CHANNEL SPARSITY AWARE POLYNOMIAL EXPANSION FILTERS FOR NONLINEAR ACOUSTIC ECHO CANCELLATION

by

Vinith Vijayarajan

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

Department of Electrical & Computer Engineering

Hammond, Indiana

December 2018

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lizhe Tan, Committee Chair, Professor

Department of Electrical and Computer Engineering

Dr. Bin Chen, Associate Professor

Department of Electrical and Computer Engineering

Dr. Yao Xu, Assistant Professor

Department of Electrical and Computer Engineering

Approved by:

Dr. Vijay Devabhaktuni

Head of the Graduate Program

Dedicated to my parents Vijayarajan Dhanushkodi and Vani Vijayarajan

ACKNOWLEDGMENTS

I wish to thank my advisor, Dr. Li-Zhe Tan, for his scientific insights and guidance in this research. Besides inculcating technical expertise within me, he also took every opportunity to make me a well- rounded researcher. I have always marveled at his scientific and engineering expertise across various fields. I have learned work ethic, focus and dedication from the best. His passion for research and scientific truth will always inspire me.

I would also like to thank my other committee members, Dr. Bin Chen, and Dr. Yao Xu, for their patience, inputs and feedbacks. I would like to thank the peers in my research group, for all the ideas and brainstorming with them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

VT	Volterra Filter
FLAF	Functional Link Adaptive filter
EMFN	Even Mirror Fourier Nonlinear
LMS	Least Mean Square
RLS	Recursive Least Squares
S-SEQ-RLS	Sparse-Sequential-Recursive Least Squares
NMSE	Normalized Mean Square Error
ERLE	Echo Return Loss Enhancement
NAEC	Nonlinear Acoustic Echo Cancellation
J	Cost Function
W	Filter Coefficients

ABSTRACT

Author: Vijayarajan, Vinith. MSECE
Institution: Purdue University
Degree Received: December 2018
Title: Channel Sparsity Aware Polynomial Expansion Filters for Nonlinear Acoustic Echo Cancellation
Committee Chair: Li-Zhe Tan

Speech quality is a demand in voice commanded systems and in telephony [1]-[5]. The voice communication system in real time often suffers from audible echoes. In order to cancel echoes, an acoustic echo cancellation system is designed and applied to increase speech quality both subjectively and objectively.

In this research we develop various nonlinear adaptive filters wielding the new channel sparsity-aware recursive least squares (RLS) algorithms using a sequential update. The developed nonlinear adaptive filters using the sparse sequential RLS (S-SEQ-RLS) algorithm apply a discard function to disregard the coefficients which are not significant or close to zero in the weight vector for each channel in order to reduce the computational load and improve the algorithm convergence rate. The channel sparsity-aware algorithm is first derived for nonlinear system modeling or system identification, and then modified for application of echo cancellation. Simulation results demonstrate that by selecting a proper threshold value in the discard function, the proposed nonlinear adaptive filters using the RLS (S-SEQ-RLS) algorithm can achieve the similar performance as the nonlinear filters using the sequential RLS (SEQ-RLS) algorithm in which the channel weight vectors are sequentially updated. Furthermore, the proposed channel sparsity-aware RLS algorithms require a lower computational load in comparison with the non-sequential and non-sparsity algorithms. The computational load for the sparse algorithms can further be reduced by using data-selective strategies.

CHAPTER 1 INTRODUCTION

1.1 Motivations

Speech is the most important mode of human communication. Voice communication is an integral part of voice commanded systems and in telephony, hence the issue of speech quality is more critical. One of the difficult artifacts to eliminate in real time is the audible echo. Acoustic echo cancellation (AEC) is one of efficient techniques for application in telecommunication. The AEC applies the adaptive filter [1]–[6] to cancel a small portion of the received signal which is designated as the echo signal and may be leaked for transmission during two-party or multi-party voice communications. The echo impairment can be very annoying to customers. It is necessary to design an effective acoustic echo cancellation system to increase speech quality both subjectively and objectively. For echo cancellation application, a linear echo path usually is assumed so that an adaptive finite impulse response (FIR) filter can be adopted to model the linear echo path. However, for low cost and power efficient mobile devices, there exits nonlinear distortion due to low-quality and overdriven audio components such as converter, amplifier, loudspeaker, and microphone, the linear adaptive filter cannot perform well with leading to strong residual echoes, which greatly degrade communication quality. Though echo cancellation has been studied for several decades, some fundamental challenges are still needed to be addressed. One of them is the non-linearity in the acoustic echo path. Recently, the diagonal-structure Volterra and functional link adaptive filters [7], [8] had been introduced to improve the nonlinear processing capability for echo cancellation using the RLS algorithm. The main problem of echo cancellation is estimating and modeling non-linear echo path accurately in order to suppress distortions.

In order to tackle problems of nonlinear echo cancellation (NAEC), several nonlinear filters have been investigated [6]-[13], which include linear FIR filter with non-linear preprocessor, static power filter, cascade and parallel dynamic power filter, hybrid Taylor-Volterra model and exponential functional link neural network and so on. However, these filters either cannot effectively model the non-linear distortion introduced by the system or are prohibitively intense computationally. These nonlinearities are introduced by low quality and

overdriven audio components (e.g., amplifiers, loudspeakers etc.). In this research we try to develop new nonlinear adaptive filters for nonlinear acoustic echo cancellation that is computationally feasible and is robust to model nonlinearity of the echo path, and to validate the developed nonlinear adaptive filters with suitable investigations and simulation results.

1.2 Objectives

In this thesis work, one of the tasks is to design efficient nonlinear adaptive filters for NAEC with realizable computational complexity and robust system modeling ability. Although the improvement using a nonlinear filter for NAEC is validated, most of the filter coefficients that need to be estimated are not significant or close to zero. This sparsity in the echo path property can be exploited while modeling. Hence the non-significant coefficients can be ignored while updating filter coefficients [9]-[14]. Usually, to improve the algorithm convergence rate, the RLS algorithm is often used [6] at a cost of a large computation load. Exploiting the sparsity property, the number of coefficients which are required to model the echo path can be minimized and furthermore can be updated sequentially for each channel in the nonlinear filter using the RLS algorithm. It is well known that the RLS type algorithm [6], [20] offers faster convergence rate and lower mean square error over the LMS type algorithm [1], [6]-[7], [19]. But it suffers a high computational load. Therefore, by exploiting the sparsity property and channel selectivity, we intend to design nonlinear adaptive filters that use the sparse sequential RLS (S-SEQ-RLS) algorithm for effective NAEC.

1.3 Echo Cancellation Problem

In telephony, the near end speakers echo is the time delayed version of the near end speakers signal transmitted to the far end speaker, which is received along with the far end speakers' signal as noise. There are two major types of echo:

1. **Hybrid echo**

   This type echo is due to different electrical connections of the telephonic system between the two speakers.

2. Acoustic echo

The more common type of echo in a phone call between two speakers is the acoustic echo, which occurs due to the near end speakers' voice getting reflected across the receiving far end speakers surrounding and then returning back to the near end speaker as unwanted noise. Figure 1.1 depicts a typical echo cancellation system.

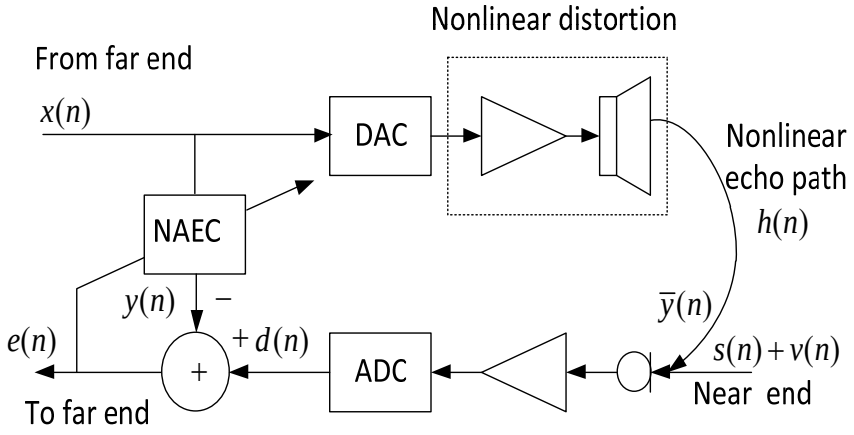

Figure 1.1. Block diagram of an echo cancellation system.

With reference to Figure 1.1, the following terms can be defined at any given time n:

1. Far End Signal: $x(n)$ is the far end signal, which becomes the echo signal when reflected around the acoustic environment of the near end speaker.
2. Acoustic Echo Path: $H(n)$ is the echo path of the near end speaker to the microphone at the near end as shown in Figure 1.1, and acts as a tapped filter, which delays the far end signal $x(n)$ to produce the echo signal. The echo path is the unknown system to be estimated.
3. Echo Signal: $\bar{y}(n)$ is the echo signal produced when the far end speaker's signal goes through the echo path, which can be estimated as

$$y(n) = X^T(n) * W(n) \tag{1.1}$$

where $W(n)$ is the vector of filter coefficients, and it is also designated as the echo path estimate.

4. Background Noise: $v(n)$ is the background noise, in addition to the near end speakers' signal, $s(n)$, which gets added to the echo signal and needs to be taken into consideration when calculating the estimate of the echo path.
5. Measured Signal: $d(n)$ is the measured signal, which is the summation of the echo signal and the background noise for a single talk case.

$$d(n) = \bar{y}(n) + v(n) \tag{1.2a}$$

For a double talk case, the measured signal becomes

$$d(n) = \bar{y}(n) + s(n) + v(n) \tag{1.2b}$$

6. Echo Path Estimate: $W(n)$ is the estimate of the echo path that is calculated by the mean square error (MSE) filter or the adaptive filter.
7. Estimated Signal: $y(n)$, defined in Equation (1), is the estimate of echo signal obtained by passing the far end speaker's signal through the estimate of the echo path.
8. Error Signal: $e(n)$ is the difference between the actual measured signal and the estimated echo signal.

Optimal signal estimation involves estimation of the unknown quantity or coefficient vector $W(n)$ (in this case the echo path) using data from other related signals of interest, namely input signal $x(n)$ and unknown systems output or echo signal $d(n)$. All these type algorithms work on minimizing the square of the error signal $e(n)$ i.e. to minimize the difference between the signal estimate and the actual signal. Another important factor that influences the performance of the estimator is the measurement or background noise that is added to the output signal, $v(n)$.

The mean squares error (MSE) estimators, such as the Wiener filter, look at the entire history of the present and past samples of input data to make an ensemble estimation at the present instance of time.

The minimum mean square MMSE estimators such as the least mean square (LMS) and recursive least squares (RLS), are time varying estimators, which means that at every given instant of time, the estimate of the unknown variable is estimated and updated based on the existing present and past values of input data. Such estimators are also known as recursive mean square estimators.

Depending on whether the estimators are ensemble estimators or recursive estimators, $W(n)$ as the estimate of $H(n)$, is either predicted once based on the causal history of the data, or is estimated at every time iteration by updating filter parameters. The difference in the update equations is what gives rise to the different types of MMSE adaptive filters, which are further discussed in detail in the following sections.

1.4 Echo Cancellation Equation

This section describes the context and usage of the equations and various signals that are defined in the previous section. As shown in Figure 1.1, at any given discrete time index n, the acoustic echo path of the near end speaker, of length L, is given by.

$$H = \left[h_0, h_1, h_2, h_3, ...h_L\right]^T \tag{1.3}$$

where L is the memory length. The input signal vector which then goes through the echo path or the far end signal vector is then given by

$$X\left(n\right) = \left[x\left(n\right)x\left(n-1\right)x\left(n-2\right)x\left(n-3\right)...x\left(n-L+1\right)\right]^T \tag{1.4}$$

The echo signal of the far end speaker can expressed as

$$\overline{y}\left(n\right) = X^T\left(n\right) * H \tag{1.5}$$

Let v(n) be the background noise or the near end speaker's signal accordingly for a single talk case (during the scenario of double talk, the combined signal of v(n) +s(n) will be the near end speaker's signal [2]) The variance of the background noise is given by $\sigma_{v2}(n)$. The total microphone signal for single talk is given by

$$d\left(n\right) = \overline{y}\left(n\right) + v\left(n\right) \tag{1.6}$$

The objective of the echo canceller is to estimate the echo path as

$$W = \left[w_0 \quad w_1 \quad \cdots \quad w_L\right]^T \tag{1.7}$$

The estimated echo signal can then be calculated as.

$$y(n) = X^T(n)W \tag{1.8}$$

The error between the estimated echo and the actual echo is the residual error, given by

$$e(n) = d(n) - y(n) \tag{1.9}$$

This error needs to be subject to the MMSE condition, which will be discussed in the next section.

1.5 Principle of Orthogonality

The cost function for optimization of the MMSE estimator, J, is chosen to minimize the mean square error, given by

$$J = E\left[|e(n)|^2\right] \tag{1.10}$$

where E denotes the expectation operator, and | | refers to the magnitude. In order to minimize this cost function, a gradient operator ∇ is defined on the basis of the coefficients of the filter. Hence, for the ith filter coefficient, w_i, we define

$$\nabla_i = \frac{\partial}{\partial w_i} \tag{1.11}$$

Applying the gradient operator ∇ to the cost function J, the gradient vector $\nabla_i J$ is obtained and is written as

$$\nabla_i J = \frac{\partial J}{\partial w_i} \tag{1.12}$$

To minimize the cost function, the elements of the vector in Equation (1.12) must all be set to zero, that is,

$$\nabla_i J = 0 \text{ for } i = 0,1,2,3,.. \tag{1.13}$$

Hence, to minimize the cost function J, with regards to Equation (1.10), simplifying the partial derivatives results in the following equation solution [3] at any given time instant n yields

$$\nabla_i J = E\left[x(n-i)e^*(n)\right] = 0 \tag{1.14}$$

Note that Equation (1.14) specifies the operating conditions required for the minimization of the cost function J, that is,

$$E\left[x(n-i)e^*(n)\right] = 0 \tag{1.15}$$

This is known as the principle of orthogonality, since two signals are known as orthogonal signals when the correlation between them is zero. Here the condition for optimal linear filtering is that the estimated error is orthogonal to all L input samples that are involved in the estimation of the echo path.

1.6 Wiener Solution

Expanding on Equation (1.15) for the principle of orthogonality, with a definition of $r_{ex(n)} = E\left[x(n-i)e^{*}(n)\right]$ as the cross correlation between the error signal and the input signal at any given time instant n and for a given filter coefficient W(n), the following can be deduced. Hence, the deduced equation at any given instant of n for the cross correlation now looks as shown below:

$$r_{ex(n)} = E\left[x(n-i)e^{*}(n)\right] = E\left[x(n-i)(d(n)-y(n))\right] = 0 \tag{1.16}$$

Simplifying Equation (1.16) results in an expression given below:

$$R_{y(n)X(n)} = P_{d(n)X(n)} \tag{1.17}$$

where $R_{y(n)X(n)}$ is the correlation between the input signal and the estimated echo signal; and $P_{dX(n)}$ is the correlation between the received echo and input signal. From observing above equation and noting that the estimated output $y(n)$ is calculated by passing the input signal $x(n)$ through the estimated echo path impulse response $W(n)$, the following relationship can be obtained;

$$W^{T}(n) * R_{XX(n)} = R_{dX(n)} \tag{1.18}$$

where

$$R_{XX(n)} = \begin{bmatrix} E[x(n)x(n)] & E[x(n)x(n-1)] & \cdots & E[x(n)x(n-L)] \\ E[x(n-1)x(n)] & E[x(n-1)x(n-1)] & \cdots & E[x(n-1)x(n-L)] \\ \vdots & \ddots & \cdots & \vdots \\ E[x(n-L)x(n)] & E[x(n)x(n-1)] & \cdots & E\{x(n-L)x(n-L)\} \end{bmatrix} \tag{1.19}$$

and

$$R_{dX(n)} = \begin{bmatrix} E[d(n)x(n)] \\ E[d(n)x(n-1)] \\ \vdots \\ E[d(n)x(n-1)] \end{bmatrix} \tag{1.20}$$

Solving for $W(n)$, we have

$$W_o = R_{XX(n)}^{-1} R_{dX(n)} \tag{1.21}$$

Hence, it is seen from Equation (1.18) that the estimated echo path $W(n)$ at any given instance of time n depends on the cumulative correlation matrices $R_{XX(n)}$ and $R_{dX(n)}$ until that instance of time n.

1.7 Organization of Thesis

To evaluate the performance of the proposed nonlinear adaptive filters, a series of simulation experiments are conducted, including nonlinear system identification and nonlinear AEC in single-talk and double-talk scenarios.

The structure of this paper is organized as follows. Chapter 2 proposes the new channel sparsity aware sequential RLS (S-SEQ-RLS) algorithm. In Chapter 3, we introduce various polynomial expansion filters for effective nonlinearity acoustic impulse response (AIR) modeling. Analysis of the sparsity and computational complexity is made in Chapter 3. Chapter 4 presents the simulation results. Finally, conclusions are presented in Chapter 6.

CHAPTER 2 LEAST MEAN SQUARE ALGORITHM, AND SPARSE SEQUENTIAL RECURSIVE LEAST SQUARE ALGORITHM

2.1 Sparse Least Mean Square (S-LMS) Algorithm

In Section 1.6, we have derived the optimal solution for the parameters of the adaptive filter. The solution leads to the minimum mean-square error for estimating the measured signal $d(n)$, echo signal and background noise The optimal (Wiener) solution is given by Equation (1.18), where the matrix inversion is required. A steepest descent based algorithm can be used to search the Wiener solution as follows:

$$W(n+1) = W(n) - \mu \hat{g}_w(n) = W(n) + 2\mu\left(P(n) - R(n)W(n)\right) \tag{2.1}$$

where $P(n)$ and $R(n)$ are the best estimate of R and P, note that

$$R = E[X(n)X^T(n)]$$

(2.2a)

$$P = E[d(n)X(n)] \tag{2.2b}$$

For n=0,1,2,3,…, $\hat{g}_w(n)$ represents an estimate of gradient vector of the objective function with respect to the filter coefficients. If we estimate the instantaneous gradient, then following results are given:

$$J = e^2(n) = \left(d(n) - W^T(n)X(n)\right)^2 \tag{2.3}$$

$$\frac{dJ}{dW(n)} = 2\left(d(n) - W^T(n)X(n)\right)\frac{d\left(d(n) - W^T(n)X(n)\right)}{dW(n)} = -2e(n)X(n) \tag{2.4}$$

Substituting the instantaneous gradient $\frac{dJ}{dW}$ into Equation (2.1), we get

$$W(n+1) = W(n) + 2\mu e(n)X(n) \tag{2.5}$$

where μ is the convergence parameter controlling the speed of convergence and must be within the range $0 < \mu < 1/\lambda_{\max}$, where $\lambda_{\max}$ is the maximum eigenvalue of the autocorrelation matrix $R = E\left[X^T(n)X(n)\right]$. Equation (2.5) is referred to the least mean square (LMS) algorithm.

We conclude the implementation of the LMS algorithm by the following steps:

1. Initialize $w_0, w_1, w_2, ..., w_N$ to arbitrary values
2. a discard function is considered, that is

$$f(w) = \begin{cases} w & |w| > \varepsilon \\ 0 & |w| \le \varepsilon \end{cases}$$

3. Read $d(n)$ and $x(n)$, and perform digital filtering

$$y(n) = w_0 x(n) + w_1 x(n) + w_2 x(n) + ... + w_N x(n - N + 1)$$

4. Compute the output error

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

5. Now update the coefficient using equation

For $k = 0, 1, \cdots, N-1$

$$w_k = w_k + 2\mu e(n) x(n-k)$$

Vector Form: $w_k = w_k + 2\mu e(n) x(n-k)$

2.2 Sparse Sequential Recursive Least Squares (S-SEQ-RLS) Algorithm

The recursive least squares (RLS) algorithm aims to minimize the sum of the squares of the difference between the desired signal and the filter output signal using the new samples of the incoming signal. The RLS algorithm updates filter coefficients in a recursive form at each iteration. The RLS algorithm is well known for its fast convergence even when the eigenvalue spread of the input signal correlation matrix is large. These algorithms offer excellent performance at the cost of larger computational complexity. However, there exists a problem of stability in comparison with the LMS algorithm. In this section, we will develop the sparse sequential-RLS(S-SEQ-RLS) algorithm [9]-[10] that aims to reduce the computation but still maintain the performance. Considering the channel sub filters for system modeling, the error at past time index is expressed as

For $1 < i < n$

$$e(i) = d(i) - \sum_{k=0}^{M} W_k^T(n) X_k(i) \tag{2.5}$$

where the kth channel sub filter has its weight vector and input vector defined below:

$$W_k(n) = [w_0(n)\ \ w_1(n)\ \ \cdots w_{M_k-1}(n)]^T \tag{2.6}$$

$$X_k(i) = [x(i)\ \ x(i-1)\ \ \cdots x(i-M_k+1)]^T \tag{2.7}$$

For the standard SEQ-RLS algorithm [11], the following objective function is minimized:

$$\zeta(n) = \frac{1}{2}\sum_{i=1}^{n}\lambda^{n-i}\left(d(i) - \sum_{k=0}^{M} W_k^T(n)X_k(i)\right)^2 \tag{2.8}$$

where $0 < \lambda < 1$. For the sparse sequential-RLS (S-SEQ-RLS) algorithm, if a discard function is considered that is,

$$f(w) = \begin{cases} w & |w| > \varepsilon \\ 0 & |w| \le \varepsilon \end{cases} \tag{2.9}$$

where ε is the small number, then the objective function in is changed to

$$\zeta(n) = \frac{1}{2}\sum_{i=1}^{n}\lambda^{n-i}\left(d(i) - \sum_{k=0}^{M} f(W_k(n))^T X_k(i)\right)^2 \tag{2.10}$$

Taking derivative of the j-th sub filter coefficient vector $W_j(n)$ and setting the result to zero lead to

$$\begin{aligned} \frac{\partial\zeta(n)}{\partial W_j(n)} &= -\sum_{i=1}^{n}\lambda^{n-i}[d(i)X_j(i)F(W_j(n)) \\ &+ \sum_{k=0,k\neq j}^{M} F(W_j(n))X_j(i)X_k^T(i)f(W_k(n))] \\ &+ \sum_{i=0}^{n}\lambda^{n-i}\left(F(W_j(n))X_j(i)X_j^T(i)f(W_j(n))\right) = 0. \end{aligned} \tag{2.11}$$

Assuming that each sub filter coefficients are time slow varying, that is, $W_k(n) \approx W_k(n-1)$ for $k \neq j$, the cross-correlation matrix in Equation (2.11) is derived as

$$\begin{aligned} P_j(n) &= \sum_{i=1}^{n}\lambda^{n-i}[d(i)F(W_j(n))X_j(i) - \sum_{k=0,k\neq j}^{M} F(W_j(n-1)X_j(i)X_k^T(i)F(W_k(n-1))] \\ &\approx \lambda\sum_{i=1}^{n-1}\lambda^{n-1-i}[d(i)F(W_j(n-1))X_j(i) - \sum_{k=0,k\neq j}^{M} X_j(i)X_k^T(i)W_k(n-1)] + d(n)F(W_j(n))X_j(n) \\ &- \sum_{k=0,k\neq j}^{M} F(W_j(n-1))X_j(n)X_k^T(n)f(W_k(n-1)). \end{aligned} \tag{2.12}$$

Its recursion can be expressed below:

$$P_j(n) = \lambda P_{j-1}(n-1) + d(n)F(W_j(n))X_j(n) - \sum_{k=0,k\neq j}^{M} F(W_j(n-1))X_j(n)X_k^T(n)\,f(W_k(n-1)). \tag{2.13}$$

Again, the autocorrelation matrix is expressed as

$$\begin{aligned} R_j(n) &= \sum_{i=1}^{n} \lambda^{n-i} F(W_j(n))X_j(i)X_j^T(i) \\ &= \sum_{i=1}^{n} \lambda^{n-i} F(W_j(n))X_j(i)X_j^T(i)F(W_j(n)) \\ &= \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i} F(W_j(n-1))X_j(i)X_j^T(i)F(W_j(n-1)) \\ &+ F(W_j(n))X_j(n)X_j^T(n)F(W_j(n)). \end{aligned} \tag{2.14}$$

Exploiting the recursion yields

$$R_j(n) = \lambda R_j(n-1) + F(W_j(n))X_j(n)X_j^T(n)F(W_j(n)) \tag{2.15}$$

The Wiener solution is then given by

$$W_j(n) = R_j^{-1}(n)P_j(n) = Q_j(n)P_j(n) \tag{2.16}$$

By using the inversion lemma, we obtain

$$k_j(n) = \frac{\lambda^{-1}Q_j(n-1)F(W_j(n))X_j(n)}{1+\lambda^{-1}X_j^T(n)F(W_j(n))Q_j(n-1)F(W_j(n))X_j(n)} \tag{2.17}$$

$$Q_j(n) = \lambda^{-1}Q_j(n-1) - \lambda^{-1}k_j(n)X_j^T(n)F(W_j(n))Q_j(n-1) \tag{2.18}$$

Since

$$k_j(n) = \lambda^{-1}Q_j(n-1)F(W_j(n))X_j(n) - \lambda^{-1}k_j(n)X_j^T(n)F(W_j(n))Q_j(n-1)F(W_j(n))X_j(n), \tag{2.19}$$

The gain vector is also expressed as

$$k_j(n) = Q_j(n)F(W_j(n))X_j(n) \tag{2.20}$$

Substituting Equations (2.13) and (2.20) in (2.16) yields the following:

$$W_j(n) = W_j(n-1) + k_j(n)[d(n) - \sum_{k=1,k\neq j}^{M} X_j^T(n)\,f(W_k(n-1))] - X_j^T(n)F(W_j(n))W_j(n-1). \tag{2.21}$$

In fact, for time slow varying coefficients, we assume

$$F(W_j(n))W_j(n-1) \approx F(W_j(n-1))W_j(n-1) \tag{2.22}$$

We obtain

$$W_j(n) = W_j(n-1) + k_j(n)[d(n) - \sum_{k=0}^{M} X_j^T(n) f(W_k(n-1))] \tag{2.23}$$

Define the innovation as

$$\alpha(n) = d(n) - \sum_{k=0}^{M} X_k^T(n) f(W_k(n-1)) \tag{2.24}$$

The update equations become

$$W_j(n) = W_j(n-1) + k_j(n)\alpha(n) \tag{2.25}$$

Then the sparse sequential RLS (S-SEQ-RLS) algorithm is summarized as

For $j = 0,1,\cdots,L$

$$k_j(n) = \frac{\lambda^{-1}Q_j(n-1)F(W_j(n-1))X_j(n)}{1+\lambda^{-1}X_j^T(n)F(W_j(n-1))Q_j(n-1)F(W_j(n-1))X_j(n)} \tag{2.26}$$

$$Q_j(n) = \lambda^{-1}Q_j(n-1) - \lambda^{-1}k_j(n)X_j^T(n)F(W_j(n-1))Q_j(n-1) \tag{2.27}$$

$$\alpha(n) = d(n) - \sum_{k=0}^{M} X_k^T(n) f(W_k(n-1)) \tag{2.28}$$

$$W_j(n) = W_j(n-1) + k_j(n)\alpha(n). \tag{2.29}$$

Note that $F(W_j(n))$ denotes the Jacobian matrix of $f(W_j(n))$. It is a diagonal matrix with elements being either ones or zeros.

CHAPTER 3 FUNCTION EXPANSION ADAPTIVE FILTERS

In order to model a nonlinear echo path due to signal companding and/or due to over driven amplifiers near to saturation, we apply the sparse LMS and sparse SEQ-RLS algorithm to the functional expansion filters.

3.1 Third-Order Volterra Filter

Let $x(n)$ and $y(n)$ be input and output signals respectively. The second-order Volterra series expansion [7],[11], [13] with a memory length of $N+1$ in terms of the diagonal channel based structure is given by

$$y(n) = f^T(W_0(n))X_0(n) + \sum_{k=1}^{N_2} f^T(W_k(n))X_k(n) \tag{3.1}$$

where N_2 is the number of significant second-order channels, $W_0(n)$ and $X_0(n)$ are the linear filter coefficient vector and linear input while $W_k(n)$ and $X_k(n)$ for $0 < k \le N_2$ are the kth second-order Volterra diagonal channels. By selecting $N_2 < N+1$, we can achieve the computational load reduction. The third-order Volterra filter according to the diagonal channel structure can be developed similarly, that is,

$$y(n) = f^T(W_0(n))X_0(n) + \sum_{k=1}^{N_2} f^T(W_k(n))X_k(n) + \sum_{k=N_2+1}^{N_2+N_3} f^T(W_k(n))X_k(n) \tag{3.2}$$

where $W_k(n)$ and $X_k(n)$ for $N_2 < k \le N_2 + N_3$ designate the kth third-order Volterra channel coefficient vector and the corresponding input vector. All input vectors for the third-order Volterra filter are listed in Table 3.1.1.

Table 3.1.1. Channel input vectors in the third-order Volterra filter

Input vector	Elements
$X_0(n)$	$x(n), x(n-1), \cdots, x(n-N)$
$X_1(n)$	$x^2(n), x^2(n-1), \cdots, x^2(n-N)$
$X_2(n)$	$x(n)x(n-1), x(n-1)x(n-2), ..., x(n-N+1)x(n-N)$
...	...
$X_{N_2}(n)$	$x(n)x(n-N_2+1), x(n-1)x(n-N_2), \cdots$
$X_{N_2+1}(n)$	$x^3(n), x^3(n-1), \cdots, x^3(n-N)$
$X_{N_2+2}(n)$	$x^2(n)x(n-1), x^2(n-1)x(n-2), \cdots, x^2(n-N+1)x(n-N)$
...	...
$X_{N_2+1+P_3}(n)$	$x^2(n)x(n-P_3), x^2(n-1)x(n-P_3-1), \cdots$
$X_{N_2+2+P_3}(n)$	$x(n)x^2(n-1), x(n-1)x^2(n-2), \cdots, x(n-N+1)x^2(n-N)$
...	...
$X_{N_2+1+2P_3}(n)$	$x(n)x^2(n-P_3), x(n-1)x^2(n-P_3-1), \cdots$
$X_{N_2+2+2P_3}(n)$	$x(n)x(n-1)x(n-2), x(n-1)x(n-2)x(n-3), \cdots, x(x-N+2)x(n-N+1)x(n-N)$
...	...
$X_{N_2+N_3}(n)$	$x(n)x(n-P_3+1)x(n-P_3), x(n-1)x(n-P_3)x(n-P_3-1), ...$

Note that $N_3 = 1 + 2P_3 + (P_3 - 1)P_3/2 = (P_3 + 1)(P_3 + 2)/2$ is the number of the third-order Volterra diagonal channels and P_3 is the maximum delay in the first signal element in the invariant third-order diagonal channel as shown in Table 3.1.1. Since $N_3 \square (N+1)(N+3)N/2$ (total number of third-order diagonal channels), a significant reduction of the computational load can be achieved. The signal vectors for time-invariant channels are listed in Table 3.1.1.

3.2 Functional Link Artificial Neural Network (FLANN) Filter

Although the Volterra filter has a property of being universal approximation for causal, time invariant, finite-memory, continuous nonlinear systems, but modeling the echo path will require a large number of filter coefficients.

An alternative choice is the functional link artificial neural network (FLANN) adaptive filter [8]. The relationship between the input and output for an FLANN filter with an order of P is given by

$$y(n) = f^T(W_0(n))X_0(n) + \sum_{k=1}^{M} f^T(W_k(n))X_k(n) \tag{3.3}$$

where $M + 1 = 2P + 1$ and the channel input vectors are listed in Table 3.2.1.

Table 3.2.1. Channel input vectors in the FLANN filter ($P = 2$)

Input vector	Elements
$X_0(n)$	$x(n), x(n-1), \cdots, x(n-N)$
$X_1(n)$	$\sin[(x(n)], \sin[\pi x(n-1)], ..., \sin[\pi x(n-N)]$
$X_2(n)$	$\cos[\pi x(n)], \cos[\pi x(n-1)], ..., \cos[\pi x(n-N)]$
$X_3(n)$	$\sin[2\pi x(n))], \sin[2\pi x(n-1)], ..., \sin[2\pi x(n-N)]$
$X_4(n)$	$\cos[2\pi x(n)], \cos[2\pi x(n-1)], ..., \cos[2\pi x(n-N)]$

3.3 Even Mirror Function Nonlinear (EMFN) Filter

The FLANN filter is based on the expansion of the trigonometric basis functions. This expansion does not satisfy Stone-Weierstrass theorem [15]. Thus, the FLANN filter cannot perfectly model the nonlinear functions with cross product terms, that is, multiplication terms with different time shift units, because the expansion of FLANN basis function does not contain cross product terms. In order to improve echo path modeling, an even-mirror Fourier nonlinear (EMFN) filter recently reported in [15] can be applied to approximate the input–output relationship of the nonlinear echo path. A third-order EMFN Filter system using the least mean square (LMS) and SEQ-RLS algorithms has been developed for NAEC. Table 3.3.1 lists the EMFN second- and third- order input vectors.

Table 3.3.1. Channel input vectors in the EMFN filter

Input vector	Elements
$X_0(n)$	$x(n), x(n-1), x(n-2), ..., x(n-N)$
$X_1(n)$	$\cos[\pi x(n)], \cos[\pi x(n-1)], ..., \cos[\pi x(n-N)]$
$X_2(n)$	$\sin[\pi x(n)/2]\sin[\pi x(n-1)/2], ...,$ $\sin[\pi x(n-N+1)/2]\sin[\pi x(n-N)/2]$
...	...
$X_{N_2}(n)$	$\sin[\pi x(n)/2]\sin[\pi x(n-N_2+1)/2], ...$
$X_{N_2+1}(n)$	$\sin[3\pi x(n)/2], \sin[3\pi x(n-1)/2], ..., \sin[3\pi x(n-N)/2]$
$X_{N_2+2}(n)$	$\cos[\pi x(n)]\cos[\pi x(n-1)/2],$ $..., \cos[\pi x(n-N+1)]\sin[\pi x(n-N)/2]$
...	...
$X_{N_2+1+P_3}(n)$	$\cos[\pi x(n)]\cos[\pi x(n-P_3)/2],$ $\cos[\pi x(n-1)]\cos[\pi x(n-P_3-1)/2], ...$
$X_{N_2+2+P_3}(n)$	$\sin[\pi x(n)/2]\cos[\pi x(n-1)],$ $..., \sin[\pi x(n-N+1)/2]\cos[\pi x(n-N)]$
...	...
$X_{N_2+1+2P_3}(n)$	$\sin[\pi x(n)/2]\cos[\pi x(n-P_3)],$ $\sin[\pi x(n-1)/2]\cos[\pi x(n-P_3-1)]...$
$X_{N_2+2+2P_3}(n)$	$\sin[\pi x(n)/2]\sin[\pi x(n-1)/2]\sin[\pi x(n-2)/2],$ $\sin[\pi x(n-1)/2]\sin[\pi x(n-2)/2]\sin[\pi x(n-3)/2], ...$ ...
$X_{N_2+N_3}(n)$	$\sin[\pi x(n)/2]\sin[\pi x(n-P_3+1)/2]\sin[\pi x(n-P_3)/2],$ $\sin[\pi x(n-1)/2]\sin[\pi x(n-P_3)/2]\sin[\pi x(n-P_3-1)/2], ...$

CHAPTER 4 COMPUTATIONAL COMPLEXITY

4.1 Sparse Least Mean Square (LMS) Algorithm

Let us denote s_j as the sparsity of channel j at time n, where the sparsity is the ratio of the number of non-zero elements over the total number of elements in the diagonal channel. We can determine the multiplications and additions per iteration for the general S-LMS algorithm. The results are listed in Table 4.1.1 and Table 4.1.2, respectively. To simplify our comparisons, we omit the computation load for generating the first element of input signal in each diagonal channel. Therefore, both Volterra and EMNF algorithms have the same computational complexity.

Table 4.1.1. Multiplications in the sparse LMS Algorithm

Items	Multiplications
$y(n)$	$\sum_{j=0}^{M} s_j N_j$
$e(n)$	None
$W_j(n)$	$\sum_{j=0}^{M} N_j$
Total	$\sum_{j=0}^{M} (s_j + 1) N_j$
Total for $s_j = 1$	$\sum_{j=0}^{M} 2N_j$

Table 4.1.2. Additions in the sparse LMS Algorithm

Items	Additions
$y(n)$	$\sum_{j=0}^{M} (s_j N_j - 1)$
$e(n)$	1
$W_j(n)$	$\sum_{j=0}^{M} N_j$
Total	$\sum_{j=0}^{M} [(s_j + 1) N_j - 1]$
Total for $s_j = 1$	$\sum_{j=0}^{M} (2N_j - 1)$

Note that the sparsity changes at the different channel and iteration. To simplify our estimation by assuming $s_j = 1$ and applying results in Tables 4.1 and 4.2, we achieve the number of multiplications per iteration for the third-order Volterra and EMFN filters as

$$\begin{aligned}\text{Number of multiplications} = 3[2(N+1)\ ] + \left[\sum_{j=1}^{N_2} 2\left(N+1-j\right)\ \right] \\ + 2\left[\sum_{j=1,\ P_3\geq 2}^{P3} 2\left(N+1-j\right)\ \right] + \sum_{j=2,\ P_3\geq 2}^{P3} (j-1)\left[2\left(N+1-j\right)\ \right].\end{aligned} \tag{4.1}$$

To calculate the number of multiplications using the standard LMS algorithm (see Table 4.1.1.), the required total number of elements can be determined by

$$\begin{aligned}N_{T-Voterra/EMFN} = 3(N+1) + \sum_{j=1}^{N_2}\left(N+1-j\right) \\ + 2\sum_{j=1,\ P_3\geq 2}^{P3}\left(N+1-j\right) + \sum_{j=2,\ P_3\geq 2}^{P3}(j-1)\left(N+1-j\right).\end{aligned} \tag{4.2}$$

For the FLANN filter, we yield the load for multiplications as

$$\text{Number of multications}=(M+1)[2(N+1)\ ] \tag{4.3}$$

where $M = 2P$ is the number of channels and P is the order of the FLANN filter. Given the total number of elements in the FLANN filter as

$$N_{T-FLANN} = (M+1)(N+1). \tag{4.4}$$

Figure 4.1.1 shows the comparison of multiplications per iteration versus the filter memory length with $s_j = 1$.

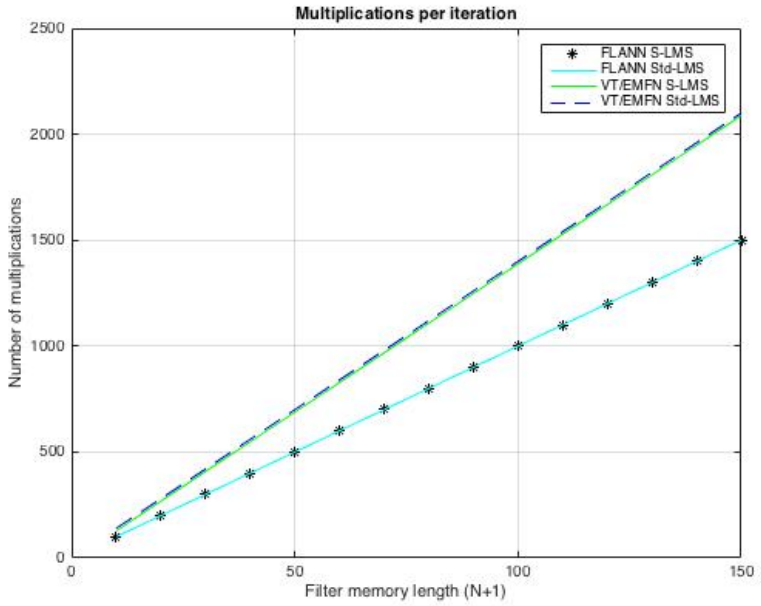

Figure 4.1. Number of multiplications per iteration versus the filter memory length with $s_j = 1$; for the Volterra filter (VT): $N_2 = 2$ and $P_3 = 3$; for the FLAAN filter: $P = 2$; for the EMFNfilter: $N_2 = 2$ and $P_3 = 3$.

Using Table 4.1.2, the required additions for the Volterra/EMFN and FLANN filters can be determined, respectively, and listed below.

$$\text{Number of additions} = 3[2(N)\ ]+\left[\sum_{j=1}^{N_2} 2(N-j)\ \right]+2\left[\sum_{j=1,\ P_3\geq 2}^{P3} 2(N-j)\ \right]+\sum_{j=2,\ P_3\geq 2}^{P3}(j-1)\left[2(N-j)\ \right]. \tag{4.5}$$

For the FLANN filter, we yield the load for multiplications as

$$\text{Number of additions}=(M+1)(2N) \tag{4.6}$$

The number of additions by using the standard LMS algorithm in Volterra, EMFN, and FLANN filters can be determined using the results in Table 4.1.2, Equations (4.2), and (4.4). Figure 4.2 displays the comparisons.

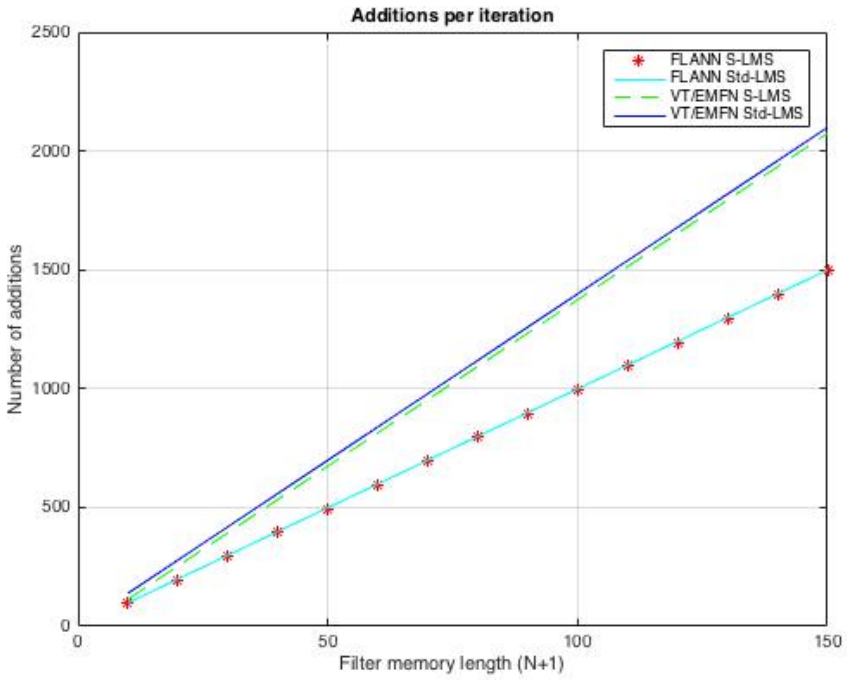

Figure 4.2 Number of additions per iteration versus the filter memory length with $s_j = 1$; for the Volterra filter (VT): $N_2 = 2$ and $P_3 = 3$; for the FLAAN filter: $P = 2$; for the EMFN filter: $N_2 = 2$ and $P_3 = 3$.

From Figures 4.1 and 4.2, we see that the computational complexity using the significant diagonal channels in LMS algorithms versus the memory length are very close. The further reduction of computation can be achieved when the sparsity factor is taken into account.

4. 2 Sparse Sequential RLS Algorithm (S-SEQ-RLS)

By denoting s_j as the sparsity of channel j at time n, similarly, we can determine the multiplications and additions per iteration for the S-SEQ-RLS algorithm. The results are listed in Table 4.2.1 and Table 4.2.2, respectively. To simplify our comparisons, we omit the computation load for generating the first element of input signal in each diagonal channel. Therefore, both Volterra and EMNF algorithms have the same computational complexity.

Table 4.2.1. Multiplications in the S-SEG-RLS algorithm

Items	Multiplications
$k_j(n)$	$\sum_{j=0}^{M}(s_jN_j)N_j+(s_jN_j)^2+2$
$Q_j(n)$	$\sum_{j=0}^{M}(s_jN_j)N_j+(N_j)^2+1$
$y(n)$	$\sum_{j=0}^{M}s_jN_j$
$\alpha(n)$	None
$W_j(n)$	$\sum_{j=0}^{M}N_j$
Total	$\sum_{j=0}^{M}(s_j^2+1)^2N_j^2+(s_j+1)N_j+3$
Total for $s_j=1$	$\sum_{j=0}^{M}4N_j^2+2N_j+3$
Standard RLS	$4N_T^2+2N_T+3,\ N_T=\sum_{j=0}^{M}N_j$

Table 4.2.2. Additions in the S-SEG-RLS algorithm

Items	Additions
$k_j(n)$	$\sum_{j=0}^{M}(s_jN_j-1)N_j+(s_jN_j-1)+1$
$Q_j(n)$	$\sum_{j=0}^{M}(s_jN_j-1)N_j+N_j^2$
$y(n)$	$\sum_{j=0}^{M}(s_jN_j-1)$
$\alpha(n)$	1
$W_j(n)$	$\sum_{j=0}^{M}N_j$
Total	$\sum_{j=0}^{M}(2s_j+1)N_j^2+(2s_j-1)N_j$
Total for $s_j=1$	$\sum_{j=0}^{M}3N_j^2+N_j$
Standard RLS	$3N_T^2+N_T,\ N_T=\sum_{j=0}^{M}N_j$

Note that the sparsity changes at the different channel and iteration. To simplify our estimation by assuming $s_j=1$ and applying results in Table 4.2.1, we achieve the number of multiplications per iteration for the third-order Volterra and EMFN filters as

Number of multiplications =

$$3[4(N+1)^2+2(N+1)+3]+\left[\sum_{j=1}^{N_2}4\left(N+1-j\right)^2+2\left(N+1-j\right)+3\right]$$
$$+2\left[\sum_{j=1,\ P_3\geq 2}^{P3}4\left(N+1-j\right)^2+2\left(N+1-j\right)+3\right] \tag{4.7}$$
$$+\sum_{j=2,\ P_3\geq 2}^{P3}(j-1)\left[4\left(N+1-j\right)^2+2\left(N+1-j\right)+3\right].$$

To calculate the number of multiplications using the standard RLS algorithm (see Table 4.2.1.), the required total number of elements can be determined

$$N_{T-Voterra/EMFN}=3(N+1)+\sum_{j=1}^{N_2}\left(N+1-j\right)+2\sum_{j=1,\ P_3\geq 2}^{P3}\left(N+1-j\right)+\sum_{j=2,\ P_3\geq 2}^{P3}(j-1)\left(N+1-j\right). \tag{4.8}$$

For the FLANN filter, we yield the load for multiplications as

$$\text{Number of multications}=(M+1)[4(N+1)^2+2(N+1)+3] \tag{4.9}$$

where $M=2P$ is the number of channels and P is the order of the FLANN filter. Given the total number of elements in the FLANN filter as

$$N_{T-FLANN}=(M+1)(N+1) \tag{4.10}$$

We can determine the complexity of multiplications via Table 4.2.1. Figure 4.1 depicts the multiplication complexity for each nonlinear chancellor versus the filter memory length of $(N+1)$.

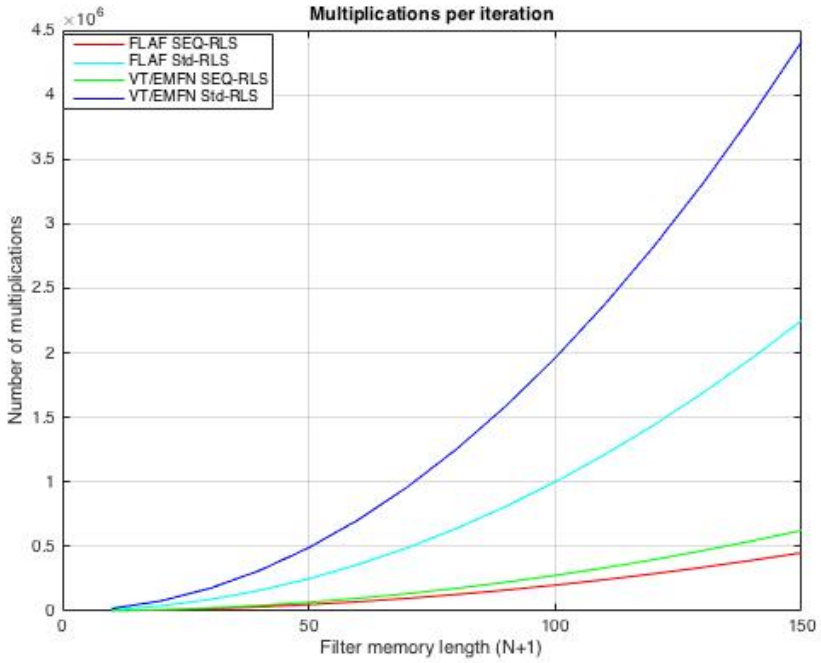

Figure 4.3. Number of multiplications per iteration versus the filter memory length with $s_j = 1$; for the Volterra filter (VT): $N_2 = 2$ and $P_3 = 3$; for the FLAAN filter: $P = 2$;for the EMFN filter: $N_2 = 2$ and $P_3 = 3$.

Using Table 4.2.2, the required additions for the Volterra/EMFN, and FLANN filters can be determined, respectively, and listed below.

$$\begin{aligned}&\text{Number of additions(Volterra/EMFN)} = \\ &3[3(N+1)^2+(N+1)]+\left[\sum_{j=1}^{N_2} 3\left(N+1-j\right)^2+\left(N+1-j\right)\right] \\ &+2\left[\sum_{j=1,\ P_3\geq 2}^{P3} 3\left(N+1-j\right)^2+\left(N+1-j\right)\right] \\ &+\sum_{j=2,\ P_3\geq 2}^{P3}(j-1)\left[3\left(N+1-j\right)^2+\left(N+1-j\right)\right]\end{aligned} \tag{4.11}$$

$$\text{Number of additions (FLANN)} = (M+1)[3(N+1)^2+(N+1)]. \tag{4.12}$$

The number of additions by using the standard RLS algorithm in Volterra, EMFN, and FLANN filters can be determined using the result in Table 4.2.2, Equations (4.11), and (4.12). Figure 4.4 displays their comparisons.

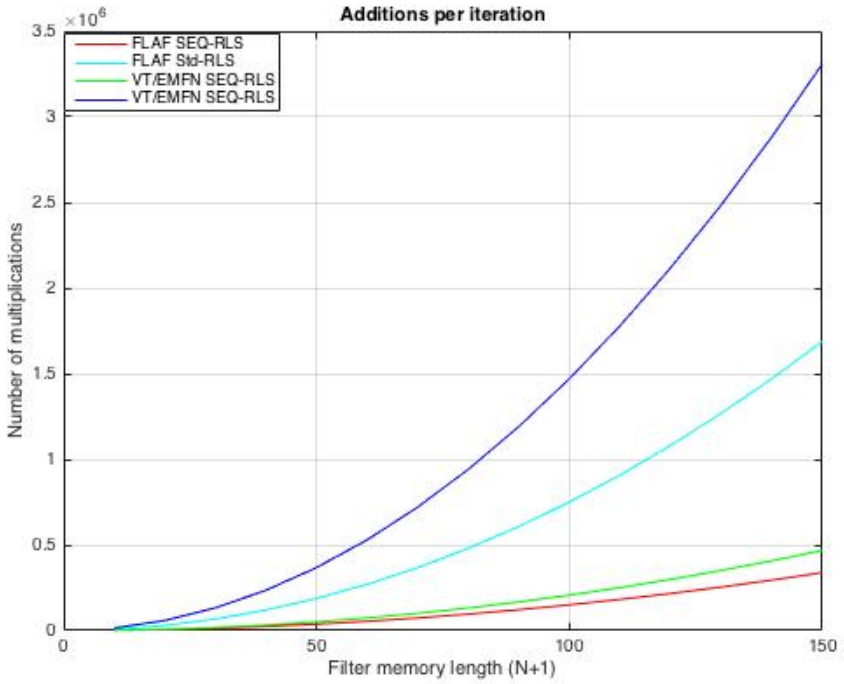

Figure 4.4. Number of multiplications per iteration versus the filter memory length with $s_j = 1$; for the Volterra filter (VT): $N_2 = 2$ and $P_3 = 3$; for the FLAAN filter: $P = 2$; for the EMFN filter: $N_2 = 2$ and $P_3 = 3$.

From Figures 4.3 and 4.4, we see that using the significant diagonal channels and SEQ-RLS algorithm can significantly reduce the computational load when the filter memory filter length increases. Clearly, the further reduction of computation is achieved when the sparsity factor is taken into account.

CHAPTER 5 SIMULATION AND RESULTS

5.1 System Identification

To validate the developed algorithms, we first perform system identification for a non-linear system given in Equations (5.1) and (5.2), and compare the performances with the Volterra, FLANN and EMFN adaptive filters using the sparse LMS and SEQ-RLS algorithms respectively. Next, we further investigate the effect of sparsity.

To begin with, we assume that the echo path is presented by the following system:

$$\begin{aligned} d(n) &= 0.3x(n-1) + 0.5x(n-3) - 0.4\cos[\pi x(n)] + \cos[\pi x(n-3)] \\ &- \sin[\pi x(n-1)/2]\sin[\pi x(n-2)/2] + \sin[3\pi x(n)/2] \\ &-0.9\sin[3\pi x(n-2)/2] + 0.5\cos[\pi x(n)]\sin[\pi x(n-1)/2] \\ &-0.6\sin[\pi x(n-1)/2]\cos[\pi x(n-3)] \\ &+ \sin[\pi x(n-1)/2]\sin[\pi x(n-2)/2]\sin[\pi x(n-3)/2] + v(n) \end{aligned} \tag{5.1}$$

and

$$\begin{aligned} d(n) &= 0.3x(n-1) + 0.5x(n-3) - 0.4x^2(n) + x^2(n-3) \\ &-x(n-1)x(n-2) + x^3(n) - 0.9x^3(n-2) + 0.5x^2(n)x(n-1) \\ &-0.6x(n-1)x^2(n-3) + x(n-1)x(n-2)x(n-3) + v(n) \end{aligned} \tag{5.2}$$

where $x(n)$ is the uniformly distributed white noise and $v(n)$ is the random noise with a Gaussian distribution. The signal to noise power ratio (SNR) of 30 dB is used for all our simulations. Each adaptive filter with a memory size of 10 is adopted.

We obtain the normalized mean square error (NMSE) for performance comparisons. The NMSE is ensemble over 100 runs versus the number of iterations defined below:

$$NMSE = 10\log_{10}\left(\frac{E\{e^2(n)\}}{\sigma_d^2}\right) \tag{5.3}$$

where σ_d^2 is the power of signal $d(n)$.

For all the simulations, we use the following parameters in the algorithms:

RLS algorithm: $\lambda = 1 - 0.1/(N+1)$; LMS algorithm: $\mu = 0.01$; Volterra filter: $N_2 = 2$ and $P_3 = 3$; FLANN filter: $P = 2$; EMFN filter: $N_2 = 2$ and $P_3 = 3$.

Figures 5.1 and 5.2 show the plots of NMSEs for system in Equation (5.1) and system in Equation (5.2), respectively. It can be seen that the EMFN filter with a sequential RLS algorithm and LMS algorithm are the best performing systems for Equation (5.1) while the Volterra filter with a sequential RLS algorithm and volterra LMS algorithm perform the best for identifying system in Equation (5.2). The EMFN filter performs the best when the system is the even-mirror Fourier expanded memory system while the Volterra filter works the best for the polynomial expanded memory system.

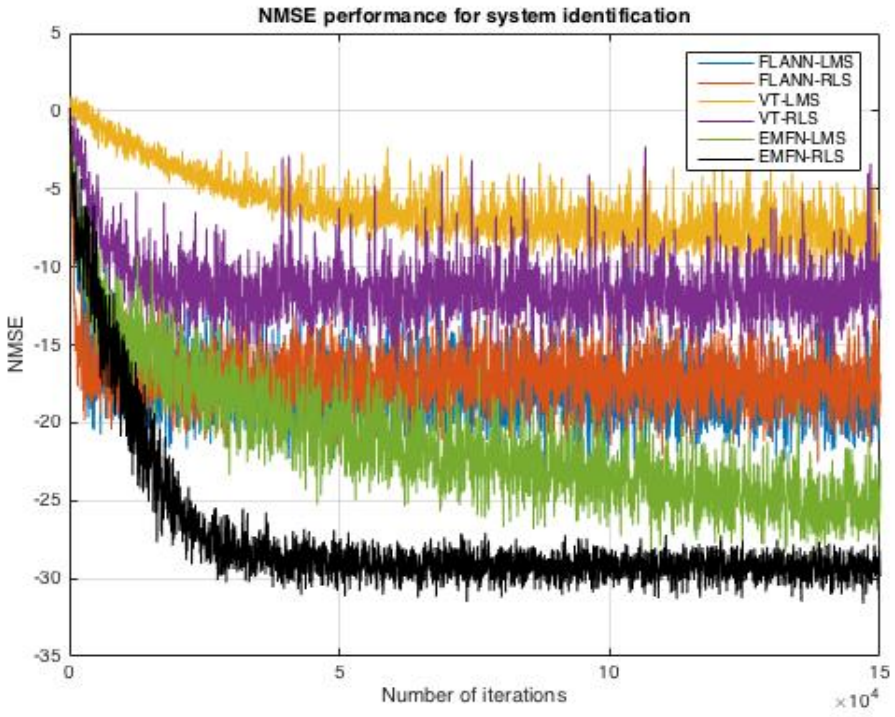

Figure 5.1. NMSE performance comparisons of system identification for Equation (5.1) with LMS and Sequential-RLS algorithms.

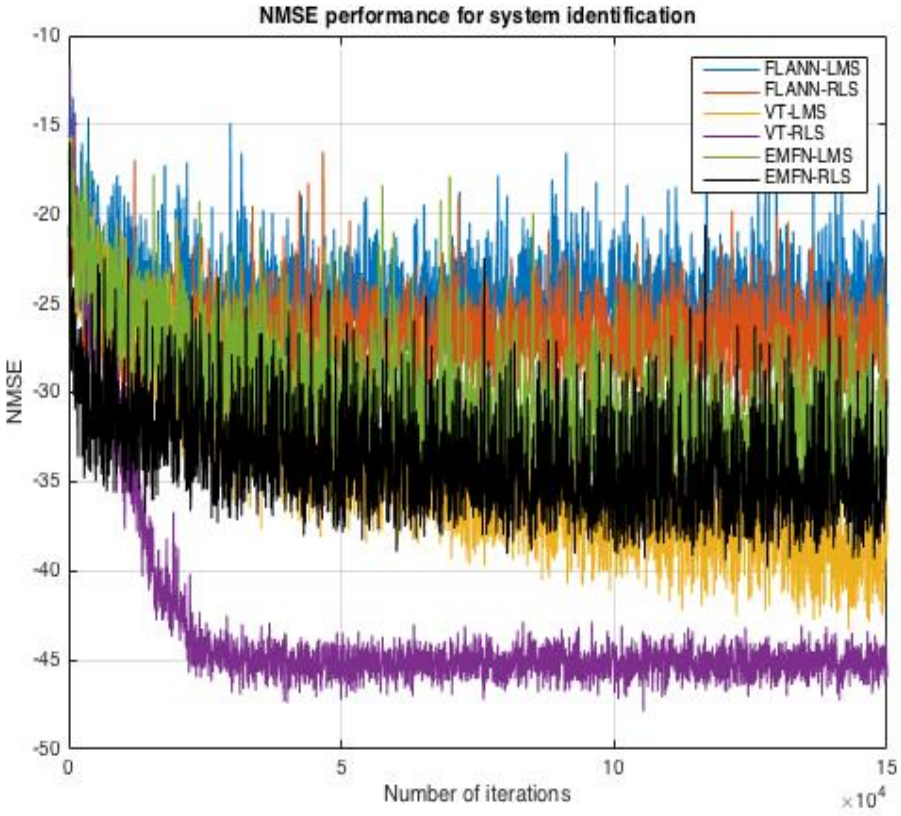

Figure 5.2. NMSE performance comparisons of system identification for Equation (5.2) with LMS and Sequential-RLS algorithms.

5.2 Effect of Sparsity

The effect of sparsity in these two systems is investigated by plotting the NMSE for various values of ε (threshold value for the discard function) ranging from 0.00001 to 0.1. Figures 5.3-5.6 depict the plots for the sparse-SEQ-RLS and sparse-LMS algorithms. It is evident that the sparse-SEQ-RLS and sparse-LMS algorithms have large performance degradation when the threshold of ε is larger than 0.001 in the discard function. The EMFN filter is more sensitive to the threshold value since the system model is the EMFN type.

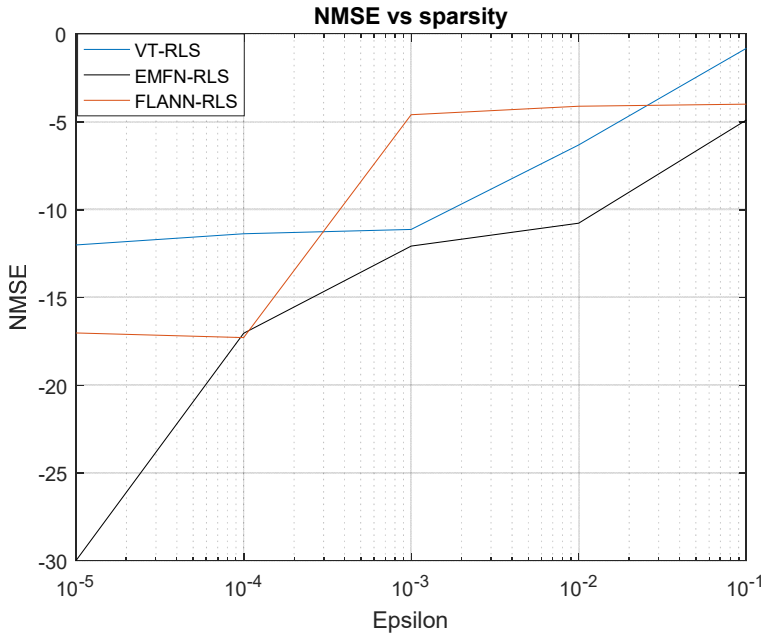

Figure 5.3 NMSEs versus ε (threshold in the discard function) for the RLS based algorithms for Equation (5.1).

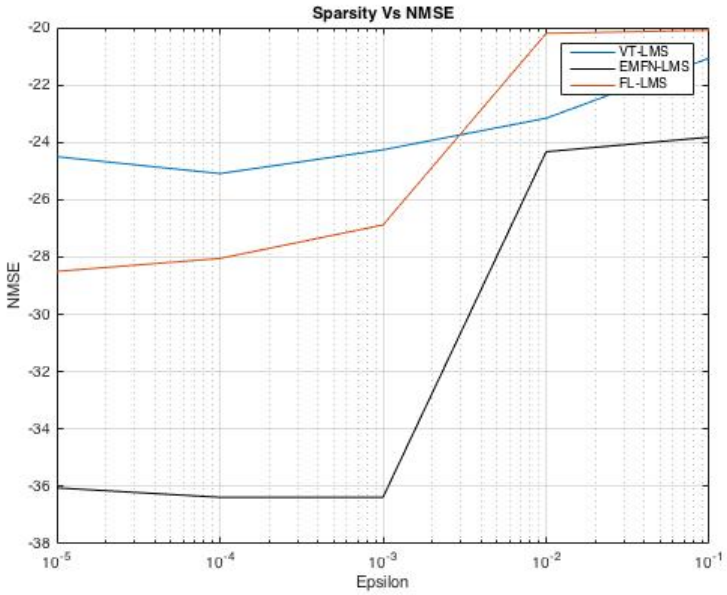

Figure 5.4 NMSEs versus ε (threshold in the discard function) for the LMS based algorithms for Equation (5.1).

A similar trend can be seen in the NMSE performance comparison as shown in Figure 5.4. Since the system model is Volterra type, the Volterra filter is more sensitive to the threshold value.

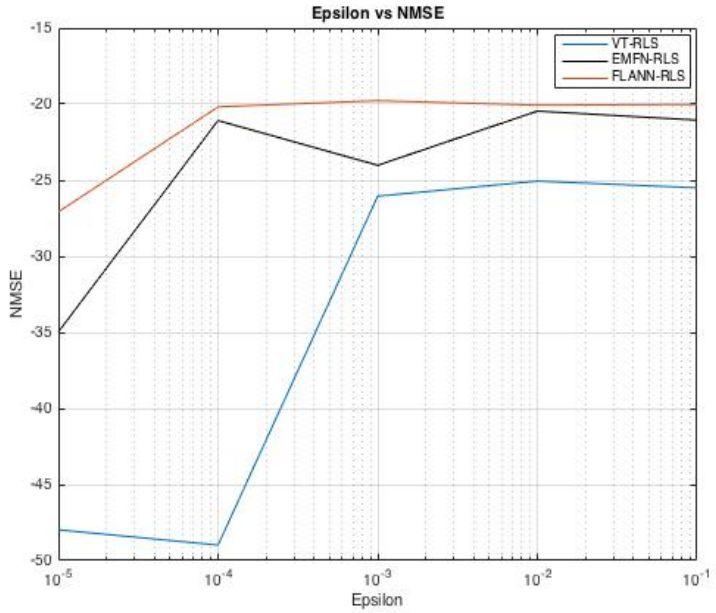

Figure 5.5 NMSEs versus ε (threshold in the discard function) for the RLS based algorithms for Equation (5.2).

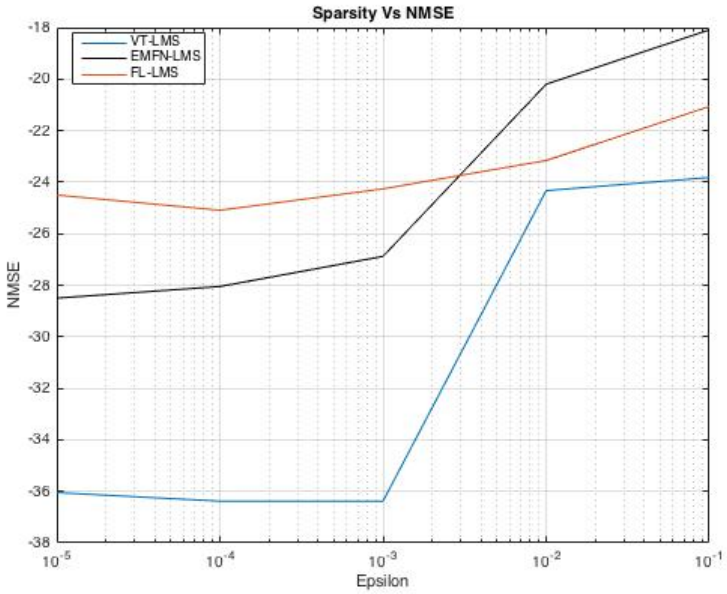

Figure 5.6 NMSEs versus ε (threshold in the discard function) for the LMS based algorithms for Equation (5.2).

5.3 Nonlinear Echo Cancellation

For nonlinear echo cancellation, we adopt the far end and near end speech segments with a sampling rate of 8000 Hz shown in Figure 5.7.

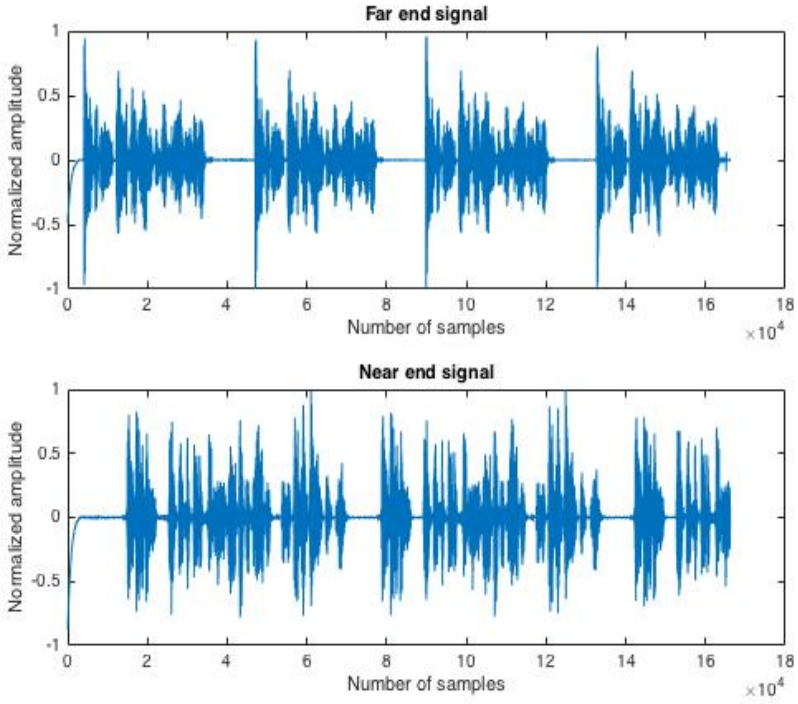

Figure 5.7. Speech signal used in the simulation (Fs=8000 Hz).

The nonlinear echo path consists of the nonlinear function and linear acoustic impulse response (AIR) echo path shown in Figure 5.8, which is a 150th order polynomial model extracted from the AIR multichannel impulse response database [21], [22].

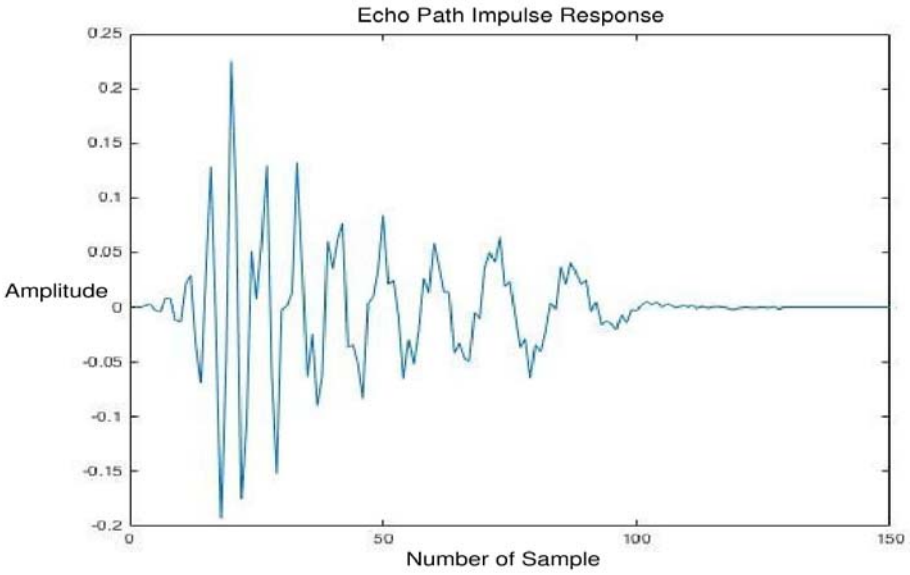

Figure 5.8. Echo cancellation path (FIR model).

For performance evaluation, the ERLE, ensemble over 100 runs versus the number of iterations defined in Equation (5.4) is used:

$$ERLE = 10\log_{10}\left(\frac{E\{d^2(n)\}}{E\{e^2(n)\}}\right) \tag{5.4}$$

Note that $d(n)$ is the echo to be cancelled while $e(n)=d(n)-y(n)$ is the residual signal from the nonlinear echo cancellation.

5.3.1 Single Talk Simulations for Memoryless Nonlinear System

The memoryless nonlinearity is introduced before the linear echo path model, which is a piecewise nonlinear system defined below:

$$f(x)=\begin{cases} 2x/(3\xi) & |x|\le\xi \\ sign(x)\left(3-(2-|x/\xi|)^2\right)/3\right) & \xi<|x|\le 2\xi \\ sign(x) & 2\xi<|x|\le 1 \end{cases} \tag{5.5}$$

where ξ is the controlling factor and is set to 0.4. The system input and output relation is shown in Figure 5.9.

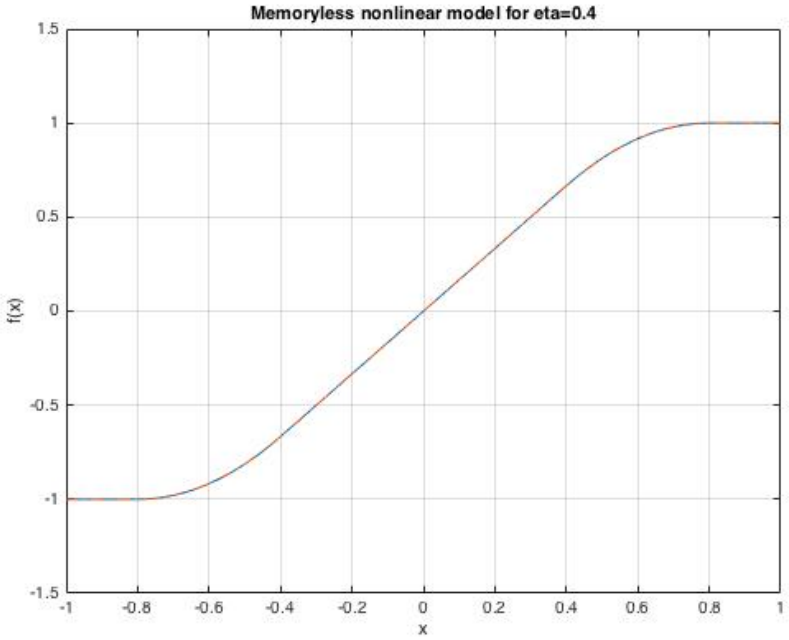

Figure 5.9. Input and output of the nonlinear system without memory.

In our single talk scenario, a SNR of 30 dB is adopted at the near end. The performance of ERLEs are obtained and displayed in Figures 5.10 and 5.11 for the S-LMS and S-SEQ-RLS algorithms, respectively.

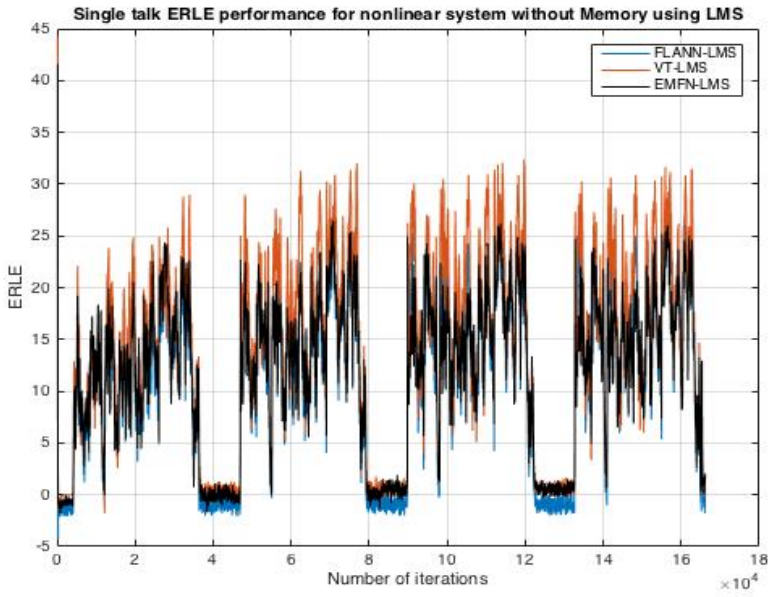

Figure 5.10. ERLE performance comparisons of single talk scenario with S-LMS algorithm and memoryless system.

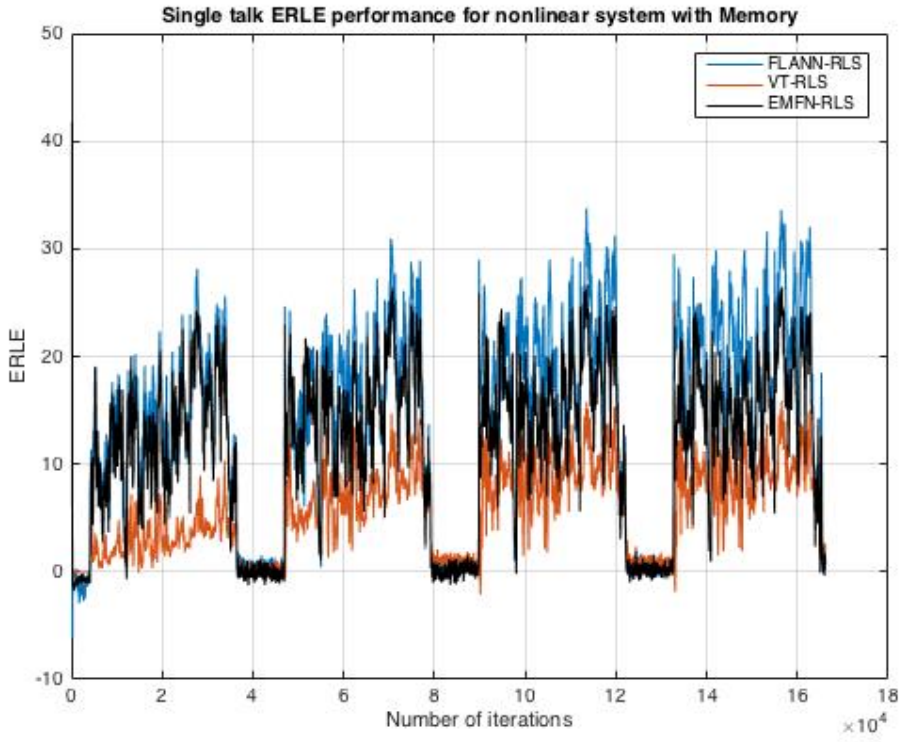

Figure 5.11. ERLE performance comparisons of single talk scenario with S-SEQ-RLS algorithm and memoryless system.

From Figures 5.10 and 5.11, it is evident that the third-order Volterra filter using the S-LMS gives a best performance when compared with that of the third-order Volterra filter using the S-SEQ-RLS algorithm performance. This might be due to the fact that the S-SEQ-RLS algorithm is a sub-optimal and is also due to the large size of Q matrix. The other two filter systems using both the algorithms give a similar performance. The S-SEQ-RLS algorithm provides the expected results for a memoryless nonlinearity, i.e., the FLANN giving the best performance followed by the EMNF and third-order VT filters.

5.3.2 Single Talk Simulations for Nonlinear System with Memory

The memory nonlinearity is introduced before the linear echo path model, which is a piecewise nonlinear system with the cross product term up to the third-order components defined below.

$$f[x(n)]=\begin{cases} 2x(n)/(3\xi) & |x(n)|\le\xi \\ sign[x(n)]\left(3-\dfrac{(2-|x(n)/\xi|)^2}{3}+\dfrac{6}{5}x(n)x(n-1)\right. & \xi<|x(n)|\le 2\xi \\ \left.-\dfrac{1}{2}x(n)x(n-2)+\dfrac{1}{2}x(n)x(n-1)x(n-2)\right) & \\ sign[x(n)] & 2\xi<|x(n)|\le 1 \end{cases} \tag{5.6}$$

The system input and output relation is shown in Figure 5.12.

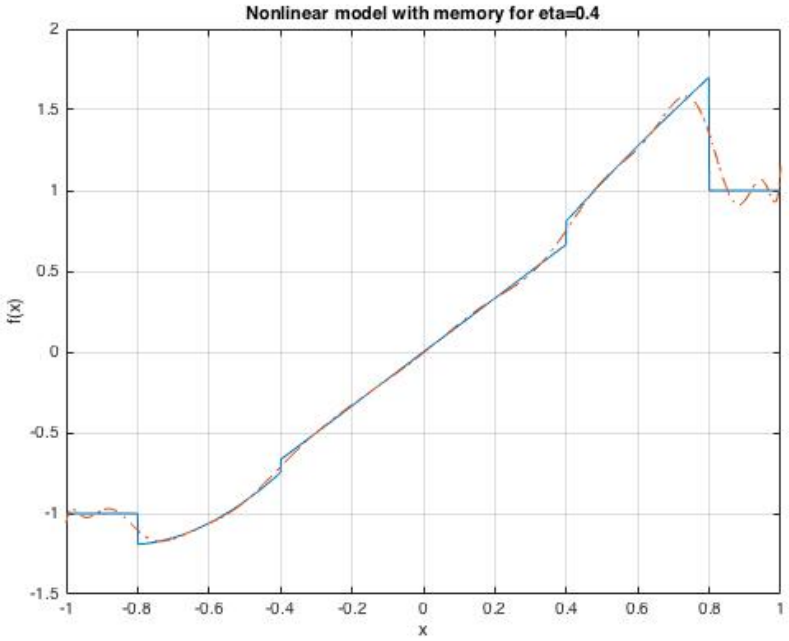

Figure 5.12. Input and output of the nonlinear system with memory.

In our single talk scenario, a SNR of 30 dB is adopted at the near end. The performance of ERLEs are obtained as displayed in Figures 5.13 and 5.14 for the S-LMS and S-SEQ-RLS algorithms, respectively.

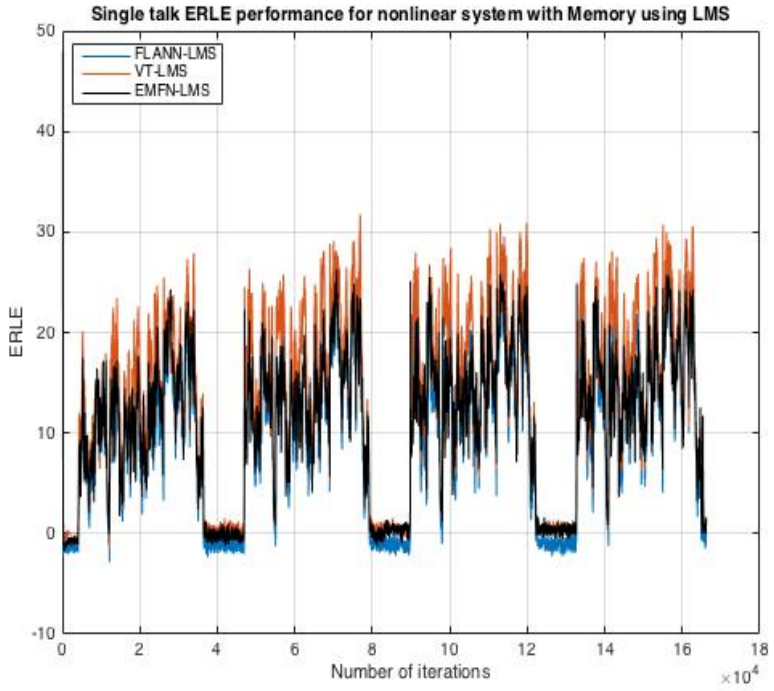

Figure 5.13. ERLE performance comparisons of single talk scenario with the S-LMS algorithm and Memory system.

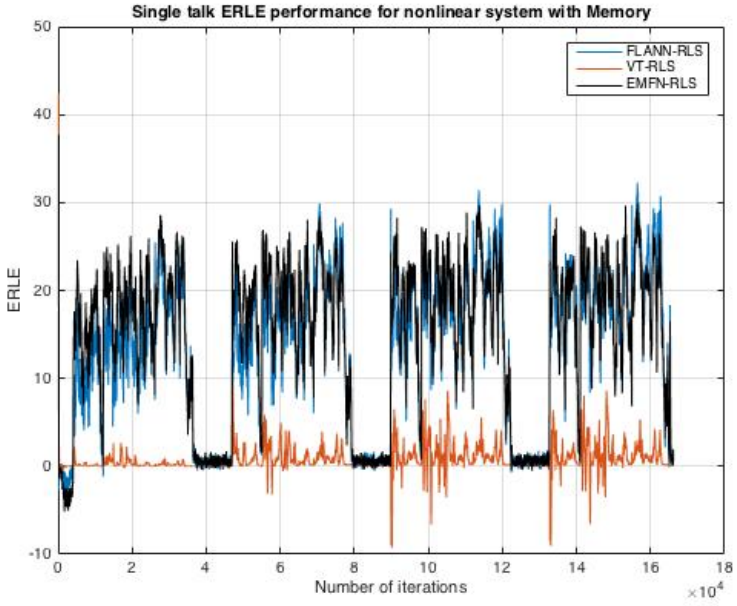

Figure 5.14. ERLE performance comparisons of single talk scenario with the S-SEQ-RLS algorithm and memory system

As shown in Figures 5.13 and 5.14, it is evident that the third-order volterra filter using the S-LMS algorithm gives a much better performance when compared with that of third-order

Volterra filter using the S-SEQ-RLS algorithm. The S-SEQ-RLS algorithm gives a better performances from the EMFN and FLANN filter system over the Volterra filter. Also, the S-LMS algorithm provides the expected result for a memory nonlinearity, i.e. third-order Volttera filter offers the best performance followed by EMNF and FLANNfilters.

5.3.3 Double Talk Simulations for Memoryless Nonlinear System

In this simulation, we repeat the previous experiments by using the same echo path in the double talk situation. The parameter settings are the same as those in the single talk case. The residue signal, that is, $e(n)-s(n)$, which is the difference between echo cancelled signal and near end signal is examined. Figure 5.15 depicts the far end, near end and observed signals (echo+near end signal) for the nonlinear ACE without memory.

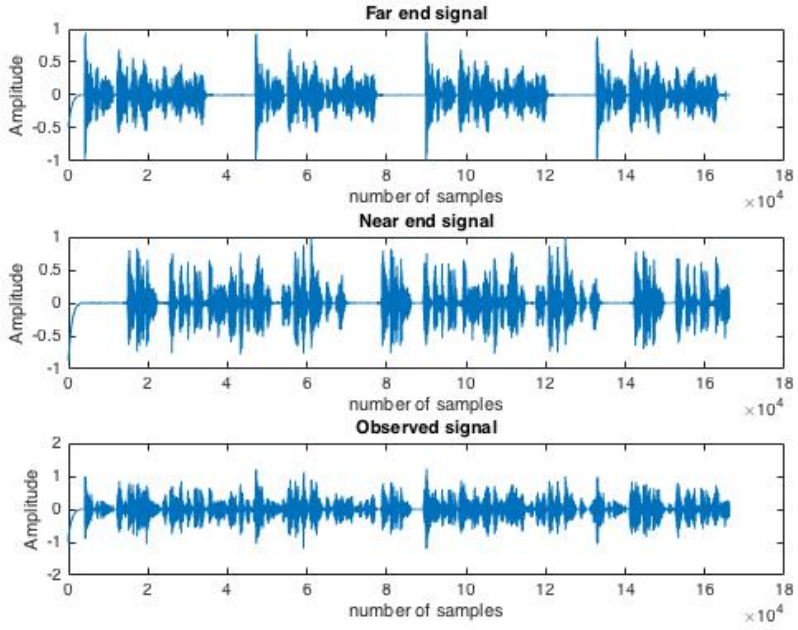

Figure 5.15. Far end, near end and observed signals
for the nonlinear ACE without memory.

According to the plotted residual signals, the method which gives the minimum amplitude is the best performing NAEC system for the double talk scenario.

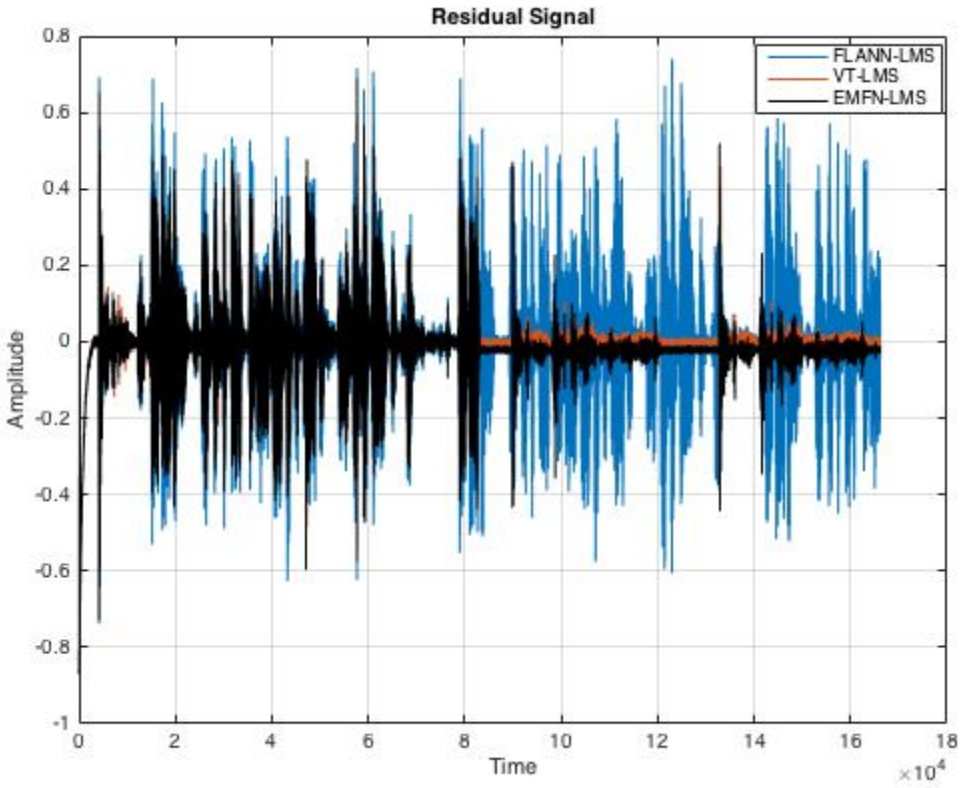

Figure 5.16. Residual signal of memoryless system with S-LMS algorithm.

From the residual signal plot in Figure5.16. It can be noted that the EMFN filter system has the lowest amplitude when compared with the other two filter systems. Hence, the EMFN adaptive filter works best for the double talk scenario for memoryless nonlinear system using the S-LMS algorithm.

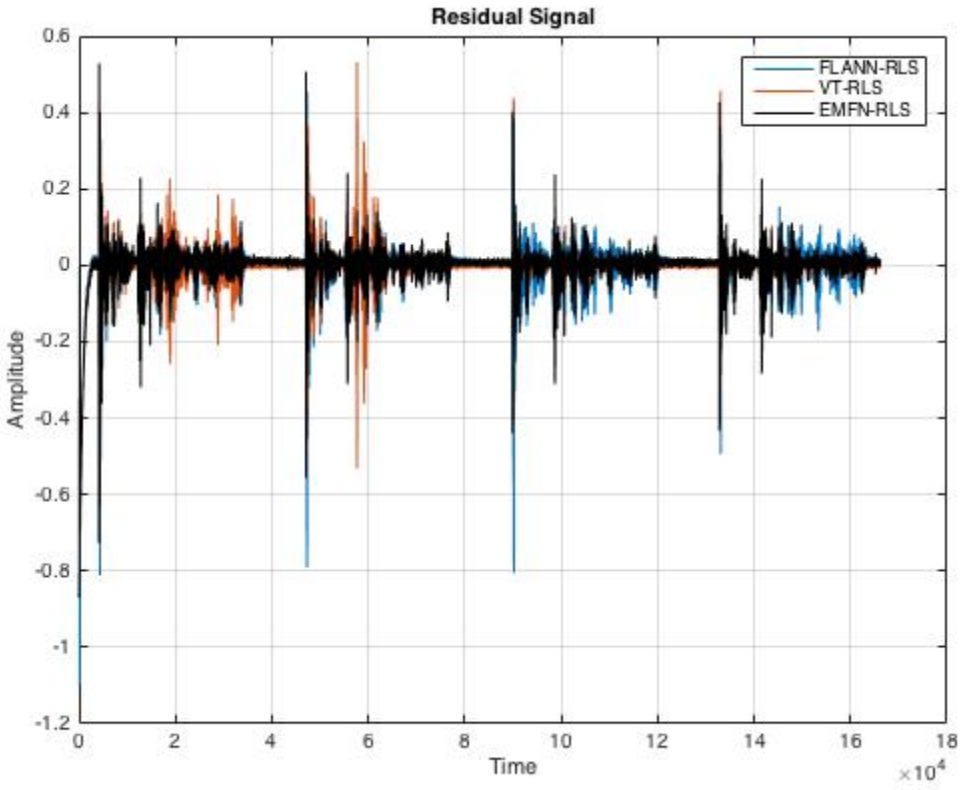

Figure 5.17. Residual signal of memoryless system with the S-SEQ-RLS algorithm.

From the residual signal plot as shown in Figure 5.17. We see that the EMFN and third-order Volterra filter systems have the small amplitude. Clearly, the EMFN adaptive filter works best for the double talk scenario for the memoryless nonlinear system using the S-SEQ_RLS algorithm.

5.3.4 Double Talk Simulations for Memory Nonlinear System

In this simulation, we repeat the previous experiments by using the same echo path in the double talk situation. The parameter settings are the same as those in the single talk. The residue signals ($e(n)-s(n)$) are plotted and investigated. Figure 5.18 depicts the far end, near end and observed signals (echo+near end signal) for the nonlinear ACE with memory.

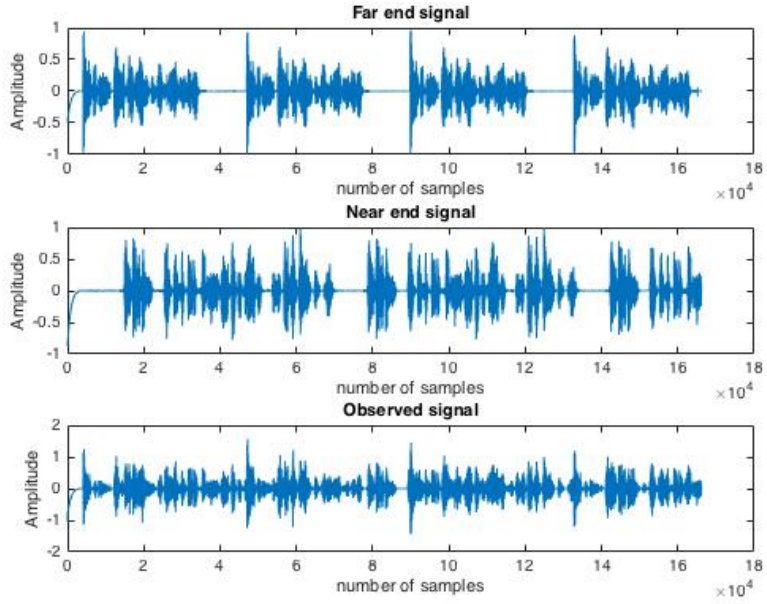

Figure 5.18. Far end, near end and observed signals for the nonlinear ACE with memory

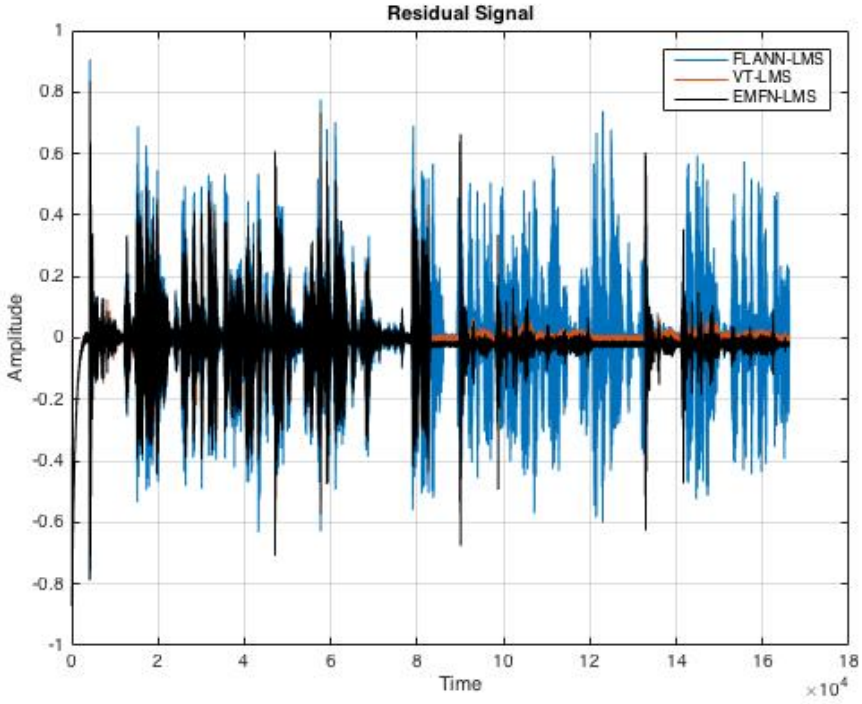

Figure 5.19. Residual signal of memory system with the S-LMS algorithm.

From the residual signal plot in Figure 5.19. We observe that the EMFN filter system has the lowest amplitude when compared with the other two filter systems. The same conclusion can be drawn as for the single talk case for the memory nonlinear system using the S-LMS algorithm.

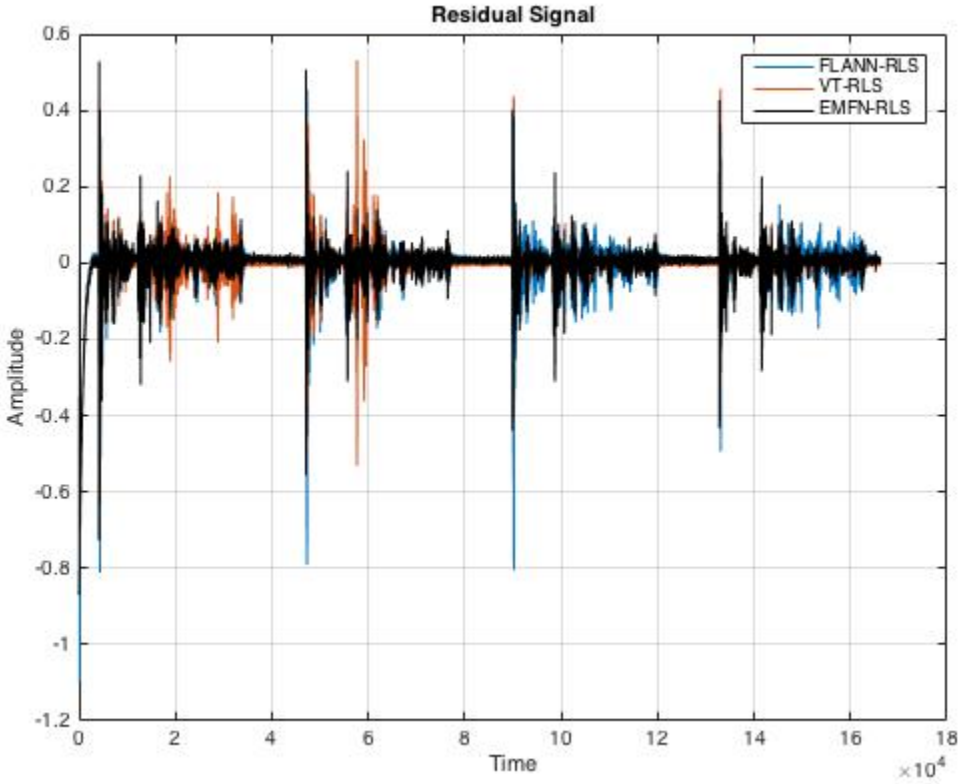

Figure 5.20. Residual signal of memory system with the S-SEQ-RLS algorithm.

From the residual signal plot in Figure 5.20. It can be seen that the EMFN and third-order Volterra filter systems have small residue amplitude. Furthermore, the EMFN adaptive filter works best for double talk scenario for the memory nonlinear system using the S-LMS algorithm.

CHAPTER 6 CONCLUSION AND FUTURE WORK

In this thesis, the significance and problem of echo cancellation are introduced in Chapter 1 and the general Wiener solution is described. In Chapter 2, the sparse-LMS and sparse sequenctail -RLS algorithms have been developed. In chapter 3, the Volterra, FLAAN and EMFN adaptive filter structures are introduced and corresponding algorithms are developed by applying the sparse LMS algorithm and sparse sequenctial RLS algorithm. The computation complexity for each nonlinear filter is derived. Their computational loads are compared. In chapter 5, the simulations are performed for the single talk and double scenarios and the all the performances are validated. It is concluded that the EMFN adaptive filter offers the performance by either using the sparse LMS algorithm or sparse sequential RLS algorithm. But Volterra filter gives close performance in comparison with the EMFN filter by using the sparse LMS algorithm. Future work could involve developing a convex combination of adaptive filters for system identification and eventually apply it to for nonlinear acoustic echo.

REFERENCES

[1] L. Tan, J. Jiang, Digital Signal Processing: Fundamentals and Applications. Third Edition, Elsevier/Academics, 2018.

[2] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters," IEEE Int. Conf. on Acoustics, Speech & Signal Processing (ICASSP), pp. 877–880, Phoenix, AZ, March 1999.

[3] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, "Nonlinear acoustic echo cancellation based on Volterra filters," IEEE Trans. Speech Audio Proc., vol. 11, no. 6, pp. 672–683, November 2003.

[4] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-Garcia, A. Uncini, "Full proportionate functional link adaptive filters for nonlinear acoustic echo cancellation," 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1145-1149, Kos, Greece, August 2017.

[5] D. Comminiello, M. Scarpiniti, L. A. Azpicueta-Ruiz, J. Arenas-García, A. Uncini, Member, "Nonlinear accoustic echo cancellation based on sparse functional link representation," IEEE Transactions/ACM on Audio, Speech, and Language Processing, vol. 22, no. 7, pp. 1172–1183, July 2014.

[6] Y. Huang, J. Skoglund, A. Luebs, "Practically efficient nonlinear acoustic echo cancelling using cascaded block RLS and FLMS adaptive filters," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.597-600, New Orleans, LA, March 2017.

[7] L. Tan, J. Jiang, "Adaptive Volterra filters for active control of nonlinear noise processes," IEEE Trans. Signal Processing, Vol 49, No. 8, pp. 1667-1676, August 2001.

[8] D. P. Das, G. Panda, “Active mitigation of nonlinear noise process using a novel filtered-s LMS algorithms,” IEEE Trans. Speech, Audio Processing, vol. 12, no. 3, pp. 313-322, May 2004.

[9] L. Tan, V. Vijayarajan, N. Chimitt, J. Jiang, A. Togbe, “Channel sparsity-aware recursive least squares algorithms for nonlinear system modeling and active noise control,” 2017 IEEE 8th Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON 2017), pp. 225-231, Columbia University, New York City, NY, October 2017.

[10] V. Vijayarajan, J. Dai, L. Tan, J. Jiang, “Channel sparsity-aware diagonal structure Volterra filters for nonlinear acoustic echo cancellation,” 2018 IEEE International Conference on Electro/Information Technology, pp. 420-423, Oakland University, Rochester, Michigan, May 2018.

[11] L. Tan, J. Jiang, “Adaptive second-order Volterra filtered-x RLS algorithms with sequential and partial updates for nonlinear active noise control,” IEEE 4th International Conference on Industrial Electronics and Applications, pp.1625-1630, Xian, China, May 2009.

[12] L. Tan, “Adaptive function expansion RLS filters with dynamic selection of channel updates for nonlinear active noise control,” 2010 International Conference on Intelligent Control and Information Processing, Vol. 2, pp.1-6, Dalian, China, August 2010.

[13] L. Tan, J. Jiang, “Adaptive second-order Volterra RLS algorithms with dynamic selection of channel updates,” 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp.1323-1328, Montreal, Canada, July 2010.

[14] X. Guo, Y. Li, J. Jiang, C. Dong, S. Du, L. Tan, “Sparse modeling of nonlinear secondary path for nonlinear active noise control,” IEEE Trans. on Instrumentation and Measurement, vol. 67, No. 3, pp. 482-496, March 2018.

[15] A. Carini, G. L. Sicuranza, “Perfect periodic sequences for even mirror Fourier nonlinear filters,” Signal Processing, vol. 104, pp. 80-93, 2014.

[16] X. Guo, C. Dong, L. Tan, S. Du, “Adaptive even mirror Fourier filtered error LMS algorithm for multichannel nonlinear active noise control,” IEEE, Information Technology, Electronics and Mobile Communication Conference. IEEE, pp. 1-6, 2016.

[17] L. Tan, J. Jiang, “An adaptive technique for modeling second-order Volterra systems with sparse kernels,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 45, No. 12, pp. 1610-1615, December 1998.

[18] L. Tan, J. Jiang, “Adaptive second-order Volterra delay filter,” IEE Electronics Letters, Vol. 32, No.9, pp. 807-809, April 1996.

[19] Y. Chen, Y. Gu, and A.O. Hero, “Sparse LMS for system identification,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3125–3128, Taipei, Taiwan, April 2009

[20] H. Yazdanpanah, and P. S. R. Diniz, “Recursive least-squares algorithms for sparse system modeling,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3879–3883, New Orleans, March 2017.

[21] Rwth Aachen University (Germany) and Bar-Ilan Uni- versity (Israel), “Multichannel impulse response database,” http://http://www.ind.rwth-aachen. de/en/research/tools-downloads/ multichannel-impulse-response-database/, 2014, [Online; Latest Accessed 16-Sep-2015].

[22] Digital Network Echo Cancellers, ITU-T Rec. G.168, 2002.

VITA

Vinith Vijayarajan was born in Madurai, India. He obtained his Bachelor of Engineering (B.E) in Electronics and communication engineering from MEPCO Schlenk engineering college, Sivakasi, India, in 2016. In December 2018, he obtained his MS degree in Electrical Engineering, from Purdue University Northwest, Indiana.

PUBLICATIONS

Conference Papers

V. Vijayarajan, J. Dai, L. Tan, J. Jiang, “Channel Sparsity-Aware Diagonal Structure Volterra Filters for Nonlinear Acoustic Echo Cancellation” 2018 IEEE INTERNATIONAL CONFERENCE ON ELECTRO INFORMATION TECHNOLOGY. PP, 420-423, Oakland University, Rochester, Michigan, May 2018

J. Dai, V. Vijayarajan, X. Peng, L. Tan, J. Jiang, “Speech recognition using sparse discrete wavelet decomposition feature extraction” 2018 IEEE INTERNATIONAL CONFERENCE ON ELECTRO INFORMATION TECHNOLOGY, pp. 812-816, Oakland University, Rochester, Michigan, May 2018.

L. Tan, V. Vijayarajan, N. Chimitt, J. Jiang, A. Togbe, “Channel Sparsity-Aware Recursive Least Squares Algorithms for Nonlinear System Modelling and Active Noise Control” 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)-2017, pp. 225-231, Columbia university, New York City,NY,October 2017.