ADAPTIVE POLYNOMIAL FILTERING FOR SYSTEM IDENTIFICATION USING MODIFIED SIGMOID VARIABLE STEP-SIZE LMS ALGORITHM

Submitted towards the partial fulfilment of requirement for the award of degree of

Master of Engineering
In
Electronics and Communication Engineering

Submitted by:

SHIV KUMAR

(801461028)

Under the guidance of:

Dr. Amit Kumar Kohli

(Associate Professor, ECED)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

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

PATIALA – 147004, PUNJAB, INDIA

JULY 2016

DECLARATION

I, **Shiv Kumar**, hereby declare that the thesis entitled **"Adaptive Polynomial Filtering for System Identification using Modified-Sigmoid Variable-Step-Size LMS Algorithm"** is an authentic record of my own work carried out towards the partial fulfillment for the award of degree of Master of Engineering in "Electronics and Communication Engineering" at Thapar University, Patiala, under the supervision of **Dr. Amit Kumar Kohli,** Associate Professor, Electronics and Communication Engineering Department.

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

Shiv Kumar

Date: 8/7/16

Shiv Kumar

Roll No. 801461028

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

Amit Kohli 8/7/16

Date: 8/7/16

Dr. Amit Kumar Kohli

Associate Professor, ECED

Thapar University, 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

Foremost, I would like to express my sincere gratitude toward my guide **Dr. Amit Kumar Kohli**, Associate Professor and P.G. coordinator, ECED, Thapar University, Patiala, for his motivation, patience, enthusiasm, immense knowledge and continuous support for my thesis. His guidance helped me in research and writing of thesis all the time. I would not have imagined having a better guide for my thesis.

Beside my guide, I am thankful to **Dr. Sanjay Sharma**, Professor and Head, Electronics and Communication Engineering Department and **Dr. Amit Mishra**, Assistant Professor, Program Coordinator of Electronics and Communication, for providing me with adequate infrastructure in carrying out my work.

I would also like to thank all the faculty members of ECED for their full support of my work. I am also thankful to my friends **Rohit Garg, Sonali** and **Shivani Sehrawat** who have helped me in correcting grammar and consistent notation in my writing and for carefully reading and commenting on countless revisions of my manuscript. I am also thankful to the authors whose work have been consulted and quoted in this work.

Last but not the least, I would also like to thank my parents for supporting me financially, morally, spiritually and always motivated me throughout my entire life to work sincerely and hardly to pursue excellance.

Place: Thapar University, Patiala Shiv Kumar

Date: Roll No. 801461028

ABSTRACT

Adaptive polynomial filtering comprises one of the primary technologies in signal processing, and it investigates many applications in the area of industry and science. These techniques are employed in a vast range of applications, for example: adaptive echo/noise cancellation system, adaptive equalization, adaptive beamforming and system identification. The current trend in the telecommunication system design is the process of identification and minimization of undesired non-linearities, as these have adverse effects on underlying system operation. The use of nonlinear models, like Volterra series, can minimize all these non-linearities. Adaptive approaches and algorithms are extensively utilized for the estimation of Volterra kernels, under the constraint of unknown non-linear system. The accuracy of the estimation of kernels is used to investigate the precision of the system model and inverse system. This thesis propounds the adaptive polynomial filtering for system identification using variable-step-size least-mean-square (VSS-LMS) algorithms, and these VSS algorithms are compared with the fixed-step-size least-mean-square (FSS-LMS) algorithm. Different VSS-LMS algorithms are also compared with each other. These all algorithms are applied to the second-order-Volterra (SOV) filter, under the various noise constraints for different values of signal-to-noise ratio (SNR). The VSS-LMS algorithm corroborates steady state behavior during convergence. The step-size of the adaptive filter is altered in compliance with a gradient based descent algorithm to minimize the squared estimation error in the course of each iteration. It also improves tracking performance in the smoothly time-varying environments for the choice of the parameters and the boundary conditions of adaptive filter.

First, we apply the sigmoid-variable-step-size least-mean-square (SVSS-LMS) algorithm to SOV filter, in which the adaptive step-size is modelled using sigmoid function. It gives fast convergence when compared with the FSS-LMS algorithm. Following this for polynomial filtering, we have used modified-sigmoid-variable-step-size least-mean-square (MSVSS-LMS) algorithm, which gives better convergence and tracking performance in comparison to SVSS-LMS algorithm under similar conditions. Other VSS-LMS algorithms like Kwong-variable-step-size least-mean-square (KVSS-LMS) algorithm (Kwong *et al*., 1992), Aboulnasr-variable-step-size-least mean-square (AVSS-LMS) algorithm (Aboulnasr *et al*., 1997) and modified-Aboulnasr-variable-step-size least-mean-square (MAVSS-LMS) algorithm (Kun *et al.*, 2009) are also compared with FSS-LMS algorithm and with each other. Simulation results are also presented to demonstrate that MSVSS-LMS algorithm is a better option than SVSS-LMS and FSS-LMS algorithm in the second-order Volterra filtering applications for non-linear system identification. In non-linear signal processing scenario, MSVSS-LMS algorithm is marginally inferior to KVSS-LMS under static environment in terms of mean squared error in the convergence and tracking mode.

Undoubtedly, MAVSS-LMS is the best algorithm under similar non-linear conditions, which is substantially better than AVSS-LMS and KVSS-LMS algorithm. But, MSVSS-LMS algorithm is finding applications in neural signal processing.

Keyword**:** Volterra filter, polynomial filter, adaptive filter, variable-step-size, fixed-step-size, LMS algorithm, mean square error

TABLE OF CONTENTS

LIST OF ACRONYMS AND ABBREVIATIONS

AF	Adaptive Filter
AVSS	Aboulnasr Variable Step-Size
AWGN	Additive White Gaussian Noise
DPCM	Differential Pulse Code modulation
FIR	Finite Impulse Response
FSS	Fixed Step-Size
FFT	Fast Fourier Transform
IID	Independent and Identically Distributed
KVSS	Kwong Variable-Step-Size
LHS	Left Hand Side
LMS	Least Mean Square
LTI	Linear Time Invariant
MAVSS	Modified Aboulnasr Variable Step-Size
MD	Multidimensional
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
MSVSS	Modified Sigmoid Variable Step-Size
NLMS	Normalized Least Mean Square
RLS	Recursive Least Square

RHS	Right Hand Side
SNR	Signal to Noise Ratio
SOV	Second Order Volterra
SS	Step-Size
SVSS	Sigmoid Variable Step-Size
VSS	Variable Step-Size
TJR	Trunk and Junction Routing
w.r.t	With Respect To

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

This chapter introduces a brief description to the subject of adaptive filter and linear & non-linear systems. The importance of the research in this area and its applications to industry are affirmed. This chapter includes the applications of adaptive filter, but the main focus is on system identification application. This chapter also includes structural implementation of adaptive filters and also the brief description of Volterra series and its application in non-linear system modeling, due to its flexible nature.

1.1 Adaptive Filters

In the last four decades, the domain of discrete-time signal processing has grown at a rapid pace. The remarkable growth and development in the sphere of discrete-time signal processing has led to adaptive filtering and its applications. The purpose of filtering is to treat a signal in order to extract significant data encompassed in it. A discrete-time filter is one that treats signals detailed in discrete-time sample format. When dealing with such signals, whose statistical characteristics are static, the analyst may simply select the most appropriate procedure to treat the signal. However, fixed signal processing algorithms cannot process a signal proficiently if its characteristics are unknown. The remedy is to utilize the adaptive filtering configuration, which may alter its characteristics automatically.

The design of discrete-time filters with fixed tap-coefficients needs prearranged detailing riders. Moreover, there are different scenarios, where the riders are unavailable, or time-variant, an adaptive filtering configuration is needed, when either the fixed specifications are unidentified or the required parameters may not be satisfied by the time-invariant filtering configurations. The role of the adaptive filtering configuration is to act adequately under unspecified environment and track time changes of input signal. This makes adaptive filter a dominant tool for discrete-time signal processing and control appliances. There are enormous various applications, in which adaptive schemes may be utilized.

Some examples include system identification, echo cancellation, equalization of dispersive channels, adaptive beamforming and automatic control. As power of discrete-time signal processors have got boosted, adaptive filtering configurations became much more usual and are presently being utilized in devices like mobile sets and various communication equipments.

Adaptive filters [1] are considered as non-linear systems because their parameters are updated from one iteration to the next, and therefore, the parameter becomes data dependent. The general setup for the adaptive filter is demonstrated in Fig. 1.1.

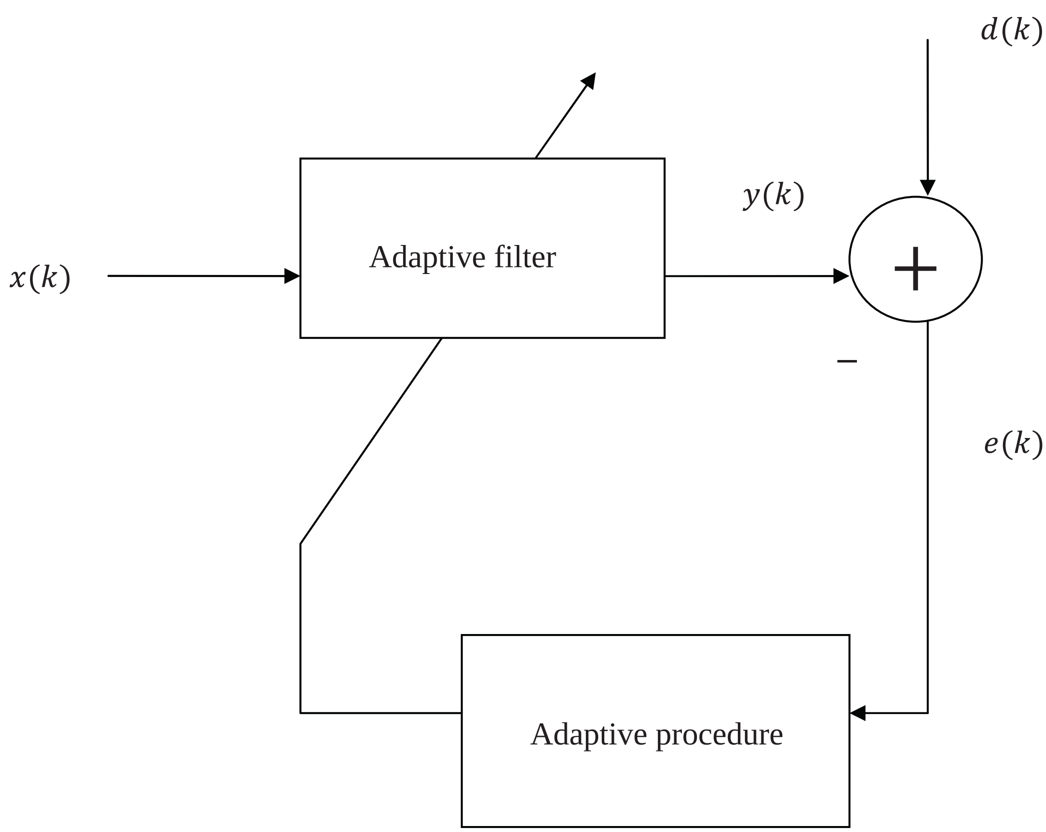

Fig. 1.1. Commonly used adaptive filtering structure.

The full detailing of the adaptive filter configuration illustrated in Fig. 1.1 encompasses three components.

1) Application: The choice of the input discrete-time signal and desired output discrete-time signal depends on the type of application. There are enormous applications of AFs such as linear and non-linear system identification, dispersive channel equalization, echo cancellation, prediction, inverse channel modeling, adaptive beamforming etc.

2) Adaptive-filter configuration: The adaptive filter may be incorporated by using various configuration or implementations. The selection of filter configuration may affect computational burden in terms of arithmetic operations or the amount of digital hardware requirements. We can implement the adaptive filter structures using various linear filters, such as FIR, IIR or by non-linear filters, such as second-order Volterra filter.

3) Algorithm: The algorithm is a methodology for updating tap- coefficients of the adaptive filtering configuration in order to optimize desired criterion used for particular application. The algorithm is determined by the aspects like error signal nature, objective function, etc. The choice of particular adaptive algorithm influences the overall aspects of the adaptive process.

1.2 Performance Measures in Adaptive Systems

The various performance measures of adaptive system [1] are described as follows

- **Convergence Rate**

The rate of convergence decides the speed, with which the filter coefficient vector converges to its optimal solution. Commonly, a quicker convergence rate is a most wanted characteristic for an adaptive algorithm. The higher the speed of convergence, better is the algorithm suited for the rapid fluctuation in the signal characteristics. Convergence rate is dependent on other performance characteristics.

- **Minimum Mean Squared Error**

The minimum-mean-squared-error (MMSE) is a measure of how accurately a system can converge to a predetermined optimal target. A low value of MMSE is an indicator that adaptive system has been appropriately modelled. A large MMSE commonly reflects that an adaptive filtering configuration cannot precisely model the underlying system; or beginning state of filter is the inappropriate beginning point to force an adaptive filtering configuration to converge to an optimal solution.

- **Computational Complexity**

Adaptive filter encompasses a discrete-time filter and an adaptive procedure, which computes tap-coefficients of the filter. For each iteration of the input discrete-time data, AF must have to perform calculations for the filter coefficients, and the required number of multiplications and additions required for this task should be less, because more the number of multiplication and additions are there, more the number of external hardware equipments are required.

- **Stability**

Stability is a significant performance estimate for an adaptive algorithm. By the nature of automatic system, there are a few entirely asymptotically stable systems that may be practically realizable.

- **Robustness**

The robustness of an adaptive system corresponds to the stability of underlying system. Robustness is an indicator of how appropriately a system may converge to optimal solution in the presence of quantization errors.

- **Misadjustment**

Misadjustment describes the steady state response of an adaptive system. It is a quantitative estimate by which ensemble average mean squared value exceeds mean-squared-error (MSE) value obtained in an optimum case.

1.3 Applications of Adaptive Filters

The capability of adaptive filtering configuration to work efficiently in the unrevealed environment and track time changes of input data make an adaptive filter an influential tool for signal processing and automatic control requisitions. Though the applications of adaptive filtering configurations are relatively different in nature, but these have one common characteristic: “ input vector and desired response”; that are utilized to evaluate the estimation error. This estimation error is utilized to adjust values of a set of tunable tap-coefficient vector. But, fundamental difference between the different applications comes into picture in the way, the desired output is estimated. From that perspective, we can categorize four primary types of adaptive filter utilities, as depicted in Table 1.1. Subsequently, we briefly introduce a few applications like system identification, echo cancellation and adaptive beamforming.

Class of Adaptive filtering	**Applications**
Identification	System identification, layered earth modelling.
Inverse modelling	Predictor-type deconvolution, adaptive equalization.
Predictor	Linear predictive coding, signal detection.
Interference cancellation	Echo cancellation, adaptive beamforming.

Table 1.1. Application of adaptive filters.

1.3.1 System Identification

In discrete-time signal processing, [1] many systems are featured by using linear or non-linear higher-order difference equation. If an unidentified system is available, it is usually beneficial to search out an analogous paradigm for this system, that is, the coefficients of its difference equation. This is referred as system identification. The order of the difference equation paradigm of the unidentified system should be estimated first, by using a little a priori information of parameters of the system. Such discrete-time difference equation with tap-coefficients is an adaptive filter. The input is fed at the same time to unidentified system and adaptive filtering configuration. The difference between instantaneous output of an unknown system and adaptive filtering configuration is termed as error signal. If adaptive filtering models the unidentified system accurately, then such error signal is approximately nill. The tap-coefficient vector of an adaptive filtering configuration is computed by the adaptive procedure, which minimizes an objective function (or cost function) based on aforementioned error signal. It is noteworthy that it is traditional approach to indicate AF by using arrow through it. Two example of this cost function are given as follows, with n as sample index.

- $J_n = E\{e^2(n)\}$, The expected value of the MSE.
- $J_n = \sum_{i=0}^{n} e^2(i)$, The sum of instantaneous MSE.

An adaptive procedure evaluates the filter tap-coefficients and gives that information to discrete-time filter, which in turn filters the input discrete-time signal by using its tap-coefficients. This adaptive procedure typically consumes a typical amount of time before it attains minimum of a cost function. It is known as convergence time. A plot of instantaneous squared error $e^2(n)$ vs. the sample index n is generally decreasing with time. A particular plot is demonstrated in Fig. 1.2, which illustrates the converging algorithm. The adaptive procedure constantly updates the filter tap-coefficients, even after convergence. Thus, if the parameters of the unidentified system vary, filter tap-coefficients also vary in order to continuously track this unidentified system.

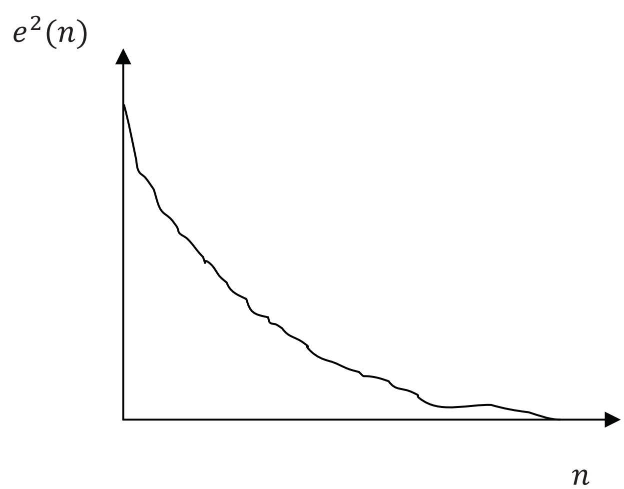

Fig. 1.2. Typical squared error plot.

Let $d(n)$ and $y(n)$ depicts the output signal of the unknown system and adaptive paradigm with $x(n)$ as their input signal respectively as depicted in Fig. 1.3. In this scenario, the motive of such AF is to appropriately demonstrate desired signal $d(n)$ at the output. If $y(n) = d(n)$, then adaptive filter is considered to be accurately modelled, and it is assumed that it has estimated the response of the unidentified system, which is governed by $x(n)$.

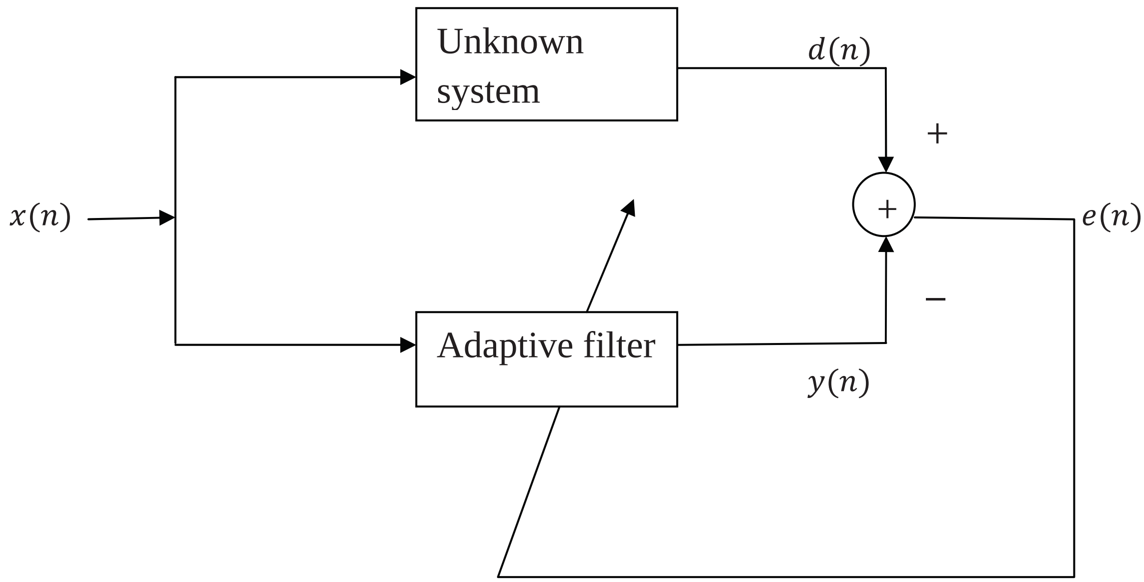

Fig. 1.3. Block diagram of system identification model.

As, a paradigm typically selected for an adaptive filtering configuration is a linear filter, the practical aim of such adaptive filtering configuration is to attain best linear paradigm that details the input-output relationship of an unidentified system. Such an algorithm makes most sense, when an unidentified system is also the linear paradigm of same configuration as an adaptive filter, as it is feasible that $y(n) = d(n)$ for a particular set of AF attributes. For the simplicity of discussion, let an unknown system and an adaptive filter both be finite-impulse response (FIR) filters, such that

$$y(n) = W_{opt}(n)X(n) \tag{1.1}$$

where, $W_{opt}(n)$ is an optimum set of filter tap-coefficients for the unidentified system at time n. In this problem formulation, an ideal adaptation algorithm would tune $W(n)$ in such a way that $W(n) = W_{opt}(n)$ as $n \rightarrow \infty$. In practical scenario, adaptive filter may only tune $W(n)$ in such a way that $y(n)$ closely approximates $d(n)$ with time.

The system identification is core of various applications of adaptive filter

- Plant identification
- Echo cancellation for long distance communication
- Adaptive noise cancelling

1.3.2 Inverse Modeling

In this application of adaptive filters, the main function of an adaptive filtering is to give inverse paradigm of unknown system. Inverse modeling is quite useful in channel equalization application. In case of linear systems, an inverse paradigm exhibits transfer function equal to the reciprocal of transfer function of an unknown system.

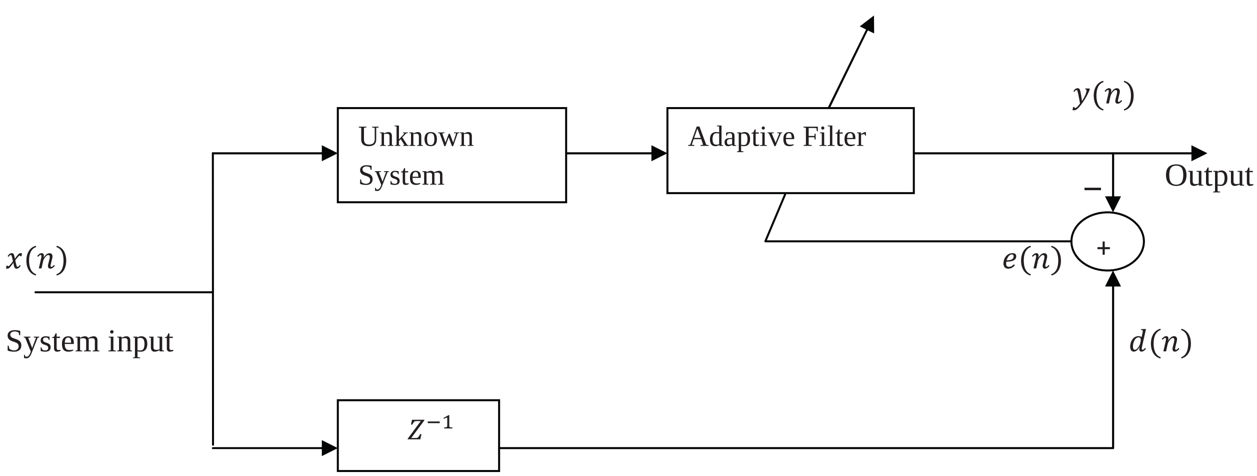

Fig. 1.4. Block diagram of inverse modeling.

1.3.3 Prediction

This application is utilized to predict a present or the future value of a random signal. In this application, a delay block is in series connection with AF block. This application of adaptive filter is used in various communication application like DPCM (differential-pulse- code-modulation), transversal filters, etc. The block diagram of prediction paradigm is depicted in Fig. 1.5.

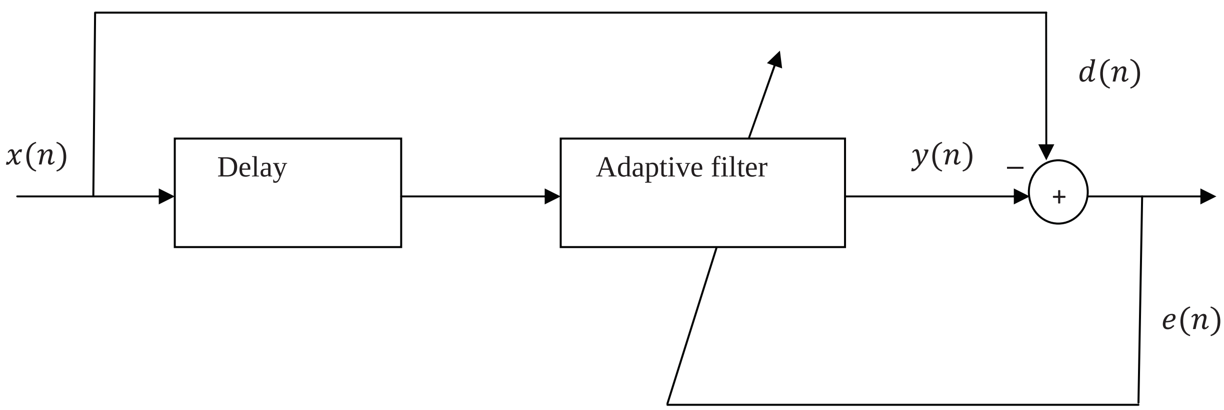

Fig. 1.5. Block diagram of prediction model.

1.3.4 Echo Cancellation

Echo signal is generally the delayed as well as distorted copy of true signal, which is bounced back to source. This class of application is used to separate the noise, echo or interference from the primary signal. The primary signal is served as a desired output of an adaptive filter. This application is very fascinating in many engineering applications. Echo problem arises basically in acoustic communication due to the reason, when both audio source as well as sink are operated in the duplex fashion. In today's telecommunication systems, users require very high quality speech quality, so this application of AF is also very popular nowadays.

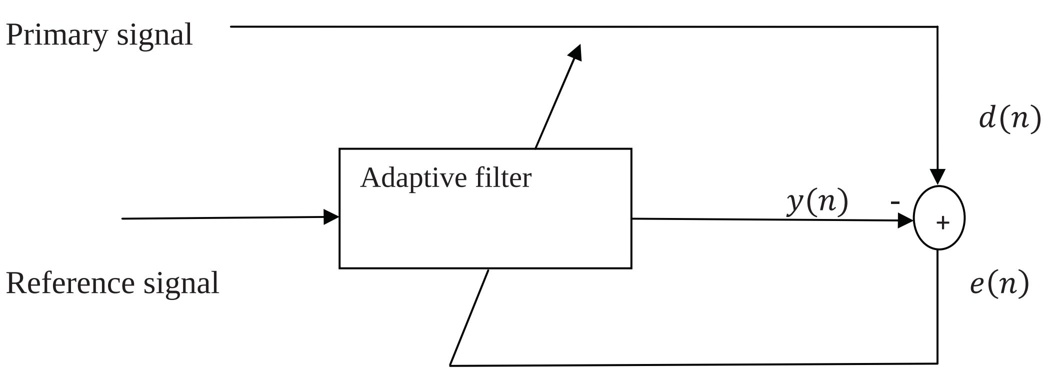

Fig. 1.6. Block diagram of echo cancellation model.

1.4 Non-Linear Filters

Linear filters [2] are used in wide range of applications and are relatively easy from conceptual and hardware implementation point of view. Linear filters are employed in the situations, where the system and input signal parameter are not changing with time, that means in a stationary environment, but practically, there are many situations in the real world, where the parameters of the signal and the system are varying with time, then the non-linear processing of the signal is required. Basically non-linear filters are those, which have non-linear relationship between input and output.

This section explains adaptive non-linear system equipped with polynomial model of non-linearity. The polynomial models are systems in which truncated Volterra series or recursive non-linear difference equation is utilized to express the relationship between input signal and output signal. A large class of non-linear systems or filters are modeled with the help of Volterra series because its expansion is a linear combination of the non-linear function of input signal. The Volterra systems are used in wide range of applications because they are capable to approximate a large number of non-linear systems with high frugality in the usage of their coefficients. The utilization of Volterra series in non-linear system identification and analysis has become extensive in last few decades. This is because of conception and mathematical tractability of the Volterra model. It is also mathematical tractable because the Volterra kernels of the non-linear systems can be attained by utilizing higher-order statistics of the input and output of underlying systems. Main task in Volterra modeling of non-linear systems is to obtain the Volterra kernels (in time-realm) or Volterra transfer function (in frequency-realm).

System using non-linear Volterra structure has several applications. High speed communication channels often use non-linear equalizer for acceptable performance, because in several applications, linear tap-delay line equalizer does not work at all. In telephone transmission, non-linearities arise due to error in signal companding. In satellite communication, non-linearities are due to the active devices and stray capacitance present in the amplifier at various stages during transmission. Other application of non-linear Volterra model and filters include echo cancellation, modeling various biological phenomenon, image processing, myoelectric signal processing, semiconductor characterization.

Linear systems are entirely specified by system's unit impulse response, but it is very difficult to detect the unit impulse response of the non-linear systems. So, we are restricted to use the non-linear filter models that are least general. Non-linear filters proposed by utilizing such paradigms include order statistic filters, homomorphic filters and the filters based on the Volterra series. Here, we are considering the two most important and generalized models like adaptive filter employing truncated Volterra series and adaptive filtering configuration utilizing non-linear recursive difference equations to relate input and output equations of underlying system.

1.4.1 Volterra Series

The Volterra series [2] was firstly showcased by Spanish mathematician Vito Volterra in his "Theory of Functionals", and first person who applied the Volterra series on non-linear application was Norbet Weiner. Volterra series found a great deal in finding the small non-linear distortion factors in the transistor amplifiers. A Volterra series paradigm is similar to that of Taylor series. It is different from Taylor series in its capability to capture 'memory' effects. The Taylor series approximates the non-linear system response in accordance with the given input, if the output signal at any instant is dependent on input signal of the system. In Volterra series, the non-linear system response depends upon the input at all other instants, and it produces the capability to capture 'memory' effect in Volterra system models. However, at present, there is no general technique for calculating the Volterra kernels for any non-linear systems, but these may be computed only for those systems, which exhibit finite as well as known order. The accuracy of Volterra kernels verify the efficiency and efficacy of underlying system paradigms.

1.4.2 Volterra Model

A causal linear-time-invarint (LTI) system can be expressed by the convolution operation as

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)\partial\tau \tag{1.2}$$

where, $x(t)$ is input signal, $y(t)$ is output signal to system; and $h(t)$ is impulse response of a system.

Whereas, any non-linear system can be expressed by Taylor series as

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

Here, $y(t)$ = output of the non-linear system.

$x(t)$ = input to system.

a_n = Taylor series coefficient.

Volterra series is the combination of the above two system representation and it may be represented as

$$y(n) = h_o + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + \cdots + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} \dots . \sum_{m_p=0}^{\infty} h_p(m_1,m_2,\dots m_p)x(n-m_1)x(n-m_2)\dots . x(n-m_p) \tag{1.4}$$

$y(n)$ = output of system.

$x(n)$ = input of system.

where, $h_p(m_1,m_2,\dots m_p)$ is called p^{th} order Volterra kernel of underlying system. Without lossing generality, we may consider that Volterra kernels are symmetrical. This means that $h_p(m_1,m_2,\dots m_p)$ remain unaltered for p! possible permutation of indices $m_1,m_2,\dots,m_p$.

1.4.3 Order of Volterra Systems

The order of the Volterra system can be obtained by putting the value of $p = 0,1,2,3,\dots$ in Eq. (1.4)

- **Zero-Order Volterra System**

The zeroth order Volterra paradigm is merely a constant and obtained by putting $p = 0$ in Eq. (1.4)

$$y_o(n) = h_o \tag{1.5}$$

where, $x(n)$ is the input, $y(n)$ is the output and h_o is considered to be static value.

- **First-Order Volterra System**

The first order Volterra system is obtained by putting $p = 1$ in Eq. (1.4)

$$Y(n) = h_o + \sum_{m_1=0}^{\infty} h_1(m_1)x(n - m_1) \tag{1.6}$$

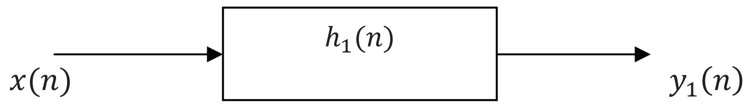

Fig. 1.4. Block diagram of first order linear system.

where, $y_1(n)$ is output of underlying system, and it is the convolution of $x(n)$ and $h_1(n)$

$$y_1(n) = x(n) * h_1(n) \tag{1.7}$$

The first-order Volterra system is same as the linear system, or in other words, we can say that linear systems are the subclass of Volterra systems. For the most general form of first-order Volterra system, we can write Eq. (1.6) in simplified form as

$$Y(n) = y_o(n) + y_1(n) \tag{1.8}$$

So, we can say that output of the first-order Volterra system is sum of a D.C term h_0 and the linear convolution of the input signal $x(n)$ and $h_1(n)$.

- **Second-Order Volterra System**

The second-order Volterra system generally can be obtained from Eq. (1.4) by putting the value $p = 2$.

$$\begin{aligned} y(n) = h_o + \sum_{m_1=0}^{\infty} h_1(m_1)x(n - m_1) \\ + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1, m_2)x(n - m_1)x(n - m_2) \end{aligned} \tag{1.9}$$

where, $h_2(m_1, m_2)$ is the second-order Volterra kernel of the system. From literature reviews, it has been found that it is symmetric, which implies that $h_2(m_1, m_2)$=$h_2(m_2, m_1)$.

1.5 Truncated Volterra Series Expansion for Non-Linear Systems

Consider a causal discrete time system with input $x(n)$ and output $y(n)$. The Volterra series expansion for the output $y(n)$ utilizing input $x(n)$ is written as

$$y(n) = h_o + \sum_{m_1=0}^{\infty} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)x(n-m_1)x(n-m_2) + \ldots + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\ldots\sum_{m_p}^{\infty} h_p(m_1,m_2,\ldots m_p)x(n-m_1)x(n-m_2)\ldots x(n-m_p) + \ldots \quad (1.10)$$

where, $h_p(m_1, m_2, \ldots m_p)$ is known as p^{th} order Volterra kernel of underlying systems. Without lossing generality, we may consider that Volterra kernels are symmetrical. This means that $h_p(m_1, m_2, \ldots m_p)$ remains unaltered for p! possible permutation of indices $m_1, m_2, \ldots, m_p$. Since, Volterra series expansion has been amicably incorporated in a vast diversity of utilities. Several researchers have been working on Volterra series expansion or other non-linear recursive difference equation for parameter estimation and non-linear system identification. Since, the infinite series in Eq. (1.10) is not desirable in filtering application. So, everyone is bounded to perform with truncated Volterra series expansion, as follows

$$y(n) = h_o + \sum_{m_1=0}^{N-1} h_1(m_1)x(n-m_1) + \sum_{m_1=0}^{N-1}\sum_{m_2=0}^{N-1}\ldots\sum_{m_p}^{N-1} h_p(m_1,m_2,\ldots m_p)x(n-m_1)x(n-m_2)x(n-m_p) + \cdots \quad (1.11)$$

1.6 Problem Statement

This thesis presents the following work

- First, the SOV filter is implemented using FSS-LMS algorithm, and the same filter implementation is then presented by using SVSS-LMS and MSVSS-LMS algorithms.
- Then, we have implemented the SOV filter using KVSS-LMS, AVSS-LMS and MAVSS-LMS algorithms. The performance of all these VSS algorithms are compared in terms of MSE.

1.7 Organization of Thesis

This thesis is organized as follows.

Chapter 2 describes literature review of different work related to adaptive polynomial filtering and system identification using different adaptive algorithms.

Chapter 3 presents the SOV-LMS algorithm, and after that the five different variable-step-size algorithms (SVSS-LMS, MSVSS-LMS, KVSS-LMS, AVSS-LMS, MAVSS-LMS) are discussed.

Chapter 4 presents the simulation results of SOV filter using FSS-LMS algorithm and compares the result with SVSS-LMS algorithm, MSVSS-LMS algorithm, KVSS-LMS algorithm, AVSS-LMS algorithm and MAVSS-LMS algorithms. Also the convergence behaviour of MSVSS-LMS algorithm and MAVSS-LMS algorithm is checked by varying their parameters.

Chapter 5 finally concludes this work by comparing the performance of different algorithms and also suggestions for future research are given.

CHAPTER 2

LITERATURE SURVEY

This chapter gives brief introduction about the work done on adaptive linear & non-linear filters and their behavior in the stationary and non-stationary environment.

V.J. Mathews [2] proposed the Volterra series as a system paradigm for mapping input-output relationship of a non-linear system. The Volterra series expansion may be used to model a large varieties of non-linear systems, and it is very useful and fascinating in adaptive filtering application because it's expansion is the linear combination of non-linear function of an input discrete-time signal. The polynomial models are those, whose input-output relationship is modelled as a truncated Volterra series and recursive nonlinear difference equations.

Y. Chen *et al*. [4] have proposed a non-linear relationship between SS (step-size) and error signal by using a modified-sigmoid function. They have proposed that SS may be varied at any stage of the stable state by functional transformation. This paper includes some tricky mathematical calculation that enhanced the steady state performance and stable misadjustment.

R. H. Kwong and E. W. Johnston [5] have proposed a new type of least mean square procedure, which uses the variable adaptation constant (SS) to diminish the swapping between misadjustment and tracking capability of LMS algorithm. In this algorithm, SS adjustment is performed by using the square of prediction error. The high prediction error causes SS to rise and yields fast convergence speed. As time progresses, the error reduces, that leads to smaller SS, and hence yields smaller misadjustment.

T. Aboulnasr and K. Mayyas [6] have presented a robust variable-step-size least mean square type procedure with fast convergence in the early stage and having a very low final misadjustment. This procedure is not affected by presence of uncorrelated noise. Simulation outcomes show that this presented procedure provides much better converge rate as compared to variable step-size in the stationary environment.

Z. Kun and Z. Xiubing [7] have proposed a modified robust variable step-size, which combines the advantages of both VSS-LMS and AVSS-LMS algorithms. This algorithm has good tracking and better noise immunity than other VSS algorithms. The theoretical, analytical and simulation results certify that an MAVSS procedure can attain the higher convergence speed and smaller MSE, and its anti-noise and tracking capability is superior to other VSS-LMS procedures.

N. Zhou and N. Holte [8] have proposed that the tracking properties of least squares estimation can be modified for a time-variant channel estimation and system identification problem, by incorporating a parametric time-variant paradigm instead of traditional paradigm, which is a constant impulse response at each iteration.

D. K. Borah and B. Hart [9] have proposed a rectangular-windowed recursive least square procedure in conjunction with a polynomial time-variant channel paradigm, which is presented for estimating a time-variant frequency-selective channel. This technique attains superior performance than the traditional FSS-LMS and RLS procedures. The performance of this procedure is dependent upon the selection of the window size and the polynomial paradigm order used, which in turn depends on fade rate and signal-to-noise ratio (SNR).

A.E. Nordsjo and L.H. Zetterberg [10] have presented the problem of identification and tracking of time-varying non-linear systems. Furthermore, the Wiener system that is consisting of a dynamical time-varying linear part followed by a fixed non-linearity and a Hammerstein system, in which the order of these two blocks is reversed, are studied. This proposed algorithm in paper depends upon certain second-order statistics that may be unknown in some typical scenario.

E. Roy *et al.* [11] have proposed that for specific type of nonlinearities, where high order nonlinearities are recursively produced, an adaptive recursive second-order Volterra filter has modified performance than conventional famous second-order and third-order polynomial filters. Furthermore, a recursive second-order polynomial filter may be updated utilizing higher convergence rate algorithm like recursive prediction error algorithms.

E. S. Olivas *et al.* [13] have developed an easy procedure to show that NLMS is modification of basic least mean square procedure, as it minimizes the MSE of an adaptive system. This demonstration is considered to be the case of simple Taylor series expansion, and they have also proposed that the NLMS algorithm, which uses adaptation constant proportional to the inverse of the input signal energy, which will provide fastest convergence.

T. Koh and E. J Power [16] have proposed some recent designs and results on the second order volterra filters. In this, a simple mean- squared- error (MSE) solution is derived by assuming that the input signal is Gaussian. They have also proposed an iterative factorization scheme to develop the subclass of polynomial filtering configurations, which may eradicate its filtering computational burdens for different applications.

E. Roy *et al.* [17] have proposed an adaptive non-linear filter based on Volterra series as well as on the IIR filter configuration. This filter is capable of identifying higher-order non-linearities where the non-linearities are found to correspond with each other in harmonical fashion.They have also shown that a recursive second-order Volterra filter gives much better performance for higher-order system identification. Simulation outcomes are also shown to demonstrate performance and computational burden of adaptive second-order infinite impulse response (IIR) Volterra filter and other second as well as third order Volterra filters in terms of MSE.

B. Widrow and S.D Stream [19] have detailed performance and characteristics of LMS filter. A discrete-time filter comprises of tapped delay line and tunable weights, whose system response is adjusted with the help of an adaptive algorithm.

M. M. Sandhi and D.A. Berkley [20] have described an automatic echo suppression method by adaptively synthesizing a copy of the echo path, by convolving the incoming signal with its replica, and then subtracting the resulting echo estimation by the echo path output.

L. L. Horowitz and K. D. Senne [21] have proposed that mean square convergence of a discrete real time LMS algorithm is slowed due to the generation of image frequency noise in the LMS loops. This paper describes a comprehensive analytical outcome in terms of mean-squared- error related to adaptation characteristics of LMS procedure.

O. Macchi and E. Eweda [22] have proposed the convergence of adaptive filter vector with the use of MSE gradient procedure with static SS. They consider the mean squared deviation between an optimal filter and a true filter weight vector during the steady state.

A. Feuer and E. Weinstein [23] have proposed the statistical analysis and study of a least mean square (LMS) procedure with uncorrelated Gaussian data. They have derived the exact analytical analysis of steady-state MSE. They have also derived the adequate and appropriate condition for the convergence of prescribed procedure with finite variance.

R. W. Harris *et al.* [25] have presented a new version of LMS adaptive filtering algorithm which implements the variable feedback adaptation constant for every value of tap-weight vector of a transversal adaptive filtering configuration. This technique is called variable-step-size (VSS) algorithm. It is an extension of the earlier ideas in stochastic approximation for varying SS (adaptation constant) in the Steepest Descent method. This VSS method is applied to the IIR filter, and the simulation results are presented for the application of VSS-FIR and VSS-IIR adaptive filters.

N. J. Bershad and L. Z. Qu [27] have presented a study of joint probability density function of a tap-weight vector in the LMS adaptation for Gaussian models. The exact expression of the characteristics function is derived for the weight vector at time $n+1$, conditioned on tap weight vector at n^{th} instant. The conditional characteristics function is expanded by Taylor series expansion, and averaged over the unknown weight density to obtain the first-order partial difference equation using the unconditioned characteristics function of the weight vectors.

D. I. Pazaitis and A. G. Constantinides [31] have proposed new variable step-size procedure. The time-variant SS pattern is tuned by using kurtosis of estimation error. Therefore, it reduces the performance deterioration arising because of the presence of the strong noise.

W. P Ang and B. F. Boroujeny [33] have presented thorough study about various LMS algorithms. A class of such algorithms, where the gradient is smoothed, after which, it attains superior tracking performance; and other class utilizes instantaneous gradient. The adaptation of LMS algorithm parameters is done according to multiplicative updating recursions (geometrical progressions), which results in superior tracking performance.

Y. Zhang *et al.* [35] have proposed new gradient-based variable-step-size least mean square procedure. The SS of presented procedure is proportional to squared norm of a smoothed gradient vector. It is the closest measure of an adaptive phenomenon. In comparison to other proposed algorithms, VSS-LMS gives better convergence properties.

J. E. Greenberg [37] has discussed a traditional LMS algorithm in an adaptive noise canceller, which may critically deteriorate performance of target signal in applications like speech processing, where short time trunk and junction-routing (TJR) exhibit large fluctuations. Moreover, the proposed method in his paper is a novel modified LMS procedure to overcome this problem. This modified method described here is not only useful in speech processing, but also applicable to any signal exhibiting large signal power.

P. Koukouplas and N. Kalouptsidis [38] have presented a technique for recognition of second order Volterra by utilizing general stationary inputs. They have used closed form expressions for determination of Volterra kernels using zero mean stationary Gaussian process as input. The approach utilized in this paper formulates the input-output cross-cumulant expressions; and further a derivation is presented using Fredholm integral equation. This equation is usually tackled by approximate linear equations. Exact closed-form expressions are determined for signals observed by filtering independent and identically distributed (IID) signals.

R. Bernardini [41] has proposed a special method for the fast convolution of the Volterra filtering. The main point, which makes the standard fast convolution unfitted to Volterra filtering configuration is that typical MD FFT procedures are not able to utilize symmetrical characterization of underlying signals entering the processing of Volterra filter. The scheme proposed in this paper is able to overcome such problem by utilizing special MD FFT. The main benefit of this scheme is that it can resorts to fast convolution, and it can be used for non-linearities of any order.

T. Ogunfunmi [42] used the Volterra and Weiner approaches for the field of non-linear adaptive system recognition. It includes latest research outcomes in the field of non-linear system recognition and also present less complex schemes to identify non-linear systems. Such techniques use adaptive filtering procedures that are famous for linear system identification. These are well applicable to non-linear systems that may be effectively modeled by using polynomial.

M. Sayadi *et al.* [43] have presented the study of steady state performance of least mean square adaptive second-order Volterra filtering configuration with the zeroth-order term for Gaussian input signals. In this paper, MSE is measured first; and afterward second-order volterra filter coefficients are adjusted accordingly.

C. Krall *et al.* [44] have proposed an equalization approach for the non-linear second-order Volterra filter. In this, firstly, a new linear equalizer is presented, that consists of a non-linear second-order Volterra filter type structure of the channel. Subsequently, a new second-order Volterra equalizer is developed by explicitly solving the minimum mean squared error (MMSE) problem for the cascade connection of two second-order Volterra systems. The performance of such equalizer is then compared with an equivalent adaptive equalizer configuration, which depicts that same performances may be attained with both schemes.

CHAPTER 3

FSS-LMS AND VSS-LMS ALGORITHMS

This chapter explains the LMS algorithm and various VSS-LMS algorithms that are used to vary the convergence rate and tracking ability of the SOV filter.

3.1 Introduction to FSS- LMS Algorithm

An LMS algorithm is simply the approximate version of Steepest Descent method. For the adaptive filtering problem with a mean square cost function, the Steepest Descent procedure is given as

$$w(n+1) = w(n) + \mu\left(p - Rw(n)\right) \quad (3.1)$$

.

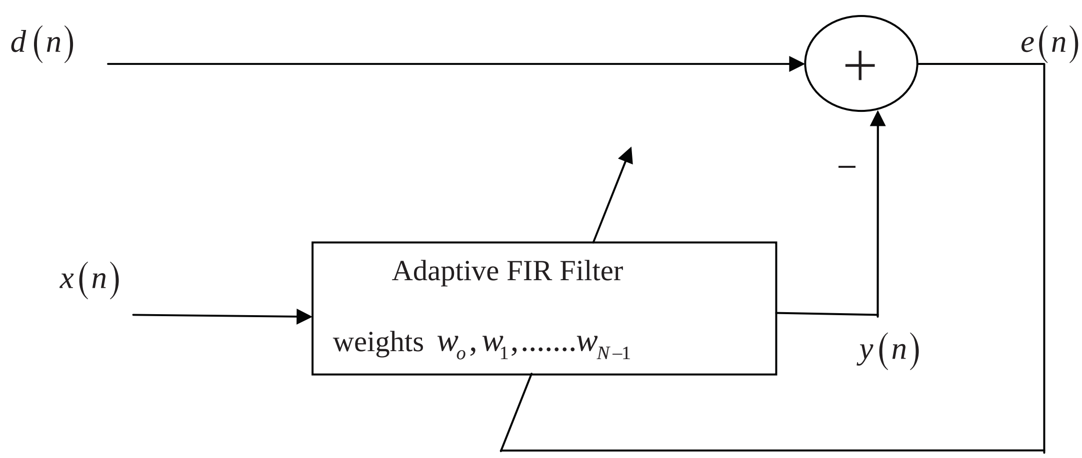

Fig. 3.1. Block diagram of adaptive FIR filtering.

For most real-time applications, the parameters R and p are not known, and therefore these must be estimated. The simplest way to do this is the following

$$\begin{aligned} R &= E\left[X(n)X^T(n)\right] \\ &= X(n)X^T(n) \end{aligned} \quad (3.2)$$

R is an autocorrelation matrix. From Fig. 3.1 $d(n)$ is an output of unidentified (desired) system, $X(n)$ is input signal, $w(n)$ is a set of filter coefficients.

$$p = E\left[d(n)X(n)\right] \\ = d(n)X(n). \tag{3.3}$$

Now,

$$w(n+1) = w(n) + \mu\left[d(n)X(n) - X(n)X^T(n)w(n)\right]. \tag{3.4}$$

$$= w(n) + \mu X(n)\left[d(n) - X^T(n)w(n)\right]. \tag{3.5}$$

Now, the terms inside square brackets on RHS is an error signal. Making this substitution, it is clear that

$$w(n+1) = w(n) + \mu e(n)X(n) \tag{3.6}$$

It is a tap-weight vector updating equation for an LMS procedure.

where, $w(n) = \{w_0, w_1, ..., \mathrm{w}_N\}$ are the filter coefficients.

$X(n) = \{X_0, X_1, ..., X_N\}$ are input vectors.

$e(n)$ is an error signal.

$$e(n) = d(n) - y(n) \tag{3.7}$$

For a filter with N taps, the update Eq. (3.6) requires $N+1$ multiplication and N additions. The filter equation $y(n) = X^T(n)w(n)$ requires N multiplications and $N-1$ additions. The computation of error signal in Eq. in (3.7) requires 2 N additions.

3.1.1 Convergence

The FSS- LMS procedure is a close match of a Steepest Descent method. Therefore, to prove its convergence, we have to made some independent assumptions, which are as follows

- The input vectors $X(1), X(2), ..., \mathrm{X}(N)$ are statistically independent of one other.
- The weight vector $w(n)$ is statistically independent of input vector $X(n)$.

- Input signal vector $X(n)$ is also statistically independent of the past (previous) discrete-time samples of the desired signal $d(n-1), d(n-1), d(n-3), \ldots$

In convergence analysis of an LMS algorithm we represent weight error vector as

$$\varepsilon(n) = w(n) - w_{opt} \tag{3.8}$$

where, w_{opt} is an optimal Weiner solution, $\varepsilon(n)$ is weight error vector, $w(n)$ is tap weight of an FIR adaptive filtering configuration.

Now subtracting w_{opt} from the both side of the LMS update Eq. (3.5), we get

$$w(n+1) - w_{opt} = w(n) - w_{opt} + \mu X(n)\left[d(n) - X^T(n)w(n)\right]. \tag{3.9}$$

Take expectation to the both side of the Eq. (3.9),

$$E\left\{\varepsilon(n+1)\right\} = E\left\{\varepsilon(n)\right\} + \mu E\left\{X(n)d(n)\right\} - \mu E\left\{X^T(n)X(n)w(n)\right\}. \tag{3.10}$$

Using above independent assumptions, we can write

$$E\left\{X^T(n)X(n)w(n)\right\} = E\left\{X^T(n)X(n)\right\}E\left\{w(n)\right\}. \tag{3.11}$$

From Eq. (3.10), it is clear that

$$E\left\{\varepsilon(n+1)\right\} = E\left\{\varepsilon(n)\right\} + \mu\left[P - RE\left\{w(n)\right\}\right] \tag{3.12}$$

From Weiner filter theory, $w_{opt} = R^{-1}P$ .Using this in Eq. (3.12), it is clear that

$$E\left\{\varepsilon(n+1)\right\} = E\left\{\varepsilon(n)\right\} + \mu\left[Rw_{opt} - RE\left\{w(n)\right\}\right] \tag{3.13}$$

or

$$E\left\{\varepsilon(n+1)\right\} = (I - \mu R)E\left\{\varepsilon(n)\right\} \tag{3.14}$$

Hence, it is apparent that $E\{\varepsilon(n)\} \to 0$, as $n \to \infty$ if and only if

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{3.15}$$

where, $\lambda_{\max}$ is the largest eigen value of R.

This implies that $E\{w(n) - w_{opt}\} \to 0$, which shows that $E\{w(n)\} \to w_{opt}$. This is called convergence in mean.

Practically, the eigen values of the correlation matrix are not known generally. Therefore, the value of μ cannot be derived using Eq. (3.15). So, for this problem, we use another formula. Let $D = Q^T RQ$, be the unitary transformation of matrix R.

where, $D = diag\{\lambda_1, \lambda_2, ...\lambda_N\}$, $\lambda_{\max} < tr\{D\}$ and $tr\{D\} = tr\{R\}$

In an LMS algorithm, the correlation matrix is approximated as $R \approx \left[X(n) X^T(n)\right]$.

$$tr\{R\} = tr\{X^T(n) X(n)\} \tag{3.16}$$

3.1.2 Misadjustment

Misadjustment in an adaptive procedure is the measure of its steady state error with respect to the Weiner solution. Mathematically, it can be expressed as

$$M \triangleq \frac{J_{ex}}{J_{\min}} \tag{3.17}$$

$$M \triangleq \frac{J_{ss} - J_{\min}}{J_{\min}} \tag{3.18}$$

where, J_{ex} is the excess MSE, $J_{\min}$ is the MSE of the optimum Weiner filter, and J_{ss} is the steady state MSE of the algorithm under consideration.

3.2 Volterra Series FSS- LMS Algorithm

The Volterra series paradigm is vastly utilized to model the non-linear systems for many different reasons. Particularly, this paradigm is very useful in non-linear adaptive filtering because traditional formulation of linear adaptive filtering configuration is simply extendable to such type of model. Here, Volterra series expansion related to input and output of a non-linear system can be expressed as

$$\begin{aligned} y(k) &= \sum_{l_1=0}^{\infty} w_{o1}(l_1)\,x(k-l_1) + \sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty} w_{o2}(l_1,l_2)\,x(k-l_1)x(k-l_2) + \ldots \\ &+ \sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}\ldots\sum_{l_i=0}^{\infty} w_{oi}(l_1,l_2,\ldots l_i)x(k-l_1)\,x(k-l_2)\ldots x(k-l_i)\ldots \end{aligned} \quad . \tag{3.19}$$

where, $w_{oi}(l_1,\ l_2,\ldots l_i)$ are tap-coefficients (Volterra kernels) of the non-linear systems, which are dependent on Volterra series expansion. As proposed by Mathews [2], Volterra series expansion may be considered as a Taylor series expansion with memory. Further, Volterra series is unsuitable to model the system, which contains discontinuities in their models. Another disadvantage of Volterra series is its computational burden, when a complete series is engaged. So by reducing this series, we can alleviate complexity at the cost of reduced accuracy of such series expansion.

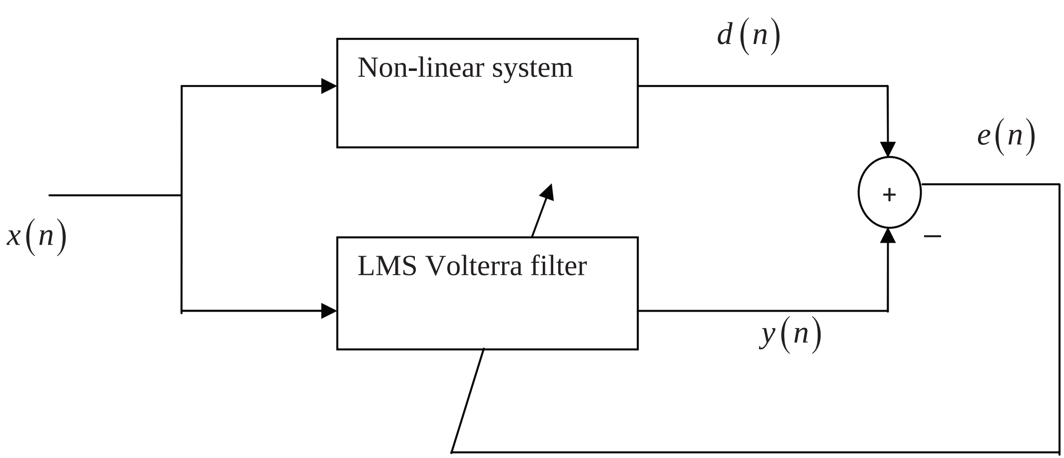

Fig. 3.2. Volterra kernel identification by adaptive LMS algorithm [3].

3.3 SOV-LMS Algorithm

In this subsection, Volterra least mean square procedure [3] is described for second-order Volterra series and N^{th}-order filter. A second-order Volterra filter (SOV) reduces complexity of an infinite Volterra series and also simplifies the derivation for some applications.

$$y(k)=\sum_{l_1=0}^{N} w_{l_1}(k)\,x(k-l_1)+\sum_{l_1=0}^{N}\sum_{l_2=0}^{N} w_{l_1,l_2}(k)\,x(k-l_1)x(k-l_2) \tag{3.20}$$

where, $w_{l_1}(k)$ and $w_{l_1,l_2}(k)$ for $l_1,l_2=0,1,2,....,\mathrm{N},$ are tap-coefficients of non-linear filtering model resting on truncated Volterra series expansion; and $y(k)$ indicates output signal of this AF. Now MSE is estimated by

$$F\left[e(k)\right]=E\left[e^2(k)\right]=E\left[d^2(k)-2d(k)\,y(k)+y^2(k)\right]. \tag{3.21}$$

and instantaneous error is represented as

$$e^2(k)=d^2(k)-2d(k)\,y(k)+y^2(k). \tag{3.22}$$

Here, input signal vector is represented as

$$X(k)=\begin{bmatrix} x(k) \\ x(k-1) \\ . \\ . \\ x(k-N) \\ x^2(k) \\ x(k)x(k-1) \\ . \\ . \\ x(k)x(k-N) \\ . \\ . \\ . \\ x(k-N)x(k-N+1) \\ x^2(k-N)) \end{bmatrix} \tag{3.23}$$

$$W(k)=\begin{bmatrix} w_o(k) \\ w_1(k) \\ . \\ . \\ w_N(k) \\ w_{o,o}(k) \\ w_{o,1}(k) \\ . \\ . \\ w_{o,N}(k) \\ . \\ . \\ w_{N,N-1}(k) \\ w_{N,N}(k) \end{bmatrix}. \tag{3.24}$$

From Fig. 3.3, output of adaptive filtering configuration is written as

$$y(k)=W^T(k)X(k). \tag{3.25}$$

where, estimate of mean squared error may be written as

$$e^2(k)=d^2(k)-2d(k)W^T(k)X(k)+W^T(k)X(k)W(k)X^T(k). \tag{3.26}$$

In LMS procedure, the main objective is to minimize a cost function.

$$W(k+1)=W(k)-\mu g_w(k). \tag{3.27}$$

$$W(k+1)=W(k)-2\mu e(k)\frac{\partial e(k)}{\partial w(k)}. \tag{3.28}$$

where, $g_w(k)$ is representing the estimate of a gradient vector of an objective function w.r.t. filter coefficients for $k=0,1,2..$.It is judicious to have various convergence factor for first-order and second-order terms of a least mean square polynomial filtering configuration. Here, corresponding updating equations can be expressed as

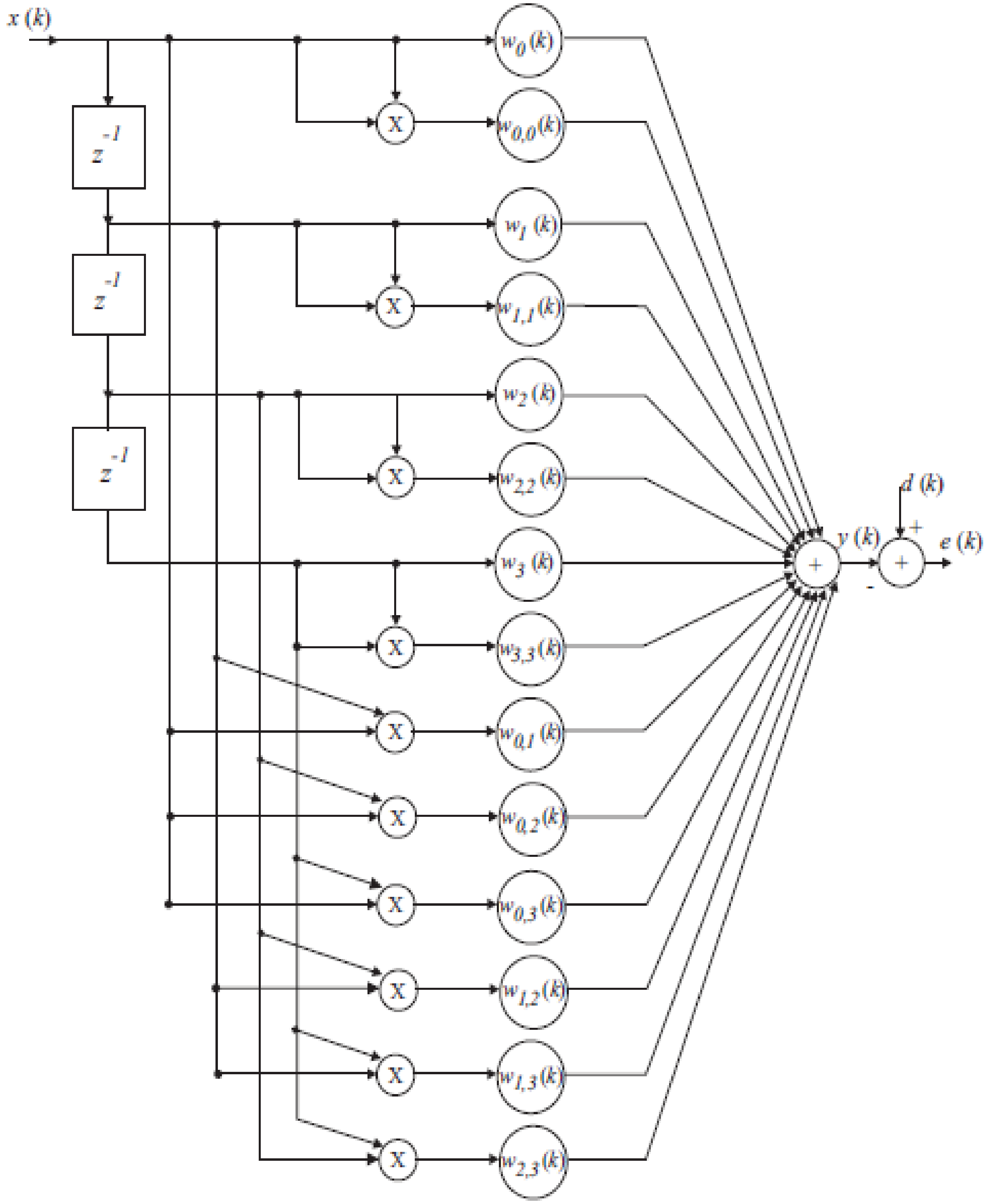

Fig. 3.3. General diagram of adaptive Volterra filter [3].

$$w_{l_1}(k+1) = w_{l_1}(k) + 2\mu_1 e(k)x(k-l_1). \tag{3.29}$$

$$w_{l_1,l_2}(k+1) = w_{l_2}(k) + 2\mu_2 e(k)x(k-l_1)x(k-l_2). \tag{3.30}$$

The Volterra LMS algorithm is identical to that of traditional LMS procedure except for the pattern of input signal $x(k)$. In an act to ensure the convergence of tap- coefficients, SS of Volterra LMS algorithm must be selected as

$$0 < \mu_1 < \frac{1}{tr(R)} < \frac{1}{\lambda_{\max}} \tag{3.31}$$

$$0 < \mu_2 < \frac{1}{tr(R)} < \frac{1}{\lambda_{\max}} \tag{3.32}$$

where, $R = E\left[X^T(k)X(k)\right]$ is an auto-correlation matrix of input signal vector. $\lambda_{\max}$ is highest eigen value of an input signal matrix *R*.

- μ controls how well and fast this procedure approaches to optimum filter tap-coefficients.
- If μ is very high, then this procedure shall not converge at all.
- If μ is high, but not much high, to prevent convergence, then this procedure will reach the steady state quickly, but consistently overshoots to an optimum weight vector.
- If μ is very small, then the algorithm will converge slowly, and it might not be able for us to track the changing conditions.
- In special cases, μ is kept high in the beginning for fast convergence, and then it is reduced to alleviate the overshoot problem.

3.4 Variable Step-Size LMS Algorithms

Usually conventional FSS-LMS algorithms [3] have contradiction among the tracking capability, convergence speed in the time-variant systems and steady state misadjustment of such systems. On the basis of convergence, bigger the value of μ, better is the convergence speed, but steady fluctuations shall rise at the same instant. On contrary, lower the convergence rate, lower is the steady state fluctuations. So, in this subsection, we discuss some of the variable step-size least mean square procedures for non-linear paradigms to achieve a balance between the above mentioned contradictions, so that LMS procedure can achieve superior convergence, stability and tracking capabilities. Various VSS-LMS algorithms are as follows.

3.4.1 SVSS-LMS Algorithm

To overcome the various contradictory statements regarding the value of SS in LMS procedure, author [4] have proposed the VSS-LMS algorithm, which makes a non-linear relationship between SS μ as well as error signal. In [4], authors have proposed about procedure based on sigmoid function, which may prevail over contradictory statements about the value of SS. However, disadvantage with this technique is that SS changes are very high in case of steady state, and it also affects error occurrence in this state.

Basic idea behind this algorithm is to make SS as large as possible at the initial stage of this procedure to obtain the rapid convergence rate, and SS must be as low as possible in the final stage of the algorithm to alleviate steady state error. In SVSS-LMS algorithm, value of SS is a non-linear function of error signal $e(n)$. It follows that

$$\mu(n) = \beta_{svss}\left(1 - \left(\exp\left(-\alpha_{svss}\left|e(n)e(n-1)\right|\right)\right)\right). \tag{3.33}$$

where, $\beta_{svss} = \left(0.99\beta_{svss}\right) + \left(0.01\left(\left|e(n)e(n-1)\right|\right)\right).$ (3.34)

3.4.2 MSVSS-LMS Algorithm

This is called modified-sigmoid-variable-step-size LMS (MSVSS-LMS) procedure. In this subsection, a new function, which is better than the sigmoid function, is presented. In MSVSS-LMS [4] algorithm, SS is much bigger than SS used in SVSS-LMS at initial stage of this procedure. Therefore, this procedure has fast convergence rate. At final stage of algorithm, that is the convergence steady state, modified SS is lower than SVSS-LMS step-size, which leads to a little steady state error.

Here, variable step-size LMS procedure according to MSVSS-LMS algorithm may be represented as

$$e(n) = d(n) - X^T(n)w(n). \tag{3.35}$$

$$\mu(n) = \beta_{msvss}\left|\frac{1}{1+\exp\left(-\alpha_{msvss}\left(e(n)-\gamma_{msvss}\right)\right)} - \frac{1}{1+\exp\left(+\alpha_{msvss}\left(e(n)+\gamma_{msvss}\right)\right)}\right| \tag{3.36}$$

$$w(n+1) = w(n) + 2\mu(n)e(n)X(n). \tag{3.37}$$

The range of values of $\mu(n)$ in conventional FSS-LMS procedure is from $1 < \mu < \frac{1}{\lambda_{\max}}$, where $\lambda_{\max}$ is highest eigen value of an input signal auto-correlation matrix. Therefore, condition of convergence in this procedure is $1 < \mu(n) < \frac{1}{\lambda_{\max}}$. Since, the value of $\mu(n)$ is lower than β_{msvss}, therefore , the maximum value of $\beta_{\max} = \frac{1}{\lambda_{\max}}$; and among these conditions, the algorithm is always convergent.

The relation between MSVSS and the error is given as

$$\mu(n) = \beta_{msvss}\left|\frac{1}{1+\exp\left(-\alpha_{msvss}\left(e(n)-\gamma_{msvss}\right)\right)} - \frac{1}{1+\exp\left(\alpha_{msvss}\left(e(n)+\gamma_{msvss}\right)\right)}\right| \tag{3.38}$$

where, β_{msvss} is a constant and controls the range of values of the function. Here, α_{msvss} and γ_{msvss} are also constants, which control the shape of the function.

In Fig.3.4, the simulation parameters used are $\beta_{msvss} = 0.2$, $\alpha_{msvss} = 10$, and $\gamma_{msvss} = 0.1, 0.5, 0.7$ [4]. We can see that when γ_{msvss} is increasing, SS becomes very slow. In a nutshell, we can say that anybody may regulate steady state error by utilizing different values of γ_{msvss}.

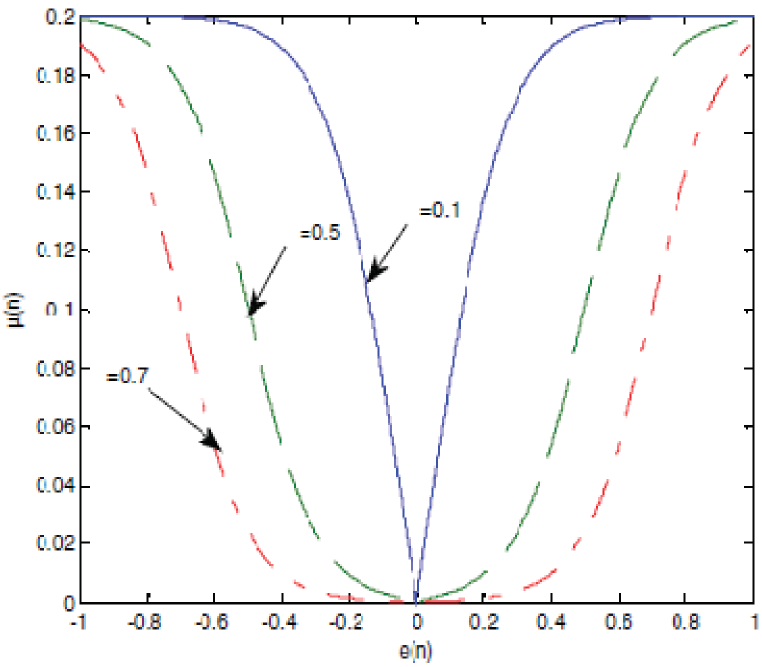

Fig. 3.4. Relationship between SS and error signal [4].

3.4.3 KVSS-LMS Algorithm

In this algorithm [5], author proposed a variable-step-size in the LMS procedure, in which SS is tuned with the help of square of predictor error. Basic concept behind this is that a high level predictor error shall cause SS to boost up and gives fast tracking, while a low value of predictor error shall cause reduction in SS to provide low misadjustment. A detailed description of this algorithm is given as

Let input vector to the system be $X(k)$ and the desired scalar output will be $d(k)$, and these processes are related by an equation as

$$d(k) = X^{T}(k)W^{*}(k) + e(k) \tag{3.39}$$

An LMS type procedure is the gradient search procedure, which calculates the set of weight vector $W(k)$ by the minimization the mean squared error. This algorithm is written as

$$W(k+1) = W(k) + \mu(k)X(k)e(k) \tag{3.40}$$

where, $\mu(k)$ is SS. In standard LMS procedure, SS is kept constant. Here according to KVSS-LMS algorithm, SS is varied and represented as follows

$$\mu_{kvss}(k+1) = \alpha_{kvss}\mu_{kvss}(k) + \gamma_{kvss}e^2(k) \tag{3.41}$$

with $0 < \alpha_{kvss} < 1$, $\gamma_{kvss} > 0$.

$$\mu_{kvss}(k+1) = \begin{cases} \mu_{\max} \; ; & if\ \mu_{kvss}(k+1) > \mu_{\max} \\ \mu_{\min} \; ; & if\ \mu_{kvss}(k+1) < \mu_{\min} \\ \mu_{kvss}(k+1) \; ; & otherwise \end{cases} \tag{3.42}$$

where, $0 < \mu_{\min} < \mu_{\max}$. The initial SS $\mu(0)$ is considered to be $\mu_{\max}$. The step-size $\mu_{kvss}(k)$ is always positive and its value is always adjusted by the size of predictor error and the tracking mode conditions. If the predictor error is small, then SS shall be reduced due to parameters α_{kvss} and γ_{kvss}. If predictor error is high, it will increase SS and give higher convergence rate and alleviate misadjustment. Therefore, constant $\mu_{\max}$ is selected such that mean squared error of this procedure must be kept bounded. A necessary and appropriate condition for $\mu_{\max}$ to ensure bounded mean squared error is

$$\mu_{\max} \le \frac{2}{3tr(R)} \tag{3.43}$$

$\mu_{\min}$ is chosen, such that , it will provide minimum level of tracking capability. Commonly, the value of $\mu_{\min}$ is chosen closest to the value of adaptation constant (SS) as in fixed-step-size LMS algorithm (FSS-LMS).The parameter α_{kvss} will be chosen in the range of zero to one to yield exponential forgetting. The particular value of α_{kvss} that will perform sufficiently better in the simulation is 0.97. Here, value of γ_{kvss} is commonly chosen very low i.e., typically 0.00048, and use in conjunction with α_{kvss} to meet misadjustment requirements.

3.4.4 AVSS-LMS [6] Algorithm

This presents a robust variable step-size least mean square type procedure. This procedure has a fascinating property of providing low value of final misadjustment, while giving rapid convergence at the initial stages of adaptation. In this procedure, approximate analysis about convergence rate and steady state working in presence of zero mean stationary Gaussian input is provided. The main significant feature of this algorithm is the addition of one parameter pertaining to time averaging operation, which permits the misadjustment control and the convergence time more independently without any inherent need to compromise between them as in other VSS algorithms.

The working of this procedure is not influenced by the presence of noise. In other algorithms like KVSS-LMS [5], due to $E\left[e^2(n)\right]$, SS updating degrades the performance at every stage of adaptation, but typically at the optimum instant. To preclude such sensitivity towards noise, AVSS-LMS algorithm is presented. The new idea behind this algorithm rests upon the fact that at optimum instants not only the error in energy is low, but also the correlation between the successive samples is low. Therefore, authors utilize auto-correlation between $e(n)$ and $e(n-1)$ to control SS updating. Now, they estimate time average of $e(n)e(n-1)$, which is represented as

$$p(n) = \beta_{avss}\, p(n-1) + (1-\beta_{avss})\, e(n)e(n-1) \tag{3.44}$$

and SS updating equation is

$$\mu_{avss}(n+1) = \alpha_{avss}\mu(n) + \gamma_{avss}\, p^2(n) \tag{3.45}$$

where, $0<\alpha_{avss}<1, \gamma_{avss}>0$ and the positive constant $0<\beta_{avss}<1$ is an exponential weighting parameter, which derives control of averaging time constant. Such auto-correlation of $e(n)$ and $e(n-1)$ basically acts for two purposes viz.

- It rejects the dependent noise components effects on SS.
- It is the efficient measure of approximation to optimum.

At steady state, $p(n) \approx 0$ causes $\mu \approx 0$, and hence achieving smaller misadjustment values.

When this adaptive processing is active and when $p(n)$ is high, it leads to high μ and hence provides fast convergence rate. Now, SS updating equation is formulated as

$$\mu_{avss}(n+1) = \alpha_{avss}\mu(n) + \gamma_{avss}E^2\{e(n)e(n-1)\} \tag{3.46}$$

Here, we consider the accurate estimation of auto-correlation of $e(n)e(n-1)$. The auto-correlation is estimated in an recursive manner, which requires only one multiplication per iteration. Simulation results show that this algorithm shows substantial performance advantage in terms of convergence speed than other VSS-LMS procedures in the same non-stationary environment for the same excess MSE. It is a straight condition for accurate convergence that

$0 < E[\mu(n)] < \frac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is largest eigen value of an input auto-correlation matrix R.

The appropriate condition [6] for efficient convergence of the mean squared error is given as

$$\frac{E\{\mu^2(\infty)\}}{E\{\mu(\infty)\}} \leq \frac{2}{3tr(R)} \tag{3.47}$$

A larger value of γ_{avss} leads to high SS in beginning states of the adaptation, and it provides fast convergence speed. However, parameter β_{avss} and γ_{avss} will give an extra degree of freedom to the algorithm that facilitates both the convergence rate and final excess mean squared error.

3.4.5 MAVSS-LMS [7] Algorithm

This algorithm is modified version of AVSS-LMS algorithm. It is a combination of KVSS-LMS and AVSS-LMS algorithm, and hence known as modified robust variable step-size LMS procedure. It has good tracking capability and better anti-noise capability. For system recognition purpose, the reference signal may be represented as

$$d(n) = W^T(n)X(n) + v(n) \tag{3.48}$$

where, $W(n)$ are adaptive filter tap-coefficients; and $v(n)$ is AWGN noise exhibiting zero mean. Here, σ_v^2 is its variance. The noise is assumed to be independent to the input sequence $X(n)$. The SS of MAVSS-LMS procedure is represented as

$$\mu_{mavss} = \begin{cases} \mu_{\max}\ ; & if\ \ \mu(n+1) > \mu_{\max} \\ \mu_{\min}\ ; & if\ \ \mu(n+1) < \mu_{\min} \\ \alpha_{mavss}\mu_{mavss}(n) + \gamma_{mavss}\, p^2(n) \end{cases} \tag{3.49}$$

$$p(n+1) = \left(1 - \beta_{mavss}(n)\right) p(n) + \beta_{mavss}(n) e(n) e(n-1) \tag{3.50}$$

$$\beta_{mavss}(n+1) = \begin{cases} \beta_{\max}; & \text{if } \beta_{mavss}(n+1) > \beta_{\max} \\ \beta_{\min}; & if\ \beta_{mavss}(n+1) < \beta_{\min} \\ \eta_{mavss}\beta_{mavss}(n) + \lambda^2{}_{mavss} e^2(n) \end{cases} \tag{3.51}$$

where, the parameters assumed are $\alpha_{mavss} > 0,\ \eta_{mavss} < 1,\ \gamma_{mavss},\ \lambda_{mavss} > 0$. The $p(n)$ is time average of error signal auto-correlation at instant n and $n+1$. $\beta_{mavss}(n)$ is time average of squared error signal, which helps in controlling the sensitivity of $p(n)$ to instantaneous error correlation. The largest bound of SS $\mu_{\max}$ satisfies mean squared stability requirement. The lowest bound of SS $\mu_{\min}$ is utilized to assure that excess mean squared error remains under desired tolerance threshold. Here, parameter β_{mavss} must be larger than zero and less than one. In MAVSS-LMS algorithm, SS is controlled by the correlation of the error signal at instants n and $n+1$; and the correlation of the error signal is statistically independent of AWGN. Hence, an MAVSS algorithm exhibits better noise tolerance capability. Its sensitivity towards error correlation function with respect to instantaneous correlation is adjusted with the help of instantaneous signal power. It means that when an algorithm is convergent, power of instantaneous error is too low and error signal correlation is insensitive to the instantaneous error, then efficiency and approximation of error signal correlation gets boosted.

CHAPTER 4

SIMULATION RESULTS AND DISCUSSION

This chapter discusses the simulation results, performance and comparisons between various VSS-LMS algorithms that we implemented in SOV filter using MATLAB.

Simulation results shown below are analyzed and verified using Matlab. This chapter is arranged as follows:

Section 4.1 shows MSE for a second order Volterra filtering configuration utilizing FSS-LMS procedure.

Section 4.2 shows the MSE for a second order Volterra filtering configuration utilizing SVSS-LMS procedure.

Section 4.3 shows the comparisons of MSE for a second order Volterra filtering configuration at different parameters of SVSS-LMS procedure.

Section 4.4 shows the MSE for a second order Volterra filtering configuration utilizing MSVSS-LMS procedure.

Section 4.5 shows the comparison of MSE for a second order Volterra filtering configuration by varying parameters of an MSVSS-LMS algorithm.

Section 4.6 shows the comparison of MSE for a second order Volterra filtering configuration for FSS-LMS, SVSS-LMS and MSVSS-LMS algorithm.

Section 4.7 shows the MSE for second order Volterra filtering configuration utilizing KVSS-LMS procedure.

Section 4.8 shows the MSE for second order Volterra filtering configuration utilizing AVSS-LMS procedure.

Section 4.9 shows the MSE for a second order Volterra filtering configuration utilizing MAVSS-LMS procedure.

Section 4.10 shows the comparison of MSE for second order Volterra filtering configuration utilizing KVSS-LMS, AVSS-LMS and MAVSS-LMS algorithms.

In the simulation results discussed and mentioned next, the system considered is a second-order Volterra filtering configuration. It is basically a truncation of a Volterra series expansion for $P = 2$. P is the order of Volterra filter or p^{th} order Volterra kernel of the system. Second-order Volterra systems consist of both linear and quadratic kernels. The Volterra kernels assumed here are considered to be symmetric. The Volterra filter, described here, is used as an adaptive non-linear filter in system identification's application mode.

$d(k)$ is considered as a desired response signal. $x(k)$ is an input signal to an adaptive Volterra filtering configuration. The objective is to choose Volterra kernels (tap-coefficients) of an adaptive Volterra filtering structure, for the minimization of a convex function of error signal. Here, we considered the length of adaptive filter L=9. All the simulations are performed under the static environment. Presented results are based on the ensemble average of 1000 statistically independent trials.

$$d(k) = -0.76 * x(k) - x(k-1) + x(k-2) + 0.5 * x(k) + 2 * x(k) * x(k-2) - 1.6 * x^2(k-1) + 1.2 * x^2(k-2) + 0.8 * x(k-1) * x(k-2) + n(k) \tag{4.1}$$

$n(k)$ = random noise signal.

4.1 Performance of SOV Filter for FSS-LMS Algorithm

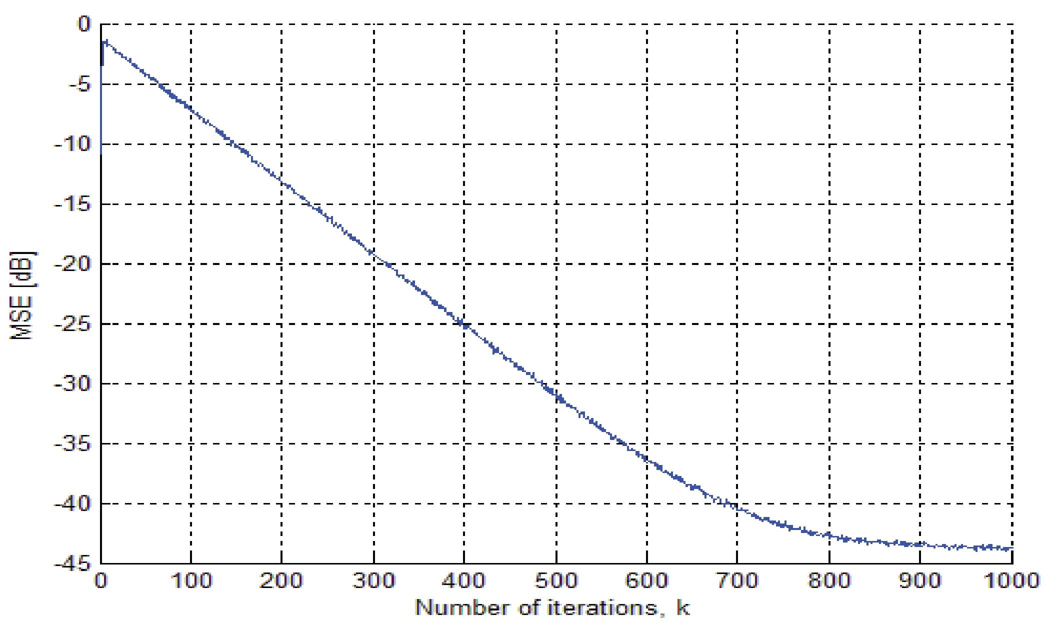

Fig. 4.1. MSE performance of SOV filter using FSS-LMS algorithm.

In Fig. 4.1, simulation results are based on the implementation of second-order Volterra filter using FSS-LMS procedure. The result is presented for parameter estimation in terms of MSE, and the system converges at approximately 700^{th} iteration; and the corresponding MSE is found to be approximately equal to $-41dB.$

4.2 Performance of SOV Filter Using SVSS-LMS Algorithm

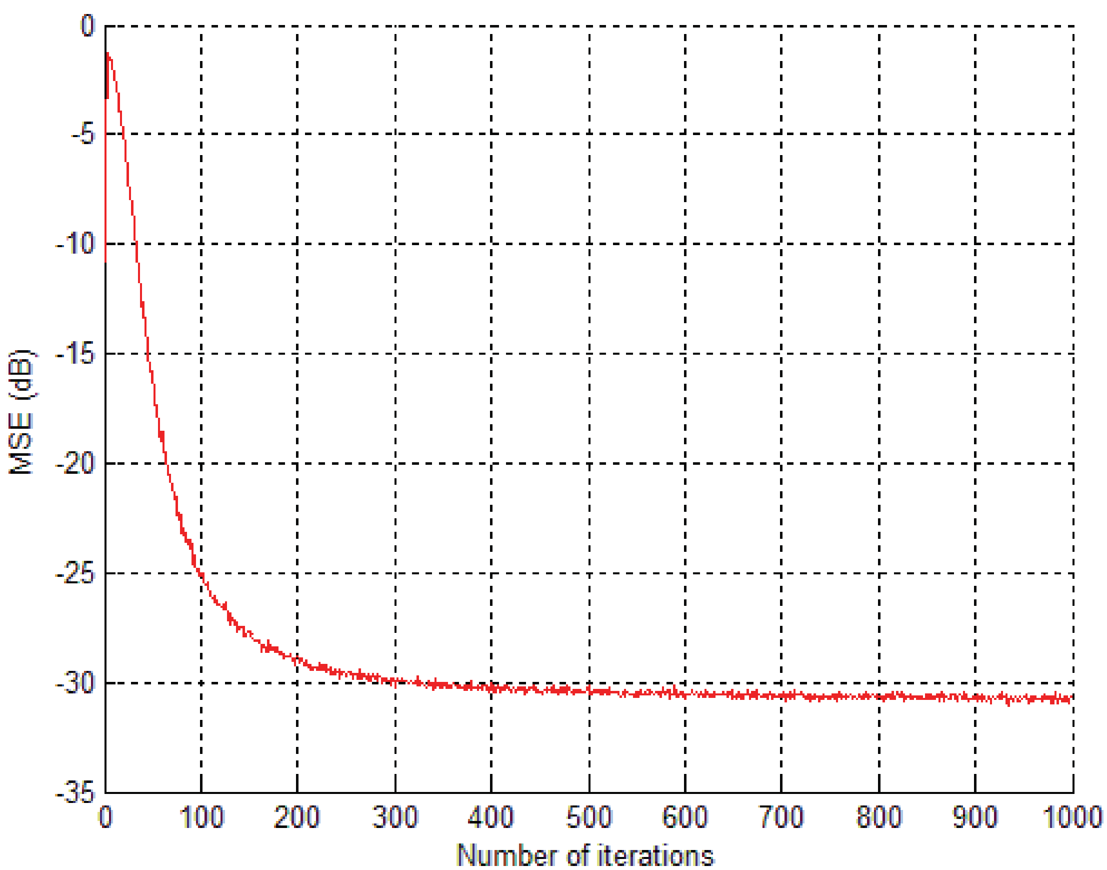

Fig. 4.2. MSE of SOV filter using SVSS-LMS algorithm.

In Fig. 4.2 , simulation results are based on the implementation of SOV filter using SVSS-LMS procedure. The result is presented for parameter estimation in terms of MSE, and the system converges at approximately 200^{th} iteration. In this algorithm, we have chosen the value of $\alpha_{svss} = 10$ and initial value of $\beta_{svss} = 0.2$. This algorithm provides better convergence rate than FSS-LMS algorithm. The corresponding MSE is found to be approximately $-31dB.$ The SNR is kept constant at $10\,dB.$

4.3 Performance Comparison of MSE for the SOV Filter at Different Parameters of SVSS-LMS Algorithm.

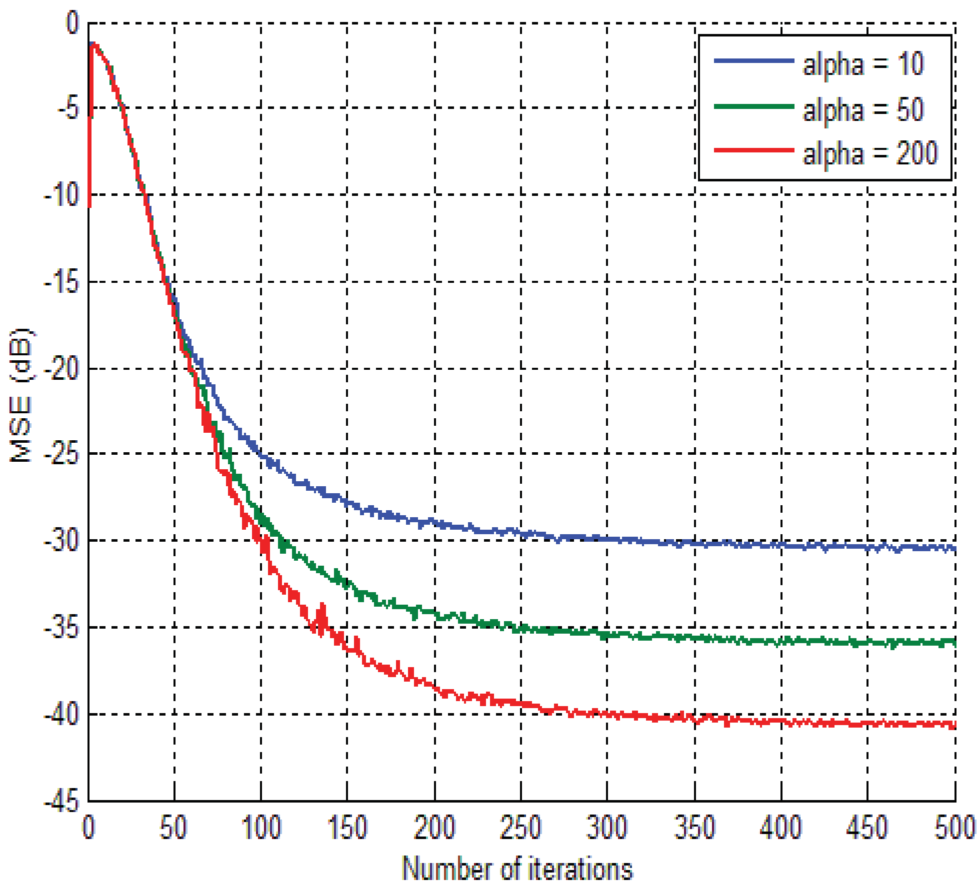

Fig. 4.3. MSE comparison of SOV filter using SVSS-LMS algorithm.

In Fig. 4.3, simulation results are based on the implementation of second-order Volterra filter using different parameters of SVSS-LMS algorithm. The results are presented for parameter estimation in terms of MSE. In these simulation results, we have taken three different values of α_{svss}. It may be referred that as the value of α_{svss} increases, the convergence of the algorithm also increases. For this simulation, the value of SNR is kept constant at $10\ dB$ with $\beta_{svss} = 0.2$.

4.4 Performance of SOV Filter Using MSVSS-LMS Algorithm

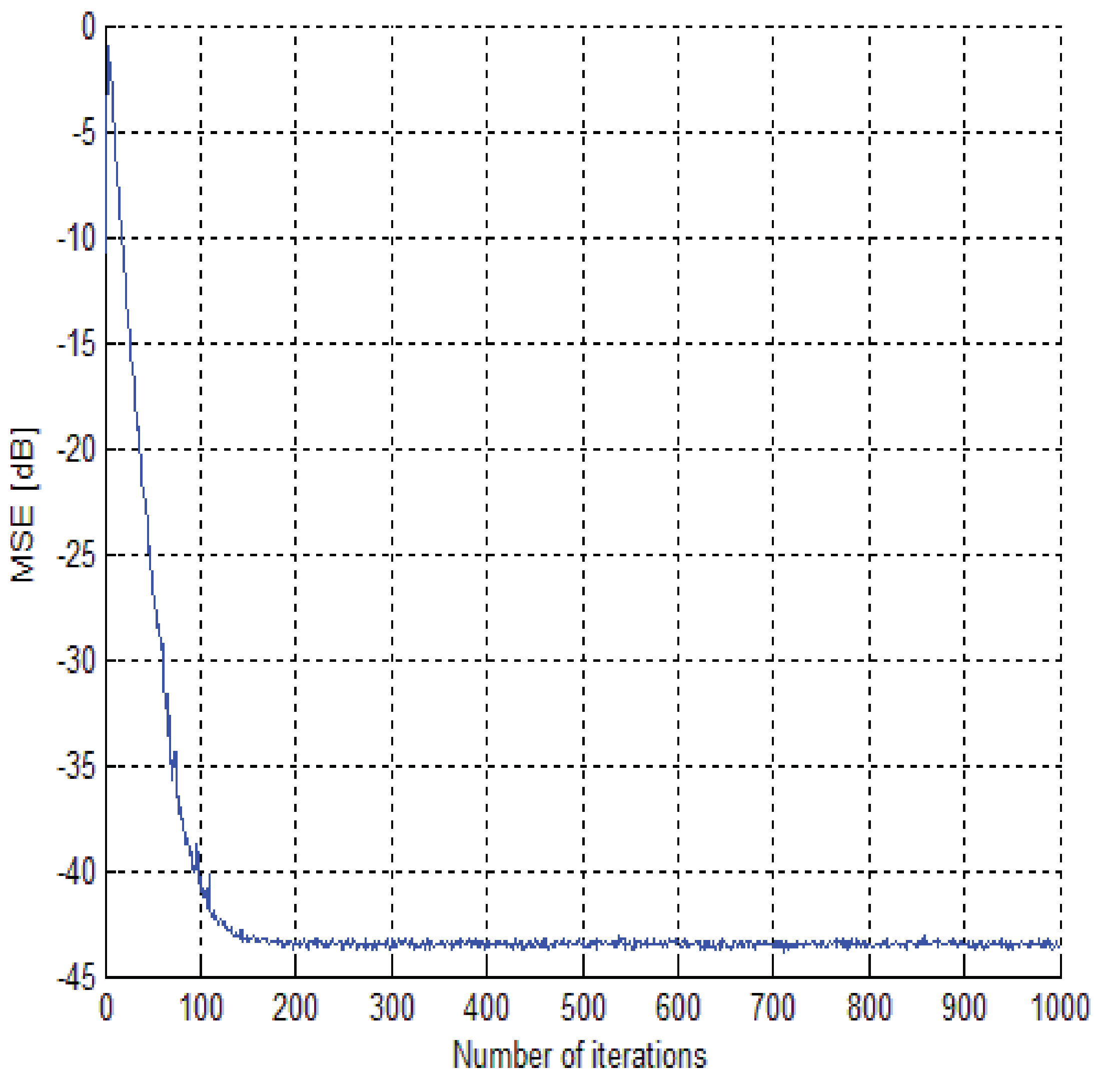

Fig. 4.4. MSE of SOV filter using MSVSS-LMS algorithm.

In Fig. 4.4, simulation results are based on the implementation of SOV filter using MSVSS-LMS procedure. The result is presented for parameter estimation in terms of MSE. This algorithm provides better convergence rate than SVSS-LMS algorithm. The values of $\alpha_{\text{msvss}} = 10$, $\beta_{msvss} = 0.2$ and $\gamma_{msvss} = 0.4$. The corresponding MSE is found to be approximately $-43\,dB$. The value of SNR is kept constant at $20\,dB$.

4.5 Performance Comparison of MSE for the SOV Filter at Different Parameters of MSVSS-LMS Algorithm

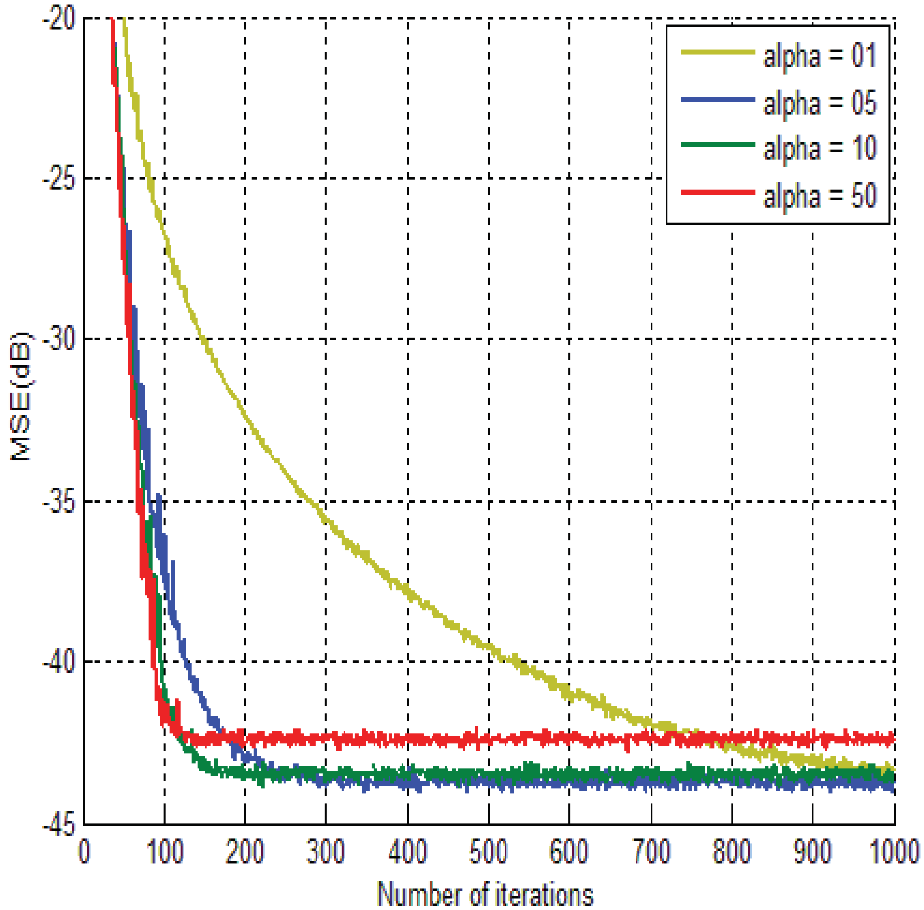

Fig. 4.5. MSE comparison of SOV filter using MSVSS-LMS algorithm.

From section 4.5, simulation results are based on the implementation of second-order Volterra filter by varying parameters of MSVSS-LMS algorithm. The results are presented for parameter estimation in terms of MSE. It is apparent that by assuming $\beta_{msvss} = 0.2$ and $\gamma_{msvss} = 0.4$ and by varying the values of α_{msvss}, the convergence rate can be significantly improved. But $\alpha_{\text{msvss}} = 10$ is optimum value, and hence, after increasing the value of alpha further, we can cause convergence precision of the algorithm to be decreased. For this simulation purpose, the value of SNR is kept constant at $20\ dB$.

4.6 Performance of SOV Filter Using FSS-LMS, SVSS-LMS, and MSVSS-LMS Algorithm

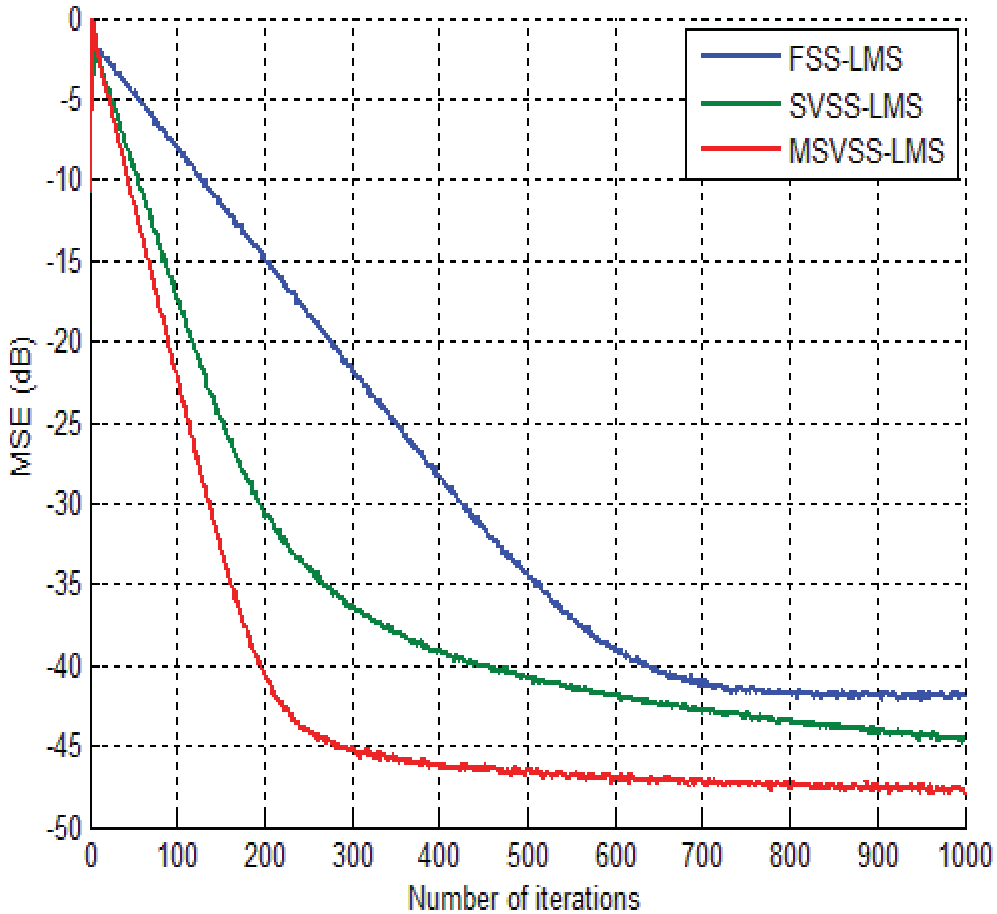

Fig. 4.6. MSE comparison of SOV filter using FSS-LMS, SVSS-LMS and MSVSS-LMS algorithm.

In Fig. 4.6, simulation results are compared based on the implementation of second-order Volterra filter using FSS-LMS, SVSS-LMS and MSVSS-LMS algorithm. The results are presented for parameter estimation in terms of MSE (dB). And we have seen that MSVSS-LMS algorithm attains faster convergence rate than SVSS-LMS and FSS-LMS algorithm. The corresponding average value of MSE in tracking mode approximately is found to be $-42\ dB$ for FSS-LMS, $-44.5\ dB$ for SVSS-LMS and $-47.5\ dB$ approximately for MSVSS-LMS algorithms. The values of parameters are kept at $\alpha_{svss} = 3$, $\alpha_{msvss} = 3$ and $\beta_{svss} = 0.2$, $\beta_{msvss} = 0.2$. and $\gamma_{msvss} = 0.4$. For this simulation purpose, the value of SNR is kept constant at 25 dB.

4.7 Performance of SOV Filter Using KVSS-LMS Algorithm

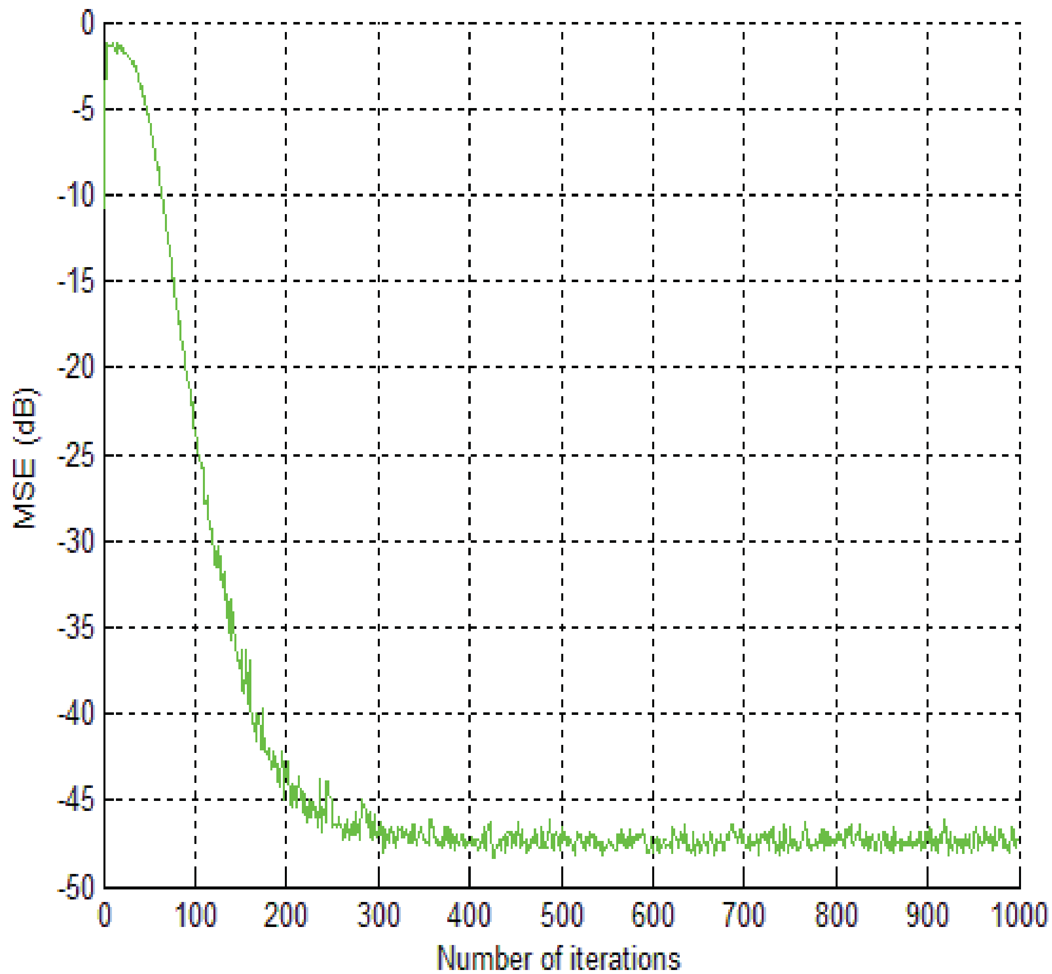

Fig. 4.7. MSE of SOV filter using KVSS-LMS algorithm.

In Fig. 4.7, simulation results are based on the implementation of SOV filter by using KVSS-LMS procedure. The result is presented for parameter estimation in terms of MSE. In this algorithm, we have chosen the value of $\alpha_{kvss} = 0.97$ and $\gamma_{kvss} = 0.00048$. The corresponding average MSE is found to be approximately $-47\,dB$ in tracking mode. For this simulation purpose, the value of SNR is kept constant at $25\,dB$.

4.8 Performance of SOV Filter Using AVSS-LMS Algorithm

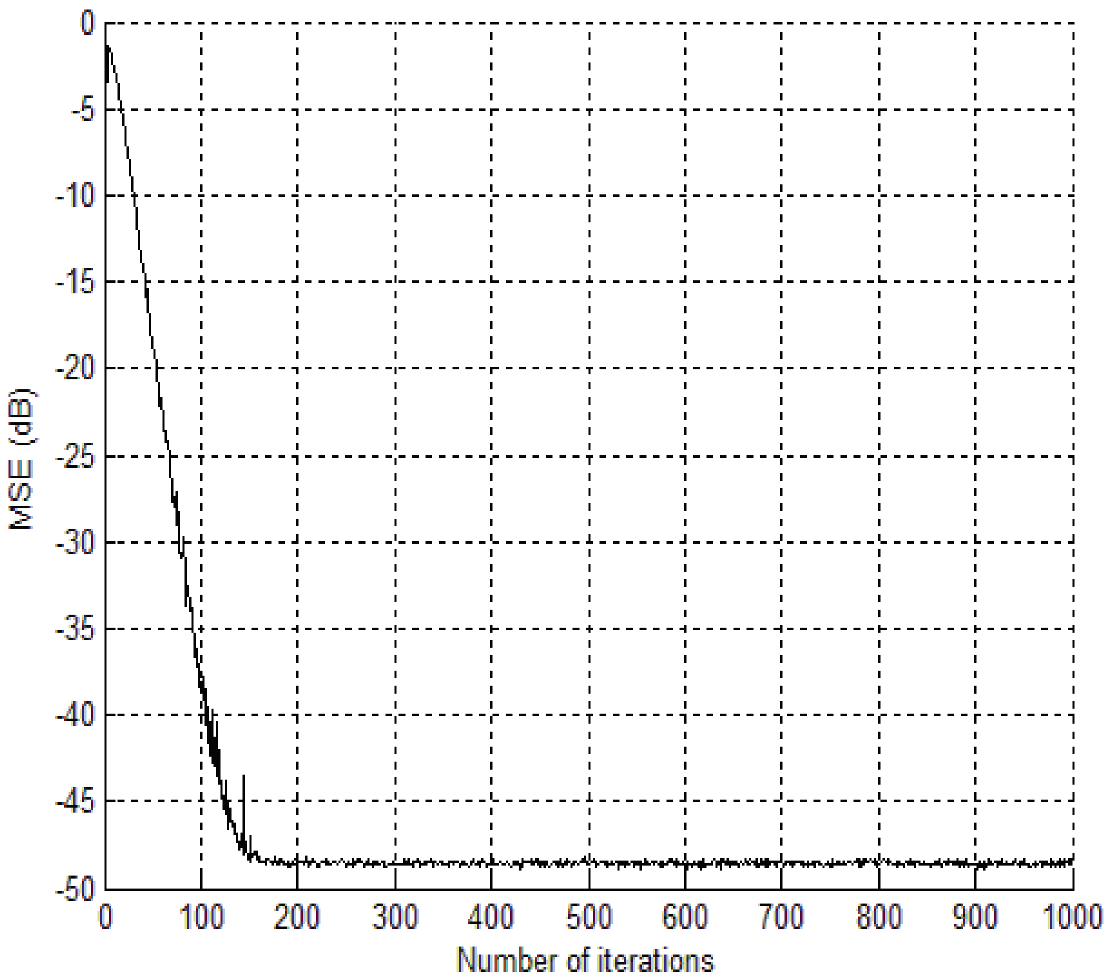

Fig. 4.8. MSE of SOV filter using AVSS-LMS algorithm.

In Fig. 4.8, simulation results are based on the implementation of SOV filter by using AVSS-LMS procedure. The result is presented for parameter estimation in terms of MSE. The AVSS-LMS is new robust form of KVSS-LMS algorithm .This algorithm provides better convergence than KVSS-LMS; and also the steady sate error is also reduced. In this algorithm, we have chosen the value $\alpha_{avss} = 0.97$, $\beta_{avss} = 0.99$, and $\gamma_{avss} = 0.00048$. The corresponding MSE is found to be approximately $-48.5\, dB$ in tracking mode. For this simulation purpose, the value of SNR is kept constant at $25\, dB$.

4.9 Performance of SOV Filter Using MAVSS-LMS Algorithm

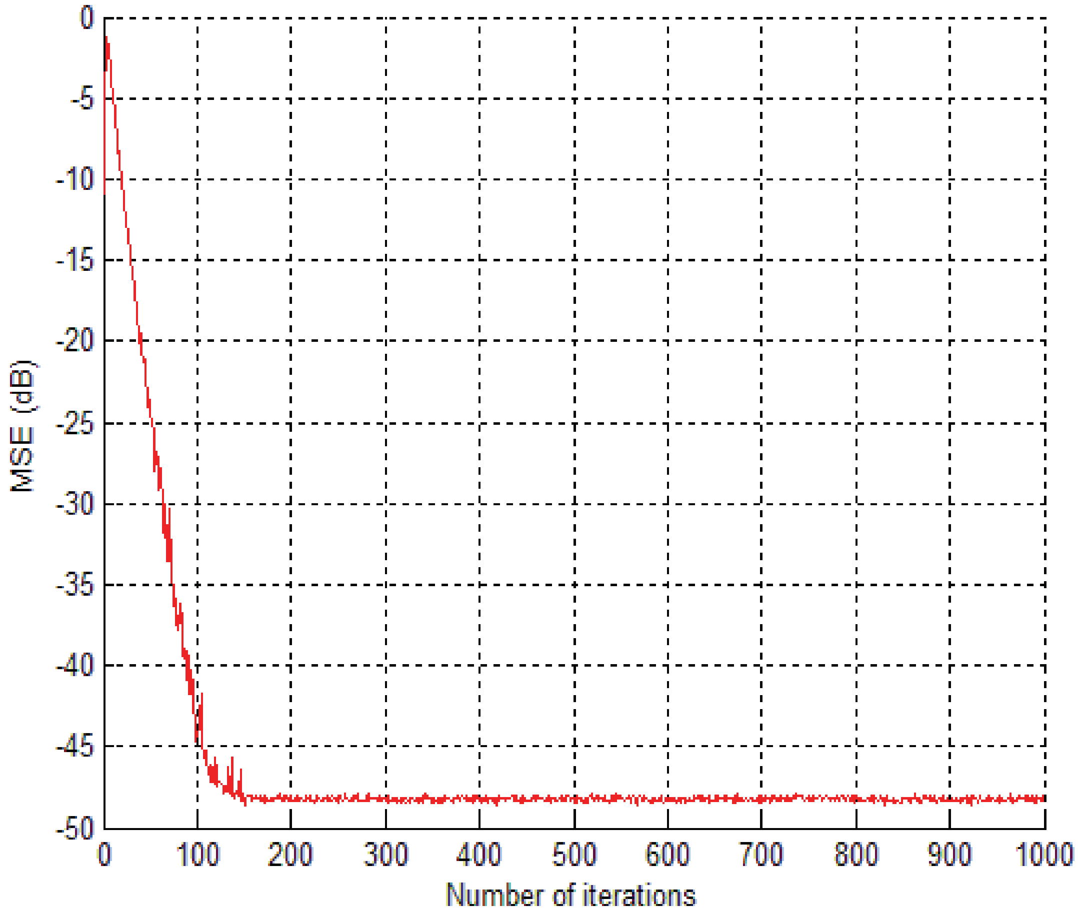

Fig. 4.9. MSE of SOV filter using MAVSS-LMS algorithm.

In Fig. 4.9, simulation results are based on the implementation of SOV filter by using MAVSS-LMS procedure. The result is presented for parameter estimation in terms of MSE. This algorithm provides better convergence than KVSS-LMS and AVSS-LMS. Its convergence is slightly better and faster than AVSS-LMS. In this simulation results, we have chosen the value of $\gamma_{avss} = 0.00048,\ \alpha_{mavss} = 0.97, \eta_{mavss} = 0.97\ and\ \lambda_{mavss} = 0.0005$. The corresponding value of average MSE is found to be approximately $-48.5\ dB$ in tracking mode.

4.10 Performance Comparison of MSE for the SOV Filter Using KVSS-LMS, AVSS-LMS and MAVSS-LMS Algorithm

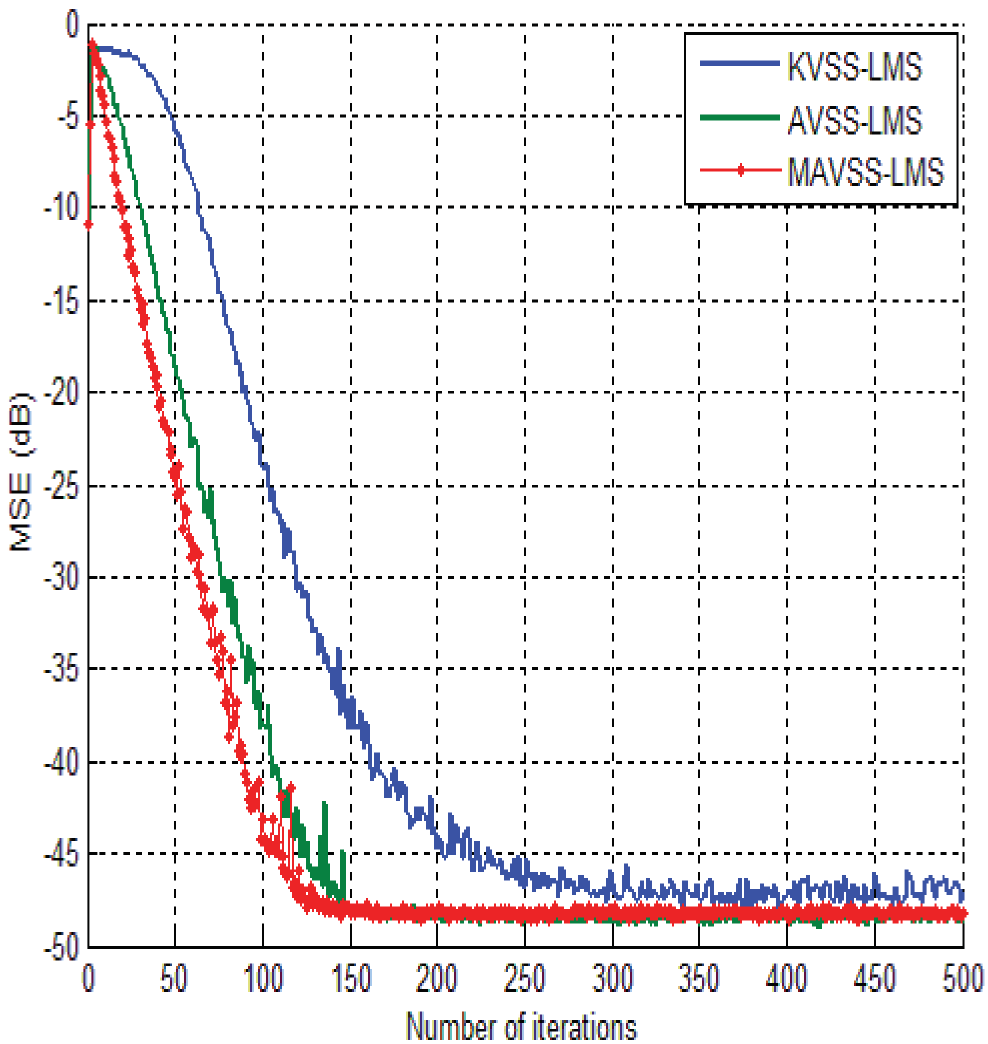

Fig. 4.10. MSE comparison for KVSS-LMS, AVSS-LMS and MAVSS-LMS algorithm.

In Fig. 4.10, simulation results show the combined performance of KVSS-LMS, AVSS-LMS and MAVSS-LMS algorithm simultaneously on the same graph, and it may be concluded that MAVSS-LMS provides highest convergence rate and followed by AVSS-LMS and after that KVSS-LMS for the same system. For this simulation purpose, the value of SNR is kept constant at $25\,dB$.

Brief Discussion

In this thesis, we initially proposed the implementation of second-order Volterra filter using LMS. We have seen that using LMS algorithm, the convergence of the system is obtained at approximately 700^{th} iteration; and to increase its convergence, we have used different mathematical formulae for variable step-size "$\mu(n)$" using the sigmoid function and the modified sigmoid functions. We have observed that by using SVSS-LMS and MSVSS-LMS, we obtained better convergence rate and tracking capability for the same adaptive system identification. We have also used KVSS-LMS, AVSS-LMS, MAVSS-LMS algorithms for varying the adaptive parameter (SS) for improving the convergence rate of the system. The MAVSS-LMS algorithm provides highest rate of convergence in comparison to AVSS-LMS and KVSS-LMS algorithms. The MAVSS-LMS converges approximately at 110^{th} iteration, while AVSS-LMS and KVSS-LMS algorithms converge at approximately140^{th} and 200^{th} iteration respectively.

CHAPTER 5

CONCLUDING REMARKS AND FUTURE SCOPE

5.1 Concluding Remarks

In this thesis, adaptive polynomial filtering for system identification application is discussed using variable-step-size LMS algorithms. The polynomial (nonlinear) filter considered here is the second-order-Volterra filter (SOV). The SOV filter is obtained by truncation of Volterra series. The performance of adaptive SOV filter is analyzed and discussed using FSS-LMS and various VSS-LMS algorithms. Firstly, we simulate the adaptive SOV filter using FSS-LMS, SVSS-LMS and MSVSS-LMS algorithm, and the convergence and tracking performances are compared in terms of MSE. Then, we have performed the simulation of adaptive SOV filter using KVSS-LMS, AVSS-LMS and MAVSS-LMS algorithms, and then the convergence as well as tracking performances are compared in terms of MSE. From the simulation results we inferred the following results

- The high value of SS gives higher convergence rate; and smaller SS gives slow convergence speed.
- SS must be high at initial stages of the algorithms to provide fast convergence rate; and SS must be small at final stage of adaptive procedure to decrease steady state error.
- The convergence rate characteristic of an MSVSS-LMS procedure is superior than a SVSS-LMS procedure and FSS-LMS procedure. The MSVSS-LMS procedure gives fastest convergence and low steady state error in comparison to SVSS-LMS procedure and FSS-LMS algorithms for same set of parameters.
- The convergence rate characteristic of an AVSS-LMS procedure is superior than a KVSS-LMS procedure.
- The convergence and steady sate performance of the MAVSS-LMS a procedure is better than AVSS-LMS procedure and KVSS-LMS procedure for same set of parameters.

The main focus is on the performance of MSVSS-LMS adaptive algorithm for nonlinear system identification. It may be an alternate approach in comparison to KVSS-LMS algorithm, in some nonlinear signal processing applications in the domain of neural networks i.e., ADALINE [51].

5.2 Future Scope

The application of signal processing is increasing day by day and also the complexity of nonlinear system is also increasing. Here we discussed only the adaptive SOV filter, but as the filter order is increased from two to three, four and so on, the sources of nonlinearity goes on increasing and we have to look-up on certain enhanced adaptive algorithms, which will overcome these nonlinear effects. All real-time applications and system models that we are dealing with are nonlinear in nature and the research is still going on nonlinear signal and system processing. We can also apply the adaptive algorithms in various applications like bio-medical engineering for genes and D.N.A identification, cell identification, communication channel estimation, noise suppression, electromechanical relays, motors, 3G, 4G etc.

REFERENCES

[1] S. Haykin, *Adaptive filter theory*, 4th ed., Englewood Cliffs, NJ: Prentice-Hall, 2002.

[2] V.J. Mathews, "Adaptive polynomial filters," *IEEE Signal Process. Mag.*, vol. 8, no. 3, pp. 10- 26, July 1991.

[3] Paulo S. R. Diniz, *Adaptive Filtering-Algorithms and Practical Implementation*, 2nd ed., LLC 233, NY: Springer, 2008.

[4] Y. Chen, J. Tian, and Liu, "Variable step-size LMS algorithm based on modified sigmoid function," *Proc. IEEE Conf. Aud. Lang., Image Process.*, Shanghai, China, pp. 627-630, July 2014.

[5] R. H. Kwong and E. W. Johnston, "A variable step-size LMS algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 7, pp. 1633–1642, July 1992.

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

[7] Z. Kun and Z. Xiubing, "A new modified robust variable step-size LMS algorithm," *Proc. IEEE conf. Ind. Elect. App.*, Xi'an, China, pp. 2699-2703, May 2009.

[8] N. Zhou and N. Holte," Least square channel estimation for a channel with fast time variation," *Proc*. ICASSP' 92, vol. 12, pp. 165-168, 1992.

[9] D. K. Borah and B. Hart, " Frequency selective fading channels estimation with polynomial time varying channel model," *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 862- 873, June 1999.

[10] A. E. Nordsjo and L. H. Zetterberg, "Identification of certain time varying non linear wiener Hammerstein system ," *IEEE Trans. Signal Process.*, vol. 49, no. 3, pp. 2721-2728, October 2001.

[11] E. Roy, W. Stewart, and T.S. Durrani, "Higher order system identification using adaptive recursive second order polynomial filters," *IEEE Signal Process. Lett.*, vol. 3, no. 10, October 1996.

[12] M. Schetzen, *The Volterra Wiener Theory of the Nonlinear Systems*, 3rd ed., New York: Wiley, 1980.

[13] E. S. Olivas, J.C. Maravilla, J. F. G. Martinez, M. M. Sober, and J. Lopez, "An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm," *IEEE Trans. Education*, vol. 41, no.1, pp. 81, February 1998.

[14] W. B. Mikhael and F. Hwu, " A fast block FIR adaptive digital filter with adaptation of individual parameters," *IEEE Trans. Circuits Syst.*, vol. 36, no. 1, pp. 1-10, January 1989.

[15] J. K. Hwong and Y. P. Li, "Variable step-size LMS algorithm with a gradient-based weighted average," *IEEE Signal Process. Lett.*, vol. 16, no. 12 , pp. 1043-1046, December 2009.

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

[17] E. Roy, R. W. Stewart, and T. S. Durrani, "Theory and applications of adaptive second order IIR Volterra filters," *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, vol. 111, Atlanta, GA, pp. 1597-160, May 1996.

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

[19] B. Widrow and S. D. Steams, *Adaptive Signal Processing*, 2nd ed., Englewood Cliffs, NJ: Prentice- Hall, 1985.

[20] M. M. Sondhi and D.A. Berkley, "Silencing echoes on the telephone network," *Proc. IEEE*, vol. 68, no. 8, pp. 948-963, August 1980.

[21] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 722-736, June 1981.

[22] O. Macchi and E. Eweda, "Second-order convergence analysis of stochastic adaptive linear filtering," *IEEE Trans. Automat. Contr.*, vol. 28, no. 1, pp. 76-85, January 1983.

[23] A. Feuer and E. Weinstein, “Convergence analysis of LMS filters with uncorrelated Gaussian data,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 1, pp. 222-230, February 1985.

[24] E. Eweda and O. Macchi, “Tracking error bounds of adaptive nonstationary filtering,” *Automarica.*, vol. 21, no. 3, pp. 293-302, 1985.

[25] R. W. Harris, D. M. Chabries, and F. A. Bishop, “A variable step (VS) adaptive filter algorithm,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 2, pp. 309-316, April 1986.

[26] N. J. Bershad, “On the optimum gain parameter in LMS adaptation,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 7, pp. 1065-1068, June 1987.

[27] S. S. Narayan, A. M. Peterson, and M. J. Narsimha, “Transform domain LMS algorithm,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 3, pp. 609–615, June 1983.

[28] N. Zhou and N. Holte, “Least square channel estimation for a channel with fast time variation,” in *Proc*. ICASSP’ 92, vol. 5, no. pp. 165-168, 1992.

[29] S. Marcos and O. Macchi, “Tracking capability of the least mean square algorithm: application to an asynchronous echo canceller,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no.11, pp.1065-1068, July 1987.

[30] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1989.

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

[32] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, “A transform domain LMS adaptive filter with variable step-size,” *IEEE Signal Process. Lett.*, vol. 9, no. 2, pp. 51-53, February 2002.

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

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

[35] Y. Zhang, N. Li, and J. A. Chambers, "A new gradient based variable step-size LMS algorithm," *Proc. ICSP*, vol. 1, pp. 16-20, Guilin, China, November 2006.

[36] Y. Zhang, J. A. Chambers, W. Wang, P. Kendrick, and T. J. Cox, "A new variable step-size LMS algorithm with robustness to non-stationary noise," *Proc. IEEE Conf. Acoust., Speech, Signal Process.*, vol. 3, pp. 1349–1352, Honolulu, Hawaii, USA, April 2007.

[37] J. E. Greenberg, "Modified LMS algorithms for speech processing with an adaptive noise canceller," *IEEE Trans. Speech, Audio Process.*, vol. 6, no. 4, pp. 338–351, July1998.

[38] P. Koukoulas and N. Kalouptsidis, "Third order Volterra system identification," Proc. ICASSP, vol. 3, pp. 2405–2408, Munich, Germany, April 1997.

[39] M. T. Akhtar, M. Abe , and M. Kawamata, "A new variable step size LMS algorithm-based method for improved online secondary path modeling in active noise control systems," *IEEE Trans. Speech, Audio Process.*, vol. 14, no. 2, pp. 720-726, March 2006.

[40] P. Koukouplas and N. Kalouptsidis, "Second order Volterra system identification," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3574-3577, December 2000.

[41] R. Bernardini, "A fast algorithm for general Volterra filtering," *IEEE Trans. Commun.*, vol. 48, no. 11, pp. 1853-1864, November 2000.

[42] T. Ogunfunmi, *Adaptive non-linear system identification (The Volterra and Wiener model approaches)*, Springer Science, 2007.

[43] M. Sayadi, F. Fnaiech, and M. Najim, "An LMS adaptive second-order Volterra filter with a zeroth-order term: steady-state performance analysis in time-varying environment," *IEEE Trans. Signal Process.*, vol. 47, no.3, pp. 872-876, March 1999.

[44] C. Krall, K. Witrisal, and H. Koeppl, "Minimum mean square error equalization for second order Volterra system," *IEEE Trans. Signal Process.*, vol. 56, no.10, October 2008.

[45] A. K. Kohli and D. K. Mehra, "Tracking of time varying channels using two-step LMS-type adaptive algorithm," *IEEE Trans. Signal Process.*, vol. 54, no. 7, pp. 2606-2615, July 2006.

[46] V. H. Nascimento and A. H. Sayed, "On the learning mechanism of the adaptive filters," *IEEE Trans. Signal Process. Lett.*, vol. 48 , no. 6, pp. 1609-1625, June 2000.

[47] J. B. Evans, P. Xue, and B. Liu," Analysis and implementation of variable step-size adaptive algorithm," *IEEE Trans. Signal Process.*, vol. 41, no. 8, pp. 2517-2535, August 1993.

[48] J. K. Hwong and Y. P. Li, “Variable step-size LMS algorithm with a gradient-based weighted average,” *IEEE Signal Process.*, vol. 16, no. 12, pp. 1043-1046, December 2009.

[49] H. Shin, A. H. Sayid, and W. Song, “ Variable step-size NLMS and affine projection algorithms,” *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132-135, February 2004.

[50] Li Xiao-Bo, Fan Yang-Yu, and Peng-Ke, “ A variable step-size LMS adaptive filtering algorithm,” *Proc. IEEE Wireless Commun., Networking, Mobile Comput.*, Beijing, China, pp. 1-4, September 2009.

[51] Savita, D. S. Kapoor, and A. K. Kohli, “Comparative evaluation of generalized ADALINE using variable learning rate parameter,” *IJCEM, International Journal of Computational Engineering & Management*, vol. 16, no. 4, pp. 44 – 48, July 2013.

801461028

by Shiv Kumar Sharma

801461028

ORIGINALITY REPORT

11%	5%	9%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Paulo S.R. Diniz. "Nonlinear Adaptive Filtering", Adaptive Filtering, 2008 Publication	1%
2	The Kluwer International Series in Engineering and Computer Science, 2002. Publication	1%
3	Zou kun. "A new modified robust variable step size LMS algorithm", 2009 4th IEEE Conference on Industrial Electronics and Applications, 05/2009 Publication	1%
4	Li, Jianhua, and Jing Li. "A novel variable step size LMS algorithm based on decorrelation", 2010 3rd International Congress on Image and Signal Processing, 2010. Publication	1%
5	www.mif.vu.lt Internet Source	1%
6	Kaur, Rashpinder, Gurjinder Singh, Balwinder Singh Dhaliwal, and Sandeep Singh Gill.	1%