VSS-LMS ALGORITHMS FOR MULTICHANNEL SYSTEM IDENTIFICATION USING VOLTERRA FILTRERING

Thesis submitted towards the partial fulfilment of requirement for the award of degree of

MASTER OF ENGINEERING

IN

WIRELESS COMMUNICATION ENGINEERING

Submitted by

Sandipta Dutta Gupta

Roll No. 801363028

Under the guidance of

Dr. Amit Kumar Kohli

Associate Professor, ECED

T.U. Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004, PUNJAB, INDIA

JUNE-2015

DECLARATION

I, **Sandipta Dutta Gupta,** hereby declare that the thesis entitled **"VSS-LMS Algorithms for Multichannel System Identification Using Volterra Filtering"** is an authentic record of my own work carried out towards the partial fulfilment for the award of degree of Master of Engineering in Wireless Communication at Thapar University, Patiala, under the supervision of **Dr. Amit Kumar Kohli**, Associate Professor, Electronics and Communication Engineering Department.

The matter presented in this thesis has not been submitted in any other University/Institute for the award of any other degree.

Sandipta

Date: 30/6/15

Sandipta Dutta Gupta

Roll No. 801363028

This is to certify that the above statement made by the student is correct to the best of my knowledge and belief.

Amit Kohli

Date: 30/6/15

Dr. Amit Kumar Kohli

Associate Professor

ECED, TU, Patiala

Countersigned by:

(Dr. Sanjay Sharma)

Professor and Head ECED

Thapar University, Patiala

(Dr. S.S. Bhatia)

Dean of Academic Affairs

Thapar University, Patiala

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor, **Dr. Amit Kumar Kohli,** Associate Professor and P.G. Coordinator, Department of Electronics and Communication Engineering, Thapar University, Patiala for his valuable guidance in carrying out work under his effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for writing of this thesis.

I shall be failing in my duties if I do not express my deep sense of gratitude towards **Dr. Sanjay Sharma,** Professor and Head of the Department of Electronics and Communication Engineering, Thapar University, Patiala, and **Mr. Hemdutt Joshi**, Assistant Professor, Program Coordinator of Wireless Communication, who have been a constant source of inspiration for me throughout this work, and for providing us with adequate infrastructure in carrying the work.

I am greatly indebted to all my friends especially **Mr. Abhishek Bansal, Mr. Bhupinder Singh**, and **Mr. D.S. Kapoor**, who constantly encouraged me and also would like to thank the entire faculty and staff members of Electronics and Communication Engineering Department for their unyielding encouragement. I am also thankful to the authors whose work have been consulted and quoted in this work.

At last but not the least my gratitude towards my parents (**Mr. Swarup Dutta Gupta** and **Mrs. Santa Dutta Gupta**), who always supported me in doing the things my way and whose everlasting desires, selfless sacrifice, encouragement, affectionate blessings and help made it possible for me to complete my degree. I would also like to render my gratitude to the Almighty God who bestowed self-confidence, ability and strength in time to complete this task and for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Place: TU, Patiala **Sandipta Dutta Gupta**

Date: **Roll No. 801363028**

ABSTRACT

Adaptive filtering comprises one of the primary technologies in signal processing and investigates numerous applications in arenas of industry and science. These techniques are employed in a vast range of applications, for example adaptive echo and noise cancellation system, adaptive equalization, and adaptive beamforming. The recent custom in the telecommunication system design is the process of identification and minimization of undesired nonlinearities, as they have an element effect on its rendition. The employ of nonlinear models can tackle all these nonlinearities. Adaptive approaches and algorithms are extensively utilized for the estimation of Volterra kernels, under the constraint of unknown nonlinear system. The correctness of the kernels will investigate the precision of the system model and inverse system, which is incorporated for compensation.

This thesis propounds the adaptive polynomial filtering deploying the multifarious variable step-size least mean square (VSS-LMS) algorithms for the nonlinear Volterra multichannel system identification, and all are compared with a fixed step-size Volterra least mean square (VLMS) algorithm, under the various noise constraints comprising an individual signal-to-noise ratio (SNR). The VSS-LMS algorithm corroborates steady behaviour during convergence, and the step-size of the adaptive filter is altered in compliance with a gradient descent algorithm delineated to abate the squared estimation error in the course of each iteration, and it also revamps tracking rendition in the smoothly time-varying environments to the choice of the parameters and the boundaries of adaptive filter. In multitudinous practical implementations, the autocorrelation matrix of the input signal has the immense eigenvalue spread in the manifestation of nonlinear Volterra filter than in respect of the linear impulse response filter. In such circumstances, an adaptive step-size is a pertinent option to mitigate the unpropitious effects of eigenvalue spread on the convergence of VLMS adaptive algorithm.

The simulation results are exhibited to reinforce the analysis, which compare the VSS-LMS algorithms with fixed step-size of the second-order Volterra filter, and also substantiate that the VSS-LMS algorithms are more robust than the fixed step-size algorithm when the input noise is logistic chaotic type.

Keywords- least mean square (LMS), minimum mean square error (MMSE), system identification, variable step-size least mean square (VSS-LMS), Volterra filter.

TABLE OF CONTENTS

ACRONYMS AND ABBREVIATIONS

2D	Two Dimensional
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
AA	Adaptive Algorithm
AEC	Acoustic Echo Cancellation
AF	Adaptive Filter
AFA	Adaptive Filter Algorithm
ANC	Active Noise Control
APA	Affine Projection Algorithm
AVSS	Aboulnasr Variable Step-Size
AWGN	Additive White Gaussian Noise
BAA	Blind Adaptation Algorithm
BLAST	Bell Laboratories Layered Space-Time
BW	Bandwidth
DSP	Digital Signal Processing
EASE	Ensemble Average Square Error
FPGA	Field Programmable Gate Array
FFF	Fixed Forgetting Factor
FIR	Finite Impulse Response
FM	Frequency Modulation
FSLMS	Filtered-s Least Mean Square

FXLMS	Filtered-x Least Mean Square
GM	Gauss-Markov
GVSS	Generalised Variable Step-Size
HVS	Human Visual System
IC	Integrated Circuit
IO	Input-Output
IRF	Impulse Response Function
ISI	Inter Symbol Interference
KVSS	Karni Variable Step-Size
KwVSS	Kwong Variable Step-Size
LMP	Least Mean p^{th} Power
LMF	Least Mean Fourth
LMS	Least Mean Square
MASE	Mean Asymptotic Square Error
MDVSS	Modified Variable Step-Size
MFE	Mean Fourth Error
MMSE	Minimum Mean Square Error
MSA	Mean Square Algorithm
MSE	Mean Square Error
MVSS	Mathews Variable Step-Size
NLMS	Normalised Least Mean Square
NVFF	Numeric Variable Forgetting Factor
PAM	Pulse Amplitude Modulation

PVSS	Pazaitis Variable Step-Size
QAM	Quadrature Amplitude Modulation
RHS	Right Hand Side
RLS	Recursive Least Square
ROM	Read Only Memory
SE-LMS	Signed Error Least Mean Square
SNR	Signal to Noise Ratio
SONAR	Sound Navigation and Ranging
SR-LMS	Signed Regressor Least Mean Square
SVD	Singular Value Decomposition
TIV	Time-Invariant
TV	Time-Variant
VFXLMS	Volterra Filtered-x Least Mean Square
VLMS	Volterra Least Mean Square
VLSI	Very Large Scale Integration
VS	Volterra Series
VSS-LMS	Variable Step-Size Least Mean Square
WPVSS	Wee-Peng Variable Step-Size
XE-NLMF	Input X and Error X Normalized Least Mean Fourth

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Abstract: *This chapter provides a brief introduction of the thesis. This include background of the nonlinear filters in the field of signal processing, LMS algorithm, VLMS algorithm, VSS-LMS algorithms, objectives, original contributions, and at the end it discusses about the organisation of the thesis.*

In the field of signal processing, linear filters have played a pivotal role because of its inherent simplicity. However, there are numerous practical situations, where non-linear processing models are needed as the behaviour of the linear filters is unacceptable and provides misleading results [1]. These system models don't follow the properties of superposition theorem and thus exhibit certain degree of nonlinearity. Thus to tackle with these problems of nonlinearity, different nonlinear filters, system equalizers, or controllers are used. The most familiar used techniques are polynomial filters, order statistics filters, homomorphic filters, and morphological filters [3, 4] and out of these most widely used polynomial filters are the Volterra filters [1, 5]. Adaptive Volterra filters have turn out to be a compelling option for various nonlinear adaptive applications, such as acoustic echo cancelling (AEC) [2], active noise control (ANC) of nonlinear processes [6,10-14], and diminution of falsifications on the systems related to loudspeakers [7]. In fact, the literature [6, 11-14] suggests numerous techniques for the development of ANC of nonlinear processes for a single channel case. The Volterra system model shows similar behaviour to the Taylor series, but it has an ability to capture 'memory' effects [8].

The LMS algorithm has been profusely used in many applications because of its simplicity, intelligibility and robustness [15]. Step size is the important parameter in LMS algorithm. As, if the step size is huge, the rate of convergence will going to be expeditious, but the steady state MSE will proliferate and vice-versa. Thus, the step size provides an accord between the steady state MSE and convergence rate of the LMS algorithm. An instinctive way to enhance the rendition of the LMS algorithm is to make the step size variable in lieu of fixed, i.e., select large step size values during the inceptive period of the convergence and use small step size when the system is in close proximity to its steady state. Thus, results in VSS-LMS algorithms [16]. Many VSS-LMS algorithms have been propounded during recent years [17-22].

1.1. Adaptive Filters

Due to increasing requirement in a vast field of signal processing applications, it motivates the growth of nonlinear adaptive filtering. The applications and research includes adaptive equalization, echo cancellation, adaptive beamforming, and ANC. All the aforestated applications includes processing of signals that are obtained by systems whose behaviours are unknown a priori. So, in adaptive behaviour, fixed filters fails to obtain a good performance. Hence, an adaptive filters are always the first priority in adaptive behaviour of the signals [28]. However, in many practical applications such as speech processing, modelling of a channel, and systems related to biological applications, TV system characteristics are needed and the parameters related to this system can be linear/nonlinear. There are some nonlinear system models, which is employed to refer nonlinear systems, and among them Volterra model, which represents that for the approximation of continuous real-valued function, a polynomial function is used. We use the filtering technique to trail the dynamics of the Volterra system, and learn the parameters which are not acknowledged to us. There are firm algorithms, which lessens the nonlinear effect in the system. When working in a TVing surroundings, the AF has the extra task of trailing the statistical variations in environmental situations [38], [39]. It is well known that the behaviour of convergence of an AF is a transient phenomenon, but its trailing behaviour is totally a steady state phenomenon i.e., generally, the good convergence nature doesn’t necessarily require to translate into a best trailing behaviour.

An AF is a self-designing filter and a computational device that uses an iterative algorithm to design the filter itself. This iterative algorithm is usually known as AA/ adaptive filtering algorithm (AFA). The adaptive system attempts to accord the relationship between the two signals, which is done in real time in a recursive manner. AFs are often consider as a set of instructions of a program running on an some mathematical processing system, like DSP chip/ microprocessor, or a compiled set of logic simulations implemented in a FPGA [40] or VLSI IC. However, rejecting any fallacy introduced by mathematical distinctness effects in the implementations, the basic action of an AF can be distinct independently of the particular physical realization. The adaptive algorithm (AA) initiates from a prior guess in successive recursion, and converges gradually, to the ideal solution in statistical sense. Thus, the main focus of us should be on the mathematical forms of AFs as compared to their realizations in hardware/software.

An AF is mainly defined by four features:

1. The filter processes the signals.
2. The system's input signal is used to compute the output signal of the filter.
3. The filter's input-output (IO) relationship is altered due to the recursively changing of the system parameters.
4. Adjustment of the system parameters from the time instantaneous to the next can also be described by the AA.

By selecting a particular AF structure, one can decides the number/ type of parameters that can be altered. The AA employed to update the values of the system parameter, which can take on a multitudinous of forms and is frequently derived as a form of idealize procedure that diminishes an error condition that is capable for the task at hand. In this section, the general problems of AFs, and an overview of mathematical analysis of the AF will be discussed.

Adaptive Filtering Problem

Fig. 1.1 depicts a block diagram, where a digital input sample $x(n)$ is given into a system, known as adaptive filter (AF), that calculates its corresponding sample of the output signal $y(n)$ at time instant n. At instant, the design of the AF is not important, excluding the fact that it constitutes adaptable parameters, whose data affect how output signal $y(n)$ is calculated, and $y(n)$ is compared to desired response signal $d(n)$, by differentiating the two samples at n. The alteration signal is known as error signal and it is given by [40]

$$\overline{e}(n) = d(n) - \tilde{y}(n) \tag{1.1}$$

The error signal, $\overline{e}(n)$ is given into a procedure which adapts / alters the parameters of the filter from time instant n to time instant $(n+1)$. The process explained above is usually denoted by the sharp arrow that pierces the AF block in the figure. As the n is increased, filter output becomes a better match to the $d(n)$ through this TVing process, and thus, the magnitude of $\overline{e}(n)$ decreases with respect to time. The term "better" is related to the form of the AA used to adjust the system parameters.

The term "adaptation" in adaptive filtering task, relates to the method/procedure by which the parameters of the models are altered from n to time index $(n+1)$. The types/ numbers of system parameters rely on the computational structure chosen for that particular system [41-43].

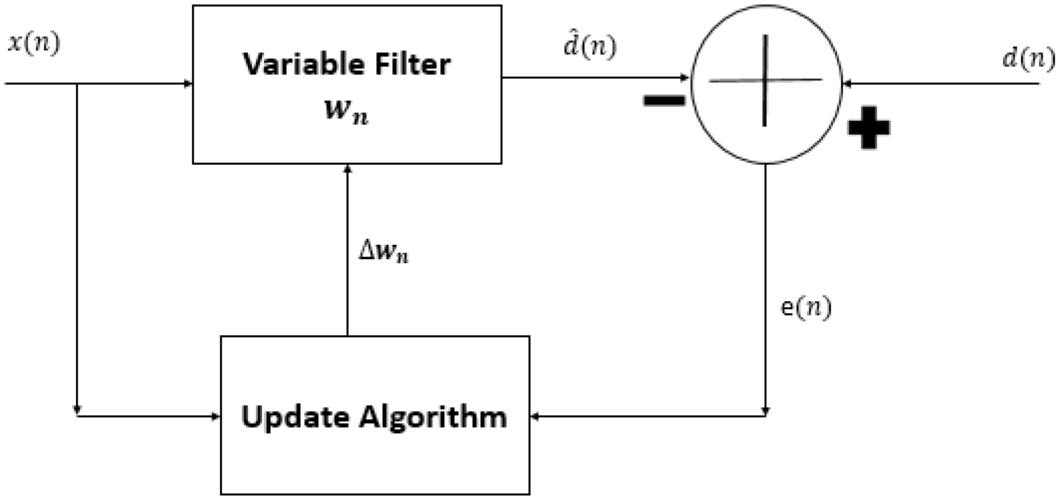

Fig. 1.1. A general block diagram of adaptive filtering problem [28]

Task of an Adaptive Filter

When contemplating the AF problem as described in Fig. 1.1 for the first time, a question comes to the mind regarding AF, “If we have the desired response signal, $d(n)$ already, what is the employ to match it using an AF?” The concept of “matching” output signal to desired response signal with some system makes it vague the delicacy of the adaptive filtering task.

Now, contemplate the following points that concern to many adaptive filtering problems:

1. The amount of interest is not always desired signal, $d(n)$. Our inclination may be to represent in output signal a certain component of desired signal that is contained in input signal, or it may be to separate a component of desired signal $d(n)$. Simultaneously, we may be totally interested in the parameters $w(n)$ and don’t bother about $x(n)$, $y(n)$, or $d(n)$.
2. Sometimes, there are some practical scenarios in which $d(n)$ is not present at all the times. In that case, adaptation prevails only when desired signal, $d(n)$ is present. When $d(n)$ is not present, we employ our most availing parameter estimates to calculate $y(n)$ in an endeavour to estimate $d(n)$.
3. There are some situations in real-world, where $d(n)$ is not at all available. In such situation, one can employ extra information about the behaviours of a “speculative” $d(n)$, like its predicted statistical behaviour to form an appropriate estimates of $d(n)$ from the signals present to the AF. These methods are generally called blind adaptation algorithm (BAA). The reality that such plan even work is a

commendation both to the originality of the developers of the AA and to the technological experience of the adaptive filtering field.

The relationship between input signal, $x(n)$ and desired signal, $d(n)$ can vary according to the time. In such cases, the AF endeavours to change its values to follow the alters in this relationship as "digitized and encoded" by the two series $x(n)$ and $d(n)$. This behaviour is referred to as trailing.

The features of an AFA is evaluated based on the following factors:

Rate of convergence

It investigates the number of iteration/recursion required to get the domain of a steady-state solution. This value explains the transient characteristic of the algorithm. This is given as the number of iterations needed for the algorithm, under static conditions to converge optimum Weiner MSE sense. It is explained to achieve the maximum rate possible. Since in many practical scenarios the system model has to meet extreme deadlines, the convergence must be expeditious enough to meet the deadlines and of course not affect the performance.

Misadjustment

This point discusses steady-state nature of the algorithm. This is a computable approximation of the expanse through which the ensemble mean final value of the MSE exceeds the MMSE produced by the ideal Winer filter.

Computational Complexity

Algorithm's demand of solution/resources is determined by a factor "Computational complexity". The time of processing and the memory data storage denotes the word "resources". AAs are recursive methods that iterate their work all rounds. Thus, to calculate the computational power required, we have to determine the number of operations in one iteration. In fact, for block-based algorithms, this data/value calculates the amount of operations required to process N simultaneous samples whereas for sample-based operations it is only and only for a single sample value. Thus, when differentiating different algorithms together, the idea is to calculate the number of operations/steps for a block of N samples.

Robustness

This can be seen as a combined need of highest immunity against errors (internal errors), like round-of errors, quantization errors, and obscureness to external errors. Sometimes, it is good for the algorithm to be fragile to certain changes of the surrounding, such as noise processes and nonstationary of acoustic signals. The accord between relative robustness and minimal sensitivity is often a tough task to correct and solve.

Application of Adaptive Filters

An AF is important whenever the filter's input signal is unknown or TVing. Since its all parameters are altering with respect to the situations of an ambient environment, hence, minute or no a-priori data/ information is needed about the signals at the input. Thus, it can be seen that the adaptive systems are efficient of working with nonstatic signals, provided that they can be taken static at least in small duration of time. A lot of application scenarios of AFs are indicated in Fig. 1.2 and Fig. 1.3 i.e., system identification process and inverse filtering process respectively. For the first case, the AF is used to approximate a system, which is not known. Both the systems, i.e., unknown and AF are driven by the equivalent input signal and the filter coefficients are altered in such a way, that the both output signal and the unknown system's output resembles each other. For the second case, i.e., inverse modelling, the AF is used in sequence with the unknown system. The learning/adaptive algorithm tries to recompense the influence of the non *a priori* system on the test signal $u[n]$ by reducing the squared difference between the AFs output and the detained test signal.

Other applications of AFs besides above explained are:

1. Teleconferencing/ videoconferencing
2. Channel equalization and suppression of interference
3. Voice control
4. Hands-free telephony

Markov Model for Time-Varying Systems

The behaviour of TVing systems can be explained by either a deterministic process or a stochastic process one. For a deterministic system, periodic TVing behaviour are reported to be an apt one. One can often finds needful to model this type of system by a first-order Markov model.

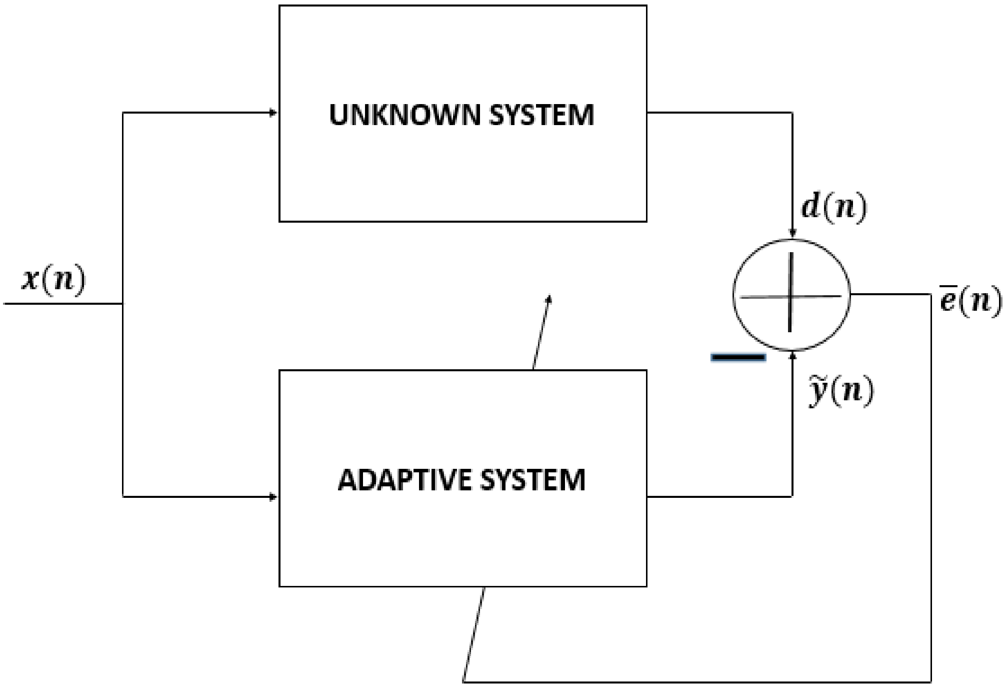

Fig. 1.2. System identification of an adaptive filter [44]

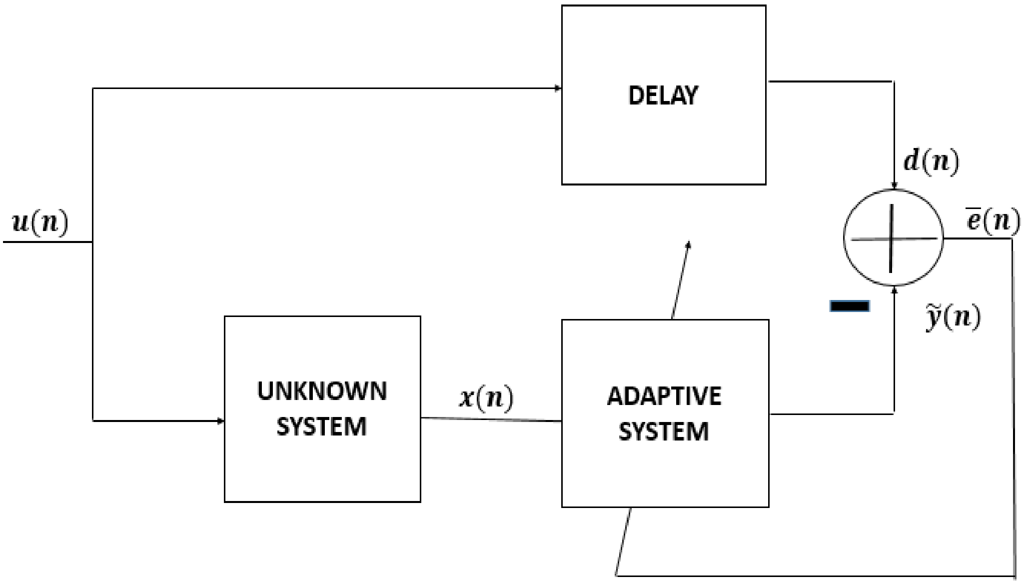

Fig. 1.3. Inverse modelling of adaptive filters [44]

Any model/system may become non static in one of the following ways given below:

1. The desired response signal, $d(n)$ provides the frame of reference, which may be TVing. Such case arises, when in system identification a transversal AF is used to model a TVing systems, where the correlation matrix of the AF's input remains fixed in this case, but the cross correlation between the $d(n)$ and $x(n)$ assumes a TVing form.

2. The stochastic system providing the AF’s tap inputs are nonstationary. This condition arises, when an AF is used to equalize a TVing channel. Thus, in that situation, both the correlation and cross correlation matrix assume TVing forms.

Hence the trailing details of a TVing systems are totally dependent on the number and types of the AF used but also are problem specific.

1.2. Nonlinear System

Linear filters are important in a vast number of applications as they are relatively simpler from implementation and conceptual view points. In today’s scenario there are numerous practical situations, where nonlinear processing of the signals are required.

This topic explains nonlinear AFs equipped with polynomial systems of nonlinearity. These systems contemplated are those nonlinear systems in which output signals can be related to the signal input be a truncated Volterra series expansion, or through an iterative nonlinear difference equation. This series expansion can model a vast class of nonlinear systems and is important in adaptive filtering applications as the expansion of the series is a linear combination of the nonlinear functions of the signal input. Such system are tempting as they can approximate numerous nonlinear systems with large parsimony in the employ of coefficients.

Linear filters have played a key role in the growth of numerous signal processing techniques [1]. The undoubted advantage of the linear filters is their simplicity according to the design, implementation, and analysis view point. The use of such filters are usually a straightforward tasks in numerous applications. But, there are many conditions, where the working performance of linear filters is unacceptable. An easy but highly penetrating type of nonlinearity: Saturation- type nonlinearity. We always get misleading results if we try to identify the above explained types of system using linear system. Another condition, where nonlinear systems will do good when linear models will not work and fail, is that of trying to correlate two signals with non-overlapping of spectral components.

Nonlinear system analysis has numerous applications, for example high-speed communications channels usually require nonlinear equalizers for good performance. Although the same system using linear weight/tap delay structures is required and used in many applications, there are many other conditions when they will not work properly at all. R.W. Lucky [45] has proposed that the performance of probability error of data transmission systems working at rates are much better than 4800 bits/s is due almost completely to nonlinear distortion.

Considering an example of telephone transmission, one can see that the nonlinearities comes from inaccuracies, when a signal is compressed. In satellite links (digital satellite), the amplifiers of the satellite are usually taken to the approximate of saturation point and they follow large nonlinear behaviour. Many researchers have employed nonlinear Volterra series representation to perform nonlinear channel equalizer.

Other implementation of nonlinear models:

1. Performance analysis of transmission of data
2. Echo cancellation
3. Adaptive nonlinear noise cancellation
4. Identifying of nonlinear function employed in Gaussian process.

Now, if one considers about the applications of nonlinear filters, they are as follows:

1. Myoelectric signal processing
2. Biological phenomena system
3. Image processing
4. Characterization of devices (especially semiconductor devices)
5. Modelling/ formation of drift oscillations in seas etc.

It is difficult to find an amalgamated framework for explaining any nonlinear systems as compared to linear systems, which are totally distinguished by the unit impulse response function, $d(n)$, of the system. In fact, the researchers doing work on nonlinear filters are enforced to regulate themselves to certain less general systems which are nonlinear.

Some nonlinear filters, which are developed using such systems/models include:

- **Homomorphic filters:** Homomorphic filters are one of the oldest types among all nonlinear filters and they have some applications like in image enhancement, removing multiplicative noise disturbance from the signal input, and in seismic signal processing. In fact one of the most used application of homomorphic filters are human visual system (HVS), which is employed in image coding applications.
- **Order statistic filters:** Order statistic are used because of its simplicity and robustness in nature. As it can be assumed from its name that they are totally based on the order statistics. By the term “order statistic”, we mean that the location/point of any stated data sample in a redisposition of all the samples under assumption in the increasing/ decreasing order of magnitude of the signal input of the filter.

- **Morphological filters:** Morphological filters mainly use geometric functions of the input signals and are used in some applications like edge detection, shape recognition, and others. All the description related to above time invariant filters may be found and can be seen in many articles.
- **Volterra filters:** These are polynomial filters, which will be discussed in detail in subsequent topics. In this thesis, our main focus is on nonlinear polynomial models. Usually these models are more applicable and used in research work than other models.

Two particular cases are taken into consideration:-

1. Representation of adaptive filtering using truncated Volterra series
2. Representation of recursive/iterative nonlinear difference equation/function, which gives IO relationship.

Though the first case can be treated as a special case of second one, we will try to explain both the cases separately. In the field of adaptive nonlinear filtering, the Volterra system is exceptionally famous and has created its own identity and name in the last few years. But on the other hand, theoretical view of nonlinear adaptive filtering using nonlinear feedback system is extremely in its infancy/ early stage; and while these type of systems are splendid from an implementation view point. One can see that there are many problems for which right solutions haven't yet been determined.

1.2.1. Volterra Series

The term "Volterra series" was first introduced by the Spanish mathematician Vito Volterra in his "Theory of Functional". The mathematician Norbert Weiner first analysed the first major/huge application of Volterra's work with respect to nonlinear circuit, who employed them in a basic way to identify a number of difficulties, which include the spectrum of FM system having an input of Gaussian noise. Since then, Volterra series used in a vast application like calculating small distortion terms in the systems (e.g. transistor amplifiers).

The Volterra series model shows same behaviour as Taylor series for nonlinearity. It is just differs from the Taylor series in terms of its capability to get 'memory' effects such as inductors and capacitors. If the output of the system, which is nonlinear in nature, depends entirely on the input at a particular time, the Taylor series can be employed to approximate the response of a nonlinear models/ systems for that particular given input at that time. At all other times, the output of the system in Volterra series depends totally on the input.

In the communication system design the two common methods:

1. Identification
2. Compensation of unrequired nonlinearities

It was observed that in a particular system, undesired nonlinearities will have a principle effect on its performance [46]. There are numerous ways and solutions to minimize the effects of unwanted nonlinearities [47]. With considerable success, the Volterra series have been vastly employed as technique in nonlinear system modelling [48], [49]. At present, however, none basic procedure exists to determine the kernels of nonlinear Volterra system. But, if the order is known and finite, they can be determined for the system. Adaptive methods and AAs are employed for the estimation of Volterra kernel, if the system order is indefinite. The exactness of the kernels will calculate the precision of the system and the precision of the inverse model employed for compensation. Another important factor is the speed of kernel estimation. A fast speed estimation method of kernel may permit the used to develop a higher-order system that allows an even better model representation.

1.2.2. Volterra Model

A simple linear and causal system having memory can be given by the following convolution representation

$$\tilde{y}(t) = \int\limits_{-\infty}^{\infty} h(\partial)\, x(t-\partial) d\partial \tag{1.2}$$

where, $x(t)$ represents input signal, $\tilde{y}(t)$ denotes the output signal, and $h(t)$ is the system impulse response. A memoryless nonlinear system with Taylor series can be given as

$$\tilde{y}(t) = \sum_{n=1}^{\infty} a_n\, [x(t)]^{\, n} \tag{1.3}$$

where, a_n represents the Taylor series coefficient.

Now, a Volterra series amalgamate the above two equations to represent nonlinear system having memory

$$\tilde{y}(t) = \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \dots \int_{-\infty}^{\infty} du_n\, g_n(u_1, \dots, u_n) \prod_{r=1}^{n} x(t-u_r) \tag{1.4}$$

where, $g_n(u_1, \dots, u_n)$ are known as kernels of Volterra system, and ui are time variables. They are labelled instead of ti so to differentiate them from t in a better way. Taking some cases into consideration, for $n = 1$,

$g_1(u_1)$: will be considered as the familiar and same impulse response, $h(t)$ as given in equation (1.2).

For $n > 1$:

g_n: are similar to higher-order impulse responses. These represents to characterize the multifarious orders of nonlinearity.

1.2.3. Order of Volterra Model

- **Zeroth and First-Order Volterra Model**

First, we will consider the zeroth order Volterra system, which is defined as

$$Y_0[x(n)] = h_0 \tag{1.5}$$

where, h_0 is just a constant.

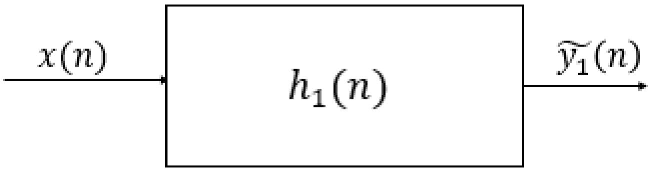

Fig.1.4. Block diagram of first-order linear system [42]

The first-order Volterra system is supremely similar to the linear system. In another way, the linear system comes under the Volterra system.

Let us consider the Fig. 1.4, which represents a general linear system and where $h_1(n)$ indicates the coefficients of the linear filter. The output of the system is given by

$$\tilde{y}_1(n) = x(n) * h_1(n) = \sum_{k=0}^{\infty} h_1(k)\, x(n-k) \tag{1.6}$$

where, * denotes convolution operation. If all the components present in $h_1(n)$ can be given by some linear amalgamation of orthonormal basis $b_m(n)$. Therefore, the equation (1.6) can be denoted as

$$h_i(k) = \sum_{M=0}^{\infty} a_1(m)\, b_m(k) \tag{1.7}$$

where, $a_1(m)$ is a constant.

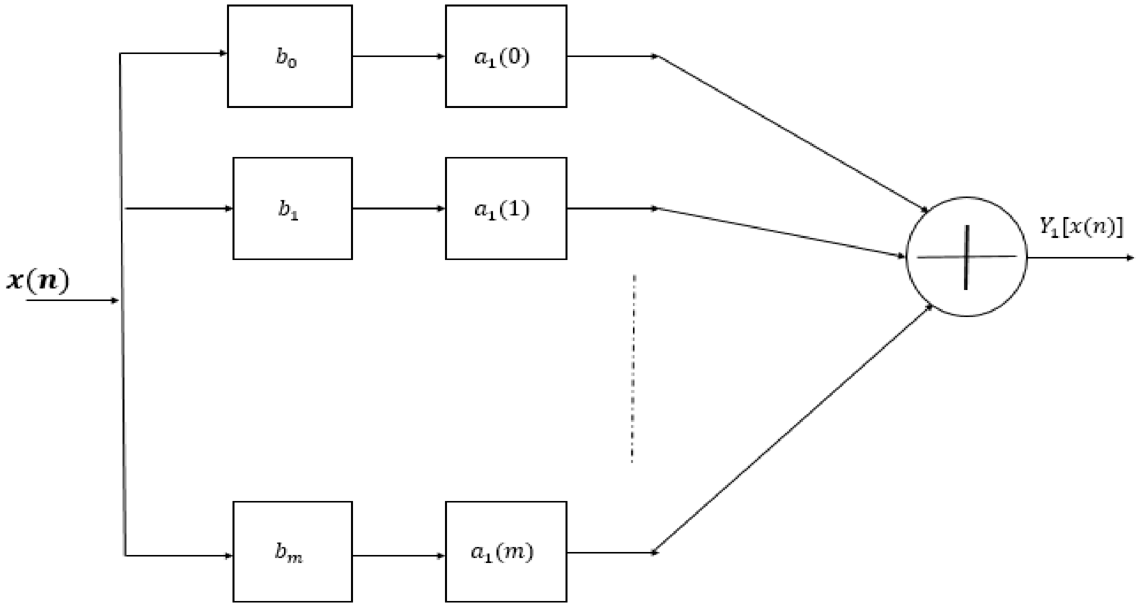

Fig. 1.5. Block diagram of first-order Volterra model [42]

Note: $\{b_m(n), 0 \leq m \leq \infty\}$ represents set of orthonormal basis i.e.,

$$[b_i(n), b_m(n)] = \delta(1 - m) \tag{1.8}$$

where [,] represents the inner product, and $\delta(1 - m)$ indicates dirac delta function.

Thus, substituting equation we get the first- order Volterra function, which is given as

$$\tilde{y}_1(n) = \sum_{k=0}^{\infty} \sum_{M=0}^{\infty} a_1(m)\, b_m(k) x(n - k) \tag{1.9}$$

In first-order Volterra system, we must include the DC component in equation (1.6), which can be represented in terms of the $Y_0[x(n)]$ and $Y_1[x(n)]$ as

$$Y(n) = h_0 + \mathrm{x(n)} * h_1(\mathrm{n}) = Y_0[x(n)] + Y_1[x(n)] \tag{1.10}$$

Thus, from equation (1.9) we can conclude that a basic first-order Volterra system having DC component is one for which the response output to a linear amalgamation of the inputs are similar to the linear amalgamation of the response output of each individual corresponding input.

- **Second-order Volterra Model**

The linear amalgamation concept explained above can be processed to the second-order case, and it is the one for which the yield reaction to a second–order Volterra system is a linear amalgamation of the individual input signals.

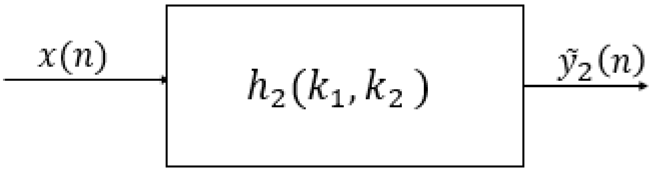

Fig. 1.6. Block diagram of second-order Volterra model

$$\tilde{y}_1 = \sum_{k_1,k_2=0}^{\infty} h_2(k_1,k_2)\, x(n-k_1)(n-k_2) \tag{1.11}$$

where, $h_2(k_1,k_2)$ represents second-order Volterra kernel.

We contemplate the kernel to be symmetric for simplicity and with no loss of generality, which indicates that $h_2(k_1,k_2) = h_2(k_2,k_1)$. Now, if $h_2(k_1,k_2)$ can be represented by the linear amalgamation of orthonormal basis set $b_k(n)$, then $h_2(k_1,k_2)$ can be given as

$$h_2(k_1,k_2) = \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} a_2(m_1,m_2)\, b_{m1}(k_1) b_{m2}(k_2) \tag{1.12}$$

In this equation, all the operations are 2D convolutions.

Substitute equation (1.12) in (1.11), we get [6]

$$\begin{aligned} Y_2(n) = &\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} a_2(0,0)\, b_0(k_1) b_0(k_2) x(n-k_1) x(n-k_2) + \\ &\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} a_2(0,1)\, b_0(k_1) b_1(k_2) x(n-k_1) x(n-k_2) + \\ &\sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty} a_2(1,0)\, b_1(k_1) b_0(k_2) x(n-k_1) x(n-k_2) + \cdots \end{aligned} \tag{1.13}$$

$$\begin{aligned} &= a_2(0,0)[b_0(n) * x(n)] + a_2(1,1)[b_1(n) * b_2(n)] + \cdots \\ &+[a_2(0,1) + a_2(1,0)][b_0(n) * x(n)]\, [b_1(n) * x(n)] + \cdots \end{aligned} \tag{1.14}$$

The following equation, demarcated as the second–order Volterra function

$$\tilde{y}_2 = Y_2[x(n)] \tag{1.15}$$

The above equation signifies second–order standardized function i.e.,

$$\tilde{y}_2[cx(n)] = c_2 Y_2[x(n)] \tag{1.16}$$

Considering a special condition such that the linear orthonormal set constitutes two bases of orthonormal $\{b_m(n), 0 \le m \le 1\}$.

Thus, from eq. (1.12), the second– order Volterra function $\tilde{y}_2(n)$ can be represented as

$$\tilde{y}_2(n) = a_2(0,0)[b_0(n) * x(n)] + a_2(1,1)[b_1(n) * x(n)]$$
$$+[a_2(0,1) + a_2(1,0)][b_0(n) * x(n)][b_1(n) * x(n)] \qquad (1.17)$$

For the above equation, the equivalent block diagram is shown below

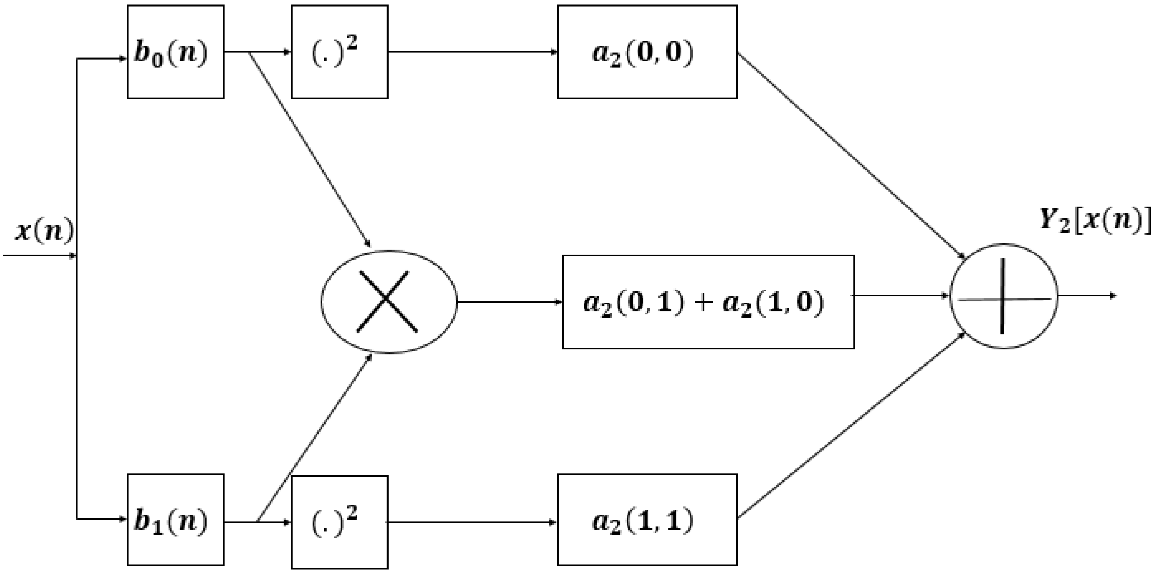

Fig. 1.7. Block diagram of second-order Volterra system with basis set of orthonormal [42]

Based on the explanation above, the basic second-order Volterra system can be described in terms of $Y_0[x(n)], Y_1[x(n)],$ and $Y_2[x(n)]$, which is given as

$$Y(n) = h_0 + \sum_{k_1}^{\infty}\sum_{k_2}^{\infty} h_2(k_1, k_2)x(n - k_1)x(n - k_2)$$
$$= Y_0[x(n)] + Y_1[x(n)] + Y_2[x(n)] \qquad (1.18)$$

To form the VS filter adaptive is simple, linear adaptive systems can be used. The two most common AAs are: **LMS and RLS algorithm.**

The LMS algorithm is simple and in terms of complexity it has computational complexity and minute convergence in minimum square sense. The RLS algorithm as compare to LMS has high complexity and having a fast convergence. It is noted that when a linear expansion is done for an input vector x_N, it will increase the eigenvalue spread, even if the signal input is white. Thus, it is utmost necessary that the selection of algorithm will be such that it has less dependent on the signal input. It is observed that when the eigenvalue spread of the matrix (autocorrelation matrix) is large in LMS, it suffers in convergence speed. One solution to this problem is to use RLS algorithm, but at a high cost of complexity.

Disadvantage of Volterra series:

It degrades its performance badly if it has to sway discontinuities, for example a saturation type of nonlinearity. The reason behind being is that the expansion of VS is referred to the Taylor series having memory which only suits the data well when the function are at least single differentiable, or we can say that when the function are smoothly working.

1.3. Truncated Volterra Series Expansion for Nonlinear Systems

Let us consider input and output signal to be $x(n)$ and $\tilde{y}(n)$ respectively, of a causal nonlinear system and for a discrete time. The VS expansion for output signal $\tilde{y}(n)$ by using the input signal $x(n)$ is expressed as

$$\begin{aligned}\tilde{y}(n) = h_0 &+ \sum_{m_1=0}^{\infty} h_1(m_1) \\ &+ \sum_{m_1=0}^{\infty} \ldots\ldots \sum_{m_2}^{\infty} h_p\left(m_1, m_2, \ldots, m_p\right) x(n-m_1) x(n-m_2) \ldots x\left(n-m_p\right) \\ &+ \cdots \end{aligned} \tag{1.19}$$

where, $h_p\left(m_1, m_2, \ldots, m_p\right)$ represents p^{th} order Volterra kernel.

Volterra system models have been widely used in numerous applications, and it becomes an attraction tool for researchers. Weiner has given a large contribution in the field of nonlinear system, and they have used VS expansion and time-invariant (TIV) nonlinear system identification. As an infinite series expansion expressed in equation (1.19) is not helpful in filtering applications, therefore one must try to use and work with truncate VS expansion, which is expressed as

$$\tilde{y}(n) = h_0 + \sum_{m_1=0}^{M-1} h_1(m_1) + \sum_{m_1=0}^{M-1} \ldots\ldots$$

$$\ldots\ldots\sum_{m_2}^{M-1} h_p\left(m_1, m_2, \ldots, m_p\right) x(n-m_1) x(n-m_2) \ldots x\left(n-m_p\right)$$

$$+\cdots \qquad (1.20)$$

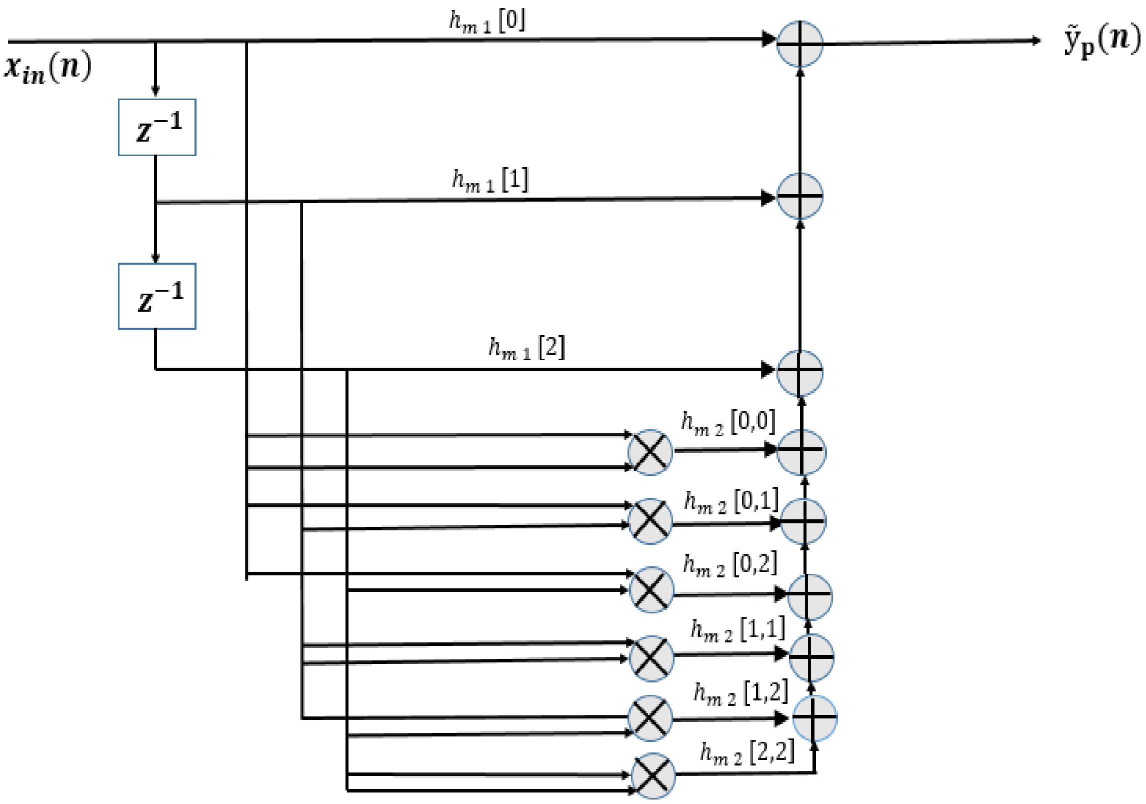

Fig.1.8. Truncated Volterra System of order and delay element of two [1]

Fig.1.8 shows a truncated Volterra system having $U = 2$ and $N - 1 = 2$ delay elements. One can see that the h_0 may be determined outside the ideal adaptive filter structure.Thus, assume that $h_0 = 0$.

Note: The above system is linear in nature at the input signal to each individual coefficient.

For moderate values of M and U: large number of coefficients. When the value of M and U are very small and minute, the representation of truncated Volterra series becomes most useful.

Disadvantage of the Volterra system model:

- Large complexity in implementing the filters, even if the value of M and U are moderate. In fact, much of the practical applications of systems using VS expansion involves low-order models.

1.4. An Overview of Variable Step-size LMS Algorithm

For ease of explanation, the LMS algorithm is framed firstly within the milieu of a system identification model, where real signals are taken into consideration.

The mathematical expression is as follows [16, 23]:

$$d(n) = X_{in}^{T}(n)\, r(n) + c(n) \tag{1.21}$$

where, $d(n)$ is the zero mean desired signal and is a filtered genre of the input signal vector $X_{in}(n)$ corrupted by the noise $c(n)$ and $r(n)$ is the impulse response of primary path $R(z)$. Also, n signifies time index, and $(.)^{T}$ is the transpose operator.

The inaccurate signal is the alteration between the desired signal and the output signal of the AF and is given by

$$\overline{e}(n) = d(n) - X_{in}^{T}(n)\overline{w}(n) \tag{1.22}$$

where, $\overline{w}(n)$ represents weights of the AF. The LMS algorithm for updating the weights is given

$$\overline{w}(n+1) = \overline{w}(n) + \mu'\overline{e}(n)X_{in}(n) \tag{1.23}$$

where, μ' is the step size.

In the VSS-LMS algorithms, equation (1.23) can be written as

$$\overline{w}(n+1) = \overline{w}(n) + \mu'(n)\overline{e}(n)X_{in}(n) \tag{1.24}$$

i.e., μ' is replaced by a variable parameter $\mu'(n)$.

The adaptive step size algorithm is delineated to eliminate the “postulate work” involved in stipulation of the step size parameter, and at the same time gratify the following requirements: a) Fast convergence speed; b) the steady state misadjustment data should be minute, when operating in a stationary scenario; c) under non-stationary environment, the algorithm should be proficient to sense the pace at which the peerless coefficients are changing and step size should be selected in such a way that can result in close estimation in mean-squared-error sense.

Several VSS-LMS algorithms have been propounded to refine the rendition of the LMS algorithm by using massive step sizes at the primary stage of the process and minute step sizes at the later stage when system approaches convergence. Distinctive methods for adaptive step sizes can be found in [17-22].

1.5. Problem Statement

The thesis propounds the following work

- Firstly, the fixed step-size VLMS algorithm is presented for single channel system. This identification model can also be extended to multichannel system of Volterra filter by considering multiple inputs.
- After that, we presented adaptive nonlinear Volterra based VSS-LMS algorithms, which are better than the fixed step-size under a certain noise conditions with different SNRs on a multichannel structure by using the convergence factor as discussed in [17-22].

1.6. Organisation of the Thesis

This thesis has four main chapters: Chapters Three, Four, Five and Six, where adaptive filtering algorithms are obtained and tested with experimental values for fixed step-size and VSS-LMS algorithms for Volterra filter. It has three substantiating chapters: Chapter One, Two, and Seventh, which gives details about the principles, literature survey conducted, and conclusions are drawn at the end of the thesis.

Chapter 1: Introduction: This chapter summarizes the AFs, its filtering problem, introduction to nonlinear systems with the expansion of VS for first-order and second-order, and then the overview of VSS-LMS algorithms are given. It also deliberates about the motivation and contributions of the thesis and gives a brief outline of the thesis.

Chapter 2: Literature Survey: This chapter conducts a literature survey of nonlinear AFs, and also survey the applications of nonlinear AFs. This chapter confers the reader with the essential background theory and data/ information on the state-of-art.

Chapter 3: Adaptive Filter: This chapter gives a detailed study of AF, the utter need for an AF, discusses about the adaptation law, nonlinear AFs, principles of both the adaptive linear FIR and IIR filters, and their present applications.

C**hapter 4: Adaptive Filtering Algorithms:** This chapter gives details on the basics of adaptive filtering practises as well as the expansion and derivation of algorithms employed within this thesis.

Chapter 5: Analysis and Methodology: In this chapter, different methods of VSS-LMS algorithms for multichannel Volterra Filter system are presented, and their mathematical formulations are given for each VSS-LMS algorithms.

Chapter 6: Simulation Results: This chapter propounds the simulation results of all the VSS-LMS algorithms, which are compared with the fixed step-size VLMS algorithms for a multichannel system under various constraint of SNRs.

Chapter 7: Conclusions and Future Scope: Finally in this chapter, we conclude the work by comparing different VSS-LMS algorithms and determining the best algorithm under different SNR conditions and it also provides suggestion for future research and applications.

CHAPTER 2

LITERATURE SURVEY

Abstract: *This chapter provides a literature survey in the field of LMS algorithm, nonlinear filter and various VSS-LMS algorithms are also presented in different literature work. The nonlinear model under contemplation is Volterra model.*

An effective research on AFs has been effectuated for about three decades. Thus, many algorithms and systems have been developed and a vivid body of literature has been formed. Firstly the adaptation law are drafted, which emphasised on the LMS algorithm. Then the linear filters are discussed and at last adaptive non-linear Volterra filters are also presented in this survey. A typical fixed filter is usually linear and time-invariant, as it is used to extricate information from an input time sequence. An adaptive filter adjusts its coefficients automatically. Most of the AFs work in the time domain. In [29], a frequency domain nonlinear adaptive Volterra filter is presented.

In [1], Mathews explained about a nonlinear AFs provided with nonlinear polynomial models. The polynomial systems contemplated are those nonlinear models whose input-output signals can be related through an iterative nonlinear difference equation, or a truncated VS expansion. The Volterra series expansion can configure a vast class of nonlinear systems and is captivating in adaptive filtering as the expansion is a linear amalgamation of nonlinear function of the signal given at the input. He also discussed about the outcome of gradient and recursive LMS adaptive Volterra filters, followed by AAs involving nonlinear difference equations. Such system are captivating because they may be proficient to approximate many nonlinear systems with substantial parsimony in the employ of coefficients. He proposed the Volterra model using different AAs i.e., RLS and LMS algorithms and he also described a lattice structure for polynomial models.

Adaptive nonlinear filters find their numerous applications in various fields. Some applications include echo cancellation [2, 30, 31, 32], equalization [33, 34] and noise reduction [35] for communication channels.In present years adaptive echo cancellation has been broadly investigated in connection with a few new services bring out in the telephone network because it permits for full-duplex transmission on two-wire networks with full bandwidth in both directions. Applications like digital subscriber loop and voiceband data modems support the above concept. In [31], a theoretical study of echo cancellation was conveyed. It was proved that the echo reduces to zero, when there is no noise as the proposed echo canceler converges exorbitantly. A nonlinear adaptive echo canceler was presented in [32], based on the Volterra filter. Distributed arithmetic was used for obtaining the hardware realizations, which were based on the combinational networks. For data signal, an adaptive nonlinear echo canceler was obtained in [30]. A derivation was obtained for a binary series expansion of a nonlinear function of the input data signals, which has finite terms and has the form of the Volterra series. Thus, based on this expansion, hardware implementation and LMS adaptation, was developed.

In earlier years, it was depicted that adaptive nonlinear Volterra filters can be employed for telephone echo cancellation, but the implicit need for a vast number of coefficients has hindered their practical use. Recently, first endeavours have been made to conflict nonlinearities in the acoustic echo path of mobile telephones, which are originated by low-cost audio components. As these endeavours are based on a memoryless nonlinear model, they are restrained to saturation effects, which cannot explain the nonlinear behaviour of small loudspeakers precisely enough. In [2], authors reported real time measurements with an adaptive nonlinear Volterra filter. They verified that the loudspeaker nonlinearities confronted in the echo paths of hand-free telephones had to be modelled with memory. Thus, they proposed an adaptive Volterra filter structure with a minimized number of coefficients. The proposed new structure was experimentally related to echo cancellers with a memoryless nonlinear model employing a real acoustic echo cancelling setup with a minute loudspeaker. With this new technique, an echo minimization improvement of 7 dB over ideal linear adaptive filters is achieved. Also, the proposed adaptive Volterra filter structure exceeded the performance compared to the memoryless approaches.

Nonlinear distortion is now an important factor hindering excessive high-speed information transmission over telephone channels. Thus equalization is the mostly apt idea to be used. Therefore, nonlinear distortion changes from one connection to another and also changes with time for each significant connection. In [33, 34], nonlinear equalizers have been studied, which are utilized by the receivers for passband QAM. It is the most favoured modulation scheme for achieving high data rates. Equalization was achieved in [33], for channel inversion by utilizing a nonlinear AF in series with the nonlinear data channel. A demonstration have been done on recorded data which shows that the nonlinear decision-feedback equalization can reduce the error rate significantly for a variety of channel characteristics. A nonlinear AF in parallel with the channel is employed to perform a cancellation of nonlinear inter symbol interference (ISI) in [34]. It provides an estimate of the interference. This cancellation process eliminates nonlinear interference terms without extreme noise enhancement and permits effective implementation with read only memory's (ROM's) to generate the required nonlinear signal variables. To increase the ability for the correction of channel nonlinearities, an orthogonalised version of the Volterra series was used.

An adaptive nonlinear finite impulse response (FIR) filter was studied for the noise cancellation in [35], where noise channels were presumed to consist of nonlinear memoryless part followed by a linear dispersive portion. The described model was used to design an adaptive nonlinear filter. The canceler removes the noise from the received signal. In an experimental simulation, when a comparison is drawn between the nonlinear canceler and linear canceler, it can be seen that the nonlinear canceler reached 22.3 dB of noise suppression, while the linear canceler was efficient of suppressing the interfering noise by only 6.4 dB.

T. Koh and E.J. Power [36] presented numerous new techniques analogous to the second-order Volterra filtering complications. The simulations and results propounded in the paper are intended basically to minimize the complexity involved with the problems of Volterra filtering. Recursive factorization and the nonlinear AA are employed to implement Volterra filters. A MMSE solution for the Volterra filter is obtained, based on the speculation that the filter has Gaussian input. Also, they proposed a recursive factorization method to design a subclass of the nonlinear adaptive Volterra filters, which can lessen the intricacy of the above filtering operations considerably. Eventually, a nonlinear AA for the Volterra filter is analysed along with its mean convergence and MSE having asymptotic excess.

In the field of nonlinear system identification, the measured noise is always a foreseeable issue, which is in general assumed to be a random process having finite-order statistics. In such condition, the MSE seems to be an appropriate metric for the error estimation. A. Rai and A.K. Kohli in [23] propounded a research work that focussed on the VSS nonlinear adaptive Volterra filtering because of its less computational complexity. They presented the polynomial AF using the GVSS-LMP algorithm for the nonlinear system identification of Volterra filter, under the environment of α-stable impulsive noise. In their research paper, they adopt the minimum fallacy/error dispersion norm instead of

MMSE as it has an apt metric for the error estimation. The Gaussian distribution is a distinctive case of the α-stable processes having α=2, which is categorized by the finite variance. By altering $p \geq 1$ in the manifestation of impulsive noise, which is characterised by $1 < \alpha < 2$, the adaptive weights are updated for the convergence of the proposed LMP algorithm. Also, they have proposed the updating criterion of GVSS in amalgamation with the LMP algorithm, to determine the slowly TVing kernels of Volterra, under the same impulsive noise scenario of the non-Gaussian α-stable.

Li Tan *et al.* [6] and Das *et al.* [9] presented a Volterra filtered-x LMS (VFXLMS) and filtered-s LMS (FSLMS) algorithms hinged on a multichannel structures, which are illustrated for feedforward ANC of a nonlinear processes, but for a fixed step size. Although, these methods execute well under certain conditions, noise can deteriorate their performance. The recent research has validated that linear ANC systems can be successfully applied to minimize the broadband noise and narrowband noise. Precisely, in reduction of low-frequency noise, such linear ANC systems are very conducive. However, in some practical cases, the noise that comes from a dynamic model may be a deterministic and nonlinear noise process despite of stochastic, tonal, or white noise process, and at the cancelling point, the primary noise may exhibit the nonlinear distortion. Also in the ANC system, the secondary path estimate may have nonminimum phase. This secondary path estimate refers the transfer function between the error microphone and secondary sources/ speaker. Thus, the causality constraint is meddled and hence the linear ANC system will endure performance degradation. In [6], an implementation of VFXLMS algorithm based on a multichannel structure is explained for feedforward ANC. Simulation results depicts that the developed algorithm attains performance improvement over the standard FXLMS algorithm for the two situations, i.e., firstly, the reference noise is a nonlinear noise, and the nonminimum phase is estimated by the secondary path. Secondly, the primary path shows the nonlinear behaviour. This developed VFXLMS algorithm in [6], can also be employed as an alternative, when the standard FXLMS algorithm does not perform well.

D.P. Das *et al.* in [9], proposed a novel nonlinear AA, i.e., FSLMS algorithm for multichannel ANC of nonlinear noise processes. A minimized complexity FSLMS algorithm employing a filter bank approach is also proposed. This method outperforms the conventional FSLMS and second-order VFXLMS algorithm for control of nonlinear noise processes and thus proposed method involved lesser number of computational complexity. It is also seen that beneath a high degree of third-order nonlinearity, second-order Volterra

FXLMS algorithm worsens its performance, but the FSLMS algorithm holds its improved performance.

In [37], J.B. MacNeil *et al.* have propounded a new technique for the system identification of time-varying (TVing) from ensemble average data using singular values decomposition (SVD). This system is capable of trailing rapid changes in dynamic systems with no *a priori* deduction referring to the system's dynamic structure. It gives a sequence of nonparametric impulse response function (IRF) representations. The system is considered to be a TVing system.

Nonlinear adaptive Volterra filtering system identification are broadly employed for identification of nonlinearities in numerous applications. In [5], the improved trailing capability of a NVFF-RLS algorithm is propounded for two types of order, i.e., first-order and second- order nonlinear TVing Volterra systems under a constraint of a nonstatic environment. The nonlinear model trailing difficulty is altered into a state estimation problem of the TV system. The TVing Volterra kernels are regulated by the first–order Gauss-Markov (GM) stochastic alteration equation, and basis upon on this, the state-space representation of the system is developed. The NVFF-RLS outplays the FFF-RLS and gives better channel estimation, and also affords superior channel trailing performance in terms of the MMSE. In account for the nonstatic signal, the NVFF-RLS algorithms is adjusted to the TVing signals by employing the refurbishing prediction error criterion.

As discussed, the LMS algorithm has always been the emphasis in many research due to its inherent robustness and simplicity, which leads to the implementation in many practical applications. It is also well known fact the MSE is dependent on the step-size of the filter selected for its performance. As the step-size diminishes, the speed of convergence time increases. Therefore, this restraint needs to compromise between the convergence rate and MSE in most of the AFs. As a consequences, many researchers have shifted their focus from a fixed step-size parameters to the variable step-size to revamp its performance. There are different algorithms which support step-size to be variable according to the need. All these VSS algorithms are given in [17-22].

S. Karni in [17] put an opinion that the convergence factor in the AF can be made TVing w.r.t the slope/ gradient of the performance surface and thus the step size of the LMS algorithm is altered from μ to $\mu(n)$, which depends on the input as well as on the output error. It also showed that the parameter α (damping parameter) is also an important parameter in the convergence, i.e. by selecting an appropriate value of α, the new algorithm retains an exorbitant convergence rate and hence have small misadjustment. This algorithm worked well under nonstationary environment.

In a nonstationary atmosphere, the ideal value of the step-size parameter of the LMS algorithm accords a balance between the quantity of gradient noise and lag noise. However, in practical scenario, the value of the step-size is difficult to determine, and it's not possible to determine it *a priori* because of its channel, which is not known. Also, fixed step-size may not be able to respond to TV channel parameters, and thus results in degraded performance. Thus from last two decades, VSS-LMS algorithms are developed to sort out from this problem.

A new VSS-LMS algorithm is proposed in [18] for AF by R.H Kwong. As the MSE proliferates or decreases, the step-size increases/decreases accordingly allowing the AF to trail changes in the system and also produces small steady state error. The main aim of the research in [18] was to provide fast trailing when a large predicted error will cause the μ to increase/ decrease. The proposed algorithm reduces the accord between trailing ability and misadjustment of the fixed step-size LMS algorithm. Thus, VSS-LMS algorithm reduces the compassion of the misadjustment to the degree of nonstationarity. A noteworthy feature of this algorithm is that the approximate formulas can be obtained to envisage the misadjustment in both the environmental conditions i.e., stationary and nonstationary.

Stochastic gradient AFs are extensively famous due to its inherent simplicity. But, they usually agonise from relatively deliberate and data reliant convergence behaviour. The performance of stochastic gradient systems is unfavourably affected by the spreading of high eigenvalue of the signal input vector (autocorrelation matrix). V.J. Mathews and Xie in [19] presented a VSS gradient AF that overcome the above problem. The step-size μ altered w.r.t a gradient descent algorithm, which is designed to minimise the MSE during each and every iteration. In the subsequent chapter, we will discuss about the gradient descent algorithms in detail. They also propounded an estimated analysis of the performance of AF, where according to a random walk model, the input signals are zero mean, Gaussian and white, and a set of prime coefficients, which are considered to be a TVing. This algorithm has very fine convergence speed and of course low steady-state misadjustment. The choice of the system parameters of the AF is subjected to change in the trailing performance of the algorithm. Their main objectives were:

- To publicize the AFA to the signal processing class
- To present an investigation of the AF that counterparts the observed characteristic of the algorithm.

A numerous number of TVing step-size algorithms have been anticipated to revamp the rendition of the conventional LMS algorithm, and all are extremely sensitive to the noise disruption. Aboulnasr *et al.* [20] have suggested a VSS-LMS AA, in which the step size of this algorithm is acclimatized according to the square of the estimated time averaged of the auto-correlation of present and past estimation error, i.e., $\overline{e}(n)$ and $\overline{e}(n-1)$ respectively. This algorithm (named as MDVSS, i.e. modified variable step-size) supports robustness behaviour, which ensures a fast convergence at initial stages of iteration and also provides small error misadjustment. The present uncorrelated noise disturbances doesn’t affect the rendition of the proposed algorithm. An estimated investigation of steady-state and convergence rendition for stationary Gaussian input and nonstationary ideal filter tap weight vector having zero-mean is provided.

In [21], D.I. Pazaitis *et al.* have anticipated a new variable step size LMS algorithm which differs from previous techniques in terms of time varying step size series, which employs the kurtosis of the estimation error signal to regulate its convergence factor. In this the properties of the convergence are investigated, and a varied estimator (kurtosis) that consider noise statistics also presented. But, compared to the fixed step-size, this algorithm shows a small proliferation in complexity. The MSE characteristic and stability criterion were provided, and results have ensured the superior trailing ability and inferior misadjustment of the algorithm proposed in [21] compared with the algorithm which has fixed step and also with other previously VSS discussed above.

WeePeng and Farhang-Boroujeny have proposed a new class of VSS-LMS algorithms which is a simplified version of the algorithms discussed above and thus, reduces complexity without observable loss in performance [22]. The algorithm proposed in [22] can be assumed as a simplified form of their equivalents in the class of algorithms anticipated by Benveniste *et al.* [65]. In fact [19] was presented to be a distinct case of the proposed algorithm. The use of the VSS parameters and multiple update equations/ formulas are also shown in the paper. It can been seen in the simulation result that the proposed algorithm works similar to [65] but outperforms the [19] algorithm. All the aforestated VSS-LMS algorithms are achieved using the linear filtering perspective.

CHAPTER 3

ADAPTIVE FILTER

Abstract: *This chapter provides some principles of adaptive linear and nonlinear filters, and a detailed explanation of an adaptive filter, the need for it, adaptive algorithms used and their applications. The adaptation laws are first delineated with the emphasis on the LMS algorithm. Also a brief concise introduction is discussed for FIR and IIR filters. Finally at the end of this chapter, an introduction of adaptive nonlinear filters and their applications are propounded.*

3.1. Introduction

From the last three decades, engrossment in adaptive system has proliferated. They have a vast and broad applications in the fields such as signal processing, in communication process, SONAR, and also utilized in biomedical engineering. An adaptive systems habituate according to the change in environment and determine for the ideal/optimal parameters of the system based on a reference signal. Optimal parameters of the systems are like tap weights in case of a filter. The performance and stability of an AA is totally dependent on the additive noise statistics and the reference input. There are certain premises in the context of Weiner filter theory, which includes the term of linearity, TIV, and Gaussian statistics such that the MSE criteria will be the perfect cost function. For the ease and simplicity of mathematical analysis these assumptions are taken into consideration. But, they don’t consider the broader problems having signal of non-Gaussian statistics.

Important factors in the digital communication system:

- Performance
- Reliability
- Maximum bandwidth utilization
- Simple

All the factors are equivalently precious for maximising profits. For simplicity, LMS algorithms are employed preferably. However, the rendition of the LMS algorithm is usually sub-optimal and the speed of convergence is small. Thus, it furnishes the motivation to investigate and study VSS-LMS algorithms for numerous applications.

3.2. The Weiner Filter

The Weiner filter comes under a class of linear prime time filter, and are prominent case of transversal FIR filters that improved upon the MSE cost function. This further minimizes the MSE signal by achieving a best and optimal filter tap weight vector. Theoretical concept for a Weiner filter is mathematically formulated for the basic case of impulse response filter having complex valued time series, as under numerous real-world situations, baseband signal comes in multifaceted form.

3.2.1. Mean Square Error Criterion

Fig. 3.1 describes a linear filter with the purpose of determining the anticipated signal, $d(n)$ from an input signal $x(n)$. Let us contemplate that, $d(n)$ and $x(n)$ are specimen of infinite length, random processes. Signal and noise are contemplated as stochastic processes in optimum filter design. The intention of the filter is to abate the MSE i.e., mean square value of the incongruity between the actual output, $\tilde{y}(n)$ of the filter and desired signal.

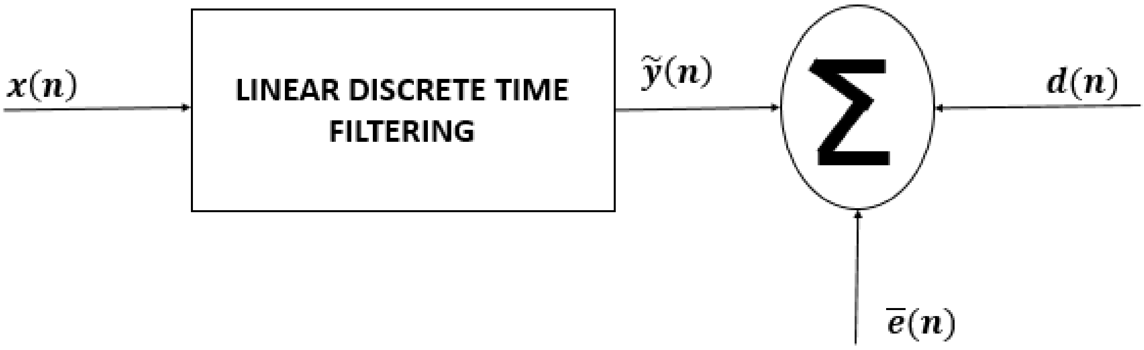

Fig. 3.1. Prototype Weiner Filtering Scheme [28]

The objective is to make the estimation error as minute as possible in some statistical sense, and this can be done by controlling the impulse response coefficients

$$w_0, w_1, \dots\dots, w_{N-1}.$$

Two basic diminutions are:

1. The filter is premeditated according to the linearity in behaviour, which makes mathematical and computational analysis easy to handle.
2. The filter is deliberate to be an FIR filter, which is symmetric and odd ordered.

The performance of the filter is calculated by the size/dimension of the estimation error, which can be easily said that if the estimation error is less, rendition will be better and vice-versa. As the $\tilde{e}(n)$ inclines to zero, $\tilde{y}(n)$ approaches the $d(n)$. In a modest way, we can say that in a filter design parameters, an apt parameter of the $\tilde{e}(n)$ as cost function is selected, which actually optimizes the performance. The cost function in Weiner filter is designated to be

$$\xi = E[\overline{e}(n)]^{2} \tag{3.1}$$

where, $E[.]$ indicates the EASE.

3.3. Weiner Filter : Transversal, Real Valued Case

Consider a transversal AF as demonstrated in Fig. 3.2. Let the inputs $x(n)$, $d(n)$, and $w_0, w_1, \dots \dots, w_{M-1}$ are real valued stationary processes, where M equals the tap weights and the number of delay units.

Both the input filter $x(n)$ and the tap weight vectors w can be given as

$$x(n) = [x(n)\; x(n-1) \dots \dots \dots x(n-M+1)]' \tag{3.2}$$

$$w = [w_0, w_1, \dots \dots \dots, w_{M-1}] \tag{3.3}$$

The output of the filter is defined as

$$\tilde{y}(n) = \sum_{i=0}^{M-1} w_i\, x(n-i) = w'x(n) = x'(n)w \tag{3.4}$$

Thus, the error signal can described as

$$e(n) = d(n) - y(n) = d(n) - w'x(n) = d(n) - x'(n)w \tag{3.5}$$

Substituting eq. (3.5) into (3.1), the EASE is obtained as,

$$\xi = E[\overline{e}(n)]^{2} = E\big[\big(d(n) - w'x(n)\big)\big(d(n) - x'(n)w\big)\big] \tag{3.6}$$

Expanding the above equation, we obtain

$$\begin{aligned}\xi = E[d(n)^{2}] - E[d(n)x'(n)w] - E[d(n)w'x(n)] \\ + E[w'x(n)x'(n)w]\end{aligned} \tag{3.7}$$

As w is not a random variable,

$$\begin{aligned}\xi = E[d(n)^{2}] - E[d(n)x'(n)]w - w'E[d(n)x(n)] \\ + w'E[x(n)x'(n)]\, w\end{aligned} \tag{3.8}$$

Now, in the next step we can write $E[d(n)x(n)]$ as a $M \times 1$ cross correlation vector

$$p = E[d(n)x(n)] = [p_0, p_1, \dots \dots \dots, p_{M-1}] \tag{3.9}$$

and $E[x(n)x'(n)]$ can be expressed as a $M \times M$ autocorrelation matrix R

$R = E[x(n)x'(n)]$, i.e.,

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0,M-1} \\ r_{10} & r_{11} & r_{12} & \cdots & r_{1,M-1} \\ r_{20} & r_{21} & r_{22} & \cdots & r_{2,M-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ r_{M-1,0} & r_{M-1,1} & r_{M-1,2} & \cdots & r_{M-1,M-1} \end{bmatrix} \tag{3.10}$$

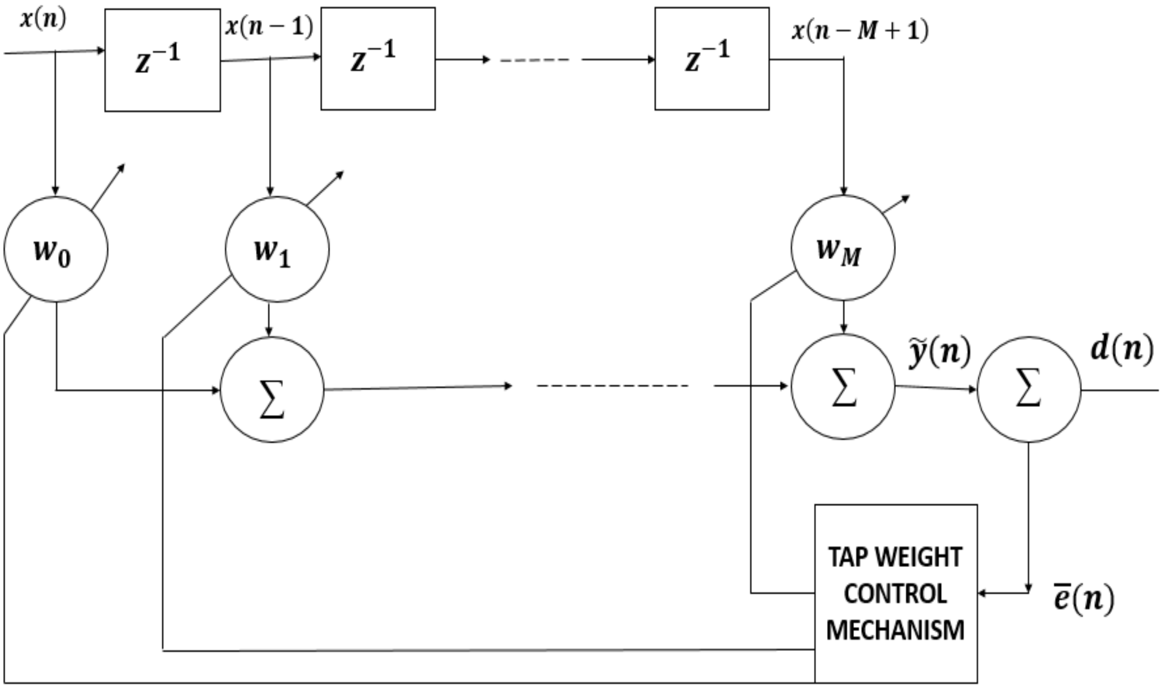

Fig. 3.2 Block diagram of an adaptive transversal filter [28]

From (3.9),

$p' = E[d(n)x'(n)]$ and thus $p'w = w'p$

Therefore,

$$E[d(n)x'(n)] = E[d(n)x\ (n)]w'$$

Subsequently, from equation (3.8) we obtain

$$\xi = E[d(n)\ ^2] - 2p'w + w'Rw \tag{3.11}$$

To minimize the cost function of the obtained set of filter tap weights w, solve the system of the equations that get results from doing the partial derivatives of ξ w.r.t. every w i.e., the gradient vector to be the zero, i.e.

$$\frac{\partial \xi}{\partial w_i} = 0 \tag{3.12}$$

where, $i = 0, 1, \dots\dots\dots, M-1$, and M characterizes amount of weights of the tap.

From the equation (3.11), the gradient vector can also be given as

$$\nabla\xi = 0 \tag{3.13}$$

where, the symbol ∇ represents gradient operator, which is defined as column vector

$$\nabla = \left[\frac{\partial}{\partial w_0} \quad \frac{\partial}{\partial w_1} \quad \frac{\partial}{\partial w_{M-1}}\right] \tag{3.14}$$

and zero on the R.H.S of (3.13) indicates the column vector comprising of M zero. It has been further manifested that the partial derivative of ξ w.r.t w can be solved such that

$$\nabla\xi = 2Rw - 2p \tag{3.15}$$

Let $\nabla\xi = 0$, thus from the above equation we obtain

$$Rw = p \tag{3.16}$$

Thus from this, we can obtain an optimum set of filter tap weights of the Weiner filter

$$w = R^{-1}p = w_0 \tag{3.17}$$

where w_0 denotes the optimum filter tap weight vector and the equation (3.17) is known as the **Weiner Hopf** equation and thus, can be solved to get the filter tap weight vector, which is analogous to the minimum point of the cost function.

3.4. Iterative Search Algorithm

From the previous section it has been shown that the Weiner Hopf equation can be expounded to obtain the optimum tap weights of the filter by minimising a cost function, if the prerequisite statistics of the underlying signals p and R are given. In some cases, it deals with the computationally complexity, especially in the case of large number of tap weights and secondly, when the data rate is high. So, an iterative search algorithm [50] is an alternative way, which minimizes the cost function at the cost of less computational complexity. This algorithm starts at some random initial point in 'w' vector space and moves continuously towards the optimum filter tap weight vector in steps. The principle of finding the w by constantly minimising the cost function ξ by intend of an iterative search algorithm is the central focus to the development of AAs (e.g. LMS). In simplified words, adaptive algorithms are referred as iterative search algorithms obtained for minimising the cost function by substituting the true statistics of tap weights with estimates that are obtained.

3.5. Method of Steepest Descent

Assume that the cost function, ξ that is to be minimised is convex. It has a unique single minimum point when ξ commensurate to a convex quadratic surface. In simpler word it can be said that the iterative search algorithm is totally guaranteed to converge to the optimal point, when the cost function is convex.

We may initiate with an arbitrary and random point on the performance surface and we consider a small step in the course in which the cost function ξ decreases faster, and this indicates to a step along the slope of steepest descent curve of the performance at that particular point. When this is repeated successively, convergence towards the minimum point of the convex surface is guaranteed, under a given set of parameters.

The method and procedure of steepest descent [50] is an another way to determine w_0, in contrast to solve the Weiner Hopf equation straight. The technique of steepest descent algorithm belongs to a class of iterative search methods of optimization. It is a basic scheme that executes an iterative search for a merest point of any convex functions, given under a set of parameters. This technique is executed in transversal filter with the function of convex relating to the cost function and filter tap weights parameters.

It follows following steps to search the merest point of the cost function w.r.t. a set of filter tap weights.

1. Start with a preliminary guess of the tap weights of the filter whose optimum values are to be determined for minimizing/reducing the cost function. If no prior knowledge is available, set all the filter tap weights to zeros and start the search process.
2. By the preliminary guess, compute the gradient vector of the cost function ξ w.r.t. to the tap weights at that particular present point.
3. Update the filter tap weights by considering step in the counter direction i.e., the sign is changed of the gradient vector that is obtained in step 2. This refers to step in the direction of the steepest descent curve in the ξ at the present input. In fact the size of the step is selected proportional to the size of the gradient vector.
4. Repeat step 2, and iterate the whole course until no further compelling changes is observed in the filter tap weights i.e. at that point we can say that the iterative search has converged to a best optimal point.

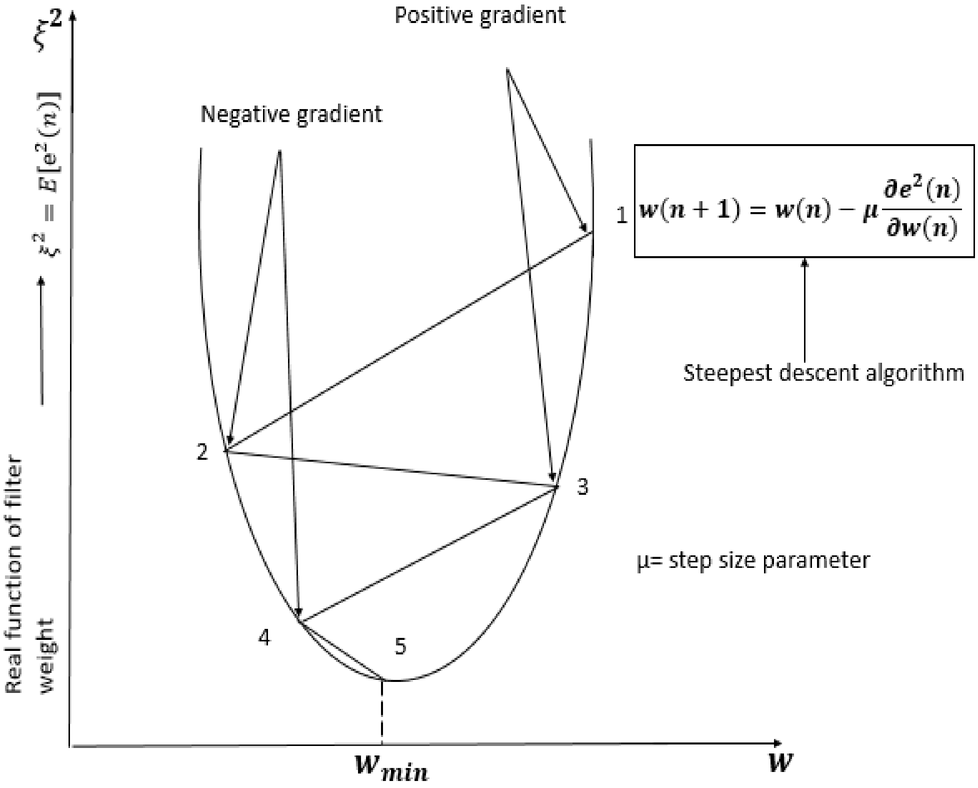

Fig. 3.3 Steepest descent curve showing positive and negative gradient

Consider the Fig. 3.3. Now, if we are at position $w(1)$ and calculate its gradient and suppose it comes out to be positive gradient i.e., no point is going further to right, as if it goes to the right, we will go further away from the minima point. Therefore we will go to the left of $w(1)$ for the next iterative i.e., we subtract some number and reach $w(2)$ as shown in the Fig. 3.3 and again we will calculate its gradient and this time suppose its gradient comes out to be negative, then for next iterative we will move to right and thus the process continue till we reach w_{min}.

NOTE:- If the gradient is steep, the quantity which we will subtract from the i^{th} will be higher in magnitude.

Steepest descent algorithm is given by

$$w(n+1) = w(n) - \mu \frac{\partial e^2(n)}{\partial w(n)} \tag{3.18}$$

where, $w(n)$ is the tap weight vector at the n-th iteration, μ represents positive scalar known as step-size, and $\frac{\partial e^2(n)}{\partial w(n)}$ denotes the gradient expression at the point $w = w(n)$.

3.6. Errror Performance Surface

The estimated erroneous signal $e(n)$ can be given as

$$e(n) = d(n) - \sum_{i=0}^{M-1} w_i\, x(n-i) \tag{3.19}$$

The cost function ξ can be expressed as

$$\xi = E[d(n)\,^2] - \sum_{i=0}^{M-1} w_i^*\; E[x(n-i)d^*(n)] - \sum_{i=0}^{M-1} w_i\, E[x^*(n-i)d(n)] \\ + \sum_{i=0}^{M-1}\sum_{k=0}^{M-1} w_i^*\, w_k E[x^*(n-i)x^*(n-k)] \tag{3.20}$$

The MSE or cost function is specifically a second-order function of the filter tap weights. The dependence of the ξ depends on the tap weights $w_0, w_1, \dots \dots \dots, w_{M-1}$, which may be envisaged as a bowl shaped having a $(M+1)$ dimensional surface with M degrees of freedom presented by w. The surface described above is known as the error rendition surface of the transversal filter. The surface is specified by an individual minimum, where the ξ achieves its minimum value, where at this minimum point, the gradient vector, $\nabla\xi = 0$. The height of the cost function ξ analogous to the physical description of $x(n-i)$, which relates to the filtering the signal with the fixed filter tap weight, from where a prediction signal error $e(n)$ with power of ξ is obtained. The theory discussed above is the base of general AAs of adaptive signal processing.

3.7. The Need for an Adaptive Filter

A conventional and regular fixed filter, which is employed to extricate information/data from an input time sequence, is linear and TIV. In contrast, an AF is a filter which naturally adjusts its parameter coefficients to optimize an objective function as shown in Fig. 1.1. The main aim of an AF is to minimize the MSE by adjusting and modifying itself and therefore, it is a time-varying system. An AF is proficient when an accurate filtering operation may be unknown and this operation may be moderately nonstationary.

AFs have found many applications in numerous fields, for example data communication, speech processing, SONAR processing, and image processing. Two applications of adaptive signal processing will be conferred about in this section to illustrate the need for an AF. First application is equalization of a data transmission in the channel

[51] and second is noise cancellation [44].

3.7.1. Equalization of a Data Transmission Channel

The continuously increasing need for computer communications has been encounter primarily by increasing data speed transmission over the general telephone network. In the process of transmission, the binary data are converted into voice signals, then it is transmitted, and again it is converted back. The time dispersion results, if the frequency response with passband 300 Hz to 3000 Hz of a telephone line deviates from the ideal/general of constant delay and amplitude. In the PAM, each individual signal is a pulse whose level of amplitude is obtained by a symbol. The effect of every symbol transmitted to a time-dispersive channel widens the time interval employed to represent that particular symbol.

The symbol of the sampled data at the receiver can be expressed as a convolution of the impulse response of the channel h_i with the data symbols $u(k)$, which are transmitted, under a given linear channel,

$$y(k) = \sum_{i=0}^{\infty} h_i u(k-i) + n_0(k) \tag{3.21}$$

where, n_0 is a noise signal. The above equation as a sampled data symbol ca also be given as

$$y(k) = h_\delta u(k-\delta) + \sum_{i \neq 0} h_i u(k-i) + n_0(k) \tag{3.22}$$

where the effective delay is denoted by $\boldsymbol{\delta}$ of the channel. In (3.22), the first term represents delayed and attenuated data symbol and the second term is ISI among symbols because of the dispersion of the channel. An AF can be used to eliminate the ISI by inverting the channel. The utter need for an AF comes from a paucity of prior knowledge of h and from the TV nature of the channel.

Fig. 3.4 depicts a block diagram of a receiver utilizing adaptive equalization. In initial stage a pre-filter is used, which suppresses the out-of-band noise. After that, a time recovery device detects the symbol rate of data so that the sampler of the system can work at this pace. An adaptive equalizer after sampling is used, which inverts the channel of the system and removes all the interference present. An adaptive equalizer for example can be an adaptive transversal filter if the data transmission used is PAM. A training sequence is originated and is employed to train the AF. A slicer is used at the output of the filter, which is employed to detect the transmitted symbols. So, after the generated training sequence, the symbols which are detected are used to adapt the filter.

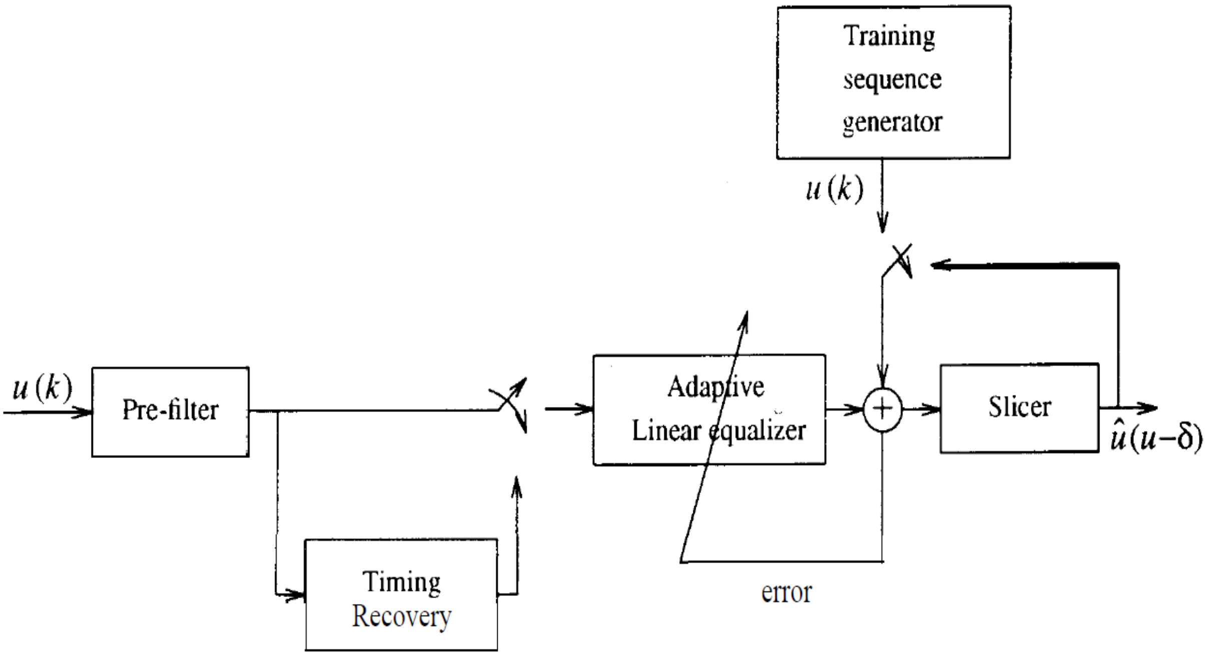

Fig. 3.4. Block diagram of a receiver utilizing adaptive equalization [51]

3.7.2. Noise Cancellation

As mentioned above, a pre-filter can be used to estimate a signal, which is corrupted by additive noise as the filter tends to suppress the maximum noise while leaving the input signal relatively unchanged. For the design of a fixed filter, it is required to know the prior knowledge of the behaviour of both the input signal and the noise. AFs are sometimes preferred as no prior knowledge of the signal and noise behaviour is needed for their design.

Adaptive noise cancellation is depicted in Fig. 3.5. The first sensor as shown receives two signal i.e., an input signal S and an uncorrelated noise signal n_1. Now, the second sensor receives the noise n_2 from the noise source, which is totally independent of S and somewhat related with the noise n_1. An AF gives an estimate of n_1 using the calculated original noise n_2. The difference is taken between the estimated n_1 and $S + n_1$, to cancel the n_1. The AF attains this by reducing the power of the system output z. The system output is the difference between the primary input signal and the output of the filter.

Both the applications above clearly signify the need for AFs. Even though a fixed filter can always be used instead of an AF in the data transmission receiver, and/or in noise cancellation, but it wouldn't be as productive as an AF since both the characteristic of the channel (data transmission channel and noise channel) are generally not known and it vary slowly with time. The characteristics of the noise that is to be cancelled are also not known to the designer mostly. All these reasons make an AF preferred and necessary.

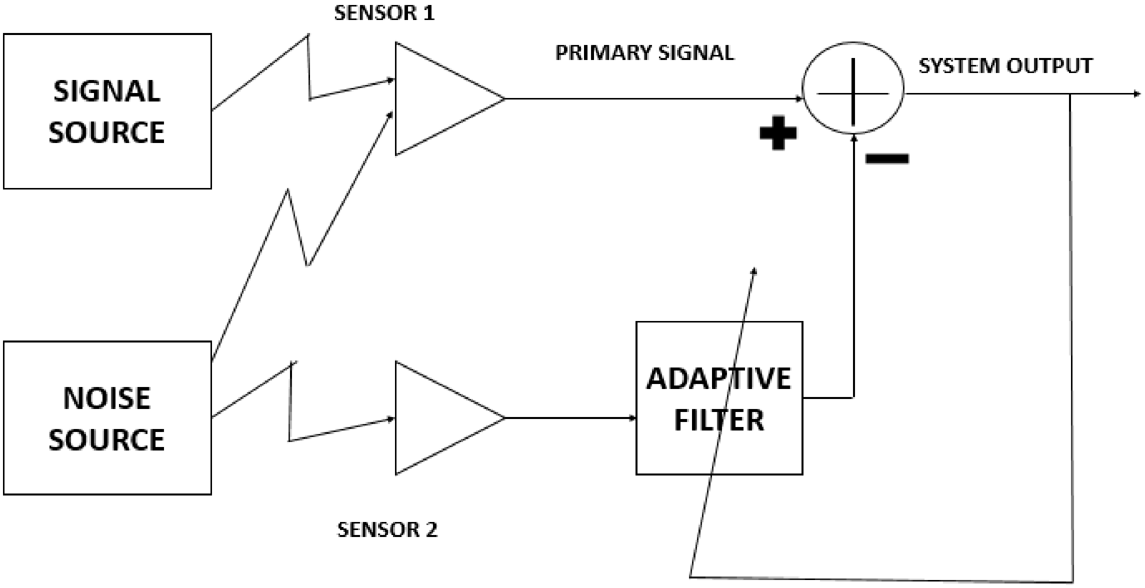

Fig. 3.5. Block diagram of an adaptive receiver cancellation [44]

3.8. Adaptation Laws

The main aim of an AF is to minimize the MSE of the output, which is given as the alteration between the desired signal and the output of the filter. This is known as the output-error formulation that is the underlying key of the majority of the algorithms presented in this thesis. The output-error formulation, which is one class of the adaptation laws is gradient based, and is given by the expression

$$p^{k+1} = p^{k} - \mu R^{-1}\left(\frac{\partial E(e^{2}(k))}{\partial p}\right) \tag{3.23}$$

where, p is a parameter vectors, k is the total number of iteration used or we can say the number of samples used, μ represents step-sizes in a diagonal matrix, R indicates a matrix, which is selected to improve the rate of convergence, $E(e^{2}(k))$ represents MSE, where the error signal is defined as

$$e(k) = d(k) - y(k) \tag{3.24}$$

In the above equation, $d(k)$ denotes the desired signal, and the $y(k)$ is the output of the filter. If the matrix is selected to be the correlation matrix of the gradient signals, the vulnerability of the filter convergence is substantially reduced on the eigenvalue spread of this gradient signal. In this AA, coefficients of the filters are updated in the different (opposite) direction of the gradient vector. Therefore, the process of adaptation moves downhill on the mean square error surface.

The computational load for real-time signal processing should be minimized to lowest value. If the MSE is approximated by the instantaneous square error $e^2(k)$, the gradient vector can be substituted by its corresponding estimate that is noisy, but unbiased in the above adaptation law. Further, if the matrix R is substituted by the unit matrix, the adaptation law turns to be the famous and well known algorithm that used in a real-time adaptation law- the LMS algorithm [44, 52]

$$p^{k+1} = p^k - 2\mu\, e(k) \left(\frac{\partial y(k)}{\partial p}\right) \tag{3.25}$$

The step-size is the key parameter of an AF, which control the convergence speed. Smaller step-size results in a slow convergence and gives low MSE, while larger step-sizes results in a fast convergence rate and high MSE. But, if the step-sizes are very much large, they make the system unstable. The choice of step-size depends on the following properties:

- Structure of the filter
- The adaptation algorithm
- Properties of the input signal

Selection of the step-size is well understood for an adaptive FIR filters, but it is uncertain for an adaptive IIR filters.

3.9. Adaptive Linear FIR Filters

Two famous kinds of adaptive linear FIR filters are:

- Transversal filters
- FIR lattice filters

We explain only the first one i.e., the linear adaptive transversal filters because of its popularity. This filter gives a guaranteed stability and a good global convergence. A linear transversal AF has the following form [52]

$$y(k) = \sum_{i=0}^{n} h_i\, u(k-i) \tag{3.26}$$

where, n represents order of the filter, u denotes the input signal, and the impulse response of the filter is given by h.

The coefficients of an AF vary slowly as shown in [53], and thus we have

$$\frac{\partial y(k)}{\partial p} = Z^{-1}\left(\frac{\partial Y(z)}{\partial p}\right) \tag{3.27}$$

where, p represents a filter coefficient that is to be adapted and Z^{-1} denotes the inverse z-transform. $Y(z)$ represents the z-transform of the time domain value $y(k)$. By using the relationship in (3.27), carry out the derivation of the gradient evaluation expression in both the z-domain and time. For an adaptive linear filter, derivation of gradient formulas in the z-domain is simpler. The gradient vector of this filter coefficients is given as

$$\frac{\partial Y(z)}{\partial h} = (z^{-1}z^{-2} \ldots z^{-n})^{\,T} U(z) \tag{3.28}$$

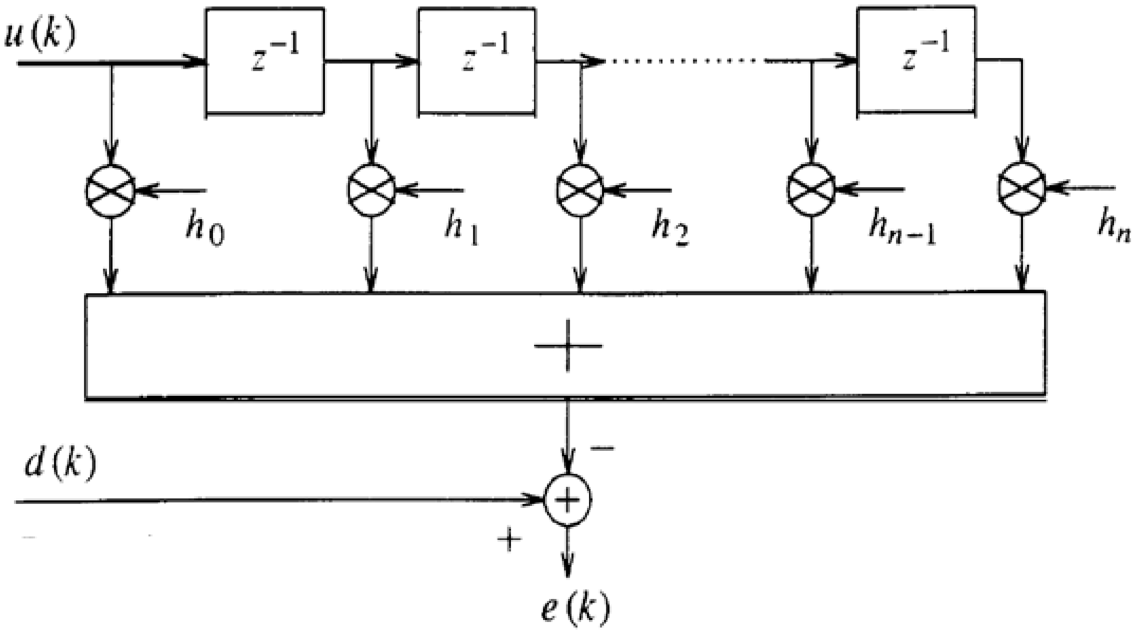

Fig. 3.6. Block diagram of an adaptive linear transversal filter [1]

where, $h = (h_0, h_1, \ldots, h_n)^{\,T}$. By using the LMS algorithm as expressed in (3.25), we can update the filter coefficients according to

$$h^{k+1} = h^k + 2\mu e(k)u(k) \tag{3.29}$$

In the above equation, the input signal is given by

$$u(k) = \big(u(k)\;u(k-1)\ldots u(k-n)\big)^{\,T} \tag{3.30}$$

Now, considering the equation (3.29) and the assumption made above, we can rewrite as

$$E\big(e_h(k+1)\big) = (I - 2\mu R)E\big(e_h(k)\big) \tag{3.31}$$

where, E denotes the expectation operator, and the $e_h(k+1)$ vector is the subtraction of the Weiner solution h_0 from the coefficient vector h^{k+1}.

Also, R represents the correlation matrix of $u(k)$, which is the input signal, and I denotes the identity matrix. It can be seen that if the step-size follows the following condition, the mean of the error of the coefficient e_h goes asymptotically to zero [28]

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{3.32}$$

where, λ_{max} is equivalent to maximum eigenvalue of R. The convergence time-constant is given as

$$\tau = \frac{1}{\lambda_{min}\mu}$$

$$= \frac{\lambda_{max}}{\lambda_{\min CX}} \tag{3.33}$$

where, CX lies between 0 and 1, and is a constant. λ_{min} is the minimum eigenvalue of the matrix R.

Step-size should be selected smaller in a practical scenario than the theoretical bounds obtained above due to the noise in gradient estimates. A minute upper bound for the step-size was acquired for an adaptive LMS transversal FIR filter [54, 55], when the two conditions, i.e. necessary and sufficient conditions are contemplated.

The convergence is confirmed if it satisfy equation (3.32) and

$$\sum_{i=0}^{n} \frac{\mu\lambda_i}{1-2\mu\lambda_i} < 1 \tag{3.35}$$

A condition, which is simpler and easier to use, is

$$0 < \mu \le \frac{1}{3tr(R)} \tag{3.36}$$

The speed of the convergence rely on the eigenvalue spread of the matrix R as given in (3.23).

3.10. Adaptive Linear IIR Filters

In spite of the fact that an adaptive FIR linear filters have fine properties and qualities, they are considered to be expensive for some practical applications, like echo cancellation in acoustic systems. Adaptive IIR filters are consider to be more computationally efficient for these type of applications. An active research is going on in the field of adaptive IIR filters. Different structures have been explored, which include

- Direct form
- Lattice form
- Recursive state-space form
- Parallel form
- Cascade form

As this thesis is mainly focussed on an adaptive FIR filter and Volterra filter, therefore we will not discuss the topic of an adaptive linear IIR filter in a descriptive way.

3.11. Adaptive Nonlinear Filters

Linearity is the property, which is usually assumed in numerous adaptive signal processing applications. But, nonlinearities in real employment limit the performances of the system. Thus, new and modern signal processing technique, theory and practice are indulging to be more and more concerned with the design of nonlinear filters to be efficient. One approach is to opt the basis of truncated Volterra series for designing an adaptive nonlinear filters. The Volterra filter is appealing from a theoretical view point, as it can assign with a basic class of nonlinear systems while the output is still linear w.r.t. numerous higher power coefficients of the system.

For a nonlinear system, the output of the system can be expanded into a Volterra series under some satisfying conditions can be given as

$$y(k) = \sum_{i_{1=0}}^{\infty} h_1(i_1)u(k-i_1) + \sum_{i_1=0}^{\infty}\sum_{i_2=0}^{\infty} h_2(i_1,i_2)u(k-i_1)u(k-i_2) + \cdots$$

$$\dots + \sum_{i_1=0}^{\infty} \dots \sum_{i_m=0}^{\infty} h_m(i_1,i_2,\dots,i_m)\, u(k-i_1)\, u(k-i_2) \dots u(k-i_m) + \cdots \tag{3.37}$$

where, $h_1, h_2, \dots, h_m$, ..., represents system coefficients. The sum with system coefficient h_1 can be contemplated as a convolution function of the signal input, $u(k)$ with $h_1(k)$ of a linear system.

Normally, the series has large in fact infinity number of terms and each individual term has an infinite sum. The memory and computation requirements make it next to impossible to base AFs on this series if no elucidation and simplification is made. In practice, if the series contains the large major terms and single term is a finite sum, the system can be made reasonably well. Therefore, truncated series are employed to develop AFs. Further, we consider that the indexes of $h_m(i_1, i_2, \dots, i_m)$ are symmetric and exchangeable. Then, an AF can be written as on the following truncated series

$$y(k) = \sum_{i_{1=0}}^{n_1} h_1(i_1)u(k-i_1) + \sum_{i_1=0}^{n_2}\sum_{i_2=0}^{n_3} h_2(i_1,i_2)u(k-i_1)u(k-i_2) + \cdots$$

$$\dots + \sum_{i_1=0}^{n_m} \dots \sum_{i_m=0}^{n_m} h_m(i_1, i_2, \dots, i_m)\, u(k-i_1)\, u(k-i_2) \dots u(k-i_m) \qquad (3.38)$$

where, m denotes the total number of terms used in the filter, n_1 represents the order of the linear term, and $n_2, n_3, \dots, n_m$ represents the order of the nonlinear system. Now, after truncating, it is obvious that the filter has a finite impulse response, therefore the name adaptive nonlinear FIR filter.

Updating the coefficients of the Volterra filter is usually based on the concept of LMS and is followed as [35, 36]

$$h_j^{k+1}(i_1, i_2, \dots, i_j) = h_j^k(i_1, i_2, \dots, i_j) + 2\mu_j e(k)u(k-i_1)u(k-i_2)\dots u(k-i_j) \qquad (3.39)$$

where, $j = 1,2, \dots \dots, m$ and μ_j represents the step-size for the j-th power term.

Volterra filter converges to the optimal and best solution asymptotically in the mean and the speed of the convergence depends on the squared ratio of maximum to minimum eigenvalues of the input matrix of correlation. A form of the LMS algorithm, known as sign algorithm [56], updates the coefficients of filter employing only the sign of the gradient. It takes only the way of gradient and mislay the proportionality factor of the correction terms to the error. Thus, the sign algorithm gives a less accurate and slower convergence in compare to the stochastic algorithm, while taking account of reduced-complexity implementation.

Also, the gradient base AAs use a step-size to sway the convergence. Usually, in most of the case the step-size is fixed in existing nonlinear AAs. A method for controlling the step-size was proposed in [57] for the sign algorithm in [56] in order to control and obtain faster convergence rate with reduced final MSE without affecting the implementational complexity.

Most of the applications related to Volterra filters work in the time domain. In [28], a frequency domain representation of a Volterra AF is given. For a case of finite memory of length *L*, the AF converges to the similar time domain nonlinear AF demonstrated previously. A fast Kalman filter was discussed in [58], which was designed for a quadratic Volterra filter. The convergence rate is faster than an LMS-based AA.

3.12. Applications of Adaptive Nonlinear Filters

Adaptive Volterra filters have numerous applications in multitudinous fields. Typical applications, which are used for communication channels are:

- Echo cancellation
- Equalization
- Noise reduction

In modern years adaptive echo cancellation [30-32] has been substantially used in connection with some advance services introduced in the telephone network due to its allowance for full-duplex transmission on 2-wire circuits with bandwidth in both directions. This concept can be employed in two fields:

- Digital subscriber loop modems
- Voiceband data modems

A theoretical survey of echo cancellation was done in [31], where the parameter coefficients of the continuous time VS were indicated by generalised Fourier series. In the absence of noise, it was proved that the echo canceler converges and reaches to zero. A nonlinear echo canceler was propounded in [32].

Nonlinear distortion is now a key factor, which hinders the speed of data transmission over telephone channels. Therefore, equalization must be designed adaptive as nonlinear distortion changes from one connection to another and also changes with time for every particular connection. In [33, 34], we have studied the receiver, which utilized the nonlinear equalization for QAM passband. This modulation scheme is the most preferred scheme for attaining high data rates. Equalization was achieved in [33] by employing a nonlinear AF in series with the nonlinear channel-data to invert that channel. Nonlinear decision-feedback is used, which reduce the maximum error in every iteration. Cancellation of nonlinear ISI in [34] is developed by using a nonlinear AF in parallel with the channel to give an estimate of the interference. The approach of cancellation removes nonlinear interference components without increasing noise and provides effective implementation with ROM’s to obtain the required nonlinear signal variables. A VS having orthogonalized version was used, resulting in a strengthen ability to modify the channel nonlinearities.

A nonlinear adaptive FIR filter was studied in [35] for noise cancellation, where the channels of the noise are assumed to comprise of a nonlinear memoryless part followed by a linear dispersive section. A nonlinear AF was designed employing this model. Comprising nonlinearity in the canceler of noise was shown to revamp the ability of the canceler to exclude noise from the signal received at the receiver. If we compare both the canceler, the nonlinear canceler will achieve 22.3 dB of noise suppression, while the linear canceler achieves 6.4 dB noise suppression.

Recent digital radio systems uses high BW efficient QAM modulation scheme to make employ of the crowded microwave radio spectrum. As the number of constellation points increases, the system seems to be more sensitive to all nonlinear and linear types of distortion. The transmitter of the high power amplifier when operates in saturation, it achieves maximum efficiency, and thus resulting in a distortion (nonlinear) in the transmitted signal. The nonlinear distortion turn out to be a critical issue. In [59], an adaptive linearization method was propounded, which compromises for amplifier nonlinearity. Before going to an amplifier, the signal is pre-distorted. The power amplifier is assumed to be memoryless and needed that the signal not to be filtered out before amplification, while developing the algorithms. Even though large number of applications of AF are used in communication systems, some efforts has to be done to employ these filters in other fields and research areas.

CHAPTER 4

ADAPTIVE FILTERING ALGORITHMS

Abstract: *This chapter presents some basics of adaptive filtering techniques, and deals with the development and derivation of the algorithms used within this thesis. It also confers the relationship of some parameters, like step-size, power spectral density and input signal power on the adaptive algorithms. The factors that calculate the performance of an adaptive algorithms are clearly discussed at the end of this chapter.*

4.1. Introduction

Since the last three decades, engrossment in adaptive systems has proliferated, leading to extensive employ of adaptive techniques in the fields like signal processing, SONAR, communication system, and biomedical engineering. The main characteristic of adaptive system is that it adapts to the environmental changes and look for the best and optimal system parameters relied on a reference signal. If we consider a case of filter, the system parameters referred to be the filter tap weights, and its performance is totally dependent on the additive noise statistics and the reference input. In the theoretical case of Weiner filter, there are contemplations of linearity, for time-invariance and statistics of Gaussian such that MSE condition will be the best and optimum cost function. These presumptions are used often for the simplicity of mathematical analysis, but never take into case of the broader problems of signals having statistic characteristics of non-Gaussian. Instead, non MSE condition can be a superior solution in the case of non-Gaussian. An efficient BW utilization, maintaining performance and system reliability are crucial in terms of maximising profits and thus supports economically. Primarily, the AF solution has to be easier and simpler, which further leads to the employ of the LMS algorithm. But, if the case of non-Gaussianity is contemplated, the performance value of the LMS algorithm is sub-optimal often. Thus, these gives the stimulus to explore and survey non-MSE AAs for the case of non-Gaussian atmosphere. E. Walch *et al*. in [60] studied the cost functions of MSE, mean fourth error (MFE) to mean sixth error, which depicted the probability of selecting the higher-order algorithm in the instance of non-Gaussian nature. The amalgamation of MSE and MFE cost function showed the capability of adaptation in the perspective of a combination of various statistics.

The above explained works have concisely encapsulated the ongoing research of multifarious orders of cost function. Fig. 4.1 depicts a chart, which categorized the MSA and the non-MSAs. These algorithms are categorised on the basis of the gradient vector adaptation. A non-MSA is referred when the gradient vector altered by help of a function or amalgamated of other vector (gradient vector).

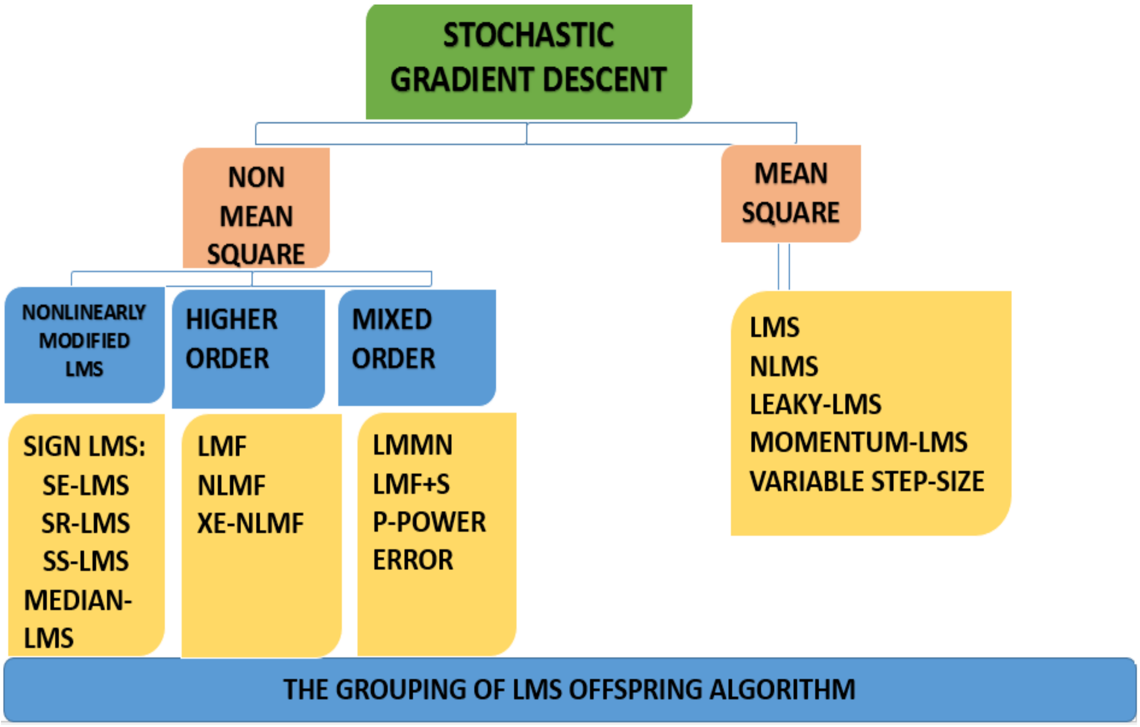

Fig. 4.1. Flow chart of the amalgamation of LMS offspring algorithm

The rendition of an AF is perilously dependent not only on its inherent structure, but also on the AA used to iteratively/recursively apprise the filter tap weights that indicate the structure. From many decades, a large number of stochastic based non-MSA have been obtained for several purposes, relishing the field of AAs. Out of all the algorithms, LMS is the most used algorithm in real-world practice. Also, the Sign LMS, LMF and LMMN are equivalently important. The LMF algorithm when executed can minimises the cost function of the mean fourth and considered as a higher order algorithm that can outplay the LMS in extensive MSE, under a condition of a sub-Gaussian noise.

As explained in section 3.5 about the method of steepest descent that initiating from some arbitrary value for the filter tap weight vector, it revamps with the increasing number of iterations. The ultimate value so obtained for filter tap weight vector, converges equivalent to Weiner solution.

Another category of algorithms is the stochastic gradient algorithm family. This term discern it with the method of steepest descent that employs deterministic gradient in an iterative and recursive computation for stochastic inputs of the Weiner filter. There is always a trade-off between two parameters:

- Filter convergence rate
- Steady state error.

If the value of a step-size is large, sooner will be the convergence and vice-versa, but we get good and more accurate value of estimation when the step-size is small. If we want to use both the step-size together, for example in initial stage large step size and then at the convergence point small step-size, we use VSS-LMS algorithm. Also, the choice and selection of step-size according to the convergence reflects an accord between speed and misadjustment of adaptation.

Thus, we conclude that:

Smaller step size gives

- Smaller misadjustment
- Longer convergence time

Larger step size provides

- Larger misadjustment
- Shorter convergence time

A VSS-LMS algorithm provides

- Faster convergence at initial stage
- Smaller convergence at the final stage
- An adjustment of step-size, which is controlled by MSE

4.2. Adaptive Filtering Algorithms

4.2.1. LMS Algorithm

The least mean square (LMS) [61] belongs to the family of a gradient descent based algorithms. It is the most commonly employed algorithm in numerous applications, principally used in channel equalization and elimination of echo. The most definite features of LMS algorithm is its simplicity and robustness, which is supportive while updating the algorithm and thus leads to huge successful applications and developments of other gradient based descent algorithms.

The LMS is instigated as a way to iteratively adjust the parameters value of filter tap weight $w(n)$ of a linear filter, but with an aim of minimising the fallacy between a $d(n)$ and $y(n)$ as shown in Fig. 4.2.

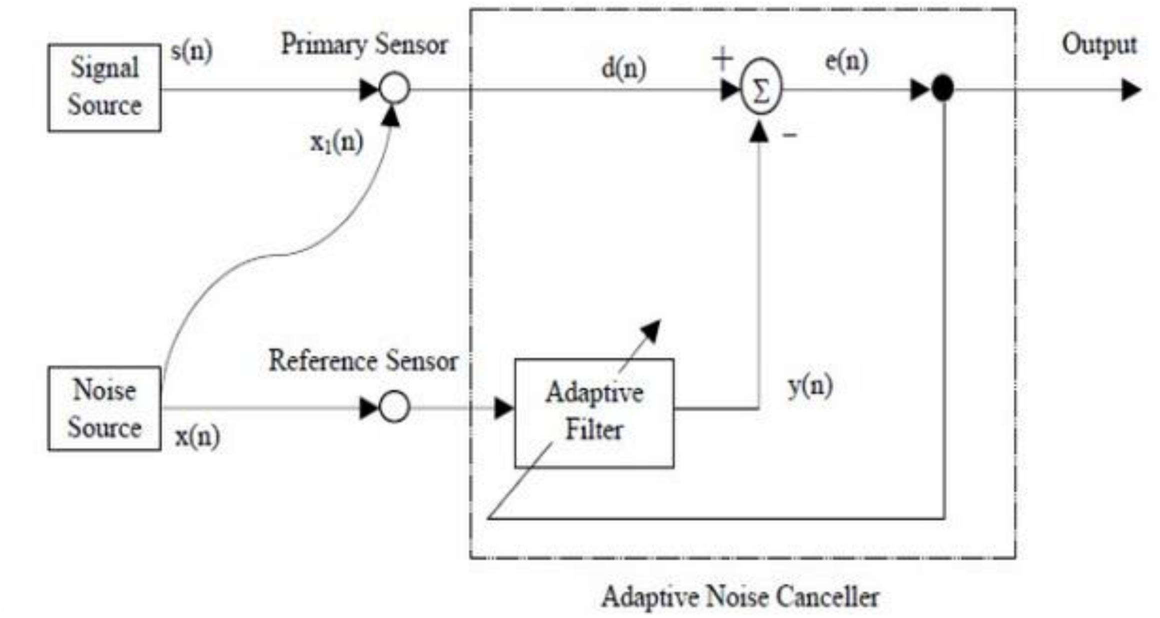

Fig. 4.2. Block diagram of LMS adaptive filter [61]

LMS is categorised as one of the many algorithms, which are apt for the work and task and whole class of algorithms have been obtained which can briefly address a multifarious problems, computational complexity, and of course minimization criteria.

4.2.2. Derivation of LMS algorithm

In this section we will derive the LMS algorithm as an expeditious approximation to the minimization (steepest descent) of a cost function. From equation (3.18), we get

$$w(n+1) = w(n) - \mu \nabla_n e^2(n) \tag{4.1}$$

where, ∇ represents gradient operator,

$w = [w_0, w_1, \dots \dots, w_{N-1}]^T$ denotes filter tap weight, and μ refers to be the step-size.

Using (3.14) and the last component in (4.1), we conclude that the i^{th} element of $\nabla_n e^2(n)$ can be expressed as

$$\frac{\partial e^2(n)}{\partial w_i} = 2e(n)\frac{\partial e(n)}{\partial w_i} \tag{4.2}$$

Take the first derivative of the equation (3.5), we get

$$\frac{\partial e\,(n)}{\partial w_i} = 0 - \frac{\partial y(n)}{\partial w_i} \tag{4.3}$$

Since the desired signal, $d(n)$ is independent of filter weight, therefore the derivative of $d(n)$ becomes zero.

Now, in the next step put (4.3) in (4.2)

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n)\frac{\partial y(n)}{\partial w_i} \tag{4.4}$$

After that take the derivative of $y(n)$ in (3.4) and put the result in (4.4), we get

$$\frac{\partial e^2(n)}{\partial w_i} = -2e(n)\,x(n-i) \tag{4.5}$$

From (3.14) and (4.5)

$$\nabla e^2(n) = -2e(n)x(n) \tag{4.6}$$

Now finally substitute (4.6) into (4.1), which gives

$$w(n+1) = w(n) + 2\mu e(n)x(n) \tag{4.7}$$

The equation (4.7) referred as the LMS algorithm for the tap weight adaptation process. In an individually and every iteration, the weight updates its value according to the change in environment in the motive of reducing the cost function to determine w_0.

It should be noted that the factor 2 in the RHS of the equation (4.7) is sometimes left out by some authors. Now, multiply the whole equation (3.15) by ½ so that the factor 2 is cancelled out, and thus we get

$$\frac{\nabla\xi}{2} = Rw - p$$

Therefore,

$$w(n+1) = w(n) - \frac{1}{2}\mu\nabla_n\xi \tag{4.8}$$

Putting the value of cost function in equation (3.8), we get

$$w(n+1) = w(n) - \frac{1}{2}\mu\nabla\, e^2(n) \tag{4.9}$$

In a brief, LMS algorithm comprises the following features:

- Adjusts the filter weight, $w(n)$ according to the change in environment.
- Adaptively controls the weight by the following equation:

    - $w(n+1) = w(n) + 2\mu e(n)x(n)$
- It gives a negative feedback to reduce the error signal (MSE)
- Simpler in implementation
- Robust and stable performance while performing against multifarious signal condition.
- Slower convergence rate and step-size parameter is the key parameter for controlling the speed of convergence.
- It needs number of iterations equivalent to the dimension of the input signal.
- **Steps for LMS algorithm**
    - Filter output :

$$y(n) = \sum_{n=0}^{M-1} x(n)w(n) \tag{4.10}$$

    - Error Estimation

$$e(n) = d(n) - y(n) \tag{4.11}$$

    - Filter tap weight adaptation

$$w(n+1) = w(n) + 2\mu x(n)e(n)$$

- Stability condition is given by

$$0 < \mu < \frac{2}{input\ signal\ power} \tag{4.12}$$

- Step-size consequences:
    - Larger step-size
        - Adaptation rate increases
        - Residual MSE increases
    - Smaller step-size
        - Adaptation rate decreases
        - Residual MSE decreases

The learning curve for LMS algorithm is depicted in Fig. 4.3. As shown, the LMS curve is not as smooth as the learning curve of steepest descent method algorithm due to its presence of gradient noise.

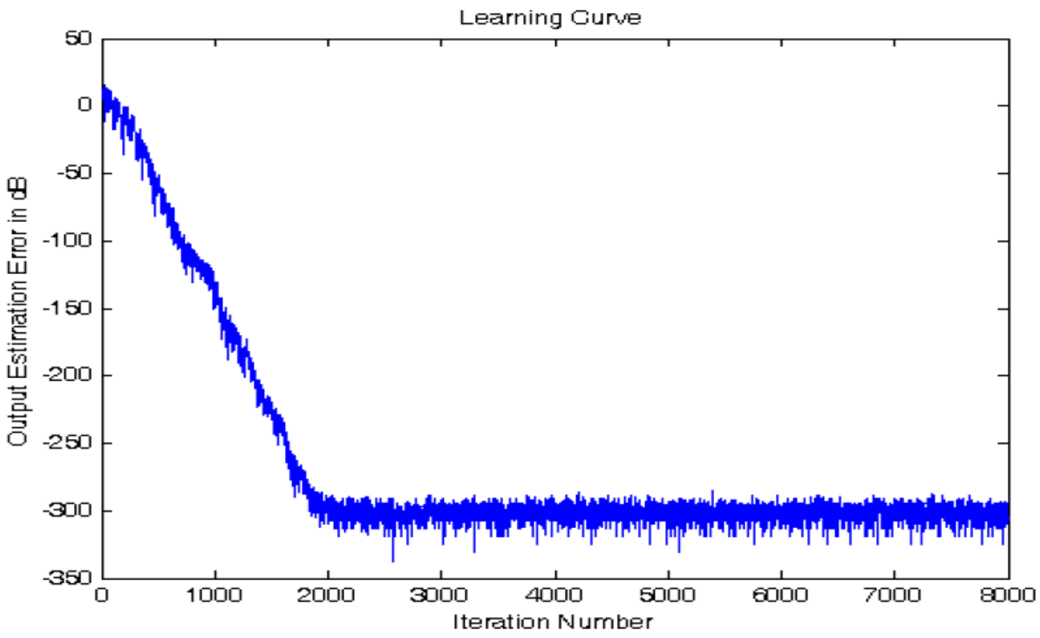

Fig. 4.3. Learning curve of LMS algorithm

Some noteworthy aspects of the LMS algorithm related to its performance are:

- LMS shows a tendency to eliminate the noisy data because of its smoothing action w.r.t small step-size parameters.
- LMS can trail slowly TVing system, and is commonly useful in non-stationary environments.
- The LMS algorithm doesn't have a tendency to stop working at undesirable local minima, thus it has a unique global minima.
- Computationally simple (m multiplication in count and m additions per iteration).

4.2.3. Step-size Optimization

As discussed earlier that the step-size plays a crucial role in adjusting and controlling the rendition of the LMS AF. The complication of the standard LMS algorithm is that the fixed step-size controls the accord between the steady state MSE and the convergence rate. If a large step is taken, it reduces the convergence/ transient time but will give result in an immense steady state MSE. Consequently, to avail a slower convergence rate, there should be selection of smaller step-size. Selection and preference of step-size vary according to the application specific with primacy such as robustness, adaptation accuracy, tracking, and fast convergence.

To attain high and qualitative performance, an optimum step-size is selected to follow the RLS performance (also known as TVing step-size algorithm). The condition for the stability of step-size is given in (4.12).

$$0 < \mu < \frac{2}{input\ signal\ power}$$

It is inherent from the above condition that the power of the input signal is inversely proportional to the step-size. Thus, smaller step-size is requisite to avoid instability or the amplification of a gradient noise.

4.2.4. Effect of Power Spectral Density of the Input Signal

There exhibit a significant interaction among the filter tap coefficients, which are adaptive in nature, and thus the rate of convergence of the LMS algorithm worsen with higher input correlation levels. To corroborate improved convergence rate, the LMS algorithm needs signal input to exhibit equal excitation over the entire range of frequency.

4.2.5. Effect of Filter Tap Length

The length of adaptive filter using LMS algorithm must be adequate to cover the impulse response of the channel, which is not known for ensuring good and fine asymptotic performance [61]. But, when the impulse response of the channel, which is not known is ‘long’, this may lead to proliferated computational complexity and hence, poor convergence rates, if the signals at the input are highly correlated [61].

4.2.6. Effect of Input Signal Power

The term in the equation (4.7), i.e., $\mu x(n)e(n)$ is directly linked to the filter input $x(n)$. This indicate that the rate of convergence and the stability criterion in the LMS algorithm is precisely dependent on the parameter value of $\mu\sigma_x^2$, where σ_x^2 represents the power variance of the input signal [61].

When the power of the signal input varies greatly, the LMS algorithm shows an unstable characteristic (gradient noise amplification), and thus leads to the best and optimal solution. To deal with such type of problem, NLMS algorithm is used.

4.2.7. Normalized Least Mean Square (NLMS) Algorithms

If the input signal to the AF is non-stationary, calculating the upper bound limit of the step-size is an issue for the VSS algorithm. If the following condition is followed, the fastest convergence can be achieved with the slection of appropriate step-size [18]

$$\mu_{max} = \frac{1}{\lambda_{max} + \lambda_{min}} \tag{4.13}$$

Howsoever, some experimental results also demonstrated that it is not required that the step-size in (4.13) doesn’t always gives the fast and stable convergence as discussed in [18]. In fact it’s given that for the LMS algorithm, $\frac{2}{3}\mu_{max}$ is a rule of thumb. NLMS algorithm is a natural option for nonstationary input and increases the convergence speed, as the step-size is generally normalised by the signal input power. When the input signal power is varied, the NLMS algorithm normalized by updating the step-size with an estimate of the variance of the input signal.

The filter tap weight adaptation is given by [63]

$$w(n+1) = w(n) + \frac{\alpha}{||x(n)||^{2}} e(n)x(n) \tag{4.14}$$

where, $||.||^2$ represents Euclidean Norm.

From equation (4.14) we can see that the NLMS algorithm is dependent on α, but independent of the power of the input signal. By selecting α in such a way so as to optimize the rate of convergence of the algorithms. Generally α varies from 0 to 2.

$$0 < \alpha < 2$$

Despite so many advantages, the NLMS algorithm also have a slight complication of its own. When the $x(n)$ is minute, there may be an occurrence of instability since we are performing numerical division by using small value of the Euclidean Norm.

However, this can be improved by adding a positive constant to the Euclidean norm in (4.14) such that

$$w(n+1) = w(n) + \frac{\alpha}{c + ||x(n)||^{2}} e(n)x(n) \tag{4.15}$$

where, $c + ||x(n)||^{2}$ represents the normalization factor, and c should always be less than 1.

Advantages of NLMS:

- Due to the normalization of the step-size, this algorithm converges faster than the LMS algorithm as compared in Fig. 4.4 and Fig. 4.5.
- In this algorithm, the estimated error value is less than the LMS algorithm.

Disadvantage of NLMS:

- NLMS is more complex and less stable than the LMS.

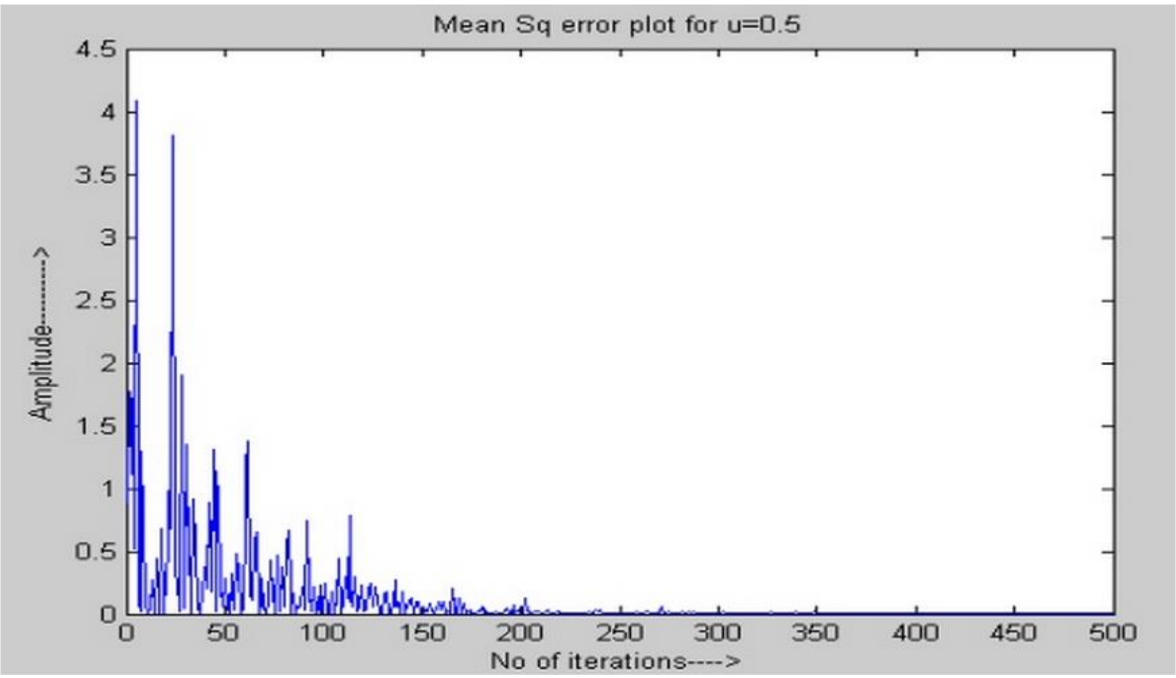

Fig. 4.4. Mean square error plot for the LMS algorithm [61].

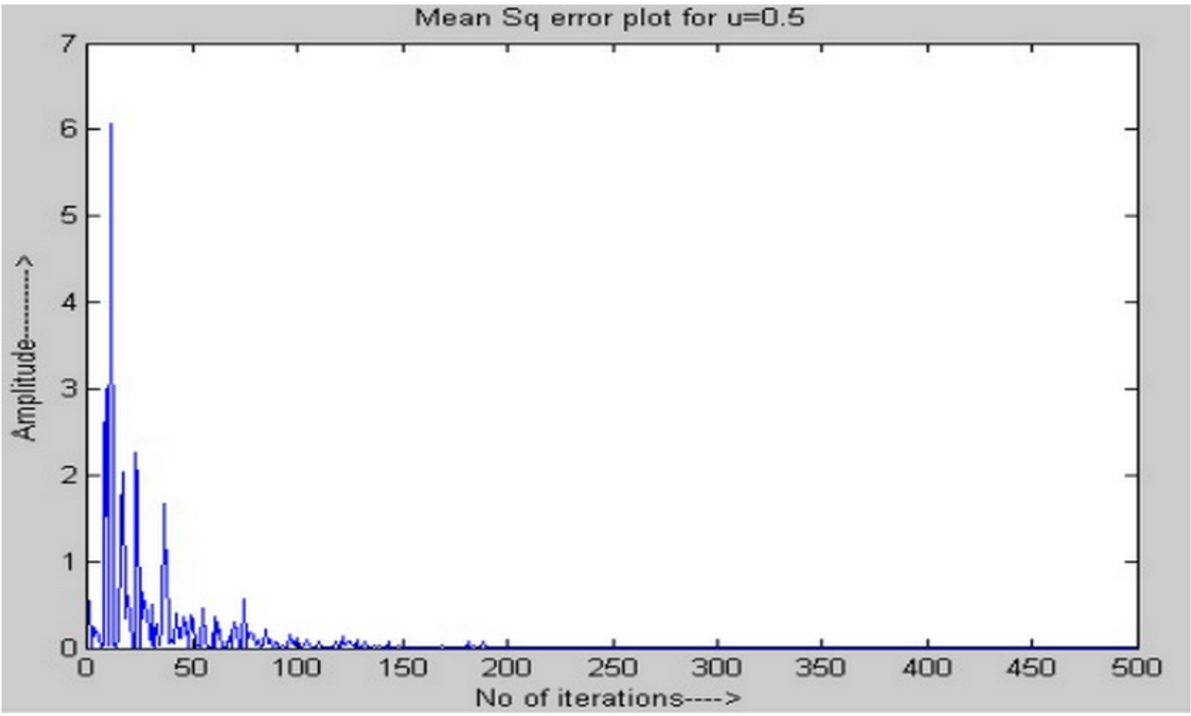

Fig. 4.5. Mean square error plot for NLMS algorithm [63]

4.2.8. RLS Algorithm

The major profit we get in the LMS algorithm is their lower computational complexity. But, at the same time, we have to compromise it with a slow convergence, especially in the case when the eigenvalue spread is large for the auto-correlation matrix, i.e. when

$$\frac{\lambda_{max}}{\lambda_{min}} \gg 1$$

Also, in the LMS algorithm, there is only one single parameters (step-size), which is used to control the convergence rate and thus, the modes referring to the eigenvalues converge slowly. Therefore, to get a faster convergence, it is important to develop more complex algorithms comprising additional parameters.

The RLS algorithm [64] recursively determine the filter coefficients that reduce a cost function of the weighted linear least square function related to the input signal as shown in Fig. 4.6.

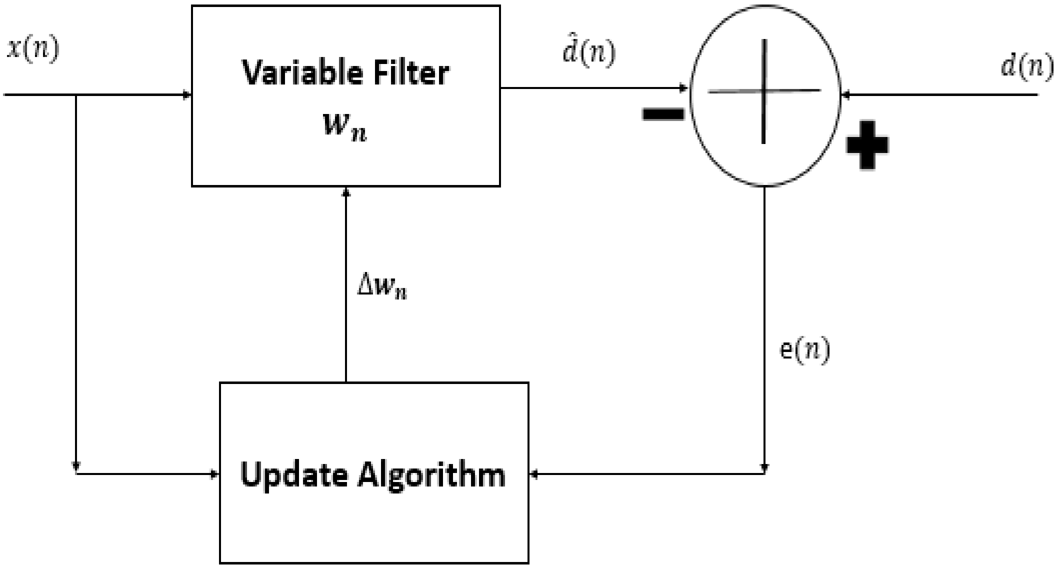

Fig. 4.6. Block diagram for the RLS algorithm [61]

The cost function for RLS algorithm is expressed as

$$\emptyset(n) = \sum_{k=1}^{n} \lambda^{n-k}\, e_n^2(k) \tag{4.16}$$

The equation (4.16) depicts that at a time instant n, all past values of the estimated error since the time the RLS algorithm commenced are required. As the time increases, the aggregate of data needed to process this algorithm also increases. Due to its limited memory and the complex computation capability makes the RLS algorithm [64] difficult and practical impossible in its ideal form. In the equation (4.16), the RLS algorithm triggers at $k = 1$ and λ represents a small positive constant, which is close to but smaller than one.

Advantages of RLS algorithm

- It converges faster than all other algorithms (LMS, NLMS, and APA) as shown in Fig. 4.7.
- It has large capacity of the noise cancellation.

Disadvantage of RLS algorithm

This is the most complex algorithm and has complex computational ability. RLS algorithms are also popular because of its capability and excellent performance in TVing environments.

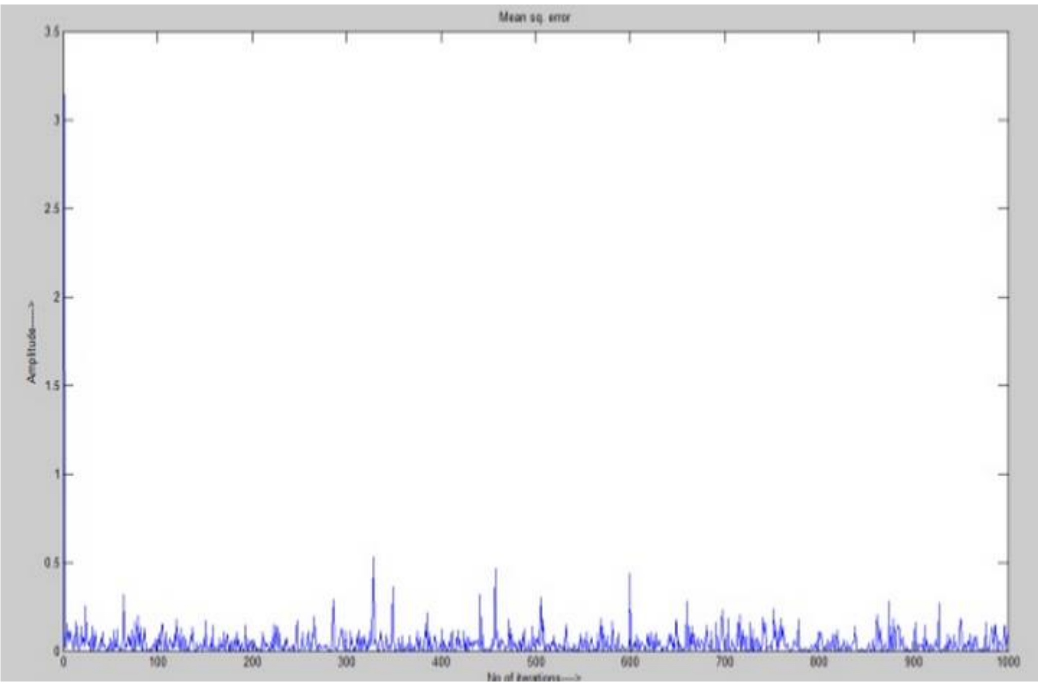

Fig. 4.7. Mean square error plot for RLS algorithm

4.3.Performance of an Adaptive Algorithms

The most crucial parameters that calculate the performance of an AAs are mentioned below [28]

- **Rate of Convergence**: The term convergence rate is explained as the aggregate number of iterations requisite for the algorithm to converge to its steady state MSE (also known as MASE, Mean asymptotic square error).
- **Misadjustment:** This parameter explains steady-state characteristic of the algorithm. This is a significant measure of the amount by which final value of EASE exceeds the MMSE obtained by the optimal Weiner filter. The asymptotic rendition of the algorithm will be better, if there is smaller misadjustment.

- **Numerical Robustness:** The employment of AFAs on a digital computer, which surely operates employing finite word-lengths, results in errors of quantization. AFA is known to be mathematically robust when its implementation employing finite-word-length is stable.
- **Computational Requirements:** This is a crucial parameter practically. It includes the total number of operations needed for full one complete iteration of the algorithm used. It also includes the amount of memory required to store the data and program.
- **Stability:** If the MSE converges to final/finite value, an algorithm is said to be stable.

In the applications of DSP for example, adaptive echo cancellation, the factors explained above play an important role. Ideally one prefers to have a robust AF having simple computational, small misadjustment and high rate of convergence that can be easily implemented on a computer.

CHAPTER 5

ANALYSIS AND METHODOLOGY

Abstract: *This chapter provides a mathematical formulation for adaptive second-order Volterra filter for single and multichannel system. After that it gives formulation for different variable step-size algorithms for multichannel system identification employing Volterra filtering.*

5.1. Introduction

The Volterra system is the most preferred paradigm among polynomial system models because of its nonlinear relationship between input-output signals. However, its output is linear in the context of kernels. There are numerous TVing nonlinear wireless communication channels, which need to be estimated by the nonlinear polynomial adaptive filtering [22].

An input-output relationship of a causal Volterra filter [1, 6] is given by the following equation

$$y(n) = \sum_{\mathrm{u}=1}^{\mathrm{U}} \tilde{y}_{\mathrm{u}}(n) \tag{5.1}$$

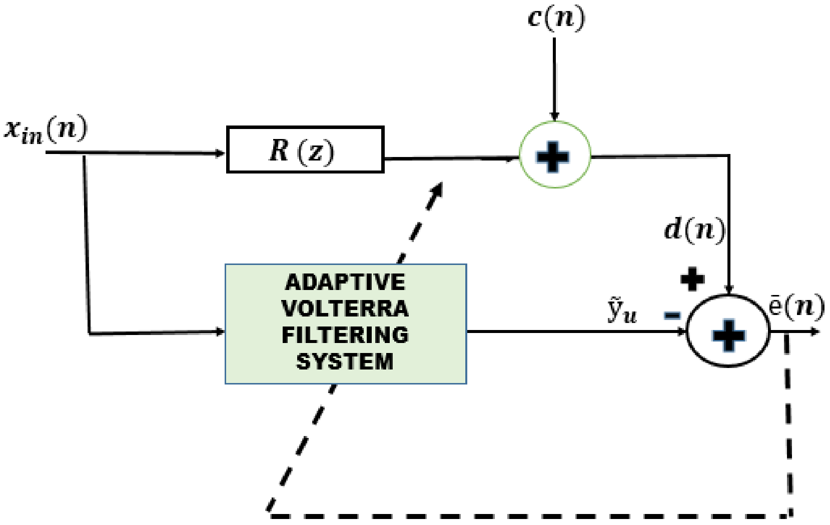

Fig. 5.1. Block diagram of an adaptive Volterra filter.

where $\tilde{y}_{\mathrm{u}}(n)$ is further expressed for an U-th order discrete filter and memory length M as

$$\tilde{\mathrm{y}}_{\mathrm{u}}(n) = h_0 + \sum_{m_1=0}^{M-1} \sum_{m_2=m_1}^{M-1} \dots\dots.. \sum_{m_k=m_u-1}^{M-1} h_u(n; m_1, \dots\dots, m_u) \,.\prod_{t=1}^{u} x_{in}(n-m_t) \,. \tag{5.2}$$

Here, $x_{in}(n)$ and $y(n)$ represent the input and output of Volterra filter, whereas h_u are the u^{th} order Volterra kernels, and n indicates the time index.

Fig.5.1 shows the schematic diagram of an adaptive Volterra filter. For simplicity, let us contemplate a second order (U=2) Volterra series expansion and thus input-output relationship is given as

$$\begin{aligned} \tilde{\mathrm{y}}_{\mathrm{u}}(n) = h_0 + & \sum_{m_1=0}^{M-1} h_1(n; m_1) \; x_{in}(n-m_1) \\ & + \sum_{m_1=0}^{M-1} \sum_{m_2=m_1}^{M-1} h_2(n; m_1, m_2) \, x_{in}(n-m_1) x_{in}(n-m_2) \end{aligned} \tag{5.3}$$

In this case, the adaptive filter (AF) would try to estimate the desired signal $d(n)$ using above second-order truncated Volterra series expansion. Here, $h_1(n; m_1)$ and $h_2(n; m_1, m_2)$ are the first-order and second-order adaptive Volterra kernels, respectively that are iteratively updated. These are updated at each time so as to decrease some convex function of the error signal designated as

$$\overline{e}(n) = d(n) - \tilde{\mathrm{y}}_u(n). \tag{5.4}$$

Further, desired signal $d(n)$ is denoted as

$$d(n) = X_{in}^{T}(n) \, r(n) + c(n) \tag{5.5}$$

where X_{in} is the input signal vector, $r(n)$ signifies the impulse response of primary path $R(z)$ as shown in Fig. 5.1 and $c(n)$ indicates additive white Gaussian noise (AWGN) with $N(0, V_{min})$in which V_{min} denotes variance.

Therefore, the estimated output received signal is indicated by

$$y'(n) = H^{T}(n) \, X_{in}(n) \tag{5.6}$$

where the adaptively approximated Volterra filter coefficient vector, H, may be represented by

$$H(n) = \begin{bmatrix} h_1(n; 0), h_1(n; 1), \dots\dots, h_1(M-1; n), h_2(n; 0{,}0), h_2(n; 0{,}1), \dots \\ \dots, h_2(n; 0, M-1), \; h_2(n; 1{,}1), \dots\dots, h_2(n; M-1, M-1) \end{bmatrix}^{T}. \tag{5.7}$$

Here, $(.)^{T}$ denotes matrix transpose operator.

For second-order Volterra filter, the input signal, X_{in} is given by

$$X_{in}(n) = [x_{in}(n), x_{in}(n-1), \dots, x_{in}(n-M+1), x_{in}^2(n), x_{in}(n)x_{in}(n-1), \dots, x_{in}(n) \\ x_{in}(n-M+1), \dots, x_{in}^2(n-M+1)]^T \quad (5.8)$$

In the nonlinear system, the final purpose is to determine the TVing Volterra kernels $h_u(n; m_1, \dots\dots, m_u)$ in (2) and thus, filter coefficients are updated at each time using a steepest descent algorithm for minimizing $\overline{e}^2(n)$ at every point. These updated filter coefficients can be expressed as [1, 6, 23]

$$H(n+1) = H(n) + \mu' X_{in}(n)\overline{e}(n) \quad (5.9)$$

More specifically, equations for updated filter coefficients for second order Volterra filter can be written as

$$h_1(n+1; m_1) = h_1(n; m_1) - \frac{\mu_1'}{2}\frac{\partial\overline{e}^2(n)}{\partial h_1(n; m_1)} \quad (5.10)$$

$$= h_1(n; m_1) + \mu_1'\overline{e}(n)X_{in}(n-m_1) \quad (5.11)$$

and

$$h_2(n+1; m_1, m_2) = h_2(n; m_1, m_2) - \frac{\mu_2'}{2}\frac{\partial\overline{e}^2(n)}{\partial h_2(n; m_1, m_2)} \quad (5.12)$$

$$= h_2(n; m_1, m_2) + \mu_2'\overline{e}(n)X_{in}(n-m_1)X_{in}(n-m_2) \quad (5.13)$$

where μ_1' and μ_2' are the step-sizes that control the steady state and speed of convergence properties of the filters and these convergence constant are chosen such that $0 < \mu_1', \mu_2' < 2/\lambda_m$, where λ_m is the maximum eigen value of the matrix $\vec{R}_{x_{in}x_{in}} = E[\vec{x}_{in}(n)\vec{x}_{in}^T(n)]$

Hence, the estimation error at the receiver output from Eq. (5.4) and (5.6) is computed by

$$\bar{e}(n) = d(n) - H^T(n)X_{in}(n) \quad (5.14)$$

This error signal starts from an initial guess that depends on the prior information available to the system and then this is fed back to the self-designing filter and thus, it converges finally to the optimal solution.

As described in [6, 24], we yield the following expression from Eq. (5.2) by placing variables $m_t = m + s_{t-1}$, for $t = 1, \dots, u$ with $t_0 = 0$, that is

$$\tilde{\mathrm{y}}_{\mathrm{u}} = \sum_{m=0}^{M-1} \sum_{s_1=0}^{M-1-m} \sum_{s_2=t_1}^{M-1-m} \dots \sum_{s_{u-1}=s_{u-2}}^{M-1-m} h_u(m, m+s_1, \dots, m+s_{u-1}) \;.$$

$$x_{in}(n-m) \prod_{t=1}^{u-1} x_{in}(n-m-s_t). \qquad (5.15)$$

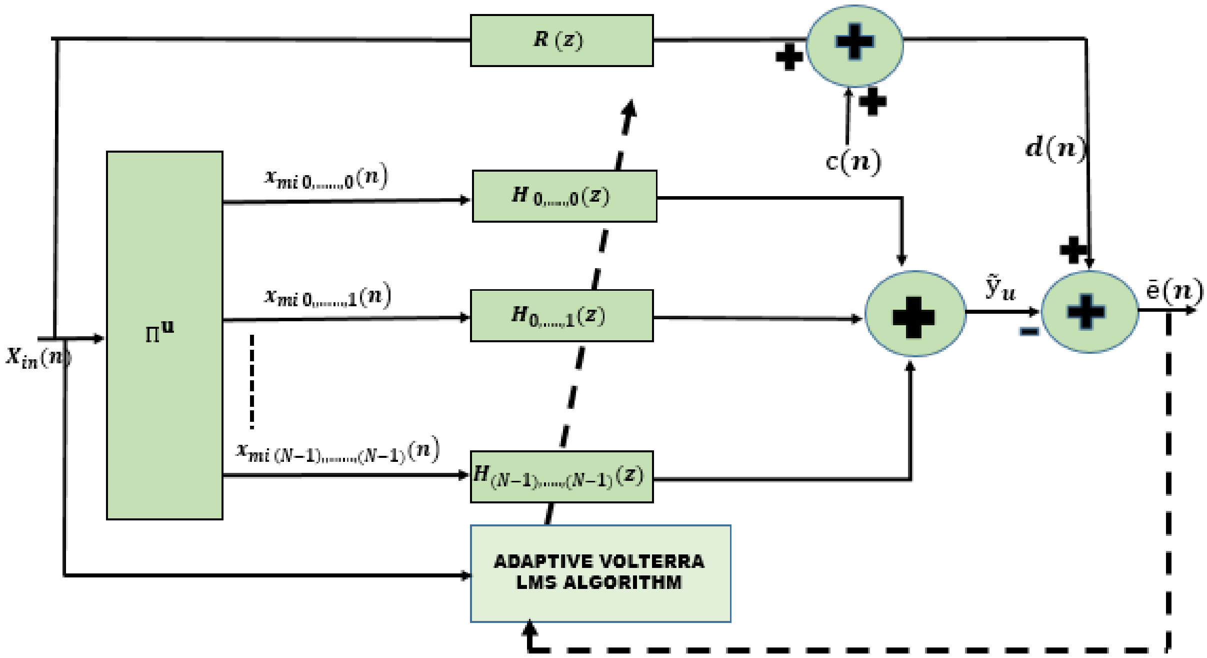

Fig. 5.2. Block diagram of a multichannel implementation of Volterra filter.

Interchanging the order of summation in (15), as shown in [6, 24], we get

$$\tilde{\mathrm{y}}_{\mathrm{u}} = \sum_{s_1=0}^{M-1} \sum_{s_2=s_1}^{M-1} \dots \sum_{s_{u-1}=s_{u-2}}^{M-1}$$

$$\left\{ \sum_{m=0}^{M-1-s_{u-1}} h_u(m, m+s_1, \dots, m+s_{u-1}) . \left(x_{in}(n-m) \prod_{t=1}^{u-1} x_{in}(n-m-s_t) \right) \right\} \qquad (5.16)$$

Simplifying Eq. (5.16),

$$\tilde{\mathrm{y}}_{\mathrm{u}} = \sum_{s_1=0}^{M-1} \sum_{s_2=s_1}^{M-1} \dots \sum_{s_{u-1}=s_{u-2}}^{M-1} \;.\; h_u(n, n+s_1, \dots, n+s_{u-1}) * x_{in_{s_1,\dots\dots,s_{u-1}}}(n)$$

$$= H_u^T(n) X_{in_u}(n) \qquad (5.17)$$

where, the * denotes linear convolution, and further $x_{in_{s_1,\dots\dots,s_{u-1}}}(n)$ is the distinct channel input with the u^{th}-order given by

$$x_{in_{s_1,\ldots\ldots,s_{u-1}}}(n) = x_{in}(n)\prod_{t=1}^{u-1} x_{in}(n-s_t) \tag{5.18}$$

where, $x_{in_{i_0}}(n) = x_{in}(n)$.

Taking Z transform on Eq. (5.17), we write

$$\overline{Y}_u(z) = \sum_{s_1=0}^{M-1}\sum_{s_2=s_1}^{M-1}\cdots\sum_{s_{u-1}=s_{u-2}}^{M-1} H_{s_{1,\ldots,}s_{u-1}}(z)X_{in_{s_{1,\ldots,}s_{u-1}}}(z) \tag{5.19}$$

where, $H_{s_{1,\ldots,}s_{u-1}}(z)$ represents FIR channel transfer function having memory size of $M-1-s_{u-1}$, which is further given by

$$H_{s_{1,\ldots,}s_{u-1}}(z) = \sum_{m=0}^{M-1-s_{u-1}} h_u(m, m+s_1,\ldots,m+s_{u-1})\ z^{-m} \tag{5.20}$$

and $X_{in_{s_{1,\ldots,}s_{u-1}}}(z)$ implies the Z transform of the analogous channel input $x_{in_{s_1,\ldots\ldots,s_{u-1}}}(n)$, and $\overline{Y}_u(z)$ is the Z transform of $\tilde{y}_u(n)$. Eq. (5.19) can be represented through block diagram as shown in Fig. 5.2 and it can be realized by $D(u,M)$ parallel FIR filters [24], which is calculated as

$$D(u,M) = \sum_{s=1}^{M} D(u-1,s)$$

with $D(1,M) = 1$, and the closed form is given by

$$D(u,M) = \frac{(u+M-2)!}{(u-1)!\,(M-1)!} = \binom{u+M-2}{u-1} \tag{5.21}$$

5.2. VSS-LMS ALGORITHMS FOR MULTICHANNEL VOLTERRA FILTER

As discussed in Section II, a multichannel system for the Volterra filter is given by equation (5.17) i.e.,

$$\tilde{y}_u = \sum_{s_1=0}^{M-1}\sum_{s_2=s_1}^{M-1}\cdots\sum_{s_{u-1}=s_{u-2}}^{M-1} .\ h_u(n, n+s_1,\ldots,n+s_{u-1}) * x_{in_{s_1,\ldots\ldots,s_{u-1}}}(n)$$

$$= H_u^T(n)X_{in_u}(n) \tag{5.22}$$

Let p be the number of inputs and k be the number of channel outputs of an adaptive Volterra filter and thus, the updated filter coefficients from Eq. (5.9) can be expressed as

$$H_{pk}(n+1) = H_{pk}(n) + \mu'_{pk}(n)\, X_{in_p}(n)\overline{e}_k(n) \tag{5.23}$$

In context to second order Volterra filter, updated filter coefficients from Eq. (5.10-5.13) can be written as

$$h_{pk1}(n+1; m_1) = h_{pk1}(n; m_1) - \frac{\mu'_{pk1}(n)}{2}\frac{\partial \overline{e}_k{}^2(n)}{\partial h_{k1}(n; m_1)} \tag{5.24}$$

$$= h_{pk1}(n; m_1) + \mu'_{pk1}(n)\overline{e_k}(n)X_{in_p}(n - m_1) \tag{5.25}$$

and

$$h_{pk2}(n+1; m_1, m_2) = h_{pk2}(n; m_1, m_2) - \frac{\mu'_{pk2}}{2}\frac{\partial \overline{e}_k{}^2(n)}{\partial h_{pk2}(n; m_1, m_2)} \tag{5.26}$$

$$= h_{pk2}(n; m_1, m_2) + \mu'_{pk2}(n)\overline{e}_k(n)X_{in_p}(n - m_1)X_{in_p}(n - m_2) \tag{5.27}$$

In the above equations, variable step-size $\mu'_{pk}(n)$ is used, which refines the performance of VLMS algorithms and can obtain a fast convergence speed and a small steady state mean square error. In [17-22], different VSS-LMS algorithms are used and all these methods are executed on the basic equations of the LMS algorithms, where the input and the noise signals are deduced to be statistically stationary. In this section we will discuss all the mathematical formulation of the VSS-LMS algorithms for a multichannel system identification process.

5.3. Methods for Variable Step Size VLMS Algorithms for Multichannel System

5.3.1. Karni's method (KVSS-VLMS)

In [17], Karni *et al.* have proposed a new convergence factor, which is TVing and are applied on adaptive LMS algorithm. In this thesis, the variable step size is applied on a multichannel system identification of the Volterra filter and is given by

$$\mu'_{pk}(n) = \mu'_{max_{pk}}\left(1 - \overline{e}_k^{\,-\alpha\left\|\overline{e}_k(n)X_{in_p}(n)\right\|^2}\right) \tag{5.28}$$

where, $\mu'_{max_{pk}} = \frac{1}{(L+1)\sigma^2_{X_{in_p}}}$, $\sigma^2_{X_{in_p}}$ is the variance of the input signal vector X_{in_p} for p inputs, L is the length of the filter vector, α is the damping parameter and should be greater than zero, $\overline{e_k}(n)$ is the error signal for k outputs. $\|.\|^2$ indicates the operation of squared Euclidean norm of the abrupt gradient vector $\overline{e_k}(n)X_{in_p}(n)$, which controls the step-size.

When $||\overline{e_k}(n)X_{in_p}(n)||$ is large, $\mu'_{pk}(n) = \mu'_{max_{pk}}$, i.e., the process is in its fast convergence state and vice-versa. For non-stationary input, the abrupt change of the input induces $||\overline{e_k}(n)X_{in_p}(n)||$ to become large and thus, brings the process back to the rapid convergence state automatically.

5.3.2. Kwong's method (KwVSS-VLMS)

In [18], Kwong and Johnston have proposed a VSS-LMS algorithm for the trailing of TVing order-I Markovian channels. Now, for a multichannel Volterra system variable step size algorithm is given by

$$\mu'_{pk}(n+1) = \beta\mu_{pk}(n) + \gamma\overline{e}_k^2(n) \tag{5.29}$$

with $0 < \beta < 1, \quad \gamma > 0$

and

$$\mu_{pk}(n+1) = \begin{cases} \mu_{max_{pk}} & if\ \ \mu'_{pk}(n+1) > \mu_{max_{pk}} \\ \mu_{min_{pk}} & if\ \ \mu'_{pk}(n+1) < \mu_{min_{pk}} \\ \mu'_{pk}(n+1) & otherwise \end{cases} \tag{5.30}$$

where, $0 < \mu_{min_{pk}} < \mu_{max_{pk}}$ and for guaranteed bounded MSE

$$\mu_{max_{pk}} \leq \frac{2}{3\ tr\left(E\left(X_{in_p}X_{in_p}^T\right)\right)} \tag{5.31}$$

The input X_{in_p} is presumed to be a zero mean independent sequence. Initially $\mu_0 = \mu_{max}$, although the algorithm is not fragile to the choice. The step size is controlled by the parameters β and γ, and the square of the prediction error, $\overline{e_k^2}(n)$. If the prediction error is large, it increases the step size and hence, provide faster tracking and similarly smaller prediction error will decrease the step size and reduce the misadjustment.

5.3.3. Mathews' method (MVSS-VLMS)

Mathews and Xie in [19], have presented an adaptive step size LMS algorithm for AF, which is changed with respect to a gradient descent algorithm. For a multichannel Volterra filter adaptive step size is given as

$$\mu'_{pk}(n) = \mu'_{pk}(n-1) + \rho\,\overline{e_k}(n)\,\overline{e_k}(n-1)X^T_{in_p}(n)X_{in_p}(n-1) \tag{5.32}$$

where, ρ is a positive constant usually small that controls the behaviour of the adaptive step size $\mu'_{pk}(n)$. To reduce the squared estimation error, a gradient descent algorithm is used at each time, which is swayed by the inner product between adjacent gradient vectors.

5.3.4. Aboulnasr's method (AVSS-VLMS)

In [20], Aboulnasr and Mayyaas have proposed a VSS-LMS AA, in which step size is controlled by the square of time-averaged autocorrelation of errors at instantaneous and past times. For a multichannel Volterra system, VSS-LMS algorithm is given as

$$\mu'_{pk}(n) = \alpha\mu'_{pk}(n-1) + \gamma\,\vartheta^2_{pk}(n) \tag{5.33}$$

where, $0 < \alpha < 1$, $\gamma > 0$ and the approximate is a time average of $\overline{e_k}(n)\overline{e_k}(n-1)$, which is described as

$$\vartheta_{pk}(n) = \beta\,\vartheta_{pk}(n-1) + (1-\beta)\overline{e_k}(n)\overline{e_k}(n-1) \tag{5.34}$$

where, β is a positive constant which lies between 0 and 1 and it is an exponential weighting specification that governs the quality of the time estimation.

$\vartheta_{pk}(n)$ is used in the update of $\mu'_{pk}(n)$, which serves two main purposes. First, the error autocorrelation is commonly a proficient measure of the contiguity to the optimum. Second, it spurned the effect of the uncorrelated noise signal on the step size update.

5.3.5. Pazaitis' method (PVSS-VLMS)

The step size in [21] is adjusted by the specimen fourth-order cumulant of the $\overline{e_k}(n)$, i.e., instantaneous error and if the adaptive step size is extended to multichannel Volterra system, it can be described as

$$\mu'_{pk}(n) = \mu'_{max_{pk}}\left(1 - \overline{e}_k^{-\alpha\,C_{pk}(n)}\right) \tag{5.35}$$

where,

$$C_{pk}(n) = \varsigma_{pk}(n) - 3\vartheta^2_{pk}(n) \tag{5.36}$$

is the kurtosis of the error, $\mu'_{max_{pk}}$ in equation (5.35) can be selected as the maximum value of μ'_{pk} that supports good convergence, α is a again a positive constant and the estimation of the second- and the fourth- order error moments are given as

$$\vartheta_{pk}(n) = \beta\, \vartheta_{pk}(n-1) + (1-\beta)\overline{e_k^2}(n) \qquad (5.37)$$

$$\varsigma_{pk}(n) = \beta\, \varsigma_{pk}(n-1) + (1-\beta)\overline{e_k^4}(n) \qquad (5.38)$$

where, β is the forgetting factor that controls the system memory and should be selected accordingly.

5.3.6. Wee-Peng's method (WPVSS-VLMS)

In [22], Wee-Peng *et al.* have proposed a new class of variable step-size algorithms, which outperforms another algorithms with reduced complexity. The adaptive step size for a multichannel Volterra system by following the approach of [22] can be given as

$$\mu'_{pk}(n) = \mu'_{pk}(n-1) + \gamma\, \overline{e_k}(n)\, X_{in_p}^{T}(n)\acute{g}_{pk}(n) \qquad (5.39)$$

where, γ is a constant, which is used to update the step size and further

$$\acute{g}_{pk}(n) = \beta\, \acute{g}_{pk}(n-1) + \overline{e_k}(n-1)X_{in_p}(n-1) \qquad (5.40)$$

where, β is very small positive constant proximity to one. When β is set equal to zero, Mathews' algorithm is obtained as seen in Eq. (5.32) and thus Mathews' algorithm is improved by the Wee-Peng's algorithm by utilizing smooth operation on one gradient vector to diminish the measured noise and can be observed in the simulation results, which are detailed in the next chapter.

CHAPTER 6

SIMULATION RESULTS

Abstract: *This chapter presents the simulation results of all the VSS-LMS algorithms individually and also propounds the results comparing with a fixed step-size VLMS algorithms under different constraints of SNR.. Different iterations are also taken into consideration for comparing all these algorithms under the same SNR conditions.*

6.1. Introduction

In this section, three cases are effectuated on noise processes in a multichannel nonlinear system identification system to analyse the performance of second-order VLMS based algorithm by using the different variable step-size algorithms under different conditions of SNR and thus comparing it with fixed step-size VLMS algorithm. In the entire case, the memory size *M* is selected to be 10, for which length of the adaptive filter is 65. According to the Monte-Carlo simulations, the performance of adaptive algorithms is differentiated on the basis of calculated performance appraisal factor as:

$$\hat{G}(n) = \sum_{g=1}^{G} \frac{\bar{e}_k^2(n)}{G} \tag{6.1}$$

where, G is the ensemble average of square error, $\bar{e}_k^2(n)$, for *k* channels of 700 independent experiments, which are plotted in semi-logarithmic scale w.r.t iterations to evaluate the performance. In this experiment, the total number of iterations are taken to be 2500. Let the number of input noise signal, *p*=1 and the number of channels, *k* =4.

In this experiment, logistic chaotic noise signal is considered and is given by the recursive equation [6, 9, 13, 14]:

$$x_{in}(n+1) = \lambda\, x_{in}(n)\big[\big(1 - x_{in}(n)\big)\big] \tag{6.2}$$

where, λ=4 and initialize x_{in} with 0.9. This nonlinear method is then normalized to possess unity signal power.

Four linear primary paths are used in this experiment, and there transfer functions are:

$$R_{1,1}(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7}$$
$$R_{1,2}(z) = z^{-5} - 0.2z^{-6} + 0.1z^{-7}$$

$$R_{1,3}(z) = z^{-5} - 0.3z^{-6} + 0.1z^{-7}$$

$$R_{1,4}(z) = z^{-5} - 0.2z^{-6} + 0.2z^{-7}.$$

Before discussing the three different cases for VSS-LMS algorithms, first we will check the result of the fixed step-size VLMS algorithm and also the results of each and every VSS-LMS algorithm for the iteration of 2500 and an ensemble average of 700.

6.2. Convergence Analysis of the multifarious algorithms

6.2.1. Fixed step-size VLMS algorithm

Fig. 6.1 shows the convergence characteristics of VLMS algorithm with a logistic chaotic input noise signal. In this experiment, we have chosen the step size to be $\mu_1 = 0.0002$, $\mu_2 = 0.00005$. It can be seen that the speed of the convergence is slow with the increase number of iteration.

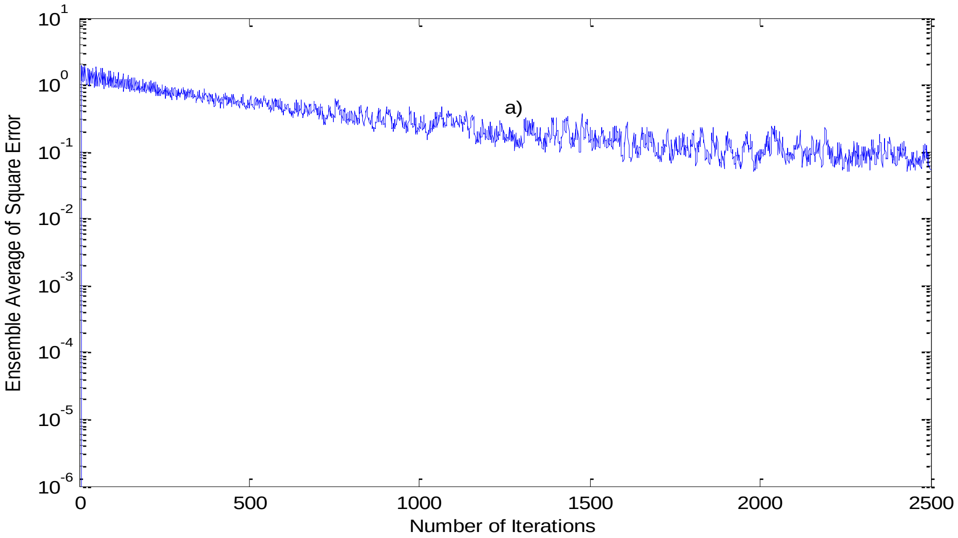

Fig. 6.1. Convergence analysis of fixed step-size Volterra least mean square algorithm

6.2.2. KVSS-VLMS (Karni’s Variable Step-size Volterra Least Mean Square) Algorithm

Fig. 6.2 shows the convergence behaviour of KVSS-VLMS algorithm for multichannelsystem identification with a logistic chaotic input noise signal having : $\mu_2 = 0.00005$, and the parameter $\alpha = 5$. Due to its variable step-size the convergence rate is better and thus there is a small misadjustment error.

6.2.3. KwVSS-VLMS (Kwong’s Variable Step-size Volterra Least Mean Square)

Another variable step-size algorithm is KwVSS-VLMS algorithm. The simulation result of the convergence analysis is given in Fig. 6.3. The parameters for KwVSS-VLMS: $\beta = 0.9997, \gamma = 5.8\text{x}10^{-4}\ \mu_{1k}(0) = 0.0054, \mu_2 = 0.00005$.

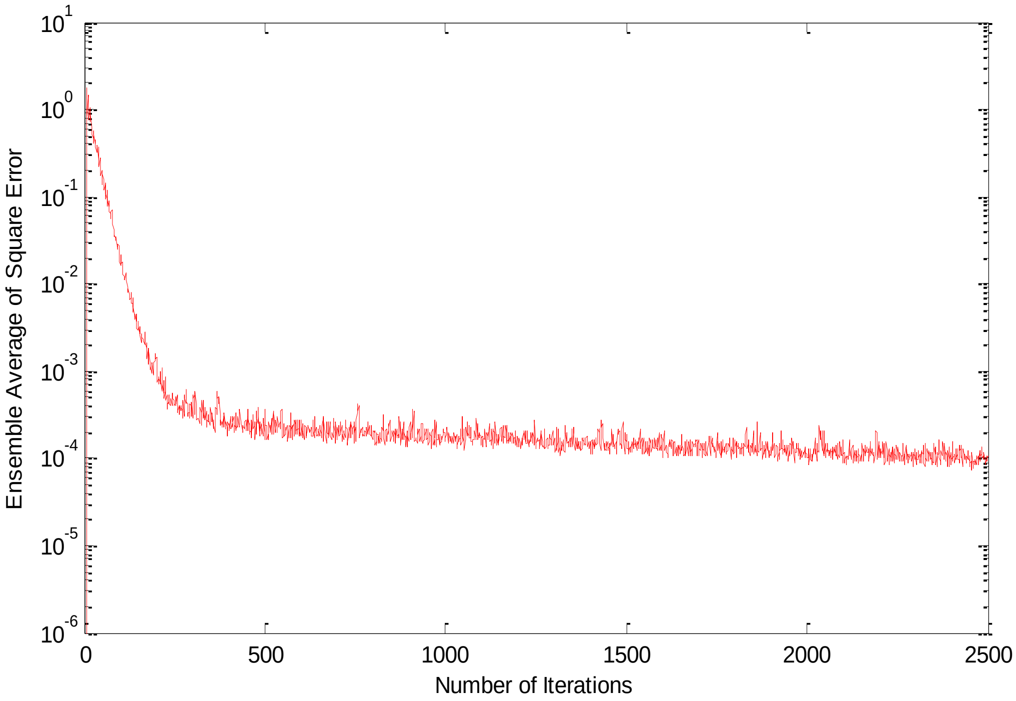

Fig. 6.2. Convergence analysis of KVSS-VLMS algorithm

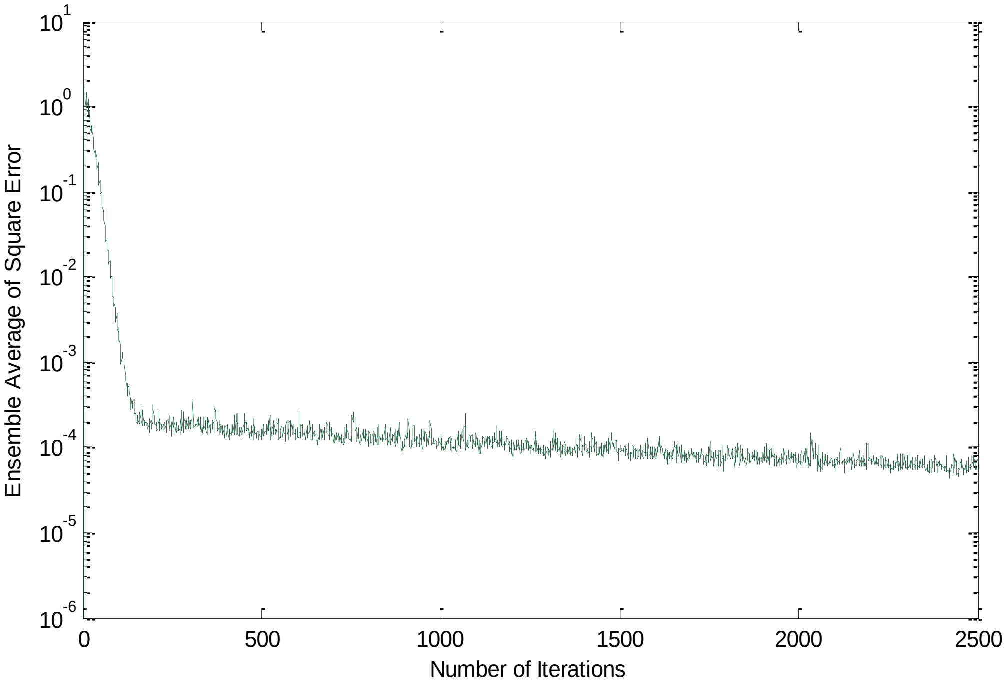

Fig. 6.3. Convergence analysis of KwVSS-VLMS algorithm

6.2.4. AVSS-VLMS (Aboulnasr Variable Step-size Volterra Least Mean Square)

As discussed in the last chapter, Aboulnasr in [20] proposed a new variable step-size algorithm for the purpose of minute misadjustment error. The same algorithm is used in this thesis for multichannel system identification by using the Volterra filter. The simulation result for the multichannel system identification is shown in Fig. 6.4. The parameters are: $\alpha = 0.97, \gamma = 10^{-3}$, $\mu_{1k}(0) = 0.02$, $\mu_2 = 0.00005, \beta = 0.9999$.

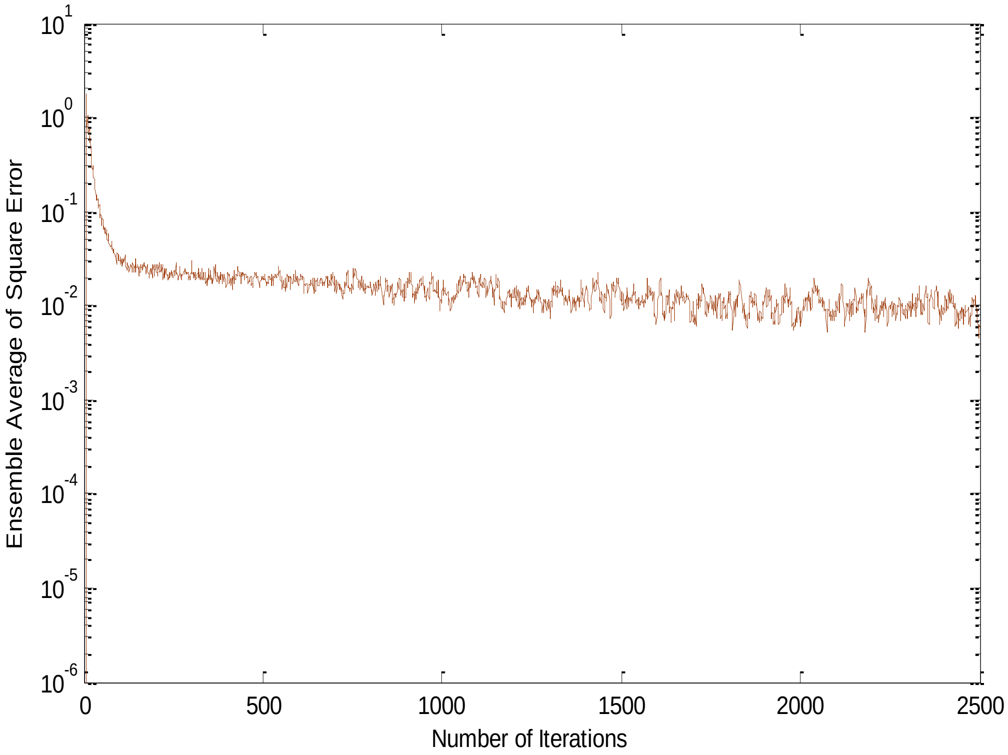

Fig. 6.4. Convergence analysis of AVSS-VLMS algorithm

6.2.5. PVSS-VLMS (Pazaitis Variable Step-size Volterra Least Mean Square)

Fig. 6.5 shows a convergence analysis of PVSS-VLMS algorithm under certain parameters. The parameters are: $\mu_2 = 0.00005$, $\alpha = 0.997, \beta = 0.999$, and $\mu'_{max_{pk}} = 0.048$. The initial adaptive step-size for PVSS-VLMS is set to zero. Also it can be seen that the PVSS-VLMS algorithm shows a good convergence.

6.2.6. MVSS-VLMS (Mathews Variable Step-size Volterra Least Mean Square)

Fig.6.6 represents the convergence analysis of MVSS-VLMS algorithms, which shows good convergence under a SNR of 15 dB. The parameters used in this case are: $\mu_{1k}(0) = 0.02$, $\mu_2 = 0.00005, \rho = 10^{-5}$.

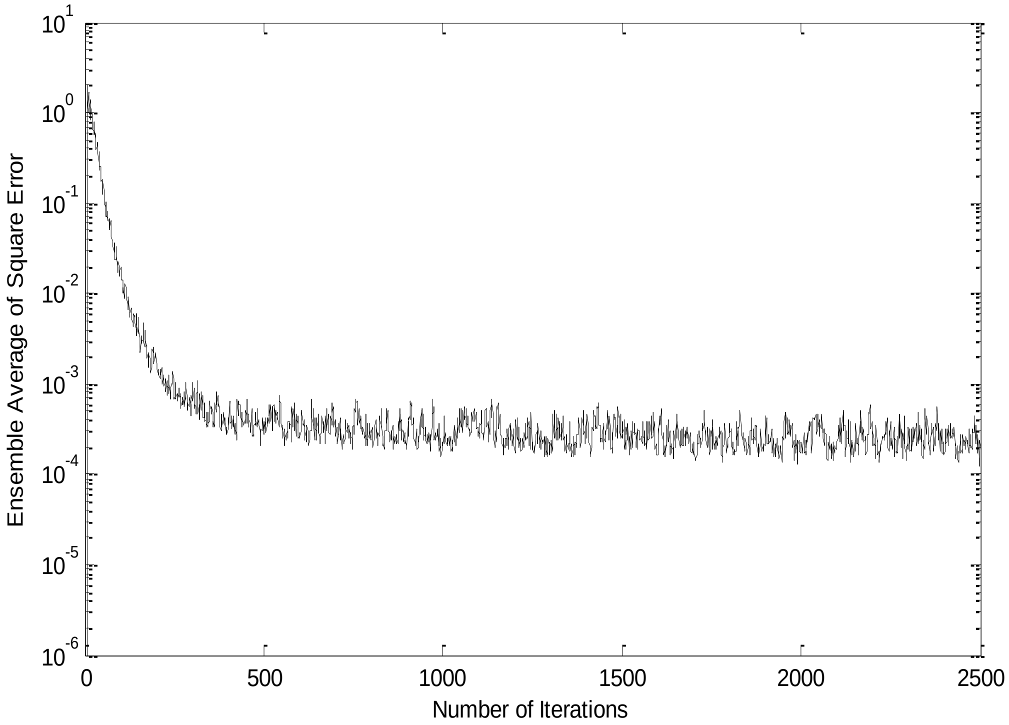

Fig. 6.5. Convergence analysis of PVSS-VLMS algorithm

6.2.7. WPVSS-VLMS (WeePeng Variable Step-size Volterra Least Mean Square)

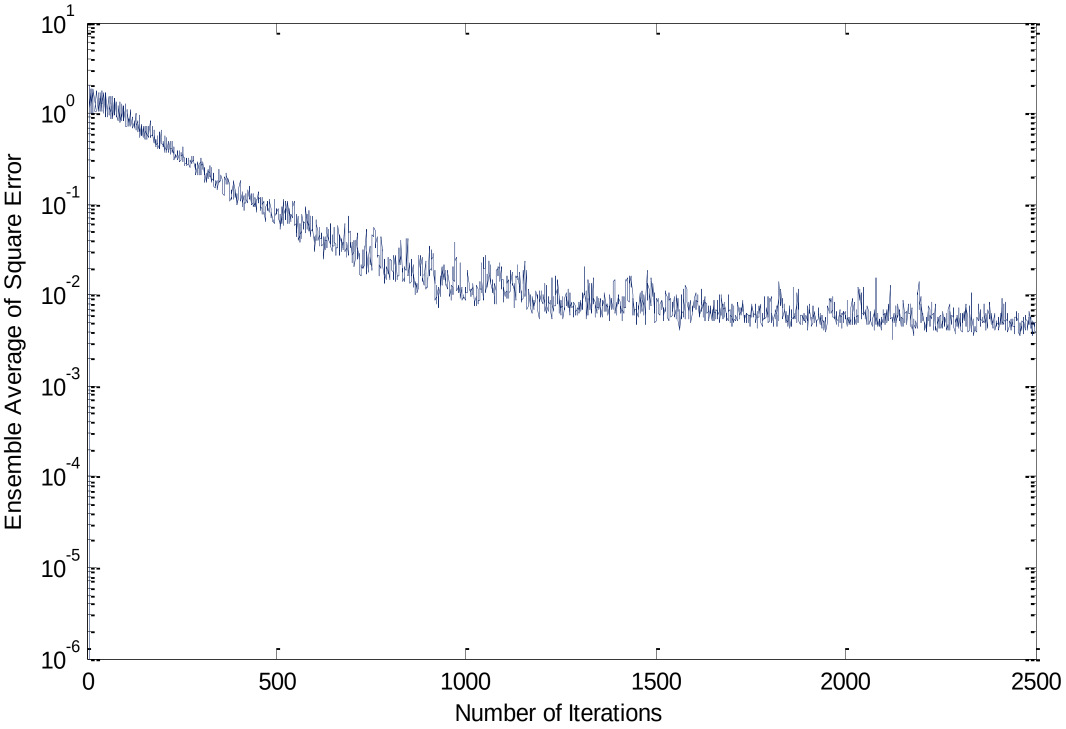

Fig. 6.6. Convergence analysis of MVSS-VLMS algorithm

As proposed in [22], WeePeng investigated a new algorithm for a single channel LMS algorithm. Thus, in this thesis the algorithm is extended to multichannel system using VLMS algorithm, and is shown in Fig. 6.7. The parameters used in the algorithms are: $\gamma = 4\text{x}10^{-2}$, $\beta = 0.9995$, $\mu_{1k}(0) = 0.02, \mu_2 = 0.00005$.

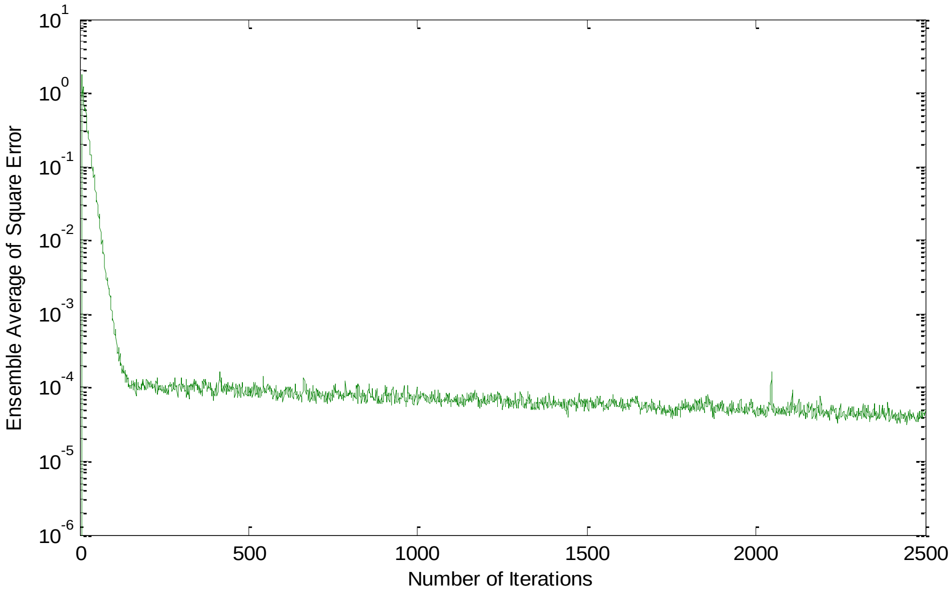

Fig. 6.7. Convergence analysis of WPVSS-VLMS algorithm

6.3. Comparison Of Convergence Characteristics of VSS-VLMS Algorithms with Fixed Step-size VLMS algorithm

Three different cases are taken into deliberation for comparison between the fixed step-size VLMS algorithm and variable step-size VLMS algorithms under different SNR conditions.

Case 1:

In this case, three algorithms (fixed VLMS, MVSS-VLMS, and WPVSS-VLMS) are taken into consideration, and are compared under low noise condition of SNR, which is equal to 10dB. The step-sizes and positive constants used are a) Fixed step VLMS: $\mu_1 = 0.0002, \mu_2 = 0.00005$; b) MVSS-VLMS: $\mu_{1k}(0) = 0.02$, $\mu_2 = 0.00005$, $\rho = 10^{-5}$;c) WPVSS-VLMS: $\gamma = 4\text{x}10^{-2}$, $\beta = 0.9995$, $\mu_{1k}(0) = 0.02, \mu_2 = 0.00005$. The variable step-size sways the issue of eigenvalue spread, and inevitably leads to the intensified convergence rate in the presence of chaotic input noise signal. It is evident from Fig. 6.8 that the performance of WPVSS-VLMS algorithm and MVSS-VLMS algorithm outperform the fixed step VLMS algorithm under the SNR of 10 dB.

Case 2:

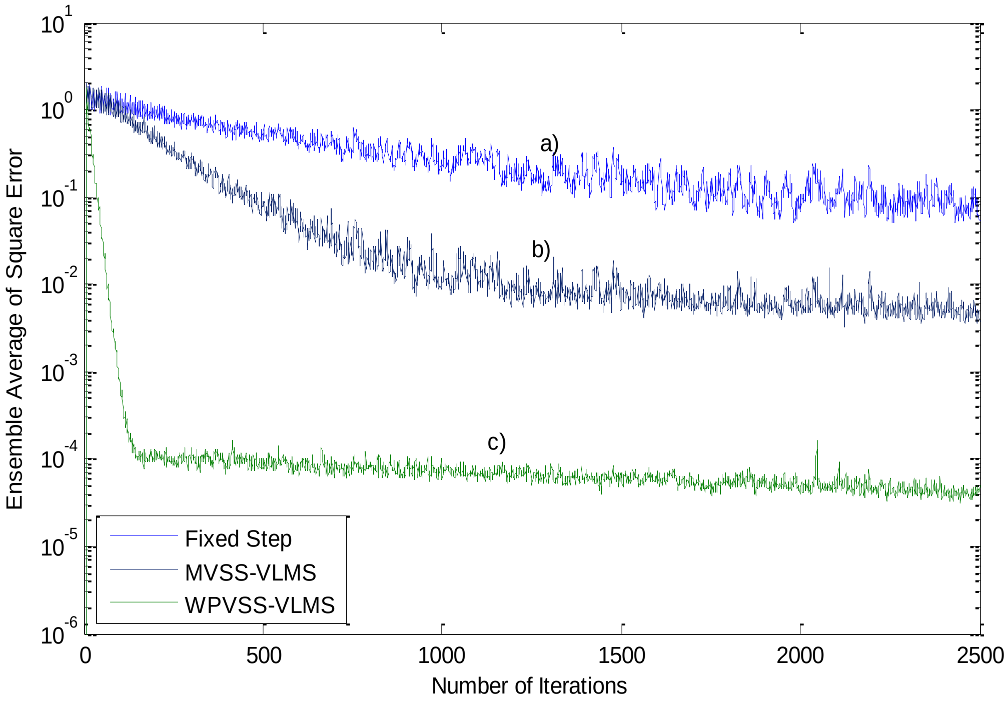

Fig.6.8. Comparison of convergence attributes with a logistic chaotic input noise signal between a) fixed step, b) MVSS-VLMS, and c) WPVSS-VLMS for second order Volterra filter.

In this simulation, convergence characteristics of the three algorithms i.e., fixed step VLMS algorithm, AVSS-VLMS, and PVSS-VLMS algorithms are analogized under the medium noise condition of SNR equivalent to 15 dB as shown in Fig. 4. Akin to the methodology opted in [20, 21], the constant parameter values of the above adaptive algorithms are stipulated to process a comparable level of misadjustment. The values of these parameters are: a) Fixed step VLMS: $\mu_1 = 0.0002, \mu_2 = 0.00005$; b) AVSS-VLMS: $\alpha = 0.97, \gamma = 10^{-3}$, $\mu_{1k}(0) = 0.02$, $\mu_2 = 0.00005, \beta = 0.9999$; c) PVSS-VLMS: $\mu_2 = 0.00005$, $\alpha = 0.997, \beta = 0.999$, and $\mu'_{max_{pk}} = 0.048$ which is the maximum value of μ'_{pk} that supports good convergence. The initial adaptive step-size for PVSS-VLMS is set to zero. It is very much apparent from Fig. 6.9 that PVSS-VLMS algorithm has good convergence, and outperforms the fixed step VLMS algorithm. Also, AVSS-VLMS algorithm has better convergence than the fixed step VLMS algorithm under medium noise condition.

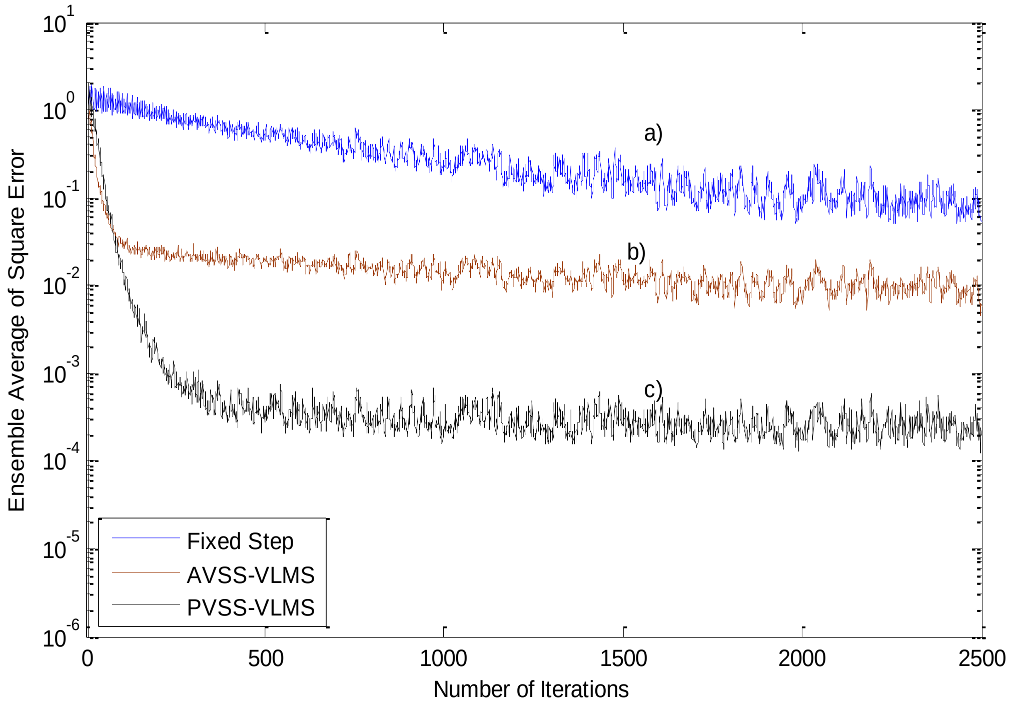

Fig.6.9. Comparison of convergence attributes with a logistic chaotic input noise signal between a) fixed step, b) AVSS-VLMS, and c) PVSS-VLMS for second order Volterra filter.

Case 3:

Now, in this case comparison is drawn between fixed step VLMS, KVSS-VLMS, and KwVSS-VLMS algorithms under the high noise condition of SNR 20 dB. The step-sizes and positive constants used are: a) Fixed step VLMS: $\mu_1 = 0.0002, \mu_2 = 0.00005$; b) KVSS-VLMS: $\mu_2 = 0.00005,\ \ \alpha = 5$; c) KwVSS-VLMS: $\beta = 0.9997,\ \ \gamma = 5.8\text{x}10^{-4}$ $\mu_{1k}(0) = 0.0054,\ \mu_2 = 0.00005$. In KVSS-VLMS [17], a proper damping parameter α should be selected for a fast convergence rate and very small misadjustment. From Fig. 6.10, it can be depicted that KwVSS-VLMS, and KVSS-VLMS outperform the fixed step VLMS and thus they yield a fast convergence rate.

It should be noted that value of step-size μ_2 is fixed in all the above cases as it will lead to misadjustment error if it is updated at each time. Thus, adaptive step-size is applied only on the linear portion of the adaptive filter.

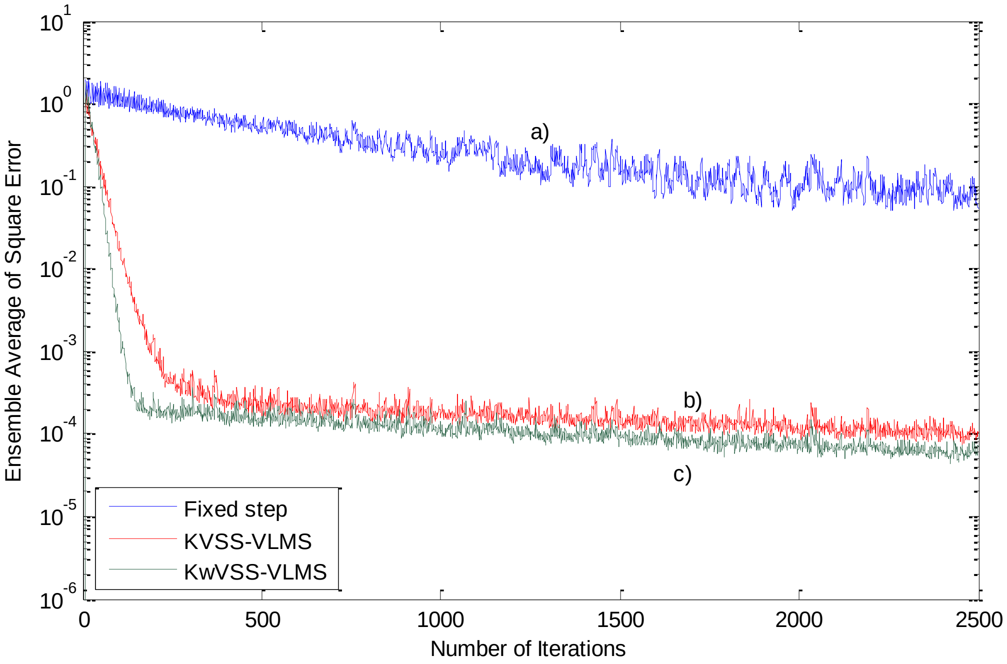

Fig. 6.10. Comparison of convergence attributes with a logistic chaotic input noise signal between a) fixed step, b) KVSS-VLMS, and c) KwVSS-VLMS for second order Volterra filter.

CHAPTER 7

CONCLUDING REMARKS AND FUTURE SCOPE

Abstract: *This chapter concludes the thesis with some noteworthy points about the VSS-LMS algorithms, and Volterra filter and some suggestions regarding the future scope.*

7.1. Conclusion

This thesis confers on an adaptive step-size LMS algorithms for the second-order Volterra filter based on a multichannel structure and compared with a fixed step-size VLMS algorithm. The objectives of this thesis was to confront the limitations of linear filtering techniques and to investigate new algorithms to overcome those limitations. There are numerous number of adaptive algorithms that can be seen in literature and every adaptive algorithm exhibits its own properties, but the motive of every algorithm is to attain MMSE rate at exorbitant convergence rate with minute computational complexity. Although several complex algorithms like RLS are having superior rendition but LMS is one of the utmost widespread algorithms in the arena of adaptive digital signal processing because of its easiness and simplicity. The VSS-LMS algorithm has been obtained from a fixed step-size LMS algorithm employing one of its constraints.

The first part of the thesis gives prominence on the basic principle of adaptive nonlinear filters, which proved to be worthwhile in adaptive system design. The drawbacks of linear adaptive filter are elucidated. Numerous practical scenarios, where nonlinear falsifications are existent were discussed. Also, nonlinear filtering methods that had been employed in practice were discussed. VSS-LMS algorithms were instigated to counter the limitations of fixed step-size algorithm. Volterra LMS algorithm extend the idea of linear adaptive filtering to nonlinear problems. VSS-VLMS algorithms showed encouraging results when applied to multichannel system identification on nonlinear ISI channel. The adaptive proficiency permits the processing of inputs whose behavior is not known and in some situations nonstationary. Output noise of the system and signal distortion are usually inferior than that can be achieved with typical optimal filter configurations.

In the second part of the thesis, expounded mathematical formulation of a multichannel system identification, and adaptive step-size algorithms was depicted. It is discerned from simulation results that variable step-size algorithms outplay and have superior rendition to the fixed step-size VLMS algorithm in controlling all sorts of nonlinear noise processes in high, medium, and low SNR environments as the variable step-size diminishes the susceptivity of the misadjustment to the degree of nonstationarity and also lessen the accord between misadjustment and trailing ability of the fixed step-size LMS algorithm.

The simulation results exhibit that the preliminary convergence rate of the adaptive Volterra filters is very rapid and after an initial span when the step-size proliferates, the step-size starts dwindling slowly and smoothly and yields to a small misadjustment errors. But, in case of nonstatic environments, the algorithms seek to calibrate the step-sizes in such a manner, so as to obtain proximate to the foremost possible performance. Variable step-size algorithm also controls the unpropitious effects of spreading of an eigenvalue of the autocorrelation matrix of the input signal.

7.2. Future Scope

There is passably substantial amount of research activity progressing in this domain at present. Many signal processing applications mandate for lessening in accord between convergence rate and misadjustment, taking insight of algorithm into account. As the significance of nonlinear system increases, the intricacy of nonlinear systems also increases. Here we have executed simulations of adaptive algorithm for second-order Volterra system, but with the proliferation in the sources of nonlinearity, we have to gaze upon certain refined adaptive algorithm that is used to overwhelm the nonlinear effects. Therefore, we can smear these adaptive algorithms in the estimation of channel for modern wireless technologies like BLAST, 3G, 4G etc.

In this thesis, only the VLMS and VSS-VLMS algorithms have been used. Other adaptive algorithms can be calculated and their appropriateness for application to adaptive noise cancellation can also be related. Additional algorithms can be used like RLS, NLMS, GVSS-LMP etc for multichannel system identification. Also, this thesis primarily dealt with transversal FIR and Volterra adaptive filters. Other methods like IIR or lattice filtering may substantiate to be active in many applications.

The fine properties and the computational intelligibility related with the algorithm makes us anticipate to see massive number of advance techniques being evolved, and will be a potential quantum leap with significant impact on practical applications like nonlinearly magnified analog as well as digital communication signal processing [25], biomedical engineering [26], equalization of nonlinear communication medium [27].

There’s a lot, which can be prepared in future for an enhancement on the method of Volterra multichannel system identification and its applications. The arena of DSP and in particular adaptive nonlinear filtering is massive and advance research and development in this field can consequence in some enhancement on the approaches studied in this thesis.

LIST OF PUBLICATION

1. Sandipta Dutta Gupta and A.K. Kohli, “VSS-LMS Algorithms for Multichannel System Identification Using Volterra Filtering,” *SSRG International Journal of Electronics and Communication Engineering*, vol. 2, no. 5, pp. 27- 35, May 2015.

REFERENCES

[1] V. John Mathews, “Adaptive Polynomial Filters,” *IEEE Signal Process. Mag.,* vol. 8, no. 3, pp. 10-26, Jul. 1991.

[2] A. Stenger, L. Trautmann, and R. Rabenstein, “Adaptive Volterra filters for nonlinear acoustic echo cancellation,” in *Proc. IEEE Workshop Nonlinear Signal Image Process.,* pp. 679-683, 1999.

[3] L. F. C. Pessoa, “ MRL-Filters: A general class of non-linear systems and their optimal design for image processing,” *IEEE Trans. on Image Process.,* vol. 7, no. 7, pp. 966-978, Jul. 1998.

[4] E. J. Coyle and J.H. Lin, “Stack filters and the mean absolute error criterion,” *IEEE Trans. Acoust. Speech Signal Process.,* vol. 36, no. 8, pp. 1244-1254, Aug. 1988.

[5] A. K. Kohli and A. Rai, “Numeric variable forgetting factor RLS algorithm for second-order Volterra filtering,” *Circuit Syst. Signal Process.,* vol. 32, no.1, pp. 223-232, Feb. 2013.

[6] L. Tan and J. Jiang, “Adaptive Volterra filters for active control of nonlinear noise processes,” *IEEE Trans. Signal Process.,* vol. 49, no. 8, pp. 1667-1676, Aug. 2001.

[7] V. J. Mathews and G.L. sicuranza, *Polynomial Signal Processing.* New York: John Wiley & Sons Inc., 2000.

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

[9] D.P. Das, S.R. Mohapatra, A. Routray and T.K. Basu, “Filtered-s LMS algorithm for multichannel active control of nonlinear processes,” *IEEE Trans. on Speech and Audio Process.,* vol. 14, no. 5, pp. 1875-1879, Sep. 2006.

[10] S.M. Kuo and D.R. Morgan, *Active noise control systems-Algorithms and DSP implementations*, New York: Wiley, 1996.

[11] P. Strauch and B. Mulgrew, “Active control of nonlinear noise processes in a linear duct,” *IEEE Trans. Signal Process.,* vol. 46, no. 9, pp. 2404-2412, Sep. 1998.

[12] M. Bouchard, B. Paillard, and C. T. L. Dinh, “Improved training of neural networks for the nonlinear active noise control of sound and vibration,” *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 391-401, Mar. 1999.

[13] D.P. Das and G. Panda, “Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm,” *IEEE Trans. Speech Audio Process.,* vol. 12, no. 3, pp. 313-322, May 2004.

[14] G.L. Sicuraranza and A.C. Carini, “Filtered-x affine projection algorithm for multichannel active noise control using second-order Volterra filters,” *IEEE Signal Process. Lett.,* vol. 11, no. 11, pp. 853-857, Nov. 2004.

[15] B. Farhang- Boroujeny, *Adaptive Filters: Theory and Applications,* New York: John Wiley & Sons, 1998.

[16] Y. Zhang, N. Li, J. A. Chambers, and Y. Hao, “New gradient-based variable step size LMS algorithms,” *EURASIP Journal on Advances in Signal Process.,* vol. 2008, no. 2, pp. 1-9, Feb. 2008.

[17] S. Karni and G. Zeng, “A new convergence factor for adaptive filters,” *IEEE Trans. on Signal Process.,* vol. 40, no. 7, pp. 1633-1642, Aug. 1989.

[18] R.H. Kwong and E. W. Johnson, “A variable step-size LMS algorithm,” *IEEE Trans. on Signal Process.,* vol. 40, no. 7, pp. 1633-1642, Jul. 1992.

[19] V.J. Mathews and Z. Xie, “A stochastic gradient adaptive filter with gradient adaptive step size,” *IEEE Trans. on Signal Process.,* vol. 41, no. 6, pp. 2075-2087, Jun. 1993.

[20] T. Aboulnasr and K. Mayyas, “A robust variable step-size LMS-type algorithm: analysis and simulations,” *IEEE Trans. on Signal Process.,* vol. 45, no. 3, pp. 631-639, Jan. 1997.

[21] D.I. Pazaitis and A. G. Constantinides, “A novel kurtosis driven variable step-size adaptive algorithm,” *IEEE Trans. on Signal Process.,* vol. 47, no. 3, pp. 864-872, Mar. 1999.

[22] W.P. Ang and B. Farhang- Boroujeny, “A new class of gradient adaptive step-size LMS algorithms,” *IEEE Trans. on Signal Process.,* vol. 49, no. 4, pp. 805-810, Apr. 2001.

[23] A. Rai and A. K. Kohli, “Adaptive polynomial filtering using generalised variable step-size least mean *p*th power (LMP) algorithm,” *Circuit System and Signal Process.,* vol. 33, no. 12, pp. 3931-3947, Dec. 2014.

[24] G.M. Raz and B. V. Veen, “ Baseband Volterra filters for implementing carrier based nonlinearities,” *IEEE Trans. Signal Process.,* vol. 46, no. 1, pp. 103-114, Jan. 1998.

[25] T. Ogunfunmi, *Adaptive non-linear system identification: Volterra and Weiner model approches*. NewYork: Springer, 2007.

[26] D. T. Westwick, R. E. Kearney, *Identification of Nonlinear Physiological Systems*. New York: Wiley, 2003.

[27] T. Ogunfunmi and T. Drullinger, "Equalization of non-linear channels using a Volterra-based non-linear adaptive filter," in *Proceedings of IEEE International Midwest Symposium on Circuits and Systems, Seoul, South Korea,* pp. 1-4, Aug. 2011.

[28] S.Haykin, *Adaptive filter theory,* 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1996.

[29] D. Mansour and A. H. Gray, "Frequency domain non-linear adaptive filter," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 6, pp. 550-553, Apr. 1981.

[30] O. Agazzi, D.G. Messerchmitt, and D. A. Hodges, "Nonlinear echo cancellation of data signals," *IEEE Trans. Commun.,* vol. COM-30, no. 11, pp. 2421-2433, Nov. 1982.

[31] E. J. Thomas, "Some considerations on the application of the Volterra representation of nonlinear networks to adaptive echo cancelers," *Bell System Technical J.,* vol. 50, no. 8, pp. 2797-2805, Oct. 1971.

[32] G. L. Sicuranza, A. Bucconi, and P. Mitt-I, "Adaptive echo cancellation with nonlinear digital filters," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 3.10.1-4, Mar. 1984.

[33] D.D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission systems," *Bell System Technical Journal,* vol. 57, no. 7, pp. 2589-2611, Sep. 1978.

[34] E. Biglieri, A. Gersho, R.D. Gitlin, and T.L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. Selected Areas in Communications,* vol. SAC-2, no. 5, pp. 765-777, Sep. 1984.

[35] M.J. Coker and D.N. Simkins, "A nonlinear adaptive noise canceller," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 5, pp. 470-473, Mar. 1980.

[36] T. Koh and E.J. Powers, "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Trans. Acoust., Speech, and Signal Processing,* vol. 33, no. 6, pp. 1445-1455, Dec. 1985.

[37] J.B. MacNeil, R.E. Kearney, and I.W. Hunter, “Identification of time varying biological systems from ensemble data,” *IEEE Trans. on Biomed. Eng.,* vol. 39, no. 12, pp. 1213-1225, Dec. 1992.

[38] A. Benveniste, “Design of adaptive algorithms for the tracking of time varying systems,” *International Journal of Adaptive Contr. Signal Processing,* vol. 1, no. 1, pp. 3-29, 1987.

[39] O. Macchi, “Optimization of adaptive identification for time-varying filters,” *IEEE Trans. Automat. Contr,* vol. AC-31, no. 3, pp. 283-287, Mar. 1986.

[40] S.C. Douglas, “Introduction to adaptive filters,” *Digital Signal Processing Handbook,* V.K. Madisetti and D.B. Williams, ed., CRC Press, pp. 414-422, 1999.

[41] R.W. Lucky, “Techniques for adaptive equalization of digital communication systems,” *Bell Sys. Tech. Journal,* vol. 45, no. 2, pp. 255-286, Feb. 1996.

[42] M. Rupp, “A family of adaptive filter algorithms with decorrelating properties,” *IEEE Trans. on Signal Process.,* vol. 46, no. 3, pp. 771-775, Sep. 1993.

[43] S.U.H. Qureshi, “Adaptive equalization,” *Proc. IEEE,* vol. 73, no. 9, pp. 1349-1387, Sep. 1985.

[44] B. Widrow and S.D. Stearns, *Adaptive Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1985.

[45] R.W. Lucky, “Modulation and detection for data transmission on the telephone channel,” *New directions in Signal Process. in Comm. and Control,* Leiden, Holland, 1975.

[46] J. Katzenelson and L.A. Gould, “The design of nonlinear filters and control systems,” *Journal of Information and Control,* vol. 5, no. 2, pp. 108-143, Jun. 1962.

[47] J.F. Barret, “The use of functionals in the analysis of nonlinear system,” *Journal of Elect. and Comm.,* vol. 15, no. 6, pp. 567-615, May 2007.

[48] P. Eykhoff, “Some fundamental aspects of process-parameter estimation,” *IEEE Trans. on Automatic Control,* vol. 8, no. 4, pp. 347-357, Oct. 1963.

[49] A.V. Barakrishnan, “A general theory of nonlinear estimation problems in control systems,” *Journal of Mathematical Analysis and Applications,* vol. 8, no. 1, pp. 4-30, Feb. 1964.

[50] B. Widrow and M. McCool, “A comparison of adaptive algorithms based on the methods of steepest descent and random search,” *IEEE Trans. on Antenna and Propagation,* vol. AP-24, no. 5, pp. 615-637, Sep. 1976.

[51] M.L. Honig and D.G. Messerschmitt, "Adaptive Filters- Structures, Algorithms, and Applications," *IEEE Trans. on Acoust., Speech, and Signal Process.,* vol. ASSP-33, no. 4, pp. 1345-1346, Oct. 1985.

[52] J.R. Treichler, CR. Johnson, and M.G. Larimore, *Theory and Design of Adaptive Filters.* New York: John Wiley & Sons, 1987.

[53] K.W. Martin and M.T. Sun, "Adaptive filters suitable for real-time spectral analysis," *IEEE Trans. on Circuits and Systems,* vol. CAS-33, no. 2, pp. 218-229, Feb. 1986.

[54] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. on Acoustics, Speech, and Signal Process.,* vol. ASSP-33, no. 1, pp. 222-230, Feb. 1985.

[55] L. I. Horowitz and K.D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-29, no. 3, pp. 722-736, Jun. 1981.

[56] G.L. Sicuranza and G. Ramponi, "Adaptive nonlinear digital filters using distributed arithmetic," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-34, no. 3, pp. 518-526, Jun. 1986.

[57] G.L. Sicuranza and G. Ramponi, "A variable-step adaptation algorithm for memory-oriented Volterra filters," *IEEE Trans. Acoustics, Speech, and Signal Process.,* vol. ASSP-35, no. 10, pp. 1492-1494, Oct. 1987.

[58] C.E. Davila, A.J. Welch, and H.G. Rylander, "A second-order adaptive Volterra filter with rapid convergence," *IEEE Trans. Acoust., Speech, and Signal Process.,* vol. ASSP-35, no. 9, pp. 1259-1263, Sep. 1987.

[59] A.A.M. Saleh and J. Salz, "Adaptive linearization of power amplifiers in digital radio systems," *Bell System Technical Journal,* vol. 62, no. 4, pp. 1019-1033, Apr. 1983.

[60] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family," *IEEE Trans. on Information Theory,* vol. IT-30, no. 2, pp. 275-283, Mar. 1984.

[61] B. Widrow, J.M. McCool, M.G. Larimore, and C.R. Johnson, "Stationary and nonstationary learning characteristic of the LMS adaptive filter," *Proceedings of the IEEE,* vol. 64, no. 8, pp. 1151-1162, Aug. 1976.

[62] B. Widrow, M. Cool, and M. Ball, "The complex LMS algorithm," *Proceedings of the IEEE,* vol. 63, no. 4 , pp. 719-720, Apr. 1975.

[63] S. Makino, Y. Kaneda, and N. Koizumi, “Exponentially weighted step-size NLMS adaptive filter based on the statistics of a room impulse response,” *IEEE Trans. on Speech and Audio Processing,* vol. 1, no. 1, pp. 101-108, Jan. 1993.

[64] D.Z. Feng, X. Zhang, D. Chang, and W. Xing, “A fast recursive total least squares algorithm for adaptive FIR filtering,” *IEEE Trans. on Signal Processing,* vol. 52, no. 10, pp. 2729-2737, Oct. 2004.

[65] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation.* New York: Springer-Verlang, 1990.