INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2004; **18**:505–521 (DOI: 10.1002/acs.811)

A novel variable-length sliding window blockwise least-squares algorithm for on-line estimation of time-varying parameters

Jin Jiang*,† and Youmin Zhang‡

Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ont., Canada N6A 5B9

SUMMARY

Motivated by the advances in computer technology and the fact that the batch/block least-squares (LS) produces more accurate parameter estimates than its recursive counterparts, several important issues associated with the block LS have been re-examined in the framework of on-line identification of systems with abrupt/gradual change parameters in this paper. It is no surprise that the standard block LS performs unsatisfactorily in such a situation. To overcome this deficiency, a novel variable-length sliding window-based LS algorithm, known as variable-length sliding window blockwise least squares, is developed. The algorithm consists of a change detection scheme and a data window with adjustable length. The window length adjustment is triggered by the change detection scheme. Whenever a change in system parameters is detected, the window is shortened to discount 'old' data and place more weight on the latest measurements. Several strategies for window length adjustment have been considered. The performance of the proposed algorithm has been evaluated through numerical studies. In comparison with the recursive least squares (RLS) with forgetting factors, superior results have been obtained consistently for the proposed algorithm. Robustness analysis of the algorithm to measurement noise have also been carried out. The significance of the work reported herein is that this algorithm offers a viable alternative to traditional RLS for on-line parameter estimation by trading off the computational complexity of block LS for improved performance over RLS, because the computational complexity becomes less and less an issue with the rapid advance in computer technologies. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: system identification; parameter estimation; time-varying systems; recursive least squares (RLS); blockwise least squares (BLS); variable-length sliding window blockwise least squares (VLSWBLS); abrupt and gradual parameter changes; change detection

1. INTRODUCTION

Since Karl Gauss proposed the technique of least squares in 1795 for predicting the motion of planets and comets using telescopic measurements, least squares (LS) and its many variants have been widely used [1–8]. The popular parameter estimation algorithms, such as least

*Correspondence to: Jin Jiang, Department of Electrical and Computer Engineering, The University of Western Ontario, London, Ontario, Canada N6A 5B9.
†E-mail: jjiang@uwo.ca
‡E-mail: ymzhang@uwo.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada

Published online 17 June 2004

Received 11 July 2003
Accepted 4 March 2004

squares, generalized/extended least squares, instrumental variables, maximum likelihood and extended Kalman filter (among others), have been reviewed in several survey papers and books, for example, References [4–6, 8, 9]. In general, LS-based algorithms can be formulated either in *block*/*batch* or *recursive*/*on-line* forms.

The classical formulation of least squares is in *block*/*batch* form, meaning that all measurements are collected first and then processed in one batch. Such a formulation is often referred to as block least squares. For on-line applications, the entire estimation process needs to be repeated every time when a new measurement becomes available. This format of implementation is referred to as blockwise least squares (BLS) in this paper. The main drawback of a BLS is its computational burden since the computational complexity is in the order of $O(\max(P^3, kP^2))$ which grows continuously with the number of measurements k and the number of parameters being estimated P. However, with the recent advance in computer technologies, the computational complexity of BLS becomes less and less an issue.

To increase the computational efficiency, a recursive variant, known as RLS, was developed in 1950. The computational complexity of RLS is only in the order of $O(P^2)$. Even though RLS algorithms are well suited for on-line estimation and possess many similar statistical properties as the block LS, they also suffer from the following drawbacks: (1) the gain of the algorithm converges to zero, which leads to the loss of the tracking ability for time-varying parameters; (2) there are also some numerical issues due to round-off errors as a result of finite-word length; and (3) the estimation results are dependent on the choice of the initial conditions. Even though BLS and RLS are equivalent analytically, it has been shown that for linear time-invariant systems, the performance of BLS is almost always superior to that of RLS.

To overcome the above drawbacks, considerable efforts have been directed towards the development of modified versions of RLS for on-line identification of time-varying systems, for example, in References [5, 8, 10–13]. The key to track time-varying parameters is to use certain type of forgetting techniques to discard 'out-of-date' data. Owing to its computational efficiency and convenience to incorporate forgetting factors in RLS-type algorithms, almost all existing techniques in dealing with parameter estimation of time-varying systems fall into the category of *recursive* LS. These techniques can be classified into three main categories: (1) variable forgetting factors [2, 5, 6, 10–17]; (2) covariance matrix modification [2, 18–20]; and (3) sliding window [8, 21–23]. However, according to the best of our knowledge, there are few results in the literature in dealing with estimation of abruptly and gradually changed parameters in the framework of BLS.

In view of the inherent advantages of BLS, it is highly desirable to develop new techniques that can improve the tracking ability of BLS for time-varying systems while retaining its other salient features. Clearly, certain effective techniques to discard 'out-of-date' data need to be developed. For this purpose, a new variable-length sliding window blockwise least squares (VLSWBLS) scheme is developed in this paper to provide satisfactory parameter tracking in the transient period and high level of estimation accuracy at the steady state for systems with both abruptly and gradually changed parameters.

The proposed scheme consists of a change detection algorithm and a variable-length sliding window. Once parameter changes are detected, the window length is shortened automatically. The performance of the developed algorithms has been evaluated in comparison with that of RLS-type algorithms with forgetting factors. The superior performance has been obtained in all cases studied. The robustness of the algorithm to measurement noise has also been examined through sensitivity analysis at various signal-to-noise ratio (SNR) levels.

The paper is organized as follows: the problem of estimating parameters in systems with abrupt and gradual parameter changes is formulated in Section 2, together with a brief review on block and recursive least squares. A VLSWBLS algorithm is developed in Section 3 with the help of a change detection scheme. The results of the illustrative example are presented in Section 4 followed by the conclusions in Section 5.

2. PROBLEM FORMULATION AND BRIEF BACKGROUND REVIEW

2.1. Estimation of time-varying parameters

To formulate the parameter estimation problem, let us consider a discrete dynamic system described by the following autoregressive moving average with auxiliary input (ARMAX) model:

$$\begin{aligned} y(k) &= \boldsymbol{\phi}^{\mathrm{T}}(k-1)\boldsymbol{\theta} \\ z(k) &= y(k) + v(k) \\ \boldsymbol{\phi}^{\mathrm{T}}(k-1) &= [-z(k-1), \ldots, -z(k-n), u(k-1), \ldots, u(k-m)] \\ \boldsymbol{\theta}^{\mathrm{T}} &= [a_1, \ldots, a_n, b_1, \ldots, b_m] \end{aligned} \tag{1}$$

where $u(k)$ and $y(k)$ are the system input and output, respectively. $z(k)$ is the measured output and $v(k)$ is a zero-mean white Gaussian sequence which counts for the measurement noise and modelling errors. $\boldsymbol{\phi}^{\mathrm{T}}(k-1)$ and $\boldsymbol{\theta}$ are the information vector and the unknown parameter vector, respectively. The number of parameters to be estimated is denoted as $P = n + m$. The individual parameters in $\boldsymbol{\theta}$ can either be constant or subject to abrupt or gradual changes. For clarity, let us represent the nominal parameter vector prior to any change by $\boldsymbol{\theta}_0$.

An abrupt change at an unknown time k_A can be described as

$$\begin{aligned} \boldsymbol{\theta}(k) &= \boldsymbol{\theta}_0, \quad k < k_A \\ \boldsymbol{\theta}(k) &= \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}, \quad k \geqslant k_A, \ \|\Delta\boldsymbol{\theta}\| \neq 0 \end{aligned}$$

where $\Delta\boldsymbol{\theta} = [\Delta a_1, \ldots, \Delta a_n, \Delta b_1, \ldots, \Delta b_m]^{\mathrm{T}}$ is assumed to be a constant vector, representing the magnitude of the abrupt change.

Gradual change at an unknown time k_{G_i} can be represented by

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}_0, \quad k < \min(k_{G_i}, \ i = 1, \ldots, P)$$

$$\boldsymbol{\theta}(k) = \boldsymbol{\theta}_0 + \begin{bmatrix} r_1 \cdot 1(k - k_{G_1}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & r_P \cdot 1(k - k_{G_P}) \end{bmatrix} \boldsymbol{\theta}(k-1), \quad k \geqslant \min(k_{G_i}, \ i = 1, \ldots, P)$$

where r_i, $i = 1, \ldots, P$, represents the rate of change in each individual parameter, and $1(k - k_{G_i})$, $i = 1, \ldots, P$, is a switching function at K_{G_i}. Even though other forms of gradual parameter changes are also possible, the paper mainly considers the above linear form of change. However, the developed algorithm does not rely on any particular form of parameter changes.

2.2. Block and recursive least squares

Traditionally, to estimate the parameter vector, $\boldsymbol{\theta}$, in a least square sense, using all available measurements up to the current instant, $j = 1, 2, \ldots, k$, one needs to minimize the following cost function:

$$J(\boldsymbol{\theta}, k) = \sum_{j=1}^{k} [z(j) - \boldsymbol{\phi}^{\mathrm{T}}(j-1)\boldsymbol{\theta}]^2 \tag{2}$$

The solution is given by

$$\hat{\boldsymbol{\theta}}(k) = \left[\sum_{j=1}^{k} \boldsymbol{\phi}(j-1)\boldsymbol{\phi}^{\mathrm{T}}(j-1)\right]^{-1} \left[\sum_{j=1}^{k} \boldsymbol{\phi}^{\mathrm{T}}(j-1)z(j)\right] \tag{3}$$

or in a compact form as

$$\hat{\boldsymbol{\theta}}(k) = [\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k]^{-1}[\boldsymbol{\Phi}_k^{\mathrm{T}}\mathbf{z}_k] \tag{4}$$

where $\hat{\boldsymbol{\theta}}(k)$ is known as least-squares estimate of $\boldsymbol{\theta}$, and

$$\mathbf{z}_k = [z(1) \quad z(2) \quad \cdots \quad z(k)]^{\mathrm{T}}_{k\times 1}$$

$$\boldsymbol{\Phi}_k = \begin{bmatrix} \boldsymbol{\phi}^{\mathrm{T}}(0) \\ \boldsymbol{\phi}^{\mathrm{T}}(1) \\ \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(k-1) \end{bmatrix} = \begin{bmatrix} -z(0) & \cdots & -z(1-n) & u(0) & \cdots & u(1-m) \\ -z(1) & \cdots & -z(2-n) & u(1) & \cdots & u(2-m) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -z(k-1) & \cdots & -z(k-n) & u(k-1) & \cdots & u(k-m) \end{bmatrix}_{k\times P}$$

If this formulation were used in on-line applications, one would have to evaluate (4) every time when a new measurement becomes available. To circumvent this problem, a recursive version of the above scheme, known as RLS, has been developed.

The basic idea of an RLS algorithm is, rather than repeatedly solving (4), to compute the parameter estimate $\hat{\boldsymbol{\theta}}(k)$ by adding a correction term to the previous parameter estimate $\hat{\boldsymbol{\theta}}(k-1)$ whenever new information becomes available.

Since RLS algorithm and its many variants can be found in textbooks and papers (see e.g. References [2, 4, 6, 8]), further elaboration on RLS-type algorithms will be omitted herein in the interest of space.

3. A VARIABLE-LENGTH SLIDING WINDOW BLOCKWISE LEAST SQUARES

3.1. Overview of the proposed algorithm

The objective of this study is to develop a parameter estimation algorithm suitable for time-varying systems. The basic idea in achieving this objective is to use a sliding window blockwise least squares with an automatically adjustable window length. The window length adjustment is triggered by a parameter change detection algorithm. The key elements in this scheme include: (1) an effective strategy to automatically adjust the length of the sliding window to achieve the best performance in both transient and steady-state intervals; (2) a fast and reliable parameter

change detection scheme; and (3) a sound mechanism to distinguish a gradual change from an abrupt one.

3.2. Sliding window blockwise least squares

It is well known that one of the effective ways to deal with time-varying parameters is to use a finite-length sliding window. However, if a fixed length window is used, it is generally difficult to achieve both good parameter tracking during the transient and high degree of accuracy at the steady-state conditions.

The unique idea proposed in this paper is to make the window length adjustable in response to the potential change in the system parameters.

To further improve the tracking capability of the algorithm, an exponential weighting technique (or other forgetting techniques) can still be used within the sliding window. Therefore, a modified cost function can be written as follows:

$$J_L(\boldsymbol{\theta}, k) = \sum_{j=k-L+1}^{k} \lambda^{k-j}[z(j) - \boldsymbol{\phi}^{\mathrm{T}}(j-1)\boldsymbol{\theta}]^2 \tag{5}$$

where L denotes the length of the sliding window. λ $(0<\lambda<1)$ is the forgetting factor within the window.

The solution to this sliding window blockwise least squares (SWBLS) algorithm can be obtained as follows:

$$\hat{\boldsymbol{\theta}}_L(k) = \left[\sum_{j=k-L+1}^{k} \lambda^{k-j}\boldsymbol{\phi}(j-1)\boldsymbol{\phi}^{\mathrm{T}}(j-1)\right]^{-1}\left[\sum_{j=k-L+1}^{k} \lambda^{k-j}\boldsymbol{\phi}^{\mathrm{T}}(j-1)z(j)\right] \tag{6}$$

or, in a compact form as

$$\hat{\boldsymbol{\theta}}_L(k) = [(\boldsymbol{\Phi}_{k-L+1}^{k})^{\mathrm{T}}\Lambda_{k-L+1}^{k}\boldsymbol{\Phi}_{k-L+1}^{k}]^{-1}[(\boldsymbol{\Phi}_{k-L+1}^{k})^{\mathrm{T}}\Lambda_{k-L+1}^{k}\mathbf{z}_{k-L+1}^{k}] \tag{7}$$

where $\boldsymbol{\Phi}_{k-L+1}^{k}$ is similarly defined as in (4), and Λ_{k-L+1}^{k} is an $L \times L$ diagonal matrix with the diagonal elements being function of forgetting factors, i.e. $\lambda^{L-1}, \lambda^{L-2}, \ldots, \lambda^{0}$. For $0<\lambda<1$, the corresponding SWBLS is referred to as sliding exponential window blockwise least squares (SEWBLS). For $\lambda = 1$, the algorithm is known as sliding rectangular window blockwise least squares (SRWBLS).

The performance of the above algorithm depends obviously on the window length. For time-invariant systems, the longer the window length, the higher the estimation accuracy. However, for systems with parameter changes, a longer window would lead to slower responses to these changes. Therefore, in this case, the window length should be adjusted accordingly so that the measurements prior to the change can be discarded effectively.

3.3. Strategies for window length adjustment

To improve the parameter tracking ability, a new strategy for window length adjustment has been developed in this paper. The essence of the proposed approach lies in its ability to quickly shrink the window length, should a change in system parameters be detected. Using a shorter window length, it allows the algorithm to track the parameter changes more responsively. In the case of an abrupt change, from the onset of the window shrinking, the window length will expand progressively and return to its original length to maintain the steady-state performance.

In case of a gradual parameter change, the window will be maintained at a desirable shortened length depending on the rate of parameter changes until the end of the change is detected. Such a fast shrinking and gradual expanding window makes the proposed approach suitable for estimating both abruptly and gradually changing parameters.

To illustrate this concept clearly, let us consider the following snap-shots in Figures 1 and 2, respectively, for a typical scenario in which an abrupt or a gradual parameter change has occurred.

In Figure 1, if there is no detected parameter change, a sliding window of length $L(k) = L_S$ is used, where L_S is a desirable window length at the steady state. This situation is shown in Figures 1(a) and 1(b). Suppose that some parameters of the system have undergone changes, and such a change is detected at k_D to be an abrupt one. At this point, the window length is reduced drastically and the new sliding window starts from $k_D - P + 1$. This case is illustrated in Figure 1(c). From this point onwards, the window length expands step-by-step in the forward direction until it reaches the full length L_S, as shown in Figure 1(d). Thereafter, the original window size resumes as shown in Figure 1(e).

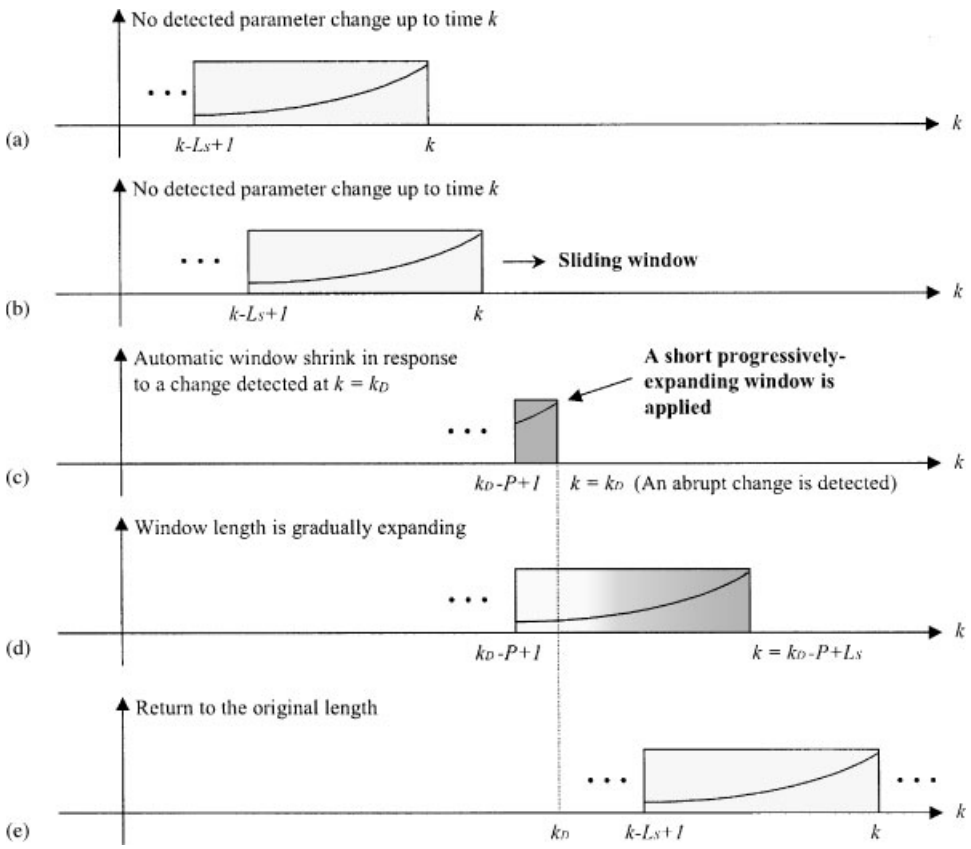

Figure 1. Window length adjustment in the presence of an abrupt change.

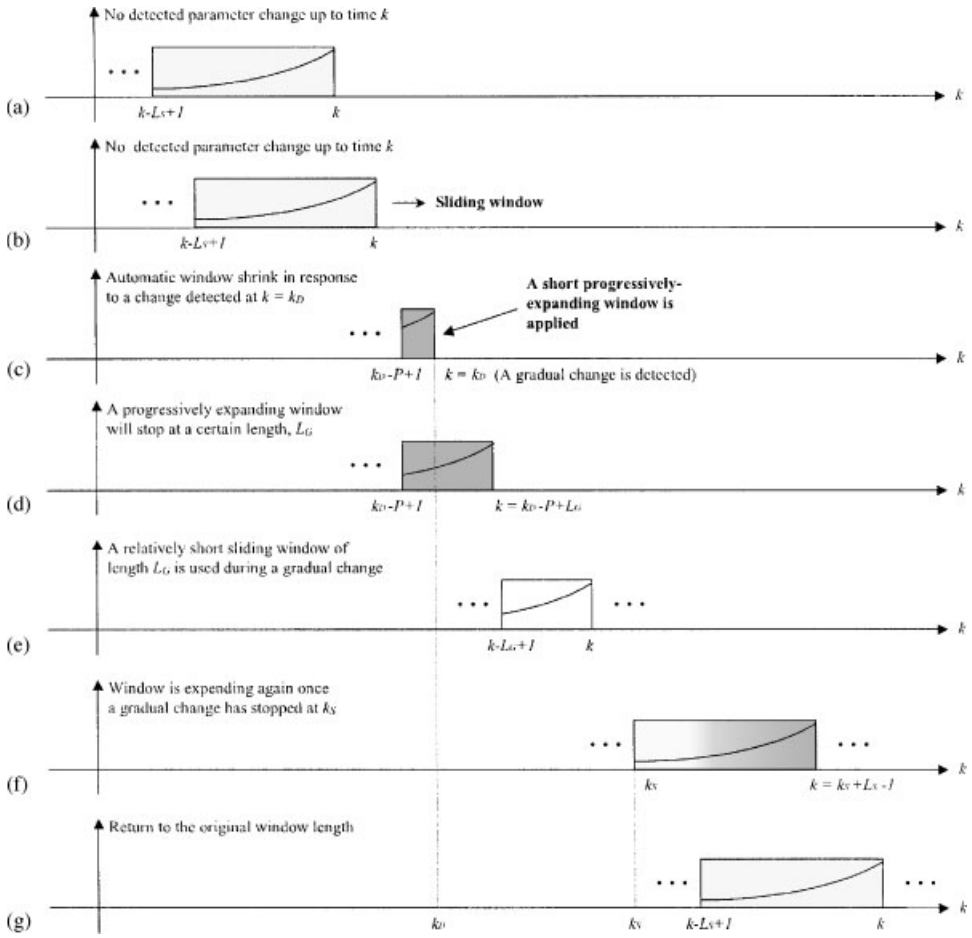

Figure 2. Window length adjustment in the presence of a gradual change.

For gradual parameter changes, the situation is slightly different: (1) Once a change is detected and classified to be a gradual one, the window will stop expanding at a certain length, denoted as L_G, as shown in Figures 2(c)–2(e). The length of this shortened window will depend on the estimated rate of the parameter change. The higher the change rate, the shorter the window becomes; (2) once the gradual change ends, the window will return gradually to its full length L_S. This situation is shown in Figure 2(f), and the normal operation resumes in Figure 2(g).

For LS parameter estimation, the minimum window size should not be less than P. At the instant of window shrinking, $\Phi_{k-L(k)+1}^{k}$ may still contain some data collected prior to the parameter change. One way to further enhance the transient performance is to use an exponential weighting within the shrunk window. It can also improve the performance of parameter tracking in case of gradual changes.

3.4. Change detection mechanisms

To automatically initiate and complete the window length adjustment, a change detection scheme must be used. The change detection scheme should also be able to distinguish an abrupt change from a gradual one. Even though there are several different ways that one can detect parameter changes in a system [24–32], for gradual changes, one not only has to detect the onset of a change, but also to estimate the rate and the termination of such a change. In view of this, two types of change detection have been used in this paper to signal the beginning and the end of a change. Such a change detection scheme relies on a combined detection index using an output prediction error and the parameter estimation error.

3.4.1. Change detection based on output prediction error. Consider the following output prediction error within the sliding window of length $L(k)$:

$$\mathbf{e}(k) = \mathbf{z}^k_{k-L(k)+1} - \hat{\mathbf{z}}^k_{k-L(k)+1} = \mathbf{z}^k_{k-L(k)+1} - \Phi^k_{k-L(k)+1}\hat{\boldsymbol{\theta}}_L(k) \tag{8}$$

where $\mathbf{z}^k_{k-L(k)+1} = [z(k-L(k)+1)\ z(k-L(k))\ \cdots\ z(k)]^{\mathrm{T}}$ and $\hat{\mathbf{z}}^k_{k-L(k)+1}$ is its estimated counterpart. Note that the window length L is now a function of time.

Occurrence of a parameter change will be declared if the following accumulated detection index:

$$d_e(k) = \frac{1}{M_e}\sum_{i=k-M_e+1}^{k} \mathbf{e}^{\mathrm{T}}(i)\mathbf{e}(i) \tag{9}$$

exceeds a pre-set threshold $\bar{\rho}_e$

$$d_e(k)\begin{cases} > \bar{\rho}_e & \text{change has started} \\ \leqslant \underline{\rho}_e & \text{change has ended} \end{cases} \tag{10}$$

where $\bar{\rho}_e$ is the threshold for detecting the occurrence of a change and $\underline{\rho}_e$ represents the threshold for detecting the end of a change. M_e is the number of data points used in calculating the above detection index. The selection of M_e and $\bar{\rho}_e$ represents certain degree of trade-off between the probability of false alarm and the probability of missed detection. If the change is detected at time k_D, this time instant will be referred to as the *detection time*.

3.4.2. Change detection based on parameter estimation error. Similarly, one can also define the following averaged parameter estimate in a sliding window of lengh M_θ:

$$\bar{\boldsymbol{\theta}}_{M_\theta}(k) = \frac{1}{M_\theta}\sum_{j=k-M_\theta+1}^{k} \hat{\boldsymbol{\theta}}_L(j) \tag{11}$$

A decision for parameter change (or lack of it) can also be made if the detection index

$$d_{\boldsymbol{\theta}}(k) = \gamma|\|\bar{\boldsymbol{\theta}}_{M_\theta}(k)\|_2 - \|\bar{\boldsymbol{\theta}}_{M_\theta}(k-1)\|_2| + (1-\gamma) \times \sum_{i=k-M_\theta+1}^{k} |\|\bar{\boldsymbol{\theta}}_{M_\theta}(i-1)\|_2 - \|\bar{\boldsymbol{\theta}}_{M_\theta}(i-2)\|_2| \tag{12}$$

exceeds the following pre-set threshold

$$d_{\boldsymbol{\theta}}(k)\begin{cases} > \bar{\rho}_{\boldsymbol{\theta}} & \text{change has started} \\ \leqslant \underline{\rho}_{\boldsymbol{\theta}} & \text{change has ended} \end{cases} \tag{13}$$

where $0<\gamma<1$ is the weighting factor. It determines the relative weights placed on the instantaneous and the accumulative parameter estimation errors. $M_{\boldsymbol{\theta}}$ is the number of data points used for calculating the above detection index.

Since a parameter change in a system usually manifests more pronounced at the output prediction error than at the parameter estimation error, the detection index $d_e(k)$ is more suitable for detecting the onset of the parameter changes, and the detection index $d_{\boldsymbol{\theta}}(k)$ is more sensitive to the detection of change termination. The time at which a change terminates is known as *stopping time* and denoted as k_S.

Remark 1

The change detection algorithm in Equations (8)–(10) is belong to a class called cumulative sum (CUSUM) algorithm, which is essentially a moving average scheme. Even though such an algorithm is intuitive to understand and simple to implement, however, it has been shown [26] that $d_e(k)$ is not a sufficient statistics for parameter changes considered in Section 2.1. Theoretically, one always runs the risk that some particular combination of parameter changes may not manifest themselves to the changes in $d_e(k)$. To deal with such a drawback, more sophisticated change detection schemes have been developed. The interested readers are encouraged to consult [26] for a number of such algorithms. It would be out of the scope of this paper to go into details. Instead, in the current work, we resort such a problem by incorporating the other parameter estimation error-based change detection scheme as defined in Equation (12). Between these two change detection schemes, our experience has indicated that almost all types of parameter changes can be detected. In comparison with other change detection schemes, the determination of changes in a particular set of parameters is done via the subsequent parameter estimation scheme, rather than using multiple hypothesis tests on the residuals.

3.5. Distinguish an abrupt from a gradual change

In the change detection scheme, it is important to differentiate abrupt change from gradual one so that the most appropriate window adjustment strategies can be employed. The basic approach is to calculate the rate of change of the detection variable $d_{\boldsymbol{\theta}}(k)$ using linear regressions. If the rate of change is slower than a certain level, the change is classified as gradual, otherwise, abrupt. Such a concept is demonstrated graphically in Figure 3.

Suppose at k_D, $d_{\boldsymbol{\theta}}(k)$ exceeds a detection threshold $\bar{\rho}_\theta$, a linear regression can be applied to the detection variable collected from $k_D - M_{\boldsymbol{\theta}} + 1$ up to k, to determine α and β in the following expression:

$$d_{\boldsymbol{\theta}}(k) = \alpha + \beta k \tag{14}$$

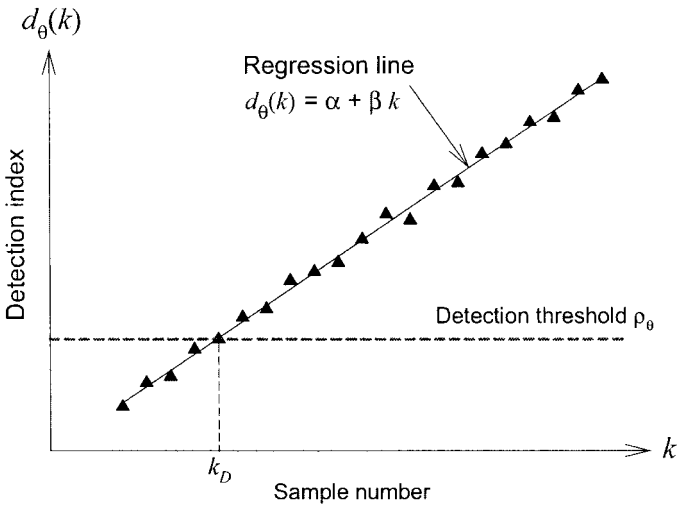

Figure 3. Determination of gradual and abrupt changes.

where the parameter β is known as the *slope*, it indicates the rate of change in the detection variable $d_{\boldsymbol{\theta}}(k)$. A hypothesis test on β against a threshold τ is given as follows:

$$\beta = \begin{cases} > \tau & \text{Abrupt change} \\ \leqslant \tau & \text{Gradual change} \end{cases} \tag{15}$$

The slope β can be estimated as

$$\begin{aligned} \beta(k) &= \frac{\sum_{j=k_D-M_{\boldsymbol{\theta}}+1}^{k} [j - \bar{\mu}(k)][d_{\boldsymbol{\theta}}(j) - \bar{\mu}_{d_{\boldsymbol{\theta}}}(k)]}{\sum_{j=k_D-M_{\boldsymbol{\theta}}+1}^{k} [j - \bar{\mu}(k)]^2} \\ &= \frac{\sum_{j=k_D-M_{\boldsymbol{\theta}}+1}^{k} [j - \bar{\mu}(k)] d_{\boldsymbol{\theta}}(j)}{\sum_{j=k_D-M_{\boldsymbol{\theta}}+1}^{k} [j - \bar{\mu}(k)]^2} \end{aligned} \tag{16}$$

where

$$\bar{\mu}(k) = \frac{1}{k - k_D + M_{\boldsymbol{\theta}}} \sum_{j=k_D-M_{\boldsymbol{\theta}}+1}^{k} j \tag{17}$$

3.6. *Algorithmic summary of the proposed scheme*

To summarize the proposed scheme, a flowchart is shown in Figure 4.

The window lengths at different stages of the algorithm are summarized as follows:

$L(k) = L_S$	before a change is detected;
$L(k) = P$	once a change is detected at $k = k_D$;
$L(k) = L(k-1) + 1$	window expending after the change is detected;
until $L(k) = L_G$	for a gradual parameter change;
or $L(k) = L_S$	for an abrupt parameter change and at the steady-state.

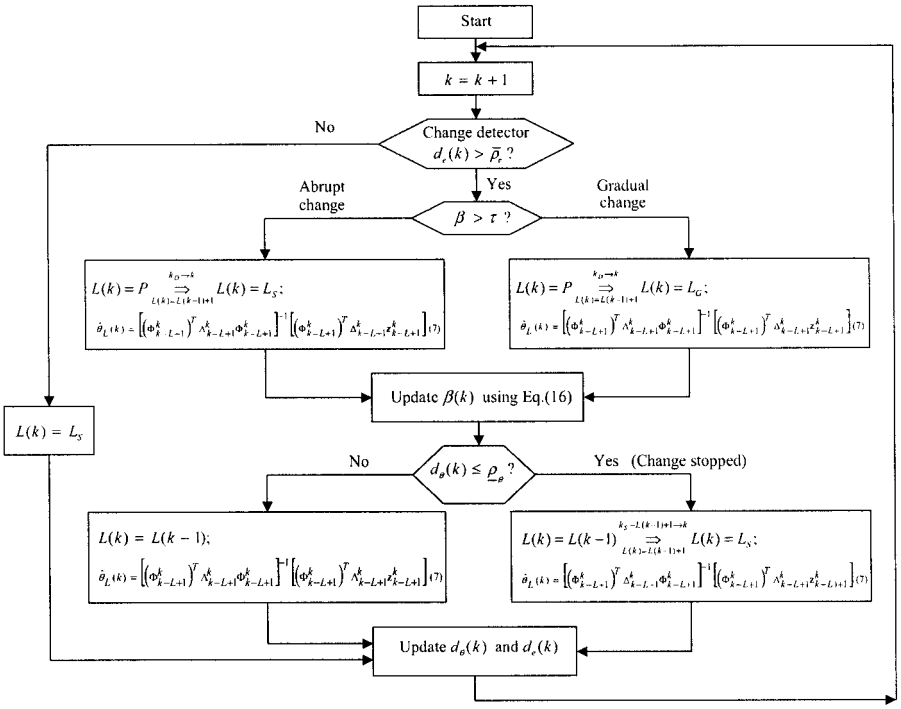

Figure 4. Algorithmic flowchart of the proposed scheme.

4. NUMERICAL ILLUSTRATIONS

To evaluate the performance of the proposed algorithm, a system with different parameter change patterns has been used for simulation studies. A complete test scenario consists of a gradual parameter change followed by an abrupt one.

4.1. System used in the study

The second-order system used in the simulation is adopted from Reference [6]. Necessary modifications have been made to simulate the parameter changes. The system is described by the following difference equation:

$$\begin{aligned} y(k) &= -a_1 y(k-1) - a_2 y(k-2) + b_1 u(k-1) + b_2 u(k-2) \\ z(k) &= y(k) + v(k) \end{aligned} \tag{18}$$

where $y(k)$ is the output of the system, and $u(k)$ the input. $v(k)$ is a zero-mean white Gaussian noise sequence. The input $u(k)$ is also a zero-mean white Gaussian sequence with unit variance $\sigma_u^2 = 1.0$, assumed to be independent of $v(k)$. The simulated parameter changes are summarized in Table I with graphical illustrations in Figure 6.

Table I. The simulated parameter changes.

k	$0<k<200$	$200\leqslant k<500$	$500\leqslant k<650$	$650\leqslant k<1000$	$1000\leqslant k\leqslant 1500$
a_1	-1.5	$10^{-3}\times(k-200)$	$-1.2\times10^{-3}\times(k-500)$	-1.5	-1.2
a_2	0.7	$-6.7\times10^{-4}\times(k-200)$	$8.0\times10^{-4}\times(k-500)$	0.7	0.5
b_1	1.0	$-10^{-3}\times(k-200)$	$1.2\times10^{-3}\times(k-500)$	1.0	0.7
b_2	0.5	$-6.7\times10^{-4}\times(k-200)$	$8.0\times10^{-4}\times(k-500)$	0.5	0.3

4.2. Indices for performance evaluation

In order to evaluate the performance of different schemes, the following averaged performance measure is used for each individual run:

$$e_{\boldsymbol{\theta}}(k)=\frac{1}{N_{\text{run}}}\sum_{j=1}^{N_{\text{run}}}\|\boldsymbol{\theta}(k)-\hat{\boldsymbol{\theta}}_L^j(k)\|_2 \tag{19}$$

where $N_{\text{run}}=100$ stands for the total number of independent runs in the Monte Carlo simulation.

In addition, the average error (AE) and the standard deviation (STD) of the $e_{\boldsymbol{\theta}}(k)$, $\forall k\in\{1,N\}$ are defined as follows to be overall performance indicators, i.e.

$$\begin{aligned}\text{AE}:&\quad \bar{e}_{\boldsymbol{\theta}}=\frac{1}{N}\sum_{k=1}^{N}e_{\boldsymbol{\theta}}(k)\\ \text{STD}:&\quad \sigma_{\boldsymbol{\theta}}=\sqrt{\frac{1}{N-1}\sum_{k=1}^{N}[e_{\boldsymbol{\theta}}(k)-\bar{e}_{\boldsymbol{\theta}}]^2}\end{aligned} \tag{20}$$

where $N=1500$ is the total number of sample points chosen for the study.

4.3. Simulation results and analysis

4.3.1. Comparison between the proposed and the RLS-type algorithms. To compare the estimation accuracy and the parameter tracking capability, the performance measure, $e_{\boldsymbol{\theta}}(k)$, is shown in Figure 5 for the proposed variable-length sliding rectangular window blockwise least squares (VLSRWBLS) and the exponentially-weighted recursive least squares (EWRLS). In the EWRLS, a constant forgetting factor of $\lambda=0.95$ is used. It can be seen that superior tracking performance has been achieved by the VLSRWBLS algorithm. The estimate converges quickly to the new parameter values with a much smaller steady-state estimation error. Compared with the estimation error prior to the change, almost the same level of the estimation accuracy has been obtained in the post-changed interval. Even though the EWRLS can still track the changed parameters, the rate of convergence is significantly lower and the steady-state estimation error is also relatively higher.

To demonstrate the time history of the parameter estimate, Figure 6 shows the trajectories of the estimated parameters using these algorithms. It can be seen clearly that the proposed approach yields excellent results in both pre- and post-change intervals. Excellent tracking performance for both gradual and abrupt changes has also been achieved for all four parameters. The proposed approach outperforms the EWRLS for abrupt parameter changes.

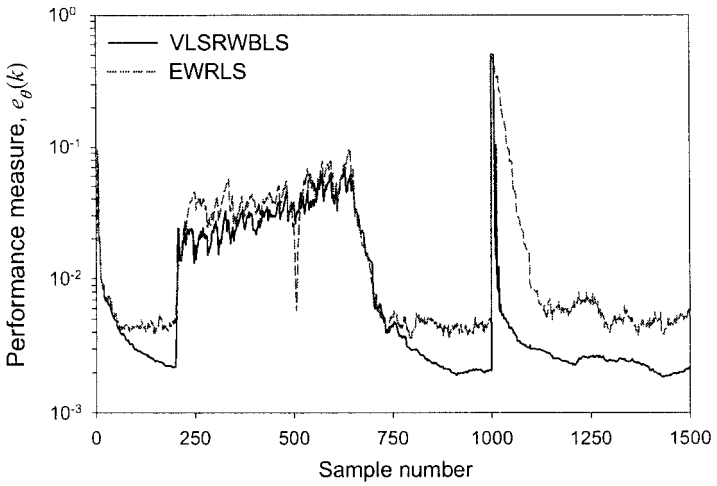

Figure 5. Comparison between the proposed and the EWRLS algorithms.

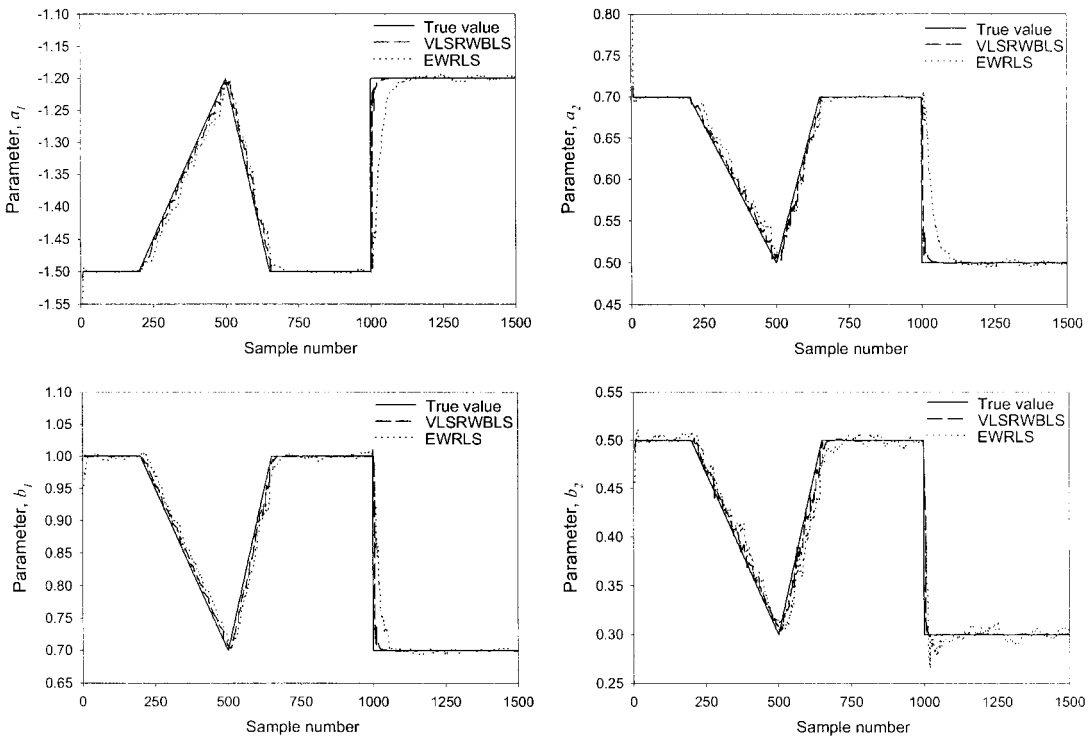

Figure 6. Parameter estimates via proposed and EWRLS algorithms.

To demonstrate the window length adjustment strategies, the time history of the corresponding window length in one entire simulation run is shown in Figure 7. For easy reference, the pattern of parameter change in a_1 is also present. The automatic window

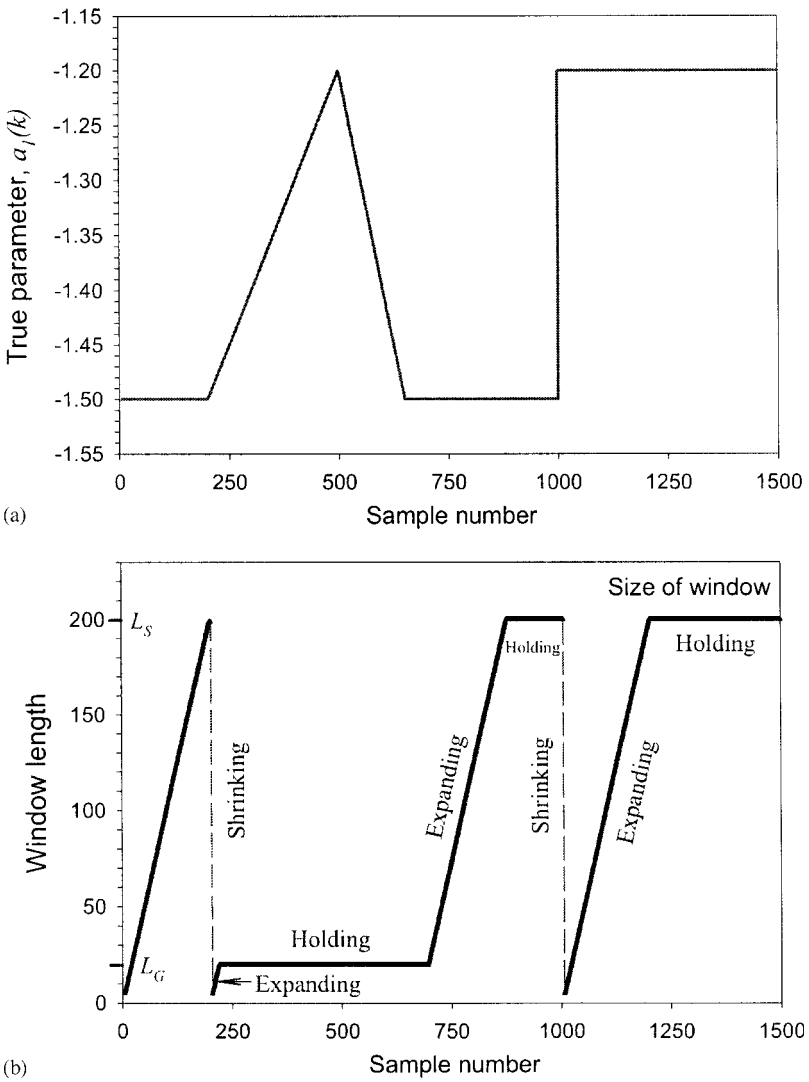

Figure 7. History of window length adjustment in response to different parameter changes: (a) history of a parameter change; and (b) history of corresponding window length adjustment.

shrinking, expanding and holding processes in response to abrupt and gradual changes and at the steady state are all illustrated. In the case of the gradual parameter change during the period $200 \leqslant k < 650$, once the change is detected and determined to be a gradual one at $k_D = 204$, the full window of length $L_S = 200$ is shrunk rapidly to $L = P = 4$, and then expanded gradually up to a length of $L_G = 20$ and maintained at the value. Once the change stops at $k = 650$ which is detected at $k_S = 696$, this window starts to expand again stepwise to the full length of L_S. When an abrupt change at $k = 1000$ is detected at $k_D = 1003$, the window is shrunk again to the minimal length of $L = 4$ and progressively expanded back to the full length $L = 200$.

Remark 2

Even though this example only illustrates a case of linearly changed parameter values, the proposed scheme can be adopted to more complex parameter change patterns by making the window length as a function of the estimated rate of parameter change. However, caution needs to be taken to avoid too frequent window adjustments to ensure a smooth parameter estimate at the duration of such gradual changes. Furthermore, in addition to the step-by-step expanding, other more sophisticated window expanding strategies may also worth exploring.

4.3.2. Performance for fixed- and variable-length windows. To evaluate the effect of different window lengths on the performance of the estimation schemes, fixed window lengths at $L = 50, 100, 200$ and the one based on the proposed variable-length sliding window algorithm have been experimented. Results are shown in Figures 8(a) and 8(b), with the rectangular- and exponential-weighted windows, respectively.

For a sliding window with a fixed-length, it can be seen that the longer the window, the higher the estimation accuracy at the steady state. However, a longer window can lead to larger delay in obtaining satisfactory post-change parameter estimation. It is clearly observable that, by using a fixed window length, one could not achieve satisfactory performance in both the steady state and the transient periods. Using the proposed variable-length sliding window strategy, significantly improved performance in both the transient and the steady-state periods has been obtained.

4.3.3. Robustness to measurement noise. To evaluate the robustness of the proposed approach at different noise levels, the system in (18) has been simulated at different SNR levels. The results are summarized in Table II. The input to the system is a zero-mean white Ganssian sequence with a fixed unit variance $\sigma_u^2 = 1.0$. while the variance of the noise term $v(k)$ is adjusted to different levels. It can be observed from this table that for different level of noise intensities, the proposed approach is robust for a set of pre-set detection thresholds ($\bar{\rho}_e^0 = 10^{-2}; \rho_e^0 = 10^{-5}; \rho_{\boldsymbol{\theta}}^0 = 10^{-5} : \tau = 18\%$, with the choice of window length $M_e = M_{\boldsymbol{\theta}} = P$). A change in SNR by a factor of 500 results in changes in AE and STD by a factor of less than 50.

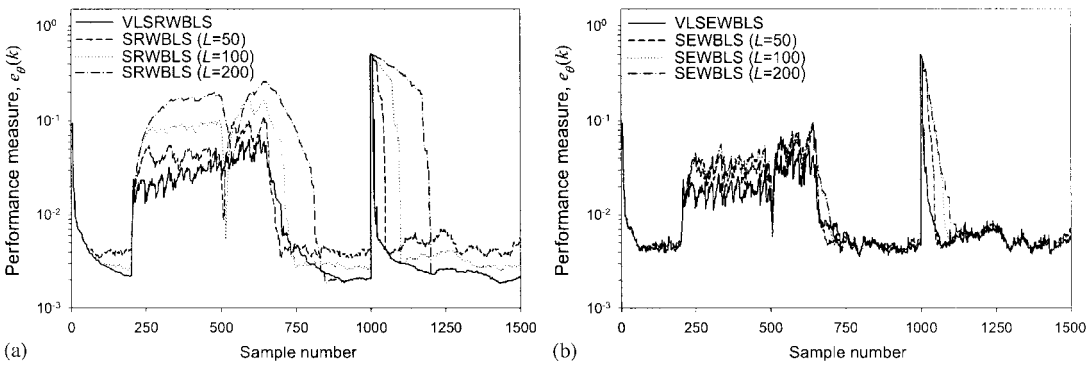

Figure 8. Comparison among different window lengths and shapes: (a) rectangular window; and (b) exponential window.

Table II. Comparison of estimation accuracy at different SNR levels.

	AE		STD	
Noise levels σ_v^2	Rectangular	Exponential	Rectangular	Exponential
0.001	1.1200×10^{-2}	1.2410×10^{-2}	3.8604×10^{-2}	3.7016×10^{-2}
0.01	1.6680×10^{-2}	1.4331×10^{-2}	3.9595×10^{-2}	3.3773×10^{-2}
0.1	8.0220×10^{-2}	8.0318×10^{-2}	3.9898×10^{-2}	3.7791×10^{-2}
0.2	1.5739×10^{-1}	1.7096×10^{-1}	9.5572×10^{-2}	9.1864×10^{-2}
0.4	4.2495×10^{-1}	4.6603×10^{-1}	1.9095×10^{-1}	2.1745×10^{-1}
0.5	5.2857×10^{-1}	5.5860×10^{-1}	2.0286×10^{-1}	1.9988×10^{-1}

5. CONCLUSIONS

With ever advancement in computer technology and inexpensive computing power at our disposal, we could not help but to re-think the necessity of some of the existing computationally more efficient but numerically less perfect algorithms. In this paper, the standard least squares (LS) parameter estimation approach has been revisited in view of its better accuracy and simpler implementation in comparison with recursive least squares (RLS) type of algorithms. To address the weak parameter tracking ability of blockwise least squares (BLS), an effective sliding window blockwise least squares approach with an automatically adjustable window length has been proposed to extend the LS parameter estimation to systems with abrupt and gradual parameter changes. A general scheme for the parameter estimation in the presence of both abrupt and gradual changes has been developed in the framework of BLS in this paper. The key to the proposed scheme is the introduction of window length adjustment strategies which make effective use of the measurements during the transient and the steady-state intervals. Simulation results have shown that the proposed scheme outperforms the RLS-type algorithms and possesses both excellent parameter tracking and steady-state performance.

The results presented in this paper is significant in the sense that they really make us to re-think about the following issue: should we use *blockwise* or *recursive* least squares? In view of its excellent performance and the rapid advance in computer technology, it is our view that the blockwise LS approaches will play even bigger role in signal processing, communications, control and other engineering applications in a very near future in the situations where RLS algorithms have been predominant up till now.

ACKNOWLEDGEMENTS

The authors would like to thank the Subject Editor and the reviewers for their insightful and constructive comments. Those comments have helped the authors to improve the quality of the paper significantly. The research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

1. Benveniste A, Metivier M, Priouret P. *Adaptive Algorithms and Stochastic Approximations*. Springer: Berlin, Germany, 1990.
2. Goodwin GC, Sin KS. *Adaptive Filtering, Prediction and Control*. Prentice-Hall: Englewood Cliffs, NJ, 1984.

3. Jiang J, Doraiswami R. Direct and blockwise identification of decomposable systems using recursive least-squares algorithms. *International Journal of Systems Science* 1988; **19**(12):2441–2447.
4. Ljung L. *System Identification: Theory for the User* (2nd edn). Prentice-Hall PTR: Upper Saddle River, NJ, 1999.
5. Ljung L, Gunnarsson S. Adaptation and tracking in system identification—a survey. *Automatica* 1990; **26**(1):7–21.
6. Ljung L, Soderstrom T. *Theory and Practice of Recursive Identification*. MIT Press: Cambridge, MA, 1983.
7. Mendel JM. *Lessons in Estimation Theory for Signal Processing*, *Communications*, *and Control*. Prentice-Hall: Englewood Cliffs, NJ, 1995.
8. Niedzwiecki M. *Identification of Time-varying Processes*. Wiley: Chichester, UK, 2000.
9. Astrom KJ, Eykhoff P. System identification—a survey. *Automatica* 1971; **7**(2):123–162.
10. Fortescue TR, Kershenbaum LS, Ydstie BE. Implementation of self-tuning regulators with variable forgetting factors. *Automatica* 1981; **17**(6):831–835.
11. Ljung L. Recursive least-squares and accelerated convergence in stochastic approximation schemes. *International Journal of Adaptive Control and Signal Processing* 2001; **15**(2):169–178.
12. Mahony RE, Lozano R. Generalized forgetting functions for on-line least-squares identification of time-varying systems. *International Journal of Adaptive Control and Signal Processing* 2001; **15**(4):393–413.
13. Nielsen HA, Nielsen TS, Joensen AK, Madsen H, Holst J. Tracking time-varying-coefficient functions. *International Journal of Adaptive Control and Signal Processing* 2000; **14**(8):813–828.
14. Kulhavy R, Kraus FJ. On duality of regularized exponential and linear forgetting. *Automatica* 1996; **32**(10): 1403–1415.
15. Song S, Lim JS, Baek S, Sung KM. Gauss Newton variable forgetting factor recursive least squares for time varying parameter tracking. *Electronics Letters* 2000; **36**(11):988–990.
16. Toplis B, Pasupathy S. Tracking improvement in fast RLS algorithms using a variable forgetting factor. *IEEE Transactions on Acoustics*, *Speech and Signal Processing* 1988; **36**(2):206–227.
17. Wang L, Langari R. A variable forgetting factor RLS algorithm with application to fuzzy time-varying systems identification. *International Journal of Systems Science* 1996; **27**(2):205–214.
18. Jiang J, Cook R. Fast parameter tracking RLS algorithm with high noise immunity. *Electronics Letters* 1992; **28**(22):2042–2045.
19. Parkum JE, Poulsen NK, Holst J. Recursive forgetting algorithms. *International Journal of Control* 1992; **55**(1): 109–128.
20. Salgado ME, Goodwin GC, Middleton RH. Modified least squares algorithm incorporating exponential resetting and forgetting. *International Journal of Control* 1988; **47**(2):477–491.
21. Belge M, Miller EL. A sliding window RLS-like adaptive algorithm for filtering alpha-stable noise. *IEEE Signal Processing Letters* 2000; **7**(4):86–89.
22. Choi BY, Bien Z. Sliding-windowed weighted recursive least-squares method for parameter estimation. *Electronics Letters* 1989; **25**(20):1381–1382.
23. Liu H, He Z. A sliding-exponential window RLS adaptive algorithm: properties and applications. *Signal Processing* 1995; **45**(3):357–368.
24. Basseville M. Detecting changes in signals and systems—a survey. *Automatica* 1988; **24**(3):309–326.
25. Basseville M. On-board component fault detection and isolation using the statistical local approach. *Automatica* 1998; **34**(11):1391–1415.
26. Basseville M, Nikiforov I. *Detection of Abrupt Changes*: *Theory and Application*. Prentice-Hall: Englewood Cliffs, NJ, 1993.
27. Gustafsson F. *Adaptive Filtering and Change Detection*. Wiley: West Sussex, England, 2000.
28. Patton RJ, Frank PM, Clark RN. *Fault Diagnosis in Dynamic Systems*: *Theory and Application*. Prentice-Hall: Englewood Cliffs, NJ, 1989.
29. Perriot-Mathonna D. Improvements in the application of stochastic estimation algorithms parameter jump detection. *IEEE Transactions on Automatic Control* 1984; **29**(11):962–969.
30. Wu NE, Zhang YM, Zhou K. Detection, estimation, and accommodation of loss of control effectiveness. *International Journal of Adaptive Control and Signal Processing* 2000; **14**(7):775–795.
31. Zhang Q, Basseville M, Benveniste A. Early warning of slight changes in systems. *Automatica* 1994; **30**(1):95–113.
32. Zhang YM, Jiang J. An active fault-tolerant control system against partial actuator failures. *IEE Proceedings—Control Theory and Applications* 2002; **149**(1):95–104.

