IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 16, 2019, accepted March 28, 2019, date of publication April 4, 2019, date of current version April 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909304

A Polarized Random Fourier Feature Kernel Least-Mean-Square Algorithm

YUQI LIU, YONGHUI XU, JINGLI YANG, AND SHOUDA JIANG
Department of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150001, China
Corresponding author: Yonghui Xu (xyh@hit.edu.cn)

ABSTRACT This paper presents a polarized random Fourier feature kernel least-mean-square algorithm that aims to overcome the dimension curve of the random Fourier feature kernel least-mean-square (RFFKLMS) algorithm. RFFKLMS is an effective nonlinear adaptive filtering algorithm based on the kernel approximation technique. However, random samples drawn from the distribution need more dimensions to achieve better-generalized performance because they are independent of the training data. To overcome this weakness, a kernel polarization method is adopted to optimize the random samples. Polarized random Fourier features demonstrate a clear advantage over a method without using the polarization method. The experimental results in the context of Lorenz time series prediction and channel equalization verify the effectiveness of the proposed method.

INDEX TERMS Kernel adaptive filtering, kernel polarization method, polarized random Fourier features, random Fourier features.

I. INTRODUCTION

Kernel adaptive filtering (KAF) has been proposed to solve various nonlinear signal processing problems [1]. Kernel least-mean-square (KLMS) is a robust and relatively low-cost method that demonstrates excellent performance in nonlinear filtering [2]. However, the growth of the weight network creates a burden of computation complexity. The quantized kernel least-mean-square (QKLMS) algorithm, which provides an appropriate time/space trade-off with good online performance, was proposed to successfully limit network size growth [3]. Recently, it was verified that the explicit kernel mapping mechanism can effectively overcome the growing sum problem of the KLMS algorithm in root [4]. A reduced Gaussian kernel least-mean-square (RGKLMS) algorithm was proposed based on a Taylor series expansion of the Gaussian kernel function [5]. Furthermore, the random Fourier feature kernel least-mean-square (RFFKLMS) algorithm also demonstrated promising performance in kernel adaptive filtering and attracted wide attention [4], [6].

However, random Fourier features (RFF) face a trade off between accuracy and computational cost for limited-dimension features. The current idea for solving this problem is to achieve a smaller approximation error with a kernel function. Quasi-random sampling is a promising solution in theory, but the smaller approximation error of a kernel function unfortunately does not result in more accurate performance in all cases [7]. The fastfood algorithm has been proposed to speed up computations of random Fourier feature [8]. A Fourier online gradient descent (FOGD) algorithm that applies the random Fourier features for approximating kernel functions was proposed to make the large scale online kernel learning task more efficient [9].

Aiming at the generalization performance of RFFKLMS in various supervised learning scenarios, the kernel polarization method is adopted as a metric of random sample parameters of RFF. The kernel polarization method as a universal kernel optimality criterion proposed by Baram, can measure the consistency between the kernel function and learning task and demonstrated advantages over Euclidean distance [10]. A pre-processing strategy based on kernel polarization is given, in this paper, for evaluating the random samples of RFFKLMS. The optimized RFF is called the polarized random Fourier feature (PRFF). Furthermore, a polarized random Fourier feature kernel least-mean-square (PRFFKLMS) algorithm is developed to effectively scale down the dimension of feature spaces with respect to RFFKLMS. Numerical simulations are utilized to demonstrate the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 reviews the related works on RFFKLMS algorithm.

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

Section 3 proposes the PRFF and the PRFFKLMS algorithm for kernel adaptive filtering. A performance evaluation is discussed in Section 4, and the proposed methods are compared with state-of-the-art methods. Finally, section 5 concludes this paper.

II. A REVIEW OF RFFKLMS

Before the introduction of the main results in Section III, a brief review of RFFKLMS algorithm is presented. The RFFKLMS algorithm overcomes the weight network growth drawback of KLMS [6] and enables kernel adaptive filtering for real-time applications. The main reason is using the RFF to approximate the Gaussian kernel function $:exp(-\frac{||\boldsymbol{x}-\boldsymbol{y}||}{2\sigma^2})$, which is supported by Bochner's theorem from harmonic analysis:

Theorem 1: (Bochner's theorem) A continuous, translation-invariant kernel $k(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x} - \boldsymbol{y})$ is positive definite if and only if the kernel is the Fourier transform of a nonnegative measure.

From Bochner's theorem, we can obtain that the Fourier transform of an appropriately scaled shift-invariant kernel is a probability density [11].

The RFFKLMS algorithm mainly include the following steps: Firt, map the input vector into a high-dimensional feature subspace:

$$\phi(\boldsymbol{x}(i)) = \sqrt{\frac{2}{D}}[\ cos(\boldsymbol{w}_1^T\boldsymbol{x}_i + b_1), ..., cos(\boldsymbol{w}_D^T\boldsymbol{x}(i) + b_D)], \quad (1)$$

where $w_i \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2 I)$, $b_i \overset{\text{i.i.d.}}{\sim} u[0, 2\pi]$, D denotes the dimension of feature mapping vector, and i.i.d. indicates independent and identically distributed. The further adaptive filtering process can be represented as

$$\begin{cases} y_i = \boldsymbol{\Omega}_i^T \phi(\boldsymbol{x}_i) \\ e_i = d_i - y_i \\ \boldsymbol{\Omega}_i = \boldsymbol{\Omega}_{i-1} + \mu e_i \phi_p(\boldsymbol{x}_i) \end{cases} \quad (2)$$

where $\boldsymbol{\Omega}_i$ represents the weight vector, e_i denotes the error, and μ is step size. As we can see from Equ.1, the random parameters of the feature network are generated by random sampling from a distribution. A strategy to optimize the selection of random samples is necessary to improve the performance of the network.

III. PROPOSED PRFFKLMS ALGORITHM

In this paper, we mainly focus on applying the kernel polarization method [10] to optimize the RFF for kernel adaptive filtering. In essence, the polarization strengthens the correspondence between the kernel proximity of inputs and the desired output signal. However, RFF method draws samples from a distribution independent of training data [12]. Generally, for a specific learning task, an ideal kernel should induce a feature subspace matching with that to be learned. Given the known sequence $\{\boldsymbol{x}_i, d_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in X$ is an m-dimensional input vector for the ith iteration and d_i denotes the desired output, the task of kernel adaptive filtering is to learn the input-output relationship $f : X \to R$ under a given sequence. The input-output relationship can be represented as:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \lambda K(\boldsymbol{x}_i, \boldsymbol{x}) \quad (3)$$

Define a kernel as $K_P(\boldsymbol{x}, \boldsymbol{x}') = \int_P p(w)\phi(\boldsymbol{x}, w)\phi(\boldsymbol{x}', w)dw$, where $p(w)$ is a probability distribution that w draws from, and $\phi(\boldsymbol{x})$ denotes explicit feature mapping. In order to measure the kernel similarity between the embedded data induced by a kernel and that induced by the desired signal, the following kernel alignment problem should be considered for getting the best kernel evaluation [13]:

$$maximize \sum_{i,j} K_P(\boldsymbol{x}_i, \boldsymbol{x}_j)d_i d_j. \quad (4)$$

Using RFF [11] and approximating (4) as a discrete sum with samples $\{w_k\}_{k=1}^{N_w} \sim p(w)$, the problem can be rewritten as:

$$maximize \sum_{i,j} d_i d_j \sum_{k=1}^{N_w} \lambda_k \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_k). \quad (5)$$

where $\boldsymbol{\theta}_k = (w_k, b_k)$ denotes the parameter of the kth dimensional feature parameter. The problem is a joint optimization of $\{\lambda_k, \boldsymbol{\theta}_k\}_{k=1}^{N_w}$, which is obviously not a convex problem. Meanwhile, for commonly used RFF-based KLMS algorithms [4], [14] in kernel adaptive filtering, samples $\{w_i\}_{i=1}^{N_w}$ drawn from the distribution $p(w)$ and samples $\{d_i\}_{i=1}^{N_w}$ drawn from a uniform distribution may not be optimal for the given learning task because of the uncertainty of random sampling. A set of optimized $\{\boldsymbol{\theta}_k\}_{k=1}^{N_w}$ is essential to be obtained before the risk function can be a convex problem over $\{\lambda_k\}_{k=1}^{N_w}$.

Here, the optimization aim is represented as

$$maximize \sum_{k=1}^{N_w} \lambda_k \left(\sum_{ij}^{N} d_i d_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_k) \right). \quad (6)$$

Assume $\{\boldsymbol{\theta}_k\}_{k=1}^{N'_w}$ satisfy i.i.d. and that there is a subset $\{\boldsymbol{\theta}_k\}_{k=1}^{N_w}$ of $\{\boldsymbol{\theta}_k\}_{k=1}^{N'_w}$ with the largest value of the object

$$\sum_{ij}^{N} d_i d_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_k). \quad (7)$$

that can maximize (6). Then, the problem is transformed into selecting a subset $\{\boldsymbol{\theta}_k\}_{k=1}^{N_w}$ that can obtain a higher value in computing (7) from $\{\boldsymbol{\theta}_k\}_{k=1}^{N'_w}$. However, direct calculation of (7) will be too complicated. In order to better solve the sample selection problem, an evaluation strategy is given as following.

For i.i.d. samples $\{\boldsymbol{\theta}_k\}_{k=1}^{N'_w}$, the polarization evaluation function as a kernel quality measures is defined as

$$P(\boldsymbol{x}, d, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} d_i \phi(\boldsymbol{x}_i, \boldsymbol{\theta}) \quad (8)$$

Then, the optimization object can be rewritten as

$$\sum_{ij}^{N} P_i(\boldsymbol{\theta}_k)P_j(\boldsymbol{\theta}_k) = \sum_{ij}^{N} d_i d_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_k) \tag{9}$$

where $P_i(\boldsymbol{\theta}_k) = P(\boldsymbol{x}_i, d_i, \boldsymbol{\theta}_k) = d_i\phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)$.

Lemma 1: Given N_w' samples $\{\boldsymbol{\theta}_k\}_{k=1}^{N_w'}$, where $k1, k2 \in \{1, 2, ..., N_w\}$, let

$$\sum_{i=1}^{N_w} P(\boldsymbol{x}_i, d_i, \boldsymbol{\theta}_{k1}) > \sum_{i=1}^{N_w} P(\boldsymbol{x}_i, d_i, \boldsymbol{\theta}_{k2}) \tag{10}$$

Then, it is satisfied that

$$\sum_{ij}^{N} d_i d_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_{k1})\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_{k1}) > \sum_{ij}^{N} d_i d_j \phi(\boldsymbol{x}_i, \boldsymbol{\theta}_{k2})\phi(\boldsymbol{x}_j, \boldsymbol{\theta}_{k2}) \tag{11}$$

The detailed notations of Lemma 1 are exposed in Appendix A. Furthermore, (6) can be realized by selecting N_w samples of $\boldsymbol{\theta}$ with a higher score of the kernel polarization function $P(\boldsymbol{\theta}_k) = \sum_{i=1}^{N_w} d_i\phi(\boldsymbol{x}_i, \boldsymbol{\theta}_k)$ from N_w' random samples, where $N_w' > N_w$. A larger N_w' may benefit to the maximization process because of the expansion of search space.

The polarization process of the random samples is given as follows. 1) Draw N_w'-dimensional random samples $\{\boldsymbol{w}_k\}_{k=1}^{N_w'}$ from a distribution $p(w)$ and $\{\boldsymbol{b}_k\}_{k=1}^{N_w'}$ from a uniform distribution $[0, 2\pi]$, where integer N_w' can be several times the required dimension N_w. For kernel adaptive filtering, a Gaussian kernel is usually the default choice. 2) Intercept the first $rN \in N^+$ pairs of training samples $\{\boldsymbol{x}_i, d_i\}_{i=1}^{rN}$, where N denotes the size of the training set, $r \in (0, 1)$, and $rN \in N^+$. 3) Evaluate N_w' random samples one by one based on the kernel polarization function (10). 4) Select N_w samples with the highest score in the evaluation as the polarized samples.

Based on above preprocessing method, a set of polarized random samples can be obtained for kernel feature mapping. An input vector $\boldsymbol{x} = \{x_1, x_2, ..., x_m\}$ is explicitly mapped to a high-dimensional feature space F under the kernel mapping mechanism of the RFF [11]. The PRFF vector can be defined as

$$\phi_p(\boldsymbol{x}_i) = \sqrt{\frac{2}{D}}\left[cos(\boldsymbol{w}_1^T\boldsymbol{x}_i + \boldsymbol{b}_1), ..., cos(\boldsymbol{w}_{N_w}^T\boldsymbol{x}_i + \boldsymbol{b}_{N_w})\right]. \tag{12}$$

where $\{\boldsymbol{w}_k, \boldsymbol{b}_k\}_{k=1}^{N_w}$ denotes the polarized samples. Then, the output of the adaptive filter can be computed by

$$y_i = \boldsymbol{\Omega}_i^T\phi(\boldsymbol{x}_i) \tag{13}$$

Meanwhile, update the weight vector

$$\boldsymbol{\Omega}_i = \boldsymbol{\Omega}_{i-1} + \mu e_i\phi(\boldsymbol{x}_i) \tag{14}$$

where, e_i denote error vector. Compared with RFFKLMS, the PRFFKLMS algorithm adds a preprocessing step that needs an $O(rNmN_w')$ computation to obtain the evaluation values and an $O(N_w'log(N_w'))$ computation to select N_w samples from N_w' samples.

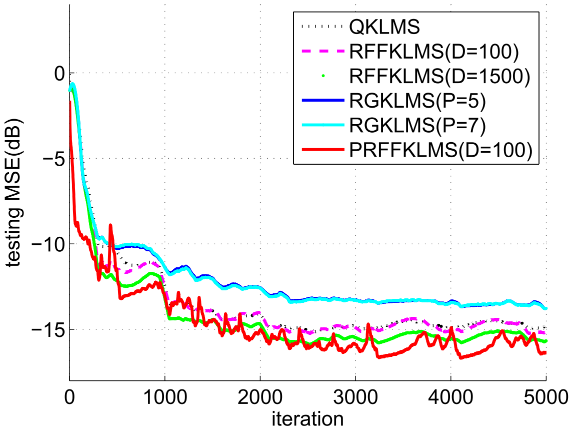

FIGURE 1. Learning curves of the testing MSE in Lorenz time series prediction.

IV. NUMERICAL SIMULATIONS

To demonstrate the advantages of the proposed PRFFKLMS algorithm, numerical simulations are conducted. Without a special mention, the Gaussian kernel is chosen as the default kernel function.

A. LORENZ TIME SERIES PREDICTION

Lorenz time series is a benchmark problem in nonlinear time series prediction, and its model can be represented as the following equations:

$$\begin{aligned}\frac{dx}{dt} &= a(y(t) - x(t))\\ \frac{dy}{dt} &= bx(n) - y(t) - x(t)z(t)\\ \frac{dz}{dt} &= x(t)y(t) - cz(t),\end{aligned} \tag{15}$$

where the model parameter a, b, and c are set as 10, 8/3, and 28, respectively [15]. The time delay of the prediction is chosen as 1, and the time embedding is 5. The number of generated training samples is 5000, and another 200 samples are used as the testing set. The time series are corrupted by white Gaussian noise with zero mean and a variance of 0.01. The kernel parameter σ of RFFKLMS algorithm is set as 1 because it can get the best MSE performance and all σ value of comparison algorithms are set the same. The step size of RGKLMS is chosen as 0.005 to ensure convergence and the step size of other algorithms are 0.1. The parameter γ of QKLMS is set to 0.4. The dimension of PRFFKLMS algorithm is chosen as 100. Taken into account both the MSE performance of PRFFKLMS and the complexity in preprocessing stage, the number of candidate samples N_w' is chosen as 10 times the dimension N_w, and r is set 0.1.

The learning curves of the testing MSE averaged over 100 independent Monte Carlo runs are shown in Fig.1. The final testing MSE of QKLMS, RFFKLMS (D = 100), RFFKLMS (D = 1500), RGKLMS (P = 5), RGKLMS (P = 7), and PRFFKLMS(D = 100) are −14.90 dB, −15.18 dB, −15.64 dB, −13.54 dB, −13.58 dB, and

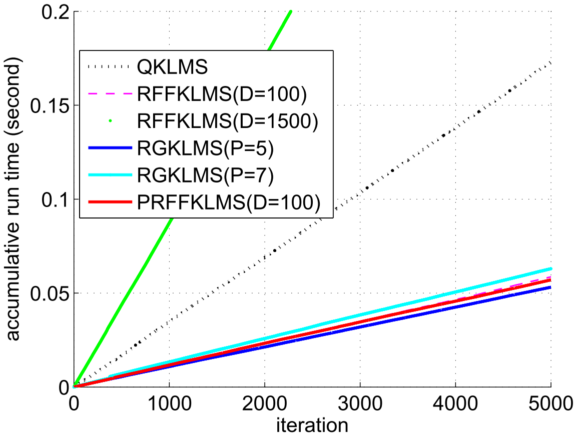

FIGURE 2. The accumulated runtimes of all the compared algorithms in Lorenz time series prediction.

−15.90 dB, respectively. The accuracy and convergence speed of the proposed PRFFKLMS algorithm is much better than the RFFKLMS algorithm, which demonstrates the effectiveness of proposed method. Although the accuracies of PRFFKLMS (D = 100) and RFFKLMS (D = 1500) are similar, PRFFKLMS demonstrates better coverage speed. As shown in Fig. 2, after 5000 iterations, the accumulated runtimes of QKLMS, RFFKLMS (D = 100), RFFKLMS (D = 1500), RGKLMS (P = 5), RGKLMS (P = 7), and PRFFKLMS(D = 100) are 0.173 s, 0.055 s, 0.441 s, 0.053 s, 0.063 s, 0.057 s, respectively. The overall performance of the proposed algorithm is better than that of the RGKLMS algorithm and RFFKLMS algorithm.

B. TIME-VARYING CHANNEL EQUALIZATION

This example aims to assess the effectiveness of the proposed method in channel equalization applications. We consider a time-varying nonlinear channel that is commonly encountered in a real satellite communication system [16]. The linear channel impulse response is $h = [h_0, h_1, h_2]^T$, and it's value is chosen as $h = [0.3482, 0.8704, 0.3482]$. The transfer function of the time-varying channel model is described as

$$H(z) = (h_0 + h_0(j)) + (h_1 + h_1(j))z^{-1} + (h_2 + h_2(j))z^{-2} \tag{16}$$

where $h_0(j)$, $h_1(j)$, and $h_2(j)$ denote the time-varying coefficients generated by using a second-order Markov model [17]. The nonlinear channel output is defined by

$$r(n) = x(n) + 0.2x^2(n) + v(n) \tag{17}$$

where $v(n)$ is 20 dB of white Gaussian noise. The time embedding is 5. For all the compared algorithms, 100 independent runs employing 6000 training samples and 200 test samples are performed. for QKLMS, RFFKLMS, and PRFFKLMS. In order to get better accuracy, the kernel parameter σ is set to be 0.6 for RGKLMS and is set as 2 for other algorithms. The step size is chosen as 0.06 for RGKLMS to ensure stable convergence and is set as 0.3 for other algorithms. The parameter γ of QKLMS is 0.1. The RFFKLMS algorithms with dimensions of 100 and 1000 are simulated for comparison. Considering both the MSE performance of PRFFKLMS and the complexity of preprocessing computation, the number of candidate samples N'_w is 10 times the dimension N_w, and r is set 0.2. Both orders P = 5 and P = 7 RGKLMS algorithm are used as comparison algorithms.

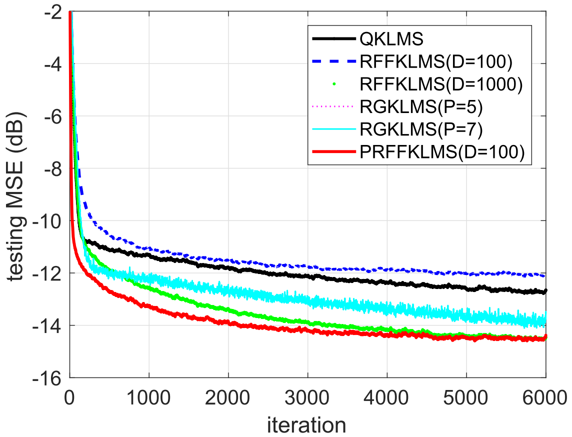

FIGURE 3. Learning curves of the testing MSE in channel equalization.

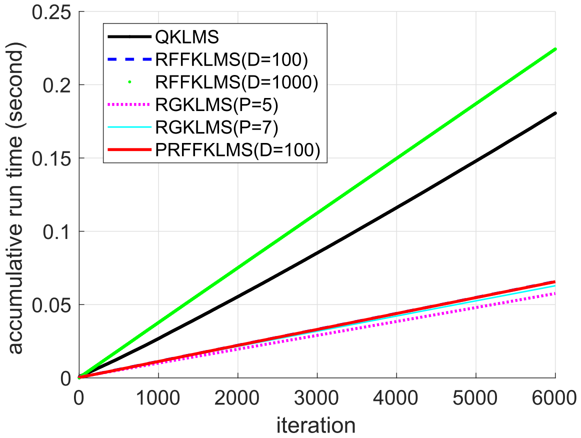

FIGURE 4. The accumulated runtimes of all compared algorithms in channel equalization.

Fig.3 shows that the proposed algorithm has an improvement in terms of both convergence speed and steady-state MSE. The final testing MSE of QKLMS, RFFKLMS (D = 100), RFFKLMS (D = 1000), RGKLMS (P = 5), RGKLMS (P = 7), and PRFFKLMS (D = 100) are −12.73 dB, −12.08 dB, −14.51 dB, −13.83 dB, −13.83 dB, and −14.52 dB, respectively. The learning curve for RGKLMS(P = 5) is the same with that of RGKLMS(P = 7). As displayed in Fig.4, after 5000 iterations, the accumulated runtimes of all the algorithms are 0.181 s, 0.069 s, 0.224 s, 0.058 s, 0.063 s, 0.066 s, respectively. It is found that the proposed PRFFKLMS algorithm improves the computational efficiency compared with the RFFKLMS algorithm and obtains better accuracy than the RGKLMS algorithm under similar computation complexity.

TABLE 1. **The final MSEs in time varying channel equalization with different SNR setting.**

SNR	QKLMS	RFFKLMS (D=100)	RFFKLMS (D=1000)	RGKLMS (P=5)	RGKLMS (P=7)	PRFFKLMS (D=100)
10 dB	-3.42 dB	-2.95 dB	-3.48 dB	-1.29 dB	-1.23 dB	-3.47 dB
12 dB	-5.07 dB	-4.48 dB	-5.15 dB	-3.05 dB	-3.06 dB	-5.13 dB
14 dB	-7.19 dB	-6.35 dB	-7.30 dB	-5.26 dB	-5.47 dB	-7.28 dB
16 dB	-9.39 dB	-8.22 dB	-9.62 dB	-7.91 dB	-8.35 dB	-9.47 dB
18 dB	-11.46 dB	-10.18 dB	-12.27 dB	-10.76 dB	-11.29 dB	-12.10 dB
20 dB	-12.73 dB	-12.08 dB	-14.51 dB	-13.83 dB	-13.83 dB	-14.52 dB

In order to further evaluate the proposed algorithm in lower SNR situations, the further simulations with SNR from 12 to 20 dB are conducted. The MSE results of all compared algorithms are shown in Table. 1. In various SNR environments, the MSE obtained by proposed PRFFKLMS (D $=$ 100) algorithm can always be less than QKLMS, RFFKLMS (D $=$ 100), RGKLMS (P $=$ 5,7) and almost the same performance with RFKLMS ($D = 1000$).

V. CONCLUSION

In this paper, the kernel polarization metric of random Fourier features is proposed and named polarized random Fourier features. Based on the PRFF, the PRFFKLMS algorithm is developed for kernel adaptive filtering. The results of an extensive simulation experiment confirm the effectiveness of proposed method. As a future work, we will seek to realize this algorithm in hardware systems and carry out research in practical application.

VI. APPENDIX: PROOF OF LEMMA 1

If it is satisfied that

$$\sum_{i=1}^{N_w} f(\boldsymbol{x}_i, d_i, \boldsymbol{\theta}_1) > \sum_{i=1}^{N_w} f(\boldsymbol{x}_i, d_i, \theta_2) \tag{18}$$

then, we have

$$\begin{aligned} &\sum_{i=1}^{N_w}\sum_{j=1}^{N_w} f(\boldsymbol{x}_i, d_i, \theta_1) f(\boldsymbol{x}_j, d_j, \theta_1) \\ &\quad > \sum_{i=1}^{N_w} f(\boldsymbol{x}_i, d_i, \theta_1) \left(\sum_{j=1}^{N_w} f(\boldsymbol{x}_j, d_j, \theta_2) \right) \\ &\quad > \sum_{i=1}^{N_w} f(\boldsymbol{x}_i, d_i, \theta_2) \left(\sum_{j=1}^{N_w} f(\boldsymbol{x}_j, d_j, \theta_2) \right) \end{aligned} \tag{19}$$

REFERENCES

[1] S. Haykin W. Liu, and J. C. Principe, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, NJ, USA: Wiley 2010.

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

[3] 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.

[4] A. Singh, N. Ahuja, and P. Moulin,"Online learning with kernels: Overcoming the growing sum problem," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2012, pp. 1–6.

[5] Y. Liu, C. Sun, and S. Jiang,"A reduced Gaussian kernel least-mean-square algorithm for nonlinear adaptive signal processing," *Circuits Syst. Signal Process.*, vol. 38, no. 1, pp. 371–394, Jan. 2019.

[6] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in RKH spaces using random Fourier features," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1920–1932, Apr. 2018.

[7] H. Avron, V. Sindhwani, J. Yang, and M. W. Mahoney, "Quasi-monte carlo feature maps for shift-invariant kernels," in *Proc. 31st Int. Conf. Mach. Learn.*, Jan. 2014, Dec. 2014

[8] Q. V. Le, T. Sarlós, and A. Smola, "Fastfood—Computing hilbert space expansions in loglinear time," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 2013, pp. 244–252.

[9] J. Lu, S. C. H. Hoi, J. Wang, P. Zhao, and Z. Y. Liu,"Large scale online kernel learning," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1613–1655, Jan. 2016.

[10] Y. Baram, "Learning by kernel polarization," *Neural Comput.*, vol. 17, no. 6, pp. 1264–1275, Jun. 2005.

[11] A. Rahimi and B. Recht,"Random features for large-scale kernel machines," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2007, pp. 1177–1184.

[12] T. Yang, Y. F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou,"Nyström method vs random fourier features: A theoretical and empirical comparison," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2012, pp. 476–484.

[13] A. Sinha and J. Duchi,"Learning kernels with random features," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.* Barcelona, Spain: Curran Associates, Dec. 2016, pp. 1306–1314.

[14] P. Bouboulis, S. Pougkakiotis, and S. Theodoridis,"Efficient KLMS and KRLS algorithms: A random Fourier feature perspective," in *Proc. IEEE Statist. Signal Process. Workshop*, Jun. 2016, pp. 1–5.

[15] A. Miranian and M. Abdollahzade,"Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 207–218, Feb. 2013.

[16] J. Choi, A. C. C. Lima, and S. Haykin,"Kalman filter-trained recurrent neural equalizers for time-varying channels," *IEEE Trans. Commun.*, vol. 53, no. 2, p. 380, Feb. 2005.

[17] Q. Liang and J. M. Mendel,"Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 551–563, Oct. 2000.

YUQI LIU received the B.S. and M.S. degrees from the Harbin Institute of Technology, China, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information Engineering. His current research interests include adaptive signal processing and kernel learning.

YONGHUI XU received the M.Sc. degree from the Harbin Institute of Technology, Harbin, China, in 2002, and the Ph.D. degree from the Beijing Institute of Technology, Beijing, China. He is currently a Lecturer with the School of Electrical Engineering and Test Automation, Harbin Institute of Technology. His current research interests include signal processing, automatic test, and fault diagnosis.

JINGLI YANG was born in 1984. He received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2013, where he is currently an Assistant Professor with the School of Electrical Engineering and Test Automation. His current research interests include adaptive filter, network communication, and fault diagnosis.

SHOUDA JIANG received the Ph.D. degree in instrument science and technology from the Harbin Institute of Technology, Harbin, China, in 2003, where he is currently a Professor with the School of Electrical Engineering and Test Automation. His current research interests include automatic test and control technology, fault diagnosis technology, and digital signal processing.