China Instrument and Control Society
Instrumentation
OPEN ACCESS
INSTRUMENTATION, Vol. 11, No. 3, September 2024 https://doi.org/10.15878/j.instr.202400193

Article

Study on a New Spline Adaptive Filter Using Convex Combination of Exponential Hyperbolic Sine

Yibo Zhao[1,2*], **Yening Li**[1,2]

[1] School of Electronic & Information Engineering, Nanjing University of Information Science & Technology, Nanjing, China

[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing, China

[*] Corresponding author email: yibozhaodn@163.com

Abstract: In this paper, a new spline adaptive filter using a convex combination of exponential hyperbolic sinusoidal is presented. the algorithm convexly combines an exponential hyperbolic sinusoidal Hammerstein spline adaptive filter and a Wiener-type spline adaptive filter to maintain the robustness in non-Gaussian noise environments when dealing with both the Hammerstein nonlinear system and the Wiener nonlinear system. The convergence analyses and simulation experiments are carried out on the proposed algorithm. The experimental results show the superiority of the proposed algorithm to other algorithms.

Keywords: Non-Gaussian noise; spline adaptive filter; Exponential hyperbolic sine cost function

Citation: Zhao, Yibo, and Yening Li. "Study on a New Spline Adaptive Filter Using Convex Combination of Exponential Hyperbolic Sine." Instrumentation 11, no. 3 (September 2024). https://doi.org/10.15878/j.instr.202400193.

0 Introduction

Since the middle and late last century, digital technology has been developing rapidly, and the related techniques and algorithms in the field of signal processing have become a hot research orientation. Adaptive filters are popular because no prior knowledge of the signal is required[1]. A series of adaptive filtering algorithms based on the fundamental principles of the adaptive filter have been proposed by researchers. One of the most well-known adaptive filtering algorithms is the least mean square (LMS) algorithm, which was first proposed by Windrow and Hoff. The LMS algorithm has been extensively researched and implemented due to its low computational complexity and robust stability in Gaussian noise environments. Since the rate of convergence of the LMS algorithm is highly uncertain for the energy magnitude of the input signal, a normalized least mean square (NLMS) algorithm was further proposed[2], in which the normalized term of the input vector is used to normalize the adjustment of the filter weight vector. The NLMS algorithm can be viewed as possessing an adjustable step size that varies with input signal changes, so it has a faster convergence rate than the LMS algorithm.

In echo cancellation, noise control, channel equalization, and other applications, nonlinear problems are inevitably encountered, and linear adaptive filters have limited ability to model nonlinear systems, so Volterra adaptive filter (VAF)[3], Kernel adaptive filter (KAF)[4], function link adaptive filter (FLAF)[5], spline adaptive filter (SAF)[6,7] and other nonlinear adaptive filters are beginning to be widely used. Compared to linear filters, nonlinear filters are more complex, so a balance between convergence performance and complexity needs to be sought in practical applications. The SAF is popular due to its flexibility and low computational complexity. Depending on the cascading order, the SAF can be categorized into Wiener-type spline adaptive filters (WSAF) and Hammerstein-type spline adaptive filters (HSAF). The WSAF is a typical linear-nonlinear model that cascades a nonlinear network after a linear network. Meanwhile, the HSAF is a nonlinear-linear model that cascades a linear network after a nonlinear network. In nonlinear system identification, the HSAF can effectively model Hammerstein-type nonlinear systems, but its performance deteriorates sharply when confronted with Wiener-type nonlinear system. Similarly, the WSAF performs to the HSAF. In general, we cannot know the a priori information of unknown nonlinear systems. To solve this problem, Ref. [8] proposes a convex combination spline adaptive filter by convexly combining different types of spline adaptive filters to maintain the convergence performance in the face of different types of nonlinear systems.

In practical application environments, adaptive filters will inevitably interfere with non-Gaussian noises. Anti-non-gaussian noise interference algorithms have been introduced into the spline adaptive filter to improve the robustness of nonlinear filters. A spline adaptive filter using maximum correlation-entropy criterion (SAF-MCC) is proposed to enhance the robustness of the WSAF, see Ref. [9]. Also, a nonlinear spline adaptive filter based on the robust Geman-McClure estimator (SAF-RGM) proposed in Ref. [10]. In addition, some momentum gradient-like spline adaptive filters[11,12] and spline-priority-optimized adaptive filters[13,14] have been developed to improve the convergence performance of spline adaptive filters.

Motivated by the excellent performance of the exponential hyperbolic sine (EHS) algorithm[15] in linear systems, a convex combination exponential hyperbolic sine spline adaptive filter (CSAF-EHS) is presented in this paper, which convexly combines an exponential hyperbolic sine Hammerstein-type spline adaptive filter (HSAF-EHS) and an exponential hyperbolic sine Wiener-type spline adaptive filter (WSAF-EHS) to maintain convergence performance in the presence of non-Gaussian noise when dealing with Hammerstein-type nonlinear systems or Wiener-type nonlinear systems and to achieve robustness. The main contributions of this paper are as follows:

(i) The exponential hyperbolic sine algorithm is used to propose a convex combination exponential hyperbolic sine spline adaptive filter.

(ii) To ensure the robust convergence of the algorithm, the convergence analysis of the proposed algorithm is performed.

(iii) Simulation experiments are conducted to verify the feasibility, correctness, and effectiveness of the proposed algorithm.

1 Review of the HSAF and WSAF

1.1 The HSAF

The basic structure of the HSAF[6] is shown in Fig.1. It consists of a nonlinear network and a linear adaptive filter, where the nonlinear network is composed of an adaptive lookup table (LUT) and a spline interpolation function, see Fig.2. The output of the nonlinear network and the whole system are represented by $s(n)$ and $y(n)$, respectively. $s(n)$ can be represented by a polynomial function $\varphi(\cdot)$ containing $x(n)$, while $\varphi(\cdot)$ contains the local parameter u and the interval index i, which can be denoted respectively as:

$$u_n = \frac{x(n)}{\Delta x} - \left\lfloor \frac{x(n)}{\Delta x} \right\rfloor \tag{1}$$

$$i = \left\lfloor \frac{x(n)}{\Delta x} \right\rfloor + \frac{Q-1}{2} \tag{2}$$

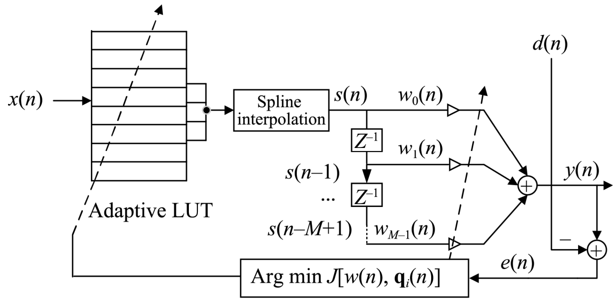

Fig.1 Basic structure of the HSAF

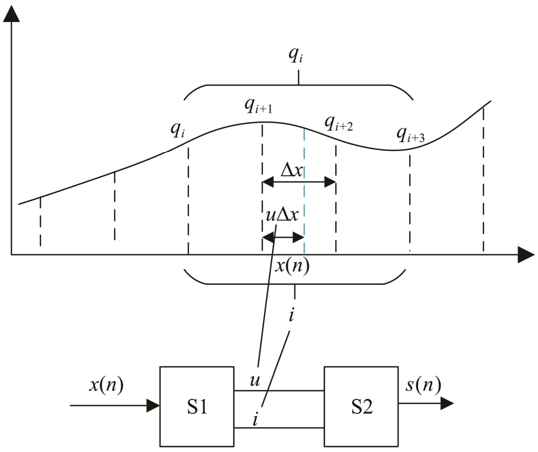

Fig.2 Spline interpolation process

where Δx is the uniform sampling interval between neighboring control points in the LUT, Q is the number of the control points and $\lfloor\cdot\rfloor$ is a floor operation. In the HSAF, the output of the nonlinear network can be denoted as:

$$s(n) = \varphi(n) = \mathbf{u}_n^T \mathbf{C} \mathbf{q}_{i,n} \tag{3}$$

where $\mathbf{u}_n = [u_n^3, u_n^2, u_n^1, 1]^T$, $\mathbf{q}_{i,n}=[q_i, q_{i+1}, q_{i+2}, q_{i+3}]^T$ is a vector of control points consisting of four neighboring control points, and $\mathbf{C}$ is a matrix of order 4, which is usually used to find the optimal approximation by using the CR spline basis matrix, which is:

$$\mathbf{C} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{4}$$

Then, the output of the HSAF can be denoted as:

$$y(n) = \mathbf{w}_n^T \mathbf{s}_n \tag{5}$$

where $\mathbf{w}_n=[w_0(n), w_1(n), \ldots, w_{M-1}(n)]^T$, and $\mathbf{s}_n=[s(n), s(n-1), \ldots, s(n-M+1)]^T$ is the input vector. Then the learning rule for linear filters and control point vectors can be represented as:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_w e(n) \mathbf{s}_n \tag{6}$$

$$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_q e(n) \mathbf{C}^T \mathbf{U}_{i,n} \mathbf{w}_n \tag{7}$$

Where $\mathbf{U}_{i,n}=[\boldsymbol{u}_{i,n}, \boldsymbol{u}_{i,n-1},\ldots,\boldsymbol{u}_{i,n-M+1}]$, and $\mathbf{U}_{i,n}$ comes from the equation $\partial\mathbf{s}_n^T/\partial\mathbf{q}_{i,n}=\mathbf{C}^T\mathbf{U}_{i,n}$. If considered in the same span i or overlapping spans of the current input $x(n)$, there is $\boldsymbol{u}_{i,n-k}=\boldsymbol{u}_{n-k}(k=0,1,2\ldots,M-1)$, otherwise it is considered as a zero vector.

1.2 The WSAF

The basic structure of the WSAF[7] is shown in Fig.3. The input signal $\mathbf{x}_n=[x(n), x(n-1), x(n-M+1)]^T$ passes through an input linear network to obtain the output as:

$$s(n)=\mathbf{w}_n^T\mathbf{x}_n \tag{8}$$

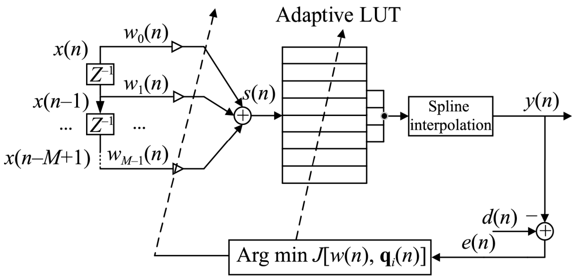

Fig.3 Basic structure of the WSAF

Spline interpolation similar to the principle of Fig. 2, and then we can get u and i:

$$u=\frac{s(n)}{\Delta x}-\left\lfloor\frac{s(n)}{\Delta x}\right\rfloor \tag{9}$$

$$i=\left\lfloor\frac{s(n)}{\Delta x}\right\rfloor+\frac{Q-1}{2} \tag{10}$$

The output of the WSAF can be represented as:

$$y(n)=\mathbf{u}_n^T\mathbf{C}\mathbf{q}_{i,n} \tag{11}$$

Subtracting $y(n)$ from the desired signal $d(n)$ yields the error signal:

$$e(n)=d(n)-y(n) \tag{12}$$

Then the learning rule of the linear filter weight vector and control point vector are respectively denoted as:

$$\mathbf{w}_{n+1}=\mathbf{w}_n+\mu_w e(n)\dot{\mathbf{u}}_n^T\mathbf{C}\mathbf{q}_{i,n}\mathbf{x}_n/\Delta x \tag{13}$$

$$\mathbf{q}_{i,n+1}=\mathbf{q}_{i,n}+\mu_q e(n)\mathbf{C}^T\mathbf{u}_n \tag{14}$$

where $\dot{\mathbf{u}}_n^T$ comes from the derivation of $\mathbf{u}_n$ with respect to u_n.

2 The proposed HSAF-EHS and WSAF-EHS

Denote the parameters of the HSAF-EHS and WSAF-EHS by $\mathbf{U}_{j,i,n}$, $\mathbf{u}_{j,n}$, $\mathbf{s}_{i,n}$, $\mathbf{w}_{j,n}$, $\mathbf{q}_{j,i,n}$, λ_j, $e_j(n)$, $y_j(n)$ respectively, where j is the index of the two algorithms. Cost functions for the HSAF-EHS and WSAF-EHS are respectively defined as:

$$J(\mathbf{w}_{1,n}, \mathbf{q}_{1,i,n})=1-\mathrm{E}[\exp[-\sinh^2(\lambda_1 e_1(n))]] \tag{15}$$

$$J(\mathbf{w}_{2,n}, \mathbf{q}_{2,i,n})=1-\mathrm{E}[\exp[-\sinh^2(\lambda_2 e_2(n))]] \tag{16}$$

Where λ_1 and λ_2 are the scale factors of HSAF-EHS and WSAF-EHS, respectively, and :

$$e_1(n)=d(n)-y_1(n) \tag{17}$$

$$e_2(n)=d(n)-y_2(n) \tag{18}$$

Further, the partial derivative of $J(\mathbf{w}_{1,n}, \mathbf{q}_{1,i,n})$ and $J(\mathbf{w}_{2,n}, \mathbf{q}_{2,i,n})$ to $\mathbf{w}_{1,n}$ and $\mathbf{w}_{2,n}$ respectively can get the updated gradient of the weight vector as:

$$\begin{aligned}\nabla_{\mathbf{w}_{1,n}}&=\frac{\partial J(\mathbf{w}_{1,n},\mathbf{q}_{1,i,n})}{\partial\mathbf{w}_{1,n}}\\&=-e_{1,\mathrm{d}}(n)\frac{\partial y(n)}{\partial\mathbf{w}_{1,n}^T}\\&=-e_{1,d}(n)\mathbf{s}_{1,n}\end{aligned} \tag{19}$$

$$\begin{aligned}\nabla_{\mathbf{w}_{2,n}}&=\frac{\partial J(\mathbf{w}_{2,n},\mathbf{q}_{2,i,n})}{\partial\mathbf{w}_{2,n}}\\&=-e_{2,d}(n)\frac{\partial y_2(n)}{\partial u_2}\frac{\partial u_2}{\partial s_{2,n}}\frac{\partial s_{2,n}}{\partial\mathbf{w}_{2,n}}\\&=-e_{2,d}(n)\dot{\mathbf{u}}_{2,n}{}^T\mathbf{C}\mathbf{q}_{2,i,n}\mathbf{x}_n/\Delta x\end{aligned} \tag{20}$$

Where

$$e_{1,d}(n)=\exp[-\sinh^2(\lambda_1 e_1(n))]\sinh^2(\lambda_1 e_1(n)) \tag{21}$$

$$e_{2,d}(n)=\exp[-\sinh^2(\lambda_2 e_2(n))]\sinh^2(\lambda_2 e_2(n)) \tag{22}$$

Using stochastic negative gradient descent, the learning rule for the weight vector of the HSAF-EHS and WSAF-EHS can be derived respectively:

$$\begin{aligned}\mathbf{w}_{1,n+1}&=\mathbf{w}_{1,n}-\mu_{1,w}\nabla_{\mathbf{w}_{1,n}}\\&=\mathbf{w}_{1,n}+\mu_{1,w}e_{1,d}(n)\mathbf{s}_{1,n}\end{aligned} \tag{23}$$

$$\begin{aligned}\mathbf{w}_{2,n+1}&=\mathbf{w}_{2,n}-\mu_{2,w}\nabla_{\mathbf{w}_{2,n}}\\&=\mathbf{w}_{2,n}+\mu_{2,w}e_{2,d}(n)\dot{\mathbf{u}}_{2,n}^T\mathbf{C}\mathbf{q}_{2,i,n}\mathbf{x}_n/\Delta x\end{aligned} \tag{24}$$

where $\mu_{1,w}$ is the learning rate for $\mathbf{w}_{1,n}$, $\mu_{2,w}$ is the learning rate for $\mathbf{w}_{2,n}$. Similarly, $\nabla_{\mathbf{q}_{1,i,n}}$ and $\nabla_{\mathbf{q}_{2,i,n}}$ can be obtained as:

$$\begin{aligned}\nabla_{\mathbf{q}_{1,i,n}}&=\frac{\partial J(\mathbf{w}_{1,n},\mathbf{q}_{1,i,n})}{\partial\mathbf{q}_{1,i,n}}\\&=-e_{1,d}(n)\frac{\partial y_1(n)}{\partial\mathbf{s}_{1,n}}\frac{\partial\mathbf{s}_{1,n}^T}{\partial\mathbf{q}_{1,i,n}}\\&=-e_{1,d}(n)\mathbf{C}^T\mathbf{U}_{1,i,n}\mathbf{w}_{1,n}\end{aligned} \tag{25}$$

$$\begin{aligned}\nabla_{\mathbf{q}_{2,i,n}}&=\frac{\partial J(\mathbf{w}_{2,n},\mathbf{q}_{2,i,n})}{\partial\mathbf{q}_{2,i,n}}\\&=-e_{2,d}(n)\frac{\partial y_2(n)}{\partial\mathbf{q}_{2,i,n}}\\&=-e_{2,d}(n)\mathbf{C}^T\mathbf{u}_{2,n}\end{aligned} \tag{26}$$

Then, the updating iteration of the control point vectors $\mathbf{q}_{1,i,n}$ and $\mathbf{q}_{2,i,n}$ can be denoted respectively as:

$$\begin{aligned}\mathbf{q}_{1,i,n+1}&=\mathbf{q}_{1,i,n}-\mu_{1,q}\nabla_{\mathbf{q}_{1,i,n}}\\&=\mathbf{q}_{1,i,n}+\mu_{1,q}e_{1,d}(n)\mathbf{C}^T\mathbf{U}_{1,i,n}\mathbf{w}_{1,n}\end{aligned} \tag{27}$$

$$\begin{aligned}\mathbf{q}_{2,i,n+1}&=\mathbf{q}_{2,i,n}-\mu_{2,q}\nabla_{\mathbf{q}_{2,i,n}}\\&=\mathbf{q}_{2,i,n}+\mu_{2,q}e_{2,d}(n)\mathbf{C}^T\mathbf{u}_{2,n}\end{aligned} \tag{28}$$

3 The CSAF-EHS

The convex combination of two filters with different characteristics is a popular solution in the application of adaptive filters. Convex combining filters have a convex combining factor that is adaptively adjusted to the error signal and can combine the advantages of the two sub-filters to achieve better performance than either of the two separate sub-filters. To maintain the convergence performance in both the Hammerstein-type nonlinear system and the Wiener-type nonlinear system, the CSAF-EHS using HSAF-EHS and WSAF-EHS are presented in Fig.4. Then the whole output is given by:

$$y(n)=\sigma(n)y_1(n)+(1-\sigma(n))y_2(n) \tag{29}$$

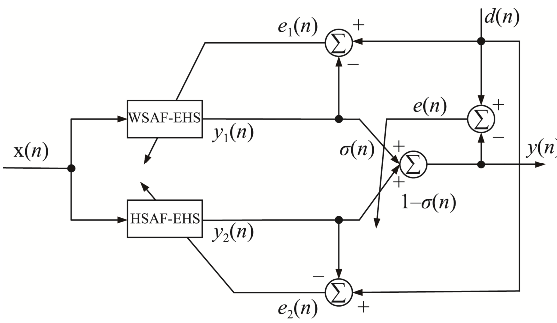

Fig.4 Basic structure of the CSAF-EHS

When the identified system is a Hammerstein-type nonlinear system, $\sigma(n)$ will tend to 0. When the identified system is a Wiener-type nonlinear system, $\sigma(n)$ will tend to 1. The two filters can work together when $\sigma(n)\in(0, 1)$. Therefore, the CSAF will behave well in both of the above nonlinear systems. To satisfy the condition $0\leqslant\sigma(n)\leqslant 1$, $\sigma(n)$ is taken as a sigmoid function[16]:

$$\sigma(n)=\left(\frac{1}{1+e^{-a(n)}}\right) \tag{30}$$

The mixing factor $\sigma(n)$ is updated by the auxiliary parameter $a(n)$. To robustly obtain an updated expression for $a(n)$, a cost function considering the over error signal $e(n)$ is defined using the EHS algorithm as follows:

$$J(n)=1-E[\exp[-\sinh^2-(\lambda_\alpha e(n))]] \tag{31}$$

Take the derivative of $J(n)$ for $e(n)$, we can get:

$$e_\alpha(n)=\exp[-\sinh^2(\lambda_\alpha e(n))]\sinh(\lambda_\alpha e(n)) \tag{32}$$

To get the update gradient, take the partial derivative of $J(n)$ to $a(n)$ and we have:

$$\begin{aligned}\nabla_{a(n)}&=\frac{\partial J(n)}{a(n)}\\&=-e_a(n)\frac{\partial y(n)}{a(n)}\\&=-e_a(n)\big(y_1(n)-y_2(n)\big)\big[\sigma(n)\big(1-\sigma(n)\big)\big]\end{aligned} \tag{33}$$

Using the stochastic gradient descent method, the updated iteration of $a(n)$ can be obtained as:

$$\begin{aligned}a(n+1)&=a(n)-\frac{\mu_a}{r(n)}\nabla_{a(n)}\\&=a(n)+\frac{\mu_a}{r(n)}e_a(n)\big(y_1(n)-y_2(n)\big)\\&\quad\big[\sigma(n)\big(1-\sigma(n)\big)\big]\end{aligned} \tag{34}$$

where μ_α is the iteration step size, $r(n)$[17] is defined as:

$$r(n)=\beta r(n-1)+(1-\beta)y_{NL}^2(n) \tag{35}$$

And the constant β is a smoothing factor. To avoid slowing down the update rate of the auxiliary parameter, the value of $a(n)$ needs to be restricted to the interval [–4, +4] [18].

4 Convergence analysis

The convergence analysis of the algorithm is carried out to ensure the stability of the proposed algorithm. The Taylor series expansions of $e_1(n)$ and $e_2(n)$ with respect to $\mathbf{w}_{1,n}$ and $\mathbf{w}_{2,n}$ are respectively as follows:

$$e_1(n+1)=e_1(n)+\frac{\partial e_1(n)}{\partial\mathbf{w}_{1,n}^T}\Delta\mathbf{w}_{1,n}+\text{h.o.t} \tag{36}$$

$$e_2(n+1)=e_2(n)+\frac{\partial e_2(n)}{\partial\mathbf{w}_{2,n}^T}\Delta\mathbf{w}_{2,n}+\text{h.o.t} \tag{37}$$

where h.o.t denotes the higher-order term of the Taylor series expansion. Combining equations (17), (18), (23), and (24), the following formula can be obtained:

$$e_1(n+1)=\left[1-\mu_{1,w}\frac{\left\|\mathbf{s}_{1,n}\right\|^2 e_{1,d}(n)}{e_1(n)}\right]e_1(n) \tag{38}$$

$$e_2(n+1)=\left[1-\mu_{2,w}\frac{(\dot{\mathbf{u}}_{2,n}{}^T\mathbf{C}\mathbf{q}_{2,i,n}/\Delta x)^2\left\|\mathbf{x}_n\right\|^2 e_{2,d}(n)}{e_2(n)}\right]e_2(n) \tag{39}$$

The convergence of the algorithms can be satisfied when $|e_1(n+1)|\leq|e_1(n)|$ and $|e_2(n+1)|\leq|e_2(n)|$. Thus, the convergence range of $\mu_{1,w}$ and $\mu_{2,w}$ is respectively given by:

$$0<\mu_{1,w}\leqslant\frac{2e_1(n)}{\left\|\mathbf{s}_{1,n}\right\|^2 e_{1,d}(n)} \tag{40}$$

$$0<\mu_{2,w}\leqslant\frac{2e_2(n)}{(\dot{\mathbf{u}}_{2,n}{}^T\mathbf{C}\mathbf{q}_{2,i,n}/\Delta x)^2\left\|\mathbf{x}_n\right\|^2 e_{2,d}(n)} \tag{41}$$

To get the range that $\mu_{1,q}$ and $\mu_{2,q}$ when the two algorithms converge, the Taylor series expansions of $e_1(n)$ and $e_2(n)$ to $\mathbf{q}_{1,i,n}$ and $\mathbf{q}_{2,i,n}$ are as follows:

$$e_1(n+1)=e_1(n)+\frac{\partial e_1(n)}{\partial\mathbf{q}_{1,i,n}}\Delta\mathbf{q}_{1,i,n}+\text{h.o.t} \tag{42}$$

$$e_2(n+1)=e_2(n)+\frac{\partial e_2(n)}{\partial\mathbf{q}_{2,i,n}}\Delta\mathbf{q}_{2,i,n}+\text{h.o.t} \tag{43}$$

And combining equations (17), (18), (27) and (28), we can get:

$$e_1(n+1)=\left[1-\mu_{1,q}\frac{\left\|\mathbf{C}^{\mathrm{T}}\mathbf{U}_{1,i,n}\mathbf{w}_{1,n}\right\|^2 e_{1,d}(n)}{e_1(n)}\right]e_1(n) \quad (44)$$

$$e_2(n+1)=\left[1-\mu_{2,q}\frac{\left\|\mathbf{C}^{\mathrm{T}}\mathbf{u}_{2,n}\right\|^2 e_{2,d}(n)}{e_2(n)}\right]e_2(n) \quad (45)$$

Similarly, the range of $\mu_{1,q}$ and $\mu_{2,q}$ to satisfy the convergence condition is:

$$0<\mu_{1,q}\leqslant\frac{2e_1(n)}{\left\|\mathbf{C}^T\mathbf{U}_{1,i,n}\mathbf{w}_{1,n}\right\|^2 e_{1,d}(n)} \quad (46)$$

$$0<\mu_{2,q}\leqslant\frac{2e_2(n)}{\left\|\mathbf{C}^T\mathbf{u}_{2,n}\right\|^2 e_{2,d}(n)} \quad (47)$$

To ensure the proper functioning of the convex combination algorithm, it is also necessary to obtain a range of step sizes for μ_α. Thus, define a Taylor series expansion of the total error vector $e(n)$ to $a(n)$ as:

$$e(n+1)=e(n)+\frac{\partial e(n)}{\partial a(n)}\Delta a(n)+h.o.t \quad (48)$$

Substituting Eqs. (10) and (23) into (34), we have:

$$e(n+1)=\left[1-\frac{\mu_a}{r(n)}\frac{e_a(n)(y_1(n)-y_2(n))\left[\sigma(n)(1-\sigma(n))\right]}{e(n)}\right]e(n) \quad (49)$$

The condition that needs to be satisfied to maintain a stable update of $a(n)$ is:

$$\left|1-\frac{\mu_a}{r(n)}\frac{e_a(n)\left(y_1(n)-y_2(n)\right)\left[\sigma(n)\left(1-\sigma(n)\right)\right]}{e(n)}\right|\leqslant 1 \quad (50)$$

Thus, the range of μ_α is given by:

$$0<\mu_a\leqslant\frac{2r(n)e(n)}{e_a(n)\left(y_1(n)-y_2(n)\right)\left[\sigma(n)\left(1-\sigma(n)\right)\right]} \quad (51)$$

5 Experiments

The simulation experiment is carried out in MATLAB software. The nonlinear system to be identified is a classical Hammerstein system[6] or a Wiener system, where the Hammerstein system contains an unknown vector $\mathbf{w}_1$*=[0.6, –0.4, 0.25, –0.15, 0.1, –0.05, 0.001]^T with length 7, and the lookup table vector $\mathbf{q}_1$*={–2.2, 2.0, …, –0.8, –0.91, –0.40, –0.20, –0.05, 0.0, –0.40, 0.58, 1.0, 1.0, 1.2, …, 2.0, 2.2}^{T} with length 23. The Wiener system contains an unknown vector $\mathbf{w}_2$*=[0.3, –0.1,0.7,–0.15,0.1,–0.2,0.01]^T [8] with length 7, and the lookup table vector $\mathbf{q}_2$*={–2.2, 2.0, …, –0.8, –0.91, –0.42, –0.01, –0.1, 0.1, –0.15, 0.58, 1.2, 1.0, 1.2, …, 2.0, 2.2}^{T} [11] with length 23. The sampling intervals for both unknown systems are Δx=0.2. All the control spline vectors in the experiment are set initially as a vector $\mathbf{q}(0)$={–2.2, 2.0, …, 2.0, 2.2}^{T} of length 23, and the filter vectors are set initially as [1, 0, 0, 0, 0, 0, 0]^T. The experiment is performed by adding the output of the unknown system with an SNR=30 dB of zero-mean Gaussian white noise and the α-stable distribution noise signal. The α-stable distribution noise is a typical non-Gaussian noise[19], which is characterized by parameters(α, β, γ, δ), The character index $\alpha\in(0, 2]$, the larger its value, the shorter the algebraic tail of the corresponding distribution, and the weaker the pulse characteristics. When α=2, α-stable distribution noise becomes Gaussian distribution noise. $\beta\in[-1, 1]$ is a symmetry parameter, γ>0 is a dispersion parameter, and $\delta\in R$ is a location parameter. The experimental input signals are $x(n)=\theta x(n-1)+\sqrt{1-\theta^2}a(n)$, where θ=0.1, $a(n)$ is a zero-mean, unit-variance Gaussian white noise signal. All results are averaged over 100 independent experiments. Mean square error (MSE) is used to evaluate the performance of the algorithm in the experiment, where the MSE is denoted as:

$$MSE=10\log_{10}(E[e^2(n)]) \quad (52)$$

5.1 The impact of different l values on algorithms

Fig.5 and Fig.6 respectively show the effects of different scale factors on HSAF-EHS and WSAF-EHS. In the experiment, (α, β, γ, δ)=(1.6, 0, 0.05, 0). The experiment is set up with the other parameters of HSAF-EHS: M=7, μ=0.008, WSAF-EHS:M=7, μ=0.005. Fig.5 shows the convergence curves of the HSAF-EHS and WSAF-EHS algorithms for different scale factors, where the unknown system changes from a Hammerstein system to a Wiener system. In the first half of the iteration, the WSAF-EHS algorithm performs poorly at different scale factors due to the unknown system being a Hammerstein system, while the steady-state error of the HSAF-EHS algorithm does not change significantly when the value of the scale factor is larger, but its convergence speed is significantly faster. In the second half of the iteration, where the unknown system is a Wiener system, the HSAF-EHS algorithm performs poorly for different scale factors, and the steady state error of the WSAF-EHS algorithm is significantly higher when the value of the scale factor of the algorithm is higher, and the speed of its convergence is significantly faster.

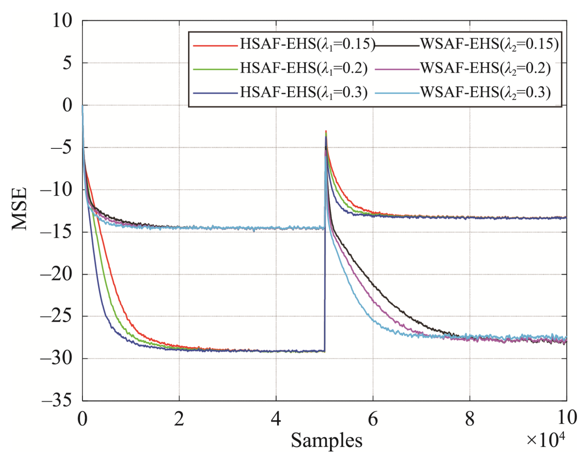

Fig.5 Performance of different λ_1 and λ_2 values when the unknown system changes from the Hammerstein nonlinear system to the Wiener nonlinear system

In Fig.6, the unknown system changes from a Wiener system to a Hammerstein system, while the two algorithms at different scale factors perform similarly to the conclusions obtained in Fig.5.

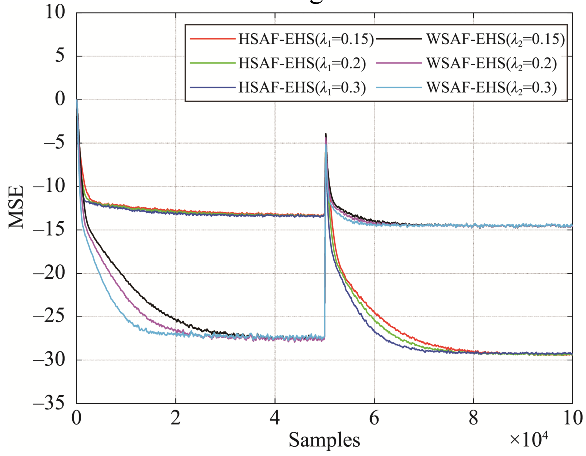

Fig.6 Performance of different λ_1 and λ_2 values when the unknown system changes from the Wiener nonlinear system to the Hammerstein nonlinear system

5.2 Performance of different algorithms

The MSE curves for the different algorithms are shown in Fig.7. (α, β, γ, δ)=(1.6, 0, 0.05, 0) was set in the experiment, and the parameter settings for each algorithm are given in Table 1. Before the first 3×10^4 iterations, the system to be recognized is a nonlinear Hammerstein system and after 3×10^4 iterations, the system changes to a nonlinear Wiener system. In the first half of the iterations, the HSAF-LMS algorithm has a lower MSE curve compared to the WSAF-LMS and WSAF-RGM algorithms, but its curve has some volatility, and the steady state error is higher than that of the CSAF-EHS algorithm proposed in this paper. In the second half of the iteration, the WSAF-RGM and CSAF-EHS uniformly have some resistance to non-Gaussian noise interference, so they have lower MSE curves than the HSAF-LMS and WSAF-LMS. Taking the whole iteration together, the CSAF-EHS has good performance when facing both unknown nonlinear systems under non-Gaussian background noise.

Table 1 The parameter settings for each algorithm

Algorithms	Parameters
HSAF	M=7, μ=0.003
WSAF	M=7, μ=0.005
SAF-RGM	M=7, μ=0.004, σ=0.3
CSAF-EHS	M=7, μ_1=0.008, μ_2=0.005, μ_α=0.1, λ_1=0.3, λ_2=0.3, λ_α=0.5, β=0.9

In addition, as shown in Fig.7, during the first half of the convergence process, $\sigma(n)$ converges towards 1, and the value of $\sigma(n)$ produces a downward fluctuation because the performance advantage of HSAF-EHS at the early stage of convergence is not obvious or even lags behind that of WSAF-EHS, which makes the CSAF-EHS to maintain the convergence performance. In the second half of the iteration process, the value of $\sigma(n)$ tends to be 0. This verifies that $\sigma(n)$ can adapt itself, which makes the CSAF-EHS continuously maintain the convergence performance.

Only the non-Gaussian noise parameter was adjusted to (α, β, γ, δ)=(1.8, 0, 0.05, 0), and the rest of the parameters were consistent with those in Fig.7, and the experiment was conducted to obtain the results shown in Fig.8. Similar to the conclusion in Fig.7, the proposed CSAF-EHS algorithm has better performance throughout the iterations, and has good applicability.

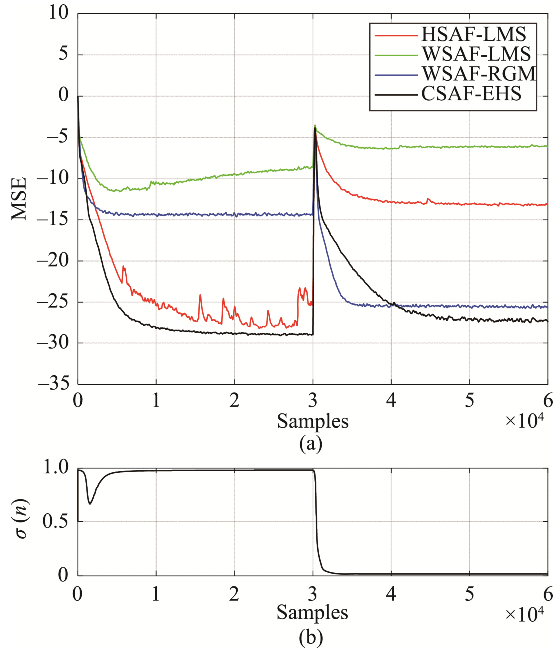

Fig.7 Performance of different algorithms when the unknown system changes from the Hammerstein nonlinear system to the Wiener nonlinear system under α=1.6 (a) MSE curves of different algorithms (b) Variation curve of the $\sigma(n)$ in CSAF-EHS

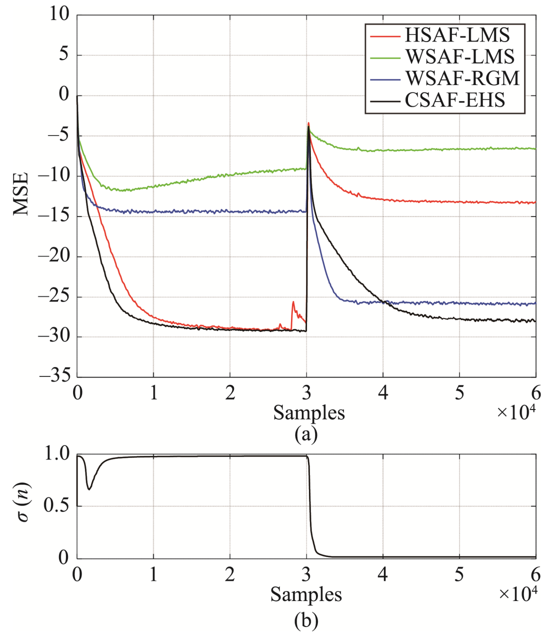

Fig.8 Performance of different algorithms when the unknown system changes from the Hammerstein nonlinear system to the Wiener nonlinear system under α=1.8 (a) MSE curves of different algorithms (b) Variation curve of the $\sigma(n)$ in CSAF-EHS

Figs.9 and 10 show the performance of different algorithms in the face of an unknown system mutating from a Wiener system to a Hammerstein system for (α, β, γ, δ)=(1.6, 0, 0.05, 0) and (α, β, γ, δ)=(1.8, 0, 0.05, 0), respectively., and the parameters of the algorithm in the experiment remain the same as in Fig.7. CSAF-EHS always maintains its performance only before and after a mutation in the unknown system, having the best convergence performance throughout the iteration. Similar to Fig.7 and Fig.8, the $\sigma(n)$ of CSAF-EHS in Fig.9 and Fig.10 undergoes an upward fluctuation at the beginning of the convergence, while converging to 0 and 1 after that to maintain the convergence performance of WSAF-EHS, respectively.

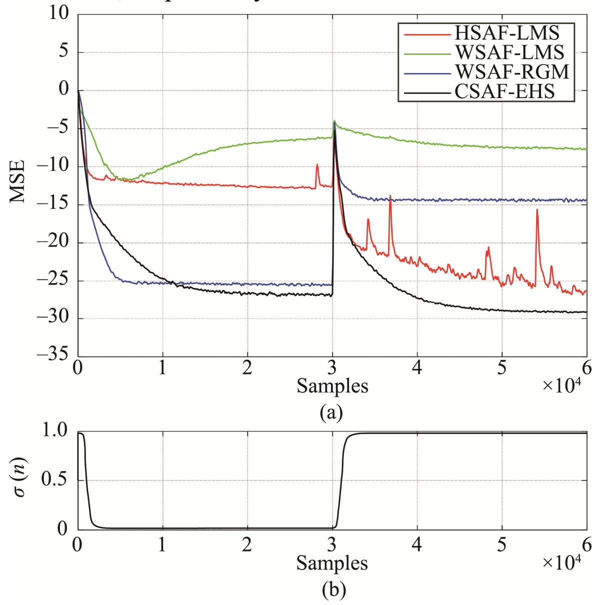

Fig.9 Performance of different algorithms when the unknown system changes from the Wiener nonlinear system to the Hammerstein nonlinear system (a) MSE curves of different algorithms under α=1.6 (b) Variation curve of the $\sigma(n)$ in CSAF-EHS

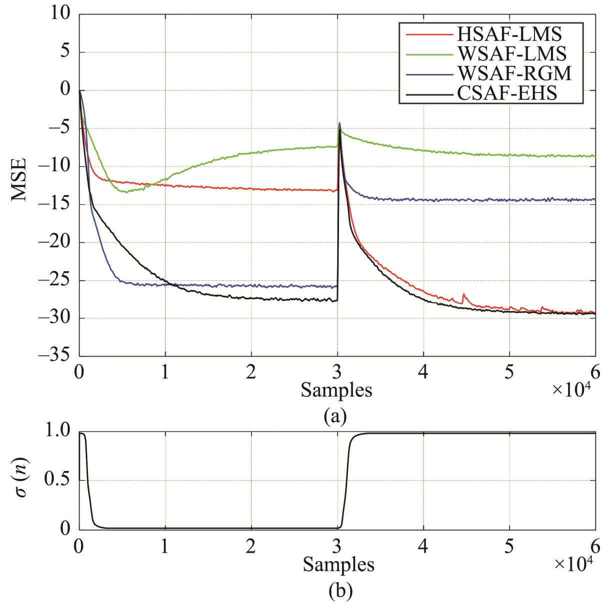

Fig.10 Performance of different algorithms when the unknown system changes from the Wiener nonlinear system to the Hammerstein nonlinear system under α=1.8 (a) MSE curves of different algorithms (b) Variation curve of the $\sigma(n)$ in CSAF-EHS

6 Conclusion

In this paper, a convex combination of exponential hyperbolic sine spline adaptive filters is proposed. The algorithm convexly combines an exponential hyperbolic sinusoidal Hammerstein spline adaptive filter and an exponential hyperbolic sinusoidal Wiener spline adaptive filter, and it has a good convergence performance in non-Gaussian noise environments when dealing with both the Hammerstein-unknown system and the Wiener-unknown system. In addition, an algorithm convergence analysis is performed to ensure its normal convergence. The final simulation experiments demonstrate the effectiveness of the proposed algorithms.

Author Contributions:

Yibo Zhao: Conceptualization; Methodology; Writing - original draft & editing.

Yening Li: Methodology, Writing - original draft.

Funding Information:

This work was supported by the National Natural Science Foundation of China (Grant No. 62371242, Grant No. 61871230)

Data Availability:

The authors declare that the main data supporting the findings of this study are available within the paper and its Supplementary Information files.

Conflict of Interest:

The authors declare no competing interests.

Dates:

Received 6 June 2024; Accepted 10 July 2024; Published online 30 September 2024

References

[1] Widrow, B., Stearns, S. D. (1985). *Adaptive signal processing*. Prentice-Hall.

[2] Slock, D. T. (1993). On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal processing*. 41(9), 2811-2825.

[3] Lu, W., Zhang, L. (2019). Collaborative block-delay Volterra filters for nonlinear acoustic echo cancellation. *Applied Acoustics*. 156, 83-91.

[4] Sankar, S., Kar, A., Burra, S., et al. (2020). Nonlinear acoustic echo cancellation with kernelized adaptive filters. *Applied Acoustics*. 166, 107329 .

[5] Comminiello, D., Scarpiniti, M., Azpicueta-Ruiz, L. A., Arenas-Garcia, J., Uncini, A. (2013). Functional link adaptive filters for nonlinear acoustic echo cancellation. *IEEE Transactions on Audio, Speech, and Language Processing*. 21(7), 1502-1512.

[6] Scarpiniti, M., Comminiello, D., Parisi, R., Uncini A. (2014). Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties. *Signal Processing*. 100, 112-123.

[7] Scarpiniti, M., Comminiello, D., Parisi, R., & Uncini, A. (2013). Nonlinear spline adaptive filtering. *Signal Processing*. 93(4), 772-783.

[8] Scarpiniti, M., Comminiello, D., Uncini, A. (2019). Convex combination of spline adaptive filters. *27th European Signal Processing Conference(EUSIPCO)*. 1-5.

[9] Wang, W., Zhao, H., Zeng, X., & Doğançay, K. (2019). Steady-state performance analysis of nonlinear spline adaptive filter under maximum correntropy criterion. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 67(6), 1154-1158.

[10] Liu, Q., He, Y. (2020). Robust Geman-McClure based nonlinear spline adaptive filter against impulsive noise. *IEEE Access*. 8, 22571-22580.

[11] Yu, T., Li, W., Yu, Y., & de Lamare, R. C. (2021). Robust spline adaptive filtering based on accelerated gradient learning: Design and performance analysis. *Signal Processing*. 183, 107965.

[12] Guo, W., Zhi, Y. (2022). Nonlinear spline adaptive filtering against non-gaussian noise. *Circuits, Systems, and Signal Processing*. 41(1), 579-596.

[13] Guo, W., Zhi, Y. (2023). Nonlinear spline prioritization optimization generalized hyperbolic secant adaptive filtering against alpha-stable noise. *Nonlinear Dynamics*. 111(15), 14351-14363.

[14] Guo, W., Zhi, Y., Feng, K. (2022). Nonlinear spline prioritization optimization adaptive filter with arctangent-exponential hyperbolic cosine. *Nonlinear Dynamics*. 110(1), 611-621.

[15] Radhika, S., Albu, F., Chandrasekar, A. (2022). Robust exponential hyperbolic sine adaptive filter for impulsive noise environments. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 69(12), 5149-5153.

[16] Shen, Z., Tang, L., Yang, L. (2021). Robust normalized subband adaptive filter algorithm with a sigmoid-function-based step-size scaler and its convex combination version. *Mathematical Problems in Engineering*. 2021, 1-11.

[17] Azpicueta-Ruiz, L. A., Figueiras-Vidal, A. R., & Arenas-Garcia, J. (2008). A normalized adaptation scheme for the convex combination of two adaptive filters. *In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. 3301-3304.

[18] Arenas-Garcia, J., Figueiras-Vidal, A. R., Sayed, A. H. (2006). Mean-square performance of a convex combination of two adaptive filters. I*EEE transactions on signal processing*. 54(3), 1078-1090.

[19] Zhao, Y. B., Li, S. H., Ge, Y. X., & Jiang, W. (2022). Research on Dual-Channel Nonlinear Acoustic Echo Cancellation in α-Stable Distributed Noise Environment. *Circuits, Systems, and Signal Processing*. 41(10), 5614-5631.