Steady-State Analysis of the Fixed-Point RLS Algorithm

Tülay Adali

Sasan H. Ardalan

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

TR-92/2
June 1992

ABSTRACT

New expressions are derived for the mean weight misadjustment in the fixed-point recursive least squares (RLS) algorithm for the case of time-varying channel estimation. The expressions derived are general, in that they take into account the correlation in the input. It is shown that the additive system noise is amplified by a correlation amplification factor which is defined as a function of the input autocorrelation matrix eigenvalues. Correlation also amplifies the effect of roundoff error due to the desired signal estimate computation. It is also demonstrated that tracking noise and weight update roundoff error behave similarly in that their contributions are proportional to the time constant of the algorithm. Finally, conditions are determined for the termination of updating which is an important problem in the finite precision implementation of adaptive filters. Since additive system noise and tracking noise can be present even if the algorithm is implemented in infinite precision, these results are applicable to the infinite precision RLS problem, and yield new expressions for teh weight misadjustment also in this case. The original results of the analysis are checked by simulations and perfect agreement with theory is observed.

I. INTRODUCTION

This report considers the steady-state analysis of fixed-point RLS algorithm for the general in which the input to the adaptive filter can be correlated. In most of the work analyzing the performance of adaptive filters, authors have made the assumption that the input is composed of independent random data samples, [6, 14, 17, 18, 21, 27]. This has been a common assumption in works considering the analysis of RLS algorithm when implemented in finite precision [6, 17]. This assumption greatly simplifies the evaluations involved because of the nice properties of the white input, such as having a diagonal autocorrelation matrix and thus providing uncoupling in the expressions involved. However, most often the input data to the adaptive filter is correlated, such as in the echo-cancellation using speech signals, and in the equalization of mobile communication channels. Therefore, in this work, the input is assumed to be a wide sense stationary random process, which is a good model for most of the stochastic processes encountered in practice.

In the analysis, the unknown system is assumed to be time-varying hence considering the interaction of fixed-point roundoff errors with the tracking noise in the RLS algorithm for the first time. The performance measure is chosen as the error between the fixed-point weights and the optimum system parameters. This choice enables us to obtain the expression for the steady-state performance of the infinite precision RLS algorithm as a special case, which is also new and an important result in itself.

In this work, first the adaptive filtering problem is formulated, followed by the introduction of the fixed-point version of the RLS algorithm. The model used for the finite precision effects is discussed. Next, the analysis leading to new results for the norm of the weight error vector is presented. A new quantity called the correlation amplification factor is introduced in order to account for the effect of correlation on the performance. These results are then checked by simulations where a close match is observed with theory.

II. The Problem Formulation

The system to be estimated 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 $v(n)$

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

Assuming that the time-varying system impulse response has insignificant terms beyond N samples, the samples $d(n)$ can be written in terms of the system impulse response coefficients, $\mathbf{w}^*(n)$ as

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

where the input data vector $\mathbf{x}(n)$ contains the last N samples of data; i.e.

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

In a system identification problem, the system coefficients at time n are being estimated based on the knowledge of the unknown system output $d(n)$ which depends on the system tap values at time $(n-1)$.

A time-varying channel is modeled as a first order Markov process

$$\mathbf{w}^*(n) = b\mathbf{w}^*(n-1) + \boldsymbol{\alpha}(n) \tag{4}$$

where the parameter b, which satisfies $|b| < 1$, controls the time constant of the nonstationarity which is $1/(1-b)$. The elements of the random vector $\boldsymbol{\alpha}(n)$ consist of zero mean, independent Gaussian noise processes with covariance matrix $\sigma_\alpha^2\mathbf{I}$.

When the system is nonstationary, the algorithm has to track the variations in the system parameters. Therefore, we use the exponentially windowed RLS algorithm, i.e. we choose the

forgetting factor λ less than 1. The factor $1/(1-\lambda)$ represents the memory of the algorithm, and is also called the time constant of the algorithm [1, 30]. To ensure that the change in the channel, i.e. in the weights $\mathbf{w}^*(n)$, is slow compared to the memory of the algorithm, the condition

$$\frac{1}{1-b} \gg \frac{1}{1-\lambda}$$

has to be satisfied. Since λ is chosen close to 1 [1, 30], in the common region of operation for the adaptive algorithm, b will be very close to but less than 1, [28].

Also, another important consideration in the choice of the two important parameters of the algorithm, the forgetting factor λ and the filter length N, is to satisfy the condition

$$\frac{1}{1-\lambda} \gg N. \tag{5}$$

This is the common region of operation for the adaptive filter because we would like the adaptive filter response to be much faster than the changes in the channel. This has been called the slow adaptation condition [21, 31].

The exponentially windowed RLS algorithm minimizes the cost function:

$$\varepsilon(n) = \sum_{i=1}^{n} \lambda^{n-i} \left(d(i) - \mathbf{w}^T(n)\mathbf{x}(i)\right)^2.$$

When there is additive noise in the system as given in (1), the RLS updating structure is then given by:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + e(n)\mathbf{k}(n) \tag{6}$$

where

$$e(n) = z(n) - \mathbf{x}^T(n)\mathbf{w}(n-1), \tag{7}$$

and

$$\mathbf{k}(n) \;=\; \mathbf{R}^{-1}(n)\mathbf{x}(n) \tag{8}$$

with the definition

$$\mathbf{R}^{-1}(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i). \tag{9}$$

III. The Fixed-Point RLS Algorithm

When an algorithm is implemented on special-purpose digital hardware or as a software routine, finite register lengths have to be used. Accordingly, the use of finite wordlength introduces errors into the algorithm, the most important being the roundoff errors introduced by the arithmetic operations.

The effect of roundoff errors, i.e. the errors introduced by the quantization of arithmetic operations, depends on whether the arithmetic is performed in floating-point or in fixed-point arithmetic. Since in fixed-point arithmetic addition does not increase the wordlength, roundoff after addition is not required and errors are not introduced, provided that overflow does not occur. On the other hand, fixed-point multiplication does increase the wordlength and the result needs to be rounded or truncated. In floating-point arithmetic both addition and multiplication introduce roundoff errors. In this work, we consider fixed-point implementation of filters. Among different works on the effects of floating-point errors on adaptive filters, reference [6] considers the analysis of floating-point RLS algorithm assuming that the input is white.

It is assumed that the number of bits to be used for the integer part of the finite precision variables is chosen such that overflows do not occur. Preventing overflow by proper scaling of the algorithmic quantities and the inputs is discussed in various works; e.g. in [17, 30], proper scaling mechanisms are given for the RLS algorithm for preventing overflow. Also, it is assumed that quantization errors are negligible compared to the roundoff errors and other noise sources.

Fixed-point roundoff errors are modeled as uniformly distributed zero mean white random processes. This model is commonly used in the analysis of finite precision effects in adaptive filtering algorithms [10, 13, 18, 17, 24]. In [19], the limitations of this assumption is studied for fixed-point

multiplication. It is pointed out that this is a valid assumption provided that the bandwidth and the dynamic range of the quantities that are being multiplied are sufficiently large, i.e. if the variance of the signal that is being multiplied is σ_x^2, then we should have $\sigma_x/\Delta \gg 1$. On the other hand, for signals with small dynamic range and narrow bandwidths, the assumption that white roundoff errors are independent from the data may fail. It is particularly important to remember the limitations of the assumptions that are being made when discussing the analysis results and interpreting the simulation results. Since it is assumed that the roundoff errors are uniformly distributed over the bin width, the variance of this error is given by

$$\sigma^2 = 2^{-2B}/12 \tag{10}$$

where B is the number of bits used to represent the fractional part of the variable used for storing the final value of the operation.

The two important roundoff error terms in the analysis are introduced as follows: the roundoff error vector $\boldsymbol{\mu}(n)$ is due to the scalar-vector multiplication in the weight error update,

$$\mathbf{w}'(n) = \mathbf{w}'(n-1) + e'(n)\mathbf{k}'(n) + \boldsymbol{\mu}(n) \tag{11}$$

and the scalar roundoff error term $\epsilon(n)$ is due the inner product computation in the desired estimate,

$$\hat{d}'(n) = \mathbf{x}'^T(n)\mathbf{w}'(n-1) + \epsilon(n). \tag{12}$$

The N zero mean components of $\boldsymbol{\mu}(n)$ in (11) are assumed to be independent, and to have identical variances $\sigma_\mu^2 = 2^{-2B_\mu}/12$ where B_μ is the number of bits used for the fractional part of the weights. The roundoff error $\epsilon(n)$ satisfy the same assumptions and has variance $c\sigma_\epsilon^2$ where $c = N$ if the individual products are rounded before forming the final inner product, and $c = 1$ if the result is rounded after forming the inner product.

The Kalman gain $\mathbf{k}'(n)$ is assumed to be perturbed by a zero mean white error term $\beta(n)$. This is the quantization error if the Kalman gain is precomputed in infinite precision, or it can be

used to model the roundoff error when the Kalman gain is computed using different RLS algorithms (conventional or fast RLS algorithms). The quantized Kalman gain can then be written as

$$k_i'(n) = k_i(n) + \beta_i(n) \tag{13}$$

where $k_i'(n)$ is the ith entry of the fixed-point gain vector $\mathbf{k}'(n)$. In the literature it has been reported that the RLS algorithm, when implemented in finite precision, is numerically unstable and the weights diverge from their optimum values [23, 26, 29, 33]. The accumulation of roundoff errors is known to cause the divergence of the RLS algorithm, forcing the inverse autocorrelation estimate, $\mathbf{R}^{-1}(n)$, to lose the property of being positive definite [23, 26, 29]. This matrix is directly related to the Kalman gain vector $\mathbf{k}(n)$ as shown in (8). A complete analysis of a full fixed-point implementation of the conventional RLS algorithm is presented in [23] together with a scenario for explaining the reasons for the divergence. In this work, our goal is to find closed form expressions that explain the effect of finite precision errors on the overall system performance. This becomes an overwhelming task when the roundoff errors at each stage are considered [23, 33]. Besides, by assuming that the Kalman gain is perturbed by an additive noise component, the analysis becomes more general, independent of the method used for the computation of the Kalman gain.

IV. Analysis of the Fixed-Point RLS Algorithm

The main performance criterion in the analysis is the weight error vector which is defined as the difference between the fixed-point weight vector estimate at time n, $\mathbf{w}'(n)$, and the system parameters at the same time instant $\mathbf{w}^*(n)$,

$$\boldsymbol{\theta}'(n) = \mathbf{w}'(n) - \mathbf{w}^*(n). \tag{14}$$

It is also possible to select the performance measure as the misadjustment in the weights due to the finite precision adjustment as done in [13] and then use some of the previous results for the infinite precision RLS algorithm. However, doing so would limit the results to the fixed-point case.

With the choice of (14) for the weight error, it is possible to get the weight misadjustment error for the infinite precision case by simply letting the fixed-point roundoff errors go to zero. The additive noise component $v(n)$ and the white noise driving the time varying system weights $\boldsymbol{\alpha}(n)$ will then be the only sources of noise, and the final result showing the effect of correlation on the weight misadjustment will also be valid for the the infinite precision RLS. In [28], the performance criterion is chosen differently, and thus the effect of input correlation on the weights is not observed. In some applications, such as system identification, the instantaneous weights are the desired information and thus it is particularly important to know their variations from the optimum. The instantaneous weight values also supply general information about the operating environment of the adaptive filter, as in the case of digital phase locked loop implementations through the monitoring of shifts in the energy distribution of tap values in the equalizer [25].

To find the expression for the weight error vector (14) by using the fixed-point error model of Section 3.2, the fixed-point weight update equation (11) is written by using the expression for prediction error (1), (7), the model for fixed-point inner product computation (12), and the weight update error model (11):

$$\mathbf{w}'(n) = \mathbf{w}'(n-1) + \mathbf{k}'(n)\left[d(n) - \mathbf{x}'^{T}(n)\mathbf{w}'(n-1)\right] + \eta(n)\mathbf{k}'(n) + \boldsymbol{\mu}(n) \tag{15}$$

where the new zero mean random variable is defined as

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

Next, substituting the fixed-point model (13) and the model for time-varying weights in (4) with $b \rightarrow 1$ into (15), and using the weight error definition, (14), gives

$$\boldsymbol{\theta}'(n) = \left[\mathbf{I} - \mathbf{k}(n)\mathbf{x}^{T}(n) - \boldsymbol{\beta}(n)\mathbf{x}^{T}(n)\right]\boldsymbol{\theta}'(n-1) + \boldsymbol{\psi}(n) + \eta(n)\mathbf{k}(n) + \eta(n)\boldsymbol{\beta}(n)$$

where

$$\boldsymbol{\psi}(n) = \boldsymbol{\mu}(n) - \boldsymbol{\alpha}(n). \tag{17}$$

The weight misadjustment $\varrho(n)$ is defined as the expected value of the inner product of the weight error vector,

$$\begin{aligned} \varrho(n) &= E\left\{\boldsymbol{\theta}'^T(n)\boldsymbol{\theta}'(n)\right\} \\ &= E\left\{\|\boldsymbol{\theta}'(n)\|^2\right\}. \end{aligned} \tag{18}$$

Defining the weight error covariance matrix as

$$\mathbf{R}_{\theta'}(n) = E\left\{\boldsymbol{\theta}'(n)\boldsymbol{\theta}'^T(n)\right\} \tag{19}$$

we have

$$\begin{aligned} \varrho(n) &= Trace\left\{\mathbf{R}_{\theta'}(n)\right\} \\ &= \sum_{i=0}^{N-1} \vartheta_{ii}(n) \end{aligned} \tag{20}$$

where

$$\vartheta_{ij}(n) = E\left\{\theta_i(n)\theta_j(n)\right\}. \tag{21}$$

The error covariance matrix is then evaluated by forming the outer product of (17) and then taking the expected value of the resulting matrix,

$$\begin{aligned} \mathbf{R}_{\theta'}(n) = \quad & \mathbf{R}_{\theta'}(n-1) - E\left\{\mathbf{k}(n)\mathbf{x}^T(n)\right\}\mathbf{R}_{\theta'}(n-1) - \mathbf{R}_{\theta'}(n-1)E\left\{\mathbf{x}(n)\mathbf{k}^T(n)\right\} \\ + \quad & E\left\{\mathbf{k}(n)\mathbf{x}^T(n)\boldsymbol{\theta}'(n-1)\boldsymbol{\theta}'^T(n-1)\mathbf{x}(n)\mathbf{k}^T(n)\right\} + \sigma_\eta^2 E\left\{\mathbf{k}(n)\mathbf{k}^T(n)\right\} \\ + \quad & E\left\{\boldsymbol{\beta}(n)\mathbf{x}^T(n)\boldsymbol{\theta}'(n-1)\boldsymbol{\theta}'^T(n-1)\mathbf{x}(n)\boldsymbol{\beta}^T(n)\right\} + \sigma_\psi^2\mathbf{I} + \sigma_\eta^2\sigma_\beta^2\mathbf{I}. \end{aligned} \tag{22}$$

Since it is assumed that all the noise processes, $\eta(n)$, $\psi(n)$ and $\boldsymbol{\beta}(n)$ are zero mean and independent of the data sequences and from each other, all the terms in (22) involving first order statistics of these processes are equal to zero and have been dropped. An important assumption that has been made in (22) is the independence of the weight error vector with the input data vector. This has been common practice in the analysis of adaptive filtering algorithms [6, 8, 9, 10, 12, 13, 17, 24, 28].The latter is a

sampling from the works included in the refences of this work. Although this assumption does not exactly reflect reality, it is always observed that final predictions made using this assumption are in excellent agreement with experiments and simulations. This is also observed in the final results of this analysis. Reference [2] studies this assumption, and concludes that the perturbation terms neglected by using this assumption may usually be disregarded. Also, since the weight error depends heavily on the past data, especially for forgetting factors very close to 1, it is not much affected by the input data vector at the current time [9]. This is similar to the *averaging principle* of [18] which is discussed in the Appendix.

Assuming that the input sequence is independent and identically distributed, reference [14] shows that, for the LMS algorithm, successive input vectors can be modeled as statistically independent when the counterpart of condition (5) for the LMS algorithm is satisfied. In [31] it is conjectured that the same property holds for the RLS algorithm. Considering (17) it is easy to see that the assumption of independent successive data vectors is the same as the assumption of independent data and error vectors. It is worth noting that satisfying the condition $N(1-\lambda) \ll 1$ is again an important consideration.

Substituting the expressions derived in the Appendix, (50), (51), (55), and (57) for the expected values in equation (22) gives

$$\begin{aligned} \mathbf{R}_{\theta'}(n) = \quad & \left[1 - 2(1-\lambda) + 2(1-\lambda)^2\right] \mathbf{R}_{\theta'}(n-1) + \sigma_\psi^2 \mathbf{I} + \sigma_\eta^2 \sigma_\beta^2 \mathbf{I} + \sigma_\eta^2 (1-\lambda)^2 \mathbf{R}_x^{-1} \\ + \quad & Trace\left\{\mathbf{R}_x \mathbf{R}_{\theta'}(n-1)\right\} \left[(1-\lambda)^2 \mathbf{R}_x^{-1} + \sigma_\beta^2 \mathbf{I}\right]. \end{aligned} \tag{23}$$

Evaluating the weight misadjustment $\varrho(n)$ at steady-state requires the computation of the $Trace$ of $\mathbf{R}_{\theta'}(n)$. Hence, the trace of all the terms in (23). We then get an expression which is function of $\mathbf{R}_{\theta'}(n)$, or its $Trace$, $\varrho(n)$. Therefore, it is preferable to write the $Trace$ of the matrix product in

equation (23) as

$$Trace\left\{\mathbf{R}_x\mathbf{R}_{\theta'}(n-1)\right\} = \sigma_x^2 \sum_{i=0}^{N-1} \vartheta_{i,i}(n-1) + 2\sum_{i=1}^{N-1} r_i \sum_{j=0}^{N-1-i} \vartheta_{j,i+j}(n-1) \tag{24}$$

where $\vartheta_{i,j}$ is defined in (21), and

$$r_i = E\{x(n)x(n-i)\}$$

with

$$r_0 = \sigma_x^2.$$

By defining

$$\gamma(n-1) = \frac{2\sum_{i=1}^{N-1} r_i \sum_{j=0}^{N-1-i} \vartheta_{j,i+j}(n-1)}{\sigma_x^2 \sum_{i=0}^{N-1} \vartheta_{ii}(n-1)} \tag{25}$$

a more compact form can be achieved for the term in (24):

$$Trace\left\{\mathbf{R}_x\mathbf{R}_{\theta'}(n-1)\right\} = \sigma_x^2\left[1+\gamma(n-1)\right]\varrho(n-1). \tag{26}$$

V. The Steady-State Weight Error Vector Norm

To evaluate the norm square of the weight error vector, take the *Trace* of the error covariance matrix in (23) and use the definition (26) and (20) to get

$$\varrho(n) = T(n-1)\varrho(n-1) + D(n-1). \tag{27}$$

where the *transition* term, $T(n-1)$, is defined as

$$T(n-1) = 1 - 2(1-\lambda) + 2(1-\lambda)^2 + \sigma_x^2\left(1+\gamma(n-1)\right)\left((1-\lambda)^2\sum_{i=0}^{N-1}\frac{1}{\rho_i} + N\sigma_\beta^2\right) \tag{28}$$

and the *driving* term, $D(n-1)$ as

$$D(n-1) = N\sigma_\psi^2 + N\sigma_\eta^2\sigma_\beta^2 + \sigma_\eta^2(1-\lambda)^2\sum_{i=0}^{N-1}\frac{1}{\rho_i}. \tag{29}$$

In the above definitions, (28) and (29), ρ_i is the ith eigenvalue of the $N\times N$ true input autocorrelation matrix $\mathbf{R}_x$, assuming that the matrix is nonsingular, i.e. it does not have any zero eigenvalues.

The sum of inverse eigenvalues appearing in (28) and (29) is a very interesting result showing the impact of input correlation on the overall error. The significance and effect of this term will be studied further in the next section. This term involving the eigenvalues results because of the *Trace* evaluation of the inverse input autocorrelation matrix, i.e., by using

$$Trace\{\mathbf{R}_x^{-1}\} = \sum_{i=0}^{N-1} \frac{1}{\rho_i}.$$

At steady-state, i.e. $n \rightarrow \infty$, the error covariance matrix $\mathbf{R}_{\theta'}(n)$, will be slowly fluctuating around its mean [28, 31]. Therefore $\varrho(n)$, the sum of the squares of its diagonal terms, will not change much in one iteration, i.e.

$$\varrho(n) \approx \varrho(n-1) = \varrho. \tag{30}$$

The only time dependent variables in $\gamma(n)$ are the entries of the error covariance matrix $\mathbf{R}_{\theta'}(n)$ which are the coefficients $\vartheta_{ij}(n)$. Since at steady-state these coefficients are slowly varying around their mean values, they can be replaced by their steady-state mean values, hence dropping the time dependence in $\gamma(n)$.

The steady-state error is then easily found to be

$$\varrho = \frac{\sigma_\psi^2 + \sigma_\eta^2\sigma_\beta^2}{2\phi(1-\lambda)}N + \frac{\sigma_\eta^2}{2\phi}(1-\lambda)\sum_{i=0}^{N-1}\frac{1}{\rho_i} \tag{31}$$

where

$$\phi = \lambda - \frac{1}{2}\sigma_x^2(1+\gamma)\left((1-\lambda)\sum_{i=0}^{N-1}\frac{1}{\rho_i} + \frac{\sigma_\beta^2}{1-\lambda}N\right).$$

One important consideration in the final form of the expression given for ϱ in (31) is determining whether the denominator can be zero for some input statistics and driving error values. Since $\lambda < 1$, it is sufficient to consider only the newly defined term ϕ which can apparently go to zero and cause the error ϱ to become infinity. First let the input is uncorrelated, then all the eigenvalues are the

same and equal to σ_x^2. The term ϕ then becomes

$$\phi = \lambda - \frac{1}{2}N(1-\lambda) - \frac{N}{2(1-\lambda)}\sigma_x^2\sigma_\beta^2. \tag{32}$$

The perturbation variance σ_β^2 given by (10) is much smaller that 1. Remembering that λ and the system order N are chosen such that

$$N(1-\lambda) \ll 1$$

and that λ is very close to 1, it is easy to see that the last four terms of ϕ in (32) will be very small compared to the first term λ giving

$$\phi \approx \lambda \approx 1.$$

This is a very good approximation for most cases, which is also observed in simulations. However, it is also important to note that the only term that might be significant and can force ϕ to be zero is the term involving σ_β^2, the term due to the perturbations in the Kalman gain vector. In reference [17] it is shown that this term might lead to the exponential divergence of the algorithm. Now, consider the correlated case for ϕ. In the next section, it will be shown that the term involving the inverse eigenvalues increases with input correlation. So, even though this term has a multiplicative factor of $(1-\lambda)$, it can be expected to be significant in the overall error at high input correlation levels. Because γ is a function of the cross error covariance values ϑ_{ij} as given in (25), it does not favor analytical evaluation. By modeling the input as a first order autoregressive process, and by adjusting the coefficient of the process, as discussed in the simulations in section VIII, the behavior of γ is predicted with correlated inputs. It is observed that γ, which equals 0 for uncorrelated inputs ($r_i = 0$, for $i = 1, 2, \cdots, N-1$ for uncorrelated inputs), takes values close to -1 for correlated inputs. Therefore, it is true that, in most cases, $\phi \approx 1$ both for the correlated and uncorrelated cases, and thus its effect is most often negligible in the overall error with respect to the other terms.

By following the above discussion for ϕ, and by letting $\sigma_\beta^2 \to 0$, (31) becomes

$$\varrho = \frac{\sigma_\psi^2}{2(1-\lambda)} N + \frac{\sigma_\eta^2}{2}(1-\lambda) \sum_{i=0}^{N-1} \frac{1}{\rho_i}. \tag{33}$$

Since in most cases, ϕ is very close to 1, it is not included in the denominator of the above expression.

In (33), the first term reflects the effect of the roundoff error due to weight update recursion and that of the Markov process driving the time varying weights. The second term is due to the additive noise and the roundoff error resulting from the desired signal estimate computation. The overall error clearly indicates a tradeoff in the choice of the forgetting factor λ. It is desirable to decrease the effect of additive noise, which is most often the dominant error source, by choosing λ very close to one. However doing so would increase the misadjustment due to σ_ψ^2, i.e., the error due to the weight roundoff errors and the time variance of the system. The forgetting factor is chosen less than 1 in order to be able to track variations in the channel. As noted before, it has to be chosen such that the channel characteristics do not change much during the memory of the algorithm, $1/(1-\lambda)$. When λ is increased, approaching 1, the algorithm can not easily *forget* the past data and thus can not emphasize the most recent data. This results in poor tracking characteristics. This is observed in the first term of (33) which indicates an increased misadjustment due to time varying system weights, when λ is close to 1. This result has also been reported in [21] and [28] where the same model of first order Markov process has been used for the time varying weights. Here, it is shown that the roundoff error noise due to weight update affects the overall error in the same way. This property is exploited in [3] to propose a new stabilization technique for the prewindowed RLS algorithm, i.e. when $\lambda = 1$. We delay the discussion of the input correlation effects on the weight misadjustment and its interaction with other noise sources until the end of next chapter where we define a new quantity: the *correlation amplification factor.*

VI. Correlation Amplification Factor

In the expressions for ϱ, mean of the weight error norm, we have a very interesting term, the sum of the inverse eigenvalues of the input autocorrelation matrix. This is an original term reflecting the impact of input correlation on the overall error. To study the characteristics of this new term, define a new quantity v, the *correlation amplification factor* as

$$v = \frac{\sigma_x^2}{N} \sum_{i=1}^{N} \frac{1}{\rho_i}. \tag{34}$$

Note that this is a normalized quantity which reduces to 1 when the input is uncorrelated.

The characteristics of this new quantity, v, as a function of signal correlation is analyzed by modeling the input as a first order autoregressive (AR) process with AR constant a, i.e.

$$x(n) = ax(n-1) + u(n) \tag{35}$$

where $u(n)$ is a white random process. Then, for a system of order N the input autocorrelation coefficients are

$$r_i = \sigma_x^2 a^i \quad \text{for} \quad i = 0, 1, \cdots, N-1$$

where $\sigma_x^2 = \sigma_u^2/(1-a^2)$. It can be shown that for this autocorrelation matrix characteristics, the correlation amplification factor v is given by

$$v = \frac{1 + \frac{N-2}{N} a^2}{1 - a^2}. \tag{36}$$

Thus, the correlation amplification factor increases when the input correlation increases, i.e., as a approaches 1. In this case, the correlation coefficient $r_i = \sigma_x^2 a^i$ becomes significant even for larger i values. Increasing the system order also tends to increase v. However, it should also be noted that since $(N-2)/N \to 1$ as N increases, for large N, the system order ceases to be important. In Figure 1, the amplification factor v is plotted as a function of the AR constant a, for second, third, fourth, and fifth order systems showing the discussed characteristics.

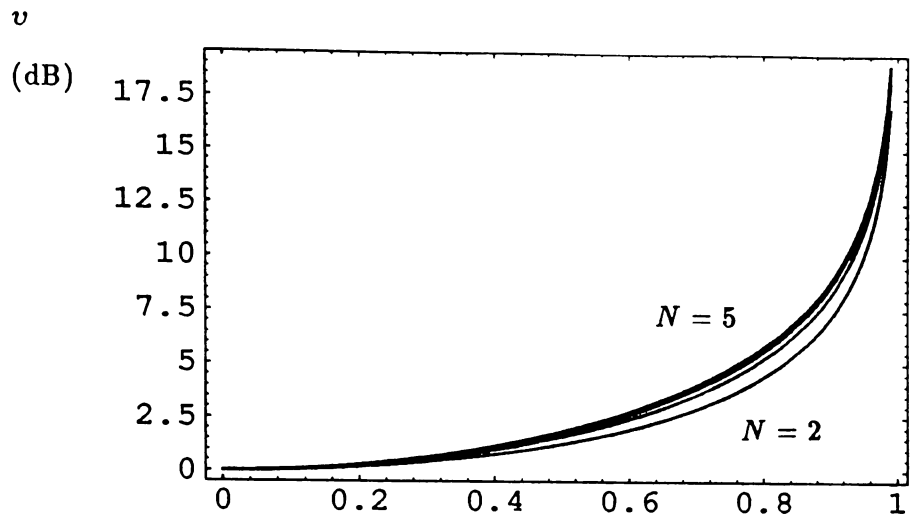

Figure 1: The Correlation Amplification Factor

Returning to the final form of the expression for the weight error misadjustment (33), it is seen that the excess noise due to the additive noise $v(n)$ and the roundoff error term $\epsilon(n)$, i.e. the error term $\eta(n)$, is dramatically enhanced by high input correlation. It is also of interest to mention that the condition number of the autocorrelation matrix, defined as the ratio of the maximum eigenvalue to the minimum [11], alone does not provide sufficient information to determine the impact of correlation on the overall error. It is the sum of inverse eigenvalues that determines the amplification of the noise $\eta(n)$. In the simulations, this fact will be emphasized by selecting two inputs whose autocorrelation matrices have the same condition number but have different eigenvalue distributions. By examining the original expression for ϱ in (31), and remembering that ϕ is not very effective in the overall error, it is seen that input statistics interact mainly with the additive noise component, and $\epsilon(n)$, i.e. $\eta(n)$, whereas it has almost no effect on the other noise sources.

VII. Termination of Updating

Our analysis assumes that the algorithm does not stop updating. However, because of the finite precision arithmetic involved, the algorithm might terminate updating before completely converging, i.e. reaching its steady-state value given in (33). This effect has also been called the *stalling phenomenon* [30] and occurs whenever the correction term $k_i'(n)e'(n)$ for the ith tap weight in the update equation (11) is smaller in magnitude than the *least significant bit* of the tap weight:

$$|k_i'(n_T)e'(n_T)| \leq 2^{-B_\mu - 1} \tag{37}$$

where n_T is the time instant at which ith tap stops updating. This is similar to the condition given in [24] for the LMS algorithm.

When the algorithm stops updating, the major error source will be the random variable $\alpha(n)$ driving the time-varying weights, and the noise source $\eta(n)$ consisting of additive noise and the roundoff error noise $\epsilon(n)$. Therefore,

$$\sigma_{e'}^2(n) \approx N\sigma_x^2\sigma_\alpha^2 + \sigma_\eta^2 \tag{38}$$

when the algorithm stops updating.

When the input is correlated, each of the entries in the Kalman gain vector takes different values. Therefore in order to have an idea about the factors affecting stalling in the RLS algorithm, it is preferable to consider the approximate condition

$$\mathbf{k}'^T(n_T)\mathbf{k}'(n_T)e'^2(n_T) = \sum_{i=0}^{N-1} k_i'^2(n_T)e'^2(n_T) \leq N2^{-2B_\mu - 2}. \tag{39}$$

for the complete stop of updating. As a first degree approximation, consider the expected value of the term in (39). Upon using (51) derived in the Appendix, and taking its *Trace*, the condition becomes

$$\frac{1-\lambda}{1-\lambda^{n_T+1}} \sum_{i=0}^{N-1} \frac{1}{\rho_i}\sigma_{e'}^2 + N\sigma_\beta^2\sigma_{e'}^2 \leq N2^{-2B_\mu - 2}. \tag{40}$$

Note that instead of the steady-state mean for the autocorrelation estimate given in (49), the original expectation given in (48) has been used. This is necessary, since the parameter to be determined is the time index n_T.

By simple manipulations, one gets the following expressions for n_T, the number of iterations after which the algorithm stops updating:

$$n_T \approx \frac{1}{\ln \lambda} \ln \left[1 - \sigma_{e'} 2(1-\lambda) \left(\frac{\sum_{i=1}^{N} \frac{1}{\rho_i}}{N 2^{-2B_\mu - 2} - N\sigma_{e'}^2 \sigma_\beta^2)} \right)^{1/2} \right]. \tag{41}$$

For $\lambda = 1$,

$$E\left\{ \mathbf{k}(n)\mathbf{k}^T(n) \right\} = \frac{1}{n^2} \mathbf{R}_x^{-1} \tag{42}$$

and the resulting expression for n_T is given by

$$n_T \approx \left(\frac{(N\sigma_x^2\sigma_\alpha^2 + \sigma_\eta^2) \sum_{i=1}^{N} \frac{1}{\rho_i}}{N(2^{-2B_\mu - 2} - (N\sigma_x^2\sigma_\alpha^2 + \sigma_\eta^2)\sigma_\beta^2)} \right)^{1/2}. \tag{43}$$

A large driving noise variance σ_α^2, or the presence of a considerably large additive noise delays the termination of updating, or if they are sufficiently large, the argument of the *ln* function in (41), or the argument of the square root operation in (43) becomes negative, indicating that the condition given in (37) is never satisfied, and the algorithm does not stop updating. This argument suggests the calculation of a limiting value for the forgetting factor, which ensures that the algorithm never stalls, which is given by

$$\lambda < 1 - \left(\frac{N(2^{-2B_\mu - 2} - (N\sigma_x^2\sigma_\alpha^2 + \sigma_\eta^2)\sigma_\beta^2)}{(N\sigma_x^2\sigma_\alpha^2 + \sigma_\eta^2) \sum_{i=1}^{N} \frac{1}{\rho_i}} \right)^{1/2}. \tag{44}$$

Recalling the discussion for the correlation amplification factor v in Section 3.5, in (41) and (43), it can be easily seen that updating will stop later when the input signal is highly correlated. Considering (41) and (43) we observe that the stop of updating is also delayed by choosing longer register lengths for representing the fractional part of the weights.

VIII. Simulation Results

The input used in the simulations is a first order AR process, with AR constant a. The innovation is a white Gaussian random process. By adjusting the AR constant, in a range between 0 and 1, the eigenvalue spread of the input can be changed. The input is always normalized such that $\sigma_x^2 \approx 1$ in order to keep its dynamic range approximately the same. This is necessary, because the analysis assumes that the weights do not stop updating, and when the variance of the process decreases with increasing a, the weights might stop updating.

A simple system identification problem is considered, and is simulated by using the block diagram system simulation package *Capsim* [7]. The unknown system output is corrupted by an additive noise term, $v(n)$, and the system order is chosen as 7. The impulse response of the initial channel characteristics is described by the raised cosine:

$$w_i^*(0) = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{2\pi}{W}(i-4)\right)\right], & i = 1, 2, \cdots, 7 \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

where the parameter W controls the amount of amplitude distortion introduced by the channel.

In the simulations, the inner product for the desired signal estimate $\hat{d}(n)$ is computed by first forming the inner product and then quantizing the result. Therefore, the variance of the roundoff error $\epsilon(n)$ is σ_ϵ^2. The control parameter W of the channel is chosen as 2.9 [30].

A time-varying system driven by a noise source of variance 10^{-6} is considered, and the additive noise variance is 10^{-2}. The fractional part of the weights is represented by 9 bits, those of the Kalman gain with 12 bits, and those of the data and the desired signal with 14 bits. The variance of the additive noise is chosen such that its effect is dominant in the overall error ϱ, and this large variance also ensures that the algorithm does not stop updating. The forgetting factor λ is chosen as 0.99, satisfying $N(1-\lambda) \ll 1$ for $N = 7$. The AR constant is increased from 0 to 0.9995 thus increasing the correlation, i.e. the eigenvalue spread of the input.

a	$\kappa(\mathbf{R}_x)$	Theory	Simulation
0.0	1	8.44×10^{-4}	$(8.17 \pm 1.28) \times 10^{-4}$
0.4	5	9.63×10^{-4}	$(9.18 \pm 1.59) \times 10^{-4}$
0.6	12	1.20×10^{-3}	$(1.15 \pm 0.21) \times 10^{-3}$
0.8	40	1.96×10^{-3}	$(1.89 \pm 0.36) \times 10^{-3}$
0.9	106	3.51×10^{-3}	$(3.42 \pm 0.66) \times 10^{-3}$
0.93	165	4.85×10^{-3}	$(4.67 \pm 0.96) \times 10^{-3}$
0.96	316	8.21×10^{-3}	$(7.89 \pm 1.66) \times 10^{-3}$
0.99	1350	3.17×10^{-2}	$(3.05 \pm 0.74) \times 10^{-2}$
0.995	2600	6.30×10^{-2}	$(6.04 \pm 1.25) \times 10^{-2}$
0.999	8155	0.31	0.30 ± 0.063
0.9995	10423	0.63	0.60 ± 0.12

Table 1: Theory and Simulation Comparison for ϱ with increasing input correlation

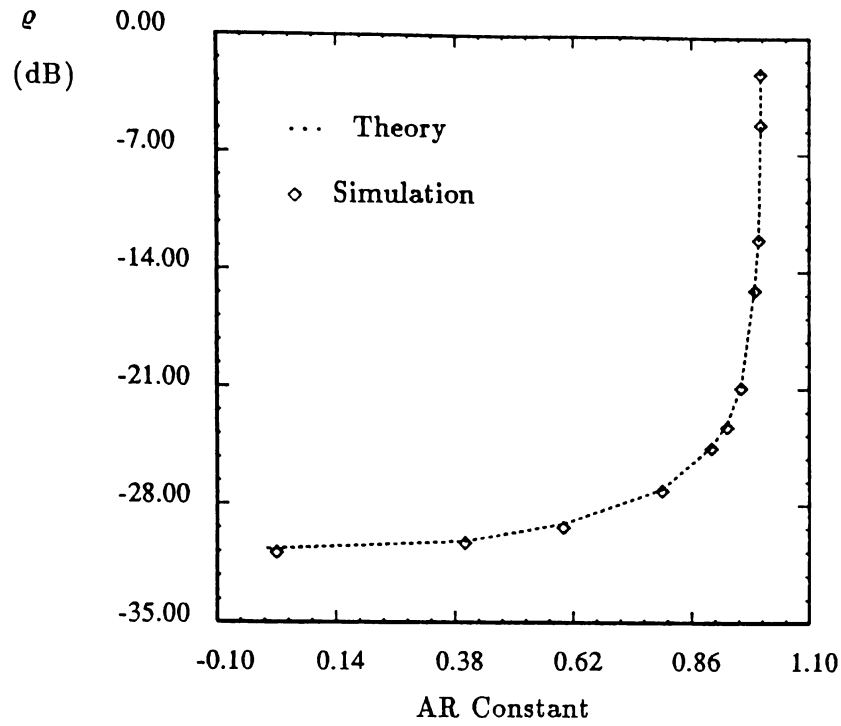

Figure 2: Simulation and Theoretical Results for ϱ as a Function of the AR Constant

The eigenvalue spread of a matrix is usually represented by its 2-norm condition number $\kappa(\mathbf{R}_x)$. This is given by ρ_{max}/ρ_{min} for a symmetric matrix where $\rho_{\max}$ and ρ_{min} are the maximum and the minimum eigenvalues of the matrix. Therefore by changing the AR constant, the condition number has been increased from 1 to 10423, as shown in Table 1. For each AR constant value, 45 independent simulations are done, and the results of these runs have been averaged to get the simulation results given in Table 1. The theoretical results are obtained by evaluating the error expression given in (33). The mean simulation results are given together with the deviation of the 45 independent simulation results. In Table 1, a perfect match between the theoretical (33) and simulation values for ϱ is observed.

Figure 2 shows the results of Table 1 plotted as a function of the AR constant again in dBs, and the same characteristics observed in Figure 1 is seen here, because the effect of the correlation amplification factor v is dominant in the overall error with the chosen parameters. In the figure,

the diamond markers show the simulation values, and the dotted lines join the theoretical values calculated by using (33).

Although in Table 1, it is the condition number of the input autocorrelation matrix that is shown as a measure of the input correlation, our analysis shows that the condition number alone does not provide adequate information on the impact of correlation on the error, ϱ. In order to emphasize the importance of the correlation amplification factor, v, the following experiment has been set up. A system identification problem is simulated by using two different inputs: a Gaussian input, and a sinusoidal input observed in a white Gaussian noise background. The unknown system is stationary, i.e.$\sigma_\alpha^2 = 0$. Both inputs have a condition number of 1000 for each independent run. Because of the presence of the sinusoidal component, the eigenvalues of the second input are distributed such that most of them are close to ρ_{min}. Both inputs have the same dynamic range, a variance of unity, and the parameters used for this case are the same with the first simulation, except for the number of bits used for the fraction of the weights which is 8 for this case, and the additive noise variance which is 10^{-4}. The result shown in Figure 3 is the average of 10 independent runs with each input. The top error curve belongs to the sinusoidal input and the bottom to the Gaussian input. Because the eigenvalues of the sinusoidal input are distributed such that most of them are closer to ρ_{min}, the error floor for the sinusoidal input is higher even though the two inputs have the same condition number. A sample set of eigenvalues for the two inputs are shown in Table 2 for comparison of their distributions. Note that the condition number for both sets of the eigenvalues is approximately 1000 whereas the correlation amplification factor v is equal to 57.329 for the Gaussian input, and 150.571 for the sinusoidal input. This results in approximately $4dB$ difference in the overall error, ϱ, of the two inputs, as the additive noise is the dominant error source in the simulation. Noting that this is just one input sample from the independent runs, it is seen that the approximate difference between the two error floors in Table 2 is about $4dB$.

Another simulation example is set up as follows: The initial values of the time-varying system

Sinusoidal Input with Gaussian Noise	Gaussian Input
4.3089	7.8199
2.6962	0.1445
0.007585	0.04021
0.005218	0.01940
0.004194	0.01274
0.004109	0.009950
0.004030	0.007195

Table 2: Eigenvalue Distribution for Two Different Inputs

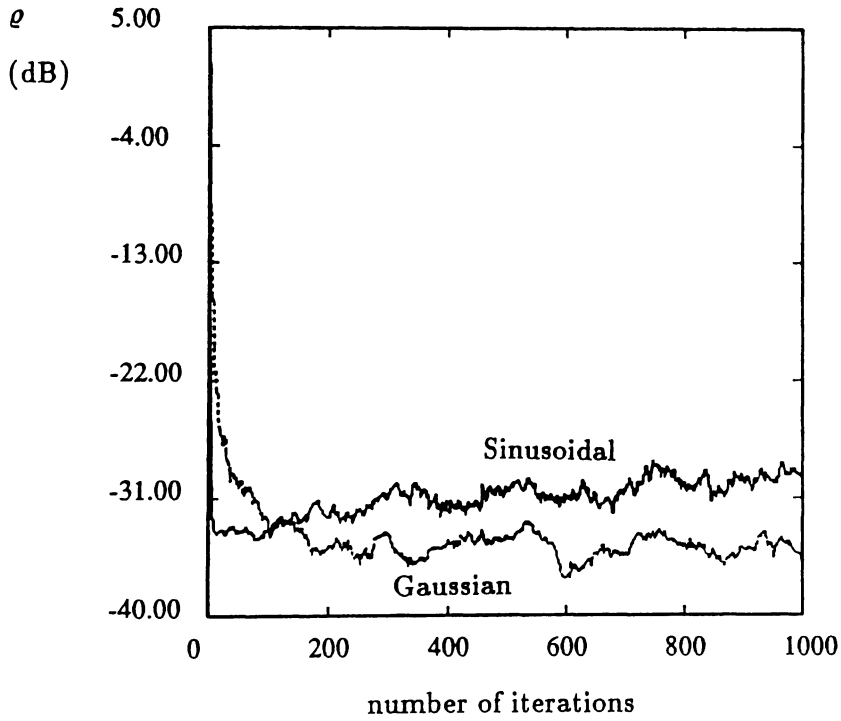

Figure 3: The weight misadjustment, ϱ, for Two Different Inputs with $\kappa(\mathbf{R}_x) = 1000$

B_μ	$\lambda = 0.97$		$\lambda = 0.98$	
	Theory	Simulation	Theory	Simulation
7	3.244×10^{-3}	$(3.028 \pm 0.333) \times 10^{-3}$	2.717×10^{-3}	$(2.329 \pm 0.348) \times 10^{-3}$
8	2.747×10^{-3}	$(2.665 \pm 0.337) \times 10^{-3}$	2.000×10^{-3}	$(1.964 \pm 0.275) \times 10^{-3}$
9	2.622×10^{-3}	$(2.561 \pm 0.321) \times 10^{-3}$	1.820×10^{-3}	$(1.811 \pm 0.242) \times 10^{-3}$
10	2.591×10^{-3}	$(2.526 \pm 0.315) \times 10^{-3}$	1.775×10^{-3}	$(1.766 \pm 0.256) \times 10^{-3}$
11	2.584×10^{-3}	$(2.514 \pm 0.319) \times 10^{-3}$	1.764×10^{-3}	$(1.758 \pm 0.253) \times 10^{-3}$
12	2.581×10^{-3}	$(2.514 \pm 0.318) \times 10^{-3}$	1.761×10^{-3}	$(1.751 \pm 0.253) \times 10^{-3}$

Table 3: Theory and Simulation Comparison for ϱ for $\lambda = 0.97$ and $\lambda = 0.98$

impulse response are the same ones used in the first simulation example. These system taps are driven by a noise source of variance $\sigma_\alpha^2 = 10^{-6}$, and the additive noise variance, σ_v^2, is 2.5×10^{-3}. For the fixed-point representation, 12 bits are used for the fractional part of the Kalman gain, and 14 for the input. The input is a first order AR process with AR constant 0.9, and has a condition number of approximately 100 for each independent run. For each data point, the ensemble average of 50 independent simulations is taken. The number of bits used for representing the fractional part of the weights is increased from 7 to 12. This is repeated for 4 different forgetting factor values, 0.97, 0.98, 0.99, and 0.999.

The results of this simulation are given in Table 3 and Table 4, again showing a close match with the theory. The results for the cases $\lambda = 0.99$, $\lambda = 0.98$ and $\lambda = 0.97$ are plotted in Figure 4 where the top curve is for $\lambda = 0.97$ and the bottom for $\lambda = 0.99$. The small deviations from the theory can be easily explained by remembering the limitations of the models used, and by means of an important finite precision effect: the termination of updating which is discussed in Section VII. In this simulation, the additive noise is not very large, therefore it is expected that the algorithm stops updating when the forgetting factor is very close to one, and when less precision is used for

	$\lambda = 0.99$		$\lambda = 0.999$	
B_μ	Theory	Simulation	Theory	Simulation
7	2.965×10^{-3}	$(1.987 \pm 0.369) \times 10^{-3}$	2.145×10^{-2}	$(1.391 \pm 0.312) \times 10^{-2}$
8	1.581×10^{-3}	$(1.395 \pm 0.227) \times 10^{-3}$	8.052×10^{-3}	$(11.020 \pm 1.988) \times 10^{-3}$
9	1.235×10^{-3}	$(1.214 \pm 0.229) \times 10^{-3}$	4.702×10^{-3}	$(7.321 \pm 1.197) \times 10^{-3}$
10	1.149×10^{-3}	$(1.168 \pm 0.197) \times 10^{-3}$	3.865×10^{-3}	$(4.699 \pm 0.527) \times 10^{-3}$
11	1.127×10^{-3}	$(1.137 \pm 0.204) \times 10^{-3}$	3.655×10^{-3}	$(3.626 \pm 0.371) \times 10^{-3}$
12	1.122×10^{-3}	$(1.133 \pm 0.198) \times 10^{-3}$	3.603×10^{-3}	$(3.364 \pm 0.267) \times 10^{-3}$

Table 4: Theory and Simulation Comparison for ϱ for $\lambda = 0.99$ and $\lambda = 0.999$

the weights. In almost all cases, for $B_\mu = 7$, there is some discrepancy between the theory and simulation. The only case where this difference is small is when the forgetting factor is 0.97. Because for $\lambda = 0.97$, the estimates become quite noisy, and the weights keep updating. However, one other point to remember is that the uniformly distributed roundoff error model holds better for larger register lengths, i.e. for $B_\mu \geq 8$ bits. It is also important to observe that, when the forgetting factor is very close to 1 (for $\lambda = 0.999$ in this example), algorithm stops updating more easily, and the overall error, ϱ, is larger because the algorithm can not track the variations in the system taps. The memory of the system $1/(1-\lambda)$ is very large, and the effect of the noise driving the weights is amplified as shown in (33). When the forgetting factor is decreased, the effect of the additive noise component becomes more pronounced as predicted in (33). This is observed for $\lambda = 0.97$, where the size of weight register B_μ is no longer effective, especially for $B_\mu \geq 9$.

Close inspection of the mean error values in Tables 3 and 4 show that the error, ϱ, first gradually decreases with increasing λ, and suddenly increases when λ is chosen as 0.999. In fact, it can be easily shown that, for

$$\lambda_{opt} = 1 - \frac{\sigma_\psi \sqrt{N}}{\sigma_\eta \sqrt{\sum_{i=1}^{N} \frac{1}{\rho_i}}}$$

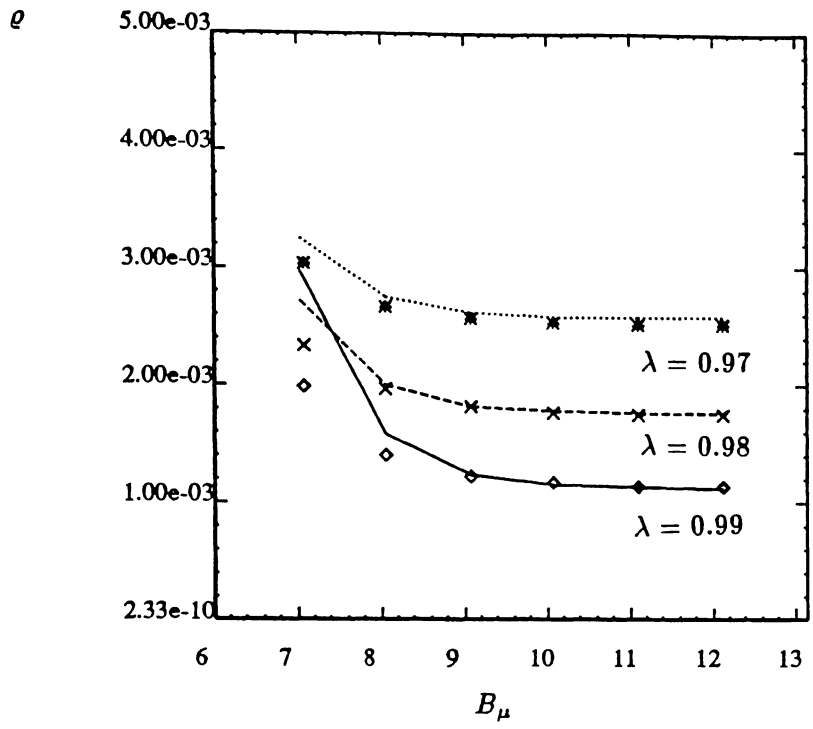

Figure 4: Change in ϱ as a function of B_μ, with $\lambda = 0.97, 0.98$, and 0.99

the weight error ϱ given in (33) attains its minimum. For λ_{opt}, the two terms in (33) defining the tradeoff become equal. This results in a final error of

$$\varrho = \sigma_\psi \sigma_\eta \sqrt{N \sum_{i=1}^{N} \frac{1}{\rho_i}} \tag{46}$$

which becomes

$$\varrho = \frac{N \sigma_\psi \sigma_\eta}{\sigma_x}$$

when the input is uncorrelated.

For the simulation example of Table 5, when $B_\mu = 10$, the optimum forgetting factor becomes $\lambda_{opt} = 0.9931$. In Table 5, the error ϱ is given as a function of forgetting factor, and it is plotted in Figure 5. In the figure, the solid line shows the simulation results for ϱ, and the dotted line is drawn through the values calculated by using (33). Note that the optimum occurs at $\lambda \approx 0.993$.

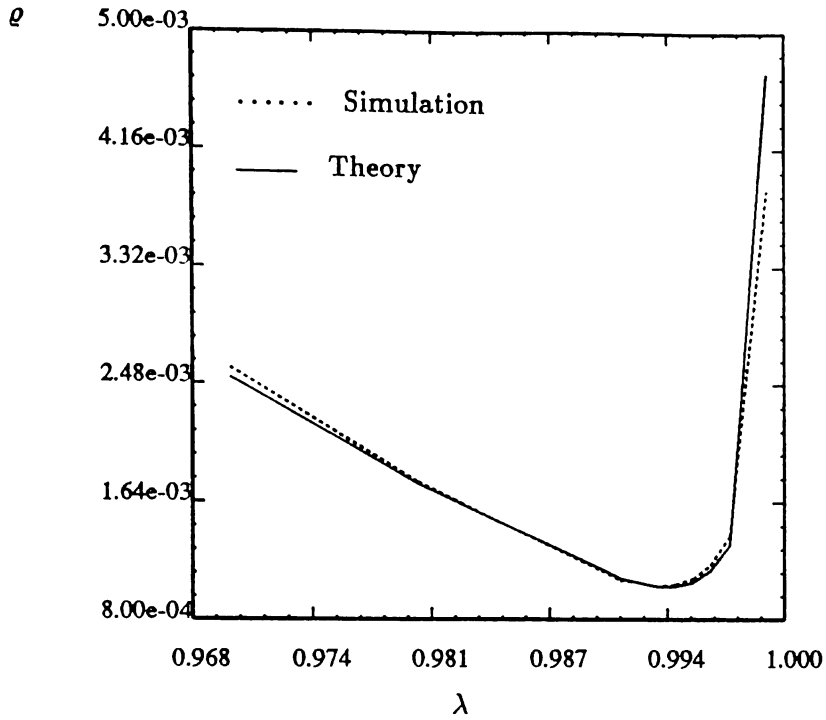

Figure 5: Change in ϱ as a function of λ

λ	Theory	Simulation
0.97	2.5910×10^{-3}	$(2.5260 \pm 0.3150) \times 10^{-3}$
0.98	1.7750×10^{-3}	$(1.7660 \pm 0.2560) \times 10^{-3}$
0.99	1.1490×10^{-3}	$(1.1680 \pm 0.1970) \times 10^{-3}$
0.991	1.0829×10^{-3}	$(1.0932 \pm 0.1309) \times 10^{-3}$
0.992	1.0613×10^{-3}	$(1.0613 \pm 0.1899) \times 10^{-3}$
0.993	1.0397×10^{-3}	$(1.0447 \pm 0.1954) \times 10^{-3}$
0.994	1.0469×10^{-3}	$(1.0448 \pm 0.1611) \times 10^{-3}$
0.995	1.0881×10^{-3}	$(1.0717 \pm 0.1771) \times 10^{-3}$
0.996	1.1884×10^{-3}	$(1.1528 \pm 0.2101) \times 10^{-3}$
0.997	1.4065×10^{-3}	$(1.3356 \pm 0.3210) \times 10^{-3}$
0.999	3.8650×10^{-3}	$(4.6990 \pm 0.5270) \times 10^{-3}$

Table 5: Theory and Simulation Comparison for ϱ as a Function of λ

IX. Conclusion

New expressions are derived for the mean weight misadjustment in the fixed-point recursive least squares (RLS) algorithm. These expressions are general in that they take into account the correlation in the input as well as the noise driving the time-varying weights. These expressions show the effect of each individual noise source on the overall error and those of input correlation and system parameters. The effect of input correlation is analyzed by defining a new quantity called the correlation amplification factor. It is shown that input correlation amplifies the excess error due to additive noise, and roundoff noise in the desired signal estimate computation. However, correlation has almost no effect on the noise due to time-varying system weights and the noise due to weight update recursion. It is also demonstrated that tracking noise and weight update roundoff error behave similarly, and contributions from both increase proportional to the time constant of the algorithm, $1/(1-\lambda)$. The final error expression also indicates a tradeoff in the choice of the forgetting factor. The effect of the additive noise, which is most often the dominant error source, can be decreased by choosing λ very close to 1. However, increasing λ amplifies the noise due to weight update and the error driving the time-varying system weights. An optimum forgetting factor which minimizes the error is also found. Finally, conditions are determined for the termination of updating which is an important problem in the finite precision implementation of adaptive filters. It is shown that, increasing input correlation tends to keep the weights active, i.e., updating. Using large register lengths for the weights, high additive noise, and rapidly changing system taps also tend to delay or prevent stop of updating, as we would expect.

APPENDIX

In this Appendix, we present the evaluations of the expectations to be used in the derivation of the final mean of the weight error norm, ϱ. The expectations derived in this Appendix are used in (22).

The evaluation of $E\left\{\mathbf{k}(i)\mathbf{x}^T(i)\right\}$ *and* $E\left\{\mathbf{k}(i)\mathbf{k}^T(i)\right\}$:

In [31], it is shown that for the condition given in (5), at steady-state, the autocorrelation estimate $\mathbf{R}(i)$ is a quasi-deterministic matrix, i.e.

$$E\left\{\left[\mathbf{R}(i) - E\left\{\mathbf{R}(i)\right\}\right]^2\right\} \ll \left[E\left\{\mathbf{R}(i)\right\}\right]^2. \tag{47}$$

This analysis is done for the recovery of a chirped sinusoid in noise, but it is conjectured that the same property holds for the RLS algorithm with stationary inputs. Also, in [28] the authors model $\mathbf{R}(i)$ as the sum of two matrices, a mean matrix and a zero mean symmetric perturbation matrix. It is then shown that the second moment of this zero mean perturbation matrix is negligible for Gaussian inputs. The behavior of the autocorrelation matrix estimate $\mathbf{R}(i)$ is also checked by simulations, and it is observed that the norm of the perturbation matrix is always much smaller than that of the mean matrix. Therefore, by using the same approach as in [31], $\mathbf{R}^{-1}(i)$ is simply replaced by its steady-state mean value which is

$$E\{\mathbf{R}^{-1}(i)\} = \frac{1-\lambda^{i+1}}{1-\lambda}\mathbf{R}_x^{-1} \tag{48}$$

for a stationary input. Here, it is assumed that the input autocorrelation matrix is nonsingular, which is the case in most practical systems [30]. The same approach, replacing the autocorrelation estimate by its mean value at steady-state is also justified by the averaging principle of [18]. In this case it is stated that when one of the two jointly stationary processes is slowly varying with respect to the other process, they are *almost* independent. The autocorrelation estimate is the sum of input vector outer products which definitely is a slowly varying term with respect to the other

terms, especially for λ close to 1. Therefore it is *almost* independent of the other terms, such as the instantaneous input vector outer product.

At steady-state, for large i, the mean of the the input autocorrelation estimate becomes

$$\lim_{i\to\infty} E\{\mathbf{R}^{-1}(i)\} = (1-\lambda)\mathbf{R}_x^{-1} \tag{49}$$

since $\lambda < 1$.

By using the definition of the true input autocorrelation matrix for wide sense stationary inputs,

$$\mathbf{R}_x = E\left\{\mathbf{x}(i)\mathbf{x}^T(i)\right\}$$

the two required expectations can be written as

$$\begin{aligned} E\left\{\mathbf{k}(i)\mathbf{x}^T(i)\right\} &= E\left\{\mathbf{R}^{-1}(i)\mathbf{x}(i)\mathbf{x}^T(i)\right\} \\ &= (1-\lambda)\mathbf{R}_x^{-1} E\left\{\mathbf{x}(i)\mathbf{x}^T(i)\right\} \\ &= (1-\lambda)\mathbf{I} \end{aligned} \tag{50}$$

and

$$\begin{aligned} E\left\{\mathbf{k}(i)\mathbf{k}^T(i)\right\} &= E\left\{\mathbf{R}^{-1}(i)\mathbf{x}(i)\mathbf{x}^T(i)\mathbf{R}^{-1}(i)\right\} \\ &= (1-\lambda)^2\mathbf{R}_x^{-1} E\left\{\mathbf{x}(i)\mathbf{x}^T(i)\right\}\mathbf{R}_x^{-1} \\ &= (1-\lambda)^2\mathbf{R}_x^{-1}. \end{aligned} \tag{51}$$

The evaluation of $E\left\{\mathbf{k}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\mathbf{k}^T(i)\right\}$:

Using the definition of Kalman gain and the quasi-deterministic property of $\mathbf{R}(i)$, (47), the expectation term to be evaluated becomes

$$\begin{aligned} &E\left\{\mathbf{k}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\mathbf{k}^T(i)\right\} = \\ &\qquad (1-\lambda)^2\mathbf{R}_x^{-1} E\left\{\mathbf{x}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\mathbf{x}^T(i)\right\}\mathbf{R}_x^{-1}. \end{aligned} \tag{52}$$

The expectation in (52) involves fourth order input statistics, and can be evaluated as follows, assuming that the input is Gaussian:

The statistical properties of a Gaussian process $g(n)$ are uniquely determined if its mean and autocorrelation are known [32]. In particular, a moment M of any order for a Gaussian process can be expressed in terms of these two statistics.

For jointly Gaussian processes with zero mean, the fourth order moment can be written as [32]

$$E\{g_1 g_2 g_3 g_4\} = E\{g_1 g_2\}E\{g_3 g_4\} + E\{g_1 g_3\}E\{g_2 g_4\} + E\{g_1 g_4\}E\{g_2 g_3\}. \tag{53}$$

If the input is not zero mean, the random process in the above expression is replaced by the centered process $g(n) - E\{g(n)\}$.

Since the samples of the Gaussian input process $x(n)$ at different time instants will be jointly Gaussian, the expectation in matrix (52) can be written as

$$\begin{aligned} E\left\{\mathbf{x}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\mathbf{x}^T(i)\right\} &= 2\mathbf{R}_x\mathbf{R}_{\theta'}(i-1)\mathbf{R}_x \\ &+ Trace\left\{\mathbf{R}_x\mathbf{R}_{\theta'}(i-1)\right\}\mathbf{R}_x \end{aligned} \tag{54}$$

by following a procedure similar to the one in [8].

The final expression for the expectation then becomes

$$\begin{aligned} &E\left\{\mathbf{k}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\mathbf{k}^T(i)\right\} = \\ &\qquad (1-\lambda)^2\left(2\mathbf{R}_{\theta'}(i-1) + Trace\left\{\mathbf{R}_x\mathbf{R}_{\theta'}(i-1)\right\}\mathbf{R}_x^{-1}\right). \end{aligned} \tag{55}$$

The evaluation of $E\left\{\boldsymbol{\beta}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\boldsymbol{\beta}^T(i)\right\}$

The perturbation error $\beta_i(n)$ is independent of the input $x(n)$ and hence of the error vector $\boldsymbol{\theta}'(n)$. Noting the presence of a scalar term which is independent of $\boldsymbol{\beta}(n)$ in the required expectation,

it is rewritten as

$$E\left\{\boldsymbol{\beta}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\boldsymbol{\beta}^T(i)\right\} = \sigma_\beta^2 E\{\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\}\,\mathbf{I}. \qquad (56)$$

Upon using the commutative property of the *Trace* operator and the fact that *Trace* of a scalar quantity is equal to itself, the above equality takes the final form

$$E\left\{\boldsymbol{\beta}(i)\mathbf{x}^T(i)\boldsymbol{\theta}'(i-1)\boldsymbol{\theta}'^T(i-1)\mathbf{x}(i)\boldsymbol{\beta}^T(i)\right\} = \sigma_\beta^2\, Trace\{\mathbf{R}_x\mathbf{R}_{\theta'}(i-1)\}\,\mathbf{I}. \qquad (57)$$

In getting the final expression of (57) we have used the commutativity of the two linear operators; *Trace* and *Expectation*, as well as the independence assumption of the input and error vectors by following the discussion of Section IV.

REFERENCES

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

[2] J. E. Mazo, "On the independence theory of equalizer convergence," *Bell Syst. Tech. J.*, vol 58, no. 5, pp. 963-993, May-June 1979.

[3] T. Adalı and S. H. Ardalan,"Analysis of a Stabilization Technique for the Fixed-point Prewindowed RLS Algorithm," *IEEE Trans. on Signal Processing*, vol. ASSP-39, no. 9, pp. 2079-2082, Sept. 1991.

[4] T. Adalı and S. H. Ardalan,"Fixed-Point Roundoff Error Analysis of the RLS Algorithm with Time-Varying Channels," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Toronto, Ontario, Canada), May 1991, pp. 1865-1868.

[5] T.Adalı, and S.H.Ardalan, "Convergence and error analysis of the fixed point RLS algorithms with correlated inputs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Albuquerque, NM), April 1990, pp. 1479-1482.

[6] 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.

[7] S.H. Ardalan, "A Graphical Simulation Tool for Communication Systems," *XCAD Corp.*, P.O. Box 5861, Raleigh, NC, 27650.

[8] N. J. Bershad, "Analysis of the Normalized LMS algorithm with Gaussian Inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 793-806, Aug. 1986.

[9] N. J. Bershad, "On the probability density function of the LMS adaptive filter weights," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, no. 1, pp. 43-56, Jan. 1989.

[10] V. J. Mathews, Z. Xie, "Fixed-point error analysis of stochastic gradient adaptive lattice filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, no. 1, pp. 70-80, Jan. 1990.

[11] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Raleigh, N.C.:, 1990

[12] G. Ungerboeck, "Theory on the speed of convergence in adaptive equalizers for digital communication," *IBM J. Res Develop.*, pp. 546-555, Nov. 1972.

[13] S. T. Alexander, "Transient weight misadjustment properties for the finite precision LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 9, pp. 1250-1258.

[14] S. Marcos, and O. D. Macchi, "Tracking capability of the Least Mean Square algorithm: Application to an asynchronous echo canceler," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 11, pp. 1570-1578, Nov. 1987.

[15] B. Farhang-Boroujeny, and S. Gazor, "Performance analysis of transform domain normalized LMS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Toronto, Ontario, Canada), pp. 2133-2136, May 1991.

[16] 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.

[17] 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.

[18] C. Samson, and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 6, pp. 1177-1191, Oct. 1983.

[19] 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.

[20] M. G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, New York, N.Y.: Marcel Dekker Inc., 1987

[21] N. J. Bershad, S. McLaughlin, and C. F. N. Cowan, "Performance Comparison of RLS and LMS algorithms for tracking a first order Markov Communications Channel," in *Proc. IEEE Int. Symp. on Circuits and Systems* (New Orleans, LA), May 1990, pp. 266-270.

[22] J. Botto, and G. V. Moustakides, "Stabilizing the fast Kalman algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, no. 9, pp. 1342-1348, Sep. 1989.

[23] G. E. Bottomley, and S.T. Alexander, "A novel approach for stabilizing recursive least squares filters," *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 8, pp. 1770-1779.

[24] C. Caraiscos, and B. Liu, "A roundoff error analysis of the LMS adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 1, pp. 34-41, Feb. 1984.

[25] J. M. Cioffi, and T. Kailath, "Fast, recursive-least squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 2, pp. 304-337, April 1984.

[26] J. M. Cioffi, "Limited-precision effects in adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 7, pp. 821-833, July 1987.

[27] S. C. Douglas, and T. H.-Y. Meng, "An optimum NLMS algorithm: performance improvement over LMS," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Toronto, Ontario, Canada) May 1991, pp. 2125-2128.

[28] E. Eleftheriou, and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 5, pp. 1097-1109, Oct. 1986.

[29] P. Fabre, C. Gueguen, "Fast recursive least-squares algorithms: preventing divergence," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* (Tampa, FL), pp. 1149-1152, May 1985.

[30] S. Haykin, *Adaptive Filter Theory*, Englewood Cliffs, N.J.: Prentice Hall, 1986.

[31] O. D. Macchi, and N. J. Bershad, "Adaptive recovery of a chirped sinusoid in noise, Part 1: Performance of the RLS algorithm," *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 3, pp. 583-594, March 1991.

[32] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York, N.Y.: McGraw-Hill, Inc., 1984, Second Edition

[33] D. T. Slock, and T. Kailath, "Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol. ASSP-39, no. 1, pp. 92-114, Jan. 1991.