Kernel Adaptive Filters With Feedback Based on Maximum Correntropy

WANG, Shiyuan; DANG, Lujuan; WANG, Wanli; QIAN, Guobing; TSE, Chi K.

Published in:
IEEE Access

Published: 01/01/2018

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2018.2808218

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 21/08/2023
Kernel Adaptive Filters With Feedback Based on Maximum Correntropy

SHIYUAN WANG1,2, (Member, IEEE), LUJUAN DANG1,2, WANLI WANG1,2, GUOBING QIAN1,2, AND CHI K. TSE3, (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,
Define \( R = X - Y \). Equation (1) is therefore rewritten as

\[
C(R) = \int \kappa_{h_1}(r)f_R(r)\,dr
\]

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.,

\[
\hat{C}(R) = \frac{1}{M} \sum_{j=1}^M \kappa_{h_1}(r_j)
\]

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).

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

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.,

\[
\max \, C(R) = \sum_{j=1}^M \kappa_{h_1}(r_j).
\]

### III. REVIEW 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 \( \{u(j), d(i)\}_{j=1}^M \), where \( u(j) \in \mathbb{R}^n \times 1 \) and \( d(i) \in \mathbb{R} \) are the \( j \)th 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 \( \phi(\cdot) \). Let \( \Psi(i) = [\phi(u(1)), \phi(u(2)), \ldots, \phi(u(j)), \ldots, \phi(u(t))] \) and \( d(i) = [d_1(i), d_2(i), \ldots, d_M(i)]^T \). 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:

\[
\hat{f}_i(\cdot) = \omega(i)^T \phi(\cdot),
\]

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

\[
\omega(i) = \sum_{j=1}^i \alpha_j \phi(u(j)).
\]
Substituting (8) into (7) yields
\[ \hat{f}_i(\cdot) = \sum_{j=1}^{i} \alpha_j \varphi(u(j))^T \varphi(\cdot). \quad (9) \]

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.,
\[ \varphi(u(j))^T \varphi(\cdot) = \kappa(u(j), \cdot). \quad (10) \]

Substituting (10) into (9) yields
\[ \hat{f}_i(\cdot) = \sum_{j=1}^{i} \alpha_j \kappa(u(j), \cdot). \quad (11) \]

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 function \( \omega(i) \) generates different kernel adaptive filters. In KRLS, \( \omega(i) \) is estimated by minimizing the following regularized loss function:
\[ \min_\omega \sum_{j=1}^{i} \left( d_j(i) - \hat{d}_j(i) \right)^2 + \lambda_1 \| \omega(i) \|^2, \quad (12) \]

where \( \lambda_1 \) is the regularization factor for avoiding overfitting [17] and \( \hat{d}_j(i) = \omega(i)^T \varphi(u(j)) \) denotes the \( j \)th estimated output at discrete time \( i \).

The solution to (12) yields
\[ \omega(i) = [ \Psi(i)^T \Psi(i) + \lambda_1 I ]^{-1} \Psi(i)d(i). \quad (13) \]

The matrix inversion lemma is described by [17]
\[ (A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}. \quad (14) \]

Let \( \Lambda = \lambda_1 I, B = \Psi(i), C = I, \) and \( D = \Psi(i)^T \). Thus, we can rewrite the weight update of KRLS in (13) as
\[ \omega(i) = \Psi(i)(K(i) + \lambda_1 I)^{-1}d(i) = \Psi(i)\alpha_s(i). \quad (15) \]

where \( \alpha_s(i) = [ \lambda_1 I + K(i) ]^{-1}d(i) \). The positive definite matrix \( K(i) \) in (15) takes the form of \( K(i) = \Psi(i)^T \Psi(i), \) the \( m_1 \)th row and \( m_2 \)th column element of which can be evaluated by
\[ K_{m_1,m_2}(i) = \kappa_h(u(m_1), u(m_2)) = \exp\left( -\frac{\|u(m_1) - u(m_2)\|^2}{2h_2^2} \right). \quad (16) \]

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
\[ \max_{\omega} \sum_{j=1}^{i} \kappa_h(d_j(i) - \hat{d}_j(i)) - \frac{1}{2} \lambda_2 \| \omega(i) \|^2, \quad (17) \]

where \( \lambda_2 \) denotes the regularization factor of KRMC and the \( j \)th estimated output \( \hat{d}_j(i) \) is given by \( \hat{d}_j(i) = \omega(i)^T \varphi(u(j)) \).

The solution to (17) can be derived as
\[ \omega(i) = \left( \Psi(i)\hat{\Lambda}(i)\Psi(i)^T + \lambda_2 h_2^2 I \right)^{-1} \Psi(i)\hat{\Lambda}(i)d(i), \quad (18) \]

where \( \hat{\Lambda}(i) = \text{diag}[(\exp((d_1(i) - \hat{d}_1(i))^2 / -2h_2^2), \exp((d_2(i) - \hat{d}_2(i))^2 / -2h_2^2), \cdots, \exp((d_i(i) - \hat{d}_i(i))^2 / -2h_2^2)] \).

Using the matrix inversion lemma shown in (14) by letting
\[ \Lambda = \lambda_2 h_2^2 I, B = \Psi(i), C = \hat{\Lambda}(i), \) and \( D = \Psi(i)^T, \)

we rewrite (18) as
\[ \omega(i) = \Psi(i)\alpha(i). \quad (19) \]

where \( \alpha(i) = \left( K(i) + \lambda_2 h_2^2 \hat{\Lambda}(i) \right)^{-1}d(i) \) with the entry of positive definite matrix \( K(i) = \Psi(i)^T \Psi(i) \) calculated by (16). Therefore, \( \alpha(i) \) is used to represent \( \omega(i) \) 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.,
\[ \hat{d}_j(i) = \varphi(u(j))^T \Psi(i)\alpha(i) + b(i)^T \alpha(i)\hat{d}_{j-1}(i - 1), \quad j = 1, 2, \cdots i \quad (20) \]

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

\[
\hat{d}(i) = (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i),
\]

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

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

\[
J = \sum_{j=1}^{i} \kappa_{h_j}(e_j(i))
\]

with \( e_j(i) = d_j(i) - \hat{d}_j(i) \) denoting the \( j \)th 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., \( e(i) = [e_1(i), e_2(i), \ldots, e_i(i)]^T \). Here, we define the desired outputs by \( d(i) = (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i) \), where \( \alpha(i) \) and \( b(i) \) are the optimal FFC and FBC, respectively; \( v(i) = [v(1), v(2), \ldots, v(i)]^T \) is the collection of disturbance noises. According to (21), we have

\[
e(i) = d(i) - \hat{d}(i)
\]

\[
= d(i) - (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i)
\]

\[
= (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i) - (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i)
\]

\[
+ (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i) - (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i)
\]

\[
= (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) - \hat{d}(i-1)b(i)^T\alpha(i)
\]

\[
= \theta_{\alpha}(i) - \theta_{\alpha}(i) - \hat{d}(i-1)b(i)^T\alpha(i)
\]

\[
\leq \theta_{\alpha}(i) - \theta_{\alpha}(i) - \hat{d}(i-1)b(i)^T\alpha(i),
\]

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

\section*{B. WEIGHT UPDATE}

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

\[
\alpha(i+1) = \alpha(i) + s_{\alpha}(i) \frac{\partial J}{\partial \alpha(i)}.
\]

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

\[
\frac{\partial \hat{d}(i)}{\partial \alpha(i)} \approx \Psi(i) \varphi(u(j)) + b(i)\hat{d}_{i-1}(i-1) + r_{\alpha}^m(i)b(i)^T\alpha(i)\frac{\partial \hat{d}_{i-1}(i-1)}{\partial \alpha(i)},
\]

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

Define \( Y_{\alpha}^m(i) = \frac{\partial \hat{d}(i)}{\partial \alpha(i)} \). According to (21), we rewrite (26) by

\[
Y_{\alpha}^m(i) \approx K(i) + b(i)\hat{d}_{i-1}(i-1)^T + r_{\alpha}^m(i)b(i)^T\alpha(i)Y_{\alpha}^m(i-1).
\]

Combining (26) and (27), we rewrite (25) by

\[
\frac{\partial J}{\partial \alpha(i)} = \frac{\partial J}{\partial \alpha(i)} - Y_{\alpha}^m(i)\Lambda(i) e(i) = \frac{\partial J}{\partial \alpha(i)} - \Lambda(i)\bar{b}^T(i),
\]

where \( \Lambda(i) = \hat{\Lambda}(i)\bar{b}^T(i) \), and \( e(i) = [e(i), \ldots, e(i)]^T \) is the step size with \( \rho_{\alpha}^m(i) \) being positive; \( \frac{\partial \hat{d}(i-1)}{\partial b(i)} \approx \frac{\partial \hat{d}(i-1)}{\partial b(i)} \), and \( Y_{\alpha}^m(i) = \frac{\partial \hat{d}(i)}{\partial b(i)} \) is approximated by

\[
Y_{\alpha}^m(i) \approx \alpha(i)\hat{d}(i-1)^T + r_{\alpha}^m(i)b(i)^T\alpha(i)Y_{\alpha}^m(i-1).
\]

According to (29), we rewrite \( e(i) \) by

\[
e(i) = d(i) - \hat{d}(i)
\]

\[
= -(K(i) + \hat{d}(i-1)b(i)^T)\alpha(i)
\]

\[
+ (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i)
\]

\[
+ (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i)
\]

\[
= -(K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) - (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i)
\]

\[
= -(K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) - (K(i) + \hat{d}(i-1)b(i)^T)\alpha(i) + v(i)
\]

\[
\leq \theta_{\alpha}(i) - \theta_{\alpha}(i) - \hat{d}(i-1)b(i)^T\alpha(i),
\]

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

For compactness of the notation, \( \Omega = [\alpha, b] \) is introduced. The parameter settings are discussed as follows.

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

1. \( s_{\Omega}^m(i) = \frac{s_{\Omega}^m(i)}{s_{\Omega}^m(i)} \) in (28) and (29) is evaluated by

\[
s_{\Omega}^m(i) = \begin{cases} 1, & \text{if } \frac{s_{\Omega}^m(i)}{s_{\Omega}^m(i)} < 2 \left( e(i) - \theta_{\Omega}(i) \right) \Lambda(i) e(i) \\ 0, & \text{else} \end{cases}
\]

where \( \theta_{\Omega}(i) = \theta_{\alpha}(i) - \theta_{\alpha}(i) \).
where $\tilde{e}_\Omega(i) = e^m_\Omega(i) + e^m_\Omega(i-1)$ with $e^m_\Omega(i) = \|Y^m_\Omega(i)\Lambda(i)e(i)\|^2/\rho^m_\Omega(i)$.

The normalization factor $\rho^m_\Omega(i)$ in (28) and (29) is designed to prevent the vanished cone problem [11] which is updated by
\[ \rho^m_\Omega(i) = \max(\rho^m_\Omega(i-1), \alpha^m_\Omega\rho^m_\Omega(i-1) + \max(\tilde{e}^m_\Omega(i), \|Y^m_\Omega(i)\Lambda(i)e(i)\|^2)), \] (33)
where $\alpha^m_\Omega \in (0, 1)$ denotes the impact degree of the previous factor on the current one and $\rho^m_\Omega$ is a regulatory factor that ensures the initial learning rate being relatively small [12].

2. $r^m_\Omega(i)$ in (26) and (30) is defined by
\[ r^m_\Omega(i) = \begin{cases} \text{sgn}(\tilde{e}^m_\Omega(i)), & \text{if } |\mu^m(i)| < \frac{1}{\eta^m + |b(i)^T\alpha(i)|} \text{ (34)} \\ 0, & \text{else} \end{cases} \]
where $\text{sgn}(\cdot)$ denotes the sign function; $\tilde{e}^m_\Omega(i) = b(i)^T\alpha(i)\mu^m(i)\rho^m_\Omega(i-1)$ and
\[ \mu^m(i) = L^m(i-1)^T(L^m(i-1)L^m(i-1)^T)^{-1}L^m(i) \] (35)
with $L^m(i-1) = \Lambda(i-1)e(i-1)$; $\eta^m$ 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 $s^m_\Omega(i) = \tilde{e}^m_\Omega(i)/\rho^m_\Omega(i)$ in (28) and (29) is crucial for filtering convergence performance. The normalization factor $\rho^m_\Omega(i)$ is used for solving the vanishing radius problem [11], [44] and $\tilde{e}^m_\Omega(i)$ can be regarded as the switch controlling the updates of FFC and FBC. $r^m_\Omega(i)$ in (26) and (30) has the function of regulation factor $\lambda_2$ 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.

**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 only on the current error as the loss function, i.e.,
\[ J = \kappa_{h_1}(e(i)), \] (36)
where $e(i) = d(i) - \hat{d}(i)$. For simplicity, (36) can be rewritten by
\[ J = \kappa_{h_1}(e(i)), \] (37)
where $e(i) = d(i) - \hat{d}(i)$ and $\hat{d}(i) = (K(i)^T + \hat{d}(i-1)b(i)^T)\alpha(i)$ with $K(i)$ being the last column of $K(i)$. In comparison

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

**Initialization:**
Start with parameters initiation: $\theta^m_a, \theta^m_b, \varrho^m_a, \varrho^m_b, \delta^m_a, \delta^m_b, \gamma^m$.

while $|\mu(i), d(i)| > 1$ available do
1) Update $s^m_a(i), s^m_b(i)$ using (32) and (33), $r^m_a(i)$ and $r^m_b(i)$ using (34).
2) The update forms of $Y^m_a(i-1)$ and $Y^m_b(i-1)$ are shown as follows.
\[ \tilde{Y}^m_a(i) \approx K(i) + b(i)[\hat{d}(i-1) - \hat{d}(i-1)]^T + r^m_a(i)b(i)^T\alpha(i)\Theta^a_{i-1}, \]
\[ \tilde{Y}^m_b(i) \approx \alpha(i)\hat{d}(i-1)^T + r^m_b(i)b(i)^T\alpha(i)\Theta^b_{i-1}, \]
with $\hat{d}(i-1) = [\hat{d}_a(i-1); \hat{d}(i-1)]$ and $\hat{d}_a(i-1) = 0$ in (22); $\Theta^a_{i-1}$ and $\Theta^b_{i-1}$ are given by
\[ \Theta^a_{i-1} = \begin{bmatrix} 0 & Y^m_a(i-1) \\ 0 & 0 \end{bmatrix}, \Theta^b_{i-1} = \begin{bmatrix} 0 & Y^m_b(i-1) \\ 0 & 0 \end{bmatrix}, \]
with $0$ being the $(i-1) \times 1$ null vector.
3) The FFC $\alpha(i)$ and FBC $b(i)$ can be approximated iteratively by the recursions
\[ \alpha(i+1) = \begin{bmatrix} \alpha(i) \\ 0 \end{bmatrix} + s^m_a(i)\tilde{Y}^m_a(i)\Lambda(i)e(i). \]
\[ b(i+1) = \begin{bmatrix} b(i) \\ 0 \end{bmatrix} + s^m_b(i)\tilde{Y}^m_b(i)\Lambda(i)e(i). \]
end while

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
\[ \alpha(i+1) = \alpha(i) + s^m_a(i)Y^m_a(i)\Lambda(i)e(i). \] (38)
where $s^m_a(i) \in [0, 1/\rho^m_a(i)]$ is the step size with a positive scalar $\rho^m_a(i)$; $\Lambda_{i,j}(i) = \exp((d(i) - \omega^i\varrho^m_a(i))^2 - 2h^2_1)$ denotes the $i$th column and $j$th row element of diagonal matrix $\Lambda(i)$ in (28).

Vector $\tilde{Y}^m_a(i) = \partial\hat{d}(i)/\partial\alpha(i)$ in (38) has a similar update form to that in (27), i.e.,
\[ \tilde{Y}^m_a(i) \approx K(i) + b(i)[\hat{d}(i-1) + r^m_a(i)b(i)^T\alpha(i)\tilde{Y}^m_a(i-1), \] (39)
where $r^m_a(i)$ is the parameter for adjusting the recurrent gradient information of LRKOL-MCC.
For FBC $b(i)$, we obtain from (29)

$$b(i+1) = b(i) + s^b(i)\hat{Y}^b(i)\Lambda_{b,i}(i)e(i),$$

(40)

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

$$\hat{Y}^b(i) \approx \alpha(i)\hat{d}(i-1) + r^b_s(i)b(i)^T\alpha(i)Y^b(i) - 1).$$

(41)

The parameter settings are discussed as follows. The $s^Q(i)$ and $r_s^Q(i)$ obey the following updating rules.

1. $s^Q(i) = \tilde{s}^Q(i)/\rho^Q_s(i)$ in (38) and (40) is given by

$$\tilde{s}^Q(i) = \begin{cases} 1, & \text{if } \tilde{e}_Q(i) < 2(e(i) - \tilde{d}_Q)\Lambda_{1,i}(i)e(i) \\
0, & \text{else,}
\end{cases}$$

(42)

with $\tilde{e}_Q(i) = e_s^Q(i) + \tilde{e}_Q(i-1)$ with $e_s^Q(i) = \|Y^Q(i)\Lambda_{1,i}(i)e(i)\|^2/\rho^Q_s(i)$. The scalar $\tilde{d}_Q(i)$ is a positive scalar larger than $\tilde{d}_Q(i)$ with $\tilde{d}_Q(i) = v(i) - \tilde{d}(i-1)b(i)^T\alpha(i)$ and $\tilde{d}_Q(i) = v(i) - \hat{d}(i-1)b(i)^T\alpha(i)$. Here, the normalization factor $\rho^Q_s(i)$ in (38) and (40) can be updated by

$$\rho^Q_s(i) = \max(\rho^Q_s(i-1), \tilde{Q}_s^Q(i-1))$$

(43)

where $\tilde{Q}_s^Q(i) = e_s^Q(i) + \tilde{e}_Q(i-1)$ with $\tilde{e}_Q(i) = \tilde{e}_Q(i-1)$ being positive. 2. $r_s^Q(i)$ in (39) and (41) is given by

$$r_s^Q(i) = \begin{cases} \text{sgn}(\tilde{Q}_s^Q(i)), & \text{if } |\mu^Q(i)| < \frac{1}{\eta^Q + |b(i)^T\alpha(i)|} \\
0, & \text{else,}
\end{cases}$$

(44)

Remark 2: Compared with Algorithm 1 and Algorithm 2, the difference between KRMC-MF and LRLKOL-MCC is the update form of $\hat{Y}^Q(i)$ and $\hat{Y}^Q(i)$. Since multiple single-delay feedbacks are incorporated into KRMC-MF, $\hat{Y}^Q(i)$ is updated recursively in the form of matrix. However, $\hat{Y}^Q(i)$ in LRLKOL-MCC is recursively updated in the form of vector due to only one single-delay feedback. Therefore, LRLKOL-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., LRLKOL, KRLS, and KRLS-MF, and the MCC-based algorithms, i.e., LRLKOL-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 $\Lambda_{b,i}(i)$ in LRLKOL-MCC, and the matrices $B(i)$ and $\Lambda(i)$ with diagonal entries calculated by (2) in MMSE-MF 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 LRLKOL-MCC and KRMC-MF have almost the same computational costs as LRLKOL and KRLS-MF, respectively, which will also be presented in the following simulations.

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

**Proof:** The optimal FFC $\alpha_\ast$ subtracted from both sides of (28) gives

$$
\hat{\alpha}(i + 1) = \hat{\alpha}(i) + s^m_\alpha(i) Y^m_\alpha(i) \Lambda(i) e(i).
$$

(45)

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

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

(46)

where

$$
\Delta_\alpha(i) = 2s^m_\alpha(i) \hat{\alpha}(i)^T Y^m_\alpha(i) \Lambda(i) e(i)
+ (s^m_\alpha(i))^2 \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2.
$$

(47)

Substituting (27) into (47), we have

$$
\Delta_\alpha(i) \approx 2s^m_\alpha(i) \hat{\alpha}(i)^T (K(i) + b(i) \hat{\alpha}(i - 1) + r^m_\alpha(i)b(i)^T \alpha(i) Y^m_\alpha(i - 1) \Lambda(i) e(i)
+ (s^m_\alpha(i))^2 \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2.
$$

(48)

where $\Gamma_\alpha^1(i)$, $\Gamma_\alpha^2(i)$, and $\Gamma_\alpha^3(i)$ are defined as follows:

$$
\begin{align*}
\Gamma_\alpha^1(i) = s^m_\alpha(i) \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2, \\
\Gamma_\alpha^2(i) = 2r^m_\alpha(i) \hat{\alpha}(i)^T (b(i) \hat{\alpha}(i - 1) + r^m_\alpha(i)b(i)^T \alpha(i) Y^m_\alpha(i - 1) \Lambda(i) e(i)
+ (s^m_\alpha(i))^2 \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2, \\
\Gamma_\alpha^3(i) = 2b(i)^T (K(i) + b(i) \hat{\alpha}(i - 1) + r^m_\alpha(i)b(i)^T \alpha(i) Y^m_\alpha(i - 1) \Lambda(i) e(i)
+ (s^m_\alpha(i))^2 \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2.
\end{align*}
$$

(49)

According to (45), we expand $\Gamma_\alpha^3$ as

$$
\begin{align*}
\Gamma_\alpha^3(i) = 2b(i)^T (K(i) + b(i) \hat{\alpha}(i - 1) + r^m_\alpha(i)b(i)^T \alpha(i) Y^m_\alpha(i - 1) \Lambda(i) e(i)
+ (s^m_\alpha(i))^2 \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2.
\end{align*}
$$

(50)

where $e^m_\alpha(i) = \| Y^m_\alpha(i) \Lambda(i) e(i) \|^2 / \rho^m_\alpha(i)$ and $\mu^m(i) = L^m(i - 1)^T (L^m(i - 1)L^m(i - 1)^T)^{-1}L^m(i - 1)$ with $L^m(i - 1)$ is the identity matrix.

$$
\begin{align*}
\Delta_\alpha(i) \approx 2s^m_\alpha(i) \hat{\alpha}(i)^T \alpha(i) \hat{\alpha}(i - 1) + s^m_\alpha(i - 1)
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
\end{align*}
$$

(51)

Hence, equality holds

$$
\begin{align*}
\Delta_\alpha(i) \approx 2s^m_\alpha(i) \hat{\alpha}(i)^T \alpha(i) \hat{\alpha}(i - 1) + s^m_\alpha(i - 1)
+ 2s^m_\alpha(i - 1) e^m_\alpha(i) \rho^m_\alpha(i)
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
\end{align*}
$$

(52)

Therefore, we simplify (50) as

$$
\begin{align*}
\Gamma_\alpha^2(i) & = 2s^m_\alpha(i) b(i)^T \alpha(i) \mu^m(i) \rho^m_\alpha(i) (\alpha(i) - \alpha(i - 1))
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ (s^m_\alpha(i - 1))^2 e^m_\alpha(i) \rho^m_\alpha(i)
\end{align*}
$$

(53)

According to (52), we have $s^m_\alpha(i - 1) \rho^m_\alpha(i - 1) = 1$. Therefore, (53) can be simplified by

$$
\begin{align*}
\Gamma_\alpha^2(i) & = \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ 2s^m_\alpha(i - 1) e^m_\alpha(i) \rho^m_\alpha(i)
\end{align*}
$$

(54)

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

$$
\ell(i) = \begin{cases}
|b(i)^T \alpha(i) \mu^m(i)|, & \text{if } |\mu^m(i)| < \frac{1}{\rho^m_\alpha(i) |b(i)^T \alpha(i)|} \\
0, & \text{else.}
\end{cases}
$$

(55)

Since $s^m_\alpha(i) = s^m_\alpha(i) / \rho^m_\alpha(i)$ holds in (24) and (42), $s^m_\alpha(i)$ can be obtained only when $s^m_\alpha(i) = 0$, generating $r^m_\alpha(i) = 0$ which is given in (44). Therefore, $r^m_\alpha(i)$ can be derived using (49). Similarly, considering $\Delta_\alpha(i - 1)$ as a function of $s^m_\alpha(i - 1)$ shown in (48), we have $\Delta_\alpha(i - 1) = 0$ on the condition of $s^m_\alpha(i - 1) = 0$, which results in $\Gamma_\alpha^2(i)$ equals 0.

In addition, based on (23), $\Gamma_\alpha^3(i)$ can be derived as

$$
\begin{align*}
\Gamma_\alpha^3(i) & = 2(\theta_\alpha(i) - e(i))^T \Lambda(i) e(i).
\end{align*}
$$

(56)

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

$$
\begin{align*}
\Delta_\alpha(i) \approx 2s^m_\alpha(i) \hat{\alpha}(i)^T \alpha(i) \hat{\alpha}(i - 1) + s^m_\alpha(i - 1)
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ \ell(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ (s^m_\alpha(i - 1))^2 e^m_\alpha(i) \rho^m_\alpha(i)
\end{align*}
$$

(57)

Therefore, (57) can be rewritten as

$$
\begin{align*}
\Delta_\alpha(i) \approx 2s^m_\alpha(i) \hat{\alpha}(i)^T \alpha(i) \hat{\alpha}(i - 1) + s^m_\alpha(i - 1)
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ \ell(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ (s^m_\alpha(i - 1))^2 e^m_\alpha(i) \rho^m_\alpha(i)
\end{align*}
$$

(58)

where

$$
\begin{align*}
\chi(i) = 2s^m_\alpha(i) \hat{\alpha}(i)^T \alpha(i) \hat{\alpha}(i - 1) + s^m_\alpha(i - 1)
+ \gamma_\alpha(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ \ell(i) \| Y^m_\alpha(i - 1) \Lambda(i - 1) \Lambda(i - 1) e(i)
+ (s^m_\alpha(i - 1))^2 e^m_\alpha(i) \rho^m_\alpha(i)
\end{align*}
$$

(59)
Here (c1) follows from the inequality 0 ≤ ℓ(i) < 1 and 0 ≤ s′′α(i) ≤ 1/pρα(i); (c2) is obtained by combining (32).

Since (58) and Δα(1) ≤ 0, Δα(i) ≤ 0 in (58) can be derived, which gives \[ \| \hat{a}(i + 1) \|^2 - \| \hat{a}(i) \|^2. \]

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

The proposed KRMC-MF is proved to be convergent based on the designed learning parameters (32), (33), and (34). Since \( \alpha_s \) is unavailable in practice, we define \( W^\alpha_1(i) = \| \alpha(i + 1) \|^2 - \| \alpha(i) \|^2 \) and \( W^\gamma_2(i) = \alpha(i + 1) - \alpha(i) \) to rewrite \( \Delta_{\alpha}(i) \) as

\[
\| \Delta_{\alpha}(i) \| = \| \alpha(i + 1) \|^2 - \| \alpha(i) \|^2 = \| \alpha(i + 1) \|^2 - 2(\alpha(i + 1) - \alpha(i))^T \alpha_s \| \leq W^\alpha_1(i) + 2 \| \alpha(i + 1) - \alpha(i) \|^T \alpha_s \| \leq W^\alpha_1(i) + 2W^\gamma_2(i)g_1 \quad (60)
\]

with \( g_1 = \| \alpha_s \| \). Similarly, we have

\[
| \Delta_{b}(i) | \leq W^b_1(i) + 2W^b_2(i)g_2, \quad (61)
\]

where \( W^b_1(i) = \| b(i + 1) \|^2 - \| b(i) \|^2, W^b_2(i) = b(i + 1) - b(i), \) and \( g_2 = \| b_s \| \).

To observe the convergence of KRMC-MF, we define the convergence curves of \( \tau_\alpha(i) \) with \( \tau_\alpha(i) = \| W^\alpha_1(i) + 2W^\gamma_2(i) \| \) and \( \tau_b(i) = \| W^b_1(i) + 2W^b_2(i) \| \), which can avoid the direct calculation of \( \alpha_s \). The convergence curves of \( \tau_\alpha(i) \) and \( \tau_b(i) \) converge to zero, leading to \( \lim_{i \to \infty} \| \Delta_{\alpha}(i) \| = 0 \) and \( \lim_{i \to \infty} | \Delta_{b}(i) | = 0 \). 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 \( \lim_{i \to \infty} | \Delta_{\alpha}(i) | = 0 \) and \( \lim_{i \to \infty} | \Delta_{b}(i) | = 0 \).

### 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], KRMCC [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; KRMCC 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]:

\[
f(t) = \exp \{ j\beta t - \gamma |t|^\beta \} \quad (62)
\]

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 \( \delta \in (-\infty, \infty) \):

\[
S(t, \sigma) = \frac{\tan \frac{\sigma \pi}{2}}{2\pi \log |t|}, \quad \sigma = 1. \quad (63)
\]

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 Monte Carlo runs. The notations \( p^m = [\theta^m, \phi^m, \sigma^m, \theta^{\phi m}, \theta^\gamma m] \), \( p^s = [\theta^s, \phi^s, \sigma^s, \theta^{\phi s}, \theta^\gamma s] \), and \( q^m = [\theta^m, \phi^m, \sigma^m] \) are adopted for clarity. Furthermore, we also introduce the collection of regularization factors \( \lambda = [\lambda_1, \lambda_2] \) with \( \lambda_1 \) and \( \lambda_2 \) 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

\[
MSE = \frac{1}{N} \sum_{j=1}^{N} (d(j) - \hat{d}(j))^2, \quad (64)
\]

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]:

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

where \( \tau = 30 \). The time series is first discretized using a sampling period of 6 seconds. The input \( u(i) = [x(i-7), x(i-6), \ldots, x(i-2), x(i-1)] \) 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, KRMCC, and KRMC-MF, we have \( h_1 = 0.17 \) in (2) for LRKOL-MCC and \( h_1 = 0.8 \) for KRMCC and KRMC-MF. The kernel size \( h_2 \) in (16) is chosen as 0.85. The collection \( \lambda = [0.001, 0.001] \) is chosen in simulations. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, \( p^m = p^s = [0.65, 0.93, 0.1, 150, 0.01] \) is configured. In this simulation, \( q^m = [0.0096, 0.0096] \) gives that \( \theta^m \) or \( \theta^{\phi m} \) 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.
i.e., $q^1 = [0.0096, 0.0096]$. 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.

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{fig2.png}
\caption{Learning curves of KAFs in MG time series prediction under the Gaussian noises.}
\end{figure}

In this simulation, the alpha stable noise $v(i)$ is shown in Fig. 3, with parameters set as $V = (1.4, 0.02, 0)$. The parameters $h_1$ and $h_2$ are configured as 1.2 and 0.23, respectively. The regularization factor $\lambda = [0.95, 0.001]$ is used in the non-Gaussian environment. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, the parameters are configured as $p^m = p^s = [0.99, 0.93, 0.5, 150, 0.0001]$, $q^m = [0.02, 0.02]$ and $q^s = [0.02, 0.02]$. 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.

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{fig3.png}
\caption{A sequence of the alpha-stable noise with $V = (1.4, 0, 0.02, 0)$ in MG time series prediction.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\columnwidth]{fig4.png}
\caption{Learning curves of KAFs in MG time series prediction under the alpha-stable noises.}
\end{figure}

\section*{B. NONLINEAR REGRESSION}

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

\begin{equation}
\begin{aligned}
x(i) &= x(i-1)(0.8 - 0.5 \exp(-x^2(i - 1))) - 0.3 \\
&\quad + 0.9 \exp(-x^2(i - 1))x(i - 2) + 0.1\sin(x(i - 1)\pi),
\end{aligned}
\end{equation}

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 $u(i) = [x(i - 1), x(i - 2)]$ 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 $\lambda$ are configured as $h_2 = 0.23$ and $\lambda = [0.001, 0.001]$. For LRKOL, LRKOL-MCC, KRLS-MF, and KRMC-MF, we choose $p^m = p^s = [0.88, 0.999, 0.1, 150, 0.01]$, $q^m = [0.015, 0.015]$, and $q^s = [0.015, 0.015]$. 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 $\Delta_{\alpha}(i)$ or $\Delta_{B}(i)$. To validate the convergence of KRMC-MF
TABLE 2. Comparison of mean consumed time in nonlinear regression under gaussian noises.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LRKO</th>
<th>LRKO-MCC</th>
<th>KRLS</th>
<th>KRMC</th>
<th>KRLS-MF</th>
<th>KRMC-MF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean consumed time (s)</td>
<td>6.33</td>
<td>6.41</td>
<td>17.76</td>
<td>17.80</td>
<td>27.70</td>
<td>30.43</td>
</tr>
</tbody>
</table>

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 $\lim_{i \to \infty} |\Delta_\alpha(i)| = 0$ and $\lim_{i \to \infty} \|\Delta_b(i)\|^2 = 0$. According to (61), $\lim_{i \to \infty} |\Delta_\alpha(i)| = 0$ is derived, which complies with Theorem 1. The similar conclusions can be obtained for FBC, i.e., $\lim_{i \to \infty} |\Delta_b(i)| = 0$. Similarly, the convergence of LRKO-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 $\lim_{i \to \infty} |\Delta_\alpha(i)| = 0$ and $\lim_{i \to \infty} |\Delta_b(i)| = 0$ in LRKO-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 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, \( p^m = p^r = [0.985, 0.93, 0.8, 150, 0.0001] \), \( q^m = [0.2, 0.2] \) and \( q^r = [0.2, 0.2] \) are configured. 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; \( \eta \) 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

\[
\alpha(i) = \alpha(i - 1) + s^m(i - 1)\mathbf{Y}_a^m(i - 1)\mathbf{A}(i - 1)e(i - 1).
\]

(68)

From (68), we obtain

\[
\mathbf{Y}_a^m(i - 1) = \frac{\alpha(i) - \alpha(i - 1)}{s^m_a(i - 1)} \mathbf{L}^m(i - 1)^T \times (\mathbf{L}^m(i - 1)\mathbf{L}^m(i - 1)^T)^{-1}, \quad (69)
\]

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

\[
\mathbf{Y}_a^m(i) = \mathbf{K}(i) + b(i)\hat{d}(i - 1)^T + r^m(i)b(i)^T \alpha(i) \mathbf{Y}_a^m(i - 1) = \mathbf{K}(i) + b(i)\hat{d}(i - 1)^T + \frac{r^m(i)b(i)^T \alpha(i)[\alpha(i) - \alpha(i - 1)]}{s^m_a(i - 1)} \mathbf{L}^m(i - 1)^T \times (\mathbf{L}^m(i - 1)\mathbf{L}^m(i - 1)^T)^{-1},
\]

(70)

Substituting (70) into (28) yields

\[
\alpha(i + 1) = \alpha(i) + s^m_a(i)\mathbf{Y}_a^m(i)\mathbf{A}(i)e(i) = \alpha(i) + s^m_a(i)[\mathbf{K}(i) + b(i)\hat{d}(i - 1)^T]\mathbf{A}(i)e(i)
\]

\[
+ [\alpha(i) - \alpha(i - 1)] \times \frac{s^m_a(i)s^m_a(i)b(i)^T \alpha(i)}{s^m_a(i - 1)} \mathbf{L}^m(i - 1)^T \times (\mathbf{L}^m(i - 1)\mathbf{L}^m(i - 1)^T)^{-1} \mathbf{A}(i)e(i)
\]

\[
= \alpha(i) + s^m_a(i)[\mathbf{K}(i) + b(i)\hat{d}(i - 1)^T]\mathbf{A}(i)e(i) + [\alpha(i) - \alpha(i - 1)] \mathbf{Y}_a(i),
\]

(71)

where \( \alpha(i) - \alpha(i - 1) \) can be regarded as a momentum term, \( \mathbf{Y}_a(i) = \frac{s^m_a(i)b(i)^T \alpha(i)\mu^m(i)}{s^m_a(i - 1)} \) is the corresponding momentum coefficient, and \( \mu^m(i) \) is defined in (35). When KRMC-MF converges, we obtain

\[
\lim_{i \to \infty} \frac{s^m_a(i)}{s^m_a(i - 1)} \approx 1.
\]

(72)

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

\[
\tilde{\mathbf{Y}}_a(i) = \frac{s^m_a(i)b(i)^T \alpha(i)\mu^m(i)}{s^m_a(i - 1)} \approx 1
\]

(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**

LUUJAN 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 Professor, 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.