Signal Processing 142 (2018) 441–449

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Momentum fractional LMS for power signal parameter estimation

Syed Zubair[a,*], Naveed Ishtiaq Chaudhary[a], Zeshan Aslam Khan[a], Wenwu Wang[b]

[a] Department of Electrical Engineering, International Islamic University, Islamabad, Pakistan
[b] Center for Vision, Speech and Signal Processing, University of Surrey, Guildford, United Kingdom

ARTICLE INFO

Article history:
Received 14 March 2017
Revised 1 August 2017
Accepted 6 August 2017
Available online 7 August 2017

Keywords:
Fractional algorithms
Parameter estimation
Signal processing
Power signal estimation

ABSTRACT

Fractional adaptive algorithms have given rise to new dimensions in parameter estimation of control and signal processing systems. In this paper, we present novel fractional calculus based LMS algorithm with fast convergence properties and potential ability to avoid being trapped into local minima. We test our proposed algorithm for parameter estimation of power signals and compare it with other state-of-the-art fractional and standard LMS algorithms under different noisy conditions. Our proposed algorithm outperforms other LMS algorithms in terms of convergence rate and accuracy.

1. Introduction

Integer order adaptive signal processing algorithms have their benefits for many signal processing, physical processes and control applications. One well-known algorithm based upon gradient descent is Least Mean Square (LMS) algorithm [1]. Many variations of the standard LMS have been proposed in the literature to improve its convergence properties and estimation accuracy [2]. All these algorithms are based on integer order gradients which find the trajectory of the solution to the optimum value in the negative direction of the gradient.

Recently, a graceful number of research activities have emerged in applying fractional order calculus for the design of adaptive algorithms. The fractional order adaptive algorithms have shown improved performance in various engineering applications compared to integer order LMS based algorithms [3,4]. In this paper, we design a new fractional LMS algorithm with improved convergence properties as compared to standard LMS and state-of-the-art fractional LMS algorithms.

1.1. Related work

Fractional order calculus has equally evolved in parallel with integer order calculus in the field of mathematics. Its application in the field of sciences and engineering was initiated in [5]. Since then, it has been applied in a variety of domains where integer order adaptive algorithms were applied ranging from signal processing [6,7], biomedical problems [8], control [9,10], to physical processes [11,12]. The newly evolved fractional adaptive algorithms borrow their ideas from LMS algorithm and its variants by introducing different ways for step-size calculation and weights updating mechanisms. For example, fractional least mean square (FLMS) identification algorithm was developed by exploiting the theories of fractional calculus for weights update in standard LMS [3].

The FLMS update equation includes integer order gradient as well as the fractional order gradient. The trade-off between these two gradients is suggested in [13] that adds a proportion of each gradient according to the value of a forgetting factor. This results in better convergence as compared to the original FLMS in [3]. The convergence properties of FLMS is further improved by introducing a sliding window which also includes previous values of the input in addition to the current input values [14]. To reduce the computational complexity, works in [15] include only fractional part of the gradient in the weight update equation. By omitting the integer order gradient and retaining only the fractional part, the overall convergence is not affected but the computational complexity caused by the integer order gradient is reduced. The fractional order used in the algorithms so far lies in the range $\in (0,1)$ and as fractional order approaches to 1, convergence rate increases. However, higher fractional order also increases the steady state error. This behavior of rapidity and accuracy was further studied in [16] for fractional order $\in (1,1.5)$. The authors found that the same behavior of rapidity and accuracy is observed as in the original FLMS [3]. Modified LMS [17] was extended to fractional version in [18].

* Corresponding author.
E-mail addresses: szubair@iiu.edu.pk, szgilani@gmail.com (S. Zubair), naveed.ishtiaq@iiu.edu.pk (N.I. Chaudhary), zeeshan.aslam@iiu.edu.pk (Z.A. Khan), w.wang@surrey.ac.uk (W. Wang).

http://dx.doi.org/10.1016/j.sigpro.2017.08.009

The above discussion shows that different variants of LMS have been extended to fractional order and their properties were studied. In this study, we extend momentum LMS (mLMS) [19] to fractional order and empirically study its convergence properties and estimation accuracy for sinusoidal signal modeling. The mLMS updates the weights by incorporating proportion of the previously calculated gradients in the current update step. This increases the convergence rate of the mLMS as compared to the standard LMS. By incorporating these previously calculated gradients in the current weights update step of the standard FLMS algorithm [3], we also intend to improve the convergence properties of the FLMS and name it as momentum FLMS (mFLMS).

Many parameter estimation techniques exist in the literature for different applications [20–23]. Recently some new methods have been introduced in [24–28]. To demonstrate promising properties of the proposed method, we consider the application of signal modeling and parameter estimation of sinusiodal signals which are important for reliability assessment and quality monitoring of power systems. Frequency, as one of the parameters, is important to be estimated for harmonic measurement and compensation [29] and in phase lock loops (PLL) for grid signal synchronization with system output [30]. The amplitude estimate is used in fault detection algorithms [31] and in under/over voltage protection algorithms [32]. The phase estimate is used in different scenarios such as PLL algorithms [33] and in the generation of control signals in a controller [34]. Recently, a novel stochastic gradient algorithm has been proposed for estimating the parameters of the sine combination signal modeling, and further a multi-innovation stochastic gradient parameter estimation method is presented by expanding the scalar innovation into the innovation vector for improving the estimation accuracy [35]. Here we apply our proposed algorithm on parameter estimation problem of power signals and compare it with LMS, mLMS and FLMS.

1.2. Our contribution

Inspired by different variations in LMS to improve its convergence and parameter estimation properties [36], we also incorporate an adaptation term in standard fractional LMS (FLMS) [3] and study its convergence properties and estimation accuracy. We design momemtum fractional LMS (mFLMS) in which a momentum term is incorporated with standard FLMS that increases the speed of the convergence and has the ability to avoid trapping in local minima. This work is different from [36] where momentum term is used with simple (non-fractional) LMS (mLMS). To show the performance of the proposed algorithm, the mFLMS is applied to estimate magnitude and phase of a sinusoidal signal [35] which is a combination of different sinusoidal harmonics having different amplitudes and phases. We compare its performance with fractional LMS (FLMS), momemtum LMS (mLMS) and LMS algorithms with varying learning rate parameters and under different noise conditions. This is also different from the works in [14,15] where no momentum term is used.

1.3. Paper outline

The paper is organized as follows: Section 2 gives brief description about FLMS. Section 3 describes the design of mFLMS and its derivation for power signal. Section 4 gives experimental details followed by results and discussion. Finally, the paper is concluded in Section 6.

2. Fractional order least mean squares (FLMS)

Application of fractional calculus to standard LMS algorithm have given rise to FLMS algorithm [3] where apart from taking simple integer order derivative, the fractional order derivative is also used to calculate fractional order gradients for the minimization of objective function. Let $y(n)$ be the estimated signal, $d(n)$ be the desired signal and $e(n)$ be the error signal, then the objective function for the minimization of the error is:

$$J(n) = E[e(n)^2] = E[d(n) - y(n)]^2 \tag{1}$$

where $E[\cdot]$ is the expectation. The estimated output $y(n)$ is written as:

$$y(n) = \hat{\mathbf{w}}^T(n)\,\mathbf{u}(n) \tag{2}$$

where $\hat{\mathbf{w}}$ is the estimated weight vector and $\mathbf{u}$ is the input vector. To find the weights, we need to minimize objective function (1) with respect to $\hat{\mathbf{w}}$, given as:

$$\frac{\partial J(n)}{\partial \hat{\mathbf{w}}} = 2e(n)\frac{\partial e(n)}{\partial \hat{\mathbf{w}}} \tag{3}$$

Substituting $e(n)$ in above equation:

$$\frac{\partial J(n)}{\partial \hat{\mathbf{w}}} = 2e(n)\frac{\partial}{\partial \hat{\mathbf{w}}}\left(d(n) - \hat{\mathbf{w}}^T(n)\mathbf{u}(n)\right) \tag{4}$$

After simplifying (4):

$$\frac{\partial J(n)}{\partial \hat{\mathbf{w}}} = -2e(n)\mathbf{u}(n) \tag{5}$$

From (5), standard LMS update equation [1] is given by

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) - \frac{\mu_1}{2}\left(\frac{\partial J(n)}{\partial \hat{\mathbf{w}}}\right) \tag{6}$$

where μ_1 represents the step size parameter for standard LMS.

In Eq. (6), the first order gradient is used to update LMS weights. In case of the fractional LMS, in addition to first order gradient, fractional order gradient is also used. The recursive weight update relation for the fractional LMS algorithm is written as:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) - \frac{\mu_1}{2}\left(\frac{\partial J(n)}{\partial \hat{\mathbf{w}}}\right) + \mu_f\left(\frac{\partial^f}{\partial \hat{\mathbf{w}}^f}J(n)\right) \tag{7}$$

where μ_f is the step size for the fraction order derivative ∂^f.

Following the Caputo and Riemann-Liouville definition [37], the fractional derivative of a function $g(t) = t^n$ is defined as:

$$D^f g(t) = \frac{\Gamma(n+1)}{\Gamma(n-f+1)}t^{n-f} \tag{8}$$

where D^f is fractional derivative operator of order f and Γ is a gamma function, defined as:

$$\Gamma(n) = (n-1)! \tag{9}$$

By using the above definitions for fractional order derivatives, the fractional order derivative in (7) becomes:

$$\frac{\partial^f}{\partial \hat{\mathbf{w}}^f}J(n) = -2(e(n)\mathbf{u}(n))\left(\frac{\partial^f}{\partial \hat{\mathbf{w}}^f}\hat{\mathbf{w}}(n)\right) \tag{10}$$

By using (8), (10) becomes:

$$\frac{\partial^f}{\partial \hat{\mathbf{w}}^f}J(n) = -2(e(n)\mathbf{u}(n))\left(\frac{\Gamma(2)}{\Gamma(2-f)}\hat{\mathbf{w}}^{1-f}(n)\right) \tag{11}$$

As $\Gamma(2) = 1$, substituting (11) in (7), we have:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu_1 e(n)\mathbf{u}(n) + \frac{\mu_f}{\Gamma(2-f)}e(n)\mathbf{u}(n) \odot |\hat{\mathbf{w}}|^{1-f}(n) \tag{12}$$

where the symbol $\odot$ denotes an element by element multiplication of vectors and the absolute value of vector $\hat{\mathbf{w}}$ is used to avoid complex values.

Eq. (12) is the weight update equation of the standard FLMS algorithm [3].

Table 1
Computational complexities of different LMS algorithms.

Method	Operations			*Big O*
	Multiplication	Addition	Power	
LMS	2M+1	2M	0	M
mLMS	3M+1	3M	0	M
FLMS	4M+2	3M	M	M
mFLMS	5M+2	4M	M	M

3. Momentum fractional LMS (mFLMS)

Using the concepts of momentum term for gradient calculation as has been used for standard LMS [38], we extend it for the FLMS. The momentum term takes care of the proportion of previous gradients and add it to the current weights. This helps in speeding up the optimum search and avoids trapping in local minima. For new momentum FLMS (mFLMS), the weight update equation is:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{v}(n+1) \tag{13}$$

where $\mathbf{v}(n+1)$ is the velocity term which contains previously calculated gradients:

$$\mathbf{v}(n+1) = \alpha\mathbf{v}(n) + \mathbf{g}(n) \tag{14}$$

while from (14), $\mathbf{g}(n)$ is the gradient part of the equation given by:

$$\mathbf{g}(n) = \mu_1 e(n)\mathbf{u}(n) + \frac{\mu_f}{\Gamma(2-f)} e(n)\mathbf{u}(n) \odot |\hat{\mathbf{w}}|^{1-f}(n) \tag{15}$$

$\alpha \in (0,\ 1)$ controls the proportion of previous gradients that is added in current update of Eq. (13). We use this equation for power signal modeling in the next section.

3.1. Computational complexity of mFLMS

The complexity of the algorithms i.e., LMS, mLMS, FLMS and mFLMS is evaluated in terms of the number of operations required for the adaptation process and the results are presented in Table 1. If M denotes the number of unknown weight parameters, then LMS algorithm requires $2M+1$ multiplications and $2M$ additions in one iteration of the weight adaptation mechanism while mLMS algorithm takes $3M+1$ multiplications and $3M$ additions. The mLMS algorithm requires M more multiplications and additions than LMS. The FLMS method requires $4M+2$ multiplications, $3M$ additions and M powers while the mFLMS scheme requires $5M+2$ multiplications, $4M$ additions and M powers. The mFLMS algorithm requires M more multiplications and additions than FLMS. In comparison to the LMS, the multiplications required by FLMS and mFLMS are, respectively, 2 and 2.5 times more than that required by the LMS. As an asymptotic complexity bound, the order of all the algorithms is $O(M)$ (i.e. in terms of operations) and the fractional or momentum parts in the update equation do not add any considerable complexity to the proposed algorithm as compared to the other LMS variants.

3.2. Convergence analysis of mFLMS

Considering Eqs. (13)–(15), assuming $\mu_f = \mu_1\Gamma(2-f)$ for simplicity in (15) and then rearranging, (13) can be written as:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \alpha[\hat{\mathbf{w}}(n) - \hat{\mathbf{w}}(n-1)] + \mu_1 e(n)\mathbf{u}(n) \odot |\hat{\mathbf{w}}|^{1-f}(n) \tag{16}$$

Defining $\Delta\mathbf{w}(n) = \mathbf{w}(n) - \mathbf{w}_{opt}$, where $\mathbf{w}_{opt}$ is the optimum value of the weight vector. The weight error update equation is written as:

$$\begin{aligned}\Delta\hat{\mathbf{w}}(n+1) = {} & \Delta\hat{\mathbf{w}}(n) + \alpha\big[\Delta\hat{\mathbf{w}}(n) - \Delta\hat{\mathbf{w}}(n-1)\big] \\ & + \mu_1\mathbf{u}(n)\big(d(n) - \mathbf{u}^T(n)\{\mathbf{w}_{opt} + \Delta\hat{\mathbf{w}}(n)\}\big) \\ & \times \Big(1 + \{\mathbf{w}_{opt} + \Delta\hat{\mathbf{w}}(n)\}^{1-f}\Big)\end{aligned} \tag{17}$$

Expanding

$$\begin{aligned}\Delta\hat{\mathbf{w}}(n+1) = {} & \Delta\hat{\mathbf{w}}(n) + \alpha\big[\Delta\hat{\mathbf{w}}(n) - \Delta\hat{\mathbf{w}}(n-1)\big] \\ & + \mu_1\mathbf{u}(n)d(n) + \mu_1\mathbf{u}^T(n)d(n)\{\mathbf{w}_{opt} + \Delta\hat{\mathbf{w}}(n)\}^{1-f} \\ & - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}_{opt} - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\Delta\hat{\mathbf{w}}(n) \\ & + \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\{\mathbf{w}_{opt} + \Delta\hat{\mathbf{w}}(n)\}^{2-f}\end{aligned} \tag{18}$$

Defining

$$\{\mathbf{w}_{opt} + \Delta\hat{\mathbf{w}}(n)\}^j = \sum_{k=0}^{j}\binom{j}{k}(\mathbf{w}_{opt}^k)^T\Delta\hat{\mathbf{w}}(n)^{j-k} \tag{19}$$

using (19) in (18),

$$\begin{aligned}\Delta\hat{\mathbf{w}}(n+1) = {} & \Delta\hat{\mathbf{w}}(n) + \alpha\big[\Delta\hat{\mathbf{w}}(n) - \Delta\hat{\mathbf{w}}(n-1)\big] + \mu_1\mathbf{u}(n)d(n) \\ & - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}_{opt} - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\Delta\hat{\mathbf{w}}(n) \\ & + \mu_1\mathbf{u}^T(n)d(n)\Delta\hat{\mathbf{w}}(n)^{1-fr} + \mu_1\mathbf{u}^T(n)d(n) \\ & \times \sum_{k=1}^{1-f}\binom{1-f}{k}(\mathbf{w}_{opt}^k)^T\Delta\hat{\mathbf{w}}(n)^{1-f-k} \\ & - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\Delta\hat{\mathbf{w}}(t)^{2-f} - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\binom{2-f}{1} \\ & \times (\mathbf{w}_{opt}^1)^T\Delta\hat{\mathbf{w}}(n)^{1-f} - \mu_1\mathbf{u}(n)\mathbf{u}^T(n)\sum_{k=1}^{1-f}\binom{1-f}{k} \\ & \times (\mathbf{w}_{opt}^{k+1})^T\Delta\hat{\mathbf{w}}(n)^{1-f-k}\end{aligned} \tag{20}$$

Assuming the input and output are uncorrelated, the weights are statistically independent of input and output. Applying expectation on both sides of (20) and equating $E\Delta\hat{\mathbf{w}}(\cdot) = \mathbf{H}(\cdot)$

$$\begin{aligned}\mathbf{H}(n+1) = {} & \mathbf{H}(n) + \alpha[\mathbf{H}(n) - \mathbf{H}(n-1)] + \mu_1\mathbf{p} - \mu_1\mathbf{R}\mathbf{w}_{opt} \\ & - \mu_1\mathbf{R}\mathbf{H}(n) + \mu_1\mathbf{p}E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\} + \mu_1\mathbf{p}\sum_{k=1}^{1-f}\binom{1-f}{k} \\ & \times \mathbf{w}_{opt}^k E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f-k}\big\} - \mu_1\mathbf{R}E\big\{\Delta\hat{\mathbf{w}}(n)^{2-f}\big\} \\ & - \mu_1\mathbf{R}\binom{2-f}{1}\mathbf{w}_{opt}^1 E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\} - \mu_1\mathbf{R}\sum_{k=1}^{1-f}\binom{1-f}{k} \\ & \times \mathbf{w}_{opt}^{k+1}E\big\{\Delta\hat{\mathbf{w}}(t)^{1-f-k}\big\}.\end{aligned} \tag{21}$$

where $\mathbf{p}$ is the cross-correlation between input and the desired value and $\mathbf{R}$ is the input auto-correlation. As we know that, for optimal values of weight; $\mathbf{p} - \mathbf{R}\mathbf{w}_{opt} = 0$, (21) becomes

$$\begin{aligned}\mathbf{H}(n+1) = {} & \mathbf{H}(n) + \alpha[\mathbf{H}(n) - \mathbf{H}(n-1)] - \mu_1\mathbf{R}\mathbf{H}(n) \\ & + \mu_1\mathbf{p}E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\} - \mu_1\mathbf{R}E\big\{\Delta\hat{\mathbf{w}}(n)^{2-f}\big\} \\ & - \mu_1\mathbf{R}\binom{2-f}{1}\mathbf{w}_{opt}^1 E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\}.\end{aligned} \tag{22}$$

Let

$$\begin{aligned}& \mu_1\mathbf{p}E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\} - \mu_1\mathbf{R}E\big\{\Delta\hat{\mathbf{w}}(n)^{2-f}\big\} \\ & -\mu_1\mathbf{R}\binom{2-f}{1}\mathbf{w}_{opt}^1 E\big\{\Delta\hat{\mathbf{w}}(n)^{1-f}\big\} \\ & = \mu_1 E\{\Delta\hat{\mathbf{w}}(n)\}G\big(\Delta\hat{\mathbf{w}}(n), f\big)\end{aligned} \tag{23}$$

using (23), (22) becomes

$$\begin{aligned}\mathbf{H}(n+1) &= \mathbf{H}(n) + \alpha[\mathbf{H}(n) - \mathbf{H}(n-1)] - \mu_1 \mathbf{R}\mathbf{H}(n) \\ &\quad + \mu_1 E\{\Delta\hat{\mathbf{w}}(n)\}\mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\mathbf{H}(n+1) \\ &= \mathbf{H}(n) + \alpha\mathbf{H}(n) - \alpha\mathbf{H}(n-1) - \mu_1\mathbf{R}\mathbf{H}(n) \\ &\quad + \mu_1\mathbf{H}(n)\mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\end{aligned} \tag{24}$$

Simplifying

$$\mathbf{H}(n+1) = \mathbf{H}(n)\big\{1 + \alpha - \mu_1\big(\mathbf{R} - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big)\big\} - \alpha\mathbf{H}(n-1) \tag{25}$$

Defining a $2M$ dimensional state vector as,

$$\mathbf{M}(n+1) = \begin{bmatrix}\mathbf{H}(n+1)\\ \mathbf{H}(n)\end{bmatrix} \tag{26}$$

we have following recursion

$$\mathbf{M}(n+1) = \begin{bmatrix}(1+\alpha)\mathbf{I} - \mu_1\big(\mathbf{R} - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big) & -\alpha\mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{M}(n) \tag{27}$$

The stability of the above expression is governed by the roots r of the determinant:

$$\det\begin{bmatrix}(1+\alpha-r)\mathbf{I} - \mu_1\big(\mathbf{R} - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big) & -\alpha\mathbf{I}\\ \mathbf{I} & -r\mathbf{I}\end{bmatrix} = 0 \tag{28}$$

for which the necessary and sufficient condition is $|r_i| < 1,\ i = 1, 2, \ldots, 2M$.

Using the following result for block matrices **A, B, C**, and **D**:

$$\det\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix} = \det[\mathbf{D}]\det\big[\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\big] = \det[\mathbf{A}]\det\big[\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}\big] \tag{29}$$

Assuming that $\mathbf{D}^{-1}$ exists, we arrive at the following characteristics equation. The stability of the method is governed by the roots r of the determinant

$$(-r)^n \det[\mathbf{D}]\det\Big[(1+\alpha-r)\mathbf{I} - \mu_1\big(\mathbf{R} - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big) - \frac{\alpha}{r}\mathbf{I}\Big] = 0 \tag{30}$$

To determine the $2M$ roots, we need to investigate the typical quadratic form

$$r_i^2 - r_i\big(1 + \alpha - \mu_1\big(\lambda_i - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big)\big) + \alpha = 0 \tag{31}$$

Applying the Routh-Hurwitz method to determine the step size bound

$$\big(1 + \alpha - \mu_1\big(\lambda_i - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big)\big) > 0 \tag{32}$$

simplifying

$$0 < \mu_1 < \frac{1+\alpha}{\big(\lambda_i - \mathbf{G}\big(\Delta\hat{\mathbf{w}}(n), f\big)\big)} \tag{33}$$

Eq. (33) gives step size bound for $0 < \alpha < 1$.

3.3. mFLMS for power signal modeling

Parameter estimation of the sinusoidal signals is very important as it helps in assessing the reliability of power systems. Here we consider a sampled multi-harmonic sinusoidal signal with different amplitudes and phase:

$$y(n) = \sum_{k=1}^{N} a_k \sin(n\omega_k + \phi_k) + \epsilon(n) \tag{34}$$

where $\epsilon(n)$ is the Gaussian noise with zero mean and constant variance σ^2.

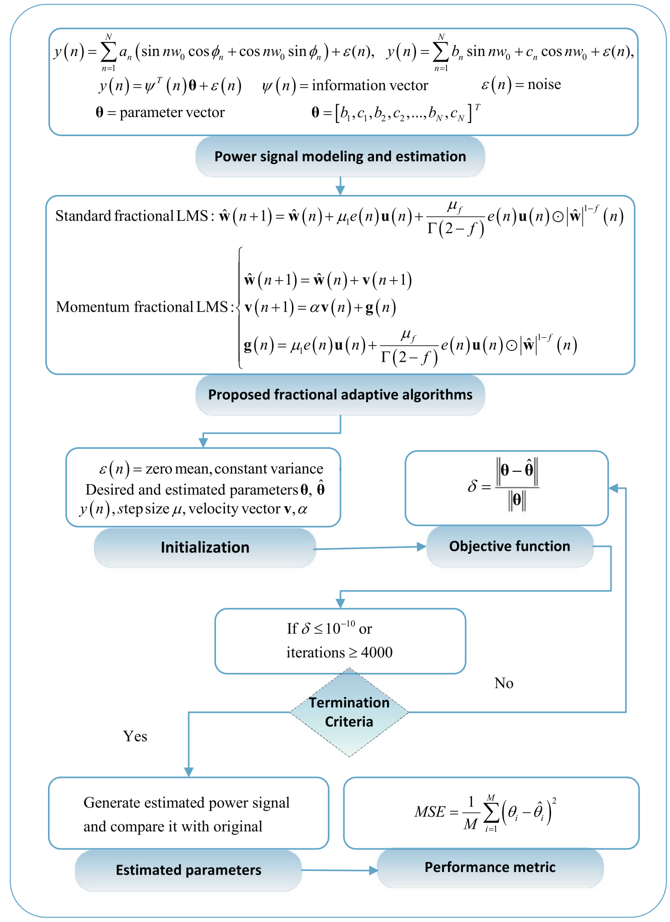

Fig. 1. Graphical Abstract of the whole system design.

Using trigonometric identity, (34) can be represented as:

$$y(n) = \sum_{k=1}^{N} a_k(\sin n\omega_k \cos\phi_k + \cos n\omega_k \sin\phi_k) + \epsilon(n) \tag{35}$$

We assume that the frequency of the signal is known, then (35) can be written as:

$$y(n) = \sum_{k=1}^{N} b_k \sin n\omega_k + c_k \cos n\omega_k + \epsilon(n) \tag{36}$$

where $b_k = a_k\cos\phi_k$ and $c_k = a_k\sin\phi_k$.

We will estimate parameters b_k and c_k since these can give us a_k and ϕ_k using relations:

$$a_k = \sqrt{b_k^2 + c_k^2} \qquad \phi_k = \tan^{-1}\frac{c_k}{b_k} \tag{37}$$

Defining the parameter vector $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta} = [b_1, c_1, b_2, c_2, \ldots, b_N, c_N]^T \in \Re^{2N} \tag{38}$$

and the corresponding information vector as:

$$\begin{aligned}\boldsymbol{\psi}(n) &= [\sin\omega_1 n, \cos\omega_1 n, \sin\omega_2 n,\\ &\quad \cos\omega_2 n, \ldots, \sin\omega_N n, \cos\omega_N n]^T \in \Re^{2N}\end{aligned} \tag{39}$$

Using Eqs. (38) and (39) in (36), the identification model for a power signal is given as:

$$y(n) = \boldsymbol{\psi}^T(n)\boldsymbol{\theta} + \epsilon(n) \tag{40}$$

Eq. (40) represents the identification model to estimate the parameters of power signal. While applying the mFLMS for parameter estimation, the parameter vector $\boldsymbol{\theta}$ is treated as weight vector in the

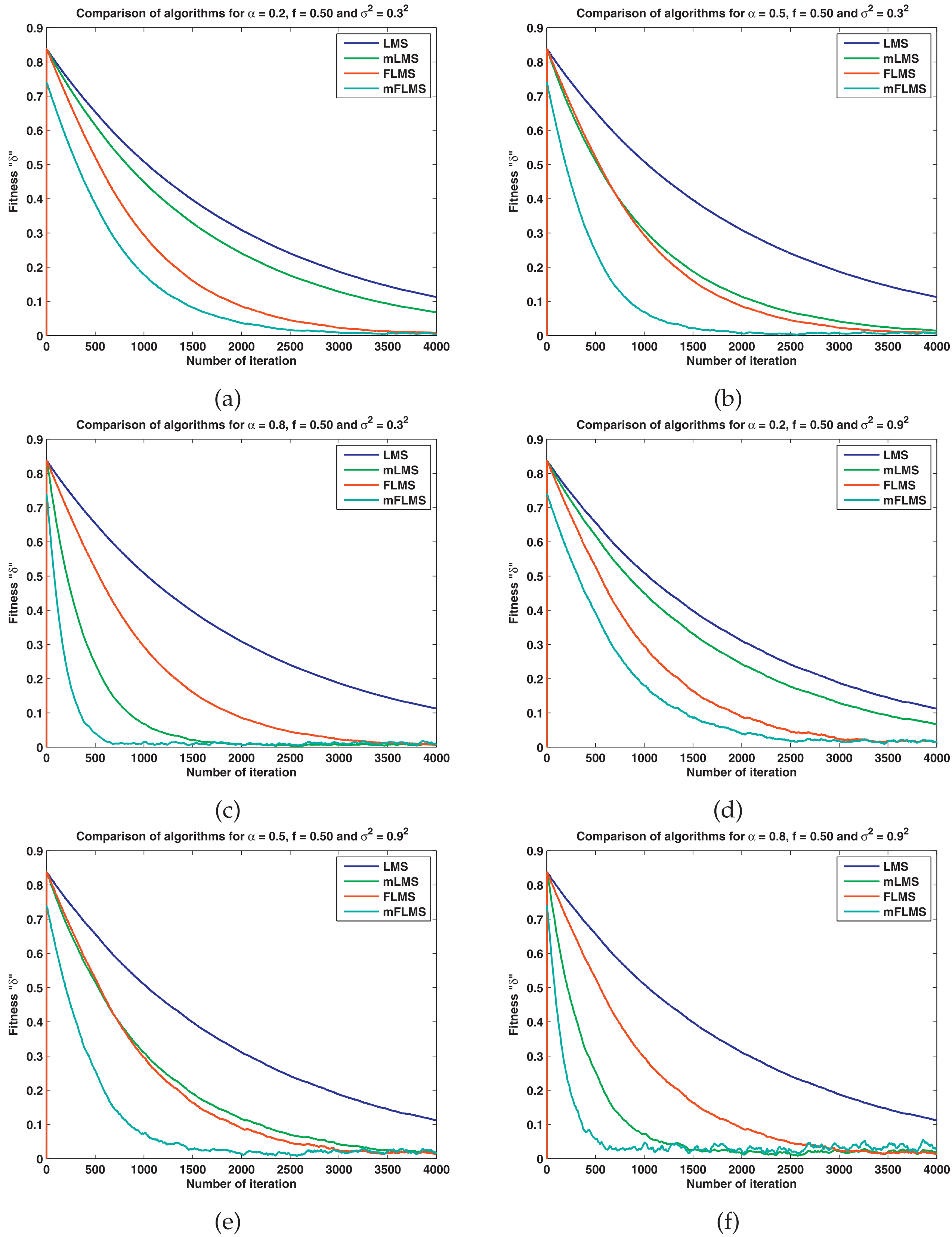

Fig. 2. Fitness curves of different types of LMS algorithms for $\mu = 0.001$ and $f = 0.5$ (where applicable). (a)-(c) Fitness curves of LMS, mLMS, FLMS and mFLMS for $\alpha = 0.2, 0.5, 0.8$ respectively and $\sigma^2 = 0.3^2$. (d)–(f) Fitness curves of LMS, mLMS, FLMS and mFLMS for $\alpha = 0.2, 0.5, 0.8$ respectively and $\sigma^2 = 0.9^2$.

Table 2
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.30^2$.

Method	α	Fitness achieved at specific iterations									
		100	200	300	400	500	600	700	800	900	1000
LMS		0.7978	0.7598	0.7226	0.6867	0.6546	0.6209	0.5910	0.5624	0.5344	0.5088
mLMS	**0.2**	0.7881	0.7415	0.6964	0.6534	0.6154	0.5761	0.5416	0.5091	0.4775	0.44900
mLMS	**0.5**	0.7600	0.6891	0.6238	0.5631	0.5116	0.4604	0.4163	0.3774	0.3403	0.3084
mLMS	**0.8**	0.6602	0.5137	0.4021	0.3104	0.2443	0.1864	0.1424	0.1113	0.0860	0.0677
FLMS (f = 0.25)		0.7774	0.7165	0.6543	0.5933	0.5385	0.4809	0.4308	0.3846	0.3408	0.3028
mFLMS (f = 0.25	**0.2**	0.6646	0.5902	0.5178	0.4504	0.3919	0.3331	0.2842	0.2431	0.2061	0.1752
mFLMS (f=0.25)	**0.5**	0.6191	0.5049	0.4027	0.3149	0.2475	0.1861	0.1402	0.1085	0.0830	0.0646
mFLMS (f = 0.25)	**0.8**	0.455	0.2444	0.1281	0.0645	0.0372	0.0185	0.0107	0.0102	0.0126	0.016
FLMS (f = 0.50)		0.7715	0.7059	0.6406	0.5778	0.5225	0.4654	0.4166	0.372	0.3301	0.294
mFLMS (f = 0.50)	**0.2**	0.6595	0.5824	0.5094	0.4429	0.386	0.3295	0.2829	0.2437	0.2083	0.1786
mFLMS (f = 0.50)	**0.5**	0.6119	0.4963	0.3963	0.3121	0.2478	0.1891	0.1448	0.1134	0.0875	0.0686
mFLMS (f = 0.50)	**0.8**	0.4459	0.2439	0.1325	0.0684	0.0403	0.0188	0.0097	0.0093	0.0118	0.0152
FLMS (f = 0.75)		0.7653	0.696	0.6291	0.5665	0.5124	0.4573	0.4108	0.3687	0.3293	0.2954
mFLMS (f = 0.75)	**0.2**	0.6559	0.5781	0.5065	0.4424	0.3882	0.3346	0.2905	0.2532	0.2193	0.1906
mFLMS (f = 0.75)	**0.5**	0.6075	0.4936	0.3978	0.318	0.2569	0.2008	0.1578	0.1264	0.0998	0.0797
mFLMS (f = 0.75)	**0.8**	0.4438	0.2525	0.1451	0.0795	0.0488	0.0235	0.0112	0.0092	0.0109	0.0142

mFLMS algorithm and is updated using Eq. (13). The overall graphical representation of the whole work is shown in Fig. 1.

4. Experimental setup

We investigate the behavior of mFLMS, FLMS, mLMS and LMS on the following signal which is a combination of different sinusoids in terms of frequency, phase and amplitudes. We assume that the sinusoids are of known frequencies and estimate amplitude and phase of individual sinusoidal signals for the above four algorithms.

Consider the following combination of sine signals with four different frequencies [35]:

$$\begin{aligned} y(n) = \ & 1.8\sin(0.07n+0.95) + 2.9\sin(0.5n+0.8) \\ & + 4\sin(2n+0.76) + 2.5\sin(1.6n+1.1) + \epsilon(n) \end{aligned}$$

The parameter vector (amplitude and phase) of the power signal is:

$$\begin{aligned} \boldsymbol{\theta} &= [a_1, a_2, a_3, a_4, \phi_1, \phi_2, \phi_3, \phi_4]^T \\ &= [1.8, 2.9, 4, 2.5, 0.95, 0.8, 0.76, 1.1]^T \end{aligned}$$

In the present study, ϵ is a noise with zero mean and a constant variance σ^2. The mFLMS and mLMS algorithms are investigated for three different values of α (proportion of previous gradients), i.e. [0.2, 0.5, 0.8], as only these two algorithms have momentum terms. The step size for all the algorithms is selected empirically after performing a set of trials to achieve the best MSE value after the convergence, i.e., 10^{-3}. In case of fractional order algorithms, the values for both step size parameters are same, i.e. $\mu_1 = \mu_f = \mu$. For higher values of μ_1 or μ_f, algorithms either did not converge smoothly or had higher value of MSE. The forgetting factor α was also chosen on the similar basis. To perform detail analysis, the proposed algorithm is studied for three different values of α (proportion of previous gradients), i.e., $= 0.2, 0.5$ and 0.8. It is observed that the convergence speed of the mFLMS method increases by increasing the value of α but at the cost of steady state performance i.e., the higher value of α provides faster convergence while lower value gives better steady state performance. The fractional order based methods are studied for three different fractional orders i.e., $f = 0.25$, 0.50 and 0.75. All the algorithms are examined for three values of Gaussian noise variance i.e., $\sigma^2 = 0.30^2, 0.60^2$ and 0.90^2. The adaptation process is performed for 4000 iterations.

4.1. Evaluation metrics

The performance of the design methodology is examined using two evaluation metrics i.e. fitness function and mean square error (MSE). The fitness function δ developed for power signal estimation is given as:

$$\delta = \frac{||\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}||}{||\boldsymbol{\theta}||} \tag{41}$$

where $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are the desired and approximated parameter vectors. The actual parameters of the power signal are adapted through proposed methods such that when $\delta \to 0$ consequently $\hat{\boldsymbol{\theta}} \to \boldsymbol{\theta}$. The performance measure based on MSE is given as:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\theta_i - \hat{\theta})^2 \tag{42}$$

where M is the number of estimated parameters.

5. Results and discussion

The learning efficiency of the algorithms for different values of parameters is shown in Fig. 2. All those sub-figures of Fig. 2 show plots of fitness function against the number of iterations for fractional order $f = 0.5$. Detailed results for other values of f are given in the tables. The comparison of the proposed mFLMS algorithm with standard LMS, mLMS and FLMS methods for $\alpha = 0.2, 0.5$ and 0.8 are given in Fig. 2 a, b and c respectively for noise variance $\sigma^2 = 0.30^2$, while the respective plots in case of $\sigma^2 = 0.90^2$ are presented in Fig. 2 d, e and f. It is observed that convergence of mLMS and mFLMS methods is faster than their counterparts, i.e., standard LMS and FLMS algorithms, and by increasing the value of α, the momentum versions provide faster convergence. It is also seen that the proposed mFLMS algorithm outperforms all other algorithms in terms of convergence for different variations in parameter values.

The performance of the proposed scheme is also evaluated for initial convergence rate and results of fitness adaptation for first 1000 iterations are presented in Tables 2–4 for $\sigma^2 = 0.30^2, 0.60^2$ and 0.30^2 respectively, for various α and fractional order values. It is seen from the results presented in Tables 2–4 that initial convergence of mFLMS is much faster than standard adaptive strategies and rate of convergence increases by increasing the proportion of previous gradients (α).

The performance of the proposed mFLMS algorithm is further verified through mean square error (MSE) metric. The results

Table 3
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.60^2$.

Method	α	Fitness achieved at specific iterations									
		100	200	300	400	500	600	700	800	900	1000
LMS		0.7979	0.76	0.7235	0.6875	0.656	0.6217	0.5913	0.5629	0.5347	0.5093
mLMS	**0.2**	0.7882	0.7418	0.6976	0.6544	0.6172	0.577	0.5418	0.5096	0.4778	0.4496
mLMS	**0.5**	0.7602	0.6895	0.6257	0.5646	0.5142	0.4617	0.4165	0.3778	0.3407	0.3091
mLMS	**0.8**	0.6604	0.5148	0.4064	0.3131	0.2492	0.1879	0.1415	0.1111	0.0861	0.0692
FLMS (f = 0.25)		0.7775	0.717	0.6561	0.5949	0.5412	0.4823	0.431	0.3852	0.3412	0.3037
mFLMS (f = 0.25)	**0.2**	0.6648	0.591	0.5201	0.4525	0.3954	0.3348	0.2842	0.2437	0.2066	0.1762
mFLMS (f = 0.25)	**0.5**	0.6192	0.5062	0.4064	0.3179	0.2527	0.188	0.1393	0.1088	0.0839	0.0674
mFLMS (f = 0.25)	**0.8**	0.4548	0.2474	0.1367	0.0698	0.0457	0.0247	0.0205	0.0184	0.0247	0.0318
FLMS (f = 0.50)		0.7717	0.7065	0.6424	0.5794	0.5253	0.4668	0.4168	0.3726	0.3305	0.2949
mFLMS (f = 0.50)	**0.2**	0.6596	0.5831	0.5116	0.4449	0.3895	0.3312	0.2829	0.2443	0.2087	0.1794
mFLMS (f = 0.50)	**0.5**	0.6119	0.4975	0.3999	0.315	0.2527	0.191	0.144	0.1137	0.0882	0.0707
mFLMS (f = 0.50)	**0.8**	0.4456	0.2469	0.1409	0.0734	0.0488	0.0238	0.0184	0.017	0.0232	0.0302
FLMS (f = 0.75)		0.7655	0.6965	0.631	0.568	0.5151	0.4587	0.411	0.3693	0.3297	0.2961
mFLMS (f = 0.75)	**0.2**	0.6559	0.5789	0.5087	0.4444	0.3914	0.3362	0.2905	0.2538	0.2197	0.1913
mFLMS (f = 0.75)	**0.5**	0.6074	0.4948	0.4012	0.3208	0.2615	0.2026	0.1573	0.1266	0.1002	0.0812
mFLMS (f = 0.75)	**0.8**	0.4434	0.2554	0.1529	0.084	0.0568	0.0269	0.0172	0.0157	0.0213	0.0282

Table 4
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.90^2$.

Method	α	Fitness achieved at specific iterations									
		100	200	300	400	500	600	700	800	900	1000
LMS		0.798	0.7603	0.7245	0.6884	0.6574	0.6225	0.5915	0.5633	0.5351	0.5098
mLMS	**0.2**	0.7883	0.7421	0.6988	0.6554	0.6189	0.578	0.5421	0.5101	0.4782	0.4502
mLMS	**0.5**	0.7603	0.69	0.6276	0.5661	0.5168	0.463	0.4166	0.3783	0.341	0.3098
mLMS	**0.8**	0.6606	0.5159	0.4108	0.3159	0.2542	0.1895	0.1407	0.111	0.0865	0.0717
FLMS (f = 0.25)		0.7777	0.7175	0.6578	0.5964	0.544	0.4837	0.4313	0.3858	0.3416	0.3046
mFLMS (f = 0.25)	**0.2**	0.6649	0.5918	0.5224	0.4545	0.399	0.3365	0.2843	0.2443	0.2071	0.1774
mFLMS (f = 0.25)	**0.5**	0.6193	0.5075	0.4101	0.321	0.2579	0.19	0.1387	0.1094	0.0854	0.0711
mFLMS (f = 0.25)	**0.8**	0.4548	0.251	0.1459	0.0756	0.0552	0.0323	0.0313	0.027	0.0368	0.0476
FLMS (f = 0.50)		0.7719	0.707	0.6443	0.5809	0.5281	0.4682	0.417	0.3732	0.3309	0.2958
mFLMS (f = 0.50)	**0.2**	0.6596	0.5839	0.5139	0.447	0.3929	0.3329	0.283	0.2449	0.2092	0.1805
mFLMS (f = 0.50)	**0.5**	0.612	0.4988	0.4036	0.318	0.2577	0.193	0.1435	0.1142	0.0892	0.0736
mFLMS (f = 0.50)	**0.8**	0.4454	0.2503	0.1498	0.0788	0.058	0.0305	0.0285	0.025	0.0347	0.0453
FLMS (f = 0.75)		0.7656	0.697	0.6329	0.5695	0.5179	0.4601	0.4111	0.3699	0.3301	0.297
mFLMS (f = 0.75)	**0.2**	0.6559	0.5796	0.5109	0.4464	0.3947	0.3379	0.2907	0.2544	0.2201	0.1922
mFLMS (f = 0.75)	**0.5**	0.6074	0.496	0.4048	0.3236	0.2662	0.2045	0.1569	0.127	0.1009	0.0833
mFLMS (f = 0.75)	**0.8**	0.4431	0.2585	0.1611	0.0889	0.0654	0.0319	0.0257	0.023	0.0318	0.0422

Table 5
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.30^2$.

Method	α	Adaptive parameters								MSE
		$\boldsymbol{\theta}_1$	$\boldsymbol{\theta}_2$	$\boldsymbol{\theta}_3$	$\boldsymbol{\theta}_4$	$\boldsymbol{\theta}_5$	$\boldsymbol{\theta}_6$	$\boldsymbol{\theta}_7$	$\boldsymbol{\theta}_8$	
LMS		1.6241	2.5608	3.5767	2.192	0.894	0.8152	0.7694	1.1025	5.29E-02
mLMS	**0.2**	1.689	2.6954	3.7477	2.3121	0.9171	0.8099	0.7643	1.1009	1.93E-02
mLMS	**0.5**	1.7662	2.8561	3.9541	2.4563	0.941	0.8037	0.7584	1.0989	8.97E-04
mLMS	**0.8**	1.7814	2.9025	4.0161	2.4991	0.938	0.7996	0.7563	1.0981	9.67E-05
FLMS (f = 0.25)		1.7684	2.8862	4.0094	2.4777	0.9416	0.8025	0.7565	1.1048	2.36E-04
mFLMS (f = 0.25)	**0.2**	1.7776	2.8991	4.0161	2.4944	0.9472	0.8009	0.7563	1.0996	1.02E-04
mFLMS (f = 0.25)	**0.5**	1.7794	2.9019	4.0181	2.4987	0.9394	0.7992	0.7567	1.0987	1.11E-04
mFLMS (f = 0.25)	**0.8**	1.7755	2.9046	4.0242	2.5054	0.9209	0.7985	0.7632	1.1003	2.62E-04
FLMS (f = 0.50)		1.7699	2.8819	4.0014	2.4759	0.9419	0.8028	0.7568	1.1028	2.39E-04
mFLMS (f = 0.50)	**0.2**	1.7788	2.8978	4.0134	2.4938	0.9467	0.8012	0.7563	1.0993	8.73E-05
mFLMS (f = 0.50)	**0.5**	1.7802	2.9018	4.0178	2.4983	0.9395	0.7995	0.7565	1.0986	1.05E-04
mFLMS (f = 0.50)	**0.8**	1.7767	2.905	4.0216	2.5043	0.921	0.7983	0.7613	1.1001	2.37E-04
FLMS (f = 0.75)		1.7693	2.873	3.9853	2.4696	0.9418	0.8032	0.7574	1.101	3.63E-04
mFLMS (f = 0.75)	**0.2**	1.7792	2.8948	4.0071	2.4918	0.9466	0.8017	0.7566	1.099	7.57E-05
mFLMS (f = 0.75)	**0.5**	1.7812	2.9016	4.0173	2.498	0.9401	0.7999	0.7563	1.0985	9.63E-05
mFLMS (f = 0.75)	**0.8**	1.7778	2.9051	4.0193	2.5032	0.922	0.7981	0.7596	1.0997	2.11E-04
True Values		1.8	2.9	4	2.5	0.95	0.8	0.76	1.1	0

obtained through mFLMS are compared with standard adaptive schemes and given in Tables 5–7 for $\sigma^2 = 0.30^2, 0.60^2$ and 0.30^2 respectively, for different variations in α and fractional order. It is observed that all algorithms are accurate and convergent but accuracy of the methods decreases by increasing noise variance. It is seen that fractional adaptive algorithms i.e., FLMS and mFLMS remain stable for all variations of fractional order and not much difference in accuracy is observed among different fractional orders. However, higher value of fractional order i.e., $f = 0.75$ gives relatively better results.

It can be observed from the results in Tables 1–6 that mFLMS algorithm provides faster convergence for higher proportion of pre-

Table 6
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.60^2$.

Method	α	Adaptive Parameters								MSE
		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	
LMS		1.6173	2.5622	3.583	2.1881	0.8936	0.8181	0.767	1.1016	5.28E-02
mLMS	**0.2**	1.6813	2.6969	3.7556	2.3085	0.9164	0.8127	0.7617	1.0998	1.91E-02
mLMS	**0.5**	1.7555	2.8577	3.9658	2.4534	0.9386	0.8059	0.7553	1.0974	9.13E-04
mLMS	**0.8**	1.7631	2.9051	4.0323	2.4984	0.9258	0.7993	0.7526	1.0961	3.87E-04
FLMS (f = 0.25)		1.7551	2.8878	4.0257	2.4742	0.9381	0.8038	0.7528	1.1034	4.64E-04
mFLMS (f = 0.25)	**0.2**	1.7619	2.9006	4.0334	2.4915	0.9421	0.8014	0.7525	1.0981	3.46E-04
mFLMS (f = 0.25)	**0.5**	1.7591	2.9036	4.036	2.4972	0.9286	0.7984	0.7534	1.0973	4.38E-04
mFLMS (f = 0.25)	**0.8**	1.7526	2.9086	4.0475	2.5104	0.8912	0.797	0.7664	1.1007	1.02E-03
FLMS (f = 0.50)		1.757	2.8835	4.0166	2.4725	0.9386	0.8043	0.7533	1.1012	4.18E-04
mFLMS (f = 0.50)	**0.2**	1.7638	2.8994	4.03	2.4909	0.9416	0.8021	0.7527	1.0977	3.04E-04
mFLMS (f = 0.50)	**0.5**	1.7607	2.9035	4.0356	2.4966	0.9289	0.7989	0.753	1.0971	4.18E-04
mFLMS (f = 0.50)	**0.8**	1.7548	2.9096	4.0426	2.5084	0.8914	0.7966	0.7626	1.1002	9.34E-04
FLMS (f = 0.75)		1.7572	2.8747	3.999	2.4662	0.9388	0.805	0.7541	1.0994	4.75E-04
mFLMS (f = 0.75)	**0.2**	1.7651	2.8965	4.0225	2.4889	0.9419	0.803	0.753	1.0974	2.49E-04
mFLMS (f = 0.75)	**0.5**	1.7628	2.9033	4.0347	2.496	0.93	0.7997	0.7527	1.0969	3.85E-04
mFLMS (f = 0.75)	**0.8**	1.757	2.9101	4.0384	2.5063	0.8933	0.7962	0.7592	1.0995	8.36E-04
True Values		1.8	2.9	4	2.5	0.95	0.8	0.76	1.1	0

Table 7
Performance comparison based on fitness achieved at specific iterations for $\sigma^2 = 0.90^2$.

Method	α	Adaptive Parameters								MSE
		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	
LMS		1.6106	2.5637	3.5894	2.1843	0.8933	0.821	0.7646	1.1007	5.26E-02
mLMS	**0.2**	1.6735	2.6984	3.7635	2.3049	0.9157	0.8155	0.7591	1.0987	1.90E-02
mLMS	**0.5**	1.7448	2.8593	3.9775	2.4505	0.9361	0.8081	0.7522	1.0959	9.99E-04
mLMS	**0.8**	1.745	2.9078	4.0486	2.4976	0.9133	0.7989	0.749	1.0942	8.70E-04
FLMS (f = 0.25)		1.7417	2.8892	4.0419	2.4706	0.9346	0.805	0.7492	1.1019	8.15E-04
mFLMS (f = 0.25)	**0.2**	1.7461	2.902	4.0506	2.4885	0.9368	0.802	0.7488	1.0965	7.39E-04
mFLMS (f = 0.25)	**0.5**	1.7389	2.9051	4.0537	2.4955	0.9176	0.7976	0.7502	1.0959	9.79E-04
mFLMS (f = 0.25)	**0.8**	1.7313	2.9121	4.07	2.5148	0.861	0.7956	0.7695	1.1012	2.25E-03
FLMS (f = 0.50)		1.7442	2.8851	4.0318	2.469	0.9352	0.8058	0.7497	1.0998	7.08E-04
mFLMS (f = 0.50)	**0.2**	1.7487	2.9009	4.0465	2.4879	0.9365	0.803	0.749	1.0961	6.59E-04
mFLMS (f = 0.50)	**0.5**	1.7413	2.9051	4.0534	2.4948	0.918	0.7984	0.7495	1.0957	9.37E-04
mFLMS (f = 0.50)	**0.8**	1.7346	2.914	4.0633	2.5121	0.8612	0.7949	0.7639	1.1004	2.07E-03
FLMS (f = 0.75)		1.7452	2.8763	4.0128	2.4629	0.9357	0.8067	0.7507	1.0979	6.81E-04
mFLMS (f = 0.75)	**0.2**	1.751	2.8981	4.0379	2.4859	0.9372	0.8042	0.7495	1.0957	5.44E-04
mFLMS (f = 0.75)	**0.5**	1.7446	2.905	4.0522	2.494	0.9198	0.7995	0.7491	1.0954	8.64E-04
mFLMS (f = 0.75)	**0.8**	1.7377	2.915	4.0573	2.5092	0.864	0.7943	0.7588	1.0993	2.11E-04
True Values		1.8	2.9	4	2.5	0.95	0.8	0.76	1.1	0

vious gradients i.e., $\alpha = 0.8$, while it gives better steady state performance for lower value of alpha i.e., $\alpha = 0.2$. Thus, the middle value i.e., $\alpha = 0.5$ seems to be an appropriate choice that is a good compromise between faster convergence and better steady state performance.

The curve fitting plot for the sinusoidal signal by mFLMS for $\alpha = 0.2$, $f = 0.5$ and $\sigma^2 = 0.60^2$, is given in Fig. 3. It can be noticed from the figure that the proposed mFLMS algorithm accurately follows the original power signal with high precision which validates the correctness and effectiveness of the proposed method.

6. Conclusion

In this work, a novel momentum fractional least mean square (mFLMS) has been proposed to increase the convergence of the standard fractional LMS algorithm by using previous proportions of the calculated gradients to update the weights. The correctness of the designed mFLMS method is established estimating the parameters of a power signal different noise variances, fractional orders and α (proportion of previous gradients) values. The proposed method outperforms all other adaptive algorithms for variations in the adaptive filter parameter values. The convergence speed of the mFLMS increases by increasing the value of α but at the cost of steady state performance i.e., the higher value of α provides faster

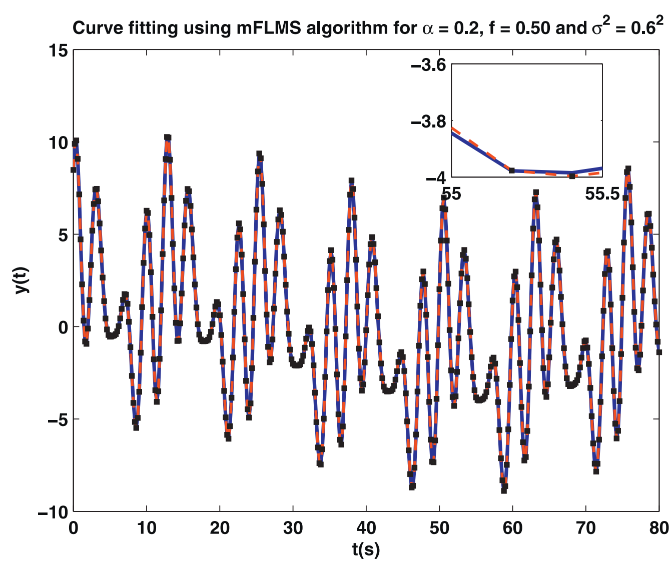

Fig. 3. Curve fitting using mFLMS for $\alpha = 0.2, \sigma^2 = 0.6^2, f = 0.5$. Here, the blue color represents the actual sinusoidal curve while the red color with data markers represents estimated curve. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article)

convergence while lower value gives better steady state performance. The performance of all algorithms decreases with an increase in noise variance but still the mFLMS achieves consistently better results in terms of accuracy and convergence. The proposed mFLMS algorithm remain convergent for all fractional orders and there is no noticeable difference in accuracy among different fractional orders. However, higher value of fractional order gives relatively better results.

7. Future work

In this work, we have devised momentum based fractional LMS for amplitude and phase estimation of a power signal. In addition to these two parameters, frequency is also an important parameter and we intend to extend our proposed algorithm for estimation of unknown frequency of a power signal in near future.

References

[1] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[2] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[3] R.M.A. Zahoor, I.M. Qureshi, A modified least mean square algorithm using fractional derivative and its application to system identification, Eur. J. Sci. Res. 35 (1) (2009) 14–21.
[4] Y. Chen, D. Xue, A. Visioli, Guest editorial for special issue on fractional order systems and controls, IEEE/CAA J. Automatica Sin. 3 (3) (2016) 255–256.
[5] M.D. Ortigueira, Introduction to fractional linear systems. part 1: continuous-time case, IEE Proc. - Vision, Image Signal Process. 147 (1) (2000) 62–70.
[6] C.-C. Tseng, S.-L. Lee, Designs of fractional derivative constrained 1-d and 2-d FIR filters in the complex domain, Signal Process. 95 (2014) 111–125.
[7] B. Kuldeep, V. Singh, A. Kumar, G. Singh, Design of two-channel filter bank using nature inspired optimization based fractional derivative constraints, ISA Trans. 54 (2015) 101–116.
[8] J.T. Machado, Fractional order description of DNA, Appl. Math. Model. 39 (14) (2015) 4095–4102.
[9] A.-C. Norelys, M.A. Duarte-Mermoud, Fractional adaptive control for an automatic voltage regulator, ISA Trans. 52 (6) (2013) 807–815.
[10] C.M. Ionescu, Y. Zhou, J.A.T. Machado, Advances in fractional dynamics and control, J. Vibration Control 22 (8) (2016). 1969–197.
[11] H. Zhangand, F. Liuand, M.S. Phanikumar, M.M. Meerschaert, A novel numerical method for the time variable fractional order mobile-immobile advection dispersion model, Comput. Math. Appl. 66 (5) (2013) 693–701.
[12] M.M. Meerschaert, R.L. Schilling, A. Sikorskii, Stochastic solutions for fractional wave equations, Nonlinear Dyn. 80 (4) (2015) 1685–1695.
[13] B. Shoaib, I.M. Qureshi, A modified fractional least mean square algorithm for chaotic and nonstationary time series prediction, Chin. Phys. B 23 (3) (2014).
[14] M.S. Aslam, N.I. Chaudhary, M.A. Raja, A sliding-window approximation-based fractional adaptive strategy for Hammerstein nonlinear armax systems, Nonlinear Dyn. 87 (1) (2017) 519–533.
[15] N.I. Chaudhary, M.A.Z. Raja, A.U.R. Khan, Design of modified fractional adaptive strategies for Hammerstein nonlinear control autoregressive systems, Nonlinear Dyn. 82 (4) (2015) 1811–1832.
[16] S. Cheng, Y. Wei, Y. Li, Y. Wang, An innovative fractional order LMS based on variable initial value and gradient order, Signal Process. 133 (2017) 260–269.
[17] F.F. Kretschmer, B.L. Lewis, An improved algorithm for adaptive processing, IEEE Trans. Aerosp. Electron. Syst. AES-14 (1) (1978) 172–177.
[18] Y. Tan, Z. He, B. Tian, A novel generalization of modified LMS algorithm to fractional order, IEEE Signal Process. Lett. 22 (9) (2015) 1244–1248.
[19] R. Sharma, W.A. Sethares, J.A. Bucklew, Analysis of momentum adaptive filtering algorithms, IEEE Trans. Signal Process. 46 (1998) 1430–1434.
[20] L. Xu, F. Ding, Parameter estimation algorithms for dynamical response signals based on the multi-innovation theory and the hierarchical principle, IET Signal Process. 11 (2) (2017) 228–237.
[21] F. Ding, F. Wang, L. Xu, T. Hayat, A. Alsaedi, Parameter estimation for pseudo-linear systems using the auxiliary model and the decomposition technique, IET Control Theory Appl. 11 (3) (2017) 390–400.
[22] F. Ding, F. Wang, L. Xu, M. Wu, Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear arma systems using the data filtering, J. Franklin Inst. 354 (3) (2017) 1321–1339.
[23] L. Xu, The damping iterative parameter identification method for dynamical systems based on the sine signal measurement, Signal Process. 120 (2016) 660–667.
[24] L. Xu, F. Ding, Y. Gu, A. Alsaedi, T. Hayat, A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay, Signal Process. 140 (2017) 97–103.
[25] X. Wang, F. Ding, Recursive parameter and state estimation for an input nonlinear state space system using the hierarchical identification principle, Signal Process. 117 (2015) 208–218.
[26] Y. Wang, F. Ding, The auxiliary model based hierarchical gradient algorithms and convergence analysis using the filtering technique, Signal Process. 128 (2016) 212–221.
[27] F. Ding, State filtering and parameter estimation for state space systems with scarce measurements, Signal Process. 104 (2014) 369–380.
[28] F. Ding, X. Liu, H. Chen, G. Yao, Hierarchical gradient based and hierarchical least squares based iterative parameter identification for CARARMA systems, Signal Process. 97 (2014) 31–39.
[29] E. Owen, A history of harmonics in power systems, IEEE Ind. Appl. Mag. 4 (1) (1998) 6–12.
[30] S. Vazquez, J.A. Sanchez, M.R. Reyes, J.I. Leon, J.M. Carrasco, Adaptive vectorial filter for grid synchronization of power converters under unbalanced and/or distorted grid conditions, IEEE Trans. Ind. Electron. 61 (3) (2014) 1355–1367.
[31] M. Shahbazi, P. Poure, S. Saadate, M.R. Zolghadri, FPGA based fast detection with reduced sensor count for a fault-tolerant three-phase converter, IEEE Trans. Ind. Inform. 9 (2013) 1343–1350.
[32] M.N. Kabir, Y. Mishra, G. Ledwich, Z.Y. Dong, K.P. Wong, Coordinated control of grid-connected photovoltaic reactive power and battery energy storage systems to improve the voltage profile of a residential distribution feeder, IEEE Trans. Ind. Inform. 10 (2) (2014) 967–977.
[33] L. Feola, R. Langella, A. Testa, On the effects of unbalances, harmonics and interharmonics on PLL systems, IEEE Trans. Instrum. Meas. 62 (9) (2013) 2399–2409.
[34] Y. Yang, K. Zhou, M. Cheng, Phase compensation resonant controller for PWM converters, IEEE Trans. Ind. Inform. 9 (2) (2013) 957–964.
[35] L. Xu, F. Ding, Recursive least squares and multi-innovation stochastic gradient parameter estimation methods for signal modeling, Circuit, Syst. Signal Process. 36 (4) (2016) 1735–1753.
[36] N.I. Chaudhary, S. Zubair, M.A.Z. Raja, Design of momentum LMS adaptive strategy for parameter estimation of Hammerstein controlled autoregressive systems, Neural Comput. Appl. (2016) 1–11, doi:10.1007/s00521-016-2762-1.
[37] M.D. Ortigueira, Fractional Calculus for Scientists and Engineers, Dordrecht: Springer, 2011.
[38] J.J. Shynk, S. Roy, The LMS algorithm with momentum updating, in: IEEE International Symposium on Circuits and Systems, 3, 1988, pp. 2651–2654.