

# New Improved Recursive Least-Squares Adaptive-Filtering Algorithms

Md. Zulfikar Ali Bhotto, *Member, IEEE*, and Andreas Antoniou, *Life Fellow, IEEE*

**Abstract**—Two new improved recursive least-squares adaptive-filtering algorithms, one with a variable forgetting factor and the other with a variable convergence factor are proposed. Optimal forgetting and convergence factors are obtained by minimizing the mean square of the noise-free *a posteriori* error signal. The determination of the optimal forgetting and convergence factors requires information about the noise-free *a priori* error which is obtained by solving a known  $L_1 - L_2$  minimization problem. Simulation results in system-identification and channel-equalization applications are presented which demonstrate that improved steady-state misalignment, tracking capability, and readaptation can be achieved relative to those in some state-of-the-art competing algorithms.

**Index Terms**—Adaptive filters, adaptive-filtering algorithms, recursive least-squares algorithms, forgetting factor, convergence factor.

## I. INTRODUCTION

AS IN classical optimization algorithms, the convergence characteristics of adaptive-filtering algorithms depend on the search directions used. Two well known search directions, namely, steepest-descent and Newton search directions, have their merits and demerits. Steepest-descent search directions are computationally simple, numerically robust, but offer a convergence speed that is highly dependent on the eigenvalue spread ratio of the Hessian matrix [1]. Newton search directions, on the other hand, offer fast convergence although a large amount of computation is required to achieve convergence. Least-mean-squares and normalized-least-mean-squares (LMS and NLMS, respectively), and affine-projection (AP) algorithms employ steepest-descent search directions and hence their convergence speed is often unsatisfactory particularly when the input signal is highly correlated [2], [3]. On the other hand, recursive-least-squares (RLS) algorithms employ Newton search directions and hence they offer faster convergence and reduced steady-state misalignment relative to algorithms that employ steepest-descent directions.

With a large forgetting factor, RLS algorithms yield a reduced steady-state misalignment at the expense of a poor readaptation capability and with a small forgetting factor they offer an improved readaptation capability at the cost of an increased

steady-state misalignment [2]. In order to achieve a reduced steady-state misalignment and good readaptation capability simultaneously, RLS algorithms with a variable forgetting factor (VFF) have been proposed in [4]–[7]. Like other RLS algorithms, the algorithms in [6], [7] involve an increased computational complexity of order  $M^2$  ( $\mathcal{O}(M^2)$ ) where  $M$  is the length of the adaptive filter. The computational complexity of the VFF fast RLS (FRLS) algorithms in [4], [5], on the other hand, is of  $\mathcal{O}(M)$ . Some other FRLS algorithms can be found in [2], [3], [8]. In [4], the variable forgetting factor varies in proportion to the inverse of the squared error and it can become negative [4] but the problem can be prevented by using a prespecified threshold (see [4] for details). In [5], the variable forgetting factor is obtained by minimizing the excess mean-squared error (EMSE) which varies in proportion to the inverse of the autocorrelation of the error signal (see (62) in [5]). The variable forgetting factor in [5] decreases gradually as time advances and, consequently, it does not yield a significant improvement in the steady-state misalignment in nonstationary environments over those achieved with other FRLS algorithms.

The known VFF RLS algorithm reported in [6], referred to hereafter as the KVFF-RLS algorithm, uses a forgetting factor which is controlled by the step size and its evolution is constrained to be bounded by two levels. In the case of system-identification applications, this algorithm works with the lower bound of the forgetting factor whenever a change in the unknown system occurs. Otherwise, it works with the larger bound of the forgetting factor. The VFF RLS algorithm reported in [7], referred to hereafter as the *switching* RLS (SRLS) algorithm, operates with a prespecified forgetting factor and whenever a change in the unknown system occurs it uses a much smaller forgetting factor that is obtained by using the power of the *a priori* error signal. Since prespecified forgetting factors are required in the VFF-RLS algorithms in [6], [7] they do not track Markov-type nonstationarities well.

A variable convergence factor (VCF) has been used before in an LMS-Newton algorithm described in [9]. This algorithm performs better than the conventional RLS (CRLS) algorithm described in [3, p. 199] in terms of steady-state misalignment in Markov-type nonstationary environments but its speed of convergence is not as good as that of the CRLS algorithm.

In this paper, we propose a new RLS algorithm that uses a VFF, referred to hereafter as the VFF-RLS algorithm that does not require a prespecified forgetting factor. The forgetting factor is obtained by minimizing the mean square of the noise-free *a posteriori* error. In doing so, an optimal convergence factor is obtained. Based on this approach, an RLS algorithm can also be developed that uses a fixed forgetting factor along with a variable convergence factor (VCF); this will be referred to here-

Manuscript received April 26, 2012; revised July 16, 2012; accepted July 31, 2012. Date of publication December 04, 2012; date of current version May 23, 2013. This paper was recommended by Associate Editor M. Chakraborty.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: zbhotto@ece.uvic.ca, aantoniou@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2012.2220452

after as the VCF-RLS algorithm. Simulation results show that the new VFF-RLS algorithm offers improved steady-state misalignment, readaptation, and tracking capability compared to those achieved with the CRLS algorithm [3], KVFF-RLS algorithm [6], and SRLS algorithm [7] for the same initial speed of convergence. On the other hand, the proposed VCF-RLS algorithm offers improved steady-state misalignment compared to that achieved with the CRLS and LMS-Newton algorithms for the same fixed forgetting factor.

## II. RECURSIVE LEAST-SQUARES ALGORITHMS

The weight-vector update formula in RLS adaptive-filtering algorithms, referred to hereafter as RLS adaptation algorithms, is obtained by solving the minimization problem [2], [3]

$$\underset{\mathbf{w}_k}{\text{minimize}} \sum_{i=1}^k \lambda^{k-i} (d_i - \mathbf{x}_i^T \mathbf{w}_k)^2 \quad (1)$$

where  $0 \ll \lambda < 1$  is the forgetting factor,  $d_i$  and  $\mathbf{x}_i \in \mathcal{R}^{M \times 1}$  are the desired signal and input signal vector at iteration  $i$ , respectively, and  $\mathbf{w}_k \in \mathcal{R}^{M \times 1}$  is the required weight vector at iteration  $k$ . The solution of the minimization problem in (1) can be obtained as

$$\mathbf{w}_k = \mathbf{R}_k^{-1} \mathbf{p}_k \quad (2)$$

where  $\mathbf{R}_k = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_i \mathbf{x}_i^T$  and  $\mathbf{p}_k = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}_i d_i$  are approximations of the autocorrelation matrix  $\hat{\mathbf{R}}$  and crosscorrelation vector  $\hat{\mathbf{p}}$  of the Wiener filter [10], respectively. The autocorrelation matrix and crosscorrelation vector can be expressed as

$$\mathbf{R}_k = \lambda \mathbf{R}_{k-1} + \mathbf{x}_k \mathbf{x}_k^T \quad (3)$$

and

$$\mathbf{p}_k = \lambda \mathbf{p}_{k-1} + \mathbf{x}_k d_k \quad (4)$$

respectively.

With  $\lambda \approx 1$ , the errors  $\mathbf{R}_k - \hat{\mathbf{R}}$  and  $\mathbf{p}_k - \hat{\mathbf{p}}$  become small and, therefore, the difference between the Wiener solution (see [10]) and (2) is also small. The initial autocorrelation matrix and crosscorrelation vector  $\mathbf{R}_0$  and  $\mathbf{p}_0$  should be chosen as  $\delta \mathbf{I}$  and  $\mathbf{0}$ , respectively, where  $\mathbf{I}$  is the identity matrix and  $\delta$  is a small positive constant of the order of  $10^{-4}$ . With this choice, the effects of  $\mathbf{R}_0$  and  $\mathbf{p}_0$  on the update formulas in (3) and (4), respectively, would quickly diminish and, therefore, the initial values of  $\mathbf{R}_k$  and  $\mathbf{p}_k$  would not contribute significantly to the steady-state values of  $\mathbf{R}_k$  and  $\mathbf{p}_k$ . On the other hand, if the entries of  $\mathbf{R}_0$  and  $\mathbf{p}_0$  are large with a  $\lambda \approx 1$  the misalignment between the Wiener solution and (2) would be large and the convergence of  $\mathbf{w}_k$  in (2) to the Wiener solution would be slow. Situations where the entries of  $\mathbf{R}_0$  and  $\mathbf{p}_0$  become quite large can arise in system-identification applications when sudden system changes occur during the learning stage. For example, if a change occurs in the system to be identified at iteration  $k = K$ , an RLS algorithm has to reconverge to the new state of the system. In such a situation, the entries of  $\mathbf{R}_0 = \lambda \mathbf{R}_{K-1}$  are much larger than the entries of  $\delta \mathbf{I}$  and  $\mathbf{p}_0 = \lambda \mathbf{p}_{K-1} \neq \mathbf{0}$ . As a result, with  $\lambda \approx 1$ , the

effect of  $\mathbf{R}_0$  and  $\mathbf{p}_0$  on (3) and (4) would persist and, therefore, a large error would be introduced in the steady-state values of  $\mathbf{R}_k$  and  $\mathbf{p}_k$ . Hence, the difference between the Wiener solution (i.e., the Wiener solution that were to be obtained if the new state of the system were to exist from  $k = 0$  to  $k = \infty$ ) and (2) would be large and the reconvergence of  $\mathbf{w}_k$  to the optimal weight vector for the new state of the system would be very slow. For the same reason, an RLS algorithm with a  $\lambda \approx 1$  could fail to reconverge in the presence of an outlier in  $d_k$  or  $\mathbf{x}_k$ ; furthermore, the RLS algorithm could lose its tracking capability in nonstationary environments. Some Newton-type algorithms that are robust with respect to outliers can be found in [11]–[14].

Improved readaptation capability has been achieved in the KVFF-RLS and SRLS algorithms reported in [6], [7] by reducing the value of  $\lambda$ , to ensure that the values of the elements of  $\mathbf{R}_0 = \lambda \mathbf{R}_{K-1}$  and  $\mathbf{p}_0 = \lambda \mathbf{p}_{K-1}$  are reduced, and then rapidly returning  $\lambda$  to its previous value which is close to unity. An alternative approach for achieving improved readaptation capability reported in [15] involves using a convex combination of the outputs of two RLS adaptive filters, one with a small value of  $\lambda$  and the other with a value of  $\lambda$  close to unity. A *sigmoid* function is used to assign more weight on the output of the adaptive filter with a small  $\lambda$  during transience and more weight on the output of the adaptive filter with a  $\lambda$  close to unity during steady state. Since  $\lambda \approx 1$  is the optimal forgetting factor for the CRLS algorithm [2] in the sense that it yields the minimum mean-square error, the performance of the CRLS algorithm would be identical with that of the algorithms in [6], [7], [15] in stationary environments.

## III. IMPROVED RECURSIVE LEAST-SQUARES ALGORITHMS

In this section, we develop VFF-RLS and VCF-RLS algorithms that offer improved performance in tracking Markov-type nonstationarities and sudden system changes and also offer reduced steady-state misalignment relative to those achieved with the CRLS, KVFF-RLS, and SRLS algorithms.

### A. VFF-RLS Algorithm

The inverse of the autocorrelation matrix in (3) can be obtained by using the matrix inversion formula [2], [3] as

$$\mathbf{S}_k = \frac{1}{\lambda} \left( \mathbf{S}_{k-1} - \frac{1}{\lambda + \tau_k} \mathbf{S}_{k-1} \mathbf{x}_k \mathbf{x}_k^T \mathbf{S}_{k-1} \right) \quad (5)$$

where  $\mathbf{S}_k$  is a positive-definite matrix for all  $k$  and  $\tau_k = \mathbf{x}_k^T \mathbf{S}_{k-1} \mathbf{x}_k > 0$ . Using (5) in (2), the weight-vector update formula for the CRLS algorithm can be expressed as

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \frac{q}{\lambda + \tau_k} \mathbf{S}_{k-1} \mathbf{x}_k e_k \quad (6)$$

where  $e_k$  is the *a priori* error signal and  $q$  is the *convergence factor* whose value assumes the value of unity in the CRLS algorithm. The *a priori* error signal  $e_k$  can be expressed as

$$e_k = e_{f,k} + v_k \quad (7)$$

where  $v_k$  is a white Gaussian noise signal with variance  $\sigma_v^2$ ,

$$e_{f,k} = \mathbf{x}_k^T \mathbf{h}_{k-1} - \mathbf{x}_k^T \mathbf{w}_{k-1} \quad (8)$$

is the noise-free *a priori* error signal and in the case of a system-identification application  $\mathbf{h}_{k-1}$  is the impulse response of the unknown system. The *a posteriori* error signal  $\epsilon_k$  can be expressed as

$$\epsilon_k = \epsilon_{f,k} + v_k \quad (9)$$

where

$$\epsilon_{f,k} = \mathbf{x}_k^T \mathbf{h}_{k-1} - \mathbf{x}_k^T \mathbf{w}_k \quad (10)$$

is the noise-free *a posteriori* error signal. In the case of a system-identification application, the desired signal  $d_k$  becomes  $d_k = \mathbf{x}_k^T \mathbf{h}_{k-1} + v_k$ . The noise-free *a posteriori* error signal at iteration  $k$  can also be obtained by using (6) and (8) in (10) as

$$\epsilon_{f,k} = (1 - q\tau_{p,k})\epsilon_{f,k} - q\tau_{p,k}v_k \quad (11)$$

where

$$\tau_{p,k} = \mathbf{x}_k^T \mathbf{S}_k \mathbf{x}_k = \frac{\tau_k}{\lambda + \tau_k} \quad (12)$$

lies in the range (0,1). If the unknown system now evolves as per a first-order Markov model, i.e.,  $\mathbf{h}_k = \mathbf{h}_{k-1} + \boldsymbol{\phi}_k$  where  $\boldsymbol{\phi}_k$  is a white Gaussian noise signal with variance  $\sigma_\phi^2$ , then  $\epsilon_{f,k}$  in (10) and (11) requires an additional term due to the lag in adaptation. This can be obtained from (11) as

$$\epsilon_{f,k} = (1 - q\tau_{p,k})\epsilon_{f,k} - q\tau_{p,k}v_k - \mathbf{x}_k^T \boldsymbol{\phi}_k. \quad (13)$$

By squaring both sides of (13), we obtain

$$\begin{aligned} \epsilon_{f,k}^2 &= (1 - q\tau_{p,k})^2 \epsilon_{f,k}^2 + q^2 \tau_{p,k}^2 v_k^2 + (\mathbf{x}_k^T \boldsymbol{\phi}_k)^2 \\ &\quad - 2q(1 - q\tau_{p,k})\tau_{p,k}v_k\epsilon_{f,k} + 2q\tau_{p,k}v_k\mathbf{x}_k^T \boldsymbol{\phi}_k \\ &\quad - 2(1 - q\tau_{p,k})\epsilon_{f,k}\mathbf{x}_k^T \boldsymbol{\phi}_k. \end{aligned} \quad (14)$$

Assuming that  $v_k$  and  $\boldsymbol{\phi}_k$  are independent and white Gaussian noise signals and taking the expectation on both sides in (14), we obtain

$$\begin{aligned} E[\epsilon_{f,k}^2] &= E[e_{f,k}^2] - 2qE[\tau_{p,k}e_{f,k}^2] + q^2E[\tau_{p,k}^2e_{f,k}^2] \\ &\quad + q^2E[\tau_{p,k}^2]E[v_k^2] + E[(\mathbf{x}_k^T \boldsymbol{\phi}_k)^2]. \end{aligned} \quad (15)$$

An optimal value of the convergence factor  $q$  can now be obtained by solving the one-dimensional minimization problem

$$\underset{q}{\text{minimize}} E[\epsilon_{f,k}^2]. \quad (16)$$

The solution of this problem can be obtained by setting the derivative of the objective function in (16) with respect to  $q$ , i.e.,

$$\begin{aligned} \frac{\partial E[\epsilon_{f,k}^2]}{\partial q} &= -2E[\tau_{p,k}e_{f,k}^2] + 2qE[\tau_{p,k}^2e_{f,k}^2] \\ &\quad + 2qE[\tau_{p,k}^2]E[v_k^2] \end{aligned}$$

to zero. In this way, we can obtain

$$q = \frac{E[\tau_{p,k}e_{f,k}^2]}{E[\tau_{p,k}^2e_{f,k}^2] + E[\tau_{p,k}^2]E[v_k^2]}. \quad (17)$$

Note that since  $E[e_{f,k}^2]$  is a measure of the excess MSE (EMSE) [2], using (17) in (6) the minimum EMSE can also be obtained.

Based on the above analysis, we can now obtain an optimal value of the forgetting factor. We start by obtaining a simplified expression for  $q$  in (17). The recursion formula in (3) can be expressed as

$$\begin{aligned} \mathbf{R}_k &= \lambda^k \mathbf{R}_0 + \mathbf{x}_k \mathbf{x}_k^T + \lambda \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T + \lambda^2 \mathbf{x}_{k-2} \mathbf{x}_{k-2}^T \\ &\quad + \lambda^3 \mathbf{x}_{k-3} \mathbf{x}_{k-3}^T + \dots + \lambda^{k-1} \mathbf{x}_1 \mathbf{x}_1^T. \end{aligned} \quad (18)$$

Taking the expectation of both sides in (18), we obtain

$$\begin{aligned} E[\mathbf{R}_k] &= \lambda^k \mathbf{R}_0 + (1 + \lambda + \lambda^2 + \lambda^3 + \dots + \lambda^{k-1}) \hat{\mathbf{R}} \\ &= \lambda^k \mathbf{R}_0 + \frac{1 - \lambda^k}{1 - \lambda} \hat{\mathbf{R}} \end{aligned} \quad (19)$$

which at steady state, i.e., as  $k \rightarrow \infty$ , becomes

$$E[\mathbf{R}_k] = \frac{1}{1 - \lambda} \hat{\mathbf{R}}. \quad (20)$$

As in [2], at steady state

$$E[\mathbf{S}_k] \approx (E[\mathbf{R}_k])^{-1} = (1 - \lambda) \hat{\mathbf{R}}^{-1}. \quad (21)$$

On the other hand,

$$\begin{aligned} E[\tau_{p,k}] &= E[\mathbf{x}_k^T \mathbf{S}_k \mathbf{x}_k] = \text{tr}(E[\mathbf{S}_k] E[\mathbf{x}_k \mathbf{x}_k^T]) \\ &= \text{tr}(E[\mathbf{S}_k] \hat{\mathbf{R}}) \end{aligned} \quad (22)$$

and from (21) and (22), we obtain

$$E[\mathbf{x}_k^T \mathbf{S}_k \mathbf{x}_k] = (1 - \lambda)M. \quad (23)$$

If we neglect the dependence of  $\tau_{p,k}$  on  $e_{f,k}$  in (17) and assume that  $M$  is large, then  $E[\tau_{p,k}^2] \approx E[\tau_{p,k}]^2$  as shown in the Appendix. Using this approximation along with (23) in (17), we get the optimality condition

$$q \approx \frac{1}{(1 - \lambda)} \frac{E[e_{f,k}^2]}{ME[e_{f,k}^2] + ME[v_k^2]}. \quad (24)$$

For any fixed  $\lambda$ , (24) yields a  $q$  that would be an approximate solution of the problem in (16). Similarly, for any fixed  $q$ , (24) yields a  $\lambda$  that is also an approximate solution of the problem in (16). Using  $q = 1$ , an optimal forgetting factor  $\lambda$  can be obtained as

$$\lambda_{opt} \approx 1 - \frac{E[e_{f,k}^2]}{M(E[e_{f,k}^2] + E[v_k^2])}. \quad (25)$$

In order to compute  $\lambda_{opt}$ , we need  $E[e_{f,k}^2]$  which is unknown *a priori*. With the noise-free *a priori* error signal  $e_{f,k}$  known, we can approximate  $E[e_{f,k}^2]$  by using the time average of  $e_{f,k}^2$ , which is given by

$$\sigma_{1,k} = \beta\sigma_{1,k-1} + (1 - \beta)e_{f,k}^2 \quad (26)$$

where  $\beta$  is the pole of a first-order moving average filter in (26) whose value should be in the range  $0 \ll \beta < 1$ . Using (26) in (25), we obtain the optimal forgetting factor at iteration  $k$  as

$$\lambda_k = 1 - \frac{\sigma_{1,k}}{M(\sigma_{1,k} + \sigma_v^2)}. \quad (27)$$

### B. VCF-RLS Algorithm

The proposed VCF-RLS algorithm is based on the following principles. Equation (24) suggests that for every fixed  $\lambda$  there is a value of  $q$  that solves the problem in (16). However, since the assumption  $E[\tau_{p,k}^2] \approx E[\tau_{p,k}]^2$  has been made and the dependence of  $e_{f,k}$  on  $\tau_{p,k}$  has been neglected in the derivation of (24),  $q$  can become greater than unity which, as can be seen from (6), would affect the stability of the adaptive filter. To circumvent this problem, we use  $\mu_k = \min(1, q_k)$  instead of  $q$  in (6). The variable convergence factor  $q_k$  at iteration  $k$  can be obtained as

$$q_k = \frac{1}{M(1 - \lambda)} \frac{\sigma_{1,k}}{\sigma_{1,k} + c_1\sigma_v^2} \quad (28)$$

by replacing  $E[e_{f,k}^2]$  in (24) by  $\sigma_{1,k}$  given by (26). In (28), integer  $c_1$  is a tuning integer whose value should lie in the range 2 to 8 based on extensive simulations (see Section IV-B). Constant  $c_1$  in (28) would further reduce the value of  $q_k$  and hence a reduced steady-state misalignment can be achieved. Since  $\sigma_{1,k}$  in (26) is a measure of the EMSE of the adaptive filter, its steady-state value would be significantly smaller than  $\sigma_v^2$ . However, due to the use of a time average in (26) the transient value of  $\sigma_{1,k}$  would be significantly larger than  $\sigma_v^2$ . Therefore, we would obtain  $\sigma_{1,k} + c_1\sigma_v^2 \approx \sigma_{1,k}$  and  $\sigma_{1,k} + c_1\sigma_v^2 \approx c_1\sigma_v^2$  in (28) during the transience and steady state of the adaptive filter, respectively. In such a situation from (28), we would get

$$q_k \approx \frac{1}{M(1 - \lambda)}$$

during transience and

$$q_k \approx \frac{1}{M(1 - \lambda)} \frac{\sigma_{1,k}}{c_1\sigma_v^2}$$

during steady state. If we now choose a  $\lambda$  in the range  $0 < \lambda < 1$ , e.g.,  $\lambda = 0.95$ , we would get  $q_k \gg 1$  during transience. On the other hand, during steady state we would get  $q_k \approx 0$  as  $\sigma_{1,k} \approx 0$  in (26) during steady state. Under these circumstances, we would obtain  $\mu_k \approx 1$  and  $\mu_k \approx 0$  during transience and steady state, respectively. Therefore, the convergence speed of the proposed VCF-RLS algorithm would remain the same as that of the CRLS algorithm while its steady-state misalignment would be reduced. When  $\lambda$  is chosen to be close to unity, e.g.,  $\lambda = 1 - 2^{-15}$ , we would obtain  $q_k \gg 1$ , i.e.,  $\mu_k = 1$  for all  $k$

and hence the performance of the proposed VCF-RLS algorithm would be similar to that of the CRLS algorithm.

The proposed VCF-RLS algorithm can be used in applications where the use of RLS algorithms with a fixed forgetting factor is preferred.

## IV. IMPLEMENTATION ISSUES OF THE PROPOSED RLS ALGORITHMS

In this section, we discuss some implementation issues associated with the proposed RLS algorithms.

### A. Noniterative Shrinkage Method

The value of  $e_{f,k}$  in (26) can be obtained from the *a priori* error signal by using a so-called *noniterative shrinkage* method which has been used to solve image denoising problems in [16], [17]. In this method, a noise-free signal  $\mathbf{o}_f$  can be recovered from a noisy signal  $\mathbf{o} = \mathbf{o}_f + \mathbf{v}$ , where  $\mathbf{v}$  is a white Gaussian noise signal, by solving the  $L_1 - L_2$  minimization problem

$$\underset{\mathbf{a}}{\text{minimize}} \ t\|\mathbf{a}\|_1 + 0.5\|\mathbf{D}\mathbf{a} - \mathbf{o}\|^2 \quad (29)$$

where  $t$  is the threshold parameter and  $\mathbf{D}$  is an orthogonal matrix.

In the proposed VFF-RLS algorithm,  $\mathbf{o}$ ,  $\mathbf{o}_f$ , and  $\mathbf{D}$  in (29) become  $e_k$ ,  $e_{f,k}$ , and 1, respectively, and the optimal solution, i.e.,  $\mathbf{o}_{f,k} = \hat{e}_{f,k}$  of the problem in (29) can be obtained as

$$\hat{e}_{f,k} = \text{sign}(e_k)\max(|e_k| - t, 0). \quad (30)$$

Since the computation of  $\hat{e}_{f,k}$  is not iterative, the above approach is suitable for real-time applications such as adaptive filtering. The formula in (30) reduces  $e_k$  by an amount  $t$ . Using an appropriate  $t$  we can obtain  $\hat{e}_{f,k} \approx e_{f,k}$ .

Different  $L_1 - L_2$  minimization problems are formulated to obtain new RLS adaptation algorithms in [18]–[20]. In the RLS algorithms in [18], [19] an  $L_1 - L_2$  minimization problem was formulated whose solution was used to bring sparsity in the weight vector. In the RLS algorithm in [20] an  $L_1 - L_2$  minimization problem was formulated and its solution was then used to detect and remove the impulsive noise component in the error signal. We formulated a different  $L_1 - L_2$  minimization problem and we then used the solution to obtain a new variable forgetting factor and convergence factor in the CRLS algorithm.

### B. Threshold Parameter

Taking the expectation of the squares of both sides in (7), we obtain

$$E[e_k^2] = E[e_{f,k}^2] + \sigma_v^2 \quad (31)$$

as  $e_{f,k}$  is independent of  $v_k$  for a white Gaussian signal  $v_k$  with variance  $\sigma_v^2$  which may suggest that the threshold parameter should be chosen as  $t = \sigma_v$ . However, since the derivation of (27) involves: a) neglecting the dependence of  $\tau_{p,k}$  on  $e_{f,k}$ , b) assuming that  $E[\tau_{p,k}^2] \approx E[\tau_{p,k}]^2$ , and c) using a time average instead of a statistical average,  $t$  needs to be tuned with respect to  $\sigma_v$  to achieve improved results. Through extensive simulations, it was found that  $t = \sqrt{c_1\sigma_v^2}$  with  $c_1$  in the range 2 to 8 yields good results.

### C. Forgetting Factor

The value of the forgetting factor given by (25) is in the range  $(1 - 1/M, 1)$  and it is optimal in the sense that it yields a minimum EMSE. Since the value of  $\sigma_{1,k}$  in (26) becomes very large during transience and very small during steady state, (27) yields  $\lambda_k \approx 1 - 1/M$  during transience and  $\lambda_k \approx 1$  during steady state. As a result, as per the discussion in Section II, the proposed VFF-RLS algorithm yields fast convergence, good readaptation capability, and reduced steady-state misalignment. As reported in [2] and [8], the range of the forgetting factor for the stable operation of the FRLS algorithm is  $\lambda \in (1 - 2/M, 1)$  which encompasses the range of the forgetting factor in (27). Further improvement regarding convergence speed and readaptation capability can be achieved in the proposed algorithm if  $\lambda_k$  in (27) assumes values close to  $1 - 2/M$  during transience while  $\lambda_k \approx 1$  during steady state. Therefore, we propose to use

$$\lambda_k = 1 - \frac{2\sigma_{1,k}}{M(\sigma_{1,k} + c_1\sigma_v^2)} \quad (32)$$

where  $c_1$  is a tuning integer in the range 2 to 8 instead of the  $\lambda_k$  given by (27). Tuning integer  $c_1$  is used to increase the value of  $\lambda_k$  to a similar level to that achieved using (27) during steady state and a value in the range 2 to 8 was found to give good results as per discussion in Section III-B. As can be seen in (26), since  $\sigma_{1,k} \gg \sigma_v^2$  during transience we obtain  $\lambda_k \approx 1 - 2/M$  and since  $\sigma_{1,k} \approx 0$  during steady state we obtain  $\lambda_k \approx 1$  during steady state. In other words, the steady-state values of  $\lambda_k$  in (32) and (27) would be very similar and hence both of them would approximate  $\lambda_{opt}$  in (25) with similar accuracy and hence would minimize  $E[\epsilon_{f,k}^2]$  in (16). The transient values of  $\lambda_k$  in (32), on the other hand, would be lower than those in (27) and hence improved readaptation capability would be achieved.

Based on the above principles, the implementations of the proposed RLS algorithms given in Table I can be obtained.

## V. STEADY-STATE ANALYSIS

In this section, we derive expressions for the MSE for the proposed RLS algorithms by using the energy conservation relation reported in [2, p. 287].

The impulse response of the unknown system is modeled as a first-order Markov model of the form [3]

$$\mathbf{h}_k = \mathbf{h}_{k-1} + \phi_k \quad (33)$$

where the elements of  $\phi_k$  are the samples of a white Gaussian noise signal with variance  $\sigma_\phi^2$ . The weight-vector update formula in (6) for the system model in (33) can be expressed in terms of the weight-error vector

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_{k-1} - \mathbf{S}_k \mathbf{x}_k e_k - \phi_k \quad (34)$$

where

$$\hat{\mathbf{w}}_k = \mathbf{h}_{k-1} - \mathbf{w}_k \quad (35)$$

and

$$\hat{\mathbf{w}}_{k-1} = \mathbf{h}_{k-1} - \mathbf{w}_{k-1}. \quad (36)$$

TABLE I  
PROPOSED RLS ALGORITHMS

Initialization: Set $\mathbf{w}_0 = \mathbf{0}$ , $0 < \lambda_f < 1$ , $\mathbf{S}_0 = 10^4 \mathbf{I}$ , $\sigma_0 = 0$ , $t = \sqrt{c_1 \sigma_v^2}$ , $\beta$ in the range 0.9 to 0.99, and $c_1$ in the range 2 to 8.
Iteration $k$ :
$e_k = d_k - \mathbf{w}_{k-1}^T \mathbf{x}_k$
$\mathbf{t}_k = \mathbf{S}_{k-1} \mathbf{x}_k$
$\hat{e}_{f,k} = \text{sign}(e_k) \cdot \max( e_k  - t, 0)$
$\sigma_k = \beta \sigma_{k-1} + (1 - \beta) \hat{e}_{f,k}^2$
VFF-RLS algorithm:
$\lambda_k = 1 - \frac{2\sigma_k}{M(\sigma_k + c_1 \sigma_v^2)}$
VCF-RLS algorithm:
$\lambda_k = \lambda_f$
$\tau_k = \frac{\lambda_k}{\lambda_k + \mathbf{x}_k^T \mathbf{t}_k}$
$\mathbf{g}_k = \sqrt{\tau_k} \mathbf{t}_k$
$\mathbf{S}_k = \lambda_k^{-1} (\mathbf{S}_{k-1} - \mathbf{g}_k \mathbf{g}_k^T)$
VFF-RLS algorithm:
$\mu_k = 1$
VCF-RLS algorithm:
$\mu_k = \min \left[ 1, \frac{\sigma_k}{(1 - \lambda_k) M (\sigma_k + c_1 \sigma_v^2)} \right]$
$\mathbf{w}_k = \mathbf{w}_{k-1} + (\mu_k \sqrt{\tau_k} e_k) \mathbf{g}_k$

This model is used in [9] and [21] to obtain the steady-state MSE of the LMS-Newton and RLS algorithms, respectively, in Markov-type nonstationary environments.

### A. MSE in Nonstationary Environments

Let us consider the case of the proposed VFF-RLS algorithm. Premultiplying both sides of (34) by  $\mathbf{x}_k^T \mathbf{S}_k \Gamma$ , we obtain

$$\mathbf{x}_k^T \mathbf{S}_k \Gamma \hat{\mathbf{w}}_k = \mathbf{x}_k^T \mathbf{S}_k \Gamma \hat{\mathbf{w}}_{k-1} - \mathbf{x}_k^T \mathbf{S}_k \Gamma \mathbf{S}_k \mathbf{x}_k e_k - \mathbf{x}_k^T \mathbf{S}_k \Gamma \phi_k \quad (37)$$

where  $\Gamma$  is a positive-definite matrix. Scaled noise-free *a posteriori* and *a priori* errors can be defined as

$$\epsilon_{f,k}^\Gamma = \mathbf{x}_k^T \mathbf{S}_k \Gamma \hat{\mathbf{w}}_k \quad (38)$$

and

$$e_{f,k}^\Gamma = \mathbf{x}_k^T \mathbf{S}_k \Gamma \hat{\mathbf{w}}_{k-1} \quad (39)$$

respectively. Also let us define

$$\tau_{p,k}^\Gamma = \mathbf{x}_k^T \mathbf{S}_k \Gamma \mathbf{S}_k \mathbf{x}_k \quad (40)$$

$$\phi_k^\Gamma = \mathbf{x}_k^T \mathbf{S}_k \Gamma \phi_k. \quad (41)$$

Using (38)–(41) in (37), we obtain

$$\epsilon_{f,k}^\Gamma = e_{f,k}^\Gamma - \tau_{p,k}^\Gamma e_k - \phi_k^\Gamma. \quad (42)$$

Now substituting  $e_k$  in (34) by using the  $e_k$  obtained from (42), we have

$$\hat{\mathbf{w}}_k + \frac{e_{f,k}^\Gamma}{\tau_{p,k}^\Gamma} \mathbf{S}_k \mathbf{x}_k = \hat{\mathbf{w}}_{k-1} + \frac{\epsilon_{f,k}^\Gamma}{\tau_{p,k}^\Gamma} \mathbf{S}_k \mathbf{x}_k + \frac{\phi_k^\Gamma}{\tau_{p,k}^\Gamma} \mathbf{S}_k \mathbf{x}_k - \phi_k. \quad (43)$$

If we take the square of the weighted  $L_2$  norm, i.e.,  $\|\cdot\|_\Gamma^2$ , of both sides in (43), we get

$$\begin{aligned} \|\hat{\mathbf{w}}_k\|_\Gamma^2 + \frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma} &= \|\hat{\mathbf{w}}_{k-1}\|_\Gamma^2 + \frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma} \\ &+ 2\frac{e_{f,k}^\Gamma\phi_k^\Gamma}{\tau_{p,k}^\Gamma} - 2\hat{\mathbf{w}}_{k-1}^T\Gamma\phi_k \\ &+ \|\phi_k\|_\Gamma^2 - \frac{\left(\phi_k^\Gamma\right)^2}{\tau_{p,k}^\Gamma}. \end{aligned} \quad (44)$$

The energy conservation relation for the system in (33) can now be obtained by taking the expectation of both sides in (44) as

$$\begin{aligned} E\left[\|\hat{\mathbf{w}}_k\|_\Gamma^2\right] + E\left[\frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right] &= E\left[\|\hat{\mathbf{w}}_{k-1}\|_\Gamma^2\right] \\ &+ E\left[\frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right] + E\left[\|\phi_k\|_\Gamma^2\right] - E\left[\frac{\left(\phi_k^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right] \end{aligned} \quad (45)$$

where we assume that vector  $\phi_k$  in (44) is an independent and identically distributed white Gaussian noise signal with variance  $\sigma_\phi^2$ . Note that no assumption has been made in obtaining the energy conservation relation in (45).

During steady state, we obtain  $E[\|\hat{\mathbf{w}}_k\|_\Gamma^2] = E[\|\hat{\mathbf{w}}_{k-1}\|_\Gamma^2]$  and, therefore, the steady-state energy conservation relation assumes the form

$$E\left[\frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right] = E\left[\frac{\left(e_{f,k}^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right] + E\left[\|\phi_k\|_\Gamma^2\right] - E\left[\frac{\left(\phi_k^\Gamma\right)^2}{\tau_{p,k}^\Gamma}\right]. \quad (46)$$

Using  $e_{f,k}^\Gamma$  given by (42) in (46) and applying some straightforward simplifications based on the assumption concerning the statistical independence of  $\phi_k$ , we obtain

$$2E\left[e_k e_{f,k}^\Gamma\right] = E\left[\tau_{p,k}^\Gamma e_k^2\right] + E\left[\|\phi_k\|_\Gamma^2\right] \quad (47)$$

If we use (7) in (47) and simplify the result by using the assumption that the measurement noise in (7) is an independent white Gaussian noise signal, we have

$$2E\left[e_{f,k}^\Gamma e_{f,k}\right] = E\left[\tau_{p,k}^\Gamma e_{f,k}^2\right] + E\left[\tau_{p,k}^\Gamma\right] \sigma_v^2 + E\left[\|\phi_k\|_\Gamma^2\right]. \quad (48)$$

Now if we neglect the dependence of  $e_{f,k}$  on  $\tau_{p,k}^\Gamma$  in (48), we obtain

$$2E\left[e_{f,k}^\Gamma e_{f,k}\right] = E\left[\tau_{p,k}^\Gamma\right] E\left[e_{f,k}^2\right] + E\left[\tau_{p,k}^\Gamma\right] \sigma_v^2 + E\left[\|\phi_k\|_\Gamma^2\right]. \quad (49)$$

In order to obtain the EMSE for the case of nonstationary environments, we let  $\Gamma = \mathbf{R}_k$  in (49) and (39)–(41) to get

$$E\left[e_{f,k}^2\right] = \frac{E\left[\tau_{p,k}\right]}{2 - E\left[\tau_{p,k}\right]} \sigma_v^2 + \frac{E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right]}{2 - E\left[\tau_{p,k}\right]}. \quad (50)$$

As can be seen, the first term at the right-hand side of the above equation represents the EMSE for stationary environments. Using (25) and (23) in (50) after some simple manipulations, we obtain

$$E\left[e_{f,k}^2\right]^2 - E\left[e_{f,k}^2\right] \left(E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right] - \sigma_v^2\right) = \sigma_v^2 E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right]. \quad (51)$$

If we now solve (51) for  $E[e_{f,k}^2]$ , we obtain the EMSE as

$$E\left[e_{f,k}^2\right] = E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right] \text{ or } E\left[e_{f,k}^2\right] = -\sigma_v^2.$$

Since  $E[e_{f,k}^2]$  is a positive quantity, we obtain the EMSE for nonstationary environments as

$$E\left[e_{f,k}^2\right] = E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right]. \quad (52)$$

Now  $E[\|\phi_k\|_{\mathbf{R}_k}^2]$  can be obtained as

$$E\left[\|\phi_k\|_{\mathbf{R}_k}^2\right] = \text{tr}\left\{\mathbf{R}_k E\left[\phi_k \phi_k^T\right]\right\} = \sigma_\phi^2 \text{tr}\left\{\mathbf{R}_k\right\} \quad (53)$$

since  $E[\phi_k \phi_k^T] = \sigma_\phi^2 \mathbf{I}$ . From (53) and (52), we obtain the EMSE as

$$E\left[e_{f,k}^2\right] = \sigma_\phi^2 \text{tr}\left\{\mathbf{R}_k\right\}. \quad (54)$$

Therefore, the MSE in nonstationary environments can be obtained as

$$\text{MSE} = \sigma_v^2 + \sigma_\phi^2 \text{tr}\left\{\mathbf{R}_k\right\}. \quad (55)$$

### B. MSE in Stationary Environments

In stationary environments, we have  $\sigma_\phi^2 = 0$  and hence the solution of (51) becomes  $E[e_{f,k}^2] = 0$  or  $E[e_{f,k}^2] = -\sigma_v^2$ . Since the EMSE is a nonnegative quantity, we obtain  $E[e_{f,k}^2] = 0$ . Therefore, for stationary environments, we have

$$\text{MSE} = \sigma_v^2. \quad (56)$$

### C. VCF-RLS Algorithm

Repeating the above analysis, it can be shown that the expressions for the MSE given by (55) and (56) also apply to the VCF-RLS algorithm. However, the analysis is based on the assumption that the exact value of  $E[e_{f,k}^2]$  is known. Since the solution of the problem in (29) would not recover  $\mathbf{o}_f$  exactly,  $\hat{e}_{f,k}$  in (30) would not be exactly equal to  $e_{f,k}$ . Furthermore, (26) would not yield the exact value of  $E[e_{f,k}^2]$  as required for

TABLE II  
MSE FOR THE VFF-RLS ALGORITHM IN DB

Noise power	$M = 11, \sigma_\phi^2 = 10^{-4}$		$M = 21, \sigma_\phi^2 = 10^{-6}$	
	Eq. (57)	Experimental	Eq. (57)	Experimental
-5	-4.98	-4.96	-5.00	-5.01
-10	-9.95	-9.88	-9.99	-9.98
-15	-14.85	-14.86	-15.00	-14.98
-20	-19.54	-19.42	-19.99	-19.90
-25	-23.66	-23.86	-24.92	-24.86

the derivation of the formula in (25). Consequently, the expressions for the MSE given by (55) and (56) would be rough approximations at best in the case of the VCF-RLS algorithm.

#### D. Verification of MSE

In this section, we provide experimental verification for the MSE formula given by (55) in a system-identification application for the case of the proposed VFF-RLS algorithm. The impulse response of the initial unknown system in (33), i.e.,  $\mathbf{h}_0$ , was the impulse response of an FIR filter obtained by using MATLAB commands  $h = fir1(M - 1, \omega_c)$ ;  $h = h/norm(h)$  where  $\omega_c = 0.4$  is the normalized cutoff frequency of the FIR filter. Eight experiments were carried out where the input signal was obtained by filtering a white Gaussian signal with unity variance through an IIR filter with a single pole at 0.95. The values of  $\{M, \sigma_\phi^2\}$  in the first four and the last four experiments were set to  $\{11, 10^{-4}\}$  and  $\{21, 10^{-6}\}$ , respectively. We used (27) with  $\sigma_0 = 0$ ,  $\beta = 0.95$ , and  $t = \sqrt{8\sigma_v^2}$  in all experiments. The desired signal was contaminated in the first four experiments by using a white Gaussian signal with variance values 0.3162, 0.1, 0.0316, 0.01, and 0.0032, respectively, which correspond to signal-to-noise ratios (SNRs) of 5, 10, 15, 20, and 25 dB, respectively. The same values of  $\sigma_v^2$  were also used in the last four experiments. The theoretical MSE was calculated using the formula

$$\text{MSE} \approx \sigma_v^2 + \sigma_\phi^2 M \sigma_x^2 \quad (57)$$

where we assumed that  $\text{tr}\{\mathbf{R}_k\} \approx M\sigma_x^2$ . The learning curve in each experiment was obtained using 500 independent trials and the experimental MSE was obtained by averaging the last 50 samples of 4000 samples in the learning curve.

The experimental and theoretical MSE values are given in Table II. As can be seen, the experimental results agree quite well with the theoretical results and, in effect, the steady-state performance of the VFF-RLS algorithm can be accurately predicted if  $e_{f,k}$  can be accurately recovered from  $e_k$ . As the noise power  $\sigma_v^2$  decreases for a constant input signal power  $\sigma_x^2$ , i.e., for a large SNR, the error between the experimental and theoretical results increases because of the effect of the approximation  $\text{tr}\{\mathbf{R}_k\} \approx M\sigma_x^2$  made in the second term on the right-hand side of (57); in effect, the second term in (57) becomes more prominent compared to the first term for a large SNR.

Note that the experimental values in Table II can also be obtained by using (32) with  $c_1 = 2$  to 8.

## VI. DISCUSSION

The *a posteriori* error signal for stationary environments i.e.,  $\phi_k = \mathbf{0}$ , can be obtained from (6) as

$$\epsilon_k = (1 - q\tau_{p,k})e_k \quad (58)$$

where  $\tau_{p,k}$  is defined in (12). Taking the expectation of the square of  $\epsilon_k$  in (58) and neglecting the dependence of  $e_{f,k}$  on  $\tau_{p,k}$ , we obtain

$$E[\epsilon_k^2] = (1 - 2qE[\tau_{p,k}] + q^2E[\tau_{p,k}^2])E[e_k^2]. \quad (59)$$

For the VFF-RLS algorithm, the convergence factor  $q$  in (58) is equal to unity. By using (25) and (23) in (59) along with the assumption that  $E[\tau_{p,k}^2] \approx E[\tau_{p,k}]^2$  after some manipulation, we get

$$E[\epsilon_k^2] = \frac{\sigma_v^2}{E[e_{f,k}^2] + \sigma_v^2} \sigma_v^2. \quad (60)$$

It is easy to show that (60) also holds true for the proposed VCF-RLS algorithm. Since  $E[e_{f,k}^2]$  is a measure of the excess MSE [2] and it is a positive quantity, we obtain  $E[\epsilon_k^2] < \sigma_v^2$  for the proposed RLS algorithms. For the SRLS algorithm of [7] we obtain  $E[\epsilon_k^2] > \sigma_v^2$ .

An advantage of the proposed RLS algorithms over the KVFF-RLS algorithm in [6] is that the forgetting or convergence factor in the proposed RLS algorithms involves less computation than the forgetting factor in the KVFF-RLS algorithm.

A disadvantage of the proposed RLS algorithms is that the information about the measurement noise variance is required to be known *a priori*. In distributed sensor networks [22]–[25], an adaptive filter at a given sensor can interact with other adaptive filters in the neighboring sensors to update its weight vector. Such interactive adaptations among adaptive filters on the entire network bring significant improvement in the steady-state misalignment in each of the adaptive filters connected to the network. During the communication interval between sensors, the variance of the measurement noise can be obtained by using a time average. On the other hand, for speech signals the variance of the measurement noise can be obtained during silent periods.

## VII. SIMULATION RESULTS

In this section, we present simulation results in system-identification and channel-equalization applications. The performance of the CRLS, SRLS [7], KVFF-RLS [6], and the proposed VFF-RLS algorithms are illustrated by using the mean-square deviation (MSD) curves for different SNRs in stationary and nonstationary environments. The MSD was evaluated as  $\text{MSD} = 20 \log_{10} \|\mathbf{h} - \mathbf{w}_k\|_2$  and it was obtained by using ensemble averages over 1000 independent trials in all experiments. Unless otherwise stated the initial parameter  $\mathbf{w}_0$  was chosen as a zero vector and  $\mathbf{S}_0 = \mathbf{R}_0^{-1}$  was set to  $10^4 \mathbf{I}$  in all algorithms. Parameters  $\sigma_0$ ,  $\beta$ ,  $t$ , and  $c_1$  for the proposed VFF-RLS algorithm given in Table I were set to 0, 0.9,  $\sqrt{8\sigma_v^2}$ , and 8, respectively. The prespecified forgetting factor  $\lambda$  was set to  $1 - 2^{-15}$  in the CRLS, SRLS, and KVFF-RLS algorithms. The parameters for the SRLS algorithm were set

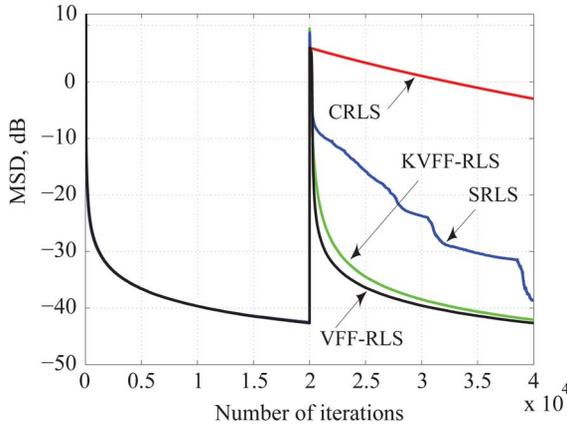


Fig. 1. Learning curves for a system identification application in a stationary environment with a SNR = 20 dB (Experiment 1).

to  $\alpha = 1 - 1/(16M)$ ,  $\beta = 1 - 1/(20M)$  and those for the KVFF-RLS algorithm were set to  $\beta = 1 - 1/(2M)$ ,  $\alpha = 1 - 1/M$ ,  $\mu = 0.003$  in all experiments.

#### A. System-Identification Application

The first experiment was concerned with a stationary environment. The unknown system to be identified was an FIR filter whose impulse response was obtained by using MATLAB commands  $\mathbf{h}_0 = \text{fir1}(M - 1, \omega_c)$ ;  $\mathbf{h}_0 = \mathbf{h}_0/\text{norm}(\mathbf{h}_0)$ ; with  $M = 51$  where  $\omega_c = 0.4$  is the normalized cutoff frequency of the FIR filter. The coefficients of the unknown system were multiplied by  $-1$  at iteration 20 000 to examine the readaptation capability of the RLS algorithms. The input signal was obtained by filtering a white Gaussian noise signal with variance  $\sigma_x^2 = 1.0$  through an IIR filter that had the transfer function [26]

$$H(z) = \frac{1}{z^4 - 0.95z^3 - 0.19z^2 - 0.09z + 0.5}.$$

The measurement noise added to the desired signal was a white Gaussian noise signal with variance  $\sigma_v^2 = 10^{-2}$  which corresponds to a SNR of 20 dB. The MSD curves obtained by using the VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 1.

In the second experiment, we repeated the first experiment with the same setup, input signal, and measurement noise as in the first experiment except that a nonstationarity was introduced as per the model given in (33) where the impulse response of the unknown system,  $\mathbf{h}_0$ , was the same as in the first experiment and the elements of  $\phi_k$  were obtained from a white Gaussian noise signal with a variance  $\sigma_\phi^2 = 10^{-3}$ . The SNR was the same as in the first experiment, i.e., 20 dB. The MSD curves obtained by using the VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 2.

In the third experiment, we repeated the second experiment except that we used  $\sigma_\phi^2 = 10^{-4}$ . The MSD curves obtained by using the VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 3.

In the fourth experiment, we repeated the first experiment for the case of a stationary environment except that the variance of the measurement noise signal was set to  $10^{-4}$  (SNR = 40

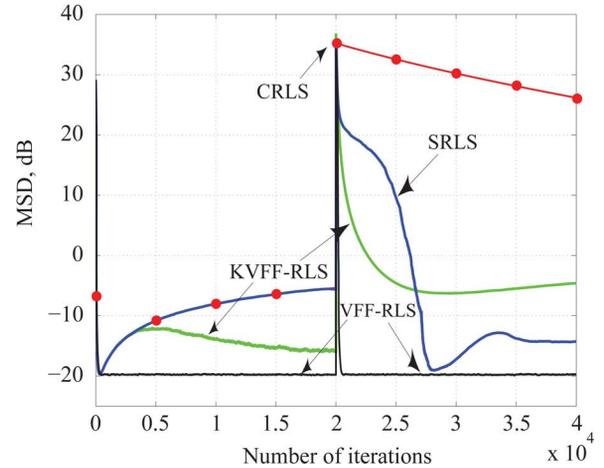


Fig. 2. Learning curves for a system identification application in a nonstationary environment with a SNR = 20 dB and  $\sigma_\phi^2 = 10^{-3}$  (Experiment 2).

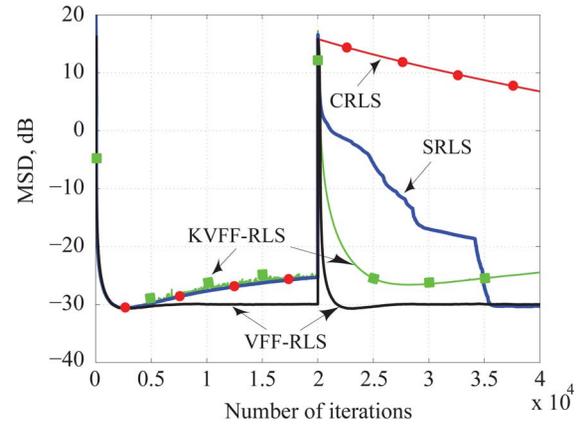


Fig. 3. Learning curves for a system identification application in a nonstationary environment with a SNR = 20 dB and  $\sigma_\phi^2 = 10^{-4}$  (Experiment 3).

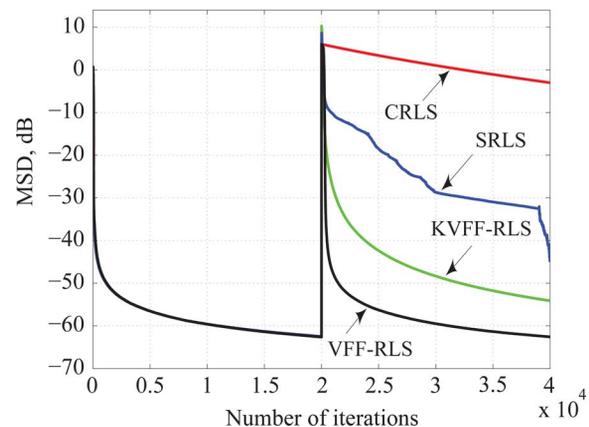


Fig. 4. Learning curves for a system identification application in a stationary environment with a SNR = 40 dB (Experiment 4).

dB). The MSD curves obtained by using the VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 4.

In the fifth experiment, we repeated the fourth experiment except that a nonstationarity was introduced as per the model given in (33) where  $\mathbf{h}_0$  was the same as in the first experiment and the elements of  $\phi_k$  were obtained from a white Gaussian noise signal with  $\sigma_\phi^2 = 10^{-4}$ . The MSD curves obtained by using the

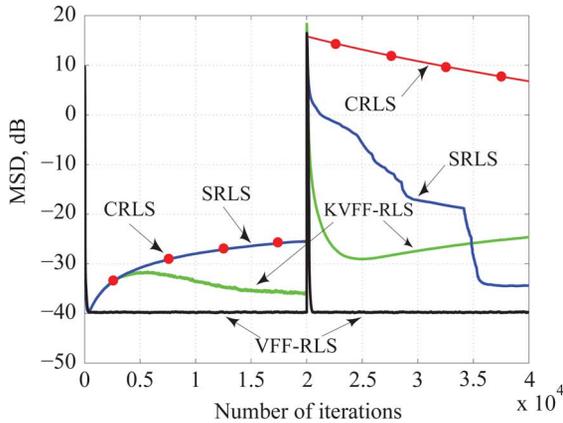


Fig. 5. Learning curves for a system identification application in a nonstationary environment with a SNR = 40 dB and  $\sigma_\phi^2 = 10^{-4}$  (Experiment 5).

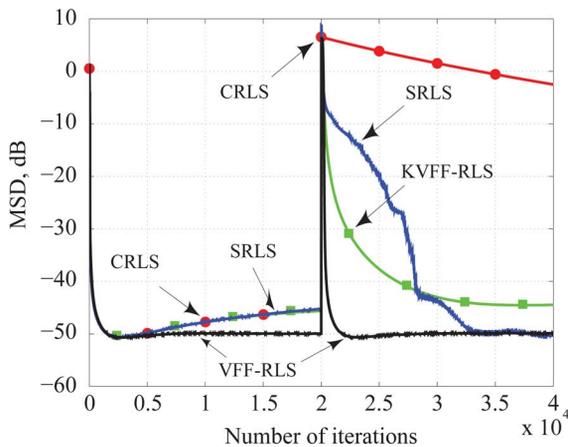


Fig. 6. Learning curves for a system identification application in a nonstationary environment with a SNR = 40 dB and  $\sigma_\phi^2 = 10^{-5}$  (Experiment 6).

VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 5.

In the sixth experiment, we repeated the fifth experiment except that we used  $\sigma_\phi^2 = 10^{-5}$ . The MSD curves obtained by using the VFF-RLS, CRLS, SRLS, and the KVFF-RLS algorithms are illustrated in Fig. 6.

As can be seen in Figs. 1–6, the proposed VFF-RLS algorithm outperforms the competing algorithms in all the experiments. Since there is no prespecified forgetting factor in the proposed algorithm, it performs better than the other algorithms in Markov-type nonstationary environments.

In the seventh experiment, we examined the effect of tuning integer  $c_1$  on the VFF-RLS algorithm. We repeated the sixth experiment for values of  $c_1$  of 2, 4, and 6. The MSD curves obtained are illustrated in Fig. 7. As can be seen from the MSD curve of the VFF-RLS algorithm in Fig. 6 and the MSD curves in Fig. 7, changes in tuning integer  $c_1$  in the range 2 to 8 do not significantly alter the performance of the VFF-RLS algorithm. In effect, the VFF-RLS algorithm is quite robust with respect to changes in  $c_1$ .

In the eighth experiment, we examined the effect of tuning integer  $c_1$  in a system-identification application in the case of the VCF-RLS algorithm and compared the performance achieved with that of the CRLS algorithm and LMS-Newton-II algorithm

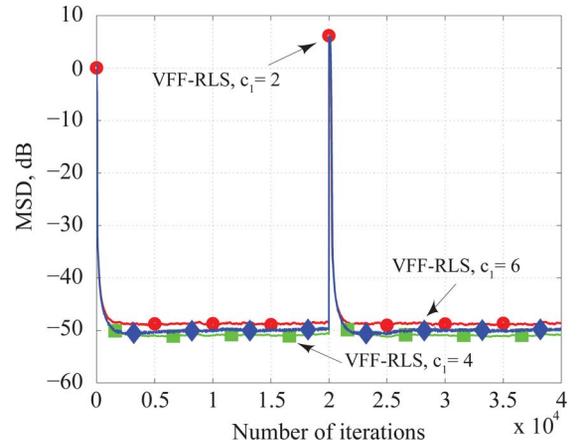


Fig. 7. Effect of changes in tuning integer  $c_1$  on the VFF-RLS algorithm in a system identification application (Experiment 7).

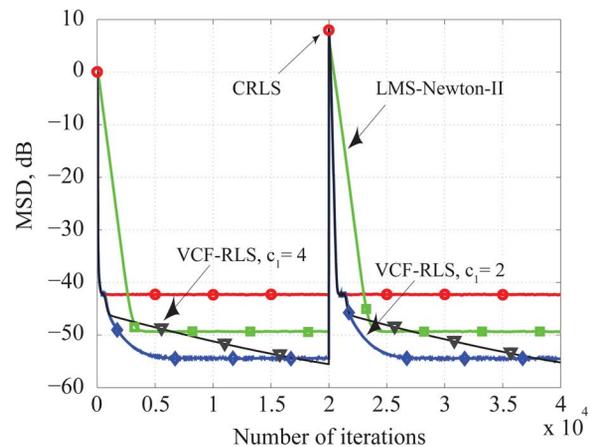


Fig. 8. Effect of changes in tuning integer  $c_1$  on the VCF-RLS algorithm in a system identification application and comparison with competing algorithms (Experiment 8).

described in [9]. The unknown system was obtained as per the model given in (33) where  $\mathbf{h}_0$  and the input signal were the same as in the first experiment and the elements of  $\phi_k$  were obtained from a white Gaussian noise signal with  $\sigma_\phi^2 = 10^{-6}$ . The measurement noise added to the desired signal was a white Gaussian noise signal with  $\sigma_v^2 = 10^{-4}$ . The forgetting factor  $\lambda$  in the CRLS and VCF-RLS algorithms was set to 0.99 in order to achieve the same readaptation capability. Parameters  $\sigma_0$ ,  $\beta$ , and  $t$  for the VCF-RLS algorithm given in Table I were set to 0, 0.99, and  $\sqrt{c_1 \sigma_v^2}$ , respectively. In the LMS-Newton-II algorithm, we used  $\alpha = 0.001$  and  $q = 0.1$ . The MSD curves obtained for the CRLS, VCF-RLS, and LMS-Newton-II algorithms are illustrated in Fig. 8. As can be seen, for the same readaptation capability the VCF-RLS algorithm yields a reduced steady-state misalignment as compared to the CRLS algorithm. The VCF-RLS algorithm, on the other hand, yields a faster convergence and a reduced steady-state misalignment as compared to the LMS-Newton-II algorithm.

### B. Channel-Equalization Application

In this section, the proposed VCF-RLS algorithm is compared with the CRLS and NLMS algorithms in a channel-equalization

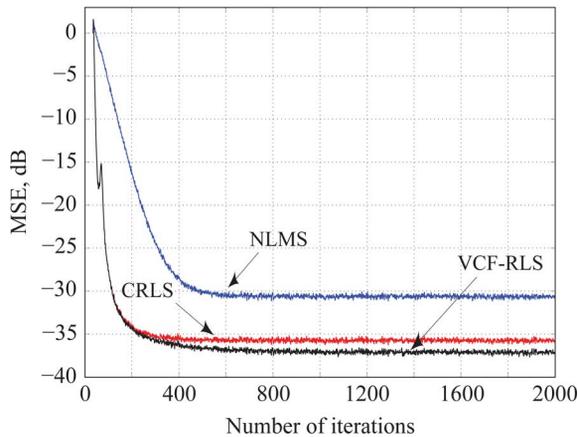


Fig. 9. Comparison of the VCF-RLS algorithm with the CRLS and NLMS algorithms for in channel equalization application using  $t = \sqrt{2M\sigma_v^2}$ ,  $c_1 = 2M$  in the VCF-RLS algorithm (Experiment 9).

application. The taps of the impulse response of the channel to be equalized were obtained as

$$\hat{h}_k = 0.1(0.5)^k$$

for  $k = 0, 1, \dots, 8$  [3]. The binary-phase-shift key (BPSK) input signal for each channel was obtained by using Bernoulli trials (see [3] and [27] for details about Bernoulli trials). The channel output was contaminated by using a white Gaussian noise signal with  $\sigma_v^2 = 10^{-4}$ . The length of the adaptive-filter weight vector was set to  $M = 60$ . The total delay introduced by the adaptive filter and the channel for the BPSK signal was  $L = 35$  [3]. The forgetting factor  $\lambda$  was set to 0.99 for both the VCF-RLS and CRLS algorithms. The MSE curves obtained in 5000 independent trials by using the VCF-RLS, NLMS, and the CRLS algorithms are illustrated in Fig. 9. As can be seen, the VCF-RLS algorithm yields a reduced steady-state misalignment as compared to the CRLS algorithm for the same convergence speed.

## VIII. CONCLUSION

Two new improved RLS algorithms, the VFF-RLS and VCF-RLS algorithms, have been developed. Both the variable forgetting factor and variable convergence factor are obtained by solving a mean-square noise-free *a posteriori* error minimization problem. Simulation results for a system-identification application were presented which show that the proposed VFF-RLS algorithm outperforms the CRLS, SRLS, and KVFF-RLS algorithms in terms of steady-state misalignment, tracking, and readaptation capability in both stationary and nonstationary environments. Simulation results for a channel-equalization application were presented which demonstrate that the proposed VCF-RLS algorithm offers improved performance compared to the CRLS and NLMS algorithms in terms of steady-state misalignment. A steady-state analysis of the proposed algorithms was also presented which led to expressions for the expected MSE in nonstationary and stationary environments and the validity of these expressions has been demonstrated by simulation results in a system-identification application.

## APPENDIX

From (12), we can write

$$\tau_{p,k} = 1 - \frac{\lambda}{\tau_k} + \left(\frac{\lambda}{\tau_k}\right)^2 - \dots$$

and since, in practice,  $\tau_k \gg \lambda$  for a large  $M$  all higher order terms can be neglected, i.e.,  $\tau_{p,k} \approx 1 - \lambda/\tau_k$ . With such a linearized  $\tau_{p,k}$ , we obtain  $E[\tau_{p,k}^2] \approx E[1 - 2\lambda/\tau_k]$  by neglecting the second-order term. As in [9], [28], and [29] it can be shown that  $E[1/\tau_k] \approx 1/M$  which becomes small for a large  $M$ ; hence we can obtain  $E[\tau_{p,k}]^2 = (E[1 - \lambda/\tau_k])^2 \approx 1 - 2E[\lambda/\tau_k]$  where we have again neglected the second-order term. Therefore, for a large  $M$  we get  $E[\tau_{p,k}^2] \approx E[\tau_{p,k}]^2$ .

## ACKNOWLEDGMENT

The authors are grateful to the Natural Sciences and Engineering Research Council of Canada for supporting this work.

## REFERENCES

- [1] A. Antoniou and W. S. Lu, *Practical Optimization*. New York: Springer, 2007.
- [2] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ: Wiley, 2003.
- [3] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. New York: Springer, 2008.
- [4] B. Toplis and S. Pasupathy, "Tracking improvements in fast RLS algorithms using a variable forgetting factor," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, pp. 206–227, Feb. 1988.
- [5] D. T. M. Slock and T. Kailath, "Fast transversal filters with data sequence weighting," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, pp. 346–359, Mar. 1989.
- [6] S.-H. Leung and C. F. So, "Gradient-based variable forgetting factor RLS algorithm in time-varying environments," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3141–3150, Aug. 2005.
- [7] C. Paleologu, J. Benesty, and S. Ciochina, "A robust variable forgetting factor recursive least-squares algorithm for system identification," *IEEE Signal Process. Lett.*, vol. 15, pp. 597–600, 2008.
- [8] D. T. M. Slock and T. Kailath, "Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 92–114, Jan. 1991.
- [9] P. S. R. Diniz, M. L. R. De Campos, and A. Antoniou, "Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor," *IEEE Trans. Signal Process.*, vol. 43, pp. 617–627, Mar. 1995.
- [10] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series With Engineering Applications*. Cambridge, MA: MIT Press, 1964, paperback ed.
- [11] S. C. Chan and Y. Zou, "A recursive least m-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.
- [12] L. R. Vega, H. Rey, J. Benesty, and S. Tressens, "A fast robust recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1209–1215, Mar. 2009.
- [13] M. Z. A. Bhotto and A. Antoniou, "Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environment," *IEEE Signal Process. Lett.*, vol. 18, no. 3, pp. 185–188, Mar. 2011.
- [14] M. Z. A. Bhotto and A. Antoniou, "Robust quasi-Newton adaptive filtering algorithms," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 8, pp. 537–541, Aug. 2011.
- [15] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3137–3149, Jul. 2008.
- [16] M. Zibulevsky and M. Elad, " $l_1 - l_2$  optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.
- [17] I. Daubechies, M. DeFrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.

- [18] K. Slavakis, Y. Kopsinis, and S. Theodoridis, "Revisiting adaptive leastsquares estimation and application to online sparse signal recovery," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4292–4295.
- [19] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted  $l_1$  balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.
- [20] S. Farahmand and G. B. Giannakis, "Robust RLS in the presence of correlated noise using outlier sparsity," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3308–3313, Jun. 2012.
- [21] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, pp. 1097–1110, Oct. 1986.
- [22] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.
- [23] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
- [24] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.
- [25] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
- [26] S. Werner and P. S. R. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Process. Lett.*, vol. 8, no. 8, pp. 231–235, Aug. 2001.
- [27] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*. New York: McGraw-Hill, 2000.
- [28] G. E. Bottomley and S. T. Alexander, "A novel approach for stabilizing recursive least squares filters," *IEEE Trans. Signal Process.*, vol. 39, pp. 1770–1779, Aug. 1991.
- [29] C. G. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, pp. 1177–1191, Oct. 1983.



**Md. Zulfikar Ali Bhotto** (M'07) obtained his B.Sc. in electrical and electronic engineering from the Department of Electrical and Electronic Engineering at Rajshahi University of Engineering and Technology, Bangladesh, in 2002. He obtained his Ph.D. in electrical and computer engineering from the Department of Electrical and Computer Engineering at the University of Victoria, Victoria, BC, Canada in 2012.

He worked as a Lecturer and Assistant Professor at Rajshahi University of Engineering and Technology, Bangladesh, from 2003 to 2007. He was awarded a University of Victoria fellowship for 2007–2008. Currently he is working as a Postdoctoral Research Associate in the Department of Electrical and Computer Engineering at the University of Victoria.



**Andreas Antoniou** (M'69–SM'79–F'82–LF'04) received the B.Sc.(Eng.) and Ph.D. degrees in electrical engineering from the University of London in 1963 and 1966, respectively.

He taught at Concordia University from 1970 to 1983, was the founding Chair of the Department of Electrical and Computer Engineering, University of Victoria, BC, Canada, from 1983 to 1990, and is now Professor Emeritus. His teaching and research interests are in the area of digital signal processing. He is the author of *Digital Signal Processing: Signals, Systems, and Filters*, McGraw-Hill, 2005, and the co-author with Wu-Sheng Lu of *Practical Optimization: Algorithms and Engineering Applications*, Springer, 2007.

Dr. Antoniou served first as Associate Editor and after that as Chief Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1983 to 1987, as a Distinguished Lecturer of the IEEE Signal Processing and the Circuits and Systems (CAS) Societies during 2003–2004 and 2006–2007, respectively, and as General Chair of the 2004 International Symposium on Circuits and Systems. He is a Fellow of the IET. He was awarded the CAS Golden Jubilee Medal by the IEEE Circuits and Systems Society, the B.C. Science Council Chairmans Award for Career Achievement for 2000, the Doctor Honoris Causa degree by the National Technical University, Athens, Greece, in 2002, the IEEE Circuits and Systems Society Technical Achievement Award for 2005, the 2008 IEEE Canada Outstanding Engineering Educator Silver Medal, the IEEE Circuits and Systems Society Education Award for 2009, the 2011 Craigdarroch Gold Medal for Career Achievement and the 2011 Legacy Award for Research both from the University of Victoria, Victoria, Canada.