Available online at www.sciencedirect.com
ScienceDirect
Digital Signal Processing 18 (2008) 334–345
Digital Signal Processing
www.elsevier.com/locate/dsp

State-space least mean square

Mohammad Bilal Malik *, Muhammad Salman

Department of Electrical Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Rawalpindi, Pakistan

Available online 23 May 2007

Abstract

In this paper, we present a generalized form of the well-known least mean square (LMS) filter. The proposed filter incorporates linear time-varying state-space model of the underlying environment and hence is termed as state-space LMS (SSLMS). This attribute results in marked improvement in its tracking performance over the standard LMS. Furthermore, the use of SSLMS in state estimation in control systems is straightforward. Overall performance of SSLMS, however, depends on factors like model uncertainty and time-varying nature of the problem. SSLMS with adaptive memory, having time-varying step-size parameter, provides solutions to such cases. The step-size parameter is iteratively tuned by stochastic gradient method so as to minimize the mean square value of the prediction error. Different computer simulations demonstrate the ability of the algorithms suggested in this paper. A detailed study of computational complexities of the proposed algorithms is carried out at the end.

Keywords: Adaptive filtering; State-space LMS; SSLMS; Tracking

1. Introduction

Adaptive filters have played a vital role in the development of a wide variety of systems, for the last three decades. The philosophy of adaptive filters revolves around recursive least squares (RLS) and least mean square (LMS) [1]. The derivations of standard RLS and LMS assume multiple linear regression model. Whereas, this model is applicable to a wide range of problems, it also causes certain restrictions in the design of adaptive filters when the underlying model of the environment is different. Model dependent nature of the tracking problem makes it specially sensitive to this limitation. Consequently, tracking performance of LMS and RLS have been thoroughly explored ([1–5], etc.). Researchers have been endeavoring to find various forms of LMS and RLS that could fit well in diverse scenarios [1]. Notable amongst these is an important contribution by Sayed and Kailath [6], who have given a state-space model for RLS. Based on [6], Haykin et al. [7] have exploited one-to-one correspondence between RLS and Kalman filter to devise extended RLS (ERLS) algorithms. In our previous work, we have developed SSRLS [8] which takes into consideration the state-space model of the system, thus resulting in a very useful generalization of the standard RLS. SSRLS shows considerable improvement in tracking performance over standard RLS and LMS. Development of

* Corresponding author.
E-mail address: mbmalik@ieee.org (M.B. Malik).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2007.05.003

SSRLS with adaptive memory (SSRLSWAM) [9] adds a level of versatility to this philosophy. SSRLSWAM proves to be an effective tracker even in difficult scenarios as shown in [9].

In this paper we develop state-space least mean square (SSLMS) by incorporating the linear state-space model of the environment, which offers two plus points. Firstly any causal linear system can be represented by a state-space model, thus a designer is not restricted to the linear regression model. Secondly, the multiple-input multiple-output (MIMO) nature of state-space model allows handling vector observations. The standard RLS and LMS, on the other hand, only deal with scalar observations. Appropriate to the nature of generalization, we use the term state-space least mean square (SSLMS). SSLMS was first introduced in [10], where we showed the ability of this new filter to track time-varying systems. As a natural extension of SSLMS, we developed SSLMS with adaptive memory [11]. This algorithm exhibits superior tracking properties under difficult conditions.

This paper commences with a discussion of the state-space model of an unforced time-varying discrete system. The output of the system that may have been corrupted by observation noise is assumed to be available for measurements. This model forms the basis of further development. The derivation of SSLMS, based on the minimum norm solution of an underdetermined system of linear equations, uses only latest observation in the recursive algorithm. The solution with normalization factor is called state-space normalized LMS (SSNLMS), where omitting this factor gives us simply SSLMS. It is shown that the linear regression model is a special case of the general state-space model used in derivation of SSLMS. This in fact shows that SSLMS is a true generalization of the standard LMS. This is followed by a noisy sinusoid tracking example. State-space formulation of SSLMS also makes it possible to use it in state estimation in control systems. This is illustrated by an example in Section 8.

The derivation of SSLMS with adaptive memory (SSLMSWAM) is then presented. The idea is to iteratively tune step-size parameter by stochastic gradient method so as to minimize the mean square value of the prediction error. To highlight the usefulness of SSLMSWAM, the application of tracking of noisy Van der Pol oscillations is discussed. The performance of SSLMSWAM is also compared with that of SSLMS, SSRLS, SSRLSWAM, RLS and LMS.

A detailed discussion of computational complexities of different algorithms under discussion concludes the paper.

2. State-space model

Consider a process $y[k] \in R^m$ that is available for measurement. Assume that the underlying process generator is an unforced linear time varying discrete-time system. Standard form of such a system is given as follows [9]:

$$\begin{aligned} x[k+1] &= A[k]x[k], \\ y[k] &= C[k]x[k] + v[k], \end{aligned} \tag{1}$$

where $x[k] \in R^n$ is the state vector at time k, the elements of which are called state variables, which may also be referred to as the process states. We assume that the maximum number of outputs of the system is less than or equal to the states, i.e., $m \leqslant n$. This is a logical assumption as a system with $m > n$ can be simplified to the one with $m \leqslant n$ without the loss of any information about the states [12]. The system matrix $A[k]$ and the output matrix $C[k]$ may be stochastic or deterministic depending on the nature of the problem. $C[k]$ is assumed to be full rank, which is a reasonable assumption. To see this, consider the case when $C[k]$ is deterministic and rank deficient. We can reduce the number of outputs to make a new full rank output matrix, without the loss of any information about the states. On the other hand, a stochastic $C[k]$ is full rank by virtue of unavoidable presence of white noise. The pair $(A[k], C[k])$ is assumed to be l-step observable [9]. Observation noise is represented by $v[k]$. It is customary to assume $v[k]$ to be a zero-mean white process, although these assumptions do not affect the derivations and development done in this paper. Such assumptions on the observation noise are important for analyzing the performance of the filters presented here. The state-transition matrix for the system (1) is given by

$$A[k, j] = \begin{cases} A[k-1]A[k-2]\ldots A[j], & k > j, \\ I, & k = j. \end{cases} \tag{2}$$

The system matrix $A[k]$ is assumed to be invertible for all k which results in the following properties [12]:

$$\begin{aligned} &A^{-1}[k, j] = A[j, k], \quad \forall j, k, \\ &A[k, i] = A[k, j]A[j, i], \quad i \leqslant j \leqslant k, \\ &A[k+1, k] = A[k]. \end{aligned} \tag{3}$$

The absence of 'process noise' or any other deterministic inputs poses a question about the usefulness of (1). However, this framework is general enough to address adaptive filtering applications as would be shown later in this paper. Moreover, this concept is a familiar one in the context of exosystems used in servomechanism problems [13]. The reference signals (to be tracked) and disturbances (to be rejected) are modeled using systems similar to (1). In all these cases, the system matrix $A[k]$ is neutrally stable. For a time-invariant system matrix $A = A[k]$, neutral stability implies that all of the eigenvalues of A are strictly on the unit circle. An extension of this philosophy to nonlinear systems can be found in the regulation problem, where once again the references/disturbances are modeled by neutrally stable exosystems [14]. Neutral stability of $A[k]$ rules out existence of unstable or exponentially stable states and hence the concern about applicability of (1) to physical systems. Having said that, the theory developed in this paper however, does not make any assumptions about the neutral stability of $A[k]$. This is only a matter of usefulness of models like (1) in practical applications.

3. State estimator

Suppose that the observations $y[k]$ start appearing at time $k = 1$. The initial state vector is $x[0] = x_0$ and is not known. The observability assumption allows us to design a state estimator. The idea is to generate the estimated state vector $\hat{x}[k]$ making use of the observations $y[1], y[2], \ldots, y[k]$.

The system equation (1) enables us to compute the predicted state estimate at time k (using observations up to time $k-1$) as follows:

$$\bar{x}[k] = A[k-1]\hat{x}[k-1]. \tag{4}$$

The prediction error can now be defined as

$$\varepsilon[k] = y[k] - \bar{y}[k] \tag{5}$$

with

$$\bar{y}[k] = C[k]\bar{x}[k] \tag{6}$$

as the predicted output. The prediction error is also referred to as innovations in the realm of Kalman filtering [1]. We can also define the estimation error as

$$e[k] = y[k] - \hat{y}[k], \tag{7}$$

where $\hat{y}[k] = C[k]\hat{x}[k]$ is the estimated output. One of the well-known estimator forms is [12]

$$\hat{x}[k] = \bar{x}[k] + K[k]\varepsilon[k], \tag{8}$$

where $K[k]$ is the observer gain, which is to be determined by different methods presented in this paper.

4. State-space least mean square (SSLMS)

In this section we derive a generalized version of LMS viz SSLMS which incorporates model dynamics as given in (1). The discussion begins with relating the prediction error (5) and estimation error (7) as follows:

$$e[k] = \varepsilon[k] - C[k]\delta[k], \tag{9}$$

where $\delta[k]$ is defined as

$$\delta[k] = \hat{x}[k] - \bar{x}[k]. \tag{10}$$

The assumption that $C[k]$ is full rank makes it possible to choose $\hat{x}[k]$ such that

$$e[k] = 0, \tag{11}$$

which gives

$$\varepsilon[k] = C[k]\delta[k]. \tag{12}$$

If $m < n$ then there are infinitely many choices of $\hat{x}[k]$ that satisfy (11). We resort to minimum norm solution of (12), which minimizes $\delta[k]$ in (10) subject to the constraint $e[k] = 0$ [1]. We get

$$\delta[k] = C^T[k]\left(C[k]C^T[k]\right)^{-1}\varepsilon[k]. \tag{13}$$

From (10) and (13)

$$\hat{x}[k] = \bar{x}[k] + C^T[k]\left(C[k]C^T[k]\right)^{-1}\varepsilon[k]. \tag{14}$$

Comparing (14) with (8), the observer gain $K[k]$ according to the method of minimum norm solution comes out to be

$$K[k] = C^T[k]\left(C[k]C^T[k]\right)^{-1}. \tag{15}$$

As apparent from its form, the gain in (15) has a limited scope. In order for a state estimator to be valid, the map from output of the system (which is the input of the estimator) to state estimates should be controllable. Alternately stating, $(A[k-1] - K[k]C[k]A[k-1], K[k])$ pair should be controllable. The choice of gain as given in (15) does not guarantee that this requirement will always be satisfied. Furthermore, a designer prefers to have a control of the rate of convergence, which is done through step-size parameter in the standard LMS [1]. In view of these considerations, we introduce a step-size parameter μ and matrix G to arrive at a more useful expression as follows:

$$K[k] = \mu G C^T[k]\left(C[k]C^T[k]\right)^{-1}. \tag{16}$$

The matrix G is chosen so as to have a valid estimator, i.e., controllable pair $(A[k-1] - K[k]C[k]A[k-1], K[k])$, whereas rate of convergence is controlled through μ. In certain cases like sinusoidal model [2,8], the controllability condition exists without the matrix G in (16). On the other hand, constant velocity and constant acceleration models [2,8] do not fulfill this requirement and hence a designer has to choose a G for the estimator to be valid. The actual choice of this matrix depends on the nature of the problem. One simple approach is illustrated in Section 8.2, where the first column of G consists of non-zero entries. The rest are all zeroes.

Finally, for the cases where invertibility of $C[k]C^T[k]$ cannot be ensured, we may use a small number γ, which modifies (16) into

$$K[k] = \mu G C^T[k]\left(\gamma I + C[k]C^T[k]\right)^{-1}. \tag{17}$$

This arrangement allows us to handle problems where $C[k]$ may become rank deficient for a short interval. $\gamma = 0$ is used in situations where $C[k]$ is guaranteed to be full rank. Defining

$$\gamma I + C[k]C^T[k] \tag{18}$$

as a normalization factor, the algorithm comprising (4)–(6), (8), (17) is termed as state-space normalized LMS (SSNLMS). It is apparent from (17) that an $m \times m$ matrix is required to be inverted. A simplification in this algorithm results by removing the normalization factor, which reduces the observer gain to

$$K[k] = \mu G C^T[k]. \tag{19}$$

The algorithm (4)–(6), (8), (19) is accordingly called state-space LMS (SSLMS). An analogy of SSLMS with the standard LMS would become clear in Section 5.

5. Analogy with the standard LMS

In order to show analogy of SSLMS with standard LMS, let

$$\boldsymbol{u}[k] = \left[u[k], u[k-1], \ldots, u[k-n+1]\right]^T \tag{20}$$

be input vector for a system with tap weights

$$\boldsymbol{w}_0[k] = \left[w_{01}[k], w_{02}[k], \ldots, w_{0n}[k]\right]^T. \tag{21}$$

The output of this filter is corrupted by additive observation noise $v[k]$. The signal $d[k]$ is called the desired signal. The tap-weights of an adaptive transversal filter

$$\hat{\boldsymbol{w}}[k] = \left[\hat{w}_1[k], \hat{w}_2[k], \ldots, \hat{w}_n[k]\right]^T \tag{22}$$

can be thought of as an estimate of the unknown tap-weights $\boldsymbol{w}_0$. The output of this filter is $s[k]$. The problem is to adjust the estimated tap-weights so as to minimize the error $e[k]$ in some sense. We choose the minimum norm least square error as our optimization criterion. If the problem is overdetermined then the filter becomes the standard RLS. If it is underdetermined then we get the normalized LMS. Finally dropping the normalization factor gives us the standard LMS. The relevant details can be found in [1]. It is not difficult to see that if the following condition holds

$$\begin{aligned} &m = 1, \\ &A = I, \\ &C[k] = \boldsymbol{u}[k], \\ &\bar{x}[k] = \hat{\boldsymbol{w}}[k], \\ &\varepsilon[k] = e[k], \\ &y[k] = d[k], \\ &\bar{y}[k] = s[k], \end{aligned} \tag{23}$$

then the filter developed in this paper turns into standard LMS. Due to this reason, we assert that SSLMS and its variants are in fact generalization of the conventional LMS filters.

6. An application of SSLMS

The problem of tracking a noisy sinusoid/chirp is of historical significance and has received considerable attention in the literature [7,15,16]. The problem arises naturally in the context of an interfering signal of known frequency. Widrow et al. considered the problem of cancellation of 60 Hz interference in electrocardiography, using LMS [17]. In this section we give a brief account of how SSLMS can be used in such cases.

The phase and amplitude of the interfering signal are assumed to be unknown. Apparently, a priori knowledge of frequency simplifies the problem into a trivial one. However in the case of standard RLS and LMS, the designer has no direct way to incorporate this information. By virtue of state-space formulation of SSLMS, this information can be incorporated in a straightforward manner. For a sinusoid represented in discrete time as

$$y[k] = \sigma_s \cos(\omega_0 kT + \phi) + v[kT], \tag{24}$$

the system matrices are given by [8]

$$\begin{aligned} &A = \begin{bmatrix} \cos(\omega_0 T) & \sin(\omega_0 T) \\ -\sin(\omega_0 T) & \cos(\omega_0 T) \end{bmatrix}, \\ &C = [1\ 0], \end{aligned} \tag{25}$$

where σ_s^2 = signal power, ω_0 = signal frequency, ϕ = phase of signal, T = sampling time, v = observation noise; SSLMS may then be used to track the interfering signal.

7. State-space least mean square with adaptive memory (SSLMSWAM)

When the model of the underlying environment is completely/partially unknown, a presumed model results in a model mismatch almost obviously. A partial compensation for this mismatch may be achieved if we adaptively tune the step-size parameter, so as to minimize a cost function. SSLMS with adaptive memory builds upon the framework of SSLMS to achieve iterative tuning of step-size parameter. Our objective is to tune the step-size parameter μ so as to minimize the following cost function

$$J[k] = \frac{1}{2} E\big[\varepsilon^T[k]\varepsilon[k]\big], \tag{26}$$

where $E[\cdot]$ is the expectation operator and $\varepsilon[k]$ is the prediction error defined in (5). Differentiating $J[k]$ with respect to μ gives

$$\nabla_\mu[k] = \frac{\partial J[k]}{\partial \mu} = E\left[\frac{\partial \varepsilon^T[k]}{\partial \mu}\varepsilon[k]\right], \tag{27}$$

where $\frac{\partial \varepsilon^T[k]}{\partial \mu}$ is a row vector. Defining

$$\psi[k] = \frac{\partial \hat{x}[k]}{\partial \mu}, \tag{28}$$

we get

$$\frac{\partial \varepsilon[k]}{\partial \mu} = \frac{\partial}{\partial \mu}\big[y[k] - C[k]A[k-1]\hat{x}[k-1]\big] = -C[k]A[k-1]\psi[k-1], \tag{29}$$

which implies that

$$\nabla_\mu[k] = -E\big[\psi^T[k-1]A^T[k-1]C^T[k]\varepsilon[k]\big]. \tag{30}$$

Differentiating (8) with respect to μ and using (4), (19), (28) and (29) we get

$$\psi[k] = \big(A[k-1] - K[k]C[k]A[k-1]\big)\psi[k-1] + GC^T[k]\varepsilon[k]. \tag{31}$$

Now we are in a position to formulate SSLMS with adaptive memory. The stochastic gradient method that updates $\mu[k]$, which in turn is a function of time, is [1]

$$\mu[k] = \mu[k-1] - \alpha\nabla_\mu[k], \tag{32}$$

where α is a small positive learning rate parameter. Based on (30), an instantaneous estimate for the scalar gradient $\nabla_\mu[k]$ can be taken as

$$\hat{\nabla}_\mu[k] = -\psi^T[k-1]A^T[k-1]C^T[k]\varepsilon[k], \tag{33}$$

which modifies (32) into

$$\mu[k] = \big[\mu[k-1] + \alpha\,\psi^T[k-1]A^T[k-1]C^T[k]\varepsilon[k]\big]_{\mu_-}^{\mu_+}. \tag{34}$$

For this algorithm to be meaningful we require $\mu > 0$. The bracket followed by μ_- and μ_+ in equation indicates truncation that restricts step-size parameter to $[\mu_-, \mu_+]$. The lower limit is generally set close to zero, whereas the upper limit depends on the nature of the problem. Its value is determined through experimentation. A simplification in the algorithm results if we ignore the normalization factor. The normalization factor improves convergence properties of the SSLMS by normalizing the step-size parameter μ. However, when we have a complete scheme to adapt step-size parameter μ, then normalization is somewhat unnecessary. Replacing μ by $\mu[k]$ in (16), the complete SSLMS algorithm with adaptive memory is summarized below in (35). First three equations constitute SSLMS, whereas the last two address the update of step-size parameter μ.

$$\begin{aligned}
&K[k] = \mu[k]GC^T[k],\\
&\varepsilon[k] = y[k] - C[k]A[k-1]\hat{x}[k-1],\\
&\hat{x}[k] = A[k-1]\hat{x}[k-1] + K[k]\varepsilon[k],\\
&\mu[k] = \big[\mu[k-1] + \alpha\psi^T[k-1]A^T[k-1]C^T[k]\varepsilon[k]\big]_{\mu_-}^{\mu_+},\\
&\psi[k] = \big(A[k-1] - K[k]C[k]A[k-1]\big)\psi[k-1] + GC^T[k]\varepsilon[k].
\end{aligned} \tag{35}$$

8. An application of SSLMSWAM

8.1. Tracking Van der Pol oscillations

We illustrate the performance of SSLMSWAM by tracking Van der Pol oscillations [18]. An electronic circuit with vacuum tubes, that acts like a resistor for high current through it, can be modeled by a Van der Pol oscillator. The circuit exhibits negative resistance behavior for low values of current through it. Small oscillations are amplified whereas larger ones are suppressed. Equations for Van der Pol oscillator are [18]

$$\begin{aligned}
&\dot{x}_1 = x_2,\\
&\dot{x}_2 = -x_1 + \varepsilon\big(1 - x_1^2\big)x_2.
\end{aligned} \tag{36}$$

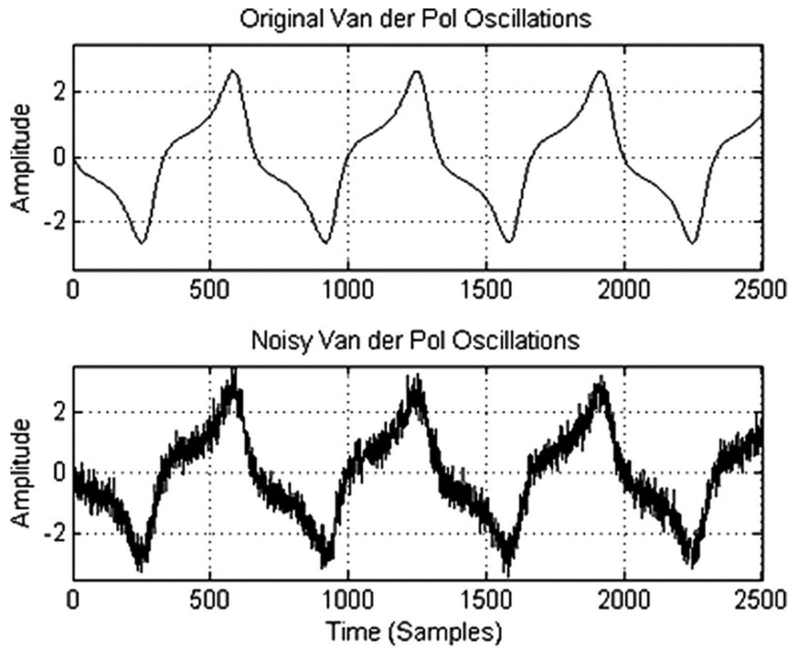

Fig. 1. Van der Pol oscillations.

Working with the assumption that actual (Van der Pol) signal model is completely unknown, the constant acceleration model [19] given below is a good choice.

$$A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \qquad C = [1\ 0\ 0]. \tag{37}$$

8.2. Computer experiment

SSLMSWAM is selected as the estimator. We observe the signal x_2 in (36) in discrete domain after sampling with sampling time $T = 0.01$ s. Zero mean white Gaussian noise with variance 0.001 corrupts the observations. The learning rate parameter is chosen to be $\alpha = 0.01$. The system is started with zero initial conditions except $\mu[0] = 0.1$. Matrix G is chosen to be

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0 & 0 \\ 0.3 & 0 & 0 \end{bmatrix}. \tag{38}$$

The noisy Van der Pol oscillations to be tracked are shown in Fig. 1. The simulation results as illustrated in Fig. 2 demonstrate the performance of the net algorithm. In order to draw a comparison with other adaptive algorithms, tracking results for SSLMS with step-size parameter $\mu = 0.1$, SSRLS [8] with forgetting factor $\lambda = 0.97$, SSRLSWAM [9] with initial forgetting factor $\lambda[0] = 0.93$, 3-tap standard LMS filter with $\mu = 0.001$ and 3-tap standard RLS filter with forgetting factor $\lambda = 0.99$ are also given. Adaptation of step-size parameter for SSLMSWAM is given in Fig. 3. Results for the case of uniformly distributed observation noise are given in Fig. 4, where LMS operates with $\mu = 0.1$ and RLS operates with $\lambda = 0.8$.

8.3. Comments

The constant acceleration model (37) used in these simulations provides an approximation by fitting 3rd-order polynomial on various segments of Van der Pol oscillations. For the segments containing lower frequency components, the fit is good and SSLMS and SSRLS perform well. On the other hand for the segments having higher frequency

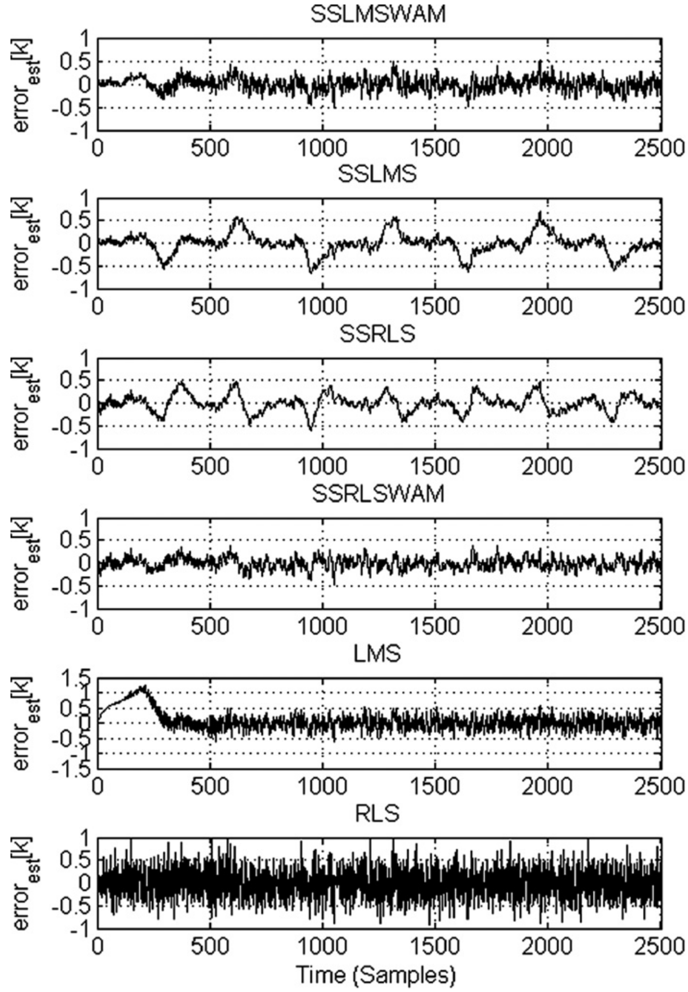

Fig. 2. Van der Pol oscillations estimation error for SSLMSWAM, SSLMS, SSRLS, SSRLSWAM, LMS and RLS (Gaussian observation noise).

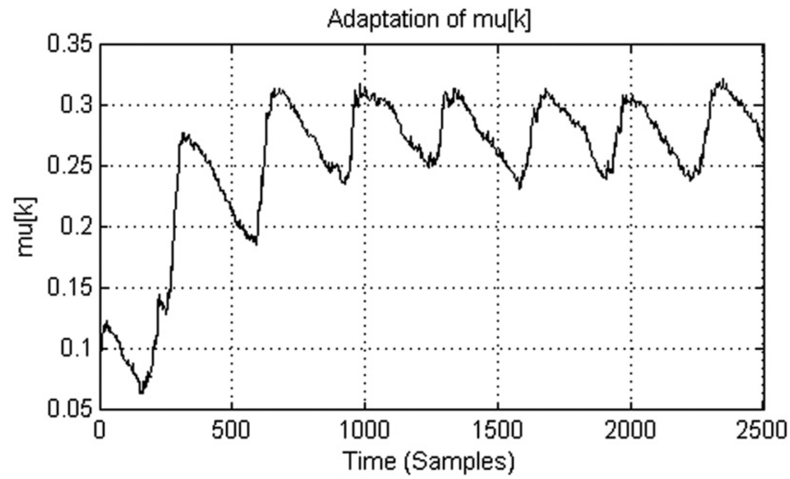

Fig. 3. Adaptation of step-size parameter for SSLMSWAM—Van der Pol oscillations tracking.

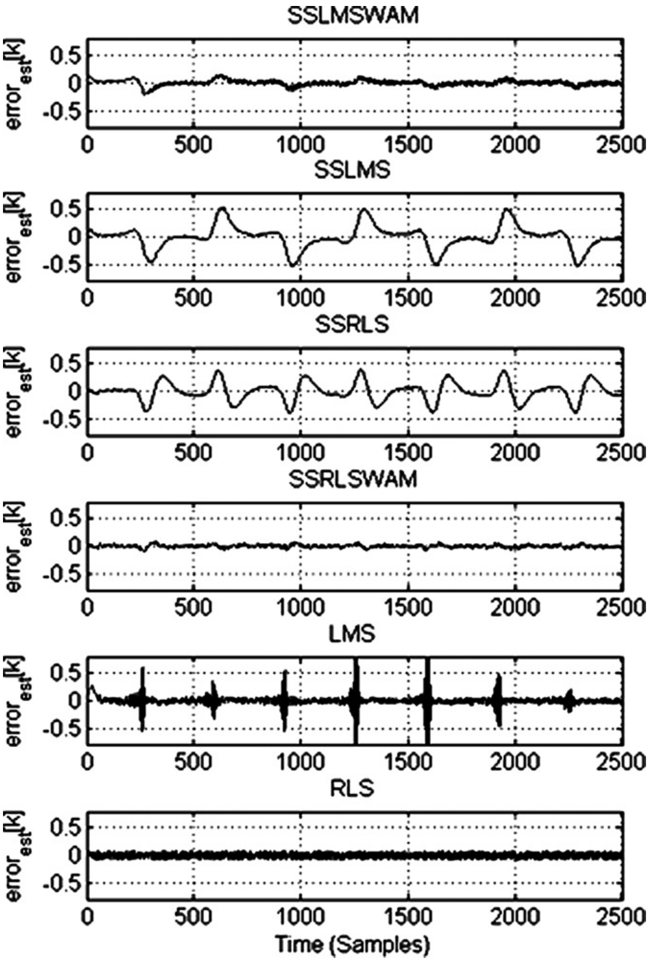

Fig. 4. Van der Pol oscillations estimation error for SSLMSWAM, SSLMS, SSRLS, SSRLSWAM, LMS and RLS (uniformly distributed observation noise).

components the fit is not good enough. SSLMSWAM and SSRLSWAM compensate partially for this *misfit* by tuning step-size parameter and forgetting factor, respectively. Consequently, the estimation error for SSLMSWAM and SSRLSWAM is less than SSLMS and SSRLS for these segments. This effect is easier to notice in case of uniformly distributed observation noise (Fig. 4).

In Fig. 4, periodic sharp rise and fall in estimation error for LMS is noticeable whereas performance of RLS for the selected value of forgetting factor is satisfactory. It is possible to improve the performance of SSLMS, LMS, SSRLS and RLS by changing values of step-size parameter and forgetting factor, respectively. However, we assume that the designer does not know beforehand the most suitable values of these parameters.

9. Computational complexity

Computational complexity of an algorithm is usually of significant importance particularly in real-time applications. In this section, we discuss this aspect of SSLMS.

The complexities of Eqs. (4)–(6), (8) are given in Table 1. These equations are common to all the variants of SSLMS. Table 2 furnishes complexity of SSLMS that comprises Eqs. (4)–(6), (8) and (19). It can be seen that complexity of SSLMS is $O(n^2)$.

Table 3 provides complexity of SSNLMS that comprises Eqs. (4)–(6), (8) and (17). Although the complexity of SSNLMS is also $O(n^2)$, it is computationally more intensive of the two due to its requirement of mth-order matrix

Table 1
Complexity of Eqs. (4)–(6), (8)

Eq. #	Operation	Multiplications	Additions	Divisions
(4)	$\bar{x}[k]_{n\times 1} = A_{n\times n}\hat{x}[k-1]_{n\times 1}$	n^2	$n^2 - n$	–
(5)	$\varepsilon[k]_{m\times 1} = y[k]_{m\times 1} - \bar{y}[k]_{m\times 1}$	–	m	–
(6)	$\bar{y}[k]_{m\times 1} = C_{m\times n}\bar{x}[k]_{n\times 1}$	mn	$mn - m$	–
(8)	$\hat{x}[k]_{n\times 1} = \bar{x}[k]_{n\times 1} + K_{n\times m}\varepsilon[k]_{m\times 1}$	mn	mn	–
	Total	$n^2 + 2mn$	$n^2 + 2mn - n$	–

Table 2
Complexity of SSLMS (Eqs. (4)–(6), (8), (19))

Eq. #	Operation	Multiplications	Additions	Divisions
	Total from Table 1	$n^2 + 2mn$	$n^2 + 2mn - n$	–
(19)[a]	$K[k]_{n\times m} = \mu_{1\times 1}G_{n\times n}C^T[k]_{n\times m}$	mn^2	$mn^2 - mn$	–
	Total	$n^2 + mn^2 + 2mn$	$n^2 + mn^2 + mn - n$	–

[a] $\mu_{1\times 1}G_{n\times n}$ computed offline.

Table 3
Complexity of SSNLMS (Eqs. (4)–(6), (8), (17))

Eq. #	Operation	Multiplications	Additions	Divisions
	Total from Table 1	$n^2 + 2mn$	$n^2 + 2mn - n$	–
a	$P_{n\times m} = \mu_{1\times 1}G_{n\times n}C^T[k]_{n\times m}$	mn^2	$mn^2 - mn$	–
	$Q_{m\times m} = C[k]_{m\times n}C^T[k]_{n\times m}$	m^2n	$m^2n - m^2$	–
b	$R_{m\times m} = \gamma_{1\times 1}I_{m\times m} + Q_{m\times m}$	–	m^2	–
	Gauss–Jordan Inversion [20] $S_{m\times m} = R^{-1}_{m\times m}$	$\frac{m^3}{3} + \frac{m^2}{2} + \frac{m}{6}$	$\frac{m^3}{3} + \frac{m^2}{2} + \frac{m}{6}$	$\frac{m^2}{2} + \frac{m}{2}$
(17)	$K[k]_{n\times m} = P_{n\times m}S_{m\times m}$	m^2n	$m^2n - mn$	–
	Total	$n^2 + \frac{m^3}{3} + 2m^2n + mn^2 + \frac{m^2}{2} + 2mn + \frac{m}{6}$	$n^2 + \frac{m^3}{3} + 2m^2n + mn^2 + \frac{m^2}{2} - n + \frac{m}{6}$	$\frac{m^2}{2} + \frac{m}{2}$

[a] $\mu_{1\times 1}G_{n\times n}$ computed offline.
[b] $\gamma_{1\times 1}I_{m\times m}$ computed offline.

Table 4
Complexity of SSLMSWAM (Eq. (36))

Eq. #	Operation	Multiplications	Additions	Divisions
	$H_{n\times m} = G_{n\times n}C^T[k]_{n\times m}$	mn^2	$mn^2 - mn$	–
	$K[k]_{n\times m} = \mu[k]_{1\times 1}H_{n\times m}$	mn	–	–
	Total from Table 1	$n^2 + 2mn$	$n^2 + 2mn - n$	–
	$P_{n\times 1} = C^T[k]_{n\times m}\varepsilon[k]_{m\times 1}$	mn	$mn - n$	–
	$Q_{n\times 1} = A^T[k-1]_{n\times n}P_{n\times 1}$	n^2	$n^2 - n$	–
	$R_{1\times 1} = \alpha_{1\times 1}\psi^T[k-1]_{1\times n}Q_{n\times 1}$	$n + 1$	$n - 1$	–
	$\mu[k]_{1\times 1} = \mu[k-1]_{1\times 1} + R_{1\times 1}$	–	1	–
	$J_{n\times 1} = H_{n\times m}\varepsilon[k]_{m\times 1}$	mn	$mn - n$	–
	$M_{m\times n} = C[k]_{m\times n}A[k-1]_{n\times n}$	mn^2	$mn^2 - mn$	–
	$N_{n\times n} = K[k]_{n\times m}M_{m\times n}$	mn^2	$mn^2 - n^2$	–
	$O_{n\times n} = A[k-1]_{n\times n} - N_{n\times n}$	–	n^2	–
	$S_{n\times 1} = O_{n\times n}\psi[k-1]_{n\times 1}$	n^2	$n^2 - n$	–
(36)	$\psi[k]_{n\times 1} = S_{n\times 1} + J_{n\times 1}$	–	n	–
	Total	$3n^2 + 3mn^2 + 5mn + n + 1$	$3n^2 + 3mn^2 + 2mn - 3n$	–

Table 5
Summary of complexities of algorithms for $m = 1$

Algorithm	Multiplications	Additions	Divisions
SSLMS (Table 2)	$2n^2 + 2n$	$2n^2$	–
SSNLMS (Table 3)	$2n^2 + 4n + 1$	$2n^2 + n + 1$	1
SSLMSWAM (Table 4)	$6n^2 + 6n + 1$	$6n^2 - n$	–
SSRLS [9]	$2n^3 + 3n^2 + 4n$	$2n^3 + n^2 + n$	1
SSRLSWAM [9]	$8n^3 + \frac{19n^2}{2} + \frac{13n}{2} + 1$	$8n^3 - n^2 + 4n$	3
RLS [1]	$4n^2 + O(n)$		1
LMS [1]	$3n + 2$		–
NLMS [1]	$5n + 2$		1

inversion. SSLMSWAM (Eq. (35)) is $O(n^2)$ but computationally more intensive (Table 4) than SSNLMS and SSLMS, due to the adaptation of step-size parameter.

We summarize the complexity of algorithms presented in this paper and compare them with SSRLS, SSRLSWAM, RLS, LMS and NLMS in Table 5, for $m = 1$ [1,8,9].

10. Conclusion

We have seen that the development of SSLMS has overcome some of the limitations of the standard LMS. This new filter exhibits superior tracking performance. The state-space formulation of SSLMS enables design of adaptive filters to be carried out in an intuitive and direct manner. The concept of adaptive memory enhances the capability of SSLMS when it comes to dealing with uncertain time-varying systems. Stochastic gradient approach that has been used to adapt step-size parameter successfully achieves the desired results. Some applications presented give insight into the potential use of the new algorithm. Stability and convergence analysis of SSLMS is left for future considerations.

Acknowledgments

The authors are thankful to their colleague Dr. Khalid Munawar for reviewing the paper and providing valuable comments for its improvement. The authors also acknowledge Higher Education Commission (HEC), Pakistan for the support received from them under Ph.D. scholarship grant for Ph.D. work of the author Muhammad Salman.

References

[1] S. Haykin, Adaptive Filter Theory, fourth ed., Prentice Hall, Delhi, 2002.
[2] Y. Bar-Shalom, X.-R. Li, T. Kirubarajan, Estimation with Applications to Tracking and Navigation, Wiley & Sons, New York, 2001.
[3] A. Benveniste, Design of adaptive algorithms for the tracking of time-varying systems, Int. J. Adaptive Control Signal Process. 1 (1987) 3–29.
[4] E. Eleftheriou, D.D. Falconer, Tracking properties and steady-state performance of RLS adaptive filter algorithms, IEEE Trans. Acoust. Speech Signal Process. ASSP-34 (1986) 1097–1110.
[5] E. Ewada, Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels, IEEE Trans. Signal Process. 42 (1994) 2937–2944.
[6] A.H. Sayed, T. Kailath, A state-space approach to adaptive RLS filtering, IEEE Signal Process. Mag. 11 (3) (1994) 18–60.
[7] S. Haykin, A.H. Sayed, J. Zeidler, P. Yee, P. Wei, Adaptive tracking of linear time-variant systems by extended RLS algorithms, IEEE Trans. Signal Process. 45 (6) (May 1997) 1118–1128.
[8] M.B. Malik, State-space recursive least-squares: Parts I & II, Signal Process. J. 84 (2004) 1709–1728.
[9] M.B. Malik, State-space recursive least squares with adaptive memory, Signal Process. J. 86 (2006) 1365–1374.
[10] M.B. Malik, R. Bhatti, Tracking of linear time-varying systems using state-space least mean square, in: IEEE International Symposium on Communications and Information Technologies, 2004, pp. 582–585.
[11] M.B. Malik, M. Salman, State-space least mean square with adaptive memory, in: IEEE Region 10 Conference, 2005.
[12] W.J. Rugh, Linear System Theory, second ed., Prentice Hall, Upper Saddle River, NJ, 1996.
[13] C.A. Desoer, Y.T. Wang, Linear time-invariant robust servomechanism problem: A self-contained exposition, in: C.T. Leondes (Ed.), Control and Dynamic Systems, vol. 16, 1980, pp. 81–129.
[14] A. Isidori, Nonlinear Control Systems, third ed., Springer-Verlag, London, 2001.
[15] M.B. Malik, M. Salman, Comparative tracking performance of SSRLS and SSLMS algorithms for chirped signal recovery, in: IEEE International Multi Topic Conference, 2005.

[16] M. Salman, M.B. Malik, Adaptive recovery of a noisy chirp: Performance of the SSLMS algorithm, in: IEEE International Symposium on Signal Processing and Its Applications, 2005, pp. 763–766.
[17] B. Widrow, J.R. Glover Jr., J.M. McCool, J. Caunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong Jr., R.C. Goodlin, Adaptive noise canceling: Principles and applications, Proc. IEEE 63 (12) (1975) 1692–1716.
[18] H.K. Khalil, Nonlinear Systems, third ed., Prentice Hall, Upper Saddle River, NJ, 2002.
[19] M.B. Malik, State-space recursive least-squares, Ph.D. dissertation, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan, 2004.
[20] E. Kreyszig, Advanced Engineering Mathematics, eighth ed., Wiley & Sons, Singapore, 1999.

Mohammad Bilal Malik received his B.S. degree in electrical engineering from College of Electrical and Mechanical Engineering (E&ME), Rawalpindi, Pakistan, in 1991. He received his M.S. degree in electrical engineering from Michigan State University (MSU), Michigan, USA, in 2001. In 2004, he received his Ph.D. degree in electrical engineering from National University of Sciences and Technology (NUST), Rawalpindi, Pakistan. He has been teaching at College of E&ME, National University of Sciences and Technology (NUST) since 1991. His research focuses on signal processing and adaptive filtering for communications and control systems.

Muhammad Salman received his B.S. degree in electrical engineering from College of Electrical and Mechanical Engineering (E&ME), Rawalpindi, Pakistan, in 1993. He received his M.S. degree in electrical engineering from College of Electrical and Mechanical Engineering (E&ME), National University of Sciences and Technology (NUST), Rawalpindi, Pakistan, in 2005. He is currently pursuing his Ph.D. in electrical engineering at College of E&ME, National University of Sciences and Technology (NUST), Rawalpindi, Pakistan under a scholarship from Higher Education Commission (HEC), Pakistan. His research focuses on signal processing and adaptive filtering for communications and control systems.