Widely Linear Complex-Valued Spline-Based Algorithm for Nonlinear Filtering

1st Long Shi
School of Computing and Artificial Intelligence
Southwestern University of Finance and Economics
Chengdu, China
shilong@swufe.edu.cn

2nd Lu Shen
School of Physics,Engineering and Technology
University of York
York, U.K
lu.shen@york.ac.uk

3rd Yuriy Zakharov
School of Physics,Engineering and Technology
University of York
York, U.K
yury.zakharov@york.ac.uk

4th Rodrigo C. de Lamare
The CETUC
PUC-Rio
Rio de Janeiro, Brazil
rodrigo.delamare@york.ac.uk

***Abstract*—Over the past decade, spline adaptive filtering (SAF) has attracted much attention in nonlinear signal processing. In this paper, we investigate the WL complex-valued SAF (WL-CSAF) algorithm in the complex domain based on the cascaded architecture combining a widely linear (WL) model and a nonlinear look-up table (LUT), which is capable of exploiting the full second-order information of signals. Moreover, the steady-state mean square error (MSE) is analyzed to provide insights into the theoretical behavior of the WL-CSAF algorithm. Simulations of nonlinear system identification scenarios demonstrate the superiority of WL-CSAF against known algorithms, and validate the accuracy of the theoretical predictions.**

***Index Terms*—complex domain, mean-square error, nonlinear filtering, spline adaptive filter, widely linear**

I. Introduction

Adaptive filtering algorithms are widely used for online learning in the field of signal processing. They have received considerable critical attention in a variety of applications, such as beamforming [1], frequency estimation in power systems [2], wind profile prediction [3], underwater acoustic systems [4], etc.

In recent years, spline adaptive filtering (SAF) has become very popular within the field of nonlinear filtering due to its simplicity [5]. Spline is a polynomial-based method that minimizes a smoothness penalty, which is different from the probability-based Gaussian process that specifies a covariance function [6]. The architecture of SAF is comprised of the combination of a finite impulse response (FIR) filter and a spline-interpolated adaptable look-up table (LUT). Therefore, SAF is actually a cascade model also known as linear-nonlinear (LN) block-oriented model. In [7], the authors analyzed the steady-state performance of the SAF algorithm. In [8], the generalized spline nonlinear adaptive filter (GSNAF) was designed for multiple-input and multiple-output scenarios. In [9], an interval variable step-size (IVSS) scheme was proposed for SAF to speed up the algorithm convergence in the block-oriented system identification. To combat impulsive noise, some robust SAF algorithms [10]–[12] have also been investigated. In [13], the frequency domain spline adaptive filter (FDSAF) was proposed to improve the computing efficiency. So far, there is scarce literature studying complex-valued SAF besides [14], where Campo *et al*. proposed the spline-based model for complex I/Q signals in the context of nonlinear radio communication systems.

In this paper, based on the cascaded architecture combining a widely linear (WL) model and a nonlinear LUT, we investigate the WL complex-valued SAF algorithm (WL-CSAF), which is capable of exploiting the full second-order information of signals. As the WL model is an augmented variant of the strictly linear (SL) model, the WL-CSAF algorithm can be easily transformed into the SL complex-valued SAF (SL-CSAF) algorithm as long as the WL model reduces to the SL model. The steady-state mean-square error (MSE) behavior of WL-CSAF is theoretically analyzed. Simulations in a nonlinear system identification scenario are conducted to test the performance of the proposed WL-CSAF algorithm. It is shown to outperform the SL-CSAF, complex-valued LMS (CLMS) [15] and WL-CLMS [16] algorithms. The presented theoretical analysis is verified to provide accurate prediction results.

Notation: Boldface letters denote vectors and matrices. Superscripts $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ are complex conjugate, transpose, and Hermitian transpose, respectively. The symbols $|\cdot|$ and $\|\cdot\|^2$ stand for the absolute operator and Euclidean norm, respectively. The symbols $\mathrm{Re}\{\cdot\}$ and $\dot{(\cdot)}$ denote the real part and the derivative, respectively.

The work of Long Shi was partially supported by National Natural Science Foundation of China (Grant: 62201475) and Fundamental Research Funds for the Central Universities (220810004005040272). The work of Yuriy Zakharov and Lu Shen was partly supported by the U.K. EPSRC (Grants EP/R003297/1 and EP/V009591/1).

II. Principles of Spline Interpolation

In this section, we introduce the model adopted, review real-valued spline interpolation and the design of WL complex-valued spline interpolation.

A. Review of Real-valued Spline Interpolation

Spline interpolation based on low-order piece-wise polynomials is a commonly used interpolation scheme capable of providing high approximation accuracy at lower complexity compared to other schemes [17]. It allows interpolation between an arbitrary set of points, called *knots*, also known as control points.

The construction of the piece-wise model starts by defining a set of *knots* $\mathbf{Q}_i = [q_{x,i}, q_{y,i}]^T$ for $i = 0, 1, \cdots, Q$, on the plane x — y. The abscissa sequence $\{q_{x,i}\}$ is required to be constrained as: $q_{x,0} < q_{x,1} < \cdots < q_{x,Q}$ [18]. We consider the uniform sampling of the abscissa, since it allows for a simple input-output relation [5], [14], and the region width is defined as $\Delta x = q_{x,i+1} - q_{x,i}$. Given a suitable spline degree P, the spline segment for each region, $[q_{x,i}, q_{x,i+1})$, is defined as an affine combination of $P+1$ spline curves, and the Pth-degree spline basis function is expressed by the Cox-deBoor recursion [19]

$$\begin{aligned} N_i^P(u) =& \frac{u - q_{x,i}}{q_{x,i+P} - q_{x,i}} N_i^{P-1}(u) \\ &+ \frac{q_{x,i+P+1} - u}{q_{x,i+P+1} - q_{x,i+1}} N_{i+1}^{P-1}(u), \end{aligned} \tag{1}$$

where $i = 0, 1, \cdots, Q - P - 1$, and the 0th-degree basis function takes the form

$$N_i^0(u) = \begin{cases} 1, & q_{x,i} \leq u \leq q_{x,i+1} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The basis functions N_i^P for different regions are shifted from each other, i.e.,

$$N_i^P(u) = N_0^P(u - i\Delta x). \tag{3}$$

The spline segment in an arbitrary region is given by

$$\varphi_i(u) = \sum_{k=i-P-1}^{i-1} q_{x,k} N_k^P(u), \tag{4}$$

where $\varphi_i(u)$ is a Pth-degree local polynomial. By substituting (1) into (4), the local polynomial can be obtained as

$$\varphi_i(u) = \mathbf{u}^T \mathbf{C} \mathbf{q}_i, \tag{5}$$

where $\mathbf{u} = [u^P, u^{P-1}, \cdots, u, 1]^T$ is a vector with u being the normalized abscissa value between two knots, $\mathbf{C}$ is a predefined spline basis matrix of size $(P+1) \times (P+1)$, and $\mathbf{q}_i = [q_i, q_{i+1}, \cdots, q_{i+P}]^T$ contains y-axes control points $q_i = q_{y,i}$. In this paper, we consider utilizing an important and widespread spline basis matrix termed as cubic Catmul-Rom (CR)-spline basis matrix that enables a local approximation [20], which is defined as

$$\mathbf{C} = \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}. \tag{6}$$

B. Design of WL Complex-valued Spline Interpolation

Motivated by the WL model, $s[n]$ is originated from:

$$s[n] = \underbrace{\mathbf{h}_n^T \mathbf{x}_n^*}_{s_h[n]} + \underbrace{\mathbf{g}_n^T \mathbf{x}_n}_{s_g[n]}, \tag{7}$$

where $\mathbf{h}_n$ and $\mathbf{g}_n$ are the filter weight vectors. With this in mind, for $s_h[n]$ and $s_g[n]$, the WL SAF calculates independently the local parameter and span index:

$$u_{h,n} = \frac{|s_h[n]|}{\Delta x} - \left\lfloor \frac{|s_h[n]|}{\Delta x} \right\rfloor, \quad i_{h,n} = \left\lfloor \frac{|s_h[n]|}{\Delta x} \right\rfloor + \frac{Q-1}{2}, \tag{8}$$

$$u_{g,n} = \frac{|s_g[n]|}{\Delta x} - \left\lfloor \frac{|s_g[n]|}{\Delta x} \right\rfloor, \quad i_{g,n} = \left\lfloor \frac{|s_g[n]|}{\Delta x} \right\rfloor + \frac{Q-1}{2}. \tag{9}$$

The filter output is composed of two components, $y_h[n]$ and $y_g[n]$, defined from

$$y[n] = \underbrace{\mathbf{u}_{h,n}^T \mathbf{C} \mathbf{q}_{h,i,n}}_{y_h[n]} + \underbrace{\mathbf{u}_{g,n}^T \mathbf{C} \mathbf{q}_{g,i,n}}_{y_g[n]}, \tag{10}$$

where $\mathbf{u}_{h,n}$, $\mathbf{u}_{g,n}$ and $\mathbf{q}_{h,i,n}$, $\mathbf{q}_{g,i,n}$ are the instantaneous realizations of $\mathbf{u}$ and $\mathbf{q}_i$ at time instant n.

III. Proposed WL-CSAF Algorithm

A. Learning Rules of the WL-CSAF Algorithm

The cost function of the adaptive filter is defined as

$$J(\mathbf{h}_n, \mathbf{g}_n, \mathbf{q}_{h,i,n}, \mathbf{q}_{g,i,n}) = e[n] e^*[n], \tag{11}$$

where $e[n] = d[n] - y[n]$. In order to derive the WL-CSAF algorithm we minimize the cost function in (11) using gradient descent optimization with respect to the filter weight vectors. We first consider the learning rule for $\mathbf{h}_n$ that is described by

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \mu_{h,a} \bigtriangledown_{\mathbf{h}_n} J(\mathbf{h}_n, \mathbf{g}_n, \mathbf{q}_{h,i,n}, \mathbf{q}_{g,i,n}), \tag{12}$$

where $\mu_{h,a}$ is the step-size. Since the maximum change of a real-valued cost function of complex variables is in the direction of the conjugate gradient, by invoking Wirtinger Calculus [21], the partial derivative is computed as

$$\frac{\partial J(\mathbf{h}_n, \mathbf{g}_n, \mathbf{q}_{h,i,n}, \mathbf{q}_{g,i,n})}{\partial \mathbf{h}_n^*} = e[n] \underbrace{\frac{\partial e^*[n]}{\partial \mathbf{h}_n^*}}_{\mathbf{\Omega}_{h,1}} + e^*[n] \underbrace{\frac{\partial e[n]}{\partial \mathbf{h}_n^*}}_{\mathbf{\Omega}_{h,2}}, \tag{13}$$

where

$$\begin{aligned}\boldsymbol{\Omega}_{h,1} &= -\frac{\partial y^*_{\mathrm{WL}}[n]}{\partial \mathbf{h}^*_n} = -\frac{\partial y^*_h[n]}{\partial \mathbf{h}^*_n} \\ &= -\left[\frac{\partial y^*_h[n]}{\partial \mathbf{u}_{h,n}}\frac{\partial \mathbf{u}_{h,n}}{\partial s^*_h[n]}\frac{\partial s^*_h[n]}{\partial \mathbf{h}^*_n} + \frac{\partial y^*_h[n]}{\partial \mathbf{u}_{h,n}}\frac{\partial \mathbf{u}_{h,n}}{\partial s_h[n]}\frac{\partial s_h[n]}{\partial \mathbf{h}^*_n}\right] \\ &= -\frac{1}{2\Delta x}\frac{s_h[n]}{|s_h[n]|}\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}^*_{h,i,n}\mathbf{x}_n,\end{aligned} \tag{14}$$

and

$$\begin{aligned}\boldsymbol{\Omega}_{h,2} &= -\frac{\partial y_{\mathrm{WL}}[n]}{\partial \mathbf{h}^*_n} = -\frac{\partial y_h[n]}{\partial \mathbf{h}^*_n} \\ &= -\left[\frac{\partial y_h[n]}{\partial \mathbf{u}_{h,n}}\frac{\partial \mathbf{u}_{h,n}}{\partial s^*_h[n]}\frac{\partial s^*_h[n]}{\partial \mathbf{h}^*_n} + \frac{\partial y_h[n]}{\partial \mathbf{u}_{h,n}}\frac{\partial \mathbf{u}_{h,n}}{\partial s_h[n]}\frac{\partial s_h[n]}{\partial \mathbf{h}^*_n}\right] \\ &= -\frac{1}{2\Delta x}\frac{s_h[n]}{|s_h[n]|}\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}_{h,i,n}\mathbf{x}_n.\end{aligned} \tag{15}$$

Substituting (14) and (15) into (13) gives rise to

$$\begin{aligned}&\frac{\partial J(\mathbf{h}_n, \mathbf{g}_n, \mathbf{q}_{h,i,n}, \mathbf{q}_{g,i,n})}{\partial \mathbf{h}^*_n} \\ &= -\frac{1}{2\Delta x}\frac{s_h[n]}{|s_h[n]|}e[n]\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}_{h,i,n}\mathbf{x}_n \\ &\qquad - \frac{1}{2\Delta x}\frac{s_h[n]}{|s_h[n]|}e^*[n]\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}^*_{h,i,n}\mathbf{x}_n \\ &= -\frac{1}{\Delta x}\frac{s_h[n]}{|s_h[n]|}\mathbf{x}_n\mathrm{Re}\{e[n]\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}^*_{h,i,n}\}.\end{aligned} \tag{16}$$

Therefore, the update of $\mathbf{h}_n$ is given by

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\mu_{h,a}}{\Delta x}\frac{s_h[n]}{|s_h[n]|}\mathbf{x}_n\mathrm{Re}\{e[n]\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}^*_{h,i,n}\}. \tag{17}$$

Following a similar derivation, the update of $\mathbf{g}_n$ takes the form

$$\mathbf{g}_{n+1} = \mathbf{g}_n + \frac{\mu_{g,a}}{\Delta x}\frac{s_g[n]}{|s_g[n]|}\mathbf{x}^*_n\mathrm{Re}\{e[n]\dot{\mathbf{u}}^T_{g,n}\mathbf{C}\mathbf{q}^*_{g,i,n}\}, \tag{18}$$

where $\mu_{g,a}$ is the step-size.

We then obtain the learning rules for both $\mathbf{q}_{h,i,n}$ and $\mathbf{q}_{g,i,n}$:

$$\mathbf{q}_{h,i,n+1} = \mathbf{q}_{h,i,n} + \mu_{q,a}e[n]\mathbf{C}^T\mathbf{u}_{h,n}, \tag{19}$$

$$\mathbf{q}_{g,i,n+1} = \mathbf{q}_{g,i,n} + \mu_{q,a}e[n]\mathbf{C}^T\mathbf{u}_{g,n}, \tag{20}$$

where $\mu_{q,a}$ is the step-size. The proposed WL-CSAF algorithm is summarized in Table I.

B. Computational Complexity

The complexity of the WL-CSAF algorithm is shown in Table II, where multiplications and additions per iteration are counted. In total, the WL-CSAF algorithm requires $4M + 6P^2 + 18P + 20$ multiplications and $4M + 6P^2 + 12P + 2$ additions.

TABLE I
SUMMARY OF THE WL-CSAF ALGORITHM

Initialization: $\mathbf{h}_0$, $\mathbf{g}_0$, $\mathbf{q}_{h,i,0}$, $\mathbf{q}_{g,i,0}$
for $n = 0, 1, \cdots$ **do**
 $s_{\mathrm{WL}}[n] = \mathbf{h}^T_n\mathbf{x}^*_n + \mathbf{g}^T_n\mathbf{x}_n$
 $u_{h,n} = \frac{|s_h[n]|}{\Delta x} - \left\lfloor\frac{|s_h[n]|}{\Delta x}\right\rfloor, \quad i_{h,n} = \left\lfloor\frac{|s_h[n]|}{\Delta x}\right\rfloor + \frac{Q-1}{2}$
 $u_{g,n} = \frac{|s_g[n]|}{\Delta x} - \left\lfloor\frac{|s_g[n]|}{\Delta x}\right\rfloor, \quad i_{g,n} = \left\lfloor\frac{|s_g[n]|}{\Delta x}\right\rfloor + \frac{Q-1}{2}$
 $y_{\mathrm{WL}}[n] = \mathbf{u}^T_{h,n}\mathbf{C}\mathbf{q}_{h,i,n} + \mathbf{u}^T_{g,n}\mathbf{C}\mathbf{q}_{g,i,n}$
 $e[n] = d[n] - y_{\mathrm{WL}}[n]$
 $\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\mu_{h,a}}{\Delta x}\frac{s_h[n]}{|s_h[n]|}\mathbf{x}_n\mathrm{Re}\{e[n]\dot{\mathbf{u}}^T_{h,n}\mathbf{C}\mathbf{q}^*_{h,i,n}\}$
 $\mathbf{g}_{n+1} = \mathbf{g}_n + \frac{\mu_{g,a}}{\Delta x}\frac{s_g[n]}{|s_g[n]|}\mathbf{x}^*_n\mathrm{Re}\{e[n]\dot{\mathbf{u}}^T_{g,n}\mathbf{C}\mathbf{q}^*_{g,i,n}\}$
 $\mathbf{q}_{h,i,n+1} = \mathbf{q}_{h,i,n} + \mu_{q,a}e[n]\mathbf{C}^T\mathbf{u}_{h,n}$
 $\mathbf{q}_{g,i,n+1} = \mathbf{q}_{g,i,n} + \mu_{q,a}e[n]\mathbf{C}^T\mathbf{u}_{g,n}$
end for

IV. STEADY-STATE MSE OF WL-CSAF

To make the subsequent analysis mathematically tractable, we introduce the following assumptions:

A1: The noise $v[n]$ is zero-mean with variance σ^2_v, independent of $\mathbf{x}_n$, $\mathbf{q}_{h,i,n}$ and $\mathbf{q}_{g,i,n}$.

A2: In the steady-state, the error signal is independent of $\mathbf{x}_n$, $\mathbf{C}^T\mathbf{u}_{h,n}$ and $\mathbf{C}^T\mathbf{u}_{g,n}$.

Assumption A1 is widely used in adaptive filtering [22]–[24]. Assumption A2 has been shown to significantly simplify the analysis and provide good prediction [7], [9]. The desired signal $d[n]$ in the WL-CSAF is modeled by

$$d[n] = \mathbf{u}^T_{h,n}\mathbf{C}\mathbf{q}_{h,o} + \mathbf{u}^T_{g,n}\mathbf{C}\mathbf{q}_{g,o} + v[n], \tag{21}$$

where $\mathbf{q}_{h,o}$ and $\mathbf{q}_{g,o}$ denote the control points of the unknown system, and $v[n]$ is the white noise. Using (21) and (10), the error signal in (15) is written as

$$e[n] = \mathbf{u}^T_{h,n}\mathbf{C}\tilde{\mathbf{q}}_{h,i,n}(n) + \mathbf{u}^T_{g,n}\mathbf{C}\tilde{\mathbf{q}}_{g,i,n}(n) + v(n), \tag{22}$$

where $\tilde{\mathbf{q}}_{h,i,n}(n) = \mathbf{q}_{h,o} - \mathbf{q}_{h,i,n}$ and $\tilde{\mathbf{q}}_{g,i,n}(n) = \mathbf{q}_{g,o} - \mathbf{q}_{g,i,n}$. By defining the augmented control point error vector $\tilde{\mathbf{q}}_{a,i,n} = [\tilde{\mathbf{q}}^T_{h,i,n}(n), \tilde{\mathbf{q}}^T_{g,i,n}(n)]^T$, (19) and (20) can be jointly transformed into

$$\tilde{\mathbf{q}}_{a,i,n+1} = \tilde{\mathbf{q}}_{a,i,n} - \mu_{q,a}e[n]\mathbf{C}^T_a\mathbf{u}_{a,n}, \tag{23}$$

where $\mathbf{u}_{a,n} = [\mathbf{u}^T_{h,n}, \mathbf{u}^T_{g,n}]^T$, and

$$\mathbf{C}_a = \begin{bmatrix} \mathbf{C}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^T \end{bmatrix}^T. \tag{24}$$

Squaring (23) results in

$$\begin{aligned}&\|\tilde{\mathbf{q}}_{a,i,n+1}\|^2 \\ &= \left\{\tilde{\mathbf{q}}_{a,i,n} - \mu_{q,a}e[n]\mathbf{C}^T_a\mathbf{u}_{a,n}\right\}^H\left\{\tilde{\mathbf{q}}_{a,i,n} - \mu_{q,a}e[n]\mathbf{C}^T_a\mathbf{u}_{a,n}\right\} \\ &= \|\tilde{\mathbf{q}}_{a,i,n}\|^2 - \mu_{q,a}\left(e[n]\tilde{\mathbf{q}}^H_{a,i,n}\mathbf{C}^T_a u_{a,n} + e^*[n]\mathbf{u}^T_{a,n}\mathbf{C}_a\tilde{\mathbf{q}}_{a,i,n}\right) \\ &\qquad + \mu^2_{q,a}e^2[n]\|\mathbf{C}^T_a\mathbf{u}_{a,n}\|^2.\end{aligned} \tag{25}$$

Taking the expectation of (25) and using $E\|\tilde{\mathbf{q}}_{a,i,n+1}\|^2 = E\|\tilde{\mathbf{q}}_{a,i,n}\|^2$ in the steady-state, we arrive at

$$\begin{aligned}&E\left\{e[n]\tilde{\mathbf{q}}^H_{a,i,n}\mathbf{C}^T_a u_{a,n} + e^*[n]\mathbf{u}^T_{a,n}\mathbf{C}_a\tilde{\mathbf{q}}_{a,i,n}\right\} \\ &\qquad = \mu_{q,a}E\left\{e^2[n]\|\mathbf{C}^T_a\mathbf{u}_{a,n}\|^2\right\}.\end{aligned} \tag{26}$$

TABLE II
COMPUTATIONAL COMPLEXITY OF WL-CSAF

WL-CSAF		
–	multiplications	additions
$s_{\mathrm{wl}}[n]$	$2M$	$2M-1$
$y_{\mathrm{wl}}[n]$	$2P^2+6P+4$	$2P^2+4P+1$
total	$2M+2P^2+6P+4$	$2M+2P^2+4P$
$\mathbf{w}_{a,n+1}$	$2M+2P^2+6P+10$	$2M+2P^2+4P$
$\mathbf{q}_{a,i,n+1}$	$2P^2+6P+6$	$2P^2+4P+2$
total	$2M+4P^2+12P+16$	$2M+4P^2+8P+2$
Total	$4M+6P^2+18P+20$	$4M+6P^2+12P+2$

Utilizing (22) and Assumption A1, the left hand side of (26) can be further transformed into

$$E\{(e[n]-v[n])^*e[n]+e^*[n](e[n]-v[n])\}=2E\{e^2[n]-\sigma_v^2\}. \tag{27}$$

Therefore, we obtain

$$2E\{e^2[n]-\sigma_v^2\}=\mu_{q,a}E\left\{e^2[n]\|\mathbf{C}_a^T\mathbf{u}_{a,n}\|^2\right\}. \tag{28}$$

Applying Assumption A2 to (28), we finally obtain

$$\mathrm{MSE}=E\{e^2[n]\}=\frac{2\sigma_v^2}{2-\mu_{q,a}E\|\mathbf{C}_a^T\mathbf{u}_{a,n}\|^2}, \tag{29}$$

where $E\|\mathbf{C}_a^T\mathbf{u}_{a,n}\|^2$ is given by

$$E\|\mathbf{C}_a^T\mathbf{u}_{a,n}\|^2=E\|\mathbf{C}^T\mathbf{u}_{h,n}\|^2+E\|\mathbf{C}^T\mathbf{u}_{g,n}\|^2. \tag{30}$$

The expectation values of $E\|\mathbf{C}^T\mathbf{u}_{h,n}\|^2$ and $E\|\mathbf{C}^T\mathbf{u}_{g,n}\|^2$ can be obtained from the ensemble average in the simulation trials.

V. SIMULATION

In this section, we will verify the performance of the WL-CSAF algorithm, as well as validate the theoretical analysis in system identification scenarios. Similar to [5], the unknown system is composed of a linear component $\mathbf{w}_o=[0.6-0.6j,-0.4+0.4j,0.25-0.25j,-0.15+0.15j,0.1-0.1j]^T$ with $j=\sqrt{-1}$ and a nonlinear memoryless target function implemented by a 23-point length LUT $\mathbf{q}_o$, interpolated by a uniform third degree spline with an interval sampling $\Delta x=0.2$, and defined as

$$\mathbf{q}_o=\{-2.2,-2,\cdots,0.8,-0.91,-0.42,-0.01,-0.1,\\ 0.1,-0.15,0.58,1.2,1.0,\cdots,2.2\}.$$

The input signal $x[n]$ is generated as

$$x[n]=ax[n-1]+\sqrt{1-a^2}\xi[n], \tag{31}$$

where $\xi[n]$ is a zero-mean white Gaussian random sequence and $0\le a<1$ is a parameter that determines the level of correlation between adjacent samples. The system output is disturbed by an additive white Gaussian noise $v[n]$. The simulation results are obtained by averaging over 100 independent trials. The noise which yields a certain signal to noise ratio (SNR) is added, where SNR is defined as

$$\mathrm{SNR(dB)}=10\log 10\left(\frac{\sigma_\xi^2}{\sigma_v^2}\right), \tag{32}$$

with σ_ξ^2 and σ_v^2 denoting the variances of $\xi[n]$ and $v[n]$.

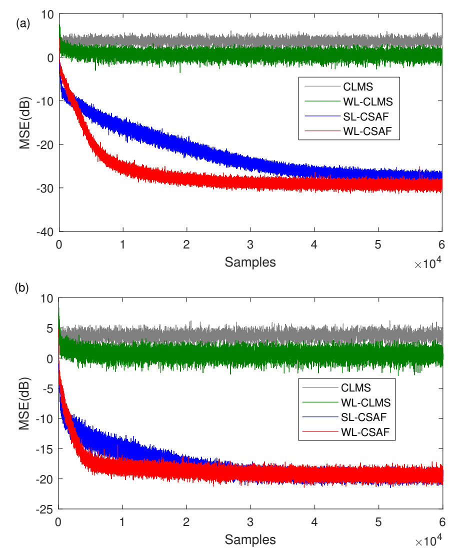

Fig. 1. Comparison of the CLMS, WL-CLMS, SL-CSAF and WL-CSAF algorithms. $x[n]$ is generated by a noncircular signal $\xi[n]$ with variance $\sigma_\xi^2=1$ and complementary variance $\widetilde{\sigma}_\xi^2=0.5$, $a=0.95$, (a) SNR=30dB. (b) SNR=20dB.

In Fig. 1, we compare the WL-CSAF algorithm with CLMS [21], WL-CLMS [2] and SL-CSAF for different SNRs. For the CLMS and WL-CLMS algorithms, their step-sizes are set to 0.005. To ensure a fair comparison, the step-sizes for weight vectors and control points are adjusted to ensure the same steady-state MSE of the investigated algorithms. In the SL-CSAF, step-sizes are set to $\mu_w=0.001$, $\mu_q=0.005$, while in the WL-CSAF, step-sizes are set to $\mu_{h,a}=\mu_{g,a}=0.004$, $\mu_{q,a}=0.005$. It is seen that the WL-CSAF algorithm outperforms the CLMS, WL-CLMS and SL-CSAF algorithms.

Fig. 2 shows the identification results of linear filter weights of the WL-CSAF, where Re(linear filter weights) denotes the real part of linear filter weights, while Im(linear filter weights) denotes the imaginary part of linear filter weights. It is seen that the proposed WL-CSAF algorithm can accurately predict the true weight vector.

The simulated and theoretical steady-state MSE values of the proposed WL-CSAF algorithm are presented in Fig. 3. The theoretical steady-state MSE is calculated using (29) and (30).

The step-sizes for weight vectors are set to $\mu_{h,a} = \mu_{g,a} = 0.004$. Fig. 2(a) shows the steady-state MSE versus the step-size, while Fig. 2(b) shows the steady-state MSE versus SNR. It is clear that the theoretical steady-state MSE values agree well with the simulated values.

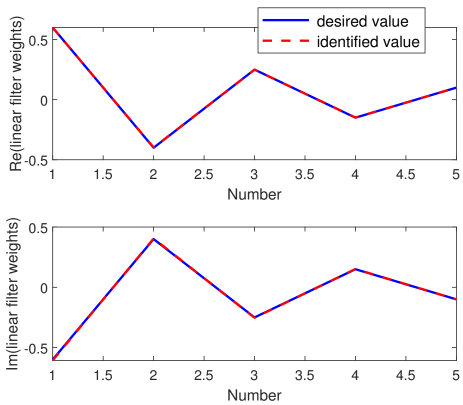

Fig. 2. Identification results of linear filter weights. $x[n]$ is generated by a noncircular signal $\xi[n]$ with variance $\sigma_\xi^2 = 1$ and complementary variance $\widetilde{\sigma}_\xi^2 = 0.5$, $a = 0.95$, SNR=30dB.

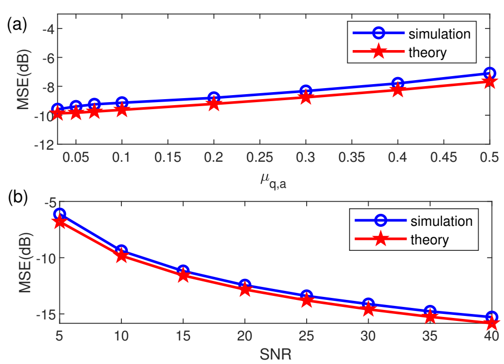

Fig. 3. Simulated and theoretical steady-state MSE of the WL-CLMS algorithm, $x[n]$ is generated by a noncircular signal $\xi[n]$ with variance $\sigma_\xi^2 = 1$ and complementary variance $\widetilde{\sigma}_\xi^2 = 0.5$, $a = 0.95$. (a) Steady-state MSE versus step-size μ_q, with SNR = 10dB. (b) Steady-state MSE versus SNR, with $\mu_{q,a} = 0.05$

VI. Conclusion

In this paper, we have proposed the WL-CSAF algorithm for nonlinear adaptive filtering in the complex domain. The proposed algorithm is based on a cascaded architecture composed of a widely linear filter and a nonlinear LUT, which is capable of exploiting the full second-order information of signals. We have also carried out a theoretical analysis for the WL-CSAF algorithm. Specifically, with some widely used assumptions, the steady-state MSE has been analyzed. Simulations in system identification scenarios verify the high performance of the WL-CSAF algorithm, and validate the accuracy of the theoretical analysis.

References

[1] N. Song, R. C. de Lamare, M. Haardt, and M. Wolf, "Adaptive widely linear reduced-rank interference suppression based on the multistage Wiener filter," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4003–4016, 2012.

[2] Y. Xia, S. C. Douglas, and D. P. Mandic, "Adaptive frequency estimation in smart grid applications: Exploiting noncircularity and widely linear adaptive estimators," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 44–54, 2012.

[3] L. Shi, H. Zhao, Y. Zakharov, B. Chen, and Y. Yang, "Variable step-size widely linear complex-valued affine projection algorithm and performance analysis," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5940–5953, 2020.

[4] L. Shen, Y. Zakharov, L. Shi, and B. Henson, "BEM adaptive filtering for SI cancellation in full-duplex underwater acoustic systems," *Signal Processing*, p. 108366, 2021.

[5] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering," *Signal Processing*, vol. 93, no. 4, pp. 772–783, 2013.

[6] L. Martino and J. Read, "A joint introduction to gaussian processes and relevance vector machines with connections to kalman filtering and other kernel smoothers," *Information Fusion*, vol. 74, pp. 17–38, 2021.

[7] M. Scarpiniti, D. Comminiello, G. Scarano, R. Parisi, and A. Uncini, "Steady-state performance of spline adaptive filters," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 816–828, 2015.

[8] M. Rathod, V. Patel, and N. V. George, "Generalized spline nonlinear adaptive filters," *Expert Systems with Applications*, vol. 83, pp. 122–130, 2017.

[9] L. Yang, J. Liu, Z. Zhao, R. Yan, and X. Chen, "Interval variable step-size spline adaptive filter for the identification of nonlinear block-oriented system," *Nonlinear Dynamics*, vol. 98, no. 3, pp. 1629–1643, 2019.

[10] C. Liu, Z. Zhang, and X. Tang, "Sign normalised spline adaptive filtering algorithms against impulsive noise," *Signal Processing*, vol. 148, pp. 234–240, 2018.

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

[12] V. Patel, S. S. Bhattacharjee, and N. V. George, "A family of logarithmic hyperbolic cosine spline nonlinear adaptive filters," *Applied Acoustics*, vol. 178, p. 107973, 2021.

[13] L. Yang, J. Liu, Q. Zhang, R. Yan, and X. Chen, "Frequency domain spline adaptive filters," *Signal Processing*, vol. 177, p. 107752, 2020.

[14] P. P. Campo, L. Anttila, D. Korpi, and M. Valkama, "Cascaded spline-based models for complex nonlinear systems: Methods and applications," *IEEE Transactions on Signal Processing*, vol. 69, pp. 370–384, 2020.

[15] Y.-M. Shi, L. Huang, C. Qian, and H.-C. So, "Shrinkage linear and widely linear complex-valued least mean squares algorithms for adaptive beamforming," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 119–131, 2014.

[16] Y. Xia and D. P. Mandic, "Augmented performance bounds on strictly linear and widely linear estimators with complex data," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 507–514, 2017.

[17] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties," *Signal Processing*, vol. 100, pp. 112–123, 2014.

[18] L. Schumaker, *Spline functions: basic theory*. Cambridge University Press, 2007.

[19] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-spline techniques*. Springer, 2002, vol. 6.

[20] E. Catmull and R. Rom, "A class of local interpolating splines," in *Computer aided geometric design*. Elsevier, 1974, pp. 317–326.

[21] D. P. Mandic and V. S. L. Goh, *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*. John Wiley & Sons, 2009, vol. 59.

[22] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[23] L. Shi, H. Zhao, X. Zeng, and Y. Yu, "Variable step-size widely linear complex-valued NLMS algorithm and its performance analysis," *Signal Processing*, vol. 165, pp. 1–6, 2019.

[24] L. Shi, H. Zhao, and Y. Zakharov, "Performance analysis of shrinkage linear complex-valued lms algorithm," *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1202–1206, 2019.