[UCI]I804:24011-200000241869

2011년 8월

석사학위 논문

Modified Adaptive Algorithm for a Sparse Reconfigurable Adaptive Filter

조선대학교 대학원

첨단부품소재공학과

Chang Hong

Modified Adaptive Algorithm for a Sparse Reconfigurable Adaptive Filter

Sparse Reconfigurable 적응 필터를 위한
변형된 적응 알고리즘

2011년 8월 25일

조선대학교 대학원

첨단부품소재공학과

Chang Hong

Modified Adaptive Algorithm for a Sparse Reconfigurable Adaptive Filter

지도교수 황 석 승

이 논문을 공학 석사학위신청 논문으로 제출함

2011년 4월

조선대학교 대학원

첨단부품소재공학과

Chang Hong

상홍의 석사학위논문을 인준함

위원장　　조선대학교 교수　　이 동 기 (인)

위　원　　조선대학교 교수　　심 재 기 (인)

위　원　　조선대학교 교수　　황 석 승 (인)

2011년 5월

조선대학교 대학원

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Sparse Reconfigurable 적응 필터를 위한 변형된 적응 알고리즘

Chang Hong

Advisor: Prof. Hwang Suk-seung, Ph.D.

Department of Advanced parts and Materials Engineering,

Graduate School of Chosun University

스파스 재설정 적응 필터(SRAF, Sparse Reconfigurable Adaptive Filter)는 입력지연장치, 재설정 연결기와 적응 가중치들을 포함하는 광스위치, 그리고 출력지연장치로 구성되어 있다. 이 시스템 구조는 다수의 적응 가중치들과 다양한 시간 지연기들로 구성된 스파스 탭이 부착된 지연선(TDL, Tapped-Delay-Line)을 구동하기 위하여 사용될 수 있다. 적응 알고리즘들은 광스위치를 위한 SRAF 사용 시에 핵심적인 역할을 한다. 본 논문에서는 SRAF에 대한 성능을 수학적으로 분석할 수 있는 변형된 시스템-기반 (MSB, Modified System-Based) 알고리즘을 제안한다. MSB 적응 알고리즘은 백색 입력 신호들과 비백색 입력신호들 모두에 대하여 좋은 성능을 가질 뿐만 아니라, 빠른 수렴 속도에 의해 낮은 복잡도를 가지므로 상호-상관-기반(CCB, Cross-correlation-based)이나 시스템-기반(SB, System-Ba- sed)과 같은 일반적인 알고리즘보다 효율적이라고 할 수 있다. 알고리즘의 수렴 속도를 개선시키기 위해 MSB는

행렬구조를 가지는 가중치들의 각 행 또는 열을 독립적으로 적응시킨다. 또한, 본 논문에서는 MSB를 위해 요구되는 중간 학습 신호들의 구조제시한다.

최대 행렬 요소들을 순차적으로 선택하는 방법을 기반으로 하는 일반적인 연결 알고리즘은 SRAF가 요구하는 연결 제한 조건을 수행할 시 동일한 절대 가중치 값들의 합이 존재하면 완벽한 성능을 기대할 수 없다. CCB와 SB의 경우를 위해 이러한 문제점 해결을 위한 시스템 식별 정확도를 계량할 하고, 일반적인 알고리즘 보다 개선된 솔루션을 얻는 것을 목적으로 진보된 계산을 사용하는 개선된 연결 제한 알고리즘을 제안한다. 또한, MSB 적응 알고리즘의 시스템 식별 성능 항상을 위한 개선된 연결 제한 알고리즘을 제시한다.

지연 연결기 들의 선택을 목적으로 하는 가중치 벡터를 계산하기 위한 일반적인 알고리즘들은 LMS(Least-Mean-Square) 알고리즘에 국한되어 있지만, 본 논문에서는 시스템 식별을 위한 수렴 속도를 향상시키기 위해 SB 알고리즘에 RLS(Recursive Least-Squares) 적응 알고리즘을 사용하고 그에 따른 결과를 제시한다.

I. INTRODUCTION

A. Research Overview

In this thesis, we focus on adaptive system identification methods for sparse reconfigurable adaptive filter (SRAF) [1], but first we need to review some important details about the sparse filter. A sparse filter is a large TDL that has relatively few and possibly widely spaced nonzero coefficients. The design of the filter has to continually adapt to the changing environment [2]. One of the first adaptive finite-impulse-response (FIR) filters with a sparse impulse response was studied in [3]-[7]. An interpolated FIR (IFIR) filter based on a sparse filter and an interpolator was proposed in [8] and [9] to reduce the computational complexity of a conventional FIR filter when the number of coefficients is large. An improved adaptive IFIR (AIFIR) structure using the least-mean-square (LMS) algorithm [10] was presented in [11], and the double AIFIR (DAIFIR) filter was proposed in [12]. The Haar transform was considered in [13] and [14] to reduce the computational complexity and improve the convergence rate for the identification of a sparse impulse response. Recently, the simplified signed sparse LMS (SSSLMS) algorithm, which is a special case of the adaptive natural gradient algorithm [15], was proposed in [16] and [17]. For a sparse system, the generalized subband decomposition (GSD) structure and the block exact fast affine projection (BEFAP) algorithm were also considered in [18] and [19], respectively. Many sparse adaptive filters have previously been studied, but they do not utilize the unique matrix architecture of the nonblocking optical switch considered here.

Although TDL filters have previously been considered for optical fiber technology [20], previous systems have been limited by the small size of the optical switches. Recently developed three-dimensional (3-D) microelectronmechanical system (MEMS) optical switches have overcome this limitation [21]-[24] with sizes up to 1200×1200, so that many applications are now possible. The SRAF considered here is based on a photonic switch with an input and output connection architecture that can be represented by a matrix of adaptive weights. These tap weights can be represented by a sparse matrix (size $N \times N$) with the constraint that at most only one element in each row and column is nonzero; the nonzero weights combine the input and output delays so that up to N^2 different time delays are possible.

In previous work [1], a cross-correlation-based (CCB) algorithm for selecting the specific switch connections was investigated. In the CCB algorithm and the similar approach in [25], the connections are determined by computing a cross-correlation function between the input and a desired response signal that depends on the particular application. The algorithm chooses the connections to maximize the norm of the cross-correlation vector, and the LMS algorithm [26]-[29] is applied to compute the weight values for the chosen delays. Although this approach has good performance for white input signals, it may not find the best delay combinations when the input signal is non-white. In order to compensate for this limitation, an algorithm that is motivated by the system identification formulation presented in [1], which we refer to as the system-based (SB) approach [30], was also be considered. This algorithm first adapts the weights using the LMS algorithm and then chooses appropriate delay combinations to have improved performance for non-white input signals compared with the basic CCB algorithm.

In order to improve the convergence rate of the algorithm, a modified system-based (MSB) algorithm whose performance can be understood mathematically is considered in this paper. The MSB separately updates each row or column of the weights with the structure of a matrix. The LMS algorithm is also be utilized to calculate weight values for choosing delay connections.

Finally, we extend the use of the method of least squares (LS) [31]-[33] to develop a recursive algorithm for the design of adaptive transversal filters [34] such that, given the least-squares estimate of the tap-weight vector of the filter at $(N-1)\mathrm{th}$ iteration, we may compute the updated estimate of this vector at $N\mathrm{th}$ iteration upon the arrival of new data. We refer to the resulting algorithm as the recursive least-squares (RLS) algorithm [35]. The RLS algorithm may be viewed as a special case of the Kalman filter [36]-[39]. An important feature of the RLS algorithm is that it utilizes information contained in the input data, extending back to the instant of time when the algorithm is initiated. The resulting rate of convergence is therefore typically faster than the LMS algorithm. This improvement in performance, however, is achieved at the expense of a large increase in computational complexity.

B. Thesis Organization

The structure of this paper is organized as follows. In section Ⅱ, we present the MSB algorithm for the SRAF to improve the convergence rate of the adaptive algorithm. An implementation of the connection constraint for the MSB algorithm is discussed in section Ⅲ, for increasing the accuracy of adaptive algorithms, an upgraded connection constraint algorithm for the CCB and SB to choose the optimal connection as the same values exist when computing the summation of the largest weights is proposed. Moreover, another upgraded connection constraint algorithm, which can improve the performance of the MSB algorithm, is also presented. In section Ⅳ, a brief introduction of the RLS adaptive algorithm for the SB to compute adaptive weight vector is presented. Finally, the conclusion of this study is summarized in section Ⅴ.

II. MSB ADAPTIVE ALGORITHM

A. Introduction

The adaptive algorithm [40] used by the SRAF chooses the appropriate time delays and computes the weight values of the optical switch according to the specific application. In previous work [1], we investigated the CCB algorithm for selecting the specific switch connections. The CCB approach has good performance for white input signals, but it may not find the optimal delay combinations for non-white input signals. In order to overcome this problem, a SB approach in [30] based on a system identification formulation that adapts the weights and chooses the appropriate delay combinations and has good performance for white and non-white input signals was investigated.

In this paper, the MSB adaptive algorithm whose performance can be understood mathematically for the SRAF, and has faster convergence rate than conventional CCB and SB algorithms, has been presented. The connection constraint algorithm for the CCB and SB adaptive algorithms considers the entire $N \times N$ weight matrix when selecting a subset of N values. The MSB adaptive algorithm initially uses N values and considers other values only if the current weight value matches previously chosen values. Thus, the convergence rate of the MSB is faster and the computational complexity of enforcing the connection constraint of the MSB algorithm is less than them of the conventional algorithms such as CCB and SB. It, also, has good performance for white and non-white input signals. Considering separately updates each row/column of weight matrix to improve the convergence rate, a special signal structure for the reference signals is presented for this algorithm. The properties of the proposed SRAF algorithm are demonstrated by computer simulation for a system identification application.

B. Signal Model for the Switch

1. Conventional Signal Model

A set of input and output signals for the $N \times N$ switch represented by the weight matrix for the SRAF can be defined by

$$\mathbf{x}(k) \triangleq [x_1(k), \dots, x_N(k)]^T \tag{1.1}$$

$$\mathbf{y}(k) \triangleq [y_1(k), \dots, y_N(k)]^T \tag{1.2}$$

where k is the discrete-time index, shown in Figure 2.1. Delays at the input and output of the switch can be represented by the following matrices

$$\mathbf{U}_I(z^{-1}) \triangleq [z^{-m_1}, \dots, z^{-m_N}]^T \tag{1.3}$$

$$\mathbf{U}_O(z^{-1}) \triangleq [z^{-n_1}, \dots, z^{-n_N}]^T\ . \tag{1.4}$$

The switch weight matrix $\mathbf{W}(k)$ connects the elements of $\mathbf{x}(k)$ and $\mathbf{y}(k)$ such that at most there is only one nonzero element in each row and each column. Combining these definitions, the overall output can be written in terms of the input as follows

$$y(k) = \mathbf{U}^T{}_O(z^{-1})\mathbf{W}(k)\mathbf{U}_I(z^{-1})x(k) \tag{1.5}$$

where z^{-1} in this time-domain expression is the delay operator (i.e., $z^{-1}x(k) = x(k - 1)$).

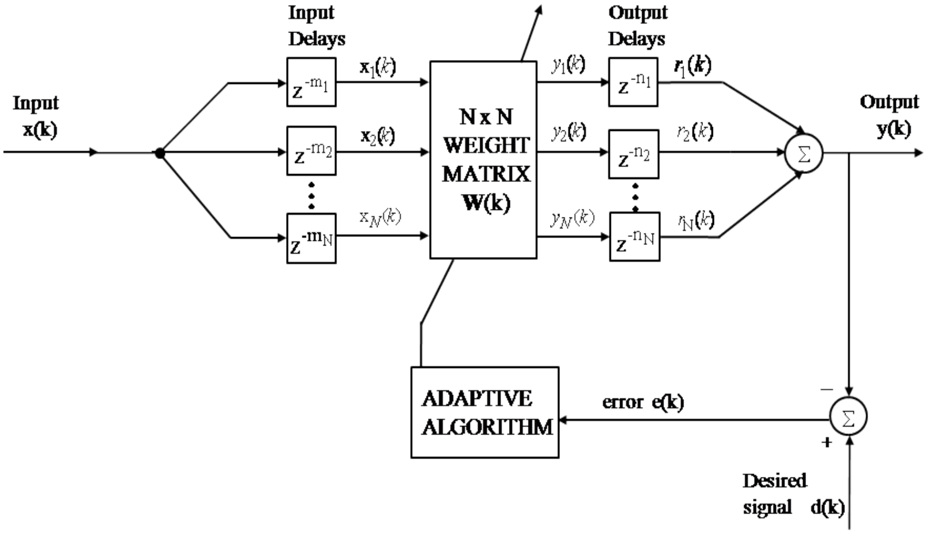


Figure 2.1: Architecture of sparse reconfigurable adaptive filter (SRAF).

The output error can be written as:

$$e(k) \triangleq d(k) - y(k) \tag{1.6}$$

where $d(k)$ is the desired signal. The LMS algorithm for computing the extended weight vector is

$$\mathbf{w}(k+1) \triangleq \mathbf{w}(k) + 2\mu \boldsymbol{r}(k) e(k) \tag{1.7}$$

where $\mu > 0$ is the step-size parameter for controlling the convergence properties of the system, and with the regression vector $\boldsymbol{r}(k)$ given by:

$$
\begin{aligned}
\boldsymbol{r}(k) &\triangleq \mathbf{U}_O(z^{-1})\mathbf{W}(k)\mathbf{U}_I(z^{-1})\mathbf{1}x(k) \\
&= \mathbf{U}_O(z^{-1})\mathbf{W}(k)\mathbf{x}(k) \\
&= \mathbf{U}_O(z^{-1})\mathbf{y}(k) \qquad (1.8)
\end{aligned}
$$

where $\mathbf{1} \triangleq [1, \dots, 1]^T$ is of size *N*.

2. Specific Structure of the Switch

The extended weight vector $\widetilde{\mathbf{w}}$ (size N^2) (we ignore the time argument (k) in this chapter) can be written as:

$$\widetilde{\mathbf{W}} = \left[\omega_{1,1}, \dots, \omega_{1,N}, \omega_{2,1}, \dots, \omega_{2,N}, \dots, \omega_{N,1}, \dots, \omega_{N,N},\right]^T \qquad (1.9)$$

For convenience, (1.9) can be rewritten in the form of a weight matrix as

$$\widetilde{\mathbf{W}} = [\mathbf{w}_1, \dots, \mathbf{w}_\mathrm{N}] \qquad (1.10)$$

where $\mathbf{w}_j \triangleq \left[\omega_{1,j}, \dots, \omega_{N,j}\right]^T$. The subscript i of $\omega_{i,j}$ corresponds to the ith input switch and the subscript j refers to the jth output switch. A block diagram of the MSB adaptive algorithm is show in Figure 2.2. Obviously, the main difference between the SB and MSB is that we have to separately update each column vector $\mathbf{w}_j$ using the LMS algorithm with the corresponding input signal vector and desired response signal. Thus, the input vector for the MSB can be written as:

$$\mathbf{x}_j(k) \triangleq \left[x_{1,j}(k), \dots, x_{N,j}(k)\right]^T. \tag{1.11}$$

Observe that $x_{i,j}(k) = x(k - m_i - n_j)$ where m_i and n_j $(i, j = 1,2,\cdots,N)$ denote the ith input and jth output delay, respectively. And the corresponding output vector is:

$$y_j(k) = \mathbf{w}_j^T \mathrm{x}_j(k). \tag{1.12}$$

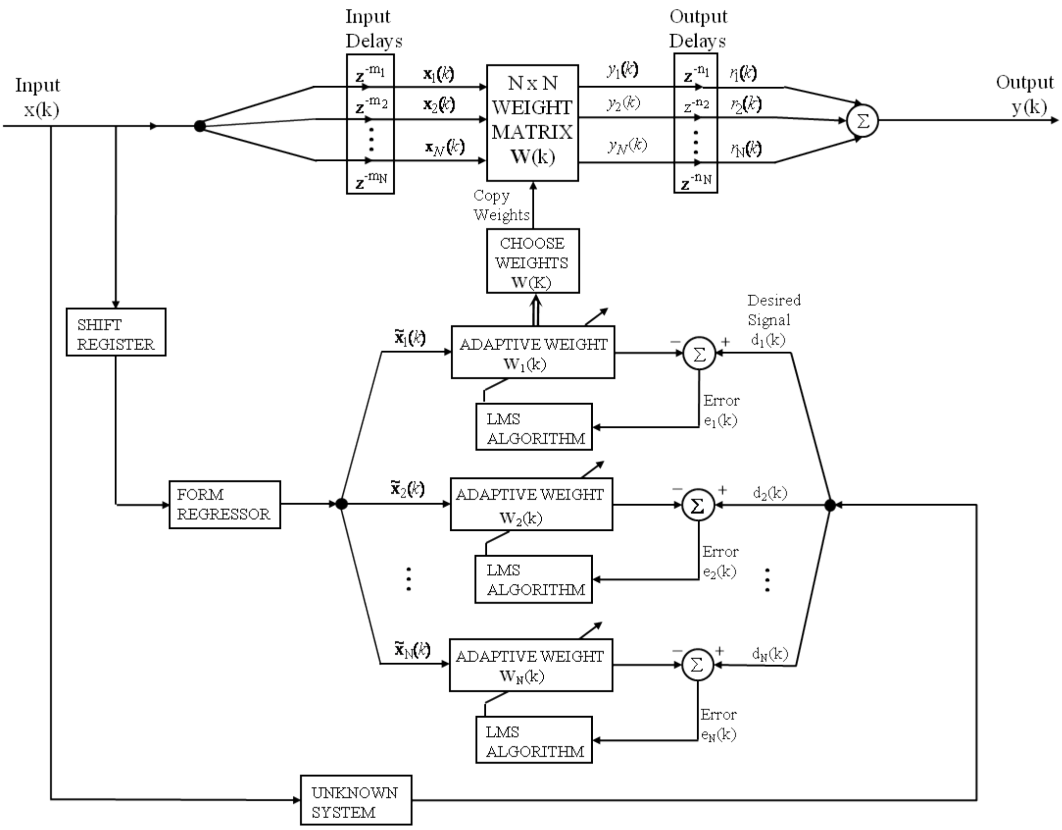


Figure 2.2: Modified system-based (MSB) adaptive algorithm.

In order to obtain the specific desired signal for updating the $j\mathrm{th}$ column vector $\mathbf{w}_j$ of $\widetilde{\mathbf{W}}$, we have to arrange the weight matrix to process the input signals to include appropriate input signal vector $\mathbf{x}_j(k)$ that only corresponding to the $j\mathrm{th}$ output $\mathrm{y}_j(k)$, and make other column input signal values with zero instead. This process can be expressed by a special structure as shown in Figure 2.3. For example, if the λth (λ = 1,2,⋯, N) column of the weight matrix is updated, (the input delays are $m_i = 1,2,\cdots,N$, and the output delays are $n_j = 0, N, \cdots, (N-1)N$) the data structure is shown in Figure 2.3(b). Meanwhile, the structure states of (λ-1) and (λ+1) columns are also represented by Figures 2.3(a) and 2.3(c), respectively. The shaded segments shown in Figure 2.3(b) are the reference data to adaptively update λth column of the weight matrix.

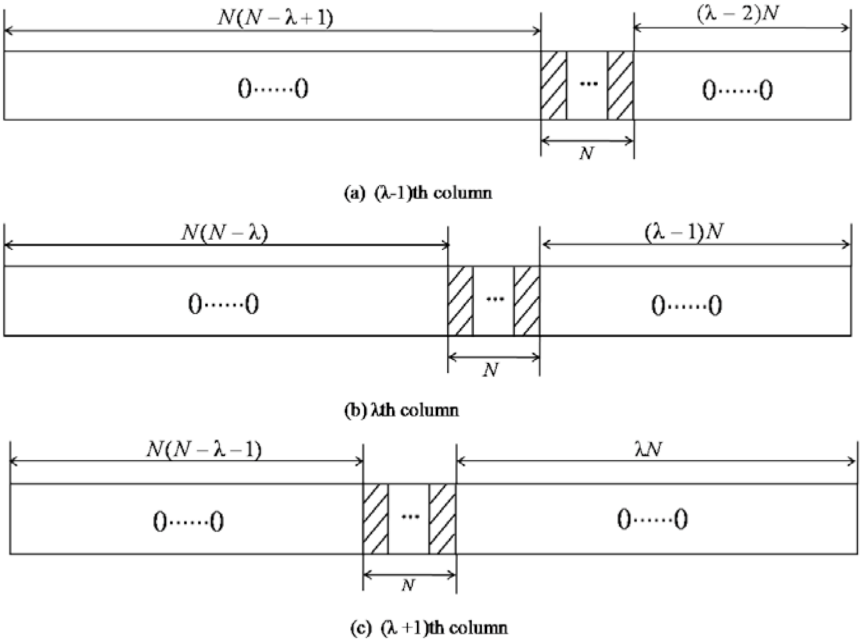


Figure 2.3: Structure of intermediate desired signals for MSB algorithm.

C. MSB Adaptive Algorithm

Using the specific structure discussed previously with intermediate desired signal $d_j(k)$, the output error of the system can be written as

$$e_j(k) \triangleq d_j(k) - y_j(k) \tag{1.13}$$

and the mean-square-error (MSE) cost function is

$$\begin{aligned} \alpha_j &\triangleq E\left[e_j^2(k)\right] \\ &= \sigma_{d\,j}^2 - 2\mathbf{w}_j^T\mathbf{P}_j + \mathbf{w}_j^T\mathbf{R}_j\mathbf{w}_j \end{aligned} \tag{1.14}$$

where $\mathbf{p}_j \triangleq E\left[\mathbf{x}_j(k)d_j(k)\right]$ is the cross-correlation vector, $\mathbf{R}_j \triangleq E\left[\mathbf{x}_j(k)\mathbf{x}_j^T(k)\right]$ is the input signal autocorrelation matrix, and $\sigma_{d\,j}^2$ is the variance of the desired signal $d_j(k)$ (assuming that it has '0' mean). Differentiating α_j with respect to $\mathbf{w}_j$, setting the result equal to the zero vector, and solving for $\mathbf{w}_j$, we can obtain the optimal weights $\mathbf{w}_{j,o} = \mathbf{R}_j^{-1}\mathbf{p}_j$, the minimum MSE (MMSE) is given by the equation

$$\begin{aligned} \alpha_{j,min} &= \sigma_{d_j}^2 - \mathrm{p}_j^T\mathbf{R}_j^{-1}\mathbf{p}_j \\ &= \sigma_{d_j}^2 - \mathbf{w}_{j,o}^T\mathbf{R}_j\mathbf{w}_{j,o}\,. \end{aligned} \tag{1.15}$$

Since from the optimal weights $\mathbf{w}_{j,o}$ we can choose only one weight element (due to the switch constraint), (1.15) can be rewritten as

$$\alpha_{j,min} = \sigma_{d_j}^2 - \omega_{j,o}^2\sigma_j^2 \tag{1.16}$$

where $\alpha_{j,o}$ is one of the elements in $\mathbf{w}_{j,o}$ and σ_j^2 is the variance of $\mathbf{x}_j(k)$ (assuming that $\mathbf{x}_j(k)$ has '0' zero mean). Since σ_j^2 is a constant value for both white and non-white input signals, the lowest $\alpha_{j,\min}$ only depends on $\omega_{j,o}^2$. From this expression, it is obvious that $\varepsilon_{j,\min}$ achieves the lowest minimum when $\omega_{j,o}^2\sigma_j^2$ is maximum. To make $\alpha_{j,\min}$ have the lowest minimum value for any input signal, we have to choose the maximum $\omega_{j,o}$ after each update of the adaptive algorithm.

From this discussion, we can summarize the MSB adaptive algorithm as follows.

1). Adapt weight vector $\mathbf{w}_j$ with size of *N* from the weight matrix $\widetilde{\mathbf{W}}$ using the LMS algorithm. The LMS algorithm for computing the weight vector is

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + 2\mu\mathbf{x}_j(k)e_j(k) \tag{1.17}$$

with $e_j(k)$ given by (1.13).

2). Choose the largest weight (magnitude) of $\mathbf{w}_j$ subject to the connection constraint (discussed in the next section).

3). Continue this process until N weights have been selected.

4). Copy these selected nonzero weights to the proper locations in the $N \times N$ switch weight matrix $\mathbf{W}(k)$.

D. Computer Simulation

In this section, we present computer simulation example to demonstrate the performance of the MSB algorithm for implementing the filter. Parks-McClellan algorithm is used to generate a linear-phase FIR bandpass filter with 64 coefficients. The impulse and frequency responses of the system are shown in Figure 2.4. It is obvious that the impulse response is symmetric about the center tap as expected since this filter has the linear-phase characteristic and we can observe the passband over the normalized frequency range [0.25 0.3], stop-bands in the range [0 0.2] and [0.35 1], and the transition bands in the range [0.2 0.25] and [0.35 1]. The magnitude of the frequency response shows that the stop-band is 30-40 dB lower than the passband.

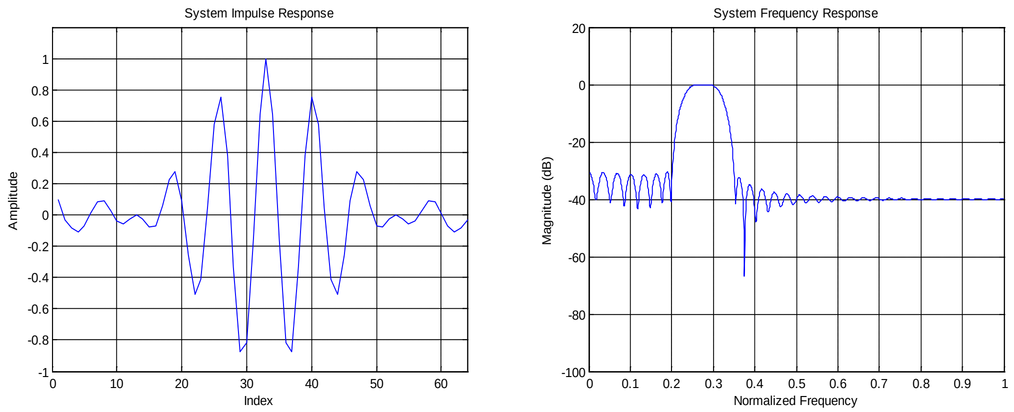


Figure 2.4: Impulse response and frequency response of the actual system with 64 coefficients.

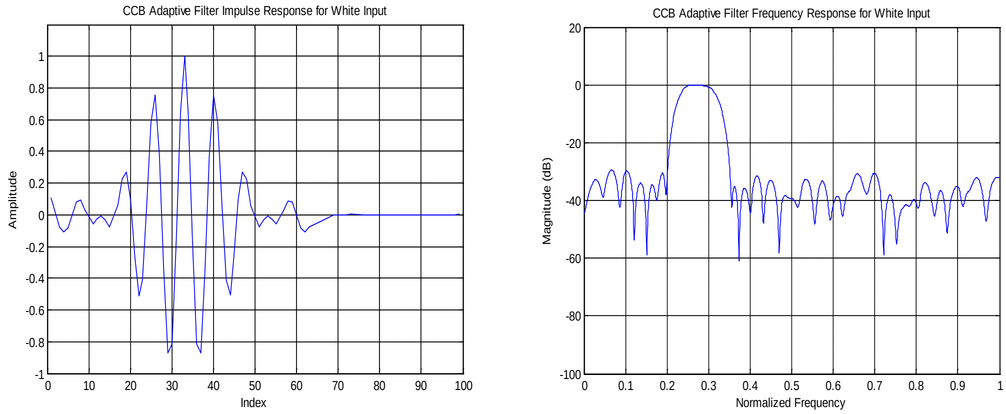


Figure 2.5: Impulse response and frequency response of the CCB adaptive system with 64 nonzero coefficients for a white input signal.

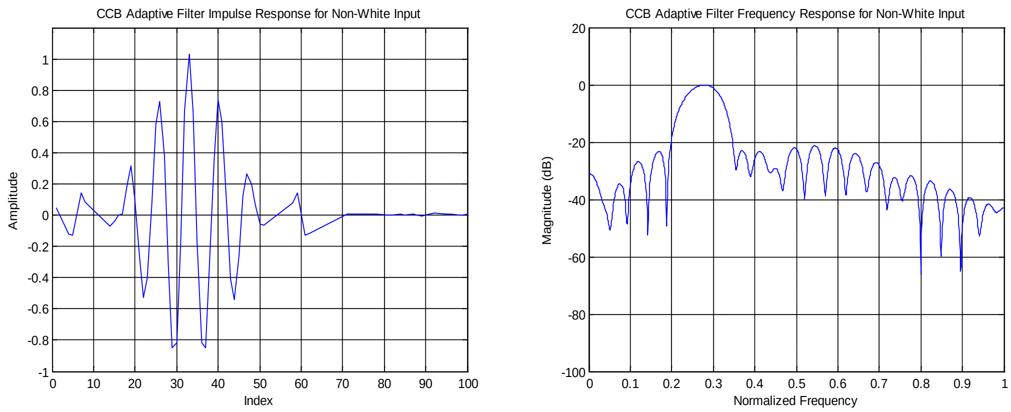


Figure 2.6: Impulse response and frequency response of the CCB adaptive system with 64 nonzero coefficients for a non-white input signal.

Figures 2.5 and 2.6 show the impulse response and frequency response of the converged CCB adaptive algorithm for white and non-white input signals, respectively. From these Figures, observe that the CCB adaptive algorithm is similar to the actual system for the white input, but has observable distortion for the non-white input.

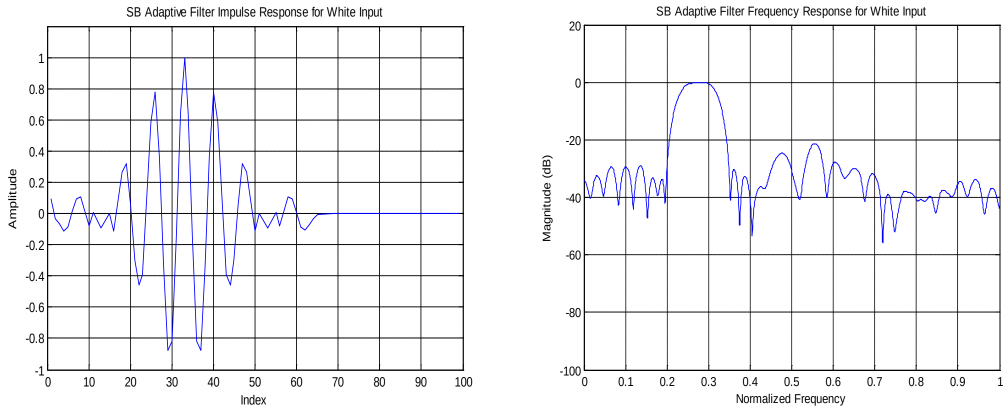


Figure 2.7: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a white input signal.

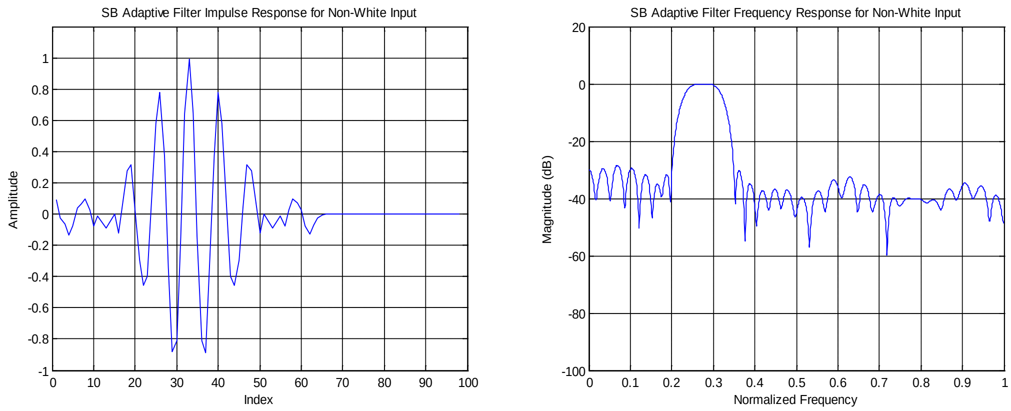


Figure 2.8: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a non-white input signal.

The impulse and frequency responses of the SB adaptive algorithm for white and non-white input signals are shown in Figures 2.7 and 2.8, respectively. Observe that the SB adaptive algorithm has a performance similar to that of the CCB adaptive algorithm for a white input signal, but the performance for a non-white input signal is better than the CCB algorithm.

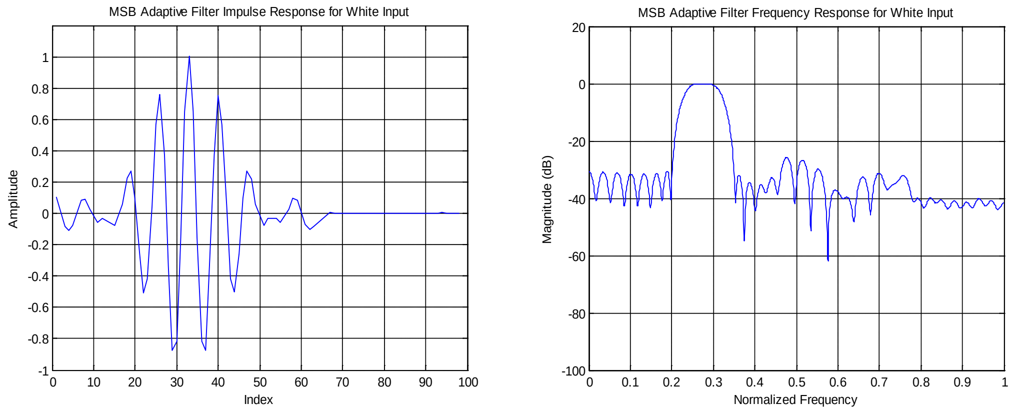


Figure 2.9: Impulse response and frequency response of the MSB adaptive system with 64 nonzero coefficients for a white input signal.

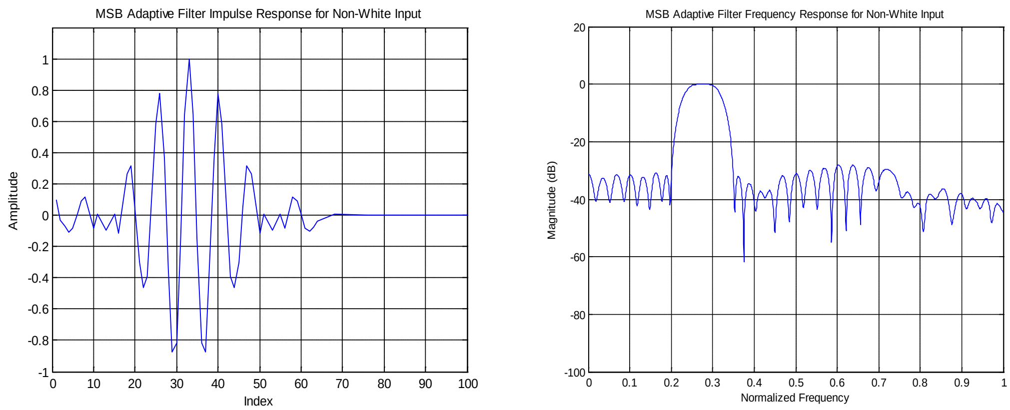


Figure 2.10: Impulse response and frequency response of the MSB adaptive system with 64 nonzero coefficients for a non-white input signal.

Figures 2.9 and 2.10 show the impulse and frequency responses of the MSB adaptive filter for white and non-white input signals, respectively. Observe that the MSB adaptive algorithm has a similar performance to that of the SB adaptive algorithm and has better performance than the CCB adaptive algorithm. It is obvious that the performance of the MSB algorithm for the non-white input signal is clear much better than it is for the CCB adaptive algorithm.

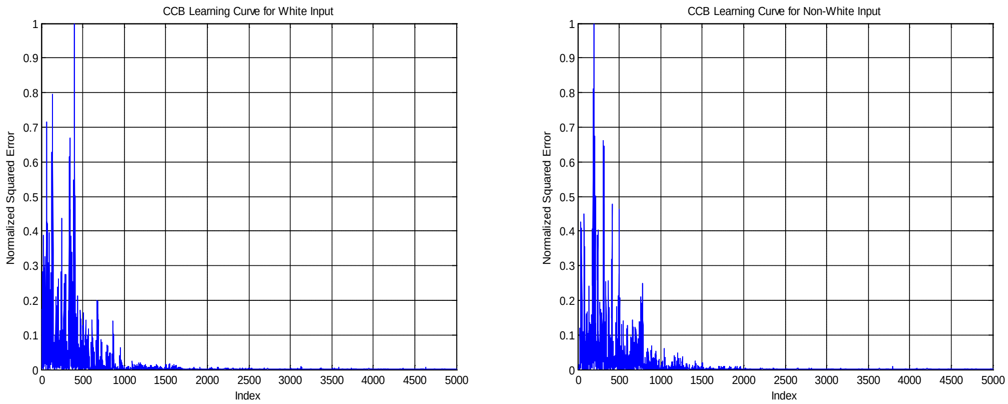


Figure 2.11: Squared-error learning curves of the CCB adaptive algorithm for white and non-white input signals.

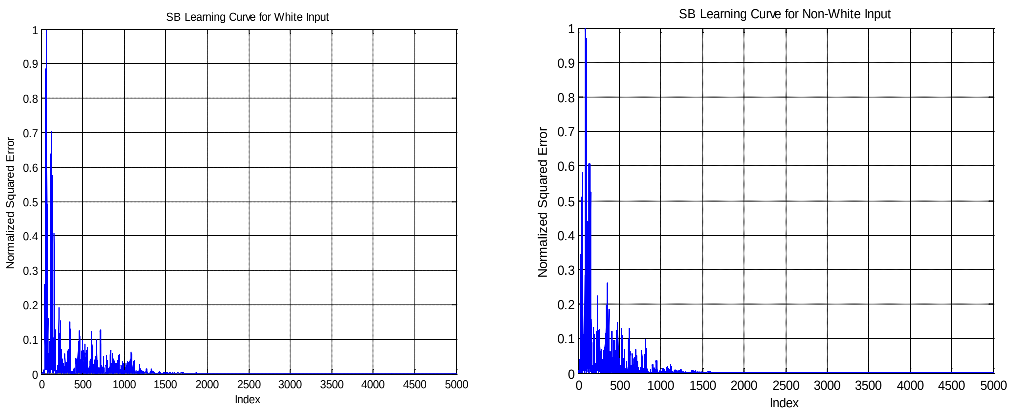


Figure 2.12: Squared-error learning curves of the SB adaptive algorithm for white and non-white input signals.

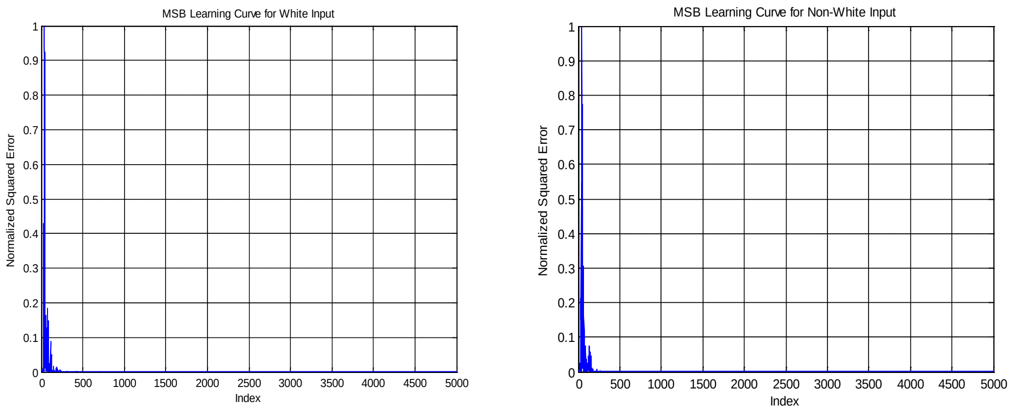


Figure 2.13: Squared-error learning curves of the MSB adaptive algorithm for white and non-white input signals.

The trajectories of the LMS squared errors for a duration of 5,000 samples for the CCB, SB and MSB adaptive algorithms are shown in Figures 2.11, 2.12 and 2.13, respectively. Observe that the MSB algorithm (converged by approximately 150) converges faster than the CCB algorithm (converged by approximately 2000) and SB algorithm (converged by approximately 1500) due to the reduced number of adaptive weights updated in the MSB algorithm.

E. Conclusion

Adaptive algorithm used by the SRAF chooses the appropriate time delays and computes the weight values of the optical switch according to the specific application. In order to improve the performance efficiency of the SRAF, the MSB adaptive algorithm whose performance can be understood mathematically was investigated in this paper. The main idea behind this method is to separately update each row or column vector of weight matrix using the LMS algorithm with the suitable input signal vector and intermediate desired response signal, which has been achieved by a specific structure of the input signals. Because of low computational complexity due to the fast convergence rate, compared with conventional SRAF adaptive algorithms such as the CCB and SB adaptive algorithms, the MSB algorithm is more efficient than them, and also has good performance for white and non-white input signals. The performance of the MSB adaptive algorithm was illustrated by computer simulation example.

III. IMPLEMENTATION OF THE CONNECTION CONSTRAINT

A. Introduction

The SRAF [41] uses adaptive algorithm [42] to choose the appropriate time delays and compute the weight values of the optical switch according to the specific application. The connection constraint algorithm for the CCB and SB adaptive algorithms considers the entire $N \times N$ weight matrix for selecting a subset of N values. The connection constraint algorithm for the MSB adaptive algorithm initially uses N values and considers other values only if the current delay corresponding weight value matches previously chosen delays. Thus, the computational complexity of enforcing the connection constraint of the MSB adaptive algorithm is more efficient than conventional connection constraint.

The purpose of this chapter is twofold. First, we describe conventional connection constraint algorithm for the MSB. Second, we propose two upgraded connection constraint algorithms: when the same values exist as computing the summation of the weight values, the conventional connection algorithm [41] for the CCB and SB adaptive algorithms might not work perfectly. In order to solve this problem, one upgraded algorithm used progressive computation to obtain the better solution is motivated, which can improve the accuracy of the system identification of the CCB and SB adaptive algorithms; and for improving the performance of system identification, another upgraded algorithm for the MSB adaptive algorithm is also be presented.

B. Conventional Connection Algorithm for the MSB

For the MSB adaptive algorithm, we describe a conventional method of choosing the N elements of each column weight vector $\mathbf{w}_j(k)$ (according to the largest magnitudes) subject to the connection constraint that at most one element of each row and column of $\mathbf{W}(k)$ can be nonzero. After the weights are computed using the LMS algorithm, they are copied to the optical switch subject to this input connected only to one output at any given moment. Suppose the candidate vector consisting of the largest elements from all the $\mathbf{w}_j$ is

$$\hat{\mathbf{w}} = [\hat{\omega}_1, \ldots, \hat{\omega}_N]^T \tag{3.1}$$

where $\hat{\omega}_j$ is the largest weight (based on magnitude) of the weight vector $\mathbf{w}_j$.

From this discussion, we can summarize the algorithm as follows:

1) Determine the largest weight (magnitude) of $\hat{\mathbf{w}}$ and save the corresponding i (row index) and j values.
2) Choose the next largest weight (magnitude) value of $\hat{\mathbf{w}}$ and check the corresponding i value.
3) Compare these i values, If the i value chosen from 2) matches previously chosen, exclude the corresponding chosen largest weight, include the next largest weight (magnitude) of the corresponding column weight vector in $\hat{\mathbf{w}}$, and go to the previous step. Otherwise, if the i value chosen from 2) does not match previously chosen, choose the corresponding weight value, save i and j values and go to next step.

4) Continue this process until N weights have been selected, and let the corresponding elements be denoted by the weight vector $\mathbf{w}$.

5) Select the input and output delays (m_i, n_j) of the optical switch associated with the N elements of $\mathbf{w}$, and copy the adaptive weights to the switch to realize the input/output connections.

This algorithm is illustrated by the flowchart in Figure 3.1.

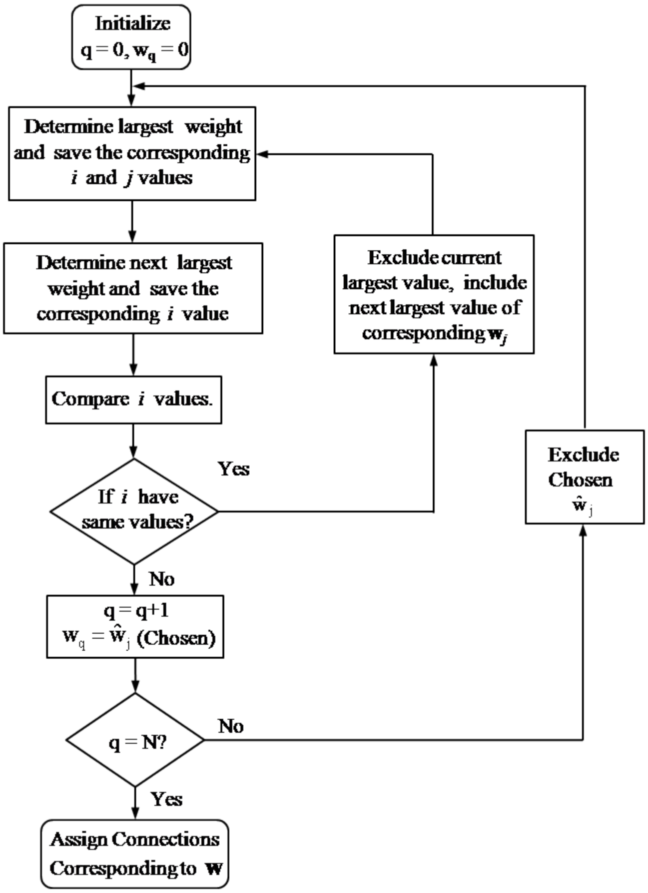


Figure 3.1: Algorithm flowchart for choosing the *N* largest weights subject to the connection constraint for the MSB algorithm.

C. Upgraded Connection Constraint Algorithms

Although a connection algorithm based on sequentially choosing the maximum elements is investigated previously, it might not work perfectly if the same values exist as computing the summation of weight values. In order to solve this problem, an upgraded connection constraint algorithm is motivated. A simple example is considered as follows:

$$W = \begin{bmatrix} 2 & 4 & 9 \\ 5 & 8 & 6 \\ 7 & 12 & 13 \end{bmatrix}. \tag{3.2}$$

Using the conventional connection algorithm, 13 [located at (3,3)], 8 [located at (2,2)], and 2 [located at (1,1)] would be chosen. Because we want to maximize the sum of the weight magnitudes, this selection is not optimal. The optimal connection is given by 12 [located at (3,2)], 9 [located at (1,3)], and 5 [located at (2,1)].

1. Upgraded Connection Constraint Algorithm for CCB and SB

The proposed connection constraint algorithm for the CCB and SB is represented as follows:

1) Choose the largest and next largest values (magnitudes) from those located in the different row and column of the weight matrix.
2) Store the sum of the two largest weights.

3) Based on original weight matrix, generate a modified weight matrix by using zero instead of the largest weight.
4) Repeat 1) and 2), and choose the largest and next largest weights based on the modified weight matrix.
5) Store the sum of the two largest weights selected from 4).
6) If the value in 2) exceeds that in 5), then the largest weight from 1) is chosen. Otherwise, if the value in 5) exceeds that in 2), the largest weight from 4) is chosen.
7) If the value in 2) equals that in 5), then based on 2) and 5), including each next largest value, restore the sums of the largest weights, respectively, and go to 6).
8) Continue this procedure until *N* weights have been selected.

Figure 3.2 shows an algorithm flowchart for the proposed connection algorithm.

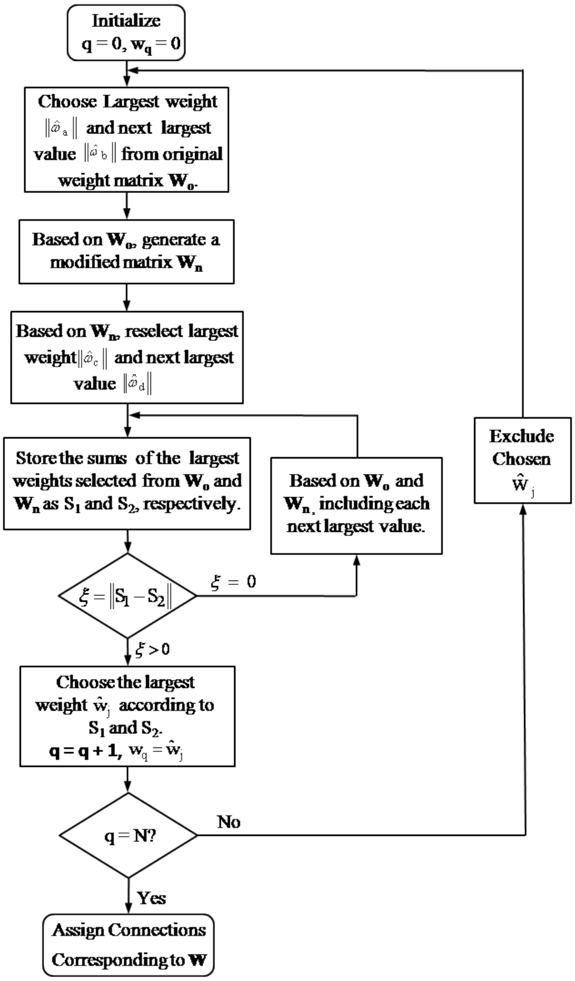


Figure 3.2: Algorithm flowchart of upgraded connection constraint algorithm for the CCB and SB.

2. Upgraded Connection Constraint Algorithm for MSB

Although the upgraded connection constraint algorithm for the CCB and SB has better performance than the conventional algorithm, at the expense of an increase in the computational complexity. In order to improve the efficiency, an upgraded connection algorithm for the MSB adaptive algorithm is also presented. The proposed connection constraint algorithm for the MSB algorithm is summarized as follows:

1) Choose N largest weights (magnitudes) from each column of the weight vector, and save the i (row index) of each largest weight.
2) Compare the i chosen previously. If there is not same value exist, then the N largest weights (magnitudes) selected form 1) are the solution. However, if the same value exists, go to next steps.
3) Based on the first largest weight of N largest weights selected from 1), search the corresponding adaptive weights using conventional algorithm for the MSB, and save the sum of the weight values as S_1.
4) Based on the second largest weight of N largest weights selected from 1), search the corresponding adaptive weights using conventional algorithm, and store the sum of the weight values as S_2.
5) If S_1 is greater than S_2, then the adaptive weights from 3) are chosen. Otherwise, the adaptive weights from 4) are chosen.
6) Continue this procedure until *N* weights have been selected.

Figure 3.3 shows an algorithm flowchart for the proposed connection constraint algorithm for the MSB.

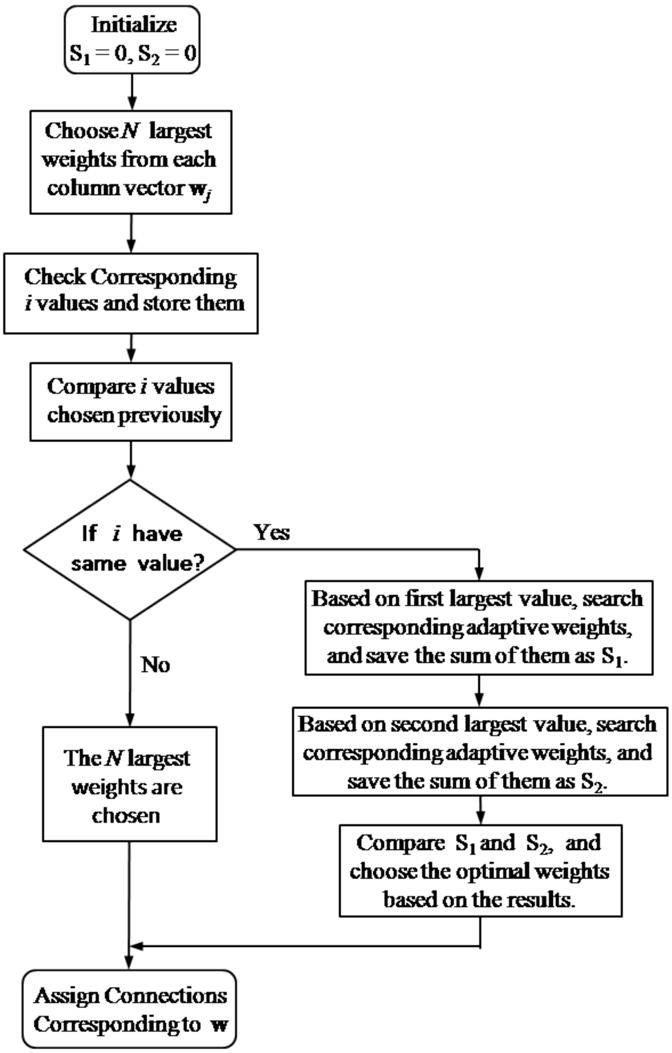


Figure 3.3: Algorithm flowchart of upgraded connection constraint algorithm for the MSB adaptive algorithm.

D. Computer Simulation

Figures 3.4 and 3.5 compare the mean-square-error (MSE) obtained by averaging the squared error over 10,000 samples independent computer runs for the conventional algorithm and upgraded connection constraint algorithms, respectively. Observe that the dotted lines (MSE of proposed algorithms) are lower than the solid lines (MSE of conventional algorithm) for all matrix sizes of the switch, and both of the proposed connection constraint algorithms have better performance than the conventional algorithm. Also it can be easily observe that the proposed algorithm for the MSB converged faster than the proposed algorithm for the CCB and SB.

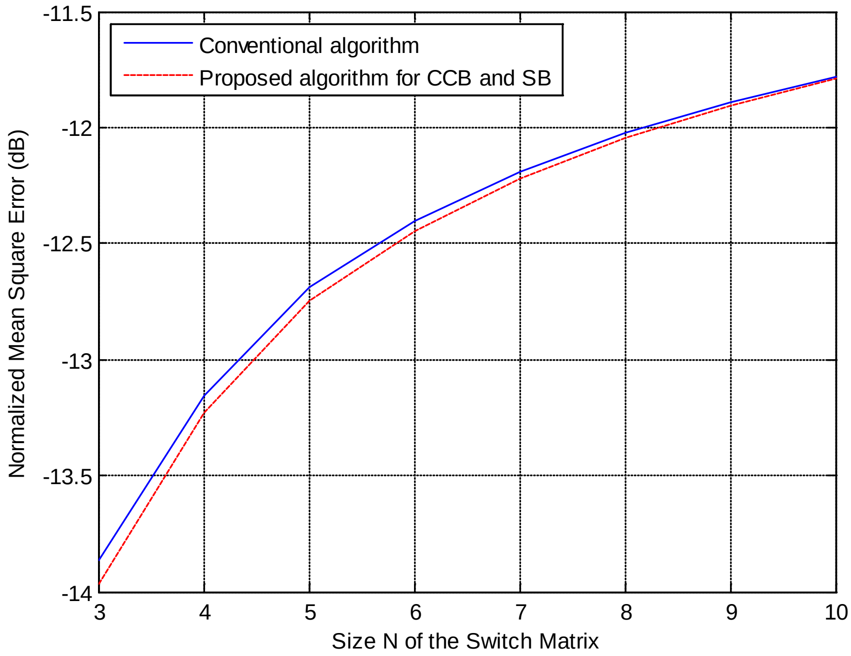


Figure 3.4: MSE curves for conventional algorithm and proposed connection constraint algorithm for CCB and SB.

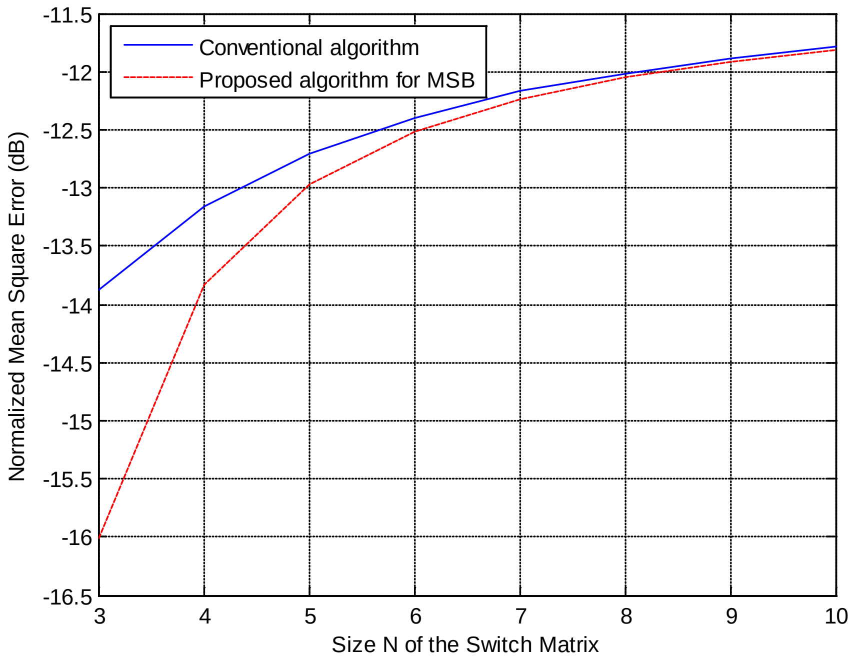


Figure 3.5: MSE curves for conventional algorithm and proposed connection constraint algorithm for MSB.

E. Conclusion

The SRAF is highly flexible due to its ability to choose from a wide range of delay values. In order to verify accurate system identification, two upgraded connection algorithms that can choose the better input and output delays for implementing the connection constraints for the SRAF are described in this chapter.

For implementing the proposed connection constraint algorithm for the CCB and SB, under the circumstance that the same values of summation of the weight magnitudes exist, based on the previous calculation, another largest weight has to be considered for choosing the better solution. Although the proposed connection algorithm for the CCB and SB has improved the accuracy of system identification, at the expense of an increase in the computational complexity. Also, the proposed connection algorithm for MSB has been investigated for better implementing the SRAF. The properties of the proposed connection algorithms were illustrated by computer simulation example.

IV. ADAPTIVE ALGORITHM BASED ON RLS FOR THE SB ALGORITHM

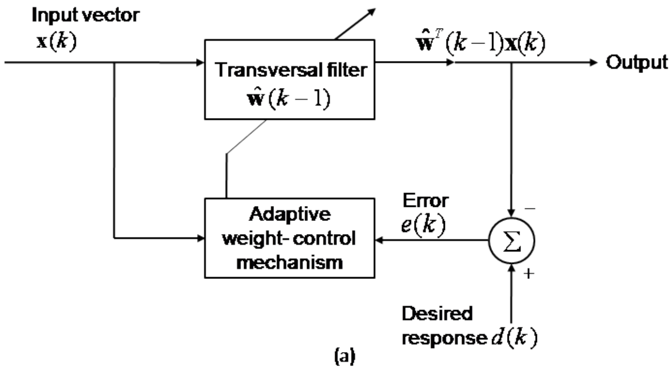
A. Introduction

The least-mean-square (LMS) algorithm and the recursive least-squares (RLS) algorithm [35] have established themselves as the principal tools for linear adaptive filtering [43]. While the LMS algorithm represents the simplest and most easily applied adaptive algorithm, the RLS algorithm represents increased complexity, computational cost, fidelity, and fast convergence rate. In performance, RLS approaches the Kalman filter [36], in adaptive filtering applications, at somewhat reduced required throughput in the signal processor. Recently, various theories have been written on a comparative evaluation of the tracking behaviors of the LMS and RLS algorithms. The convergence behaviors of both of these algorithms are now well understood [44]. Typically, the RLS algorithm has a faster rate of convergence than the LMS algorithm.

In previous work [41], the cross-correlation-based (CCB) and system-based (SB) adaptive algorithms have focused on the LMS algorithm to compute the weight vector for choosing delay connections. In this chapter, the RLS adaptive algorithm is considered for the SB algorithm to improve the convergence rate of system identification.

B. RLS Adaptive Algorithm

Similar to the LMS algorithm, from which it can be derived, the RLS adaptive algorithm minimizes the total squared error between the desired signal and the output from the unknown system. An important feature of the RLS algorithm is that it utilizes information contained in the input data, extending back to the instant of time when the algorithm is initiated. Figure 4.1 shows the representation of the RLS adaptive algorithm. Figure 4.1(a) shows the block diagram of the RLS algorithm, and Fig. 4.1(b) depicts a signal-flow-graph representation of the RLS algorithm that complements the Figure 4.1(a).



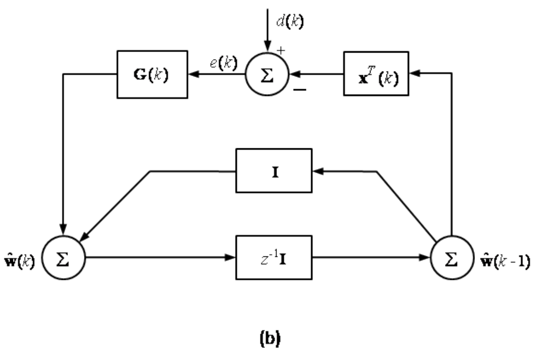


Figure 4.1: Response of the RLS algorithm: (a) block diagram; (b) signal-flow graph.

Where $\mathbf{x}(k)$ is the tap-weight vector at time k, and $\mathrm{e}(k)$ is a priori estimation error defined by

$$\mathrm{e}(k) = d(k) - \widehat{\mathbf{w}}^T(k-1)\mathbf{x}(k) \tag{4.1}$$

the inner product $\widehat{\mathbf{w}}^T(k-1)\mathbf{x}(k)$ represents an estimate of the desired response $d(k)$. The RLS algorithm for computing the extended weight vector $\widehat{\mathbf{w}}$ in (1.9) is given by

$$\widehat{\mathbf{w}}(k) = \widehat{\mathbf{w}}(k-1) + \mathbf{G}(k)e(k). \tag{4.2}$$

The vector $\mathbf{G}(k)$ (size $N \times 1$) is referred to as the gain vector, defined by

$$\mathbf{G}(k) = \frac{\lambda^{-1}\boldsymbol{\beta}(k-1)\mathbf{x}(k)}{1 + \lambda^{-1}\mathbf{x}^T(k)\boldsymbol{\beta}(k-1)\mathbf{x}(k)} \tag{4.3}$$

where matrix $\boldsymbol{\beta}(k)$ (size $N \times N$) is referred to as the inverse correlation matrix, defined by

$$\boldsymbol{\beta}(k) = \lambda^{-1}\boldsymbol{\beta}(k-1) - \lambda^{-1}\mathbf{G}(k)\,\mathbf{x}^T(k)\boldsymbol{\beta}(k-1). \tag{4.4}$$

The forgetting factor λ is a positive constant close to, but less than 1. When λ equals 1, we have the ordinary method of least squares. The inverse of 1 – λ is, roughly speaking, a measure of the memory of the algorithm.

C. Computer Simulation

Next, we present computer simulation results of the SB based on the LMS and RLS adaptive algorithms. Figures 4.2 and 4.3 show the impulse and frequency responses of the SB adaptive algorithm for white and non-white input signals based on the LMS algorithm, respectively. Also, the impulse and frequency responses of the SB adaptive

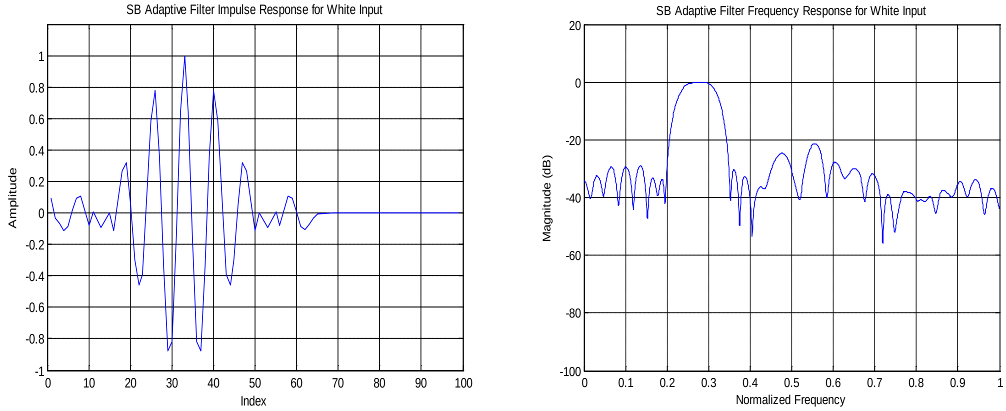


Figure 4.2: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a white input signal based on the LMS algorithm.

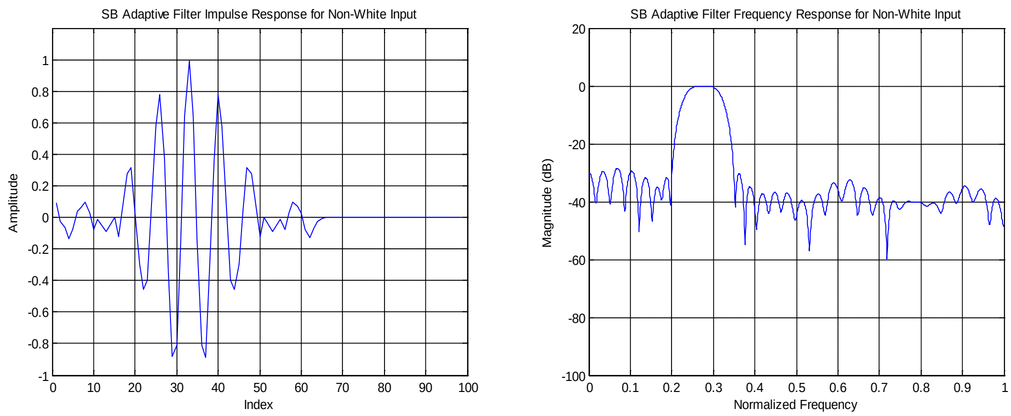


Figure 4.3: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a non-white input signal based on the LMS algorithm.

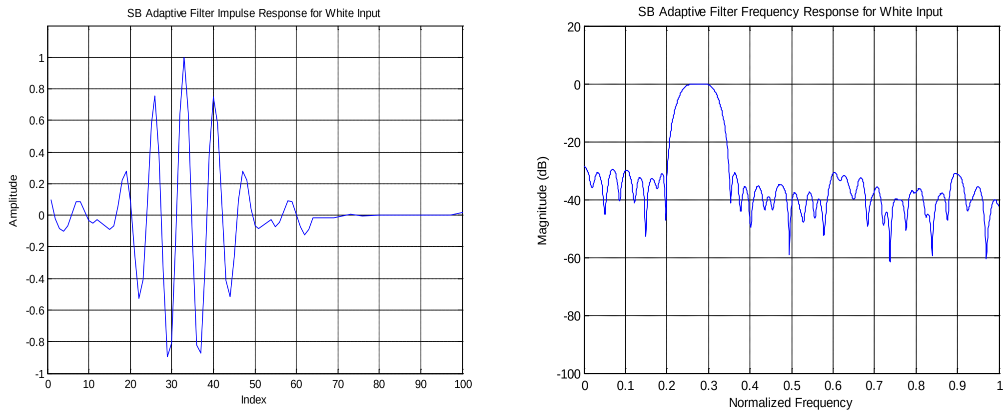


Figure 4.4: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a white input signal based on the RLS algorithm.

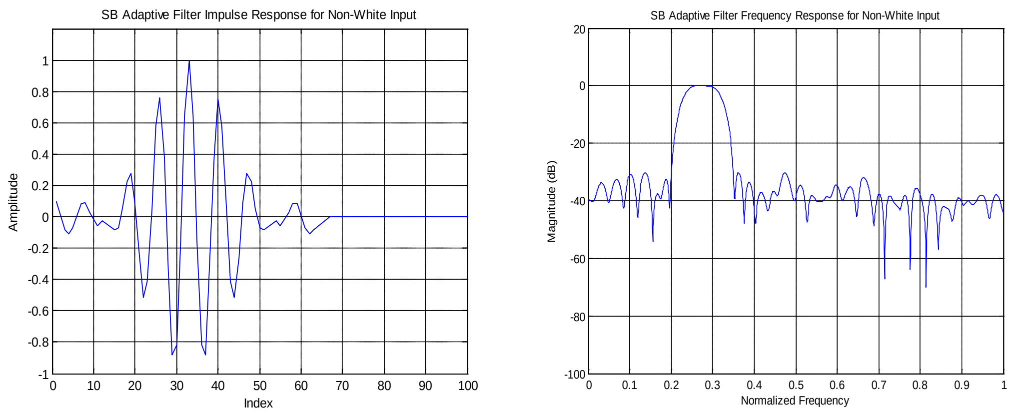


Figure 4.5: Impulse response and frequency response of the SB adaptive system with 64 nonzero coefficients for a non-white input signal based on the RLS algorithm.

algorithm for white and non-white input signals based on the RLS algorithm are shown in Figures 4.4 and 4.5, respectively. Observe that the performance of the SB adaptive algorithm based on the RLS has a performance similar to that of based on the LMS algorithm.

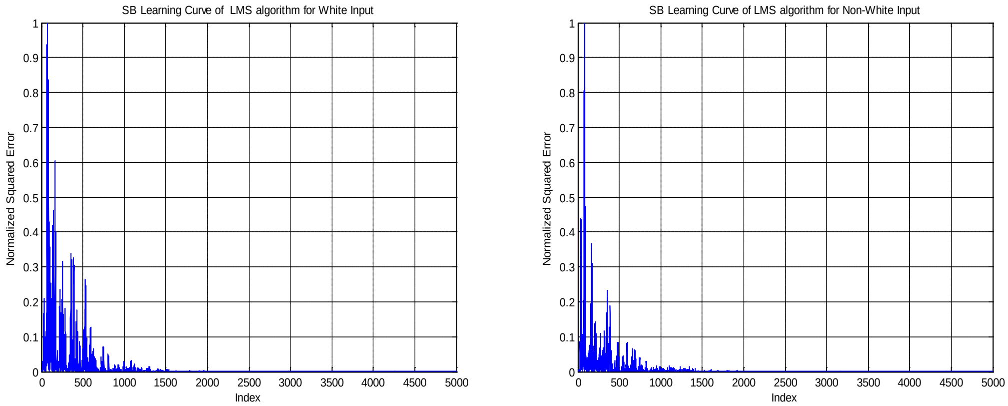


Figure 4.6: LMS squared-error learning curves of the SB adaptive algorithm for white and non-white input signals.

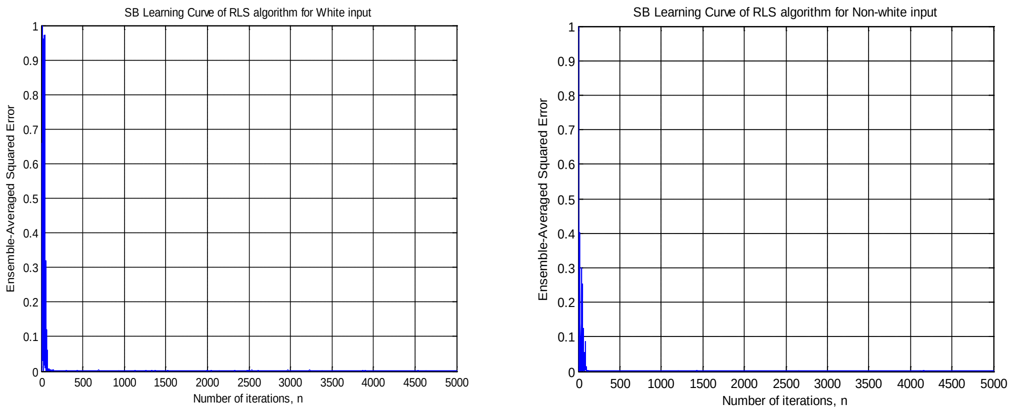


Figure 4.7: RLS squared-error learning curves of the SB adaptive algorithm for white and non-white input signals.

Finally, we present computer simulation results of the LMS and RLS algorithms for the SB algorithm. For this computer simulation, the algorithms were run for $L = 5{,}000$ iterations, the step-size parameter was $\mu = 0.001$, and the forgetting factor $\lambda = 0.98$. Figure 4.6 shows the trajectory of the LMS squared error for the SB adaptive algorithm with both types of input signals, observing that the algorithm has converged by approximately sample 1500. Also, the trajectory of the RLS squared error for the SB adaptive algorithm with both types of input signals is shown in Figure 4.7. Observe that the algorithm has converged by approximately sample 100. From these figures, we observe that the convergence rate of the RLS algorithm is faster than the LMS algorithm. This improvement in performance, however, is achieved at the expense of a large increase in computational complexity.

D. Conclusion

The RLS is an adaptive algorithm which recursively finds the filter coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is similar to other algorithms such as the LMS that aim to reduce the mean square error. Compared to the LMS algorithm, the RLS approach offers faster convergence. However, this benefit comes at the expense of requiring more computations. In order to improve the convergence rate of the SB algorithm, the RLS algorithm was considered in this chapter. Also, the performance of the LMS and RLS algorithms for the SB algorithm were illustrated by computer simulation example.

V. CONCLUSION

Due to MEMS technology, large optical switches can be efficiently implemented, thus broadening the range of possible applications. The SRAF is highly flexible due to its ability to choose from a wide range of delay values. In order to improve the performance of the sparse reconfigurable adaptive filter (SRAF), a MSB adaptive algorithm whose performance can be understood mathematically was investigated in this thesis. The main idea behind this method is to separately update each row or column vector of weight matrix using the LMS algorithm with the suitable input signal vector and intermediate desired response signal, which has been achieved by a specific structure of the input signals. Because of less calculation compared with conventional CCB and SB algorithms, the MSB algorithm is more efficient, and also has good performance for white and non-white input signals. The performance of the MSB adaptive algorithm was illustrated by a computer simulation example.

The connection constraint algorithm for the CCB and SB adaptive algorithms considers the entire $N \times N$ weight matrix when selection a subset of N values. The connection constraint algorithm for the MSB adaptive algorithm initially uses N values and considers other values only if the current weights value matches previously chosen values. When the same values exist as computing the summation of the weight values, the previous connection algorithm based on sequentially choosing the maximum elements might not work perfectly. In order to ensure accurate system identification, two upgraded connection algorithms that can choose the best input and output delay values for implementing the connection constraints for the SRAF is described in this thesis. The proposed connection constraint algorithm for the CCB and SB is more accurate than the conventional algorithm, but at the expense of an increase in computational complexity; in order to improve the efficiency, another proposed connection constraint algorithm for the MSB was also proposed.

In order to improve the convergence rate of the SB algorithm, an adaptive algorithm based on RLS was discussed. The convergence rate of the RLS algorithm is much faster than the LMS algorithm. The improvement in performance, however, is achieved at the cost of high computational complexity.

REFERENCES

[1] J. J. Shynk, J. E. Bowers, and S. Hwang, “A sparse reconfigurable adaptive filter based on a photonic switch,” in *Proc*. *38th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, pp. 235-240, Nov. 2004.

[2] R. K. Mehra. “Approaches to adaptive filtering,” *IEEE Trans. Autom. Control*, vol. AC-17, pp. 693-698. 1972.

[3] J. R. Treichler and M. G. Larimore, “Thinned impulse response for adaptive FIR filters,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* Paris, France, May 1982, pp. 631-634.

[4] M. Smith and D. Farden, “Thinning the impulse response of FIR digital filters,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,*1981, vol. 6, pp. 240-242.

[5] D. Mattera, F. Palmieri, and S. Haykin, “Efficient sparse FIR filter design,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* 2002, vol. 2, PP. 1537-1540.

[6] J. L. H. Webb and D. C. Munson Jr., “Chebyshev optimization of sparse FIR filters using linear programming with an application to beamforming,” *IEEE Trans. Signal Process.,* vol. 44, no. 8, pp. 1912-1922, Aug. 1996.

[7] L. Rabiner, “Linear program design of finite impulse response (FIR) digital filters,” *IEEE Trans. Audio Electroacoust.,* vol. AE-20, no. 4, pp. 280-288, Oct. 1972.

[8] Y. Neuvo, D. Cheng-Yu, and S. K. Mitra, “Interpolated finite impulse response filters,” *IEEE Trans. Acoust., Speech, Signal Process.,* vol. ASSP-32, no. 3, pp. 563-570, Jun. 1984.

[9] T. Saramaki, Y. Neuvo, and S. K. Mitra, “Design of computationally efficient interpolated FIR filters,” *IEEE Trans. Circuits Syst.,* vol. 35, no. 1, pp. 70-88, Jan. 1988.

[10] S. S. Haykin, B. Widrow (Editor): *Least-Mean-Square Adaptive Filters*, Wiley, 2003.

[11] O. J. Tobias, R. Seara, and C. A. F. da Rocha, “Analytical model for the mean weights of two adaptive interpolated-FIR filter structures,” in *Proc. IEEE Int. Symp. Circuits Syst.,* Geneva, Switzerland, May 2000, vol. 3, pp. 93-96.

[12] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, “On adaptive interpolated FIR filters,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* Montreal, QC, Canada, May 2004, vol. 2, pp. 665-668.

[13] K. C. Ho and S. D. Blunt, “Rapid identification of a sparse impulse response using an adaptive algorithm in the Haar domain,” *IEEE Trans. Signal Process.,* vol. 51, no. 3, pp. 628-638, Mar. 2003.

[14] N. J. Bershad and A. Bist, “Fast coupled adaptation for sparse impulse responses using a partial Haar transform,” *IEEE Trans. Signal Process.,* vol. 53, no. 3, pp. 966-976, Mar 2005.

[15] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson Jr., “Exploiting sparsity in adaptive filters,” *IEEE Trans. Signal Process.,* vol. 50, no. 8, pp. 1883-1894, Aug. 2002.

[16] F. O’Regan and C. Henegham, “Algorithmic analysis and implementation of a novel natural gradient adaptive filter for sparse system,” in *Proc. IEEE Int. Conf. Acoust*., *Speech, Signal Process.,* Hong Kong, China, Apr. 2003, vol. 2, pp.553-556.

[17] J. M. Medel. “Gradient estimation algorithms for equation error formalutions,” *IEEE Trans. Autom. Control.* Vol. AC-19, pp. 820-824.

[18] K. Dogancay, “Generalized subband decomposition adaptive filters for sparse systems,” *Signal Process.,* vol. 83, pp. 1093-1103, May 2003.

[19] G. Rombouts and M. Moonen, “Sparse-BEFAP: A fast implementation of fast affine projection avoiding explicit regularization,” in *Proc, Eur. Signal Process. Conf.,* TAMPERE, Finland, Sep. 2000, pp. 1871-1874.

[20] M. Tur, J. W. Goodman, B. Moslehi, J. E. Bowers, and H. J. Shaw, “Fiber-optic signal processor with applications to matrix-vector multiplication and lattice filtering,” *Opti. Lett.*, vol. 7, pp. 463-465, Sep. 1982.

[21] R. Helkey, S. Adams, J. Bowers, T. Davis, O. Jerphagnon, V. Kaman, A. Keating, B. Liu, C. Pusarla, D. Xu, S. Yuan, and X. Zheng, “Design of large, MEMS-based photonic switches,” *Opt. Photon. News*, vol. 13, pp. 42-45, May 2002.

[22] V. Kaman, X. Zheng, R. J. Helkey, C. Pusarla, and J. E. Bowers, “A 32-element 8-bit photonic true-time-delay system based on a 288×288 3-D MEMS optical switch,” *IEEE Photon. Technol. Lett.,* vol. 15, pp. 849-851, Jun. 2003.

[23] T. Yamamoto, J. Yamaguchi, R. Sawada, and Y. Uenishi, “Development of a large -scale 3-D MEMS optical switch module,” *NTT Tech. Rev.*, vol. 1, pp. 37-42, Oct. 2003.

[24] X. Zheng, V. Kamen, S. Yuan, Y. Xu, O. Jerphagnon, A. Keating, R. C. Anderson, H. N. Poulsen, B. Liu, J. R. Sechrist, C. Pusarla, R. Helkey, D. J. Blumenthal, and J. E. Bowers, “Three-dimensional MEMS photonic cross-connect switch design

and performance," *IEEE J. Sel. Topics Quantum Electron*., vol. 9, no. 2, pp. 571-578, Apr. 2003.

[25] J. H. Gross and D. M. Etter, "Comparison of echo cancellation algorithms for the adaptive delay filter," *in Proc. IEEE Veh. Technol. Conf.,* Denver, CO, May 1992, pp. 574-576.

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

[27] S. S. Narayan and A. M. Peterson. Frequency domain least-mean-square algorithm. *Proc. IEEE*, 69(1):124-126, Jan. 1981.

[28] B. Widrow and E. Walach. On the statistical efficiency of the LMS algorithm with nonstationary inputs. *IEEE Transactions on Information Theory*, IT-30(2):211-222, March 1984.

[29] J. Homer, I. Mareels, R. R. Bitmead, B. Wahlberg, and F. Gustafsson, "LMS estimation via structural detection," *IEEE Trans. Signal Process.* Vol. 46, no. 10, pp. 2651-2663, Oct. 1998.

[30] S. Hwang, J. J. Shynk, and J. E. Bowers, "A system-based adaptive algorithm for a reconfigurable photonic switch," in *Proc. 39th Conf. Inf. Sci. Syst.*, Baltimore, MD, Mar. 2005, pp. TA8.3.1-8.3.4.

[31] O. Macchi, N. J. Bershad, and M. M-boup, "Steady-state superiority of LMS over LS for time varying line enhancer in noisy environment," *IEEE Proc. London*. *part F,* vol. 138, pp. 354-360. 1991.

[32] F. Ling. Efficient least-squares lattice algorithms based on givens rotation with systolic array implementations. In *Proc. ICASSP-89*, pp. 1290-2193, Glasgow, Scotland, 1989.

[33] T. S. Huang, S. D. Blostein, and E. A. Margerum, "Least-squares estimation of motion parameter from 3-D point correspondences," *In proc. IEEE Conf. Computer Vison and Pattern Recognition, Miami Beach,* FL, 1986, pp. 24-26.

[34] R. De Cristofaro, "A study of transversal filter design techniques in digital communications," M. Eng. Thesis, Concordia Univ., Montreal, P. Q., Canada, Apr. 1976.

[35] S. Haykin, *Adaptive Filter Theory,* 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[36] B. Mulgrew, "Kalman filter techniques in adaptive filtering," *IEEE Proc,* part F, vol. 134, pp. 239-243. 1987.

[37] B. Mulgrew and C. F. N. Gowan, "An adaptive Kalman equalizer: structure and performance," *IEEE Trans. Acoust. Speech Signal Process.,* vol. ASSP-35, pp. 1727-1735. 1987.

[38] D. N. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM J. Res. Dev.,* vol. 18, pp. 267-273, 1974.

[39] Q. T. Zhang and S. Haykin, "Tracking characteristics of the Kalman filter in a nonstationary environment for adaptive filter applications," *Proc. ICASSP,* Boston, MA, 1983, pp. 671-674.

[40] T. Kang and J. J. Shynk, "Adaptive algorithm properties for a reconfigurable photonic switch," in *Proc. 39th Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, Nov. 2005, pp. 340-343.

[41] S. Hwang, J. J. Shynk, T. Kang and J. E. Bowers, “Algorithms for a sparse reconfigurable adaptive filter and a photonic switch architecture,” *IEEE Transactions on circuits and systems,* Regular papers, vol. 55, no. 1, pp. 347-360, Feb. 2008.

[42] A. Benveniste, “Design of adaptive algorithms for the tracking of time-varying systems,” *Int. J. Adaptive Contr. Signal Processing,* vol. 1, pp. 3-29, 1987.

[43] O. Macchi, “Optimization of adaptive identification for time-varying filters,” *IEEE Trans. Automat. Contr.,* vol. AC-31, pp. 283-287, 1986.

[44] E. Eweda, “Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels,” *IEEE Trans. Signal Processing.* vol. 42, pp. 2937-2944, 1994.