UNIVERSITY OF BLIDA 1

Faculty of Technology

Department of Electronics

MAGISTER REPORT

in Electronics

Option : Signals and Systems

STUDY OF SPARSE ADAPTIVE ALGORITHMS FOR THE IDENTIFICATION OF ACOUSTIC IMPULSE RESPONSES

by

Ayoub TEDJANI

Before the jury composed of :

M. BAHOURA	Professor	U. of Blida 1	President
M. DJENDI	Professor	U. of Blida 1	Examiner
A. RECIOUI	MCA	IGEE of Boumerdès	Examiner
A. BENALLAL	Professor	U. of Blida 1	Supervisor

Blida, May 15th, 2016.

UNIVERSITÉ DE BLIDA 1

Faculté des Technologie

Département d'Électronique

MÉMOIRE DE MAGISTER

en Électronique

Option : Signaux et Systèmes

ÉTUDE DES ALGORITHMES ADAPTATIFS PARCIMONIEUX POUR L'IDENTIFICATION DES RÉPONSES IMPULSIONNELLES ACOUSTIQUES

par

Ayoub TEDJANI

Devant le jury composé de :

M. BAHOURA	Professeur	U. de Blida 1	Président
M. DJENDI	Professeur	U. de Blida 1	Examinateur
A. RECIOUI	MCA	IGEE de Boumerdès	Examinateur
A. BENALLAL	Professeur	U. de Blida 1	Promoteur

Blida, le 15 Mai 2016.

ملخص

يعد تحديد الاستجابات الصوتية النبضية وإلغاء الصدى من المشاكل الشائعة في مجال معالجة الإشارة. في هذا العمل، نقدم أولا لمحة عامة عن المصافي (الفلاتر) المتكيفة و أهم التطورات في مجال الخوارزميات ذات التكيف المقتصد (sparse) ، بدءا من أشهر الأعمال المنجزة على الخوارزمية القياسية المتناسبة لأقل متوسط للمربعات (في اللغة الإنجليزية، :PNLMS Proportionate Normalized Least Mean Square) و أنواعها المختلفة، و ختاما بأحدث التقنيات التي تستخدم مبدأ الاستشعار المضغوط (باللغة الإنجليزية، Compressed Sensing).

بعد ذلك نقوم بسرد التفاصيل النظرية لأهم وأحدث الخوارزميات ذات التكيف المقتصد التي تستند أساسا إلى مبدأ الخوارزمية القياسية لأقل متوسط للمربعات (NLMS) وتستخدم في التصفية المتكيفة، مع القيام بتحليل درجة التعقيد الحسابي لكل منها. الخوارزميات التي تستهدفها هذه الدراسة هي NLMS و أهم إصداراته الكلاسيكية المتناسبة التي تستخدم المعلومة مسبقا عن تناثر النظام (PNLMS، IPNLMS وMPNLMS) بالإضافة إلى الترقيات التي أجريت عليها لتكون مواكبة لتغير درجة تناثر النظام (SC-PNLMS ، SC-IPNLMS وSC-MPNLMS) كذلك بعض الخوارزميات الحديثة التي تقوم على مبدأ الاستشعار المضغوط (Compressed Sensing) مثل ZA-NLMS ، RZA-NLMS، VSS-ZA-NLMS و VSS-RZA-NLMS .

وأخيرا، بمساعدة لغة البرمجة MATLAB® ، نقوم بتنفيذ سلسلة من عمليات المحاكاة لعدة استجابات صوتية نبضية مصطنعة وحقيقية ذات درجات تناثر مختلفة مع إدخالات (inputs) مستقرة وغير مستقرة، وذلك من أجل تحليل، دراسة ومقارنة نقاط القوة و الضعف في الخوارزميات المدروسة من حيث سرعة التقارب، دقة التقدير والتعقيد الحسابي.

كلمات البحث: التصفية المتكيفة، الخوارزميات ذات التكيف المقتصد (sparse) ، الاستجابات الصوتية النبضية، تعريف و تحديد الأنظمة، الاستشعار المضغوط (Compressed Sensing)، الخوارزمية القياسية لأقل متوسط للمربعات (NLMS) ، دقة التقدير و التقريب، درجة التعقيد الحسابي.

RÉSUMÉ

L'identification de réponses impulsionnelles acoustiques (RIAs) et l'annulation d'écho sont des problèmes communs dans le domaine du traitement du signal. Dans ce travail, on présente, premièrement, un aperçu du filtrage adaptatif et les principaux développements dans le domaine des algorithmes parcimonieux, commençant à partir des célèbres ouvrages sur l'algorithme proportionné normalisé de moindres carrés moyens (en anglais, PNLMS : Proportionate Normalized Least Mean Square) et ses nombreuses variantes jusqu'aux techniques les plus récentes qui utilisent le principe de l'acquisition comprimée (en anglais, CS : compressed sensing).

Ensuite, on présente les détails théoriques des algorithmes parcimonieux les plus importants et les plus récents qui sont basés sur l'algorithme NLMS et utilisés pour le filtrage adaptatif. De plus, leurs complexités de calcul sont analysées. Les algorithmes ciblés par cette étude sont le NLMS, ses versions classiques proportionnées qui utilisent l'information, *a priori*, sur la parcimonie de système (PNLMS, IPNLMS et MPNLMS) et leurs mises à jour contrôlés par la variation du degré de parcimonie de système (SC-PNLMS, SC-IPNLMS et SC-MPNLMS) ainsi que certaines algorithmes récentes basés sur le principe de l'acquisition comprimée (CS) à savoir le ZA-NLMS, RZA-NLMS, VSS-ZA-NLMS et VSS-RZA-NLMS.

Enfin, en utilisant le logiciel MATLAB®, une série de simulations a été effectuée à la fois pour différentes réponses impulsionnelles acoustiques synthétiques et réelles avec des entrées stationnaires et des entrées non-stationnaires afin d'analyser, d'étudier et de comparer les points forts et les points faibles des algorithmes concernés en termes de la vitesse de convergence, la précision d’estimation et la complexité de calcul.

Mots clés: Filtrage Adaptatif, Algorithmes Parcimonieux, Réponses Impulsionnelles Acoustiques, Identification, Acquisition comprimée (en anglais, Compressed Sensing), LMS, NLMS, Parole, Performance à l’état d’équilibre, Complexité.

ABSTRACT

Acoustic impulse responses (AIRs) identification and echo cancellation are common problems in the field of signal processing. In this work, firstly, we provide an overview of adaptive filtering and the major developments in the area of sparse algorithms, starting from the celebrated works on proportionate normalized least mean square (PNLMS) algorithm and its several variants to more recent approaches that use compressed sensing (CS) framework.

Then, the theoretical details of the most important and recent sparse NLMS-based adaptive filtering algorithms are presented, and their computational complexity is analyzed. The algorithms of interest include NLMS, its classical proportionate sparseness-aware (SA) versions (PNLMS, IPNLMS and MPNLMS) and their sparseness-controlled (SC) upgrades (SC-PNLMS, SC-IPNLMS and SC-MPNLMS) as well as some recent CS-based algorithms namely ZA-NLMS, RZA-NLMS, VSS-ZA-NLMS and VSS-RZA-NLMS.

Finally, using MATLAB® software, a series of simulations were carried out both in synthetic and real different-sparseness acoustic impulse responses with stationary and non-stationary inputs in order to analyze, investigate and compare the algorithms strengths and weaknesses in terms of convergence speed, estimation accuracy and computational complexity.

Keywords: Adaptive Filtering, Sparse Algorithms, Acoustic Impulse Responses, System Identification, Compressed Sensing, LMS, NLMS, Speech, Steady-State Performance, Complexity.

DEDICATION

To my beloved mother

To my father, my cherished brothers and my dear sister

To all my lovely family

To the memory of all my grandparents

To all my teachers and instructors

To all my friends and classmates

To everybody with whom I exchanged a smile

To the pursuit of knowledge, the thrill of discovery and the hope of their use for the betterment of the humankind

ACKNOWLEDGMENTS

First and foremost, I have to thank greatly God that gave me the power to complete this work.

I would like to express my deepest gratitude to my supervisor Pr. BENALLAL Ahmed for all of his constant support, motivation and friendship throughout the completion of this work. His wide knowledge, logical thinking, continuous encouragement and wise guidance made it an absolute privilege to work with him.

I am sincerely grateful to all jury members that gave me the honor of judging my work.

Special thanks to all my teachers in the electronics department of Blida-1 University for their valuable edification.

Finally, I would like to send my greatest respect and gratitude to my parents for their unconditional love and support which added an essence to my life and made my successes possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS, GRAPHICS AND TABLES

INTRODUCTION

The term *filter* is often used to describe the form of a piece of physical hardware or software that is applied to a set of noisy data in order to extract information about a prescribed quantity of interest. A filter is said to be *linear* if the filtered, smoothed, or predicted quantity at the output of the device is a *linear function* of the observation applied to the filter input. Otherwise, the filter is *nonlinear* [1]. A linear filtering system is called *sparse* if its impulse response contains a small number of *active* taps (taps with considerably large magnitude) in the presence of a large number of *non-active* taps (taps of magnitudes equal to or close to zero). The network echo canceller is one example of such systems [2]-[3].

Another example is the acoustic echo generated due to coupling between microphone and loudspeaker in hands free mobile telephony, where the sparseness (or sparsity) of the acoustic channel impulse response varies with the loudspeaker-microphone distance [4]. Other well known examples of sparse systems include HDTV where clusters of dominant echoes arrive after long periods of silence [5], wireless multipath channels which, on most of the occasions, have only few clusters of significant paths [6], and underwater acoustic channels where the various multipath components caused by reflections off the sea surface and sea bed have long intermediate delays [7].

The LMS algorithm and its different versions used for conventional system identification do not use the *a priori knowledge* of the sparseness of the system. Consequently, they perform poorly both in terms of steady state excess mean square error (EMSE) and convergence speed. Recently, several alternate algorithms have been proposed to exploit the sparse nature of the system impulse response and achieve better performance. The most famous amongst them is the proportionate normalized LMS (PNLMS) algorithm [8] and its various versions.

Each coefficient, in the PNLMS algorithm, is updated independently with a step-size that varies proportionally with respect to the magnitude of the particular coefficient estimate of the system, resulting in fast *initial* convergence for sparse systems. However, the rate of convergence slows down afterwards considerably, sometimes slower than the NLMS algorithm. In [9], an attempt had been made to overcome this limitation by proposing an algorithm that switches in alternate cycles between NLMS and PNLMS algorithms. After that, in [10], the well-known improved PNLMS (IPNLMS) algorithm is proposed where a *controlled* mixture of the PNLMS and the NLMS algorithms is used.

Independently, certain modifications have been introduced to the gain matrix of the PNLMS algorithm in order to maintain its initial high rate of convergence, resulting in the so-called the μ-law PNLMS (MPNLMS) algorithm [11]. Many other approaches for sparse system identification have been proposed separately from the PNLMS family, like partial update LMS which deploys statistical detection of active taps [12]-[13] and the exponentiated gradient (EG) algorithm [14]-[16] which is shown in [17] that it is equivalent to the IPNLMS algorithm [10] under certain conditions.

Most of the aforementioned algorithms are very sensitive to the degree of system sparsity. Although, all of them perform satisfactorily for systems that are highly sparse, many of them have a performance that deteriorates as the degree of system sparsity reduces and generally, they perform worse than the usual *sparsity-agnostic* adaptive algorithms like LMS, NLMS and RLS for dispersive (non-sparse) systems. The IPNLMS algorithm [10] is an exception. It uses a free manually-tunable parameter that enables the algorithm to adapt to the existing sparsity level of the system. Recently, in [18]-[20], more enhancements have been achieved mainly by imposing a sparseness measure on the PNLMS, IPNLMS and MPNLMS algorithms resulting in a class of *sparseness-controlled* (SC) algorithms.

It is worthy to state that the subject of sparse adaptive filtering algorithms has known a renewed dynamism in the last few years. This was due to the emergence of the framework of compressive sensing (CS) [21]-[23], where a linear superposition of a small number of stored signals (called *atoms*) is used to construct a *sparse representation* of the signal. Unlike the usual basis in vector space, the atoms are drawn from an *over-complete* dictionary and thus the representation of the signal using atoms is not unique [24]. Therefore, numerous heuristic iterative methods have been recently developed in order to compute a sparse representation of the signal, notably from them being the matching pursuit (MP) method [25]-[27] and the basis pursuit (BP) method [28] or least absolute shrinkage and selection operator (LASSO) [29].

Motivated by LASSO, two sparsity-aware LMS algorithms were proposed in [30], namely, the zero-attracting LMS (ZA-LMS) and the reweighted-zero-attracting LMS (RZA-LMS). This has been achieved by introducing two different sparsity constraints (the ℓ_1-norm and the log sum penalty function) into the convex quadratic cost function of the LMS algorithm. The results presented in [30] showed that both the ZA-LMS and RZA-LMS algorithms behave better than the standard LMS in both transient and steady state performance for highly sparse systems but, for less-sparse systems, their performance degrades noting that the RZA-LMS is more robust against the degree of system sparsity. Further extension of the above philosophy was made on the NLMS algorithm [31]-[32].

Recently, the conventional invariable step-size (ISS) ZA-NLMS and RZA-NLMS have been upgraded to a variable step-size (VSS) version resulting in two more improved-performance algorithms named VSS-ZA-NLMS and VSS-RZA-NLMS [33], [34].

In a separate side, several RLS-based sparse adaptive filtering algorithms have been developed using the LASSO as well [35]-[38] resulting in interesting improvements both in terms of transient convergence speed and steady state excess MSE.

It should be addressed that most of the aforementioned studies on sparse adaptive filtering algorithms assume white Gaussian inputs and used simple impulse responses with relatively small number of taps (8, 16, 64 taps). However, in this report, we used different synthetic and real different-sparsity acoustic impulse responses (AIRs) that have larger sizes (256, 1024, 2048 & 8192 taps) with different-nature inputs (stationary and non-stationary). These relatively long AIRs are used in order to approach and simulate more effectively the real acoustic applications.

Because of the implementation simplicity and low complexity cost of the NLMS algorithm and in front of this huge number of new sparse algorithms, we focused our study on the most known and recent NLMS-based ones. Accordingly, the overall goal of this work is to present, analyze and compare the most important sparse adaptive filtering NLMS-based algorithms, in order to outline their capabilities and performances in the context of AIRs identification and echo cancellation. The report consists of three chapters organized as follows:

Chapter 1 "*State of the Art*": it starts by describing sparse impulse responses giving an example of room acoustic system. Then, it provides a review of the general principles of adaptive system identification and acoustic echo cancellation (AEC) process, as well as brief discussions of Wiener filter, the steepest descent method, NLMS and variable step-size NLMS (VSS-NLMS) algorithms. It concludes by defining the used performance-measures.

Chapter 2 "*Sparse Adaptive Filtering NLMS-Based Algorithms"*: it provides the mathematical details about all NLMS-Based algorithms of interest, and then gives an analysis of their computational complexity.

Chapter 3 "*Simulation, Results and Discussion"*: it starts by explaining the simulation details and the obeyed comparison criteria. Then, it presents analyses and discussions of the obtained comparison details accompanied with MATLAB® simulation figures in order to well demonstrate the performances of the studied algorithms. It ends with a summary of the main achieved results.

The report ends with a conclusion that summarizes the whole work, future works for forward researches in the field as well as appendices that list the used abbreviations and mathematical symbols and provide brief pseudo-codes of the studied algorithms.

CHAPTER 1
STATE OF THE ART

1.1. Sparse impulse response

A simple definition of a *sparse* impulse response is: “an impulse response (IR) that has only a small percentage of its components with a significant magnitude while the rest are zero or very small”. It could be defined also as: “an impulse response is sparse if a large fraction of its energy is concentrated in a small fraction of its duration” [39]. For example, in a network impulse response, only about 8-12 ms in a 64 or 128 ms time duration are active and the others are zeros (or inactive). The inactive part accounts for bulk delay due to network loading, encoding … etc.

Notable amongst the acoustic sparse impulse responses, there are network echo paths, marine and land geophysical seismic impulses responses. However, an acoustic echo path in a conference or a hands-free communication system is not usually considered as sparse as a network echo path impulse response, but with the increase of user demand, it was needed to provide an echo canceller of as many as 512-2048 taps in order to deal with possible propagation delays of sound [40].

1.2. Room acoustic impulse response

In a room system, the first sound to be heard is the one coming via the direct path from the source. Then, after a short while, the listener will hear the attenuated reflections of the sound (echoes) off the walls, see Figure 1.1.

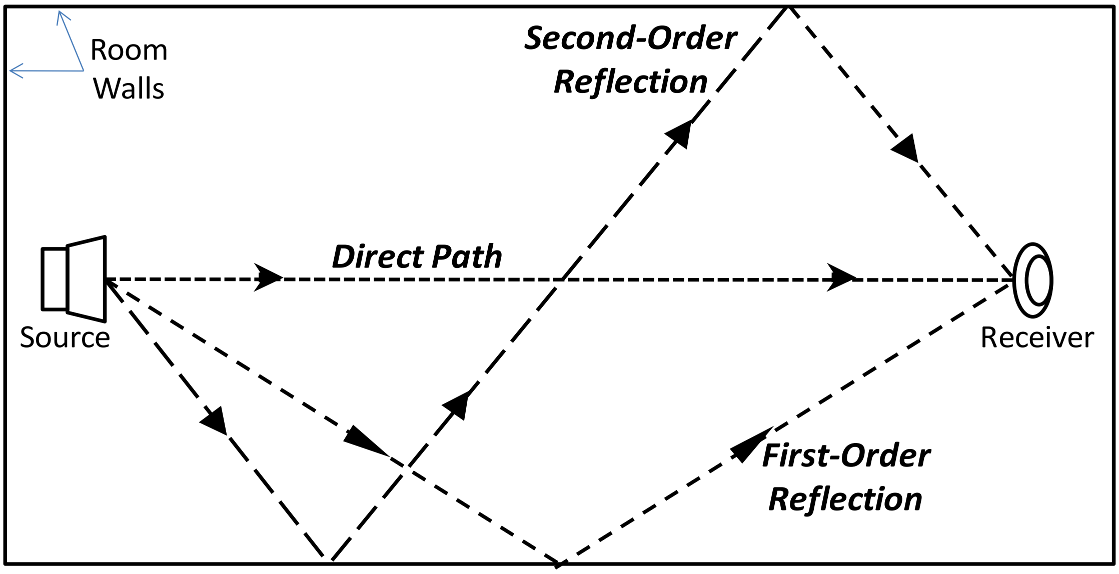

Figure 1.1: A sketch illustrating sound propagation in a room.

As the sound is reflected again and again off the walls, each reflection will be further delayed and attenuated. More inspections of the room impulse response gave the observation that the sound decays at an exponential rate [41]. Therefore, the impulse response of a room may be as shown in Figure 1.2.

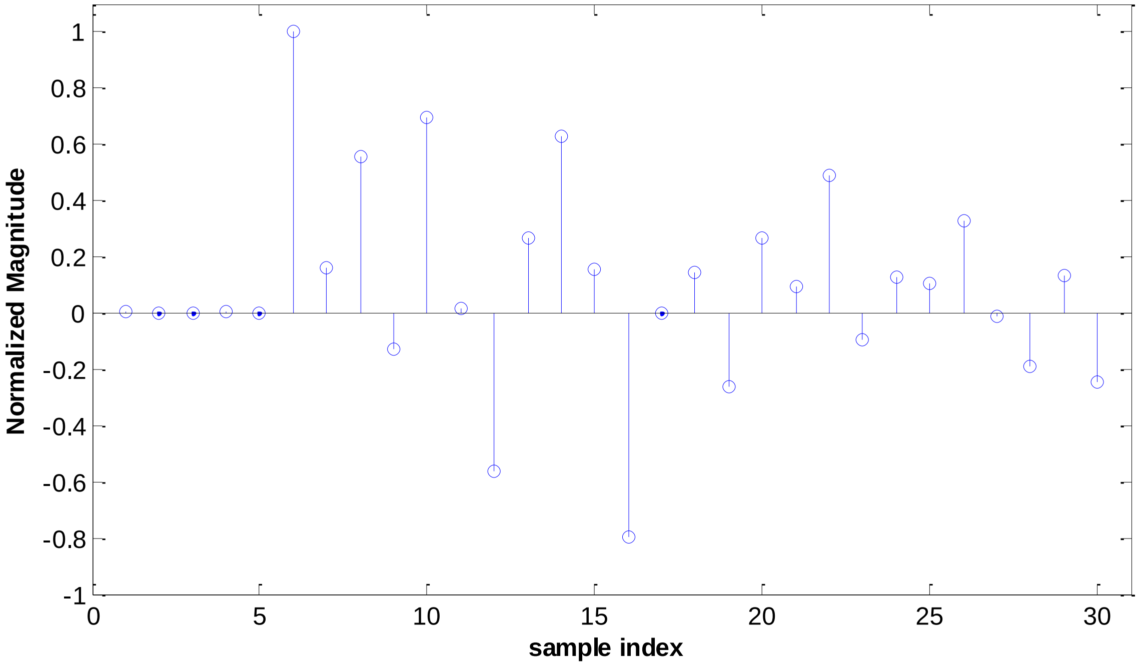

Figure 1.2: Impulse response for a room as shown in Figure 1.1.

By having absorbers around the wall, the effects of echoes can be reduced resulting in a room impulse response with less active coefficients, as shown in Figure 1.3. The latter impulse response is said to be *more sparse* system than the former, due to having a majority of inactive filter taps.

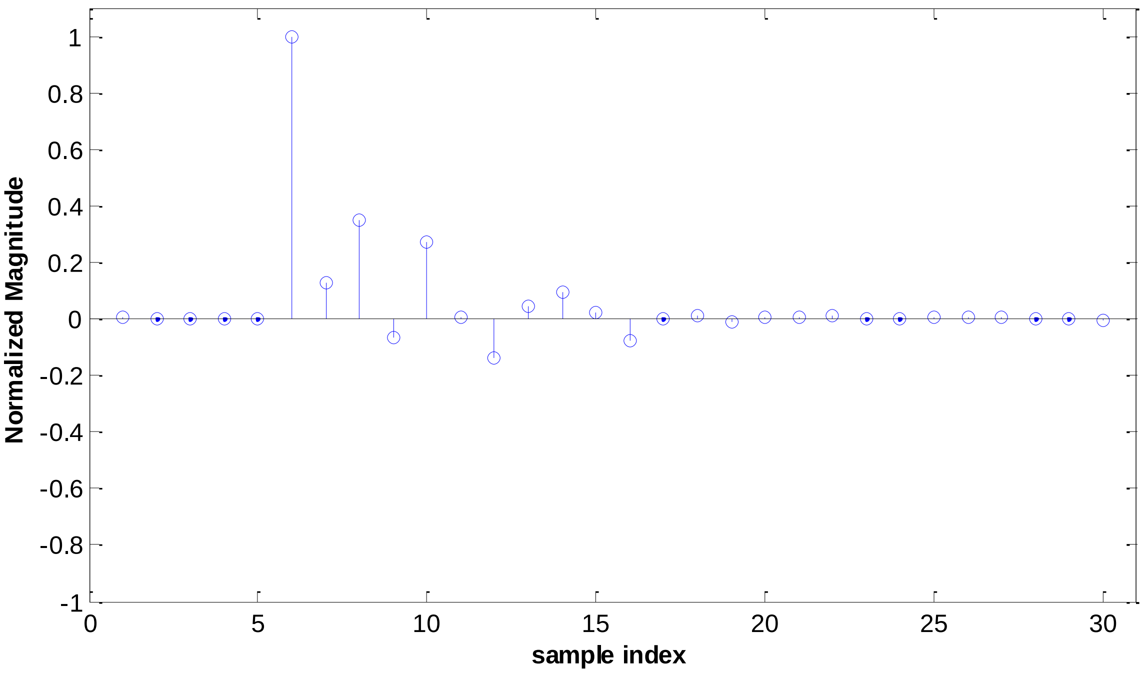

Figure 1.3: Sparse impulse response for a room in the presence of echo absorbers.

The identification of different types of system impulse responses is a common problem in signal processing. Moreover, there are many practical application problems that cannot be successfully solved by using fixed digital filters because either we do not have sufficient information to design a digital filter with fixed coefficients or the design criteria change during the normal operation of the filter. Most of these applications can be successfully solved by using special *smart* filters known as *adaptive filters*. The distinguishing feature of adaptive filters is that they can modify their response to improve performance during operation without any intervention from the user [42].

The wide range of applications of adaptive filters can be mainly subdivided into four classes: (1) system identification, (2) system inversion, (3) signal prediction, and (4) multisensor interference cancellation [42]. In this report, the classification of interest is system identification especially in the purpose of echo cancellation.

1.3. Concept of adaptive system identification and echo cancellation

In general the problem of system identification involves the build of an approximation of an unknown system given only two signals, the input signal and a reference (or a desired) signal. Typically, the unknown system is modeled linearly with a finite impulse response (FIR), and adaptive filtering algorithms are employed to iteratively converge upon an estimate of the response [44]. Figure 1.4 illustrates the concept of adaptive linear system identification process.

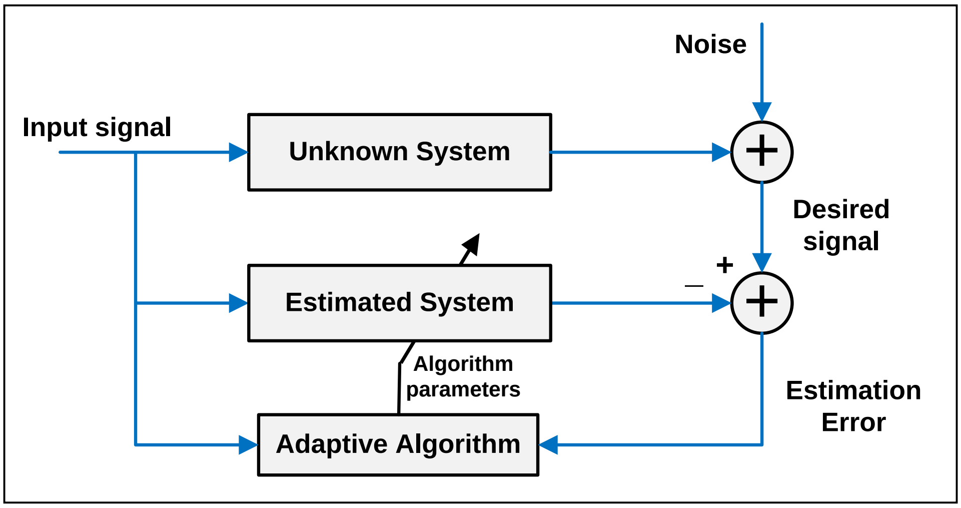

Figure 1.4: The concept of adaptive linear system identification.

1.3.1. Adaptive acoustic echo cancellation

An *echo* is the delayed and distorted version of an original signal that returns to its source. In some applications (radar, sonar, or ultrasound), the echo is the wanted signal; however, in communication applications, the echo is an unwanted signal that must be eliminated. There are two types of echoes in communication systems: (1) *electrical or line echoes*, which are generated electrically due to impedance mismatches at points along the transmission medium, and (2) *acoustic echoes*, which result from the reflection of sound waves and acoustic coupling between a microphone and a loudspeaker (Figure 1.5) [42].

An acoustic echo may be imperceptibly 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], [45]-[47].

Several methods to deal with acoustic echoes have been developed. One of the best techniques to prevent or control echoes is *adaptive echo cancellation*. Its basic idea is simple: to cancel the echo, a replica or pseudo-echo is generated and then subtracted from the real echo. Figure 1.5 illustrates how to synthesize the echo replica by passing the signal at the loudspeaker through a device designed to duplicate the reverberation and echo properties of the room (echo path).

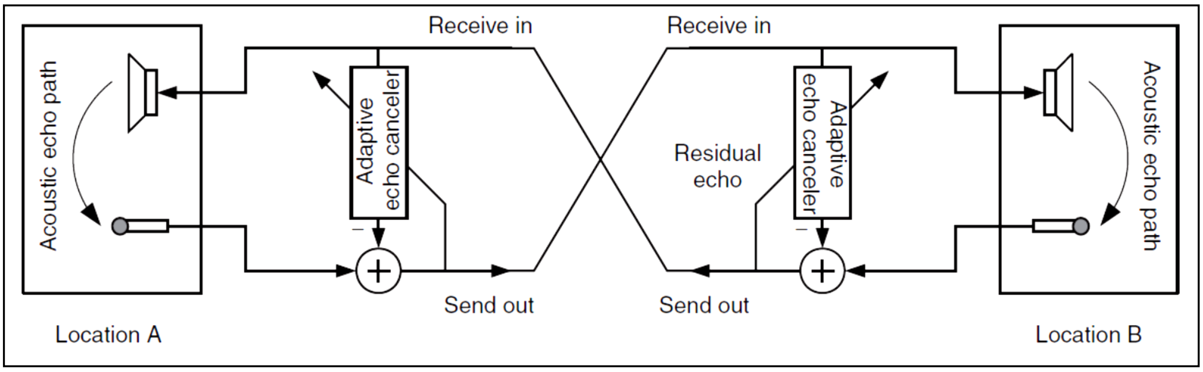

Figure 1.5: Principle of acoustic echo cancellation using an adaptive echo canceller [42].

An *adaptive acoustic echo canceller* is a device that attempts to cancel the acoustic echo. Figure 1.6 shows a loudspeaker-room-microphone system (LRMS) describing a typical acoustic echo cancellation (AEC) system, with an echo canceller employing an adaptive filter.

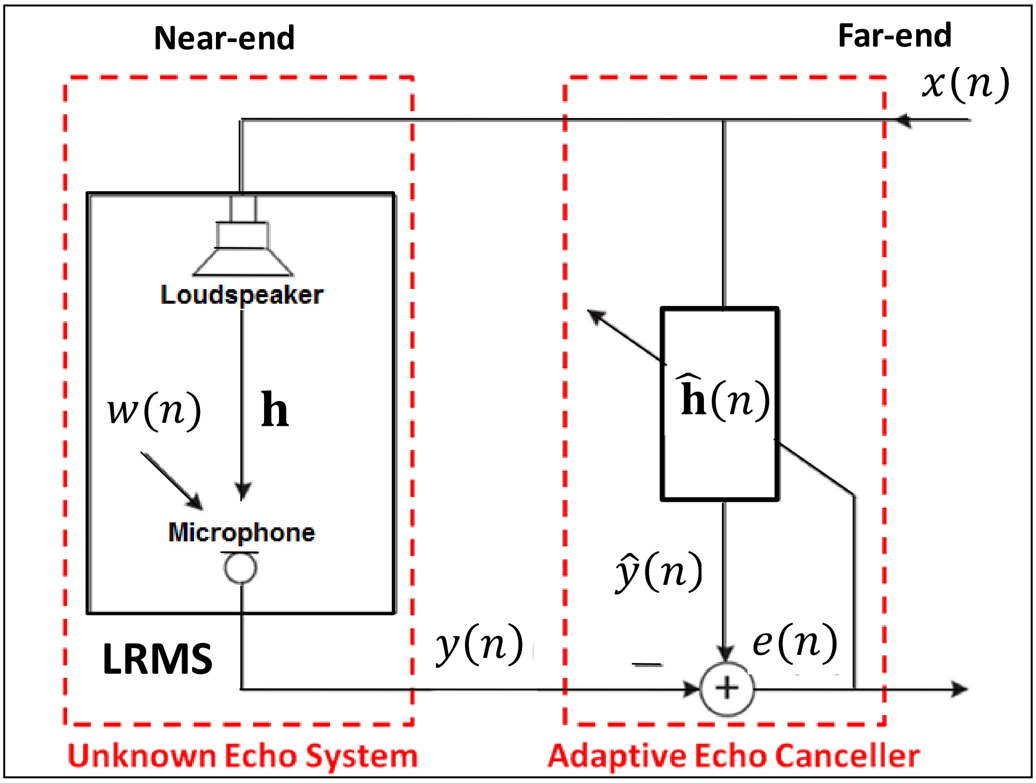

Figure 1.6: Adaptive system for acoustic echo cancellation in a loudspeaker-room-microphone system (LRMS) [43-modified].

1.3.2. Formulations and definitions

An adaptive acoustic echo canceller assumed with FIR model configuration (due its stability characteristics) has the coefficients

$$\widehat{\mathbf{h}}(n) = \left[\hat{h}_0(n), \hat{h}_1(n), \ldots, \hat{h}_{L-1}(n)\right]^{\boldsymbol{T}} \qquad (1.1)$$

where L is the length of the adaptive filter assumed equal to length of the unknown room impulse response $\mathbf{h}$ which is defined by

$$\mathbf{h} = [h_0, h_1, \ldots, h_{L-1}]^{\boldsymbol{T}} \qquad (1.2)$$

The superscript $^{\boldsymbol{T}}$ denotes the transposition operator. $x(n)$ is the time-varying far-end signal which is transmitted to the near-end loudspeaker in the LRMS. The microphone in the near-end room receives the desired signal (the output of the LRMS) that is given by

$$y(n) = \mathbf{h}^{\boldsymbol{T}}\mathbf{x}(n) + w(n) \qquad (1.3)$$

where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^{\boldsymbol{T}}$ is a vector containing L samples of the input signal and $w(n)$ is a stationary, zero-mean and independent noise that is uncorrelated with any other signal [44].

The difference of the output of the echo canceller $\hat{y}(n)$ and the desired signal $y(n)$

$$e_p(n) = y(n) - \hat{y}(n) = \left[\mathbf{h}^{\boldsymbol{T}} - \widehat{\mathbf{h}}^{\boldsymbol{T}}(n)\right]\mathbf{x}(n) + w(n) \qquad (1.4)$$

is called *a posteriori* error signal $e_p(n)$. Since this latter is computed after the adaptive filter coefficients have been updated, the previous estimation of the impulse response $\widehat{\mathbf{h}}^{\boldsymbol{T}}(n-1)$is used to compute the *a priori* error signal $e(n)$ at each iteration. It is computed as

$$e(n) = y(n) - \widehat{\mathbf{h}}^{\boldsymbol{T}}(n-1)\mathbf{x}(n) \qquad (1.5)$$

Since the objective of an echo canceller is to estimate the unknown system $\mathbf{h}$ as closely as possible, $e(n)$ must come significantly smaller at each iteration, as the filter coefficients converge to the unknown true impulse response $\mathbf{h}$ [43].

It is necessary for the echo canceller to be *adaptive* in order to be capable of converging and tracking the time-varying nature of the echo path which may arise due to a change in temperature [48] and pressure or to changes in the acoustic environment [49], for example movements of people, doors, windows or furniture. For these reasons, adaptive algorithms are utilized effectively to track and compensate any changes in the LRMS.

Before getting into the relevant details of adaptive filtering algorithms, we start first by reviewing the Wiener filter, the way to get the optimum solution and the steepest decent algorithm, the recursive way to reach a sub-optimum solution.

1.4. Wiener filter

Figure 1.7 shows a simplified conceptual diagram to illustrate the use of Wiener filter for system estimation.

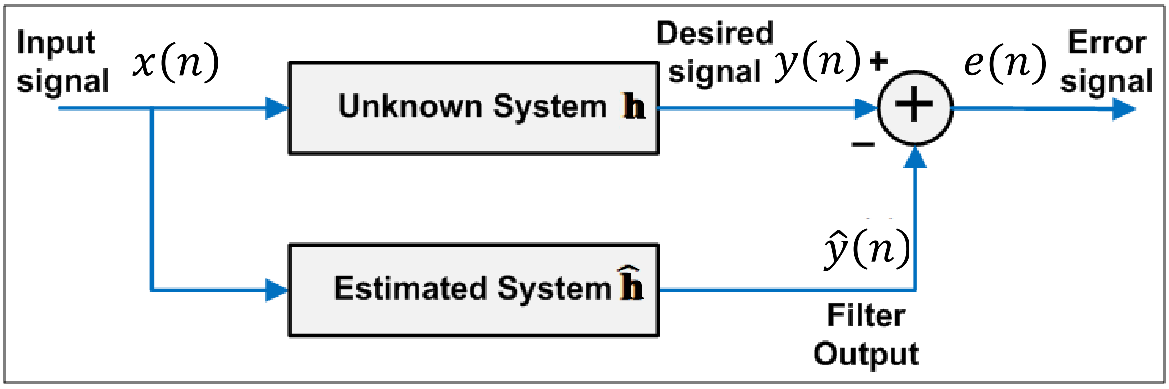

Figure 1.7: Linear conceptual diagram for system estimation.

The difference between the true system and the estimated system represents the quality of the filter estimation which is a function of the error signal. This function can be viewed as a *cost* incurred when the estimation is incorrect. A very-common choice for *cost function* $\mathcal{J}$ (or criterion of performance) is the mean-square error (MSE) which is always positive and defined [42] as

$$\mathcal{J}(n) = E\{e^2(n)\} \qquad (1.6)$$

where the operator $E\{.\}$ denotes the expectation value.

Assuming that $e^2(n)$, $\mathbf{x}(n)$ and $y(n)$ are *statistically stationary* and that the input signal is always a real-valued. Hence, during derivations and calculations, the non-conjugate matrix transpose $([.]^{\boldsymbol{T}})$ is used instead of the Hermitian transposition ($[.]^{\boldsymbol{H}}$). Therefore, the MSE cost function can be written as

$$\mathcal{J}(n) = E\{e^2(n)\} = E\{y^2(n)\} - 2\mathbf{p}^{\boldsymbol{T}}\,\widehat{\mathbf{h}} + \widehat{\mathbf{h}}^{\boldsymbol{T}}\mathbf{R}\,\widehat{\mathbf{h}} \tag{1.7}$$

where $\mathbf{p}$ is the L-by-1 *cross-correlation vector* between $y(n)$ and $x(n)$ defined as follows:

$$\mathbf{p} = E\{y(n)\mathbf{x}(n)\} \tag{1.8}$$

and $\mathbf{R}$ is the L-by-L *auto-correlation matrix* of the tap inputs in the FIR filter and can be defined as

$$\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^{\boldsymbol{T}}(n)\} \tag{1.9}$$

The minimum of the error surface can be obtained by setting the partial derivatives of $\mathcal{J}$, with respect to each filter coefficient, to zero. That is

$$\nabla\mathcal{J} = \frac{\partial\mathcal{J}}{\partial\,\widehat{\mathbf{h}}} = 2\mathbf{R}\,\widehat{\mathbf{h}} - 2\mathbf{p} = 0 \tag{1.10}$$

Assuming that the auto-correlation matrix $\mathbf{R}$ is nonsingular [1] and its inverse is $\mathbf{R}^{-\mathbf{1}}$, the unique *optimum* impulse response $\widehat{\mathbf{h}}_{opt}$ which minimizes the MSE is obtained by

$$\widehat{\mathbf{h}}_{opt} = \mathbf{R}^{-\mathbf{1}}\mathbf{p} \tag{1.11}$$

which is known as *Wiener-Hopf solution.* This latter can be used to estimate the unknown room impulse response. However, this approach is not appropriate in dealing with non-stationary signals like speech signals and furthermore, the autocorrelation and cross-correlations are unknown. In this case, an alternative procedure is to use the method of *steepest-descent*, which is a gradient type iterative technique that has been employed to optimize cost functions [1]. It defines the filter coefficient update equation with of the following recursive relation

$$\begin{aligned}\widehat{\mathbf{h}}(n) &= \widehat{\mathbf{h}}(n-1) + \mu\frac{1}{2}\left[-\nabla\mathcal{J}\left(\widehat{\mathbf{h}}(n-1)\right)\right]\\ &= \widehat{\mathbf{h}}(n-1) + \mu\left[\mathbf{p} - \mathbf{R}\,\widehat{\mathbf{h}}(n-1)\right]\end{aligned} \tag{1.12}$$

where $\nabla\mathcal{J}\left(\widehat{\mathbf{h}}(n-1)\right)$ is the *gradient* vector of the cost function and μ is a positive real-valued constant known as the *step-size parameter* or the *adaptation constant*.

It controls the size of the descent in the direction of the negative gradient [42]. The rate of convergence of the steepest-descent method is faster for higher step size μ. For stability it must be lie with the range

$$0 < \mu < \frac{2}{\vartheta_{max}} \tag{1.13}$$

where ϑ_{max} is the *largest eigenvalue* of the auto-correlation matrix $\mathbf{R}$ [1].

According to eq. (1.12), the computations using the steepest descent method require an explicit knowledge of the input signal statistics [1]. However, in reality, this is not always possible. To solve this issue, the *least mean square (LMS) algorithm* can be used.

1.5. Stochastic gradient-based adaptive algorithms

Basically, we may identify two distinct major approaches for adaptive algorithms. One is the *stochastic gradient approach* that is attained by the least mean square (LMS) algorithm. The other approach minimizes a deterministic sum of weighted errors squared. It is known as the *recursive least-squares (RLS) approach* [1]. In this work, we focus on the first approach, more precisely, the LMS-based algorithms.

1.5.1. Least mean square (LMS) algorithm

The LMS algorithm is by far the most popular member of the family of *stochastic gradient algorithms*. The term ‘stochastic gradient’ is intended to distinguish the LMS algorithm from the method of the steepest descent that uses a ‘deterministic gradient’ in a recursive computation of the Wiener filter for stochastic inputs. A significant feature of the LMS algorithm is its *simplicity*. Moreover, it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion [1].

The filter coefficients update equation for LMS substitutes the gradient vector in eq. (1.12) with an *instantaneous estimate* of the gradient vector, as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\frac{1}{2}\left[-\hat{\nabla}\mathcal{J}\left(\hat{\mathbf{h}}(n-1)\right)\right] \tag{1.14}$$

where $\hat{\nabla}\mathcal{J}\left(\hat{\mathbf{h}}(n-1)\right)$ is defined by using the *instantaneous estimates* for $\mathbf{R}$ and $\mathbf{p}$ that are based on sample values of $\mathbf{x}(n)$ and $y(n)$ [1]. This gives

$$\hat{\nabla}\mathcal{J}\left(\hat{\mathbf{h}}(n-1)\right) = -2\mathbf{x}(n)e(n) \tag{1.15}$$

Hence, the filter coefficients update equation for LMS is expressed as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu 2\mathbf{x}(n)e(n) \tag{1.16}$$

The LMS algorithm is convergent in the mean square if its step size μ_{LMS} (has the dimensions of the *inverse power*) satisfies the following condition [42]

$$0 < \mu_{LMS} < \frac{2}{\mathrm{trace}(\mathbf{R})} \tag{1.17}$$

where $\mathbf{R}$ is the autocorrelation matrix of the input.

1.5.2. Normalized least mean square (NLMS) algorithm

The selection of μ_{LMS} in practical applications is complicated because usually the power of the input signal is either unknown or varies with time. In order solve this issue, μ_{LMS} is normalized over the squared Euclidean norm (ℓ_2-norm) of $\mathbf{x}(n)$ resulting in the well-known *normalized LMS* (or NLMS) algorithm [51] and its filter coefficients update equation is given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n).\mathbf{x}(n) + \delta_{NLMS}} \tag{1.18}$$

where δ_{NLMS} is a *regularization parameter* used to prevent division by zero and stabilizes the solution [52]. We take $\delta_{NLMS} = cst.\sigma_x^2$ [10], [39]. σ_x^2 is the variance of the input signal and cst is a positive constant. The NLMS algorithm is convergent in the mean square if its step size μ (*dimensionless*) satisfies the following condition [1]

$$0 < \mu < 2 \tag{1.19}$$

The main drawback of the NLMS (or LMS) algorithm is its *low convergence rate*, especially when operating on highly correlated input signals; its convergence depends on *spectral dynamic range* of the input signal [54]. Consequently, several modifications have been proposed in the literature to improve its performance such as the variable-step-size NLMS (VSS-NLMS) [1], [55], [56]. The NLMS algorithm is summarized in Table B.1 (Appendix B).

1.5.3. Variable step-size NLMS (VSS-NLMS) algorithm

In case of the conventional (invariable step-size or ISS) normalized least mean square (ISS-NLMS), the step-size governs the rate of convergence and the steady-state excess mean-square error. To meet the conflicting requirements of fast convergence and low misadjusment (good estimation accuracy), the step-size needs to be controlled. In [57], they had proposed a variable step-size NLMS (VSS-NLMS) algorithm. The update equation of the VSS-NLMS algorithm is

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} \tag{1.20}$$

where

$$\mu(n) = \mu_{max}\frac{\boldsymbol{p}^T(n)\boldsymbol{p}(n)}{\boldsymbol{p}^T(n)\boldsymbol{p}(n) + C} \tag{1.21}$$

where C is a positive constant parameter satisfying $C \sim \mathcal{O}(1/S\mathrm{NR})$, where SNR is the input signal-to-noise ratio (SNR), and μ_{max} is the *maximal* step-size [33].

According to eq. (1.21), the range of VSS is given by $\mu(n) \in (0, \mu_{max})$. To ensure the stability of the adaptive algorithm, the maximal step-size μ_{max} is usually set to be less than 2 [55]. $\boldsymbol{p}(n)$ is approximated as follows

$$\boldsymbol{p}(n) = \beta\boldsymbol{p}(n-1) + (1-\beta)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + C} \tag{1.22}$$

where $\beta \in [0, 1)$ is the *smoothing factor* to control the value of the VSS and the estimation error [33]. The VSS-NLMS algorithm is given briefly in Table B.2 (Appendix B).

1.6. Performance measures

Some of the most important factors that influence the choice of the adaptive algorithms are: the *convergence rate,* the *accuracy* of the obtained solution, the *computational complexity* of the algorithm, the *robustness* to numerical errors when implemented in finite-precision arithmetic, and finally, the *tracking ability*, i.e., the performance of the algorithm when operating in a non-stationary environment [44]. In the following subsections, the adopted measures are reviewed.

1.6.1. Mean square error

The mean square error (MSE) is one of the ways to define an objective function. It is defined as the expected value of the square of the error ($E\{e^2(n)\}$). But in almost all our simulations in this report, we use the *time-average normalized MSE* (denoted as NMSE) [44] defined as

$$\mathrm{NMSE(dB)} = 10\log_{10}\left(\frac{\langle e^2(n)\rangle}{\langle y^2(n)\rangle}\right) \tag{1.23}$$

where $\langle\,.\,\rangle$ the symbol denotes a time averaging (over 256 samples for most of our simulations), and $y(n)$ is the desired signal. For the *speech* input signal, we prefer to use the *time-average MSE* [44] defined as

$$\mathrm{MSE(dB)} = \ 10\log_{10}(\langle e^2(n)\rangle) \tag{1.24}$$

and, as can be seen from equations (1.23) and (1.24), a lower MSE value is favorable [43].

1.6.2. Learning curves

Plots of the NMSE or MSE as a function of the number of iterations n, are known as *learning curves*. They characterize the performance of an adaptive filter and are widely used in theoretical and experimental studies [42].

CHAPTER 2
SPARSE ADAPTIVE FILTERING NLMS-BASED ALGORITHMS

In view of this chapter, we reviewed the basic proportionate-type NLMS adaptive algorithms; the classical PNLMS [8], the improved PNLMS (IPNLMS) [10] and the μ-law PNLMS (MPNLMS) [11], [61]. Their sparseness-controlled (SC) versions are also discussed in order to show how the use of the sparseness measure enhances the algorithms robustness against the sparsity variation. Finally, some of most recent developments in the field, due to the introduction of *compressed sensing* concept, are presented such as the zero-attracting and reweighted zero-attracting NLMS (ZA-NLMS and RZA-NLMS) and their variable step-size (VSS) version.

2.1. Basic sparseness-aware NLMS-based adaptive filtering algorithms

2.1.1. Proportionate NLMS (PNLMS) algorithm

One of the first sparse adaptive filtering algorithms considered as a milestone for network echo cancellation (NEC) is the so called proportionate NLMS (PNLMS) which was proposed a decade ago [8]. The coefficient update equation of PNLMS is slightly different from NLMS so that each filter coefficient is updated with an independent step-size that is linearly proportional to the magnitude of that estimated filter coefficient [41]. More precisely, the PNLMS algorithm assigns higher step-sizes for coefficients with higher magnitude using a control matrix $\mathbf{Q}$ as shown below [8] and the rest of terms are carried over from NLMS.

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{PNLMS}} \tag{2.1}$$

where δ_{PNLMS} is the regularization parameter for PNLMS. It is usually is taken as $\delta_{PNLMS} = \delta_{NLMS}/L$, [18], [39] and the diagonal matrix

$$\begin{aligned}\mathbf{Q}(n-1) &= diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\} \\ &= \begin{Bmatrix} q_0(n-1) & 0 & \dots & 0 \\ 0 & q_1(n-1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & q_{L-1}(n-1) \end{Bmatrix}\end{aligned} \tag{2.2}$$

The elements of the control matrix can be expressed as

$$q_l(n-1) = \frac{\kappa_l(n-1)}{\frac{1}{L}\sum_{i=0}^{L-1} \kappa_i(n-1)}, \tag{2.3}$$

where $\kappa_l(n-1) = max\{\kappa_{min}(n), |\hat{h}_l(n-1)|\}$ (2.4)

and $\kappa_{min}(n-1) = \rho \times max\{\gamma, |\hat{h}_0(n-1)|, |\hat{h}_1(n-1)|, \cdots, |\hat{h}_{L-1}(n-1)|\}$ (2.5)

with $0 \le l \le L-1$ being the tap-indices. The parameters γ and ρ are positive numbers with typical values $\gamma = 0.01$ and $\rho \in [1/L, 5/L]$ [62]. The parameter γ prevents $\hat{h}_l(n-1)$ from stalling during initialization stage where $\widehat{\mathbf{h}}(0) = \mathbf{0}_{L\times 1}$while ρ prevents individual filter coefficients from stalling when their magnitudes are much smaller than the magnitude of the largest coefficient [18], [39], [62].

It can be seen that for $q_l = 1, \forall l$, PNLMS is equivalent to NLMS [18]. The PNLMS algorithm is shown in Table B.3 (Appendix B).

It is worthy to mention here that the PNLMS algorithm was developed in an *intuitively* manner [41], because the equations used to calculate the step-size control factors are *not* based on any optimization criteria but are designed in *an ad-hoc way*. Since, in sparse systems, PNLMS algorithm employs larger step-sizes for active coefficients (to speed up their convergence rate) than for inactive coefficients [18], it achieves a fast-convergence initial phase compared to NLMS, however, this is followed by a second phase of *slower*-convergence rate. Moreover, when the impulse response is less sparse or dispersive, the NLMS works better than PNLMS [41].

2.1.2. Improved proportionate NLMS (IPNLMS) algorithm

An improvement of PNLMS is the IPNLMS algorithm [10] which was originally developed for NEC and was further used for the identification of acoustic room impulse responses [64]. It employs a combination of proportionate (PNLMS) and non-proportionate (NLMS) updating techniques, with the relative significance of each controlled by a *factor* α. Elements of the control matrix $\mathbf{Q}$ for IPNLMS are given [43] by

$$q_l(n-1) = \frac{(1-\alpha)}{2L} + \frac{(1+\alpha)\left|\hat{h}_l(n-1)\right|}{2\left\|\hat{\mathbf{h}}(n-1)\right\|_1 + \delta_{IPNLMS}}, \qquad 0 \le l \le L-1 \tag{2.6}$$

where $\|.\|_1$ is defined as the ℓ_1-norm and the first and second terms are the NLMS and the proportionate terms respectively. The most used values for α are 0, -0.5 and -0.75. The latter values are favorable choices for most applications of echo cancellation [10], [63]. The regularization parameter for IPNLMS should be taken [10], [39] as

$$\delta_{IPNLMS} = \frac{(1-\alpha)}{2L}\delta_{NLMS} \tag{2.7}$$

This choice of regularization is to ensure that the IPNLMS algorithm achieves the *same steady-state* level compared to that of the NLMS algorithm for the *same* step-size [18]. Equation (2.6) is made up of the sum of two terms, where the first is a constant and the second term is a function of the weight coefficients. It can be noticed that, when $\alpha = -1$ the second term becomes zero and therefore the q_l becomes $1/L$. It means that the same update will be made to all filter coefficients regardless of their individual magnitudes. So, for this value of α, IPNLMS performs as NLMS. For α close to unity, the second term dominates the equation, and as a result it behaves as PNLMS. The IPNLMS algorithm is recapitulated in Table B.4 (Appendix B).

2.1.3. *μ*-law proportionate NLMS (MPNLMS) algorithm

The MPNLMS algorithm was proposed to improve the convergence of PNLMS. This is achieved by computing the *optimal* proportionate step-size during the adaptation process. The MPNLMS algorithm was derived such that all coefficients attain converged values to within a vicinity ϵ of their optimal values in the same number of iterations [11]. The definition for $\kappa_l(n-1)$ of MPNLMS is differed from that of previous PNLMS, as follows

$$\kappa_l(n-1) = max\left\{\kappa_{min}(n), F\left(\left|\hat{h}_l(n-1)\right|\right)\right\}, \qquad 0 \le l \le L-1 \tag{2.8}$$

where $\kappa_{min}(n-1) = \rho \times max\{\, \gamma, |\hat{h}_0(n-1)|, |\hat{h}_1(n-1)|, \cdots, |\hat{h}_{L-1}(n-1)|\, \}$ (2.9)

$$F(|\hat{h}_l(n-1)|) = \ln(1 + \nu . |\hat{h}_l(n-1)|) \ , \qquad 0 \le l \le L-1 \qquad (2.10)$$

with $\nu = 1/\epsilon$ and ϵ (vicinity) is a very small positive number chosen as a function of the noise level [11]. It has been shown in [11] that $\epsilon = 0.001$ is a good choice for typical echo cancellation, as the echo below -60 dB is negligible. Therefore, we can choose $\nu = 1000$. The positive bias of 1 in $\ln(1 + \nu . |\hat{h}_l(n-1)|)$ is introduced to avoid numerical instability during the initialization stage when $|\hat{h}_l(0)| = 0,\ \forall l$. The MPNLMS algorithm is given in Table B.5 (Appendix B).

As PNLMS, it has been shown that MPNLMS suffers from a significant degradation in convergence performance when the impulse response is dispersive such as can occur in AIRs [65].

In reality, the acoustic echo in the receiving room does not always follow a same response. The path may vary with time influenced by the *distance* between the loudspeaker and the microphone and due to change in room characteristics, including temperature [48], pressure and movement of the talker. The acoustic characteristics of environment can be evaluated by the reverberation time, which is proportional to the volume of the enclosed space and inversely proportional to the absorption area [49]. In such a time-varying environment, the impulse response may vary over a sufficiently large range that its sparsity could change from sparse to dispersive. Therefore, it is needed to have a *robust* algorithm that can cope with sparseness variations of the acoustic path and maintain its good performance.

2.2. Sparseness-controlled NLMS-based adaptive filtering algorithms

In this section, we present the concept of more flexible approaches proposed to improve the convergence of the classical sparseness-aware (SA) proportionate adaptive algorithms in *dispersive* impulse responses estimation [18], [19]. These algorithms compute a *sparseness measure* of the estimated impulse response at each iteration of the adaptive process and incorporate it into their methods. It is found that the *sparseness-controlled* (SC) algorithms achieve faster convergence for both sparse and dispersive AIRs and are *robust* to the sparseness variation of AIRs. Hence, they are a good effective choice for AEC.

Before starting with the SC-algorithms, we present first a very-used definition of sparseness measure and its characteristics.

2.2.1. Sparseness measure

The degree of sparseness can be qualitatively referred to as a range of *strongly dispersive* to *strongly sparse* [66].The sparseness of an impulse response of length L can be quantified by the sparseness measure [67], [68]

$$\xi(\mathbf{h}) = \frac{L}{L-\sqrt{L}}\left\{1 - \frac{||\mathbf{h}||_1}{\sqrt{L}\,||\mathbf{h}||_2}\right\}, \tag{2.11}$$

where $||\mathbf{h}||_1$ and $||\mathbf{h}||_2$ are the ℓ_1-norm and the ℓ_2-norm respectively. That is

$$||\mathbf{h}||_1 = \sum_{i=0}^{L-1} |h_i| \tag{2.12}$$

$$||\mathbf{h}||_2 = \sqrt{\sum_{i=0}^{L-1} h_i^2} = \sqrt{\mathbf{h}^T\mathbf{h}} \tag{2.13}$$

By considering impulse responses with various degrees of sparseness it can be shown that $0 \leq \xi(\mathbf{h}) \leq 1$.

- *Spareseness of a delta function*

  An impulse of length L is defined by

  $$h_i = \begin{cases} \pm v, & i = n_1 \\ 0, & 0 \leq i \leq L-1\,, i \neq n_1 \end{cases} \tag{2.14}$$

  where n_1 defines the location of the impulse. Using equations (2.12) and (2.13), we find that $||\mathbf{h}||_1 = v$ and $||\mathbf{h}||_2 = \sqrt{v^2}$ from which substituting in equation (2.11) gives the sparseness $\xi(\mathbf{h}) = 1$.

- *Sparseness of a signal with constant magnitude*

  For $\mathbf{h}$ of length L with a constant magnitude $\pm v$, we have $||\mathbf{h}||_1 = L|v|$ and $||\mathbf{h}||_2 = \sqrt{Lv^2}$. Substituting these results into eq. (2.11), the sparseness for $\mathbf{h}$ is then $\xi(\mathbf{h}) = 0$.

It is also interesting to note that the measure is independent of the sorting order of the impulse response coefficients and not affected by a non-zero scaling factor c [43], i.e.:

$$\xi(c\mathbf{h}) = \xi(\mathbf{h}) \qquad \forall c \neq 0 \tag{2.15}$$

Direct use $\xi(\mathbf{h})$ is not feasible since $\mathbf{h}$ is unknown during adaptation. Therefore, $\hat{\xi}(n)$ is employed to estimate the sparseness of an impulse response at each sample iteration [18], [19]. That is

$$\hat{\xi}(n) = \frac{L}{L-\sqrt{L}}\left\{1 - \frac{\|\hat{\mathbf{h}}(n-1)\|_1}{\sqrt{L}\,\|\hat{\mathbf{h}}(n-1)\|_2}\right\}, \qquad n \geq L \tag{2.16}$$

which uses the estimation of the impulse response at the iteration ($\hat{\mathbf{h}}(n-1)$), instead of the unknown impulse response $\mathbf{h}$.

2.2.2. Sparseness-controlled improved PNLMS (SC-IPNLMS) algorithm [18]

The proposed SC-IPNLMS algorithm further improves the performance of IPNLMS by emphasizing the proportionate term if the impulse response is significantly sparse. For relatively less sparse impulse responses, SC-IPNLMS allocates a higher weighting to the NLMS term. This can be achieved by expressing the elements of the update control matrix as

$$q_l(n-1) = \left[\frac{\left(1-0.5\hat{\xi}(n)\right)}{L}\right]\frac{(1-\alpha)}{2L} + \left[\frac{\left(1+0.5\hat{\xi}(n)\right)}{L}\right]\frac{(1+\alpha)\left|\hat{h}_l(n-1)\right|}{2\|\hat{\mathbf{h}}(n-1)\|_1 + \delta_{SC-IPNLMS}},$$
$$0 \leq l \leq L-1 \tag{2.17}$$

The weighting of 0.5 included in eq. (2.17) is chosen empirically in order to balance the performance between sparse and dispersive cases, which could be further optimized for a specific application [43]. In addition, normalization by L is introduced to reduce significant coefficient noise when the effective step-size is large for sparse AIRs with high $\hat{\xi}(n)$.

It is found in [18] that, for dispersive AIRs, SC-IPNLMS allocates a uniform step-size across $\hat{h}_l(n)$ while, for sparse AIRs, the algorithm distributes $q_l(n)$ proportionally to the magnitude of the coefficients. Consequently, the SC-IPNLMS algorithm varies the degree of NLMS and proportionate adaptations according to the sparseness nature of the AIRs while, in the standard IPNLMS, the mixing coefficient α in eq. (2.6) is fixed *a priori*. The SC-IPNLMS algorithm is described in Table B.6 (Appendix B).

2.2.3. Effect of the parameter ρ on step-size control matrix $\mathbf{Q}$ for PNLMS [19]

In subsection 2.1.1, the parameter ρ in eq. (2.5) was originally introduced to prevent freezing of the filter coefficients when they are much smaller than the largest coefficient.

Results presented in [19] showed that a higher value of ρ will reduce the degree of proportionality due to the $\mathbf{Q}$ matrix meaning that all filter coefficients are updated at a more uniform rate. This gives a good convergence performance of PNLMS when the AIR is dispersive. On the other hand, a lower value of ρ will increase influence of the $\mathbf{Q}$ matrix, hence, giving a good convergence performance for a sparse AIR. As a consequence of this important observation, it was proposed in [19] to incorporate $\hat{\xi}(n)$ into ρ for both PNLMS and MPNLMS.

2.2.4. The SC-PNLMS and SC-MPNLMS algorithms

In order to overcome the problem of slow convergence in dispersive AIRs, the PNLMS and MPNLMS algorithms need to have step-size control elements $q_l(n)$ robust to sparseness variations of the impulse response. In order to achieve a high ρ when $\hat{\xi}(n)$ is small (dispersive system), several choices can be employed. A very-used choice is the exponential-function form [19] as

$$\rho(n) = e^{-\lambda\hat{\xi}(n)}, \qquad \lambda \in \mathbb{R}^{+} \tag{2.18}$$

Replacing ρ by $\rho(n)$ in the PNLMS update equation gives the sparseness-controlled PNLMS algorithm (SC-PNLMS). Tests discussed in [19] showed that $\lambda = 6$ gives a good compromise of convergence performance in dispersive and sparse systems. Moreover, the range of $4 \leq \lambda \leq 6$ could be considered as a good choice for the application of AEC.

Incorporating $\rho(n)$ in a similar manner for the MPNLMS algorithm gives the sparseness-controlled MPNLMS algorithm (SC-MPNLMS) which inherits more of the MPNLMS properties when the estimated AIR is sparse and distributes uniform step-size across $\hat{h}_l(n)$, as in NLMS, when the estimated AIR is dispersive [19].

In addition, it can be noted that when $n = 0$, $\left\| \hat{\mathbf{h}}(0) \right\|_2 = 0$ and hence, to prevent division by a small number or zero, $\hat{\xi}(n)$ can be computed for $n \geq L$ in both SC-PNLMS and SC-MPNLMS. When $n < L$, we can set $\rho(n) = \rho = 5/L$ [10].

The SC-PNLMS and SC-MPNLMS algorithms are summed up in Tables B.7 and B.8 respectively (Appendix B).

It should be addressed that most of the mentioned NLMS-algorithm improvements result in a significant increase of the computational complexity. In the next section, we present some more-recent sparse algorithms that have a little bit less computational complexity. These algorithms have been emerged from the concept of *compressed sensing*.

2.3. Compressed-sensing NLMS-based adaptive filtering algorithms

First and foremost, *compressed sensing* (CS) is known also by the names of *compressive sensing* and *compressed* or *compressive sampling* or *sparse sampling* [69]. It is a very useful concept when dealing with limited and redundant data. It has flourished exceptionally fast over the past few years [24] and has been extensively studied and applied in the following domains: medical image processing [70], compression [71], coding and machine learning including face recognition, detection and tracking of objects in video [72, 73, 74], sensor networks [75] and cognitive radio.

2.3.1. What’s compressed sensing?

Compressed sensing is an umbrella term for the methodologies and concepts involved in reconstructing compressed representations of mathematical objects using limited amount of data, typically much less than the objects’ ambient dimension (i.e., the object dimension when it is uncompressed). For more details about the CS-theory and its applications, see references [24] and [69].

Since the computation of a sparse representation for a signal is a non-deterministic polynomial time (NP)-hard problem [76], [77], the LASSO method [29] uses ℓ_1-norm minimization to obtain sparse signal reconstruction and provide the guarantee for convergence. However, this method suffers from slower convergence rate compared to the MP method [25]-[27] which employs a greedy algorithm for faster convergence but it has lesser guarantee of achieving a sparse representation.

2.3.2. Zero-attracting NLMS (ZA-NLMS) algorithm

The use of LASSO method permitted to introduce two different sparsity constraints (the ℓ_1-norm and the log-sum penalty function) into the convex quadratic cost function of the LMS algorithm, resulting in two sparsity-aware LMS algorithms, namely, the *zero-attracting* LMS (ZA-LMS) and the *reweighted zero-attracting* LMS (RZA-LMS) [30]. These algorithms obey the following [24] updating scheme

$$\begin{Bmatrix} new \\ parameter \\ estimate \end{Bmatrix} = \underbrace{\underbrace{\begin{Bmatrix} old \\ parameter \\ estimate \end{Bmatrix} + \{stepsize\} \begin{Bmatrix} new \\ information \end{Bmatrix}}_{\text{LMS}} + \begin{Bmatrix} zero \\ attraction \\ term \end{Bmatrix}}_{\text{Sparse LMS}}$$

where the new information term is the error vector between the outputs of the filter and the desired signal vector. The *Zero-Attraction* (ZA) term (or zero attractor) is a norm-related regularization function which applies an attraction to zero on small parameters [24].

In order to exploit the system sparsity in time domain, the cost function of ZA-LMS [30] is given by

$$\mathcal{J}_{ZA-LMS}(n) = \frac{1}{2}e^2(n) + \lambda_{ZA}\left\| \hat{\mathbf{h}}(n)\right\|_1 \qquad (2.19)$$

which means that the sparse cost function $\mathcal{J}_{ZA-LMS}(n)$ combines the instantaneous error $e(n)$ with a sparseness-inducing penalty term ($\lambda_{ZA}\left\| \hat{\mathbf{h}}(n)\right\|_1$) [24]. Where λ_{ZA} is a regularization parameter to balance the estimation error and sparse penalty of $\hat{\mathbf{h}}(n)$. The corresponding updated equation [24] of ZA-LMS is

$$\begin{aligned} \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + \mu[-\nabla\mathcal{J}_{ZA-LMS}(n)] \\ &= \hat{\mathbf{h}}(n-1) + \mu.\mathbf{x}(n)e(n) - \rho_{ZA}.\operatorname{sgn}\left(\hat{\mathbf{h}}(n-1)\right) \qquad (2.20) \end{aligned}$$

where $\rho_{ZA} = \mu\lambda_{ZA}$ is referred to as the *zero-attraction controller* or the *regularization step-size* in the adaptive filtering context. It controls the strength of the zero-attractor ($-\rho_{ZA}.\operatorname{sgn}\left(\hat{\mathbf{h}}(n-1)\right)$) [9]. Usually the regularization step-size is fine tuned offline (via exhaustive simulations) or in an ad–hoc manner [24].

In [32]-[34], a systematic approach is used. It expresses ρ_{ZA} in terms of the noise level. In [32], the input signal power E_0 is set equal to unity (i.e. $E_0 = 1$), which makes the noise power $\sigma_n^2 = 10^{-\mathrm{SNR}/10}$, where SNR ($10\log_{10}(E_0/\sigma_n^2)$) is the input signal-to-noise ratio. Note that $\mathrm{sgn}(.)$ is a component-wise *signum function* defined as

$$\mathrm{sgn}(\,h\,) = \begin{cases} 1\ , & h > 0 \\ 0\ , & h = 0 \\ -1\ , & h < 0 \end{cases} \qquad (2.21)$$

Observing the update equation (2.20), its second term attracts small-value filter coefficients to zero in high probability. In other words, most of the small-value filter coefficients can be replaced by zero. This will speed up the convergence and mitigate the noise on zero positions as well. To be more practical and facilitate the choice of the step-size, an improved algorithm was proposed (i.e., ZA-NLMS) [32], [78]. The update equation of ZA-NLMS [32] was proposed as follows

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{ZA}.\mathrm{sgn}\left(\hat{\mathbf{h}}(n-1)\right) \qquad (2.22)$$

The ZA-NLMS algorithm is presented in Table B.9 (Appendix B).

2.3.3. Reweighted zero-attracting NLMS (RZA-NLMS) algorithm

It is found in [30] that the shrinkage in the ZA-LMS does not distinguish between zero taps and non-zero taps. Since all the taps are forced to zero uniformly, its performance would deteriorate for less sparse systems. Therefore, in [30], they used the *reweighted* ℓ_1-norm minimization recovery algorithm [79] and proposed a heuristic approach to reinforce the zero-attractor. The resultant algorithm was termed as the reweighted ZA-LMS (or RZA-LMS). The cost function of RZA-LMS is expressed as

$$\mathcal{J}_{RZA-LMS}(n) = \frac{1}{2}e^2(n) - \lambda_{RZA}\sum_{i=1}^{L} log(1 + \varepsilon_{RZA}|h_i(n)|) \qquad (2.23)$$

where λ_{RZA} is the regularization parameter and $\varepsilon_{RZA} > 0$ is the positive *threshold*. According to the stochastic gradient approach [24], the resulting filter update iteration is

$$\hat{h}_i(n) = \hat{h}_i(n-1) + \mu . x(n-i)e(n) - \rho_{RZA} \frac{\operatorname{sgn}\left(\hat{h}_i(n-1)\right)}{1 + \varepsilon_{RZA}\left|\hat{h}_i(n-1)\right|} \qquad (2.24)$$

where $0 \leq i \leq L-1$. Equation (2.24) is written also in the vector form [24] as

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu . \mathbf{x}(n)e(n) - \rho_{RZA} \frac{\operatorname{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1 + \varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|} \qquad (2.25)$$

where $\rho_{RZA} = \mu\lambda_{RZA}\varepsilon_{RZA}$ and $\left|\widehat{\mathbf{h}}(n-1)\right|$ is an $L \times 1$ vector with elements that are absolute values of the elements of the vector $\widehat{\mathbf{h}}(n-1)$. The term $\frac{\operatorname{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1+\varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|}$ is obtained by L element-by-element divisions.

The RZA-LMS selectively shrinks taps with large magnitudes and the ones with small magnitudes. The reweighted zero-attractor (third term in the update equation (2.25)) takes significant effect only on those taps whose magnitudes are comparable to $1/\varepsilon_{RZA}$; and there is a little shrinkage exerted on the taps whose $|h_i(n-1)| \gg 1/\varepsilon_{RZA}$ [30].

In [32] and [78], the same analogy is applied further to RZA-LMS in order to obtain the RZA-NLMS algorithm that has the following update equation

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{RZA} \frac{\operatorname{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1 + \varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|} \qquad (2.26)$$

Observing equation (2.26), if ε_{RZA} is very small, $\varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|$ tends to zero and the update equations of the RZA-NLMS and the ZA-NLMS algorithms become equivalent [80]. The RZA-NLMS algorithm is given in Table B.10 (Appendix B).

2.3.4. The effect of $\rho_{(R)ZA}$ (ρ_{ZA} or ρ_{RZA})

Regularization plays a fundamental role in adaptive filtering. However, the better performance is not obtained if the regularization step-size is not chosen properly. According to equations (2.22) and (2.26), a large $\rho_{(R)ZA}$ results in a faster convergence since the intensity of attraction increases as $\rho_{(R)ZA}$ increases. However, steady-state misalignment increases as $\rho_{(R)ZA}$ increases (means that the estimation accuracy decreases as $\rho_{(R)ZA}$ increases). Therefore, the parameter $\rho_{(R)ZA}$ is determined by the trade-off between adaptation speed and adaptation quality in particular applications [80]. In applications that require high estimation accuracy, $\rho_{(R)ZA}$ should be chosen small enough and the performance of the RZA-NLMS approaches to that of the ZA-NLMS.

2.3.5. The effect of ε_{RZA}

Observing equation (2.26), if ε_{RZA} is very small, $\varepsilon_{RZA}\left|\hat{\mathbf{h}}(n-1)\right|$ tends to zero and the update equations of the RZA-NLMS and the ZA-NLMS algorithms become equivalent [80]. Since ε_{RZA} is in the denominator part of the third term of the RZA-NLMS update equation (2.26), a larger ε_{RZA} will make the step-size in the reweighted-zero-attractor term smaller. Therefore, the convergence rate becomes slower and the steady-state level decreases due to the relatively smaller [80] step-size.

The conventional ZA-NLMS and RZA-NLMS algorithms are called also the invariable step-size (μ) ZA-NLMS (or ISS-ZA-NLMS) and ISS-RZA-NLMS respectively. This nomenclature is to differentiate them from the variable step-size ones (VSS-ZA-NLMS and VSS-RZA-NLMS).

2.3.6. Variable step-size ZA-NLMS (VSS-ZA-NLMS) algorithm

Although the variable step-size NLMS (VSS-NLMS) algorithm was proposed for AEC to improve the estimation accuracy [57], system sparsity has not been considered in the VSS-NLMS algorithm. The VSS-NLMS algorithm is adaptive to the estimation error in each iteration, i.e., large step-size is used in the case of large estimation error to accelerate the convergence speed, while small step-size is used when the estimation error is small to improve the steady-state estimation accuracy [34].

Recently, by jointly taking advantage of system sparsity and VSS-NLMS, two improved adaptive sparse channel estimation (ASCE) algorithms have been proposed in [33], [34]. They are named as variable step-size zero-attracting NLMS (VSS-ZA-NLMS) and VSS reweighted ZA-NLMS (VSS-RZA-NLMS).

Based on the observation that large step-sizes are preferred to achieve fast convergence while small step-sizes are preferred for accurate estimation, the proposed VSS-ZA-NLMS and VSS-RZA-NLMS algorithms replace the ISS by VSS in conventional NLMS-based algorithms in order to improve the adaptive sparse channel estimation in terms of bit error rate (BER) and mean square error (MSE) metrics [34]. The update equation for the VSS-ZA-NLMS algorithm [33] is expressed as

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{ZA}.\,\mathrm{sgn}\left(\widehat{\mathbf{h}}(n-1)\right) \qquad (2.27)$$

where $\mu(n)$ is calculated as explained in subsection 1.5.3.

In the case of small step-sizes, good estimation accuracy is achieved while high convergence speed is obtained for large step-sizes. Analysis presented in [33] showed that the value of the VSS $\mu(n)$ will increase if the estimation error decreases and vice versa. In view of that, as the updating error decreases, VSS-ZA-NLMS reduces its step-size adaptively to ensure the algorithm stability as well as to achieve better steady-state estimation performance [33]. The main steps of the VSS-ZA-NLMS algorithm are listed in Table B.11 (Appendix B).

2.3.7. Variable step-size reweighted ZA-NLMS (VSS-RZA-NLMS) algorithm

The same philosophy, as in subsection 2.3.6, was extended further to the case of RZA-NLMS (see subsection 2.3.3) in order to obtain the VSS-RZA-NLMS with the following update equation [34]

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{RZA}\frac{\mathrm{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1 + \varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|} \qquad (2.28)$$

The algorithm VSS-RZA-NLMS is summarized in Table B.12 (Appendix B).

In the next chapter, we conduct simulations to test the behavior of the discussed algorithms in different-sparsity synthetic and real systems for stationary and non-stationary inputs.

2.5. Computational complexity

The computational complexity of an algorithm is a very important criterion that should be examined since it has a direct relationship with the hardware implementation and the operating time of the system. Although many factors contribute to the complexity of an algorithm, the relative complexity of NLMS, PNLMS, SC-PNLMS, IPNLMS, SC-IPNLMS, MPNLMS, SC-MPNLMS, ZA-NLMS, RZA-NLMS, VSS-ZA-NLMS and VSS-RZA-NLMS in terms of the total number of additions, multiplications, divisions and logarithms (Log) per iteration is assessed in Table 2.1.

The followings should be noted:

- The computation of the ℓ_2-norm ($\|\mathbf{x}(n)\|_2^2 = \mathbf{x}^{\boldsymbol{T}}(n)\mathbf{x}(n) = p_x(n)$, where $\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$) can be updated recursively using an exponential window [81]. That is;

$$p_x(n) = \tau . p_x(n-1) + (1-\tau) . L . x^2(n) \tag{2.29}$$

  where τ is a forgetting factor and L is the filter length. This method requires only four multiplications and two additions.
- The error update equation $e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$ requires L additions and L multiplications.
- The comparison between two numbers takes one subtraction. But, in this content, comparison is regarded as an operator.
- The additional complexity of the SC-algorithms, on top of their conventional SA method, arises from the computation of the sparseness measure $\hat{\xi}(n)$. Given that $(L/L-\sqrt{L})$ in eq. (2.16) can be computed offline, the remaining ℓ-norms require an additional $2L$ additions and L multiplications. Therefore, the total computation of $\hat{\xi}(n)$, requires $2L+1$ additions, $L+2$ multiplications and 1 division.
- The SC-PNLMS and SC-MPNLMS algorithms additionally require computations for eq. (2.18). Alternatively, a look-up table with values of $\rho(n)$ defined in eq. (2.18) can be computed for $0 \leq \hat{\xi}(n) \leq 1$ [43].
- Since $\left\|\widehat{\mathbf{h}}(n)\right\|_1$ computation is already available from IPNLMS in eq. (2.6), SC-IPNLMS only requires an additional $L+3$ additions, $L+6$ multiplications and 1 division.
- Because of the zero-attractor term $(-\rho_{ZA}\mathrm{sgn}\left(\widehat{\mathbf{h}}(n-1)\right))$, the ZA-NLMS algorithm has an additional L additions with respect to NLMS.

- Because of the reweighted-zero attractor term $(-\rho_{RZA}\frac{\mathrm{sgn}\left(\hat{h}_i(n-1)\right)}{1+\varepsilon_{RZA}\left|\hat{h}_i(n-1)\right|})$, the RZA-NLMS algorithm requires more additional L additions, L multiplications and L divisions with respect to ZA-NLMS.
- For VSS-algorithms, the increase of computational complexity is due to the calculation of the variable step-size $\mu(n)$. That is;

$$\begin{cases} p_x(n) = \tau.p_x(n-1) + (1-\tau).L.x^2(n) \\ \boldsymbol{p}(n) = \beta\boldsymbol{p}(n-1) + (1-\beta)\dfrac{\mathbf{x}(n)e(n)}{p_x(n)} \\ \mu(n) = \mu_{max}\dfrac{\boldsymbol{p}^T(n)\boldsymbol{p}(n)}{\boldsymbol{p}^T(n)\boldsymbol{p}(n)+C} \end{cases}$$

which requires an additional $2L+3$ additions, $3L+6$ multiplications and 2 divisions more than the ISS-algorithms.

Table 2.1 gives the computational-complexity values concerning the filtering-part of algorithms. For example, for NLMS algorithm, we have:

$$\begin{cases} e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1) \\ p_x(n) = \tau.p_x(n-1) + (1-\tau).L.x^2(n) \\ \hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\dfrac{\mathbf{x}(n)e(n)}{p_x(n)+\delta_{NLMS}} \end{cases}$$

which requires $2L+3$ additions, $2L+5$ multiplications and 1 division. By the same analogy, we continue for the other algorithms and the results are summarized in Table 2.1.

Table 2.1: Complexity of algorithms of interest in terms of: addition, multiplication, division and logarithm (Log).

Algorithm	Addition	Multiplication	Division	Log
NLMS	$2L$+3	$2L$+5	1	0
PNLMS	$3L$+1	$6L$+4	2	0
SC-PNLMS	$5L$+2	$7L$+6	3	0
IPNLMS	$4L$+2	$6L$+4	2	0
SC-IPNLMS	$5L$+5	$7L$+10	3	0
MPNLMS	$4L$+1	$7L$+4	2	L
SC-MPNLMS	$6L$+2	$8L$+6	3	L

ZA-NLMS	$3L$+3	$2L$+5	1	0
VSS-ZA-NLMS	$5L$+6	$5L$+11	3	0
RZR-NLMS	$4L$+3	$3L$+5	L+1	0
VSS-RZA-NLMS	$6L$+6	$6L$+11	L+3	0

It can be seen from Table 2.1 that the overall computational complexities of the discussed sparse NLMS-based algorithms are increased compared to NLMS. To compensate these increased complexities, their convergence performances must be efficiently higher. Consequently, the trade-off between complexity and performance depend on the design choice for a particular application.

CHAPTER 3

SIMULATION, RESULTS AND DISCUSSION

3.1. Synthetic generation of sparse acoustic impulse responses

In this chapter, we presented some simulation results to demonstrate the performance of the interested algorithms in our study. They are first tested for *synthetic* systems where the degree of sparseness is controlled more easily, then we further for real systems. The method proposed in [18] provides a means of generating synthetic impulse responses (Synthetic IRs) with different degrees of sparsity using random sequences. This can be achieved by first defining an $L \times 1$ vector $\mathbf{u}$ (where L is the length of the impulse response) as

$$\mathbf{u}_{L\times 1} = \left[\mathbf{0}_{L_p\times 1} \quad 1 \quad e^{-1/\psi} \quad e^{-2/\psi} \quad \ldots \quad e^{-(L_u-1)/\psi}\right]^{\boldsymbol{T}} \tag{3.1}$$

where the L_p leading zeros models the length of the bulk delay and $L_u = L - L_p$ is the length of the decaying window while $\psi \in \mathbb{Z}^+$ is the decay constant. If we define an $L_u \times 1$ vector $\mathbf{b}$ as a zero-mean white Gaussian noise (WGN) sequence with variance σ_b^2, we can express an $L \times 1$ synthetic impulse response as

$$\mathbf{h}(n) = \begin{bmatrix} \mathbf{0}_{L_p\times L_p} & \mathbf{0}_{L_p\times L_u} \\ \mathbf{0}_{L_u\times L_p} & \mathbf{B}_{L_u\times L_u} \end{bmatrix} \mathbf{u} + \mathbf{p'} \tag{3.2}$$

where $\mathbf{B}_{L_u\times L_u} = diag\{\mathbf{b}\}$ and the $L \times 1$ vector $\mathbf{p'}$ ensures elements in the inactive region are small but non-zero and is an independent zero-mean WGN sequence with variance $\sigma_{p\prime}^2$.

3.2. Computer-simulations setup

All simulations were performed in floating-point representation using MATLAB® software. We used three different types of input signals, sampled at 16 KHz, and filtered by different-sparsity synthetic and real impulse responses to obtain the desired signals. Firstly, we used different-sparsity synthetic acoustic impulse responses generated using the approach described in section 3.1 (see figures 3.1, 3.2 and 3.3), then we used two acoustic real impulse responses in different sparsity levels; one is measured in a car enclosure (see Figure 3.4) where the other is measured in a real audio-conference (denoted ACN) room (see Figure 3.5).

In order to use our two real systems in different sparsity levels, the car system is used truncated at the first 256 taps to give ξ =0.5138 (less sparse), then with all of its 1024 taps which gives ξ =0.7410 (more sparse) where the 'long' ACN system is used truncated at the first 2048 taps to obtain ξ =0.3673 (less sparse) and with all of its 8192 taps which obtains ξ =0.6199 (more sparse).

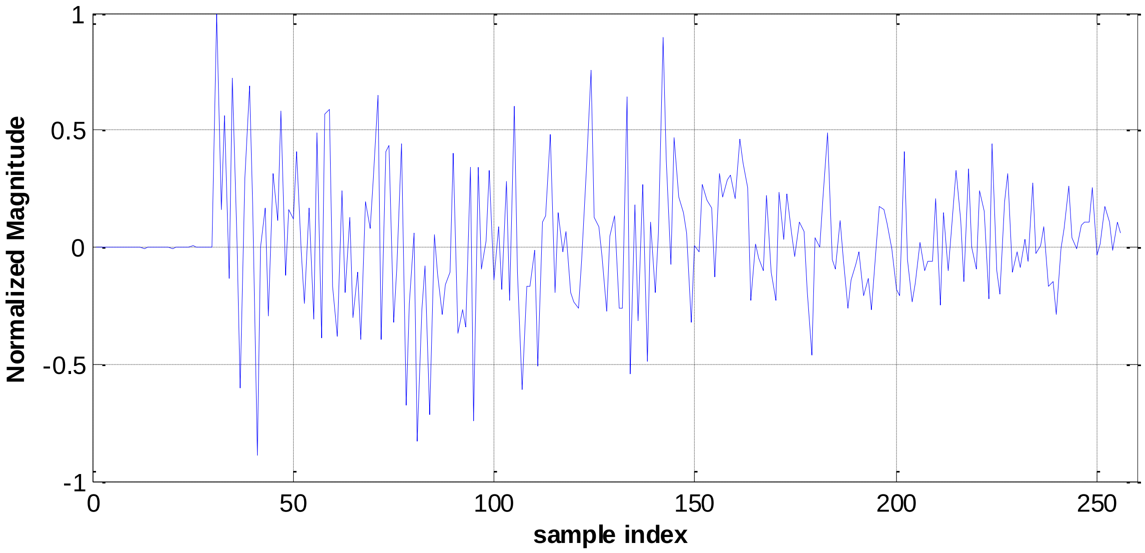

Figure 3.1: Synthetic impulse response with length L=256, the bulk delay length L_p= 30, ψ =160 and ξ =0.3028 (non-sparse or dispersive).

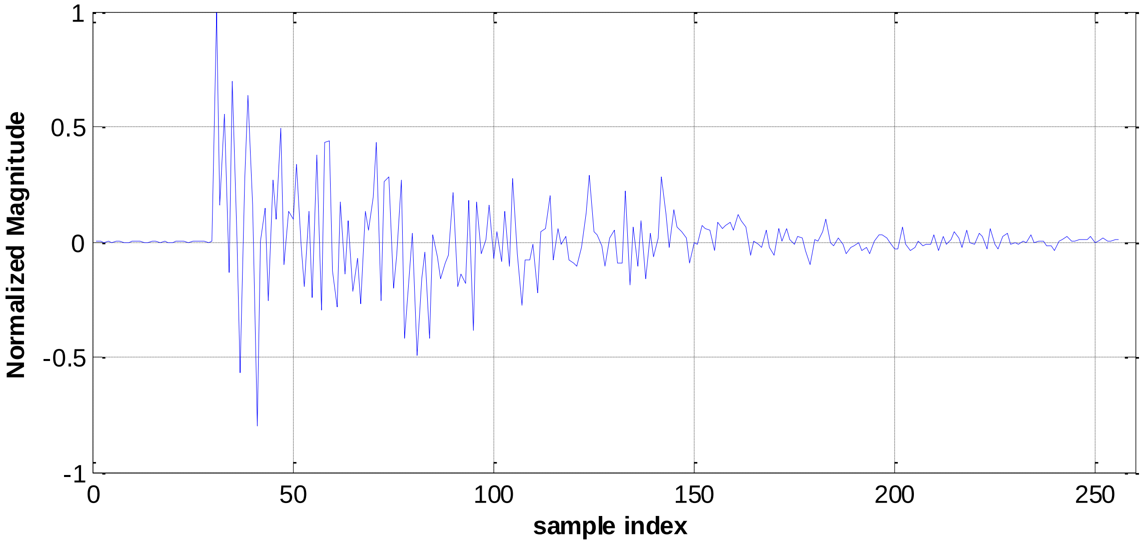

Figure 3.2: Synthetic impulse response with length L=256, the bulk delay length L_p= 30, ψ =60 and ξ =0.4743 (less sparse).

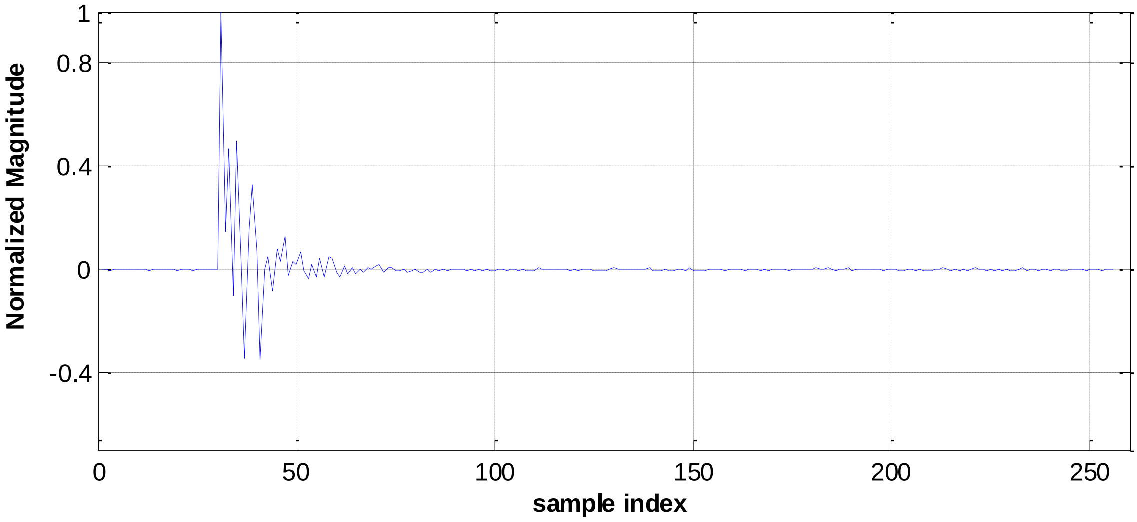

Figure 3.3: Synthetic impulse response with length L=256, the bulk delay length L_p= 30, ψ =10 and ξ =0.8296 (very sparse).

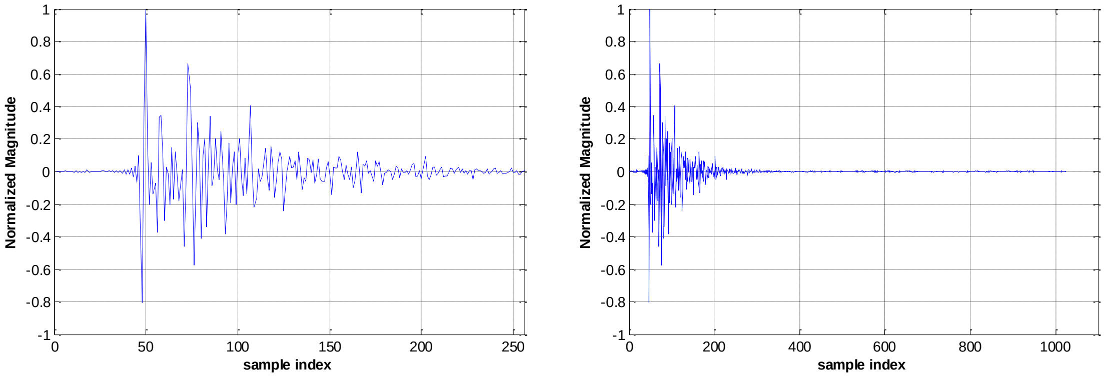

Figure 3.4: (Left) Car impulse response with length L=256 and ξ =0.5138 (less sparse). (Right) Car impulse response with L=1024 and ξ =0.7410 (more sparse).

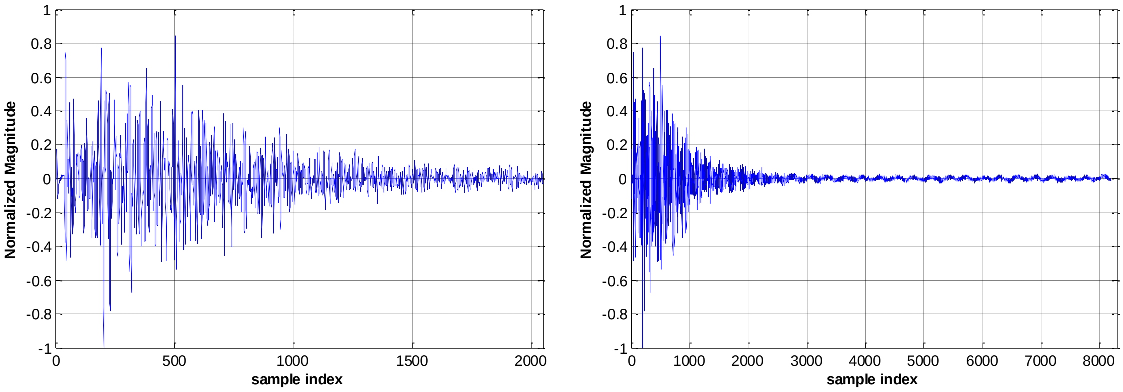

Figure 3.5: (Left) ACN impulse response with length L=2048 and ξ =0.3673 (less sparse). (Right) ACN impulse response with L=8192 and ξ =0.6199 (more sparse).

Concerning inputs, the first one used is a stationary zero-mean correlated noise, with a spectrum equivalent to the average spectrum of speech. It is usually called USASI (USA Standards Institute, now ANSI) [82] noise in the field of acoustic echo cancellation. Its spectral dynamic range is 29 dB. Figure 3.6 shows the used USASI noise normalized on its maximum.

The second used input signal is a simulated auto-regressive (AR) stationary process of order 20. This AR(20) model was obtained from a linear prediction analysis of the French word “UN” . Its frequency spectrum is represented in Figure 3.7 where we can see four main peaks. The spectral dynamic range of the AR input signal is about 40 dB.

Thirdly, and since in real situations the input signal is non-stationary, we use a long speech signal that was obtained by concatenation of a man voice and a woman voice in the same sequence (see Figure 3.8). The estimated spectral dynamic range for this signal is 40 dB.

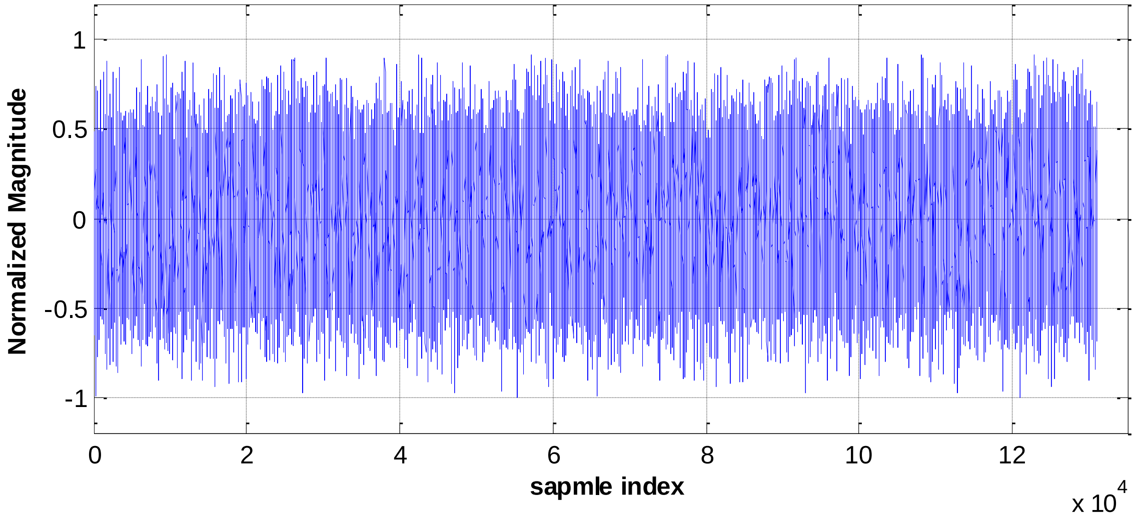

Figure 3.6: The used USASI noise (input signal). Its length is 131072 samples.

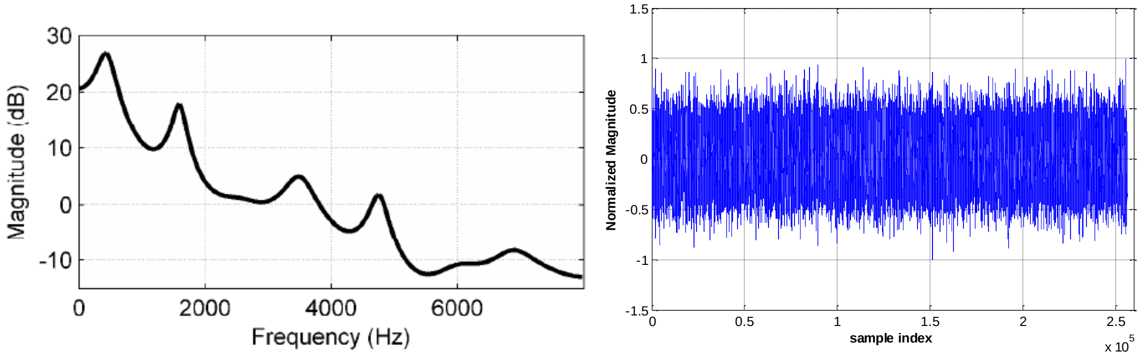

Figure 3.7: The used WGN AR20-model (input signal): (Left) in the frequency domain [44] and (Right) in the time domain. Its length is 256000 samples.

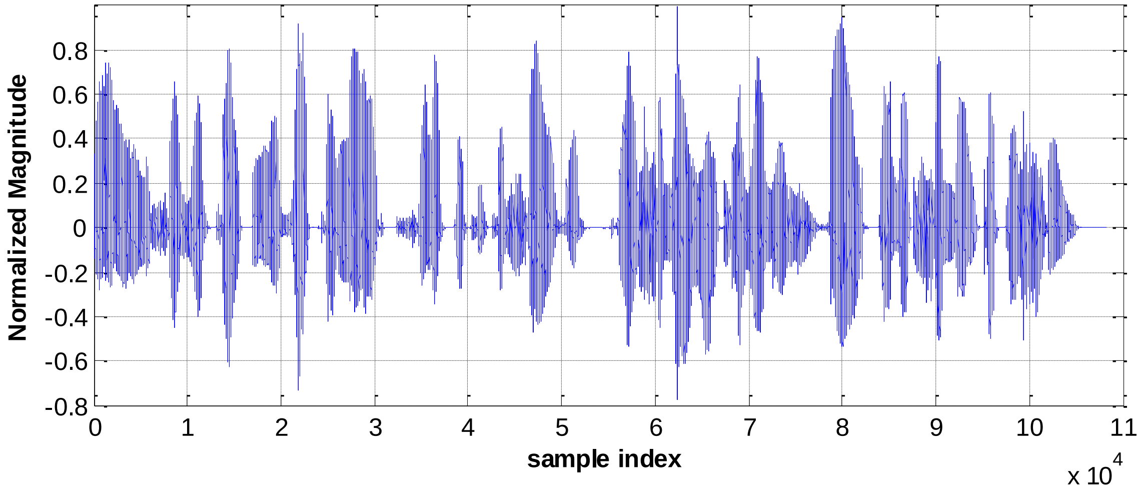

Figure 3.8: The used SPEECH input signal with length of 108208 samples.

The followings should be noted:

- For the sake of uniformity and in order to avoid operations on big numbers, the input and output signals are normalized to their maxima during simulations.
- Many parameters that are usually determined via exhaustive-simulations or chosen in an ad-hoc manner are chosen in a similar way as used in simulations of their relative original published papers.
- To be more practical, for most cases, we added a white-noise to the desired signal with a signal-to-noise ratio (SNR) equals to 50dB as used in [44], which gives a good quality of the signal.
- The step-size was chosen μ=0.3 as in [18], [19], [43]. This value gives a good trade-off between convergence speed and estimation accuracy.
- For IPNLMS, we took α=-0.5 because it leads IPNLMS to behave better than both NLMS and PNLMS in most cases [39].
- In order to compare the re-convergence behavior, we made an abrupt change (a jump) in the impulse response by multiplying the desired signal by 1.5 at a chosen instant.
- As explained in subsection 1.6.1, for stationary-input signals, we used the time-average (over blocks of 256 samples for USASI-noise and 1024 samples for WGN-AR20) normalized MSE (NMSE) learning curve where for the speech input signal (non-stationary), we preferred using the time-average MSE (over blocks of 256 samples).
- For the sake of clarification, the corresponding simulation details are given in the title of each obtained figure.

3.3. Comparison-criteria setup

To compare algorithms belonging to the same family, we kept the similarity of the main ‘stem’ parameters but we were free in the selection or optimization of the additional control parameters because they represent a distinguishing feature that should be well exploited for the newer algorithm versions in order to compensate their increase in the computational complexity.

To compare algorithms of different frameworks (e.g. CS-based and PNLMS-type versions), a systematic used procedure is designing them to achieve approximately the same final steady-state MSE-level as the NLMS algorithm which is taken as reference. Then, we feel free in the selection of their parameters and investigate if there is any superiority of one algorithm with respect to the other.

3.4. Comparison between classical sparsity-aware NLMS-based algorithms

We have made tests using three synthetic impulse responses of different-sparsity levels with USASI-noise input (see figures 3.9). Then we used two real car impulse responses with USASI-noise input. One with length L=256 (less sparse), and the other with L=1024 (more sparse), see figures 3.10.

Investigating these figures, we observe that both PNLMS and MPNLMS have faster *initial* convergence than NLMS but followed by a second phase of slower convergence rate and worse steady-state performance with respect to NLMS when the impulse response is dispersive. This is because when the unknown system is dispersive, $\kappa_l(n-1)$ in eq. (2.4) becomes significantly large for most $0 \leq l \leq L-1$. As a consequence, the denominator of $q_l(n-1)$ in eq. (2.3) is large, giving rise to a small step-size for each large coefficient. This causes a significant degradation in convergence performance for PNLMS and MPNLMS when the impulse response is dispersive.

It is also observed from simulations that, although the IPNLMS algorithm (for α =-0.5) has faster convergence than NLMS and PNLMS regardless of the impulse-response nature (sparse or dispersive) or the input nature (stationary or non-stationary), but it does not outperform the MPNLMS algorithm for highly sparse impulse responses. However, MPNLMS is more complex than IPNLMS.

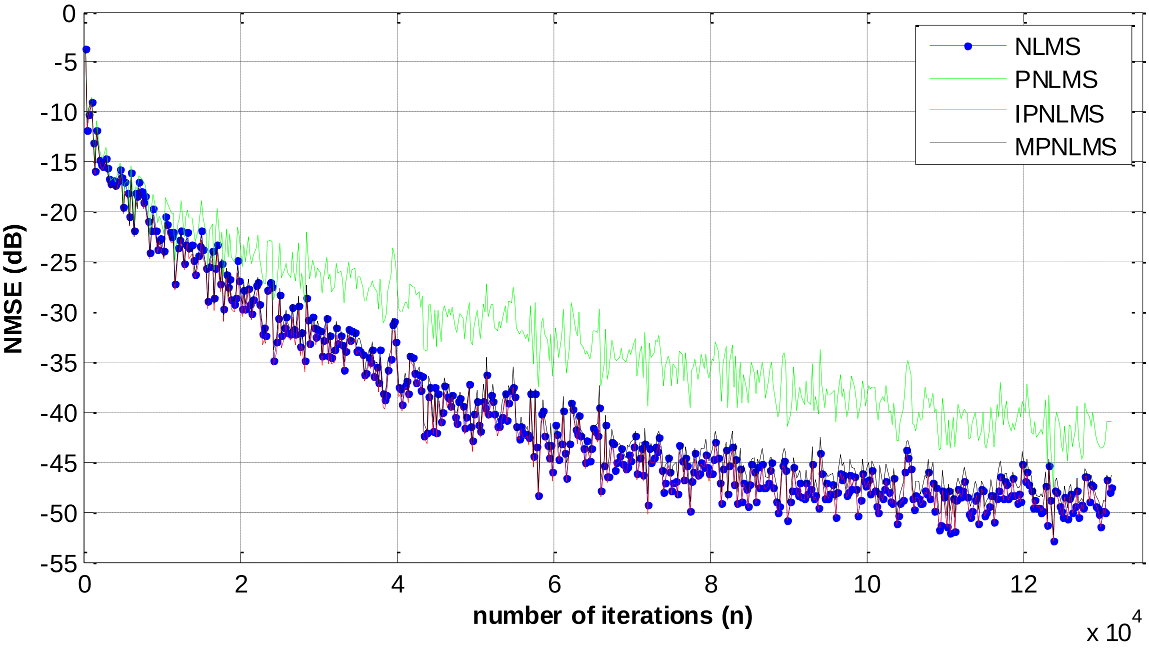

Figure 3.9a: USASI-noise input. Synthetic system with L=256, ψ =160 and ξ =0.3028 (dispersive). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

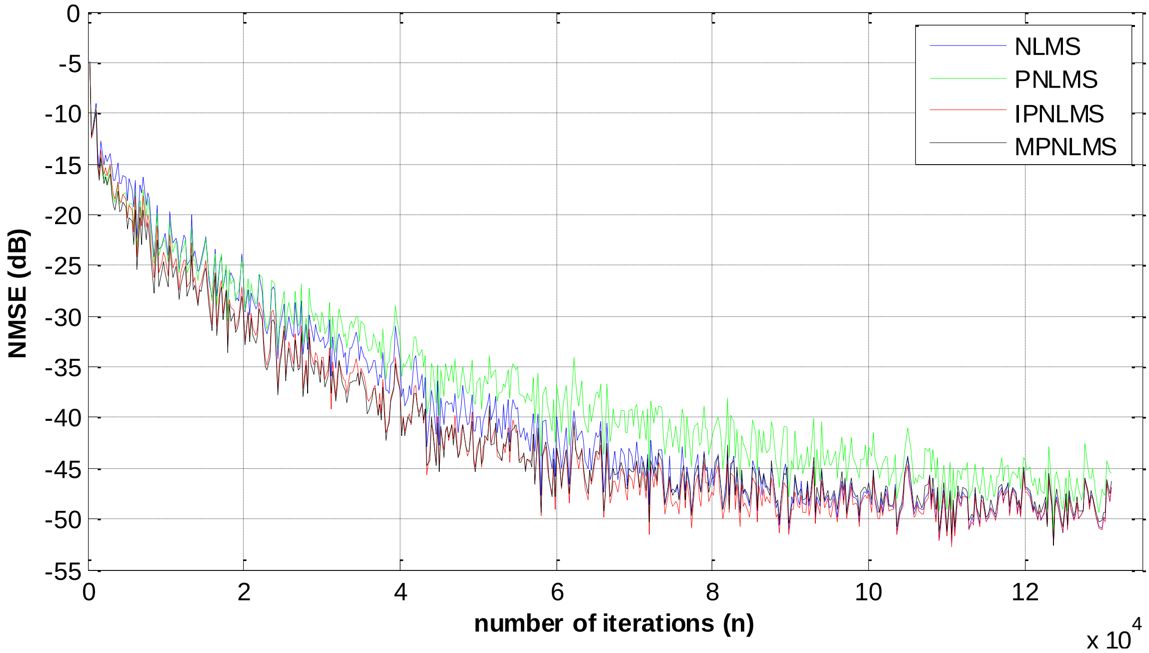

Figure 3.9b: USASI-noise input. Synthetic system with L=256, ψ =60 and ξ =0.4743 (less sparse). NLMS (mu=0.3), PNLMS (mu=0.3), IPNLMS (mu=0.3, α=-0.5) and MPNLMS (mu=0.3). Output with SNR=50dB.

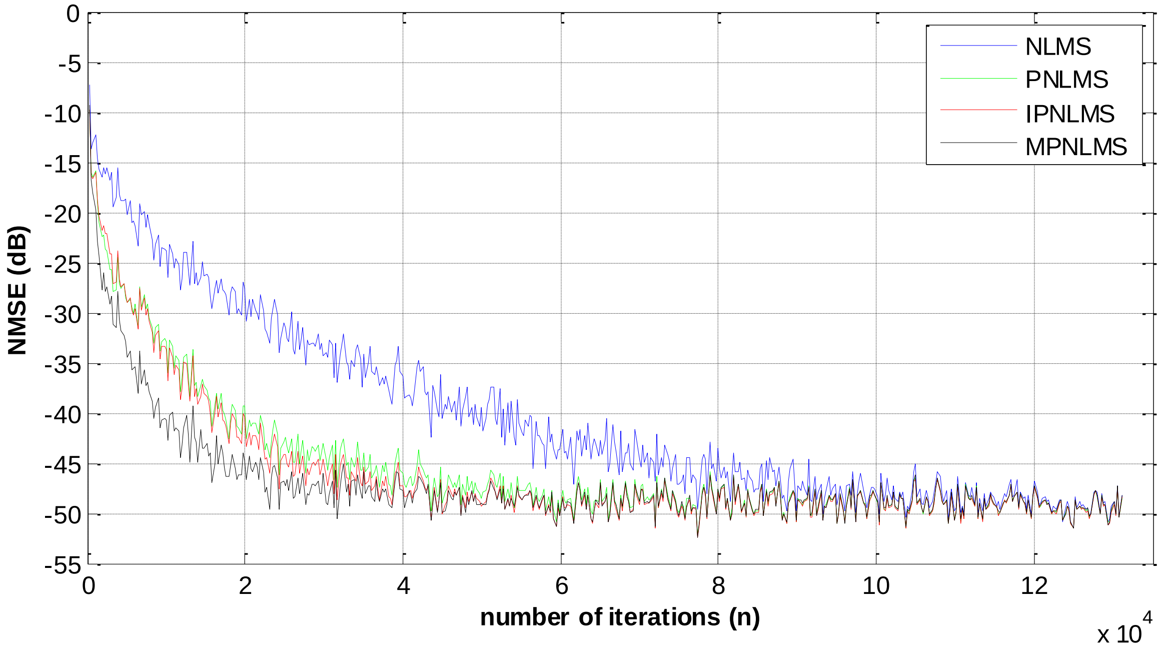

Figure 3.9c: USASI-noise input. Synthetic system with L=256, ψ =10 and ξ =0.8296 (very sparse). NLMS (mu=0.3), PNLMS (mu=0.3), IPNLMS (mu=0.3, α=-0.5) and MPNLMS (mu=0.3). Output with SNR=50dB.

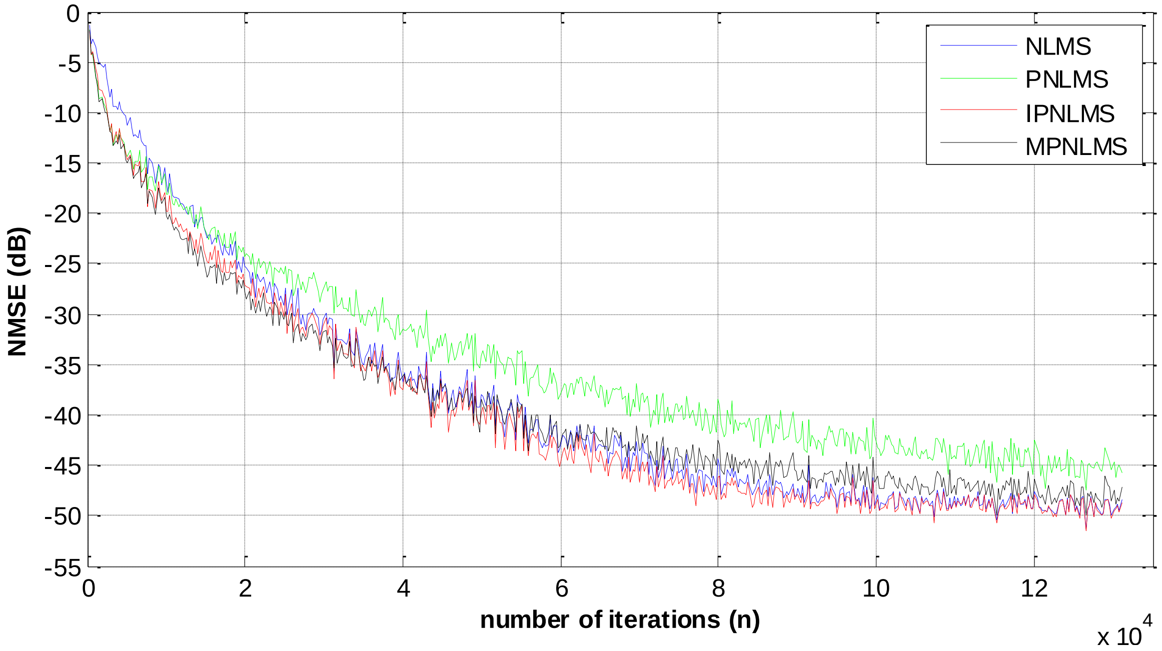

Figure 3.10a: USASI-noise input. Real Car system with L=256 and ξ =0.5138 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

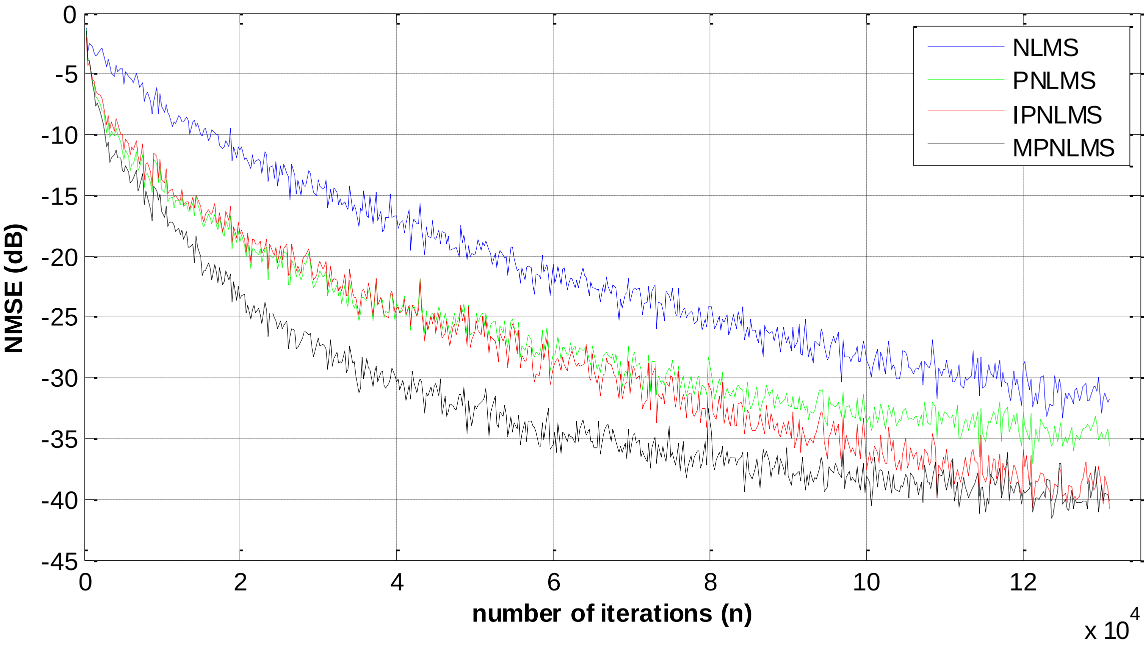

Figure 3.10b: USASI-noise input. Real Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

By making an abrupt change in the real car impulse responses, we obtained the same results concerning the *re-convergence* behavior of algorithms, see figures 3.11.

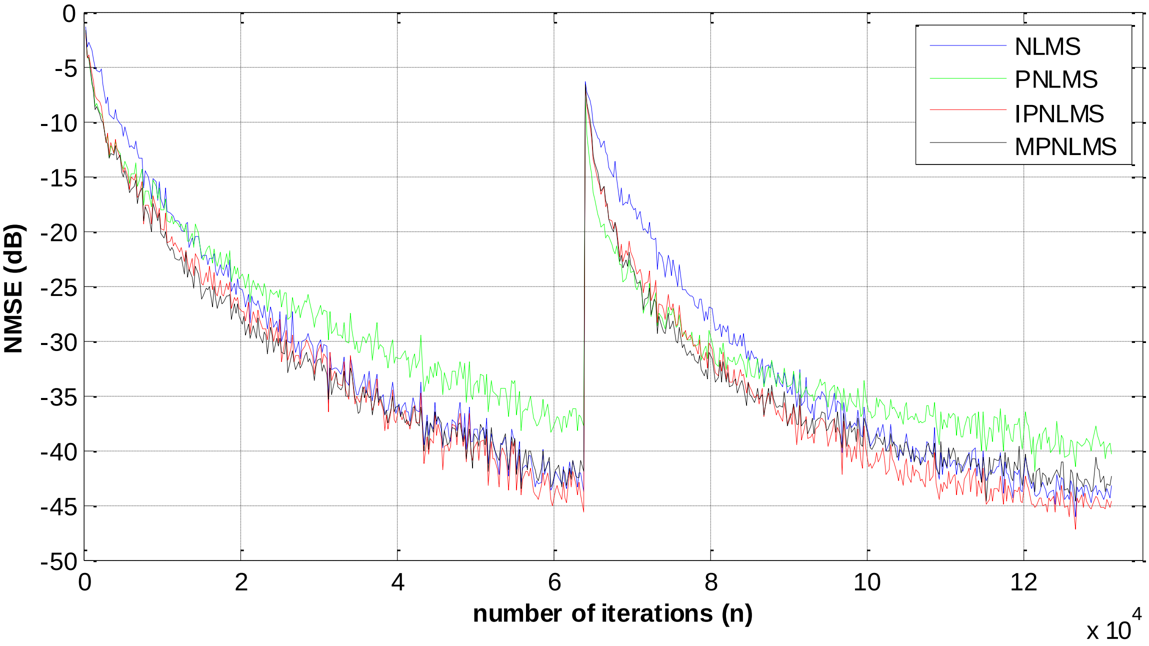

Figure 3.11a: USASI-noise input. Car system with L=256 and ξ =0.5138 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB. An abrupt change of the impulse response is applied at n=63744.

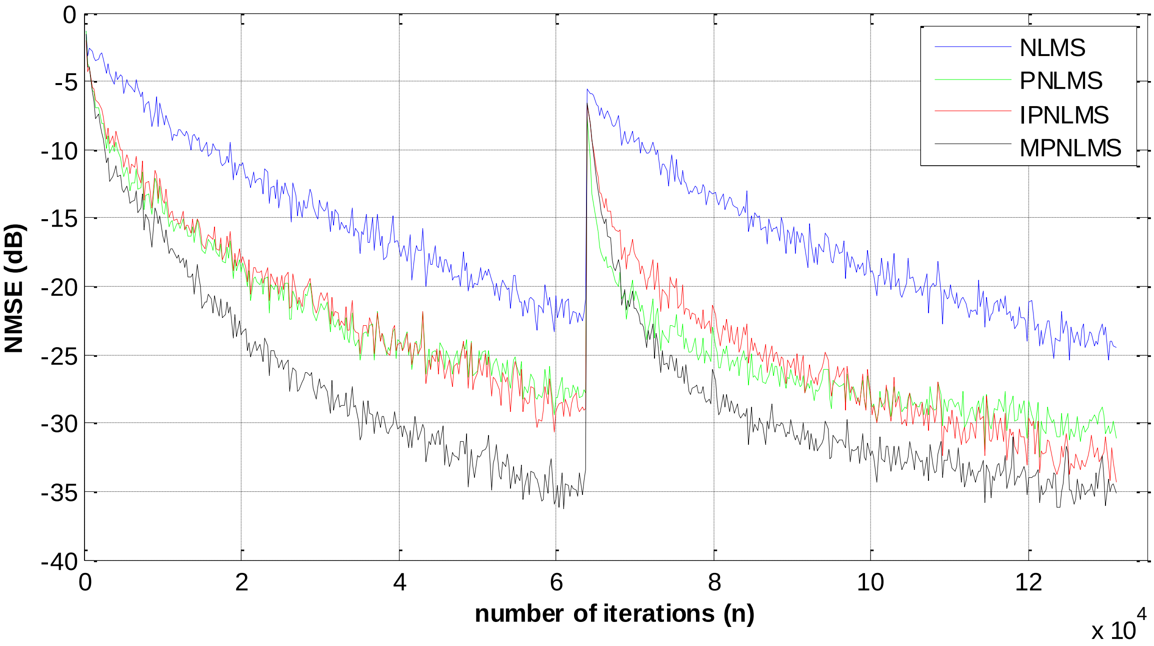

Figure 3.11b: USASI-noise input. Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB. An abrupt change of the impulse response is applied at n=63744.

The same thing is found also when testing with WGN-AR20 input or when using real ACN impulse responses (L=2048 for less sparse and L=8192 for more sparse). See figures 3.12.

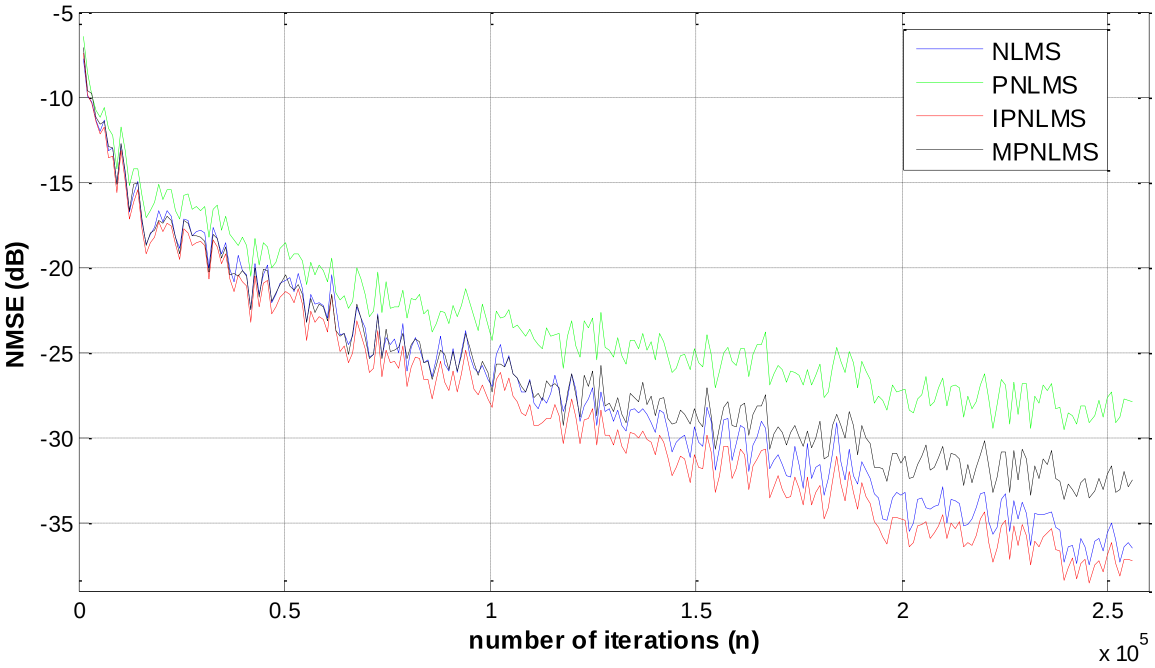

Figure 3.12a: WGN-AR20 input. Real ACN system with L=2048 and ξ =0.3673 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

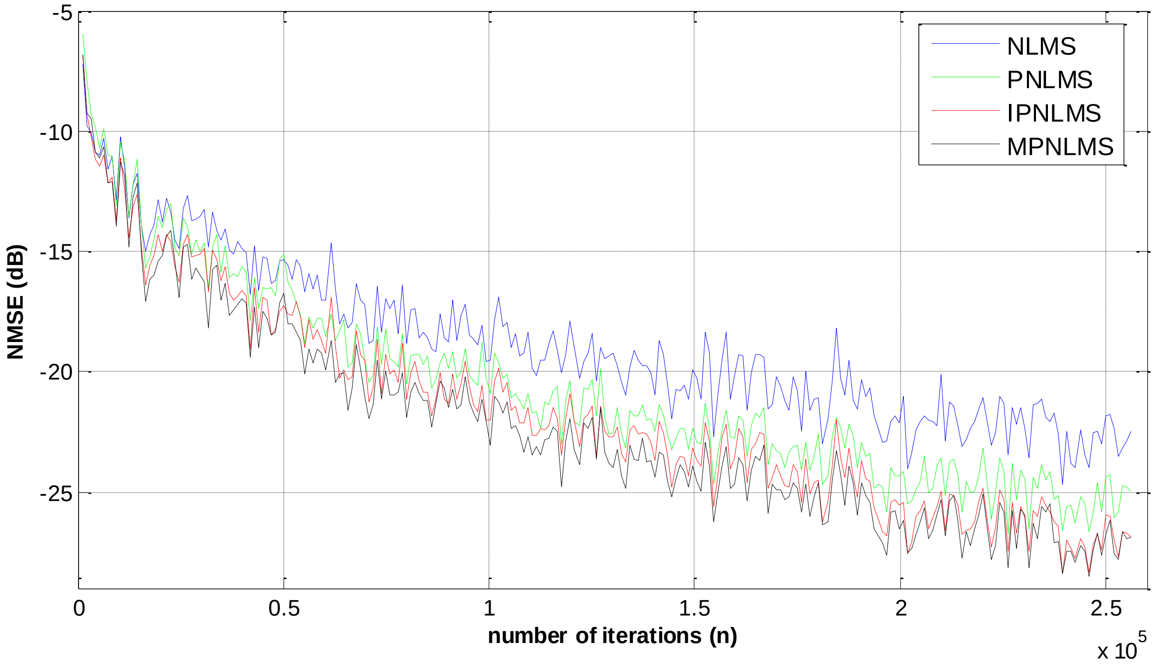

Figure 3.12b: WGN-AR20 input. Real ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

Extending to non-stationary input (speech), we made first a test to compare noisy and noiseless output cases, see figures 3.13. In the case of noisy output, there is only less sharp peaks down and a shift in the final MSE due to the added output SNR (50dB) for the compared algorithms. Therefore, simulations using noisy-outputs are more practical and better in avoiding zero-division issues during calculations that correspond to very small or zero inputs (silences).

- *Noiseless case*

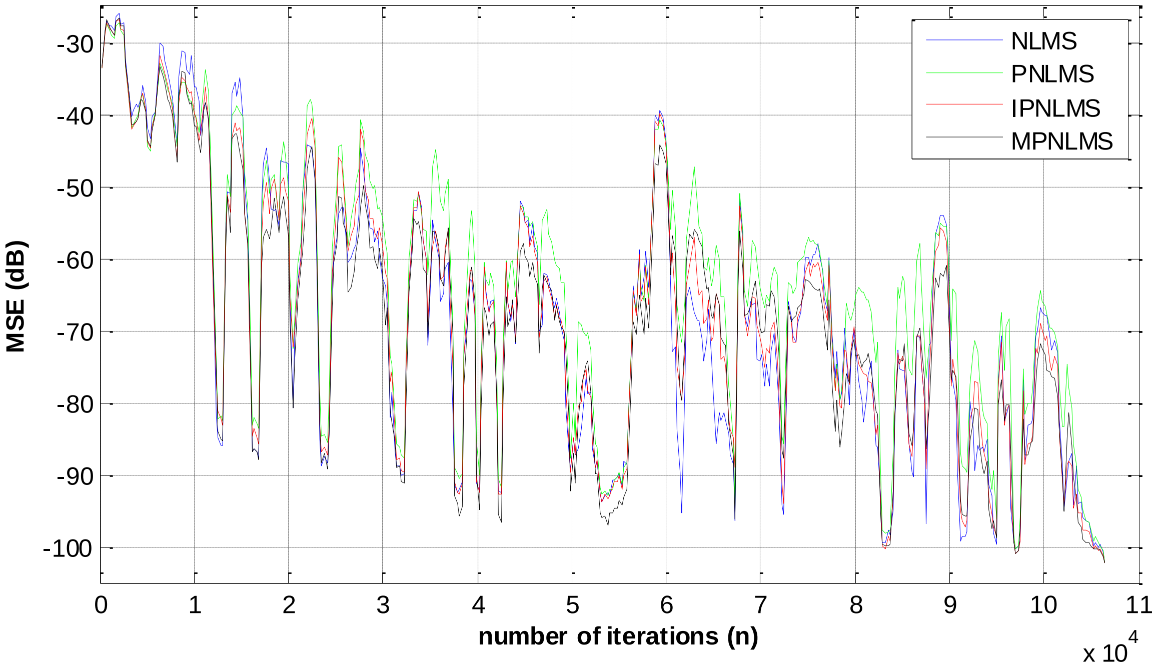

Figure 3.13a: Speech input. Real car system with L=256. NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). No output noise.

- *Noisy case (SNR=50dB)*

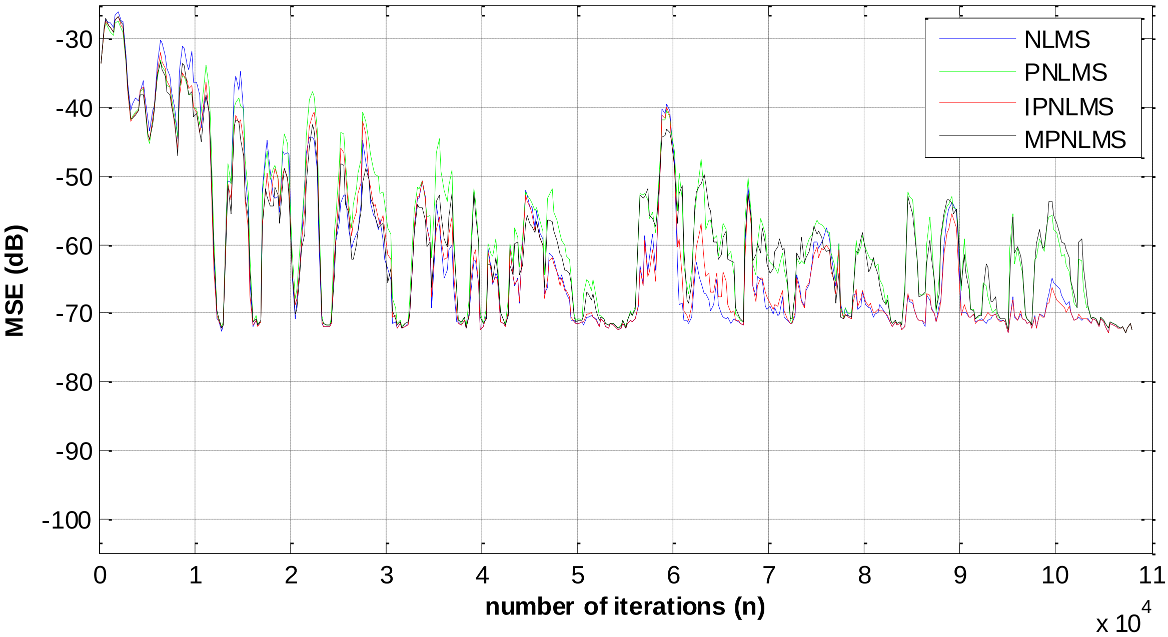

Figure 3.13b: Speech input. Real car system with L=256. NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

Figures 3.14 are obtained using two real long ACN systems (less sparse with L=2048 and more sparse with L=8192) with WGN with speech input. It could be noted that the same analogy is still valid for non-stationary case. That is, although PNLMS and MPNLMS have good initial convergence, they, later, perform worse than NLMS in dispersive systems. They behave significantly better than

NLMS in sparse systems. IPNLMS algorithm obtains a good overall performance in both sparse and dispersive systems.

We found the same results when testing with car systems and when using a jump in the impulse response.

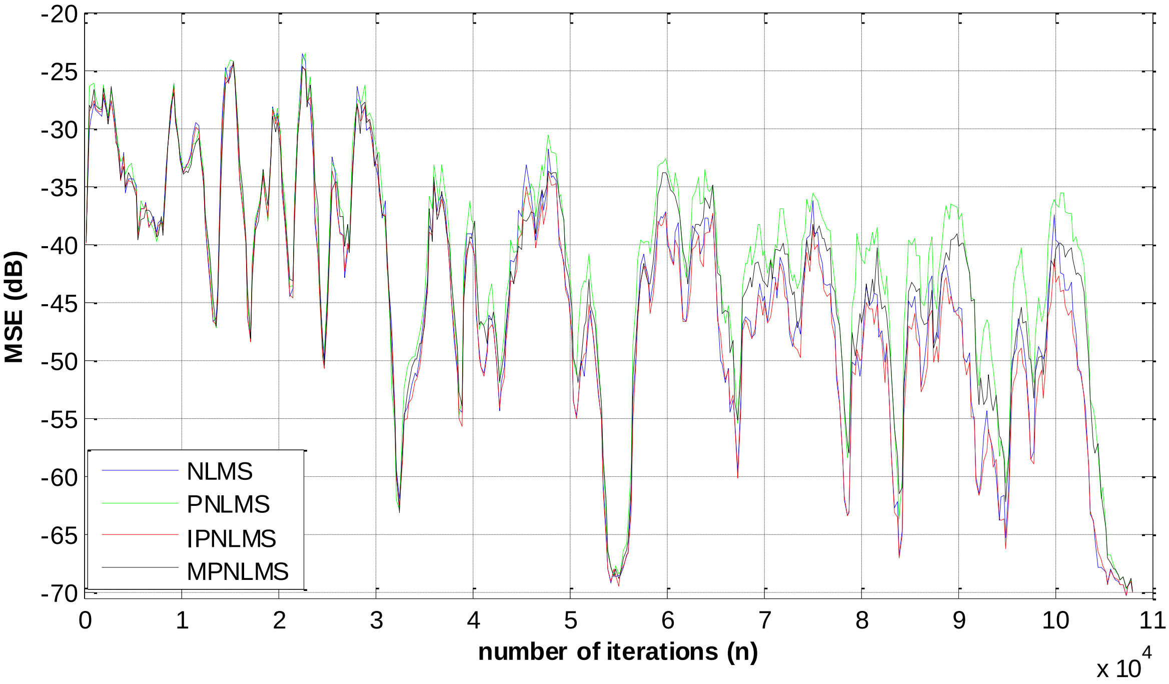

Figure 3.14a: Speech input. Real 'long' ACN system with L=2048 and ξ =0.3673 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

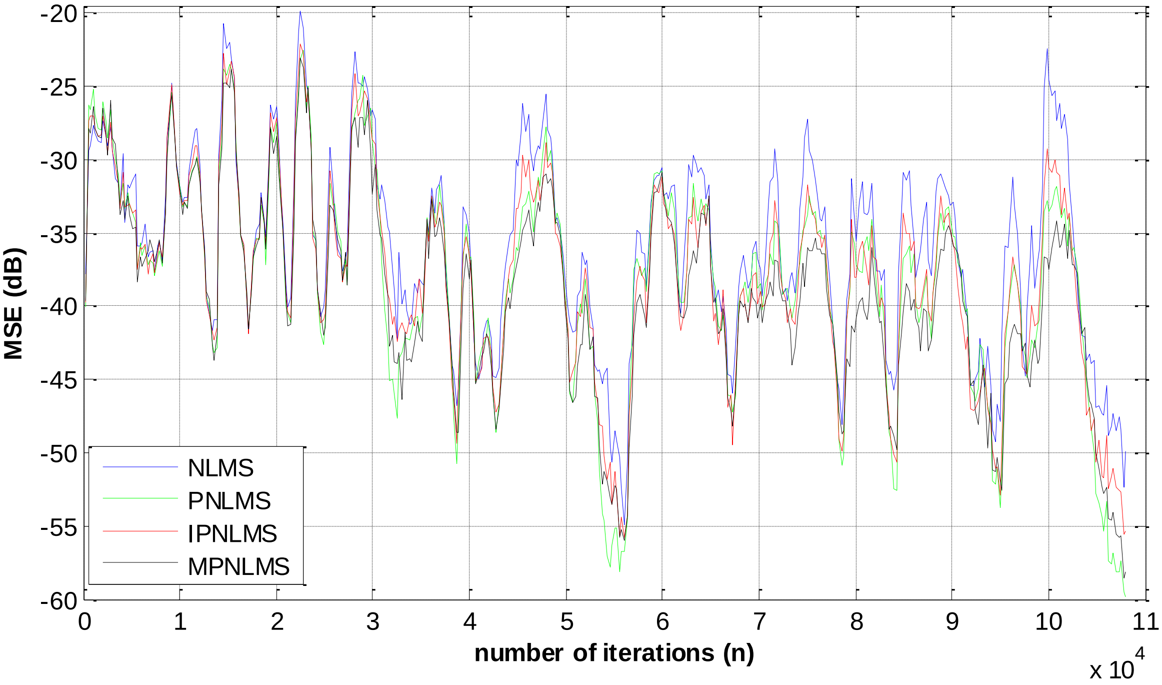

Figure 3.14b: Speech input. Real 'long' ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5) and MPNLMS (μ=0.3). Output with SNR=50dB.

3.5. Comparison between CS-based ISS and VSS algorithms

3.5.1. Choice of regularization step-size $\rho_{(R)ZA}$ and threshold parameter ε_{RZA}

In this subsection, we conduct tests to determine experimentally the values of $\rho_{(R)ZA}$ and ε_{RZA} that give a good performance of the (R)ZA-NLMS algorithm for a moderate-sparsity real car system (ξ =0.5138).

A). Variation of $\rho_{(R)ZA}$

In order to determine systematically a good value of the regularization step-size $\rho_{(R)ZA}$, the procedure used in [32]-[34] is followed (i.e. $\rho_{(R)ZA}$ is expressed in terms of σ_n^2). Initially, we considered $\sigma_n^2 = 10^{-\mathrm{SNR}/10}$ as in [32], where SNR is 50dB. Then, since the input signal power is not set equal to 1, we continued testing experimentally for many values of $\rho_{(R)ZA}$ as shown in Figure 3.15a.

It can be observed from Figure 3.15a that better steady-state performance is obtained for the ZA-NLMS algorithm when ρ_{ZA} is smaller, and since the convergence speed increases with the increase of ρ_{ZA} (see subsection 2.3.4), a convenient value for our simulations is $\rho_{(R)ZA}$=0.003σ_n^2.

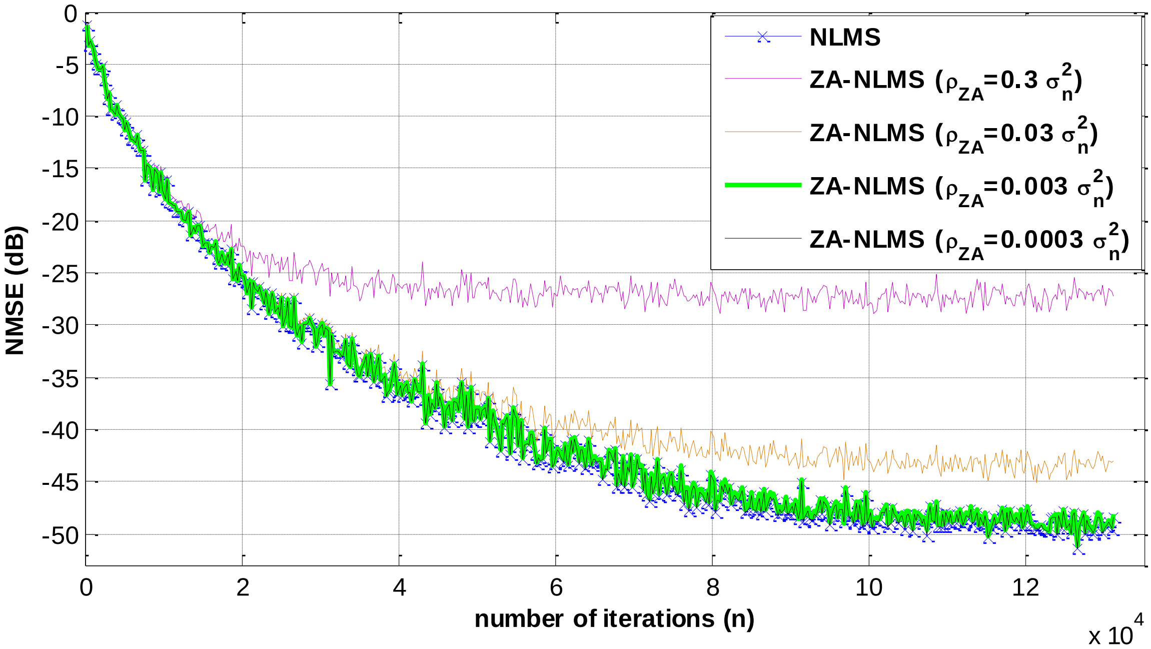

Figure 3.15a: USASI-noise input. Car system with L=256 and ξ =0.5138. NLMS (μ=0.3), ZA-NLMS (μ=0.3, ρ_{ZA}=0.3σ_n^2 / 0.03σ_n^2 / 0.003σ_n^2 / 0.0003σ_n^2). Output with SNR=50dB.

B). Variation of ε_{RZA}

From Figure 3.15b, we observe that better steady-state performance is obtained for the RZA-NLMS algorithm when ε_{RZA} is larger and since the reweighted zero-attractor (third term in the update equation (2.26)) takes significant effect only on those taps whose magnitudes are comparable to $1/\varepsilon_{RZA}$, a good choice of ε_{RZA} for our simulations can be 30 (to zero-attract significantly taps of values less than 1/30).

For the VSS-ZA-NLMS and VSS-RZA-NLMS algorithms, the smoothing factor is taken to be β=0.1, the initialization maximal step-size μ_{max}=1.0 and the positive threshold parameter C=10^{-7}.

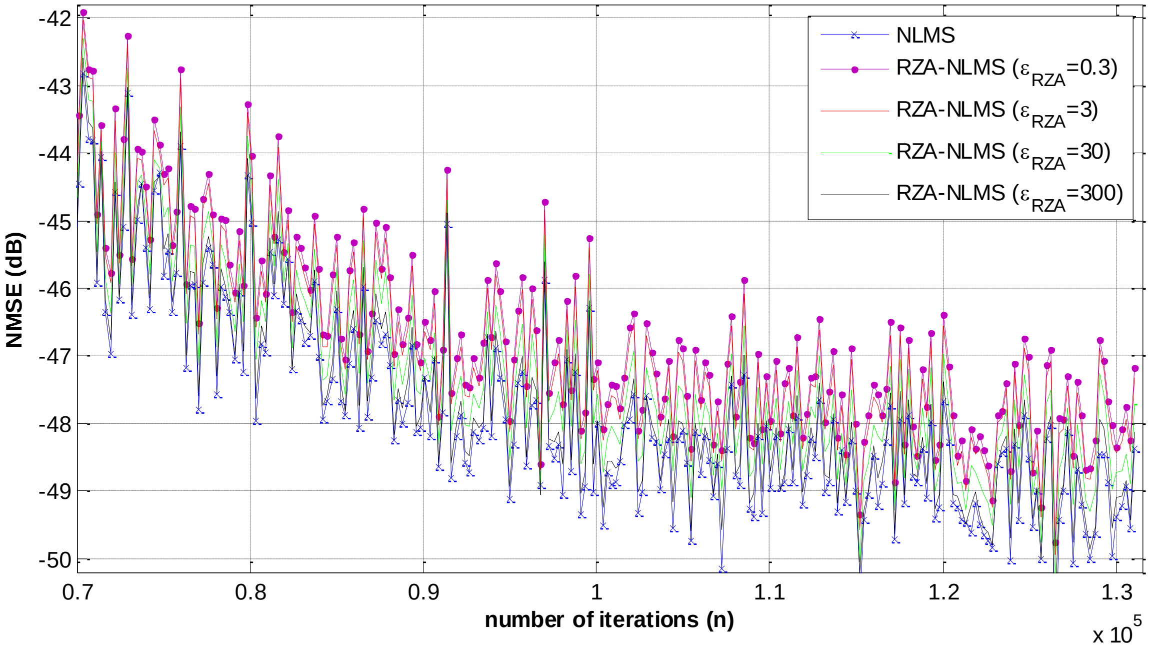

Figure 3.15b: USASI-noise input. Car system with L=256 and ξ =0.5138. NLMS (μ=0.3), RZA-NLMS (μ=0.3, ε_{RZA}=0.3 / 3.0 / 30 / 300). Output with SNR=50dB.

Before starting with comparisons, it is worthy to mention here that most of published papers dealing with ZA-NLMS and RZA-NLMS algorithms assumed white Gaussian inputs and used impulse responses with relatively small number of taps (8, 16, 64 taps) [30]-[34], [83], [84] and they found slight performance improvements of the ZA-NLMS and RZA-NLMS algorithms with respect to the NLMS algorithm for highly sparse systems where the NLMS was better for less sparse and dispersive systems. This is because the ℓ_1-constraint for significant non-zero tap-weights is useless [85].

In our study, longer systems and more complicated inputs are used. Firstly, we use synthetic impulse responses with USASI-noise input, see figures 3.16. Then, we extend for ACN systems (figures 3.17) and speech input (figures 3.18).

Investigating, we could observe that the performance of ZA-NLMS and RZA-NLMS algorithms is similar to the NLMS algorithm and even worse in less sparse cases where the VSS-ZA-NLMS and the VSS-RZA-NLMS algorithms outperform significantly NLMS (and R/ZA-NLMS) for all cases (stationary and non-stationary) in terms of convergence speed and estimation accuracy. Furthermore, VSS-RZA-NLMS has a little bit better steady-state MSE (better accuracy) than VSS-ZA-NLMS but more computational complexity.

The relatively small value of $\rho_{(R)ZA}$ chosen to satisfy the requirement of good estimation-accuracy led the reweighted algorithms (RZA-NLMS and VSS-RZA-NLMS) to behave approximately the same as the non-reweighted ones (ZA-NLMS and VSS-ZA-NLMS respectively).

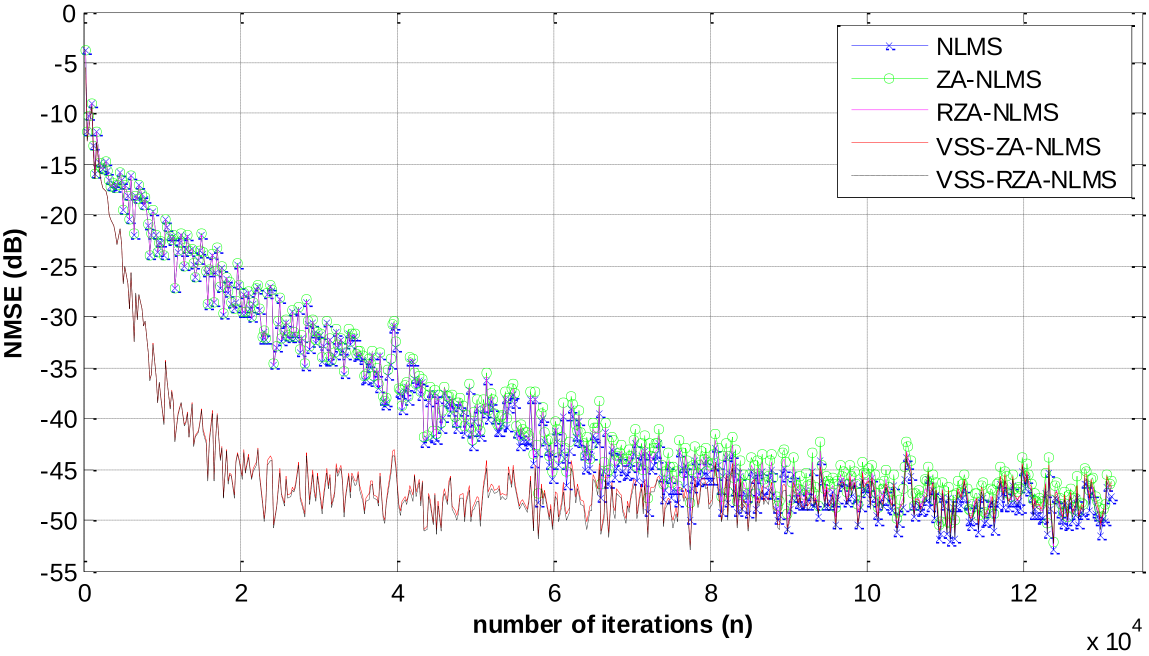

Figure 3.16a: USASI-noise input. Synthetic system with length L=256, ψ =160 and ξ =0.3028 (dispersive). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

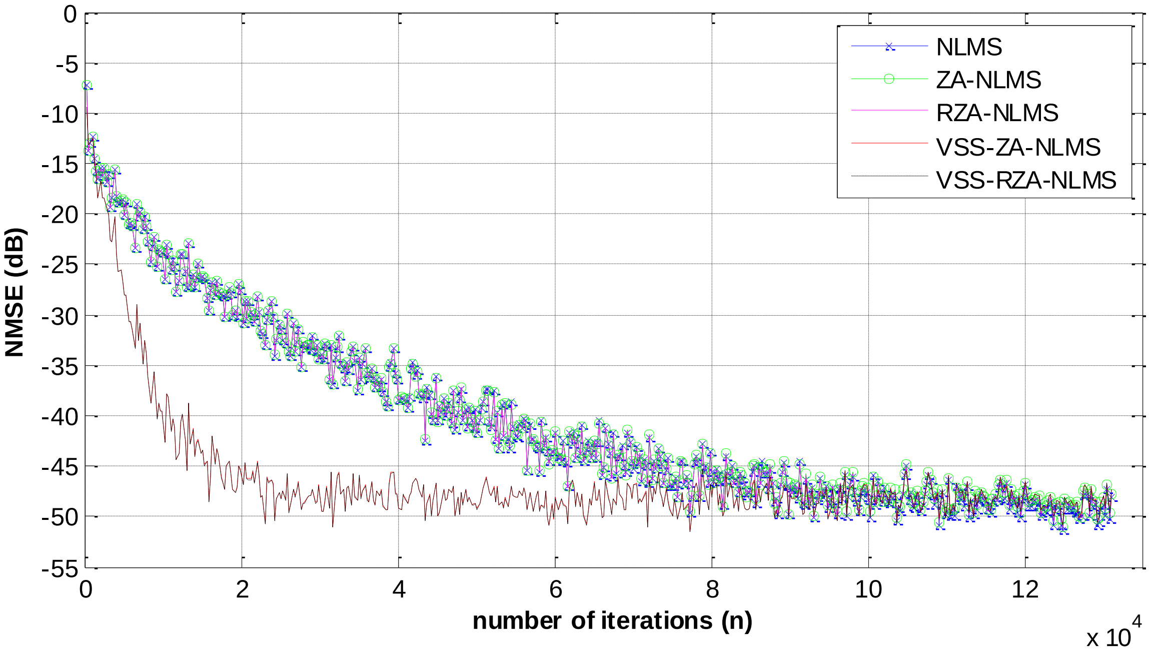

Figure 3.16b: USASI-noise input. Synthetic system with length L=256, ψ =10 and ξ =0.8296 (very sparse). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

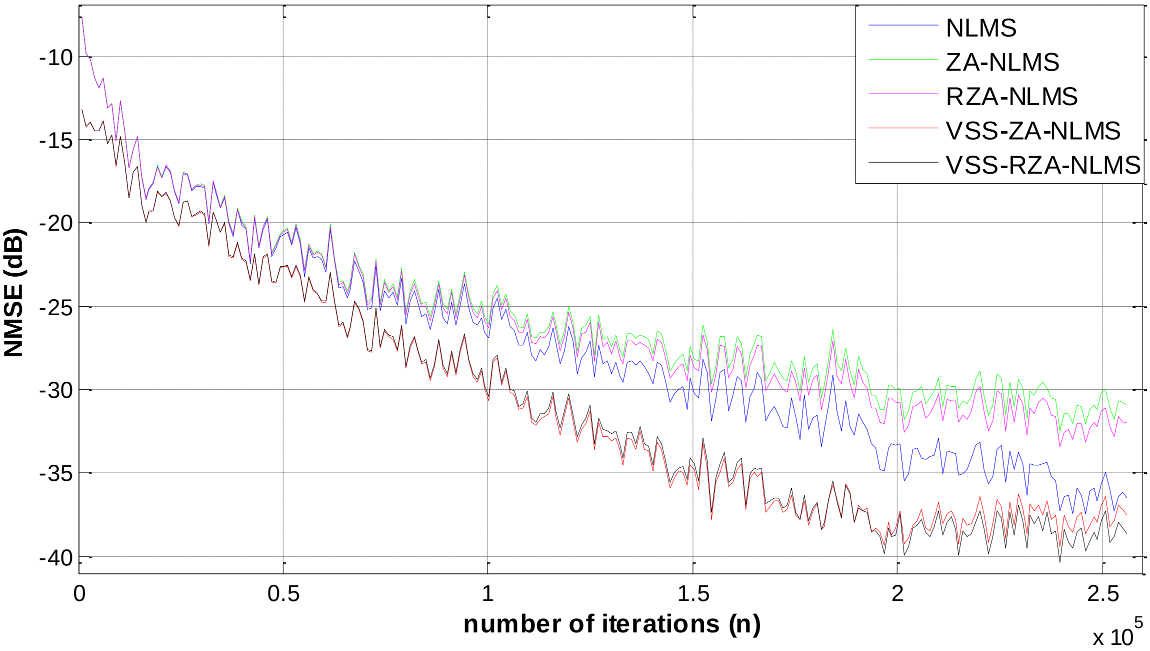

Figure 3.17a: WGN-AR20 input. ACN system with L=2048 and ξ =0.3673 (less sparse). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

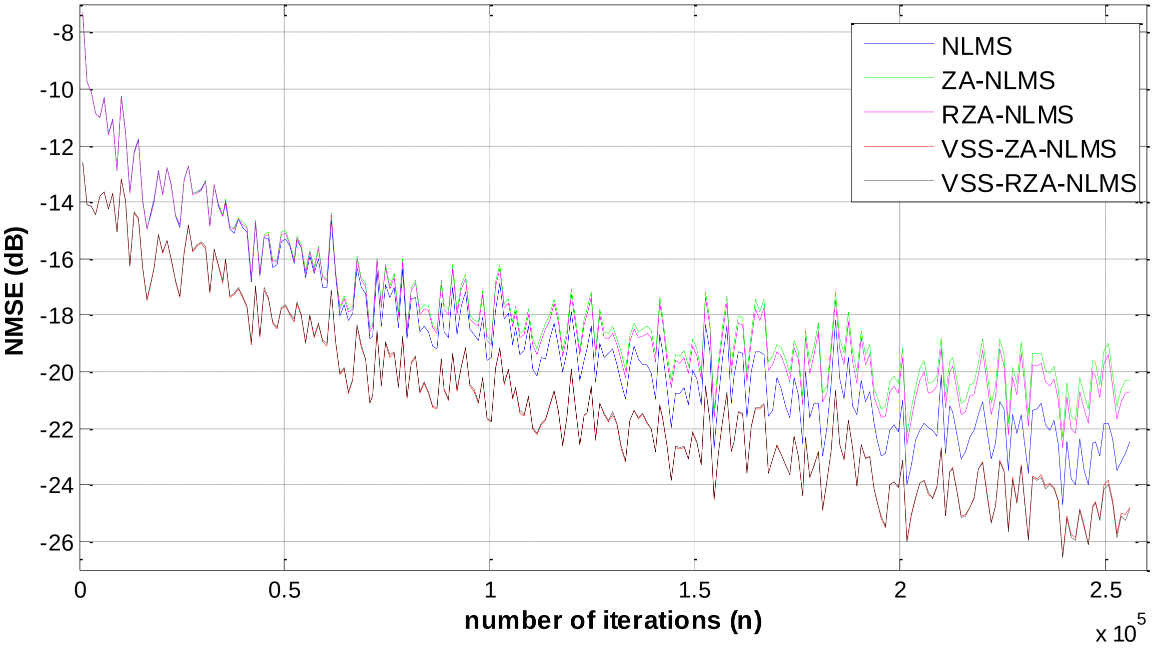

Figure 3.17b: WGN-AR20 input. ACN system with L=8192 and ξ =0.6199 (more sparse). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

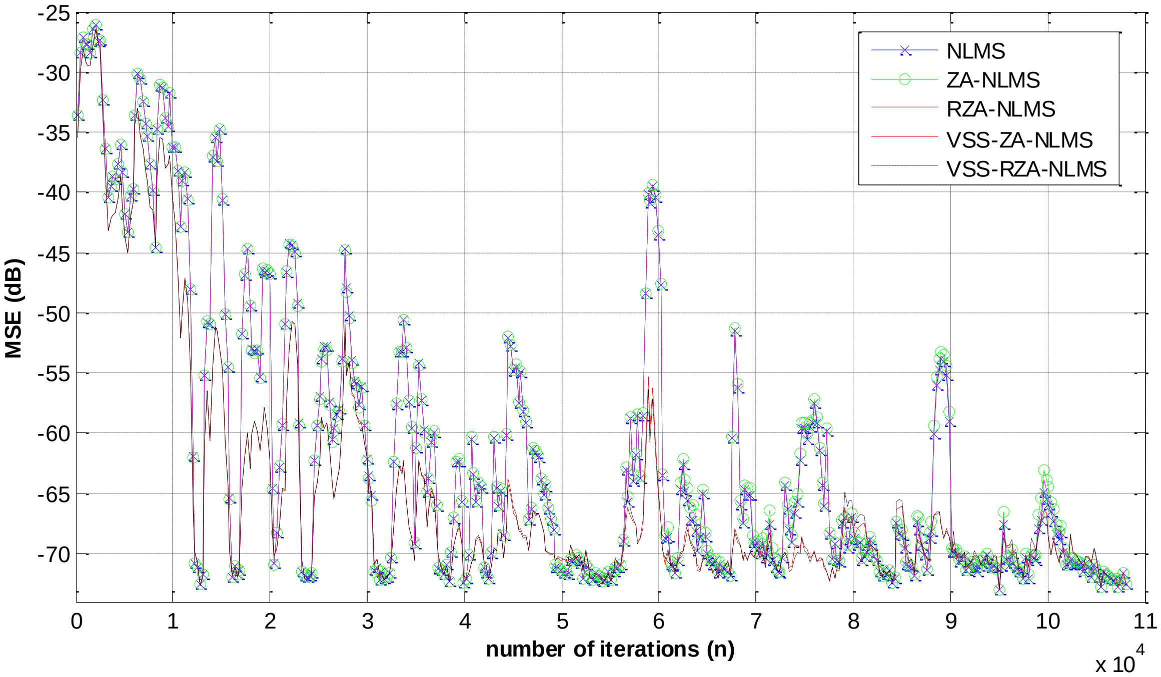

Figure 3.18a: Speech input. Car system with L=256 and ξ =0.5138 (less sparse). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

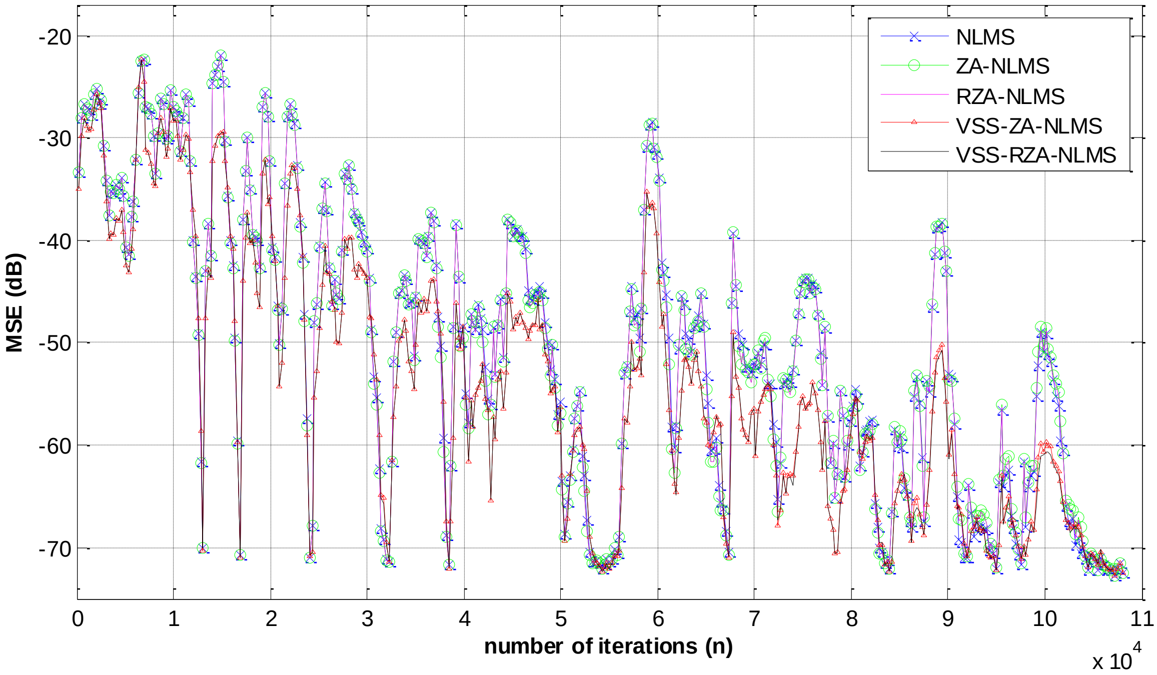

Figure 3.18b: Speech input. Car system with L=1024 and ξ =0.7410 (more sparse). μ=0.3 for all algorithms, $\rho_{(R)ZA}$=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}. Output with SNR=50dB.

The same results are found for the re-convergence behavior when using an abrupt change (jump) in the impulse response and when using WGN-AR20 input.

3.6. Comparisons between pairs of conventional sparsity-aware & SC-algorithms and the VSS-RZA-NLMS algorithm

In all comparisons figures, the NLMS learning-curve is added as a reference.

In order to reduce graphs crowding and since the VSS-RZA-NLMS attains the best performance of the CS-based algorithms compared in the previous section, we consider it as the best candidate to represent its family in the following comparisons.

3.6.1. Comparison between PNLMS, SC-PNLMS and VSS-RZA-NLMS

In figures 3.19, we made our comparisons using USASI-noise input with two synthetic impulse responses, then, we tested the re-convergence using two real car systems with a jump at n=63744, see figures 3.20. After that, the case of non-stationary speech input is tested (figures 3.20).

From figures 3.19-3.21 (a & b), one can notice that the SC-PNLMS algorithm outperforms the PNLMS algorithm especially for less-sparse and dispersive systems with a performance very close to NLMS but slightly worse. However, for sparse cases, SC-PNLMS and PNLMS become very similar and behave better in terms of convergence rate and accuracy than NLMS.

For stationary inputs, the VSS-RZA-NLMS algorithm has the best overall convergence-speed and steady-state performance particularly in less-sparse and dispersive systems where PNLMS and SC-PNLMS have, most of the time, faster initial convergence (or re-convergence) speed which reduces later to be slower than the convergence speed of VSS-RZA-NLMS. In addition, the longer and the more-sparse the system is, the closer the learning-curves of PNLMS/SC-PNLMS to the learning-curve of VSS-RZA-NLMS and SC-PNLMS can achieve even better steady-state level than VSS-RZA-NLMS as found when testing with ACN system.

For speech input (non-stationary), PNLMS and SC-PNLMS are more favorable in highly-sparse systems where VSS-RZA-NLMS has superior performance for less-sparse and dispersive systems.

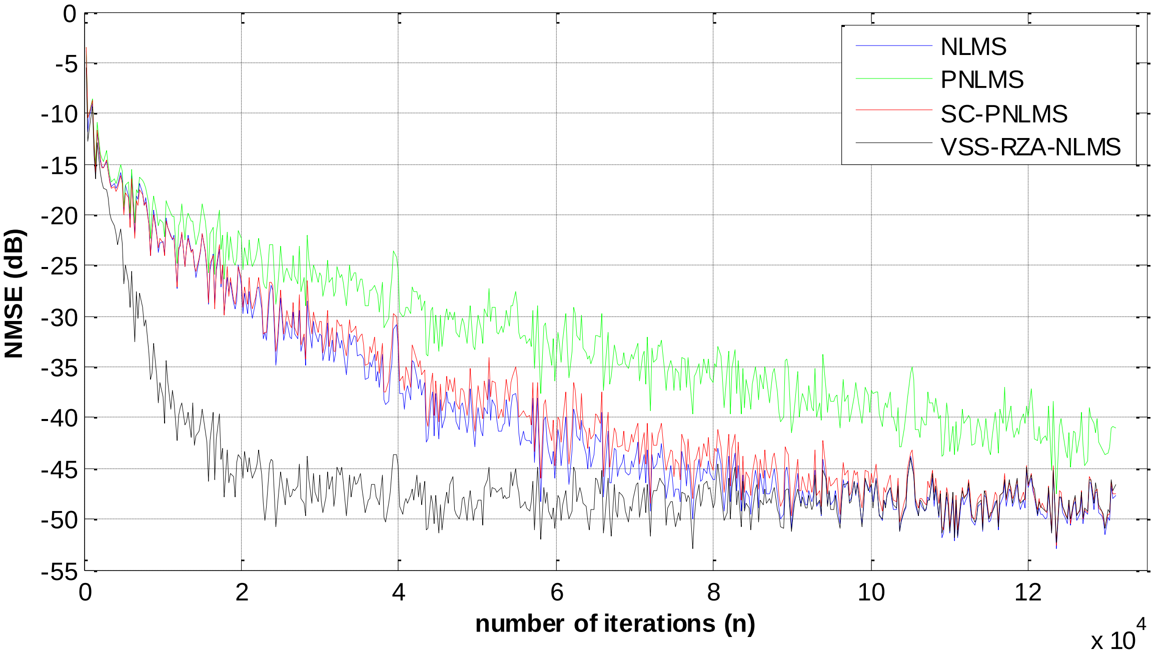

Figure 3.19a: USASI-noise input. Synthetic system with length L=256, ψ =160 and ξ =0.3028 (dispersive). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

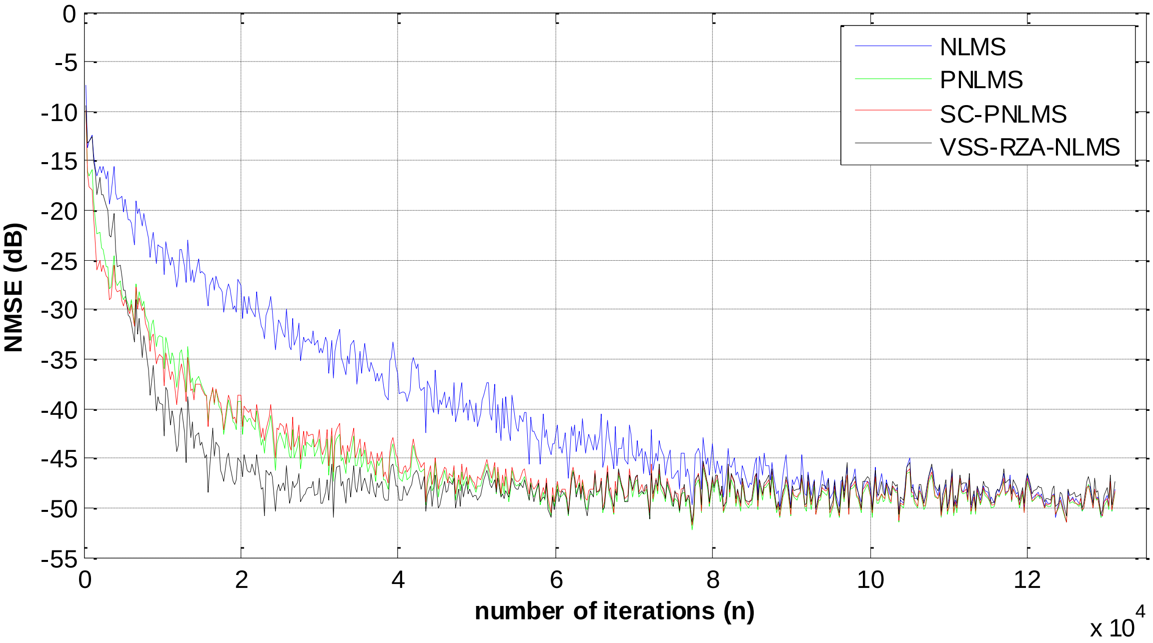

Figure 3.19b: USASI-noise input. Synthetic system with length L=256, ψ =10 and ξ =0.8296 (very sparse). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

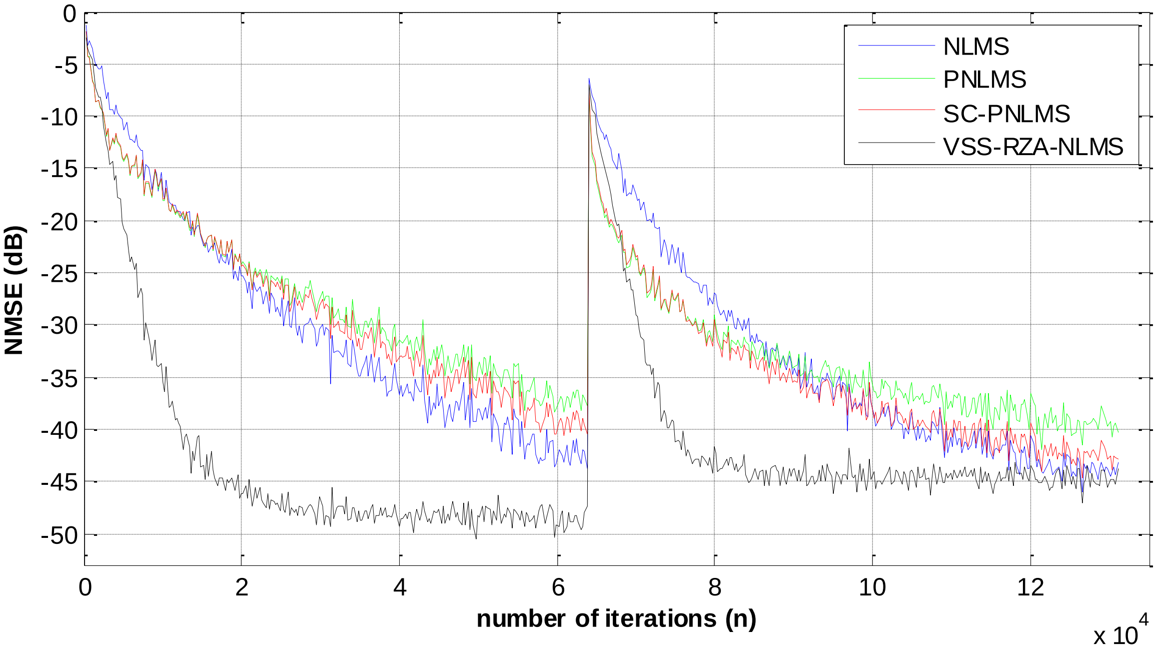

Figure 3.20a: USASI-noise input. Car system with L=256 and ξ =0.5138 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB. An abrupt change of the impulse response is applied at n=63744.

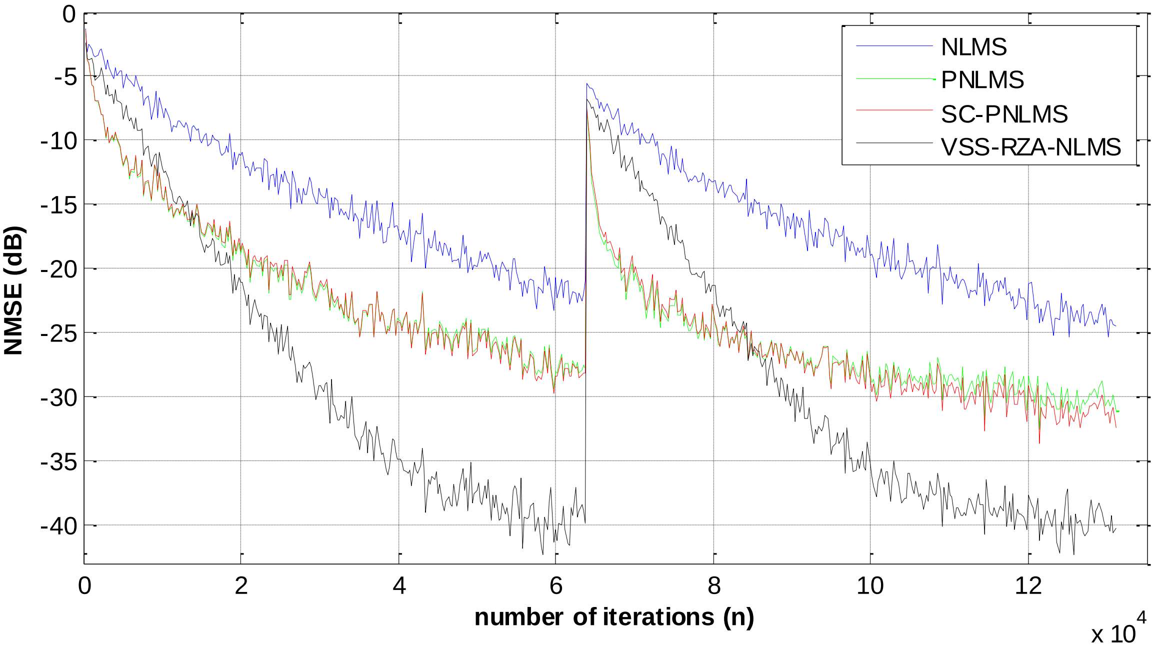

Figure 3.20b: USASI-noise input. Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB. An abrupt change of the impulse response is applied at n=63744.

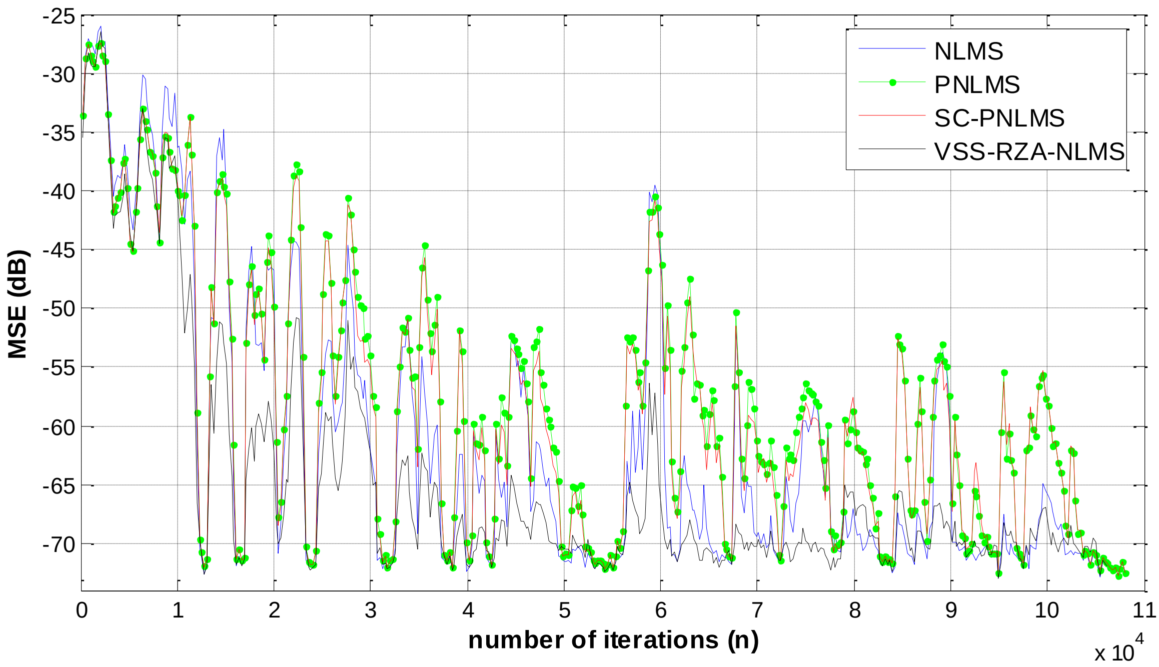

Figure 3.21a: Speech input. Car system with L=256 and ξ =0.5138 (less sparse). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

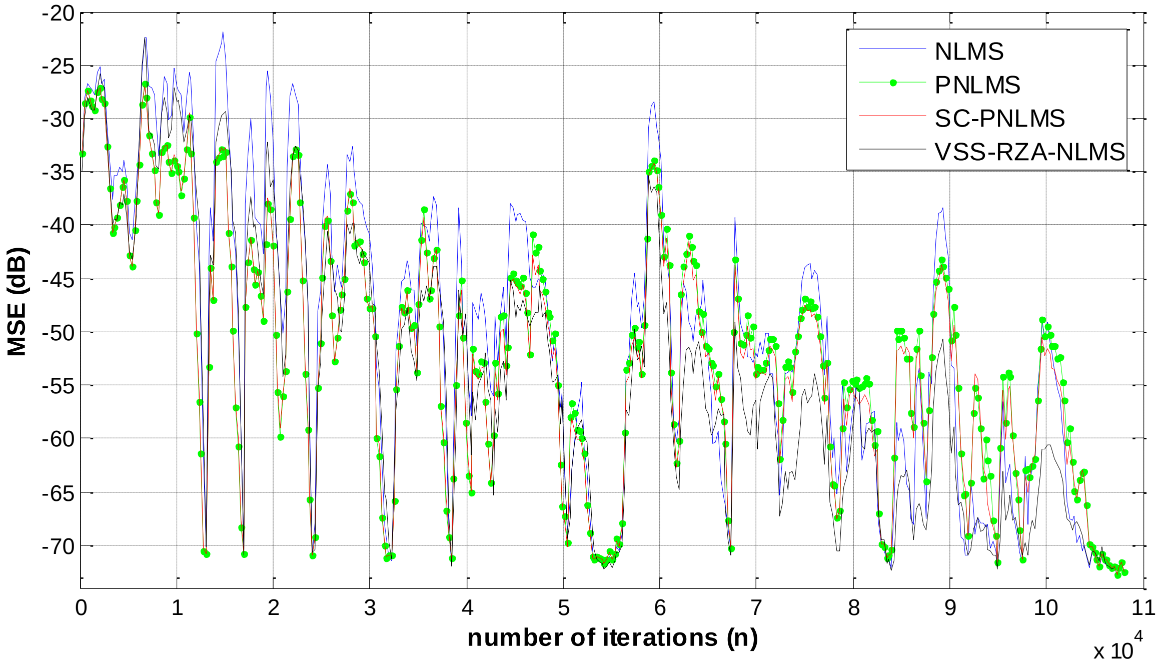

Figure 3.21b: Speech input. Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), PNLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

3.6.2. Comparison between IPNLMS, SC-IPNLMS and VSS-RZA-NLMS

As previously, tests are made first for synthetic systems with USASI-noise input (figures 3.22) and real ACN systems with WGN-AR20 input (figures 3.23), then, for the case of speech input (figures) in real ACN systems. Other tests have made using an abrupt change is made at the impulse response. The results were as follows

The SC-IPNLMS algorithm outperforms the IPNLMS algorithm especially for more-sparse systems. However, for less sparse and dispersive cases, SC-IPNLMS and IPNLMS behave similarly and their performances are slightly better than NLMS.

For stationary inputs, it is clear that VSS-RZA-NLMS has better global convergence-speed and steady-state performance than IPNLMS and SC-IPNLMS when the system is less sparse or dispersive where they have, for most cases, a little bit faster initial convergence (or re-convergence) speed which reduces later to be slower than the convergence speed of VSS-RZA-NLMS. In addition, the longer and the more-sparse the system is, the closer the learning-curves of IPNLMS/SC-IPNLMS to the learning-curve of VSS-RZA-NLMS and they can achieve even better steady-state level than VSS-RZA-NLMS as in Figure 3.23b.

For speech input (non-stationary), the SC-IPNLMS algorithm has the best global performance in more-sparse systems where in less sparse and dispersive systems the VSS-RZA-NLMS algorithm is the most favorable.

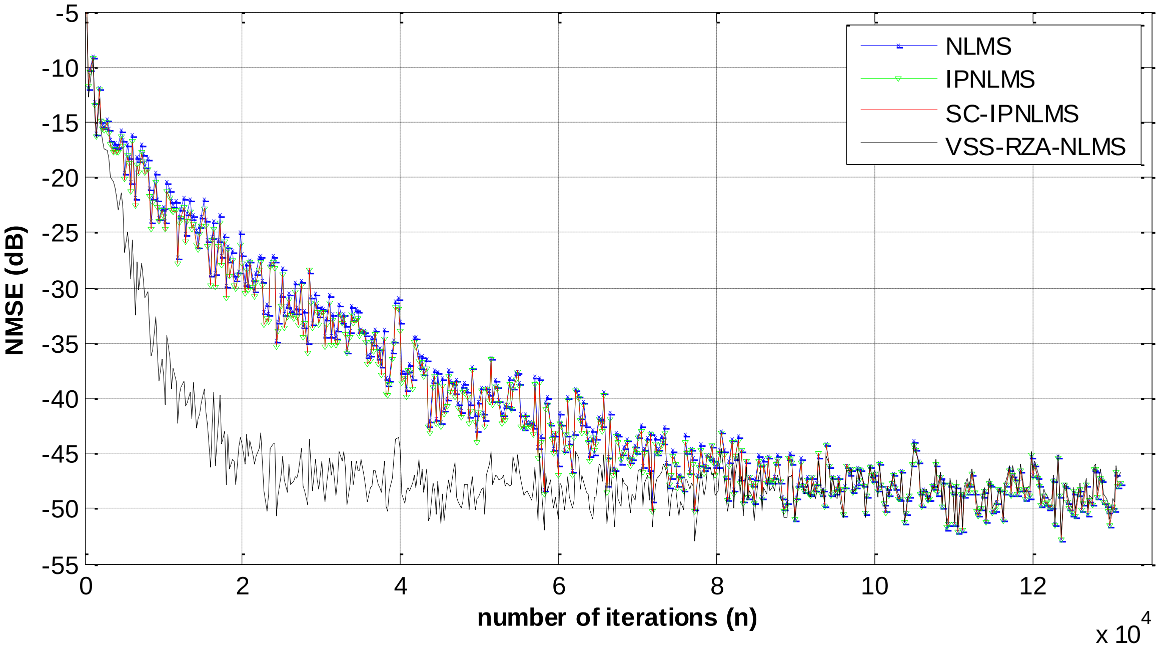

Figure 3.22a: USASI-noise input. Synthetic system with length L=256, ψ =160 and ξ =0.3028 (dispersive). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

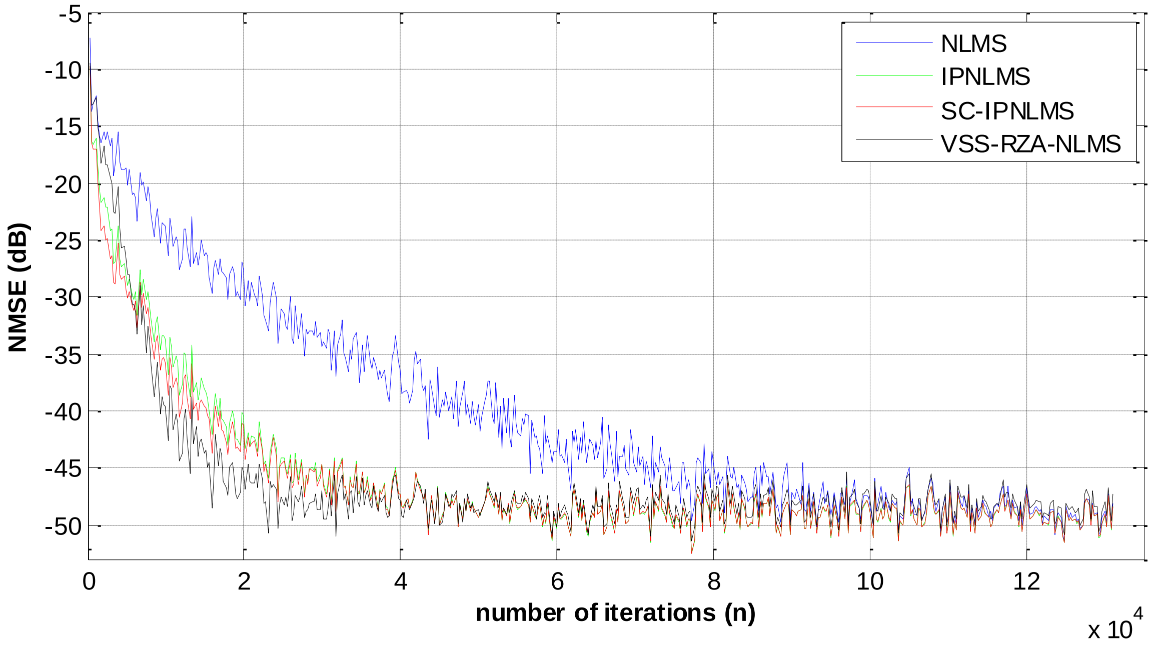

Figure 3.22b: USASI-noise input. Synthetic system with length L=256, ψ =10 and ξ =0.8296 (very sparse). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

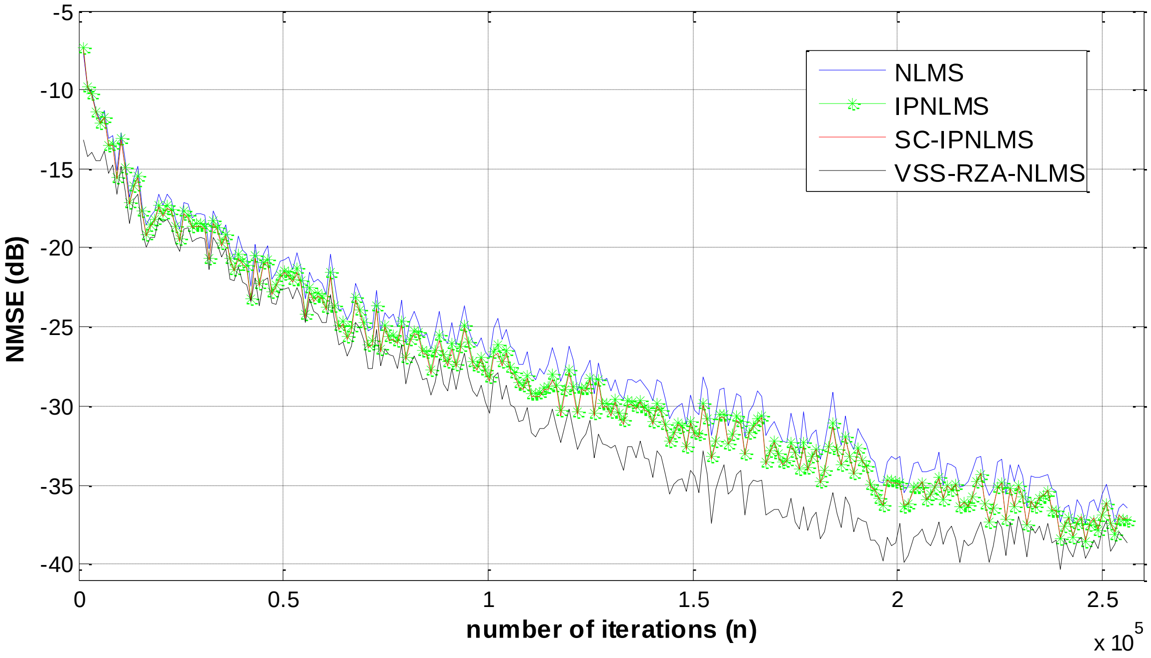

Figure 3.23a: WGN-AR(20) input. ACN system with L=2048 and ξ =0.3673 (less sparse). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

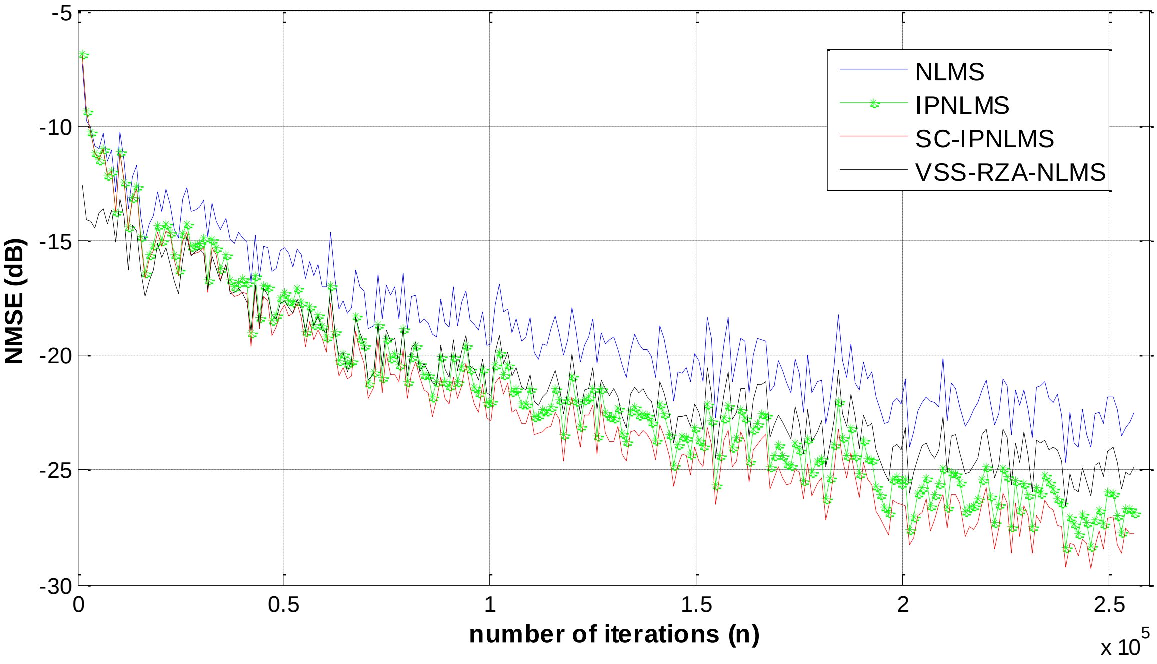

Figure 3.23b: WGN-AR(20) input. ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

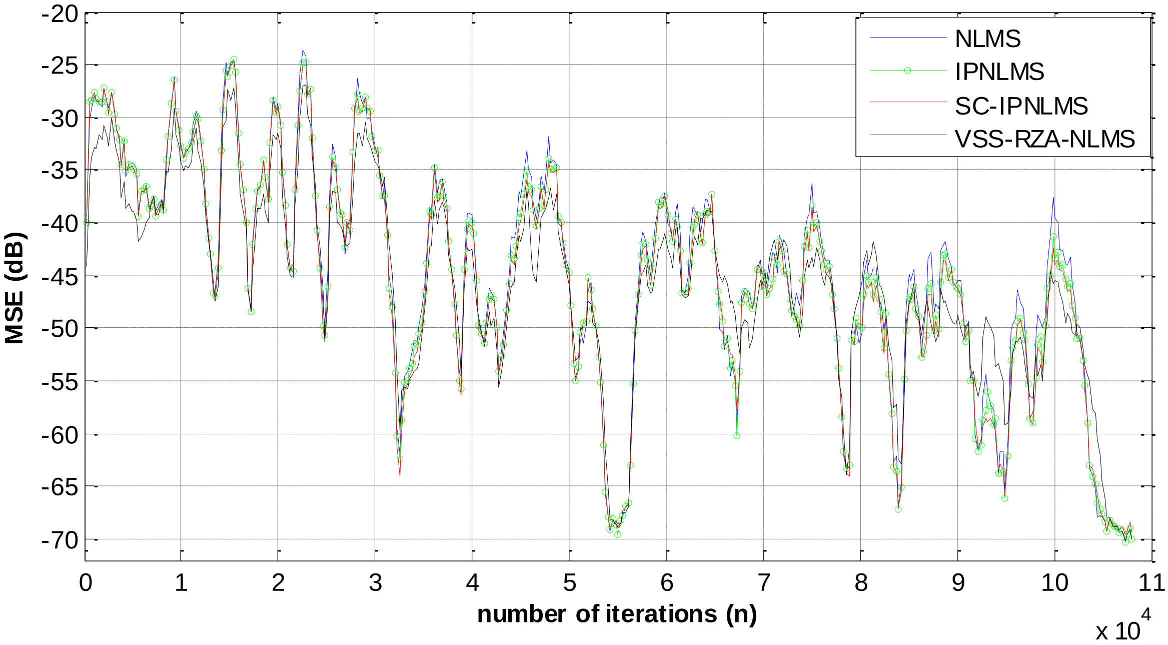

Figure 3.24a: Speech input. ACN system with L=2048 and ξ =0.3673 (less sparse). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

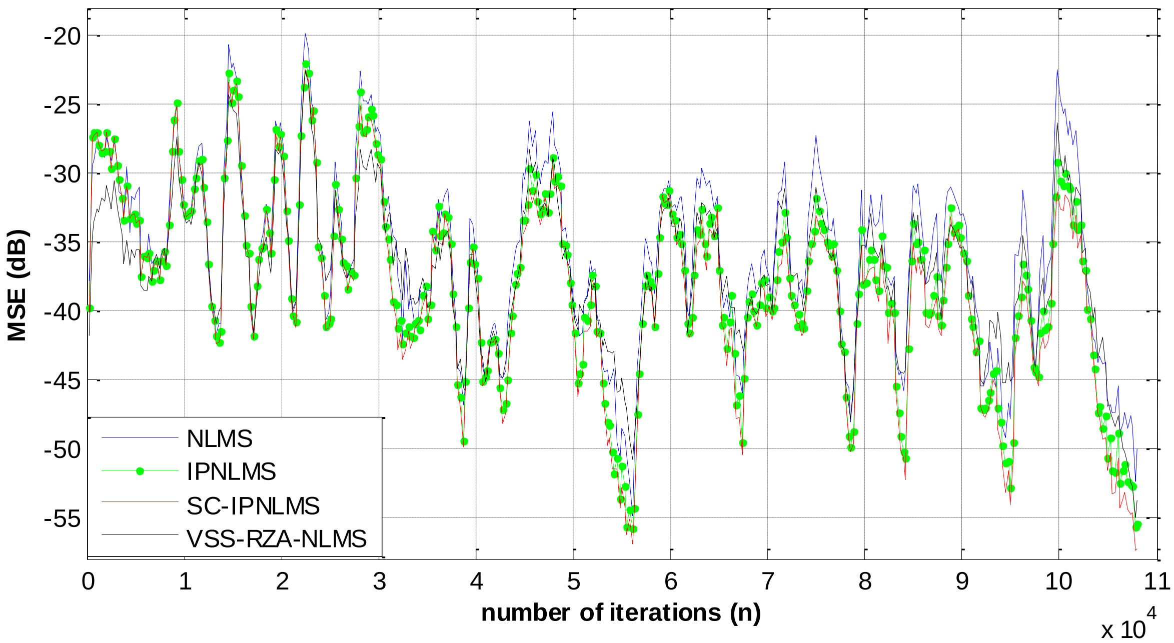

Figure 3.24b: Speech input. ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), IPNLMS (μ=0.3, α=-0.5), SC-IPNLMS (μ=0.3, α=-0.5) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

3.6.3. Comparison between MPNLMS, SC-MPNLMS and VSS-RZA-NLMS

The same procedure as in the precedent subsection, we started comparisons using synthetic systems with USASI-noise input in figures 3.25, then, we used real ACN systems with WGN-AR20 input (figures 3.26) and finally speech input is used in real car systems (figures 3.27). Tests for re-convergence behavior have also been conducted.

Observations made at figures 3.25-3.27 (a & b) gave the result that the SC-MPNLMS algorithm slightly surpasses the MPNLMS algorithm especially for less-sparse and dispersive systems.

For stationary inputs, it is clear that the overall performance of VSS-RZA-NLMS is better than MPNLMS and SC-MPNLMS for most less-sparse and dispersive systems where they have faster initial convergence (or re-convergence) speed in more-sparse impulse responses. In addition, the longer and the more-sparse the system is, the closer the learning-curves of MPNLMS/SC-MPNLMS to the one of VSS-RZA-NLMS and can achieve even better steady-state level than VSS-RZA-NLMS, which is the case in Figure 3.26b.

For speech input (non-stationary), the SC-MPNLMS and MPNLMS algorithms are better in more-sparse systems where in less-sparse and dispersive systems the VSS-RZA-NLMS algorithm behaves the best.

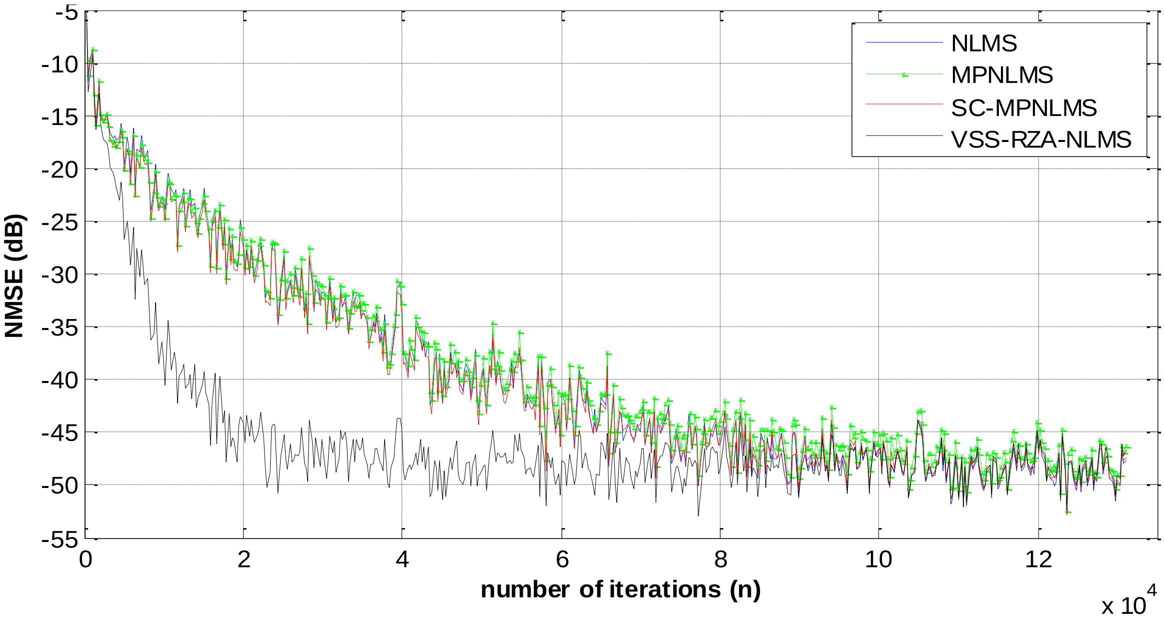

Figure 3.25a: USASI-noise input. Synthetic system with length L=256, ψ =160 and ξ =0.3028 (dispersive). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

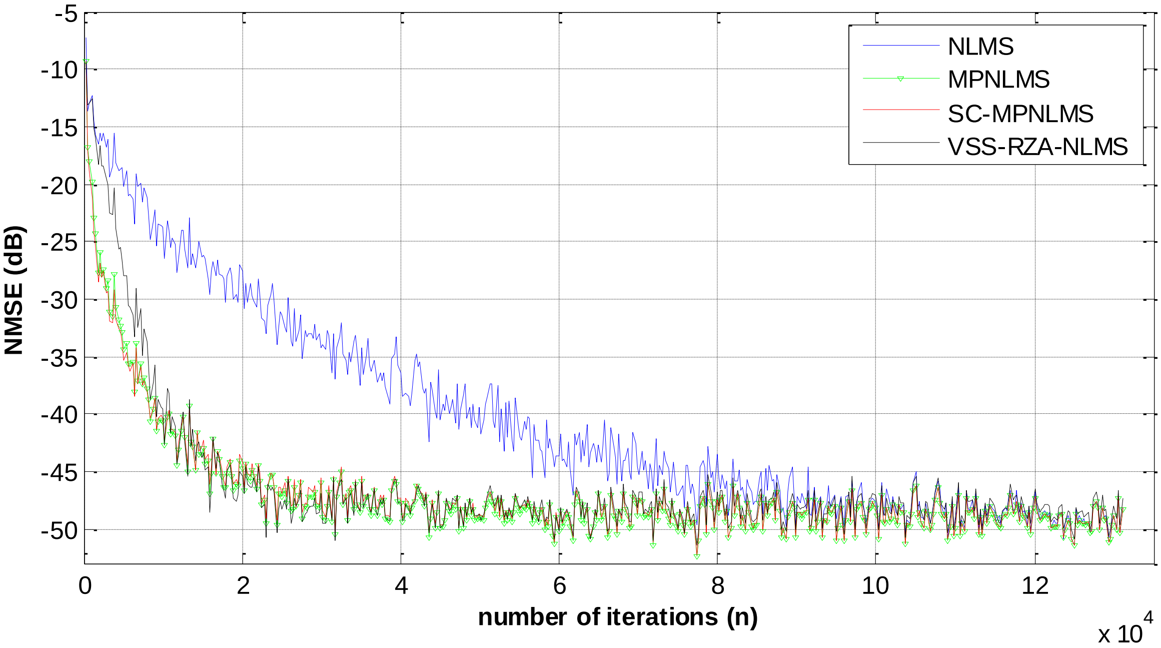

Figure 3.25b: USASI-noise input. Synthetic system with length L=256, ψ =10 and ξ =0.8296 (very sparse). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

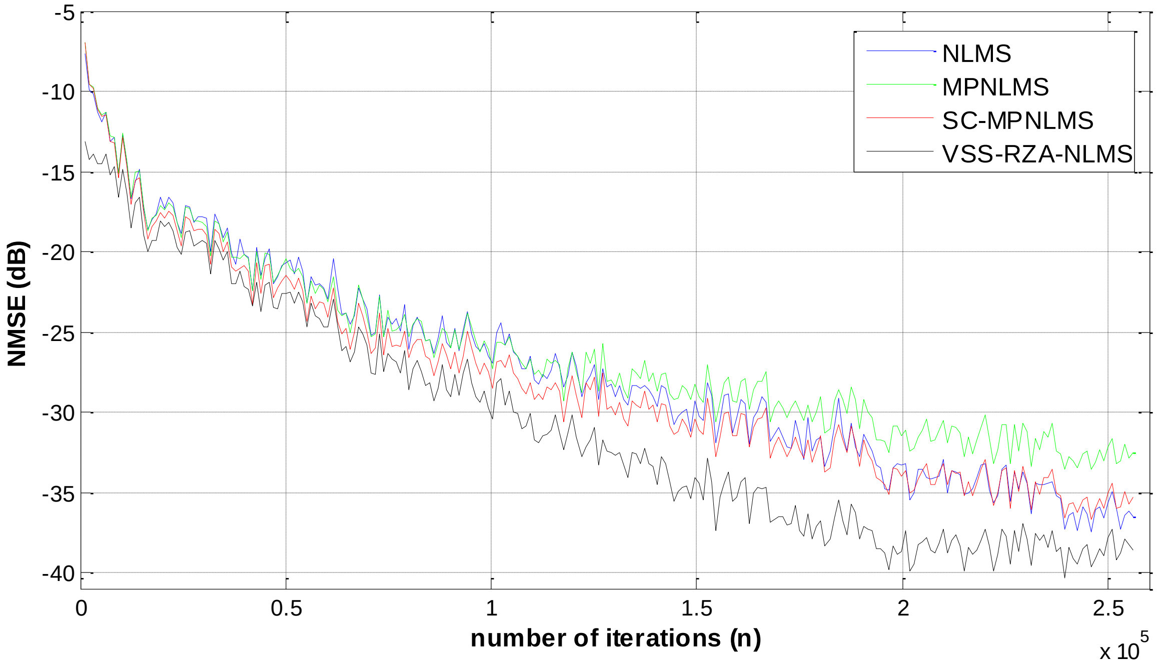

Figure 3.26a: WGN-AR(20) input. ACN system with L=2048 and ξ =0.3673 (less sparse). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

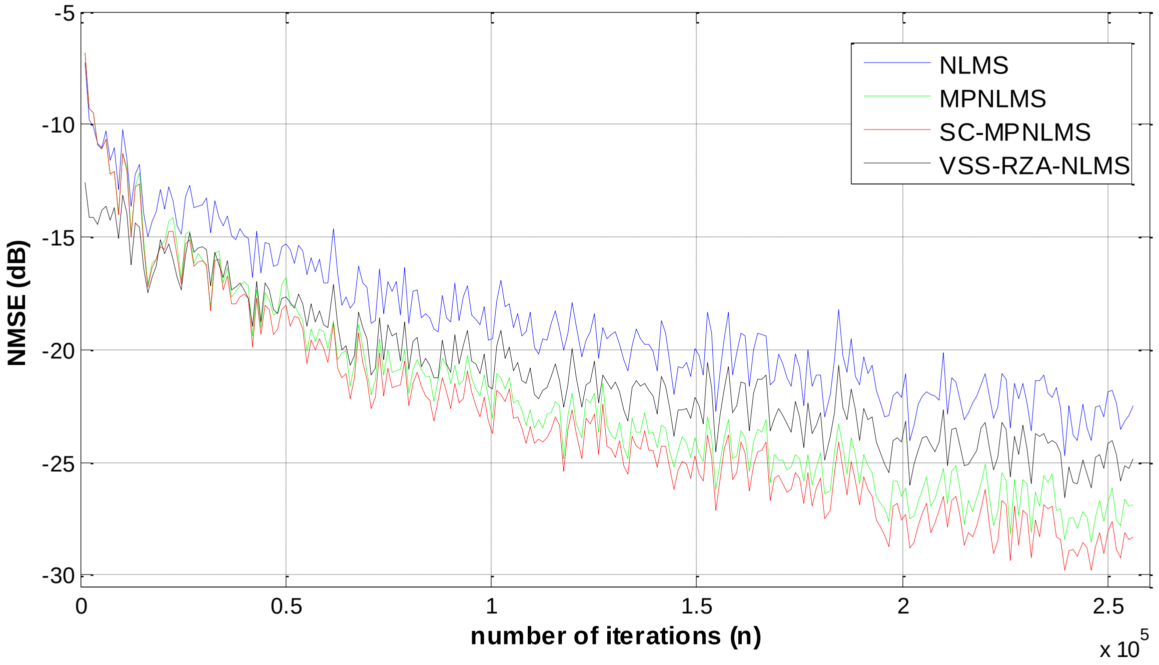

Figure 3.26b: WGN-AR(20) input. ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

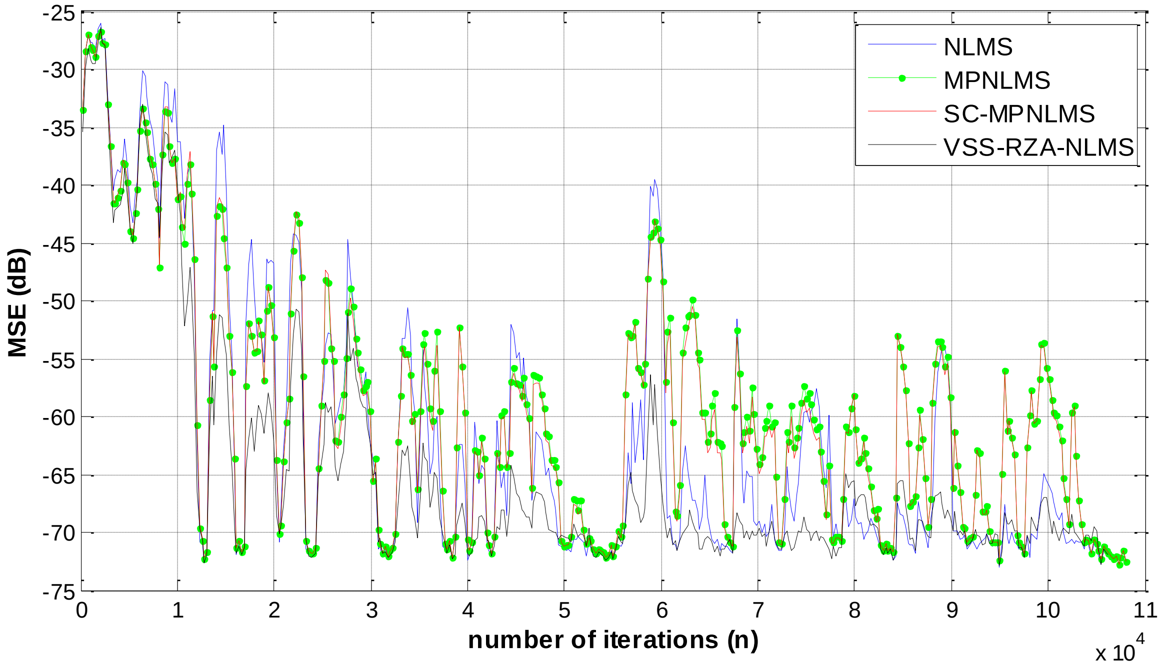

Figure 3.27a: Speech input. Car system with L=256 and ξ =0.5138 (less sparse). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

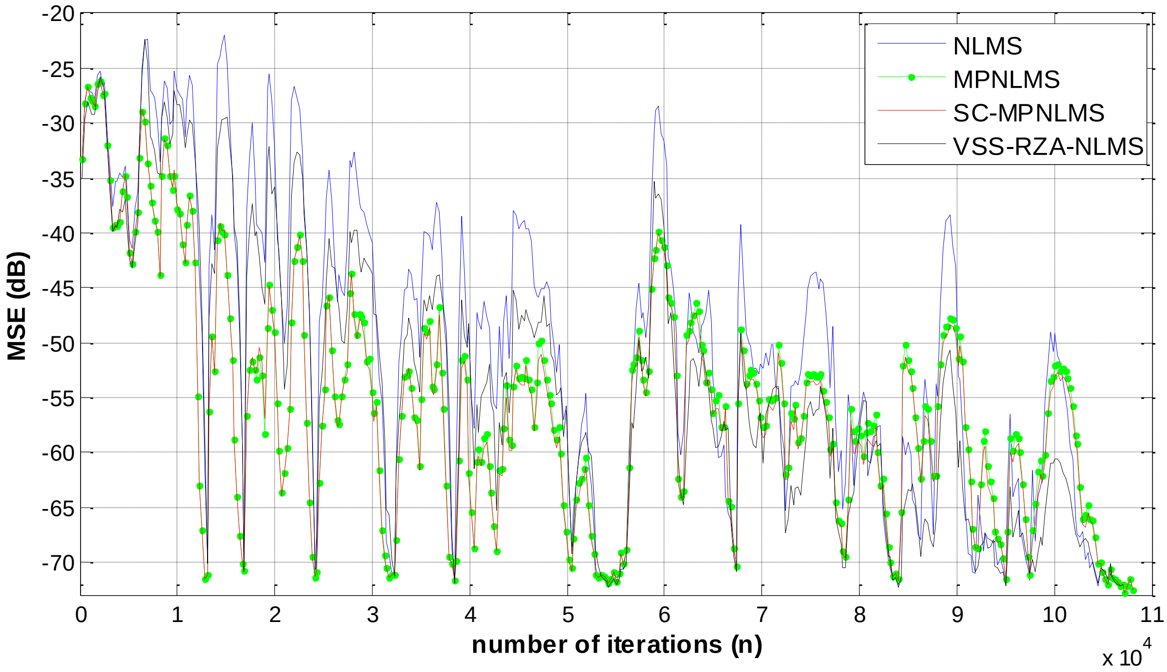

Figure 3.27b: Speech input. Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), MPNLMS (μ=0.3), SC-MPNLMS (μ=0.3, λ=6.0) and VSS-RZA-NLMS (ρ_{RZA}=0.003σ_n^2, ε_{RZA}=30, μ_{max}=1.0 and C=10^{-7}). Output with SNR=50dB.

3.7. Summary

Each one of the discussed algorithms has its advantages and its drawbacks in terms of many criteria such as convergence speed, estimation accuracy and computational complexity. Depending on system nature, application requirements and user objectives, some algorithms are more favorable than others. There is always a trade-off to optimize the quality-to-cost ratio of the chosen algorithm. If two algorithms behave similarly for a particular application, then, the one with less computational complexity is the most favorable.

From the previous results, we conclude generally that VSS and SC algorithms outperform ISS and conventional SA algorithms respectively but they have more computational complexity.

The main achieved results from the previous simulations in stationary and non-stationary cases are summarized in Tables 3.1 and 3.2 respectively. The value 5 is given for the best algorithm performance (either fastest convergence speed, or the most precise estimation accuracy, or the lowest computational complexity) where the value 1 is assigned for the worst (either slowest convergence speed, or the least precise estimation accuracy, or the highest computational complexity). The values in between indicate the closeness either to the best (e.g. 4) or to the worst (e.g. 2).

- *Stationary-inputs case*

Table 3.1: Recapitulation of the main obtained results (stationary inputs).

STATIONARY	Dispersive and less sparse IR		Strongly sparse systems IR		Comput. complexity
Algorithm	**Speed of convergence**	**Estimation accuracy**	**Speed of convergence**	**Estimation accuracy**	
NLMS	4	5	1	2	5
PNLMS	1	1	4	4	4
SC-PNLMS	3	3	4	4	3
IPNLMS	4	4	4	4	3.5
SC-IPNLMS	4	4	4.5	4.5	3
MPNLMS	2	2	5	5	2
SC-MPNLMS	3	3.5	5	5	1
ZA-NLMS	4	4	1	1	4.5
VSS-ZA-NLMS	5	5	3.5	4.5	3.5
RZR-NLMS	4	4	1	1.5	4
VSS-RZA-NLMS	5	5	3.5	5	2

- *Non-stationary-inputs (speech) case*

Figures 3.28 and 3.29 show the performance of the SC algorithms (SC-PNLMS, SC-IPNLMS and SC-MPNLMS) for speech input (non-stationary) with sparse car system and sparse ACN system respectively. Tests for dispersive cases have also been investigated. The comparison results of non-stationary-inputs case are summarized in Table 3.2.

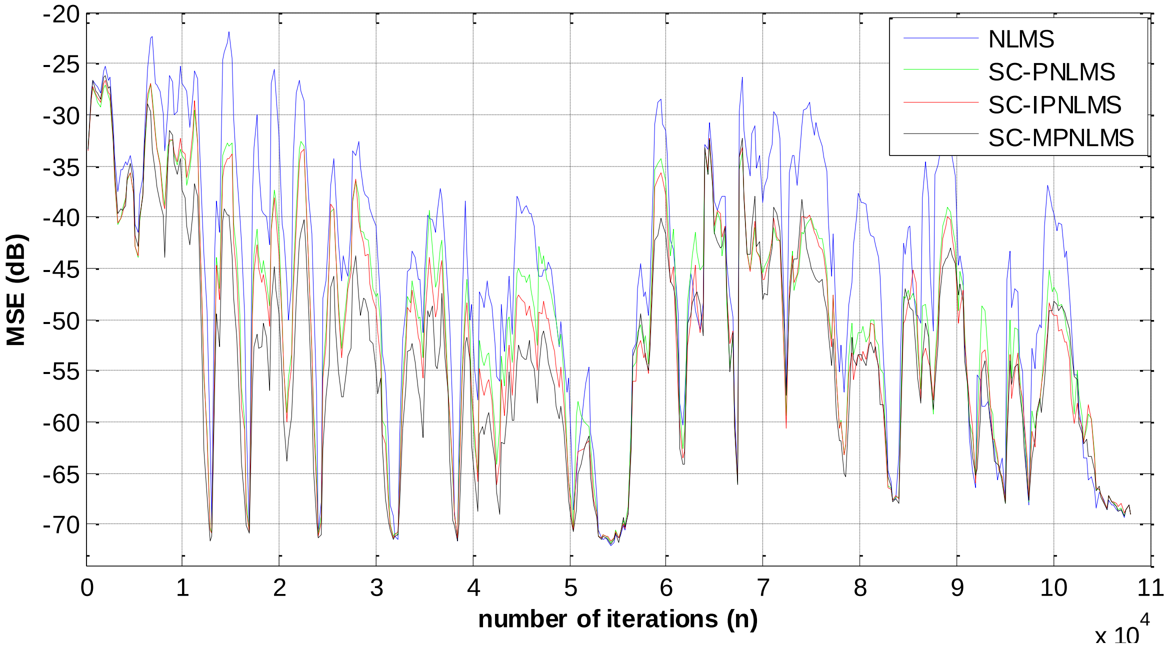

Figure 3.28: Speech input. Car system with L=1024 and ξ =0.7410 (more sparse). NLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0), SC-IPNLMS (μ=0.3, α=-0.5) and SC-MPNLMS (μ=0.3, λ=6.0). Output with SNR=50dB. An abrupt change of the impulse response is applied at n=63744.

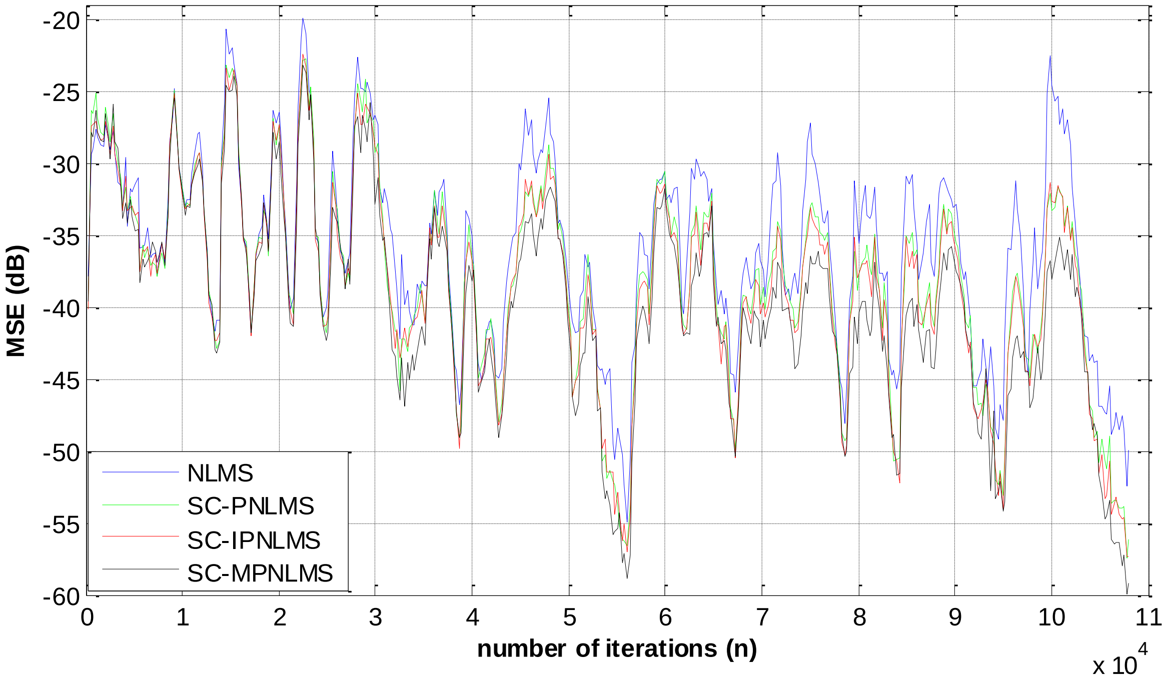

Figure 3.29: Speech input. ACN system with L=8192 and ξ =0.6199 (more sparse). NLMS (μ=0.3), SC-PNLMS (μ=0.3, λ=6.0), SC-IPNLMS (μ=0.3, α=-0.5) and SC-MPNLMS (μ=0.3, λ=6.0). Output with SNR=50dB.

Table 3.2: Recapitulation of the main obtained results (speech-input).

SPEECH	Dispersive and less sparse IR		Strongly sparse systems IR		Comput. complexity
Algorithm	**Speed of convergence**	**Estimation accuracy**	**Speed of convergence**	**Estimation accuracy**	
NLMS	4	4	2	2	5
PNLMS	1	1	3.5	3.5	4
SC-PNLMS	1.5	1.5	3.5	3.5	3
IPNLMS	2.5	2.5	4	4	3.5
SC-IPNLMS	3	3	4	4	3
MPNLMS	2	2	5	5	2
SC-MPNLMS	2	2	5	5	1
ZA-NLMS	3	3	1	1	4.5
VSS-ZA-NLMS	5	5	3	3	3.5
RZR-NLMS	3	3	1	1	4
VSS-RZA-NLMS	5	5	3	3	2

CONCLUSION

This report addressed the problem of identifying long acoustic impulse responses using sparse adaptive filtering algorithms. Because of the implementation easiness and low complexity cost of the NLMS algorithm, this work focuses on the study and comparison of the most known and recent adaptive filtering NLMS-based algorithms for sparse and non-sparse AIRs, emphasizing on the achievement of fast convergence rate and good accuracy with relatively low computational complexity.

The trade-off between convergence speed and the steady state MSE is an important issue in the context of system identification and AEC. This issue can be balanced by choosing the suitable algorithm with the appropriate parameters for the adaptive filtering process.

A series of simulations were carried out both in synthetic and real different-sparsity AIRs for NLMS, its classical proportionate SA versions (PNLMS, IPNLMS and MPNLMS) and their SC upgrades (SC-PNLMS, SC-IPNLMS and SC-MPNLMS) as well as some recent CS-based algorithms namely ZA-NLMS, RZA-NLMS, VSS-ZA-NLMS and VSS-RZA-NLMS.

These experiments helped to analyze and investigate the algorithms strengths and weaknesses in terms of convergence and re-convergence speed, estimation accuracy and computational complexity.

For classical ISS-algorithms, NLMS gives better performance in non-sparse systems, whereas MPNLMS performs well in sparse impulse responses. It is found also that, generally, VSS-algorithms are better than ISS ones and SC-algorithms are better than SA ones for most sparseness levels.

Moreover, IPNLMS/SC-IPNLMS exhibit an overall better performance in a stationary-input variable-sparsity system where VSS-ZA-NLMS/VSS-RZA-NLMS perform better for non-stationary inputs (speech) especially in non-sparse systems. If many algorithms give approximately the same performance, the one with the least computational complexity is the most favorable.

FUTURE WORKS

The work within this report could be further extended in a number of directions, as below

Firstly, it should be addressed that all sparse and non-sparse NLMS-based algorithms still suffer from the limitation that the convergence time is determined by the *ratio* of the maximum to the minimum eigenvalues of the correlation matrix of the input signal (spectral dynamic range) [54]. This ratio is related to the ratio of the maximum to the minimum input power spectrum (commonly called *spectral dynamic range* or the *spectral flatness measure* [42] of the input signal).

In another hand, RLS-based algorithms have fast convergence rate that is independent to the spectral dynamic range of the input signal [1]. Several recent works have been conducted to reduce the complexity of RLS and Fast-RLS (FRLS) algorithms while keeping their good performance [82], [86]-[92]. Efficient applications of sparse algorithms techniques on these algorithms are worthy topics of future studies that may obtain a new family of robust and low-complexity sparse algorithms.

Moreover, a combination of the ZA-NLMS (or RZA-NLMS) and the PNLMS algorithms is proposed in [93]. The resultant algorithms are named the (R)ZA-PNLMS which incorporate the PNLMS control-matrix $\mathbf{Q}$ into the (R)ZA-NLMS update equations. Although this combination had not been discussed in this report, because it increases the complexity with a negligible improvement in the performance of the classical PNLMS algorithm, but it opens a wide range of future possibilities to design new improved sparse algorithms that combine CS-based and PNLMS-type families.

Further developments and more designed algorithms could always be obtained by making more combinations of different techniques used in sparse adaptive filtering algorithms whether in the time-domain or in the frequency-domain. Likewise, on the way to yield new possibilities, different frameworks (LMS; affine projection algorithms (APA) [45], [46], [57], [83], [94], RLS [35]-[37], [95]; the least mean fourth (LMF) adaptive algorithm framework [96]-[101]; compressed sensing [6], [21]-[24], [70]-[74]; wavelets [27], [69], [102], [103]; … etc.) may possibly be combined using some reciprocity schemes such as the so-called cooperative-learning (CL) [104] and different sparseness measures can be incorporated [41], and other norms (e.g.: $\boldsymbol{\ell_p}$-norm,…) can be used [32], [105], [106].

Furthermore, the rigorous analysis [85], [107], [108] concerning the performance of any new algorithm is a commendable subject of research too. It involves many details such as stability conditions and effects of variation of different parameters for different input signals (stationary, non-stationary) and systems with different sparseness levels.

APPENDICES

APPENDIX A

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

ACN	Audio-conference
AEC	Acoustic echo cancellation
AEC/NEC	Acoustic echo cancellation or Network echo cancellation
AIR(s)	Acoustic impulse response(s)
ANSI	American National Standards Institute
APA	Affine projection algorithms
AR	Auto-regressive
AR(20)	Auto-regressive stationary process of order 20
ASCE	Acoustic sparse channel estimation
BER	Bit error rate
BP	Basis pursuit
CL	Cooperative-learning
COP	Criterion of performance
CS	Compressed or compressive sensing or sampling
dB	Decibel
e.g.	‘exempli gratia’ (Latin) which means for example
EG	Exponentiated gradient
EM	Expectation maximization
EMSE	Excess MSE
eq.	Equation
Figures x.xx-y.yy(a&b)	Means all figures; from Figure ‘x.xx(a)’ to ‘y.yy(b)’

FIR	Finite impulse response
FRLS	Fast-RLS
HDTV	High definition television
i.e.	Used for clarification and to show the true meaning
IPNLMS	Improved PNLMS
IR(s)	Impulse response(s)
ISS	Invariable (or invariant)-step-size
ISS-NLMS	Invariable -step-size NLMS
ISS-RZA-NLMS	Invariable-step-size Reweighted zero attracting NLMS
ISS-ZA-NLMS	Invariable-step-size zero attracting NLMS
KHz	Kilo-Hertz
LASSO	Least absolute shrinkage and selection operator
LMF	Least mean fourth
LMS	Least mean square
Log	Logarithm
LRMS	Loudspeaker-room-microphone system
MATLAB	Matrix laboratory
MP	Matching pursuit
MPNLMS	Mu-law (μ-law) PNLMS
ms	Millisecond
MSE	Mean square error
NEC	Network echo cancellation
NLMS	Normalized LMS
NMSE	Normalized MSE (time-average)
(N)MSE	NMSE or MSE
NP	Non-deterministic polynomial
PNLMS	Proportionate NLMS
R/ZA-NLMS	ZA-NLMS and RZA-NLMS
RIAs	Réponses impulsionnelles acoustiques
RLS	Recursive least square
RZA-LMS	Reweighted zero attracting (or attractor) LMS
RZA-NLMS	Reweighted zero attracting (or attractor) NLMS
RZA-ISS-NLMS	Reweighted zero attracting invariable-step-size NLMS

RZA-VSS-NLMS	Reweighted zero attracting variable-step-size NLMS
R/ZA-NLMS	ZA-NLMS and RZA-NLMS
(R)ZA-NLMS	ZA-NLMS or RZA-NLMS
SA	Sparseness-aware
SC	Sparseness-controlled improved NLMS
SC-IPNLMS	Sparseness-controlled
SC-MPNLMS	Sparseness-controlled μ-law proportionate NLMS
SC-PNLMS	Sparseness-controlled proportionate NLMS
SNR	Signal noise ratio
SOE	Signal operating environment
USA	United States of America
USASI	USA Standards Institute
VSS	Variable-step-size
VSS-NLMS	Variable-step-size NLMS
VSS-RZA-NLMS	Variable-step-size Reweighted zero attracting NLMS
VSS-ZA-NLMS	Variable-step-size zero attracting NLMS
WGN	White Gaussian noise
ZA-LMS	Zero attracting (or attractor) LMS
ZA-NLMS	Zero attracting (or attractor) NLMS
ZA-ISS-NLMS	Zero attracting invariable-step-size NLMS
ZA-VSS-NLMS	Zero attracting variable-step-size NLMS
[xx]-[yy]	Means including all references; from number ‘xx’ up to number ‘yy’

General Notations

$\mathbf{X}$	Matrix quantity
$\mathbf{x}$ or $\boldsymbol{x}$	Vector quantity
x or X	Scalar quantity
$x(n)$	Function of a discrete variable at time index n
$\hat{\cdot}$	Estimated quantity

Operators

$[.]^{\boldsymbol{H}}$	Hermitian transposition operator
$[.]^{\boldsymbol{T}}$	Non-conjugate matrix transposition operator
$[.]^{\mathbf{-1}}$	Matrix inverse operator
$\|.\|$	Absolute operator
$\langle\,.\,\rangle$	Time averaging (over 256 samples in our simulations)
$diag\{.\}$	Diagonal operator
$E\{.\}$	Expectation operator
$e^{(.)}$	Exponential function
$\ln(.)$	Natural logarithmic function
$\log_{10}(.)$	Common (base 10) logarithmic function
$max\{.\}$	Maximum function
$\mathrm{sgn}(\,.\,)$	Signum function
$\|\|.\|\|_1$	ℓ_1-norm
$\|\|.\|\|_2$	ℓ_2-norm (Euclidean norm)
$\|\|.\|\|_p$	ℓ_p-norm
∇	Gradient vector
$\hat{\nabla}$	Estimation of gradient vector
$\frac{\partial(.)}{\partial(.)}$	Partial derivative

Symbols and variables

$[.]_{m\times n}$	Matrix of dimension m rows and n columns
$\mathbf{0}_{m\times n}$	Null matrix of dimension m rows and n columns
C	Positive threshold parameter for VSS-NLMS based algorithms
$C\sim\mathcal{O}(1/\mathrm{SNR})$	C is proportional to the order of $(1/\mathrm{SNR})$
E_0	The input signal power which is set equal to unity in reference [32]
$\mathbf{Q}(n)$	Diagonal step-size control matrix
$\mathbf{R}$	Auto-corellation matrix
c	Non-zero scalar
cst	Constant
$e(n)$	*a priori* error

$e_p(n)$	*a posteriori* error
$\mathbf{h}$	True impulse response
h_i	i^{th} element of $\mathbf{h}$
$\widehat{\mathbf{h}}(n)$	Estimated impulse response
$\hat{h}_i(n)$	i^{th} element of $\widehat{\mathbf{h}}(n)$
$\widehat{\mathbf{h}}_{opt}$	Wiener-Hopf optimum estimated solution for the IR
L	Length of adaptive filter
L_p	Leading zeros models the length of the bulk delay
L_u	$L_u = L - L_p$, the length of the decaying window
n	Sample iteration
$\mathbf{b}$	$L_u \times 1$ vector, defined as a zero-mean white Gaussian noise (WGN) sequence with variance σ_b^2
$\boldsymbol{p}(n)$	Update vector used to estimate $\mu(n)$ of VSS-NLMS based algorithms
$\mathbf{p}$	Cross-correlation vector
$\mathbf{p'}$	$L \times 1$ vector, it ensures elements in the 'inactive' region are small but non-zero and is an independent zero-mean WGN sequence with variance $\sigma_{p'}^2$
$q_l(n)$	l^{th} diagonal element of $\mathbf{Q}(n)$
$\mathbf{u}$	$L \times 1$ vector, used to model a synthetic IR
$w(n)$	Additive noise in LRMS
$y(n)$	True system output (desired signal)
$\hat{y}(n)$	Adaptive filter output
$\mathcal{O}(.)$	Order of
$\mathcal{J}(n)$	Cost function
$\mathcal{J}_{..\mathrm{X}..}(n)$	Cost function of the algorithm ' X '
α	Control parameter for the IPNLMS based algorithms
β	Smoothing factor for VSS-NLMS based algorithms
γ	Parameter to prevent filter coefficients from stalling during initialization stage
$\delta_{..\mathrm{X}..}$	Regularization parameter of the algorithm ' X ' to avoid division by zero

ε_{RZA}	Positive threshold for RZA-NLMS based algorithms
ϵ	Vicinity value for MPNLMS based algorithms
κ_l	l^{th} threshold value for PNLMS based algorithms
κ_{min}	Minimum threshold value for PNLMS based algorithms
λ_{RZA}	Regularization parameter for RZA-NLMS based algorithms
λ_{ZA}	Regularization parameter for ZA-NLMS based algorithms
μ	Invariable step-size parameter
μ_{max}	Maximum step-size for VSS-NLMS based algorithms
$\mu(n)$	Variable step-size parameter
$\mu_{..\mathrm{X}..}$	Invariable step-size parameter of the algorithm ‘ X ’
ν	Compensation factor for MPNLMS based algorithms
$\xi(\mathbf{h})$	Sparseness measure of the vector $\mathbf{h}$
$\hat{\xi}(n)$	Sparseness measure of the vector $\widehat{\mathbf{h}}(n-1)$
ρ	Parameter to prevent individual filter coefficients from stalling when their magnitudes are much smaller than the magnitude of the largest coefficient
$\rho(n)$	Variable ρ with respect to $\hat{\xi}(n)$ for SC-PNLMS and SC-MPNLMS
ρ_{RZA}	Zero-attraction controller (regularization step-size) for RZA-NLMS based algorithms
ρ_{ZA}	Zero-attraction controller (regularization step-size) for ZA-NLMS based algorithms
σ_b^2	Variance of the vector $\mathbf{b}$
σ_n^2	Variance of the system noise
$\sigma_{p\prime}^2$	Variance of the vector $\mathbf{p}'$
σ_x^2	Variance of the input signal
ψ	$\in \mathbb{Z}^+$, the decay constant
ϑ_{max}	The largest eigen-value of $\mathbf{R}$

APPENDIX B

ALGORITHMS TABLES

This Appendix provides the pseudo-codes of all studied algorithms in this work.

Table B.1: Pseudo-code of NLMS algorithm [1], [39], [53].

```
Initialization (typical values):    $L = length\ of\ the\ filter$
$0 < \mu < 2$,    $\delta_{NLMS} = \frac{L}{100} . \sigma_x^2$
$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:
For $n = 1, 2, \ldots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$

$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n).\mathbf{x}(n) + \delta_{NLMS}}$
End
```

Table B.2: Pseudo-code of VSS-NLMS algorithm [33], [57].

```
Initialization (typical values):    $L = length\ of\ the\ filter$, $C \sim \mathcal{O}(1/\mathrm{SNR}) = 10^{-6}$
$0 < \mu_{max} < 2$,    $0 < \beta < 1$,    $\delta_{NLMS} = \frac{L}{100} . \sigma_x^2$
$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$,    $\boldsymbol{p}(0) = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:
For $n = 1, 2, \ldots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$
```

```
$\boldsymbol{p}(n) = \beta\boldsymbol{p}(n-1) + (1-\beta)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + C}$

$\mu(n) = \mu_{max}\frac{\boldsymbol{p}^T(n)\boldsymbol{p}(n)}{\boldsymbol{p}^T(n)\boldsymbol{p}(n) + C}$

$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}}$

End
```

Table B.3: Pseudo-code of PNLMS algorithm [1], [11], [39], [62].

```
Initialization (typical values):    $L = length\ of\ the\ filter$

$0 < \mu < 2\,,\quad \delta_{NLMS} = \frac{L}{100}\,.\,\sigma_x^2\,,\quad \delta_{PNLMS} = \frac{\delta_{NLMS}}{L},\quad \rho = \frac{5}{L},\quad \gamma = 0.01$

$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:

For $n = \mathbf{1}, \mathbf{2}, \ldots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
$\kappa_{min}(n-1) = \rho \times max\{\gamma, |\hat{h}_0(n-1)|, |\hat{h}_1(n-1)|, \cdots, |\hat{h}_{L-1}(n-1)|\}$
$\kappa_l(n-1) = max\{\kappa_{min}(n)\,, |\hat{h}_l(n-1)|\},\quad 0 \le l \le L-1$
$q_l(n-1) = \frac{\kappa_l(n-1)}{\frac{1}{L}\sum_{i=0}^{L-1}\kappa_i(n-1)}\,,\quad 0 \le l \le L-1$
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu\frac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{PNLMS}}$
End
```

Table B.4: Pseudo-code of IPNLMS algorithm [1], [11], [39], [43].

```
Initialization (typical values):    $L = length\ of\ the\ filter$

$0 < \mu < 2\,,\quad -1 \le \alpha \le 1,\quad \delta_{NLMS} = \frac{L}{100}\,.\,\sigma_x^2\,,\quad \delta_{IPNLMS} = \frac{(1-\alpha)}{2L}\delta_{NLMS},$

$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$
```

Update, processing and adaptation:

```
For $n = \mathbf{1}, \mathbf{2}, \dots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
$q_l(n-1) = \dfrac{(1-\alpha)}{2L} + \dfrac{(1+\alpha)\left|\hat{h}_l(n-1)\right|}{2\left\|\hat{\mathbf{h}}(n-1)\right\|_1 + \delta_{IPNLMS}}, \qquad 0 \le l \le L-1$
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \dfrac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{IPNLMS}}$
End
```

Table B.5: Pseudo-code of MPNLMS algorithm [1], [11], [39], [62].

Initialization (typical values): $L = length\ of\ the\ filter, \quad \nu = 1000,$

$$0 < \mu < 2, \qquad \delta_{NLMS} = \frac{L}{100}\,.\,\sigma_x^2, \qquad \delta_{MPNLMS} = \frac{\delta_{NLMS}}{L}, \qquad \rho = \frac{5}{L}, \qquad \gamma = 0.01,$$

$$\hat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$$

Update, processing and adaptation:

```
For $n = \mathbf{1}, \mathbf{2}, \dots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
$F\left(\left|\hat{h}_l(n-1)\right|\right) = \ln\left(1 + \nu.\left|\hat{h}_l(n-1)\right|\right), \qquad 0 \le l \le L-1$
$\kappa_{min}(n-1) = \rho \times max\left\{\gamma, F\left(\left|\hat{h}_0(n-1)\right|\right), F\left(\left|\hat{h}_1(n-1)\right|\right), \cdots, F\left(\left|\hat{h}_{L-1}(n-1)\right|\right)\right\}$
$\kappa_l(n-1) = max\left\{\kappa_{min}(n), F\left(\left|\hat{h}_l(n-1)\right|\right)\right\}, \qquad 0 \le l \le L-1$
$q_l(n-1) = \dfrac{\kappa_l(n-1)}{\frac{1}{L}\sum_{i=0}^{L-1}\kappa_i(n-1)}, \qquad 0 \le l \le L-1$
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \dfrac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{MPNLMS}}$
End
```

Table B.6: Pseudo-code of SC-IPNLMS algorithm [19], [43].

```
Initialization (typical values):  $L = length\ of\ the\ filter$ ,  $4 \le \lambda \le 6$

$0 < \mu \le 1$ ,   $-1 \le \alpha \le 1$,   $\delta_{NLMS} = \frac{L}{100} \cdot \sigma_x^2$ ,   $\delta_{SC-IPNLMS} = \frac{(1-\alpha)\delta_{NLMS}}{2L}$,

$\hat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:

For $n = \mathbf{1}, \mathbf{2}, \ldots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
    if $n < L$
        $q_l(n-1) = \frac{(1-\alpha)}{2L} + \frac{(1+\alpha)|\hat{h}_l(n-1)|}{2\|\hat{h}_l(n-1)\|_1 + \delta_{SC-IPNLMS}}$ ,   $0 \le l \le L-1$
    else
        $\hat{\xi}(n) = \frac{L}{L-\sqrt{L}}\left\{1 - \frac{\|\hat{\mathbf{h}}(n-1)\|_1}{\sqrt{L}\,\|\hat{\mathbf{h}}(n-1)\|_2}\right\}$,   $n \ge L$
$q_l(n-1) = \left[\frac{(1-0.5\hat{\xi}(n))}{L}\right]\frac{(1-\alpha)}{2L} + \left[\frac{(1+0.5\hat{\xi}(n))}{L}\right]\frac{(1+\alpha)|\hat{h}_l(n-1)|}{2\|\hat{\mathbf{h}}(n-1)\|_1 + \delta_{SC-IPNLMS}}$ ,
        $0 \le l \le L-1$
    end
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\frac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{SC-IPNLMS}}$
End
```

Table B.7: Pseudo-code of SC-PNLMS algorithm [19], [43].

```
Initialization (typical values):  $L = length\ of\ the\ filter$, $4 \le \lambda \le 6$

$0 < \mu \le 1$ ,   $\delta_{NLMS} = \frac{L}{100} \cdot \sigma_x^2$ ,   $\delta_{SC-PNLMS} = \frac{\delta_{NLMS}}{L}$,  $\gamma = 0.01$

$\hat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$
```

```
Update, processing and adaptation:
For $n = \mathbf{1, 2, \ldots}$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
    if n $< L$
        $\rho(n) = \frac{5}{L}$
    else
        $\hat{\xi}(n) = \frac{L}{L-\sqrt{L}}\left\{1 - \frac{\|\hat{\mathbf{h}}(n-1)\|_1}{\sqrt{L}\,\|\hat{\mathbf{h}}(n-1)\|_2}\right\}, \qquad n \geq L$
        $\rho(n) = e^{-\lambda\hat{\xi}(n)}, \qquad n \geq L$
    end
$\kappa_{min}(n-1) = \rho(n) \times max\{\gamma, |\hat{h}_0(n-1)|, |\hat{h}_1(n-1)|, \cdots, |\hat{h}_{L-1}(n-1)|\}$
$\kappa_l(n-1) = max\{\kappa_{min}(n), |\hat{h}_l(n-1)|\}, \qquad 0 \leq l \leq L-1$
$q_l(n-1) = \frac{\kappa_l(n-1)}{\frac{1}{L}\sum_{i=0}^{L-1}\kappa_i(n-1)}, \qquad 0 \leq l \leq L-1$
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\frac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{SC-PNLMS}}$
End
```

Table B.8: Pseudo-code of SC-MPNLMS algorithm [19], [43].

```
Initialization (typical values):   $L = length\ of\ the\ filter,\ 4 \leq \lambda \leq 6,\ \nu = \frac{1}{\varepsilon} = 1000$
$0 < \mu \leq 1, \qquad \delta_{NLMS} = \frac{L}{100}.\sigma_x^2, \qquad \delta_{SC-MPNLMS} = \frac{\delta_{NLMS}}{L}, \quad \gamma = 0.01$
$\hat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:
For $n = \mathbf{1, 2, \ldots}$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
```

```
    if n < $L$
        $\rho(n) = \frac{5}{L}$
    else
        $\hat{\xi}(n) = \frac{L}{L-\sqrt{L}}\left\{1 - \frac{\|\hat{\mathbf{h}}(n-1)\|_1}{\sqrt{L}\,\|\hat{\mathbf{h}}(n-1)\|_2}\right\}, \qquad n \geq L$
        $\rho(n) = e^{-\lambda\hat{\xi}(n)}, \qquad n \geq L$
    end
$F\left(|\hat{h}_l(n-1)|\right) = \frac{\ln\left(1+\nu|\hat{h}_l(n-1)|\right)}{\ln(1+\nu)}, \qquad 0 \leq l \leq L-1$
$\kappa_{min}(n-1) = \rho(n) \times max\{\gamma, F\left(|\hat{h}_0(n-1)|\right), F\left(|\hat{h}_1(n-1)|\right), \cdots, F\left(|\hat{h}_{L-1}(n-1)|\right)\}$
$\kappa_l(n-1) = max\{\kappa_{min}(n), F\left(|\hat{h}_l(n-1)|\right)\}, \qquad 0 \leq l \leq L-1$
$q_l(n-1) = \frac{\kappa_l(n-1)}{\frac{1}{L}\sum_{i=0}^{L-1}\kappa_i(n-1)}, \qquad 0 \leq l \leq L-1$
$\mathbf{Q}(n-1) = diag\{q_0(n-1) \quad q_1(n-1) \quad \cdots \quad q_{L-1}(n-1)\}$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\frac{\mathbf{Q}(n-1)\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{Q}(n-1)\mathbf{x}(n) + \delta_{SC-MPNLMS}}$
End
```

Table B.9: Pseudo-code of ZA-NLMS (or ISS-ZA-NLMS) algorithm [32].

```
Initialization (typical values):    $L = length\ of\ the\ filter,$
$0 < \mu < 2, \qquad \delta_{NLMS} = \frac{L}{100}.\sigma_x^2, \ \sigma_n^2 = 10^{-\mathrm{SNR}/10}, \ \rho_{ZA} = 0.003\sigma_n^2,$
$\hat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:
For $n = \mathbf{1}, \mathbf{2}, \ldots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$

$e(n) = y(n) - \mathbf{x}^T(n)\,\hat{\mathbf{h}}(n-1)$
```

```
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{ZA}.\mathrm{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)$
End
```

Table B.10: Pseudo-code of RZA-NLMS (or ISS-RZA-NLMS) algorithm [32].

```
Initialization (typical values):   $L = length\ of\ the\ filter,$
$0 < \mu < 2\,,\quad \delta_{NLMS} = \frac{L}{100}\,.\sigma_x^2\ ,\quad \sigma_n^2 = 10^{-\mathrm{SNR}/10},\ \rho_{RZA} = 0.003\sigma_n^2,\ \ \varepsilon_{RZA} = 30$
$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0)\quad \hat{h}_1(0)\quad \cdots\quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$

Update, processing and adaptation:
For $n = \mathbf{1}, \mathbf{2}, \dots$ (iterations)
$\mathbf{x}(n) = [x(n)\quad x(n-1)\quad \cdots\quad x(n-L+1)]^T$
$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{RZA} \frac{\mathrm{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1 + \varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|}$
End
```

Table B.11: Pseudo-code of VSS-ZA-NLMS (or ZA-VSS-NLMS) algorithm [33].

```
Initialization (typical values):   $L = length\ of\ the\ filter,\ C \sim \mathcal{O}(1/\mathrm{SNR}) = 10^{-7}$
$0 < \mu_{max} < 2,\ \ \delta_{NLMS} = \frac{L}{100}\,.\sigma_x^2\ ,\quad \sigma_n^2 = 10^{-\mathrm{SNR}/10},\ \rho_{ZA} = 0.003\sigma_n^2,$
$\widehat{\mathbf{h}}(0) = [\hat{h}_0(0)\quad \hat{h}_1(0)\quad \cdots\quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1},\qquad \boldsymbol{p}(0) = \mathbf{0}_{L\times 1},\qquad 0 < \beta < 1,$

Update, processing and adaptation:
For $n = \mathbf{1}, \mathbf{2}, \dots$ (iterations)
$\mathbf{x}(n) = [x(n)\quad x(n-1)\quad \cdots\quad x(n-L+1)]^T$
$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$
$\boldsymbol{p}(n) = \beta\boldsymbol{p}(n-1) + (1-\beta)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + C}$
$\mu(n) = \mu_{max}\frac{\boldsymbol{p}^T(n)\boldsymbol{p}(n)}{\boldsymbol{p}^T(n)\boldsymbol{p}(n) + C}$
```

```
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{ZA}.\operatorname{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)$
End
```

Table B.12: Pseudo-code of VSS-RZA-NLMS (or RZA-VSS-NLMS) algorithm [34].

```
Initialization (typical values):   $L = length\ of\ the\ filter$, $C \sim \mathcal{O}(1/\text{SNR}) = 10^{-7}$
$0 < \mu_{max} < 2$,   $\delta_{NLMS} = \frac{L}{100}.\sigma_x^2$ ,   $\sigma_n^2 = 10^{-\text{SNR}/10}$,  $\rho_{ZA} = 0.003\sigma_n^2$,   $\varepsilon_{RZA} = 30$
    $\widehat{\mathbf{h}}(0) = [\hat{h}_0(0) \quad \hat{h}_1(0) \quad \cdots \quad \hat{h}_{L-1}(0)]^T = \mathbf{0}_{L\times 1}$,   $\boldsymbol{p}(0) = \mathbf{0}_{L\times 1}$,   $0 < \beta < 1$,

Update, processing and adaptation:
For $n = \mathbf{1}, \mathbf{2}, \dots$ (iterations)
$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-L+1)]^T$
$e(n) = y(n) - \mathbf{x}^T(n)\,\widehat{\mathbf{h}}(n-1)$

$\boldsymbol{p}(n) = \beta\boldsymbol{p}(n-1) + (1-\beta)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + C}$
$\mu(n) = \mu_{max}\frac{\boldsymbol{p}^T(n)\boldsymbol{p}(n)}{\boldsymbol{p}^T(n)\boldsymbol{p}(n) + C}$
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu(n)\frac{\mathbf{x}(n)e(n)}{\mathbf{x}^T(n)\mathbf{x}(n) + \delta_{NLMS}} - \rho_{RZA}\frac{\operatorname{sgn}\left(\widehat{\mathbf{h}}(n-1)\right)}{1 + \varepsilon_{RZA}\left|\widehat{\mathbf{h}}(n-1)\right|}$
End
```

REFERENCES

1. Haykin, S., “Adaptive Filter Theory”, 3rd ed. Prentice Hall, Upper Saddle River, New Jersey, (1996), 989 p.

2. Radecki, J., Zilic, Z. and Radecka, K., “Echo cancellation in IP networks”, Proceedings of the 45th Midwest Symposium on Circuits and Systems, Tulsa, Okla, USA, vol. 2, n° 8, (August 2002), 219-222.

3. Krishna, V.V., Rayala, J. and Slade, B., “Algorithmic and Implementation Apsects of Echo Cancellation in Packet Voice Networks”, Proc. Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, vol. 2, (2002), 1252-1257.

4. Hansler, E. and Schmidt, G., Eds., ‘‘Topics in Acoustic Echo and Noise Control’’, Springer-Verlag, Berlin, (2006), 642 p.

5. Schreiber, W., “Advanced Television Systems for Terrestrial Broadcasting”, Proc. IEEE, vol. 83, n° 6, (1995), 958-981.

6. Bajwa, W., Haupt, J., Raz, G. and Nowak, R., “Compressed Channel Sensing”, Prof. IEEE CISS, Princeton, New Jersey, (March 2008), 5-10.

7. Kocic, M., Brady, D. and Stojanovic, M., “Sparse Equalization for Real-Time Digital Underwater Acoustic Communications”, Proc. IEEE OCEANS, vol. 3, (1995), 1417-1422.

8. Duttweiler, D.L., “Proportionate normalized least-mean-squares adaptation in echo cancelers”, IEEE Trans. Speech Audio Process., vol. 8, n° 5, (September 2000), 508-518.

9. Gay, S.L., “An efficient, fast converging adaptive filter for network echo cancellation”, Proc. Asilomar Conf. Signals, Systems, Comput., (November 1998), 394-398.

10. Benesty, J. and Gay, S.L., “An improved PNLMS algorithm”, Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing, Orlando, Florida, USA, (2002), 1881-1884.

11. Deng, H. and Doroslovacki, M., “Improving convergence of the PNLMS algorithm for sparse impulse response identification”, IEEE Signal Processing Letters, vol. 12, n° 3, (2005), 181-184.

12. Homer, J., Mareels, I., Bitmead, R.R., Wahlberg, B. and Gustafsson, A., “LMS estimation via structural detection”, IEEE Trans. Signal Process, vol. 46, (October 1998), 2651-2663.

13. Li, Y., Gu, Y. and Tang, K., “Parallel NLMS filters with stochastic active taps and step-sizes for sparse system identification”, Proc. ICASSP, vol. 3, (2006), 109-112.

14. Kivinen, J. and Warmuth, M.K., “Exponentiated Gradient versus Gradient Descent for Linear Predictors”, Information and Computation, (January 1997), vol. 132, n° 1, 1-64.

14. Hill, S.I. and Williamson, R.C., “Convergence of Exponentiated Gradient Algorithms”, IEEE Trans. on Signal Processing, vol. 49, n° 6, (June 2001), 1208-1215.

16. Martin, R.K., Sethares, W.A., Williamson, R.C. and Johnson, C.R., ‘‘Exploiting sparsity in adaptive filters’’, IEEE Trans. Signal Process, vol. 50, n° 8, (August 2002), 1883-1894.

17. Paleologu, C., Benesty, J. and Ciochina, S., ‘‘Sparse Adaptive Filters for Echo Cancellation’’, Synthesis Lectures on Speech and Audio Processing, Morgan and Claypool Publishing Series, (2010), 124 p.

18. Khong, A.W.H. and Naylor, P.A., “Efficient use of sparse adaptive filters,” Proc. Fortieth Asilomar Conf. Signals, Systems and Computers, (2006), 1375-1379.

19. Loganathan, P., Khong, A.W.H. and Naylor, P.A., “A Class of Sparseness-Controlled Algorithms for Echo Cancellation”, IEEE Trans. Audio, Speech and Language Processing, vol. 17, n° 8, (November 2009), 1591-1601,

20. Garcia, J.A. and Figueiras-Vidal, A.R., “Adaptive Combination of Proportionate Filters for Sparse Echo Cancellation”, IEEE Trans. Audio, Speech and Language Processing, vol. 17, n° 6, (2009), 1087-1098.

21. Donoho, D.L., “Compressed sensing”, IEEE Trans. Information Theory, vol. 52, n° 4, (April 2006), 1289-1306.

22. Candes, E., “Compressive sampling”, Int. Congress of Mathematics, vol.3, (2006), 1433-1452.

23. Baraniuk, R., “Compressive sensing”, IEEE Signal Processing Magazine, vol. 25, (March 2007), 21-30.

24. Avishy, Y.C., Lyudmila S.M. and Simon J.G., ‘‘Compressed Sensing and Sparse Filtering’’, Springer-Verlag, Berlin Heidelberg, (2014), 501 p.

25. Mallat, S.G. and Zhang, Z., “Matching pursuits with time-frequency dictionaries”, IEEE Trans. Signal Process., vol. 41, n° 12, (December 1993), 3397-3415.

26. Hoffman, A.J., “On greedy algorithms that succeed”, Surveys in Combinatorics, (1985), 97-112.

27. Pati, Y.C., Rezaiifar, R., and Krishnaprasad, P.S., “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition”, Proc. 27th Asilomar Conf. Signals, Syst. Comput., (1993), 40-44.

28. Chen, S. and Donoho, D.L., “Basis pursuit”, in Proc. 28th Asilomar Conf. Signals, Syst. Comput., (1994), 41-44.

29. Tibshirani, R., “Regression shrinkage and selection via the lasso”, J. Royal. Statist. Soc B., vol. 58, (1996), 267-288.

30. Chen, Y., Gu, Y. and Hero, A.O., “Sparse LMS for system identification”, Proc. IEEE ICASSP, Taipei, Taiwan, (April 2009), 3125-3128.

31. Chen, Y., Gu, Y., and Hero, A.O., “Regularized Least-Mean-Square Algorithms”, Arxiv preprint, (December 2010), 1-9.

32. Gui, G. and Adachi, F., ‘‘Improved least mean square algorithm with application to adaptive sparse channel estimation’’, EURASIP J. Wirel. Commun. Netw., vol. 2013, n° 1, (2013),1-18.

33. Gui, G., Peng, W., Xu, Li., Liu, B. and Adachi, F., ‘‘Variable-step-size based sparse adaptive filtering algorithm for channel estimation in broadband wireless communication systems’’. EURASIP J. Wirel. Commun. Netw., vol. 2014, (2014),1-7.

34. Gui, G., Dai, L., Shinya, K., and Adachi, F., ‘‘Variable Earns Profit: Improved Adaptive Channel Estimation using Sparse VSS-NLMS Algorithms’’. EURASIP J. Wirel. Commun. Netw., vol. 2014, (2014),1-7.

35. Babadi, B., Kalouptsidis, N., and V. Tarokh, “SPARLS: The sparse RLS algorithm”, IEEE Trans. Signal Processing, vol. 58, n° 8, (2010), 4013-4025.

36. Angelosante, D. and Giannakis, G.B., “RLS-Weighted LASSO for Adaptive Estimation of Sparse Signals”, Proc. IEEE ICASSP, Taipei, Taiwan, (April 2009), 3245-3248.

37. Angelosante, D., Bazerque, J.A., and Giannakis, G.B., “Online Adaptive Estimation of Sparse Signals: Where RLS Meets the $\boldsymbol{\ell_1}$-Norm”, IEEE Trans. Signal Processing, vol. 58, n° 7, (2010), 3436-3447.

38. Kopsinis, Y., Slavakis, K. and Theodoridis, S., “Online Sparse System Identification and Signal Reconstruction using Projections onto Weighted ℓ_1-Balls”, IEEE Trans. Signal Processing, vol. 59, n° 3, (March 2011), 936-952.

39. Huang, Y., Benesty, J., and Chen, J., ‘‘Acoustic MIMO Signal Processing’’, Springer-Verlag New York Inc., Secaucus, New Jersey, USA, (2005), Chap. 4, 59-84.

40. Liu, L., ‘‘On Improvement of Proportionate Adaptive Algorithms for Sparse Impulse Response’’, PhD thesis, Kochi University of Technology, Japan, (September 2009), 6-7.

41. Samalla, K. and Satyanarayan, Ch., ‘‘Modified Sparseness Controlled IPNLMS Algorithm Based on $\boldsymbol{\ell_1}$, $\boldsymbol{\ell_2}$ and $\boldsymbol{\ell}_\infty$ Norms’’, I.J. Image, Graphics and Signal Processing, (April 2013), 18-29.

42. Manolakis, D.G., Ingle, V.K. and Kogon, S.M., ‘‘Statistical and Adaptive Signal Processing’’, Artech House, Inc., (2005), 796 p.

43. Loganathan, P., ‘‘Sparseness-controlled Adaptive Algorithms for Supervised and Unsupervised System Identification’’, PhD thesis, Department of Electrical and Electronic Engineering, Imperial College, London, (April 2011), 168 p.

44. Benallal, A. and Arezki, M., ‘‘A fast convergence normalized least-mean-square type algorithm for adaptive filtering’’, Int. J. Adapt. Control Signal Process., (2013), 1-8.

45. Yang, Z., Zheng, R. and Grant, S., ‘‘Proportionate affine projection sign algorithms for network echo cancellation’’, IEEE Transactions on Audio, Speech and Language Processing, vol. 19, n° 8, (November 2011), 2273-2284.

46. Gansler, T., Gay, S.L., Benesty, J. and Sondhi, M. M., ‘‘A robust proportionate affine projection algorithm for network echo cancellation’’, Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, (June 2000), 793-796.

47. Lamare, R.C., Yukawa, M. and Sampaio-Neto, R., ‘‘Efficient acoustic echo cancellation with reduced-rank adaptive filtering based on selective decimation and adaptive interpolation’’, IEEE Transactions on Speech, Audio and Language Processing, vol. 16, n° 4, (May 2008), 696-710.

48. Elko, G.W., Diethorn, E. and Gansler, T., ‘‘Room impulse response variation due to thermal fluctuation and its impact on acoustic echo cancellation’’, in Proc. Intl. Workshop Acoust. Echo Noise Control (IWAENC), (2003), 67-70.

49. Sabine, H., ‘‘Room acoustics’’, Transactions of the IRE Professional Group on Room Acoustics, vol. 1, n° 4, (July 1953), 4-12.

50. Loganathan, P., ‘‘Adaptive Echo Cancellation’’, Final year project report, Department of Electrical and Electronic Engineering, Imperial College, London, (2007), 71 p.

51. Farhang-Boroujeny, B., “Adaptive Filters: Theory and Applications”, 2nd ed. John Wiley, UK, (2013), 800 p.

52. Benesty, J. et al., ‘‘Advances in Network and Acoustic Echo Cancellation’’, Springer-Verlag, Berlin, Heidelberg, (2001), 31-54.

53. Petraglia, M.R., ‘‘New Adaptive Algorithms for the Rapid Identification of Sparse Impulse Responses’’, Signal Processing, Sebastian Miron (Ed.), ISBN: 978-953-7619-91-6, InTech, (March 2010), 1-18.

54. Treichler, J.R., Johnson, C.R. and Larimore, M.G., ‘‘Theory and Design of the Adaptive Filters’’, Prentice Hall, (2001), 350 p.

55. Widrow, B. and Stearns, S.D., ‘‘Adaptive Signal Processing’’, Prentice-Hall, Englewood Cliffs, New Jersey, (1985), 528 p.

56. Sayed, A.H., ‘‘Fundamentals of Adaptive Filtering’’, John Wiley & Sons, New Jersey, (2003), 1125 p.

57. Shin, H.C., Sayed, A.H. and Song W.J., “Variable Step-Size NLMS and Affine Projection Algorithms’’, IEEE Signal Processing Letters, vol. 11, N° 2, (February 2004), 132-135.

58. Adapa, N.S. and Bollu, S., ‘‘Performance Analysis of different Adaptive Algorithms based on Acoustic Echo Cancellation’’, MSc Thesis in Electrical Engineering, Blekinge Institute of Technology, Sweden (2013), 54 p.

59. Homer, J., Mareels I., Bitmead R.R., Wahlberg B. and Gustafsson, A., ‘‘LMS estimation via structural detection’’, IEEE Trans. Signal Processing, vol. 46, (October 1998), 2651-2663.

60. Sugiyama, A., Sato, H., Hirano, A. and Ikeda S., ‘‘A fast convergence algorithm for adaptive FIR filters under computational constraint for adaptive tap-position control’’, IEEE Trans. Circuits Syst. II, vol. 43, (September 1996), 629-636.

61. Deng, H. and Doroslovacki, M., ‘‘Proportionate adaptive algorithms for network echo cancellation’’, IEEE Trans. Signal Processing, vol. 54, (May 2006), 1794-1803.

62. Xu, L. and Ju, Y., ‘‘Improving Convergence of the MPNLMS Algorithm for Echo Cancellation’’, 3rd International Conference on Advanced Computer Control (ICACC), (2011), 198-201.

63. Khong, A.W.H., Benesty, J. and Naylor, P.A., “An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation,” in Proc. European Signal Processing Conference, (September 2005), 1-4.

64. Ahmad, R., Khong, A.W. and Naylor P.A., ‘‘Proportionate frequency domain adaptive algorithms for blind channel identification’’, in Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 5, (May 2006), 29-32.

65. Deshpande, A. and Grant S.L., ‘‘A new multi-algorithm approach to sparse system adaptation’’, in Proc. European Signal Processing Conf. (EUSIPCO), (2005), 1-4.

66. Naylor, P.A, Cui, J. and Brookes, M., ‘‘Adaptive algorithms for sparse echo cancellation’’, Signal Processing, Elsevier, vol. 86, (2006), 1182-1192.

67. Hoyer, P.O., ‘‘Non-negative matrix factorization with sparseness constraints’’, Journal of Machine Learning Research, vol. 5, (November 2004), 1457-1469.

68. Benesty, J., Huang, Y.A., Chen, J. and Naylor, P.A., ‘‘Selected methods for acoustic echo and noise control’’, E. Hänsler and G. Schmidt, Eds. Springer, (2006), Chap. 5, 125-153.

69. Starck, J.L., Fionn, M. and Fadili, J.M., ‘‘Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity’’, Cambridge University Press, (2010), 316 p.

70. Qiu, C., Lu, W. and Vaswani, N., ‘‘Real-time dynamic MR image reconstruction using Kalman filtered compressed sensing’’, In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, (2009), 393-396.

71. Carmi, A., Kanevsky, D. and Ramabhadran, B., ‘‘Bayesian compressive sensing for phonetic classification’’, in: Proceedings of the international conference on acoustics, speech and signal processing, (2010), 4370-4373.

72. Li, H., Shen C. and Shi, Q. ‘‘Real-time visual tracking using compressive sensing’’, in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), (2011), 1305-1312.

73. Mei, X., Ling, H., ‘‘Robust visual tracking and vehicle classification via sparse representation’’, IEEE Trans Pattern Anal Mach Intell (PAMI), vol. 33, n° 11, (2011), 2259-2272.

74. Warnell, G. and Chellappa, R., ‘‘Compressive sensing in visual tracking, recent developments in video surveillance’’, in: El-Alfy H (ed), InTech, (2012), 1-20.

75. Joshi, S. and Boyd, S., ‘‘Sensor selection via convex optimization’’, IEEE Trans. Signal Process., vol. 57, n° 2, (2009), 451-462.

76. Kingsbury, B., ‘‘Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling’’, in: Proceedings of the ICASSP, (2009), 3761-3764.

77. Natarajan, B.K., “Sparse approximate solutions to linear systems”, SIAM J. Comput., vol. 24, n° 2, (1995), 227-234.

78. Gui, G., Peng, W. and Adachi, F., ‘‘Improved adaptive sparse channel estimation based on the least mean square algorithm’’, in IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, (2013), 3130-3134.

79. Candes, E.J., Wakin, M.B. and Boyd, S.P., ‘‘Enhancing sparsity by reweighted ℓ_1-minimization’’, J. Fourier Anal. Appl., vol. 14, n° 5-6, (2008), 877-905.

80. Wen, F., ‘‘System Identification Using Reweighted Zero Attracting Least Absolute Deviation Algorithms’’, (2013), 1-5.

81. Gilloire, A., Moulines, E., Slock, D. and P. Duhamel, ‘‘State of art in echo cancellation’’, in A. R. Figuers-vidal, ed., Digital Signal processing in telecommunication, Springer Berlin, (1996), 45-91.

82. Benallal, A. and Benkrid, A., ‘‘A simplified FTF-type algorithm for adaptive filtering’’, Signal processing, Elsevier, vol. 87, n° 5, (2007), 904-917.

83. Meng, R., ‘‘Sparsity-aware Adaptive Filtering Algorithms and Application to System Identification’’, MSc. Thesis, Department of Electronics, University of York, (December 2011), 77 p.

84. Yang, J. and Sobelman, G.E., ‘‘Sparse LMS with Segment Zero attractors for Adaptive Estimation of Sparse Signals’’, (2010), 422-425.

85. Zhang, S., Zhang, J., ‘‘Transient analysis of zero attracting NLMS algorithm without Gaussian inputs assumption’’, Signal Processing, Elsevier, vol. 97, (2014), 100-109.

86. Djendi, M., Benallal, A., Guessoum, A., Berkani, D., ‘‘Three new versions of the Newton type adaptive filtering algorithm’’, Proceedings of IEEE-ISSPA, Paris, France, vol. 2, (2003), 559-562.

87. Djendi, M., Rahim, M., Guessoum, A., Bouchard, M., Berkani, D., ‘‘Comparative study of new version of the Newton type adaptive filtering algorithm’’, Proceedings of IEEE-ICASSP, Montreal, Quebec, Canada, vol. 2, (2004), 677-680.

88. Djendi, M. and Guessoum, A., ‘‘A new fast Newton-type adaptive filtering algorithm for stereophonic acoustic echo cancellation (SAEC)’’, International Journal of Adaptive Control Signal Processing, vol. 35, n° 2, (2010), 435-444.

89. Kalouptsidis, N., Mileounis, G., Babadi, B. and Tarokh, V., ‘‘Adaptive algorithms for sparse system identification’’, Signal Processing, Elsevier, vol. 91, (2011), 1910-1919.

90. Djendi, M., ‘‘An efficient stabilized fast Newton adaptive filtering algorithm for stereophonic acoustic echo cancellation’’, Computers Electrical Engineering, vol. 38, n° 2, (2012), 938-952.

91. Djendi, M., Bouchard, M., Guessoum, A., Benallal, A., Berkani, D., ‘‘Improvement of the convergence speed and the tracking ability of the fast newton

type adaptive filtering (FNTF) algorithm'', Signal Processing, vol. 86, n° 7, (2006), 1704-1719.

92. Bendoumia, R. and Djendi, M., ''Two-channel variable-step-size forward and-backward adaptive algorithms for acoustic noise reduction and speech enhancement'', Signal Processing, vol. 108, (2015), 226-244.

93. Das, R.L. and Chakraborty, M., ''A Zero Attracting Proportionate Normalized Least Mean Square Algorithm'', Proc. APSIPA-ASC, Hollywood, California, (December 2012), 1-4.

94. Darazirar, I., Djendi, M., ''A two-sensor Gauss-Seidel fast affine projection algorithm for speech enhancement and acoustic noise reduction'', Applied Acoustics, vol. 96, (2015), 39-52.

95. Kalouptsidis, N., Mileounis, G., Babadi, B. and Tarokh, V., ''Adaptive algorithms for sparse system identification'', Signal Processing, Elsevier, vol. 91, (2011), 1910-1919.

96. Walach, E. and Widrow, B., ''The least mean fourth (LMF) adaptive algorithm and its family'', IEEE Transactions on Information Theory, vol. 30, n° 2, (1984), 275-283.

97. Hubscher, P.I. and Bermudez, J.C.M., ''An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm'', IEEE Transactions on Signal Processing, vol. 51, n° 3, (2003), 664-671.

98. Hubscher, P.I., Bermudez, J.C.M. and Nascimento, V.H., ''A mean-square stability analysis of the least mean fourth adaptive algorithm'', IEEE Transactions on Signal Processing, vol. 55, n° 8, (2007), 4018-4028.

99Eweda, E., ''Global stabilization of the least mean fourth algorithm'', IEEE Transactions on Signal Processing, vol. 60, n° 3, (2012), 1473-1477.

100. Eweda, E. and Bershad, N.J., ''Stochastic analysis of a stable normalized least mean fourth algorithm for adaptive noise canceling with a white Gaussian reference'', IEEE Trans. Signal Process., vol. 60, n° 12, (2012), 6235-6244.

101. Gui, G., Mehbodniya, A. and Adachi, F., ‘‘Least mean square/fourth algorithm for adaptive sparse channel estimation’’, in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), London, UK, (2013), 1-5.

102. Noskoski, O.A. and Bermudez, J.C.M., ‘‘Wavelet-packet-based adaptive algorithm for sparse impulse response identification’’, in: Proc. Int. Conf. Acoust., Speech, and Signal Process, Honolulu, Hawaii, USA, vol. 3, (2007), 1321-1324.

103. Ribas, C.H.H., Bermudez, J.C.M. and Bershad, N.J., ‘‘Identification of sparse impulse responses - design and implementation using the partial Haar block wavelet transform’’, Digital Signal Processing, Elsevier, vol. 22, (2012), 1073-1084.

104. Das, R.L. and Chakraborty, M., ‘‘Sparse Adaptive Filters - an Overview and Some New Results’’, Proc. ISCAS, COEX, Seoul, Korea, (May 2012), 2745-2748.

105. Aliyu, M.L., Alkassim, M.A. and Salman, M.S. ‘‘A $\boldsymbol{p}$-norm variable step-size LMS algorithm for sparse system identification’’, Springer-Verlag, London, (2014), 1-7.

106. Zhang, S. and Zhang, J., ‘‘Enhancing the tracking capability of recursive least $\boldsymbol{p}$-norm algorithm via adaptive gain factor’’, Digital Signal Processing, Elsevier, vol. 30, (2014), 67-73.

107. Jahromi, M.N.S., Salman M.S., Hocanin, A. and Kukrer, O., ‘‘Convergence analysis of the zero-attracting variable step-size LMS algorithm for sparse system identification’’, Springer-Verlag, London, (2013), 1-4.

108. Das, R.L. and Chakraborty, M., “Zero Attracting PNLMS Algorithm and its Convergence in Mean”, Arxiv preprint, (July 2015), 1-11.