Kernel Adaptive Hammerstein Filter

Yunfei Zheng[1], Jiyao Dong[1], Wentao Ma[2], and Badong Chen[1,*]
[1] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[2] Department of Electrical Engineering, Xi'an University of Technology, Xi'an 710048, China
Corresponding author. E-mail: chenbd@mail.xjtu.edu.cn (B. Chen)

***Abstract*—To identify Hammerstein systems, a variety of Hammerstein filters have been proposed. However, most of them assume the nonlinear part in Hammerstein systems to be polynomial in the process of modeling, which restricts their applicability in many practical situations. In this paper, a simple kernel adaptive filter (KAF) called kernel least mean square (KLMS) combined with coherence criterion (CC) is used to approximate the nonlinear part of a Hammerstein system, resulting in the kernel adaptive Hammerstein filter (KAHF). The KAHF can identify various Hammerstein systems well without any prior knowledge of nonlinear part. Simulation results confirm the desirable performance of the new method.**

***Index Terms*—Hammerstein system identification, kernel adaptive filter, infinite impulse response system**

I. Introduction

The Hammerstein system, which consists of a memoryless nonlinear part followed by a linear subsystem, can describe a great deal of practical applications, such as the power amplifier [1], the wind turbine [2], the valve actuator [3], etc. Generally, the memoryless nonlinear part in the Hammerstein system can be modeled as polynomial function, piecewise linear function, saturation function, etc [4]–[7]. As for the linear subsystem, the finite impulse response (FIR) and the infinite impulse response (IIR) are two commonly used models [8], [9].

To identify Hammerstein systems, more and more research efforts have been dedicated to developing the adaptive Hammerstein filter (AHF), thereby generating the least-mean-square based AHF [10], [11], affine-projection based AHF [12], Kalman filtering based AHF [13], and so on. These methods show powerful identification performance for Hammerstein systems. However, they focus mainly on the polynomial function based Hammerstein model, which restricts their applicability in many real world situations. When the nonlinear part is not polynomial or unknown, the performance of these methods may degrade seriously. To address this issue, a universal and efficient method for approximating the nonlinear part of the Hammerstein system is necessary.

As a class of universal approximators, the KAFs [14] are developed in a high dimensional reproducing kernel Hilbert space (RKHS) by using the linear structure (inner product) of this space to implement the traditional linear adaptive filers. Well-known KAFs include the KLMS [15], kernel affine projection algorithms (KAPAs) [16], and kernel recursive least squares (KRLS) [17]. They find successful applications in nonlinear learning tasks such as chaotic time series prediction, nonlinear channel equalization, adaptive noise cancellation, etc [18]–[20]. Overall, the KAFs have several desirable features: 1) with a universal kernel, they are universal learner, 2) under the mean square error criterion, the performance surface is quadratic in RKHS, so gradient descent learning does not suffer from local minima, and 3) they have moderate complexity in terms of computation and memory especially when the redundant features are pruned. In this work, the KAFs are considered to approximate the nonlinear part of the Hammerstein system. It is worth noting that a similar idea had been proposed in [21] and [22]. Since the FIR filter is the default choice for the linear subsystem, they can update the linear and nonlinear parameters separately. However, for practical applications, the FIR filter as the linear subsystem is, of course, not always the good choice. Moreover, once the linear subsystem is modeled as the IIR filter in [21] and [22], how to update the linear and nonlinear parameters will become an issue.

In the present paper, the KLMS is used to approximate the nonlinear part of the Hammerstein system, and the IIR filter is used to build the model for the linear part, resulting in the kernel adaptive Hammerstein filter (KAHF). Similarly to [11], the recursive iteration method is used to perform online update. Since the KLMS can approximate well any nonlinear function with a proper kernel function and step-size [14], the KAHF can perform well for any form of nonlinear subsystem. Further, to reduce the computational burden of the KAHF, the coherence criterion (CC) [23]–[25] is used to constrain the dictionary size.

II. Kernel Adaptive Hammerstein Filter

A. Identification model

Fig. 1 shows the proposed model for identifying the Hammerstein system, in which the nonlinear part is modeled as a KAF, and the linear subsystem is modeled as an IIR filter. Without loss of generality, the input signal $u(n)$ and the desired response $d(n)$ that has been polluted by a disturbance noise $v(n)$ are known. As the output of the KAF, $\hat{z}(n)$ can be obtained by using the KLMS [14], [15] method, namely

$$\hat{z}(n) = \sum_{l=1}^{L} \hat{p}_l(n-1)\kappa\big(u(n), \mathcal{D}_l(n-1)\big), \tag{1}$$

where $\hat{p}_l(n-1)$ is the coefficient of the KLMS, $\mathcal{D}(n-1) = \{\mathcal{D}_1(n-1), \cdots, \mathcal{D}_L(n-1)\}$ represents the dictionary, and κ

This work was supported by 973 Program (No. 2015CB351703) and National Natural Science Foundation of China (No. 91648208).

ISBN 978-90-827970-1-5 © EURASIP 2018

is the kernel function. In the following, the Gaussian kernel with kernel width σ will be the default choice, i.e., $\kappa(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2})$.

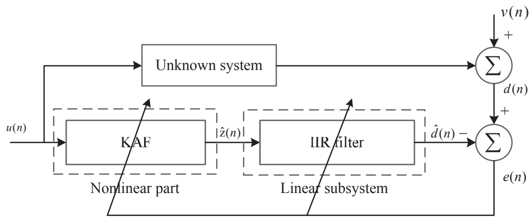

Fig. 1. Block diagram of the proposed adaptive Hammerstein system model.

Regarding $\hat{z}(n)$ as the input of the IIR filter, we get

$$\hat{d}(n) = -\sum_{i=1}^{N} \hat{a}_i(n-1)\hat{d}(n-i) + \sum_{j=0}^{M} \hat{b}_j(n-1)\hat{z}(n-j), \quad (2)$$

where $\hat{a}_i(n-1)$ is the feedback coefficient with length being N, and $\hat{b}_i(n-1)$ is the feedforward coefficient with length being M. Our purpose is to make the $\hat{d}(n)$ and $d(n)$ close enough, so that the system can be identified well. When $\hat{b}_0(n) = 1$, Equation (2) can be rewritten as

$$\hat{d}(n) = -\sum_{i=1}^{N} \hat{a}_i(n-1)\hat{d}(n-i) + \sum_{j=1}^{M} \hat{b}_j(n-1)\hat{z}(n-j) + \sum_{l=1}^{L} \hat{p}_l(n-1)\kappa\big(u(n), \mathcal{D}_l(n-1)\big). \quad (3)$$

Thus, the goal is simplified to update the coefficients $\hat{a}_i(n-1)$, $\hat{b}_j(n-1)$, as well as $\hat{p}_l(n-1)$ and, meanwhile, construct the dictionary $\mathcal{D}_l(n-1)$ such that the estimated output is close to the desired response.

B. Online update

In this subsection, we adopt the recursive iteration method to update the coefficients in (3), and the CC [23]–[25] is used to construct the corresponding dictionary. Without loss of generality, the cost function is defined as

$$J(n) = \frac{1}{2}e^2(n), \quad (4)$$

where $e(n) = d(n) - \hat{d}(n)$ is the prediction error which reflects the difference between the estimated output $\hat{d}(n)$ and the desired response $d(n)$. To minimize $J(n)$, the gradients of $J(n)$ with respect to $\hat{a}_i(n-1)$, $\hat{b}_j(n-1)$ and $\hat{p}_l(n-1)$ are firstly written as

$$\frac{\partial J(n)}{\partial \hat{a}_i(n-1)} = -e(n)\frac{\partial \hat{d}(n)}{\partial \hat{a}_i(n-1)}, \quad 1 \le i \le N, \quad (5)$$

$$\frac{\partial J(n)}{\partial \hat{b}_j(n-1)} = -e(n)\frac{\partial \hat{d}(n)}{\partial \hat{b}_j(n-1)}, \quad 1 \le j \le M, \quad (6)$$

and

$$\frac{\partial J(n)}{\partial \hat{p}_l(n-1)} = -e(n)\frac{\partial \hat{d}(n)}{\partial \hat{p}_l(n-1)}, \quad 1 \le l \le L. \quad (7)$$

To get the explicit expression for the gradients update, we need to estimate the partial derivatives in (5), (6) and (7), respectively. For Equation (5), the partial derivative of $\hat{d}(n)$ with respect to $\hat{a}_i(n-1)$ is given by

$$\frac{\partial \hat{d}(n)}{\partial \hat{a}_i(n-1)} = -\hat{d}(n-i) - \sum_{s=1}^{N} \hat{a}_s(n-1)\frac{\partial \hat{d}(n-s)}{\partial \hat{a}_i(n-1)}. \quad (8)$$

When the adaptation is slowly enough, there exists

$$\frac{\partial \hat{d}(n)}{\partial \hat{a}_i(n-1)} \approx \frac{\partial \hat{d}(n)}{\partial \hat{a}_i(n)}, \quad (9)$$

and

$$\frac{\partial \hat{d}(n-s)}{\partial \hat{a}_i(n-1)} \approx \frac{\partial \hat{d}(n-s)}{\partial \hat{a}_i(n-s)}. \quad (10)$$

Substituting (9) and (10) into (8), we have

$$\frac{\partial \hat{d}(n)}{\partial \hat{a}_i(n)} \approx -\hat{d}(n-i) - \sum_{s=1}^{N} \hat{a}_s(n-1)\frac{\partial \hat{d}(n-s)}{\partial \hat{a}_i(n-s)}. \quad (11)$$

Similarly, we get

$$\frac{\partial \hat{d}(n)}{\partial \hat{b}_j(n)} \approx \hat{z}(n-j) - \sum_{s=1}^{N} \hat{a}_s(n-1)\frac{\partial \hat{d}(n-s)}{\partial \hat{b}_j(n-s)}. \quad (12)$$

Finally, due to the space limitation, the partial derivative of $\hat{d}(n)$ with respect to $\hat{p}_l(n-1)$ is directly given by

$$\frac{\partial \hat{d}(n)}{\partial \hat{p}_l(n-1)} \approx \kappa(u(n), \mathcal{D}_l(n-1)) - \sum_{s=1}^{N} \hat{a}_s(n-1)\frac{\partial \hat{d}(n-s)}{\partial \hat{p}_l(n-1)} + \sum_{s=1}^{M} \hat{b}_s(n-1)\kappa(u(n-s), \mathcal{D}_l(n-1))\frac{\partial \hat{p}_l(n-s)}{\partial \hat{p}_l(n-1)}. \quad (13)$$

With a similar procedure used from (9) to (10), we have

$$\frac{\partial \hat{d}(n)}{\partial \hat{p}_l(n-1)} \approx \frac{\partial \hat{d}(n)}{\partial \hat{p}_l(n)}, \frac{\partial \hat{d}(n-s)}{\partial \hat{p}_l(n-1)} \approx \frac{\partial \hat{d}(n-s)}{\partial \hat{p}_l(n-s)}, \quad (14)$$

and $\frac{\partial \hat{p}_l(n-s)}{\partial \hat{p}_l(n-1)} \approx 1$. Thus, Equation (13) can be rewritten as

$$\frac{\partial \hat{d}(n)}{\partial \hat{p}_l(n)} \approx \kappa(u(n), \mathcal{D}_l(n-1)) - \sum_{s=1}^{N} \hat{a}_s(n-1)\frac{\partial \hat{d}(n-s)}{\partial \hat{p}_l(n-s)} + \sum_{s=1}^{M} \hat{b}_s(n-1)\kappa(u(n-s), \mathcal{D}_l(n-1)). \quad (15)$$

Let

$$\phi(n) = \Big[\frac{\partial \hat{d}(n)}{\partial \hat{a}_1(n)}, \cdots, \frac{\partial \hat{d}(n)}{\partial \hat{a}_N(n)}, \frac{\partial \hat{d}(n)}{\partial \hat{b}_1(n)}, \cdots, \frac{\partial \hat{d}(n)}{\partial \hat{b}_M(n)}, \frac{\partial \hat{d}(n)}{\partial \hat{p}_1(n)}, \cdots, \frac{\partial \hat{d}(n)}{\partial \hat{p}_L(n)}\Big]^T, \quad (16)$$

ISBN 978-90-827970-1-5 © EURASIP 2018

and

$$\psi(n)=\Big[-\hat{d}(n-1),\cdots,-\hat{d}(n-N),\hat{z}(n-1),\cdots,\hat{z}(n-M),$$
$$\kappa(u(n),\mathcal{D}_1(n-1))+\sum_{s=1}^{M}\hat{b}_s(n-1)\kappa(u(n-s),\mathcal{D}_1(n-1)),\cdots,$$
$$\kappa(u(n),\mathcal{D}_L(n-1))+\sum_{s=1}^{M}\hat{b}_s(n-1)\kappa(u(n-s),\mathcal{D}_L(n-1))\Big]^T. \tag{17}$$

Then, we obtain a compact expression for the gradients update, i.e.,

$$\phi(n) = \psi(n) - \sum_{s=1}^{N}\hat{a}_s(n-1)\phi(n-s). \tag{18}$$

For simplifying representation, we define the data vector and the coefficient vector, respectively, as

$$\hat{\mathbf{h}}(n)=[-\hat{d}(n-1),\cdots,-\hat{d}(n-N),\hat{z}(n-1),\cdots,\hat{z}(n-M),$$
$$\kappa(u(n),\mathcal{D}_1(n-1))),\cdots,\kappa(u(n),\mathcal{D}_L(n-1))]^T, \tag{19}$$

and

$$\hat{\mathbf{w}}(n-1) =[\hat{a}_1(n-1),\cdots,\hat{a}_N(n-1),\hat{b}_1(n-1),\cdots,$$
$$\hat{b}_M(n-1),\hat{p}_1(n-1),\cdots,\hat{p}_L(n-1)]^T. \tag{20}$$

Thus, the normalized iteration for updating the coefficients of the KAHF can be written as

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \frac{\mathbf{\Lambda}\phi(n)}{\rho + \hat{\mathbf{h}}^T(n)\mathbf{\Lambda}\phi(n)}e(n), \tag{21}$$

where $\mathbf{\Lambda}=\mathrm{diag}[\eta_1,\cdots,\eta_{N+M+L}]$ denotes diagonal matrix of the step sizes, and ρ is a small positive constant to avoid the denominator being 0. The normalization is a commonly used method in AHFs, and the details can be found in [11].

As we known, the main challenge of the KAFs is their linear growing structure with each new training sample, which causes large computation and storage burdens under the case of large number of training data. Therefore, a proper pruning criteria is necessary for removing the redundant training samples. In recent years, the quantization method [26]–[30] and the CC [23]–[25] have been proved very efficient in compressing the dictionary of KAFs and meanwhile can keep a better filter performance. For simplicity, the CC is adopted to update the dictionary in this work. When a new input arrives, we first compute its coherence with the dictionary, i.e., $\mu = \max k\big(x(n),\mathcal{D}_l(n-1)\big)$, where $l \in [1, L]$, and then make a comparison between μ and a given threshold ε. According to the comparison results, we have

$$\begin{cases} \mathcal{D}_l(n) = \{\mathcal{D}_l(n-1), x(n)\}, & if\ \mu \le \varepsilon \\ \mathcal{D}_l(n) = \mathcal{D}_l(n-1), & otherwise. \end{cases} \tag{22}$$

Once the size of $\mathcal{D}_l(n)$ changes, the dimension of $\hat{\mathbf{h}}(n+1)$ will change, too. Hence, the dimension of $\hat{\mathbf{w}}(n)$ should be changed accordingly, that is

$$\begin{cases} \hat{\mathbf{w}}(n) = [\hat{\mathbf{w}}(n)^T, 0]^T, & if\ \mu \le \varepsilon \\ \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n), & otherwise. \end{cases} \tag{23}$$

The proposed KAHF algorithm is then summarized in **Algorithm 1**.

Algorithm 1: KAHF algorithm

Definitions:
σ: kernel width, ε: sparsification threshold,
ρ: regulization parameter, $\mathbf{\Lambda}$: diagonal matrix of step sizes,
$\mathcal{D}(n)$: dictionary at iteration n, $\hat{\mathbf{w}}(n)$: coefficient vector.

Main Loop:
Compute $\hat{z}(n)$ according to (1).
Construct the data vector $\hat{\mathbf{h}}(n)$ according to (19).
Compute the estimated output: $\hat{d}(n) = \hat{\mathbf{w}}(n-1)^T\hat{\mathbf{h}}(n)$.
Compute the prediction error: $e(n) = d(n) - \hat{d}(n)$.
Construct $\phi(n)$ according to (16).
Construct $\psi(n)$ according to (17).
Update the gradient vector $\phi(n)$ according to (18).
Update the coefficient vector $\hat{\mathbf{w}}(n-1)$ according to (21).
Adjust the coefficient vector $\hat{\mathbf{w}}(n)$ according to (23).

Remark: Unlike the traditional adaptive Hammerstein filter (AHF) with polynomial nonlinearity, the KAHF uses a simple KAF, i.e., KLMS, to learn the nonlinear component of a Hammerstein system, and thus can identify various Hammerstein systems well without requiring any prior knowledge of nonlinear part. For further improving the performance of the KAHF, one can consider using the regularization technology applied in [31] and [32], design methodology for the Gaussian KLMS algorithm discussed in [33], and so on.

III. Simulation results

In this section, the proposed KAHF is applied to identify different Hammerstein systems, in which the nonlinear parts including the polynomial as well as other forms are considered. For all experiments, the input signal $x(n)$ is generated by a zero mean white Gaussian process with variance 0.1. The desired response $d(n)$ is obtained by corrupting the output of the system with an additive white Gaussian noise $v(n)$ with zero mean and variance such that the output signal-to-noise ratio (SNR) is 30dB. In the following, 30000 data are used for training and 50 data are used for testing. The kernel width of the Gaussian kernel is fixed at $\sigma = 0.1$, and the matrix of step sizes is set to $\mathbf{\Lambda} = 1\times10^{-5}\mathbf{I}$, where $\mathbf{I}$ denotes the identify matrix. To reduce the disturbance caused by experiment itself, all simulation results are conducted over 30 Monte Carlo runs.

First, the Hammerstein system with polynomial nonlinearity is considered, namely the input-output relationship of the memoryless nonlinear part is given by

$$z(n) = 0.3x(n) + 0.2x^2(n) + 0.4x^3(n). \tag{24}$$

For the linear part, the transfer function is set as

$$H(z) = \frac{1 + 0.3z^{-1}}{1 - 0.8z^{-1} + 0.1z^{-2}}. \tag{25}$$

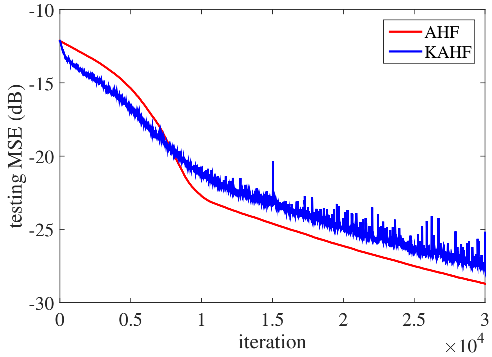

Fig. 2. Testing MSE curves of different algorithms for identifying Hammerstein system with polynomial nonlinearity.

Fig. 2 shows the testing mean square error (MSE) curves of different algorithms, where the parameters for regularization is fixed at $\rho = 0.005$ and for sparsification is set as $\varepsilon = 0.996$. In addition, the traditional AHF with polynomial nonlinearity is chosen for comparison. As can be seen from Fig. 2 that, the final testing MSE of KAHF is close to but slightly larger than that of the AHF. However, one should note that the Hammerstein system in this experiment has the determined polynomial nonlinearity, and hence can be modeled accurately by the AHF method.

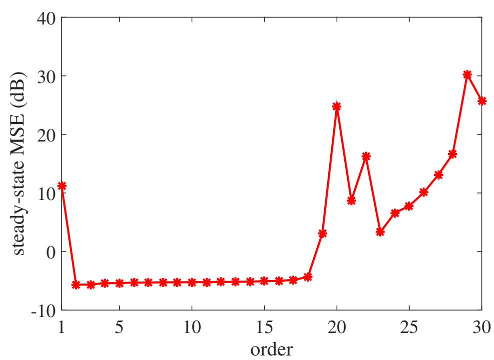

Fig. 3. Steady-state MSE of AHF versus polynomial order.

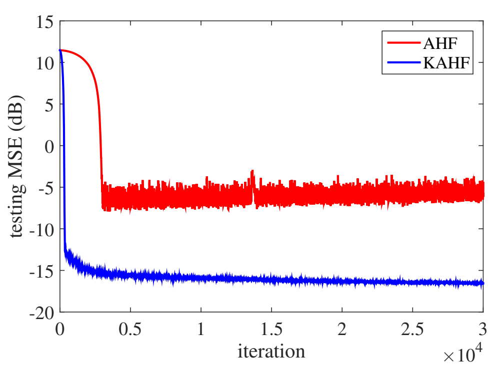

Fig. 4. Testing MSE curves of different algorithms for identifying Hammerstein system with sinc function nonlinearity.

Then, we change the nonlinear function in (24) to $z(n) = \mathrm{sinc}(x(n))$, and keep the transfer function in (25) unchanged. For choosing the optimal order of polynomial for AHF, the corresponding steady-state MSE versus polynomial order is shown in Fig. 3. Herein, $\rho = 0.02$ and $\varepsilon = 0.97$ are used. For each polynomial order, the steady-state MSE is obtained by averaging the last 500 iterations of its corresponding testing MSE. One can see from Fig. 3 that, when the order is about 3, the steady-state MSE of AHF can reach the minimum. Therefore, the order of the polynomial for AHF is selected as 3 in Fig. 4 which shows the testing MSE curves of AHF and KAHF. As expected, the KAHF can obtain a better identification performance, but the performance of the AHF degenerates seriously.

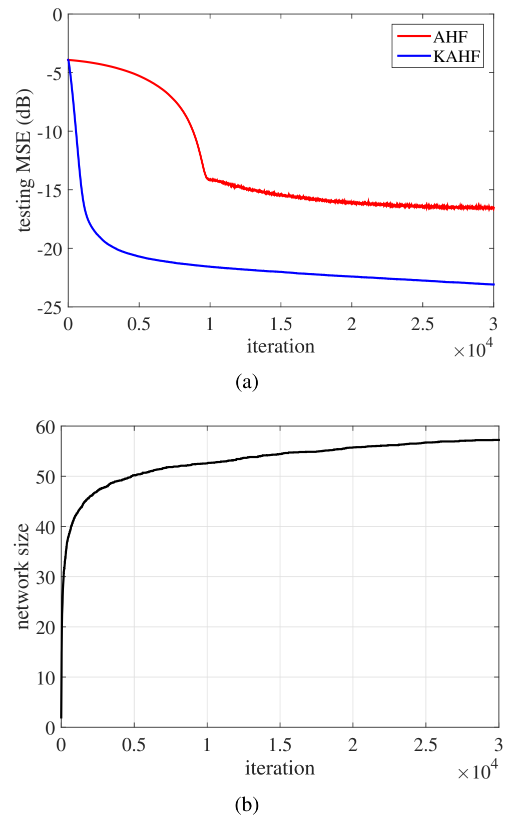

Fig. 5. Performance curves of different algorithms: (a) testing MSE curves of AHF and KAHF; (b) network size evolution curve of KAHF with $\varepsilon = 0.95$.

Further, a general Hammerstein system composed of the memoryless nonlinearity with input-output relationship

$$z(n) = 0.1\sin\big(\pi * x(n)\big) + 0.3\exp\big(-x^2(n)\big), \tag{26}$$

and linear component with the transfer function

$$H(z) = \frac{1 + 0.8z^{-1}}{1 - 0.5z^{-1} + 0.3z^{-2}}, \tag{27}$$

is identified using both AHF and KAHF. The corresponding performance curves of different algorithms are shown in Fig. 5, where $\rho = 0.01$ is used for regularization, and $\varepsilon = 0.95$ is used for compressing dictionary. In addition, the polynomial order is set as 6 for AHF according to the result of extended

ISBN 978-90-827970-1-5 © EURASIP 2018

experiments. It can be seen from Fig. 5(a) that, the KAHF has the better identification performance for such nonlinear Hammerstein system than AHF, which is consistent with the result in Fig. 4. Moreover, as shown in Fig. 5(b), the final network size of KAHF is only 57 when the sparsification parameter is set as $\varepsilon = 0.95$, which means the KAHF with CC can effectively curb its increasing network structure, and hence wouldn't cause a large computational burden.

IV. Conclusion

This paper proposed a new Hammerstein filter named as KAHF, which consists of a kernel adaptive filter (KAF) nonlinear part followed by an IIR linear subsystem. Different from the traditional adaptive Hammerstein filter (AHF) with polynomial nonlinearity, the KAHF uses a simple KAF, i.e., KLMS, to learn the nonlinear component of a Hammerstein system, and thus can identify various Hammerstein systems well without requiring any prior knowledge of nonlinear part. To reduce the computational complexity, the coherence criterion (CC) is adopted to constrain the network size of KLMS. Simulation results demonstrate the desirable performance of the new method. Future works will focus on studying the convergence properties of the KAHF, and trying to apply it to some practical applications, like the acoustic echo cancellation of loudspeakers.

References

[1] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[2] G.V.D. Veen, J.W.V. Wingerden, and M. Verhaegen, "Global identification of wind turbines using a Hammerstein identification method," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1471–1478, Aug. 2013.

[3] J. Wang and Q. Zhang, "Detection of asymmetric control valve stiction from oscillatory data using an extended Hammerstein system identification method," *J. Process Control*, vol. 24, no. 1, pp. 1–12, Jan. 2014.

[4] F. Ding, X. Liu, and G. Liu, "Identification methods for Hammerstein nonlinear systems," *Digit. Signal Prog.*, vol. 21, no. 2, pp. 215–238, Mar. 2011.

[5] Y. Liu and E. Bai, "Iterative identification of Hammerstein systems with piecewise-linear nonlinearities," in *Proceedings of the 2006 American Control Conference*, 2006, pp. 2357–2362.

[6] Y. Liu and E. Bai, "Iterative identification of Hammerstein systems," *Automatica*, vol. 43, no. 2, pp. 346–354, Feb. 2007.

[7] J. Vörös, "Iterative algorithm for parameter identification of hammerstein systems with two-segment nonlinearities," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2145–2149, Nov. 1999.

[8] 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, Oct. 2015.

[9] T. Onur and R. Hacioğlu, "Nonlinear acoustic echo cancellation using an adaptive Hammerstein block structure based on a generalized basis function," *J. Electr. Eng. Comput. Sci.*, vol. 25, no. 2, pp. 939–949, Apr. 2017.

[10] J. Jeraj and V. J. Mathews, "Stochastic mean-square performance analysis of an adaptive Hammerstein filter," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2168–2177, Jun. 2006.

[11] J. Jeraj and V. J. Mathews, "A stable adaptive Hammerstein filter employing partial orthogonalization of the input signals," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1412–1420, Apr. 2006.

[12] J.U. Ifiok and O. Tokunbo, "An affine projection-based algorithm for identification of nonlinear Hammerstein systems," *Signal Process.*, vol. 90, no. 6, pp. 2020 – 2030, Jun. 2010.

[13] J. Ma, F. Ding, W. Xiong, and E. Yang, "Combined state and parameter estimation for Hammerstein systems with time delay using the Kalman filtering," *J. Adapt. Control Signal Process.*, vol. 31, no. 8, pp. 1139–1151, Jan. 2017.

[14] J. Príncipe, W. Liu, and S. Haykin, *Kernel adaptive filtering : a comprehensive introduction*, New York: Wiley, 2010.

[15] W. Liu, P. Pokharel, and J.C. Príncipe, "The kernel least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Jan. 2008.

[16] W. Liu and J.C. Príncipe, "Kernel affine projection algorithms," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–12, Mar. 2008.

[17] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Jul. 2004.

[18] S. Wang, J. Feng, and K. Chi, "Kernel affine projection sign algorithms for combating impulse interference," *IEEE Trans. Circuits Syst. II-Express Briefs*, vol. 60, no. 11, pp. 811–815, Oct. 2013.

[19] Z. Mazloomi, H. Shandiz, and H. Faramarzpour, "Kernel least mean square algorithm in control of nonlinear systems," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, May 2013, pp. 1–5.

[20] G. Alipoor and M. Savoji, "Wide-band speech coding using kernel methods and bandwidth extension based on parametric stereo," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 2767–2771.

[21] S. Vaerenbergh and L. Azpicueta-Ruiz, "Kernel-based identification of Hammerstein systems for nonlinear acoustic echo-cancellation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3739–3743.

[22] F. Albu and K. Nishikawa, "New iterative kernel algorithms for nonlinear acoustic echo cancellation," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec. 2015, pp. 734–739.

[23] C. Richard, J.C.M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Nov. 2009.

[24] H. Fan and Q. Song, "A linear recurrent kernel online learning algorithm with sparse updates," *Neural Netw.*, vol. 50, no. 2, pp. 142–153, Feb. 2014.

[25] H. Fan, Q. Song, X. Yang, and Z. Xu, "Kernel online learning algorithm with state feedbacks," *Know.-Based Syst.*, vol. 89, pp. 173–180, Nov. 2015.

[26] B. Chen, S. Zhao, P. Zhu, and J.C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 22–32, Dec. 2012.

[27] B. Chen, S. Zhao, and P. Zhu, "Quantized kernel recursive least squares algorithm.," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1484–1491, May 2013.

[28] S. Zhao, B. Chen, and P. Zhu, "Fixed budget quantized kernel least-mean-square algorithm," *Signal Process.*, vol. 93, no. 9, pp. 2759–2770, Sept. 2013.

[29] S. Nan, L. Sun, B. Chen, Z. Lin, and K. A. Toh, "Density-dependent quantized least squares support vector machine for large data sets," *IEEE Trans.Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 94–106, Jan. 2017.

[30] S. Wang, Y. Zheng, S. Duan, L. Wang, and H. Tan, "Quantized kernel maximum correntropy and its mean square convergence analysis," *Digit. Signal Prog.*, vol. 63, pp. 164 – 176, Jan. 2017.

[31] Steven Van Vaerenbergh, Javier Vía, and Ignacio Santamaría, "Adaptive kernel canonical correlation analysis algorithms for nonparametric identification of wiener and hammerstein systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 875351, Feb. 2008.

[32] S. VanVaerenbergh, J. Vła, and I. Santamarła, "Blind identification of simo wiener systems based on kernel canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2219–2230, May 2013.

[33] P. Pedrosa, J. C. M. Bermudez, and C. Richard, "A design methodology for the gaussian klms algorithm," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug. 2017, pp. 2634–2638.