

A Family of Shrinkage Adaptive-Filtering Algorithms

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

Abstract—A family of adaptive-filtering algorithms that uses a variable step size is proposed. A variable step size is obtained by minimizing the energy of the noise-free *a posteriori* error signal which is obtained by using a known L_1 - L_2 minimization formulation. Based on this methodology, a shrinkage affine projection (SHAP) algorithm, a shrinkage least-mean-squares (SHLMS) algorithm, and a shrinkage normalized least-mean-squares (SHNLMS) algorithm are proposed. The SHAP algorithm yields a significantly reduced steady-state misalignment as compared to the conventional affine projection (AP), variable-step-size AP, and set-membership AP algorithms for the same convergence speed although the improvement is achieved at the cost of an increase in the average computational effort per iteration in the amount of 11% to 14%. The SHLMS algorithm yields a significantly reduced steady-state misalignment and faster convergence as compared to the conventional LMS and variable-step-size LMS algorithms. Similarly, the SHNLMS algorithm yields a significantly reduced steady-state misalignment and faster convergence as compared to the conventional normalized least-mean-squares (NLMS) and set-membership NLMS algorithms.

Index Terms—Adaptive-filtering algorithms, affine projection algorithms, least-mean-squares algorithms, normalized least-mean-squares algorithms, set-membership algorithms.

I. INTRODUCTION

LEAST-MEAN-SQUARES (LMS) algorithms are widely used due to their simplicity [1], [2]. The variable-step-size LMS (VLMS) and normalized LMS (NLMS) algorithms reported in [3] and [4], respectively, offer improved performance compared to the conventional LMS algorithm. The non-parametric NLMS (NPNLMS) and the set-membership NLMS (SMNLMS) algorithms reported in [5] and [6], respectively, offer improved performance relative to the conventional NLMS algorithm. Unfortunately, these algorithms do not perform well in situations where the input signal is colored [1], [2]. In such situations, the affine projection (AP) algorithm yields improved performance as compared to the NLMS algorithm [7].

The AP algorithm in [7] uses the L most recent input-signal vectors, where L is also known as the *projection order*, in updating the weight vector whereas the NLMS algorithm uses only the current input-signal vector. As a result, the AP algorithm in

[7] offers improved performance with respect to the NLMS algorithm in [4]. Several variants of the AP algorithm have been proposed in [8]–[13]. The AP algorithms in [8] and [9], known as the set-membership AP (SMAP) and variable-step-size AP (VSSAP) algorithms, respectively, yield a reduced steady-state misalignment relative to the AP algorithm in [7] for similar convergence speed. On the other hand, the variable-regularization AP algorithm in [10] offers a significantly reduced steady-state misalignment compared to the AP algorithm for the same convergence speed. The AP algorithm in [11] employs a so-called *reuse time* whereby each input-signal vector is retained in the input-signal matrix over a specified time and by using a variable reuse time, reduced steady-state misalignment can be achieved. An additional feature of the SMAP algorithm is that it entails a reduced average computational complexity due to its data-selective feature. Some other variants of the SMAP algorithm are described in [12] and [13]. The algorithm in [12] yields a slightly improved steady-state misalignment and the algorithm in [13] yields improved convergence speed for sparse system-identification applications as compared to the SMAP algorithm. The binormalized data-reusing and set-membership binormalized data-reusing LMS algorithms in [14] and [15], respectively, are derived for a projection order $L = 2$. These algorithms are alternative implementations of the AP and SMAP algorithms that offer reduced computational complexity compared to the direct implementations of the AP and SMAP algorithms with $L = 2$. A rigorous analysis of the AP algorithm in [7] was presented in [16]. Note that information about the variance of the measurement noise needs to be available *a priori* for the successful operation of the AP algorithms [8]–[13] and [15].

In this paper, we propose a family of so-called *shrinkage* adaptive-filtering algorithms, namely, a shrinkage AP (SHAP) algorithm, a shrinkage LMS (SHLMS) algorithm, and a shrinkage NLMS (SHNLMS) algorithm. The proposed algorithms use a variable step size which is obtained by minimizing the energy of the noise-free *a posteriori* error signal. The noise-free error signal is obtained by using an L_1 - L_2 minimization method described in [17], [18]. Simulation results in a system-identification application and an acoustic echo-cancellation application are used to demonstrate the superior performance of the proposed SHAP, SHLMS, and SHNLMS algorithms in terms of reduced steady-state misalignment as compared to the performance achieved with corresponding state-of-the-art competing algorithms reported in [3]–[9]. Simulation results for the SHAP algorithm show that the improved performance comes about at the cost of an increase in the average computational effort per iteration over that in the known AP algorithm in the amount of 11 to 14%.

The paper is organized as follows. In Sections II and III, we discuss the derivation of the proposed SHAP, SHLMS, and SHNLMS algorithms. Issues concerning the application

Manuscript received December 06, 2011; revised August 15, 2012 and November 08, 2012; accepted December 03, 2012. Date of publication December 28, 2012; date of current version March 12, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Ut-Va Koc. The authors are grateful to the Natural Sciences and Engineering Research Council of Canada for supporting this work.

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/TSP.2012.2236831

of the proposed algorithms are discussed in Section IV. The computational complexity of the SHAP algorithm is discussed in Section V. Simulation results are presented in Section VI and conclusions are drawn in Section VII.

II. SHRINKAGE AFFINE-PROJECTION ALGORITHMS

In the case of system-identification adaptive-filtering applications, the desired signal samples d_k are obtained as

$$d_k = d_{f,k} + v_k \quad (1)$$

where $d_{f,k} = \mathbf{x}_k^T \mathbf{w}_{\text{opt}}$ is the noise-free desired signal, v_k is the measurement noise signal, $\mathbf{w}_{\text{opt}} \in \mathcal{R}^{M \times 1}$ is the impulse response of the unknown system, and $\mathbf{x}_k \in \mathcal{R}^{M \times 1}$ is the input-signal vector.

In the conventional AP algorithm [7], the weight-vector update equation assumes the form

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mu \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} \mathbf{e}_k \quad (2)$$

where μ is the step size, $\mathbf{X}_k = [\mathbf{x}_k \ \mathbf{x}_{k-1} \ \cdots \ \mathbf{x}_{k-L+1}] \in \mathcal{R}^{M \times L}$ is the input-signal matrix, and $\mathbf{e}_k = [e_k \ \varepsilon_{k-1} \ \cdots \ \varepsilon_{k-L+1}]^T$ is the *a priori* error-signal vector where $e_k = d_k - \mathbf{x}_k^T \mathbf{w}_{k-1}$ and $\varepsilon_{k-i} = d_{k-i} - \mathbf{x}_{k-i}^T \mathbf{w}_{k-1}$ is the *a posteriori* error at iteration $k-1$. Vector \mathbf{e}_k can also be expressed as

$$\mathbf{e}_k = \mathbf{e}_{f,k} + \mathbf{v}_k \quad (3)$$

where

$$\mathbf{e}_{f,k} = \mathbf{X}_k^T (\mathbf{w}_{\text{opt}} - \mathbf{w}_{k-1}) \quad (4)$$

is the noise-free *a priori* error vector and $\mathbf{v}_k = [v_k \ v_{k-1} \ \cdots \ v_{k-L+1}]^T$ contains the measurement noise signal samples. Similarly, the *a posteriori* error vector can be expressed as

$$\boldsymbol{\epsilon}_k = \boldsymbol{\epsilon}_{f,k} + \mathbf{v}_k \quad (5)$$

where

$$\boldsymbol{\epsilon}_{f,k} = \mathbf{X}_k^T (\mathbf{w}_{\text{opt}} - \mathbf{w}_k) \quad (6)$$

is the noise-free *a posteriori* error vector.

In the conventional AP algorithm, \mathbf{w}_k is obtained by minimizing the mean-square error (MSE), $\|\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_k\|^2$. On the other hand, in the Wiener solution, which is the optimal solution [1], [2], \mathbf{w}_k is obtained by minimizing the energy of the *a posteriori* error signal, $E[(d_k - \mathbf{x}_k^T \mathbf{w}_r)^2]$. The two solutions are obviously different and in order to bring the weight vector \mathbf{w}_k closer to the Wiener solution, the step size μ in (2) can be optimized.

The update equation in (2) yields the noise-free *a posteriori* error vector as

$$\boldsymbol{\epsilon}_{f,k} = (1 - \mu) \boldsymbol{\epsilon}_{f,k} - \mu \mathbf{v}_k. \quad (7)$$

Taking the expectation of the square of the L_2 norm of $\boldsymbol{\epsilon}_{f,k}$ in (7), we obtain

$$E[\|\boldsymbol{\epsilon}_{f,k}\|^2] = (1 - \mu)^2 E[\|\boldsymbol{\epsilon}_{f,k}\|^2] + \mu^2 E[\|\mathbf{v}_k\|^2]. \quad (8)$$

We can now minimize $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ in (8) with respect to μ in order to obtain a solution that is closer to the optimal Wiener solution. The optimum value of μ can be obtained as

$$\mu_k = \frac{E[\|\boldsymbol{\epsilon}_{f,k}\|^2]}{E[\|\boldsymbol{\epsilon}_{f,k}\|^2] + E[\|\mathbf{v}_k\|^2]} \quad (9)$$

by setting the derivative of $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ with respect to μ to zero. As can be seen in (9), the optimum step size, μ_k , will lie in the range (0,1). Since $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$, where $\boldsymbol{\epsilon}_{f,k}$ is the most recent element in the *a priori* error vector in (4), is a measure of the excess mean-square error (EMSE) of the AP algorithm [16], the minimum EMSE can be obtained by using (9) in (2).

One obvious difficulty in computing the step size in (9) is to obtain $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$. Although the time average of the squares of $\boldsymbol{\epsilon}_{f,k}$, i.e.,

$$\sigma_{e,f,k}^2 = \lambda \sigma_{e,f,k-1}^2 + (1 - \lambda) \|\boldsymbol{\epsilon}_{f,k}\|^2 \quad (10)$$

where $0 \ll \lambda < 1$ is the forgetting factor, can be used to replace the statistical mean $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ in (9), the problem cannot be solved as $\boldsymbol{\epsilon}_{f,k}$ is unknown. An easy solution would be to recover $\boldsymbol{\epsilon}_{f,k}$ from the noisy *a priori* error vector \mathbf{e}_k in (3). Once $\boldsymbol{\epsilon}_{f,k}$ is known, it can be used in (10) to obtain an estimate of $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$. On the other hand, one can obtain $E[\|\mathbf{v}_k\|^2] = L\sigma_v^2$, where L is the projection order. With $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ and $E[\|\mathbf{v}_k\|^2]$ known, the step size in (9) can be evaluated. Since the L_2 norm of $\boldsymbol{\epsilon}_{f,k}$ is used in (10), μ_k cannot approach zero unless the adaptive filter reaches steady state.

A. Shrinkage Denoising Method

Vector $\boldsymbol{\epsilon}_{f,k}$ can be recovered from the noisy *a priori* error vector \mathbf{e}_k by using a method described in [17], [18]. This method involves the minimization problem

$$\underset{\mathbf{a}_k}{\text{minimize}} \quad (t \|\mathbf{a}_k\|_1 + 0.5 \|\mathbf{D}\mathbf{a}_k - \mathbf{e}_k\|^2) \quad (11)$$

where $\|\mathbf{a}_k\|_1$ is the L_1 norm of vector \mathbf{a}_k , $\mathbf{D} \in \mathcal{R}^{L \times L}$ is an orthonormal transform matrix, $\mathbf{D}\mathbf{a}_k$ is an estimate of $\boldsymbol{\epsilon}_{f,k}$, \mathbf{e}_k is given by (3), and t is an empirical constant known as the threshold parameter. This type of optimization problem has been used to solve image denoising applications in [17], [18]. Its solution can be obtained as

$$\mathbf{a}_k = \text{sign}(\mathbf{a}_{o,k}) \odot \hat{\mathbf{a}}_k \quad (12)$$

where $\mathbf{a}_{o,k} = \mathbf{D}^T \mathbf{e}_k$, $\hat{\mathbf{a}}_k$ is a vector whose r th element is given by $\max(|a_{o,k}(r)| - t, 0)$, and \odot denotes the element-wise product. Signal $\mathbf{a}_{o,k}$ essentially represents \mathbf{e}_k in terms of the column vectors of \mathbf{D} . With \mathbf{a}_k in (12) known, an estimate of $\boldsymbol{\epsilon}_{f,k}$ in (3) can be obtained as $\hat{\boldsymbol{\epsilon}}_{f,k} = \mathbf{D}\mathbf{a}_k$. This can then be used in (10) to obtain an estimate of $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ which can be used in (9).

In the next subsection, the required update equations for the SHAP algorithm are deduced.

B. Update Equations for SHAP Algorithm

An important issue associated with the SHAP algorithm involves the choice of a suitable value for the threshold parameter

t . Assuming a white Gaussian noise signal \mathbf{v}_k and $\mathbf{D} = \mathbf{I}$ and then choosing the threshold parameter as $t = \sqrt{L\sigma_v^2}$ in (11) would give

$$\|\mathbf{D}\mathbf{a}_k - \mathbf{e}_k\|^2 \approx L\sigma_v^2 \quad (13)$$

for all k . Taking the expectation of both sides of (13) we obtain

$$E[\|\mathbf{D}\mathbf{a}_k - \mathbf{e}_k\|^2] \approx L\sigma_v^2 \quad (14)$$

For a white Gaussian noise signal with variance σ_v^2 , from (3) we obtain

$$E[\|\mathbf{e}_{f,k} - \mathbf{e}_k\|^2] = L\sigma_v^2. \quad (15)$$

Comparing (14) and (15), we can see that $\mathbf{D}\mathbf{a}_k$ would be approximately equal to $\mathbf{e}_{f,k}$. In effect, the noise signal would be eliminated and we would obtain $\mathbf{D}\mathbf{a}_k \approx \mathbf{e}_{f,k}$. The above formula could be used to obtain a good initial value for t which can be tuned to suit the application at hand.

The update equations for the SHAP algorithm can be summarized as follows:

$$\begin{aligned} t &= \sqrt{L\sigma_v^2} \\ \mathbf{a}_{o,k} &= \mathbf{D}^T \mathbf{e}_k \\ \mathbf{a}_k &= \text{sign}(\mathbf{a}_{o,k}) \odot \hat{\mathbf{a}}_k \\ \hat{\mathbf{e}}_{f,k} &= \mathbf{D}\mathbf{a}_k \\ \sigma_{e,f,k}^2 &= \lambda \sigma_{e,f,k-1}^2 + (1 - \lambda) \|\hat{\mathbf{e}}_{f,k}\|^2 \\ \mu_k &= \frac{\sigma_{e,f,k}^2}{\sigma_{e,f,k}^2 + L\sigma_v^2} \\ \mathbf{w}_k &= \mathbf{w}_{k-1} + \mu_k \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} \mathbf{e}_k. \end{aligned} \quad (16)$$

Using $L = 1$ in the SHAP algorithm, we obtain the SHNLMS algorithm as a special case.

As in (15) we can also obtain from (3) the energy relation of the noisy and noise-free *a priori* error signals as

$$E[\|\mathbf{e}_k\|^2] = E[\|\mathbf{e}_{f,k}\|^2] + L\sigma_v^2. \quad (17)$$

As can be seen, $E[\|\mathbf{e}_{f,k}\|^2]$ can be obtained directly as $E[\|\mathbf{e}_{f,k}\|^2] = E[\|\mathbf{e}_k\|^2] - L\sigma_v^2$ and it can then be used in (9) to obtain μ_k . However, since we have to use the time average of $\|\mathbf{e}_k\|^2$, i.e., $E[\|\mathbf{e}_k\|^2]$, instead of its statistical average in (9) and the relation in (17) does not hold true for the time average, the algorithm can become unstable as μ_k can assume values outside the range (0,1).

The energy of the *a posteriori* error signal in (5) for the shrinkage AP algorithm becomes

$$E[\|\boldsymbol{\epsilon}_k\|^2] = \frac{(E[\|\mathbf{v}_k\|^2])^2}{E[\|\mathbf{e}_{f,k}\|^2] + E[\|\mathbf{v}_k\|^2]} \quad (18)$$

which is less than $E[\|\mathbf{v}_k\|^2]$. Note that $\boldsymbol{\epsilon}_{f,k}$ in (6) is not independent of \mathbf{v}_k as can be seen from (1)–(3). As a result, a relation such as that in (17) cannot be obtained by using (5). Hence, we

have $E[\|\boldsymbol{\epsilon}_k\|^2] < E[\|\mathbf{v}_k\|^2]$ even though $E[\|\boldsymbol{\epsilon}_{f,k}\|^2]$ is a positive quantity for all L . For the NPNLMS algorithm in [5], the energy of the *a posteriori* error signal becomes

$$E[\boldsymbol{\epsilon}_k^2] = pE[v_k^2] + (1 - p)E[e_k^2] > E[v_k^2]$$

where p is the probability of update. In adaptive-filter theory, $E[\|\mathbf{e}_{f,k}\|^2] \neq 0$ for $\mu \neq 0$ and, therefore, from (18) we have $E[\boldsymbol{\epsilon}_k^2] < E[v_k^2]$. A similar statement holds true for the SHAP algorithm with respect to the AP algorithm in [19], which was designed for acoustic echo-cancellation applications.

In the next section, we describe the shrinkage LMS algorithm.

III. SHRINKAGE LMS ALGORITHM

The update equation for the conventional LMS algorithm [1], [2] is

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mu e_k \mathbf{x}_k \quad (19)$$

where μ is the step size. The noise-free *a posteriori* error signal for this algorithm becomes

$$\boldsymbol{\epsilon}_{f,k} = (1 - \mu \mathbf{x}_k^T \mathbf{x}_k) e_{f,k} - \mu \mathbf{x}_k^T \mathbf{x}_k v_k. \quad (20)$$

Taking the expectation of the square of the noise-free *a posteriori* error signal in (20), we obtain

$$E[\boldsymbol{\epsilon}_{f,k}^2] = E[(1 - \mu \mathbf{x}_k^T \mathbf{x}_k)^2 e_{f,k}^2] + \mu^2 E[(\mathbf{x}_k^T \mathbf{x}_k)^2] E[v_k^2] \quad (21)$$

where we assume that v_k is an independent and identically distributed white Gaussian noise signal. In order to proceed further, we neglect the dependence of $e_{f,k}$ on the input signal \mathbf{x}_k as the amplitudes of $e_{f,k}$ become quite small at steady state. Under these circumstances, (21) can be simplified as

$$E[\boldsymbol{\epsilon}_{f,k}^2] = E[1 - 2\mu \mathbf{x}_k^T \mathbf{x}_k + \mu^2 (\mathbf{x}_k^T \mathbf{x}_k)^2] E[e_{f,k}^2] + \mu^2 E[(\mathbf{x}_k^T \mathbf{x}_k)^2] E[v_k^2]. \quad (22)$$

Setting the derivative of $E[\boldsymbol{\epsilon}_{f,k}^2]$ in (22) with respect to μ to zero, we obtain

$$\mu_k = \frac{1}{E[\|\mathbf{x}_k\|^2]} \frac{E[e_{f,k}^2]}{E[e_{f,k}^2] + \sigma_v^2}. \quad (23)$$

In the derivation of the above expression, we have assumed that $E[(\mathbf{x}_k^T \mathbf{x}_k)^2] \approx E[\mathbf{x}_k^T \mathbf{x}_k]^2$ (see Appendix) in order to obtain a simple expression for μ_k and thus preserve the simplicity of the LMS algorithm in the SHLMS algorithm. Since the *a priori* error signal e_k is a scalar quantity, $e_{f,k}$ can be recovered from e_k by using (11) as

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

Following the discussion in Section II.B, the threshold parameter t should be chosen as $t = \sigma_v$ where σ_v^2 is the variance of the noise signal. However, since the dependence of $e_{f,k}$ on the input signal \mathbf{x}_k was neglected in deriving (23), the step size in

(23) does not assure the minimization of $E[\epsilon_{f,k}^2]$. As a result, fine tuning of threshold t around σ_v could yield improved performance. From our extensive simulation results, we found out that $t = \sqrt{Q\sigma_v^2}$ with $Q = 1$ to 4 works well. Based on these principles, the update equations of the shrinkage LMS algorithm become

$$\begin{aligned} \hat{e}_{f,k} &= \text{sign}(e_k) \odot \max(|e_k| - t, 0) \\ \sigma_{e,f,k}^2 &= \lambda \sigma_{e,f,k-1}^2 + (1 - \lambda) \hat{e}_{f,k}^2 \end{aligned} \quad (24)$$

$$\begin{aligned} \mu_k &= \frac{q}{E[\|\mathbf{x}_k\|^2] \sigma_{e,f,k}^2 + \sigma_v^2} \\ \mathbf{w}_k &= \mathbf{w}_{k-1} + \mu_k \mathbf{x}_k e_k \end{aligned} \quad (25)$$

where we have used $\sigma_{e,f,k}^2$ in place of $E[\epsilon_{f,k}^2]$ in (23) for the step size μ_k in (25) and $q \in (0, 1)$. In many practical applications, $E[\|\mathbf{x}_k\|^2]$ is known. For a white Gaussian input signal, $E[\|\mathbf{x}_k\|^2]$ can easily be obtained as $M\sigma_x^2$ where σ_x^2 is the variance of the input signal.

IV. DISCUSSION

A key requirement of the proposed SHAP and SHLMS algorithms is that the variance of the noise should be available.

Some recent applications of adaptive filtering, which include adaptive mobile networks, source localization, environment monitoring, etc., [20]–[24], require a set of sensors distributed over a region and each sensor requires an adaptive filter. Two topologies, namely, fusion and network topologies, are used for distributed computing. In the fusion topology, the central sensor can be used to obtain the noise variance and then transmit the information to all the other sensors. In the network topology, each sensor can be used to estimate the noise variance during the communication interval with the neighboring sensors. In acoustic echo-cancellation applications, on the other hand, the noise variance can be estimated during off periods in the speech signal. With the noise variance known, we can use the threshold parameter as discussed in Section II.B.

In situations where the variance of the noise is not available, an optimal value of parameter t in (11) can be estimated by using a technique based on the dichotomous search described in ([25], p. 82), as detailed below:

1. Let t_l and t_u be sufficiently small and sufficiently large values, respectively, such that the optimal value of t , t_{opt} , lies in the range t_l to t_u .
2. Obtain threshold $t_k = (t_l + t_u)/2$.
3. Find the solutions of the problem in (11) at thresholds $t_a = t_k - 0.5\epsilon$ and $t_b = t_k + 0.5\epsilon$ where ϵ is a small positive number, say, 10^{-6} .
4. Compute the residual errors r_a and r_b for the solutions obtained in step 3.
5. If $r_a < r_b$, use $t_l = t_l$ and $t_u = t_b$. Otherwise, use $t_l = t_a$ and $t_u = t_u$.
6. If $|t_u - t_l| \leq \epsilon$, obtain $t_{\text{opt}} = t_l$. Otherwise, repeat from step 2.
7. Obtain the optimal values of t for $k = 1, 2, \dots, N_s$, where N_s is the number of samples in the input signal; then compute the average optimal value of t , \bar{t}_{opt} .

The number of samples in the input signal would tend to depend on the application but a value of $N_s = 100$ would be typical. The technique would be applied once at the time of the deployment of the adaptive filter.

Since the proposed shrinkage algorithms are actually variants of the basic LMS, NLMS, and AP algorithms, they are very reliable.

V. COMPUTATIONAL COMPLEXITY

The numbers of additions and multiplications for the AP algorithm can be obtained from the update equation

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mu_k \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} \mathbf{e}_k$$

as $L^3 + (M+1)L^2 + (2M-1)L$ and $L^3 + ML^2 + (2M+1)L$, respectively, (see ([2], p. 244)) and, similarly, for the SMAP algorithm using the update equation

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \max\left(0, 1 - \frac{\gamma}{|e_k|}\right) \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} \hat{\mathbf{e}}_k \quad (26)$$

where $\hat{\mathbf{e}}_k = [e_k \ 0 \ \dots \ 0]^T$ we obtain $L^3 + (M+1)L^2 + (M-1)L + M + 1$ and $L^3 + ML^2 + (M+1)L + M + 1$ additions and multiplications, respectively. On the other hand, using (16) with $\mathbf{D} \neq \mathbf{I}$ it can be shown that $L^3 + (M+3)L^2 + (2M-1)L + 1$ additions and $L^3 + (M+2)L^2 + 2(M+1)L + 3$ multiplications are required for the SHAP algorithm.

From the above formulas for the arithmetic operations, it can be readily shown that the proposed SHAP algorithm requires $2L^2 + 1$ more additions and $2L^2 + L + 3$ more multiplications than the AP algorithm or $2L^2 + ML - M + 1$ more additions and $2L^2 + (M+1)L - M + 2$ more multiplications than the SMAP algorithm. In effect, the proposed SHAP algorithm entails an increased computational complexity of the order of L^2 relative to the AP and SMAP algorithms.

VI. SIMULATION RESULTS

In this section, we present simulation results for the proposed algorithms as well as corresponding state-of-the-art competing algorithms in two adaptive-filtering applications, namely, a system-identification application and an acoustic echo-cancellation application. The proposed SHAP algorithm is compared with the NLMS, AP, VSSAP [9], and SMAP [8] algorithms. The proposed SHLMS algorithm is compared with the conventional LMS and VLMS [3] algorithms. The proposed SHNLMS algorithm is compared with the NLMS, SMNLMS [6], and NPNLMS [5] algorithms. A regularization matrix $\delta \mathbf{I}$ with $\delta = 10^{-8}$ was added to matrix $\mathbf{X}_k^T \mathbf{X}_k$ in (2) in all experiments to assure its invertibility for all L . The initial weight vector \mathbf{w}_0 was set to the zero vector in all algorithms and experiments. The error bound for the SM algorithms was set to $\gamma = \sqrt{5\sigma_v^2}$ [5], [8] in all experiments. Unless otherwise stated, the orthonormal matrix \mathbf{D} , forgetting factor λ , and $\sigma_{e,f,0}$ for the SHAP algorithm were chosen as the discrete cosine transform matrix, 0.90, and 0, respectively, in all experiments. The learning curves were obtained by averaging the MSE or the mean-square deviation (MSD), defined as

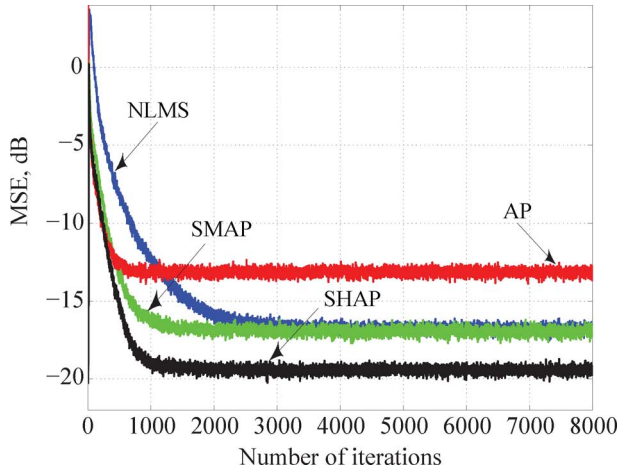


Fig. 1. Learning curves for SHAP, SMAP, AP, and NLMS algorithms with $L = 2$, $t = \sqrt{L\sigma_v^2}$, and SNR = 20 dB (Experiment 1).

$MSD = 20 \log_{10}(\|\mathbf{w}_{opt} - \mathbf{w}_k\|/\|\mathbf{w}_{opt}\|)$, over 1000 trials in each experiment.

A. System-Identification Application

A series of experiments were carried out in a system-identification application where the unknown system was an FIR filter whose impulse response was obtained as $\mathbf{w}_{opt} = \text{fir1}(M - 1, f_c)$ using MATLABTM, where M is the filter order and f_c is the normalized cut-off frequency which was set to 0.3. The elements of the unknown \mathbf{w}_{opt} were normalized to ensure that \mathbf{w}_{opt} has unit norm. Nine experiments were performed as detailed below.

In Experiment 1, the order of the unknown system was set to 27 and the input signal was a white Gaussian noise signal with unity variance. The input signal was colored by using an IIR filter with transfer function [8]

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

The measurement noise added to the desired signal was also a white Gaussian noise signal with variance $\sigma_v^2 = 10^{-2}$ which corresponds to an SNR of 20 dB. The projection order L was set to 2 in all AP-type algorithms. The learning curves obtained with the SHAP, SMAP, AP, and NLMS algorithms are illustrated in Fig. 1. As can be seen, the proposed SHAP algorithm yields a significantly reduced steady-state misalignment for the same convergence speed as compare to the competing algorithms.

In Experiment 2, the order of the unknown system and projection order were changed to 37 and 4, respectively. In addition, the variance of the measurement noise was changed to 10^{-4} (SNR = 40 dB). The learning curves obtained are illustrated in Fig. 2. As can be seen, the proposed SHAP algorithm yields a significantly reduced steady-state misalignment as compared to the competing algorithms. The NLMS algorithm yields a reduced steady-state misalignment as compared to the AP and SMAP algorithms due to the increased projection order but the speed of convergence is also reduced.

In Experiment 3, we reduced the degree of correlation of the input signal of the adaptive filter by using an IIR filter with a single pole at 0.95 to filter the unity-variance white Gaussian signal. The order of the unknown system was set to 111 and the

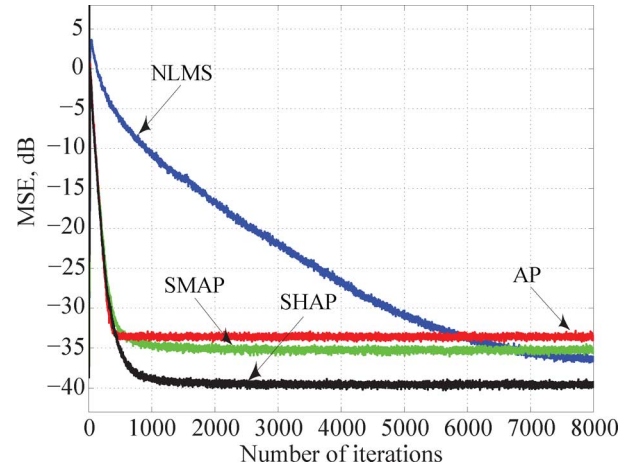


Fig. 2. Learning curves for SHAP, SMAP, AP, and NLMS algorithms with $L = 4$, $t = \sqrt{L\sigma_v^2}$, and SNR = 40 dB (Experiment 2).

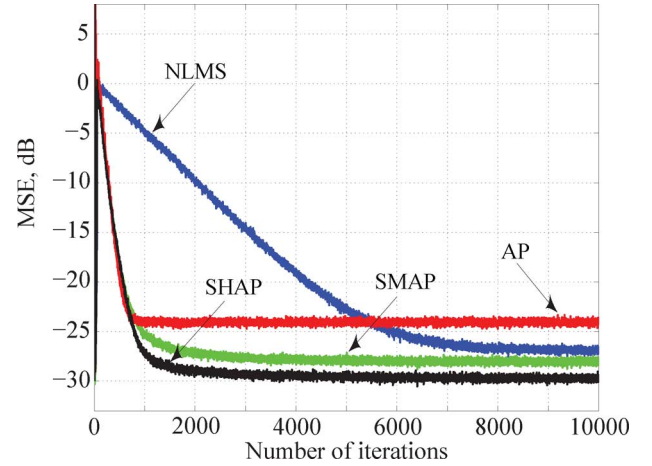


Fig. 3. Learning curves for SHAP, SMAP, AP, and NLMS algorithms with $L = 6$, $t = \sqrt{L\sigma_v^2}$, and SNR = 30 dB (Experiment 3).

variance of the measurement noise was set to 10^{-3} (SNR = 30 dB). The projection order was set to $L = 6$ in all algorithms. The learning curves obtained are illustrated in Fig. 3. As can be seen, the SHAP algorithm yields a much reduced steady-state misalignment for the same convergence speed as compared to the other AP algorithms.

In Experiment 4, we repeated Experiment 3 with $M = 63$ and $L = 3$ and we set the variance of the measurement noise to 0.0316 (SNR = 15 dB). The learning curves obtained are illustrated in Fig. 4. As can be seen, the SHAP algorithm yields a reduced steady-state misalignment for the same convergence speed as compared to the other AP algorithms.

In the above four experiments, we computed the average CPU time using the formula

$$\tau_{av}(j) = \frac{1}{K} \sum_{i=1}^K \tau(i, j)$$

for $j = 1, 2, \dots, J$ where $\tau(i, j)$ denotes the required CPU time for the j th iteration in the i th trial where J is the total number of iterations and K is the total number of trials. The CPU time $\tau(i, j)$ was measured by using command `cputime` of MATLABTM 2011b. The simulations were carried out using an

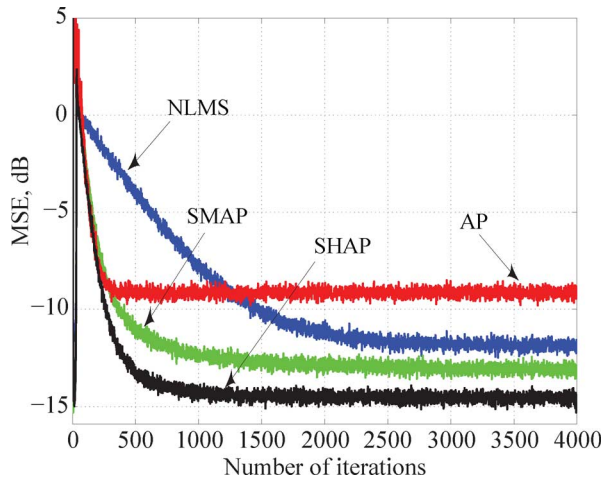


Fig. 4. Learning curves for SHAP, SMAP, AP, and NLMS algorithms with $L = 3$, $t = \sqrt{L\sigma_v^2}$, and SNR = 15 dB (Experiment 4).

TABLE I
AVERAGE CPU TIME PER ITERATION (MICROSECONDS)

| Algs. | Exp #1 | Exp #2 | Exp #3 | Exp #4 |
|-------|--------|--------|--------|--------|
| NLMS | 39 | 42 | 47 | 44 |
| AP | 55 | 67 | 96 | 68 |
| SMAP | 8 | 17 | 13 | 7 |
| SHAP | 64 | 78 | 108 | 79 |

Intel(R) Core(TM)2 CPU 6400 @2.13 GHz processor. The average value of τ_{av} , i.e., $\tau_{av,avg} = 1/J \sum_{j=1}^J \tau_{av}(j)$, referred to hereafter as the average CPU time per iteration, for the NLMS, AP, SMAP and SHAP algorithms in Experiments 1 to 4 are given in Table I. As can be seen, the reduced steady-state misalignment of the proposed SHAP algorithm comes at the cost of an increased average CPU time per iteration, of the order of 11 to 14%, relative to that of the AP algorithm. The average CPU time per iteration for the NLMS algorithm is approximately 39 to 56% that of the SHAP algorithm but the required computational effort to achieve convergence is significantly larger than those required by the competing algorithms as the NLMS algorithm is very slow to converge. Furthermore, the NLMS algorithm leads to a large steady-state misalignment. The number of updates in the SMAP algorithm in Experiments 1 to 4 were 4236, 6208, 4177, and 2642, respectively. The number of updates in the AP, NLMS, and SHAP algorithms was equal to the number of iterations, i.e., 8000 in Experiments 1 and 2, 10,000 in Experiment 3, and 4000 in Experiment 4. As can be seen in Table I, the SMAP algorithm yields the lowest $\tau_{av,avg}$. However, a lower $\tau_{av,avg}$ in the SMAP algorithm only indicates that it entails a low computational load compared to the other algorithms. In other words, if we were to increase J from 8000 to 12000 in Experiment 1, the number of updates in the SMAP algorithm would stay approximately the same, i.e., around 4236, and, therefore, $\tau_{av,avg} = 1/J \sum_{j=1}^J \tau_{av}(j)$ would be lower than that given in Table I but it would remain the same for the other algorithms.

The SHAP algorithm entails an increased average CPU time per iteration as compared to the AP algorithm due to the increased computational complexity associated with the SHAP algorithm as discussed in Section V. The average CPU time per iteration for the SMAP algorithm is lower as most of the time

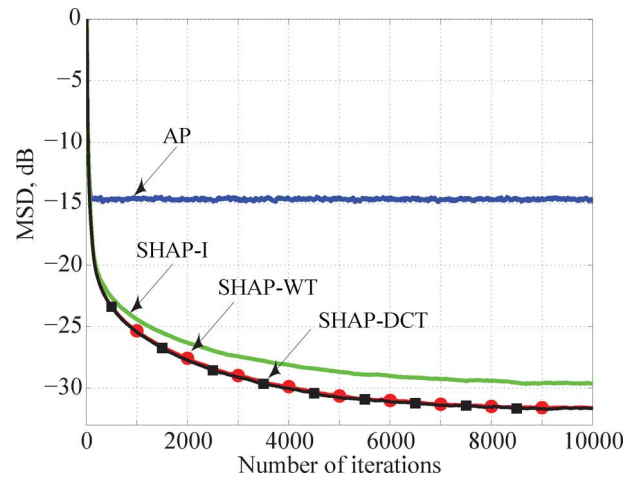


Fig. 5. Learning curves for AP algorithm and SHAP algorithm for three different matrices \mathbf{D} with $L = 8$, $t = \sqrt{L\sigma_v^2}$, and SNR = 20 dB (Experiment 5).

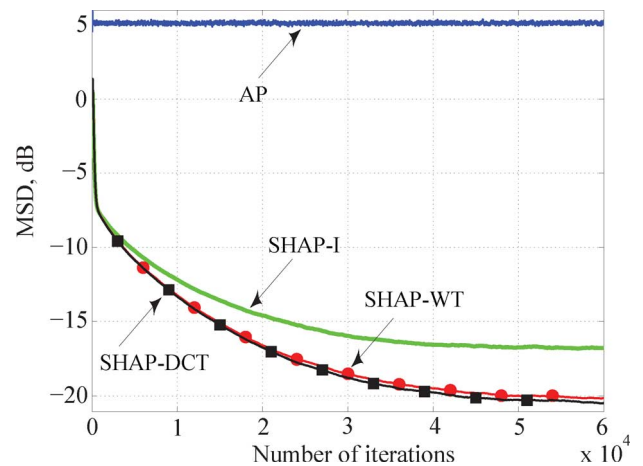


Fig. 6. Learning curves for AP algorithm and SHAP algorithm for three different matrices \mathbf{D} with $L = 8$, $t = \sqrt{10\sigma_v^2}$, and SNR = 0 dB (Experiment 6).

the inequality $|e_k| < \gamma$ in (26) is satisfied and hence the update in (26) does not need to be carried out.

In Experiment 5, we investigated the effect of using different orthonormal matrices \mathbf{D} on the convergence characteristics of the SHAP algorithm. We repeated Experiment 1 with the AP algorithm and the SHAP algorithm using the identity matrix (SHAP-I), DCT matrix (SHAP-DCT) [26], and Daubechies wavelet matrix (SHAP-WT) [27] as \mathbf{D} with $L = 8$, $\lambda = 0.93$, and SNR = 20 dB. The MSD curves obtained are illustrated in Fig. 5. As can be seen, the SHAP-DCT and SHAP-WT algorithms perform better than the AP algorithm and slightly better than the SHAP-I algorithm. The threshold parameter t in this experiment was small and hence similar noise removal would be achieved using different \mathbf{D} . Consequently, different \mathbf{D} would result in similar values of μ_k and similar steady-state misalignment during steady state.

In Experiment 6, we repeated Experiment 5, i.e., Experiment 1 with $L = 8$, except that we used $\sigma_v^2 = 1$ and $\lambda = 0.99$. The MSD curves obtained are illustrated in Fig. 6. As can be seen,

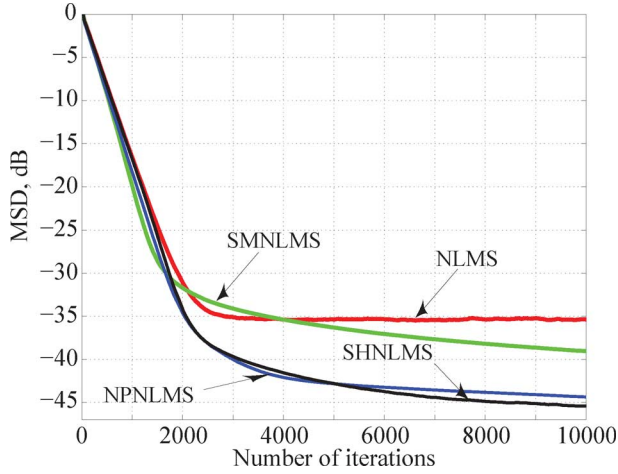


Fig. 7. Learning curves for SHNLMS, NPNLMS, SMNLMS, and NLMS algorithms with $L = 1$, $t = \sqrt{L\sigma_v^2}$, and SNR = 30 dB (Experiment 7).

the SHAP-DCT and SHAP-WT algorithms performed significantly better than the SHAP-I algorithm. This is due to the fact that with an increased σ_v^2 , an increased threshold $t = \sqrt{10\sigma_v^2}$ was obtained and, consequently, different \mathbf{D} resulted in different amounts of noise removal.

In Experiment 7, we examined the performance of the SHAP algorithm with $L = 1$, namely, the SHNLMS algorithm, and that of the NLMS, SMNLMS [6], and NPNLMS [5] algorithms. The order of the unknown system was set to 63. The input signal was obtained by filtering a white Gaussian noise signal with variance 1 by an IIR filter with a pole at 0.9. The variance of the measurement noise was set to 10^{-3} (SNR = 30 dB). The parameters for the NPNLMS algorithm were set to $\lambda = 1 - 1/(6M)$, $\delta = 10^{-8}$, and $\hat{\sigma}_{e,0}^2 = 0$. Parameter λ in (24) was set to 0.99. The MSD curves obtained are illustrated in Fig. 7. As can be seen, the NPNLMS and SHNLMS algorithms offer improved performance compared to the competing algorithms.

Next we carried out two experiments to examine the performance of the SHLMS algorithm.

In Experiment 8, we used $M = 31$ and $f_c = 0.4$ to obtain the unknown system and the input signal was a white Gaussian noise signal with variance 1. The measurement noise was a white Gaussian noise signal with variance 10^{-3} (SNR = 30 dB). The learning curves obtained by using the LMS and the SHLMS algorithms are illustrated in Fig. 8. As can be seen, the proposed SHLMS algorithm yields a reduced misalignment compared to the LMS algorithm when the speed of convergence of the two algorithms is constant. On the other hand, the SHLMS algorithm yields a faster convergence compared to the LMS algorithm when the steady-state misalignments of the two algorithms are approximately the same.

In Experiment 9, we used $M = 16$ and $f_c = 0.4$ to obtain the unknown system. We used the same input signal as in Experiment 8 except that it was correlated by using an IIR filter with a single pole at 0.9. The measurement noise was a white Gaussian noise signal with variance 10^{-1} (SNR = 10 dB). The MSD curves obtained by using the LMS, SHLMS, and the variable step size LMS (VLMS) algorithm reported in [3] are illustrated in Fig. 9. The parameters for the VLMS algorithm were set to

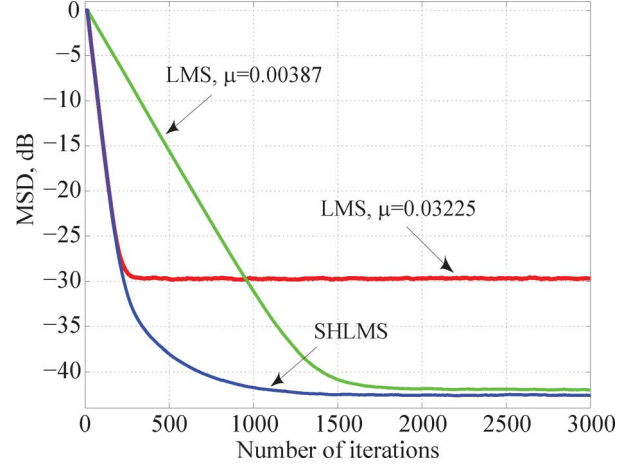


Fig. 8. Learning curves for SHLMS and LMS algorithms with $\lambda = 0.95$, $q = 1$, $t = \sqrt{2\sigma_v^2}$, and SNR = 30 dB (Experiment 8).

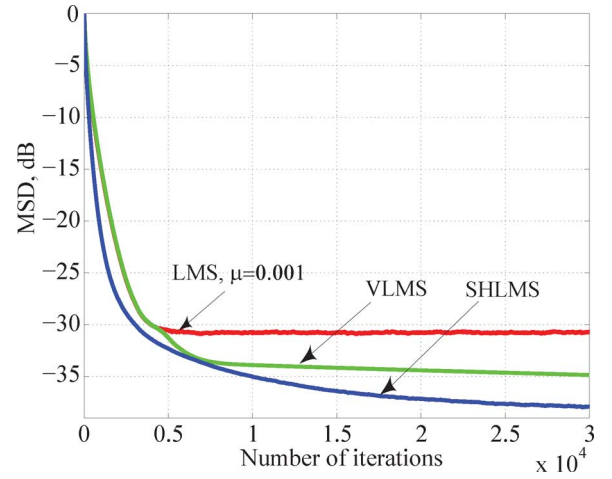


Fig. 9. Learning curves for SHLMS, VLMS, and LMS algorithms with $\lambda = 0.999$, $t = \sqrt{2\sigma_v^2}$, and SNR = 10 dB (Experiment 9).

$\mu_{\min} = 0.00001$, $\mu_{\max} = 0.001$, $\alpha = 0.999$, $\beta = 0.992$, and $\gamma = 0.001$ and $q/E[\|\mathbf{x}_k\|^2]$ for the SHLMS algorithm was set to $1/300$. As can be seen, the proposed SHLMS algorithm yields a reduced steady-state misalignment and, in addition, it converges slightly faster than the competing algorithms.

B. Acoustic Echo-Cancellation Application

In Experiment 10, the performance of the various algorithms in an acoustic echo-cancellation application was investigated. The length of the acoustic channel was set to 1024. The impulse response of the acoustic channel is illustrated in Fig. 10. The input to the loudspeaker was assumed to be amplified by a factor of 10^3 . The signal was contaminated by a white Gaussian noise signal with variance 10^{-2} . The input signal in the 36th trial is illustrated in Fig. 11. The variance of the measurement noise added to the desired signal was set 10^{-2} (SNR = 60 dB). The projection order of the AP algorithms was set to 10. The MSD curves obtained by using the LMS, SHLMS, SHNLMS, NLMS, AP, VSSAP, and SHAP algorithms are illustrated in Fig. 12. As can be seen, the SHAP algorithm offers a reduced steady-state misalignment for the same convergence speed as

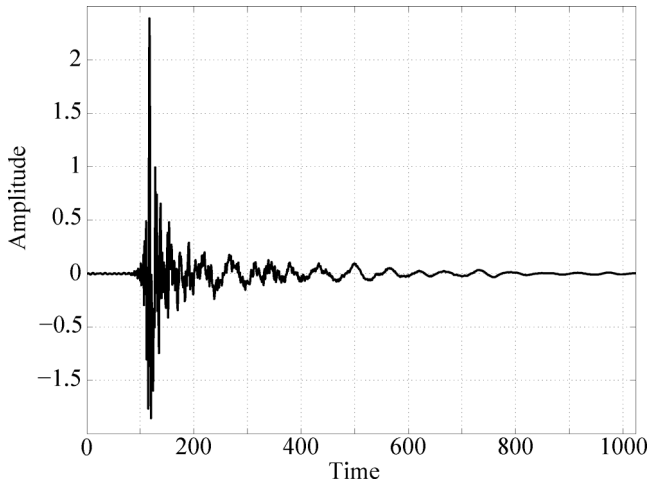


Fig. 10. Impulse response of the acoustic echo path (Experiment 10).

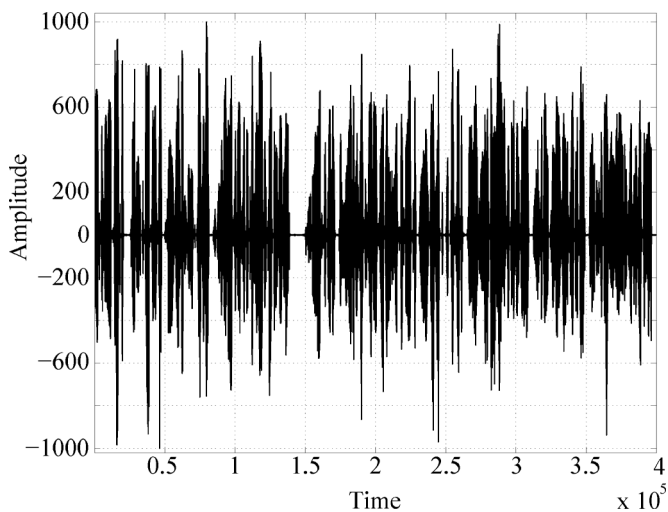


Fig. 11. Input signal for the 36th trial for the acoustic echo-cancellation application (Experiment 10).

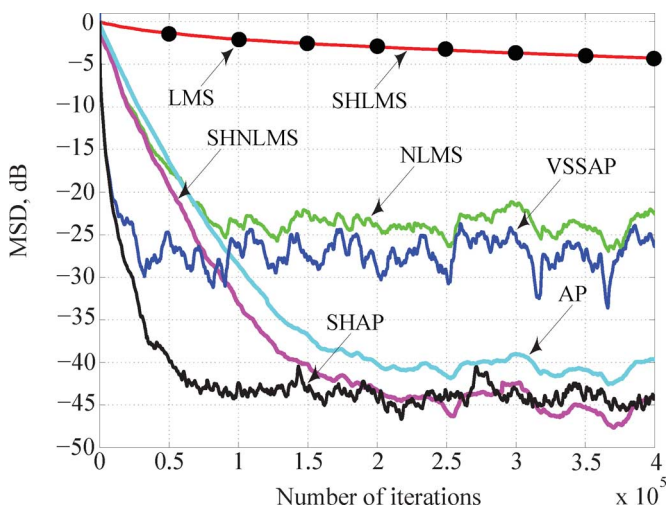


Fig. 12. Learning curves for SHAP, VSSAP, AP, LMS, SHLMS, SHNLMS, and NLMS algorithms with $t = \sqrt{10\sigma_v^2}$ and $\lambda = 0.9$ for the SHAP algorithm (SNR = 60 dB); $t = \sqrt{2\sigma_v^2}$ and $\lambda = 0.99$ for the SHLMS and SHNLMS algorithms (Experiment 10).

compared to the VSSAP algorithm, and a faster convergence as well as a somewhat improved steady-state misalignment as compared to the AP algorithm. The SHNLMS algorithm offers a reduced steady-state misalignment and converges slightly faster than the NLMS algorithm. The SHNLMS algorithm also performs slightly better than the AP algorithm due to the large projection order used in the AP algorithm. The SHLMS algorithm, on the other hand, converges slowly due to the high power of the input signal and large M .

VII. CONCLUSION

A family of adaptive-filtering algorithms, namely, the shrinkage AP, shrinkage NLMS, and shrinkage LMS algorithms have been proposed based on a known L_1 - L_2 minimization scheme described in [17], [18]. Simulation results obtained in system-identification and echo-cancellation applications have shown that the SHAP algorithm performs much better than the conventional AP, SMAP, and VSSAP algorithms in terms of steady-state misalignment and convergence speed. However, the improved performance comes about at the cost of an increased average computational effort per iteration of the order of 11 to 14%. The SHLMS algorithm, on the other hand, offers faster convergence and a reduced steady-state misalignment as compared to the LMS and the VLMS algorithms. Similarly, the SHNLMS algorithm offers a reduced steady-state misalignment as compared to the NLMS and SMNLMS algorithms.

In view of their advantages, the proposed shrinkage adaptive-filtering algorithms can provide improved solutions in many recent adaptive-filtering applications, for example, adaptive mobile networks, source localization, and environment monitoring.

APPENDIX

The computation of $E[(\mathbf{x}_k^T \mathbf{x}_k)^2] = \text{tr}\{E[\mathbf{x}_k \mathbf{x}_k^T \mathbf{x}_k \mathbf{x}_k^T]\}$ in (22) requires information about the fourth-order moment of the input signal \mathbf{x}_k . For a white Gaussian input signal, the fourth-order moment can be obtained by using its second-order moment as

$$E[\mathbf{x}_k \mathbf{x}_k^T \mathbf{x}_k \mathbf{x}_k^T] = 2E[\mathbf{x}_k \mathbf{x}_k^T] E[\mathbf{x}_k \mathbf{x}_k^T] + E[\mathbf{x}_k \mathbf{x}_k^T] \text{tr}\{E[\mathbf{x}_k \mathbf{x}_k^T]\} \quad (27)$$

(see ([1], p. 87) and ([28], p. 991) for details). Now taking the trace on both sides of (27) and using the properties $\text{tr}\{A+B\} = \text{tr}\{A\} + \text{tr}\{B\}$ and $\text{tr}\{c \cdot A\} = c \text{tr}\{A\}$ where c is a scalar, we obtain

$$E[(\mathbf{x}_k^T \mathbf{x}_k)^2] = 2 \text{tr}\{E[\mathbf{x}_k \mathbf{x}_k^T] E[\mathbf{x}_k \mathbf{x}_k^T]\} + \text{tr}\{E[\mathbf{x}_k \mathbf{x}_k^T]\} \text{tr}\{E[\mathbf{x}_k \mathbf{x}_k^T]\}. \quad (28)$$

When the dimension, M , of signal \mathbf{x}_k becomes large, the first term on the right-hand side of (28) becomes insignificant relative to the second term. For example, for a white Gaussian signal we obtain $E[\mathbf{x}_k \mathbf{x}_k^T] = \sigma_x^2 \mathbf{I}$ for which the first term on the right-hand side of (28) would be of the order of M while the

second term would be of the order of M^2 . Therefore, for a large M , the first term would be negligible and this would also be the case for a correlated input signal [28]. In effect, for a large M

$$E \left[(\mathbf{x}_k^T \mathbf{x}_k)^2 \right] \approx (\text{tr} \{ E [\mathbf{x}_k \mathbf{x}_k^T] \})^2 = E [\mathbf{x}_k^T \mathbf{x}_k]^2.$$

REFERENCES

- [1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. New York, NY, USA: Springer, 2008.
- [2] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ, USA: Wiley, 2003.
- [3] T. Aboulnasr and K. Mayyas, "A robust variable step size LMS-type algorithm: Analysis and simulations," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 631–639, Mar. 1997.
- [4] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. Signal Process.*, vol. 41, no. 9, pp. 2811–2825, Sep. 1993.
- [5] J. Benesty, H. Rey, L. R. Vega, and S. Tressens, "A nonparametric VSS NLMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 581–584, Oct. 2006.
- [6] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Process. Lett.*, vol. 5, no. 5, pp. 111–114, May 1998.
- [7] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Jpn. A.*, vol. 67, pp. 126–132, 1984.
- [8] S. Werner and P. S. R. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Process. Lett.*, vol. 8, no. 8, pp. 231–235, 2001.
- [9] H. C. Shin, A. H. Sayed, and W. J. Song, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132–135, Feb. 2004.
- [10] H. Rey, L. R. Vega, S. Tressens, and J. Benesty, "Variable explicit regularization in affine projection algorithm: Robustness issues and optimal choice," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2096–2109, May 2007.
- [11] M. Chang, N. Kong, and P. Park, "An affine projection algorithm based on reuse time of input vectors," *IEEE Signal Process. Lett.*, vol. 7, no. 8, pp. 750–753, Aug. 2010.
- [12] S. Werner, P. S. R. Diniz, and J. E. W. Moreira, "Set-membership affine projection algorithm with variable data-reuse factor," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 261–264.
- [13] S. Werner, J. A. Apolinario, Jr., and P. S. R. Diniz, "Set-membership proportionate affine projection algorithms," *EURASIP J. Audio, Speech, Music Process.*, vol. 2007, p. 10, 2007, Article ID 34242.
- [14] J. A. Apolinario, Jr., M. L. R. de Campos, and P. S. R. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," *IEEE Trans. Signal Process.*, vol. 48, pp. 3235–3242, Nov. 2000.
- [15] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Trans. Signal Process.*, vol. 51, pp. 124–134, Jan. 2003.
- [16] H. C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 90–102, Jan. 2004.
- [17] I. Daubechies, M. Defrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.
- [18] M. Zibulevsky and M. Elad, " l_1 - l_2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, pp. 76–88, May 2010.
- [19] C. Paleologu, J. Benesty, and S. Ciochina, "A variable step-size affine projection algorithm designed for acoustic echo cancellation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 8, pp. 1466–1478, Nov. 2008.
- [20] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 90, no. 8, pp. 1504–1510, Aug. 2007.
- [21] 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.
- [22] 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.
- [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 and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
- [25] A. Antoniou and W. S. Lu, *Practical Optimization*. New York, NY, USA: Springer, 2007.
- [26] S. A. Khayam, "The Discrete Cosine Transform (DCT): Theory and Application," Michigan State Univ., East Lansing, MI, USA, Tech. Rep., Mar. 2003.
- [27] I. Daubechies, "Orthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 41, pp. 909–996, Nov. 1988.
- [28] J. E. Mazo, "On the independence theory of equalizer convergence," *Bell Syst. Tech. J.*, vol. 58, no. 5, pp. 963–993, May 1979.



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

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

Dr. Ali Bhotto was awarded a University of Victoria fellowship for 2007–2008.



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, London, U.K., in 1963 and 1966, respectively.

He taught at Concordia University from 1970 to 1983. From 1983 to 1990, he was the founding Chair of the Department of Electrical and Computer Engineering, University of Victoria, BC, Canada, where he is currently 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 coauthor

(with W.-S. Lu) of *Practical Optimization: Algorithms and Engineering Applications* (Springer, 2007).

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