A New Stabilization Technique for the Fixed Point Prewindowed RLS Algorithm

Tulay Adali
Sasan H. Ardalan

Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
North Carolina State University

CCSP TR-90/5
June 1990

A NEW STABILIZATION TECHNIQUE FOR THE FIXED-POINT PREWINDOWED RLS ALGORITHM

Tülay Adalı and Sasan H. Ardalan

Center for Communications and Signal Processing
North Carolina State University
Raleigh, North Carolina 27695-7914

ABSTRACT

In this correspondence, a stable finite precision Recursive Least Squares (RLS) algorithm is derived for the prewindowed growing memory case (forgetting factor, $\gamma = 1$). Previously, it has been shown that the prewindowed growing memory RLS algorithm diverges under fixed-point implementation [2,1]. The random walk phenomenon due to roundoff errors in the weight update causes the divergence of the algorithm. To overcome this effect, these roundoff errors are modeled such that their effect is incorporated into the algorithm. The steady-state behaviour of this new algorithm is analyzed, and it is shown that the divergence phenomenon is actually eliminated, and the new algorithm converges.

I. Introduction

It has been shown that, for the prewindowed RLS algorithm ($\gamma = 1$), the roundoff error associated with the weight update recursion leads to divergence as the algorithm iterates, [2]. In the literature, this phenomenon has been explained as a random walk process for the weight vector [2,8]. For the exponentially windowed RLS algorithm, ($\gamma < 1$), this roundoff error is amplified if the forgetting factor, γ is chosen very close to one. For the prewindowed RLS, the random walk process can be modeled as a stochastically varying weight vector, and the algorithm can be rederived for this case.

In the analysis, we have considered a system identification problem which is a common adaptive filtering application. After a brief introduction of the conventional RLS algorithm, the fixed point modeling used is presented. The basis for the divergence of the algorithm is discussed. A new model for the fixed-point weights is introduced, and then the stabilized RLS algorithm, using this model is given. The steady-state behaviour of this new algorithm is analyzed and finally simulation results comparing the two versions of the prewindowed algorithm, the original and the stabilized one, are presented.

II. RLS Algorithm

The problem is modeled as a linear system with input signal $x(n)$, and output signal $d(n)$. The response of the system, available for measurement at the filter input $z(n)$, is the sum of the desired signal, $d(n)$, and a random, additive white noise component, $v(n)$.

$$z(n) = d(n) + v(n) \tag{1}$$

The samples $d(n)$ can then be written in terms of the system impulse response coefficients, $\mathbf{w}^*$ as

$$d(n) = \mathbf{w}^{*T}\mathbf{x}(n) = \sum_{i=0}^{N-1} w_i^* x(n-i). \tag{2}$$

It is assumed that the system response has insignificant terms beyond N samples. The sample vectors then, have the last N samples of data; that is, the input data vector at time n is

$$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-N+1)]^T. \tag{3}$$

Now, consider the system identification problem, where the system parameters, $\mathbf{w}^*$ are to be estimated. In the RLS algorithm, the weights $\mathbf{w}(n)$ are calculated such that the accumulated sum of the error residuals is minimized. The error residual is the difference between the output of the system, $z(n)$, and an estimate of the desired response obtained from $\mathbf{w}(n)$, at time n,

$$\epsilon(n) = z(n) - \mathbf{x}^T(n)\mathbf{w}(n) \tag{4}$$

and the quantity minimized is

$$\varepsilon(n) = \sum_{i=1}^{n} \gamma^{n-i} \epsilon^2(n), \tag{5}$$

where γ is the forgetting factor and is chosen less than one.

The conventional RLS algorithm can then be summarized as follows [4]:

$$\begin{aligned} e(n) &= z(n) - \mathbf{x}^T(n)\mathbf{w}(n-1) && (6)\\ \mathbf{k}(n) &= \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\gamma + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)} && (7)\\ \mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n)e(n) && (8)\\ \mathbf{P}(n) &= \frac{1}{\gamma}\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1) && (9) \end{aligned}$$

with initializations:

$$\begin{aligned} \mathbf{P}(0) &= \delta^{-1}\mathbf{I} \quad (\delta \ll 1)\\ \mathbf{w}(0) &= 0. \end{aligned}$$

III. Fixed Point Roundoff Error Model for RLS

In the fixed-point model of the RLS algorithm, we assume that the Kalman gain is precomputed and then quantized. Thus, it is perturbed by the error term $\boldsymbol{\beta}(n)$ such that;

$$\mathbf{k}'(n) = \mathbf{k}(n) + \boldsymbol{\beta}(n) \tag{10}$$

where we have used primed variable to represent the fixed point quantity. Then, two calculations are important in the analysis: the calculation of the weight vector update, (8), and the calculation of the prediction of the desired signal, $d(n)$, (2). Using subscripts for a single element of the vector quantities, and primes for the finite register lengths, we have

$$w_i'(n) = w_i'(n-1) + e'(n)k_i'(n) + \mu_i(n) \tag{11}$$

where $\mu_i(n)$ is the roundoff error introduced by rounding the product in the i*th* entry of the equation. Similarly, we introduce the roundoff error vector, $\boldsymbol{\epsilon}(n)$. Again, $\epsilon_i(n)$ is the i*th* component of the error vector, such that;

$$e_i'(n) = z_i'(n) - \mathbf{x}_i'(n)\mathbf{w}_i'(n-1) + \epsilon_i(n). \tag{12}$$

Both of these roundoff errors are modeled as uniformly distributed, uncorrelated additive noise processes with zero mean and variances $\sigma_\mu^2 = 2^{-2B_\mu}/12$ and $\sigma_\epsilon^2 = 2^{-2B_\epsilon}/12$, [10]. Here, B_μ and B_ϵ are the register lengths in bits used for the fractional part of the weights and the data respectively. This model is commonly used for finite register effects [10], and is quite accurate if the quantities

that are multiplied do not become very small and if the step size, δ is chosen such that σ_x^2/δ is a considerably large quantity [11], which is most often the case. This ratio is a measure of the signal dynamic range and here, σ_x^2 is the variance of the input. In the analysis, we assume that the Kalman gain is computed in infinite precision and then quantized.

We define the error vector as the difference between the fixed-point weight vector estimate at time n, $\mathbf{w}'(n)$, and the optimum system parameters, $\mathbf{w}^*$:

$$\boldsymbol{\theta}'(n) = \mathbf{w}'(n) - \mathbf{w}^*. \tag{13}$$

Using this definition for the error, the steady-state value of the mean norm square error is [1]:

$$E\left\{\|\boldsymbol{\theta}'(n)\|^2\right\} = N\sigma_\mu^2\frac{n}{3} + \frac{\sigma_\eta^2}{n}\sum_{i=1}^{N}\frac{1}{\lambda_i}. \tag{14}$$

Here, λ_i's are the eigenvalues of the input autocorrelation matrix, $\mathbf{R}_x$, and we have introduced the term, σ_η^2;

$$\sigma_\eta^2 = E\{\eta^2(n)\} \tag{15}$$

where

$$\eta(n) = v(n) - \epsilon(n). \tag{16}$$

In (14), the first term which is proportional to n, the number of iterations, is the most important term, leading to the divergence as the algorithm iterates. This is due to the roundoff error in the weight update recursion. The roundoff error variance σ_μ^2 is typically much smaller than σ_η^2, the additive noise component. Hence, correlation enhances the additive noise error initially in the weight error but its effect eventually dies out as the number of iterations increases. The random walk phenomenon due to roundoff errors in the weight update dominates, leading to the divergence. This behaviour of the roundoff error, $\mu(n)$, can also be explained as follows:

For $\gamma = 1$, we have

$$E\{\mathbf{P}(n)\} = \frac{1}{n}\mathbf{R}_x^{-1} \tag{17}$$

where $\mathbf{R}_x^{-1}$ is the inverse of the true input autocorrelation matrix. We also have

$$\mathbf{k}(n) = \mathbf{P}(n)\mathbf{x}(n), \tag{18}$$

which can be easily derived by using (9) and (7). So, as the algorithm iterates, $\mathbf{k}(n) \rightarrow 0$, and the weight vector becomes:

$$\mathbf{w}'(n) = \mathbf{w}'(n-1) + \boldsymbol{\mu}(n). \tag{19}$$

Since $\mu(n)$ is a vector of white noise processes, the estimates for the filter taps, and consequently the coefficients of $\boldsymbol{\theta}'(n)$ behave as random walk processes, [8,2].

IV. Stabilization of the Fixed Point Prewindowed RLS Algorithm

In the prewindowed RLS algorithm ($\gamma = 1$), the estimate for the inverse of the autocorrelation matrix, $\mathbf{P}(n)$ goes to zero as the algorithm iterates (17), and we end up with the random walk process given in (19). The roundoff error term due to weight update $\boldsymbol{\mu}(n)$, then causes the linear growth of the mean square weight error term as depicted in (14). Since the elements of $\boldsymbol{\mu}(n)$ constitute a set of independent white-noise processes each having zero mean and variance σ_μ^2, we have exactly the case of random-walk state model. This is the model used for modeling the nonstationary environment in the application of Kalman filter approach to systems with stochastically varying dynamics. The model used in this approach is given by [9]:

$$\mathbf{w}^*(n) = \mathbf{w}^*(n-1) + \boldsymbol{\mu}(n), \tag{20}$$

where $\mathbf{w}^*(n)$ is the vector of dynamic system parameters. We can use the same model by looking at the same problem in another way. We consider that the computed weights have the stochastic variance property. This is a good model of the actual case, especially for stationary systems and after a large number of iterations, since then we shall have $\mathbf{k}(n) \rightarrow 0$.

When this model of stochastically varying weights (20), is used, the only change in the RLS algorithm, given in (6)-(9) is the addition of a new constant diagonal matrix term,$\mathbf{D}_\mu$, to the estimate of $\mathbf{P}(n)$. The new algorithm is then given as, [9]:

$$\begin{aligned}
e(n) &= z(n) - \mathbf{x}^T(n)\mathbf{w}(n-1) && (21)\\
\mathbf{k}(n) &= \frac{\mathbf{P}^*(n-1)\mathbf{x}(n)}{1 + \mathbf{x}^T(n)\mathbf{P}^*(n-1)\mathbf{x}(n)} && (22)\\
\mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n)e(n) && (23)\\
\mathbf{P}^*(n) &= \mathbf{P}^*(n-1) + \mathbf{D}_\mu - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}^*(n-1). && (24)
\end{aligned}$$

Although all the algorithmic quantities calculated above will be different than those obtained using (6)-(9), we have used a different notation only for $\mathbf{P}(n)$ to indicate that it is computed differently, since it will be the quantity we shall concentrate on.

The new term $\mathbf{D}_\mu$, added to the matrix update equation, (24) is the correlation matrix of the random variable $\boldsymbol{\mu}(n)$, and is equal to

$$\mathbf{D}_\mu = E\{\boldsymbol{\mu}(n)\boldsymbol{\mu}^T(n)\} = \sigma_\mu^2\mathbf{I}. \tag{25}$$

To analyze this new algorithm, we first write (24) using (22) as

$$\mathbf{P}^*(n) = \mathbf{D}_\mu + \left(\mathbf{P}^*(n-1) - \frac{\mathbf{P}^*(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{P}^*(n-1)}{1 + \mathbf{x}^T(n)\mathbf{P}^*(n-1)\mathbf{x}(n)}\right). \tag{26}$$

We then use the matrix inversion lemma [4], for the term in paranthesis in (26), and rewrite it as

$$\mathbf{P}^*(n) - \mathbf{D}_\mu = (\mathbf{P}^{*-1}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n))^{-1}. \tag{27}$$

Now, we define

$$\mathbf{P}^{*-1}(i) = \mathbf{R}^*(i). \tag{28}$$

After taking the inverse of both sides in equation (27), and factoring out $\mathbf{P}(n)$ on the left side, we are able to write the same equation as

$$(\mathbf{I} - \sigma_\mu^2 \mathbf{R}^*(n))^{-1}\mathbf{R}^*(n) = \mathbf{R}^*(n-1) + \mathbf{x}(n)\mathbf{x}^T(n). \tag{29}$$

We can now assume that matrix $|\sigma_\mu^2 \mathbf{R}^*(n)|$ is small, i.e. the absolute value of its each entry is smaller than 1. This is a valid assumption provided that the condition $|r_{max}(n)| \ll 1/\sigma_\mu^2$ is satisfied, where $r_{max}(n)$ is the largest entry of $\mathbf{R}^*(n)$. Since σ_μ^2 is typically very small and the analysis shows that $\mathbf{R}^*(n)$ is a bounded matrix, this condition is easily satisfied. Thus, for sufficiently large n, we can make the approximation:

$$(\mathbf{I} - \sigma_\mu^2 \mathbf{R}^*(n))^{-1} \approx \mathbf{I} + \sigma_\mu^2 \mathbf{R}^*(n). \tag{30}$$

With this approximation, we are now able to write

$$\mathbf{R}^*(n) + \sigma_\mu^2 \mathbf{R}^{*2}(n) \approx \mathbf{R}^*(n-1) + \mathbf{x}(n)\mathbf{x}^T(n). \tag{31}$$

Taking the expectation of both sides, and using the approximation in the steady-state that, $E\{\mathbf{R}^*(n)\} = E\{\mathbf{R}^*(n-1)\}$, and $E\{\mathbf{x}(n)\mathbf{x}^T(n)\} = \mathbf{R}_x$, we finally have

$$E\{\mathbf{R}^{*2}(n)\} = \frac{1}{\sigma_\mu^2}\mathbf{R}_x. \tag{32}$$

Assume that the input is a white Gaussian random process. Then, for large n, and assuming small variations about the mean of $\mathbf{R}^*(n)$ [1], we obtain from (32) that

$$E\{\mathbf{R}^*(n)\} = \frac{\sigma_x}{\sigma_\mu}\mathbf{I}. \tag{33}$$

This expression is checked by simulations and it is seen that the assumptions of small variations is a quite reasonable one for uncorrelated data, and for small σ_μ^2, and the expression predicts the steady-state behaviour of matrix $\mathbf{R}^*(n)$ quite well. For the correlated case, the expected value is the same as given by (33), but the variance about the mean increases with correlation.

Initially, the two estimates for the autocorelation function $\mathbf{R}(n)$ and $\mathbf{R}^*(n)$ follow each other very closely, but after sufficiently large number of iterations $\mathbf{R}^*(n)$ reaches a steady-state value

[1] The variance of a random variable x is given by $\sigma_x^2 = E\{x^2\} - E\{x\}^2$. Its mean value is then written as $E\{x\} = \sqrt{E\{x^2\} - \sigma_x^2}$. If σ_x^2 is small compared to $E\{x^2\}$, the expected value of x can be approximated as $E\{x\} \approx \sqrt{E\{x^2\}}$.

and stays there whereas $\mathbf{R}(n)$ continues to increase linearly, causing the eventual divergence of the algorithm. The number of iterations n, before the algorithm reaches steady-state depends on the value of σ_μ^2. It takes longer for $\mathbf{R}^*(n)$ to reach its steady-state value when σ_μ^2 is small.

We can get an expression similar to (18) for this case also. We multiply the right hand side of equation (24) by $\mathbf{x}(n)$, and use equation (22), to get

$$\mathbf{k}(n) = \mathbf{P}^*(n)\mathbf{x}(n) - \mathbf{D}_\mu \mathbf{x}(n). \tag{34}$$

Hence, at steady-state, $\mathbf{k}(n) \rightarrow \alpha \mathbf{x}(n)$ where $\alpha = (\sigma_\mu/\sigma_x) - \sigma_\mu^2$, if the input is uncorrelated. Since $0 < \sigma_\mu^2 \ll 1$, the first term in α, σ_μ/σ_x dominates, and Kalman gain is now limited from below with this positive value, instead of becoming zero.

Thus, for large n, the algorithm behaves as the LMS algorithm, with α as the step size. Before reaching the steady-state value, the algorithm behaves as the normal RLS algorithm, since the added term is not significant at this stage. But as the algorithm iterates the new added component $\mathbf{D}_\mu$ dominates and prevents the new estimate of the inverse autocorrelation matrix, $\mathbf{R}^{*-1}(n)$ from going to zero. The presence of this quantity also ensures the positive definiteness of matrix $\mathbf{P}^*(n)$. The loss of positive definiteness, together with the loss of symmetry of the $\mathbf{R}^{-1}(n)$ matrix, is shown to be one of the main reasons for the unstable behaviour of the RLS algorithm, [5],[6].

By an analysis similar to [7], we can derive the steady-state excess weight error equation for the new algorithm, which is also the application of Kalman filter approach to the systems with stochastically varying dynamics, as

$$E\{\|\boldsymbol{\theta}'(n)\|^2\} = \frac{\sigma_\mu^2 \|\mathbf{w}^*\|^2}{2\alpha\sigma_x^2} + \frac{1}{2}\alpha N \sigma_\eta^2 \tag{35}$$

where $\alpha = \sigma_\mu/\sigma_x$ for uncorrelated inputs.

Thus by using the new algorithm, we eliminated the linear growth term in the prediction error in (14), but it should be also noted that now the effects of additive noise component and the roundoff error due to desired response calculation are amplified.

V. Simulation Results

For the simulations, we define a new error vector measure as

$$\zeta(n) = \mathbf{w}'(n) - \mathbf{w}(n) \tag{36}$$

where $\mathbf{w}(n)$ is the weight vector calculated using infinite precision, which is a *double precision* quantity in the simulations. This new quantity is defined since the linear growth observed in (14) is the characteristics we are trying to eliminate. Since σ_μ^2 is typically much smaller than σ_η^2, with the earlier definition (13) as the performance criteria, the algorithm should iterate for a very long

time before this linear growth can be seen. By using this definition, (36), the effect of second term in (14) is left out, and only the bias due to the roundoff error, $\mu(n)$ is present, [2].

The same system identification problem, described in the second section is simulated, using a white gaussian random process as the input. The order of the system used in the simulations is 9, and the variance of the additive noise to the system is 1.0, in order to prevent the algorithm from stopping updating. Kalman gain is computed in infinite percision, and then quantized using 12 bits for its fractional part. 14 bits are used for representing the fractional part of the data and 9 for the weights. The values of the optimum system weights used in the simulations are given in Table 1.

In Figure 1, the norm squared value of the weight error (36), is plotted for the original, prewindowed RLS, which is implemented as given in (6)-(9) together with the stabilized algorithm, (22)-(24). The theoretical mean value for the original algorithm, $nN\sigma_\mu^2/3$, the linear growth is also shown in the graph. When the curves for $E\{||\zeta(n)||^2\}$ for the two cases, original prewindowed RLS algorithm and the stabilized one, are compared, it is clearly seen that the excess error due to roundoff errors have been reduced and the divergence is prevented.

VI. Conclusions

The effect of roundoff errors on the performance of the fixed point RLS algorithm is presented, explaining the reason for the unstable behaviour of the algorithm with a choice of 1 for the forgetting factor γ. Thus, the motivation for the use of a new model is explained, and a stable finite precision RLS algorithm is derived for the prewindowed growing memory case. The steady-state behaviour of this new algorithm is analyzed and it is shown that the divergence phenomenon is actually eliminated and the new algorithm converges.

References

[1] T.Adalı and S.H.Ardalan, "Convergence and error analysis of the fixed point RLS algorithms with correlated inputs," *in Proc. ICASSP 90, pp. 1479-1482, April 3-6 1990, Albuquerque, New Mexico.*

[2] S.H.Ardalan and S.T.Alexander, "Fixed point roundoff error analysis of the exponentially windowed RLS algorithm for time varying systems," *IEEE Trans. Acoust.,Speech,Signal Processing*, vol. ASSP-35,No. 6, pp. 770-783, June 1987.

[3] Graham C. Goodwin and Robert L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, New York, N.Y.: Academic Press. Inc., 1977

[4] Simon Haykin, *Adaptive Filter Theory*, New Jersey: Prentice-Hall, 1986

[5] P. Fabre, C. Gueguen, "Fast recursive least-squares algorithms: preventing divergence," *in Proc. ICASSP 85, pp. 1149-1152, May 1985, Tampa, Florida.*

[6] G.E. Bottomley and S.T. Alexander, "A theoretical basis for the divergence of conventional recursive least squares filters," *in Proc. ICASSP 89, pp. 908-911, May 1989, Glasgow, Scotland.*

[7] S.H.Ardalan, "Floating point roundoff error analysis of the RLS and LMS algorithms ," *IEEE Trans. Circuits Syst.*, vol. CAS-33,No. 12, pp. 1192-1208, Dec. 1986.

[8] Slock and T. Kailath, "Numerically stable fast recursive least-squares transversal filters," *in Proc. ICASSP 88, pp. 1473-1477, April 3-6 1990, New York, N.Y.*

[9] L.Ljung and T.Söderström, *Theory and Practice of Recursive Identification*, Cambridge, M.A.:MIT Press, 1987

[10] C.Caraiscos and B.Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust.,Speech,Signal Processing*, vol. ASSP-32,No. XX, pp. 34-41, Feb. 1984.

[11] C.W.Barnes B.N.Tran, and S.H.Leung, "On the statistics of fixed point roundoff error," *IEEE Trans. Acoust.,Speech,Signal Processing*, vol. ASSP-33, pp. 595-606, June 1985.

i	Impulse Response Coefficients w_i^*
0	-4.320000
1	-8.697600
2	-9.511681
3	-3.783629
4	4.925197
5	9.722113
6	8.121600
7	3.744000
8	0.800000

Table 1

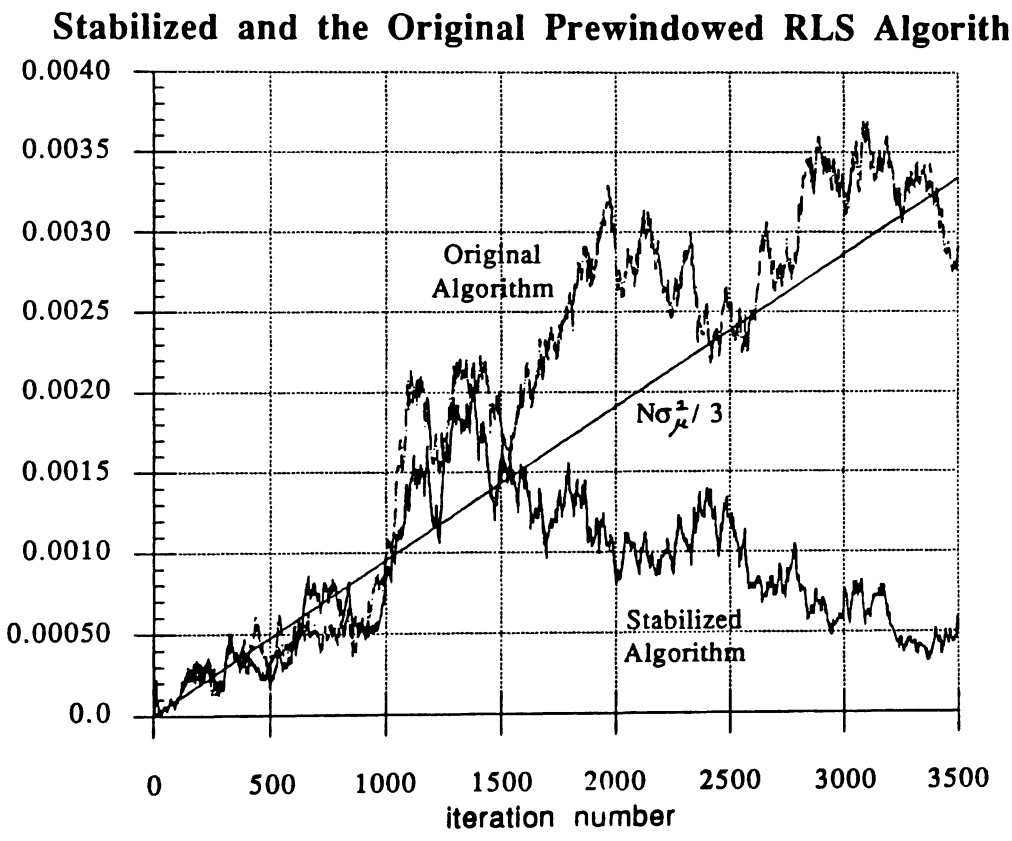
Stabilized and the Original Prewindowed RLS Algorithms
0.0040
0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.00050
0.0
Original
Algorithm
$N\sigma_\mu^2 / 3$
Stabilized
Algorithm
0
500
1000
1500
2000
2500
3000
3500
iteration number

Figure 1