RESEARCH   **Open Access**

Family of state space least mean power of two-based algorithms

Muhammad Moinuddin[1,2]*, Ubaid M Al-Saggaf[1,2] and Arif Ahmed[1,2]

Abstract

In this work, a novel family of state space adaptive algorithms is introduced. The proposed family of algorithms is derived based on stochastic gradient approach with a generalized least mean cost function $\mathbf{J}[k] = \mathbf{E}\left[\|\boldsymbol{\varepsilon}[k]\|^{2L}\right]$ for any integer L. Since this generalized cost function is having power '$2L$', it includes the whole family of the power of two-based algorithms by having different values of L. The novelty of the work resides in the fact that such a cost function has never been used in the framework of state space model. It is a well-known fact that the knowledge of state space model improves the estimation of state parameters of that system. Hence, by employing the state space model with a generalized cost function, we provide an efficient way to estimate the state parameters. The proposed family of algorithms inherit simplicity in its structure due to the use of stochastic gradient approach in contrast to the other model-based algorithms such as Kalman filter and its variants. This fact is supported by providing a comparison of the computational complexities of these algorithms. More specifically, the proposed family of algorithms has computational complexity far lesser than that of the Kalman filter. The stability of the proposed family of algorithms is analysed by providing the convergence analysis. Extensive simulations are presented to provide concrete justification and to compare the performances of the proposed family of algorithms with that of the Kalman filter.

Keywords: Adaptive filters; State space least mean algorithms; State space estimation algorithms; Convergence and stability analysis

1 Introduction

Over the past few decades, adaptive filters have gained huge recognition in innumerable applications extending over a wide range of fields. Adaptive filtering is an important part of statistical signal processing, and adaptive filters have been successfully applied in diverse fields such as equalization, noise cancellation, linear prediction, and in system identification [1,2]. Adaptive filters are preferred over conventional filters because of their accuracies due to adaptive capability in the domain of the problem in which it is being used. Adaptive filters automatically adjust their weights according to some adaptive algorithm which is usually based on minimization of a function of the difference between the desired signal and the observed signal [1,2]. The most widely used algorithms for adaptive filters

are the least mean squares (LMS) algorithm [1,2] and the recursive least squares algorithm [1,2].

It is observed that the adaptive filters designed by incorporating the knowledge of state space (SS) model of the system performs better than the ones without it. Amongst the plethora of literature found on adaptive filtering, there are many algorithms that deal with the SS model. For example, the very well-known Kalman filter (KF) [3], which gives the linear optimal solution by calculating the minimum mean square error (MMSE) while utilizing the system model. It optimally estimates on the basis of observations which are subjected to noise and other disturbances. In the nonlinear filtering domain, we have the extended Kalman filter (EKF) [3], unscented Kalman filter (UKF) [4], cubature Kalman filter (CKF) [5], quadrature Kalman filter (QKF) [6] and many other variants of KF [3]. These provide suboptimal solution to the filtering problem. Although numerous techniques exist in literature for estimation of state parameters utilizing state space model but most of them are either highly computationally complex or have less accuracy in estimating the

*Correspondence: mmsansari@kau.edu.sa
[1] Center of Excellence in Intelligent Engineering Systems (CEIES), King Abdul Aziz University, Abdullah Rida Street, P.O. Box. 80204, Jeddah 21859, Saudi Arabia
[2] Electrical and Computer Engineering Department, King Abdul Aziz University, Computer Street, P.O. Box. 80204, Jeddah 21859, Saudi Arabia

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 2 of 16

state parameters. Recently, the SS version of LMS and RLS are developed in [7-10] with the aim to provide an alternative to the highly-complex KF techniques. This is the very reason for carrying out our investigation.

It is found in the literature that all of the state space-based adaptive filtering estimation algorithms can be formulated using the generalized form [3]:

$$\hat{x}[k] = \bar{x}[k] + K[k]\,\varepsilon[k] \qquad (1)$$

where:

$$\varepsilon[k] = y[k] - \bar{y}[k] \qquad (2)$$

Equation 1 forms the basis of most of the state-space-based estimation algorithms [3-6]. The usual practice to derive the gain $\mathbf{K}[k]$ is to employ least squares solution [1]. Usually the square of the error is minimized; however, non-mean square errors have also been studied [11]. In this paper, we propose to minimize a more general cost function given by $J = E[\|\varepsilon\|^{2L}]$, where the notation '$\|.\|$' is used to represent the Euclidean norm and L is a positive integer value where $L = 1, 2, 3, ...$ for the basic state space least mean square (SSLMS) algorithms. Our main contribution in this work is to develop a family of adaptive algorithms which has much lesser computational cost as compared to the existing state space model-based adaptive algorithms [3-6]. We provide a detailed comparison of computational cost to support this argument in Section 6.

The paper is organized as such: Section 2 of the paper introduces the state space model. Then in Section 3, the proposed general SSLM algorithm is derived. Section 4 presents the convergence analysis followed by simulation results and comparison of the different algorithms in Section 5. Section 6 presents the computational complexity of the algorithms, and finally, X we conclude the paper in Section 7.

2 State space model
We begin by defining the general state space model of a linear time varying system.

$$\mathbf{x}[k+1] = \mathbf{A}[k]\,\mathbf{x}[k] + \mathbf{B}[k]\,\mathbf{u}[k] + \mathbf{w}[k], \qquad (3a)$$
$$\mathbf{y}[k] = \mathbf{C}[k]\,\mathbf{x}[k] + \mathbf{D}[k]\,\mathbf{u}[k] + \mathbf{v}[k] \qquad (3b)$$

where $\mathbf{x} \in \Re^n$ are the process states, $\mathbf{y} \in \Re^m$ are the measured outputs such that $m \leq n$. $\mathbf{A}[k]$ is the state transition matrix, $\mathbf{B}[k]$ is the input matrix, $\mathbf{u}[k]$ is the input vector where $\mathbf{u} \in \Re^p$, $\mathbf{w} \in \Re^n$ is the process noise vector and $\mathbf{v} \in \Re^m$ is the measurement noise vector. The matrix $\mathbf{C}[k]$ is the output matrix where $\mathbf{dim}[\mathbf{C}[k]] = m \times n$, $\mathbf{D}[k]$ is the feed through matrix with $\mathbf{dim}[\mathbf{D}[k]] = m \times p$. It is assumed that the above system is observable. A special

case is the unforced (autonomous) linear time varying system, represented as:

$$\mathbf{x}[k+1] = \mathbf{A}[k]\,\mathbf{x}[k] + \mathbf{w}[k], \qquad (4a)$$
$$\mathbf{y}[k] = \mathbf{C}[k]\,\mathbf{x}[k] + \mathbf{v}[k] \qquad (4b)$$

The state space representation for a nonlinear continuous time system is:

$$\dot{x} = f(x, u, w), \qquad (5a)$$
$$y = h(x, u, v) \qquad (5b)$$

where f and h are nonlinear functions and the parameters are as defined before.

3 Derivation of the proposed general SSLM algorithm
Considering the system described by Equation 4 above, a model-based adaptive estimation process can be divided into the following two steps. Step 1 is the time update which is given by:

$$\bar{x}[k] = A[k-1]\hat{x}[k-1] \qquad (6)$$

Step 2 is the measurement update which is given by:

$$\hat{x}[k] = \bar{x}[k] + K[k]\,\varepsilon[k] \qquad (7)$$

where $\varepsilon[k]$ is the prediction error defined as:

$$\varepsilon[k] = y[k] - \bar{y}[k] \qquad (8)$$

here, $\mathbf{y}[k]$ is as mentioned in Equation 4, $\mathbf{K}[k]$ is the gain matrix and:

$$\bar{y}[k] = C[k]\,\bar{x}[k] \qquad (9)$$

Equations 6 to 9 constitute the basic structure employed in all KF techniques [3-6]. From Equations 6, 8, and 9, Equation 7 can be written as:

$$\hat{x}[k] = [I - K[k]\,C[k]]\,A[k-1]\,\hat{x}[k-1] + K[k]\,y[k] \qquad (10)$$

The measurement update of the class of adaptive filter governed by Equation 7 can be set up in a more generalized form as follows:

$$\hat{x}[k] = \bar{x}[k] - \mu \nabla J[k] \qquad (11)$$

where $J[k]$ is the cost function to be minimized, $\nabla J[k]$ is the gradient of the cost function with respect to the predicted states, and μ is the step size parameter. To derive the proposed SSLM algorithms, we start by defining the general cost function as:

$$J[k] = \mathbf{E}\left[\|\varepsilon[k]\|^{2L}\right] \text{ for } L = 1, 2, 3, ... \qquad (12)$$

Minimizing the cost function $J[k]$ with respect to the predicted states ($\bar{x}[k]$) result in:

$$\nabla J[k] = -2L\|\varepsilon[k]\|^{2L-2}\mathbf{C}^T[k]\,\varepsilon[k] \qquad (13)$$

Substituting Equation 13 in Equation 11 and generalizing, the Equation 11 can hence be written as:

$$\hat{x}[k] = \bar{x}[k] + \mu \|\varepsilon[k]\|^{2L-2} GC^T[k]\,\varepsilon[k] \tag{14}$$

which is our general estimator algorithm. A matrix **G** was imposed for the condition of controllability [7] which is required due to the dynamics of the system where the algorithm is being applied. Comparing Equations 7 and 14 yields:

$$\mathbf{K}[k] = \mu \|\varepsilon[k]\|^{2L-2} \mathbf{G}C^T[k] \tag{15}$$

It should be noted that for $L = 1$, the algorithm results in the basic SSLMS [7].

4 Convergence in the mean analysis

Before proceeding to the convergence analysis of the proposed algorithm, we set up the stage by putting forth the following assumptions.

1. The noise vectors $\mathbf{w}[k]$ and $\mathbf{v}[k]$ are zero-mean white processes with covariance matrices $\mathbf{Q}_w = \sigma_w^2\mathbf{I}$ and $\mathbf{Q}_v = \sigma_v^2\mathbf{I}$, respectively. Moreover, they are independent of the input and state variables of the system.
2. The system matrices $\mathbf{A}[k]$ and $\mathbf{C}[k]$ are independent of the state variables to be estimated. Hence, they can be treated as deterministic variables.
3. The filter's length is long enough to apply the law of long adaptive filters [2].

The first assumption is a well-known assumption and is also true in real practice. The second assumption is true for linear seperable systems but not true for nonlinear and inseparable systems. However, we can employ this assumption to make the analysis tractable. Moreover, this assumption is valid in most of the practical scenarios. The third assumption is also well-known in literature and often used in the analysis of adaptive filters [2].

Considering the general SSLMS algorithm given in Equation 14 which is reproduced here for convenience:

$$\hat{x}[k] = \bar{x}[k] + \mu \|\varepsilon[k]\|^{2L-2} GC^T[k]\,\varepsilon[k] \tag{16}$$

After substituting the expression for predicted state from Equation 6 in the above equation and using the definition of vector norm, we can rewrite Equation 16 as:

$$\hat{x}[k] = A[k-1]\hat{x}[k-1] + \mu(\varepsilon^T[k]\,\varepsilon[k])^{L-1}C^T[k]\,\varepsilon[k] \tag{17a}$$

$$= A[k-1]\hat{x}[k-1] + \mu((y[k]-\bar{y}[k])^T(y[k]-\bar{y}[k]))^{L-1}$$
$$\times GC^T[k](y[k]-\bar{y}[k]) \tag{17b}$$

To proceed further, we express the prediction error $\varepsilon[k]$ in terms of the actual states and the estimated states. To do so, we substitute the values of $y[k]$ from Equation 4 and $\bar{y}[k]$ from Equation 9 in the expression of $\varepsilon[k]$ to obtain:

$$y[k]-\bar{y}[k] = C[k](x[k]-\bar{x}[k]) + v[k] \tag{18a}$$

$$= C[k](A[k-1]x[k-1] + w[k-1]$$
$$- A[k-1]\hat{x}[k-1]) + v[k] \tag{18b}$$

$$= C[k]A[k-1](x[k-1]-\hat{x}[k-1])$$
$$+ C[k]w[k-1] + v[k] \tag{18c}$$

Defining the state estimation error vector $\tilde{x}[k]$ as:

$$\tilde{x}[k] = \hat{x}[k] - x[k] \tag{19}$$

enables Equation 18 to be rewritten as:

$$y[k]-\bar{y}[k] = -C[k]A[k-1]\tilde{x}[k-1]$$
$$+ C[k]w[k-1] + v[k] \tag{20}$$

Now, the substitution of the above result in Equation 17 and subtraction of $\mathbf{x}[k]$ from both sides results in:

$$\hat{x}[k]-x[k] = A[k-1]\hat{x}[k-1] - x[k]$$
$$+ \mu[(-C[k]A[k-1]\tilde{x}[k-1]$$
$$+ C[k]w[k-1] + v[k])^T(-C[k]$$
$$\times A[k-1]\tilde{x}[k-1] + C[k]w[k-1]$$
$$+ v[k])]^{L-1}GC^T[k](-C[k]A[k-1]\tilde{x}[k-1]$$
$$+ C[k]w[k-1] + v[k]) \tag{21}$$

Next, by substituting $x[k] = A[k-1]x[k-1] + w[k-1]$ in Equation 21 on the right hand side and simplifying will give:

$$\tilde{x}[k] = A[k-1]\tilde{x}[k-1] - w[k-1]$$
$$+ \mu[(-C[k]A[k-1]\tilde{x}[k-1]$$
$$+ C[k]w[k-1] + v[k])^T(-C[k]A[k-1]$$
$$\times \tilde{x}[k-1] + C[k]w[k-1] + v[k])]^{L-1}GC^T[k]$$
$$\times (-C[k]A[k-1]\tilde{x}[k-1] + C[k]w[k-1] + v[k]) \tag{22}$$

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 4 of 16

The complete mean convergence analysis for the generalized case with arbitrary value of L is quite involved. Therefore, we present here special scenarios of $L = 1$ and $L = 2$. Thus, simplifying Equation 22 for the case when $L = 1$ results in:

$$\tilde{x}[k] = \mathbf{A}[k-1]\tilde{x}[k-1] - \mathbf{w}[k-1]$$

$$+ \mu \mathbf{G}\mathbf{C}^T[k]\left(-\mathbf{C}[k]\mathbf{A}[k-1]\tilde{x}[k-1]\right.$$

$$+ \mathbf{C}[k]\mathbf{w}[k-1] + \mathbf{v}[k]) \tag{23a}$$

$$= \mathbf{A}[k-1]\tilde{x}[k-1]$$

$$- \mathbf{w}[k-1] - \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{A}[k-1]\tilde{x}[k-1]$$

$$+ \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{w}[k-1] \tag{23b}$$

$$+ \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{v}[k] \tag{23c}$$

$$= (\mathbf{I} - \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k])\mathbf{A}[k-1]\tilde{x}[k-1]$$

$$+ \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{w}[k-1]$$

$$+ \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{v}[k] - \mathbf{w}[k-1] \tag{23d}$$

Now, taking expectation on both sides of the above equation and by employing Assumptions 1 and 2 will result in:

$$\mathbf{E}[\tilde{x}[k]] = (\mathbf{I} - \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k])\mathbf{A}[k-1]\mathbf{E}[\tilde{x}[k-1]] \tag{24}$$

Hence, it can be observed that the state estimation error vector will converge in the mean sense provided that the following conditions are fulfilled:

1. The matrices $\mathbf{A}[k]$, $\mathbf{A}[k-1]$, ... should have bounded entries, i.e.:

   $$|\mathbf{A}[k]\{i,j\}| < 1, \quad \forall k \tag{25}$$

   for $i = 1, 2, 3, ..., n$ and $j = 1, 2, 3, ..., n$.
2. $|\mathbf{I} - \mu \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]| < 1$ which implies that the step size μ of the algorithm is bounded by:

   $$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k])}, \quad \forall k \tag{26}$$

   where $\lambda_{\max}(\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k])$ represents the largest eigenvalue of $\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]$.
3. System or the states should be observable, i.e. matrix $\mathbf{C}[k]$ is full rank.

The above conditions correspond to the bounds for the basic SSLMS algorithm ($L = 1$). Thus, our analysis provides the mean convergence bounds of the existing SSLMS algorithm which was not provided in the original work [7]. Similarly, for $L = 2$, the mean convergence analysis could be performed for the corresponding algorithm in

the family. On substitution of $L = 2$, simplification and taking expectation of Equation 22 results in:

$$\mathbf{E}[\tilde{x}[k]] = \{\mathbf{I} - \mu(\gamma \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{Q}_w\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \mathbf{G}\mathbf{C}^T[k]\mathbf{Q}_v\mathbf{C}[k] + \mathbf{G} \tag{27a}$$

$$\times \mathbf{C}^T[k]\mathbf{C}[k]\mathbf{Q}_w\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \sigma_w^2 Tr(\mathbf{C}^T[k]\mathbf{C}[k])\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \mathbf{G}\mathbf{C}^T[k]\mathbf{Q}_v\mathbf{C}[k] \tag{27b}$$

$$+ \sigma_v^2\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k])\}\mathbf{A}[k-1]\mathbf{E}[\tilde{x}[k-1]] \tag{27c}$$

where $\gamma = \|\tilde{x}\|_\zeta^2$ which is considered a constant due to the employment of the Assumption 3. Here, the term ζ is given by:

$$\zeta = \mathbf{A}^T[k-1]\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{A}[k-1] \tag{28}$$

Let us assume a matrix $\mathbf{Z}[k]$, such that:

$$\mathbf{Z}[k] = (\gamma \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k] + \mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]\mathbf{Q}_w\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \mathbf{G}\mathbf{C}^T[k]\mathbf{Q}_v\mathbf{C}[k] + \mathbf{G} \tag{29a}$$

$$\times \mathbf{C}^T[k]\mathbf{C}[k]\mathbf{Q}_w\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \sigma_w^2 Tr(\mathbf{C}^T[k]\mathbf{C}[k])\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]$$

$$+ \mathbf{G}\mathbf{C}^T[k]\mathbf{Q}_v\mathbf{C}[k] \tag{29b}$$

$$+ \sigma_v^2\mathbf{G}\mathbf{C}^T[k]\mathbf{C}[k]) \tag{29c}$$

Therefore, Equation 27 can be written as:

$$\mathbf{E}[\tilde{x}[k]] = (\mathbf{I} - \mu \mathbf{Z}[k])\mathbf{A}[k-1]\mathbf{E}[\tilde{x}[k-1]] \tag{30}$$

Consequently, the mean convergence of the state estimation error vector for $L = 2$ is obtained by fulfilling the Conditions 1 and 3 reported for $L = 1$ except the bound for step size μ which is given by:

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{Z}[k])}, \quad \forall k \tag{31}$$

The analysis for the higher values of L can be similarly carried out.

5 Simulation results and discussion

In this section, we present simulation results to validate the performance of the proposed family and to compare its performance with that of the standard KF. More specifically, we aim to compare mean square error performance and the convergence speed of the aforementioned estimators. For this purpose, we investigate the following three experiments:

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 5 of 16

1. Estimation of the state parameters of a noisy sinusoid tracking problem.
2. Estimation of the state parameters of a Van der pol oscillator.
3. Estimation of the state parameters of a symmetrical three-phase induction machine.

For all of the above experiments, we investigate three different noise environments, namely Gaussian, Uniform, and Laplacian. We also investigate the effect of exponent L for the aforementioned noise environments. More specifically, we set $L = 1, 2, 3$, and 4 which corresponds to SSLMS, state space least mean fourth (SSLMF), state space least mean sixth (SSLMSi), and state space least mean eighth (SSLME) algorithm, respectively. The results are reported after averaging of 100 independent simulation experiments.

5.1 Example 1. Tracking sinusoids

In the first example, we consider the system reported in [7]. More specifically, we investigate a second order transversal filter with known frequency and unknown
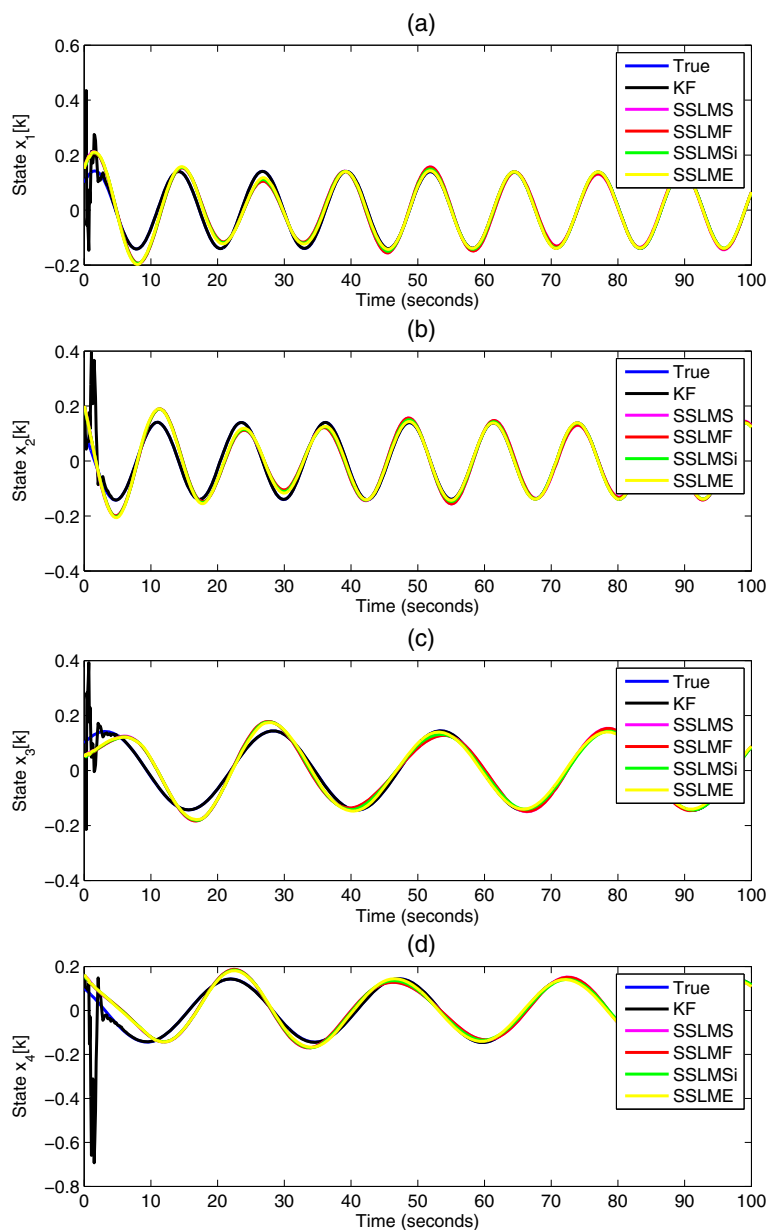
Figure 1 State $x_1[k]$ **(a)**, $x_2[k]$ **(b)**, $x_3[k]$ **(c)**, and $x_4[k]$ **(d)**.

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39
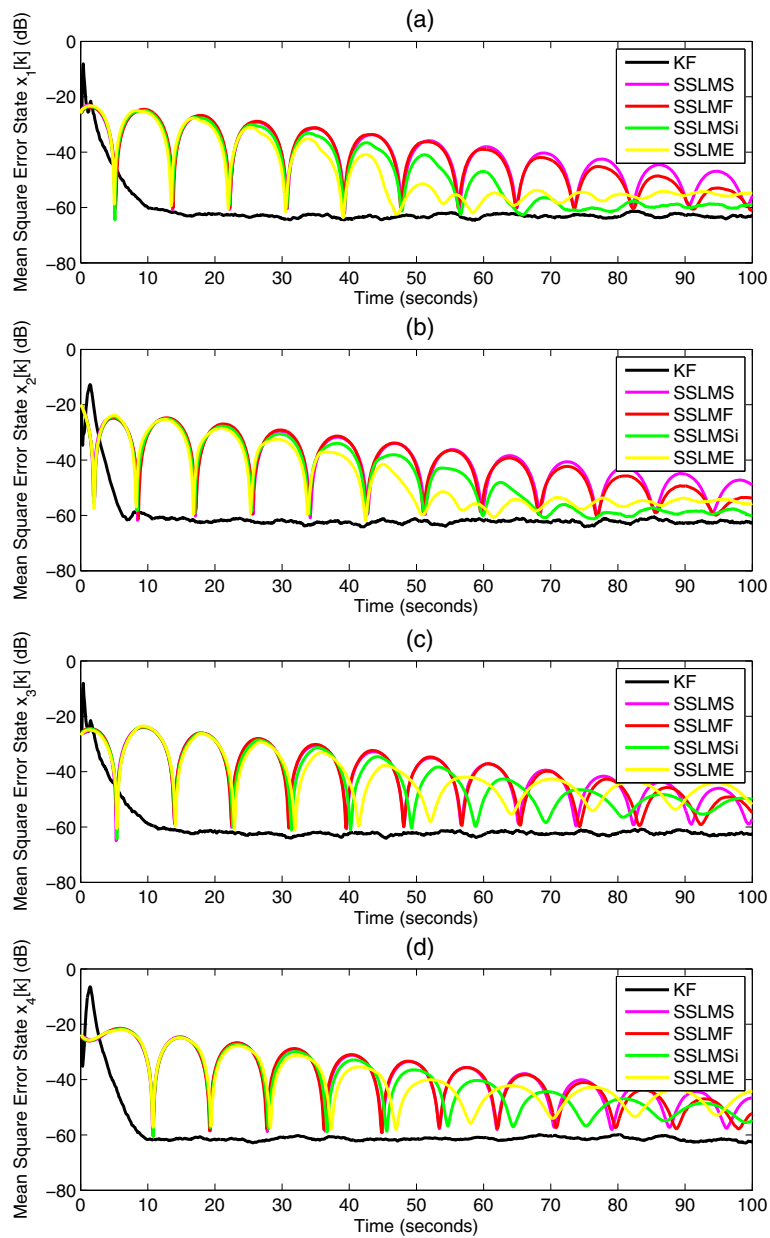
Page 6 of 16

Figure 2 Mean square error state $x_1[k]$ **(a)**, $x_2[k]$ **(b)**, $x_3[k]$ **(c)**, and $x_4[k]$ **(d)**.
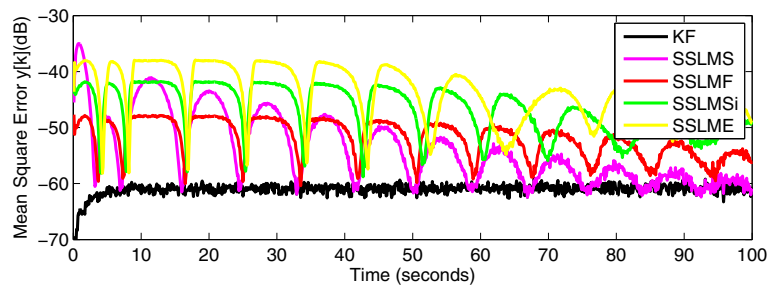
Figure 3 Mean square error $y[k]$ (Example 1).

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 7 of 16

Table 1 Root mean square error of example 1

| | | Root mean square error (dB) | | | | |
| | | Example 1 | | | | |
		KF	SSLMS	SSLMF	SSLMSi	SSLME
State x_1	Gaussian	−17.3726	−16.4997	−16.4695	−16.7676	−16.9518
	Uniform	−17.1419	−16.4963	−16.4671	−16.7656	−16.9406
	Laplace	−17.6115	−16.4901	−16.5112	−16.8499	−17.0366
State x_2	Gaussian	−17.0558	−16.4167	−16.3911	−16.6666	−16.8323
	Uniform	−16.8535	−16.4157	−16.3910	−16.6672	−16.8250
	Laplace	−17.2894	−16.4092	−16.4314	−16.7432	−16.9055
State x_3	Gaussian	−17.3597	−16.3199	−16.2747	−16.5047	−16.6701
	Uniform	−17.1301	−16.3164	−16.2717	−16.5023	−16.6601
	Laplace	−17.5963	−16.3101	−16.2960	−16.5595	−16.7142
State x_4	Gaussian	−14.0092	−15.5096	−15.4791	−15.6395	−15.7546
	Uniform	−13.8130	−15.5070	−15.4768	−15.6376	−15.7471
	Laplace	−14.2523	−15.5024	−15.4936	−15.6762	−15.7891
Observation y	Gaussian	−30.4409	−24.3831	−25.3871	−22.6547	−20.8293
	Uniform	−30.4612	−24.3844	−25.3869	−22.6497	−20.9188
	Laplace	−30.4614	−24.3817	−24.8789	−21.9582	−19.3682

phase and amplitude of sinusoids which produces a 4th order system given by:

$$A[k] = \mathbf{diag}\left\{ \begin{bmatrix} \cos(\omega_i T) & \sin(\omega_i T) \\ -\sin(\omega_i T) & \cos(\omega_i T) \end{bmatrix} \right\}, \ i = 1, 2 \quad (32a)$$

$$C[k] = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \quad (32b)$$

where ω_i's represent frequencies of sinusoids which are known and constant. For the purpose of our study, we set the values of the frequencies ω_1, ω_2 to 0.5 and 0.25, respectively. T represents the sampling time which is considered to be 0.1 s for our study. The step size μ for the family of algorithms is adjusted to achieve optimum performance. Variance of the observation noise used is $\sigma_v^2 = 0.001^2$ while the process noise is considered with variance $\sigma_w^2 = 0.0001^2$. The **G** matrix is taken as all zero entries except for the first column with 1's. The actual initial system states are considered to be $\mathbf{x}[0] = [0.1 \ 0.1 \ 0.1 \ 0.1]^T$, and
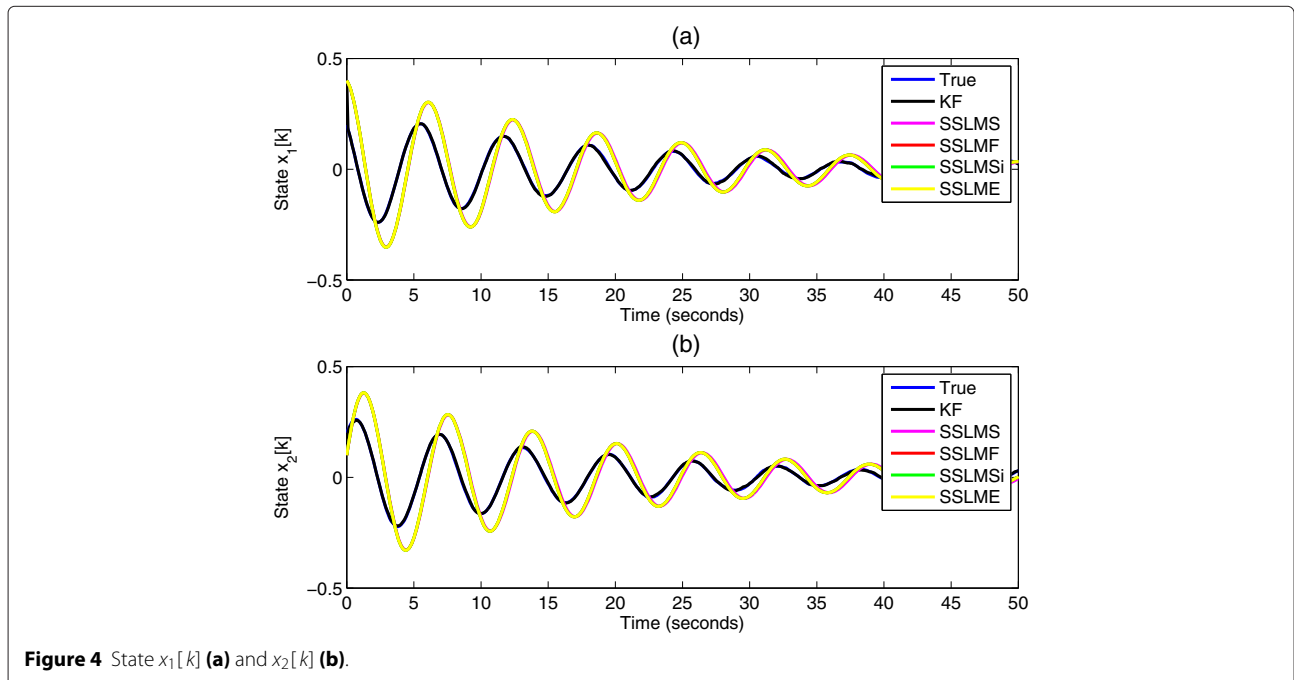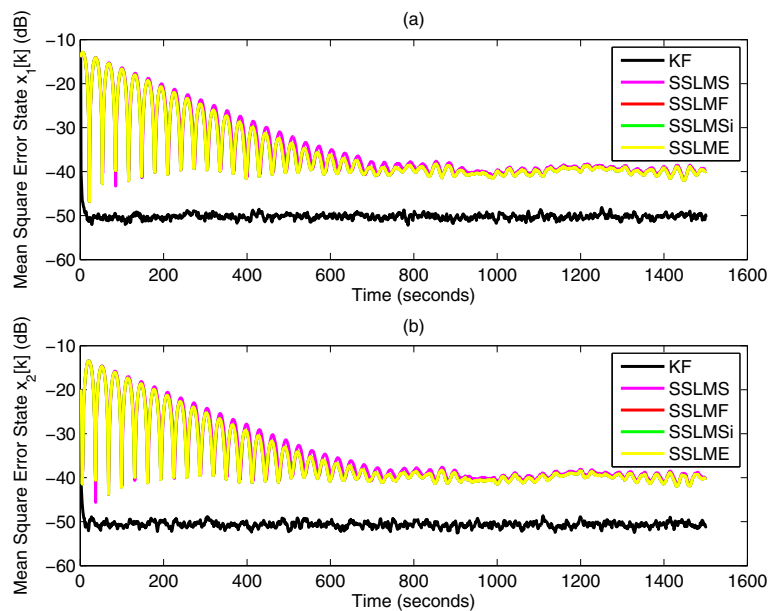

Figure 4 State $x_1[k]$ **(a)** and $x_2[k]$ **(b)**.

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 8 of 16

Figure 5 Mean square error $x_1[k]$ **(a)** and $x_2[k]$ **(b)**.

the initial estimate for tracking are chosen to be $\hat{\mathbf{x}}[0] = [0.15\ 0.2\ 0.05\ 0.16]^T$. In Figure 1, the true value of all the four states are compared with the considered algorithms. It can be depicted from Figure 1 that the performance of KF is better than the proposed family of algorithms. This fact is further elaborated in Figure 2 where the mean square error for these estimators are compared. However, this superiority of the KF is achieved at the expense of

high-computational cost. The KF performs approximately six times as many operations as the proposed family of algorithms (refer Section 6). The plot of the mean square observation error in the presence of Gaussian noise are presented in Figure 3. It is observed from Figure 1 that all of the SSLM algorithms are performing well in terms of estimation. A closer look at Figure 2 reveals that SSLME converged faster. Moreover, Table 1 verifies that SSLME
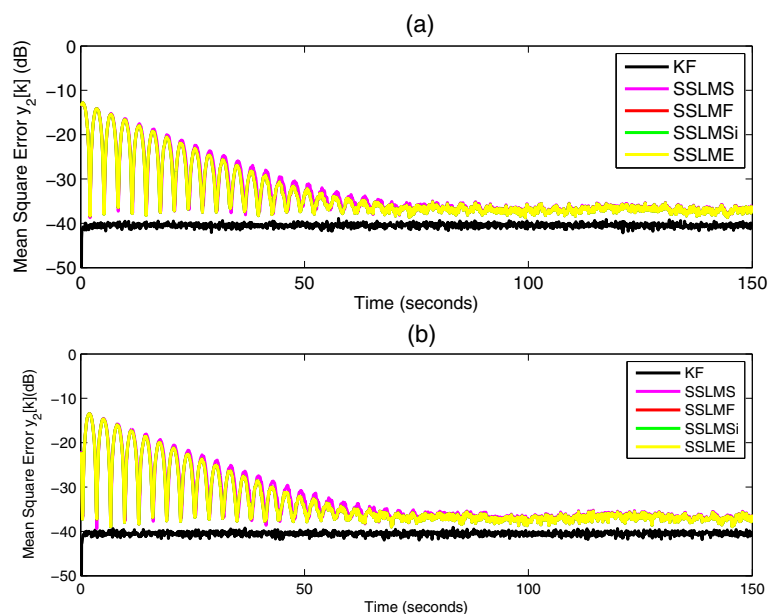

Figure 6 Mean square error **(a)** $y_1[k]$ and **(b)** $y_2[k]$ (Example 2).

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 9 of 16

Table 2 Root mean square error of Example 2

| | | Root mean square error (dB) | | | | |
| | | Example 2 | | | | |
		KF	SSLMS	SSLMF	SSLMSi	SSLME
State x_1	Gaussian	−22.1985	−12.4088	−12.5740	−12.5760	−12.5760
	Uniform	−22.1974	−13.2778	−13.4407	−13.4426	−13.4427
	Laplace	−22.1995	−12.2850	−13.4015	−13.4027	−13.4053
State x_2	Gaussian	−24.0418	−12.5487	−12.7486	−12.7511	−12.7511
	Uniform	−24.0494	−13.4124	−13.6097	−13.6121	−13.6121
	Laplace	−24.0450	−12.2753	−13.5716	−13.5730	−13.5763
Observation y_1	Gaussian	−20.2142	−12.3685	−12.5290	−12.5309	−12.5310
	Uniform	−20.2185	−13.2086	−13.3642	−13.3660	−13.3660
	Laplace	−20.2284	−12.2460	−13.3274	−13.3285	−13.3311
Observation y_2	Gaussian	−20.2526	−12.4860	−12.6800	−12.6824	−12.6824
	Uniform	−20.2403	−13.3160	−13.5048	−13.5071	−13.5071
	Laplace	−20.2426	−12.2205	−13.4735	−13.4749	−13.4780

performs the best in comparison to other algorithms in the family.

5.2 Example 2. Van der pol oscillator

In this example, we explore the commonly used Van der pol oscillator which is a highly nonlinear system exhibiting both stable and unstable limit cycles [12]. We consider the case of unstable limit cycle therefore, as time proceeds the system states approach to zero. The system is represented by the following differential equations:

$$\dot{x}_1 = -x_2 \tag{33a}$$

$$\dot{x}_2 = x_1 - \alpha(1 - x_1{}^2)x_2 \tag{33b}$$

where α is a constant which is set to 0.2. The state space representation for the system shown above takes the following form:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & -1 \\ 1 & -\alpha(1 - x_1{}^2) \end{bmatrix} \mathbf{x}(t) \tag{34a}$$

$$\mathbf{y}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(t) \tag{34b}$$

The system is discretized with a sampling time of 0.1 s. The observation is subject to a noise of variance $\sigma_v^2 = 0.01^2$. The process noise is considered to be of variance $\sigma_w^2 = 0.001^2$. The true initial system states are considered to be $\mathbf{x}[0] = [0.2\ 0.2]^T$, and the initial estimate for the algorithms are $\hat{\mathbf{x}}[0] = [0.4\ 0.1]^T$. The **G** matrix is taken as all zero entries except for the first column with 1's. For this experiment, the states, their mean square error, and observation mean square error are plotted only for uniform noise scenario in Figures 4, 5, and 6, respectively. However, the root mean square error (RMSE) results for all the noise environments are reported in Table 2. It can

be depicted from Figures 4, 5, and 6 that the KF performs better and has a faster convergence in comparison to the SSLM algorithms. However, this comes at the cost of high computational complexity as will be discussed in Section 6. The SSLM algorithms have slower convergence with SSLMSi, and SSLME having the best performance among the others in the family.

5.3 Example 3. Symmetrical three phase induction machine

Example 3 presents estimation of the states of a nonlinear symmetrical three phase induction machine, more specifically, the flux and angular velocity estimation ([13-15]). The state space representation of the induction machine is as follows:

$$\dot{\mathbf{x}}(t) = [x_1(t)\ x_2(t)\ x_3(t)\ x_4(t)\ x_5(t)]^T \tag{35a}$$

$$\mathbf{u}(t) = [z_1(t)\ z_2(t)\ z_3(t)]^T \tag{35b}$$

$$\mathbf{y}(t) = [y_1(t)\ y_2(t)]^T \tag{35c}$$

where:

$$\dot{x}_1(t) = k_1 x_1(t) + z_1(t)x_2(t) + k_2 x_3(t) + z_2(t) \tag{36a}$$

$$\dot{x}_2(t) = -z_1(t)x_1(t) + k_1 x_2(t) + k_2 x_4(t) \tag{36b}$$

$$\dot{x}_3(t) = k_3 x_1(t) + k_4 x_3(t) + (z_1(t) - x_5(t))x_4(t) \tag{36c}$$

$$\dot{x}_4(t) = k_3 x_2(t) - (z_1(t) - x_5(t))x_3(t) + k_4 x_4(t) \tag{36d}$$

$$\dot{x}_5(t) = k_5(x_4(t)x_1(t) - x_2(t)x_3(t)) + k_6 z_3(t) \tag{36e}$$

$$y_1(t) = k_7 x_1(t) + k_8 x_3(t) \tag{36f}$$

$$y_2(t) = k_7 x_2(t) + k_8 x_4(t) \tag{36g}$$

The normalized state variables $x_1(t)$, $x_2(t)$, and $x_3(t)$, $x_4(t)$ are the components of the stator and the rotor flux,
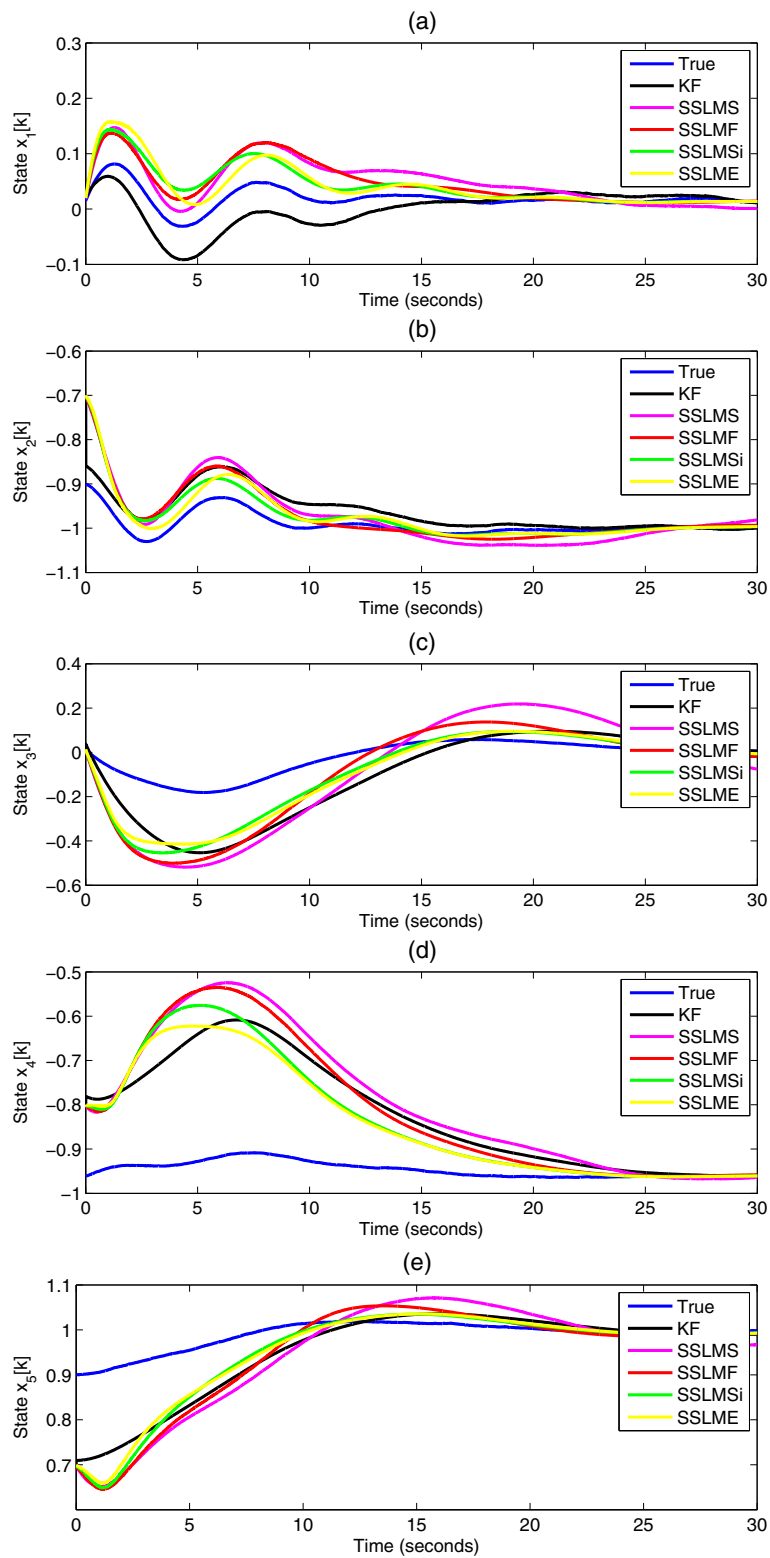
Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 10 of 16

Figure 7 Induction machine state x_1 **(a)**, x_2 **(b)**, x_3 **(c)**, x_4 **(d)**, and $\boldsymbol{x_5}$ **(e)**.

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39
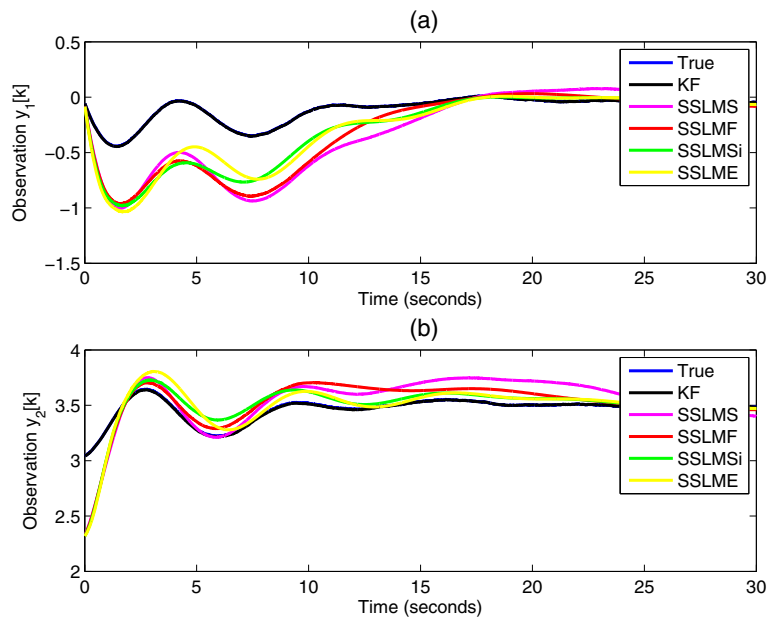
Page 11 of 16

Figure 8 Observation y_1 **(a)** and y_2 **(b)**.

respectively, and $x_5(t)$ is the angular velocity. The inputs $z_1(t)$, $z_2(t)$, and $z_3(t)$ are the frequency, amplitude of the stator voltage, and the load torque, respectively. $k_1 = -0.186$, $k_2 = 0.178$, $k_3 = 0.225$, $k_4 = -0.234$, $k_5 = -0.081$, $k_6 = 4.643$, and $k_7 = -4.448$, $k_8 = 1$. The values of ks depend on the induction machine considered. Outputs $y_1(t)$ and $y_2(t)$ are the normalized stator currents. For the simulation purpose, the system is discretized at a

sampling interval of 0.01 s and $z_1(t)$, $z_2(t)$, and $z_3(t)$ are chosen as 1, 1, and 0, respectively. System was simulated in the presence of all types of noises, and μs were well tuned to get optimum performance. The **G** matrix is taken as all zero entries except for the first column with tens. The process noise variance is considered as $\sigma_w^2 = 0.0001^2$, and the observation noise variance is $\sigma_v^2 = 0.001^2$. The initial true state is considered to be $\mathbf{x}[0] = [0.0147 \quad -$
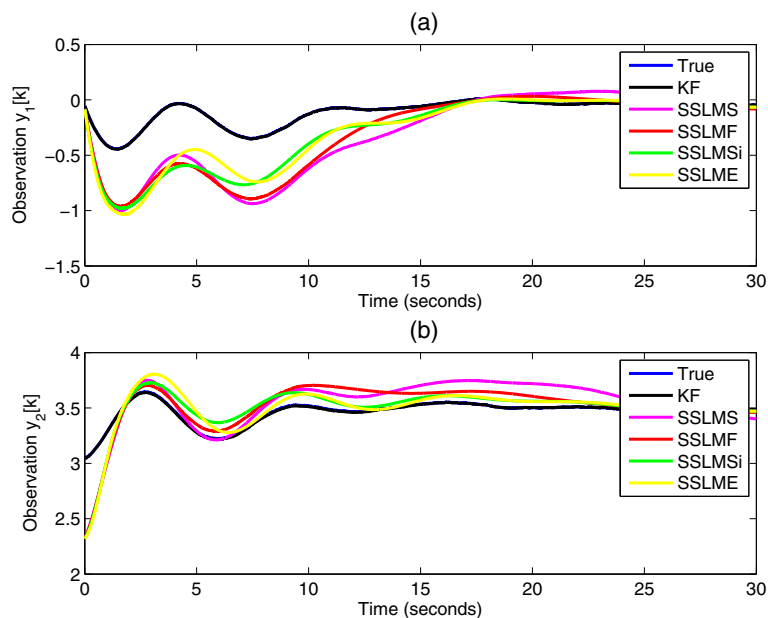

Figure 9 Mean square error state x_1 **(a)**, x_2 **(b)**, x_3 **(c)**, x_4 **(d)**, and x_5 **(e)**.

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39
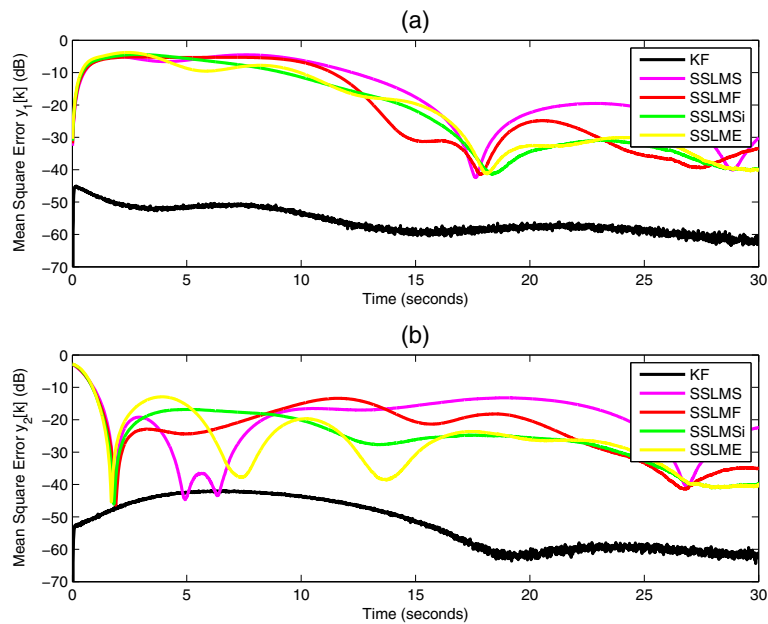
Page 12 of 16

Figure 10 Mean square error **(a)** $y_1[k]$ and **(b)** $y_2[k]$ (Example 3).

Table 3 Root mean square error of Example 3

		Root mean square error (dB)				
		Example 3				
		KF	**SSLMS**	**SSLMF**	**SSLMSi**	**SSLME**
State x_1	Gaussian	−14.9484	−13.9622	−14.1395	−14.6637	−14.6965
	Uniform	−14.9438	−13.9619	−14.1506	−14.6729	−14.7018
	Laplace	−14.9330	−13.9600	−14.1397	−14.6654	−14.6985
State x_2	Gaussian	−14.2212	−13.1412	−13.7200	−14.3676	−14.3167
	Uniform	−14.2242	−13.1367	−13.7295	−14.3708	−14.4115
	Laplace	−14.2198	−13.1355	−13.7170	−14.3652	−14.3153
State x_3	Gaussian	−8.4244	−7.2323	−7.9750	−8.7182	−8.8679
	Uniform	−8.4201	−7.2197	−7.9772	−8.7156	−8.7881
	Laplace	−8.4095	−7.2197	−7.9665	−8.7122	−8.8612
State x_4	Gaussian	−7.8682	−6.8915	−7.1522	−7.8059	−8.1534
	Uniform	−7.8713	−6.8930	−7.1616	−7.8149	−7.9130
	Laplace	−7.8669	−6.8911	−7.1511	−7.8057	−8.1535
State x_5	Gaussian	−11.1405	−9.9399	−10.2366	−10.7061	−8.1534
	Uniform	−11.1391	−9.9325	−10.2373	−10.7057	−7.9130
	Laplace	−11.1303	−9.9349	−10.2329	−10.7033	−8.1535
Observation y_1	Gaussian	−26.9886	−4.7978	−5.0163	−5.4818	−5.6242
	Uniform	−26.9860	−4.7960	−5.0221	−5.4861	−5.5343
	Laplace	−26.9872	−4.7963	−5.0150	−5.4816	−5.6243
Observation y_2	Gaussian	−23.66444	−7.9033	−8.5351	−9.0782	−8.9918
	Uniform	−23.6616	−7.8941	−8.54861	−9.0833	−9.0834
	Laplace	−23.6558	−7.8925	−8.5351	−9.0773	−8.9919

Table 4 Computational complexity of SSLM algorithm

Equation number	Operation	Multiplication	Additions
6	$\bar{\mathbf{x}}[k]_{n\times 1} = \mathbf{A}[k-1]_{n\times n}\,\hat{\mathbf{x}}[k-1]_{n\times 1}$	n^2	$n^2 - n$
8	$\varepsilon[k]_{m\times 1} = \mathbf{y}[k]_{m\times 1}\,\bar{\mathbf{y}}[k]_{m\times 1}$	0	m
9	$\bar{\mathbf{y}}[k]_{m\times 1} = \mathbf{C}[k]_{m\times n}\,\bar{\mathbf{x}}[k]_{n\times 1}$	mn	$nm - m$
7	$\hat{\mathbf{x}}[k]_{n\times 1} = \bar{\mathbf{x}}[k]_{n\times 1} + \mathbf{K}[k]_{n\times m}\,\varepsilon[k]_{m\times 1}$	mn	nm
15	$\mathbf{K}[k]_{n\times m} = \mu_{1\times 1}||\varepsilon[k]_{m\times 1}||^{2L-2}\mathbf{G}_{n\times n}\mathbf{C}^{T}[k]_{n\times m}$	$mn^2 + mn + m + L - 1$	$mn^2 - mn + m - 1$
Total for the SSLM algorithm		$3mn + n^2 + mn^2 + m + L - 1$	$m + mn^2 + n^2 + mn - n - 1$

Table 5 Computational complexity of KF algorithm

Step	Operation	Multiplication	Additions
Predict	$\bar{\mathbf{x}}[k]_{n\times 1} = \mathbf{A}[k-1]_{n\times n}\,\hat{\mathbf{x}}[k-1]_{n\times 1}$	n^2	$n^2 - n$
	$\bar{\mathbf{P}}[k]_{n\times n} = \mathbf{A}[k-1]_{n\times n}\,\mathbf{P}[k-1]_{n\times n}\,\mathbf{A}^T[k-1]_{n\times n} + \mathbf{Q}[k]_{n\times n}$	$2n^3$	$2n^3 - n^2$
Update	$\varepsilon[k]_{m\times 1} = \mathbf{y}[k]_{m\times 1} - \mathbf{C}[k]_{m\times n}\,\bar{\mathbf{x}}[k]_{n\times 1}$	mn	nm
	$\mathbf{S}[k]_{m\times m} = \mathbf{C}[k]_{m\times n}\,\bar{\mathbf{P}}[k]_{n\times n}\,\mathbf{C}^T[k]_{n\times m} + \mathbf{R}[k]_{m\times m}$	$mn^2 + m^2 n$	$m^2 n + mn^2 - mn$
	$\mathbf{K}[k]_{n\times m} = \bar{\mathbf{P}}[k]_{n\times n}\,\mathbf{C}^T[k]_{n\times m}\,\mathbf{S}^{-1}[k]_{m\times m}$	$m^3 + m^2 n + mn^2$	$m^3 + m^2 n$
	$\hat{\mathbf{x}}[k]_{n\times 1} = \bar{\mathbf{x}}[k]_{n\times 1} + \mathbf{K}[k]_{n\times m}\,\varepsilon[k]_{m\times 1}$	mn	nm
	$\mathbf{P}[k]_{n\times n} = (\mathbf{I}_{n\times n} - \mathbf{K}[k]_{n\times m}\,\mathbf{C}[k]_{m\times n})\bar{\mathbf{P}}[k]_{n\times n}$	$mn^2 + n^3$	$mn^2 + n^3 - n^2$
Total for the KF algorithm		$m^3 + 2m^2 n + 3mn^2 + 2mn + 3n^3 + n^2$	$m^3 + 2m^2 n + 3mn^2 - mn + 3n^3 + n^2 - n$

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing* (2015) 2015:39

Page 15 of 16

Table 6 Computational complexity summary of Example 1

	Example 1 ($m = 1, n = 4$)				
Operation	KF	SSLMS	SSLMF	SSLMSi	SSLME
Multiplications	273	45	46	47	48
Additions	257	32	32	32	32
Total operations	530	77	78	79	80

Table 8 Computational complexity summary of Example 3

	Example 3 ($m = 2, n = 5$)				
Operation	KF	SSLMS	SSLMF	SSLMSi	SSLME
Multiplications	618	107	108	109	110
Additions	583	81	81	81	81
Total operations	1201	188	189	190	191

0.9 0.0136 $-$ 0.9616 0.9$]^T$ and the initial estimate for the SSLM algorithms as $\hat{\mathbf{x}}[0] = [\, 0.02 - 0.7\ 0.0136 - 0.8\ 0.7\,]$. Figure 7 presents the five states of the induction machine in presence of Laplacian noise while Figure 8 presents the observations in presence of Laplacian noise. The mean square estimation error and the mean square observation error in the presence of Laplacian noise are presented in Figure 9 and Figure 10, respectively. Figure 9 shows that the algorithms perform well in terms of estimating the states and have comparable performance to the KF. Figure 10 shows us that the estimated observation from KF is near real; however, the corresponding state estimates are not near the true values. The RMSE for this example is presented in Table 3.

An overview of the RMSE of the states and observations can be referred to in Tables 1, 2, and 3.

It is clear from our investigation that as the value of L increases, better performance is observed from the SSLM algorithms. Although the performance of the algorithms is not better in comparison to the KF, nevertheless, the algorithms are extremely low in terms of computational power requirement as will be discussed in Section 6.

6 Computational complexity
When dealing with real-time applications, it is essential to calculate the computational complexities of the algorithm. For this purpose, we provide a comparison of computational complexities of the proposed family of SSLM algorithms with that of the KF. To evaluate the computational complexity of the proposed SSLM family, we compute the total number of operations required by the Equations 6, 8, 9, 7, and 15 which define its implementation in order of execution. The details of operations required by the SSLM family of algorithms and the KF are presented in Tables 4 and 5, respectively. Note that in evaluating the computational cost of inversion of a matrix having dimension $n \times n$, we assume a total of n^3 multiplications and n^3 additions.

Table 7 Computational complexity summary of Example 2

	Example 2 ($m = 2, n = 2$)				
Operation	KF	SSLMS	SSLMF	SSLMSi	SSLME
Multiplications	84	26	27	28	29
Additions	70	15	15	15	15
Total operations	154	41	42	43	44

To get more insight on the computational complexity, we present a comparison of computational complexities of the algorithms for all the three examples investigated in simulation experiments in Tables 6, 7, and 8.

The results clearly suggest that the proposed algorithm has very low complexity as compared to the standard KF algorithm. According to our presented investigation, the KF algorithm perform approximately 6 times as many operations in example 1, 3.5 times as many in example 2, and 6 times as many in example 3 when compared to our algorithms.

7 Conclusions
In this work, the general family of SSLM algorithms is proposed. The proposed family is based on minimizing the general least mean cost function via stochastic gradient optimization. In order to assess the performance of the proposed family, simulation results are carried out for three different examples with different types of noise environments. In these simulations, effect of noise and exponent L on state estimation are investigated. The simulation results show that the performance of the proposed family is efficient and comparable to that of the Kalman filter. However, the computational complexity of the proposed family of algorithms is far lesser than that of the Kalman filter. More specifically, the computational complexity of the proposed family is 3.5 to 6 times lesser than the Kalman filter as presented in the reported examples. This gives a motivation to use our proposed family of algorithm in real-time application where computational complexity is of major concern. For future research, an adaptive μ could be proposed and investigated keeping in focus the effect of varying the sampling time and the noise. Moreover, different variants of KF algorithms and other linear estimation algorithms should be investigated along with the proposed SSLM algorithms family and compared to get more insightful outcome.

Moinuddin *et al. EURASIP Journal on Advances in Signal Processing*   (2015) 2015:39

Page 16 of 16

References
1. S Haykin, *Adaptive Filter Theory*, 3rd edn. (Prentice-Hall, Upper-Saddle River, NJ, 1996)
2. AH Sayed, *Fundamentals of Adaptive Filtering*. (Wiley-Interscience, New York, 2003)
3. CK Chui, G Chen, *Kalman Filtering With Real-Time Applications*, 4th edn. (Springer, Berlin, Heidelberg, 2009)
4. EA Wan, R Van der Merwe, in *IEEE Proceedings of Symposium on Adaptive Systems for Signal Processing, Communication and Control*. The unscented Kalman filter for nonlinear estimation (Lake Louise, Alberta, Canada, 2000)
5. I Arasaratnam, S Haykin, Cubature Kalman filters. Automatic Control IEEE Trans **54**(6), 1254–1269 (2009)
6. I Arasaratnam, S Haykin, RJ Elliott, Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature. Proc. IEEE **95**(5), 953–977 (2007)
7. MB Malik, Salman M, State-space least mean square. Digital Signal Process **18**(3), 334–345 (2008)
8. MB Malik, RA Bhatti, Tracking of linear time-varying systems using state-space least mean square. IEEE Int. Symp. Commun. Inform. Technol **1**, 582–585 (2004)
9. MB Malik, State-space recursive least squares: part I. Signal Process **84/9**, 1709–1718 (2004)
10. MB Malik, State-space recursive least squares: part II. Signal Process **84/9**, 1719–1728 (2004)
11. E Walach, B Widrow, The least-mean fourth (LMF) adaptive algorithm and its family. IEEE Trans. Inform. Theory **IT30**, 275–283 (1984)
12. HK Khalil, *Nonlinear Systems*, 3rd edn. (Prentice Hall, Upper-Saddle River, NJ, 2000)
13. K Reif, F Sonnemann, R Unbehauen, An EKF-based nonlinear observer with a prescribed degree of stability. Automatica **34**(9), 1119–1123 (1998)
14. R Kandepu, B Foss, L Imsland, Applying the unscented Kalman filter for nonlinear state estimation. J. Process Control **18**, 753–768 (2008)
15. L Salvatore, S Stasi, L Tarchioni, A new EKF based algorithm for flux estimation in induction machines. IEEE Trans. Ind. Electron **40**(5), 496–504 (1993)