

Received April 2, 2021, accepted April 15, 2021, date of publication April 28, 2021, date of current version May 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3076345

Online Sequential Complex-Valued ELM for Noncircular Signals: Augmented Structures and Learning Algorithms

HUI WANG¹, YUANYUAN WANG², SHUAI ZHU¹, AND HUI SHENG ZHANG¹

¹School of Science, Dalian Maritime University, Dalian 116026, China

²School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China

Corresponding author: Huisheng Zhang (zhhuisheng@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61671099 and Grant 61301202.

ABSTRACT Online sequential extreme learning machine (OS-ELM) has become a popular online learning strategy for single-hidden layer feedforward neural networks, and complex-valued signals are ubiquitous in real applications. In order to cater for complex-valued signals, especially for noncircular signals, in this paper we extend OS-ELM to complex domain, and propose two augmented online sequential complex ELM models by incorporating the conjugates of the network input and the hidden layer respectively. In this way, the proposed models are equipped with the capability to capture the complete second-order statistics of noncircular signals, which results in the enhanced generalization ability. The corresponding regularized models are derived to avoid the possible overfitting problem. By exploiting the algebra structure resulted from the augmented architecture, several approaches to reducing the computational complexity are also proposed. Simulation results validate the efficiency of the proposed algorithms.

INDEX TERMS Complex extreme learning machine (CELM), online sequential learning algorithms, noncircular signals, augmented complex statistics, computational complexity.


I. INTRODUCTION

Neural networks have become powerful tools which imitate the human brain's hierarchical architecture to learn complicated representation, and have been successfully applied to various fields [1]. Neural networks can usually be trained in two modes: batch learning and online learning [2], [3]. The batch learning approach accumulates the weight correction over all training samples before actually performing the update, nevertheless the online learning approach updates the network weights immediately after a block of training samples are fed. Gradient learning used to be a popular training method. However, it suffers from slow convergence and easily gets trapped in local minima. Thus, gradient-free methods are desired for fast training.

Extreme learning machine is a promising learning strategy for single-hidden layer neural networks, which have been proved to be universal approximators [4]. By randomly choosing hidden nodes and analytically determining the output weights with least-squares solution, ELM retains

the merits such as fast learning and good generalization ability [5]. ELM was first proposed by Prof. Huang, and then extended to online sequential learning [6]. In order to cater for various application scenarios, many variants of ELM have been proposed, such as convex ELM [7], ordinal ELM [8], semi-supervised ELM [9], Bayesian ELM [10], voting based ELM [11], PCA-ELM [12], robust ELM [13], microgenetic ELM [14], fuzzy ELM [15], hierarchical ELM [16], kernel ELM [17], ELM for multilayer perceptron [18], stacked ELM [19], wavelet ELM [20], graph embedded ELM [21], ELM for interval neural networks [22], memetic ELM [23], and ELM with binary output layer [24], ELM for imbalance learning [25], and ELM for residual learning [26]. The readers can refer to [27]–[29] for a thorough review of the essence and trends of ELM. Recently online sequential ELM has been applied to various fields such as fault diagnosis [30] and air quality forecasting [31].

In applications such as array signal processing, radar and magnetic resonance data imaging, communication systems, and interval data processing, signals are usually represented as complex numbers; this motivates the extension of neural networks from real domain to complex domain. Two main

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif .

superiorities of complex-valued neural networks (CVNNs) have been identified: (i) The XOR problem and the detection of symmetry problem that cannot be solved with two-layered real-valued neural networks, can be solved by two-layered complex-valued neural networks due to the orthogonal decision boundaries [32]; (ii) CVNN exhibits smaller generalization error in particular for signals having high coherence due to the enhanced capability in capturing the correlation between channels [33]. Other advantages of CVNNs have also been reported in models such as complex-valued recurrent neural networks [34] and complex independent component analysis [35].

As a gradient-free CVNN model, complex ELM (CELM) [36] naturally avoids the restriction caused by Liouville’s theorem on the choice of activation functions, and outperforms several other CVNN models. CELM has been proved to be a universal approximator in complex domain through the approximation theory of the incremental CELM [37]. By performing a one-to-one transformation of real-valued features to the complex plane, CELM also exhibited better performance for real-valued classification problems than traditional real-valued neural models, especially when data sets are highly unbalanced [38].

From the point of view of statistics, the essence of the learning of neural networks is to extract the statistical features from the input signals and encode them with network weights [39]. As revealed by the statistical theory, complex-valued random signals can be categorized into two types according to the second-order statistics: proper signals and noncircular signals [40]–[42]. Probability distributions of proper signals are rotation invariant, and then the second-order statistics can be characterized by the covariance matrix, which coincides with the case of real-valued signals. However, in real applications the signals usually exhibit noncircularity, for which both the covariance and the pseudo-covariance matrices should be collaboratively considered. The above so called augmented statistics for noncircular signals have been introduced into several important learning models, such as the augmented complex least means square algorithms [43]–[46], augmented echo state network [47], augmented algorithm for fully complex-valued neural networks [48], and augmented CELM [49]. However, it is still unknown how the augmented statistical theory benefits the online sequential learning of CELM for noncircular signals.

In this paper, motivated by the augmented statistical theory, we aim to establish a framework for online sequential learning of CELM to deal with noncircular signals. The key contributions of our work are as follows:

(i) In order to capture the full second-order statistics of the input signals and the hidden layer signals, we propose two augmented online sequential CELM (OS-CELM) models by incorporating the conjugates of the network input and hidden layer, respectively. In this way, the improved performance can be observed compared with the original OS-CELM when dealing with noncircular signals.

(ii) In order to avoid the overfitting problem, the regularized algorithms for the proposed models are mathematically derived.

(iii) By considering the algebraic structure resulted from the augmented networks, we propose several computational tricks to reduce the computational complexity.

The remainder of this paper is organized as follows. We give a brief review of CELM and OS-CELM in the next section. In Section III, we propose two augmented OS-CELM models, namely, OS-CELMAI and OS-CELMAH. The corresponding regularized algorithms are derived in Section IV. In Section V, we discuss how to use the algebraic structure to reduce the computational complexity. The superiority of the proposed models is illustrated with simulation results in Section VI. Finally, this paper is concluded in Section VII.

The following notations are adopted: Bold-faced quantities with uppercase and lowercase letters denote matrices and vectors respectively, \mathbb{R} the set of real numbers, \mathbb{C} the set of complex numbers, $\| \cdot \|$ the Euclidean norm of a vector, $j = \sqrt{-1}$, $(\cdot)^*$ the complex conjugate, $(\cdot)^T$ the transpose of a vector or a matrix, $(\cdot)^{-1}$ the matrix inversion, $(\cdot)^H$ the Hermitian transpose of a vector or a matrix, $\Re(\cdot)$ the real part of a complex number or a vector, and $\Im(\cdot)$ the imaginary part of a complex number or a vector.

II. COMPLEX-VALUED ELM MODEL

In this section, we first give a brief review of the complex-valued extreme learning machine (CELM), then we extend the online sequential ELM from real domain to complex domain to establish the online sequential CELM (OS-CELM) model, which serves as a basis of the two augmented models proposed in the next section. CELM and OS-CELM basically inherit the network structures and the learning procedures of their real-valued counterparts, while the network parameters and the activation functions are all complex-valued, and the algebraic manipulations are performed in the complex domain.

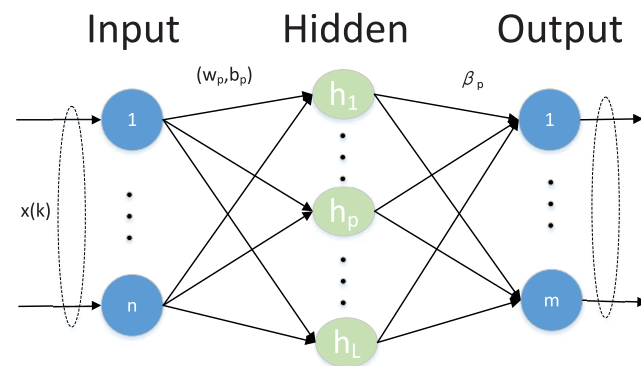


FIGURE 1. Architecture of CELM.

A. CELM

CELM is a training strategy for complex-valued feedforward neural networks with single hidden layer (CFNNsL), whose architecture is illustrated in Fig. 1.

Suppose that a CFNNLSL consists of n input nodes, L hidden nodes, and m output nodes. Write $\mathbf{w}_p = [w_{1p}, w_{2p}, \dots, w_{np}]^T \in \mathbb{C}^{n \times 1}$ as the input weight vector connecting the p th hidden node and the input nodes, and $b_p \in \mathbb{C}$ as the threshold (bias) of the p th hidden node. For an input vector $\mathbf{x} \in \mathbb{C}^{1 \times n}$, the output of the p th hidden node is $h(\mathbf{x}\mathbf{w}_p + b_p)$, where $h(\cdot)$ is a fully complex-valued activation function.

Suppose that complex-valued training samples $\aleph = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=1}^N$ are provided, where $\mathbf{x}(k) \in \mathbb{C}^{1 \times n}$ is a network input vector and $\mathbf{t}(k) \in \mathbb{C}^{1 \times m}$ is the corresponding target output vector, for $k = 1, \dots, N$. Then the functionality of a CELM can be mathematically modelled by

$$\sum_{p=1}^L h(\mathbf{x}(k)\mathbf{w}_p + b_p)\boldsymbol{\beta}_p = \mathbf{t}(k), \quad k = 1, \dots, N, \quad (1)$$

where $\boldsymbol{\beta}_p = [\beta_{1p}, \beta_{2p}, \dots, \beta_{mp}] \in \mathbb{C}^{1 \times m}$ is the output weight vector connecting the p th hidden node and the output nodes.

Formula (1) can be reformulated in a compact form

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \quad (2)$$

where the hidden layer input matrix $\mathbf{H} = h(\mathbf{Y})$,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{x}(1)\mathbf{w}_1 + b_1 & \cdots & \mathbf{x}(1)\mathbf{w}_L + b_L \\ \vdots & \cdots & \vdots \\ \mathbf{x}(N)\mathbf{w}_1 + b_1 & \cdots & \mathbf{x}(N)\mathbf{w}_L + b_L \end{bmatrix} = \mathbf{X}\mathbf{W} \oplus \mathbf{b}, \quad (3)$$

$\mathbf{X} = [\mathbf{x}(1)^T, \dots, \mathbf{x}(N)^T]^T$, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_L]$, $\mathbf{b} = [b_1, \dots, b_L]$,

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1 \\ \vdots \\ \boldsymbol{\beta}_L \end{bmatrix}_{L \times m}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}(1) \\ \vdots \\ \mathbf{t}(N) \end{bmatrix}_{N \times m}, \quad (4)$$

and the notation \oplus indicates that the vector \mathbf{b} is added elementwise to each row of the matrix $\mathbf{X}\mathbf{W}$.

The input weight vectors \mathbf{w}_p and the hidden layer bias $b_p (p = 1, \dots, L)$ of CELM are randomly generated and need not to be tuned. Thus we only need to determine the output weight matrix $\boldsymbol{\beta}$ to satisfy (2). By solving the least-squares problem $\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2$, we obtain an explicit solution

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{T}, \quad (5)$$

where $\mathbf{H}^\dagger = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H$ is the Moore-Penrose generalized inverse of the matrix \mathbf{H} .

B. ONLINE SEQUENTIAL CELM (OS-CELM)

The CELM reviewed above is really a batch learning model, as it needs the training samples to be available beforehand. However, in many real-world applications, the training data may arrive one-by-one or chunk-by-chunk. Thus CELM should be extended to online sequential learning mode to cater for these situations.

Suppose that an initial complex-valued training data set $\aleph_0 = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=1}^{N_0}$ is provided, where $N_0 \geq L$. By solving the least-squares problem $\min_{\boldsymbol{\beta}} \|\mathbf{H}_0 \boldsymbol{\beta} - \mathbf{T}_0\|^2$, we obtain

an initial estimate of the output weight matrix $\boldsymbol{\beta}^{(0)} = \mathbf{K}_0^{-1} \mathbf{H}_0^H \mathbf{T}_0$, where $\mathbf{K}_0 = \mathbf{H}_0^H \mathbf{H}_0$, $\mathbf{H}_0 = h(\mathbf{Y}_0)$,

$$\mathbf{Y}_0 = \begin{bmatrix} \mathbf{x}(1)\mathbf{w}_1 + b_1 & \cdots & \mathbf{x}(1)\mathbf{w}_L + b_L \\ \vdots & \cdots & \vdots \\ \mathbf{x}(N_0)\mathbf{w}_1 + b_1 & \cdots & \mathbf{x}(N_0)\mathbf{w}_L + b_L \end{bmatrix}_{N_0 \times L} = \mathbf{X}_0 \mathbf{W} \oplus \mathbf{b}, \quad (6)$$

$\mathbf{X}_0 = [\mathbf{x}(1)^T, \dots, \mathbf{x}(N_0)^T]^T$, and

$$\mathbf{T}_0 = \begin{bmatrix} \mathbf{t}(1) \\ \vdots \\ \mathbf{t}(N_0) \end{bmatrix}_{N_0 \times m}. \quad (7)$$

Suppose that another chunk of data $\aleph_1 = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=N_0+1}^{N_0+N_1}$ are provided, where N_1 denotes the number of observations in this chunk. We try to solve the least-squares problem of minimizing

$$\left\| \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} \boldsymbol{\beta} - \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \right\|^2, \quad (8)$$

where

$$\mathbf{H}_1(\mathbf{w}_1, \dots, \mathbf{w}_L, \mathbf{x}(N_0+1), \dots, \mathbf{x}(N_0+N_1), b_1, \dots, b_L) = \begin{bmatrix} h(\mathbf{x}(N_0+1)\mathbf{w}_1 + b_1) & \cdots & h(\mathbf{x}(N_0+1)\mathbf{w}_L + b_L) \\ \vdots & \cdots & \vdots \\ h(\mathbf{x}(N_0+N_1)\mathbf{w}_1 + b_1) & \cdots & h(\mathbf{x}(N_0+N_1)\mathbf{w}_L + b_L) \end{bmatrix} = h(\mathbf{X}_1 \mathbf{W} \oplus \mathbf{b}), \quad (9)$$

$\mathbf{X}_1 = [\mathbf{x}(N_0+1)^T, \dots, \mathbf{x}(N_0+N_1)^T]^T$, and

$$\mathbf{T}_1 = \begin{bmatrix} \mathbf{t}(N_0+1) \\ \vdots \\ \mathbf{t}(N_0+N_1) \end{bmatrix}_{N_1 \times m}. \quad (10)$$

Noticing that this problem depends on the training sets \aleph_0 and \aleph_1 , the solution can be obtained as

$$\boldsymbol{\beta}^{(1)} = \mathbf{K}_1^{-1} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix}, \quad (11)$$

where

$$\mathbf{K}_1 = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}. \quad (12)$$

Instead of computing $\boldsymbol{\beta}^{(1)}$ from scratch according to (11), we try to obtain it based on $\boldsymbol{\beta}^{(0)}$. By (11) and (12), we have

$$\begin{aligned} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} &= \mathbf{H}_0^H \mathbf{T}_0 + \mathbf{H}_1^H \mathbf{T}_1 \\ &= \mathbf{K}_0 \mathbf{K}_0^{-1} \mathbf{H}_0^H \mathbf{T}_0 + \mathbf{H}_1^H \mathbf{T}_1 \\ &= \mathbf{K}_0 \boldsymbol{\beta}^{(0)} + \mathbf{H}_1^H \mathbf{T}_1 \\ &= (\mathbf{K}_1 - \mathbf{H}_1^H \mathbf{H}_1) \boldsymbol{\beta}^{(0)} + \mathbf{H}_1^H \mathbf{T}_1 \\ &= \mathbf{K}_1 \boldsymbol{\beta}^{(0)} - \mathbf{H}_1^H \mathbf{H}_1 \boldsymbol{\beta}^{(0)} + \mathbf{H}_1^H \mathbf{T}_1 \end{aligned} \quad (13)$$

and

$$\mathbf{K}_1 = [\mathbf{H}_0^H, \mathbf{H}_1^H] \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} = \mathbf{K}_0 + \mathbf{H}_1^H \mathbf{H}_1. \quad (14)$$

Combing (11), (13), and (14), we arrive at

$$\begin{aligned} \boldsymbol{\beta}^{(1)} &= \mathbf{K}_1^{-1} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix} \\ &= \mathbf{K}_1^{-1} (\mathbf{K}_1 \boldsymbol{\beta}^{(0)} - \mathbf{H}_1^H \mathbf{H}_1 \boldsymbol{\beta}^{(0)} + \mathbf{H}_1^H \mathbf{T}_1) \\ &= \boldsymbol{\beta}^{(0)} + \mathbf{K}_1^{-1} \mathbf{H}_1^H (\mathbf{T}_1 - \mathbf{H}_1 \boldsymbol{\beta}^{(0)}). \end{aligned} \quad (15)$$

In this way, when the $(l+1)$ th chunk of data set

$$\mathfrak{S}_{l+1} = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=(\sum_{j=0}^l N_j)+1}^{\sum_{j=0}^{l+1} N_j}$$

is available, where $l \geq 0$ and N_{l+1} denotes the number of training samples of this chunk, we have

$$\begin{aligned} \boldsymbol{\beta}^{(l+1)} &= \boldsymbol{\beta}^{(l)} + \mathbf{K}_{l+1}^{-1} \mathbf{H}_{l+1}^H (\mathbf{T}_{l+1} - \mathbf{H}_{l+1} \boldsymbol{\beta}^{(l)}), \\ \mathbf{K}_{l+1} &= \mathbf{K}_l + \mathbf{H}_{l+1}^H \mathbf{H}_{l+1}. \end{aligned} \quad (16)$$

We apply the Woodbury identity formula to avoid directly calculating \mathbf{K}_{l+1}^{-1} and obtain that

$$\begin{aligned} \mathbf{K}_{l+1}^{-1} &= (\mathbf{K}_l + \mathbf{H}_{l+1}^H \mathbf{H}_{l+1})^{-1} \\ &= \mathbf{K}_l^{-1} - \mathbf{K}_l^{-1} \mathbf{H}_{l+1}^H (\mathbf{I} + \mathbf{H}_{l+1} \mathbf{K}_l^{-1} \mathbf{H}_{l+1}^H)^{-1} \mathbf{H}_{l+1} \mathbf{K}_l^{-1}. \end{aligned} \quad (17)$$

Based on (17), \mathbf{K}_{l+1}^{-1} can be obtained recursively from \mathbf{K}_0^{-1} .

Let $\mathbf{P}_{l+1} = \mathbf{K}_{l+1}^{-1}$, then the update formulae for $\boldsymbol{\beta}^{(l+1)}$ can be rewritten as

$$\begin{aligned} \mathbf{P}_{l+1} &= \mathbf{P}_l - \mathbf{P}_l \mathbf{H}_{l+1}^H (\mathbf{I} + \mathbf{H}_{l+1} \mathbf{P}_l \mathbf{H}_{l+1}^H)^{-1} \mathbf{H}_{l+1} \mathbf{P}_l, \\ \boldsymbol{\beta}^{(l+1)} &= \boldsymbol{\beta}^{(l)} + \mathbf{P}_{l+1} \mathbf{H}_{l+1}^H (\mathbf{T}_{l+1} - \mathbf{H}_{l+1} \boldsymbol{\beta}^{(l)}). \end{aligned} \quad (18)$$

Remark 1: It used to be a difficult problem to choose the activation functions of CVNNs due to the conflict between the boundedness and analyticity of a complex-valued function. Benefited from the gradient-free nature, such a conflict does not remain a restriction for CELM. Thus, the popular activation functions for RVNNs could work well for CELM when they are extended to the complex domain.

Remark 2: Online sequential learning may face the concept drift problem when the distribution of the data steaming is non-stationary. By considering the forgetting factor into the algorithm design, this problem can be avoided for OS-ELM. For more details, readers can refer to [50].

III. AUGMENTED OS-CELM

A. SECOND-ORDER STATISTICS OF COMPLEX SIGNALS

The recently introduced augmented complex statistics [41], [45] revealed that second-order statistical properties of complex signals are characterized by both their covariance matrix $\mathcal{C}_{\mathbf{xx}} = E[\mathbf{xx}^H]$ and pseudocovariance matrix $\mathcal{P}_{\mathbf{xx}} = E[\mathbf{xx}^T]$. The covariance conveys the information concerning the total power of the signal, while the pseudocovariance matrix captures the information about the power difference and cross-correlation between the real and imaginary parts of

the signal. Processes with the vanishing pseudo-covariance (i.e. $\mathcal{P}_{\mathbf{xx}} = 0$) are termed second-order circular (or proper). However, in most real-world applications, complex signals are second-order noncircular or improper. Thus the traditional algorithms for complex-valued signal processing which are only based on covariance matrix may result in suboptimal solutions for noncircular signal scenarios.

The so-called augmented model is designed to cater for second-order noncircular signals, whereby an augmented signal \mathbf{x}^a is produced by concatenating the original signal \mathbf{x} with its conjugate \mathbf{x}^* , that is

$$\mathbf{x}^a = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}. \quad (19)$$

The corresponding augmented covariance matrix then becomes

$$\begin{aligned} \mathcal{C}_{\mathbf{x}^a \mathbf{x}^a} &= E[\mathbf{x}^a (\mathbf{x}^a)^H] = E \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix} \begin{bmatrix} \mathbf{x}^H & \mathbf{x}^T \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{C}_{\mathbf{xx}} & \mathcal{P}_{\mathbf{xx}} \\ \mathcal{P}_{\mathbf{xx}}^* & \mathcal{C}_{\mathbf{xx}}^* \end{bmatrix}, \end{aligned} \quad (20)$$

which contains the full second-order statistical information [41] available in the complex domain.

B. OS-CELM WITH AUGMENTED INPUT (OS-CELMAI)

As shown by the aforementioned statistical theory of complex signals, OS-CELM is insufficient to capture the complete second-order statistics of training data as it does not incorporate the conjugate information of signals during the learning process. That is to say, OS-CELM may lead to suboptimal solutions for noncircular signal processing problems. To fill this gap, in the following of this subsection, we propose an OS-CELM model with augmented input (OS-CELM).

The topological graph of OS-CELMAI is shown in Fig.2. In order to capture the complete second-order statistics of training data, we augment the original network input $\mathbf{x}(k)$ with

$$\mathbf{x}^a(k) = [\mathbf{x}(k), \mathbf{x}^*(k)] \in \mathbb{C}^{1 \times 2n}. \quad (21)$$

For an initial training data set $\mathfrak{S}_0 = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=1}^{N_0}$, we solve the least-squares problem $\min_{\boldsymbol{\beta}} \|\mathbf{H}_0^{ai} \boldsymbol{\beta} - \mathbf{T}_0\|^2$, and obtain

an initial estimate of the output weight matrix $(\boldsymbol{\beta}^{ai})^{(0)} = (\mathbf{K}_0^{ai})^{-1} (\mathbf{H}_0^{ai})^H \mathbf{T}_0$, where $\mathbf{K}_0^{ai} = (\mathbf{H}_0^{ai})^H (\mathbf{H}_0^{ai})$, $\mathbf{H}_0^{ai} = h(\mathbf{Y}_0^{ai})$,

$$\begin{aligned} \mathbf{Y}_0^{ai} &= \begin{bmatrix} \mathbf{x}^a(1) \mathbf{w}_1^{ai} + b_1 & \cdots & \mathbf{x}^a(1) \mathbf{w}_L^{ai} + b_L \\ \vdots & \cdots & \vdots \\ \mathbf{x}^a(N_0) \mathbf{w}_1^{ai} + b_1 & \cdots & \mathbf{x}^a(N_0) \mathbf{w}_L^{ai} + b_L \end{bmatrix} \\ &= \mathbf{X}_0^a \mathbf{W}^{ai} \oplus \mathbf{b}, \end{aligned} \quad (22)$$

$\mathbf{X}_0^a = [\mathbf{X}_0, \mathbf{X}_0^*]$, and $\mathbf{W}^{ai} = [\mathbf{W}_1^T, \mathbf{W}_2^T]^T$. Here \mathbf{W}_1 is the weight matrix connecting the original input nodes and the hidden layer, and \mathbf{W}_2 is the weight matrices connecting other

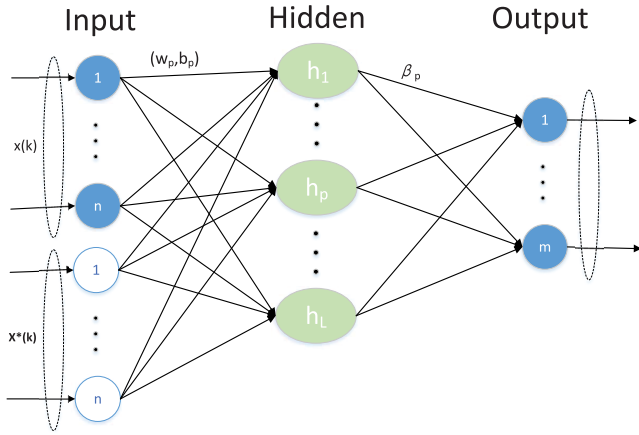


FIGURE 2. Architecture of OS-CELMAI.

input nodes of the augmented input layer and the hidden layer, respectively.

When $(l + 1)$ th chunk of data

$$\mathfrak{N}_{l+1} = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=(\sum_{j=0}^l N_j)+1}^{\sum_{j=0}^{l+1} N_j}$$

are received, the output weight matrix can be updated as follows

$$\mathbf{P}_{l+1}^{ai} = \mathbf{P}_l^{ai} - \mathbf{P}_l^{ai} (\mathbf{H}_{l+1}^{ai})^H (\mathbf{I} + \mathbf{H}_{l+1}^{ai} \times \mathbf{P}_l^{ai} (\mathbf{H}_{l+1}^{ai})^H)^{-1} \mathbf{H}_{l+1}^{ai} \mathbf{P}_l^{ai}, \quad (23)$$

$$(\boldsymbol{\beta}^{ai})^{(l+1)} = (\boldsymbol{\beta}^{ai})^{(l)} + \mathbf{P}_{l+1}^{ai} (\mathbf{H}_{l+1}^{ai})^H (\mathbf{T}_{l+1} - \mathbf{H}_{l+1}^{ai} (\boldsymbol{\beta}^{ai})^{(l)}), \quad (24)$$

where $\mathbf{P}_{l+1}^{ai} = (\mathbf{K}_{l+1}^{ai})^{-1}$.

Remark 3: By incorporating the conjugate information of the network input, OS-CELMAI is equipped with the ability to capture the complete second-order statistics of training samples. However, when the number of input features of the sample is large enough, the augmented input may lead to information redundancy. This problem can usually be tackled by applying feature reduction techniques such as PCA to perform data preprocessing.

C. OS-CELM WITH AUGMENTED HIDDEN LAYER (OS-CELMAH)

The topological graph of OS-CELMAH is shown in Fig. 3. Different from OS-CELMAI aforementioned, OS-CELMAH incorporates the conjugate of the original hidden nodes of OS-CELM to form a new augmented hidden layer. This augmented structure contributes to capturing the complete second-order statistics of the hidden layer, which is a transformation of the network input signal.

Given an initial training data set $\mathfrak{N}_0 = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=1}^{N_0}$, resulted from the augmented structure, the hidden layer output matrix of the OS-CELMAH appears to be

$$\mathbf{H}_0^{ah} = [\mathbf{H}_0, \mathbf{H}_0^*]_{N_0 \times 2L}, \quad (25)$$

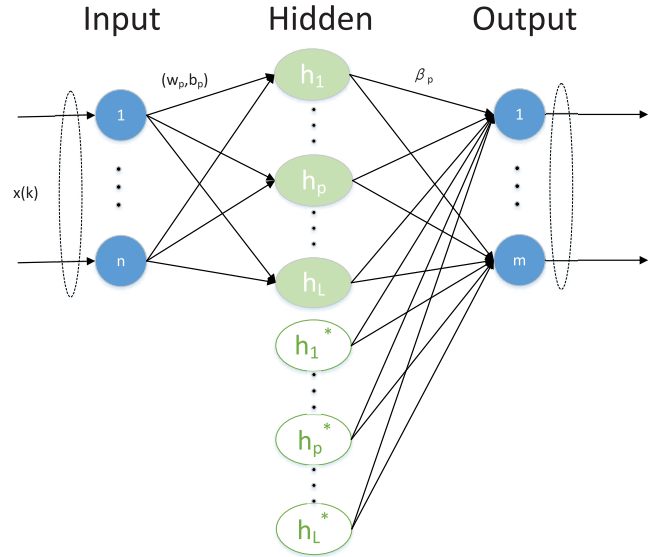


FIGURE 3. Architecture of OS-CELMAH.

where \mathbf{H}_0 is defined in (6). Then the output weight matrix can be initially estimated by solving the least-squares problem $\min_{\boldsymbol{\beta}} \|\mathbf{H}_0^{ah} \boldsymbol{\beta} - \mathbf{T}_0\|^2$ with the solution

$$(\boldsymbol{\beta}^{ah})^{(0)} = \mathbf{P}_0^{ah} (\mathbf{H}_0^{ah})^H \mathbf{T}_0, \quad (26)$$

where $\mathbf{P}_0^{ah} = ((\mathbf{H}_0^{ah})^H \mathbf{H}_0^{ah})^{-1}$.

When the $(l + 1)$ th chunk of data

$$\mathfrak{N}_{l+1} = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=(\sum_{j=0}^l N_j)+1}^{\sum_{j=0}^{l+1} N_j}$$

are received, the output weight matrix can be updated as follows

$$\mathbf{P}_{l+1}^{ah} = \mathbf{P}_l^{ah} - \mathbf{P}_l^{ah} (\mathbf{H}_{l+1}^{ah})^H (\mathbf{I} + \mathbf{H}_{l+1}^{ah} \times \mathbf{P}_l^{ah} (\mathbf{H}_{l+1}^{ah})^H)^{-1} \mathbf{H}_{l+1}^{ah} \mathbf{P}_l^{ah}, \quad (27)$$

$$(\boldsymbol{\beta}^{ah})^{(l+1)} = (\boldsymbol{\beta}^{ah})^{(l)} + \mathbf{P}_{l+1}^{ah} (\mathbf{H}_{l+1}^{ah})^H (\mathbf{T}_{l+1} - \mathbf{H}_{l+1}^{ah} (\boldsymbol{\beta}^{ah})^{(l)}). \quad (28)$$

Remark 4: It can be observed from Equation (27) that the complexity of computing $(\mathbf{I} + \mathbf{H}_{l+1}^{ah} \mathbf{P}_l^{ah} (\mathbf{H}_{l+1}^{ah})^H)^{-1}$ mainly depends on N_l , but not L . Thus this augmented structure does not increase the complexity for computing this inverse matrix.

IV. REGULARIZED MODELS

In this section, we derive the regularized OS-CELM models based on Wirtinger calculus to avoid the possible overfitting problem, which may greatly degenerate the generalization performance of the network in scenarios such as overtraining and noisy data. The regularized OS-CELMAI and regularized OS-CELMAH are also provided.

A. REGULARIZED OS-CELM

Suppose that an initial complex-valued training data set $\mathfrak{N}_0 = \{(\mathbf{x}(k), \mathbf{t}(k))\}_{k=1}^{N_0}$ is provided, where $N_0 \geq L$. Instead of solely

minimizing $\|\mathbf{H}_0\boldsymbol{\beta}^{(0)} - \mathbf{T}_0\|$, we try to find $\boldsymbol{\beta}^{(0)}$ to minimize the following new cost function

$$\begin{aligned} \mathbf{J} &= \frac{1}{2}\|\mathbf{H}_0\boldsymbol{\beta}^{(0)} - \mathbf{T}_0\|^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}^{(0)}\|^2 \\ &= \frac{1}{2}(\mathbf{H}_0\boldsymbol{\beta}^{(0)} - \mathbf{T}_0)^H(\mathbf{H}_0\boldsymbol{\beta}^{(0)} - \mathbf{T}_0) + \frac{\lambda}{2}(\boldsymbol{\beta}^{(0)})^H\boldsymbol{\beta}^{(0)}, \end{aligned} \quad (29)$$

where $\lambda > 0$ is the regularization parameter to balance the tradeoff between the training error and the output weight norm. According to Wirtinger calculus theory [41], [51]–[53], the solution of this optimization problem can be obtained by solving

$$\frac{\partial \mathbf{J}}{\partial (\boldsymbol{\beta}^{(0)})^*} = 0. \quad (30)$$

Since

$$\frac{\partial \mathbf{J}}{\partial (\boldsymbol{\beta}^{(0)})^*} = \frac{1}{2} \left[((\mathbf{H}_0)^H\mathbf{H}_0\boldsymbol{\beta}^{(0)}) - (\mathbf{H}_0)^H\mathbf{T}_0 \right] + \frac{\lambda}{2}\boldsymbol{\beta}^{(0)}, \quad (31)$$

we obtain the optimal solution

$$\boldsymbol{\beta}^{(0)} = ((\mathbf{H}_0)^H\mathbf{H}_0 + \lambda\mathbf{I})^{-1}(\mathbf{H}_0)^H\mathbf{T}_0. \quad (32)$$

Similarly, when a new chunk of data are received, the optimal estimation of the output weight matrix is

$$\boldsymbol{\beta}^{(1)} = \left(\begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} + \lambda\mathbf{I} \right)^{-1} \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{T}_0 \\ \mathbf{T}_1 \end{bmatrix}. \quad (33)$$

Denote $\mathbf{K}_0 = (\mathbf{H}_0)^H\mathbf{H}_0 + \lambda\mathbf{I}$ and

$$\mathbf{K}_1 = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix}^H \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{bmatrix} + \lambda\mathbf{I},$$

then we have $\mathbf{K}_1 = \mathbf{K}_0 + \mathbf{H}_1^H\mathbf{H}_1$. This equation reveals that the relationship between \mathbf{K}_1 and \mathbf{K}_0 here is identical to Equation (14), and the parameter λ in (33) can be absorbed into \mathbf{K}_0 . Thus, in the sequel the output weight matrix can be recursively updated in the form identical to OS-CELM.

The outline for the regularized OS-CELM can be briefed as follows. Let $\mathbf{P}_0 = ((\mathbf{H}_0)^H\mathbf{H}_0 + \lambda\mathbf{I})^{-1}$. Based on an initial estimation $\boldsymbol{\beta}^{(0)} = \mathbf{P}_0\mathbf{H}_0^H\mathbf{T}_0$, the equations for updating $\boldsymbol{\beta}^{(l+1)}$ can be written as

$$\begin{aligned} \mathbf{P}_{l+1} &= \mathbf{P}_l - \mathbf{P}_l\mathbf{H}_{l+1}^H(\mathbf{I} + \mathbf{H}_{l+1}\mathbf{P}_l\mathbf{H}_{l+1}^H)^{-1}\mathbf{H}_{l+1}\mathbf{P}_l, \\ \boldsymbol{\beta}^{(l+1)} &= \boldsymbol{\beta}^{(l)} + \mathbf{P}_{l+1}\mathbf{H}_{l+1}^H(\mathbf{T}_{l+1} - \mathbf{H}_{l+1}\boldsymbol{\beta}^{(l)}). \end{aligned} \quad (34)$$

B. REGULARIZED OS-CELMAI

Let $\mathbf{P}_0^{ai} = ((\mathbf{H}_0^{ai})^H\mathbf{H}_0^{ai} + \lambda\mathbf{I})^{-1}$. Then the initial estimation for the output weight matrix is given by $(\boldsymbol{\beta}^{ai})^{(0)} = \mathbf{P}_0^{ai}(\mathbf{H}_0^{ai})^H\mathbf{T}_0$, and the recursive estimation for $(\boldsymbol{\beta}^{ai})^{(l+1)}$ can be given by

$$\begin{aligned} \mathbf{P}_{l+1}^{ai} &= \mathbf{P}_l^{ai} - \mathbf{P}_l^{ai}(\mathbf{H}_{l+1}^{ai})^H(\mathbf{I} + \mathbf{H}_{l+1}^{ai}\mathbf{P}_l^{ai}(\mathbf{H}_{l+1}^{ai})^H)^{-1} \\ &\quad \times \mathbf{H}_{l+1}^{ai}\mathbf{P}_l^{ai}, \\ (\boldsymbol{\beta}^{ai})^{(l+1)} &= (\boldsymbol{\beta}^{ai})^{(l)} + \mathbf{P}_{l+1}^{ai}(\mathbf{H}_{l+1}^{ai})^H(\mathbf{T}_{l+1} \\ &\quad - \mathbf{H}_{l+1}^{ai}(\boldsymbol{\beta}^{ai})^{(l)}). \end{aligned} \quad (35)$$

C. REGULARIZED OS-CELMAH

Let $\mathbf{P}_0^{ah} = ((\mathbf{H}_0^{ah})^H\mathbf{H}_0^{ah} + \lambda\mathbf{I})^{-1}$. Then the initial estimation for the output weight matrix is given by $(\boldsymbol{\beta}^{ah})^{(0)} = \mathbf{P}_0^{ah}(\mathbf{H}_0^{ah})^H\mathbf{T}_0$, and the recursive estimation for $(\boldsymbol{\beta}^{ah})^{(l+1)}$ can be given by

$$\begin{aligned} \mathbf{P}_{l+1}^{ah} &= \mathbf{P}_l^{ah} - \mathbf{P}_l^{ah}(\mathbf{H}_{l+1}^{ah})^H(\mathbf{I} + \mathbf{H}_{l+1}^{ah}\mathbf{P}_l^{ah}(\mathbf{H}_{l+1}^{ah})^H)^{-1} \\ &\quad \times \mathbf{H}_{l+1}^{ah}\mathbf{P}_l^{ah}, \\ (\boldsymbol{\beta}^{ah})^{(l+1)} &= (\boldsymbol{\beta}^{ah})^{(l)} + \mathbf{P}_{l+1}^{ah}(\mathbf{H}_{l+1}^{ah})^H(\mathbf{T}_{l+1} - \mathbf{H}_{l+1}^{ah} \\ &\quad \times (\boldsymbol{\beta}^{ah})^{(l)}). \end{aligned} \quad (36)$$

V. APPROACHES TO REDUCING THE COMPUTATIONAL COMPLEXITY

The augmented structures of OS-CELMAI and OS-CELMAH contribute to capturing the complete second-order statistics of noncircular signals. At the same time, these specified structures may be exploited in reducing the computational complexity of the proposed learning algorithms [54]. In this section, we introduce several techniques to reduce the computational complexity of the proposed learning algorithms by utilizing the algebraic structure resulted from the augmented network structures.

A. A TECHNIQUE FOR OS-CELMAI

It is well known that the real part and imaginary part of a complex vector \mathbf{x} can be expressed by

$$\Re(\mathbf{x}) = \frac{1}{2}(\mathbf{x} + \mathbf{x}^*), \quad \Im(\mathbf{x}) = \frac{1}{2j}(\mathbf{x} - \mathbf{x}^*). \quad (37)$$

We expand $\mathbf{X}_0^a\mathbf{W}^{ai}$ in Equation (22) as follows

$$\mathbf{X}_0^a\mathbf{W}^{ai} = \mathbf{X}_0\mathbf{W}_1 + \mathbf{X}_0^*\mathbf{W}_2. \quad (38)$$

Applying (37) to (38), we obtain

$$\begin{aligned} \Re[\mathbf{X}_0^a\mathbf{W}^{ai}] &= \Re[\mathbf{X}_0\mathbf{W}_r], \\ \Im[\mathbf{X}_0^a\mathbf{W}^{ai}] &= \Im[\mathbf{X}_0\mathbf{W}_i], \end{aligned} \quad (39)$$

where $\mathbf{W}_r = \mathbf{W}_1 + \mathbf{W}_2^*$, $\mathbf{W}_i = \mathbf{W}_1 - \mathbf{W}_2^*$. Combining (39) and (22), we have

$$\mathbf{Y}_0^{ai} = (\Re[\mathbf{X}_0\mathbf{W}_r] + j\Im[\mathbf{X}_0\mathbf{W}_i]) \oplus \mathbf{b}. \quad (40)$$

Similarly, for any $l > 0$, we have

$$\mathbf{Y}_l^{ai} = (\Re[\mathbf{X}_l\mathbf{W}_r] + j\Im[\mathbf{X}_l\mathbf{W}_i]) \oplus \mathbf{b}. \quad (41)$$

Remark 5: An operation of complex-complex multiplication, ab , requires 4 real multiplications and 2 real sums, whereas an operation $\Re[ab]$ (or $\Im[ab]$) only needs 2 real multiplications and 1 real sum. Thus, computing $\mathbf{X}_0^a\mathbf{W}^{ai}$ via (40) only require a half of computations compared with that via (22).

B. DUAL-CHANNEL OS-CELMAH (DC-OS-CELMAH)

In this subsection, we reformulate the update equations for the augmented output weight matrix of OS-CELMAH in order to reduce the computation complexity.

We have derived the initial estimate for the output weight matrix $(\beta^{(ah)})^{(0)} = \mathbf{P}_0^{ah}(\mathbf{H}_0^{ah})^H \mathbf{T}_0$, where $\mathbf{P}_0^{ah} = (\mathbf{K}_0^{ah})^{-1} = ((\mathbf{H}_0^{ah})^H \mathbf{H}_0^{ah})^{-1}$, and $\mathbf{H}_0^{ah} = [\mathbf{H}_0, \mathbf{H}_0^*]$. Now we represent \mathbf{K}_0^{ah} as a block matrix

$$\begin{aligned} \mathbf{K}_0^{ah} &= \begin{bmatrix} [\mathbf{H}_0 & \mathbf{H}_0^*]^H [\mathbf{H}_0 & \mathbf{H}_0^*] \\ \left(\begin{matrix} \mathbf{H}_0^H \\ (\mathbf{H}_0^*)^H \end{matrix} \right) \begin{pmatrix} \mathbf{H}_0 & \mathbf{H}_0^* \end{pmatrix} \\ \begin{bmatrix} \mathbf{H}_0^H \mathbf{H}_0 & \mathbf{H}_0^H \mathbf{H}_0^* \\ (\mathbf{H}_0^*)^H \mathbf{H}_0 & (\mathbf{H}_0^*)^H \mathbf{H}_0^* \end{bmatrix} \\ \begin{bmatrix} \mathcal{A}_0 & \mathcal{B}_0^* \\ \mathcal{B}_0 & \mathcal{A}_0^* \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{A}_0 & \mathcal{B}_0^* \\ \mathcal{B}_0 & \mathcal{A}_0^* \end{bmatrix}. \end{aligned} \quad (42)$$

Recalling $\mathbf{P}_0^{ah} = (\mathbf{K}_0^{ah})^{-1}$, \mathbf{P}_0^{ah} can also be represented as a block matrix with the same block structure

$$\mathbf{P}_0^{ah} = \begin{bmatrix} \mathcal{S}_0 & \mathcal{T}_0^* \\ \mathcal{T}_0 & \mathcal{S}_0^* \end{bmatrix}. \quad (43)$$

Combining (25), (26), and (43), we have

$$\begin{aligned} (\beta^{(ah)})^{(0)} &= \begin{bmatrix} \mathcal{S}_0 & \mathcal{T}_0^* \\ \mathcal{T}_0 & \mathcal{S}_0^* \end{bmatrix} \begin{bmatrix} \mathbf{H}_0^H \\ (\mathbf{H}_0^*)^H \end{bmatrix} \mathbf{T}_0 \\ &= \begin{bmatrix} \delta_0 \\ \delta_0^* \end{bmatrix} \mathbf{T}_0 \\ &= \begin{bmatrix} \mathbf{h}_0 \\ \mathbf{g}_0 \end{bmatrix}, \end{aligned} \quad (44)$$

where

$$\mathbf{h}_0 = \delta_0 \mathbf{T}_0, \quad \mathbf{g}_0 = \delta_0^* \mathbf{T}_0, \quad (45)$$

and $\delta_0 = \mathcal{S}_0 \mathbf{H}_0^H + \mathcal{T}_0^* (\mathbf{H}_0^*)^H$. Accordingly, \mathbf{P}_l^{ah} can also be represented as a block matrix, for $l > 0$. In this way, the recursive update formula for the augmented output weight matrix can be reformulated as

$$(\beta^{(ah)})^{(l+1)} = \begin{bmatrix} \mathbf{h}_{l+1} \\ \mathbf{g}_{l+1} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_l \\ \mathbf{g}_l \end{bmatrix} + \begin{bmatrix} \delta_{l+1} \\ \delta_{l+1}^* \end{bmatrix} e_{l+1}^{ah}, \quad (46)$$

where $\delta_{l+1} = \mathcal{S}_{l+1} \mathbf{H}_{l+1}^H + \mathcal{T}_{l+1}^* (\mathbf{H}_{l+1}^*)^H$,

$$\mathbf{h}_{l+1} = \mathbf{h}_l + \delta_{l+1} e_{l+1}^{ah}, \quad \mathbf{g}_{l+1} = \mathbf{g}_l + \delta_{l+1}^* e_{l+1}^{ah}, \quad (47)$$

and $e_{l+1}^{ah} = \mathbf{T}_{l+1} - \mathbf{H}_{l+1}^{ah} (\beta^{(ah)})^{(l)}$.

Equation (45) can be reformulated as

$$\begin{aligned} \omega_{r,0} &= \mathbf{h}_0 + \mathbf{g}_0^* = 2\delta_0 \Re[\mathbf{T}_0], \\ \omega_{i,0} &= \mathbf{h}_0 - \mathbf{g}_0^* = 2\delta_0 \Im[\mathbf{T}_0]J, \end{aligned} \quad (48)$$

and (47) can be reformulated as

$$\begin{aligned} \omega_{r,l+1} &= \mathbf{h}_{l+1} + \mathbf{g}_{l+1}^* = \omega_{r,l} + 2\delta_{l+1} \Re[e_{l+1}^{ah}], \\ \omega_{i,l+1} &= \mathbf{h}_{l+1} - \mathbf{g}_{l+1}^* = \omega_{i,l} + 2\delta_{l+1} \Im[e_{l+1}^{ah}]J, \end{aligned} \quad (49)$$

where $e_{l+1}^{ah} = \mathbf{T}_{l+1} - (\Re[\mathbf{H}_{l+1} \omega_{r,l}] + J \Im[\mathbf{H}_{l+1} \omega_{i,l}])$. Equations (48) and (49) are equivalent to (44) and (46), thus

they can also be used to adjust the output weight matrix. When a chunk of data are available, we update $\omega_{r,l}$ and $\omega_{i,l}$ separately. We name this learning strategy as dual-channel OS-CELMAH.

Remark 6: Compared with OS-CELMAH, the dual-channel OS-CELMAH could effectively save the computational cost of (28) due to the following two aspects: (i) Computing δ_{l+1} only needs one half of computations for $\mathbf{P}_0^{ah}(\mathbf{H}_0^{ah})^H$; (ii) Since $\Re[e_{l+1}^{ah}]$ and $\Im[e_{l+1}^{ah}]$ are real-valued vectors, the matrix multiplications $\delta_{l+1} \Re[e_{l+1}^{ah}]$ and $\delta_{l+1} \Im[e_{l+1}^{ah}]$ only need no more than one half computational cost of the corresponding multiplication operations $\delta_{l+1} e_{l+1}^{ah}$ and $\delta_{l+1}^* e_{l+1}^{ah}$ in (46), which is identical to (28).

C. SIMPLIFIED CALCULATION OF THE INVERSE OF COVARIANCE MATRIX

As illustrated by (42), the covariance matrix \mathbf{K}_0^{ah} enjoys a particular structure. In the following we utilize this structure to simplify the computation of its inverse \mathbf{P}_0^{ah} .

Combining (42), (43) and $\mathbf{K}_0^{ah} \mathbf{P}_0^{ah} = \mathbf{I}$, we have

$$\begin{bmatrix} \mathcal{A}_0 & \mathcal{B}_0^* \\ \mathcal{B}_0 & \mathcal{A}_0^* \end{bmatrix} \begin{bmatrix} \mathcal{S}_0 & \mathcal{T}_0^* \\ \mathcal{T}_0 & \mathcal{S}_0^* \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (50)$$

which leads to

$$\begin{aligned} \mathcal{A}_0 \mathcal{S}_0 + \mathcal{B}_0^* \mathcal{T}_0 &= \mathbf{I}, & \mathcal{A}_0 \mathcal{T}_0^* + \mathcal{B}_0^* \mathcal{S}_0^* &= \mathbf{0}, \\ \mathcal{B}_0 \mathcal{S}_0 + \mathcal{A}_0^* \mathcal{T}_0 &= \mathbf{0}, & \mathcal{B}_0 \mathcal{T}_0^* + \mathcal{A}_0^* \mathcal{S}_0^* &= \mathbf{I}. \end{aligned} \quad (51)$$

Solving the above equations, we have

$$\mathcal{S}_0 = (\mathcal{A}_0 - \mathbf{R} \mathcal{B}_0)^{-1}, \quad \mathcal{T}_0^* = -\mathcal{S}_0 \mathbf{R}, \quad (52)$$

where $\mathbf{R} = \mathcal{B}_0^* (\mathcal{A}_0^*)^{-1}$.

In this way, the computation of the inverse of a $2L \times 2L$ complex matrix has been reduced to that of two $L \times L$ complex matrices. Thus, great computational cost can be saved especially when L is considerable large.

VI. SIMULATION RESULTS

In this section, we compare the performance of the proposed augmented OS-CELM models with the original OS-CELM model in dealing with complex-valued signal processing problems. Two benchmark problems are used to evaluate the performance: (1) Noncircular chaotic Ikeda map signal and (2) Complex nonminimum-phase channel model.

The degree of noncircularity of complex-valued signals can be measured by the index [55]

$$\rho = 1 - \det(C_{\mathbf{x}^a \mathbf{x}^a}) \det^{-2}(C_{\mathbf{x}\mathbf{x}}), \quad (53)$$

where $\det(\cdot)$ denotes the matrix determinant operator, the augmented covariance matrices $C_{\mathbf{x}^a \mathbf{x}^a}$ and $C_{\mathbf{x}\mathbf{x}}$ are defined in Section III.A, and the index ρ is normalized within $[0, 1]$ with the value 0 indicating perfect circularity. According to (53), it can be calculated that the noncircularity of the two benchmark problems are 0.8936 and 0.4253 respectively, which indicates that the noncircular chaotic Ikeda map signal

is of great noncircularity, whereas the other one is of less noncircularity.

The CFNNs used in simulations are single output networks, and input weights and hidden biases are randomly generated by drawing the real parts and imaginary parts from a uniform distribution $U(-0.1, 0.1)$. As reported by [56]–[58], such a setting for random weights following a uniform distribution with small variance can ensure the stability of ELM models. As shown in Table 1, where the experiment results for a sensitivity analysis of the proposed models based on 50 independent tests are reported, the stability of ELMs with this setting can be observed. The activation function was chosen to be the complex inverse hyperbolic function $\text{arcsinh}(\cdot)$, which is nonlinear and piecewise continuous in the complex domain. The number of hidden nodes was adjusted by gradually increasing its value and the almost optimal number of the hidden nodes is determined based on cross-validation method.

The performance of each model was measured using the normalized root mean square error (NRMSE) which is defined by

$$NRMSE = \sqrt{\frac{\sum_{k=1}^N ((\mathbf{t}(k) - \mathbf{y}(k))^H (\mathbf{t}(k) - \mathbf{y}(k)))}{N}}, \quad (54)$$

where N is the number of testing samples. For each problem, the average performance curves were plotted based on 50 independent trials.

A. NONCIRCULAR CHAOTIC IKEDA MAP SIGNAL

The nonlinear and noncircular chaotic Ikeda map signal is given by [59]

$$\begin{aligned} x(k+1) &= 1 + u(x(k) \cos[t(k)] - y(k) \sin[t(k)]), \\ y(k+1) &= u(x(k) \sin[t(k)] - y(k) \cos[t(k)]), \end{aligned} \quad (55)$$

where $u = 0.9$ and $t(k) = 0.4 - 6/(1 + (x^2(k) + y^2(k)))$.

We conducted one-step-ahead prediction of the above chaotic time series. Based on (55) we obtained 2992 samples, where 1992 samples were used for training and the remaining for testing. The number of training samples in the initial stage was 992, and the chunk size in the sequential learning was fixed to 40. The CFNN for this problem consisted of 8 input units and 1 output unit. The number of the hidden nodes was initially set as 10, and then gradually increased to 600 with the increment 80.

We first conducted tests to compare the generalization performance of the OS-CELM, OS-CELMAI and OS-CELMAH. The simulation results are plotted in Fig.4. It can be observed that OS-CELMAI and OS-CELMAH outperformed OS-CELM due to the enhanced ability in capturing the second-order statistics of the complex-valued signals. However, with the increase of the number of the hidden nodes, the models tends to be suffered from the overfitting problems. We then conducted the tests on regularized models with the regularization parameter $\lambda = 2^{-15}$, and the results are plotted in Fig.5. It can be observed that the generalization

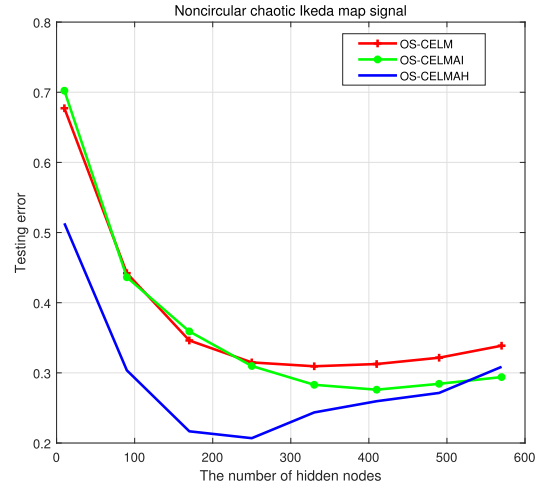


FIGURE 4. Performance comparison of the augmented OS-CELM models and the original OS-CELM for Noncircular chaotic Ikeda map signal.

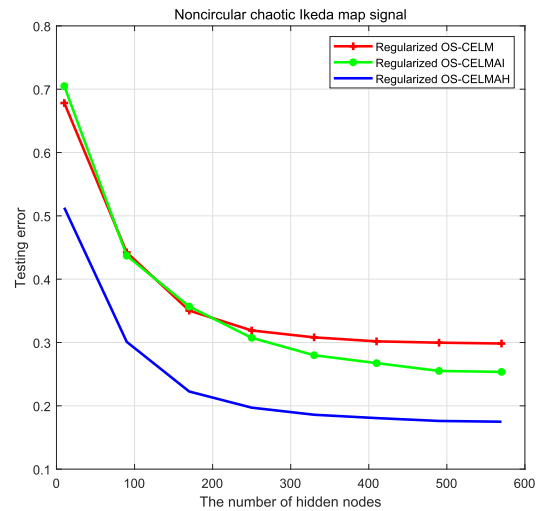


FIGURE 5. Performance comparison of the regularized models for noncircular chaotic Ikeda map signal.

performance was further improved as regularized models successfully avoided the overfitting problems, and the regularized OS-CELMAI and the regularized OS-CELMAH still outperformed the regularized original OS-CELM.

B. COMPLEX NONMINIMUM-PHASE CHANNEL MODEL

Complex nonminimum-phase channel model is of order 3 with nonlinear distortion for 4-quadrature amplitude modulation signaling. The channel output y_n is given by [60]

$$\begin{aligned} y_n &= o_n + 0.1o_n^2 + 0.05o_n^3 + v_n, v_n \sim N(0, 0.01), \\ o_n &= (0.34 - 0.27i)s_n + (0.87 + 0.43i)s_{n-1} \\ &\quad + (0.34 - 0.21i)s_{n-2}, \end{aligned} \quad (56)$$

where $N(0, 0.01)$ is the white Gaussian noise with mean 0 and variance 0.01. Given two initial values $s_0 = -0.7 - 0.7i$ and $s_{-1} = -0.7 + 0.7i$, s_n ($n = 1, 2, \dots$) were then generated by randomly choosing the real and imaginary parts from interval $[-0.7, 0.7]$.

TABLE 1. Sensitivity analysis of the proposed models.

Dataset	Hidden nodes	Testing error	OS-CELM	OS-CELMAI	OS-CELMAH
Noncircular chaotic Ikeda map signal	10	Mean	0.6782	0.7049	0.5129
		Variance	$7.8867e - 05$	0.0025	$9.5172e - 05$
	330	Mean	0.3081	0.2799	0.1859
		Variance	$1.1856e - 05$	$1.3049e - 04$	$2.5914e - 05$
	570	Mean	0.2984	0.2536	0.1749
		Variance	$1.9171e - 04$	$5.8778e - 04$	$2.3420e - 04$
Complex nonminimum-phase channel model	5	Mean	0.3976	0.6381	0.3959
		Variance	$2.1870e - 05$	0.0118	$1.7717e - 05$
	150	Mean	0.2831	0.1907	0.2643
		Variance	$4.9423e - 05$	$3.8731e - 05$	$1.4466e - 04$
	300	Mean	0.2639	0.1832	0.2523
		Variance	$2.3109e - 05$	$5.5491e - 04$	$1.3537e - 04$

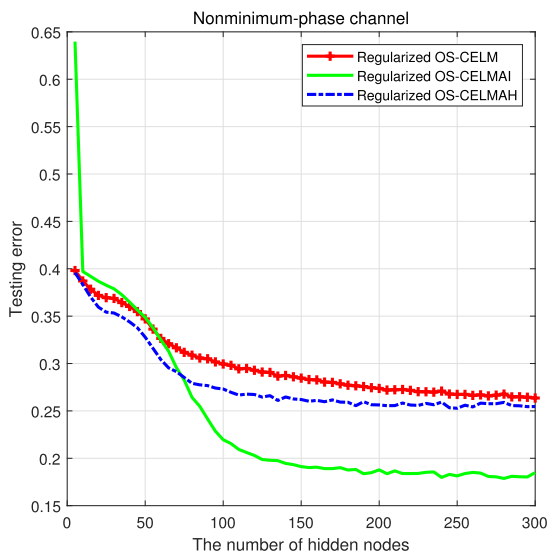


FIGURE 6. Performance comparison of the regularized models for nonminimum-phase channel.

Based on (56), we obtained 998 training samples and 998 testing samples separately. The number of training samples in the initial stage was 338, and the chunk size in the sequential learning phase was fixed to 30. The CFNNSL for this problem consisted of 3 input units and 1 output unit. The number of hidden nodes was initially set as 5, and then gradually increased to 300 with the increment 5.

The simulation results for the regularized CELM models are illustrated in Fig.6, where the regularization parameter was set as $\lambda = 2^{-30}$. It can be observed that both the regularized augmented models outperformed the regularized OS-CELM, and the regularized OS-CELMAI illustrated the best performance in this experiment. Recalling that (regularized) OS-CELMAH outperformed (regularized) OS-CELMAI for noncircular chaotic Ikeda map signal, the two augmented models can be complementary for different applications.

VII. CONCLUSION

In this paper we proposed two augmented online sequential CELM models for noncircular signal processing.

By incorporating the conjugates of the network input and the hidden layer respectively into the construction of network and the learning procedure, the augmented models can capture the complete second-order statistics of noncircular signals, which contribute to enhancing the generalization capability of learning systems. The corresponding regularized algorithms to avoid overfitting problems were derived, and several tricks to reduce the computational complexity by exploiting the algebra structure due to the augmented architecture were also provided. The simulation results illustrated the superiority of the proposed models and showed that the two augmented approaches could be complementary depending on the application scenarios.

We should mention that, as the two proposed augmented models are used to cater for noncircular signals, performance promotion may not be observed for circular signals. Additionally, as shown by simulation results, for the two augmented models, we cannot justify which one is always a better choice than the other one. Thus, to make clear how to choose the augmented models for different scenarios will be our future work. Moreover, the augmented structures introduce more computation cost. Though in this paper we have introduced several tricks to reduce the computational complexity, there is still the potential to further reduce the computation cost, which is left for our future work.

REFERENCES

- [1] F. Li, J. M. Zurada, Y. Liu, and W. Wu, "Input layer regularization of multilayer feedforward neural networks," *IEEE Access*, vol. 5, pp. 10979–10985, 2017.
- [2] H. Zhang and W. Wu, "Boundedness and convergence of online gradient method with penalty for linear output feedforward neural networks," *Neural Process. Lett.*, vol. 29, no. 3, pp. 205–212, Jun. 2009.
- [3] J. Wang, B. Zhang, Z. Sang, Y. Liu, S. Wu, and Q. Miao, "Convergence of a modified gradient-based learning algorithm with penalty for single-hidden-layer feed-forward networks," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2445–2456, Apr. 2020.
- [4] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.
- [5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.
- [6] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1411–1423, Nov. 2006.

- [7] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, nos. 16–18, pp. 3056–3062, Oct. 2007.
- [8] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, and X. Wang, "Ordinal extreme learning machine," *Neurocomputing*, vol. 74, nos. 1–3, pp. 447–456, Dec. 2010.
- [9] J. Liu, Y. Chen, M. Liu, and Z. Zhao, "SELM: Semi-supervised ELM with application in sparse calibrated location estimation," *Neurocomputing*, vol. 74, no. 16, pp. 2566–2572, Sep. 2011.
- [10] E. Soria-Olivas, J. Gomez-Sanchis, J. D. Martin, J. Vila-Frances, M. Martinez, J. R. Magdalena, and A. J. Serrano, "BELM: Bayesian extreme learning machine," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 505–509, Mar. 2011.
- [11] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Inf. Sci.*, vol. 185, no. 1, pp. 66–77, Feb. 2012.
- [12] A. Castaño, F. Fernández-Navarro, and C. Hervás-Martínez, "PCA-ELM: A robust and pruned extreme learning machine approach based on principal component analysis," *Neural Process. Lett.*, vol. 37, no. 3, pp. 377–392, Jun. 2013.
- [13] P. Horata, S. Chiewchanwattana, and K. Sunat, "Robust extreme learning machine," *Neurocomputing*, vol. 102, pp. 31–44, Feb. 2013.
- [14] D. Lahoz, B. Lacruz, and P. M. Mateo, "A multi-objective micro genetic ELM algorithm," *Neurocomputing*, vol. 111, pp. 90–103, Jul. 2013.
- [15] W. B. Zhang and H. B. Ji, "Fuzzy extreme learning machine for classification," *Electron. Lett.*, vol. 49, no. 7, pp. 448–449, 2013.
- [16] H.-G. Han, L.-D. Wang, and J.-F. Qiao, "Hierarchical extreme learning machine for feedforward neural network," *Neurocomputing*, vol. 128, pp. 128–135, Mar. 2014.
- [17] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognit. Lett.*, vol. 54, pp. 11–17, Mar. 2015.
- [18] J. Tang, C. Deng, and G. B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2016.
- [19] H. Zhou, G. B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2013–2025, Sep. 2015.
- [20] S. F. Ding, J. Zhang, X. Xu, and Y. Zhang, "A wavelet extreme learning machine," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1033–1040, 2016.
- [21] A. Iosifidis, A. Tefas, and I. Pitas, "Graph embedded extreme learning machine," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 311–324, Jan. 2016.
- [22] D. K. Yang, Z. Li, and W. Wu, "Extreme learning machine for interval neural networks," *Neural Comput. Appl.*, vol. 27, no. 1, pp. 3–8, 2016.
- [23] Y. Zhang, J. Wu, Z. Cai, P. Zhang, and L. Chen, "Memetic extreme learning machine," *Pattern Recognit.*, vol. 58, pp. 135–148, Oct. 2016.
- [24] S. B. Yang, C. Zhang, Y. Bao, J. Yang, and W. Wu, "Binary output layer of extreme learning machine for solving multi-class classification problems," *Neural Process. Lett.*, vol. 52, no. 1, pp. 153–167, 2020.
- [25] W. Xiao, J. Zhang, Y. Li, S. Zhang, and W. Yang, "Class-specific cost regulation extreme learning machine for imbalanced classification," *Neurocomputing*, vol. 261, pp. 70–82, Oct. 2017.
- [26] J. Zhang, W. Xiao, Y. Li, and S. Zhang, "Residual compensation extreme learning machine for regression," *Neurocomputing*, vol. 311, pp. 126–136, Oct. 2018.
- [27] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, "Non-iterative and fast deep learning: Multilayer extreme learning machines," *J. Franklin Inst.*, vol. 357, no. 13, pp. 8925–8955, 2020.
- [28] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.
- [29] Z. Wang, L. Sui, J. Xin, L. Qu, and Y. Yao, "A survey of distributed and parallel extreme learning machine for big data," *IEEE Access*, vol. 8, pp. 201247–201258, 2020, doi: [10.1109/ACCESS.2020.3035398](https://doi.org/10.1109/ACCESS.2020.3035398).
- [30] J. Liu, Q. Li, W. Chen, Y. Yan, and X. Wang, "A fast fault diagnosis method of the PEMFC system based on extreme learning machine and Dempster-Shafer evidence theory," *IEEE Trans. Transport. Electric.*, vol. 5, no. 1, pp. 271–284, Mar. 2019.
- [31] E. Sharma, R. C. Deo, R. Prasad, and A. V. Parisi, "A hybrid air quality early-warning framework: An hourly forecasting model with online sequential extreme learning machines and empirical mode decomposition algorithms," *Sci. Total Environ.*, vol. 709, Mar. 2020, Art. no. 135934.
- [32] T. Nitta, "Orthogonality of decision boundaries in complex-valued neural networks," *Neural Comput.*, vol. 16, no. 1, pp. 73–97, Jan. 2004.
- [33] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.
- [34] S. L. Goh and D. P. Mandic, "A complex-valued RTRL algorithm for recurrent neural networks," *Neural Comput.*, vol. 16, no. 12, pp. 2699–2713, Dec. 2004.
- [35] X.-L. Li and T. Adali, "Noncircular complex ICA by generalized householder reflections," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6423–6430, Dec. 2013.
- [36] M.-B. Li, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, Oct. 2005.
- [37] G.-B. Huang, M.-B. Li, L. Chen, and C.-K. Siew, "Incremental extreme learning machine with fully complex hidden nodes," *Neurocomputing*, vol. 71, nos. 4–6, pp. 576–583, Jan. 2008.
- [38] R. Savitha, S. Suresh, and N. Sundararajan, "Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems," *Inf. Sci.*, vol. 187, pp. 277–290, Mar. 2012.
- [39] E. A. Wan, "Neural network classification: A Bayesian interpretation," *IEEE Trans. Neural Netw.*, vol. 1, no. 4, pp. 303–305, Dec. 1990.
- [40] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, Aug. 1995.
- [41] D. P. Mandic and S. L. Goh, "Complex valued nonlinear adaptive filters: Noncircularity," in *Widely Linear and Neural Models*. New York, NY, USA: Wiley, 2009.
- [42] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data—The Theory of Improper and Noncircular Signals*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
- [43] S. L. Goh and D. P. Mandic, "An augmented CRTRL for complex-valued recurrent neural networks," *Neural Netw.*, vol. 20, no. 10, pp. 1061–1066, Dec. 2007.
- [44] S. L. Goh and D. P. Mandic, "An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks," *Neural Comput.*, vol. 19, no. 4, pp. 1039–1055, Apr. 2007.
- [45] Y. Xia, S. C. Douglas, and D. P. Mandic, "Performance analysis of the deficient length augmented CLMS algorithm for second order noncircular complex signals," *Signal Process.*, vol. 144, pp. 214–225, Mar. 2018.
- [46] Y. Xia and D. P. Mandic, "A full mean square analysis of CLMS for second-order noncircular inputs," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5578–5590, Nov. 2017.
- [47] Y. Xia, B. Jelfs, M. M. Van Hulle, J. C. Principe, and D. P. Mandic, "An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 74–83, Jan. 2011.
- [48] D. Xu, J. Dong, and H. Zhang, "Deterministic convergence of wirtinger-gradient methods for complex-valued neural networks," *Neural Process. Lett.*, vol. 45, no. 2, pp. 445–456, Apr. 2017.
- [49] H. Zhang, Y. Wang, D. Xu, J. Wang, and L. Xu, "The augmented complex-valued extreme learning machine," *Neurocomputing*, vol. 311, pp. 363–372, Oct. 2018.
- [50] W. Cao, Z. Ming, Z. Xu, J. Zhang, and Q. Wang, "Online sequential extreme learning machine with dynamic forgetting factor," *IEEE Access*, vol. 7, pp. 179746–179757, 2019.
- [51] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen ver änderlichen," *Ann. Math.*, vol. 97, no. 1, pp. 357–375, 1927.
- [52] H. Li and T. Adali, "Algorithms for complex ML ICA and their stability analysis using Wirtinger calculus," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6156–6167, Dec. 2010.
- [53] H. Zhang and D. P. Mandic, "Is a complex-valued stepsize advantageous in complex-valued gradient learning algorithms?" *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2730–2735, Dec. 2016.
- [54] C. Jahanchahi, S. Kanna, and D. Mandic, "Complex dual channel estimation: Cost effective widely linear adaptive filtering," *Signal Process.*, vol. 104, pp. 33–42, Nov. 2014.
- [55] P. J. Schreier, "The degree of impropriety (noncircularity) of complex random vectors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 3909–3912.
- [56] W. Cao, J. Gao, Z. Ming, S. Cai, and H. Zheng, *Impact of Probability Distribution Selection on RVFL Performance* (Lecture Notes in Computer Science), vol. 10699. Cham, Switzerland: Springer, 2008, pp. 114–124.
- [57] W. Cao, M. J. A. Patwary, P. Yang, X. Wang, and Z. Ming, "An initial study on the relationship between meta features of dataset and the initialization of NNRW," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

- [58] W. Cao, L. Hu, J. Gao, X. Wang, and Z. Ming, "A study on the relationship between the rank of input data and the performance of random weight neural network," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12685–12696, Aug. 2020.
- [59] K. Aihara, *Applied Chaos and Applicable Chaos*. Tokyo, Japan: Science-Sha, 1994.
- [60] I. Cha and S. A. Kassam, "Channel equalization using adaptive complex radial basis function networks," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 122–131, Jan. 1995.



HUI WANG is currently pursuing the M.S. degree in applied mathematics with Dalian Maritime University, in 2018. Her research interests include intelligent computing and neural networks.



YUANYUAN WANG received the M.S. degree from Dalian Maritime University, in 2017, where she is currently pursuing the Ph.D. degree in marine electric engineering.



SHUAI ZHU is currently pursuing the M.S. degree in applied mathematics with Dalian Maritime University. His research interests include intelligent computing and neural networks.



HUISHENG ZHANG received the M.S. degree from Xiamen University, in 2003, and the Ph.D. degree from the Dalian University of Technology, in 2009. From April 2014 to March 2015, he was supported by the China Scholarship Council (CSC) to work as a Visiting Scholar at Imperial College London (ICL), U.K. He is currently a Professor with Dalian Maritime University. His research interests include neural networks, signal processing, and learning theory.

...