IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2018, accepted February 13, 2018, date of publication February 20, 2018, date of current version March 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808218

Kernel Adaptive Filters With Feedback Based on Maximum Correntropy

SHIYUAN WANG[1,2], (Member, IEEE), LUJUAN DANG[1,2], WANLI WANG[1,2], GUOBING QIAN[1,2], AND CHI K. TSE[3], (Fellow, IEEE)

[1]College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China
[2]Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China
[3]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Shiyuan Wang (wsy@swu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671389 and Grant 61701419, in part by the China Postdoctoral Science Foundation Funded Project under Grant 2016M590853 and Grant 2017M610583, in part by the Chongqing Postdoctoral Science Foundation Special Funded Project under Grant Xm2017107, and in part by the Fundamental Research Funds for the Central Universities under Grant XDJK2017D177 and Grant XDJK2017D178.

ABSTRACT This paper presents novel kernel adaptive filters with feedback, namely, kernel recursive maximum correntropy with multiple feedback (KRMC-MF) and its simplified version, a linear recurrent kernel online learning algorithm based on maximum correntropy criterion (LRKOL-MCC). In LRKOL-MCC and KRMC-MF, single output and multiple outputs based on single delay are utilized to construct their feedback structure, respectively. Compared with the minimum mean square error criterion, the maximum correntropy criterion (MCC) adopted by LRKOL-MCC and KRMC-MF captures higher order statistics of errors. The proposed filters are, therefore, robust against outliers. Therefore, the past information can be reused to improve filtering performance in terms of the steady-state mean square error. The convergence characteristics of the filter parameters in LRKOL-MCC and KRMC-MF are also derived. Simulations on chaotic time-series prediction and nonlinear regression illustrate the desirable accuracy and robustness of the proposed filters.

INDEX TERMS Kernel adaptive filters, maximum correntropy, minimum mean square error, feedback structure, convergence.

I. INTRODUCTION

Over the past decade, kernel methods have been applied to solving nonlinear issues in signal processing, remote sensing, and machine learning [1]. In the kernel method, the input space is translated into a high or even infinite dimension reproducing kernel Hilbert space (RKHS) for modeling the nonlinear relation existing in the input and the output spaces [2]–[4] using some nonlinear mapping functions. To avoid the direct calculation of these functions, the nonlinear issue is changed into the inner product in RKHS. Further, by virtue of the reproducing property of RKHS, the inner product in RKHS is evaluated only by a Mercer kernel that is a continuous, symmetric and positive definite function [5], [6]. Based on the kernel method, kernel adaptive filters (KAFs) can therefore solve the nonlinear adaptive filtering problem by taking the linear form in RKHS. The classical KAFs include the kernel least mean square (KLMS) [7], kernel affine projection algorithm (KAPA) [8], and kernel recursive least squares (KRLS) [9]. Generally, feedback networks (FNs) can be introduced to improve performance in the fields of signal processing or neural networks [10]. Hence, the feedback structures are also applied to kernel adaptive filters, generating the linear recurrent kernel online learning (LRKOL) algorithm [11], kernel least mean square with single feedback (SF-KLMS) algorithm [12], regularized kernel least mean square algorithm with multiple-delay feedback (RKLMS-MDF) algorithm [13], and kernel recursive least squares with multiple feedback (KRLS-MF) [14]. The linearly increasing network sizes in these KAFs limit their online applications [15]. Therefore, sparsification methods such as the novelty criterion [16], surprise criterion [17], and quantization method [18], [19] are required to reduce the network size of KAFs.

The aforementioned KAFs based on the minimum mean square error (MMSE) criterion usually achieve better filtering performance for the case of Gaussian noises. Actually,

MMSE merely contains the second order statistics of errors, and is therefore sensitive to non-Gaussian noises or large outliers. To overcome this issue, information theoretic learning (ITL) has been introduced as the cost function for the non-Gaussian cases. Compared with MMSE, ITL can incorporate the complete distribution of errors into the learning process, resulting in an improvement on filtering precision and robustness against outliers. A variety of ITL criterions [20]–[22], e.g., minimum error entropy (MEE) [23], [24] and maximum correntropy criterion (MCC) [25]–[36], have already been combined into adaptive learning systems. As a similarity measure of two random variables, MCC with a lower computational cost has been applied to KRLS, generating the corresponding kernel recursive maximum correntropy (KRMC) algorithm. The KRMC algorithm inheriting the advantages of both KRLS and MCC, can achieve excellent performance for solving nonlinear problems from the aspects of filtering accuracy and stability in impulsive noise environments [37].

In this paper, by virtue of high accuracy and robustness against outliers of KRMC, a novel feedback structure based on multiple single-delay outputs is introduced into KRMC, generating novel kernel recursive maximum correntropy with multiple feedback (KRMC-MF). To further reduce the computational complexity, a simplified feedback structure based on only one single-delay output is also presented to construct a linear recurrent kernel online learning algorithm based on maximum correntropy criterion (LRKOL-MCC), which was also not reported before.

The rest of this paper is organized as follows. Section II briefly reviews the definition of correntropy. KRLS and KRMC are introduced in Section III. In Sections IV and V, a new KRMC-MF and its simplified version LRKOL-MCC are derived. Seciton VI presents the convergence analysis of the proposed filters. In Section VII, simulations on chaotic time series prediction and nonlinear regression illustrate the filtering accuracy and robustness of the proposed filters. Section VIII concludes this paper.

II. CORRENTROPY

The similarity between two variables can be measured by correntropy [38], [39]. Given two random variables X and Y with $f_{XY}(x, y)$ denoting the joint probability density function and x, y being realizations of X, Y, correntropy is defined as follows:

$$C(X, Y) = \iint \kappa(x, y) f_{XY}(x, y)\, dxdy. \tag{1}$$

Hence, $C(X, Y)$ can also be regarded as the expectation of the kernel function $\kappa(\cdot, \cdot)$. Generally, the Gaussian kernel with the universal approximation is widely utilized and takes the form of [40]–[42]

$$\kappa_{h_1}(x, y) = \exp\left(\frac{-(x-y)^2}{2h_1^2}\right), \tag{2}$$

where h_1 denotes the kernel width of correntropy. This implies that the Gaussian kernel is a shift-invariant kernel function, i.e., $\kappa_{h_1}(x, y) = \kappa_{h_1}(x - y)$.

Define $R = X - Y$. Equation (1) is therefore rewritten as

$$C(R) = \int \kappa_{h_1}(r) f_R(r)\, dr \tag{3}$$

where $r = x - y$ is the realization of the variable R associated with probability density function $f_R(r)$.

However, in practice, we may not know the exact joint probability density function $f_R(r)$, which precludes direct calculation of $C(R)$ by (3). Hence, the correntropy $C(R)$ is generally estimated using a finite number of independent identical distribution data $\{x_j, y_j\}_{j=1}^{M}$, i.e.,

$$\hat{C}(R) = \frac{1}{M}\sum_{j=1}^{M} \kappa_{h_1}(r_j) \tag{4}$$

with $r_j = x_j - y_j$.

According to (2), the optimization for minimizing r_j in (4) can be changed into maximizing the kernel function $\kappa_{h_1}(r_j)$, resulting in the following cost function under the maximum correntropy criterion (MCC).

$$\max \ \hat{C}(R) = \frac{1}{M}\sum_{j=1}^{M} \kappa_{h_1}(r_j). \tag{5}$$

Note that the coefficient $1/M$ as a scale factor only controls the magnitude of the correntropy $\hat{C}(R)$ and has no influence on the solution to (5). Therefore, the cost function under MCC is simplified by eliminating the coefficient $1/M$, i.e.,

$$\max \ \bar{C}(R) = \sum_{j=1}^{M} \kappa_{h_1}(r_j). \tag{6}$$

III. REVIEWS OF THE KERNEL RECURSIVE LEAST SQUARES AND KERNEL RECURSIVE MAXIMUM CORRENTROPY

A. KERNEL RECURSIVE LEAST SQUARES ALGORITHM

In the original Euclidean space, given the training input-output data at discrete time i denoted by $\{\boldsymbol{u}(j), d_j(i)\}_{j=1}^{i}$, where $\boldsymbol{u}(j) \in \mathbb{R}^{n\times 1}$ and $d_j(i) \in \mathbb{R}$ are the jth input vector and the desired output respectively, a continuous mapping $f : \mathbb{R}^{n\times 1} \rightarrow \mathbb{R}$ hidden in the training set is required to be learned. The nonlinear mapping transforming the input data from the Euclidean space into RKHS is denoted by $\varphi(\cdot)$. Let $\boldsymbol{\Psi}(i) = [\varphi(\boldsymbol{u}(1)), \varphi(\boldsymbol{u}(2)), \ldots, \varphi(\boldsymbol{u}(j)), \ldots, \varphi(\boldsymbol{u}(i))]$ and $\boldsymbol{d}(i) = [d_1(i), d_2(i), \ldots, d_j(i), \ldots, d_i(i)]^T$. According to the kernel method, the estimate of the latent function f_i at discrete time i can be modeled by the following inner product:

$$\hat{f}_i(\cdot) = \boldsymbol{\omega}(i)^T \varphi(\cdot), \tag{7}$$

where $\boldsymbol{\omega}(i)$ is the weight vector in RKHS. Based on all the training data at discrete time i, $\boldsymbol{\omega}(i)$ can be denoted by a linear combination of the feature input $\varphi(\boldsymbol{u}(j))$ and the coefficient α_j, i.e.,

$$\boldsymbol{\omega}(i) = \sum_{j=1}^{i} \alpha_j \varphi(\boldsymbol{u}(j)). \tag{8}$$

Substituting (8) into (7) yields

$$\hat{f}_i(\cdot) = \sum_{j=1}^{i} \alpha_j \varphi\left(\boldsymbol{u}\left(j\right)\right)^T \varphi(\cdot). \tag{9}$$

However, it is hard to find the mapping function $\varphi(\cdot)$ for different input data. Therefore, to avoid the direct calculation of this mapping function, the kernel trick [17] is used to transform the calculation of inner product based on $\varphi(\cdot)$ in (9) into the evaluation of kernel function $\kappa(\cdot, \cdot)$, i.e.,

$$\varphi\left(\boldsymbol{u}\left(j\right)\right)^T \varphi(\cdot) = \kappa(\boldsymbol{u}\left(j\right), \cdot). \tag{10}$$

Substituting (10) into (9) yields

$$\hat{f}_i(\cdot) = \sum_{j=1}^{i} \alpha_j \kappa(\boldsymbol{u}\left(j\right), \cdot). \tag{11}$$

Since there exists only the calculation of kernel functions in (11), the estimate $\hat{f}_i(\cdot)$ in (11) is more efficient than that in (9). Therefore, (11) is used in kernel adaptive filters generally.

According to (8) and (11), different approaches for updating the weight $\boldsymbol{\omega}(i)$ generates different kernel adaptive filters. In KRLS, $\boldsymbol{\omega}(i)$ is estimated by minimizing the following regularized loss function:

$$\min_{\boldsymbol{\omega}} \sum_{j=1}^{i} \left(d_j(i) - \hat{d}_j(i)\right)^2 + \lambda_1 \|\boldsymbol{\omega}(i)\|^2, \tag{12}$$

where λ_1 is the regularization factor for avoiding overfitting [17] and $\hat{d}_j(i) = \boldsymbol{\omega}(i)^T \varphi(\boldsymbol{u}(j))$ denotes the jth estimated output at discrete time i.

The solution to (12) yields

$$\boldsymbol{\omega}(i) = [\boldsymbol{\Psi}(i)\boldsymbol{\Psi}(i)^T + \lambda_1 \mathbf{I}]^{-1} \boldsymbol{\Psi}(i)\boldsymbol{d}(i). \tag{13}$$

The matrix inversion lemma is described by [17]

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}. \tag{14}$$

Let $\mathbf{A} = \lambda_1 \mathbf{I}$, $\mathbf{B} = \boldsymbol{\Psi}(i)$, $\mathbf{C} = \mathbf{I}$, and $\mathbf{D} = \boldsymbol{\Psi}(i)^T$. Thus, we can rewrite the weigh update of KRLS in (13) as

$$\begin{aligned} \boldsymbol{\omega}(i) &= \boldsymbol{\Psi}(i)[\mathbf{K}(i) + \lambda_1 \mathbf{I}]^{-1} \boldsymbol{d}(i) \\ &= \boldsymbol{\Psi}(i)\boldsymbol{\alpha}_s(i), \end{aligned} \tag{15}$$

where $\boldsymbol{\alpha}_s(i) = [\lambda_1 \mathbf{I} + \mathbf{K}(i)]^{-1} \boldsymbol{d}(i)$. The positive define matrix $\mathbf{K}(i)$ in (15) takes the form of $\mathbf{K}(i) = \boldsymbol{\Psi}(i)^T \boldsymbol{\Psi}(i)$, the m_1th row and m_2th column element of which can be evaluated by

$$\begin{aligned} \mathbf{K}_{m_1,m_2}(i) &= \kappa_{h_2}(\boldsymbol{u}\left(m_1\right), \boldsymbol{u}\left(m_2\right)) \\ &= \exp\left(\frac{-\|\boldsymbol{u}\left(m_1\right) - \boldsymbol{u}\left(m_2\right)\|^2}{2h_2^2}\right), \end{aligned} \tag{16}$$

where h_2 is the kernel width.

B. KERNEL RECURSIVE MAXIMUM CORRENTROPY

Kernel recursive maximum correntropy (KRMC) incorporating the maximum correntropy criterion into KRLS, has been proven to be robust against outliers. Therefore, the regularized MCC in KRMC is described by

$$\max_{\boldsymbol{\omega}} \sum_{j=1}^{i} \kappa_{h_1}\left(d_j(i) - \hat{d}_j(i)\right) - \frac{1}{2}\lambda_2 \|\boldsymbol{\omega}(i)\|^2, \tag{17}$$

where λ_2 denotes the regularization factor of KRMC and the jth estimated output $\hat{d}_j(i)$ is given by $\hat{d}_j(i) = \boldsymbol{\omega}(i)^T \varphi(\boldsymbol{u}(j))$. The solution to (17) can be derived as

$$\boldsymbol{\omega}(i) = \left(\boldsymbol{\Psi}(i)\bar{\boldsymbol{\Lambda}}(i)\boldsymbol{\Psi}(i)^T + \lambda_2 h_1^2 \mathbf{I}\right)^{-1} \boldsymbol{\Psi}(i)\bar{\boldsymbol{\Lambda}}(i)\boldsymbol{d}(i), \tag{18}$$

where $\bar{\boldsymbol{\Lambda}}(i) = \mathrm{diag}[\exp((d_1(i) - \hat{d}_1(i))^2/-2h_1^2), \exp((d_2(i) - \hat{d}_2(i))^2/-2h_1^2), \cdots, \exp((d_i(i) - \hat{d}_i(i))^2/-2h_1^2)]$.

Using the matrix inversion lemma shown in (14) by letting $\mathbf{A} = \lambda_2 h_1^2 \mathbf{I}$, $\mathbf{B} = \boldsymbol{\Psi}(i)$, $\mathbf{C} = \bar{\boldsymbol{\Lambda}}(i)$, and $\mathbf{D} = \boldsymbol{\Psi}(i)^T$, we rewrite (18) as

$$\boldsymbol{\omega}(i) = \boldsymbol{\Psi}(i)\boldsymbol{\alpha}(i), \tag{19}$$

where $\boldsymbol{\alpha}(i) = \left(\mathbf{K}(i) + \lambda_2 h_1^2 \bar{\boldsymbol{\Lambda}}(i)^{-1}\right)^{-1} \boldsymbol{d}(i)$ with the entry of positive define matrix $\mathbf{K}(i) = \boldsymbol{\Psi}(i)^T \boldsymbol{\Psi}(i)$ calculated by (16). Therefore, $\boldsymbol{\alpha}(i)$ is used to represent $\boldsymbol{\omega}(i)$ with the help of (19) hereafter.

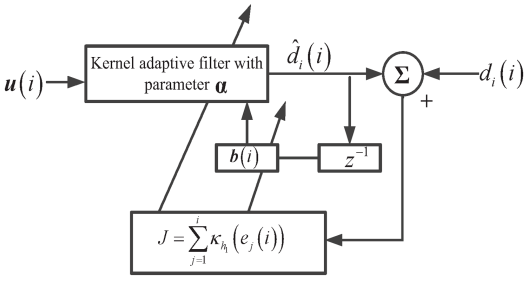


FIGURE 1. Block diagram of KRMC-MF at discrete time i.

IV. KERNEL RECURSIVE MAXIMUM CORRENTROPY WITH MULTIPLE FEEDBACK

A. KRMC-MF

Figure 1 shows the block diagram of KRMC-MF. In Fig. 1, we incorporate a recurrent scalar $\hat{d}_{j-1}(i-1)$ into the evaluation of $\hat{d}_j(i)$, i.e.,

$$\hat{d}_j(i) = \varphi(\boldsymbol{u}(j))^T \boldsymbol{\Psi}(i)\boldsymbol{\alpha}(i) + \boldsymbol{b}(i)^T \boldsymbol{\alpha}(i)\hat{d}_{j-1}(i-1), \quad j = 1, 2, \cdots i \tag{20}$$

where the column vector parameters $\boldsymbol{\alpha}(i)$ and $\boldsymbol{b}(i)$ represent the feedforward coefficient (FFC) and the feedback coefficient (FBC), respectively; the first term in the right side of (20) is the feedforward part; the second term in the right side of (20) includes the past information which can be regarded as the feedback term. At discrete time i,

$\hat{\boldsymbol{d}}(i)$ is formed by stacking $\left\{\hat{d}_j(i)\right\}_{j=1}^{i}$ into column vector $\hat{\boldsymbol{d}}(i)=[\hat{d}_1(i),\cdots,\hat{d}_j(i),\cdots,\hat{d}_i(i)]^T$, and $\boldsymbol{\Psi}(i)$ is formed by stacking $\{\varphi(\boldsymbol{u}(j))\}_{j=1}^{i}$. Thus, according to (20), the collection of estimated outputs $\hat{\boldsymbol{d}}(i)$ has the following linear form:

$$\hat{\boldsymbol{d}}(i)=(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i), \tag{21}$$

where column vector $\hat{\boldsymbol{d}}(i-1)$ is the multiple single-delay feedbacks.

According to the correntropy (4), we give the loss function of KRMC-MF as follows:

$$J=\sum_{j=1}^{i}\kappa_{h_1}(e_j(i)) \tag{22}$$

with $e_j(i)=d_j(i)-\hat{d}_j(i)$ denoting the jth estimated error at discrete time i. Each $\hat{d}_j(i)$ evaluated by (20) includes the feedback with single delay. Since the loss function (22) is the sum of Gaussian kernels, multiple single-delay feedbacks are included in KRMC-MF.

For the convenience of derivation, we first give the evaluation about the collection of errors at discrete time i, i.e., $\boldsymbol{e}(i)=[e_1(i),e_2(i),\cdots,e_i(i)]^T$. Here, we define the desired outputs by $\boldsymbol{d}(i)=(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}_*+\boldsymbol{v}(i)$, where $\boldsymbol{\alpha}_*$ and $\boldsymbol{b}_*$ are the optimal FFC and FBC, respectively; $\boldsymbol{v}(i)=[v(1),v(2),\cdots,v(i)]^T$ is the collection of disturbance noises. According to (21), we have

$$\begin{aligned}\boldsymbol{e}(i)&=\boldsymbol{d}(i)-\hat{\boldsymbol{d}}(i)\\&=\boldsymbol{d}(i)-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)\\&=(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}_*-(\mathbf{K}(i)\\&\quad+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)+\boldsymbol{v}(i)\\&=-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)+(\mathbf{K}(i)\\&\quad+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}_*-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}_*\\&\quad+(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}_*+\boldsymbol{v}(i)\\&=-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\hat{\boldsymbol{\alpha}}(i)-\hat{\boldsymbol{d}}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}_*\\&\quad+\boldsymbol{v}(i)\\&=\boldsymbol{\theta}_\alpha(i)-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\hat{\boldsymbol{\alpha}}(i)\\&\le\bar{\boldsymbol{\theta}}_\alpha^m-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\hat{\boldsymbol{\alpha}}(i),\end{aligned} \tag{23}$$

where $\hat{\boldsymbol{\alpha}}(i)=\boldsymbol{\alpha}(i)-\boldsymbol{\alpha}_*$; $\boldsymbol{\theta}_\alpha(i)=\boldsymbol{v}(i)-\hat{\boldsymbol{d}}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}_*$; the equivalent disturbance $\bar{\boldsymbol{\theta}}_\alpha^m$ is the column vector with each entry being not smaller than that of $\boldsymbol{\theta}_\alpha(i)$.

B. WEIGHT UPDATE

Based on the steepest ascent method [17], the FFC is updated iteratively by

$$\boldsymbol{\alpha}(i+1)=\boldsymbol{\alpha}(i)+s_\alpha^m(i)\frac{\partial J}{\partial\boldsymbol{\alpha}(i)}, \tag{24}$$

where $s_\alpha^m(i)\in\left\{0,\ 1\big/\rho_\alpha^m(i)\right\}$ denotes the step size with a positive scalar $\rho_\alpha^m(i)$. Taking the derivative of (22) regarding $\boldsymbol{\alpha}(i)$, we have

$$\frac{\partial J}{\partial\boldsymbol{\alpha}(i)}=\sum_{j=1}^{i}\kappa_{h_1}(d_j(i)-\hat{d}_j(i))\frac{e_j(i)}{h_1^2}\frac{\partial\hat{d}_j(i)}{\partial\boldsymbol{\alpha}(i)}. \tag{25}$$

Further, according to (20), the differential $\partial\hat{d}_j(i)/\partial\boldsymbol{\alpha}(i)$ can be expressed as

$$\begin{aligned}\frac{\partial\hat{d}_j(i)}{\partial\boldsymbol{\alpha}(i)}&\approx\boldsymbol{\Psi}(i)^T\varphi(\boldsymbol{u}(j))+\boldsymbol{b}(i)\hat{d}_{j-1}(i-1)\\&\quad+r_\alpha^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\frac{\partial\hat{d}_{j-1}(i-1)}{\partial\boldsymbol{\alpha}(i-1)},\end{aligned} \tag{26}$$

where $r_\alpha^m(i)$ is the parameter for adjusting the recurrent gradient information [43] and we use the approximation $\partial\hat{d}_{j-1}(i-1)\big/\partial\boldsymbol{\alpha}(i)\approx\partial\hat{d}_{j-1}(i-1)\big/\partial\boldsymbol{\alpha}(i-1)$.

Define $\boldsymbol{\Upsilon}_\alpha^m(i)=\partial\hat{\boldsymbol{d}}(i)/\partial\boldsymbol{\alpha}(i)$. According to (21), we rewrite (26) by

$$\boldsymbol{\Upsilon}_\alpha^m(i)\approx\mathbf{K}(i)+\boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T+r_\alpha^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_\alpha^m(i-1). \tag{27}$$

Combining (26) and (27), we rewrite (25) by $\partial J\big/\partial\boldsymbol{\alpha}(i)=\boldsymbol{\Upsilon}_\alpha^m(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)$ with $\boldsymbol{\Lambda}(i)=\bar{\boldsymbol{\Lambda}}(i)/h_1^2$, where $\bar{\boldsymbol{\Lambda}}(i)$ is defined after (18). Hence, the recursion of $\boldsymbol{\alpha}(i)$ in (24) can be updated by

$$\boldsymbol{\alpha}(i+1)=\boldsymbol{\alpha}(i)+s_\alpha^m(i)\boldsymbol{\Upsilon}_\alpha^m(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i). \tag{28}$$

Similar to (28), the recursive form of the FBC $\boldsymbol{b}(i)$ is given by

$$\boldsymbol{b}(i+1)=\boldsymbol{b}(i)+s_b^m(i)\boldsymbol{\Upsilon}_b^m(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i), \tag{29}$$

where $s_b^m(i)\in\left\{0,\ 1\big/\rho_b^m(i)\right\}$ is the step size with $\rho_b^m(i)$ being positive; $\partial\hat{\boldsymbol{d}}(i-1)\big/\partial\boldsymbol{b}(i)\approx\partial\hat{\boldsymbol{d}}(i-1)\big/\partial\boldsymbol{b}(i-1)$ and $\boldsymbol{\Upsilon}_b^m(i)=\partial\hat{\boldsymbol{d}}(i)\big/\partial\boldsymbol{b}(i)$. Hence, $\boldsymbol{\Upsilon}_b^m(i)$ is approximated by

$$\boldsymbol{\Upsilon}_b^m(i)\approx\boldsymbol{\alpha}(i)\hat{\boldsymbol{d}}(i-1)^T+r_b^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_b^m(i-1). \tag{30}$$

According to (29), we rewrite $\boldsymbol{e}(i)$ by

$$\begin{aligned}\boldsymbol{e}(i)&=\boldsymbol{d}(i)-\hat{\boldsymbol{d}}(i)\\&=-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)\\&\quad+(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}_*+\boldsymbol{v}(i)\\&=-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)\\&\quad+(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}(i)-(\mathbf{K}(i)\\&\quad+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}(i)+(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\boldsymbol{\alpha}_*+\boldsymbol{v}(i)\\&=-\hat{\boldsymbol{d}}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}(i)-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\hat{\boldsymbol{\alpha}}(i)+\boldsymbol{v}(i)\\&=\boldsymbol{\theta}_b(i)-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\hat{\boldsymbol{\alpha}}(i)\\&\le\bar{\boldsymbol{\theta}}_b^m-(\mathbf{K}(i)+\hat{\boldsymbol{d}}(i-1)\boldsymbol{b}_*^T)\hat{\boldsymbol{\alpha}}(i),\end{aligned} \tag{31}$$

where $\hat{\boldsymbol{b}}(i)=\boldsymbol{b}(i)-\boldsymbol{b}_*$; $\boldsymbol{\theta}_b(i)=\boldsymbol{v}(i)-\hat{\boldsymbol{d}}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}(i)$; the equivalent disturbance $\bar{\boldsymbol{\theta}}_b^m$ is the column vector with each element being not smaller than that of $\boldsymbol{\theta}_b(i)$.

For compactness of the notation, $\boldsymbol{\Omega}=\{\boldsymbol{\alpha},\boldsymbol{b}\}$ is introduced. The parameter settings are discussed as follows.

To guarantee the convergence of KRMC-MF, we establish the following updating rules regarding $s_{\boldsymbol{\Omega}}^m(i)$ and $r_{\boldsymbol{\Omega}}^m(i)$:

1. $s_{\boldsymbol{\Omega}}^m(i)=z_{\boldsymbol{\Omega}}^m(i)\big/\rho_{\boldsymbol{\Omega}}^m(i)$ in (28) and (29) is evaluated by

$$z_{\boldsymbol{\Omega}}^m(i)=\begin{cases}1, & \text{if}\quad\bar{\epsilon}_{\boldsymbol{\Omega}}^m(i)<2\left(\boldsymbol{e}(i)-\bar{\boldsymbol{\theta}}_{\boldsymbol{\Omega}}^m\right)^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)\\0, & \text{else},\end{cases} \tag{32}$$

where $\bar{\epsilon}^m_{\boldsymbol{\Omega}}(i) = \epsilon^m_{\boldsymbol{\Omega}}(i) + \epsilon^m_{\boldsymbol{\Omega}}(i-1)$ with $\epsilon^m_{\boldsymbol{\Omega}}(i) = \left\|\boldsymbol{\Upsilon}^m_{\boldsymbol{\Omega}}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)\right\|^2 / \rho^m_{\boldsymbol{\Omega}}(i)$.

The normalization factor $\rho^m_{\boldsymbol{\Omega}}(i)$ in (28) and (29) is designed to prevent the vanished cone problem [11] which is updated by

$$\rho^m_{\boldsymbol{\Omega}}(i) = \max(\rho^m_{\boldsymbol{\Omega}}(i-1), \vartheta^m_{\boldsymbol{\Omega}}\rho^m_{\boldsymbol{\Omega}}(i-1) + \max(\bar{\varrho}^m_{\boldsymbol{\Omega}}, ||\boldsymbol{\Upsilon}^m_{\boldsymbol{\Omega}}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)||^2)), \quad (33)$$

where $\vartheta^m_{\boldsymbol{\Omega}} \in (0, 1)$ denotes the impact degree of the previous factor on the current one and $\bar{\varrho}^m_{\boldsymbol{\Omega}}$ is a regulatory factor that ensures the initial learning rate being relatively small [12].

2. $r^m_{\boldsymbol{\Omega}}(i)$ in (26) and (30) is defined by

$$r^m_{\boldsymbol{\Omega}}(i) = \begin{cases} sgn\left(\bar{\zeta}^m_{\boldsymbol{\Omega}}(i)\right), & \text{if } |\mu^m(i)| < \dfrac{1}{\eta^m + |\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)|} \\ 0, & \text{else,} \end{cases} \quad (34)$$

where sgn$(\cdot)$ denotes the sign function; $\bar{\zeta}^m_{\boldsymbol{\Omega}}(i) = \boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^m(i)z^m_{\boldsymbol{\Omega}}(i-1)$ and

$$\mu^m(i) = \boldsymbol{L}^m(i-1)^T(\boldsymbol{L}^m(i-1)\boldsymbol{L}^m(i-1)^T)^{-1}\boldsymbol{L}^m(i) \quad (35)$$

with $\boldsymbol{L}^m(i-1) = \boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1)$; η^m is a small positive constant to make the denominator nonzero. The proposed KRMC-MF is therefore summarized in **Algorithm 1**.

Remark 1: The choice of step size $s^m_{\boldsymbol{\Omega}}(i) = z^m_{\boldsymbol{\Omega}}(i)/\rho^m_{\boldsymbol{\Omega}}(i)$ in (28) and (29) is crucial for filtering convergence performance. The normalization factor $\rho^m_{\boldsymbol{\Omega}}(i)$ is used for solving the vanishing radius problem [11], [44] and $z^m_{\boldsymbol{\Omega}}(i)$ can be regarded as the switch controlling the updates of FFC and FBC. $r^m_{\boldsymbol{\Omega}}(i)$ in (26) and (30) has the function of regulation factor λ_2 in (17). Furthermore, the update form of weight in KRMC-MF is similar to that in the momentum least mean square (MLMS) algorithm [43], which has been proven in Appendix. Therefore, the feedback manner in KRMC-MF can reuse the past information efficiently, and thus improve the filtering performance.

Algorithm 1 Kernel Recursive Maximum Correntropy With Multiple Feedback (KRMC-MF) Algorithm

Initiation:

Start with parameters initiation: $\boldsymbol{\theta}^m_\alpha$, $\boldsymbol{\theta}^m_b$, $\bar{\varrho}^m_\alpha$, $\bar{\varrho}^m_b$, ϑ^m_α, ϑ^m_b, η^m.

while $\{\boldsymbol{u}(i), d(i)\}(i > 1)$ available **do**

1) Update $s^m_\alpha(i)$, $s^m_b(i)$ using (32) and (33), $r^m_\alpha(i)$ and $r^m_b(i)$ using (34).

2) The update forms of $\boldsymbol{\Upsilon}^m_\alpha(i-1)$ and $\boldsymbol{\Upsilon}^m_b(i-1)$ are shown as follows.

$$\boldsymbol{\Upsilon}^m_\alpha(i) \approx \mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T + r^m_\alpha(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Theta}^\alpha_{i-1},$$
$$\boldsymbol{\Upsilon}^m_b(i) \approx \boldsymbol{\alpha}(i)\hat{\boldsymbol{d}}(i-1)^T + r^m_b(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Theta}^b_{i-1},$$

with $\hat{\boldsymbol{d}}(i-1) = [\hat{\boldsymbol{d}}_0(i-1); \hat{\boldsymbol{d}}(i-1)]$ and $\hat{\boldsymbol{d}}_0(i-1) = 0$ in (22); $\boldsymbol{\Theta}^\alpha_{i-1}$ and $\boldsymbol{\Theta}^b_{i-1}$ are given by

$$\boldsymbol{\Theta}^\alpha_{i-1} = \begin{bmatrix} \mathbf{0} & \boldsymbol{\Upsilon}^m_\alpha(i-1) \\ 0 & \mathbf{0}^T \end{bmatrix}, \quad \boldsymbol{\Theta}^b_{i-1} = \begin{bmatrix} \mathbf{0} & \boldsymbol{\Upsilon}^m_b(i-1) \\ 0 & \mathbf{0}^T \end{bmatrix},$$

with $\mathbf{0}$ being the $((i-1)\times 1)$ null vector.

3) The FFC $\boldsymbol{\alpha}(i)$ and FBC $\boldsymbol{b}(i)$ can be approximated iteratively by the recursions

$$\boldsymbol{\alpha}(i+1) = \begin{bmatrix} \boldsymbol{\alpha}(i) \\ 0 \end{bmatrix} + s^m_\alpha(i)\boldsymbol{\Upsilon}^m_\alpha(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i),$$
$$\boldsymbol{b}(i+1) = \begin{bmatrix} \boldsymbol{b}(i) \\ 0 \end{bmatrix} + s^m_b(i)\boldsymbol{\Upsilon}^m_b(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i).$$

end while

V. LINEAR RECURRENT KERNEL ONLINE LEARNING ALGORITHM BASED ON MAXIMUM CORRENTROPY CRITERION

A. LRKOL-MCC

In KRMC-MF, the summation of correntropy based on all errors in (22) is used as the loss function. Therefore, to further reduce the computational complexity of KRMC-MF, a linear recurrent kernel online learning algorithm based on maximum correntropy criterion (LRKOL-MCC) uses the simplified correntropy based on only the current error as the loss function, i.e.,

$$J = \kappa_{h_1}(e_i(i)), \quad (36)$$

where $e_i(i) = d_i(i) - \hat{d}_i(i)$. For simplicity, (36) can be rewritten by

$$J = \kappa_{h_1}(e(i)), \quad (37)$$

where $e(i) = d(i) - \hat{d}(i)$ and $\hat{d}(i) = (\mathbf{K}_i(i)^T + \hat{d}(i-1)\boldsymbol{b}(i)^T)\boldsymbol{\alpha}(i)$ with $\mathbf{K}_i(i)$ being the last column of $\mathbf{K}(i)$. In comparison with (22) and (37), LRKOL-MCC has lower computational complexity than KRMC-MF.

B. WEIGHT UPDATE

Similar to (28) and (29) in KRMC-MF, the update form of FFC in LRKOL-MCC is given by

$$\boldsymbol{\alpha}(i+1) = \boldsymbol{\alpha}(i) + s^s_\alpha(i)\boldsymbol{\Upsilon}^s_\alpha(i)\Lambda_{i,i}(i)e(i), \quad (38)$$

where $s^s_\alpha(i) \in \left\{0,\ 1/\rho^s_\alpha(i)\right\}$ is the step size with a positive scalar $\rho^s_\alpha(i)$; $\Lambda_{i,i}(i) = \exp((d(i) - \boldsymbol{\omega}^T\varphi(\boldsymbol{u}(i)))^2/-2h_1^2)$ denotes the ith column and ith row element of diagonal matrix $\boldsymbol{\Lambda}(i)$ in (28).

Vector $\boldsymbol{\Upsilon}^s_\alpha(i) = \partial\hat{d}(i)/\partial\boldsymbol{\alpha}(i)$ in (38) has a similar update form to that in (27), i.e.,

$$\boldsymbol{\Upsilon}^s_\alpha(i) \approx \mathbf{K}_i(i) + \boldsymbol{b}(i)\hat{d}(i-1) + r^s_\alpha(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}^s_\alpha(i-1), \quad (39)$$

where $r^s_\alpha(i)$ is the parameter for adjusting the recurrent gradient information of LRKOL-MCC.

For FBC $\boldsymbol{b}(i)$, we obtain from (29)

$$\boldsymbol{b}(i+1) = \boldsymbol{b}(i) + s_b^s(i)\boldsymbol{\Upsilon}_b^s(i)\Lambda_{i,i}(i)e(i), \tag{40}$$

where the step size $s_b^s(i) \in \{0,\ 1/\rho_b^s(i)\}$ with $\rho_b^s(i)$ being positive and $\boldsymbol{\Upsilon}_b^s(i) = \partial\hat{d}(i)/\partial\boldsymbol{b}(i)$ takes the form of

$$\boldsymbol{\Upsilon}_b^s(i) \approx \boldsymbol{\alpha}(i)\ \hat{d}(i-1) + r_b^s(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_b^s(i-1). \tag{41}$$

The parameter settings are discussed as follows.
The $s_{\boldsymbol{\Omega}}^s(i)$ and $r_{\boldsymbol{\Omega}}^s(i)$ obey the following updating rules.
1. $s_{\boldsymbol{\Omega}}^s(i) = z_{\boldsymbol{\Omega}}^s(i)/\rho_{\boldsymbol{\Omega}}^s(i)$ in (38) and (40) is given by

$$z_{\boldsymbol{\Omega}}^s(i) = \begin{cases} 1, & \text{if } \ \bar{\epsilon}_{\Omega}^s(i) < 2\left(e(i) - \bar{\theta}_{\Omega}^s\right)^T\Lambda_{i,i}(i)e(i) \\ 0, & \text{else,} \end{cases} \tag{42}$$

where $\bar{\epsilon}_{\boldsymbol{\Omega}}^s(i) = \epsilon_{\boldsymbol{\Omega}}^s(i) + \epsilon_{\boldsymbol{\Omega}}^s(i-1)$ with $\epsilon_{\boldsymbol{\Omega}}^s(i) = \left\|\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^s(i)\Lambda_{i,i}(i)e(i)\right\|^2/\rho_{\boldsymbol{\Omega}}^s(i)$. The scalar $\bar{\theta}_{\Omega}^s$ is a positive number larger than θ_{Ω}^s with $\theta_{\alpha}^s = v(i) - \hat{d}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}_*$ and $\theta_b^s(i) = v(i) - \hat{d}(i-1)\hat{\boldsymbol{b}}(i)^T\boldsymbol{\alpha}(i)$. Here, the normalization factor $\rho_{\boldsymbol{\Omega}}^s(i)$ in (38) and (40) can be updated by

$$\rho_{\boldsymbol{\Omega}}^s(i) = \max(\rho_{\boldsymbol{\Omega}}^s(i-1),\ \vartheta_{\boldsymbol{\Omega}}^s\rho_{\boldsymbol{\Omega}}^s(i-1) + \max(\bar{\varrho}_{\boldsymbol{\Omega}}^s,\ ||\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^s(i)\Lambda_{i,i}(i)e(i)||^2)), \tag{43}$$

with $\vartheta_{\boldsymbol{\Omega}}^s \in (0,\ 1)$ and $\bar{\varrho}_{\boldsymbol{\Omega}}^s$ being positive.
2. $r_{\boldsymbol{\Omega}}^s(i)$ in (39) and (41) is given by

$$r_{\boldsymbol{\Omega}}^s(i) = \begin{cases} sgn\left(\bar{\zeta}_{\boldsymbol{\Omega}}^s(i)\right), & \text{if } \ |\mu^s(i)| < \dfrac{1}{\eta^s + |\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)|} \\ 0, & \text{else,} \end{cases} \tag{44}$$

where $\bar{\zeta}_{\boldsymbol{\Omega}}^s(i) = \boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^s(i)z_{\boldsymbol{\Omega}}^s(i-1)$; $\mu^s(i) = L^s(i)/L^s(i-1)$ with $L^s(i-1) = \bar{\Lambda}_{i,i}(i-1)e(i-1)$; and η^s is a relatively small positive constant to make the denominator nonzero. The proposed KRMC-MF is therefore summarized in **Algorithm 2**.

Remark 2: Compared with **Algorithm 1** and **Algorithm 2**, the difference between KRMC-MF and LRKOL-MCC is the update form of $\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^m(i)$ and $\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^s(i)$. Since multiple single-delay feedbacks are incorporated into KRMC-MF, $\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^m(i)$ is updated recursively in the form of matrix. However, $\boldsymbol{\Upsilon}_{\boldsymbol{\Omega}}^s(i)$ in LRKOL-MCC is recursively updated in the form of vector due to only one single-delay feedback. Therefore, LRKOL-MCC requires lower computational burden than KRMC-MF.

C. COMPUTATIONAL COMPLEXITY

The comparison of computational costs at discrete time i between the MMSE-based algorithms, i.e., LRKOL, KRLS, and KRLS-MF, and the MCC-based algorithms, i.e., LRKOL-MCC, KRMC, and KRMC-MF is shown in TABLE 1. In comparison with the MMSE-based algorithms, the MCC-based algorithms incur more computational burden owing to the calculation of the additional entropy functions, i.e., the scalar $\Lambda_{i,i}(i)$ in LRKOL-MCC, and the matrices $\boldsymbol{B}(i)$ [37] and $\boldsymbol{\Lambda}(i)$ with diagonal entries calculated by (2) in KRMC and KRMC-MF. In addition, compared with the KAFs with no feedback, i.e., KRLS and KRMC, other KAFs have the additional computational burden induced by the recurrent terms. Therefore, we see from TABLE 1 that the proposed LRKOL-MCC and KRMC-MF have almost the same computational costs as LRKOL and KRLS-MF, respectively, which will also be presented in the following simulations.

Algorithm 2 Linear Recurrent Kernel Online Learning Algorithm Based on Maximum Correntropy Criterion (LRKOL-MCC)

Initiation:
Start with parameters initiation: $\bar{\theta}_{\alpha}^s$, $\bar{\theta}_b^s$, $\bar{\varrho}_{\alpha}^s$, $\bar{\varrho}_b^s$, ϑ_{α}^s, ϑ_b^s, η^s.

while $\{\boldsymbol{u}(i), d(i)\}(i > 1)$ available **do**

1) Update $s_{\alpha}^s(i)$, $s_b^s(i)$ by (42) and (43), $r_{\alpha}^s(i)$ and $r_b^s(i)$ by (44).

2) The recursions of $\boldsymbol{\Upsilon}_{\alpha}^s(i-1)$ and $\boldsymbol{\Upsilon}_b^s(i-1)$ are shown as follows.

$$\boldsymbol{\Upsilon}_{\alpha}^s(i) \approx \mathbf{K}(i) + \boldsymbol{b}(i)\hat{d}(i-1) + r_{\alpha}^s(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i) \times \begin{bmatrix} \boldsymbol{\Upsilon}_{\alpha}^s(i-1) \\ 0 \end{bmatrix},$$

$$\boldsymbol{\Upsilon}_b^s(i) \approx \boldsymbol{\alpha}(i)\hat{d}(i-1) + r_b^s(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\begin{bmatrix} \boldsymbol{\Upsilon}_b^s(i-1) \\ 0 \end{bmatrix}.$$

3) The FFC $\boldsymbol{\alpha}(i)$ and FBC $\boldsymbol{b}(i)$ can be updated recursively.

$$\boldsymbol{\alpha}(i+1) = \begin{bmatrix} \boldsymbol{\alpha}(i) \\ 0 \end{bmatrix} + s_{\alpha}^s(i)\boldsymbol{\Upsilon}_{\alpha}^s(i)\Lambda_{i,i}(i)e(i),$$

$$\boldsymbol{b}(i+1) = \begin{bmatrix} \boldsymbol{b}(i) \\ 0 \end{bmatrix} + s_b^s(i)\boldsymbol{\Upsilon}_b^s(i)\Lambda_{i,i}(i)e(i).$$

end while

TABLE 1. Computational costs of KAFs at discrete time i.

Algorithm	Computational Costs	
LRKOL [11]	Update $\boldsymbol{a}(i)/\boldsymbol{\lambda}(i)$	$O(i)$
	Update $\boldsymbol{D}^{\alpha}(i)/\boldsymbol{D}^{\lambda}(i)$	$O(i)$
LRKOL-MCC	Update $\Lambda_{i,i}(i)$	$O(1)$
	Update $\boldsymbol{a}(i)/\boldsymbol{b}(i)$	$O(i)$
	Update $\boldsymbol{\Upsilon}_{\alpha}^s(i)/\boldsymbol{\Upsilon}_b^s(i)$	$O(i)$
KRLS [9]	Update $\boldsymbol{a}(i)$	$O(i)$
	Update $\boldsymbol{Q}(i)$	$O(i^2)$
KRMC [37]	Update $\boldsymbol{B}(i)$	$O(i)$
	Update $\boldsymbol{a}(i)$	$O(i)$
	Update $\boldsymbol{Q}(i)$	$O(i^2)$
KRLS-MF [14]	Update $\boldsymbol{a}(i)/\boldsymbol{b}(i)$	$O(i)$
	Update $\boldsymbol{D}^{\alpha}(i)/\boldsymbol{D}^{\lambda}(i)$	$O(i^2)$
KRMC-MF	Update $\boldsymbol{\Lambda}(i)$	$O(i)$
	Update $\boldsymbol{a}(i)/\boldsymbol{b}(i)$	$O(i)$
	Update $\boldsymbol{\Upsilon}^{\alpha}(i)/\boldsymbol{\Upsilon}^{\lambda}(i)$	$O(i^2)$

VI. CONVERGENCE ANALYSIS

This section presents the convergence analysis of KRMC-MF based on the designed learning parameters in (32), (33), and (34).

Theorem 1: Define $\Delta_{\boldsymbol{\alpha}}(i) = \left\|\hat{\boldsymbol{\alpha}}(i+1)\right\|^2 - \left\|\hat{\boldsymbol{\alpha}}(i)\right\|^2$ and $\Delta_{\boldsymbol{b}}(i) = \left\|\hat{\boldsymbol{b}}(i+1)\right\|^2 - \left\|\hat{\boldsymbol{b}}(i)\right\|^2$. For KRMC-MF, the update recursions in (28) and (29) guarantee the convergence of FFC and FBC in terms of $\lim_{i\to\infty}|\Delta_{\boldsymbol{\alpha}}(i)| = 0$ and $\lim_{i\to\infty}|\Delta_{\boldsymbol{b}}(i)| = 0$, respectively.

Proof: The optimal FFC $\boldsymbol{\alpha}_*$ subtracted from both sides of (28) gives

$$\hat{\boldsymbol{\alpha}}(i+1) = \hat{\boldsymbol{\alpha}}(i) + s_{\alpha}^{m}(i)\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i). \tag{45}$$

Squaring the Euclidean norms on both sides of (45), we obtain

$$||\hat{\boldsymbol{\alpha}}(i+1)||^2 = ||\hat{\boldsymbol{\alpha}}(i)||^2 + \Delta_{\boldsymbol{\alpha}}(i), \tag{46}$$

where

$$\begin{aligned}\Delta_{\boldsymbol{\alpha}}(i) = {} & 2s_{\alpha}^{m}(i)\,\hat{\boldsymbol{\alpha}}(i)^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\,\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i) \\ & + (s_{\alpha}^{m}(i))^2\left\|\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\,\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i)\right\|^2.\end{aligned} \tag{47}$$

Substituting (27) into (47), we have

$$\begin{aligned}\Delta_{\boldsymbol{\alpha}}(i) \approx {} & 2s_{\alpha}^{m}(i)\,\hat{\boldsymbol{\alpha}}(i)^T(\mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T \\ & + r_{\alpha}^{m}(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1))\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i) \\ & + (s_{\alpha}^{m}(i))^2\left\|\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\,\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i)\right\|^2 \\ = {} & s_{\alpha}^{m}(i)\,(2\hat{\boldsymbol{\alpha}}(i)^T(\mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T \\ & + r_{\alpha}^{m}(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1))\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i) \\ & + s_{\alpha}^{m}(i)\left\|\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\,\boldsymbol{\Lambda}(i)\,\boldsymbol{e}(i)\right\|^2) \\ = {} & s_{\alpha}^{m}\left(\boldsymbol{\Gamma}_{\alpha}^{1}(i) + \boldsymbol{\Gamma}_{\alpha}^{2}(i) + \boldsymbol{\Gamma}_{\alpha}^{3}(i)\right),\end{aligned} \tag{48}$$

where $\boldsymbol{\Gamma}_{\alpha}^{1}(i)$, $\boldsymbol{\Gamma}_{\alpha}^{2}(i)$, and $\boldsymbol{\Gamma}_{\alpha}^{3}(i)$ are shown as follows:

$$\begin{cases}\boldsymbol{\Gamma}_{\alpha}^{1}(i) = s_{\alpha}^{m}(i)\left\|\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)\right\|^2 \\ \boldsymbol{\Gamma}_{\alpha}^{2}(i) = 2r_{\alpha}^{m}(i)\hat{\boldsymbol{\alpha}}(i)^T(\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i))\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ \boldsymbol{\Gamma}_{\alpha}^{3}(i) = 2\hat{\boldsymbol{\alpha}}(i)^T(\mathbf{K}(i) + \boldsymbol{b}(i)\hat{d}(i-1)^T)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i).\end{cases} \tag{49}$$

According to (45), we expand $\boldsymbol{\Gamma}_{\alpha}^{2}$ as

$$\begin{aligned}\boldsymbol{\Gamma}_{\alpha}^{2}(i) = {} & 2r_{\alpha}^{m}(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)(\hat{\boldsymbol{\alpha}}(i-1) + s_{\alpha}^{m}(i-1) \\ & \times\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1))^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ = {} & 2r_{\alpha}^{m}(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)(\hat{\boldsymbol{\alpha}}(i-1)^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1) \\ & + s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1))\mu^{m}(i),\end{aligned} \tag{50}$$

where $\epsilon_{\boldsymbol{\alpha}}^{m}(i) = \left\|\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^{m}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)\right\|^2/\rho_{\boldsymbol{\alpha}}^{m}(i)$ and $\mu^{m}(i) = \boldsymbol{L}^{m}(i-1)^T(\boldsymbol{L}^{m}(i-1)\boldsymbol{L}^{m}(i-1)^T)^{-1}\boldsymbol{L}^{m}(i)$ with $\boldsymbol{L}^{m}(i-1) = \boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1)$.

Since $s_{\boldsymbol{\alpha}}^{m}(i) = z_{\boldsymbol{\alpha}}^{m}(i)/\rho_{\boldsymbol{\alpha}}^{m}(i)$ holds in (24) and (42), we have

$$\begin{aligned}(s_{\alpha}^{m}(i-1))^2 &= z_{\alpha}^{m}(i-1)/\left(\rho_{\alpha}^{m}(i-1)\right)^2 \\ &= s_{\alpha}^{m}(i-1)/\rho_{\alpha}^{m}(i-1).\end{aligned} \tag{51}$$

On the condition of $s_{\alpha}^{m}(i-1) \neq 0$, the following equality holds

$$s_{\alpha}^{m}(i-1) = 1/\rho_{\alpha}^{m}(i-1). \tag{52}$$

Therefore, we simplify (50) as

$$\begin{aligned}&\boldsymbol{\Gamma}_{\alpha}^{2}(i) \\ &= 2r_{\alpha}^{m}(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^{m}(i)\rho_{\alpha}^{m}(i-1) \\ &\quad\times[s_{\alpha}^{m}(i-1)\hat{\boldsymbol{\alpha}}(i-1)^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1) \\ &\quad+(s_{\alpha}^{m}(i-1))^2\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1)] \\ &= \ell(i)\rho_{\alpha}^{m}(i-1)\,[2s_{\alpha}^{m}(i-1)\hat{\boldsymbol{\alpha}}(i-1)^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i-1) \\ &\quad\times\boldsymbol{e}(i-1) + 2(s_{\alpha}^{m}(i-1))^2\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1)].\end{aligned} \tag{53}$$

According to (52), we have $s_{\alpha}^{m}(i-1)\,\rho_{\alpha}^{m}(i-1) = 1$. Therefore, (53) can be simplified by

$$\begin{aligned}\boldsymbol{\Gamma}_{\alpha}^{2}(i) = {} & \ell(i)\rho_{\alpha}^{m}(i-1)\,[2s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1) \\ & + 2s_{\alpha}^{m}(i-1)\hat{\boldsymbol{\alpha}}(i-1)^T\boldsymbol{\Upsilon}_{\alpha}^{m}(i-1)\boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1)] \\ = {} & \ell(i)[\rho_{\alpha}^{m}(i-1)\,\Delta_{\alpha}(i-1) \\ & + s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1)],\end{aligned} \tag{54}$$

where the scalar $\ell(i)$ takes the form of

$$\ell(i) = \begin{cases}|\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^{m}(i)|, & \text{if} \quad |\mu^{m}(i)| < \dfrac{1}{\eta^{m} + |\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)|} \\ 0, & \text{else.}\end{cases} \tag{55}$$

Since $s_{\boldsymbol{\alpha}}^{m}(i) = z_{\boldsymbol{\alpha}}^{m}(i)/\rho_{\boldsymbol{\alpha}}^{m}(i)$ holds in (24) and (42), $s_{\boldsymbol{\alpha}}^{m}(i) = 0$ can be obtained only when $z_{\boldsymbol{\alpha}}^{m}(i) = 0$, generating $r_{\boldsymbol{\alpha}}^{m}(i) = 0$ which is given in (34). Therefore, $\boldsymbol{\Gamma}_{\alpha}^{2}(i) = 0$ can be derived using (49). Similarly, considering $\Delta_{\alpha}(i-1)$ as a function of $s_{\alpha}^{m}(i-1)$ shown in (48), we have $\Delta_{\alpha}(i-1) = 0$ on the condition of $s_{\boldsymbol{\alpha}}^{m}(i-1) = 0$, which results in $\boldsymbol{\Gamma}_{\alpha}^{2}(i) = 0$ in (54). Therefore, (54) holds regardless of whether $s_{\alpha}^{m}(i-1)$ equals 0.

In addition, based on (23), $\boldsymbol{\Gamma}_{\alpha}^{3}(i)$ can be derived as

$$\boldsymbol{\Gamma}_{\alpha}^{3}(i) = 2(\boldsymbol{\theta}_{\alpha}(i) - \boldsymbol{e}(i))^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i). \tag{56}$$

Substituting (49), (54), and (56) into (48), we rewrite (48) as

$$\begin{aligned}\Delta_{\boldsymbol{\alpha}}(i) \approx {} & s_{\alpha}^{m}(i)(s_{\alpha}^{m}(i)\left\|\boldsymbol{\Upsilon}_{\alpha}^{m}(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)\right\|^2 \\ & + \ell(i)(\rho_{\alpha}^{m}(i-1)\Delta_{\boldsymbol{\alpha}}(i-1) \\ & + s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1)) \\ & + 2(\boldsymbol{\theta}_{\alpha}(i) - \boldsymbol{e}(i))^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)) \\ = {} & s_{\alpha}^{m}(i)(s_{\alpha}^{m}(i)\epsilon_{\alpha}^{m}(i)\rho_{\alpha}^{m}(i) + \ell(i)s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1) \\ & \times\rho_{\alpha}^{m}(i-1) + 2(\boldsymbol{\theta}_{\alpha}(i) - \boldsymbol{e}(i))^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)) \\ & + \ell(i)s_{\alpha}^{m}(i)(\rho_{\alpha}^{m}(i-1)\Delta_{\boldsymbol{\alpha}}(i-1)).\end{aligned} \tag{57}$$

Therefore, (57) can be rewritten as

$$\Delta_{\boldsymbol{\alpha}}(i) \approx \chi(i) + \ell\,(i)\,s_{\alpha}^{m}\,(i)\,\rho_{\alpha}^{m}\,(i-1)\,\Delta_{\boldsymbol{\alpha}}(i-1), \tag{58}$$

where

$$\begin{aligned}\chi(i) = {} & s_{\alpha}^{m}(i)(s_{\alpha}^{m}(i)\epsilon_{\alpha}^{m}(i)\rho_{\alpha}^{m}(i) + 2(\boldsymbol{\theta}_{\alpha}(i) - \boldsymbol{e}(i))^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ & + \ell(i)s_{\alpha}^{m}(i-1)\epsilon_{\alpha}^{m}(i-1)\rho_{\alpha}^{m}(i-1)) \\ \overset{(c_1)}{<} {} & s_{\alpha}^{m}(i)(\epsilon_{\alpha}^{m}(i) + 2(\bar{\boldsymbol{\theta}}_{\alpha}^{m} - \boldsymbol{e}(i))^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ & + \epsilon_{\alpha}^{m}(i-1)) \\ = {} & s_{\alpha}^{m}(i)(\bar{\epsilon}_{\boldsymbol{\alpha}}^{m}(i) - 2(\boldsymbol{e}(i) - \bar{\boldsymbol{\theta}}_{\alpha}^{m})^T\boldsymbol{\Lambda}(i)\boldsymbol{e}(i)) \\ \overset{(c_2)}{\leq} {} & 0.\end{aligned} \tag{59}$$

Here (c_1) follows from the inequality $0 \leq \ell(i) < 1$ and $0 \leq s_\alpha^m(i) \leq 1/\rho_\alpha^m(i)$; (c_2) is obtained by combining (32).

Since (58) and $\Delta_{\boldsymbol{\alpha}}(1) \leq 0$, $\Delta_{\boldsymbol{\alpha}}(i) \leq 0$ in (58) can be derived, which gives $\left\|\hat{\boldsymbol{\alpha}}\,(i+1)\right\|^2 \leq \left\|\hat{\boldsymbol{\alpha}}\,(i)\right\|^2$.

Thus, $\lim_{i\to\infty} \left\|\hat{\boldsymbol{\alpha}}\,(i)\right\|^2 = 0$ and $\lim_{i\to\infty} \Delta_{\boldsymbol{\alpha}}(i) = 0$ can be obtained, which completes the convergence proof FFC in *Theorem 1*. Similarly, the convergence of FBC can also be proved using $\lim_{i\to\infty} \left\|\hat{\boldsymbol{b}}\,(i)\right\|^2 = 0$ and $\lim_{i\to\infty} \Delta_{\boldsymbol{b}}(i) = 0$.

The proposed KRMC-MF is proved to be convergent based on the designed learning parameters (32), (33), and (34). Since $\boldsymbol{\alpha}_*$ is unavailable in practice, we define $W_1^{\boldsymbol{\alpha}}(i) = \|\boldsymbol{\alpha}\,(i+1)\|^2 - \|\boldsymbol{\alpha}\,(i)\|^2$ and $W_2^{\boldsymbol{\alpha}}(i) = \boldsymbol{\alpha}\,(i+1) - \boldsymbol{\alpha}\,(i)$ to rewrite $\Delta_{\boldsymbol{\alpha}}(i)$ as

$$\begin{aligned}
|\Delta_{\boldsymbol{\alpha}}(i)| &= \left|\left\|\hat{\boldsymbol{\alpha}}(i+1)\right\|^2 - \left\|\hat{\boldsymbol{\alpha}}(i)\right\|^2\right| \\
&= \left|\|\boldsymbol{\alpha}\,(i+1)\|^2 - \|\boldsymbol{\alpha}\,(i)\|^2 - 2(\boldsymbol{\alpha}\,(i+1) - \boldsymbol{\alpha}\,(i))^T \boldsymbol{\alpha}_*\right| \\
&\leq \left|W_1^{\boldsymbol{\alpha}}(i)\right| + 2\left|(\boldsymbol{\alpha}\,(i+1) - \boldsymbol{\alpha}\,(i))^T \boldsymbol{\alpha}_*\right| \\
&\leq \left|W_1^{\boldsymbol{\alpha}}(i)\right| + 2|W_2^{\boldsymbol{\alpha}}(i)|g_1 \qquad (60)
\end{aligned}$$

with $g_1 = \|\boldsymbol{\alpha}_*\|$. Similarly, we have

$$|\Delta_{\boldsymbol{b}}(i)| \leq \left|W_1^{\boldsymbol{b}}(i)\right| + 2|W_2^{\boldsymbol{b}}(i)|g_2, \qquad (61)$$

where $W_1^{\boldsymbol{b}}(i) = \|\boldsymbol{b}\,(i+1)\|^2 - \|\boldsymbol{b}\,(i)\|^2$, $W_2^{\boldsymbol{b}}(i) = \boldsymbol{b}\,(i+1) - \boldsymbol{b}\,(i)$, and $g_2 = \|\boldsymbol{b}_*\|$

To observe the convergence of KRMC-MF, we define the convergence curves of $\tau_\alpha(i)$ with $\tau_\alpha\,(i) = \left|W_1^{\boldsymbol{\alpha}}(i)\right| + 2|W_2^{\boldsymbol{\alpha}}(i)|$ and $\tau_b\,(i) = \left|W_1^{\boldsymbol{b}}(i)\right| + 2|W_2^{\boldsymbol{b}}(i)|$, which can avoid the direct calculation of $\boldsymbol{\alpha}_*$. The convergence curves of $\tau_\alpha(i)$ and $\tau_b\,(i)$ converge to zero, leading to $\lim_{i\to\infty} |\Delta_{\boldsymbol{\alpha}}(i)| = 0$ and $\lim_{i\to\infty} |\Delta_{\boldsymbol{b}}(i)| = 0$. Therefore, in following simulations, $\tau_\alpha(i)$ and $\tau_b\,(i)$ are used to prove the convergence of the proposed filters.

Since LRKOL-MCC can be regarded as a simplified version of KRMC-MF, we have the following corollary.

Corollary 1: Based on the same conditions as those in *Theorem 1*, the convergence characteristic of LRKOL-MCC is given by $\lim_{i\to\infty} |\Delta_{\boldsymbol{\alpha}}(i)| = 0$ and $\lim_{i\to\infty} |\Delta_{\boldsymbol{b}}(i)| = 0$.

VII. SIMULATION RESULTS

In this section, the Gaussian and non-Gaussian noise environments are considered to validate the filtering performance of the proposed LRKOL-MCC and KRMC-MF in the context of nonlinear time-series prediction and nonlinear regression. In the following simulations, LRKOL [11], KRLS [9], KRMC [37], and KRLS-MF [14] are chosen for comparison, where LRKOL has a filter feedback structure based on MMSE; KRLS and KRLS-MF are KAFs based on least squares without and with feedback, respectively; KRMC is the extension of KRLS based on MCC. In the non-Gaussian case, the alpha stable distribution is chosen as the disturbance noise with the following characteristic function [45], [46]:

$$f\,(t) = \exp\left\{j\delta t - \gamma|t|^\sigma\,[1 + j\beta sgn\,(t)\,S\,(t,\sigma)]\right\}, \qquad (62)$$

where the parameter set $V = (\sigma, \beta, \gamma, \delta)$ includes the characteristic factor $\sigma \in (0, 2]$, $\beta \in (-1, 1)$ measuring asymmetry, the dispersion parameter $\gamma > 0$, and the location parameter $\delta \in (-\infty, \infty)$;

$$S\,(t,\sigma) = \begin{cases} \tan\dfrac{\sigma\pi}{2}, & \sigma \neq 1 \\ \dfrac{2}{\pi}\log|t|, & \sigma = 1. \end{cases} \qquad (63)$$

In the following simulations, a segment of 1000 samples is used as the training data and another 200 samples as the testing data. To remove the stochastic nature of simulations, all simulation results are averaged 100 independent Monter Carlo runs. The notations $\boldsymbol{p}^m = [\vartheta_\alpha^m, \vartheta_b^m, \bar{\varrho}_\alpha^m, \bar{\varrho}_b^m, \eta^m]$, $\boldsymbol{p}^s = [\vartheta_\alpha^s, \vartheta_b^s, \bar{\varrho}_\alpha^s, \bar{\varrho}_b^s, \eta^s]$, $\mathbf{q}^m = [\bar{\boldsymbol{\theta}}_\alpha^m, \bar{\boldsymbol{\theta}}_b^m]$, and $\boldsymbol{q}^s = [\bar{\theta}_\alpha^s, \bar{\theta}_b^s]$ are adopted for clarity. Furthermore, we also introduce the collection of regularization factors $\boldsymbol{\lambda} = [\lambda_1, \lambda_2]$ with λ_1 and λ_2 in (12) and (17), respectively. In the following, all these parameters are set by trials to obtain the best filtering performance.

In order to evaluate the filtering accuracy, the mean square error (MSE) is defined by

$$\text{MSE} = \frac{1}{N}\sum_{j=1}^{N} (d(j) - \hat{d}(j))^2, \qquad (64)$$

where $\hat{d}(j)$ is the estimate of the desired output $d(j)$ and N is the data length.

A. MACKEY-GLASS TIME-SERIES PREDICTION

The Mackey-Glass (MG) chaotic time series, which presents periodic and chaotic dynamics, is generated by the following nonlinear time delay differential equation [17]:

$$\frac{dx(t)}{dt} = -0.1x(t) + \frac{0.2x(t-\tau)}{1 + x(t-\tau)^{10}}, \qquad (65)$$

where $\tau = 30$. The time series is first discretized using a sampling period of 6 seconds. The input $\boldsymbol{u}(i) = [x(i-7), x(i-6), \ldots, x(i-2), x(i-1)]$ is chosen to predict the current $x(i)$ that is the desired output $d(i)$.

1) GAUSSIAN ENVIRONMENT

In the Gaussian environment, the data set is corrupted by the Gaussian noise with zero mean and variance 0.013. For the compared filters based on MCC, i.e., LRKOL-MCC, KRMC, and KRMC-MF, we have $h_1 = 0.17$ in (2) for LRKOL-MCC and $h_1 = 0.8$ for KRMC and KRMC-MF. The kernel size h_2 in (16) is chosen as 0.85. The collection $\boldsymbol{\lambda} = [0.001, 0.001]$ is chosen in simulations. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, $\boldsymbol{p}^m = \boldsymbol{p}^s = [0.65, 0.93, 0.1, 150, 0.01]$ is configured. In this simulation, $\mathbf{q}^m = [\boldsymbol{0.0096}, \boldsymbol{0.0096}]$ gives that $\bar{\boldsymbol{\theta}}_m^\alpha$ or $\bar{\boldsymbol{\theta}}_m^b$ is a column vector with each entry being 0.0096 in KRMC-MF but a scalar being equal to 0.0096 in LRKOL [11] and LRKOL-MCC,

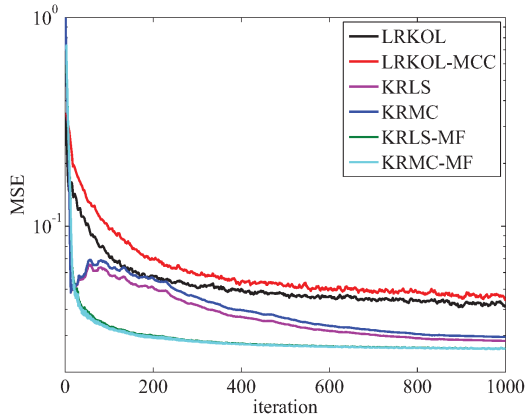


FIGURE 2. Learning curves of KAFs in MG time series prediction under the Gaussian noises.

i.e., $\boldsymbol{q}^s = [0.0096, 0.0096]$. The learning curves of several KAFs in the Gaussian environment are shown in Fig. 2. It can be seen from Fig. 2 that KRLS based on MMSE has a lower steady-state MSE than KRMC based on MCC [37]. However, the proposed LRKOL-MCC and KRMC-MF achieve almost the same estimation accuracy as LRKOL and KRLS-MF, respectively. Among the compared filters, the proposed KRMC-MF generates the highest filtering accuracy.

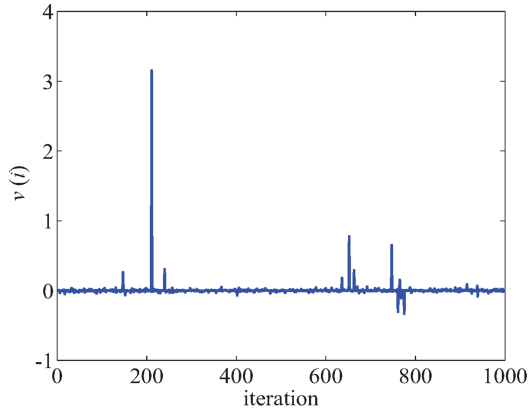


FIGURE 3. A sequence of the alpha-stable noise with $\boldsymbol{V} = (1.4, 0, 0.02, 0)$ in MG time series prediction.

2) NON-GAUSSIAN ENVIRONMENT

In this simulation, the alpha stable noise $v(i)$ is shown in Fig. 3, with parameters set as $V = (1.4, 0, 0.02, 0)$. The parameters h_1 and h_2 are configured as 1.2 and 0.23, respectively. The regularization factor $\boldsymbol{\lambda} = [0.95, 0.001]$ is used in the non-Gaussian environment. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, the parameters are configured as $\boldsymbol{p}^m = \boldsymbol{p}^s = [0.99, 0.93, 0.5, 150, 0.0001]$, $\mathbf{q}^m = [\boldsymbol{0.02}, \boldsymbol{0.02}]$ and $\boldsymbol{q}^s = [0.02, 0.02]$. The learning curves of KAFs in the non-Gaussian environment are shown in Fig. 4. It is seen clearly from this figure that LRKOL-MCC outperforms LRKOL, and KRMC-MF achieves the best filtering performance in terms of steady-state MSE and robustness.

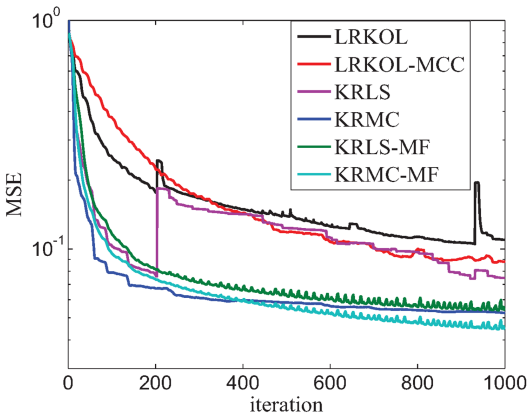


FIGURE 4. Learning curves of KAFs in MG time series prediction under the alpha-stable noises.

B. NONLINEAR REGRESSION

We consider a nonlinear dynamical data set, which is derived from the following nonlinear difference equation [11]:

$$x(i) = x(i-1)(0.8 - 0.5\exp(-x^2(i-1))) - (0.3 + 0.9\exp(-x^2(i-1)))x(i-2) + 0.1\sin(x(i-1)\pi), \quad (66)$$

where $x(i)$ is the output at discrete time i. The data are generated with the initiation conditions: $x(-1) = 0.1$ and $x(-2) = 0.1$. The latest two previous outputs $\boldsymbol{u}(i) = [x(i-1), x(i-2)]$ are used as the input to estimate the current output $x(i)$.

1) GAUSSIAN ENVIRONMENT

In the Gaussian environment, the data are corrupted by the Gaussian noises with zero mean and variance 0.013. The kernel parameter h_1 is set as 0.35 for LRKOL-MCC, and $h_1 = 0.75$ for KRMC and KRMC-MF. The parameters h_2 and $\boldsymbol{\lambda}$ are configured as $h_2 = 0.23$ and $\boldsymbol{\lambda} = [0.001, 0.001]$. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, we choose $\boldsymbol{p}^m = \boldsymbol{p}^s = [0.88, 0.999, 0.1, 150, 0.01]$, $\mathbf{q}^m = [\boldsymbol{0.015}, \boldsymbol{0.015}]$, and $\boldsymbol{q}^s = [0.015, 0.015]$. The comparison of KAFs under Gaussian noises is depicted in Fig. 5. To perform the comparison of computational complexity, Table 2 presents the mean consumed time of different filters in nonlinear regression under Gaussian noises. Combining Table 2 and Fig. 5, we see that proposed KRMC-MF with similar computational complexity achieves higher filtering accuracy than KRMC. Similarly, compared with LRKOL, LRKOL-MCC can also obtain better filtering accuracy without increasing significant computational burden. In addition, KRMC-MF achieves almost the same filtering performance as KRLS-MF under Gaussian noises in terms of the steady-state MSE and computational cost.

Since the dimensions of FFC and FBC in (28) and (29) (or (38) and (40)) increase with each new sample, the newly added dimension is not included in the computations of $\Delta_{\boldsymbol{\alpha}}(i)$ or $\Delta_{\boldsymbol{b}}(i)$. To validate the convergence of KRMC-MF

TABLE 2. Comparison of mean consumed time in nonlinear regression under gaussian noises.

Algorithm	LRKOL	LRKOL-MCC	KRLS	KRMC	KRLS-MF	KRMC-MF
Mean consumed time (s)	6.33	6.41	17.76	17.80	27.70	30.43

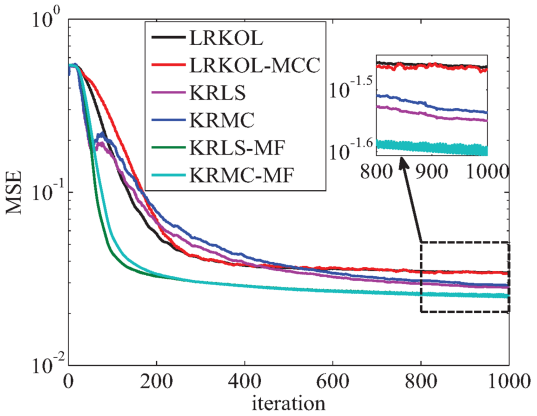


FIGURE 5. Learning curves of KAFs in nonlinear regression under the Gaussian noises.

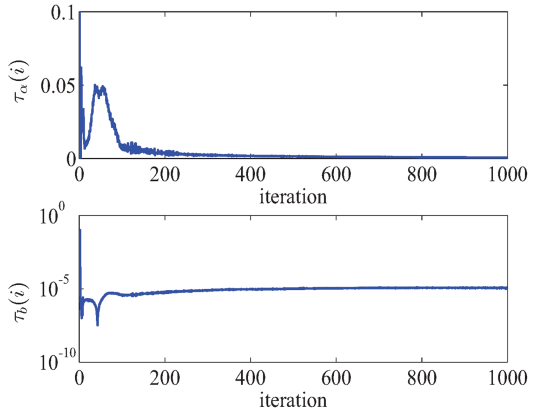


FIGURE 6. Convergence curves of $\tau_{\alpha}(i)$ and $\tau_{b}(i)$ in KRMC-MF under the Gaussian noises.

given in *Theorem* 1, the convergence curves of $\tau_{\alpha}(i)$ and $\tau_{b}(i)$ in the Gaussian noises are shown in Fig. 6. It can be shown in Fig. 6 that $\tau_{\alpha}(i)$ converges to zero, leading to $\lim_{i\to\infty}|\Delta_{\boldsymbol{\alpha}}(i)| = 0$ and $\lim_{i\to\infty}\|\Delta_{\boldsymbol{b}}(i)\|^2 = 0$. According to (61), $\lim_{i\to\infty}|\Delta_{\boldsymbol{\alpha}}(i)| = 0$ is derived, which complies with *Theorem 1*. The similar conclusions can be obtained for FBC, i.e., $\lim_{i\to\infty}|\Delta_{\boldsymbol{b}}(i)| = 0$. Similarly, the convergence of LRKOL-MCC can also be demonstrated by $\tau_{\alpha}(i)$ and $\tau_{b}(i)$, which is shown in Fig. 7. From this figure, we also obtain $\lim_{i\to\infty}|\Delta_{\boldsymbol{\alpha}}(i)| = 0$ and $\lim_{i\to\infty}|\Delta_{\boldsymbol{b}}(i)| = 0$ in LRKOL-MCC.

2) NON-GAUSSIAN ENVIRONMENT

In the non-Gaussian environment, the alpha stable noise $v(i)$ is configured as $V = (1.32, 0, 0.008, 0)$. To study the effect of the kernel parameters h_1 and h_2 on filtering accuracy in KRMC-MF, the steady-state MSEs versus

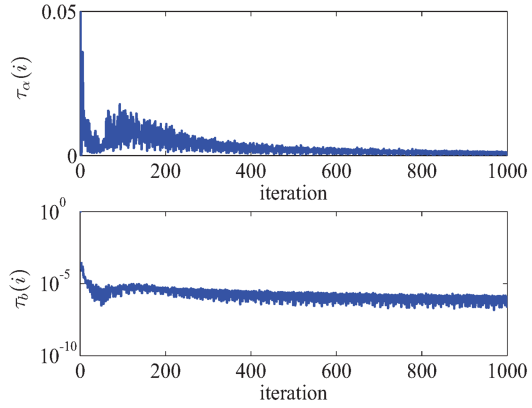


FIGURE 7. Convergence curves of $\tau_{\alpha}(i)$ and $\tau_{b}(i)$ in LRKOL-MCC under the Gaussian noises.

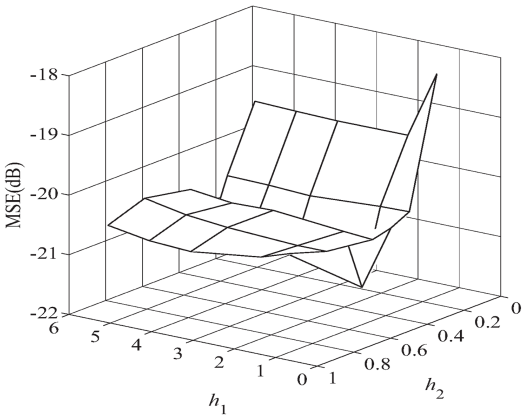


FIGURE 8. Steady-state MSE versus h_1 and h_2 in nonlinear regression under the alpha-stable noises.

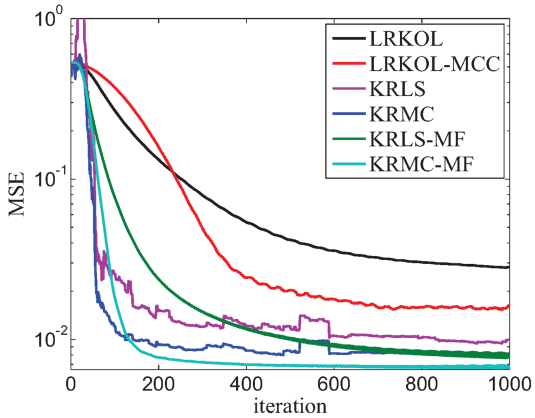


FIGURE 9. Learning curves of KAFs in nonlinear regression under the alpha-stable noises.

the kernel size $h_1 \in [1.1, 1.8, 3.5, 4.5, 5.5]$ and $h_2 \in [0.05, 0.1, 0.35, 0.7, 0.9]$ are shown in Fig. 8. We see from this figure that the steady-state MSE reaches a bottom at

$h_1 = 1.8$ and $h_2 = 0.35$. Therefore, we set $h_1 = 1.8$ and $h_2 = 0.35$ in KRLS, KRMC, and KRMC-MF. As for LRKOL and LRKOL-MCC, the same methods for parameters selection as that in Fig. 8 are used, and we set $h_1 = 3$ and $h_2 = 0.35$ thereby. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, $\boldsymbol{p}^m = \boldsymbol{p}^s = [0.985, 0.93, 0.8, 150, 0.0001]$, $\mathbf{q}^m = [\boldsymbol{0.2}, \boldsymbol{0.2}]$ and $\boldsymbol{q}^s = [0.2, 0.2]$ are configured. Fig. 9 shows the learning curves of KAFs in the alpha stable noise. From Fig. 9, we see that the proposed LRKOL-MCC and KRMC-MF achieve higher filtering accuracy than other KAFs, and KRMC-MF obtains the lowest steady-state MSE, which validates the efficiency of LRKOL-MCC and KRMC-MF.

VIII. CONCLUSION

This paper introduces novel kernel adaptive filters with feedback based on the maximum correntropy criterion, i.e., KRMC-MF and its simplification LRKOL-MCC. In KRMC-MF, a novel feedback structure with multiple single-delay outputs is incorporated into KRMC to improve the filtering accuracy at the expense of increasing computational burden. Compared with KRLS-MF adopting the least squared errors as the cost function, KRMC-MF based on the maximum correntropy criterion is robust against outliers. As a simplified version of KRMC-MF, LRKOL-MCC only takes the current error as the feedback structure to achieve an acceptable filtering accuracy, resulting in reduction of computational cost. LRKOL-MCC improves the filtering accuracy and robustness of LRKOL. In addition, the convergence of LRKOL-MCC and KRMC-MF is established theoretically for guaranteeing their stabilities, which is also validated by simulations. Simulations on time-series prediction and nonlinear regression show the superior filtering performance of the proposed LRKOL-MCC and KRMC-MF under the non-Gaussian noises. Generally, MCC is designed to combat the non-Gaussian noise, which may be not applicable for the Gaussian noise. Therefore, it is also interesting to note that the filtering performance of LRKOL-MCC and KRMC-MF can approach that of LRKOL and KRLS-MF under the Gaussian noises, respectively.

APPENDIX

The update of MLMS is denoted by [43]

$$\mathbf{w}(i+1) = \mathbf{w}(i) + \eta \frac{\partial J(i)}{\partial \mathbf{w}(i)} + \tau(\mathbf{w}(i) - \mathbf{w}(i-1)), \quad (67)$$

where $\mathbf{w}(i+1)$ is the estimated weight in MLMS; η is the learning rate; $(\mathbf{w}(i) - \mathbf{w}(i-1))$ is the momentum term and $|\tau| < 1$ is the corresponding coefficient. The convergence rate is accelerated by $\tau > 0$ and the filtering accuracy is improved by $\tau < 0$.

Theorem 2: The weigh updates in KRMC-MF and LRKOL-MCC have the similar form to that in (67).

Proof: In KRMC-MF, according to (28), we have

$$\boldsymbol{\alpha}(i) = \boldsymbol{\alpha}(i-1) + s_{\boldsymbol{\alpha}}^m(i-1)\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^m(i-1)\boldsymbol{\Lambda}(i-1)\boldsymbol{e}(i-1). \quad (68)$$

From (68), we obtain

$$\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^m(i-1) = \frac{\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)}{s_{\boldsymbol{\alpha}}^m(i-1)} \boldsymbol{L}^m(i-1)^T \times (\boldsymbol{L}^m(i-1)\boldsymbol{L}^m(i-1)^T)^{-1}, \quad (69)$$

where $\boldsymbol{L}^m(i-1)$ is defined in (35). Substituting (69) into (27) gives

$$\begin{aligned}\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^m(i) &= \mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T + r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^m(i-1) \\ &= \mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T \\ &\quad + \frac{r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)[\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)]}{s_{\boldsymbol{\alpha}}^m(i-1)} \boldsymbol{L}^m(i-1)^T \\ &\quad \times (\boldsymbol{L}^m(i-1)\boldsymbol{L}^m(i-1)^T)^{-1}, \end{aligned} \quad (70)$$

Substituting (70) into (28) yields

$$\begin{aligned}\boldsymbol{\alpha}(i+1) &= \boldsymbol{\alpha}(i) + s_{\boldsymbol{\alpha}}^m(i)\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}^m(i)\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ &= \boldsymbol{\alpha}(i) + s_{\boldsymbol{\alpha}}^m(i)[\mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T]\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ &\quad + [\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)] \times \frac{s_{\boldsymbol{\alpha}}^m(i)r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)}{s_{\boldsymbol{\alpha}}^m(i-1)} \\ &\quad \times \boldsymbol{L}^m(i-1)^T(\boldsymbol{L}^m(i-1)\boldsymbol{L}^m(i-1)^T)^{-1}\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ &= \boldsymbol{\alpha}(i) + s_{\boldsymbol{\alpha}}^m(i)[\mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T]\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ &\quad + [\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)] \times \frac{s_{\boldsymbol{\alpha}}^m(i)r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^m(i)}{s_{\boldsymbol{\alpha}}^m(i-1)} \\ &= \boldsymbol{\alpha}(i) + s_{\boldsymbol{\alpha}}^m(i)[\mathbf{K}(i) + \boldsymbol{b}(i)\hat{\boldsymbol{d}}(i-1)^T]\boldsymbol{\Lambda}(i)\boldsymbol{e}(i) \\ &\quad + [\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1)]\bar{\boldsymbol{\Upsilon}}_{\boldsymbol{\alpha}}(i), \end{aligned} \quad (71)$$

where $(\boldsymbol{\alpha}(i) - \boldsymbol{\alpha}(i-1))$ can be regarded as a momentum term, $\bar{\boldsymbol{\Upsilon}}_{\boldsymbol{\alpha}}(i) = \frac{s_{\boldsymbol{\alpha}}^m(i)r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^m(i)}{s_{\boldsymbol{\alpha}}^m(i-1)}$ is the corresponding momentum coefficient, and $\mu^m(i)$ is defined in (35). When KRMC-MF converges, we obtain

$$\lim_{i\to\infty} \frac{s_{\boldsymbol{\alpha}}^m(i)}{s_{\boldsymbol{\alpha}}^m(i-1)} \approx 1. \quad (72)$$

Based on (34) and (72), it holds

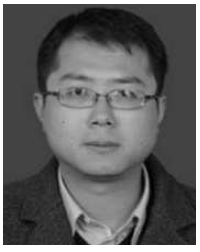
$$\begin{aligned}\bar{\boldsymbol{\Upsilon}}_{\boldsymbol{\alpha}}(i) &= \frac{s_{\boldsymbol{\alpha}}^m(i)r_{\boldsymbol{\alpha}}^m(i)\boldsymbol{b}(i)^T\boldsymbol{\alpha}(i)\mu^m(i)}{s_{\boldsymbol{\alpha}}^m(i-1)} \\ &\leq \left|\frac{s_{\boldsymbol{\alpha}}^m(i)}{s_{\boldsymbol{\alpha}}^m(i-1)}\right| \\ &\approx 1 \end{aligned} \quad (73)$$

Therefore, the update in (28) is similar to that in (67). This proof can be extended to the update in (29). Similarly, we obtain the similar proofs regarding the weight update in LRKOL-MCC.

REFERENCES

[1] G. Camps-Valls and L. Bruzzone, *Kernel Methods for Remote Sensing Data Analysis*. New York, NY, USA: Wiley, 2009.
[2] P. P. Pokharel, W. Liu, and J. C. Príncipe, "Kernel least mean square algorithm with constrained growth," *Signal Process.*, vol. 89, no. 3, pp. 257–265, Mar. 2009.
[3] M. A. Takizawa and M. Yukawa, "Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4257–4269, Aug. 2015.

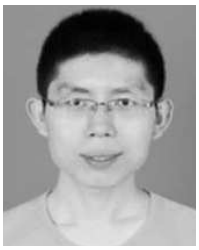
[4] M. Signoretto, L. De Lathauwer, and J. A. K. Suykens, "A kernel-based framework to tensorial data analysis," *Neural Netw.*, vol. 24, no. 8, pp. 861–874, Oct. 2011.
[5] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Mean square convergence analysis of the kernel least mean square algorithm," *Signal Process.*, vol. 92, no. 11, pp. 2624–2632, Nov. 2012.
[6] S. Wang, J. Feng, and C. K. Tse, "Kernel affine projection sign algorithms for combating impulse interference," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 11, pp. 811–815, Nov. 2013.
[7] W. Liu, P. P. Pokharel, and J. C. Príncipe, "The kernel least-mean square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Feb. 2009.
[8] W. Liu and J. C. Príncipe, "Kernel affine projection algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–12, 2008, Art. no. 784292.
[9] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.
[10] J. Feng, C. K. Tse, and F. C. M. Lau, "A neural-network-based channel-equalization strategy for chaos-based communication systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 7, pp. 954–957, Jul. 2003.
[11] H. Fan and Q. Song, "A linear recurrent kernel online learning algorithm with sparse updates," *Neural Netw.*, vol. 50, pp. 142–153, Feb. 2014.
[12] J. Zhao, X. Liao, S. Wang, and C. K. Tse, "Kernel least mean square with single feedback," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 953–957, Jul. 2015.
[13] S. Wang, Y. Zheng, and C. Ling, "Regularized kernel least mean square algorithm with multiple-delay feedback," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 98–101, Jan. 2016.
[14] S. Wang, W. Wang, and S. Duan, "Kernel recursive least squares with multiple feedback and its convergence analysis," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 10, pp. 1237–1241, Oct. 2017.
[15] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 57–67, Jan. 2005.
[16] J. Platt, "A resource-allocating network for function interpolation," *Neural Comput.*, vol. 3, no. 2, pp. 213–225, 1991.
[17] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. New York, NY, USA: Wiley, 2010.
[18] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 22–32, Jan. 2012.
[19] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel recursive least squares algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1484–1491, Sep. 2013.
[20] L. Cheng, M. F. Ren, and G. Xie, "Multipath estimation based on centered error entropy criterion for non-Gaussian noise," *IEEE Access*, vol. 4, pp. 9978–9986, 2016.
[21] Y. Zhang *et al.*, "Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation," *IEEE Access*, vol. 4, pp. 8375–8385, Nov. 2016.
[22] X. Li, Y. Hu, E. Zio, and R. Kang, "A Bayesian optimal design for accelerated degradation testing based on the inverse Gaussian process," *IEEE Access*, vol. 5, pp. 5690–5701, Mar. 2017.
[23] J. C. Príncipe, D. Xu, and J. Fisher, "Information theoretic learning: Renyi's entropy and kernel perspectives," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. New York, NY, USA: Wiley, 2000.
[24] Z. Wu, S. Peng, W. Ma, B. Chen, and J. C. Príncipe, "Minimum error entropy algorithm with sparsity penalty constraints," *Entropy*, vol. 17, no. 5, pp. 3419–3437, May 2015.
[25] B. Chen, L. Xing, J. Liang, N. Zheng, and J. C. Príncipe, "Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion," *IEEE Signal Process. Lett.*, vol. 21, no. 7, pp. 880–884, Jul. 2014.
[26] C. Liu, Y. Qi, and W. Ding, "The data-reusing MCC-based algorithm and its performance analysis," *Chin. J. Electron.*, vol. 25, no. 4, pp. 719–725, Jul. 2016.
[27] L. Lu and H. Zhao. (Jan. 2017). "Subband adaptive filter trained by differential evolution for channel estimation." [Online]. Available: https://arxiv.org/abs/1701.08407
[28] J. J. Wang, Y. J. Wang, B. Y. Jing, and X. Gao, "Regularized maximum correntropy machine," *Neurocomputing*, vol. 160, no. 21, pp. 85–92, Jul. 2015.
[29] F. Huang, J. Zhang, and S. Zhang, "Adaptive filtering under a variable kernel width maximum correntropy criterion," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 10, pp. 1247–1251, Oct. 2017.
[30] X. Chen, J. Yang, J. Liang, and Q. Ye, "Recursive robust least squares support vector regression based on correntropy criterion," *Neurocomputing*, vol. 97, no. 15, pp. 63–73, Nov. 2012.
[31] Y. Feng, X. Huang, L. Shi, Y. Yang, and J. A. K. Suykens, "Learning with the maximum correntropy criterion induced losses for regression," *J. Mach. Learn. Res.*, vol. 16, pp. 993–1034, 2015.
[32] X. T. Yuan and B. G. Hu, "Robust feature extraction via information theoretic learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1193–1200.
[33] H. J. Xing and H. R. Ren, "Regularized correntropy criterion based feature extraction for novelty detection," *Neurocomputing*, vol. 133, no. 10, pp. 483–490, Jun. 2014.
[34] H. J. Xing and X. M. Wang, "Training extreme learning machine via regularized correntropy criterion," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 1977–1986, Dec. 2013.
[35] H. J. Xing and X. Z. Wang, "Selective ensemble of SVDDs with Renyi entropy based diversity measure," *Pattern Recognit.*, vol. 61, pp. 185–196, Jan. 2017.
[36] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic-based iterative minimization for robust sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 261–275, Feb. 2014.
[37] Z. Wu, J. Shi, X. Zhang, W. Ma, and B. Chen, "Kernel recursive maximum correntropy," *Signal Process.*, vol. 117, pp. 11–16, Dec. 2015.
[38] B. Chen, Y. Zhu, J. Hu, and J. C. Príncipe, *System Parameter Identification: Information Criteria and Algorithms*. Amsterdam, The Netherlands: Elsevier, 2013.
[39] Z. Wu, S. Peng, B. Chen, and H. Zhao, "Robust Hammerstein adaptive filtering under maximum correntropy criterion," *Entropy*, vol. 17, no. 10, pp. 7149–7166, 2015.
[40] M. Yukawa, "Multikernel Adaptive Filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4672–4682, Sep. 2012.
[41] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4635–4643, Oct. 2006.
[42] S. Wang *et al.*, "Multiple sclerosis detection based on biorthogonal wavelet transform, RBF kernel principal component analysis, and logistic regression," *IEEE Access*, vol. 4, pp. 7567–7576, 2016.
[43] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.
[44] W. R. Cluett, S. L. Shah, and D. G. Fisher, "Robustness analysis of discrete-time adaptive control systems using input-output stability theory: A tutorial," *IEE Proc. D Control Theory Appl.*, vol. 135, no. 2, pp. 133–141, Mar. 1998.
[45] B. Weng and K. E. Barner, "Nonlinear system identification in impulsive environments," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2588–2594, Jul. 2005.
[46] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, no. 7, pp. 986–1010, Jul. 1993.



SHIYUAN WANG (M'13) received the B.Eng. and M.Eng. degrees in electronic and information engineering from Southwest Normal University, Chongqing, China, in 2002 and 2005, respectively, and the Ph.D. degree in circuit and system from Chongqing University, Chongqing, China, in 2011. He was a Research Associate and a Research Fellow with The Hong Kong Polytechnic University from 2012 to 2013 and from 2017 to 2017, respectively. He is currently an Associate Professor with the College of Electronic and Information Engineering, Southwest University, China. His research interests include adaptive signal processing, nonlinear dynamics and chaos, and bioinformatics. He has authored one book and over 50 papers.

LUJUAN DANG received the B.Eng. degree in information science and technology from Northwest University, Xian, China, in 2015. She is currently pursuing the M.Eng. degree with the College of Electronic and Information Engineering, Southwest University, Chongqing, China. Her current interests focus on kernel adaptive filtering and information theoretic learning.

WANLI WANG received the B.Eng. degree in electronic and information engineering from the Inner Mongolia University of Science and Technology, Inner Mongolia, China, in 2015. He is currently pursuing the M.Eng. degree with the College of Electronic and Information Engineering, Southwest University, Chongqing, China. His interests focus on kernel adaptive filtering and information theoretic learning.



GUOBING QIAN received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China in 2015. He is currently with the College of Electronic and Information Engineering, Southwest University. His current research mainly focuses on adaptive filter, and adaptive signal processing.

CHI K. TSE (M'90–SM'97–F'06) received the B.Eng. degree (Hons.) in electrical engineering and the Ph.D. degree from the University of Melbourne, Australia, in 1987 and 1991, respectively. He served as the Head of the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong, from 2005 to 2012, where he is currently the Chair Professor. His research interests include power electronics, nonlinear systems, and complex network applications. He serves as Panel Member of the Hong Kong Research Grants Council, and a member of several professional and government committees. He has received a number of research and industry awards, including the Prize Paper Award by the IEEE TRANSACTIONS ON POWER ELECTRONICS in 2001 and 2015, respectively, the Best paper Award from the *International Journal of Circuit Theory and Applications* in 2003, two Gold Medals from the International Inventions Exhibition in Geneva in 2009 and 2013, respectively, the *RISP Journal of Signal Processing* Best Paper Award in 2014, the Silver Medal from the International Invention Innovation Competition in Canada in 2016, and a number of recognitions from the academic and research communities, including the honorary professorship by several Chinese and Australian universities, the Chang Jiang Scholar Chair Professorship, the IEEE Distinguished Lectureship, the Distinguished Research Fellowship from the University of Calgary, the Gledden Fellowship and the International Distinguished Professorship-at-Large from the University of Western Australia. While with The Hong Kong Polytechnic University, he received the President's Award for Outstanding Research Performance twice, the Faculty Research Grant Achievement Award twice, the Faculty Best Researcher Award, and several teaching awards. He serves and has served as Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II (2016–2019), *IEEE Circuits and Systems Magazine* (2012–2015), the Editor-in-Chief of the IEEE Circuits and Systems Society Newsletter (since 2007), an Associate Editor for three IEEE journal/transactions, an Editor for the *International Journal of Circuit Theory and Applications*, and is on the editorial boards of a few other journals. He is currently the Chair of the Steering Committee of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.