Adaptive step-size modified fractional least mean square algorithm for chaotic time series prediction*

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

[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 8 October 2013; revised manuscript received 21 November 2013; published online 1 April 2014)

This paper presents an adaptive step-size modified fractional least mean square (AMFLMS) algorithm to deal with a nonlinear time series prediction. Here we incorporate adaptive gain parameters in the weight adaptation equation of the original MFLMS algorithm and also introduce a mechanism to adjust the order of the fractional derivative adaptively through a gradient-based approach. This approach permits an interesting achievement towards the performance of the filter in terms of handling nonlinear problems and it achieves less computational burden by avoiding the manual selection of adjustable parameters. We call this new algorithm the AMFLMS algorithm. The predictive performance for the nonlinear chaotic Mackey Glass and Lorenz time series was observed and evaluated using the classical LMS, Kernel LMS, MFLMS, and the AMFLMS filters. The simulation results for the Mackey glass time series, both without and with noise, confirm an improvement in terms of mean square error for the proposed algorithm. Its performance is also validated through the prediction of complex Lorenz series.

Keywords: fractional LMS, kernel LMS, Reimann–Lioville derivative, Mackey glass and Lorenz time series

PACS: 05.45.Df, 05.30.Pr, 05.45.Tp, 92.60.Wc **DOI:** 10.1088/1674-1056/23/5/050503

1. Introduction

The theory of fractional calculus is as old as that of classical calculus and it is currently drawing considerable attention in many scientific areas.[1,2] Successful applications have been found in signal processing, robotics, system identification, robotics, system identification, genetic algorithm, control systems, and finance and economics.[3–5] Much work has been carried out for the solution of linear, nonlinear, and fractional order systems, which has motivated researchers to analyze numerical computing in this field.[6,7] In short, fractional calculus has found wide applications for numerous, multifarious problems in engineering. Possible advancements in the field of signal processing have recently been observed through improvement in the design of variable fractional order finite impulse response (FIR) differentiators.[9]

The fractional signal processing approach is gaining great interest nowadays from scientists and engineers. One such development in this direction is the fractional least mean square algorithm.[10] The fractional derivative term is added to the classical stochastic gradient LMS algorithm to adapt the parameters and also to improve its convergence behavior. There have been many successful applications, including the parameter identification of input nonlinear controlled autoregressive system and the modeling of fractional Brownian motion.[11] A normalized fractional least mean square algorithm has also been presented for dual channel speech enhancement[12] in the literature to exhibit the performance of this fractional algorithm.

Despite its advantages, this algorithm suffers from somewhat slow convergence behavior. As in the case of classical LMS algorithm, researchers have been constantly searching for an alternative means to improve its convergence and performance. A traditional but powerful approach is to employ a time-varying step size in the weight update recursion of the standard LMS.[8] Based on this idea, when the algorithm is far from the optimal solution, large values of the step size are chosen so that it further approaches to its optimum, small steps are used to lower the misadjustments and to achieve better performance. This can be done by adjusting the values of the step size according to some criteria that may offer an approximation of the adaptation process. Several criteria have been used, including changing the sign of the gradient samples, reducing the squared error of each state or instant, cross correlation of error and input, time averaged autocorrelation of the *posteriori* and *a priori* error.[14–16]

The main contribution of this article is the development of an adaptive step-size modified fractional least mean square (AMFLMS) algorithm and its application to chaotic Mackey glass time series prediction. Different cases are discussed to validate and counter the performance of the proposed algorithm. To avoid the manual selection of adaptive step size pa-

*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

rameters and the order of fractional derivative, a mechanism is introduced here to tune them adaptively through a gradient-based approach.

The organization of this paper is as follows. A brief summary of the LMS and KLMS algorithms is presented in Section 2. Section 3 gives a detailed description of FLMS, MFLMS, and the proposed AMFLMS. Section 4 discusses the problem setting mechanism for time series prediction. The results and detail simulations are presented in Section 5. Section 6 makes a number of recommendations for future research directions.

2. Least mean square and kernel least mean square algorithms

In order to minimize the empirical risk by using the example sequence $[\boldsymbol{x}(i), d(i)]$ through a stochastic gradient method[18]

$$J(n) = \frac{1}{2}E[|e(n)|^2] = \frac{1}{2}\sum_{i=0}^{n}(d(i) - w^{\mathrm{T}}(i)\boldsymbol{x}(i))^2, \tag{1}$$

where $e(n)$ is the error at each instant of time and the gradient of 1 with respect to $\boldsymbol{w}$ is given by

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

2.1. Least mean square algorithm

i) The tap-weight adaptation process for the LMS algorithm is

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

ii) The estimation error is

$$e(i) = d(i) - y(i). \tag{4}$$

iii) The filter output is

$$y(i) = \boldsymbol{w}^{\mathrm{T}}(i)\boldsymbol{x}(i). \tag{5}$$

Here, $e(i)$ $(i = 1, 2, \ldots, n)$ defines the estimation error and μ is the step size parameter of the filter. It is customary to initialize the weight vector to zero.

2.2. Kernel least mean square algorithm

A degradation in performance of the LMS algorithm can be expected when the difference or the mapping between the input vector $\boldsymbol{x}$ and the desired vector $\boldsymbol{d}$ is highly nonlinear. A nonlinear mapping mechanism is introduced in Ref. [14] and [15] as $\boldsymbol{\varphi}(\boldsymbol{x}(i))$, which gives a powerful model $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}(i))$ than $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$. Hence, using this model, and finding $\boldsymbol{w}$ through the stochastic gradient technique, may provide an efficient mechanism towards nonlinear filtering (as LMS ensures for linear problems). Thus, by using the example sequence $[\boldsymbol{\varphi}(i), d(i)]$, the KLMS algorithm is described as

$$e(i) = d(i) - \boldsymbol{w}^{\mathrm{T}}(i-1)\boldsymbol{\varphi}(i). \tag{6}$$

Through iteration, the repeated application of the weight-update equation (6) yields

$$\boldsymbol{w}(i) = \boldsymbol{w}(i-1) + \eta e(i)\boldsymbol{\varphi}(i), \tag{7}$$

where η is the step size parameter. If we take $w(0) = 0$, then

$$\boldsymbol{w}(i) = \eta\sum_{j=1}^{n} e(j)\boldsymbol{\varphi}(j). \tag{8}$$

Substituting Eq. (8) into Eq. (5), we have

$$y = \eta\sum_{j=1}^{i} e(j) < \boldsymbol{\varphi}(j), \boldsymbol{\varphi}(i) > = \eta\sum_{j=1}^{n} e(j)k(\boldsymbol{x}(j), \boldsymbol{x}(i)). \tag{9}$$

It is clear from Equation (9) that the output of the KLMS is the sum of all past errors multiplied by the kernel evaluation of the previously received data. The η is the step size parameter for the kernel LMS, and k is the kernel function using Gaussian distribution.

3. Fractional least mean square algorithm

3.1. Introduction to fractional derivative

A type of nonlinear adaptive algorithm was introduced by Raja and Qureshi[10] in their work on the identification of autoregressive (AR) systems. Since its organization, it has been successfully utilized in various fields of signal processing, such as acoustic echo cancellation, dual channel speech enhancement and performance analysis of the Bessel beamformers.[11,12] Before discussing the FLMS algorithm, we will present the Rieman–Lioville fractional derivative method.[3]

$$(D^{\nu}f)(t) = \frac{1}{\Gamma(\alpha)}\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)^{n}\int_0^t (t-\tau)^{n-\nu-1}f(\tau)\,\mathrm{d}\tau, \tag{10}$$

where D^{ν} is the fractional derivative, and n is an integer. It is notable that we apply the Reimann–Lioville fractional derivative to power function $f(t) = (t-a)^{\alpha}$. Here a and α are real constants. It can be evaluated as

$$D^{\nu}(t-a)^{\alpha} = \frac{\Gamma(1+\alpha)}{\Gamma(1+\alpha+\nu)}(t-a)^{\alpha-\nu}. \tag{11}$$

3.2. Derivation of the FLMS algorithm

First we start with the standard LMS filter.[17] The error is minimized according to the cost function given by Eq. (1). The adaptive weight mechanism of the LMS filter considers only the first derivative. We use here the fractional derivative in addition to the first derivative; that is,

$$w_k(n+1) = w_k(n) - \mu\left(\frac{\partial J(n)}{\partial w_k}\right) - \mu_{\mathrm{f}}\left(\frac{\partial}{\partial w_k}\right)^{f} J(n), \tag{12}$$

$$w_k(n) = w_k(n-1) + \mu e(n)x(n-k) + \mu_f e(n)x(n-k)\frac{w_k^{1-f}(n)}{\Gamma}(2-f), \quad (13)$$

where μ_f is the fractional step size, and f is the order of fractional derivative, typically between 0 and 1. To pace up the computation for equation of nonlinear nature in cyclic single step iteration, we assume $w_k^{1-f}(n) \cong w_k^{1-f}(n-1)$, and then we have

$$w_k(n) = w_k(n-1) + \mu e(n)x(n-k) + \mu_f e(n)x(n-k)\frac{w_k^{1-f}(n-1)}{\Gamma(2-f)}. \quad (14)$$

3.3. Modified FLMS Algorithm

A mechanism is introduced[23] to sidestep the computation of Gamma function, which is probably one of the major functions of fractional calculus. By gripping this function in the term μ_f, and presenting the adjustable gain parameter β, we reformulate (14) as

$$w_k(n) = w_k(n-1) + \beta\mu e(n)x(n-k) + (1-\beta)\mu_f e(n)x(n-k)w_k^{1-f}(n), \quad (15)$$

where β is adjusted manually from 0 and 1. We 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.

3.4. Proposed AMFLMS algorithm

The variable step size algorithm is introduced here to eliminate the "guesswork" in the selection of step size parameters. In order to speed up the performance and convergence of the modified FLMS algorithm, the step size parameters μ, μ_f, and f are made to be adaptive by using the gradient-based approach. Let us consider the goal of reducing the squared estimation error at each instant of time by adapting these parameters.

3.4.1. Adaptation of parameter μ

First, we consider the goal to adapt the step size μ in Eq. (15) in order to reduce the squared estimation error at each instant of time

$$\mu(n+1) = \mu(n) - \frac{\alpha}{2}\nabla_\mu(n) = \mu(n) - \frac{\alpha}{2}\left[\frac{\partial e(n)}{\partial\mu(n)}e^*(n) + \frac{\partial e^*(n)}{\partial\mu(n)}e(n)\right]. \quad (16)$$

Here, we only consider the real case where α is a small positive constant.

$$\frac{\partial e(n)}{\partial\mu(n)} = \frac{\partial}{\partial\mu(n)}[d(n) - \boldsymbol{w}^{\mathrm{T}}(n)\boldsymbol{x}(n)] = -\frac{\partial\boldsymbol{w}^{\mathrm{T}}(n)}{\partial\mu(n)}\boldsymbol{x}(n) = -\boldsymbol{\Psi}^{\mathrm{T}}(n)\boldsymbol{x}(n). \quad (17)$$

Inserting the value of Eq. (17) into Eq. (16), we have

$$\mu(n+1) = \mu(n) - \frac{\alpha}{2}\left[-2e(n)\frac{\partial e(n)}{\partial\mu(n)}\right] = \mu(n) + \alpha e(n)\boldsymbol{\Psi}^{\mathrm{T}}(n)\boldsymbol{x}(n). \quad (18)$$

Now we formulate the vector $\Psi(n)$, denotes the weight vector with respect to the step size parameter μ according to Eq. (15) as

$$\frac{\partial w_k(n+1)}{\partial\mu(n)} = \frac{\partial w_k(n)}{\partial\mu(n)} + \beta e(n)x(n-k) + \beta\mu(n)x(n-k)\frac{\partial e(n)}{\partial\mu(n)} + (1-\beta)\mu_f(n)\frac{\partial e(n)}{\partial\mu(n)}x(n-k)w_k^{1-f}(n) + (1-\beta)\mu_f(n)e(n)x(n-k)\frac{\partial w_k^{1-f}(n)}{\partial\mu(n)}. \quad (19)$$

Then, we have

$$\begin{aligned}&\Psi_k(n+1)\\ &= \Psi_k(n) + \beta e(n)x(n-k) + \beta\mu(n)x(n-k)(\boldsymbol{\Psi}^{\mathrm{T}}(n)\boldsymbol{x}(n))\\ &\quad + (1-\beta)\mu_f(n)(\boldsymbol{\Psi}^{\mathrm{T}}(n)\boldsymbol{x}(n))x(n-k)w_k^{1-f}(n)\\ &\quad + (1-\beta)\mu_f(n)e(n)x(n-k)(1-f)w_k^{-f}(n)\Psi_k(n). \quad (20)\end{aligned}$$

In order to find an estimate of the particular value of the parameter μ that minimizes the cost function 1, we follow the process of adaptation in Eqs. (15)–(20).

3.4.2. Adaptation of parameter μ_f

Now, we consider the goal to adapt the fractional step size parameter μ_f. The procedure is described below.

$$\mu_f(n+1) = \mu_f(n) - \frac{\alpha_f}{2}\nabla_{\mu_f}(n) = \mu_f(n) - \frac{\alpha_f}{2}\left[-2e(n)\frac{\partial e(n)}{\partial\mu_f(n)}\right], \quad (21)$$

where α_f is a small positive constant,

$$\frac{\partial e(n)}{\partial\mu_f(n)} = -\frac{\partial\boldsymbol{w}^{\mathrm{T}}(n)}{\partial\mu_f(n)}\boldsymbol{x}(n) = -\boldsymbol{\Psi}_f^{\mathrm{T}}(n)\boldsymbol{x}(n), \quad (22)$$

and

$$\mu_f(n+1) = \mu_f(n) + \alpha_f e(n)(\boldsymbol{\Psi}_f^{\mathrm{T}}(n)\boldsymbol{x}(n)). \quad (23)$$

We formulate $\boldsymbol{\Psi}(n)$ according to Eq. (15) as

$$\frac{\partial w_k(n+1)}{\partial\mu_f(n)} = \frac{\partial w_k(n)}{\partial\mu_f(n)} + \beta\mu(n)\frac{\partial e(n)}{\partial\mu_f(n)}x(n-k) + (1-\beta)e(n)x(n-k)w_k^{1-f}(n)$$

$$+(1-\beta)\mu_{\rm f}(n)\frac{\partial e(n)}{\partial \mu(n)}x(n-k)w_k^{1-f}(n)$$
$$+(1-\beta)\mu_{\rm f}(n)e(n)x(n-k)\frac{\partial w_k^{1-f}(n)}{\partial \mu_{\rm f}(n)}, \quad (24)$$

$$\Psi_{k,f}(n+1)$$
$$= \Psi_{k,f}(n)+\beta\mu(n)(\boldsymbol{\Psi}^{\rm T}(n)\boldsymbol{x}(n))$$
$$+(1-\beta)e(n)x(n-k)w_k^{1-f}(n)$$
$$-(1-\beta)\mu_{\rm f}(n)(\boldsymbol{\Psi}^{\rm T}(n)\boldsymbol{x}(n))x(n-k)w_k^{1-f}(n)$$
$$+(1-\beta)\mu_{\rm f}(n)e(n)x(n-k)(1-f)w_k^{-f}(n)\Psi_{k,f}(n). \quad (25)$$

3.4.3. Adaptation of the order of fractional derivative f

The proposed method for adapting the order of fractional derivative by using the gradient descent approach is given by

$$f(n+1) = f(n)-\frac{\alpha_{\rm fr}}{2}\nabla_{\rm f}(n)$$
$$= f(n)+\alpha_{\rm fr}e(n)\frac{\partial e(n)}{\partial f(n)}, \quad (26)$$

where $\alpha_{\rm fr}$ is a small positive constant,

$$\frac{\partial e(n)}{\partial f(n)} = -\frac{\partial \boldsymbol{w}^{\rm T}(n)}{\partial f(n)}\boldsymbol{x}(n) = -\boldsymbol{\Phi}_{\rm f}^{\rm T}(n)\boldsymbol{x}(n). \quad (27)$$

Now we formulate the adaptation rule for f as

$$f(n+1) = f(n)+\alpha_{\rm fr}e(n)(\boldsymbol{\Phi}_{\rm f}^{\rm T}(n)\boldsymbol{x}(n)). \quad (28)$$

Differentiating Eq. (15) with respect to $f(n)$, we have

$$\Phi_{k,f}(n+1)$$
$$= \Phi_{k,f}(n)-\beta\mu(n)(\boldsymbol{\Phi}^{\rm T}(n)\boldsymbol{x}(n))$$
$$+(1-\beta)e(n)x(n-k)w_k^{1-f}(n)$$
$$-(1-\beta)\mu_{\rm f}(n)(\boldsymbol{\Phi}^{\rm T}(n)\boldsymbol{x}(n))x(n-k)w_k^{1-f}(n)$$
$$+(1-\beta)\mu_{\rm f}(n)e(n)x(n-k)(-w_k^{1-f}(n)\ln w_k(n)). \quad (29)$$

The order of the fractional derivative is adjusted adaptively through the method described by Eqs. (26)–(29).

4. Prediction problem setting

One of the most celebrated problems in signal processing is to predict a future value in nonlinear time series analysis. Prediction involves the method of taking advantage of time series history to estimate the future values in the series. A time series can be defined as a sequence of vectors or scalers depending upon time. Let us consider a time series of $\{x(n_0),x(n_1),x(n_2),\ldots,x(n_{k-1}),x(n_k),x(n_{k+1}),x(n_{k+2}),\ldots\}$ To predict the future value at a certain time, a tapped delay line is created through a process called embedding. The output is a matrix containing a tapped delay line shifted by one time sample in its each column, and will be written as

$$\begin{pmatrix} x(n_k) & x(n_{k+1}) & \ldots & x(n_{k+N-1}) \\ x(n_{k-1}) & x(n_k) & \ldots & x(n_{k+N-2}) \\ x(n_{k-2}) & x(n_{k-1}) & \ldots & x(n_{k+N-3}) \\ \ldots & \ldots & \ldots & \ldots \\ x(n_{k-M-1}) & x(n_{k-M}) & \ldots & x(n_{k+N-M}) \end{pmatrix}. \quad (30)$$

The number of columns represents the input patterns that the predictor used for training or testing. M is the number of events in the past or the order of the filter. Now, let us consider the first input pattern, which contain M events. This vector is then delivered to the predictor in order to estimate the future value using the weight vector. Using a law based on the function of mean square error (MSE), we update this weight vector as

$$e = |x(n_{k+1}-\widehat{x}(n_{k+1}))|^2. \quad (31)$$

The above-mentioned cost function (31) is used to adapt the parameter of LMS, MFLMS, and AMFLMS. The formulation of KLMS is a little bit different because it uses error vector obtained during training to test the validation set or desired data samples. The architecture of the predictor is given in Fig. 1.

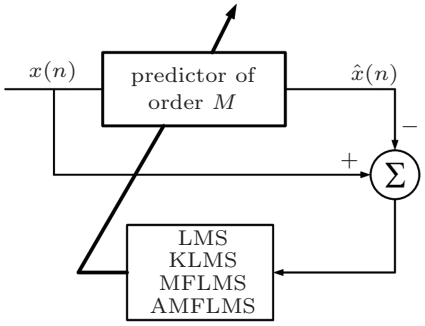


Fig. 1. Architecture of the predictor.

5. Experimental results

In this section we will demonstrate the performance of the proposed algorithm. First, we evaluate the algorithm by considering the example of Mackey glass series under stationary as well as nonstationary situation, and also with different noise levels. Next, a Lorenz time series is considered, and only the performance is evaluated on X component of the series. We mainly focus on the comparison of different algorithms with a gradient-based approach.

5.1. Mackey glass time series

The experimental results of prediction using the LMS, KLMS, MFLMS, and AMFLMS predictors are discussed. To counter the performance of the proposed filter, a Chaotic Mackey–Glass time series is generated by the following delay differential equation:

$$\frac{{\rm d}x(t)}{{\rm d}t} = -bx(t)+\frac{ax(t-\tau)}{1+x^{10}(t-\tau)}. \quad (32)$$

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

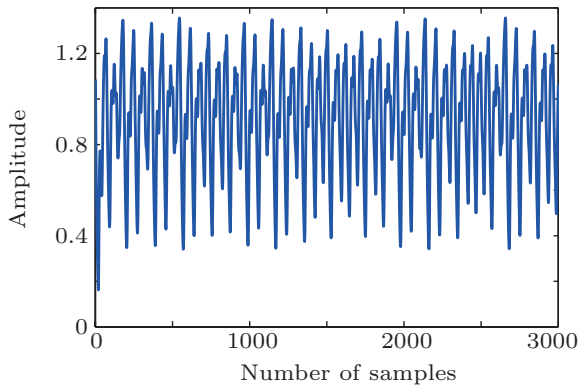


Fig. 2. (color online) Mackey–Glass time series.

5.1.1. Case I: simulation for stationary series

The experiment is performed to test the proposed AMFLMS algorithm on the stationary series. Sample points 1100–1600 are used to train and also to adjust the weights of the filter. These weights are then used to validate the test samples 1600–1800. The parameters of the filters are adjusted as: LMS ($\mu = 0.015$), KLMS ($\eta = 0.2$), and a Gaussian kernel with a width of 0.1 is used, MFLMS (($\mu = 0.08, \mu_f = 0.03, f = 0.0972$, $\beta = 0.48$)), and AMFLMS (($\beta = 0.48$)). The predictive performances of LMS, KLMS, MFLMS, and the proposed AMFLMS are depicted in Fig. 3. To validate superiority, the learning curves of the MSE are plotted in Fig. 4.

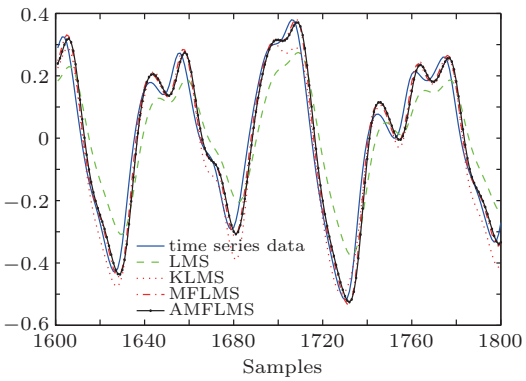


Fig. 3. (color online) Prediction analysis of stationary time series.

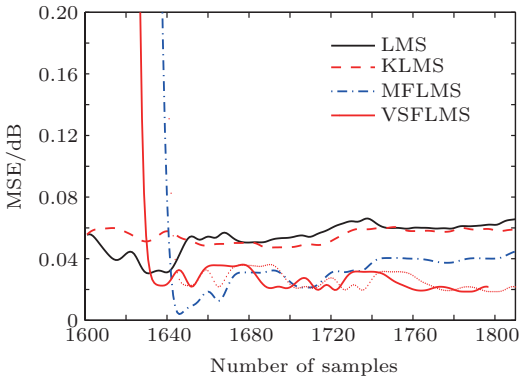


Fig. 4. (color online) Learning curves of stationary time series.

The mean behaviors of the step size parameter μ and μ_f are depicted in Fig. 5 for the stationary series during training. One can easily observe that value of both these parameters goes up very fast initially and then settles. This behavior clearly explains the fast convergence. In Fig. 6, the value of the fractional order derivative is plotted.

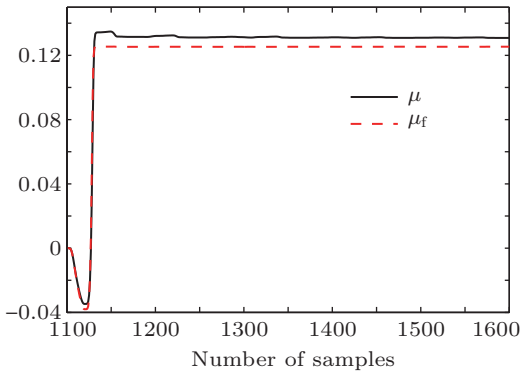


Fig. 5. (color online) Behavior of μ and μ_f during training.

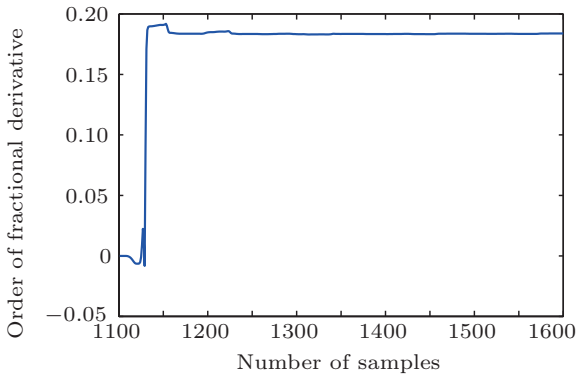


Fig. 6. (color online) Fractional order derivative during training in stationary environment.

5.1.2. Case II: simulation with stationary series with different noise levels

All the stationary of time series data is corrupted by the Gaussian noise with zero mean and different noise variances σ_n^2 to further validate the proposed algorithm. The values of

Table 1. Performance comparison of stationary series with different noise levels.

Algorithm	LMS	KLMS	MFLMS	AMFLMS
Traning MSE($\sigma = 0.005$)	$0.0024785 \pm 1.7523 \times 10^{-5}$	$0.004514 \pm 6.1332e \times 10^{-18}$	$0.001795 \pm 2.1904 \times 10^{-18}$	$0.001625 \pm 1.7523 \times 10^{-18}$
Testing MSE($\sigma = 0.005$)	$0.0024785 \pm 4.5714 \times 10^{-19}$	$0.0049036 \pm 6.1332 \times 10^{-18}$	$0.001795 \pm 2.1904 \times 10^{-18}$	$0.064462 \pm 2.8037e \times 10^{-17}$
Traning MSE($\sigma = 0.02$)	$0.0024785 \pm -4.571 \times 10^{-19}$	$0.06970 \pm 1.4101 \times 10^{-17}$	$0.02247 \pm -3.657e \times 10^{-18}$	$0.0017952 \pm 2.2857e \times 10^{-19}$
Testing MSE($\sigma = 0.02$)	$0.0024075 \pm 4.571 \times 10^{-19}$	$0.074906 \pm 1.4628 \times 10^{-17}$	$0.03756 \pm 3.868 \times 10^{-18}$	$0.067227 \pm 1.48932e \times 10^{-19}$
Traning MSE($\sigma = 0.04$)	0.0046267 ± 0.00028867	0.009670 ± 0.0005167	0.002137 ± 0.000398	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 ± 0.0009315	0.018072 ± 0.0019643	0.015315 ± 0.00091788	0.012115 ± 0.000808
Testing MSE($\sigma = 0.1$)	0.016209 ± 0.0014763	0.018942 ± 0.0021438	0.073852 ± 0.00484	0.01213 ± 0.0013113
Traning MSE($\sigma = 0.5$)	0.47252 ± 0.24589	0.20107 ± 0.01235	0.3209 ± 0.058172	0.4908 ± 0.34772
Testing MSE($\sigma = 0.5$)	0.4487 ± 0.18748	0.29321 ± 0.032418	0.34689 ± 0.090635	0.46899 ± 0.09916

the step size are adjusted accordingly, in the same way as Case 1. In this experiment, 200 independent Monte Carlo simulations are performed to test the proposed algorithm, and the results are displayed in Table 1.

It is clear from Table 1 that the proposed AMFLMS algorithm gives the best results at low noise levels, but its performance is slightly degraded in the presence of high noise levels (as mentioned in the table for noise level $\sigma_n^2 = 0.5$). Noise $\sigma_n^2 = 0.1$ is added to the stationary series and the MSE is plotted in Fig. 7. The performance of AMFLMS is still better.

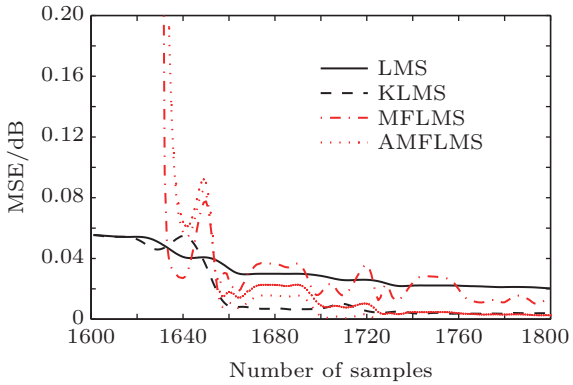


Fig. 7. (color online) Learning curves of time series with noise level σ_n^2=0.1.

5.1.3. Case III: experiment with nonstationary series

In order to examine the behavior and to further validate the performance of the proposed algorithm, experiments are performed on the nonstationary time series[23] data. We generate a modified series by adding a sinusoid ($y = 0.3\sin(2\pi ft)$, $f = 1/3000$, t is a continuous curve 1:3000) with an amplitude of 0.3 and a frequency of 3000 samples per second to stationary Mackey glass time series, as shown in Fig. 8 with the indication of training and testing samples.

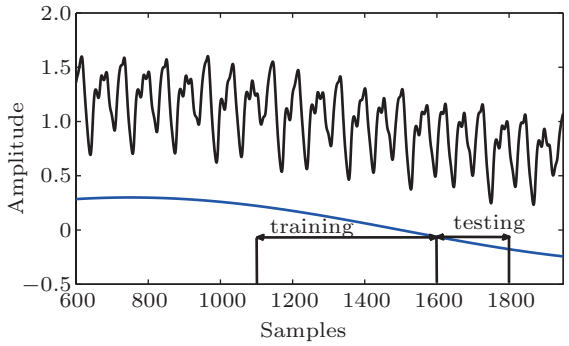


Fig. 8. (color online) Modified time series.

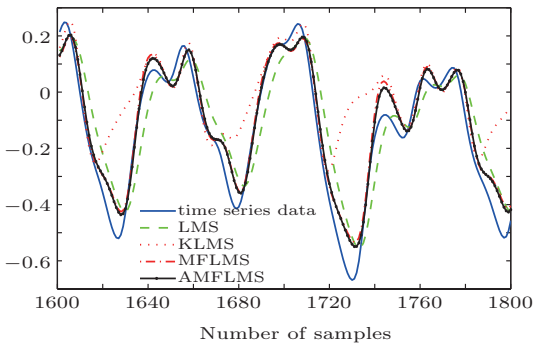


Fig. 9. (color online) Prediction analysis of modified time series.

Next, we perform the experiment to test the proposed algorithm on nonstationary series with a time varying mean. Training data is taken from samples 1100–1600. After adjusting the parameters of the filters, the predictors are then given for the test data of sample points 1600–1800, and their predictive performance are shown in Fig. 9. The filter parameters are selected as: LMS ($\mu = 0.02$), KLMS ($\eta = 0.2$), and Gaussian kernel with a width of 0.1, MFLMS ($\mu = 0.01$, $\mu_f = 0.03$, $f = 0.0972$, $\beta = 0.45$, and AFLMS $\beta = 0.45$. The performance comparison of the proposed algorithm with LMS,

KLMS, and MFLMS is detailed in terms of MSE in Fig. 10. The results show the superiority of the proposed AMFLMS filter.

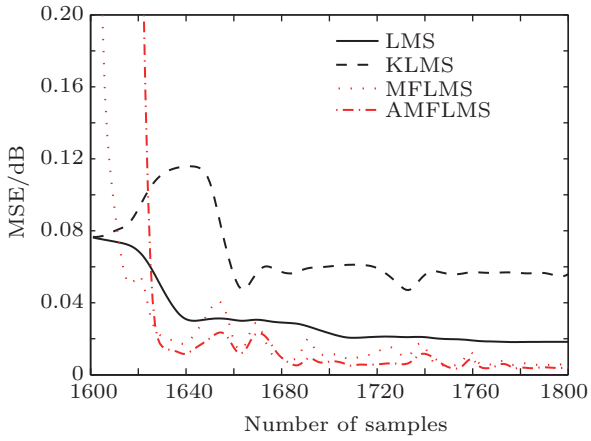


Fig. 10. (color online) Learning curves of nonstationary series.

The behaviors of the step size parameters μ, μ_f and f are displayed in Fig. 11 during training of nonstationary series.

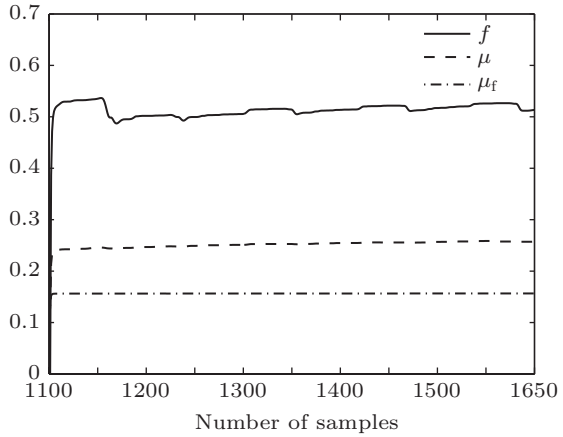


Fig. 11. Behavior of μ, μ_f and f during training of nonstationary series.

In order to further validate the performance of the proposed algorithm, the experiment is performed with a fewer number of training samples for modified nonstationary series. 200 samples (1600–1800) are taken to train the algorithms, while another 300 sample points (1600–1900) are used as a test or validation set for the LMS, KLMS, MFLMS, and AMFLMS. The results are displayed in terms of MSE in Fig. 12, which shows that the performance of the proposed AMFLMS algorithm is better than that of the other algorithms.

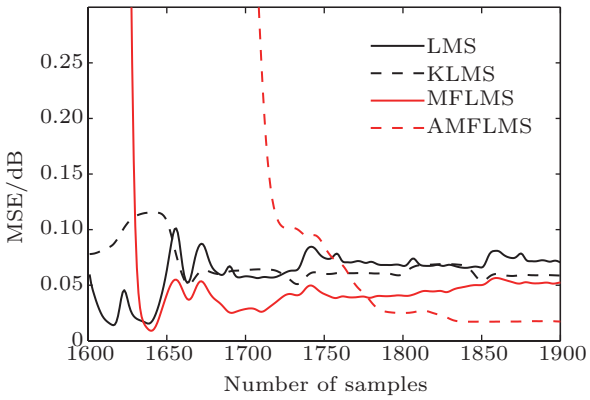


Fig. 12. (color online) Learning curves of modified series with different training and test samples.

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

In Table 2, we summarize the behavior of the filters towards nonstationary time series prediction with different noise variances σ_n^2. The results are displayed after 200 independent runs of Monte Carlo simulations. The results show that the proposed algorithm achieves better performance in terms of low and high noise levels.

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

Algorithm	LMS	KLMS	MFLMS	AMFLMS
Traning MSE($\sigma = 0.005$)	$0.0024999\pm2.9452\times10^{-5}$	$0.004534\pm6.0666\times10^{-5}$	$0.001818\pm2.3034\times10^{-5}$	$0.001646\pm9.033\times10^{-6}$
Testing MSE($\sigma = 0.005$)	$0.002487\pm3.5334\times10^{-5}$	$0.0049278\pm6.8589\times10^{-5}$	0.067229 ± 0.00021486	0.064473 ± 0.0001984
Traning MSE($\sigma = 0.02$)	0.0030239 ± 0.00013662	0.005153 ± 0.00038711	0.0023982 ± 0.000117	$0.0022105\pm5.4614\times10^{-5}$
Testing MSE($\sigma = 0.02$)	0.002944 ± 0.00017584	0.0055061 ± 0.00040318	0.067354 ± 0.0010987	0.064812 ± 0.00095545
Traning MSE($\sigma = 0.04$)	0.0046267 ± 0.00028867	0.0070108 ± 0.0007506	0.0041104 ± 0.000254	0.0039353 ± 0.00019148
Testing MSE($\sigma = 0.04$)	0.0045764 ± 0.00033865	0.0073131 ± 0.00084083	0.0067554 ± 0.002234	0.065667 ± 0.00019555
Traning MSE($\sigma = 0.1$)	0.015673 ± 0.0009315	0.018072 ± 0.0019643	0.015461 ± 0.0009336	0.015315 ± 0.0009178
Testing MSE($\sigma = 0.1$)	0.016209 ± 0.0014763	0.018942 ± 0.0021438	0.073457 ± 0.0057177	0.073852 ± 0.00484
Traning MSE($\sigma = 0.5$)	0.47252 ± 0.24589	0.20107 ± 0.01235	0.47498 ± 0.24988	0.329 ± 0.058172
Testing MSE($\sigma = 0.5$)	0.4487 ± 0.18748	0.29321 ± 0.032418	0.49428 ± 0.25345	0.34689 ± 0.090365

Noise with variance of σ_n^2 is added to the modified time series and the performance of LMS, KLMS, MFLMS and AMFLMS is observed in terms of MSE, as shown in Fig. 13.

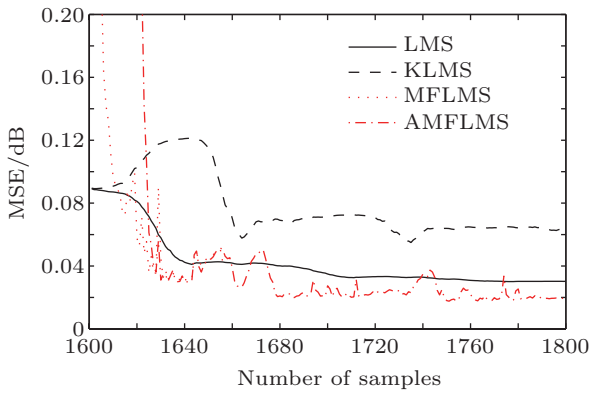


Fig. 13. (color online) Learning curves of modified series with noise $\sigma_n^2 = 0.15$.

5.2. Lorenz time series

The Lorenz time series represents a dynamical system with chaotic flow, it is also prominent for its unique butterfly shape. The Lorenz system is nonlinear, three-dimensional, and deterministic, which can be described by the following set of differential equations:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y(t)-x(t)), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= -x(t)z(t)+\gamma x(t)-y(t), \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= x(t)y(t)-Bz(t). \end{aligned} \tag{33}$$

The values of the unknown parameters upon which the Lorenz system exhibits chaotic behavior are set as $\sigma = 10$, $\gamma = 28$, and $B = 8/3$. Consider the initial values as $x(0) = 1$, $y(0) = 1$, $z(0) = 1$, and a sampling period of 0.01 second is fixed to obtain the sample data using a first-order approximation method. The state trajectory of the Lorenz system is depicted in Fig. 14.

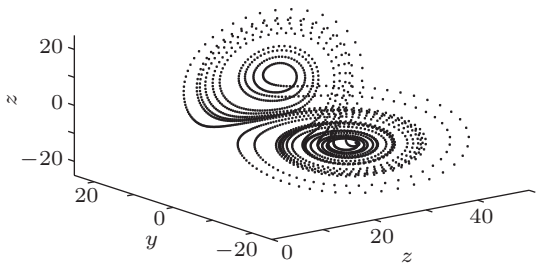


Fig. 14. State trajectory of the lorenz system.

The X component of the Lorenz series is plotted in Fig. 15. Another experiment is then performed with training sample points 1100–1600 and test sample points 1600–2000 to evaluate the performance of the filters as depicted in Fig. 16. To further validate the performance of the proposed algorithm, the MSE is plotted in Fig. 17.

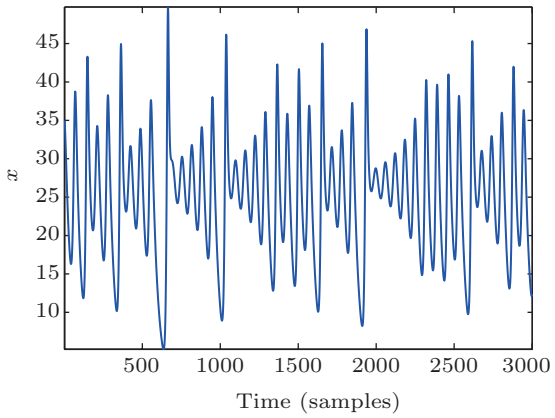


Fig. 15. (color online) The x component of the Lorenz time series.

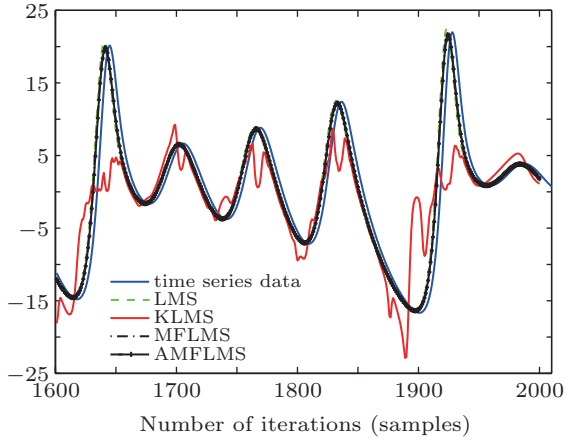


Fig. 16. (color online) Prediction analysis of the x component of Lorenz time series during training.

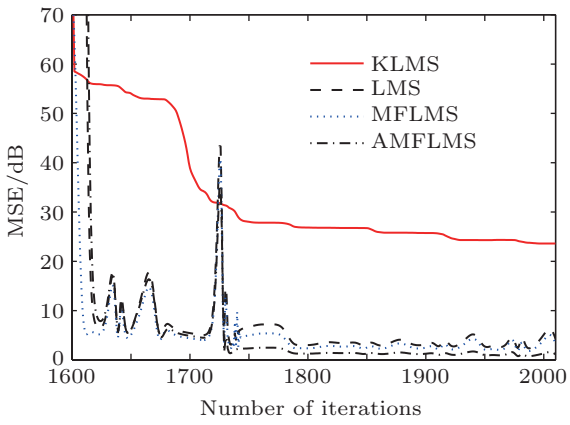


Fig. 17. (color online) Learning curves for the x component of Lorenz series.

6. Conclusion and future work

An adaptive step size modified fractional LMS algorithm employing a time varying step size parameters in the weight

adaptation equation is introduced. These parameters are adjusted according to the square of a time averaged error through gradient-based technique. As a result, the proposed algorithm can effectively handle the prediction of chaotic series. An application of predicting a chaotic Mackey glass and Lorenz series is presented, which illustrates that our proposed algorithm achieves better performance in terms of MSE. The performance of the proposed algorithm is also tested and analyzed for the stationary and nonstationary time series, as well as with different noise levels in comparison with that of the LMS, KLMS and MFLMS. Hopefully, this new algorithm (i.e. AMFLMS) is another contribution in the field of nonlinear fractional signal processing.

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

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

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

[6] Ross B 1975 *International Conference at the University of New Heaven*, June 15-16, Berlin, Germany

[7] Machado J T, Kiryakova V and Mianardi F 2011 *Communication in Nonlinear Science and Numerical Simulation* **16** 1140

[8] Methews J V and Xie Z 1993 *IEEE Trans. Signal Processing* **41** 2075

[9] Tseng, Cheng C and Lee S L 2012 *Signal Processing* **92** 553

[10] Raja M A and Qureshi I M 2009 *Eur. J. Sci. Res.* **35** 14

[11] Akhtar P and Yasin M 2012 2*nd International Multi-topic Conference IMTIC*, March 28–30, Jamshoro, Pakistan

[12] Geravanchizadeh, Masood and Osgouei S G 2011 19*th Iranian Conference on Electrical Engineering*, May 17–19, Tehran, Iran

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

[14] Harris R W, Chabries D M and Bishop F A 1985 *IEEE Trans. Accoustic Speech and Signal Processing* **33** 222

[15] Kwong C P 1992 *IEEE Trans. Sign. Process.* **40** 2613

[16] Mikheal W B, *et al*. 1986 *IEEE Trans. Circ. Syst.* **33** 677

[17] Wei R, Lu J G, Li J and Wang Z Q 2002 *Acta Electron. Sin.* **30** 73

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

[19] Liu W, Pockarel P and Principe J C 2008 *IEEE Trans. Sign. Process.* **56** 543

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

[21] Pockarel P, Liu W and Principe J C 2009 *Sign. Process.* **89** 257

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

[23] Shoaib B, Qureshi I M, Ihsanulhaq and Shafqatullah 2014 *Chin. Phys. B* **23** 030502

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