Spline NLMS Adaptive Filter Algorithm based on the Signed Regressor of Input Signal

Hossein Tavakoli[a], Mohammad Shams Esfand Abadi[b,*]

[a]Faculty of Electrical Engineering, Shahid Rajaee Teacher Training University, P.O.Box:16785-163, Tehran, Iran
[b]Faculty of Electrical Engineering, Shahid Rajaee Teacher Training University, P.O.Box:16785-163, Tehran, Iran

ARTICLE INFO

Keywords:
Normalized least mean squares (NLMS)
Spline adaptive filtering (SAF)
Signed regressor (SR)
Control point
L_1-norm

ABSTRACT

This paper presents a new spline adaptive filtering (SAF) algorithm based on signed regressor (SR) of input signal. The algorithm is called SR-SAF normalized least mean squares (SR-SAF-NLMS). The SR-SAF-NLMS is established through L_1-norm constraint to the proposed cost function. In this algorithm, the polarity of the input signal is used to adjust the weight coefficients and control point vectors. Therefore, the computational complexity, especially the number of multiplications, is significantly reduced. Furthermore, the performance of the SR-SAF-NLMS is close to the conventional SAF-NLMS. The good performance of the proposed algorithm is demonstrated through several simulation results in different scenarios.

1. Introduction

System identification is an important task in many engineering applications such as channel estimation, acoustic echo cancellation, channel equalization, and active noise control [1-3]. In a system identification task, an adaptive filter is applied to model the unknown system. This goal is achieved by adaptation of the filter coefficients based on a suitable algorithm. The linear adaptive filter (LAF) algorithms such as least mean squares (LMS) and normalized LMS (NLMS) algorithms are widely used for identifying the linear system [4]. These algorithms are simple and easy to implement. But, it is well-known that many real systems are nonlinear [5, 6]. Therefore, several nonlinear adaptive filter (NLAF) algorithms were developed [7-9].

Based on Volterra expansions of the input signal, the Volterra adaptive filtering (VAF) is applied for nonlinear system identification [10]. But, this algorithm requires large number of free parameters and suffers from analytical problems. The neural networks (NN) can also be utilized to approximate the nonlinear systems [11]. The high computational complexity and getting trapped in local minima are the main problems in this approach. Another method is called kernel adaptive filtering (KAF) [12, 13]. In this method, the input signal is transformed into the high-dimensional feature space by reproducing the kernel. Unfortunately, KAF have the problem of a continuously increasing network growth. In functional link network (FLN), the input data is expanded by a nonlinear function series such as Legendre, and trigonometric [14-16]. The computational complexity and the accuracy of this technique are related to the various functional expansions [17].

A new family of nonlinear adaptive filter algorithms, called spline adaptive filtering (SAF), has been developed in [18]. The SAF utilizes adaptive spline interpolation to adjust nonlinear filters. The nonlinear filter learns by updating linear weight coefficients and a small set of spline control points. In comparison with other algorithms, the SAF is simple to implement, flexible, and has low computational complexity. Based on different spline structures, Wiener [18], Hammerstein [19], and cascade [20], various spline adaptive filter algorithms have been introduced. In all these algorithms, the LMS algorithm is used to adapt the weight coefficients and control points. Also, the theoretical steady-state performance of SAF-LMS was studied in [21]. In the following, this approach was extend to infinite impulse response [22], NLMS [23], and subband adaptive filters [24]. The SAF has also been developed in different applications such as active noise control [25, 26] and adaptive distributed networks [27]. The frequency domain of SAF algorithm can be found in [28]. The SAF algorithm has also been extended to two-dimensional function in [29]. Another researches in SAF have been focused on nonlinear system identification under impulsive noise environments [30-32].

It is obvious that during the update process, the convergence speed, the steady-state error, and the computational complexity are important features. In some applications, the number of filter coefficients is very large. Therefore, the high computational complexity is one of the main problem in these situations. To solve this problem, various approaches such as selective partial update (SPU) [33-35] and signed regressors (SR) were proposed [36-40]. In the signed regressor LMS (SR-LMS), the signum of the input regressors is utilized. In this algorithm, the polarity of the input signal is used to adjust the filter coefficients, which requires no multiplications. The SR-LMS has a convergence speed and a steady-state error level that are only slightly inferior to those of the LMS algorithm for the same parameter setting [41]. To increase the convergence speed of SR-LMS, the signed regressor NLMS (SR-NLMS) was firstly proposed in [36]. Also, the modified version of this algorithm (MSR-NLMS) was presented in [42]. In [43], this approach was successfully extend to subbnad adaptive filter and SR subband adap-

*Corresponding author. Tel./Fax : +98 21 22970006
ORCID(s): 0000-0003-3060-4211 (H. Tavakoli); 0000-0002-9856-6592 (M.S.E. Abadi)
htavakoli@sru.ac.ir, mshams@sru.ac.ir

tive filter was established which has better performance than SR-NLMS. Also, the application of SR-SAF in adaptive distributed diffusion networks can be found in [44].

The SAF-NLMS works well in nonlinear system identification application. However, in comparison with SR-LMS, this algorithm has higher computational complexity. This problem is significantly important when the number of weight coefficients increases. Therefore, developing the new adaptive filter algorithm with low computational complexity and close performance to SAF-NLMS, is highly desirable. According to what we said, one of the proper solution to reduce the computational complexity is applying the signum of input signals in weight update equations. This strategy remarkably causes to reduce the number of multiplications. In this research, by introducing the novel cost function, the SR approach is extended to SAF-NLMS and SR-SAF-NLMS is established. The L_1-norm criterion is applied to the cost function with the proper constraint which leads to the appearance of the sign operator into the derived relations. In the proposed algorithms, the new update equations for weight coefficients and control point vectors are developed. Since the sign operator is inserted into the update equation, the computational complexity is significantly reduced. Also, the convergence speed of the introduced algorithm is close to conventional SAF-NLMS. We demonstrate the good performance of the proposed algorithms through several simulations in different nonlinear systems.

The contribution of this paper can be summarized as follows:

- The establishment of the exact SAF-NLMS algorithm according to proposed cost function.
- The establishment of the SR-SAF-NLMS algorithms via L_1-norm criterion to the introduced cost function.
- Study of the computational complexity in the derived update relations.
- Performance analysis of the proposed algorithms for different simulation setups.
- Study of the tracking performance of the proposed algorithms.
- Study of the steady-state performance of the proposed algorithms in various scenarios.

This paper is organized as follows: Sect. 2 describes the nonlinear data model. The NLMS algorithm is reviewed in Sect. 3. In Sect. 4, the SR-NLMS is derived based on the L_1-norm constraint. In Sect. 5, the SAF-NLMS algorithm is established. The SR-SAF-NLMS is proposed in Sect. 6. The computational complexity of the proposed algorithm is discussed in Sect. 7. Finally, the paper ends with a comprehensive set of simulations supporting the good performance of the proposed algorithm.

Throughout the paper, $(.)^T$ represents transpose of a vector or matrix, $\|.\|_1$ indicates L_1-norm of a vector, $\|.\|^2$ takes the squared Euclidean norm of a vector, $\lfloor . \rfloor$ describes the Floor function, and sign[.] shows the sign function.

2. Nonlinear data model

In this research, the desired signal is generated according to the following linear and nonlinear data model as

$$d(k) = \Im[\mathbf{u}^T(k)\mathbf{h}_o] + v(k), \tag{1}$$

where $\mathbf{h}_o$ is an $M \times 1$ unknown linear coefficients, $\mathbf{u}(k) = [u(k), u(k-1), \dots, u(k-M+1)]^T$ is an $M \times 1$ input regressor vector, $\Im(.)$ is a desired nonlinear function and $v(k)$ is an additive noise with variance σ_v^2. It is assumed that $v(k)$ is zero mean, white, Gaussian, and independent of $\mathbf{u}(k)$. Fig. 1 shows the structure of the desired signal generation.

3. Review of NLMS algorithm

The output $r(k)$ of an adaptive filter at iteration k is given by

$$r(k) = \mathbf{h}^T(k)\mathbf{u}(k), \tag{2}$$

where $\mathbf{h}(k) = [h_0(k), h_1(k), \dots, h_{M-1}(k)]^T$ is the $M \times 1$ of adaptive filter coefficients. The NLMS algorithm minimizes the following cost function

$$\min \frac{1}{2}\|\mathbf{h}(k+1) - \mathbf{h}(k)\|^2, \tag{3}$$

subject to $d(k) = \mathbf{u}^T(k)\mathbf{h}(k+1)$. By using the method of Lagrange Multiplier, the weight coefficients update equation becomes

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu \frac{\mathbf{u}(k)}{\epsilon + \|\mathbf{u}(k)\|^2} e(k), \tag{4}$$

where μ is the step-size, ϵ is the regularization parameter, and $e(k)$ is the output error which is obtained by

$$e(k) = d(k) - \mathbf{h}^T(k)\mathbf{u}(k). \tag{5}$$

4. Review of SR-NLMS algorithm

The SR-NLMS algorithm minimizes the following cost function

$$\min \|\mathbf{h}(k+1) - \mathbf{h}(k)\|_1, \tag{6}$$

subject to $d(k) = \mathbf{u}^T(k)\mathbf{h}(k+1)$. Now define the cost function as

$$\begin{aligned} J(k) = \|\mathbf{h}(k+1) - \mathbf{h}(k)\|_1 + \\ \alpha[d(k) - \mathbf{u}^T(k)\mathbf{h}(k+1)] \end{aligned} \tag{7}$$

where α is the Lagrange Multipliers. Using $\frac{\partial J(k)}{\partial \mathbf{h}(k+1)} = 0$ and $\frac{\partial J(k)}{\partial \alpha} = 0$, we get

$$d(k) = \mathbf{u}^T(k)\mathbf{h}(k+1), \tag{8}$$

and

$$\text{sign}(\mathbf{h}(k+1) - \mathbf{h}(k)) = \alpha \mathbf{u}(k). \tag{9}$$

By multiplying both sides of Eq. (9) into $\text{sign}(\mathbf{u}(k))\mathbf{u}^T(k)$ from the left, we obtain

$$\text{sign}(\mathbf{u}(k))\mathbf{u}^T(k)\text{sign}(\mathbf{h}(k+1)-\mathbf{h}(k)) = \alpha\text{sign}(\mathbf{u}(k))\|\mathbf{u}(k)\|^2. \quad (10)$$

Now define the following relation

$$\text{sign}(\mathbf{u}(k)) = \mathbf{\Theta}(k)\mathbf{u}(k), \quad (11)$$

and

$$\text{sign}(\mathbf{h}(k+1)-\mathbf{h}(k)) = \mathbf{\Upsilon}(k)(\mathbf{h}(k+1)-\mathbf{h}(k)), \quad (12)$$

where

$$\mathbf{\Theta}(k) = \text{diag}[\frac{1}{|u(k)|}, \frac{1}{|u(k-1)|}, \dots, \frac{1}{|u(k-M+1)|}], \quad (13)$$

and

$$\mathbf{\Upsilon}(\mathbf{k}) = \text{diag}[\frac{1}{|h_0(k+1)-h_0(k)|}, \dots, \frac{1}{|h_{M-1}(k+1)-h_{M-1}(k)|}]. \quad (14)$$

Thus, Eq. (10) can be stated as

$$\mathbf{\Theta}(k)\mathbf{u}(k)\mathbf{u}^T(k)\mathbf{\Upsilon}(k)(\mathbf{h}(k+1)-\mathbf{h}(k)) = \alpha\mathbf{\Theta}(k)\mathbf{u}(k)\|\mathbf{u}(k)\|^2 \quad (15)$$

Assuming that the diagonal elements of $\mathbf{u}(k)\mathbf{u}^T(k)$ is larger than off diagonal elements and rearranging the matrices, we obtain

$$\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)[\mathbf{u}^T(k)\mathbf{h}(k+1)-\mathbf{u}^T(k)\mathbf{h}(k)] = \alpha\mathbf{\Theta}(k)\mathbf{u}(k)\|\mathbf{u}(k)\|^2 \quad (16)$$

Since $d(k) = \mathbf{u}^T(k)\mathbf{h}(k+1)$, we get

$$\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)e(k) = \alpha\mathbf{\Theta}(k)\mathbf{u}(k)\|\mathbf{u}(k)\|^2 \quad (17)$$

This equation can be written as

$$\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)e(k) = \alpha\mathbf{\Theta}(k)\mathbf{u}^T(k)\mathbf{u}(k)\mathbf{u}(k) \quad (18)$$

We know $\|\mathbf{u}(k)\|_1 = \text{sign}(\mathbf{u}^T(k))\mathbf{u}(k)$ and $\mathbf{\Theta}(k)\mathbf{u}^T(k) = \text{sign}(\mathbf{u}^T(k))$. Therefore,

$$\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)e(k) = \alpha\mathbf{u}(k)\|\mathbf{u}(k)\|_1 \quad (19)$$

By multiplying both sides of Eq. (19) into $\text{sign}(\mathbf{u}^T(k))$ from the left, α is given by

$$\alpha = \frac{\text{sign}(\mathbf{u}^T(k))\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)e(k)}{(\|\mathbf{u}(k)\|_1)^2} \quad (20)$$

Substituting Eq. (20) into Eq. (9), we have

$$\mathbf{\Upsilon}(k)[\mathbf{h}(k+1)-\mathbf{h}(k)] = \frac{\text{sign}(\mathbf{u}^T(k))\mathbf{\Upsilon}(k)\mathbf{\Theta}(k)\mathbf{u}(k)e(k)}{(\|\mathbf{u}(k)\|_1)^2}\mathbf{u}(k) \quad (21)$$

Multiplying both sides of Eq. (21) into $\mathbf{\Upsilon}^{-1}(k)$ and rearranging the diagonal matrices, the SR-NLMS is established as

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu\frac{\text{sign}(\mathbf{u}(k))}{\epsilon + \|\mathbf{u}(k)\|_1}e(k) \quad (22)$$

5. The SAF-NLMS algorithm

Fig. 2 indicates the spline adaptive filter setup. Splines are smooth parametric curves defined by interpolation of properly defined control points collected in a lookup table (LUT). The spline estimation provides an approximation of $y(k) = \mathfrak{I}[r(k)]$ based on two parameters, $p(k)$, and $j(k)$, which are directly depending on $r(k)$. In cubic spline curves and for each input signal, $r(k)$, the spline uses four control points selected inside the LUT. Each spline span controlled by four adjacent points of the LUT, is addressed by the so called span index $j(k)$ and, inside each span, a normalized local span-abscissa parameter $p(k) \in [0,1]$ is defined. From $r(k)$, the parameters $p(k)$ and $j(k)$ are obtained as follows

$$p(k) = \frac{r(k)}{\Delta x} - [\frac{r(k)}{\Delta x}], \quad (23)$$

$$j(k) = [\frac{r(k)}{\Delta x}] + \frac{Q-1}{2}, \quad (24)$$

where Δx is the uniform space between control points and Q is the total number of control points. The output of the spline interpolation is given by

$$y(k) = \mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k), \quad (25)$$

where

$$\mathbf{p}(k) = [p^3(k), p^2(k), p(k), 1]^T, \quad (26)$$

$$\mathbf{q}_j(k) = [q_j(k), q_{j+1}(k), q_{j+2}(k), q_{j+3}(k)]^T. \quad (27)$$

The vector $\mathbf{q}_j(k)$ is the control point vector and the matric $\mathbf{C}$ is the spline basis matrix. For cubic spline the basis matric $\mathbf{C}$ is defined as

$$\mathbf{C} = \frac{1}{6}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}. \quad (28)$$

Imposing different constraints to approximation relationship, several spline basis with different properties, can be evaluated in a similar manner. An example of such a basis is the case of Catmul-Rom (CR) spline [18], very important in many applications, and the CR-spline basis matric $\mathbf{C}$ has the form

$$\mathbf{C} = \frac{1}{2}\begin{pmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{pmatrix}. \quad (29)$$

CR-spline, in fact, specifies a curve that pass through all of the control points, a feature which is not necessarily true for other spline methodologies. Overall, the CR-spline results in a more local approximation with respect to the B-spline. The update equation in SAF-NLMS needs to calculate for both weight coefficients and control pints vectors. It is important to note that in [23], there is not complete relations to establish SAF-NLMS in details. In our research, we derive the exact update equation for SAF-NLMS algorithm. First, we focus on control points vector. The SAF-NLMS minimizes the following cost function

$$\min\|\mathbf{q}_j(k+1)-\mathbf{q}_j(k)\|^2, \tag{30}$$

subject to

$$d(k)=\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1). \tag{31}$$

The cost function is defined as

$$\begin{aligned}J(k)=\|\mathbf{q}_j(k+1)-\mathbf{q}_j(k)\|^2+\\ \alpha(d(k)-\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1)).\end{aligned} \tag{32}$$

Using $\frac{\partial J(k)}{\partial \mathbf{q}_j(k+1)}=0$ and $\frac{\partial J(k)}{\partial \alpha}=0$, we get

$$d(k)=\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1), \tag{33}$$

and

$$\mathbf{q}_j(k+1)=\mathbf{q}_j(k)+\frac{\alpha}{2}\mathbf{C}^T\mathbf{p}(k). \tag{34}$$

By substituting Eq. (34) into Eq. (33), we have

$$\alpha=\frac{2e(k)}{\|\mathbf{C}^T\mathbf{p}(k)\|^2}, \tag{35}$$

where $e(k)=d(k)-\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k)$. Therefore, the updated equation for control points vector becomes

$$\mathbf{q}_j(k+1)=\mathbf{q}_j(k)+\mu_q\frac{\mathbf{C}^T\mathbf{p}(k)}{\epsilon+\|\mathbf{C}^T\mathbf{p}(k)\|^2}e(k), \tag{36}$$

where μ_q is the step-size in control points vector update equation. For the weight coefficients vector, we define the following cost function

$$\begin{aligned}J(k)=\|\mathbf{h}(k+1)-\mathbf{h}(k)\|^2+\\ \alpha(d(k)-\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k))\end{aligned} \tag{37}$$

It is important to note that $\mathbf{p}^T(k)$ is related to $\mathbf{h}(k+1)$. Now, we select the same Largrange Multiplier as Eq. (35), $\alpha=\frac{2e(k)}{\|\mathbf{C}^T\mathbf{p}(k)\|^2}$, then by $\frac{\partial J(k)}{\partial \mathbf{q}_j(k+1)}=0$, we get

$$\mathbf{h}(k+1)=\mathbf{h}(k)+\mu_h\frac{\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)\mathbf{u}(k)}{\epsilon+\|\mathbf{C}^T\mathbf{p}(k)\|^2}e(k), \tag{38}$$

where

$$\dot{\mathbf{p}}(k)=[3p^2(k),2p(k),1,0]^T, \tag{39}$$

and it is the derivative of $\mathbf{p}(k)$. Also, μ_h is the step-size in weight coefficients vector update equation.

6. The SR-SAF-NLMS algorithm

Fig. 3 depicts the SR spline adaptive filter setup. The aim of this section is providing the new algorithm which has low computational complexity and close performance to SAF-NLMS. Now define the following cost function

$$\begin{aligned}J(k)=\|\mathbf{q}_j(k+1)-\mathbf{q}_j(k)\|_1+\\ \beta(d(k)-\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1)),\end{aligned} \tag{40}$$

where β is the Lagrange Multiplier. Using $\frac{\partial J(k)}{\partial \mathbf{q}_j(k+1)}=0$ and $\frac{\partial J(k)}{\partial \beta}=0$, we get

$$d(k)=\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1), \tag{41}$$

and

$$\mathrm{sign}[\mathbf{q}_j(k+1)-\mathbf{q}_j(k)]-\beta\mathbf{C}^T\mathbf{p}(k)=0. \tag{42}$$

By multiplying $\mathrm{sign}[\mathbf{C}^T\mathbf{p}(k)]\mathbf{p}^T(k)\mathbf{C}$ from the left, we have

$$\begin{aligned}\mathrm{sign}[\mathbf{C}^T\mathbf{p}(k)]\mathbf{p}^T(k)\mathbf{C}\mathrm{sign}[\mathbf{q}_j(k+1)-\mathbf{q}_j(k)]=\\ \beta\mathrm{sign}[\mathbf{C}^T\mathbf{p}(k)]\mathbf{p}^T(k)\mathbf{C}\mathbf{C}^T\mathbf{p}(k).\end{aligned} \tag{43}$$

Define the following relations

$$\mathrm{sign}[\mathbf{C}^T\mathbf{p}(k)]=\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k), \tag{44}$$

and

$$\mathrm{sign}[\mathbf{q}_j(k+1)-\mathbf{q}_j(k)]=\mathbf{\Psi}(k)(\mathbf{q}_j(k+1)-\mathbf{q}_j(k)), \tag{45}$$

then, we obtain

$$\begin{aligned}\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)\mathbf{p}^T(k)\mathbf{C}\mathbf{\Psi}(k)(\mathbf{q}_j(k+1)-\mathbf{q}_j(k))=\\ \beta\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)\mathbf{p}^T(k)\mathbf{C}\mathbf{C}^T\mathbf{p}(k)\end{aligned} \tag{46}$$

By relocate the matrices and using the fact that $\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k+1)-\mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k)=e(k)$, we get

$$\mathbf{\Psi}(k)\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)e(k)=\beta\mathbf{C}^T\mathbf{p}(k)\|\mathbf{C}^T\mathbf{p}(k)\|_1. \tag{47}$$

Multiplying both sides of Eq. (47) into $\mathrm{sign}[\mathbf{p}^T(k)\mathbf{C}]$ from the left, we get

$$\mathrm{sign}[\mathbf{p}^T(k)\mathbf{C}]\mathbf{\Psi}(k)\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)e(k)=\beta(\|\mathbf{C}^T\mathbf{p}(k)\|_1)^2. \tag{48}$$

Therefore

$$\beta=\frac{\mathrm{sign}[\mathbf{p}^T(k)\mathbf{C}]\mathbf{\Psi}(k)\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)e(k)}{(\|\mathbf{C}^T\mathbf{p}(k)\|_1)^2}. \tag{49}$$

We know

$$\mathrm{sign}[\mathbf{q}_j(k+1)-\mathbf{q}_j(k)]=\beta\mathbf{C}^T\mathbf{p}(k), \tag{50}$$

then

$$\mathbf{\Psi}(k)(\mathbf{q}_j(k+1)-\mathbf{q}_j(k))=$$

$$\frac{\text{sign}[\mathbf{p}^T(k)\mathbf{C}]\mathbf{\Psi}(k)\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)}{(\|\mathbf{C}^T\mathbf{p}(k)\|_1)^2}\mathbf{C}^T\mathbf{p}(k)e(k). \tag{51}$$

Now by multiplying the above equation from the left by $\mathbf{\Psi}^{-1}(k)$, we have

$$\mathbf{q}_j(k+1) - \mathbf{q}_j(k) = \frac{\mathbf{\Phi}(k)\text{sign}[\mathbf{p}^T(k)\mathbf{C}\mathbf{C}^T\mathbf{p}(k)]}{(\|\mathbf{C}^T\mathbf{p}(k)\|_1)^2}\mathbf{C}^T\mathbf{p}(k)e(k), \tag{52}$$

and we obtain

$$\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q\frac{\mathbf{\Phi}(k)\mathbf{C}^T\mathbf{p}(k)}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|_1}e(k). \tag{53}$$

Finally, the update equation for control points vector in SR-SAF-NLMS becomes

$$\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q\frac{\text{sign}[\mathbf{C}^T\mathbf{p}(k)]}{\|\mathbf{C}^T\mathbf{p}(k)\|_1}e(k). \tag{54}$$

The same as SR-NLMS, the update equation for weight coefficients in SR-SAF-NLMS is given by

$$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h\frac{\text{sign}[\mathbf{u}(k)]}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|_1}\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)e(k). \tag{55}$$

Table 1 shows the Pseudocode of SR-SAF-NLMS algorithm.

7. Computational complexity

Tables 2 and 3 present the exact computational complexity of SAF-NLMS and SR-SAF-NLMS algorithms. These tables describes the number of multiplications, divisions and additions at each relation in the update equation process. Table 4 compares the computational complexity of the SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms. It shows that the number of multiplications in SR-SAF-NLMS-1 is significantly lower than SAF-NLMS. The SR-SAF-NLMS-2 has close computational complexity to SAF-NLMS. Furthermore, the SR-SAF-NLMS-3 has the lowest computational complexity. Fig. 4 shows the number of multiplications and divisions versus filter length (M) for the mentioned algorithms. We observe that SR-SAF-NLMS-1 and SR-SAF-NLMS-3 have lower computational complexity than other algorithms especially for large values of M. The CPU run time has also been presented in Table 5. This table indicates that the CPU run-time of SR-SAF-NLMS-1 is lower SAF-LMS and SAF-NLMS algorithm. The SR-SAF-NLMS-3 has the lowest CPU run-time.

8. Simulation results

In this section, several simulation and experiment results are performed in the context of system identification to validate the performance of the aforementioned SR-SAF-NLMS algorithm. In all simulations, we show the MSE learning curves which are evaluated by ensemble averaging over 100 independent trials. Table 6 demonstrates three different learning algorithms called SR-SAF-NLMS-1, SR-SAF-NLMS-2 and SR-SAF-NLMS-3 algorithms in which, $\mathbf{h}(k)$, and $\mathbf{q}_j(k)$ are updated. In addition, the performance of SR-SAF-NLMS algorithms have been compared with SAF-LMS and SAF-NLMS in all experiments. A suitable choice of initial conditions that have always guaranteed excellent results is $\mathbf{h}(-1) = \delta(k)$ for FIR filter coefficients, while the initial value of spline control knots $\mathbf{q}(-1)$ set to a straight line with unitary slope [18].

8.1. Experiment 1

The first experiment is performed in order to evaluate the convergence behavior of proposed SR-SAF-NLMS algorithms with focusing on identification of an unknown Wiener system which is illustrated in Fig. 2 [18]. The unknown linear filter is $\mathbf{h}_o = [0.6, -0.4, 0.25, -0.15, 0.1]^T$ and a nonlinear spline function interpolated by a 23 control points LUT with an uniform interval sampling, $\Delta x = 0.2$, defined by

$$\mathbf{q}_{j,o} = [-2.2, -2.0, \dots, -1.0, -0.8, -0.91, 0.42, \\ -0.01, -0.1, 0.1, -0.15, 0.58, 1.2, 1, 1.2, \dots, 2.2]. \tag{56}$$

The input signal $\mathbf{u}(k)$ consists of 30,000 samples which is generated by AR(1) process defined by

$$u(k) = au(k) + \sqrt{1-a^2}\zeta(k), \tag{57}$$

where $\zeta(k)$ is a zero mean white Gaussian noise with unitary variance and $a \in [0, 1)$ is a correlation factor which can interpret the correlation between adjacent samples. In addition, an additive white Gaussian noise is added to the unknown system output, setting the signal-to-noise ratio (SNR) to 30 dB. We also studied the performance of the algorithms for uniform, binary and Laplace noise distributions [45, 46]. The learning update rates are set to $\mu_h = \mu_q = 0.01$ and $\epsilon = 0.001$.

Fig. 5 shows the MSE learning curves of the algorithms for slightly colored input signal ($a = 0.5$). This figure compares the convergence rate of SAF-LMS, SAF-NLMS, SR-SAF-NLMS-1 and SR-SAF-NLMS-2 algorithms. We observe that SR-SAF-NLMS-1 has close performance to SAF-NLMS algorithm. Also, the SR-SAF-NLMS-1 has good convergence speed. It is important to note that in SR-SAF-NLMS algorithms, the sign of input regressors are utilized in update equation. Therefore, the computational complexity of these algorithms is lower than SAF-NLMS. Fig. 6 compares the performance of SR-SAF-NLMS-1 for different distribution of noise. The results show that the SR-SAF-NLMS-1 has good performance for all distributions. In Fig. 7, various values for Δx have been selected. Again, the algorithms are robust when this parameter changes. As we said, we use CR-spline basic matrix in the simulations. Fig. 8 compares the performance of the SAF-NLMS and SR-SAF-NLMS-1 for CR-spline and B-spline basic matrices. We observe that the results based on CR-spline have better performance. The tracking performance of the proposed algorithm has been studied in Fig. 9. In this simulation the unknown linear filter is suddenly changed to $\mathbf{h}_o = [0.7, -0.1, 0.5, -0.3, 0.4]^T$ at iteration 20000. As we see, the SR-SAF-NLMS-1 has a good tracking performance.

8.2. Experiment 2

The purpose of this setup is to identify a dynamic nonlinear system constitute of three blocks which has been interpreted in [18]. The first and third blocks are two fourth order IIR filter, Butterworth and Chebychev, respectively, with transfer functions

$$H_B(z) = \frac{(0.2851 + 0.5704z^{-1} + 0.2851z^{-2})}{(1 - 0.1024z^{-1} + 0.4475z^{-2})} \times \frac{(0.2851 + 0.5701z^{-1} + 0.2851z^{-2})}{(1 - 0.0736z^{-1} + 0.0408z^{-2})}, \quad (58)$$

and

$$H_C(z) = \frac{(0.2025 + 0.2880z^{-1} + 0.2025z^{-2})}{(1 - 1.01z^{-1} + 0.5861z^{-2})} \times \frac{(0.2025 + 0.0034z^{-1} + 0.2025z^{-2})}{(1 - 0.6591z^{-1} + 0.1498z^{-2})}, \quad (59)$$

while second block is nonlinearity as follow

$$y(k) = \frac{2u(k)}{1 + |u(k)|^2}. \quad (60)$$

This system is similar to radio frequency amplifiers for satellite communications (high power amplifier), in which the linear filters model the dispersive transmission paths, while the nonlinearity models the amplifier saturation [18]. The input signal is a zero mean white Gaussian noise with unitary variance which is generated by AR(1) process. The MSE learning curves of the proposed SR-SAF-NLMS algorithms for slightly and highly colored input signal ($a = 0.5$ and $a = 0.95$) are studied in Figs. 10 and 11. The learning update rates are set to $\mu_h = \mu_q = 0.01$, $\epsilon = 0.001$, and $M = 15$. Fig. 10 indicates that the performance of SR-SAF-NLMS algorithms are close to SAF-NLMS algorithm. The convergence rate of SR-SAF-NLMS algorithms is comparable with SAF-NLMS. Furthermore, the computational complexity of SR-SAF-NLMS algorithms is significantly lower than SR-NLMS. Fig. 11 shows the results for highly colored input signal. The same performance can be seen in this figure.

In Fig. 12, various values for μ have been chosen in SR-SAF-NLMS-3 algorithm. By increasing the step-size, the speed of convergence increases but the steady-state MSE increases too. Fig. 13 shows the tracking performance of the proposed algorithms. For tracking we change $H_B(z)$ to

$$H_B(z) = \frac{(1.3851 + 0.6704z^{-1} + 1.3851z^{-2})}{(1.1 - 0.2024z^{-1} + 0.5475z^{-2})} \times \frac{(1.3851 + 0.6701z^{-1} + 1.3851z^{-2})}{(1.1 - 0.0936z^{-1} + 0.0608z^{-2})}, \quad (61)$$

at iteration 20000. The results indicate that the proposed algorithms have good tracking ability in this experiment but the steady-state MSE increases.

Fig. 14 shows the steady-state MSE versus step-size. The step-size changes from 0.01 to 0.96. We observe that by increasing the step-size, the steady-state MSE increases. The SAF-LMS, SAF-NLMS, and SR-SAF-NLMS-1 algorithms are stable for large values of the step-size. But, the SR-SAF-NLMS-2 and SR-SAF-NLMS-3 are not stable for large values of the step-size and steady-state MSE extremely increases. Table 7 presents the steady-state MSE values for three step-sizes. This table confirms that for small step-size, all algorithms are stable. For large values, the SR-SAF-NLMS-1 is still stable and other algorithms are not stable. Fig. 15 shows the steady-state MSE versus SNR. The SNR changes from 10 to 50dB. When SNR increases, the steady-state MSE decreases. Table 8 presents the steady-state MSE values for various SNR levels.

8.3. Experiment 3

In this respect, the algorithms are verified for complex recurrent network in the identification of a nonlinear dynamic system which is delineated with transfer function [18]

$$y(k) = \frac{y(k-1)}{(1 + y^2(k-1))} + u^3(k). \quad (62)$$

In addition, the input signal is modelled as

$$\mathbf{u}(k) = 1.79u(k-1) - 1.85u(k-2) + 1.27u(k-3) - 0.41u(k-4) + \zeta(k), \quad (63)$$

where $\zeta(k)$ is a zero mean white Gaussian noise with unitary variance. In order to validate the flexibility of algorithms, the simulations have been done for different input signal normalization circumstances $[0, 0.1]$, $[-0.1, 0.1]$, and $[-0.25, 0.25]$. The parameters of the simulations are set to $M = 15$, $\mu_h = \mu_q = 0.5$, $\Delta x = 0.05$, and $\epsilon = 0.001$. Fig. 16 shows the results for $[0, 0.1]$. We observe that the SR-SAF-NLMS-1 has the same convergence speed with SR-NLMS and lower steady-sate error than SR-NLMS. Because of sign operation in SR-SAF-NLMS-1, it is obvious that the computational complexity of SR-SAF-NLMS-1 is also lower than SR-NLMS. In this case, the SR-SAF-NLMS-2 has good convergence speed, but the steady-state error is higher than other algorithms. The performance of the algorithms for various input signal normalization conditions has been presented in Figs. 17 and 18. These figures indicate that the SR-SAF-NLMS-1 has convenient performance for both simulations. However, the performance of SR-SAF-NLMS-2 is deviated especially for $[-0.25, 0.25]$. In Fig. 19, the MSE learning curves of SR-SAF-NLMS-1 have been compared for different conditions. In comparison with SR-NLMS, we observe that the performance of SR-SAF-NLMS-1 is well for various situations.

8.4. Experiment 4

In this experiment, the impulse response of the car echo path with 256 taps ($M = 256$) has been used as an unknown system [33]. Fig. 20 indicates the impulse response of the car echo path. The input signal is slightly colored input signal and learning rates are set to $\mu_h = \mu_q = 0.5$. Fig. 21 shows that the proposed algorithms have good convergence speed and low steady-state error for large value of M. In SR-SAF-NLMS-3, both control points and filter coefficients apply sign operation during the update process. We observe that the SR-SAF-NLMS-3 as well as other SR-SAF-NLNS algorithms behave well in this simulation.

9. Conclusion

This paper presents the new spline adaptive filter algorithms called SR-SAF-NSAF. The introduced algorithm was derived based on a L_1-norm constraint. The SR-NSAF utilized the polarity of the input signal to adjust the filter coefficients. The computational complexity of the proposed algorithm is significantly lower than conventional SAF-NLMS algorithm. Furthermore, the convergence speed of SR-SAF-NLMS is close to the SAF-NLMS. The good performance of SR-SAF-NLMS was demonstrated through several simulation results.

10. Future recommendation

Since the sign regressor method is useful approach in adaptive filter algorithms, the other important algorithms such as AP and subband adaptive filter can be developed to establish the SR-SAF-AP and SR-SAF subband adaptive filters. Furthermore, the theoretical mean square performance analysis of the proposed algorithms can be studied and accordingly, closed form relations for steady-state MSE will be achieved.

References

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

[2] Haykin, S., *Adaptive Filter Theory*, NJ: Prentice-Hall, 4th edition (2002).

[3] Farhang-Boroujeny, B., *Adaptive Filters: Theory and Applications*, Wiley (1998).

[4] Sayed, A.H., *Adaptive Filters*,Wiley (2008).

[5] Mandic, D.P. and Goh, V.S.L., *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*, John Wiley Sons, Chichester, West Sussex, UK (2009).

[6] Comminiello, D. and Principe, J.C., *Adaptive Learning Methods for Nonlinear System Modeling*, Elsevier, Kidlington, Oxford, UK (2018).

[7] Carini, A., Orcioni, S., Terenzi, A., et al. "Nonlinear system identification using Wiener basis functions and multiple-variance perfect sequences", *Signal Processing*, **160** (3), pp. 137-149 (2019).

[8] Zhang, S. and Zheng, W.X. "Recursive adaptive sparse exponential functional link neural network for nonlinear AEC in impulsive noise environment", *IEEE Trans. Neural Netw*, **29** (9), pp. 4314-4323 (2018).

[9] Yin, K.-L., Pu, Y.-F., Lu, L. "Hermite functional link artificial neural network assisted adaptive algorithms for IoV nonlinear active noise control", *IEEE Internet Things J*, **7** (9), pp. 8372-8383 (2020).

[10] Tan, L. and Jiang, J. "Adaptive Volterra filters for active control of nonlinear noise processes", *IEEE Trans. Signal Process*, **49** (8), pp. 1667-1676 (2001).

[11] Zhao, H., Gao, S., He, Z., et al. "Identification of nonlinear dynamic system using a novel recurrent wavelet neural network based on the pipelined architecture", *IEEE Trans. Ind. Electron*, **61** (8), pp. 4171-4182 (2014).

[12] Chen, B., Xing, L., Xu, B., et al. "Kernel risk-sensitive loss: Definition, properties and application to robust adaptive filtering", *IEEE Trans. Signal Process*, **65** (11), pp. 2888-2901 (2017).

[13] Flores, A., de Lamare, R.C. "Set-membership adaptive kernel NLMS algorithms: Design and analysis", *Signal Process*, **154**, pp. 1-14 (2019).

[14] Patel, V., Gandhi, V., Heda, S., et al. "Design of adaptive exponential functional link network-based nonlinear filters", *IEEE Trans. Circuits Syst. I*, **63** (9), pp. 1434-1442 (2016).

[15] Carini, A., Cecchi, S., Romoli, L., et al. "Legendre nonlinear filters", *Signal Process*, **109**, pp. 84-94 (2015).

[16] Carini, A. and Sicuranza, G.L. "A study about Chebyshev nonlinear filters", *Signal Process*, **122**, pp. 24-32 (2016).

[17] Sicuranza, G.L. and Carini, A. "On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control", *IEEE Trans. Audio Speech Lang. Process*, **20** (1), pp. 234-245 (2011).

[18] Scarpiniti, M., Comminiello, D., Parisi, R., et al. "Nonlinear spline adaptive filtering", *Signal Process*, **93** (4), pp. 772-783 (2013).

[19] Scarpiniti, M., Comminiello, D., Parisi, R., et al. "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties", *Signal Process*, **100**, pp. 112-123 (2014).

[20] Scarpiniti, M., Comminiello, D., Parisi, R., et al. "Novel cascade spline architectures for the identification of nonlinear systems", *IEEE Trans. Circuits Syst. I*, **62** (7), pp. 1825-1835 (2015).

[21] Scarpiniti, M., Comminiello, D., Scarano, G. et al. "Steady-state performance of spline adaptive filters", *IEEE Trans. Signal Process*, **64** (4), pp. 816-828 (2016).

[22] Scarpiniti, M., Comminiello, D., Parisi, R., et al. "Nonlinear system identifica-tion using IIR spline adaptive filters", *Signal Process*, **108**, pp. 30-35 (2015).

[23] Guan, S. and Li, Z. "Normalised spline adaptive filtering algorithm for nonlinear sys- tem identification", *Neural Process. Lett*, **46** (2), pp. 595-607 (2017).

[24] Wen, P., Zhang, J., Zhang, S., et.al "Normalized subband spline adaptive filter: algorithm derivation and analysis", *Circuits, Systems, and Signal Processing*, **40**, pp. 2400-2418 (2021).

[25] Patel, V. and George, N.V. "Compensating acoustic feedback in feed-forward active noise control systems using spline adaptive filters", *Signal Process*, **120**, pp. 448-455 (2016).

[26] Patel, V. and George, N.V. "Multi-channel spline adaptive filters for non-linear active noise control", *Applied Acoustics*, **161** (9), pp. 1-9 (2020).

[27] Mishra, B.P., Wilson, A.M., Panigrahi, T. "Estimation of nonmlinear wireless channel based of incremental spline adaptive filtering", *Wireless Personal Communication*, **126**, pp. 445-480 (2022).

[28] Yang, L., Liu, J., Zhang, Q., et al "Frequency domain spline adaptive filters", *Signal Process*, **177**, pp. 1-10 (2020).

[29] Saito, S. and Sugita, Y. "Spline nonlinrar adaptive filters considering cross terms", *Signal Process*, **184**, pp. 1-6 (2021).

[30] Liu, C., Zhang, Z., Tang, X. "Sign normalised spline adaptive filtering algorithms against impulsive noise", *Signal Process*, **148**, pp. 234-240 (2018).

[31] Yang, L., Liu, J., Yan, R., et al "Spline adaptive filter with arctangent-momen- tum strategy for nonlinear system identification", *Signal Process*, **164**, pp. 99-109 (2019).

[32] Liu, Q. and He, Y. "Robust Geman-McClure based nonlinear spline adaptive filter against impulsive noise", *IEEE Access*, **8**, pp. 22571-22850 (2020).

[33] Dogancay, K. and Tanrikulu, O. "Adaptive filtering algorithms with selective partial updates", *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process*, **48**, pp. 762-769 (2001).

[34] Abadi, M.S.E., Husoy, J.H., Ahamdi, M.J. , "Two low computational complexity improved multiband structured subband adaptive filter algorithms", *Scientia Iranica*, **28** (6), pp. 3396-3411 (2021).

[35] Abadi, M.S.E., Husoy, J.H., Ahamdi, M.J. "Two improved multiband structured subbnad adaptive filter algorithm with reduced computational complexity", *Signal Process*, **154** (1), pp. 15-29 (2019).

[36] Nagumo, J. and Noda, A. "A learning method for system identifica-

tion", *IEEE Trans. Automat. Contr*, **12**, pp. 282-287 (1967).
[37] Duttweiler, D. "Adaptive filter performance with nonlinearities in the correlation multipliers", *IEEE Trans Acoust Speech Signal Process*, **30** (8), pp. 578-586 (1982).
[38] Gersho, A. "Adaptive filtering with binary reinforcement", *IEEE Trans. Inform. Theory*, **30** (3), pp. 191-199 (1984).
[39] Sethares, W.A. "Adaptive algorithms with nonlinear data and error functions", *IEEE Trans Signal Process*, **40** (9), pp. 2199-2206 (1992).
[40] Koike, S. "Analysis of adaptive filters using normalized signed regressor LMS algorithm", *IEEE Trans Signal Process*, **47** (10), pp. 2710-2723 (1999).
[41] Bershad,N.j. "Comments on comparison of the convergence of two algorithms for adaptive FIR digital filters", *IEEE Trans Acoust Speech Signal Process*, **33** (12), pp. 1604-1606 (1985).
[42] Takahashi, K. and Mori, S. "A new normalized signed regressor LMS algorithm", *in Proc. ICCS/ISITA* , pp. 1181-11185 (1992).
[43] Abadi, M.S.E., Shafiee, M.S., Zalaghi, M. "A low computational complexity normalized subband adaptive filter algorithm employing signed regressor of input signal", *Eurasip J. Advances in Signal Processing*, **2018**(1), pp. 1-23 (2018).
[44] Abadi, M.S.E. and Shafiee, M.S., "Diffusion normalized subband adaptive algorithm for distributed estimation employing signed regressor of input signal", *Digit. Signal Process*, **70**, pp. 73-83 (2017).
[45] Kumar, K., Pandey, R., Bhattacharjee, S.S., et al. "Exponential Hyperbolic Cosine Robust Adaptive Filters for Audio Signal Processing", *IEEE Signal Processing Letters*, **28**, pp. 1410-1414 (2021).
[46] Kumar, K., Karthik, M.L.N.S., Bhattacharjee, S.S., et al. "Affine Projection Champernowne Algorithm for Robust Adaptive Filtering", *IEEE Tras. Circuits Syst. II*, **69**, pp. 1947-1951 (2022).

Biographies

Hossein Tavakoli accomplished the BS program, with honors, in Electrical Engineering from Hamedan University of Technology in 2020, and he has been studying for the MS degree in Integrated Electronic Circuits field at Shahid Rajaee Teacher Training University in Tehran since 2021. He has started his voyage of discovery in the research field with adaptive filtering and signal processing, and he is enthusiastic about pursuing this field. His research interests include non-linear adaptive filter processing, Spline adaptive filters, and digital image processing.

Mohammad Shams Esfand Abadi obtained the BS degree in Electrical Engineering from Mazandaran University, Mazandaran, Iran and the MS degree in the same field of study from Tarbiat Modares University, Tehran, Iran in 2000 and 2002, respectively, and the PhD degree in biomedical engineering from Tarbiat Modares University in 2007. Since 2004, he has been with the Faculty of Electrical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, where he is currently a Professor. His research interests include digital image processing, digital filter theory, adaptive distributed networks, and adaptive filter algorithms.

Figure and table captions

Table captions

Caption	**Table No.**
The SR-SAF-NLMS algorithm	Table 1
The computational complexity of SAF-NLMS algorithm	Table 2
The computational complexity of SR-SAF-NLMS algorithm	Table 3
The number of multiplications, divisions, and additions at each iteration	Table 4
The CPU run-time for each algorithms	Table 5
The SR-SAF-NLMS algorithms	Table 6
The Steady-state MSE for different values of μ (AR(1), $a = 0.5$)	Table 7
The Steady-state MSE for different SNR values (AR(1), $a = 0.5$)	Table 8

Figure captions

Caption	**Figure No.**
Structure of the desired signal generation.	Figure 1
Structure of spline adaptive filter.	Figure 2
Structure of SR spline adaptive filter.	Figure 3
The number of multiplications and divisions versus filter length.	Figure 4
The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 1, input signal: AR(1), $a = 0.5$).	Figure 5
The MSE learning curves of SR-SAF-NLMS1 algorithm for different noise distributions (Experiment 1, input signal: AR(1), $a = 0.5$).	Figure 6
The MSE learning curves of SR-SAF-NLMS1 algorithm for various Δx values (Experiment 1, input signal: AR(1), $a = 0.5$).	Figure 7
The MSE learning curves of SAF-NLMS and SR-SAF-NLMS1 algorithms with CR-spline and B-spline basic matrices (Experiment 1, input signal: AR(1), $a = 0.5$).	Figure 8
The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS-1 algorithms for tracking performance (Experiment 1, input signal: AR(1), $a = 0.5$).	Figure 9
The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 2, input signal: AR(1), $a = 0.5$).	Figure 10

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 2, input signal: AR(1), $a = 0.95$). Figure 11

The MSE learning curves of SR-SAF-NLMS-3 algorithm for different μ values (Experiment 2, input signal: AR(1), $a = 0.5$). Figure 12

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms for tracking performance (Experiment 2, input signal: AR(1), $a = 0.5$). Figure 13

The steady-state MSE values versus step-size (μ) (Experiment 2, input signal: AR(1), $a = 0.5$). Figure 14

The steady-state MSE values versus SNR (Experiment 2, input signal: AR(1), $a = 0.5$). Figure 15

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 3, input signal normalization condition:[0,0.1]). Figure 16

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 3, input signal normalization condition:[-0.1,0.1]). Figure 17

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms (Experiment 3, input signal normalization condition:[-0.25,0.25]). Figure 18

The Comparison of MSE learning curve of SR-SAF-NLMS-1 algorithm with different input normalization conditions. Figure 19

Impulse response of the car echo path. Figure 20

The MSE learning curves of SAF-LMS, SAF-NLMS, and SR-SAF-NLMS algorithms for car echo path unknown system (Experiment 4, input signal: AR(1), $a = 0.5$). Figure 21

Table 1

For $k = 0, 1, \dots$ **do**:

Input signal:

$\mathbf{u}(k) = [u(k), u(k-1), \dots, u(k-M+1)]^T$

Output of the linear filter:

$r(k) = \mathbf{h}^T(k)\mathbf{u}(k)$

Calculate spline local parameters:

$p(k) = \frac{r(k)}{\Delta x} - [\frac{r(k)}{\Delta x}]$

$j(k) = [\frac{r(k)}{\Delta x}] + \frac{Q-1}{2}$

$\mathbf{p}(k) = [p^3(k), p^2(k), p(k), 1]^T$

$\dot{\mathbf{p}}(k) = [3p^2(k), 2p(k), 1, 0]^T$

$\mathbf{q}_j(k) = [q_j(k), q_{j+1}(k), q_{j+2}(k), q_{j+3}(k)]^T$

Generate error signal respect to output of the nonlinear filter:

$e(k) = d(k) - \mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k)$

Update equations:

$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\text{sign}[\mathbf{u}(k)]}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|_1}\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)e(k)$

$\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\text{sign}[\mathbf{C}^T\mathbf{p}(k)]}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|_1}e(k)$

End.

Table 2

Term	$\times$	$\div$	$+$
$r(k) = \mathbf{h}^T(k)\mathbf{u}(k)$	M	0	$M-1$
$p(k) = \frac{r(k)}{\Delta x} - [\frac{r(k)}{\Delta x}]$	0	1	1
$j(k) = [\frac{r(k)}{\Delta x}] + \frac{Q-1}{2}$	0	0	2
$\mathbf{p}(k) = [p^3(k), p^2(k), p(k), 1]^T$	3	0	0
$\dot{\mathbf{p}}(k) = [3p^2(k), 2p(k), 1, 0]^T$	2	0	0
$\mathbf{q}_j(k) = [q_j(k), q_{j+1}(k), q_{j+2}(k), q_{j+3}(k)]^T$	0	0	0
$e(k) = d(k) - \mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k)$	20	0	16
$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)\mathbf{u}(k)}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|^2}e(k)$	$M+9$	M	$2M+4$
$\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\mathbf{C}^T\mathbf{p}(k)}{\epsilon + \|\mathbf{C}^T\mathbf{p}(k)\|^2}e(k)$	5	4	7

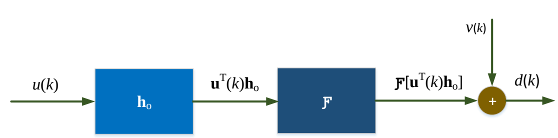

Figure 1:

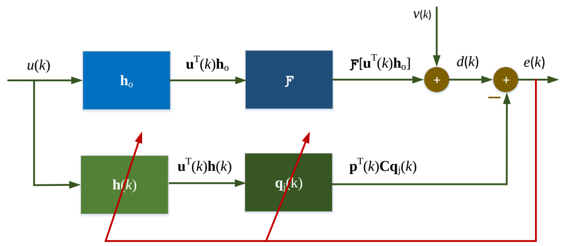

Figure 2:

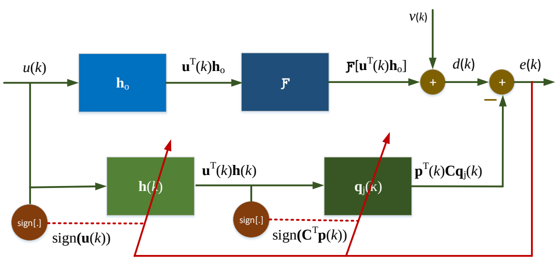

Figure 3:

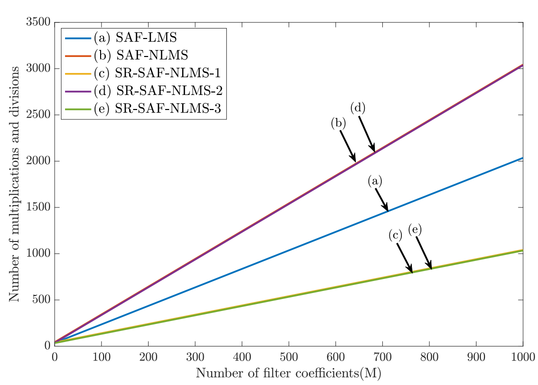

Figure 4:

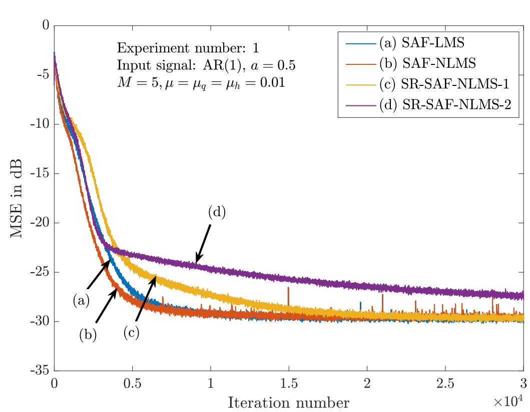

Figure 5:

Table 3

Term	$\times$	$\div$	$+$
$r(k) = \mathbf{h}^T(k)\mathbf{u}(k)$	M	0	$M-1$
$p(k) = \frac{r(k)}{\Delta x} - [\frac{r(k)}{\Delta x}]$	0	1	1
$j(k) = [\frac{r(k)}{\Delta x}] + \frac{Q-1}{2}$	0	0	2
$\mathbf{p}(k) = [p^3(k), p^2(k), p(k), 1]^T$	3	0	0
$\dot{\mathbf{p}}(k) = [3p^2(k), 2p(k), 1, 0]^T$	2	0	0
$\mathbf{q}_j(k) = [q_j(k), q_{j+1}(k), q_{j+2}(k), q_{j+3}(k)]^T$	0	0	0
$e(k) = d(k) - \mathbf{p}^T(k)\mathbf{C}\mathbf{q}_j(k)$	20	0	16
$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\text{sign}[\mathbf{u}(k)]\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)}{\epsilon + \Vert\mathbf{C}^T\mathbf{p}(k)\Vert_1} e(k)$	5	1	$M+5$
$\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\text{sign}[\mathbf{C}^T\mathbf{p}(k)]}{\epsilon + \Vert\mathbf{C}^T\mathbf{p}(k)\Vert_1} e(k)$	1	1	7

Table 4

Algorithm	Multiplications	Divisions	Additions
SAF-LMS	$2M + 35$	1	$3M + 23$
SAF-NLMS	$2M + 39$	$M + 5$	$3M + 29$
SR-SAF-NLMS-1	$M + 35$	6	$2M + 30$
SR-SAF-NLMS-2	$2M + 35$	$M + 2$	$3M + 29$
SR-SAF-NLMS-3	$M + 31$	3	$2M + 30$

Table 5

Algorithm	Time(s)
SAF-LMS	12.5
SAF-NLMS	14.6
SR-SAF-NLMS-1	7.4
SR-SAF-NLMS-2	13.7
SR-SAF-NLMS-3	6.8

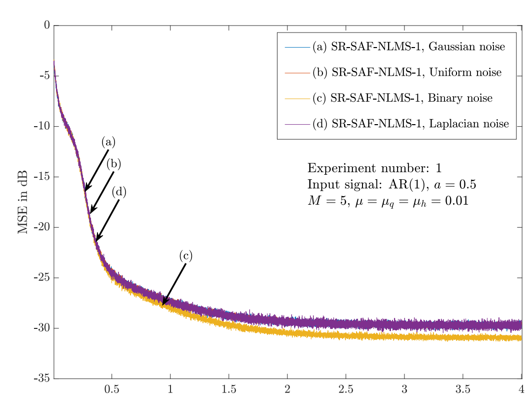

Figure 6:

Table 6

Algorithms	Update equations
SR-SAF-NLMS-1	$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\text{sign}[\mathbf{u}(k)]}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert_1}\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)e(k)$ $\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\mathbf{C}^T\mathbf{p}(k)}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert^2}e(k)$
SR-SAF-NLMS-2	$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)\mathbf{u}(k)}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert^2}e(k)$ $\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\text{sign}[\mathbf{C}^T\mathbf{p}(k)]}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert_1}e(k)$
SR-SAF-NLMS-3	$\mathbf{h}(k+1) = \mathbf{h}(k) + \mu_h \frac{\text{sign}[\mathbf{u}(k)]}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert_1}\dot{\mathbf{p}}^T(k)\mathbf{C}\mathbf{q}_j(k)e(k)$ $\mathbf{q}_j(k+1) = \mathbf{q}_j(k) + \mu_q \frac{\text{sign}[\mathbf{C}^T\mathbf{p}(k)]}{\epsilon+\lVert\mathbf{C}^T\mathbf{p}(k)\rVert_1}e(k)$

Table 7

Algorithm	μ=0.01	μ=0.51	μ=0.96
SAF-LMS	−15.4801	−8.8105	−7.7568
SAF-NLMS	−15.4886	−8.5163	−6.5661
SR-SAF-NLMS-1	−14.4180	−8.6572	−7.8254
SR-SAF-NLMS-2	−15.4744	−1.3365	4.9860
SR-SAF-NLMS-3	−14.8380	0.4511	4.3704

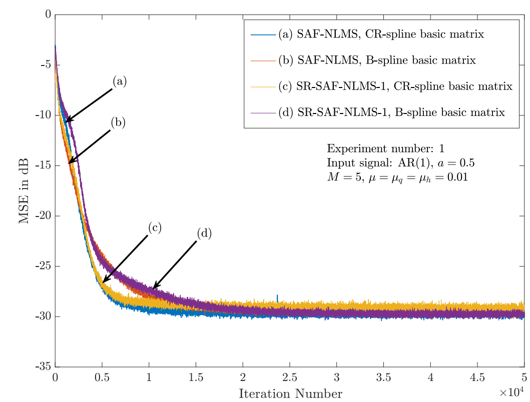

Figure 8:

Table 8

Algorithm	SNR=10	SNR=30	SNR=50
SAF-LMS	−8.8302	−15.3394	−15.5453
SAF-NLMS	−8.8021	−15.3258	−15.4631
SR-SAF-NLMS-1	−8.6416	−14.1640	−14.3133
SR-SAF-NLMS-2	−8.8135	−15.2765	−15.4658
SR-SAF-NLMS-3	−8.7564	−14.8874	−14.9947

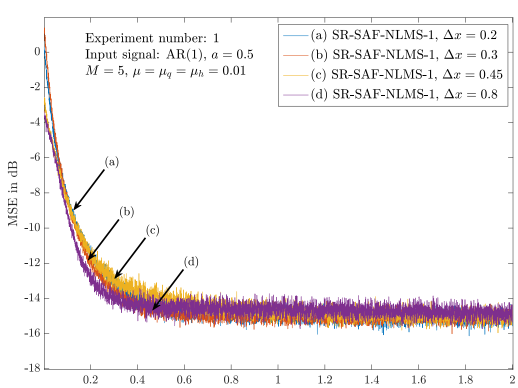

Figure 7:

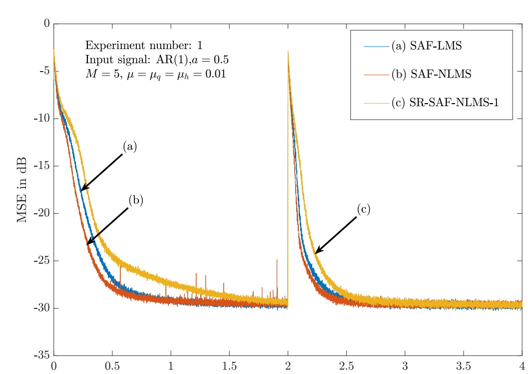

Figure 9:

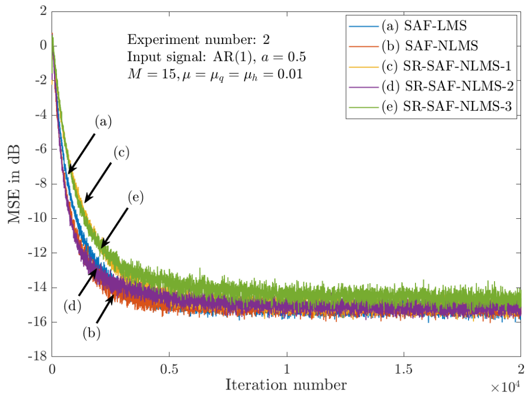

Figure 10:

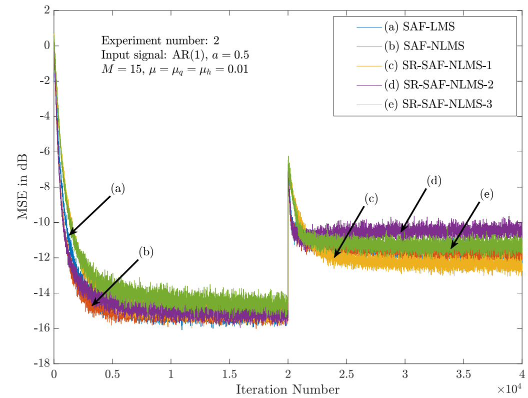

Figure 13:

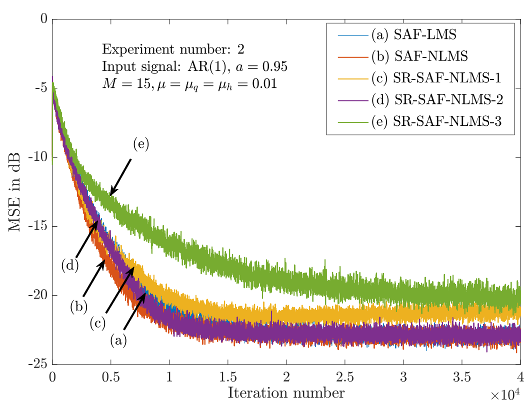

Figure 11:

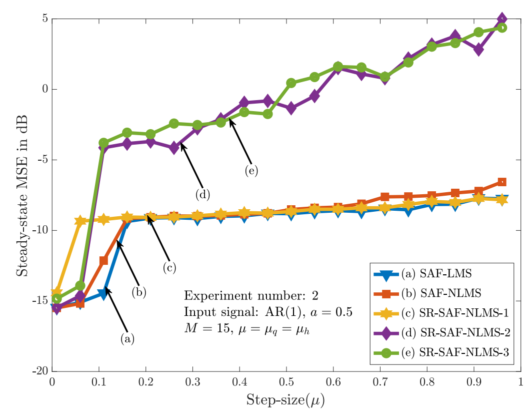

Figure 14:

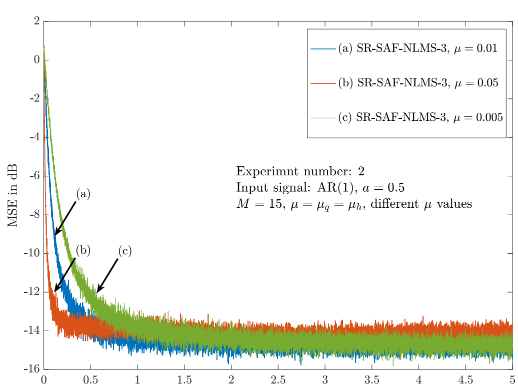

Figure 12:

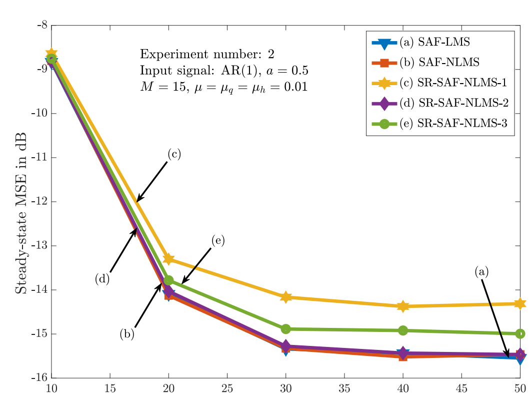

Figure 15:

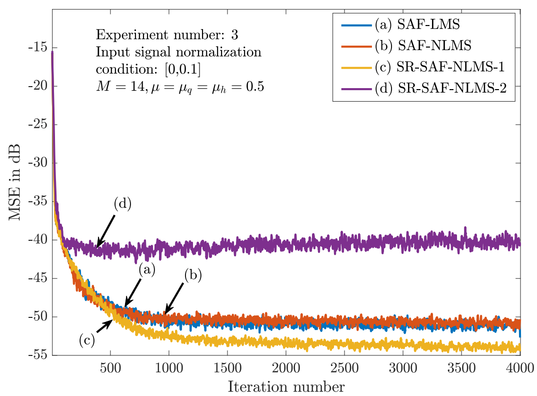

Figure 16:

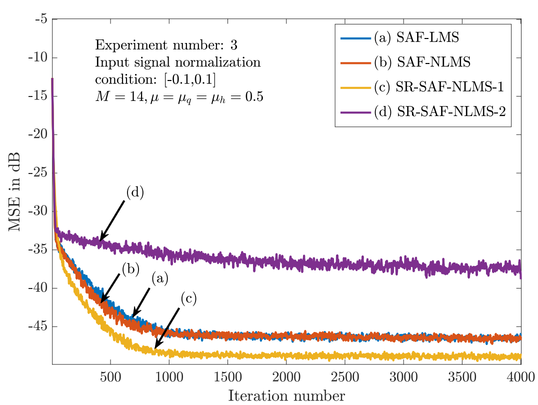

Figure 17:

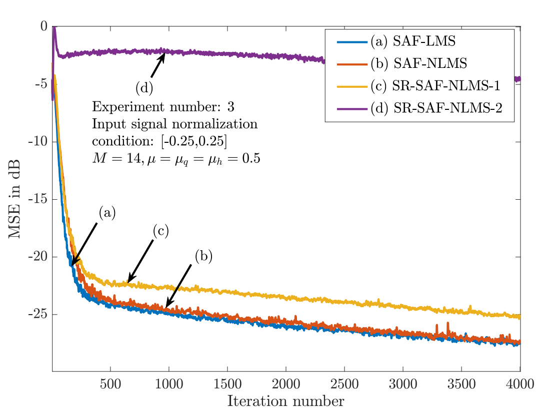

Figure 18:

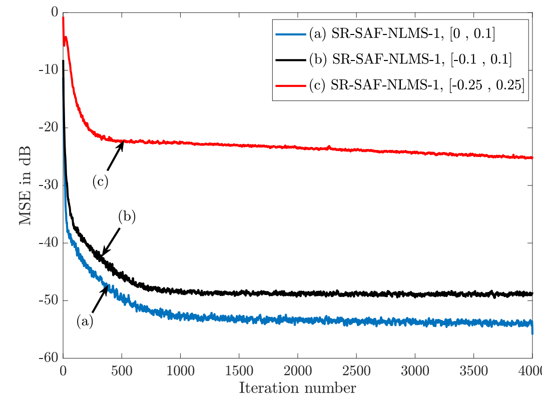

Figure 19:

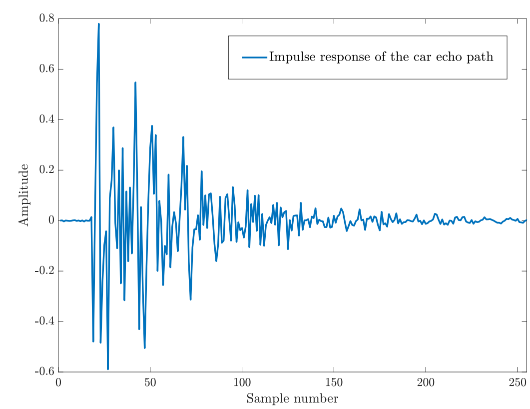

Figure 20:

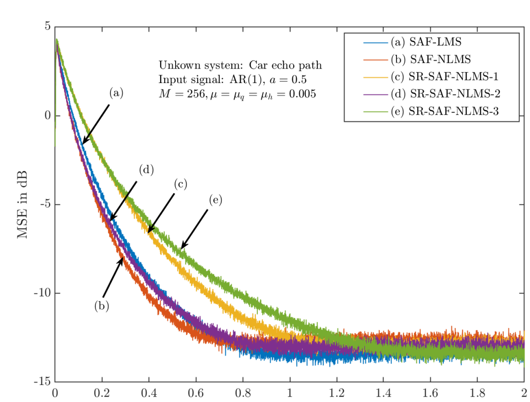

Figure 21: