A modified fractional least mean square algorithm for chaotic and nonstationary time series prediction*

Bilal Shoaib[a)†], Ijaz Mansoor Qureshi[b)], Ihsanulhaq[a)], and Shafqatullah[c)]

[a)] Department of Electronic Engineering, Faculty of Engineering and Technology, International Islamic University, Islamabad, Pakistan
[b)] Department of Electrical Engineering, AIR University, Islamabad, Pakistan
[c)] School of Engineering and Applied Sciences, ISRA University, Islamabad, Pakistan

(Received 14 July 2013; revised manuscript received 6 September 2013; published online 12 February 2014)

A method of modifying the architecture of fractional least mean square (FLMS) algorithm is presented to work with nonlinear time series prediction. Here we incorporate an adjustable gain parameter in the weight adaptation equation of the original FLMS algorithm and absorb the gamma function in the fractional step size parameter. This approach provides an interesting achievement in the performance of the filter in terms of handling the nonlinear problems with less computational burden by avoiding the evaluation of complex gamma function. We call this new algorithm as the modified fractional least mean square (MFLMS) algorithm. The predictive performance for the nonlinear Mackey glass chaotic time series is observed and evaluated using the classical LMS, FLMS, kernel LMS, and proposed MFLMS adaptive filters. The simulation results for the time series with and without noise confirm the superiority and improvement in the prediction capability of the proposed MFLMS predictor over its counterparts.

Keywords: fractional least mean square, kernel methods, Reimann–Lioville derivative, Mackey glass time series

PACS: 05.45.Df, 05.30.Pr, 05.45.Tp, 92.60.Wc **DOI:** 10.1088/1674-1056/23/3/030502

1. Introduction

The theory of fractional calculus is as old as that of classical calculus. Researchers in many scientific areas are currently paying attention to this concept. Successful applications are found in signal processing, robotics, system identification, robotics, system identification, genetic algorithm, control systems, as well as finance and economy.[1,2] Extensive research has been carried out for the solution of linear and nonlinear differential equations through fractional calculus, which motivates researchers to analyze the numerical computing in this field.[3–5] In short, fractional calculus has found wide applications for multifarious problems in science and engineering.

The fractional least mean square algorithm is introduced by adding a fractional derivative term in the standard least mean square algorithm to improve the convergence of the algorithm.[6] The fractional LMS has been applied to parameter identification of a nonlinear controlled autoregressive system, modeling of the fractional Brownian motion, dual channel speech enhancement,[7,8] etc.

Chaotic systems show unstable bounded dynamic behavior that reveals sensitive dependence on the initial conditions and comprises infinite unstable periodic motions. They are sensitive to the initial conditions and system parameters. In the real world, parameter estimation has become a difficult task due to the complexity of the chaotic systems. Genetic algorithm, particle swarm optimization, and cuckoo search algorithm[9] are used to handle this task. Chaotic time series prediction has gained significant progress during the past decade. Due to their chaotic nature, linear prediction methods are inapplicable. Hence, a number of nonlinear prediction methods have been developed in literatures. However, most of them handle efficiently the stationary series.

The main contribution of this article is the development of modified fractional least mean square algorithm and its application to the prediction of chaotic, nonlinear, and nonstationary time series. Different cases are discussed to compare the performance of the proposed algorithm with appropriate gains and step sizes. For clarity, real variables are used to carry out all derivations and the formulation of the algorithm. The simulation results are presented under different scenarios.

This paper is organized as follows. In Section 2, a detailed description of the problem based on time series prediction along with LMS and kernel LMS is presented. In Section 3, details about FLMS and modified FLMS are discussed. Section 4 presents the experimental results and simulation. Section 5 provides future directions.

2. Least mean square and kernel least mean square algorithms

2.1. Prediction problem setting

One of the most celebrated problems in signal processing is that of predicting a future value in nonlinear time se-

*Project supported by the Higher Education Commission of Pakistan.
†Corresponding author. E-mail: bilal.phdee42@iiu.edu.pk
© 2014 Chinese Physical Society and IOP Publishing Ltd

ries analysis.[10] Consider the time series $x(n), x(n-1), x(n-2), x(n-3), \ldots, x(n-M)$, representing $M+1$ samples of such a process. In the process of prediction, samples $x(n-1), x(n-2), x(n-3), \ldots, x(n-M)$ are used to predict the current value of $x(n)$. The specified data are first chosen for training upon which the filter taps are adjusted and use these weights to predict the next value of the time series. The taps are adjusted here according to the stochastic gradient algorithm (LMS), nonlinear feature mapping technique (KLMS), and fractional derivative technique (FLMS and MFLMS). The basic architecture of the predictor of order M is given in Fig. 1.

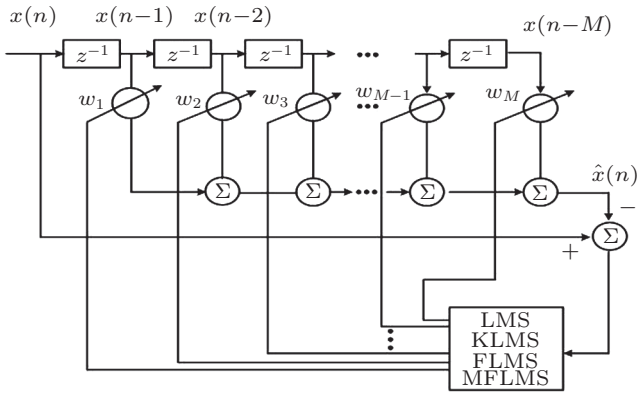


Fig. 1. Basic architecture of the predictor.

2.1.1. Least mean square algorithm

Widow and Hopf[11] used stochastic gradient technique to formulate the least mean square algorithm. In order to minimize the empirical risk by using the example sequence $[\boldsymbol{x}(i), d(i)]$ as

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{i=0}^{n}(d(i) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(i))^2. \tag{1}$$

The gradient of the above-mentioned cost function with respect to w is given by

$$\nabla_w J = -\sum_{i=1}^{n} x(i)(d(i) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}(i)). \tag{2}$$

Finally, the weights are adjusted according to the method of steepest descent

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu e(n)x(n). \tag{3}$$

It is customary to initialize the tap weight vector to zero.

2.1.2. Kernel least mean square algorithm

A poor performance can be expected when the difference or the mapping between $\boldsymbol{x}$ and $\boldsymbol{d}$ is highly nonlinear. A nonlinear mapping was introduced in Refs. [12] and [13] as $\varphi(\boldsymbol{x}(i))$, which gives a more powerful model $\boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}(i))$ than $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$. Hence, using this model and finding $\boldsymbol{w}$ through stochastic gradient technique may prove an efficient mechanism towards nonlinear filtering as LMS ensures for linear problems. Using the example sequence $[\varphi(i), d(i)]$, we evaluate the algorithm as follows:

$$\begin{aligned} e(i) &= d(i) - \boldsymbol{w}^{\mathrm{T}}(i-1)\varphi(i). \qquad (4)\\ \boldsymbol{w}(i) &= \boldsymbol{w}(i-1) + \mu e(i)\varphi(i),\\ \boldsymbol{w}(i) &= \boldsymbol{w}(i-2) + \mu e(i-1)\varphi(i-1) + \mu e(i)\varphi(i),\\ \boldsymbol{w}(i) &= \boldsymbol{w}(i-3) + \mu e(i-2)\varphi(i-2)\\ &\quad + \mu e(i-1)\varphi(i-1) + \mu e(i)\varphi(i)\\ &= \boldsymbol{w}(0) + \mu\sum_{j=1}^{n} e(j)\varphi(j). \qquad (5) \end{aligned}$$

If we take $\boldsymbol{w}(0) = 0$, then

$$\boldsymbol{w}(i) = \mu\sum_{j=1}^{n} e(j)\varphi(j). \tag{6}$$

The repeated application of the weight update equation yields

$$y = \mu\sum_{j=1}^{n} e(j)k(\boldsymbol{x}(j), \boldsymbol{x}(i)). \tag{7}$$

There is an inability to work directly with the filter weights in the feature space (RKHS), so we take a difference between the output of the network and desired signal as coefficient of the filter.[14,15] They are adjusted instead of the weight vector for KLMS in order to predict the next value of the time series.

3. Fractional least mean square algorithm

A nonlinear and fractional adaptive algorithm has been introduced[6] for identification of the autoregressive (AR) systems. Owing to its formulation, it has been immensely utilized in various fields effectively, such as echo cancellation, dual channel speech enhancement,[7] and performance analysis of the Bessel beamformer.[8] Before discussing the algorithm, we hereby present some basic concepts about the fractional derivative. The idea stems from the Riemann–Liouville fractional integral and derivative method. The definition of the method is given by

$$I^{\nu}f(t) = \frac{1}{\Gamma\alpha)}\int_0^t (t-\nu)^{\nu-1} f(\tau)\,\mathrm{d}\tau, \tag{8}$$

where I^{ν} is the fractional integral of order ν. Accordingly, the fractional derivative is given by

$$\begin{aligned} (D^{\nu}f)(t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^n (I^{n-\nu}f)(t),\\ (D^{\nu}f)(t) &= \frac{1}{\Gamma(\alpha)}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^n \int_0^t (t-\tau)^{n-\nu-1} f(\tau)\,\mathrm{d}\tau, \qquad (9) \end{aligned}$$

where D^{ν} is defined as the fractional derivative and n is an integer. In order to add a little more details, we present the

Reimann–Lioville fractional derivative applied to the power function. The ν derivative of $f(t) = (t-a)^{\alpha}$ is

$$D^{\nu}(t-a)^{\alpha} = \frac{\Gamma(1+\alpha)}{\Gamma(1+\alpha+\nu)}(t-a)^{\alpha-\nu}, \tag{10}$$

where a and α are real constants.

3.1. Derivation of FLMS algorithm

We start with the standard LMS FIR filter.[10,11] The error is minimized according to the cost function

$$J(n) = \frac{1}{2}E[|e(n)|^2]. \tag{11}$$

The adaptive weight mechanism of the LMS filter having only the first derivative is given by

$$w_k(n+1) = w_k(n) - \mu\left(\frac{\partial J(n)}{\partial w_k}\right). \tag{12}$$

We use the fractional derivative in addition to the first derivative to Eq. (12)

$$w_k(n+1) = w_k(n) - \mu\left(\frac{\partial J(n)}{\partial w_k}\right) - \mu_2\left(\frac{\partial}{\partial w_k}\right)^{\nu} J(n). \tag{13}$$

Evaluating Eq. (13), we have

$$\frac{\partial J(n)}{\partial w_k} = -e(n)x(n-k), \tag{14}$$

$$\left(\frac{\partial}{\partial w_k}\right)^{\nu} J(n) = -e(n)x(n-k)D^{\nu}w_k(n),$$

$$\left(\frac{\partial}{\partial w_k}\right)^{\nu} J(n) = -e(n)x(n-k)\left[\frac{1}{\Gamma(2-\nu)}w_k^{1-\nu}(n)\right]. \tag{15}$$

By substituting Eqs. (14) and (15) into Eq. (13), we have a method for the weight adaptation of k^{th} element for FLMS

$$w_k(n) = w_k(n-1) + \mu_1 e(n)x(n-k) + \mu_2 e(n)x(n-k)\frac{w_k^{1-\nu}(n)}{\Gamma(2-\nu)}, \tag{16}$$

where ν is a real number between 0 and 1. To pace up the computation for equations of nonlinear nature in a cyclic single step iteration, we assume $w_k^{1-\nu}(n) \cong w_k^{1-\nu}(n-1)$. Then equation (16) becomes

$$w_k(n) = w_k(n-1) + \mu_1 e(n)x(n-k) + \mu_2 e(n)x(n-k)\frac{w_k^{1-\nu}(n-1)}{\Gamma(2-\nu)}. \tag{17}$$

3.2. Modified FLMS algorithm

Here we introduce a mechanism to avoid the computation of the Gamma function, probably one of the most fundamental function in the fractional calculus. By absorbing this function in μ_{f} and introducing the adjustable gain parameter β, we reformulate Eq. (17) as

$$w_k(n) = w_k(n-1) + \beta\mu e(n)x(n-k) + (1-\beta)\mu_f e(n)x(n-k)w_k^{1-\nu}(n), \tag{18}$$

where $0 < \beta < 1$ is adjusted manually. μ and μ_{f} are the step size fractional parameters, with their values adjusted between 0 and 1. We do make sure that the algorithm does not diverge. β has to be adjusted for optimal results. Obviously, if β is taken as 0.9, we are giving dominance to LMS. If $\beta < 0.5$, we are giving preference to FLMS. Furthermore, it is also possible that β may be adapted accordingly with the number of iterations.

4. Simulation results

In this section, we present some simulation results of time series prediction using LMS, KLMS, FLMS, and MFLMS predictors. To evaluate the performance of these filters, chaotic Mackey–Glass time series is generated by the following delay differential equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = -bx(t) + \frac{ax(t-\tau)}{1+x^{10}(t-\tau)}. \tag{19}$$

With $b = 0.1$, $a = 0.2$, and $\tau = 20$, initial conditions $x(t-\tau) = 0$ for $0 \le t \le \tau$. Thus, we have a series $x(k)$ for $k = 1, 2, \ldots, 3000$, obtained by Eq. (19). It is achieved by sampling the continuous curve $x(t)$ with the interval of 1 second. The time embedding length or the order of the filter M is 10 for all the experiments, a segment of 700 samples is used for training, while another 200 is used for testing. Different cases are discussed here for the analysis of the Mackey–Glass time series prediction. The time series generated is shown in Fig. 2.

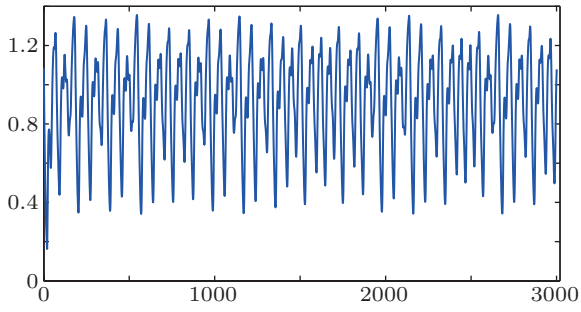


Fig. 2. (color online) Mackey–Glass time series.

4.1. Case I: simulation for stationary series

The performance of the proposed MFLMS algorithm is tested on the stationary series. Data samples of points 900–1600 are used for the training of the filters and sample points 1600–1801 are used as the test or validation set. The weights are adjusted in 50 epochs for these experiments. These weights are then used to validate the test data. The parameters are adjusted as LMS ($\mu = 0.15$), KLMS ($\eta = 0.1$, and Gaussian kernel with a width of 0.1), FLMS ($\mu = 0.019, \mu_{\text{f}} = 0.03$ and $\nu = 0.0972$), and MFLMS ($\mu = 0.08$, $\mu_{\text{f}} = 0.03$,

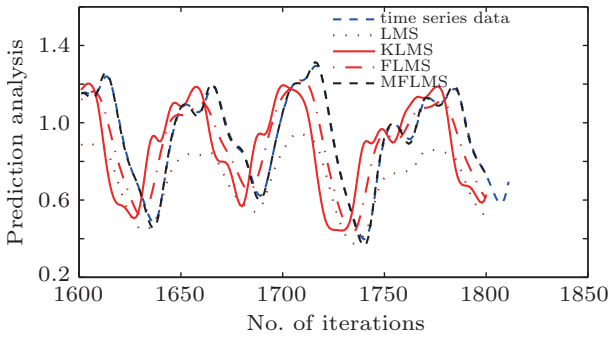


Fig. 3. (color online) Prediction analysis of stationary series without noise.

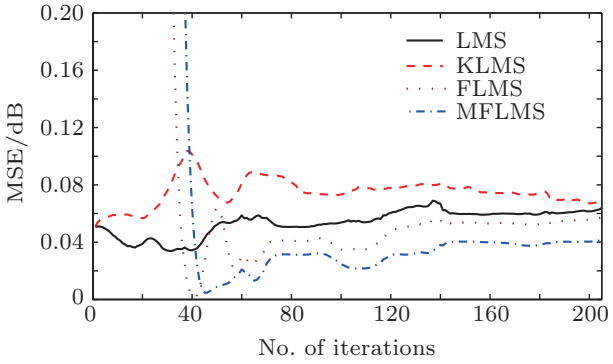


Fig. 4. (color online) Learning curves of stationary series for LMS, KLMS, FLMS, and MFLMS.

$\nu = 0.0972$, and $\beta = 0.48$) for achieving the most optimal results. Figure 3 shows the predictive performance of LMS, KLMS, FLMS, and proposed MFLMS. The superiority of the proposed method is clearly observed. To validate this, the learning curves of these performances are compared in terms of the mean square error displayed in Fig. 4.

4.2. Case II: simulation for stationary series with different noise levels

Experiments are performed with a segment of 500 samples used for training and another 200 for testing. All the stationary time series data are corrupted by Gaussian noise with zero mean and different noise variances to further validate the MFLMS algorithm μ, μ_f, ν, and β. Throughout this set of simulation, the same values of μ, μ_f, ν, and β as that in case I are used. Gaussian kernel with a width of 0.1 is used for KLMS predictor. An independent Monte Carlo simulation runs 100 times with different noise variances σ_n^2 to further validate the MFLMS applicability. The results are summarized in Table 1, which shows that MFLMS is the best performer, specifically in terms of low noise variances. Results are displayed using the form mean plus or minus standard deviation. As observed in Table 1, the MFLMS algorithm performs effectively and consistently in both training and testing data.

Table 1. Performance comparison with different noise levels.

Algorithm	LMS	KLMS	FLMAS	MFLMS
Traning MSE($\sigma = 0.005$)	$0.00250\pm3.85\times10^{-5}$	$0.00454\pm9.26\times10^{-5}$	$0.001827\pm2.93\times10^{-5}$	$0.00156\pm1.28\times10^{-5}$
Testing MSE ($\sigma = 0.005$)	$0.00243\pm4.219\times10^{-5}$	0.04927 ± 0.0001069	$0.0017738\pm3.28\times10^{-5}$	$0.001538\pm1.927\times10^{-5}$
Traning MSE ($\sigma = 0.02$)	0.00303 ± 0.000149	0.005175 ± 0.000358	0.002399 ± 0.000121	$0.0021152\pm6.50\times10^{-5}$
Testing MSE ($\sigma = 0.02$)	0.002970 ± 0.00170	0.005556 ± 0.0004324	0.002356 ± 0.000131	$0.0020892\pm9.49\times10^{-5}$
Traning MSE ($\sigma = 0.04$)	0.004640 ± 0.000297	0.00680 ± 0.000716	0.004117 ± 0.000251	0.0038219 ± 0.0001680
Testing MSE ($\sigma = 0.04$)	0.0046379 ± 0.0003128	0.007162 ± 0.0007767	0.004117 ± 0.000282	0.003822 ± 0.0002391
Traning MSE ($\sigma = 0.1$)	0.015863 ± 0.0009678	0.01832 ± 0.0015963	0.015628 ± 0.0009414	0.015301 ± 0.00086739
Testing MSE ($\sigma = 0.1$)	0.016166 ± 0.001296	0.019066 ± 0.0024241	0.015929 ± 0.0012814	0.0155606 ± 0.0013143
Traning MSE ($\sigma = 0.5$)	0.42356 ± 0.1011	0.20001 ± 0.010242	0.42533 ± 0.1115	0.3209 ± 0.030332
Testing MSE ($\sigma = 0.5$)	0.41752 ± 0.12074	0.29218 ± 0.028178	0.41918 ± 0.12231	$0.0.32013\pm0.047689$

4.3. Case III: simulation for nonstationary series

To examine the behavior of predictors for nonstationary series, we use a modified time series with sinusoid.[16] This new series is generated by adding a sinusoid ($y = 0.3\sin(2\pi ft)$, $f = 1/3000$, and t is a continuous curve between $1:3000$) with an amplitude 0.3 and a frequency of 3000 samples per second to 19. The modified time series is displayed in Fig. 5, with clear indication of training and testing samples.

Training data are taken from the samples 900–1600, as shown in Fig. 5, and testing is performed on the samples 1600–1800. After adjusting the best possible values of the filter parameters, the predictors are then trained to efficiently handle the time varying mean of the test data. The filter parameters are chosen as LMS ($\mu = 0.04$), KLMS ($\eta = 0.2$, and Gaussian kernel with a width of 0.1), FLMS ($\mu = 0.4$, $\mu_f = 0.03$, and $\nu = 0.0972$), and MFLMS ($\mu = 0.01$, $\mu_f = 0.03, \nu = 0.0972$ and $\beta = 0.35$)

accordingly. Figure 6 shows the predictive performance of the algorithms. The authentication towards performance is

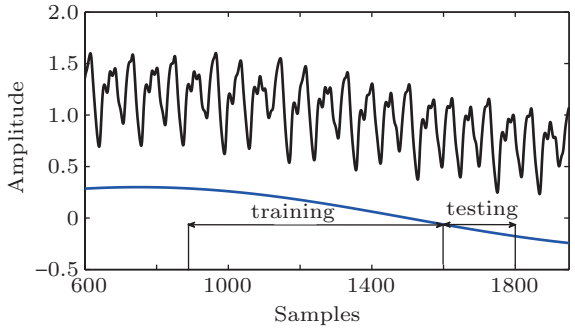


Fig. 5. (color online) Modified time series.

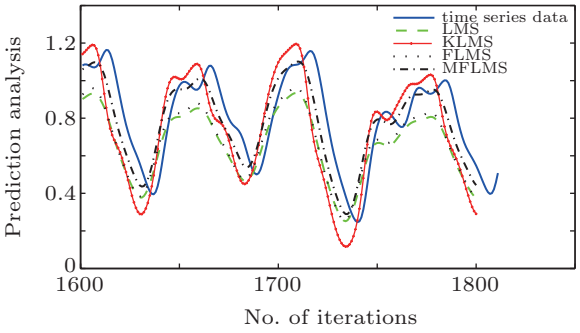


Fig. 6. (color online) Prediction Analysis of modified Time Series

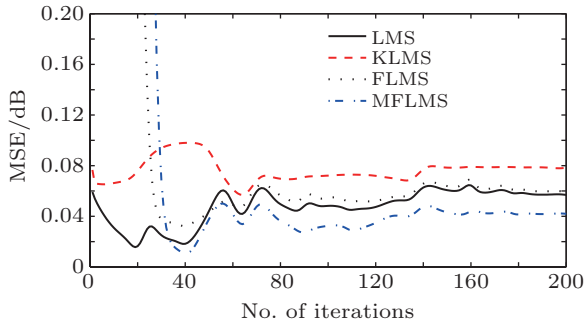


Fig. 7. (color online) Learning curves of nonstationary series.

observed in Fig. 7, which displays the learning curves of the mean square error performance comparison for LMS, KLMS, FLMS, and MFLMS.

4.4. Case IV: simulation for nonstationary series with different noise levels

In Table 2, we summarize the algorithm behavior towards nonstationary series prediction, setting different noise level parameters and testing each case on 100 Monte Carlo simulations. The proposed MFLMS attains the achievable performance. In this experiment, all the filter parameters are alloted as given in case III. The results are summarized in Table 2 after 100 Monte Carlo simulations using the form mean plus or minus standard deviation. The results are displayed with different noise variances σ_n^2 to further validate the MFLMS applicability towards nonstationary series prediction.

Table 2. Performance comparison of modified series with different noise levels.

Algorithm	LMS	KLMS	FLMS	MFLMS
Traning MSE ($\sigma = 0.005$)	0.0075878±0.00018345	0.0042825±6.8525×10^{-5}	0.0095947±0.0002324	0.0096714±0.000123
Testing MSE ($\sigma = 0.005$)	0.003413±6.7908×10^{-5}	0.039595±0.00022386	0.004302±8.8553×10^{-5}	0.004523±5.32×10^{-5}
Traning MSE ($\sigma = 0.02$)	0.0082071±0.00078202	0.0048971±0.00028805	0.01018±0.0010086	0.010158±0.00055381
Testing MSE ($\sigma = 0.02$)	0.0039908±0.00029233	0.040221±0.0010017	0.0048459±0.000381	0.0050118±0.0002199
Traning MSE ($\sigma = 0.04$)	0.010862±0.0016886	0.0066764±0.00070018	0.012715±0.0020468	0.012111±0.0010296
Testing MSE ($\sigma = 0.04$)	0.0060947±0.0006786	0.042194±0.0023461	0.0068924±0.000811	0.0067999±0.00049113
Traning MSE ($\sigma = 0.1$)	0.027542±0.0059527	0.017685±0.0017916	0.029103±0.0065121	0.026047±0.0033939
Testing MSE ($\sigma = 0.1$)	0.019771±0.0025419	0.05688±0.005641	0.020757±0.002782	0.019248±0.0017778
Traning MSE ($\sigma = 0.5$)	0.4559±0.11013	0.21233±0.015882	0.45755±0.11098	0.37931±0.058416
Testing MSE ($\sigma = 0.5$)	0.40591±0.082598	0.34762±0.03449	0.40716±0.083371	0.0.34391±0.049129

5. Conclusion and future work

In this paper, a new modified fractional least mean square algorithm is presented, which intrinsically provides a modification towards FLMS algorithm with an adjustable gain parameter. Fractional and kernel Least mean square algorithms are also discussed. One application of predicting a chaotic Mackey–Glass time series is presented, which illustrates the performance of the proposed algorithm in comparison to KLMS, LMS, and FLMS. The learning curves of these algorithms are observed and analyzed for both the stationary and nonstationary chaotic series. The proposed method is also tested on both series with different noise levels. This new formulation hopefully offers a new era towards nonlinear fractional signal processing algorithms.

References

[1] Miller K S and Ross B 1993 *An Introduction to the Fractional Calculus and Fractional Derivative* (New York: John Willey & Sons)

[2] Samko S, Killbas A and Marichev O 1993 *Fractional Integrals and Derivatives: Theory and Applications* (Verdon: Breach Science Publishers)

[3] Ross B 1992 *Int. J. Math. Statist. Sci.* **1** 21

[4] Ross B 1997 *Historia Math.* **4** 75

[5] Anatoly K A, Srivastav H M and Trijillo J J 2006 *Theory and Applications of Fractional Differential Equations* (North Holland: Mathematics Studiesn)

[6] Raja M A, Qureshi I M and Khan J A 2009 *European J. Scientific Res.* **1** 14

[7] Geravanchizadeh, Masood and Sina G O 2011 19th *Iranian Conference on Electrical Engineering*, May 17–19, 2011, Tehran, Iran

[8] Akhtar P and Yasin M 2012 2nd *Intl. Multitopic Conf. IMTIC*, March 28–30, 2012, Jamshoro, Pakistan

[9] Li X T and Yin M H 2012 *Chin. Phys. B* **21** 050507

[10] Ali H S 2003 *Fundamental of Adaptive Filtering* (New York: John Willey & Sons)

[11] Bernard W and Streams S D 1985 *Advance Signal Processing* (New Jersy: Prentice Hall)

[12] Pokharel P, Liu W and Principe J C 2009 *Signal Process.* **89** 257

[13] Liu W, Principe J C and Haykin S 2010 *Kernel Adaptive Filtering* (New York: John Willey & Sons)

[14] Liu W, Pokharel P and Principe J C 2008 *IEEE Trans. Signal Process.* **56** 543

[15] Liu W and Principe J C 2008 *Eurasip J. Adv. Signal Process.* **89** 257

[16] Cheng E S, Chen S and Mulgrew B 1996 *IEEE Trans. Neural Networks* **7** 190