A RESIDUAL-BASED SELECTIVE WINDOW FOR ROBUST RECURSIVE LEAST SQUARES ESTIMATION

S.F. Hsieh
Dept. of Communication Engineering
Nat'l Chiao Tung University
Hsinchu, Taiwan 30039

K.J.R. Liu
Electrical Engineering Dept.
Systems Research Center
University of Maryland
College Park, MD 20742

ABSTRACT

A new algorithm performing recursive least squares (RLS) estimation is proposed. It is based on selectively rejecting outliers arising from noise spikes; therefore, this method can avoid the bias of parameters estimation due to some large noise perturbations. Unlike a sliding fixed-window scheme, this new windowing scheme can be *non*-continuous. It depends on the estimated level of observed errors (residual). By monitoring the residuals in a recursive manner, we can effectively remove those spurious observed data by *downdating* them. The proposed scheme is very useful especially when some short-time large interferences perturb the system occasionally. In this respect, it outperforms existing schemes, either exponentially growing or sliding window. Computer simulations will be given to justify this.

1 Introduction

A least-squares (LS) fit method assumes that the occurrences of errors or residuals associated with observed data are equally likely and Gaussian distributed. However, in nonstationary cases, especially under bursty errors conditions, this assumption becomes invalid. The interest of robust LS methods hence emerges. Robust estimation is used by statisticians to describe an estimating process that is insensitive to large perturbations to a fraction of its input data or a slight deviation of the full input data. A robust LS estimation [11] can be cast as finding a fitting vector $\mathbf{w}$, such that the *size* of the residual vector $\mathbf{r} \in \Re^n$,

$$\sum_{i=1}^{n} \sigma_i(r_i), \qquad (1)$$

is minimized over all possible $\mathbf{w}$ in $\Re^p$. Here the residual vector is defined as $\mathbf{r} = X\mathbf{w} - \mathbf{y}$, where $X \in \Re^{n\times p}$ and $\mathbf{y} \in \Re^n$ are known. In (1), we need to define the robust functions $\sigma_.(\cdot)$. If we choose $\sigma_i(r_i)$ such that it depends on the time index i, then (1) reduces to present nonstationary LS methods. Examples include $\sigma_i(r) = \lambda^i r^2$, $i = 1,\ldots,n$ for an exponentially weighted window, and $\sigma_i(r) = r^2(u(i-n-1+\ell) - u(i-n-1))$ for a sliding window, where ℓ represents the fixed-window size, $u(\cdot)$ is an unit step function, and $n \geq \ell$ is assumed. On the other hand, we can let the robust function $\sigma_i(r_i)$ be identical for all time indices, namely, $\sigma_i(r) = \sigma(r)$, $\forall 1 \leq i \leq n$, and choose $\sigma(r)$ in a manner such that the influence of outlying residuals can be deemphasized. One example is to let

$$\sigma(r) = \begin{cases} r^2/2, & |r| \leq \eta, \\ \eta|r| - \eta^2/2, & |r| > \eta, \end{cases}$$

where η is a parameter[7]. While there are many robust estimation methods, we will choose a simple scheme which will first find a conventional LS fit and its associated residual vector, followed by downweighting or removing those badly contaminated data as revealed from the feedback of the outliers in the residuals [2,4,12].

In next section, we will first derive a recursive formula based on up/down-dating QRD to monitor all of the residuals *without* explicitly computing the optimum fitting vector $\hat{\mathbf{w}}$, and substituting back to $\mathbf{r} = X\mathbf{w} - \mathbf{y}$. This derivation shares the same spirit and, in the sense of bypassing the optimum fitting vetor in obtaining the residual vector, is a generalization of the systolic recursive LS filtering proposed by McWhirter [10]. In Section 3, we will describe the residual-based selective window for robust RLS estimation.. Comparisons of computer simulations to other existing windows and conclusions will be given in Section 4.

2 QRD-based Residual Monitoring

Consider a time recursive LS problem:

$$X(n)\mathbf{w}(n) \approx \mathbf{y}(n),$$

where $X(n)$ and $\mathbf{y}(n)$ have growing dimensions in the number of rows (growing window),

$$X(n) = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \in \Re^{n\times p}, \qquad \mathbf{y}(n) = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \Re^n.$$

The LS solution $\mathbf{w}(n) \in \Re^p$ is computed such that the Euclidean norm of the residual vector $\mathbf{r}(n) = X(n)\mathbf{w}(n) - \mathbf{y}(n)$ is minimized. We note that $r_i(n) = \mathbf{x}_i^T\mathbf{w}(n) - y_i$, $1 \leq i \leq n$ denotes the residual (error) of the i-th row (observation) at

CH2977-7/91/0000-1009 $1.00 © 1991 IEEE

time n when we want to make the best fit from the columns of $X(n)$ to $\mathbf{y}(n)$.

Suppose the QR decomposition of the *augmented* matrix $[\, X(n) \;\; \mathbf{y}(n)\,]$ is known at time n,

$$[\, X(n) \;\; \mathbf{y}(n)\,] = [\, Q(n) \;\; Q^{\perp}(n)\,] \begin{bmatrix} R(n) & \mathbf{u}(n) \\ 0 & \mathbf{v}(n) \end{bmatrix}, \quad (2)$$

where $Q(n) \in \mathcal{R}^{n \times p}$ and $Q^{\perp}(n) \in \mathcal{R}^{n \times (n-p)}$ represent the orthogonal range and null spaces of the data matrix $X(n)$, and $\mathbf{u}(n) \in \mathcal{R}^{p}$ is the projection of $\mathbf{y}(n)$ onto $Q(n)$, $\mathbf{v}(n) \in \mathcal{R}^{n-p}$ is its counterpart projected onto $Q^{\perp}(n)$, and $R(n) \in \mathcal{R}^{p \times p}$ is an upper-triangular matrix and assumed to be full-rank. $R(n)$ is sometimes called the *Cholesky factor* of the covariance matrix of $X(n)$ in that the Cholesky factorization of $X^T(n)X(n)$ can be uniquely expressed as $R^T(n)R(n)$ subject to the signs in each rows of $R(n)$ as long as $X(n)$ has full column rank.

Because an orthogonal transformation preserves the Euclidean norms of of a vector, it can be shown that[8]

$$\begin{aligned} \|\mathbf{e}(n)\| &= \|X(n)\mathbf{w}(n) - \mathbf{y}(n)\| & (3) \\ &= \left\| [\, Q(n) \;\; Q^{\perp}(n)\,] \begin{bmatrix} R(n)\mathbf{w}(n) - \mathbf{u}(n) \\ -\mathbf{v}(n) \end{bmatrix} \right\| & (4) \\ &= \| - Q^{\perp}(n)\mathbf{v}(n)\| & (5) \end{aligned}$$

as long as

$$R(n)\mathbf{w}(n) = \mathbf{u}(n)\,. \quad (6)$$

(5) means that the residual vector while estimating $\mathbf{y}(n)$ from $X(n)$ must lie in the null space of $X(n)$ which corresponds well with the geometrical interpretation of the orthogonal principle of LS problems.

. As the time index n advances by one, i.e., a new data row $[\mathbf{x}_{n+1}^T \;\; y_{n+1}\,]$, is acquired, we can write the recurrence formula for QRD as follows:

$$\begin{aligned} [\, X(n+1) \;\; \mathbf{y}(n+1)\,] &= \begin{bmatrix} X(n) & \mathbf{y}(n) \\ \mathbf{x}_{n+1}^T & y_{n+1} \end{bmatrix} & (7) \\ = \begin{bmatrix} Q(n) & Q^{\perp}(n) & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} Q_{n+1} & & Q_{n+1}^{\perp} \\ & I_{n-p} & \\ \widehat{Q}_{n+1} & & \widehat{Q}_{n+1}^{\perp} \end{bmatrix} \cdot \\ & \begin{bmatrix} R(n+1) & \mathbf{u}(n+1) \\ 0 & \mathbf{v}(n) \\ 0 & v_{n+1} \end{bmatrix} & (8) \\ = [\, Q(n+1) \;\; Q^{\perp}(n+1)\,] & \begin{bmatrix} R(n+1) & \mathbf{u}(n+1) \\ 0 & \mathbf{v}(n+1) \end{bmatrix} & (9) \end{aligned}$$

By defining

$$\widetilde{Q}(n+1) \equiv \begin{bmatrix} Q_{n+1} & Q_{n+1}^{\perp} \\ \widehat{Q}_{n+1} & \widehat{Q}_{n+1}^{\perp} \end{bmatrix} \in \mathcal{R}^{(p+1)\times(p+1)}, \quad (10)$$

we note that $\widetilde{Q}(n+1)$ constitutes an orthogonal transformation to annihilate the newly appended data row $\mathbf{x}_{n+1}$, and $Q_{n+1} \in \mathcal{R}^{p \times p}$ and $\widehat{Q}_{n+1} \in \mathcal{R}^{1 \times p}$ represent the operation of modifying the *range* space while $Q_{n+1}^{\perp} \in \mathcal{R}^{p \times 1}$, and $\widehat{Q}_{n+1}^{\perp} \in \mathcal{R}_{1 \times 1}$ that of the *null* space. We use a *hat* ^ to denote the new dimensional growth due to the appended data. To sum up, we have the following recurrence formula:

$$\begin{aligned} Q(n+1) &= \begin{bmatrix} Q(n)Q_{n+1} \\ \widehat{Q}_{n+1} \end{bmatrix} \in \mathcal{R}^{(n+1) \times p}, & (11) \\ Q^{\perp}(n+1) &= \begin{bmatrix} Q^{\perp}(n) & Q(n)Q_{n+1}^{\perp} \\ 0 & \widehat{Q}_{n+1}^{\perp} \end{bmatrix}, & (12) \\ \mathbf{v}(n+1) &= \begin{bmatrix} \mathbf{v}(n) \\ v_{n+1} \end{bmatrix} \in \mathcal{R}^{(n+1)-p}, & (13) \end{aligned}$$

$$\begin{aligned} \begin{bmatrix} R(n+1) & \mathbf{u}(n+1) \\ 0 & v_{n+1} \end{bmatrix} &= \widetilde{Q}(n+1) \begin{bmatrix} R(n) & \mathbf{u}(n) \\ \mathbf{x}_{n+1}^T & y_{n+1} \end{bmatrix} \\ &= \begin{bmatrix} Q_{n+1} & Q_{n+1}^{\perp} \\ \widehat{Q}_{n+1} & \widehat{Q}_{n+1}^{\perp} \end{bmatrix} \begin{bmatrix} R(n) & \mathbf{u}(n) \\ X_{n+1} & \mathbf{y}_{n+1} \end{bmatrix}. & (14) \end{aligned}$$

The desired optimum weighting vector $\mathbf{w}(n+1)$ and the residual vector $\mathbf{r}(n+1)$ are thus given by

$$R(n+1)\mathbf{w}(n+1) = \mathbf{u}(n+1), \quad (15)$$

which can be solved by back substitution, and

$$\begin{aligned} \mathbf{r}(n+1) &= -Q^{\perp}(n+1)v_{n+1} \qquad \text{(see (5))} & (16) \\ &= \begin{bmatrix} \mathbf{r}(n) - Q(n)Q_{n+1}^{\perp}v_{n+1} \\ -\widehat{Q}_{n+1}^{\perp}v_{n+1} \end{bmatrix} \in \mathcal{R}^{n+1}. & (17) \end{aligned}$$

To see the changes of residuals in each previous data blocks due to a new observation of $\mathbf{x}_{n+1}^T$ and y_{n+1}, we can write down the following lemma.

Lemma 1 *(updating residual)*

$$\mathbf{r}(n+1) = \begin{bmatrix} r_1(n) - \widehat{Q}_1 Q_2 \cdots Q_n Q_{n+1}^{\perp} v_{n+1} \\ r_2(n) - \widehat{Q}_2 Q_3 \cdots Q_n Q_{n+1}^{\perp} v_{n+1} \\ \vdots \\ r_n(n) - \widehat{Q}_n Q_{n+1}^{\perp} v_{n+1} \\ -\widehat{Q}_{n+1}^{\perp} v_{n+1} \end{bmatrix} \in \mathcal{R}^{n+1} \quad (18)$$

Proof.(18) can be derived from (11) and by noting that $Q(1) = \widehat{Q}_1$, *i.e.,*

$$\begin{aligned} Q(2) &= \begin{bmatrix} \widehat{Q}_1 Q_2 \\ \widehat{Q}_2 \end{bmatrix} \\ Q(3) &= \begin{bmatrix} Q(2)Q_3 \\ \widehat{Q}_3 \end{bmatrix} = \begin{bmatrix} \widehat{Q}_1 Q_2 Q_3 \\ \widehat{Q}_2 Q_3 \\ \widehat{Q}_3 \end{bmatrix} \\ &\vdots \\ Q(n) &= \begin{bmatrix} \widehat{Q}_1 Q_2 \cdots Q_n \\ \widehat{Q}_2 Q_3 \cdots Q_n \\ \vdots \\ \widehat{Q}_n \end{bmatrix} & (19) \end{aligned}$$

and substituting $Q(n)$ *back into (17).* ■

(18) explains that the overall residual vector at time $n+1$ comprises of two parts: one of them is equal to $-\widehat{Q}^{\perp}_{n+1}v_{n+1}$, the new *dimensional growth due to* $\mathbf{x}^T_{n+1}$, *while the* other one is equal to the old residual vector at the previous time n, $\mathbf{r}(n)$, offset by $Q(n)Q^{\perp}_{n+1}v_{n+1}$. Therefore, if we are only interested in $R(n+1)$ and/or r_{n+1}, then we can simply maintain the information of $R(n)$ and $\mathbf{u}(n)$, which is usually the case for many applications such as beamforming[10]. However, if we need to monitor all of those previously block residual vectors r_i, $i = 1, \cdots, n$, then the previously computed range space $Q(n)$ is still required to update those old residual vectors. This monitoring may aid in the determination of some *spurious* observations(rows) such that they can be deleted (downdated) from the LS estimation problem and mitigate the possible bias caused by them. For linear regression [3,8], this diagnosis in monitoring all the residuals is especially very important. Our method, following the approach first proposed by McWhirter [10], provides a *one-pass* direct way of keeping track of all of the residuals, without explicitly computing $\mathbf{w}(n)$ followed by $X(n)\mathbf{w}(n) - \mathbf{y}(n)$ which requires *two-passes* (involving the use of back substitution twice) and can be objectionable from the throughput point of view. We will elaborate on this later in next section.

3 Robust RLS estimation based on Residual-Outliers Rejection

For a *robust* LS problem we need to examine the residual associated with the LS problem, from which some data rows may be discarded or deemphasized. An exponentially weighted windowing scheme always assumes the old data should be gradually deemphasized, hence a forgetting factor is used in this method, while a fixed-window demands the old data to be discarded completely. Both methods are commonly used for adaptive signal processing; their performances are not satisfactory when the system is perturbed by occasional noise spikes. Accordingly, the need of residual-based robust LS estimation arises.

After obtaining the optimum fitting coefficient $\mathbf{w}(n)$ and also the corresponding residual $r_1(n), \ldots, r_n(n)$, we can determine an index set $\mathcal{I}$ based on some criteria, e.g., $\mathcal{I} = \{ i \mid 1 \le i \le n, r_i(n) < \text{ threshold } \}$. This is called the first pass for the robust LS solution. Next, remove all $\mathbf{x}_i^T$ and y_i, $\forall i \notin \mathcal{I}$, from $X(n)\mathbf{w}(n) \approx \mathbf{y}(n)$, which now becomes $X_{\mathcal{I}}(n)\mathbf{w} \approx \mathbf{y}_{\mathcal{I}}(n)$. Resolve it and this is called the second pass for the LS solution. To avoid the cumbersome two-times back substitutions (two-passes), we propose to monitor the residual whenever a new data row is appended to our system. By doing so, the first pass of back substitution can be bypassed, and a one-pass robust LS solution is possible. A recursive formula based on QR decomposition to update the residual is derived in the previous Lemma. After the residual is updated, a decision based on the magnitude of each entry in the residual is made to determine which data rows are to be discarded (downdated). Similar recursive formulas to the updating operations also exists for the downdating operations [1,6,12]. Figure 1 depicts the block diagrams of the two-passes and one-pass robust LS estimation.

4 Simulations and Conclusions

A second order AR model is used to demonstrate the advantages of the new window scheme. Let $\{u(i)\}$ be an AR process [5, pp. 204–6] given by $u(i)+a_1u(i-1)+a_2u(i-2) = v(i), ; i = 1, \ldots, 250$, with $a_1 = -0.9750$ and $a_2 = 0.9500$. The excitation noise $v(i)$ is a white Gaussian noise with a standard deviation of 0.1 except that from $i = 55$ to 57 and also from $i = 155$ to 157 $v(i)$ will be intentionally increased by a factor of 30 to account for temporary large noisy spikes. This is equivalent to lowering the SNR by about 30 dB during these intervals.

To make a fair comparison between the fixed-window scheme with a window size ℓ to the the exponentially weighting scheme with a forgetting factor λ in the sense that both schemes have the same *self noise* (*i.e.,* fluctuation of the estimated parameters with respect to the optimum AR parameters) [9], we choose $\ell = 50$ and $\lambda = \sqrt{(\ell-1)/(\ell+1)}$. 100 simulations with different noise realizations using MATLAB are performed.

Figs. 2 and 3 compare the biases of estimating the AR parameters a_1 and a_2. Figs. 4 and 5 compare the standard deviations of estimating the AR parameters a_1 and a_2. Fig. 6 compares the standard deviations of the residuals. Four windowing schemes are compared: (1). no windows are imposed (or equivalently, forgetting factor $\lambda = 1$); (2). exponentially weighted window with forgetting factor $\lambda = \sqrt{49/51}$; (3). fixed-size sliding window with window size $\ell = 50$; and (4). selective window with residual threshold = 1.0.

From these figures, we can see that the newly proposed window by selectively rejecting data rows with large residuals gives the least bias in tracking the AR parameters and converges most rapidly. This is obviously because this method discarded those highly perturbed data.

References

[1] S. T. Alexander, C.-T. Pan, and R. J. Plemmons, "Analysis of a recursive least squares hyperbolic rotation algorithm for signal processing," *Linear Algebra and its Applications:* **98**, pp. 3–40, 1988.

[2] A. C. Atkinson, *Plots, Transformations, and Regression,* Clarendon Press, Oxford, 1985.

[3] R. W. Farenbrother, *Linear Least Squares Computations.* Marcel Dekker, Inc., New York and Basel, 1988.

[4] C. Goodall, "Examining residuals," in *Understanding Robust and Exploratory Data Analysis,* edited by D. C. Hoaglin, F. Mosteller, and J. W. Tukey, John Wiley & Sons, Inc., pp. 211–246, 1983.

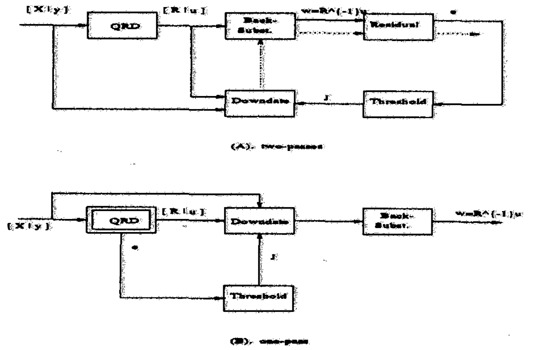

Figure 1: Two-passes and one-pass block diagrams of robust LS estimations with outlier removal

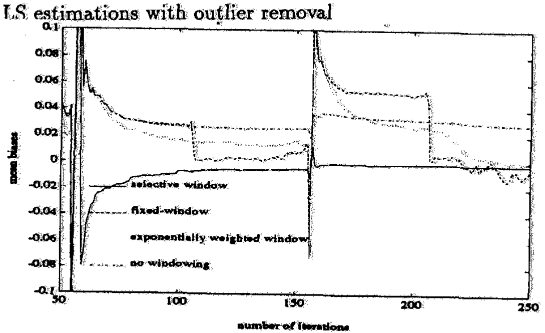

Figure 2: Comparisons of mean bias of estimating AR parameter a_1 for various windows under noisy spikes.

[5] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[6] S. F. Hsieh and K. Yao, "Systolic implementation of windowed recursive LS estimation," *Proc. of IEEE Int'l Symp. on Circuits and Systems*, New Orleans, pp. 1931–1934, 1990.

[7] P. J. Huber, "Robust estimation of a location parameter," *Annals Math. Statist.*, **35**, pp. 73–101, 1964.

[8] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, N.J., 1974.

[9] D. Manolakis, F. Ling, and J. G. Proakis, "Efficient time-recursive least-squares algorithms for finite-memory adaptive filtering," *IEEE Trans. on Circuits and Systems*, Vol. CAS-34, No. 4, pp. 400–407, Apr. 1987.

[10] J. G. McWhirter, "Recursive least-squares minimisation using a systolic array," *Proc. SPIE 431, Real time signal processing VI*, pp. 105–112, 1983.

[11] D. P. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM J. Matrix Anal. Appl.*, Vol. 11, No. 3, pp. 466–480, July 1990.

[12] C. M. Rader and A. O. Steinhardt, "Hyperbolic Householder transformations," *IEEE Trans. on Acoust., Speech, Signal Processing*, Vol. ASSP-34, No. 6, pp. 1589–1602, Dec. 1986.

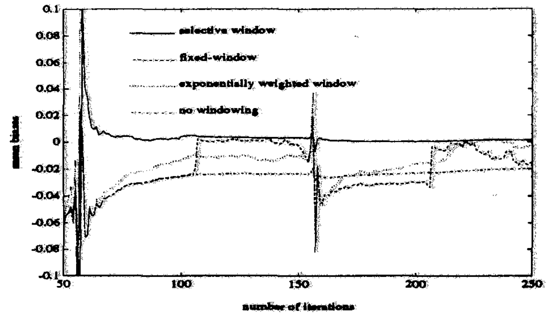

Figure 3: Comparisons of mean bias of estimating AR parameter a_2 for various windows under noisy spikes.

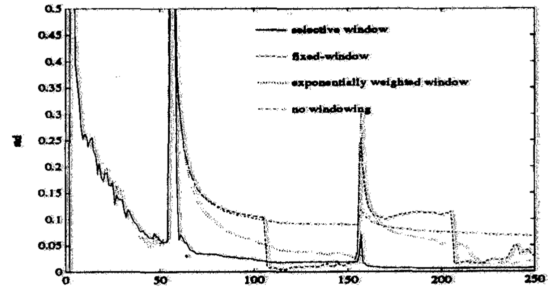

Figure 4: Comparisons of standard deviations of estimating AR parameter a_1 for various windows under noisy spikes.

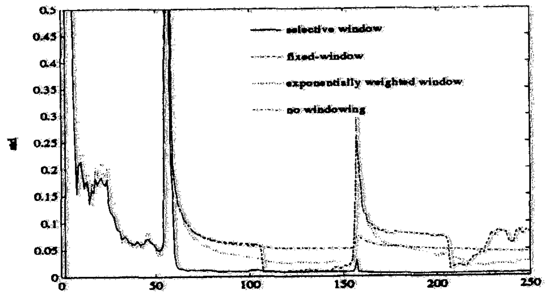

Figure 5: Comparisons of standard deviations of estimating AR parameter a_2 for various windows under noisy spikes.

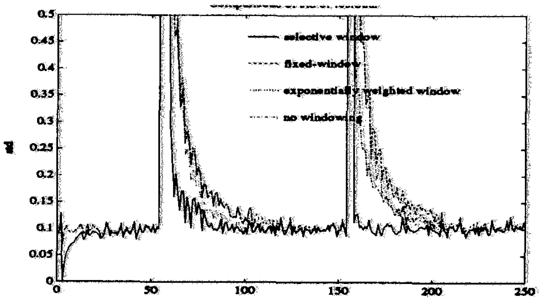

Figure 6: Comparisons of standard deviations of residuals of estimating AR parameters for various windows under noisy spikes.