Comparison of DC Offset Effects in Four LMS Adaptive Algorithms

Ayal Shoval, *Student Member, IEEE,* David A. Johns, *Member, IEEE,* and W. Martin Snelgrove, *Member, IEEE*

***Abstract*— It is well known that dc offsets degrade the performance of analog adaptive filters. In this paper, the effects of dc offsets on four variations of the stochastic gradient algorithm are analyzed. Assuming a Gaussian probability distribution for the input signal and error signal, the output mean squared error (MSE) performance in the presence of dc offsets is evaluated for each of the algorithms. The theoretical work is compared with computer simulations and the results, together with convergence properties of each of the algorithms and their respective hardware requirements, are used in selecting the most appropriate algorithm. Although a Gaussian input distribution is assumed, it may reasonably be inferred that the critical results obtained should also hold for other input distributions.**

I. Introduction

THE ESSENCE of an adaptive filter is the implementation of the algorithm that controls the coefficients of the programmable filter. Among the many possible algorithms, the least-mean-square (LMS) algorithm has been widely used due to its implementation simplicity. For even greater implementation simplicity, the sign-data, the sign-error and the sign-sign LMS (SD-LMS, SE-LMS and SS-LMS, respectively) algorithms have been proposed and investigated extensively in the technical literature [1]–[7]. The findings of these works show that all variants of the LMS algorithm converge only if the input signal is sufficiently exciting [4] and that even when sufficiency conditions are met, the SS-LMS and the SD-LMS algorithms can diverge due to misalignment of the gradient signals [4]–[6]. That is, unlike the LMS or the SE-LMS algorithms which force the coefficient updates vector to move along a line in the coefficient space parallel to its gradient signal vector, the SD-LMS and the SS-LMS algorithms force the coefficient updates vector to move along a line in the coefficient space misaligned from its gradient signal vector and parallel to the *sign* of its gradient signal vector. Consequently, whereas in the former case the coefficient updates will, on average, move in a direction of "steepest descent" of the squared error surface, in the SD-LMS and the SS-LMS algorithm case the misalignment can lead to coefficient divergence and may also cause the coefficient updates to "climb" the error surface. In addition, it has been shown that while both the LMS and the SD-LMS algorithms will ideally force the filter coefficients to their optimal locations as the error signal is reduced to zero (i.e., zero MSE), the SE-LMS and the SS-LMS algorithms will experience finite minimum MSE. This finite MSE results from the fact that slicing the error signal prevents the *effective* error signal from reducing to zero. In fact, it has been claimed [8] that as the coefficients reach their optimal values and the error signal is reduced, the *effective* error signal increases potentially causing the coefficients to jerk. Thus, it is tempting to use the LMS algorithm and dispense with the SS-LMS algorithm. However when considering algorithm implementation, the LMS algorithm is the most complex while the SS-LMS algorithm is the simplest. The SD-LMS algorithm, being simpler than the LMS algorithm, requires N slicers and N trivial multipliers which is more complex than the SE-LMS algorithm requiring 1 slicer and N trivial multipliers where N is the number of coefficients begin adapted. Thus the choice of which algorithm to use is difficult.

When implementing *analog* adaptive filters, not only are algorithm architecture complexity and algorithm convergence important issues, but also dc offsets. Although some publications have treated dc offsets in adaptive filters [2], [9]–[12], few results are available on the effects of all sources of dc offsets on all four variations of the LMS algorithm. Since algorithm misalignment and algorithm convergence rate are covered extensively in the technical literature [3]–[7], this paper focuses on the performance of the four variants of the LMS algorithm from a dc offset point of view. The results presented here should assist the designer is overcoming the perplexing issue of selecting the appropriate hardware implementation for the coefficient update algorithm.

To keep the analysis simple and tractable, discrete-time systems are used and, as a working example, an adaptive linear combiner whose input is zero-mean Gaussian noise will be assumed. Although this input forms a special case, intuitive comments will be given for arbitrary input statistics. The accuracy of the discrete-time system in analyzing the effects of dc offsets in a continuous-time linear combiner might be questionable, however the relations obtained here are based on taking the mean and variance of the product of filter gradient and error signals. Since a continuous-time linear combiner can be well approximated by a discrete-time system running at a very high oversampling rate, the relations for such a system would not depart severely from those discussed herein as the sampling rate is increased. In addition, we would like to point out that often continuous-time

Manuscript received July 7, 1993; revised March 10, 1994. This work was supported by Micronet. This paper was recommended by Associate Editor L. R. Carley.

A. Shoval and D. A. Johns are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario M5S 1A4 Canada.

W. M. Snelgrove is with the Department of Electronics, Carleton University, Ottawa, Ontario K1S 5B6 Canada.

IEEE Log Number 9407650.

techniques are used in implementing the signal path (i.e., the filter) while sampled-time techniques are used in realizing the adaptation algorithm. Thus the analysis performed here, which focuses on dc offsets in the algorithm circuitry, is sufficient to provide detail as to the comparative performance of each of the candidate algorithms. Finally, while some approximations are made in deriving analytical expressions, simulation results are presented showing close agreement which is sufficient since typically only rough estimates of dc offset values are known.

Section II briefly reviews the underlying expressions that characterize an adaptive linear combiner and identifies the location of the offsets for each of the four algorithms: LMS, SD-LMS, SE-LMS and SS-LMS. In Sections III–VI the effect of the above offsets on the performance of each of the algorithms assuming a Gaussian input, is analyzed. In Section VII simulation results are given and compared with the analytical predictions.

II. Problem Formulation

For an adaptive linear combiner, as shown in Fig. 1, the output at time index k is given by

$$y(k) = \sum_{i=1}^{N} w_i(k)x_i(k) \tag{1}$$

where $w_i(k)$ is the i^{th} coefficient value and $x_i(k)$ is the i^{th} gradient signal as well as the i^{th} input signal. In vector notation, (1) can be represented as

$$\mathbf{y}_k = \mathbf{x}_k^T \mathbf{w}_k. \tag{2}$$

The error signal is

$$\begin{aligned} e(k) &= \delta(k) - y(k) \\ &= \mathbf{x}_k^T[\mathbf{w}^* - \mathbf{w}_k] \end{aligned} \tag{3}$$

where $\delta(k)$ is the desired response and $\mathbf{w}^*$ is a vector of optimal coefficients. Defining $\mathbf{c}_k$ to be the present coefficient estimate, or mathematically

$$\mathbf{c}_k = \mathbf{w}^* - \mathbf{w}_k \tag{4}$$

then (3) can be re-written as

$$e(k) = \mathbf{x}_k^T \mathbf{c}_k. \tag{5}$$

Assuming the input is zero-mean, we have

$$E[\mathbf{x}_k] = \mathbf{0} \tag{6}$$

where $E[\bullet]$ represents the expectation operator. To allow a solution of otherwise very complicated expressions, it is also assumed that the gradient signals and the filter coefficient estimates are statistically independent, thus

$$E[\mathbf{x}_k^T \mathbf{c}_k] = E[\mathbf{x}_k^T]E[\mathbf{c}_k]. \tag{7}$$

This assumption is not uncommon [5] and is acknowledged to be an approximation since coefficient computation depends on the gradient signals. However, for slow adaptation the coefficient estimates are weakly dependent on the gradient signals and the assumption invoked by (7) provides satisfactory steady-state results as will be noted from the simulation results herein. Equations (5)–(7) also yield

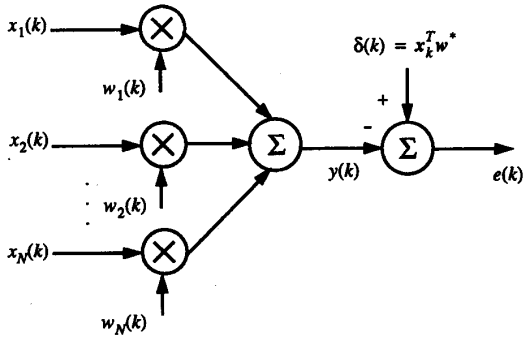


Fig. 1. A general adaptive linear combiner.

$$E[e(k)] = 0. \tag{8}$$

We also define $\sigma_x^2 \equiv E[x_i^2(k)]$ and $\sigma_e^2 \equiv E[e^2(k)]$ to be the mean-squared value of the gradient and the error signals, respectively. The quantity σ_e^2 represents the filter output MSE and is the performance measure to be evaluated for each of the four algorithms.

The LMS algorithm used to update the filter coefficients is given below with modeled dc offsets inserted at appropriate locations

$$\mathbf{w}_{k+1} = \mathbf{w}_k + 2\mu((\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m}) \tag{9}$$

where

$$\mathbf{m}_\mathbf{x} = [m_{x1} m_{x2} \cdots m_{xN}]^T \tag{10}$$

is a vector representing the unwanted dc offsets on each of the gradient signals, m_e represents the unwanted dc offset on the error signal, μ is a small step size the governs the rate of adaptation and $\mathbf{m}$ is a vector representing the unwanted equivalent dc offsets at the input of the accumulator (integrator) and at the output of the multiplier where

$$\mathbf{m} = [m_1\, m_2 \cdots m_N]^T. \tag{11}$$

Fig. 2 depicts the equivalent block diagram representing (9) for the i^{th} coefficient. Upon substituting (4) into (9) one obtains

$$\text{LMS} \quad \mathbf{c}_{k+1} = \mathbf{c}_k - 2\mu((\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m}). \tag{12}$$

The equivalent expression of (12) for the three other variants of the LMS algorithm SD-LMS, SE-LMS, and SS-LMS, respectively, are:

$$\begin{aligned} \text{SD-LMS} \quad \mathbf{c}_{k+1} &= \mathbf{c}_k - 2\mu(\text{sgn}[\mathbf{x}_k + \mathbf{m}_\mathbf{x}] \\ &\quad \times (e(k) + m_e) + \mathbf{m}) \end{aligned} \tag{13}$$

$$\begin{aligned} \text{SE-LMS} \quad \mathbf{c}_{k+1} &= \mathbf{c}_k - 2\mu((\mathbf{x}_k + \mathbf{m}_\mathbf{x}) \\ &\quad \times \text{sgn}[e(k) + m_e] + \mathbf{m}) \end{aligned} \tag{14}$$

$$\begin{aligned} \text{SS-LMS} \quad \mathbf{c}_{k+1} &= \mathbf{c}_k - 2\mu(\text{sgn}[\mathbf{x}_k + \mathbf{m}_\mathbf{x}] \\ &\quad \times \text{sgn}\,[e(k) + m_e] + \mathbf{m}) \end{aligned} \tag{15}$$

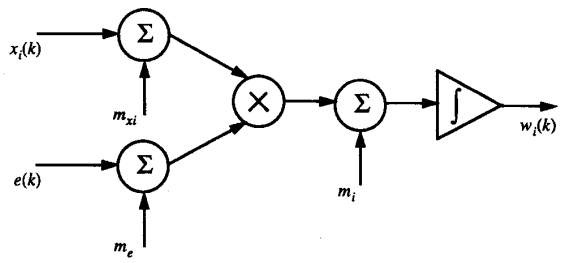


Fig. 2. Details of the LMS update circuitry showing dc offset sources.

III. The LMS Algorithm

Taking the expectation of both sides of (12) we obtain

$$E[\mathbf{c}_{k+1}] = E[\mathbf{c}_k] - 2\mu E[(\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m}]. \quad (16)$$

At steady-state (i.e., as $k \to \infty$), we have $E[\mathbf{c}_{k+1}] = E[\mathbf{c}_k]$. Using this fact together with (6) and (8), (16) simplifies to

$$E[\mathbf{x}_k e(k)] = -(\mathbf{m} + m_e \mathbf{m}_\mathbf{x}). \quad (17)$$

Substituting (5) and (7) into (17) results in

$$E[\mathbf{x}_k \mathbf{x}_k^T]\, E[\mathbf{c}_k] = -(\mathbf{m} + m_e \mathbf{m}_\mathbf{x}). \quad (18)$$

Letting

$$\mathbf{R} \equiv [\mathbf{x}_k \mathbf{x}_k^T] \quad (19)$$

and dropping the time index k (for mathematical convenience), at steady-state the following relations hold:

$$E[\mathbf{c}] = -\mathbf{R}^{-1}(\mathbf{m} + m_e \mathbf{m}_\mathbf{x}) \quad (20)$$

$$E[\mathbf{c}^T] = -(\mathbf{m} + m_e \mathbf{m}_\mathbf{x})^T \mathbf{R}^{-T}. \quad (21)$$

To solve for the residual MSE due to offsets, consider once again the expression in (12). Taking the mean-squared value of both sides yields

$$E[\mathbf{c}_{k+1}^T \mathbf{c}_{k+1}] = E[\mathbf{c}_k^T \mathbf{c}_k] - 4\mu E[c_k^T((\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m})] + 4\mu^2 E[((\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m})^T \times ((\mathbf{x}_k + \mathbf{m}_\mathbf{x})(e(k) + m_e) + \mathbf{m})]. \quad (22)$$

Noting that at steady-state $E[\mathbf{c}_{k+1}^T \mathbf{c}_{k+1}] = E[\mathbf{c}_k^T \mathbf{c}_k]$, substituting (6) and (8) into (22) and dropping the time index as before, yields

$$0 = \mu E[(\mathbf{x}^T\mathbf{x} + 2\mathbf{x}^T\mathbf{m}_\mathbf{x} + \mathbf{m}_\mathbf{x}^T\mathbf{m}_\mathbf{x}) \times (e^2 + 2em_e + m_e^2)] - \mu \mathbf{m}^T\mathbf{m} - E[\mathbf{c}^T](\mathbf{m} + m_e\mathbf{m}_\mathbf{x}) - E[e^2] - E[\mathbf{c}^T e]\mathbf{m}_\mathbf{x}. \quad (23)$$

The solution of (23) for arbitrary μ is tedious and results in a value for the MSE that has a weak dependence on μ. Thus assuming $\mu \to 0$, making use of (21) and noting from (5) and (7) that for slow adaptation the last term in (23) is proportional to $E[\mathbf{x}_k]$ and is therefore negligible, (23) can be solved for the excess MSE at steady-state

$$\sigma_e^2 \approx (\mathbf{m} + m_e\mathbf{m}_\mathbf{x})^T \mathbf{R}^{-T}(\mathbf{m} + m_e\mathbf{m}_\mathbf{x}). \quad (24)$$

The result in (24) shows that the excess MSE is inversely proportional to the power of the input signal through the $\mathbf{R}^{-T}$term; lower input signal powers, for fixed offset levels, produce higher excess MSE. The excess MSE is also directly sensitive to all offset sources. In analog implementations the dc offset at the output of the multiplier and offsets at the input to the integrator, $\mathbf{m}$, would typically dominate (relative to m_e or $\mathbf{m}_\mathbf{x}$). Clearly, to minimize the excess MSE nulling of m_e or $\mathbf{m}_\mathbf{x}$ and $\mathbf{m}$ would be required and may be plausible in certain applications using ac-coupling and offset cancelled integrators, respectively. On observing (24), it is also interesting to note that it is possible to minimize the excess MSE by adjusting the dc offsets to cancel one another rather than nulling m_e or $\mathbf{m}_\mathbf{x}$ and $\mathbf{m}$. However, satisfying this equality implies adaptively tracking a vector of integrator input offset and multiplier output offset, $\mathbf{m}$, to a vector of gradient signal bias, $\mathbf{m}_\mathbf{x}$, scaled by $-m_e$. This approach is not a trivial one when considering hardware implementation. Finally, notice that the excess MSE due to this offset cannot be compensated by reducing μ.

IV. The Sign-Data LMS Algorithm

Taking the expectation of both sides of (13), using (5)–(8) and simplifying as before yields

$$-\mathbf{m} = E[\mathrm{sgn}[\mathbf{x} + \mathbf{m}_\mathbf{x}]\mathbf{x}^T]E[\mathbf{c}] + E[\mathrm{sgn}[\mathbf{x} + \mathbf{m}_\mathbf{x}]]m_e. \quad (25)$$

For a zero-mean Gaussian noise input with variance $\sigma_x^2 = R_{ii}$, it can be shown that (see Appendix A):

$$E[\mathrm{sgn}[\mathbf{x} + \mathbf{m}_\mathbf{x}]] = \left[\mathrm{erf}\left[\frac{m_{x1}}{\sqrt{2\sigma_x^2}}\right] \mathrm{erf}\left[\frac{m_{x2}}{\sqrt{2\sigma_x^2}}\right] \cdots \mathrm{erf}\left[\frac{m_{xN}}{\sqrt{2\sigma_x^2}}\right]\right]^T \equiv \mathbf{k}_\mathbf{mx}. \quad (26)$$

Using Price's Theorem [14] it can be shown that (see Appendix B):

$$E[\mathrm{sgn}[\mathbf{x} + \mathbf{m}_\mathbf{x}]\mathbf{x}^T] = \frac{1}{\sigma_x}\sqrt{\frac{2}{\pi}} \begin{bmatrix} e^{-m_{x1}^2/2\sigma_x^2}[R_{11}\, R_{12} \cdots R_{1N}] \\ e^{-m_{x2}^2/2\sigma_x^2}[R_{21}\, R_{22} \cdots R_{2N}] \\ e^{-m_{xN}^2/2\sigma_x^2}[R_{N1}\, R_{N2} \cdots R_{NN}] \end{bmatrix} \equiv \mathbf{R}_\mathbf{MX} \quad (27)$$

where $R_{ij} = E[x_i x_j]$ as before. Substituting (26) and (27) into (25) yields

$$E[\mathbf{c}^T] = -(\mathbf{m} + m_e\mathbf{k}_\mathbf{mx})^T \mathbf{R}_\mathbf{MX}^{-T}. \quad (28)$$

Taking the mean-squared value of both sides of (13) and simplifying as done previously one obtains

$$0 = \mu(N\sigma_e^2 + Nm_e^2 - \mathbf{m}^T\mathbf{m}) - E[\mathbf{c}^T] \times (\mathbf{m} + m_e\mathbf{k}_\mathbf{mx}) - E[\mathbf{c}^T \mathrm{sgn}[\mathbf{x} + \mathbf{m}_\mathbf{x}]e]. \quad (29)$$

Using (7) and (27) the last term in (29) simplifies to

$$E[\mathbf{c}^T \text{sgn}[\mathbf{x}+\mathbf{m_x}]e] = E[\mathbf{c}^T \mathbf{R_{MX}} \mathbf{c}]. \tag{30}$$

An analytical expression for the excess MSE requires the evaluation of (30). Consider the case for a Gaussian white noise; (30) reduces to

$$E[\mathbf{c}^T \text{sgn}[\mathbf{x}+\mathbf{m_x}]e] = \sqrt{\frac{2}{\pi}}\sigma_x \sum_{i=1}^{N} \hat{\sigma}_{c_i}^2 e^{-m_{xi}^2/2\sigma_x^2} \tag{31}$$

where $\hat{\sigma}_{c_i}^2 = \sigma_{c_i}^2 + E[c_i]^2$ and $\sigma_{c_i}^2$ represents the variance of c_i. Making use of the assumption in (7), one can derive

$$\sigma_e^2 = \sigma_x^2 \sum_{i=1}^{N} \hat{\sigma}_{c_i}^2. \tag{32}$$

Assuming the mean-squared value of all the coefficient estimates equal the same value, or mathematically, $\hat{\sigma}_{c_i}^2 \approx \hat{\sigma}_{c_j}^2 \equiv \hat{\sigma}_c^2$, the following expression, making use of (28)–(32) is obtained for the excess MSE as a function of the interfering offsets

$$\sigma_e^2 \approx \frac{\mu(Nm_e^2 - \mathbf{m}^T\mathbf{m}) + (\mathbf{m}+m_e\mathbf{k_{mx}})^T \mathbf{R_{MX}^{-T}} (\mathbf{m}+m_e\mathbf{k_{mx}})}{\frac{1}{N\sigma_x}\sqrt{\frac{2}{\pi}}\sum_{i=1}^{N} e^{-m_{xi}^2/2\sigma_x^2} - \mu N}. \tag{33}$$

The expression in (33) assumes the case where the input signals $x_i(k)$ are Gaussian white, however it is not clear if the same expression can be used for nonwhite inputs. Fortunately, (33) does give reasonable estimates for general inputs in the practical case where the square of the offsets on the gradient signals, m_{xi}^2 are sufficiently small compared to the variance of the gradient signals, σ_x^2. In this case we can then approximate the exponential terms in (27) by unity and (30) can be reduced to

$$E[\mathbf{c}^T \text{sgn}[\mathbf{x}+\mathbf{m_x}]e] = \frac{1}{\sigma_x}\sqrt{\frac{2}{\pi}}\sigma_e^2. \tag{34}$$

Upon substituting (28) and (34) into (29) an expression for the excess MSE is obtained that is given by (33) with m_{xi} in the denominator set to zero.

The expression in (33) shows that the performance of the SD-LMS algorithm is similar to the LMS algorithm from a dc offset point of view; the dominant offset terms appear explicitly in the numerator of (33). The difference here is that the excess MSE is a weak function of the input signal power[1] for small μ. This effect is a consequence of the slicing operation which results in the loss of information regarding the amplitude of the signal and would be similarly manifested for arbitrary input distributions.

[1] Signal power, σ_x^2, appears both in the numerator (via $\mathbf{R_{MX}^{-T}}$) and denominator of (33).

V. The Sign-Error LMS Algorithm

Taking the expectation of both sides of (14) and simplifying as before yields

$$-\mathbf{m} = E[\mathbf{x}\ \text{sgn}[e+m_e]] + E[\text{sgn}[e+m_e]]\mathbf{m_x}. \tag{35}$$

Assuming $e(k)$ has a Gaussian distribution at steady-state,[2] using the results in Appendexes A and B as well as (7), it can be shown from (35) that

$$E[\mathbf{c}^T] = -\sigma_e\sqrt{\frac{\pi}{2}}e^{m_e^2/2\sigma_e^2}\left(\mathbf{m} + \text{erf}\left[\frac{m_e}{\sqrt{2\sigma_e^2}}\right]\mathbf{m_x}\right)^T \mathbf{R}^{-T}. \tag{36}$$

Taking the mean-squared value of both-sides of (14), simplifying as before and collecting terms the following expression results

$$0 = \mu(N\sigma_x^2 + \mathbf{m_x^T}\mathbf{m_x} - \mathbf{m}^T\mathbf{m}) - \sigma_e\sqrt{\frac{2}{\pi}}e^{-m_e^2/2\sigma_e^2} - E[\mathbf{c}^T]\mathbf{m} - E[\mathbf{c}^T\text{sgn}[e+m_e]]\mathbf{m_x}. \tag{37}$$

Defining $\tilde{\mathbf{c}}^T$ to be a vector representing the ac component of the filter coefficient estimates, or mathematically, $\tilde{\mathbf{c}}^T \equiv \mathbf{c}^T - E[\mathbf{c}^T]$, and substituting into the last term in (37) yields

$$0 = \mu(N\sigma_x^2 + \mathbf{m_x^T}\mathbf{m_x} - \mathbf{m}^T\mathbf{m}) - \sigma_e\sqrt{\frac{2}{\pi}}e^{-m_e^2/2\sigma_e^2} - E[\mathbf{c}^T]\left(\mathbf{m} + \text{erf}\left[\frac{m_e}{\sqrt{2\sigma_e^2}}\right]\mathbf{m_x}\right) - E[\tilde{\mathbf{c}}^T\ \text{sgn}[e+m_e]]\mathbf{m_x}. \tag{38}$$

The last term in (38) measures the correlation of $\tilde{\mathbf{c}}^T$ with $\text{sgn}[e+m_e]$ and is approximated to zero since for slow adaptation the ac component of the filter coefficient estimates, $\tilde{\mathbf{c}}^T$, is small. Thus, (38) together with (36) provide a nonlinear function in σ_e^2 that describes the MSE as function of μ and the interfering offsets.

While (38) is the main result for this section, it is also of interest to solve (36) for two limiting cases. To find the limiting value of the MSE for the case of small μ, set $\mu = 0$ and solve (38) to obtain (39) shown at the bottom of the page. For the case of nonzero μ and $m_e = 0$, it can be shown from (36) and (38) that the excess MSE is

$$\sigma_e^2\,|_{m_e=0} = \frac{\pi}{2}\mu^2\left(\frac{N\sigma_x^2 + \mathbf{m_x^T}\mathbf{m_x} - \mathbf{m}^T\mathbf{m}}{1 - \frac{\pi}{2}\mathbf{m}^T\mathbf{R}^{-T}m}\right)^2. \tag{40}$$

Comparing (39) with the excess MSE for the LMS algorithm (24), observe that minimizing (39) implies the minimization of m_e or $\mathbf{m}$ while the minimization of (24) implies the minimization of the dominant offset term $\mathbf{m}$. Thus in analog

[2] This assumption becomes better for small μ for which the ac component of the coefficients is small and thus the distribution of the error signal follows that of the input.

$$\lim_{\mu\to 0}\sigma_e^2 = \frac{-m_e^2}{\ln\left[\frac{\pi}{2}\left(\mathbf{m} + \text{erf}\left[\frac{m_e}{\sqrt{2\sigma_e^2}}\right]\mathbf{m_x}\right)\mathbf{R}^{-T}\left(\mathbf{m} + \text{erf}\left[\frac{m_e}{\sqrt{2\sigma_e^2}}\right]\mathbf{m_x}\right)\right]}. \tag{39}$$

implementations where the offsets represented by $\mathbf{m}$ typically dominate (relative to m_e or $\mathbf{m_x}$), much better MSE performance in the presence of dc offsets can be achieved using the SE-LMS algorithm. This result can be seen from another perspective by taking the limiting values of ± 1 for the erf $[\bullet]$ terms in (39) and keeping only the dominant offset terms in (24) to obtain the following ratio (see (41) shown at the bottom of the page). The ratio typically exceeds unity for practical offset levels including the case where the offset terms represented by m_e and $\mathbf{m}$ are of the same size, owing to the natural logarithm operator. Similar reasoning can be applied to the SD-LMS algorithm. The minimization of the offset term m_e is not difficult as it entails the minimization of the input offset of a comparator.[3] This can be achieved by using a clocked comparator or the technique in [15]. Compensation of the dominant offset term, $\mathbf{m}$, is feasible in integrated form but its practical limiting value would be higher than that obtained by compensating m_e.

In the limiting case of $m_e = 0$, notice that in (40) the MSE is shaped by μ and therefore achieves better MSE performance for small μ than (24) or (33). However observe from (40) that in the absence of dc offsets[4] the SE-LMS algorithm, unlike the LMS or the SD-LMS algorithm, will sustain a finite excess MSE that depends on μ. This is a consequence of slicing the error signal which prevents the effective error signal from going to zero at steady-state. As well, notice that offset cancellation between offsets can also improve the excess MSE as mentioned for the LMS algorithm.

It is also of interest to note that the degrading effects of dc offsets can be alleviated by passing the error signal through a high gain stage prior to coefficient computation [2]. As a result, the MSE can be shown to be reduced by a factor proportional to the gain factor. This solution is intuitively simple but becomes more difficult to achieve in high-frequency applications. It is instructive to point out that the SE-LMS algorithm inherently provides this high gain which, although nonlinear, is frequency independent.

Finally, note that unlike the LMS and the SD-LMS algorithms, the effective error signal in (14) cannot exceed unity in magnitude. Thus, if on average $|m_i| > |x_i + m_{xi}|$, then c_i will diverge. Intuitively this means that if the signal component, x_i, is small relative to the offset component, $m_i - m_{xi}$, then the parenthesized term in (14) will be dominated by the offset component, resulting in the respective coefficient to saturate at its limiting value.

VI. The Sign-Sign LMS Algorithm

Although the circuit implementation of the SS-LMS algorithm is quite simple, the analysis of its performance from an offset point of view is the most complex of the algorithms discussed so far. Thus various approximations will be used to obtain results which depict the behavior of the excess MSE as a function of the interfering offsets. Simulation results will show that the analytical results obtained by using the approximations satisfactorily predict the behavior of the excess MSE.

Assuming $e(k)$ is Gaussian, taking the expectation of both sides of (15) one obtains

$$-\mathbf{m} = E[\operatorname{sgn}[\mathbf{x} + \mathbf{m_x}]\operatorname{sgn}[e + m_e]]. \tag{42}$$

Making use of the work in [16] and the results of the previous sections, (42) can be approximated to give[5]

$$E[\mathbf{c}^T] \approx -\sqrt{\frac{\pi}{2}}\sigma_e e^{m_e^2/2\sigma_e^2}\mathbf{m}^T\mathbf{R}_{MX}^{-T}. \tag{43}$$

Taking the mean-squared value of both sides of (15) one obtains

$$0 = \mu(N - \mathbf{m}^T\mathbf{m}) - E[\mathbf{c}^T \operatorname{sgn}[\mathbf{x} + \mathbf{m_x}] \operatorname{sgn}[e + m_e]] - E[\mathbf{c}^T]\mathbf{m}. \tag{44}$$

Using [16], the procedure in obtaining (31)–(33), (43), and substituting (43) into (44) yields

$$0 \approx \mu(N - \mathbf{m}^T\mathbf{m}) - \frac{2}{\pi}\frac{\sigma_e}{\sigma_x}e^{-m_e^2/2\sigma_e^2}\frac{1}{N}\sum_{i=1}^{N} e^{-m_{xi}^2/2\sigma_x^2} + \sqrt{\frac{\pi}{2}}\sigma_e e^{m_e^2/2\sigma_e^2}\mathbf{m}^T\mathbf{R}_{MX}^{-T}\mathbf{m}. \tag{45}$$

Again, a nonlinear function in σ_e^2 describes the MSE as function of μ and the interfering offsets.

As in the SE-LMS case, (45) can be solved for the limiting case of a small μ to give the excess MSE for the SS-LMS algorithm as

$$\lim_{\mu \to 0} \sigma_e^2 \approx \frac{-m_e^2}{\ln\left[\left(\frac{\pi}{2}\right)^{3/2}\sigma_x \frac{\mathbf{m}^T\mathbf{R}_{MX}^{-T}\mathbf{m}}{\frac{1}{N}\sum_{i=1}^{N} e^{-m_{xi}^2/2\sigma_x^2}}\right]}. \tag{46}$$

With $m_e = 0$, (45) can be solved to give the excess MSE as

$$\sigma_e^2\big|_{m_e=0} \approx \left(\frac{\pi}{2}\mu\right)^2\sigma_x^2 \times \left(\frac{N - \mathbf{m}^T\mathbf{m}}{\frac{1}{N}\sum_{i=1}^{N} e^{-m_{xi}^2/2\sigma_x^2} - \sigma_x\left(\frac{\pi}{2}\right)^{3/2}\mathbf{m}^T\mathbf{R}_{MX}^{-T}\mathbf{m}}\right)^2. \tag{47}$$

The results show that the SS-LMS algorithm in the presence of dc offsets has much better excess MSE performance than the LMS algorithm or the SD-LMS algorithm for the same reasons as the SE-LMS algorithm. Notice in (46), as noted in

[3] AC-coupling $e(t)$ to eliminate signal offset can be feasibly done offset-free using passive IC components for most high-speed applications.

[4] Not the case for analog circuits.

[5] Although we cannot rigorously derive the result of (43), we believe the approximation models the actual result. The validity thereof, can be noted from the previous results and the simulations. The derivation is based on the assumption of Gaussian signalling and repeated use of Price's Theorem.

$$\frac{\sigma_{e,\mathrm{LMS}}^2}{\sigma_{e,\mathrm{SE\text{-}LMS}}^2} \propto \frac{(\mathbf{m}^T\mathbf{R}^{-T}\mathbf{m})\ln\left[\frac{\pi}{2}(\mathbf{m} \pm \mathbf{m_x})^T\mathbf{R}^{-T}(\mathbf{m} \pm \mathbf{m_x})\right]}{-m_e^2}. \tag{41}$$

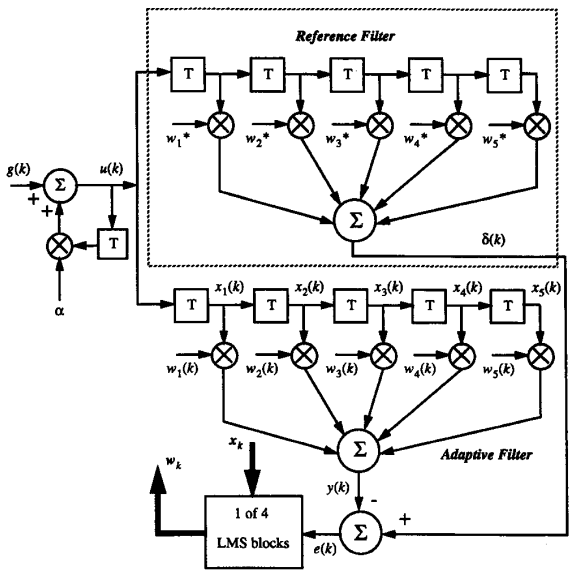


Fig. 3. The setup used to simulate the adaptive filter.

(33), the predicted MSE is weakly dependent on the input signal power, σ_x^2. As well, from (47) we see that in the absence of dc offsets, the SS-LMS algorithm, like the SE-LMS algorithm, will experience a residual excess MSE that is shaped by μ. Finally note that if $|m_i| > 1$, it can be inferred from (15) that the sign of the parenthesized term will be governed by m_i and the coefficient estimate $c_i(k)$ will drift in a direction governed by this offset and the SS-LMS algorithm will diverge. This behavior is similar to that alluded to for the SE-LMS algorithm.

VII. Numerical Verification

A 5-tap ($N = 5$) linear combiner, as illustrated in Fig. 3, was investigated to compare the simulated performance of the filter with the analytical predictions. The input, $g(k)$, was a zero-mean white Gaussian distribution. The first-order lowpass filter was used to vary the input statistics to the linear combiner through the parameter α where

$$U(z) = \frac{1}{1 - \alpha z^{-1}} G(z). \tag{48}$$

The results of the simulations and the predicted analytical calculations for various cases are provided in Fig. 4. The circles depict the predicted MSE calculated from (24), (33), (38), and (45) and the simulated MSE at the respective value for μ. A nonlinear equation solver provided by the software package MATLAB [17] was used to solve (38) and (45). The dotted lines and the solid lines connect the circles obtained from the analytical expressions and the simulations, respectively, to exemplify the behavior of the MSE as function of μ. The offsets levels for Fig. 4(a)–(d) are:

$$m_e = 0.01$$
$$\mathbf{m}_\mathbf{x}^T = [0.02 \quad -0.01 \quad -0.03 \quad -0.005 \quad 0.07]$$
$$\mathbf{m}^T = [0.08 \quad 0.01 \quad -0.05 \quad -0.02 \quad -0.06]$$

while the offset levels for Fig. 4(e) and (f) are:

$$m_e = 0.02$$
$$\mathbf{m}_\mathbf{x}^T = [0.02 \quad -0.0 \quad -0.07 \quad 0.05 \quad -0.008]$$
$$\mathbf{m}^T = [0.03 \quad -0.1 \quad 0.005 \quad -0.08 \quad -0.06].$$

The value for α, σ_x^2 and $\mathbf{R}$ for each sub-figure sequentially are:

$\alpha = 0, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 0.9968 & -0.0010 & -0.0005 & -0.0040 & 0.0006 \\ -0.0010 & 0.9968 & -0.0010 & -0.0005 & -0.0040 \\ -0.0005 & -0.0010 & 0.9968 & -0.0010 & -0.0005 \\ -0.0040 & -0.0005 & -0.0010 & 0.9968 & -0.0010 \\ 0.0006 & -0.0040 & -0.0005 & -0.0010 & 0.9968 \end{bmatrix}$$

$\alpha = 0.4, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 1.1849 & 0.4722 & 0.1869 & 0.0711 & 0.0291 \\ 0.4722 & 1.1849 & 0.4722 & 0.1869 & 0.0711 \\ 0.1869 & 0.4722 & 1.1849 & 0.4722 & 0.1869 \\ 0.0711 & 0.1869 & 0.4722 & 1.1849 & 0.4722 \\ 0.0291 & 0.0711 & 0.1869 & 0.4722 & 1.1849 \end{bmatrix}$$

$\alpha = 0.8, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 2.7560 & 2.2019 & 1.7592 & 1.4051 & 1.2261 \\ 2.2019 & 2.7560 & 2.2019 & 1.7592 & 1.4051 \\ 1.7592 & 2.2019 & 2.7560 & 2.2019 & 1.7592 \\ 1.4051 & 1.7592 & 2.2019 & 2.7560 & 2.2019 \\ 1.1261 & 1.4051 & 1.7592 & 2.2019 & 2.7560 \end{bmatrix}$$

$\alpha = 0.4, \sigma_x^2 = 0.25,$

$$\mathbf{R} = \begin{bmatrix} 0.2962 & 0.1180 & 0.0467 & 0.0178 & 0.0073 \\ 0.1180 & 0.2962 & 0.1180 & 0.0467 & 0.0178 \\ 0.0467 & 0.1180 & 0.2962 & 0.1180 & 0.0467 \\ 0.0178 & 0.0467 & 0.1180 & 0.2962 & 0.1180 \\ 0.0073 & 0.0178 & 0.0467 & 0.1180 & 0.2962 \end{bmatrix}$$
$$= \frac{\mathbf{R}(\text{Fig.4b})}{4}$$

$\alpha = 0.65, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 1.7232 & 1.1192 & 0.7240 & 0.4636 & 0.3028 \\ 1.1192 & 1.7232 & 1.1192 & 0.7239 & 0.4636 \\ 0.7240 & 1.1192 & 1.7232 & 1.1192 & 0.7239 \\ 0.4636 & 0.7239 & 1.1192 & 1.7232 & 1.1192 \\ 0.3028 & 0.4636 & 0.7239 & 1.1192 & 1.7232 \end{bmatrix}$$

$\alpha = 0.9, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 5.2106 & 4.6857 & 4.2141 & 3.7899 & 3.4121 \\ 4.6857 & 5.2106 & 4.6857 & 4.2141 & 3.7899 \\ 4.2141 & 4.6857 & 5.2106 & 4.6857 & 4.2141 \\ 3.7899 & 4.2141 & 4.6857 & 5.2106 & 4.6857 \\ 3.4121 & 3.7899 & 4.2141 & 4.6857 & 5.2106 \end{bmatrix}$$

Fig. 4(a)–(d) depicts the effects of μ and $\mathbf{R}$ on the excess MSE for each of the four algorithms with the same offset

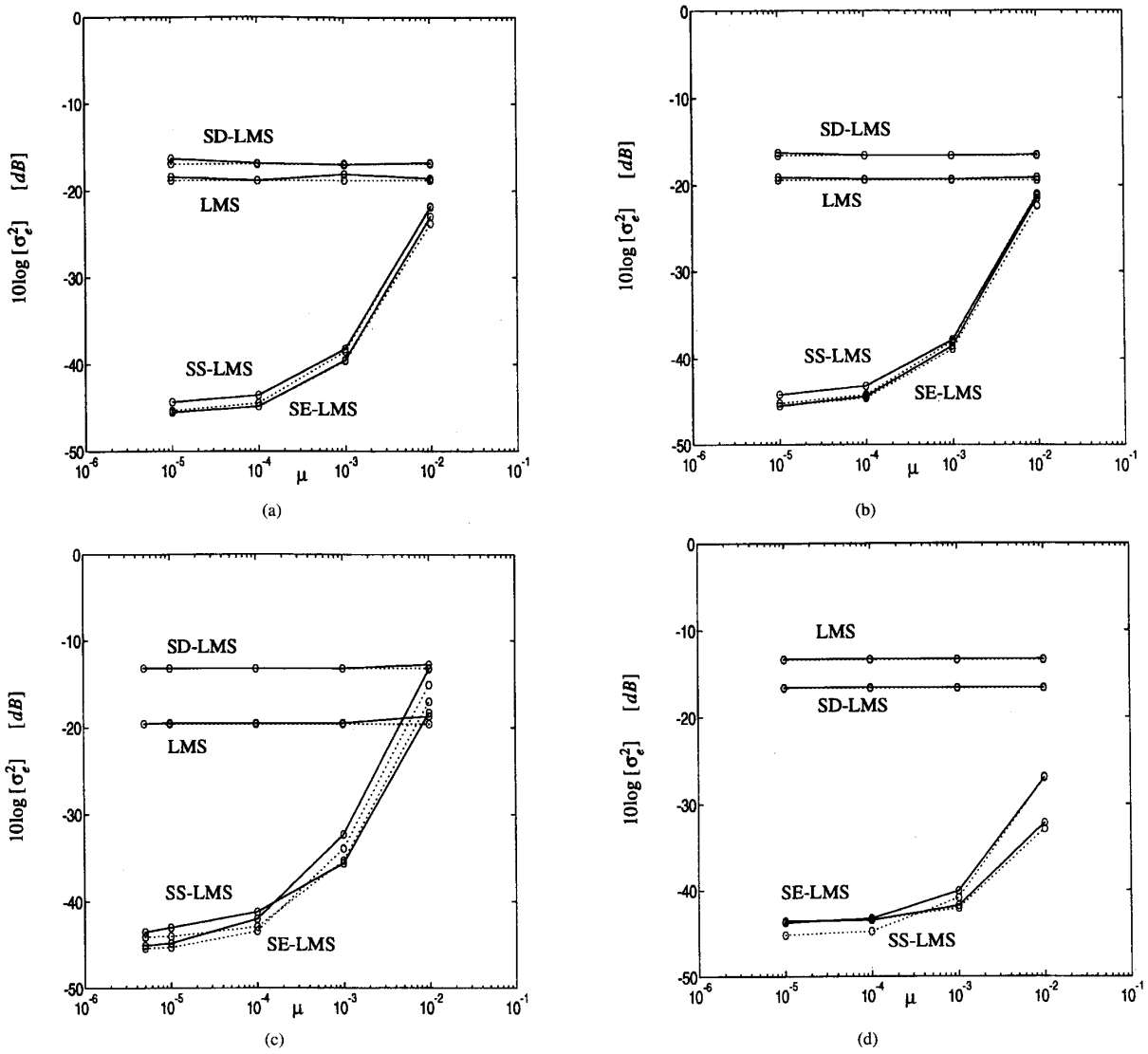
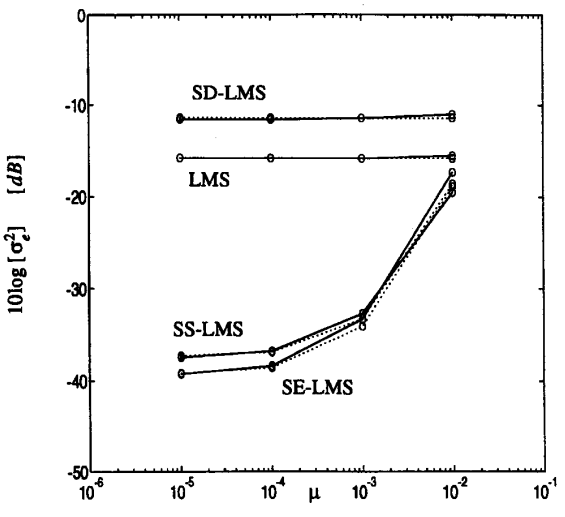


Fig. 4. (a) Theoretical (dotted lines) and simulated (solid lines) MSE as function of μ, $\mathbf{R}$, and different offset levels for the four LMS based algorithms.

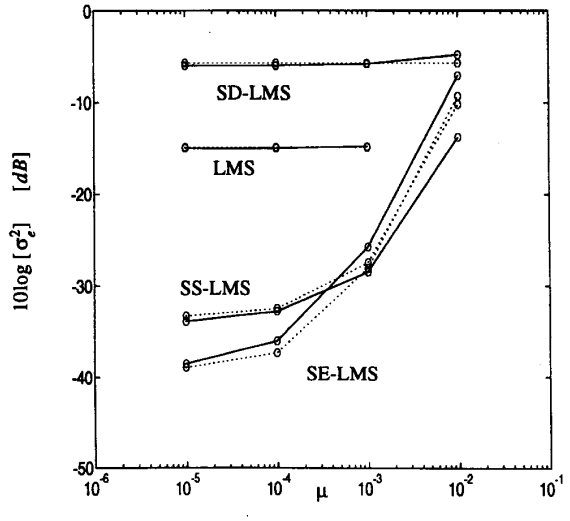
levels. Fig. 4(a) shows the case for a Gaussian white noise input. Fig. 4(b) and (c) show the results for more colored Gaussian inputs as given by the $\mathbf{R}$ matrices above. Fig. 4(d), unlike Fig. 4(a)–(c), shows the results when the input power is smaller than unity. Observe that in this case (compared with Fig. 4(b)) the excess MSE using the LMS algorithm is greatly increased, while the others are less sensitive to input power as was discussed herein. Fig. 4(e) and (f) shows another case for different offset levels. For the case of Fig. 4(f), the LMS algorithm showed evidence of divergence for $\mu = 0.01$ hence this point is omitted from the plot. The results of Fig. 4 verify the derived analytical expressions given by (24), (33), (38), and (45) for arbitrary offset levels and arbitrary $\mathbf{R}$ matrices. Specifically, note that the SE-LMS and the SS-LMS algorithms are shaped by μ and that the limiting cases for $\mu \to 0$ expressed by (39) and (46) compare well with simulated data. Observe also from all the results that the SE-LMS and the SS-LMS algorithms achieve much better MSE performance in the presence of dc offsets. In addition, it is evident (Fig. 4(c) and (f)) that the analytical results deviate from the simulation results at larger μ and for more colored inputs, as the approximations made become less appropriate for these conditions.

Fig. 5 depicts the excess MSE for each of the algorithms as function of μ with $m_e = 0$. The offsets and the matrix $\mathbf{R}$ (corresponding to a highly correlated input vector) for this simulation were:

$$\mathbf{m}_x^T = [0.02 \quad -0.01 \quad -0.03 \quad -0.005 \quad 0.07]$$
$$\mathbf{m}^T = [0.08 \quad 0.01 \quad -0.05 \quad -0.02 \quad -0.06]$$



(e)



(f)

Fig. 4. *Continued.*

$\alpha = 0.95, \sigma_x^2 = 1,$

$$\mathbf{R} = \begin{bmatrix} 10.0934 & 9.5821 & 9.0970 & 8.6364 & 8.2027 \\ 9.5821 & 10.0934 & 9.5820 & 9.0970 & 8.6364 \\ 9.0970 & 9.5820 & 10.0934 & 8.5820 & 9.0970 \\ 8.6364 & 9.0970 & 9.5820 & 10.0934 & 9.5820 \\ 8.2027 & 8.6364 & 9.0970 & 9.5820 & 10.0934 \end{bmatrix}$$

These results validate the predicted behavior of (24), (33), (40), and (47). Specifically, it appears that as long as m_e is nulled, the MSE of an adaptive filter using the SE-LMS and the SS-LMS algorithms is shaped by μ. This is not true for the LMS algorithm or the SD-LMS algorithm. Consequently, much better MSE performance in the presence of dc offsets can be attained using the SE-LMS or the SS-LMS algorithms. However, as mentioned earlier, the SS-LMS algorithm can diverge depending on input statistics due to gradient misalignment. Thus we conclude that the SE-LMS algorithm is the best choice. Table I summarizes the results presented and the issues discussed in this paper.

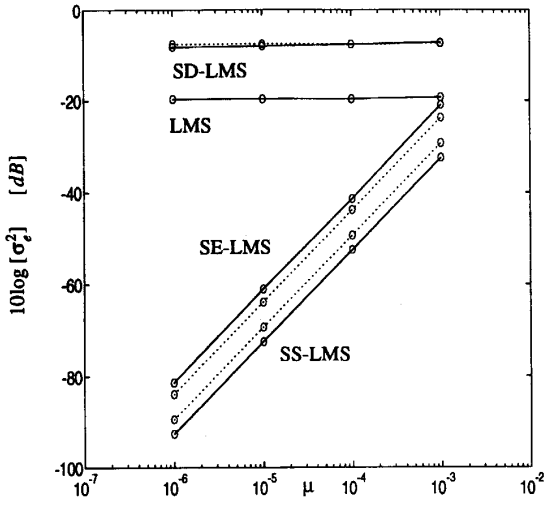


Fig. 5. MSE as function of μm for the four LMS based algorithm with $m_e = 0$.

VIII. Conclusion

We have analyzed the performance and provided analytical expressions for the performance of four coefficient update algorithms for analog adaptive filters from an offset point of view. We have found that both the SE-LMS and the SS-LMS algorithms achieve better MSE performance when dc offsets are present; especially when integrator offsets, which dominate in a practical analog system, are unavoidable and in high frequency applications where simply passing the error signal through a gain stage to reduce the effects of dc offsets [2] is impractical. For the SE-LMS and the SS-LMS algorithms, it was shown that by minimizing m_e the MSE can be reduced, whereas this is not the case for the LMS or the SD-LMS algorithms. The practicality of minimizing m_e was briefly discussed. It was observed that if offsets can be controlled it is possible to reduce the excess MSE by having the offsets cancel one another (most likely impractical). Also, some comments were given on the possibility of algorithm divergence due to excessive dc offsets.

In terms of implementation complexity the LMS algorithm is the most complex while the SS-LMS algorithm is the simplest. Between the SD-LMS algorithm and the SE-LMS algorithm the former is more hardware intensive as N slicers for the N gradient signals will be required while only 1 slicer would be required for the latter.

Having lower offset sensitivity, minimal circuit complexity combined with the fact that the SD-LMS and the SS-LMS algorithms can diverge due to gradient signal misalignment [5], it appears the SE-LMS algorithm is the best choice as an algorithm for practical high-frequency analog adaptive filters.

Finally, we would like to point out that the analysis done here is idealized in the sense that the effects of noise, coef-

TABLE I
RESULT SUMMARY

Test Case	LMS	SD-LMS	SE-LMS	SS-LMS
input power	$\sigma_e^2 \propto 1/\sigma_x^2$	no effect	$\sigma_e^2 \propto 1/ln[\sigma_x^2]$	no effect
no offsets	$\sigma_e^2 \to 0$ for $\mu \to 0$	$\sigma_e^2 \to 0$ for $\mu \to 0$	$\sigma_e^2 \propto \mu^2\sigma_x^4$	$\sigma_e^2 \propto \mu^2\sigma_x^2$
all offsets	σ_e^2 weakly depends on μ for $\mu \to 0$ LMS $\sigma_e^2 \propto (\boldsymbol{m} + m_e\boldsymbol{m}_x)^T(\boldsymbol{m} + m_e\boldsymbol{m}_x)$ SD-LMS $\sigma_e^2 \propto (\boldsymbol{m} + m_e\boldsymbol{k}_{mx})^T(\boldsymbol{m} + m_e\boldsymbol{k}_{mx})$		σ_e^2 strongly depends on μ for $\mu \to 0$ SE-LMS $\sigma_e^2 \propto \frac{m_e^2}{ln[(\boldsymbol{m}+\boldsymbol{m}_x)^T(\boldsymbol{m}+\boldsymbol{m}_x)]}$ SS-LMS $\sigma_e^2 \propto \frac{m_e^2}{ln[\boldsymbol{m}^T\boldsymbol{m}/e^{m_{xi}^2/\sigma_x^2}]}$	
$m_e = 0$	$\sigma_e^2 \propto \boldsymbol{m}^T\boldsymbol{m}$	$\sigma_e^2 \propto \boldsymbol{m}^T\boldsymbol{m}$	σ_e^2 is scaled by μ^2 SE-LMS $\sigma_e^2 \propto \mu^2(\boldsymbol{m}^T\boldsymbol{m},\ \boldsymbol{m}_x^T\boldsymbol{m}_x)^2$ SS-LMS $\sigma_e^2 \propto \mu^2(\boldsymbol{m}^T\boldsymbol{m},\ e^{m_{xi}^2/\sigma_x^2})^2$	
algorithm circuit complexity	1 multiplier/tap 1 integrator/tap	1 slicer/tap 1 trivial multiplier/tap 1 integrator/tap	1 trivial multiplier/tap 1 integrator/tap 1 slicer/filter	1 slicer/tap 1 XOR gate/tap 1 counter/tap 1 DAC/tap 1 slicer/filter
convergence	no gradient misalignment	gradients misaligned	no gradient misalignment	gradients misaligned

ficient leakage due to damped integrators and other analog circuit nonidealities were not considered. These issues are addressed in [10].

APPENDIX A

Here we evaluate

$$E[\mathrm{sgn}[\mathbf{x} + \mathbf{m_x}]] = \begin{bmatrix} E[\mathrm{sgn}[x_1 + m_{x1}]] \\ E[\mathrm{sgn}[x_2 + m_{x2}]] \\ \cdots \\ E[\mathrm{sgn}[x_N + m_{xN}]] \end{bmatrix}. \quad \text{(A-1)}$$

Considering the i^{th} element, we obtain

$$\begin{aligned} E[\mathrm{sgn}[x_i + m_{xi}]] &= -P(x_i + m_{xi} \le 0) + P(x_i + m_{xi} > 0) \\ &= -P(x_i \le -m_{xi}) + P(x_i > -m_{xi}) \\ &= -F_X[-m_{xi}] + (1 - F_X[-m_{xi}]) \\ &= 1 - 2F_X[-m_{xi}] \end{aligned} \quad \text{(A-2)}$$

where $P(\bullet)$ denotes the probability operator and $F_X[\bullet]$ denotes the cumulative distribution function. For a zero-mean Gaussian distribution with $m_{xi} \le 0$ we have [13]

$$\begin{aligned} F_X[-m_{xi}] &= \frac{1}{\sqrt{2\pi}\sigma_x}\int_{-\infty}^{-m_{xi}} e^{-\left(\frac{x_i^2}{2\sigma_x^2}\right)}\,dx_i \\ &= \frac{1}{2} + \frac{1}{2}\ \mathrm{erf}\left[-\frac{m_{xi}}{\sqrt{2\sigma_x^2}}\right] \end{aligned} \quad \text{(A-3)}$$

where $\mathrm{erf}[u] = \frac{2}{\sqrt{\pi}}\int_0^u e^{-z^2}dz$. Noting that $\mathrm{erf}[u]$ is an odd function in u and substituting (A-3) into (A-2) yields:

$$E[\mathrm{sgn}[x_i + m_{xi}]] = \mathrm{erf}\left[\frac{m_{xi}}{\sqrt{2\sigma_x^2}}\right]. \quad \text{(A-4)}$$

Had we taken $m_{xi} > 0$, a similar result would have been obtained, thus (A-4) is true for all m_{xi}. The result of (26) is then easily obtained from (A-4) and (A-1).

APPENDIX B

Here we evaluate

$$E[\mathrm{sgn}[\mathbf{x} + \mathbf{m_x}]\mathbf{x}^T] = E\left[\begin{bmatrix} \mathrm{sgn}[x_1 + m_{x1}] \\ \mathrm{sgn}[x_2 + m_{x2}] \\ \cdots \\ \mathrm{sgn}[x_N + m_{xN}] \end{bmatrix}[x_1 x_2 \cdots x_N]\right]. \quad \text{(B-1)}$$

Considering the i,j^{th} element of the above matrix, for any two zero-mean Gaussian variables x_i, x_j, with covariance $E[x_i x_j] = \sigma_{x_i}\sigma_{x_j}\rho_{i,j}$, using Price's Theorem [14] we obtain:

$$\begin{aligned} &\frac{\partial}{\partial\rho_{i,j}}E[\mathrm{sgn}[x_i + m_{xi}]x_j] \\ &\quad = \sigma_{x_i}\sigma_{x_j}E\left[\frac{d}{dx_i}\mathrm{sgn}[x_i + m_{xi}]\frac{d}{dx_j}x_j\right]. \end{aligned} \quad \text{(B-2)}$$

For the case of Gaussian signals (B-2) becomes

$$\frac{\partial}{\partial\rho_{i,j}}E[\mathrm{sgn}[x_i+m_{xi}]x_j] = 2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{\delta(x_i+m_{xi})}{2\pi\sqrt{1-\rho_{i,j}^2}} \times e^{-\left(\frac{x_i^2}{\sigma_{x_i}^2}+\frac{x_j^2}{\sigma_{x_j}^2}-\frac{2\rho_{i,j}x_ix_j}{\sigma_{x_i}\sigma_{x_j}}\right)\Big/2(1-\rho_{i,j}^2)}dx_idx_j. \quad \text{(B-3)}$$

Integrating the right-hand side of (B-3) with respect to x_i gives

$$\frac{\partial}{\partial\rho_{i,j}}E[\mathrm{sgn}[x_i+m_{xi}]x_j] = \frac{1}{\pi\sqrt{1-\rho_{i,j}^2}}\int_{-\infty}^{\infty}e^{-\left(\frac{m_{xi}^2}{\sigma_{x_i}^2}+\frac{x_j^2}{\sigma_{x_j}^2}-\frac{2\rho_{i,j}m_{xi}x_j}{\sigma_{x_i}\sigma_{x_j}}\right)\Big/2(1-\rho_{i,j}^2)}dx_j. \quad \text{(B-4)}$$

After rearranging terms and completing the square (B-4) simplifies to

$$\frac{\partial}{\partial\rho_{i,j}}E[\mathrm{sgn}[x_i+m_{xi}]x_j] = \frac{e^{-m_{xi}^2/2\sigma_{x_i}^2}}{\pi\sqrt{1-\rho_{i,j}^2}}\int_{-\infty}^{\infty}e^{-\left(\frac{x_j}{\sigma_{x_j}^2}+\frac{\rho_{i,j}m_{xi}}{\sigma_{x_i}}\right)^2\Big/2(1-\rho_{i,j}^2)}dx_j. \quad \text{(B-5)}$$

Solving for (B-5) yields

$$\frac{\partial}{\partial\rho_{i,j}}E[\mathrm{sgn}[x_i+m_{xi}]x_j] = \sqrt{\frac{2}{\pi}}\sigma_{x_j}e^{-m_{xi}^2/2\sigma_{x_i}^2}. \quad \text{(B-6)}$$

Integrating both sides of (B-6) with respect to $\rho_{i,j}$ gives

$$E[\mathrm{sgn}[x_i+m_{xi}]x_j] = \sqrt{\frac{2}{\pi}}\sigma_{x_j}e^{-m_{xi}^2/2\sigma_{x_i}^2}\int_0^{\rho_{i,j}}d\hat{\rho}_{i,j}. \quad \text{(B-7)}$$

Solving this trivial integral and substituting for $\rho_{i,j}$ results in

$$E[\mathrm{sgn}[x_i+m_{xi}]x_j] = \sqrt{\frac{2}{\pi}}\frac{1}{\sigma_{x_i}}e^{-m_{xi}^2/2\sigma_{x_i}^2}E[x_ix_j]. \quad \text{(B-8)}$$

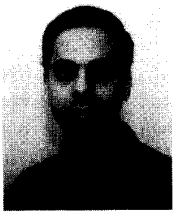
The expression in (27) follows readily by substituting (B-8) into (B-1).

ACKNOWLEDGMENT

In addition, they thank the reviewers of the manuscript for their valuable input.

REFERENCES

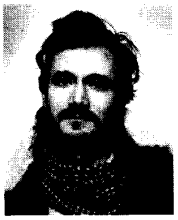
[1] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*. New York: Wiley, 1987.
[2] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Continuous-time LMS adaptive recursive filters," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 769–778, July 1991.
[3] H. Sari, "Performance evaluation of three adaptive equalization algorithms," *IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, pp. 1385–1389, May 1982.
[4] W. A. Sethares *et al.*, "Excitation conditions for signed regressor least mean squares adaptation," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 613–624, June 1988.
[5] T. A. C. M. Claasen and W. F. G. Mecklenbrauker, "Comparison of the convergence of two algorithms for adaptive FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, pp. 670–678, June 1981.
[6] S. Dasgupta and C. R. Johnson Jr., "Some comments on the behavior of sign-sign adaptive identities," *Syst. Contr. Lett.*, vol. 7, April 1986.
[7] D. L. Duttweiler, "Adaptive filter performance with nonlinearities in the correlation multiplier," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-30, pp. 578–586, Aug. 1982.
[8] C. R. Rohrs, C. R. Johnson, and J. D. Mills, "A stability problem in sign-sign adaptive algorithms," *IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, pp. 2999–3001, April 1986.
[9] T. Enomoto *et al.*, "Integrated MOS offset error canceller for analogue adaptive transversal filter," *Electron. Lett.*, vol. 19, pp. 968–970, Nov. 1983.
[10] U. Menzi and G. S. Moschytz, "Adaptive switched capacitor filters based on the LMS algorithms," *IEEE Trans. Circuits Syst. I*, vol. 40, no. 12, pp. 929–942, Dec. 1993.
[11] C.-P. J. Tzeng, "An adaptive offset cancellation technique for adaptive filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 389, pp. 799–803, May 1990.
[12] H. Qiuting, "Offset compensation scheme for analogue LMS adaptive filters," *Electron. Lett.*, vol. 28, pp. 1203–1205, June 1992.
[13] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1991, third ed.
[14] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 69–72, June 1958.
[15] A. Shoval, D. A. Johns, and W. M. Snelgrove, "Median-based offset cancellation circuit technique," in *Proc. IEEE Symp. Circuits Syst.*, vol. 4, pp. 2033–2036, May 1992.
[16] H. Sari, "Algorithms d'egalisation adaptive d'un canal dispersif," Ph.D. dissertation, Ecole Nationale Superieure des Telecommunications, Paris, Oct. 1980.
[17] MATLAB, *Matrix Laboratory*, The MathWorks, Inc., Natwick, MA, USA, 1992.



Ayal Shoval (S'89) received the B.A.Sc., M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Canada, in 1989, 1991, and 1995, respectively.

His research focused on efficient implementations of analog adaptive filters for high-speed data communications. He is currently at AT&T Bell Labs, Allentown, PA, working on high-speed LAN's.

David A. Johns (M'88) for a photograph and biography, see this issue, p. 175.



W. Martin Snelgrove (S'75–M'78) received the B.A.Sc. degree in chemical engineering in 1975, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Canada, in 1977 and 1982, respectively.

In 1982 he was with INAOE in Mexico as a Visiting Researcher in CAD, returning to Toronto as an Assistant Professor and then Associate Professor until 1992. He spent sabbatical leave in 1989 and 1990 as a Resident Visitor at AT&T Bell Labs, Reading, PA, working in CMOS analog design. In July 1992 he joined the Department of Electronics at Carleton University, Ottawa, Canada as a Full Professor, where he holds the OCRI/NSERC Industrial Research Chair in High-Speed Integrated Circuits. His recent research work has been in adaptive analog and digital filtering, data-converter architecture and circuits, and highly parallel architectures for signal processing. He has done collaborative research with about a dozen companies.

Dr. Snelgrove has published about 70 papers, one of which won the 1986 CAS Society Guillemin-Cauer award.