37), we obtain

$$\begin{aligned} \text{RHS} =& E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} \\ &- E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\} \\ &- E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} \\ &+ E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\} \\ &- \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\} \\ &+ \alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}. \end{aligned} \tag{4.39}$$

Combining (4.38) and (4.39) , we obtain

$$\begin{aligned}(2\mu-\mu^2)E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} =&\mu\, E\left[\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n))\right\}\\ &+\mu\, E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\}\\ &+\alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}\\ &-\alpha^2\, E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}.\end{aligned} \tag{4.40}$$

With the assumption that the noise $\mathbf{v}(n)$ is statistically independent of the input signal $\mathbf{u}(n)$, we obtain

$$\begin{aligned}E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} =&\frac{2}{2-\mu}E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\}\\ &+\frac{\alpha^2}{2\mu-\mu^2}\,E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\,\mathbf{U}^+(n)\mathbf{U}^H(n)\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}\\ &-\frac{\alpha^2}{2\mu-\mu^2}\,E\left\{\mathrm{sgn}^H[\hat{\mathbf{w}}(n)]\mathrm{sgn}[\hat{\mathbf{w}}(n)]\right\}.\end{aligned} \tag{4.41}$$

Ref. [48] suggests that we can assume that at steady-state, $\mathbf{U}(n)$ is statistically independent of $\mathbf{e}(n)$, and moreover, $E[\mathbf{e}(n)\mathbf{e}^H(n)] = E|e_{top}(n)|^2\,\mathbf{S}$, where $\mathbf{S}\approx\mathbf{I}$ when μ is small and $\mathbf{S}\approx\mathbf{1}\cdot\mathbf{1}^T$ for large μ, where $\mathbf{1}^T = [1\ 0\ ...\ 0]$ and $e_{top}(n)$ is the top entry of $\mathbf{e}(n)$. Then we obtain

$$\begin{aligned}E\left\{\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{e}(n)\right\} \approx&\mathrm{Tr}\left\{E[\mathbf{e}(n)\mathbf{e}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}]\right\}\\ \approx&E|e_{top}(n)|^2\mathrm{Tr}\left\{\mathbf{S}\cdot E\left[[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right]\right\}.\end{aligned} \tag{4.42}$$

Note that the MSE at time instant n equals $E|e_{top}(n)|^2$.

Similar manipulations can be applied to the first terms in the RHS of (4.41), and we obtain

$$\begin{aligned}E\left\{\mathbf{v}^H(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{v}(n)\right\} \approx&\mathrm{Tr}\left\{E[\mathbf{v}^H(n)\mathbf{v}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}]\right\}\\ \approx&N\sigma_v^2\mathrm{Tr}\left\{E\left[[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right]\right\}.\end{aligned} \tag{4.43}$$

In addition, the remaining term in (4.41) can also be rewritten as

$$E\left\{\text{sgn}^H[\hat{\mathbf{w}}(n)]\,[\mathbf{U}^+(n)\mathbf{U}(n)]\text{sgn}[\hat{\mathbf{w}}(n)]\right\} \approx \text{sgn}^H(\hat{\mathbf{w}}_0)E[\mathbf{U}^+(n)\mathbf{U}^H(n)]\text{sgn}(\hat{\mathbf{w}}_0) - \text{sgn}^H(\hat{\mathbf{w}}_0)\text{sgn}(\hat{\mathbf{w}}_0). \tag{4.44}$$

In order to find an approximation for this expression, we need to restrict the analysis to the case of uncorrelated input, i.e., we assume now that $\mathbf{R} = \sigma_u^2\mathbf{I}_M$, where $\mathbf{I}_M$ is the $M \times M$ identity matrix. In this case, for large M we have

$$\begin{aligned} E[\mathbf{U}^+(n)\mathbf{U}^H(n)] &= E\left\{\mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{U}^H(n)\right\} \\ &\approx E\left\{\mathbf{U}(n)(E[\mathbf{U}^H(n)\mathbf{U}(n)])^{-1}\mathbf{U}^H(n)\right\}. \end{aligned} \tag{4.45}$$

Since $E[\mathbf{u}^H(n)\mathbf{u}(n-1)] = 0$, the inner expectation reduces to

$$\begin{aligned} E[\mathbf{U}^H(n)\mathbf{U}(n)] &= \begin{bmatrix} \mathbf{u}^H(n) \\ \mathbf{u}^H(n-1) \\ \vdots \\ \mathbf{u}^H(n-N+1) \end{bmatrix} [\mathbf{u}(n)\ \mathbf{u}(n-1)\ \ldots\ \mathbf{u}(n-N+1)] \\ &= \begin{bmatrix} \|\mathbf{u}(n)\|^2 & \mathbf{u}^H(n)\mathbf{u}(n-1) & \ldots & \mathbf{u}^H(n)\mathbf{u}(n-N+1) \\ \mathbf{u}^H(n-1)\mathbf{u}(n) & \|\mathbf{u}(n-1)\|^2 & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}^H(n-N+1)\mathbf{u}(n) & \ldots & \ldots & \|\mathbf{u}(n-N+1)\|^2 \end{bmatrix} \\ &= \begin{bmatrix} \|\mathbf{u}(n)\|^2 & 0 & \ldots & 0 \\ 0 & \|\mathbf{u}(n-1)\|^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \|\mathbf{u}(n-N+1)\|^2 \end{bmatrix}, \end{aligned} \tag{4.46}$$

with $\mathbf{R} = \sigma_u^2\,\mathbf{I}_M$, (4.46) equals

$$E[\mathbf{U}^H(n)\mathbf{U}(n)] \approx \text{Tr}(\mathbf{R})\,\mathbf{I}_N = M\,\sigma_u^2\,\mathbf{I}_N, \tag{4.47}$$

where $\mathbf{I}_N$ is the $N \times N$ identity matrix.

Moreover, we also assume that

$$E\left\{[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right\} \approx E\left[\mathbf{U}^H(n)\mathbf{U}(n)\right]^{-1} = \frac{1}{M\,\sigma_u^2}\,\mathbf{I}_N. \tag{4.48}$$

Assuming that M is large, we can approximate $E[\mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{U}^H(n)]$ by

$$\begin{aligned} E\left\{\mathbf{U}(n)[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\mathbf{U}^H(n)\right\} &\approx E\left\{\mathbf{U}(n)\frac{1}{M\,\sigma_u^2}\,\mathbf{I}_N\mathbf{U}^H(n)\right\} \\ &\approx \frac{N\,\mathbf{R}}{M\sigma_u^2}. \end{aligned} \tag{4.49}$$

When μ is small, $\mathbf{S} \approx \mathbf{I}$. Combine with (4.42), we obtain

$$\begin{aligned} \mathrm{Tr}\left\{\mathbf{S}\cdot E[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right\} &= \mathrm{Tr}\left\{\mathbf{I}\cdot E[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right\} \\ &= \frac{N}{M\sigma_u^2}. \end{aligned} \tag{4.50}$$

So the MSE of ZA-APA with small step-size can be written as

$$\begin{aligned} \mathrm{MSE} =& \frac{2}{2-\mu}\sigma_v^2 + \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\frac{M\sigma_u^2}{N}\cdot\frac{N\,\mathbf{R}}{M\sigma_u^2}\mathrm{sgn}(\hat{\mathbf{w}}_0) \\ & - \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\frac{M\sigma_u^2}{N}\mathrm{sgn}(\hat{\mathbf{w}}_0) \\ =& \frac{2}{2-\mu}\sigma_v^2 + \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\left(\mathbf{R}-\frac{M\sigma_u^2}{N}\mathbf{I}\right)\mathrm{sgn}(\hat{\mathbf{w}}_0). \end{aligned} \tag{4.51}$$

Then we can also obtain the EMSE of ZA-APA with small step-size μ

$$\mathrm{EMSE} = \frac{\mu}{2-\mu}\sigma_v^2 + \frac{\alpha^2}{2\mu-\mu^2}\mathrm{sgn}^H(\hat{\mathbf{w}}_0)\left(\mathbf{R}-\frac{M\sigma_u^2}{N}\mathbf{I}\right)\mathrm{sgn}(\hat{\mathbf{w}}_0). \tag{4.52}$$

Moreover, when μ is large, $\mathbf{S} \approx \mathbf{1}\cdot\mathbf{1}^T$. In this case, we have [48]

$$\mathrm{Tr}\left\{\mathbf{S}\cdot E[\mathbf{U}^H(n)\mathbf{U}(n)]^{-1}\right\} = \frac{1}{M\sigma_u^2}. \tag{4.53}$$

Then the MSE of ZA-APA with large step-size can be written as

$$\text{MSE} = \frac{2}{2-\mu}\sigma_v^2 N + \frac{\alpha^2}{2\mu-\mu^2}\text{sgn}^H(\hat{\mathbf{w}}_0)\left(N\mathbf{R} - M\sigma_u^2\mathbf{I}\right)\text{sgn}(\hat{\mathbf{w}}_0). \tag{4.54}$$

Finally, the excess MSE can be written as

$$\text{EMSE} = \frac{\mu}{2-\mu}\sigma_v^2 N + \frac{\alpha^2}{2\mu-\mu^2}\text{sgn}^H(\hat{\mathbf{w}}_0)\left(N\mathbf{R} - M\sigma_u^2\mathbf{I}\right)\text{sgn}(\hat{\mathbf{w}}_0). \tag{4.55}$$

In section 4.3, simulation results will evaluate the expressions obtained for the EMSE in detail.

4.3 Simulation Results and Analysis

In this section, simulation results are given to show the steady-state properties of the proposed sparsity-aware algorithms. In these experiments, all the unknown sparse systems have 16 taps with only 2 non-zero taps.

4.3.1 Simulation and Analysis for ZA-NLMS Algorithm

Firstly, we present two simulation examples to compare the steady-state MSE of the ZA-NLMS algorithm with different step-sizes μ. The zero-attractor parameter we choose is $\alpha = 10^{-3}$, and the signal-to-noise ratio (SNR) is set to 30dB. Two different types of signals, viz., independent and identically distributed (i.i.d.) and uniformly distributed signals, are used for the input signal, viz.,

$$u(n) = \tau u(n-1) + \rho(n), \tag{4.56}$$

which is a first-order autoregressive (AR) process with a pole at τ. For the i.i.d. case, $\rho(n)$ is a white, zero-mean, Gaussian random sequence having unit variance, and is τ set to 0. As a result, an i.i.d. Gaussian input signal is generated. For the uniform case, $\rho(n)$ is a uniform random sequence between -1.0 and 1.0, and τ is set to 0.5. In addition, the measurement noise here are i.i.d Gaussian white noise as well. Moreover, in these figures, the optimal line stands for the power of the noise.

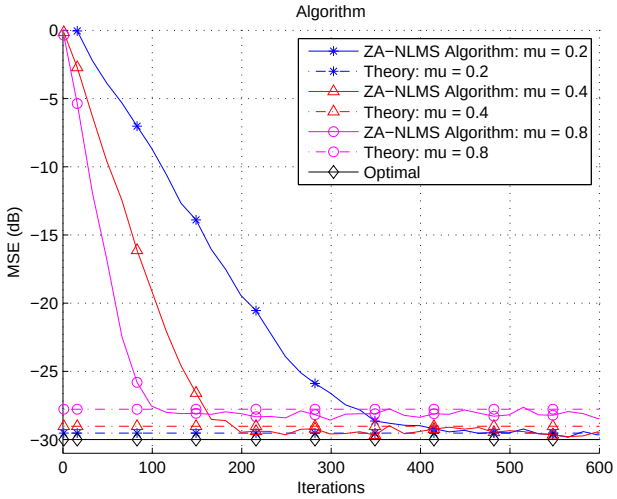

Figure 4.3: Steady-state MSE curve of the ZA-NLMS alogorithm with lower step-sizes μ [$\alpha = 10^{-3}$, SNR=30dB, Input: i.i.d. Gaussian signal].

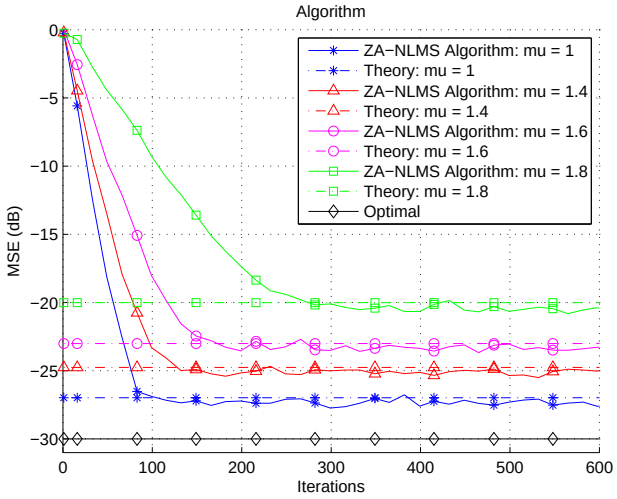

Figure 4.4: Steady-state MSE curve of the ZA-NLMS alogorithm with larger step-sizes μ [$\alpha = 10^{-3}$, SNR=30dB, Input: i.i.d. Gaussian signal].

Fig. 4.3 and 4.4 show the steady-state MSE curves of the ZA-NLMS algorithm as a function of the step-size for i.i.d. Gaussian input signal. The step-size μ varies from 0.2 to 1.9. The theoretical results are calculated using (4.29), and

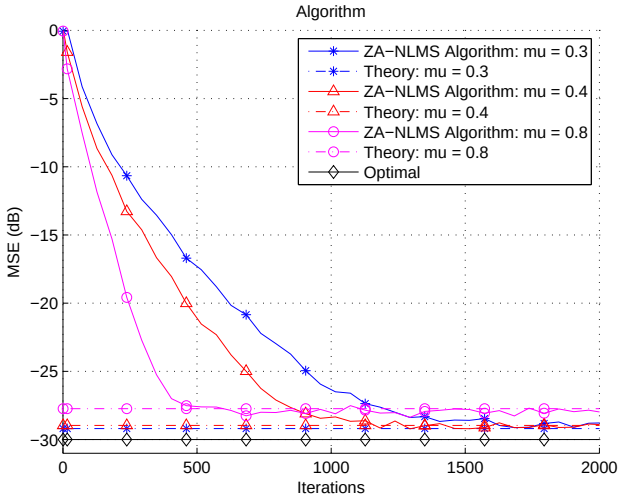

Figure 4.5: Steady-state MSE curve of the ZA-NLMS alogorithm with lower step-sizes μ [$\alpha = 10^{-3}$, SNR=30dB, Input: uniform AR(1)].

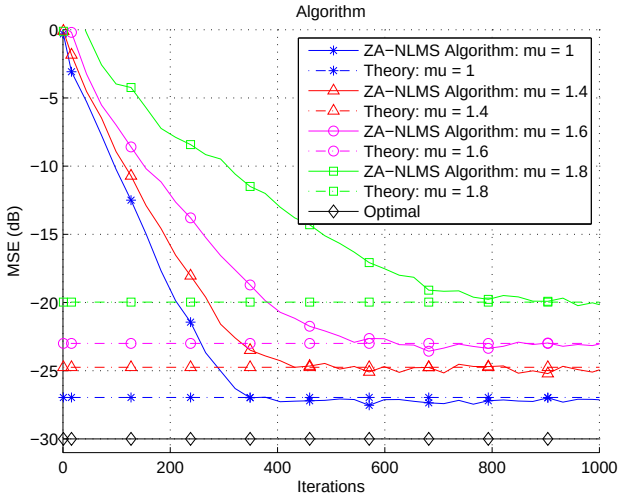

Figure 4.6: Steady-state MSE curve of the ZA-NLMS alogorithm with larger step-sizes μ [$\alpha = 10^{-3}$, SNR=30dB, Input: uniform AR(1)].

the simulation results are obtained by averaging 1000 independent trials. The simulated results present good agreement with the theoretical results for different step-sizes. We can easily see from the figures that larger step-sizes lead to larger

misadjustment, but faster convergence. So in most cases we will choose a step-size to balance the steady-state performance and the convergence rate. In addition, from Fig. 4.5 and 4.6, for colored uniform input signal, we theoretical results do not match the simulated ones accurately. However, for colored uniform input signals, μ have the same effect as for i.i.d. Gaussian input signal.

Furthermore, we also make a simulation to learn how the zero-attractor α influences the EMSE. In the experiment, we choose a fixed step-size $\mu = 0.5$.

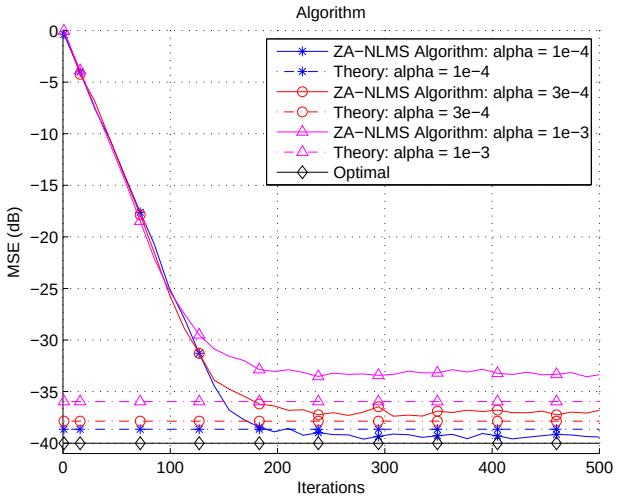

Figure 4.7: Steady-state MSE curve of the ZA-NLMS algorithm with different zero-attractor parameters α [$\mu = 0.5$, SNR=40dB, Input: i.i.d. Gaussian signal].

From Fig. 4.7 and 4.8, we can see that the simulation results present a perfect match with the theoretical results for small zero-attractor α, but deviates from the theory for a large zero-attractor. Generally speaking, as α increases, the MSE also increases. Although a higher zero-attractor leads to a higher misadjustment, it can also help the algorithm to converge faster at the initial period. A wisely chosen α can improve the steady-state performance and convergence rate.

4.3.2 Simulation and Analysis for the ZA-APA

The following experiments are made to analyze the EMSE of the ZA-APA.

In the first 4 experiments, the steady-state MSE curves of the ZA-APA are

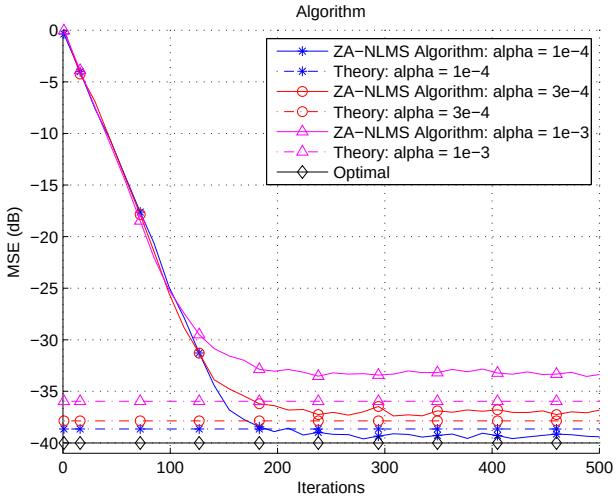

Figure 4.8: Steady-state MSE curve of the ZA-NLMS algorithm with different zero-attractor parameters α [$\mu = 0.5$, SNR=40dB, Input: uniform AR(1)].

shown as a function of the step-size μ. The step-size we test varies from 0.2 to 1.8. we use (4.52) to predict the theoretical MSE, when μ varies from 0.2 to 0.8. Moreover (4.55) is used when μ varies from 1 to 1.8.

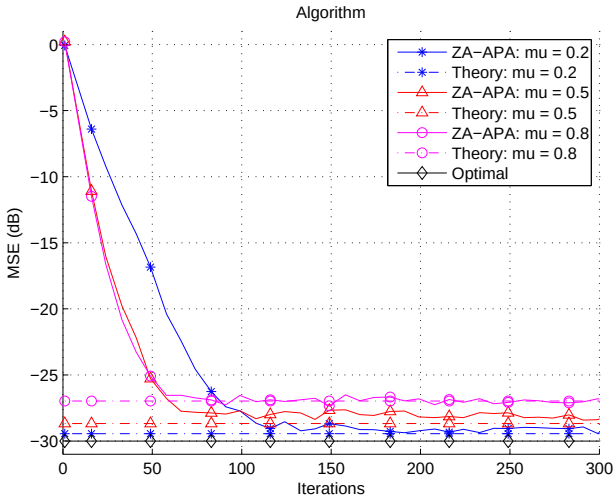

Figure 4.9: Steady-state MSE curve of the ZA-APA with different step-sizes μ [$\alpha = 10^{-3}$, $K = 4$, SNR=30dB, Input: i.i.d. Gaussian signal].

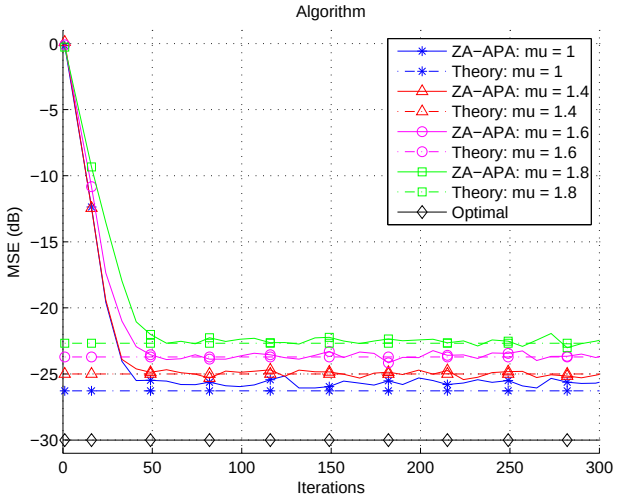

Figure 4.10: Steady-state MSE curve of the ZA-APA with different step-sizes μ [$\alpha = 10^{-3}$, $K = 4$, SNR=30dB, Input: i.i.d. Gaussian signal].

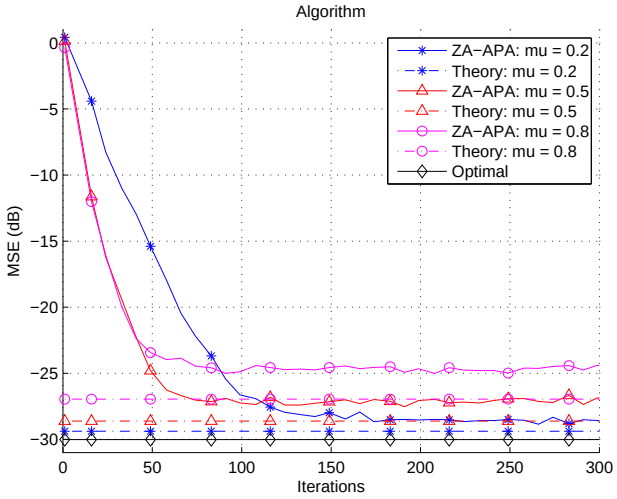

Figure 4.11: Steady-state MSE curve of the ZA-APA with different step-sizes μ [$\alpha = 10^{-3}$, $K = 4$, SNR=30dB, Input: uniform AR(1)].

As shown in Fig. 4.9 and 4.10, the curves of the steady-state MSE match the theoretical ones with i.i.d. Gaussian input signal. We can also easily see that as the step-size μ increases, the misadjustment increases. However, increasing the

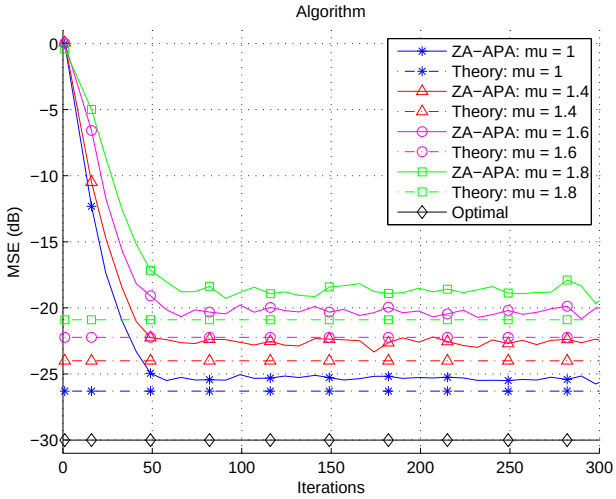

Figure 4.12: Steady-state MSE curve of the ZA-APA with different step-sizes μ [$\alpha = 10^{-3}$, $K = 4$, SNR=30dB, Input: uniform AR(1)].

step-sizes can also lead to faster convergence rate during the period when the step-size increases from 0.2 to 1. From 4.10, the curves of the steady-state MSE also match the theoretical ones well. In addition, a larger step-size μ implies a higher misadjustment. But differently from Fig. 4.9, we can see that the convergence rate does not improve with the increase of the step-size. In contrast, the convergence rate becomes slower and the steady-state MSE becomes higher during the period when the step-size increases from 1 to 1.8. In Fig. 4.11 and Fig. 4.12, uniform input the signal is used for the simulation. In these figures, the theoretical results do not accurately match the simulated ones accurately. However, μ shows the same effect as for i.i.d. Gaussian input signal.

The second experiment's objective is to analyze the effects of the affine projection order N. We can easily see from Fig. 4.13 and 4.13 that a higher N leads to a faster convergence rate. However, increasing N can also bring a higher misadjustment.

We also make a simulation to learn how the zero-attractor parameter α influences the EMSE. In the experiment, we choose the step-size $\mu = 0.5$ and affine projection order $K = 4$.

From Fig. 4.15 and 4.16, we can see that the simulation results present a

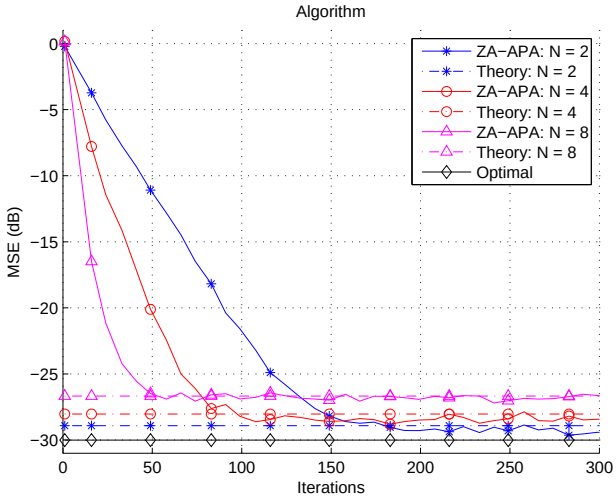

Figure 4.13: Steady-state MSE curve of the ZA-APA with different affine projection order N [$\mu = 0.5$, $\alpha = 10^{-3}$, SNR=30dB, Input: i.i.d. Gaussian signal].

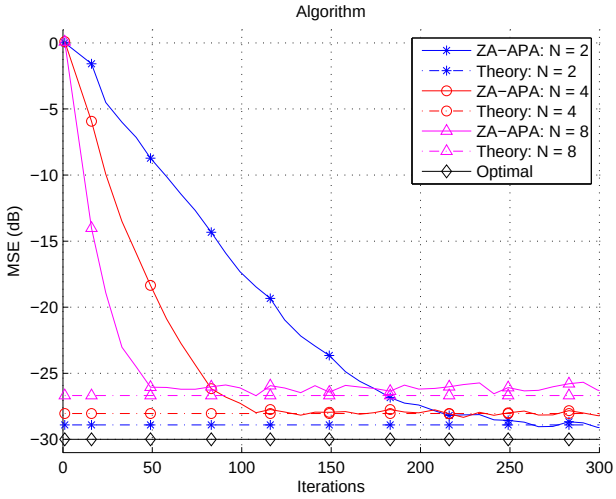

Figure 4.14: Steady-state MSE curve of the ZA-APA with different affine projection order N [$\mu = 0.5$, $\alpha = 10^{-3}$, SNR=30dB, Input: i.i.d. Gaussian signal].

perfect match with the theoretical results. Generally speaking, as α increases, the MSE also increases. Although higher zero-attractor leads to higher misadjustment, it can also help the algorithm to converge fast at the initial period. A

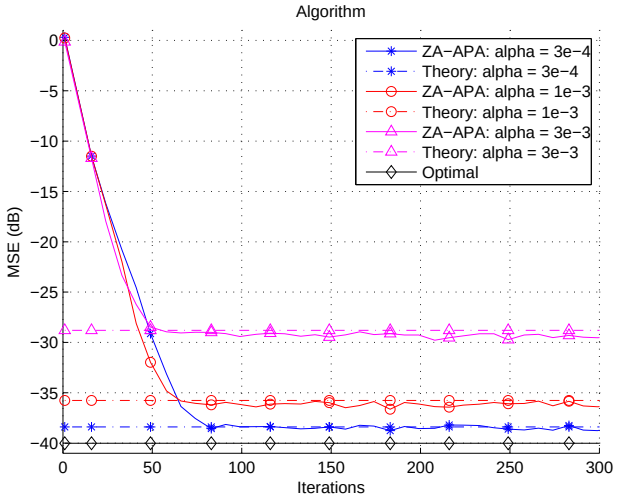

Figure 4.15: Steady-state MSE curve of the ZA-APA with different zero-attractor parameter parameters α [$\mu = 0.5$, $K = 4$, SNR=40dB, Input: i.i.d. Gaussian signal].

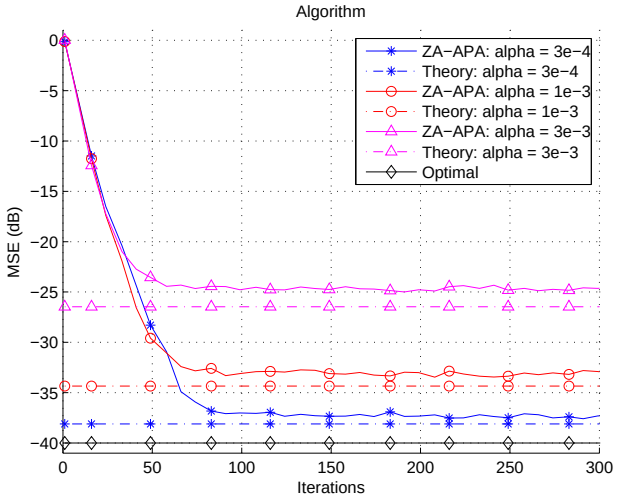

Figure 4.16: Steady-state MSE curve of the ZA-APA with different zero-attractor parameter parameters α [$\mu = 0.5$, $K = 4$, SNR=40dB, Input: uniform AR(1)].

wisely chosen α can improve the steady-state performance and convergence rate.

4.4 Summary

In this chapter, an analysis of the proposed algorithms has been presented, including the computational complexity requirements, a convergence analysis and a steady-state analysis. The theoretical expressions of the EMSE for ZA-NLMS and ZA-APA have been derived by using the energy-conservation approach, and we have also made use of an extensive set of simulations to verify the theory. Simulation results show that the theoretical expressions can predict the steady-state MSE for ZA-NLMS and ZA-APA effectively. Note that we did not perform a tracking analysis in the non-stationary scenarios, which is a topic for future investigation.

Chapter 5

Conclusions and Future Work

5.1 Summary of Work

In this thesis, low-complexity adaptive filtering algorithms that exploit the sparsity of signals and systems have been derived and investigated.

In Chapter 2, a general review of adaptive filtering techniques has been given. Firstly, we have introduced the main objectives of the adaptive filters and then we have discussed some popular applications of adaptive filters that include system identification, echo cancellation and adaptive beamforming. Finally, some of the most commonly used adaptive algorithms have been introduced.

In Chapter 3, a set of sparsity-aware zero-attracting adaptive algorithms have been developed by incorporating the l_1-norm penalty into the coefficients with the conventional NLMS and APA algorithms. Firstly, we have incorporated the l_1-norm optimization strategy with the conventional NLMS algorithm, which resulted in the proposed ZA-NLMS algorithm. Moreover, the same strategy has been applied to the conventional APA algorithm to obtain the proposed sparsity-aware ZA-APA algorithm. However, the ZA-NLMS and ZA-APA algorithms do not distinguish between zero taps and non-zero taps. Since all the taps are forced to zero uniformly, the performance of ZA-APA can be deteriorated when applied to systems with a low degree of sparsity, i.e., when the number of non-zero coefficients is significant. In order to solve this problem, we have adopted a heuristic approach to reinforce the zero attractor and proposed the RZA-NLMS and RZA-APA algorithms. We have also developed a zero-forcing technique to further

improve the performance of the proposed zero-attracting algorithms.

Finally, simulation results have been carried out to compare the proposed algorithms with the conventional ones. As we can see from the results of an extensive set of simulations, the proposed algorithms possess a faster convergence rate and a better steady-state performance.

In Chapter 4, the analysis of the proposed algorithms has been presented in detail, including their computational complexity, the convergence analysis and the steady-state analysis. By using the energy-conservation approach, we have derived the theoretical expressions of the EMSE of the ZA-NLMS and the ZA-APA algorithms. We have also carried out a large number of simulations to verify the theoretical expressions derived. Moreover, we have introduced a set of simulation results to analyze the effects of the most important parameters used in the ZA-APA algorithms.

5.2 Future Work

In this thesis, the tracking performance analysis of the proposed sparsity-aware algorithms has not been performed yet. Therefore, one of the possible future works could be the development of the tracking analysis of the proposed algorithms. Moreover, another area for investigation is the application of the proposed in other scenarios different from system identification, such as adaptive beamforming and channel equalization. Furthermore, the zero-attractor strategy considered for the proposed algorithms has been studied analytically, however, an analytical study of the reweighted zero-attractor and other similar strategies remains an open problem. Another possible future work could be to introduce a self-adaptive selection parameter to control the zero-attracting weight, by which the performance is expected to be further improved.

References

[1] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, New Jersey, fourth edition, 2002. 1, 5, 8, 10, 12, 14, 15

[2] Ali H. Sayed. *Fundamentals of Adaptive Filtering*. John Wiley & Sons, New Jersey, 2003. 1, 2, 11, 16

[3] Harry L. Van Trees. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. John Wiley & Sons, New Jersey, 2002. 1

[4] Alan V. Oppenheim. *Digital Signal Processing*. Prentice-Hall, New Jersey, 1975. 1, 6

[5] Paulo S. R. Diniz. *Adaptive Filtering: Algorithms and Practical Implementations*. Springer, Boston, second edition, 2002. 1, 2, 6, 7, 12, 14

[6] Paulo S. R. Diniz, E. A. B. da Silva, and S. Lima Netto. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, Cambridge, 2010. 1, 2, 12

[7] K. Pelekanakis and M. Chitre. Comparison of sparse adaptive filters for underwater acoustic channel equalization/estimation. *Proc. IEEE International Conference on Communication Systems (ICCS)*, 10:395–399, 2010. 2

[8] Y. Chen, Y. Gu, and A. Hero. Sparse LMS for system identification. *Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing*, pages 3125–3128, April 2009. 2, 18, 23, 39

[9] Y. Chen, Y. Gu, and A. Hero. Regularized least-mean-square algorithms. *IEEE Transactions on Signal Processing*, 2010. 2, 18, 23

[10] J. D. Gordy and R. A. Goubran. Fast system identification using affine projection and a critically sampled subband adaptive filter. *IEEE Transactions on Instrumentation and Measurement Society*, 55:1242–1249, July 2006. 2, 18

[11] F. das Chagas de Souza, O.J. Tobias, and R. Seara. A PNLMS algorithm with individual activation factors. *IEEE Transactions on Signal Processing*, 58(4):2036–2047, April 2010. 2, 15

[12] D. L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Transactions Speech Audio Process*, 8(5):508C518, September 2000. 2, 15

[13] D. L. Duttweiler. Comparison of sparse adaptive filters for underwater acoustic channel equalization/estimation. *Proc. IEEE International Conference on Communication Systems (ICCS)*, pages 395–399, November 2010. 2, 15

[14] D. L. Duttweiler. Improving convergence of the PNLMS algorithm for sparse impulse response identification. *IEEE Signal Processing Letters*, 12(3):181–184, March 2005. 2, 15

[15] Z. Yang, R. Zheng, and S. Grant. Proportionate affine projection sign algorithms for network echo cancellation. *IEEE Transactions on Audio, Speech and Language Processing*, 19:2273–2284, November 2011. 2, 10, 11

[16] R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, July 2007. 2

[17] M. Zibulevsky and M. Elad. $l_1 - l_2$ optimization in signal and image processing. *IEEE Signal Processing Magazine*, 27(3):76–88, May 2010. 2

[18] M. Elad. Why simple shrinkage is still relevant for redundant representations. *IEEE Transactions on Information Theory*, 52(12):5559–5569, December 2006. 2

[19] D. Angelosante, J. A. Bazerque, and G. B. Giannakis. Online adaptive estimation of sparse signals: where RLS meets the l_1-norm. *IEEE Transactions on Signal Processing*, 58(7):3436–3446, July 2010. 2

[20] J. Friedman, T. hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, January 2010. 2

[21] J. Jin, Y. Gu, and S. Mei. A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):409–420, April 2010. 2

[22] E. M. Eksioglu. RLS adaptive filtering with sparsity regularization. *10th International Conference on Information Science, Signal Processing and their Applications*, pages 550–553, May 2010. 2

[23] D. Angelosante and G. B. Giannakis. RLS-weighted lasso for adaptive estimation of sparse signals. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3245–3248, 2009. 2

[24] R. C. de Lamare Z. Yang and X. Li. l_1 regularized STAP algorithm with a generalized sidelobe canceler architecture for airborne radar. *Proc. IEEE Workshop on Statistical Signal Processing*, pages 329–332, June 2011. 2, 11

[25] Kun Qiu and Aleksandar Dogandzic. Variance-component based sparse signal reconstruction and model selection. *IEEE Transactions on Signal Processing*, 58(6):2935–2952, June 2010. 2

[26] E. J. Candes, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 55(5). 2, 23

[27] Alessandro J. S. Dutra, Lisandro Lovisolo, Eduardo A. B. da Silva, and Paulo S. R. Diniz. Successive approximation FIR filter design. *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 149–152, May 2011. 6

[28] J. Chambers and A. Avlonitis. A robust mixed-norm adaptive filter algorithm. *IEEE Signal Processing Letters*, 4(2), July 1997. 8

[29] O. Tanrikulu J. Chambers and A. G. Constantinides. Least mean mixed-norm adaptive filtering. *Electronics Letters*, 30:1574C1575, July 1994. 8

[30] ZenglinYang, Yahong Rosa Zheng, and Steven L. Grant. Proportionate affine projection sign algorithms for network echo cancellation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2273–2284, November 2011. 10, 16, 36, 43

[31] T. Gansler, S. L. Gay, J. Benesty, and M. M. Sondhi. A robust proportionate affine projection algorithm for network echo cancellation. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2:793–796, June 2000. 10, 17

[32] Z. Yang, R. C. de Lamare, and X. Li. l_1 regularized STAP algorithm with a generalized sidelobe canceler architecture for airborne radar. 2012. 11

[33] L. Wang and R. C. de Lamare. Constrained adaptive filtering algorithms based on conjugate gradient techniques for beamforming. *IET Signal Processing*, 4(6):686–697, December 2010. 11

[34] L. Wang and R. C. de Lamare. Adaptive reduced-rank LCMV beamforming algorithms based on joint iterative optimisation of filters. *Electronics Letters*, 44(9):565–566, April 2008. 11

[35] L. Wang and R. C. de Lamare. Adaptive reduced-rank least squares beamforming algorithm based on the set-membership framework. *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 1935–1939, November 2010. 11

[36] R. C. de Lamare and R. Sampaio-Neto. Low-complexity variable step-size mechanisms for stochastic gradient algorithms in minimum variance CDMA receivers. *IEEE Transactions on Signal Processing*, 54(6):2302–2317, June 2006. 12

[37] R. C. de Lamare, L. Wang, and R. Fa. Adaptive reduced-rank LCMV beamforming algorithms based on joint iterative optimization of filters: Design and analysis. *Elsevier Signal Processing*, 90:640–652, February 2010. 12

[38] R. C. de Lamare and R. Sampaio-Neto. Reduced-rank adaptive filtering based on joint iterative optimization of adaptive filters. *IEEE Signal Processing Letters*, 14(12):980–983, December 2007. 12

[39] Hyun-Chool Shin, Ali H. Sayed, and Woo-Jin Song. Variable step-size NLMS and affine projection algorithms. *IEEE Signal Processing Letters*, 11(2):132–135, February 2004. 14

[40] Stefan Werner, Marcello L. R. de Campos, and Paulo S. R. Diniz. Partial-update NLMS algorithms with data-selective updating. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2:1413–1416, May 2002. 14

[41] R. C. de Lamare and R. Sampaio-Neto. Adaptive reduced-rank MMSE filtering with interpolated FIR filters and adaptive interpolators. *IEEE Sig. Proc. Letters*, 12(3):177–180, 2005. 16

[42] R. C. de Lamare and P. S. R. Diniz. Set-membership adaptive algorithms based on time-varying error bounds for CDMA interference suppression. *IEEE Trans. Vehicular Technology*, 58:644–654, February 2009. 16

[43] R. C. de Lamare and P. S. R. Diniz. Set-membership adaptive algorithms based on time-varying error bounds for DS-CDMA systems. *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2006. 16

[44] R. C. de Lamare and R. Sampaio-Neto. Adaptive reduced-rank processing based on joint and iterative interpolation, decimation, and filtering. *IEEE Transactions on Signal Processing*, 57(7):2503–2514, July 2009. 16

[45] G. Rombouts and M. Moonen. A sparse block exact affine projection algorithm. *IEEE Transactions on Speech and Audio Processing*, 10:100–108, July 2002. 16

[46] T. Kailath J. M. Cioffi. Windowed fast transversal filters adaptive algorithms with normalization. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 33(3), June 1985. 17

[47] S. L. Gay and S. Tavathia. The fast affine projection algorithm. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5:3023–3026, May 1995. 17

[48] Hyun-Chool Shin and Ali H. Sayed. Mean-square performance of a family of affine projection algorithms. *IEEE Transactions on Signal Processing*, 52(1):90–102, January 2004. 36, 43, 45, 47

[49] Hyun-Chool Shin, Woo-Jin Song, and Ali H. Sayed. Mean-square performance of data-reusing adaptive algorithms. *IEEE Signal Processing Letters*, 12(12):851–854, December 2005. 36, 43

[50] Paulo S. R. Diniz. Convergence performance of the simplified set-membership affine projection algorithm. *Circuits, Systems and Signal Processing*, 30:439–462, April 2011. 36, 43

[51] G. Barrault, M. H. Costa, J. C. M. Bermudez, and A. Lenzi. A new analytical model for the NLMS algorithm. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 4:41–44, 2005. 38

[52] M. H. Costa and J. C. M. Bermudez. An improved model for the normalized LMS algorithm with Gaussian inputs and large number of coefficients. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2:1385–1388, May 2002. 38