

Time series prediction using kernel adaptive filter with least mean absolute third loss function

Lu Lu · Haiquan Zhao · Badong Chen

Received: 21 January 2017 / Accepted: 27 July 2017 / Published online: 7 August 2017
© Springer Science+Business Media B.V. 2017

Abstract In this paper, a novel kernel adaptive filter, based on the least mean absolute third (LMAT) loss function, is proposed for time series prediction in various noise environments. Combining the benefits of the kernel method and the LMAT loss function, the proposed KLMAT algorithm performs robustly against noises with different probability densities. However, an important limitation of the KLMAT algorithm is a trade-off between the convergence rate and steady-state prediction error imposed by the selection of a certain value for the learning rate. Therefore, a variable learning rate version (VLR–KLMAT algorithm) is also proposed based on a *Lorentzian function*. We analyze the stability and convergence behavior of the KLMAT algorithm and derive a sufficient condition to predict its learning rate behavior. Moreover, a kernel recursive extension of the KLMAT algorithm is further proposed for performance improvement. Simulation results in the context of time series prediction verify the effectiveness of the proposed algorithms.

Keywords Kernel adaptive filter · Least mean absolute third algorithm · Variable learning rate (VLR) · Recursive algorithm · Series prediction

1 Introduction

1.1 Previous work

The least mean square (LMS) algorithm is the most widely used algorithm for adaptive filters because of its low-cost. However, it suffers from slow convergence rate and its performance may degrade when the measurement noise is non-Gaussian noise. To address these problems, several LMS-type algorithms were proposed, including sign algorithms [22,39], adjustable step size algorithms [16,21,62], and algorithms that employ the convex combination method [19,32,36]. Apart from these algorithms, some approaches have been developed for combating non-Gaussian interference, such as entropy criterion [24,25,38] and M-estimate [66,67].

An important issue in adaptive prediction is the effect of measurement noise on the results. The measurement noise is often assumed to be a random process with finite second-order statistics (SOS), making the mean square error (MSE) performs well for prediction. However, in real-world situations, the noise may not possess finite SOS and can be described more accurately using non-Gaussian noise models [24,57]. When adaptive algorithms are used in time series pre-

L. Lu · H. Zhao (✉)
School of Electrical Engineering, Southwest Jiaotong
University, Chengdu, China
e-mail: hqzhao_swjtu@126.com

L. Lu
e-mail: lulu@my.swjtu.edu.cn

B. Chen
School of Electronic and Information Engineering,
Xi'an Jiaotong University, Xi'an, China
e-mail: chenbd@mail.xjtu.edu.cn

diction with non-Gaussian noise, it is reasonable to take the higher order information of measurement noise and error signal into account.

The loss functions based on the high-order error power (HOEP) criterion are a good solution to learn with non-Gaussian data in general [45]. The HOPE-based algorithms, which are derived by minimizing the p -norm, can achieve improved performance in the presence of non-Gaussian noise. For $p = 2$, it reduces to the LMS algorithm. When the signal is contaminated by impulsive noise, the sign algorithm with $p = 1$ is preferred [22,39]. If we select $p = 3$, the HOEP becomes the least mean absolute third (LMAT) algorithm [23,45,65], and for the choice of $p = 4$, the least mean-fourth (LMF) algorithm is obtained [18,48]. It is worth to note that both LMAT and LMF algorithms outperform the conventional LMS algorithm for the case where the measurement noise is a non-Gaussian noise. Furthermore, many variants based on the mixed-norm and p -power ($1 < p < 2$) were proposed for suppressing specific noises, such as those proposed in [5,8,9,31,33,43] and references therein.

Kernel method has become rather popular and has been successfully applied to machine learning [47,51], kernel principal component analysis [46,63], and information/signal processing [2–4,7,20,29,64]. Due to its universal nonlinear-modeling capability, the kernel adaptive filters (KAFs) have attracted considerable attention. The main idea of the KAF is that recast the input data into a high-dimensional feature space via a reproducing kernel Hilbert space (RKHS). Then, the linear adaptive filter is applied in the feature space [29]. Based on these considerations, several kernel adaptive algorithms were proposed, e.g., kernel LMS (KLMS) algorithms [13,27,38,55,58,60], kernel recursive least squares (KRLS) algorithms [14,17,26], and kernel affine projection (KAPAs) [10,28,54]. By applying the minimum error entropy (MEE) criterion to KAF, Chen et al. proposed the kernel minimum error entropy (KMEE) algorithm and its quantized version [11]. As we noticed, the KMEE algorithm has moderate computational complexity and achieves improved performance in low impulsive noise environment. To further improve the performance of KAF in the presence of α -stable noise, Chen et al. proposed some KAFs based on maximum correntropy criterion (MCC) [52,59], which can diminish the significance of the outliers on the nonlinear system. Alternatively, to improve

the performance of the KAF, an interesting and effective way is to use p -norm and mixed-norm criterions. Several classes of KAFs for nonlinear estimation have been proposed, including [30,35,37,41,49]. Particularly, Ma et al. [37] developed the kernel least mean p -power (KLMP) and the kernel recursive least mean p -power (KRLP) algorithms, which overcome the performance degradation of the algorithm when training data are corrupted by impulsive noises. However, the adaptation of KLMP and KRLP is largely dependent on p , and it is necessary for the prior knowledge or estimation of p in the presence of α -stable noise. Very recently, the KAFs based on diffusion LMS and suitable for nonlinear distributed network were proposed [15,42]. These algorithms follow the Adapt-Then-Combine (ATC) mode of cooperation and can be extended to the distributed parameter estimation applications like cooperative spectrum sensing and massive multiple input multiple output (MIMO) receiver design [42]. Although the above-mentioned algorithms have some advantages, they carry one main drawback in common: the algorithms are not suitable for different noise environments.

1.2 Motivation

In the last decades, the fields of adaptive signal processing have witnessed remarkable advances in cost function based on the HOEP criterion. Closer to the LMAT loss is the LMF function (LMF algorithm) [48], which minimizes the fourth power of the error signal, but its stability about the Wiener solution depends upon the adaptive filter input power and noise power [48]. Moreover, the LMF algorithm with Gaussian noise is not mean square stability even for a small step size. The LMAT loss function, in contrast, has a stable performance in Gaussian scenarios and its convergence performance only depends on the power of the input signal [65]. The advantages of using the LMAT loss function are listed as below. (1) The LMAT loss is a convex function, so it has no local minima. (2) When the measurement noise is non-Gaussian processes, the LMAT may have better optimum solution than the Wiener solution. Another closer to the LMAT loss is the work of Chambers et al. [9] which combines the error norms and can deal with non-stationary signal statistics through an appropriate combination. In [41], Miao and Li proposed the kernel least mean mixed-

Table 1 Aforementioned contributions

Cost function used in kernel method (criterion)	Proposed algorithm	Comments
MSE and least squares (LS) criterion (2-norm)	KLMS Liu et al [27] KAPA Liu and Principe [28] KRLS Engel et al. [17]	Not suitable for non-Gaussian environment
HOPE criterion		
LMAT loss function	This study ^a	Perform robustly against noise with different probability densities
LMF loss function	–	Unstable in Gaussian environment
Mixed-norm criterion		
Least mean mixed-norm (LMMN)	KLMMN Miao and Li [41]	The selection of the mixing parameter of the KLMMN algorithm may prohibit its practical applications
Robust mixed-norm (RMN)	KRMN Liu et al. [30] QKRMN and VPKRMN Lu et al. [35]	Only applicable to impulsive noise environment
Sign algorithm (1-norm)	KAPSA Wang et al. [49]	Slow convergence rate and only applicable to impulsive noise environment
p -Power criterion ($1 < p < 2$)	KLMP and KRLP Ma et al. [37]	Only applicable to impulsive noise environment

^a Part of this work appeared in [arXiv:1603.03564](https://arxiv.org/abs/1603.03564) ‘L. Lu, H. Zhao, B. Chen, KLMAT: A kernel least mean absolute third algorithm’ [34]

norm (KLMMN) algorithm for nonlinear system identification. However, the selection of the mixing parameter of the KLMMN algorithm may prohibit its practical applications. Table 1 summarizes the aforementioned contributions. According to this table, the KAF based on LMAT loss has not been developed. From the above analysis, we can see that the use of LMAT loss in kernel method is reasonable.

1.3 Contributions of this paper

The contribution of this paper is threefold. (1) To improve the robustness of KAF against interferences with various probability densities, we proposed a kernel LMAT (KLMAT) algorithm. Moreover, the stability and convergence property analysis of the KLMAT algorithm are performed. (2) To address the conflicting requirement of fast convergence rate and low steady-state prediction error for the fixed learning rate, a novel variable learning rate (VLR) adjustment process based on the *Lorentzian function* is incorporated into the KLMAT algorithm. (3) The recursive version of the KLMAT algorithm is developed for time series predic-

tion. The motivation in developing this kernel extension is based on the KRLS algorithm. The weights in the kernel at each iteration are solved recursively by an exponentially weighted mechanism, which emphasizes on recent data and de-emphasizes data from the remote past.

This paper is structured as follows. In Sect. 2, we briefly review the kernel method. In Sect. 3, a KLMAT and a VLR–KLMAT algorithms are proposed based on the kernel method. In Sect. 4, we analyze the convergence property of the KLMAT algorithm. In Sect. 5, an extension of KLMAT is developed. In Sect. 6, we show the advantages of our proposal through some simulation results. Finally, in Sect. 7, we give the conclusion.

2 Kernel method

Kernel method is a powerful nonparametric modeling tool. The key to kernel method is transforming input data (input space \mathbb{U}) into a high-dimensional feature space \mathbb{F} by using a certain nonlinear mapping

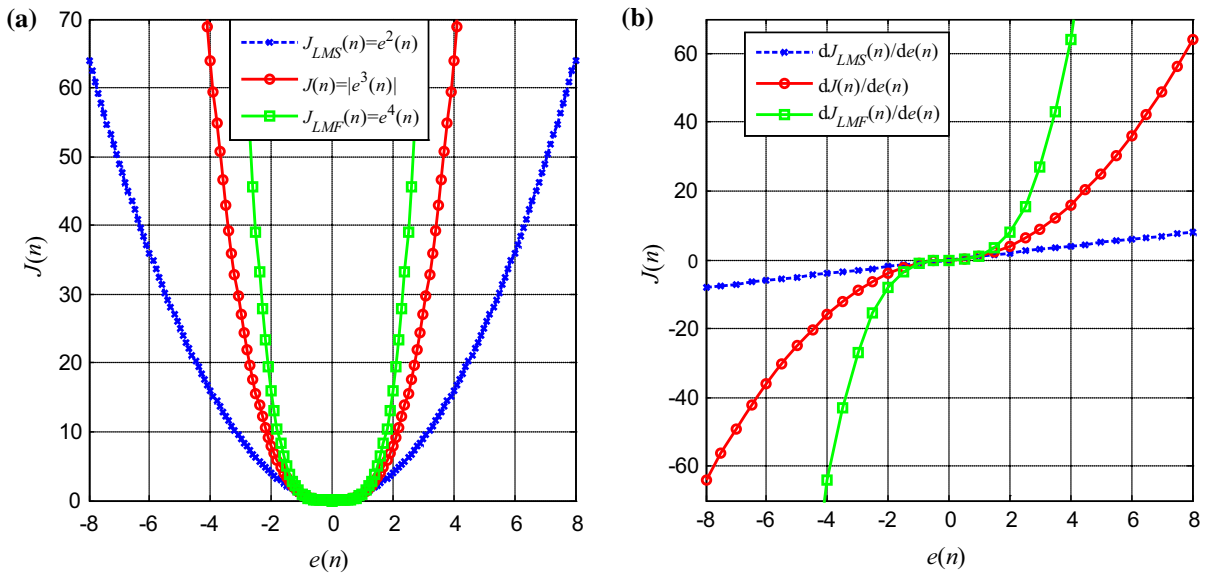


Fig. 1 a Comparison of the cost functions. b The gradients of the cost functions

$$\varphi : \mathbb{U} \rightarrow \mathbb{F} \tag{1}$$

where φ is the feature vector in kernel method. To apply the kernel method in linear adaptive filter, the kernel function κ is developed. As a result, the inner product operations in the linear adaptive filters are translated into the calculation of a kernel function κ in the feature space without knowing the nonlinear mapping.

By using the Mercer theorem [29], the inner products in RKHS can be calculated as

$$\kappa(\mathbf{u}, \mathbf{u}') = \varphi(\mathbf{u})\varphi^T(\mathbf{u}') \tag{2}$$

where \mathbf{u} is the input data. It is well known that Mercer kernel is a continuous, symmetric, and positive-definite kernel. Thus, the output of KAF can be expressed by the inner product with test data $\varphi(\mathbf{u})$ and training data $\varphi(\mathbf{u}_j)$

$$f(\mathbf{u}) = \sum_{j=1}^n a_j \langle \varphi(\mathbf{u}), \varphi(\mathbf{u}_j) \rangle \tag{3}$$

where a_j is the coefficient n and $\langle \cdot \rangle$ is the inner product operation, respectively.

3 Proposed algorithms

3.1 KLMAT algorithm

To improve the performance of the KLMS algorithm, the LMAT algorithm is first applied in RKHS. This strategy yields a novel KAF (KLMAT) for adaptive prediction. The $M \times 1$ input data $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-1+M)]$ at time n are transformed into RKHS as $\varphi(\mathbf{u}(n))$. For convenience of notation, $\varphi(\mathbf{u}(n))$ is replaced by $\varphi(n)$ throughout this paper. The weight vector $\mathbf{w}(n)$ in feature space is defined as $\Omega(n)$. Define $\Omega(1) = \mathbf{0}$ and the error signal $e(n) = d(n) - \Omega^T(n)\varphi(n)$, where $d(n)$ denotes the desired signal. The cost function of KLMAT algorithm is defined as

$$J(n) \triangleq |e(n)|^3. \tag{4}$$

As shown in Fig. 1a, $J(n)$ is less steep than $J_{LMF}(n)$ and both are steeper than a quadratic function. Because the cost function $J(n)$ is much steeper than the square of the error, the value of the gradient is significantly larger than that of the gradient of the squared error with respect to the coefficients (Fig. 1b). Therefore, the kernel algorithm with LMAT loss function converges faster than the KLMS algorithm for a given constant learning rate.

Minimizing the instantaneous third power of absolute error value, the adaptation of KLMAT algorithm in RKHS can be expressed as

$$\mathbf{\Omega}(n + 1) = \mathbf{\Omega}(n) - \frac{\mu}{3} \nabla_{\mathbf{\Omega}(n)} J(n) \tag{5}$$

where $\nabla_{\mathbf{\Omega}(n)} J(n) = -3e^2(n) \text{sign}\{e(n)\} \boldsymbol{\varphi}(n)$ is the gradient vector, μ is the learning rate (step size), and $\text{sign}\{x\}$ denotes the sign function of the variable x , i.e., if $x \geq 0$, then $\text{sign}\{x\} = 1$, otherwise $\text{sign}\{x\} = -1$. Thus, we can use (5) to obtain a recursion on the new example sequence $\{\boldsymbol{\varphi}(n), d(n)\}$

$$\mathbf{\Omega}(n + 1) = \mathbf{\Omega}(n) + \mu e^2(n) \text{sign}\{e(n)\} \boldsymbol{\varphi}(n). \tag{6}$$

Repeating the application of (6), we obtain

$$\begin{aligned} \mathbf{\Omega}(n + 1) &= \mathbf{\Omega}(n - 1) + \mu e^2(n - 1) \\ &\quad \times \text{sign}\{e(n - 1)\} \boldsymbol{\varphi}(n - 1) \\ &\quad + \mu e^2(n) \text{sign}\{e(n)\} \boldsymbol{\varphi}(n). \end{aligned} \tag{7}$$

Rearranging (7), we have

$$\mathbf{\Omega}(n + 1) = \mu \sum_{j=1}^n \left[e^2(j) \text{sign}\{e(j)\} \right] \boldsymbol{\varphi}(j). \tag{8}$$

Here, $\boldsymbol{\varphi}(n)$ is only implicitly known and its dimensionality is infinite for the Gaussian kernel. For this reason, the derivation method of the KLMS algorithm [27] is adopted to the KLMAT algorithm. That is, compute the filter output $y(n + 1)$ directly rather than expressing the weight vector. By using the Mercer kernel, the filter output can be calculated through kernel evaluations

$$\begin{aligned} y(n + 1) &= \mathbf{\Omega}^T(n) \boldsymbol{\varphi}(n + 1) \\ &= \mu \sum_{j=1}^n \left[e^2(j) \text{sign}\{e(j)\} \right] \kappa(j, n + 1) \end{aligned} \tag{9}$$

where $\kappa(\mathbf{u}, \mathbf{u}') = \exp\left(-h \|\mathbf{u} - \mathbf{u}'\|^2\right)$ stands for the Gaussian kernel and h denotes the kernel size. The KLMAT algorithm adds a new space $e(n) \text{sign}\{e(n)\}$ for $\mathbf{u}(n + 1)$ at each iteration, slightly increasing the computational complexity as compared with the KLMS algorithm. We define f_n as a nonlinear mapping at the n th iteration, and the learning process of the KLMAT algorithm can be summarized as follows:

$$\begin{aligned} f_n &= \mu \sum_{j=1}^n \left[e^2(j) \text{sign}\{e(j)\} \right] \kappa(\mathbf{u}(j), \cdot), \\ f_n(\mathbf{u}(n)) &= \mu \sum_{j=1}^n \left[e^2(j) \text{sign}\{e(j)\} \right] \kappa(\mathbf{u}(j), \mathbf{u}(n + 1)), \end{aligned}$$

$$\begin{aligned} e(n + 1) &= d(n + 1) - f_n(\mathbf{u}(n + 1)), \\ f_{n+1} &= f_n + \mu e^2(n + 1) \text{sign}\{e(n + 1)\} \\ &\quad \times \kappa(\mathbf{u}(n + 1), \cdot). \end{aligned} \tag{10}$$

From (10), it can be observed that if the kernel function is replaced by a radial kernel, the KLMAT algorithm reduces to the radial basis function (RBF) network by allocating a new kernel unit for every new example with input. For simplicity, the coefficient $a_j(n + 1)$ is defined as:

$$a_j(n + 1) = \mu \left[e^2(j) \text{sign}\{e(j)\} \right], \quad j = 1, \dots, n + 1 \tag{11}$$

and

$$\mathbf{C}(n + 1) = [\mathbf{C}(n), \mathbf{u}(n + 1)] \tag{12}$$

where $\mathbf{C}(n) = \{\mathbf{c}_j\}_{j=1}^n$ is the center set or dictionary which stores the new center at each iteration. For $n=1$, $\mathbf{C}(1) = [\mathbf{u}(1)]$.

3.2 VLR–KLMAT algorithm

An important limitation of the KLMAT algorithm is the convergence rate vs. misadjustment trade-off imposed by the selection of a certain value for the learning rate. Motivated by the VLR scheme and the Lorentzian function in [6], in this section, we proposed a novel VLR scheme for the KLMAT algorithm. Replacing μ with $\mu(n)$ for each iteration, $\mu(n)$ is adapted by using the following expression

$$\mu(n) = \beta \log \left(1 + \frac{1}{2} \frac{e^2(n)}{l^2} \right) \tag{13}$$

where β is a scalar factor which controls the value range of the function and l is the positive parameter. The large value of β leads to fast convergence rate at the initial stage and high misadjustment. By using some nonlinear function, the VLR scheme has been developed in several previous studies, including Sigmoid function [53] and Versiera function [56]. It can be observed from Fig. 2 that the Lorentzian function is much steeper than the other functions for the same small error signal. Therefore, such function can achieve fast convergence rate and improved tracking capability.

To further improve the performance of the VLR scheme, an estimation of $e^2(n)$ is introduced to (13)

$$\delta_e(n + 1) = \theta \delta_e(n) + (1 - \theta) e^2(n) \tag{14}$$

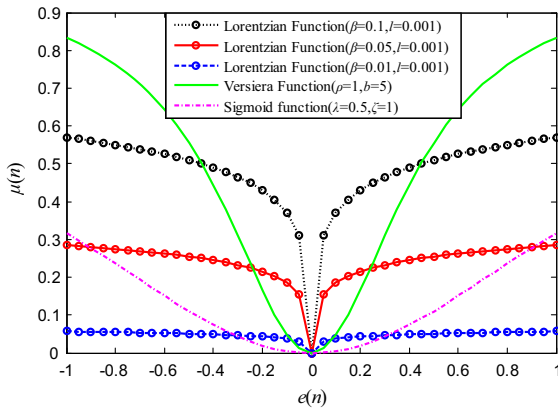


Fig. 2 Comparison of Lorentzian function, Versiera function, and Sigmoid function

where θ is the forgetting factor that governs the averaging time constant and $\delta_e(n + 1)$ is a low-pass filtered estimation of $e^2(n)$. In stationary environments, the previous samples include information that is relevant to determining a measure of update, i.e., the proximity of the adaptive filter coefficients to the optimal ones. Hence, θ is close to 1. We set to $\theta = 0.9$ for the VLR–KLMAT algorithm. Moreover, for the stability of the VLR strategy, $\mu(n)$ is further limited by

$$\mu(n + 1) = \begin{cases} \mu_{\max}, & \mu(n) > \mu_{\max} \\ \mu_{\min}, & \mu(n) < \mu_{\min} \\ \mu(n), & \text{otherwise} \end{cases} \quad (15)$$

where $\mu_{\max} = 2$ and $\mu_{\min} = 0.01 (0 < \mu_{\min} < \mu_{\max})$.

Remark 1 $\mu_{\max} = 2$ is normally selected near the point of instability of the algorithm to provide the maximum possible convergence rate and $\mu_{\min} = 0.01$ is chosen as a trade-off between the steady-state prediction error and the tracking capabilities of the algorithm [1, 12].

4 Performance analysis

The convergence analysis of the algorithm is performed in this section. For tractable analysis, the following assumptions are made:

- (A1) The measurement noise $v(n)$ is zero mean, independent, identically distributed (i.i.d.) and is independent of input $\varphi(n)$. The variance of measurement noise is σ_v^2 .
- (A2) The a priori error $e_a(n)$ is zero mean and independent of the noise $v(n)$.

The above assumptions have been successfully used in analyzing KAFs [12, 44, 50]. It can solve the problem of calculating expected values of some expressions involving contaminated Gaussian noise which is often used to model interference environments in the literature.

Consider the desired response arising from the model

$$d(n) = \mathbf{\Omega}_o^T \varphi(n) + v(n) \quad (16)$$

where $\mathbf{\Omega}_o$ denotes the optimal weight vector.

The error weight vector $\mathbf{V}(n)$ is defined as

$$\mathbf{V}(n) = \mathbf{\Omega}_o - \mathbf{\Omega}(n). \quad (17)$$

Thus, the error signal of the algorithm can be expressed as

$$\begin{aligned} e(n) &= d(n) - \mathbf{\Omega}^T(n) \varphi(n) \\ &= \mathbf{V}^T(n) \varphi(n) + v(n). \end{aligned} \quad (18)$$

Define the *a priori* error and *a posteriori* error of the KLMAT algorithm, respectively, as

$$e_a(n) = \mathbf{V}^T(n) \varphi(n) \quad (19)$$

and

$$e_a(n) = \mathbf{V}^T(n + 1) \varphi(n). \quad (20)$$

Combining (19) and (20), we have

$$\begin{aligned} \mathbf{V}(n + 1) - \mathbf{V}(n) &= [e_p(n) - e_a(n)] / \varphi(n) \\ &= [e_p(n) - e_a(n)] \varphi(n) / \kappa(\mathbf{u}(n), \mathbf{u}(n)). \end{aligned} \quad (21)$$

Squaring both sides of (21), we have the energy conservation relation (ECR) for KLMAT:

$$\begin{aligned} \|\mathbf{V}(n + 1)\|_{\mathbb{F}}^2 + \frac{e_a^2(n)}{\kappa(\mathbf{u}(n), \mathbf{u}(n))} &= \|\mathbf{V}(n)\|_{\mathbb{F}}^2 \\ + \frac{e_p^2(n)}{\kappa(\mathbf{u}(n), \mathbf{u}(n))} \end{aligned} \quad (22)$$

where $\|\mathbf{V}(n)\|_{\mathbb{F}}^2 \triangleq \mathbf{V}^T(n) \mathbf{V}(n)$ denotes the weight error power in \mathbb{F} . Taking expectations of both sides of (22), we have

$$\begin{aligned} \mathbb{E} \left[\|\mathbf{V}(n + 1)\|_{\mathbb{F}}^2 \right] + \mathbb{E} \left[\frac{e_a^2(n)}{\kappa(\mathbf{u}(n), \mathbf{u}(n))} \right] \\ = \mathbb{E} \left[\|\mathbf{V}(n)\|_{\mathbb{F}}^2 \right] + \mathbb{E} \left[\frac{e_p^2(n)}{\kappa(\mathbf{u}(n), \mathbf{u}(n))} \right] \end{aligned} \quad (23)$$

where $\mathbb{E}[\cdot]$ stands for taking expectation.

Substituting (19) and (20) into (23), the ECR can be given as

$$\begin{aligned} E \left[\|\mathbf{V}(n+1)\|_{\mathbb{F}}^2 \right] &= E \left[\|\mathbf{V}(n)\|_{\mathbb{F}}^2 \right] \\ &\quad - 2\mu E [e_a(n) f(e(n))] \\ &\quad + \mu^2 E \left[\kappa(\mathbf{u}(n), \mathbf{u}(n)) f^2(e(n)) \right] \end{aligned} \tag{24}$$

where $f(e(n)) = e^2(n) \text{sign}\{e(n)\}$ is the error function. For Gaussian kernel $\kappa(\mathbf{u}(n), \mathbf{u}(n)) \equiv 1$, we obtain

$$\begin{aligned} E \left[\|\mathbf{V}(n+1)\|_{\mathbb{F}}^2 \right] &= E \left[\|\mathbf{V}(n)\|_{\mathbb{F}}^2 \right] \\ &\quad - 2\mu E [e_a(n) f(e(n))] + \mu^2 E [f^2(e(n))]. \end{aligned} \tag{25}$$

Hence, the weight vector in the KLMAT algorithm can converge if and only if

$$\begin{aligned} E \left[\|\mathbf{V}(n+1)\|_{\mathbb{F}}^2 \right] &\leq E \left[\|\mathbf{V}(n)\|_{\mathbb{F}}^2 \right] \\ \Leftrightarrow -2\mu E [e_a(n) f(e(n))] + \mu^2 E [f^2(e(n))] &\leq 0 \\ \Leftrightarrow \mu &\leq \frac{2E [e_a(n) f(e(n))]}{E [f^2(e(n))]} \end{aligned} \tag{26}$$

Combing $f(e(n)) = e^2(n) \text{sign}\{e(n)\}$ results in

$$\mu \leq \frac{2E [e_a(n) e^2(n) \text{sign}\{e(n)\}]}{E [E^4(n)]} \tag{27}$$

Consider $e(n) = e_a(n) + v(n)$ and A2, we have

$$\mu \leq \frac{2 \{ E [e_a^3(n) \text{sign}\{e(n)\}] + E [e_a(n) v^2(n) \text{sign}\{e(n)\}] \}}{E [e_a^4(n)] + 2E [e_a^2(n) v^2(n)] + E [v^4(n)]} \tag{28}$$

According to the Price theorem [36,40], we obtain

$$\mu \leq \frac{2\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} E [e_a^3(n) e(n)] + 2\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} E [e_a(n) v^2(n) e(n)]}{E [e_a^4(n)] + 2E [e_a^2(n)] \sigma_v^2 + E [v^4(n)]} \tag{29}$$

where σ_e is the standard deviation of $e(n)$. Thus, a sufficient condition for the mean square convergence of KLMAT is formulated as

$$\begin{aligned} \mu &\leq \frac{2\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} \{ E [e_a^4(n)] + E [e_a^3(n) v(n)] \} + 2\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} \{ E [E_a^2(n) v^2(n)] + E [e_a(n) v^3(n)] \}}{E [e_a^4(n)] + 2E [e_a^2(n)] \sigma_v^2 + \sigma_v^4} \\ &= \frac{2\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e} \{ E [e_a^4(n)] + E [e_a^2(n)] \sigma_v^2 \}}{E [e_a^4(n)] + 2E [e_a^2(n)] \sigma_v^2 + \sigma_v^4}, \quad \text{if } \forall n \end{aligned} \tag{30}$$

Note that the energy of weight error $E [\|\mathbf{V}(n)\|_{\mathbb{F}}^2]$ should decrease monotonically if the learning rate satisfies the above inequality.

Remark 2 The above sufficient condition for the mean square convergence of the KLMAT algorithm is only of theoretical importance. In practical test, it is difficult to check exactly. In conventional LMS algorithm, the mean square convergence behavior can be rigorously analyzed. However, since the central limit theorem is not applicable in the nonlinear model (nonlinear prediction), $E [e_a^2(n)]$ cannot be assumed to be Gaussian.

5 Extension of KLMAT

To further improve the performance of the KLMAT and VLR-KLMAT algorithms, the recursive strategy is applied in the LMAT loss function, namely the kernel recursive least mean absolute third (KRLAT) algorithm. To derive the KRLAT algorithm in RKHS, the novel LMAT function using an exponentially weighted [29] is firstly defined by

$$\begin{aligned} J_r(n) &= \min_{\mathbf{\Omega}(n)} \left\{ \sum_{j=1}^n \lambda^{n-j} \left| d(j) - \mathbf{\Omega}^T(n) \boldsymbol{\varphi}(j) \right|^3 \right. \\ &\quad \left. + \frac{1}{2} \lambda^n \chi \|\mathbf{\Omega}(n)\|^2 \right\} \end{aligned} \tag{31}$$

where $0 \ll \lambda < 1$ is the forgetting factor and χ is the small regularization factor which de-emphasizes regularization as time progresses. The algorithm achieves slow convergence rate and small misadjustment when λ is close to one. When λ is small, the algorithm converges fast while has high steady-state error. Note that, $\frac{1}{2} \lambda^n \chi \|\mathbf{\Omega}(n)\|^2$ is a norm penalizing term, which can guarantee the existence of the inverse of the autocorrelation matrix, especially during the initial update stages [29]. Then, taking the gradient of $J_r(n)$ with respect to $\mathbf{\Omega}(n)$, we get

$$\begin{aligned} \frac{\partial J_r(n)}{\partial \Omega(n)} &= - \sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} \\ &\quad \times (d(j) - \varphi^T(j)\Omega(n)) \varphi(j) + \lambda^n \chi \Omega(n) \\ &= - \sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} d(j) \varphi^T(j) \\ &\quad + \sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} \varphi^T(j) \varphi(j) \Omega(n) \\ &\quad + \lambda^n \chi \Omega(n) \\ &= - \sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} d(j) \varphi^T(j) \\ &\quad + \left(\sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} \right. \\ &\quad \left. \times \varphi^T(j) \varphi(j) + \lambda^n \chi \right) \Omega(n). \end{aligned} \tag{32}$$

Setting (32) to zero gives

$$\Omega(n) = \Xi \Psi \tag{33}$$

where

$$\Xi = \left(\sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} \varphi^T(j) \varphi(j) + \lambda^n \chi \right)^{-1}$$

$$\text{and } \Psi = \sum_{j=1}^n \lambda^{n-j} \frac{(d(j) - \Omega^T(n)\varphi(j))^2}{|d(j) - \Omega^T(n)\varphi(j)|} d(j) \varphi^T(j).$$

Define the desired signal vector and global input vector at time n , respectively, as

$$d(n) = [d(1), d(2), \dots, d(n)] \tag{34}$$

$$\Phi(n) = [\varphi(1), \varphi(2), \dots, \varphi(n)],$$

$$\Phi(n) = \{\Phi(n-1), \varphi(n)\} \tag{35}$$

and let

$$\Lambda(n) = \text{diag} \left[\lambda^{n-1} \frac{(d(1) - \Omega^T(1)\varphi(1))^2}{|d(1) - \Omega^T(1)\varphi(1)|}, \lambda^{n-2} \frac{(d(2) - \Omega^T(2)\varphi(2))^2}{|d(2) - \Omega^T(2)\varphi(2)|}, \dots, \frac{(d(n) - \Omega^T(n)\varphi(n))^2}{|d(n) - \Omega^T(n)\varphi(n)|} \right]. \tag{36}$$

Then, (33) can be rewritten as below

$$\Omega(n) = \left(\Phi(n)\Lambda(n)\Phi^T(n) + \lambda^n \chi \mathbf{I} \right)^{-1} \times \Phi(n)\Lambda(n)d(n). \tag{37}$$

Applying the matrix inversion lemma [29]

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{38}$$

to (37), with the identifications

$$A = \chi \lambda^n, \quad B = \Phi(n), \quad C = \Lambda(n), \quad D = \Phi^T(n) \tag{39}$$

we obtain

$$\begin{aligned} &\left(\Phi(n)\Lambda(n)\Phi^T(n) + \lambda^n \chi \mathbf{I} \right)^{-1} \Phi(n)\Lambda(n) \\ &= \Phi(n) \left(\Phi(n)\Phi^T(n) + \lambda^n \chi \Lambda(n)^{-1} \right)^{-1}. \end{aligned} \tag{40}$$

Substituting the above result into (37), we obtain

$$\Omega(n) = \Phi(n) \left(\Phi(n)\Phi^T(n) + \lambda^n \chi \Lambda(n)^{-1} \right)^{-1} d(n). \tag{41}$$

The weight vector can be expressed explicitly as a linear combination of the transformed data, that is

$$\Omega(n) = \Phi(n)\Upsilon(n) \tag{42}$$

where $\Upsilon(n) = \left(\Phi(n)\Phi^T(n) + \lambda^n \chi \Lambda(n)^{-1} \right)^{-1} d(n)$ is the coefficients vector of the weight, which can be computed by kernel method. For simplicity, we define

$$\Theta(n) = \left(\Phi(n)\Phi^T(n) + \lambda^n \chi \Lambda(n)^{-1} \right)^{-1}. \tag{43}$$

Then, we have

$$\begin{aligned} \Theta(n) &= \begin{bmatrix} \Phi(n)\Phi^T(n-1) + \lambda^n \chi \Lambda(n)^{-1} & \Phi^T(n-1)\varphi(n) \\ \varphi^T(n)\varphi(n-1) & \varphi^T(n)\varphi(n) + \lambda^n \chi \left[\frac{(d(n) - \Omega^T(n)\varphi(n))^2}{|d(n) - \Omega^T(n)\varphi(n)|} \right]^{-1} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \Phi(n)\Phi^T(n-1) + \lambda^n \chi \Lambda(n)^{-1} & \Phi^T(n-1)\varphi(n) \\ \varphi^T(n)\Phi(n-1) & \varphi^T(n)\varphi(n) + \lambda^n \chi \vartheta(n) \end{bmatrix}^{-1} \end{aligned} \tag{44}$$

where $\vartheta(n) = \left[\frac{(d(n) - \mathbf{\Omega}^T(n)\boldsymbol{\varphi}(n))^2}{|d(n) - \mathbf{\Omega}^T(n)\boldsymbol{\varphi}(n)|} \right]^{-1}$. We can observe that

$$\Theta(n)^{-1} = \left[\begin{array}{cc} \Theta(n-1)^{-1} & \boldsymbol{\theta}(n) \\ \boldsymbol{\theta}^T(n)\boldsymbol{\varphi}(n)\Phi(n-1) & \lambda^n \chi \vartheta(n) + \boldsymbol{\varphi}^T(n)\boldsymbol{\varphi}(n) \end{array} \right]^{-1} \tag{45}$$

where $\boldsymbol{\theta}(n) = \Phi^T(n-1)\boldsymbol{\varphi}(n)$. Then applying the following block matrix inversion identity

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}. \tag{46}$$

Using (46), (42) becomes

$$\Theta(n)^{-1} = \rho^{-1}(n) \times \begin{bmatrix} \Theta(n-1)\rho(n) + \mathbf{q}(n)\mathbf{q}^T(n) & -\mathbf{q}(n) \\ -\mathbf{q}^T(n) & 1 \end{bmatrix} \tag{47}$$

where $\mathbf{q}(n) = \Theta(n-1)\boldsymbol{\theta}(n)$ and $\rho(n) = \lambda^n \chi \vartheta(n) + \boldsymbol{\varphi}^T(n)\boldsymbol{\varphi}(n) - \mathbf{q}^T(n)\boldsymbol{\theta}(n)$.

Combining (37) and (47), we arrive the following relation

$$\begin{aligned} \Psi(n) &= \Theta(n)d(n) \\ &= \begin{bmatrix} \Theta(n-1) + \mathbf{q}(n)\mathbf{q}^T(n)\rho^{-1}(n) & -\mathbf{q}(n)\rho^{-1}(n) \\ -\mathbf{q}^T(n)\rho^{-1}(n) & \rho^{-1}(n) \end{bmatrix} \\ &\quad \begin{bmatrix} d(n-1) \\ d(n) \end{bmatrix} \\ &= \begin{bmatrix} \Psi(n-1) - \mathbf{q}(n)\rho^{-1}(n)e(n) \\ \rho^{-1}(n)e(n) \end{bmatrix}. \end{aligned} \tag{48}$$

6 Simulation results

We conduct a series of simulations to evaluate the performance of the proposed algorithms, including simulations on a Mackey–Glass (MG) chaotic time series prediction and simulations on a sunspot number time series analysis. We compare the estimation results of the proposed algorithms with those of the KLMS algorithm and the KRLS algorithm. In the simulation study, the effectiveness is assessed in terms of MSE in the testing stage, which is defined as $MSE = 10 \log_{10} \{e^2(n)\}$ [37]. The parameters in algorithms (learning rate, kernel size, etc.) are selected to guarantee the fast and stable convergence of the algorithms. All the simulation results below are averaged over 100 independent Monte Carlo runs.

6.1 Example 1: MG chaotic time series prediction

In this example, the simulation studies are carried out for the MG chaotic time series prediction. The MG series is generated by a delay ordinary differential equation [27, 29]:

$$\frac{dx(t)}{dt} = -qx(t) + \frac{mx(t - \tau)}{1 + x(t - \tau)^{10}} \tag{49}$$

where $q = 0.1$, $m = 0.2$, and $\tau = 30$. The sampling period is 6 s, and time embedding (filter order) is 10. The white Gaussian noise (WGN) with zero mean and standard deviation $\sigma_G = 0.02$ is used as a measurement noise. A segment of 2000 samples is used as the training data and another 2000 as the test data. If the number of testing iteration 2000 is achieved, then stop the algorithm.

Firstly, we let h of the KLMAT algorithm vary within $0.1 \sim 2$ to test the performances of algorithms under different kernel sizes. When the kernel size is too small for the data samples, the performance of the algorithm may degrade since the lack of information needed in inner product calculation. When the kernel size is relatively small but in a reasonable range for the data samples, the algorithm converges quickly with relatively high steady-state error. When the kernel size is getting larger, the crest around the global optimum becomes wider, so under the fixed learning rate, the smaller misadjustment and lower convergence rate are obtained. From Fig. 3, we observed that, for $h \in (0.5, 2)$, the results are quite similar. In following simulations, we set $h = 1.5$, for small steady-state error. Then, we test the prediction performance of the proposed algorithms in Figs. 4 and 5. Figure 4 shows the

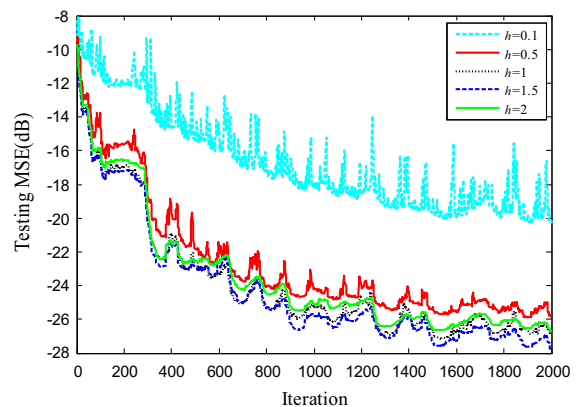


Fig. 3 Testing MSE curves for kernel sizes ($\mu = 1$)

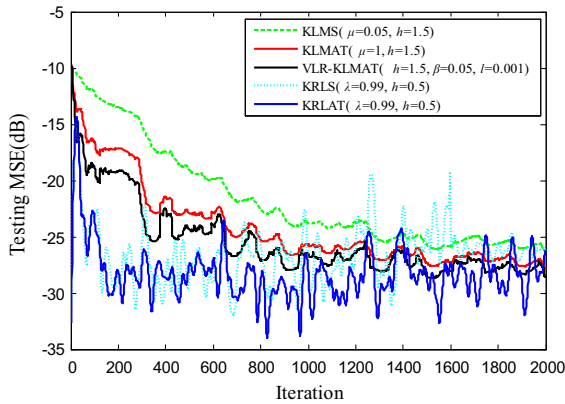


Fig. 4 Testing MSE curves of the algorithms

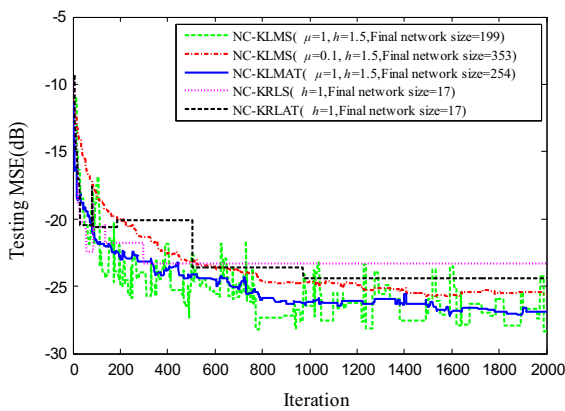


Fig. 5 Testing MSE curves of the NC-based algorithms

testing MSE under WGN environment. Observe that the KLMAT and VLR-KLMAT algorithms achieve improved performance as compared with the KLMS algorithm. Besides, the VLR-KLMAT algorithm outperforms the KLMAT algorithm, since it reaches the similar steady-state error level within fewer iterations. Owing to using the recursive method, the KRLS and KRLAT algorithms converge faster than other algorithms. Additionally, the KRLAT algorithm provides improvement with WGN noise.

Note that the network size of the KAF linearly increases with the number of training data, which means that there is a total of 2000 expansion coefficients at the end of our simulations. This may prohibit the practical implementation of these algorithms for a large training set. Therefore, the novelty criterion (NC) strategy [29] is considered to be a possible solution to the limitation of the proposed algorithms. The NC com-

putes the distance of $\mathbf{u}(n + 1)$ to the present dictionary c_j . If the distance is smaller than some preset threshold $\zeta_1 (\zeta_1 > 0)$, new input data $\mathbf{u}(n + 1)$ will not be added to the dictionary. Only if the magnitude of the prediction error is larger than another preset threshold $\zeta_2 (\zeta_2 > 0)$, new input data will be accepted as a new center. Guided by this method, the NC-KLMAT and NC-KRLAT algorithms can be easily derived by introducing NC in the KLMAT and KRLAT algorithms. To curb the network size, we have used a NC-KLMAT filter with $\zeta_1 = 0.1$ and $\zeta_2 = 0.001$. For the NC-KRLAT algorithm, we use 20 values of ζ_1 and ζ_2 uniformly in the interval of $[0.04, 0.2]$ for the enhanced NC [29]. As can be seen from Fig. 5, the NC method provides overwhelmingly smaller network size as comparison to other algorithms with sacrifice of the prediction accuracy.

Finally, to quantify the computational burden, we measured the average run execution time of the algorithm on a 2.1-GHz AMD processor with 2 GB of RAM, running MATLAB R2013a on Windows 7 environment. As one see from Table 2, the KLMS algorithm is the fastest method owing to utilizing gradient descent. The KRLAT algorithm increases the computation time, but the faster convergence rate and stable performance are achieved as compared to other algorithms.

6.2 Example 2: Effect of the measurement noise

In the second example, we focus on the performance of the proposed algorithms in MG time series prediction for various probability densities. The experimental conditions are considered to be same as in the example 1, but the measurement noise is considered as non-Gaussian noise. Since in many physical environments, the noise is characterized by non-

Table 2 Average computation time per run of the algorithms

Algorithms	Computation time (s)
KLMS	75.165688
KLMAT	77.249014
VLR-KLMAT	79.350315
KRLS	121.553110
KRLAT	152.816778

The bold values stand for the proposed algorithms

Table 3 Steady-state testing MSEs

Algorithm	Uniform noise with the variance $u \sim U(-0.1, 0.1)$	Rayleigh noise with variance $\sigma_R^2 = 0.05$	Rectangular noise with amplitude $A = 0.1$, frequency $f = 1000\text{Hz}$	Sinusoidal noise
KLMS				
Parameters	$\mu = 1, h = 1.5$	$\mu = 0.05, h = 1.5$	$\mu = 0.05, h = 1.5$	$\mu = 0.05, h = 1.5$
Testing MSE	0.019710	0.0036203	0.0018737	0.0018737
KLMAT				
Parameters	$\mu = 0.3, h = 1.5$	$\mu = 1, h = 1.5$	$\mu = 1, h = 1.5$	$\mu = 1, h = 1.5$
Testing MSE	0.0057810	0.0030222	0.0013024	0.0013024
VLR-KLMAT				
Parameters	$h = 1.5, \beta = 0.05, l = 0.001$	$h = 1.5, \beta = 0.05, l = 0.001$	$h = 1.5, \beta = 0.05, l = 0.001$	$h = 1.5, \beta = 0.05, l = 0.001$
Testing MSE	0.0048022	0.0028152	0.0008931	0.0008931
KRLS				
Parameters	$\lambda = 0.99, h = 0.5$	$\lambda = 0.99, h = 1$	$\lambda = 0.99, h = 1$	$\lambda = 0.99, h = 1$
Testing MSE	0.0001165	4.1514e-05	59.1932 (Unstable)	0.0001772
KRLAT				
Parameters	$\lambda = 0.99, h = 0.5$	$\lambda = 0.99, h = 1$	$\lambda = 0.99, h = 1$	$\lambda = 0.99, h = 1$
Testing MSE	0.0001032	3.3797e-05	2.2709e-06	2.2708e-06

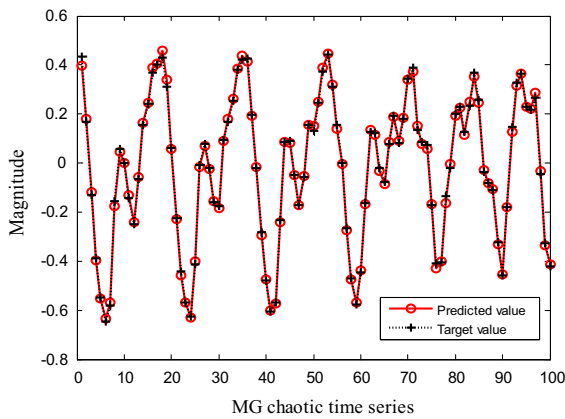


Fig. 6 Predicted values of the KRLAT algorithm and target values

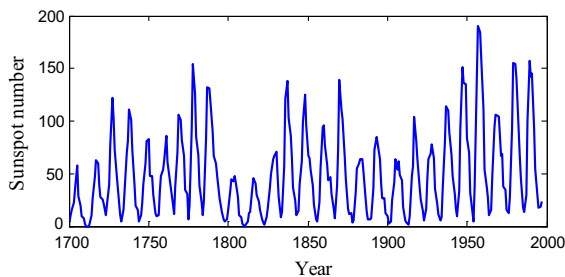


Fig. 7 Sunspot number time series

Gaussian distribution. We generate the uniform noise from the uniform distribution of probability density function $u \sim \frac{1}{b-a}$, where $b = 0.1$ and $a = -0.1$. We set $u(n) = \sqrt{-2\sigma_R^2 \log(1 - y(n))}$ for Rayleigh noise, where $y(n)$ is a uniform random variable in $(0,1)$ and $\sigma_R^2 = 0.05$ is the variance. The square function is employed to generate the rectangular noise in MATLAB. The Sinusoidal noise is generated by $u(n) = \sin(200\pi n) + \sin(1000\pi n) + \sin(1800\pi n)$. To compare the steady-state performance of the algorithms fairly, we change the parameters so that they

has the similar initial convergence rate. Table 3 shows a comparison of the KLMS, KLMAT, VLR–KLMAT, KRLS and KRLAT algorithms. The predicted values of the KRLAT and Target values are shown in Fig. 6. We can clearly see that the KLMAT and VLR–KLMAT algorithms outperform the KLMS and LMAT algorithms for all of the probability densities. The KRLS and KRLAT algorithms perform better than the KLMS-based algorithms, and the KRLAT algorithm achieves good prediction results in all these cases. The predicted value of the KRLAT algorithm agrees with the target well. Specifically, the performance of the proposed algorithms is much better than that of the existing algorithms with uniform noise and rectangular noise.

6.3 Example 3: Application of the sunspot number time series analysis

To test the performance of algorithms in realistic applications, the proposed algorithms are applied to adaptive prediction of the annually recorded sunspot time series for the years 1700–1997 [61]. A segment of the processed sunspot number series is shown in Fig. 7. The testing MSE is calculated based on 100 test data. The time embedding is 2. We choose to termination the algorithm when the testing iteration 100 is achieved. For the sake of a fair comparison, we let the algorithms use the different learning rates to accomplish similar convergence rate. In Table 4, the prediction results for WGN with $\sigma_G = 0.1$ are illustrated. It is concluded that the proposed algorithms are superior to the existing algorithms in terms of the steady-state prediction error.

7 Conclusion

We proposed a KAF based on LMAT loss function named KLMAT, which was derived by utilizing the ker-

Table 4 Steady-state testing MSEs

Algorithm	Parameters	Testing MSE
KLMS	$\mu = 0.5, h = 1.5$	0.049139
KLMAT	$\mu = 1, h = 1.5$	0.039445
VLR–KLMAT	$h = 1.5, \beta = 0.05, l = 0.001$	0.032900
KRLS	$\lambda = 0.99, h = 1$	6.3603e–05
KRLAT	$\lambda = 0.99, h = 1$	5.7681e–05

nel method and gradient descent method. Besides, its VLR version was proposed by using Lorentzian function to accelerate the initial convergence rate. In the analysis, it has been shown the upper bound of the KLAMT algorithm for mean square convergence. To further enhance the performance of the proposed algorithms, we developed a kernel extension of the KLAMT algorithm in a recursive strategy. The new KRLAT algorithm incorporates an exponentially weighted into the LMAT loss function to adapt the tap-weight vector in feature space which can maintain robustness against different noise environments and can increase the convergence rate for time series prediction. We carried out simulations that confirmed the superiority of the proposed algorithms. Our future works will concern with using real data in adaptive prediction. Some initial works have been done.

Acknowledgements The authors want to express their deep thanks to the anonymous reviewers for many valuable comments which greatly helped to improve the quality of this work. This work was partially supported by National Science Foundation of P.R. China (Grant: 61571374, 61271340, 61433011). The first author would like to acknowledge the China Scholarship Council (CSC) for providing him with financial support to study abroad (No. 201607000050).

References

- Aboulnasr, T., Mayyas, K.: A robust variable step-size LMS-type algorithm: analysis and simulations. *IEEE Trans. Signal Process.* **45**(3), 631–639 (1997)
- Arqub, O.A.: Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations. *Neural Comput. Appl.* (2015). doi:[10.1007/s00521-015-2110-x](https://doi.org/10.1007/s00521-015-2110-x)
- Arqub, O.A., Al-Smadi, M., Momani, S., Hayat, T.: Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput.* (2016). doi:[10.1007/s00500-016-2262-3](https://doi.org/10.1007/s00500-016-2262-3)
- Arqub, O.A., Al-Smadi, M., Momani, S., Hayat, T.: Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. *Soft Comput.* **20**, 3283–3302 (2016)
- Bershad, N.J., Bermudez, J.C.M.: Stochastic analysis of the Least Mean Kurtosis algorithm for Gaussian inputs. *Digit. Signal Process.* **54**, 35–45 (2016)
- Black, M.J., Rangarajan, A.: The outlier process: unifying line processes and robust statistics. In: *IEEE CS Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 15–22 (1994)
- Cao, Z., Yu, S., Principe, J.C., Xu, G., Chen, B.: Multiple adaptive kernel size KLMS for Beijing PM2.5 prediction. In: *International Joint Conference on Neural Networks*, pp. 1403–1407 (2016)
- Chambers, J.A., Avlonitis, A.: A robust mixed-norm adaptive filter algorithm. *IEEE Signal Process. Lett.* **4**(2), 46–48 (1997)
- Chambers, J.A., Tanrikulu, O., Constantinides, A.G.: Least mean mixed-norm adaptive filtering. *Electron. Lett.* **30**(19), 1574–1575 (1994)
- Chen, B., Yang, X., Ji, H., Qu, H., Zheng, N., Principe, J.C.: Trimmed affine projection algorithms. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 1923–1928 (2014)
- Chen, B., Yuan, Z., Zheng, N., Principe, J.C.: Kernel minimum error entropy algorithm. *Neurocomputing* **121**, 160–169 (2013)
- Chen, B., Zhao, S., Zhu, P., Principe, J.C.: Mean square convergence analysis for kernel least mean square algorithm. *Signal Process.* **92**(11), 2624–2632 (2012)
- Chen, B., Zhao, S., Zhu, P., Principe, J.C.: Quantized kernel least mean square algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **23**(1), 22–32 (2012)
- Chen, B., Zhao, S., Zhu, P., Principe, J.C.: Quantized kernel recursive least squares algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **24**(9), 1484–1491 (2013)
- Chouvardas, S., Draief, M.: A diffusion kernel LMS algorithm for nonlinear adaptive networks. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4164–4168 (2016)
- Costa, M.H., Bermudez, J.C.M.: A noise resilient variable step-size LMS algorithm. *Signal Process.* **88**(3), 733–748 (2008)
- Engel, Y., Mannor, S., Meir, R.: The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* **52**(8), 2275–2285 (2004)
- Eweda, E., Bershad, N.J.: Stochastic analysis of a stable normalized least mean fourth algorithm for adaptive noise canceling with a white Gaussian reference. *IEEE Trans. Signal Process.* **60**(12), 6235–6244 (2012)
- García, J.A., Vidal, A.R.F., Sayed, A.H.: Mean-square performance of a convex combination of two adaptive filters. *IEEE Trans Signal Process.* **54**(3), 1078–1090 (2006)
- Gil-Cacho, J.M., Signoretto, M., van Waterschoot, T., Moonen, M., Jensen, S.H.: Nonlinear acoustic echo cancellation based on a sliding-window leaky kernel affine projection algorithm. *IEEE Trans. Audio Speech Lang. Process.* **21**(9), 1867–1878 (2013)
- Huang, B., Xiao, Y., Ma, Y., Wei, G.G., Sun, J.: A simplified variable step-size LMS algorithm for Fourier analysis and its statistical properties. *Signal Process.* **117**, 69–81 (2015)
- Koike, S.I.: Convergence analysis of adaptive filters using normalized sign-sign algorithm. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **88**(11), 3218–3224 (2005)
- Lee, Y.H., Mok, J.D., Kim, S.D., Cho, S.H.: Performance of least mean absolute third (LMAT) adaptive algorithm in various noise environments. *Electron. Lett.* **34**(3), 241–243 (1998)
- Li, C., Shen, P., Liu, Y., Zhang, Z.: Diffusion information theoretic learning for distributed estimation over network. *IEEE Trans. Signal Process.* **61**(16), 4011–4024 (2013)
- Liu, Z., Hu, Q., Cui, Y., Zhang, Q.: A new detection approach of transient disturbances combining wavelet packet and Tsallis entropy. *Neurocomputing* **142**, 393–407 (2014)

26. Liu, W., Park, Il, Wang, Y., Principe, J.C.: Extended kernel recursive least squares algorithm. *IEEE Trans. Signal Process.* **57**(10), 3801–3814 (2009)
27. Liu, W., Pokharel, P., Principe, J.C.: The kernel least mean square algorithm. *IEEE Trans. Signal Process.* **56**(2), 543–554 (2008)
28. Liu, W., Principe, J.C.: Kernel affine projection algorithms. *EURASIP J. Adv. Signal Process.* **2008**(1), 1–13 (2008)
29. Liu, W., Principe, J.C., Haykin, S.: *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley, Hoboken, NJ (2010)
30. Liu, J., Qu, H., Chen, B., Ma, W.: Kernel robust mixed-norm adaptive filtering. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 3021–3024 (2014)
31. Lu, L., Zhao, H.: Adaptive Volterra filter with continuous lp-norm using a logarithmic cost for nonlinear active noise control. *J. Sound Vib.* **364**, 14–29 (2016)
32. Lu, L., Zhao, H., Chen, B.: Collaborative adaptive Volterra filters for nonlinear system identification in α -stable noise environments. *J. Frankl. Inst.* **353**(17), 4500–4525 (2016)
33. Lu, L., Zhao, H., Chen, B.: Improved-variable-forgetting-factor recursive algorithm based on the logarithmic cost for Volterra system identification. *IEEE Trans. Circuits Syst. II* **63**(6), 588–592 (2016)
34. Lu, L., Zhao, H., Chen, B.: KLMAT: A Kernel Least Mean Absolute Third Algorithm (2016). [arXiv:1603.03564](https://arxiv.org/abs/1603.03564)
35. Lu, L., Zhao, H., Chen, B.: Variable-Mixing Parameter Quantized Kernel Robust Mixed-Norm Algorithms for Combating Impulsive Interference (2015). [arXiv:1508.05232](https://arxiv.org/abs/1508.05232)
36. Lu, L., Zhao, H., Li, K., Chen, B.: A novel normalized sign algorithm for system identification under impulsive noise interference. *Circuits Syst. Signal Process.* **35**(9), 3244–3265 (2015)
37. Ma, W., Duan, J., Man, W., Zhao, H., Chen, B.: Robust kernel adaptive filters based on mean p -power error for noisy chaotic time series prediction. *Eng. Appl. Artif. Intell.* **58**, 101–110 (2017)
38. Machado, J.A.T.: Entropy analysis of integer and fractional dynamical systems. *Nonlinear Dyn.* **62**(1), 371–378 (2010)
39. Masry, E., Bullo, F.: Convergence analysis of the sign algorithm for adaptive filtering. *IEEE Trans. Inf. Theory* **41**(2), 489–495 (1995)
40. Mathews, V.J., Cho, S.H.: Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **35**(4), 450–454 (1987)
41. Miao, Q.Y., Li, C.G.: Kernel least-mean mixed-norm algorithm. In: *International Conference on Automatic Control and Artificial Intelligence (ACAI)*, Mar. 3–5, pp. 1285–1288 (2012)
42. Mitra, R., Bhatia, V.: Diffusion-KLMS Algorithm and Its Performance Analysis for Non-linear Distributed Networks (2015). [arXiv:1509.01352](https://arxiv.org/abs/1509.01352)
43. Papoulis, E.V., Stathaki, T.: A normalized robust mixed-norm adaptive algorithm for system identification. *IEEE Signal Process. Lett.* **11**(1), 56–59 (2004)
44. Parreira, W.D., Bermudez, J.C.M., Richard, C., Tourneret, J.Y.: Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm. *IEEE Trans. Signal Process.* **60**(5), 2208–2222 (2012)
45. Pei, S.C., Tseng, C.C.: Least mean p -power error criterion for adaptive FIR filter. *IEEE J. Sel. Areas Commun.* **12**(9), 1540–1547 (1994)
46. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**(5), 1299–1319 (1998)
47. Tian, J., Gu, H., Gao, C., Lian, J.: Local density one-class support vector machines for anomaly detection. *Nonlinear Dyn.* **64**(1–2), 127–130 (2011)
48. Walach, E., Widrow, B.: The least mean fourth (LMF) adaptive algorithm and its family. *IEEE Trans. Inf. Theory* **30**(2), 275–283 (1984)
49. Wang, S., Feng, J., Tse, C.K.: Kernel affine projection sign algorithms for combating impulse interference. *IEEE Trans. Circuits Syst. II* **60**(11), 811–815 (2013)
50. Wang, S., Zheng, Y., Duan, S., Wang, L., Chi, K.T.: A class of improved least sum of exponentials algorithms. *Signal Process.* **128**, 340–349 (2016)
51. Wu, P., Duan, F., Guo, P.: A pre-selecting base kernel method in multiple kernel learning. *Neurocomputing* **165**, 46–53 (2015)
52. Wu, Z., Shi, J., Zhang, X., Ma, W., Chen, B.: Kernel recursive maximum correntropy. *Signal Process.* **117**, 11–16 (2015)
53. Yang, J., Guan, J., Ning, Y., Liu, Q.: A novel adaptive harmonic detecting algorithm applied to active power filters. In: *Proceedings of the 3rd International Congress of Image Signal Processing Oct. 16–18, vol. 7, pp. 3287–3290* (2010)
54. Yang, X., Qu, H., Zhao, J., Chen, B.: Hybrid affine projection algorithm. In: *13th International Conference on Control Automation Robotics and Vision (ICARCV)*, pp. 964–968 (2014)
55. Yazdi, H.S., Pakdaman, M., Modagheh, H.: Unsupervised kernel least mean square algorithm for solving ordinary differential equations. *Neurocomputing* **74**(12), 2062–2071 (2011)
56. Yu, Y., Zhao, H.: An improved variable step-size NLMS algorithm based on a Versiera function. In: *IEEE International Conference on Signal Processing, Communications and Computing, Aug. 5–8, pp. 1–4* (2013)
57. Zhang, H., Yang, T., Xu, W., Xu, Y.: Effects of non-Gaussian noise on logical stochastic resonance in a triple-well potential system. *Nonlinear Dyn.* **76**(1), 649–656 (2014)
58. Zhao, S., Chen, B., Cao, Z., Zhu, P., Principe, J.C.: Self-organizing kernel adaptive filtering. *EURASIP J. Adv. Signal Process.* **2016**(1), 106 (2016)
59. Zhao, S., Chen, B., Principe, J.C.: Kernel adaptive filtering with maximum correntropy criterion. In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 2012–2017 (2011)
60. Zhao, S., Chen, B., Zhu, P., Principe, J.C.: Fixed budget quantized kernel least-mean-square algorithm. *Signal Process.* **93**(9), 2759–2770 (2013)
61. Zhao, H., Gao, S., He, Z., Zeng, X., Jin, W., Li, T.: Identification of nonlinear dynamic system using a novel recurrent wavelet neural network-based the pipelined architecture. *IEEE Trans. Ind. Electron.* **61**(8), 4171–4182 (2014)
62. Zhao, S., Man, Z., Khoo, S., Wu, H.R.: Variable step-size LMS algorithm with a quotient form. *Signal Process.* **89**(1), 67–76 (2009)

63. Zhao, X., Shang, P.: Principal component analysis for non-stationary time series based on detrended cross-correlation analysis. *Nonlinear Dyn.* **84**(2), 1033–1044 (2016)
64. Zhao, J., Wang, T., Ma, W., Qu, H.: The kernel-LMS based network traffic prediction. In: *Information Science and Control Engineering (ICISCE 2012)*, pp. 1–5 (2012)
65. Zhao, H., Yu, Y., Gao, S., Zeng, X., He, Z.: A new normalized LMAT algorithm and its performance analysis. *Signal Process.* **105**, 399–409 (2014)
66. Zhao, J., Zhang, G.: A robust Prony method against synchrophasor measurement noise and outliers. *IEEE Trans. Power Syst.* (2016). doi:[10.1109/TPWRS.2016.2612887](https://doi.org/10.1109/TPWRS.2016.2612887)
67. Zhou, Y., Chan, S.C., Ho, K.L.: New sequential partial-update least mean M-estimate algorithms for robust adaptive system identification in impulsive noise. *IEEE Trans. Ind. Electron.* **58**(9), 4455–4470 (2011)