A New Adaptive Filter for Estimating and Tracking the Delay and the Amplitude of a Sinusoid

Mrityunjoy Chakraborty, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a new adaptive filter for estimating and tracking the delay and the relative amplitude of a sinusoid vis-a-vis a reference sinusoid of the same frequency. By careful choice of the sampling period, a two-tap finite-impulse response (FIR) filter model is constructed for the delayed signal. The delay and the amplitude are estimated by identifying the FIR filter for which a delay variable and an amplitude variable are updated in an LMS-like manner, deploying, however, separate step sizes. Convergence analysis proving convergence (in mean) of the delay and the amplitude updates to their respective true values is provided, and necessary convergence conditions are established. Stability regions in the step-size plane are also identified that guarantee bounded steady-state error variance for the delay and the amplitude estimates. The proposed method is computationally simple as the primary computation is a rotation of a vector that can efficiently be implemented using CORDIC processors. MATLAB-based simulation studies confirm satisfactory estimation performance of the proposed algorithm.**

***Index Terms*—Adaptive filters, convergence analysis, least mean square (LMS) algorithm, mean square error (MSE), sinusoidal signals, time-delay estimation.**

I. Introduction

ESTIMATION of time delay(s) between the noisy versions of a signal received at two or more spatially separated sensors [1] has been an important topic in areas such as radar and sonar ranging, target localization and tracking, speed sensing, direction finding, synchronization in communication receivers, biomedicine, exploration geophysics etc. For stationary time delay, a popular offline technique is the generalized cross correlator [2], which estimates the delay by locating the peak of the cross-correlation function of the filtered versions of the observed data. While this approach can provide maximum-likelihood estimation performance under Gaussian signal and noise assumptions, the resolution of the delay estimate in this technique, however, is limited by the sampling period. For the case of deterministic signals, offline techniques have been proposed that achieve subsample accuracy. These include the discrete-time Fourier transform-based method [3], quadrature delay estimator [4], and estimators based on a combination of cross correlation and autocorrelation [5].

When the time delay is time-varying due to relative source/receiver motion, adaptive tracking of it is necessary. In [6], an adaptive tracker was proposed, which adapts a finite-impulse response (FIR) filter to model the time delay and estimates it by interpolating the filter coefficients. Alternatively, explicit delay adjustments have been achieved [7]–[10] by constraining the filter coefficients to be some functions of the time delay.

For sinusoidal signals that commonly occur in radar, sonar, and digital communication, an adaptive filter was proposed in [11] that estimates the delay between the received copies of a sinusoid of known frequency, received at two spatially separated sensors. This algorithm estimates the delay by directly updating a delay variable, but it considers only a special case where both the received sinusoids are assumed to have unity amplitude simultaneously. In practice, however, the relative amplitude of the signals is also unknown and is often time-varying, like the delay. In such cases, estimation and tracking of the delay is not possible without simultaneous estimation and tracking of the amplitude. Presence of two unknown time-varying parameters, namely, delay and amplitude, however, makes the estimation problem much more complicated than for a single unknown (i.e., delay) case. In this paper, we present a more general treatment to the problem and propose a new adaptive filter that estimates and tracks both the delay and the relative amplitude of one of the two received signals vis-a-vis the other. The development is based on choosing the sampling rate from a specific set of permissible values, which generates a two-tap FIR filter model for the delayed signal and also results in a diagonal autocorrelation matrix for the 2×1 filter input vector. An adaptive algorithm is developed to estimate and track both the delay and the relative amplitude of the delayed signal by identifying the filter coefficients. For this, a delay variable and also an amplitude variable are time updated in an LMS-like manner, with the former shown to converge in mean to the true delay value and the latter shown to converge in mean to the true amplitude with a bias that is negligible (under high input SNRs) and also which can easily be corrected. To ensure convergence, separate step sizes are, however, necessary for the delay and the amplitude variables. A detailed stability analysis is also carried out, and stability regions in the step-size plane are worked out for keeping the steady-state delay and amplitude error variance bounded. The major computation in the proposed method is a rotation of a vector that can be implemented efficiently by CORDIC processors. MATLAB-based simulation studies

Manuscript received August 27, 2009; revised January 4, 2010; accepted January 5, 2010. Date of publication May 17, 2010; date of current version October 13, 2010. The Associate Editor coordinating the review process for this paper was Dr. Jerome Blair.

The author is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721 302, India (e-mail: mrityun@ece.iitkgp.ernet.in).

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/TIM.2010.2046590

confirm satisfactory estimation performance of the proposed algorithm.

The organization of this paper is as follows. In Section II-A, we develop the proposed delay estimation algorithm; in Sections II-B and II-C, we present convergence analysis in mean and mean square, respectively; and in Section III, we provide MATLAB-based simulation results.

II. Adaptive Delay Estimation

A. Proposed Algorithm

Consider the following model for the signals received at the two sensors, namely, $x_a(t)$ and $y_a(t)$:

$$x_a(t) = s_a(t) + u_a(t) \quad (1)$$

$$y_a(t) = A\, s_a(t - D) + z_a(t) \quad (2)$$

where $s_a(t) = \cos(\Omega t + \phi)$ is a sinusoid with known analog frequency Ω and random phase ϕ that is uniformly distributed over $[0, 2\pi)$. The two terms $u_a(t)$ and $z_a(t)$ represent two zero-mean additive white Gaussian noise processes independent of each other and also of ϕ and, thus, of $s_a(t)$. The variable D is the delay between the received copies of the signal $s_a(t)$ at the two sensors, which is unknown and is to be estimated and tracked. The term A is a gain factor associated with $s_a(t - D)$ and is assumed to be unknown. It is also assumed that the net phase shift ΩD due to the delay lies within $[0, 2\pi)$, which eliminates possibilities of ambiguity over D.

The signals $x_a(t)$ and $y_a(t)$ are digitized with a sampling period τ to generate the sequences $x(n) \equiv x_a(n\tau) = \cos(\Omega n\tau + \phi) + u_a(n\tau) \equiv s(n) + u(n)$ and $y(n) \equiv y_a(n\tau) = A\cos(\Omega(n\tau - D) + \phi) + z_a(n\tau) = A\cos(\Omega n\tau + \phi)\cos(\Omega D) + A\sin(\Omega n\tau + \phi)\sin(\Omega D) + z(n)$, where $s(n) \equiv s_a(n\tau) = \cos(\Omega n\tau + \phi)$, $u(n) \equiv u_a(n\tau)$, and $z(n) \equiv z_a(n\tau)$. Our key idea is to select a specific sampling period τ satisfying $\Omega\tau = \pi/2r$, $r \in Z = \{1, 2, \ldots\}$, or, equivalently, $\Omega_s = 4r\Omega$, where $\Omega_s = 2\pi/\tau$ is the sampling frequency. It is then possible to write

$$y(n) = A\cos(\Omega D)\, s(n) + A\sin(\Omega D)\, s(n - r) + z(n), r \in Z. \quad (3)$$

Using (3), one can model $y(n)$ as the noisy output of a system with transfer function $A(\cos(\Omega D) + \sin(\Omega D) z^{-r})$ (unknown). Estimation and tracking of the delay then turn out to be a system identification problem with noisy input, where one can use a standard LMS-based two-tap adaptive filter with coefficients, say, $w_0(n)$ and $w_r(n)$, and take the delay estimate as $D(n) = (1/\Omega)\tan^{-1}(w_r(n)/w_0(n))$. However, while this approach guarantees convergence of $E[w_0(n)]$ and $E[w_r(n)]$ to the respective true system parameters, the same cannot be said about the mean delay estimate $E[D(n)]$ vis-a-vis the true delay value D (at least theoretically). In addition, stability conditions required to keep the steady-state value of $E[D(n) - D]^2$ within bound are difficult to establish in this case. In this respect, a simpler and also more appropriate approach, in our opinion, would be to use the *a priori* knowledge of the specific forms of the system coefficients, viz., $A\cos(\Omega D)$ and $A\sin(\Omega D)$, and precondition the adaptive filter as $\mathbf{w}(n) = a(n)\overline{\mathbf{w}}(n)$, where $\overline{\mathbf{w}}(n) = [\cos(\Omega D(n)), \sin(\Omega D(n))]^T$, with T denoting transposition. The filter weights are then adjusted by directly updating the parameter estimate vector $\boldsymbol{\theta}(n) = [D(n), a(n)]^T$ in an LMS-like manner, as shown in Fig. 1. However, unlike the conventional LMS algorithm [12], we choose two different step sizes, namely, μ_1 and μ_2, respectively, for $D(n)$ and $a(n)$, for reasons explained later. Defining $\boldsymbol{\mu} = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}$, the corresponding LMS update equations are given by

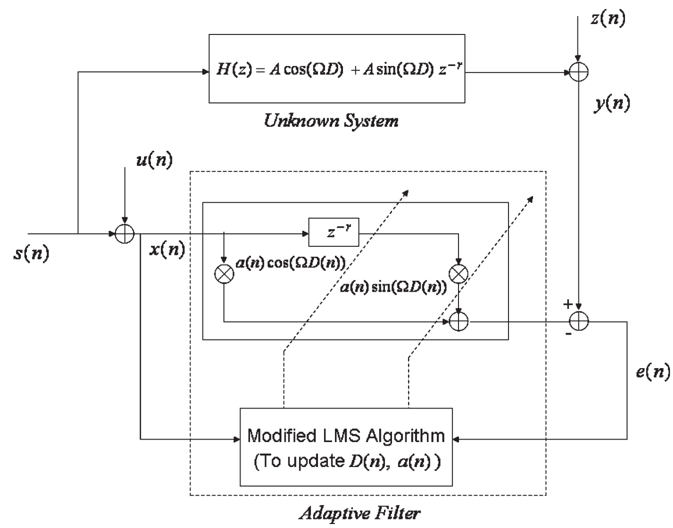

Fig. 1. System identification model for delay and amplitude estimation.

$$\begin{aligned}\boldsymbol{\theta}(n+1) &= \boldsymbol{\theta}(n) - \boldsymbol{\mu}\,\nabla_\theta e^2(n) \\ &= \boldsymbol{\theta}(n) - 2\,\boldsymbol{\mu}\,\nabla_\theta e(n)\, e(n) \end{aligned} \quad (4)$$

where $\nabla_\theta(.) = [\partial(.)/\partial D(n), \partial(.)/\partial a(n)]^T$, and the error signal $e(n)$ is given as $e(n) = y(n) - \mathbf{w}^T(n)\mathbf{x}(n)$, with $\mathbf{x}(n) = [x(n), x(n-r)]^T$, $r \in Z$. It is easy to verify that $\partial e(n)/\partial D(n) = -\Omega a(n)\mathbf{w}'^T(n)\mathbf{x}(n)$, where $\mathbf{w}'(n) = [-\sin(\Omega D(n)),\ \cos(\Omega D(n))]^T$. Similarly, $\partial e(n)/\partial a(n) = -\overline{\mathbf{w}}^T(n)\mathbf{x}(n)$. This results in the following update equation:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + 2\,\boldsymbol{\mu}\,\boldsymbol{\Omega}\,\mathbf{U}(n)\,\mathbf{x}(n)\, e(n) \quad (5)$$

where

$$\mathbf{U}(n) = \begin{bmatrix} a(n)\,\mathbf{w}'^T(n) \\ \overline{\mathbf{w}}^T(n) \end{bmatrix} \quad \boldsymbol{\Omega} = \begin{bmatrix} \Omega & 0 \\ 0 & 1 \end{bmatrix}.$$

Note that the main computation in the update term in (5) is actually a rotation of the vector $\mathbf{x}(n)$ by the angle $\omega(n) = -\Omega D(n)$, which is given by

$$\begin{bmatrix} \mathbf{w}'^T(n) \\ \overline{\mathbf{w}}^T(n) \end{bmatrix} \mathbf{x}(n) \equiv \begin{bmatrix} -\sin(\omega(n)) & \cos(\omega(n)) \\ \cos(\omega(n)) & \sin(\omega(n)) \end{bmatrix} \mathbf{x}(n).$$

Further note that the above computation also evaluates the term $\overline{\mathbf{w}}^T(n)\mathbf{x}(n)$ which, after multiplication by $a(n)$, generates the filter output $y(n)$. The above rotation can be carried out by a sequence of CORDIC rotations [14], [15] and, thus, can efficiently be implemented on a pipelined array of CORDIC processors. Such computational simplifications are, however, not available in a standard LMS based identification of (3) and estimation of D.

In the following, we show that the update equation (5) results in convergence of $\boldsymbol{\theta}(n)$ in mean as $\lim_{n\to\infty} E[\boldsymbol{\theta}(n)] = [D, A + \epsilon(A)]^T$, where $\epsilon(A) = -2A\sigma_u^2$ is a bias term (negligible under high input SNRs), provided μ_1 and μ_2 are chosen to satisfy $0 < \mu_1 < 2/\Omega^2 A^2$ and $0 < \mu_2 < 2$.

B. Convergence (in Mean) Analysis

First, recall that the phase ϕ is uniformly distributed over $[0, 2\pi)$, meaning the signal autocorrelation is given as $r_{ss}(k) = E[s(n)s(n-k)] = (1/2\pi)\int_0^{2\pi} \cos(\Omega n\tau + \phi)\cos(\Omega(n-k)\tau + \phi)d\phi = (1/2)\cos(k\pi/2r)$, $r \in Z$, where k is any integer. From this and also from the fact that $u(n)$ is a zero-mean white process independent of $s(n)$, it then follows that the 2×2 input autocorrelation matrix is given as $\mathbf{R}_{xx} = E[\mathbf{x}(n)\mathbf{x}^T(n)] = ((1/2) + \sigma_u^2)\mathbf{I}$, where $\sigma_u^2 = E[u^2(n)]$ and $\mathbf{I}$ denotes the 2×2 identity matrix. Next, define the instantaneous parameter error vector as $\boldsymbol{\Delta}(n) = \boldsymbol{\theta}(n) - \boldsymbol{\theta}$. Then, from (5), we can write

$$E[\boldsymbol{\Delta}(n+1)] = E[\boldsymbol{\Delta}(n)] + 2\boldsymbol{\mu}\boldsymbol{\Omega} E[\mathbf{U}(n)\mathbf{x}(n)e(n)]. \quad (6)$$

Note that the two vectors $\overline{\mathbf{w}}(n)$ and $\mathbf{w}'(n)$ are mutually orthogonal, i.e., $E[\overline{\mathbf{w}}^T(n)\mathbf{w}'(n)] = 0$, and each has norm unity, at each index n. We now invoke the "independence assumption" as is common with the analysis of the LMS algorithm [12] and assume $D(n)$ and $a(n)$ to be statistically independent both of $s(n)$, or, equivalently, of ϕ and, thus, of $\mathbf{s}(n) = [s(n), s(n-r)]^T$, and of $\mathbf{u}(n) = [u(n), u(n-r)]^T$. Together, this means $\boldsymbol{\theta}(n)$ is statistically independent of $\mathbf{x}(n)$. Replacing $e(n)$ in (6) by $y(n) - \mathbf{x}^T(n)\mathbf{w}(n)$, using the orthogonality between $\overline{\mathbf{w}}(n)$ and $\mathbf{w}'(n)$ and the fact that $\mathbf{R}_{xx} = (1/2 + \sigma_u^2)\mathbf{I}$, we then first observe

$$\begin{aligned} &E\left[\mathbf{U}(n)\mathbf{x}(n)\mathbf{x}^T(n)\,\mathbf{w}(n)\right] \\ &= E\left[\mathbf{U}(n)E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\}\,\mathbf{w}(n)\right] \\ &= \left(\frac{1}{2} + \sigma_u^2\right)\begin{bmatrix} 0 \\ A + E[\Delta_a(n)] \end{bmatrix} \end{aligned} \quad (7)$$

where we have used the fact that $E[a(n)] = A + E[\Delta_a(n)]$. Next, we consider the term $E[\mathbf{U}(n)\mathbf{x}(n)y(n)]$ on the right-hand side of (6). To evaluate this, we first replace $y(n)$ by $\mathbf{s}^T(n)\mathbf{w}_{\text{opt}} + z(n)$, where $\mathbf{w}_{\text{opt}} = [A\cos(\Omega D), A\sin(\Omega D)]^T$. Next, we observe that, since $D(n)$ and $a(n)$ only depend on the past samples of $z(n)$ and $z(n)$ is a zero-mean white Gaussian process (i.e., samples of $z(n)$ are independent and identically distributed) independent of the process $x(n)$, $E[\mathbf{U}(n)\mathbf{x}(n)z(n)] = E[z(n)]E[\mathbf{U}(n)\mathbf{x}(n)] = \mathbf{0}_{2\times 1}$. From this and the fact that $\mathbf{U}(n)\mathbf{w}_{\text{opt}} = A[-a(n)\sin(\Omega\Delta_D(n)), \cos(\Omega\Delta_D(n))]^T$, where $\Delta_D(n) = D(n) - D$, we then have

$$\begin{aligned} E[\mathbf{U}(n)\,\mathbf{x}(n)\,y(n)] &= E\left[\mathbf{U}(n)\,E\left\{\mathbf{x}(n)\mathbf{s}^T(n)\right\}\,\mathbf{w}_{\text{opt}}\right] \\ &= \frac{A}{2}E\begin{bmatrix} -a(n)\sin(\Omega\Delta_D(n)) \\ \cos(\Omega\Delta_D(n)) \end{bmatrix} \end{aligned}$$

since $s(n)$ and $u(n)$ are mutually independent zero-mean processes, and $E[\mathbf{s}(n)\mathbf{s}^T(n)] = (1/2)\mathbf{I}$. When $D(n)$ is sufficiently close to D, $\Omega\Delta_D(n)$ is small, and we can approximate $\sin(\Omega\Delta_D(n))$ by $\Omega\Delta_D(n)$ and $\cos(\Omega\Delta_D(n))$ by 1, resulting in

$$E[\mathbf{U}(n)\,\mathbf{x}(n)\,y(n)] \approx \begin{bmatrix} -\frac{\Omega A}{2}\,E[a(n)\Delta_D(n)] \\ \frac{A}{2} \end{bmatrix}. \quad (8)$$

Combining (6)–(8), we then have

$$\begin{aligned} &E[\boldsymbol{\Delta}(n+1)] \\ &= E[\boldsymbol{\Delta}(n)] + 2\,\boldsymbol{\mu}\,\boldsymbol{\Omega}\begin{bmatrix} -\frac{\Omega A}{2}\,E[a(n)\Delta_D(n)] \\ -A\sigma_u^2 - \left(\sigma_u^2 + \frac{1}{2}\right)\,E[\Delta_a(n)] \end{bmatrix} \end{aligned} \quad (9)$$

The above results in two recurrence relations, one for $E(\Delta_D(n))$ and the other for $E(\Delta_a(n))$. For the latter, the recurrence equation is given as

$$E(\Delta_a(n+1)) = \left[1 - 2\mu_2\left(\sigma_u^2 + \frac{1}{2}\right)\right]\,E(\Delta_a(n)) - 2\mu_2\,A\sigma_u^2. \quad (10)$$

For stability of $E(\Delta_a(n))$, i.e., to have $E(\Delta_a(n))$ bounded in the steady state, we should have $|1 - 2\mu_2(\sigma_u^2 + (1/2))| < 1$, or, equivalently, $0 < \mu_2 < 1/(\sigma_u^2 + (1/2)) \approx 2$. In such a case

$$\lim_{n\to\infty} E[\Delta_a(n)] = \frac{-2\mu_2 A\sigma_u^2}{1 - \left(1 - 2\mu_2\left(\sigma_u^2 + \frac{1}{2}\right)\right)} \approx -2A\sigma_u^2 = \epsilon(A)$$

or, equivalently, $\lim_{n\to\infty} E[a(n)] = A + \epsilon(A)$, meaning the estimate of A will be given by $(1/(1 - 2\sigma_u^2))\lim_{n\to\infty} E[a(n)]$.

In practice, the relative bias $\epsilon(A)/A$ is negligibly small (especially for high input SNRs), meaning $\lim_{n\to\infty} E[\Delta_a(n)] \approx 0$. Separately, we derive conditions in the next section for keeping the steady-state value of $E(\Delta_a^2(n))$ bounded and small. Under such a case, we can then write $E[a(n)\Delta_D(n)] = E[(A + \Delta_a(n))\Delta_D(n)] \approx E[A\Delta_D(n)]$. Substituting this in (9), we obtain

$$E(\Delta_D(n+1)) = [1 - \mu_1\Omega^2 A^2]\,E(\Delta_D(n)) \quad (11)$$

implying $\lim_{n\to\infty} E[\Delta_D(n)] = 0$ if $0 < \mu_1 < 2/\Omega^2 A^2$.

Note that, if a common step size μ is used for both $D(n)$ and $a(n)$, it would imply that $0 < \mu < 2$ and also $0 < \mu < 2/\Omega^2 A^2$, meaning the upper bound of μ will be given by the lower of the two bounds, i.e., $2/\Omega^2 A^2$. Even for frequencies in the kilohertz range, this would mean an upper bound in the range of 10^{-8} or less. Such a small value of μ, however, gives rise to some serious problems. Firstly, the update term for $a(n)$ in (5) in such cases will be negligibly small, resulting in $a(n+1) \approx a(n)$, or, equivalently, $\Delta_a(n+1) \approx \Delta_a(n)$ in (9). In other words, $a(n)$ will remain more or less static at its initial value $a(0)$ and will not converge to A. [The update term for $E[\Delta_D(n)]$ in (9), however, is free of this problem as μ is actually multiplied there by $\Omega^2 A^2$ and $0 < \mu\Omega^2 A^2 < 2$. However, as $a(n) \approx a(0)$, the update equation and the corresponding convergence condition in such a case will be given by $E(\Delta_D(n+1)) = [1 - \mu\Omega^2 A a(0)] E(\Delta_D(n))]$ and $0 < \mu\Omega^2 A a(0) < 2$, respectively—note that convergence will be very slow if $a(0) \approx 0$.] Secondly, as can easily be seen from the next section and the Appendix, the steady-state variance $E[\Delta_D(n)]^2$ in such cases will have a contribution coming from the initial error variance $[a(0) - A]^2$, which may become substantial since A is unknown. The proposed scheme avoids these problems by separating out the step size μ_2 for $a(n)$ from the step size for $D(n)$, which enables $a(n)$ to converge faster to A independent of the convergence of $D(n)$.

C. Mean Square Error Analysis

We start with the parameter error correlation matrix $\mathbf{R}_{\Delta\Delta}(n) = E[\boldsymbol{\Delta}(n)\boldsymbol{\Delta}^T(n)]$. Since our primary interest lies in $E[\Delta_D(n)]^2$ and $E[\Delta_a(n)]^2$, we will be evaluating only the diagonal elements of $\mathbf{R}_{\Delta\Delta}(n)$ and leave aside the nondiagonal entries with a "$*$" symbol. Then, from (6), we have

$$\mathbf{R}_{\Delta\Delta}(n+1) = \mathbf{R}_{\Delta\Delta}(n) + \mathbf{A}(n) + \mathbf{B}(n) + \mathbf{B}^T(n) \tag{12}$$

where $\mathbf{A}(n) = 4\boldsymbol{\mu\Omega} E[\mathbf{U}(n)\mathbf{x}(n)e(n)e(n)\,\mathbf{x}^T(n)\mathbf{U}^T(n)]\boldsymbol{\Omega\mu}$, and $\mathbf{B}(n) = 2\boldsymbol{\mu\Omega}E[\mathbf{U}(n)\mathbf{x}(n)e(n)\boldsymbol{\Delta}^T(n)]$. To evaluate $\mathbf{B}(n)$, we replace $e(n)$ by $y(n) - \mathbf{x}^T(n)\mathbf{w}(n) \equiv (\mathbf{s}^T(n)\mathbf{w}_{\rm opt} + z(n)) - \mathbf{x}^T(n)\mathbf{w}(n)$ and write $\mathbf{B}(n)$ as $\mathbf{B}(n) = \mathbf{B}_1(n) - \mathbf{B}_2(n)$, where $\mathbf{B}_1(n) = 2\boldsymbol{\mu\Omega}E[\mathbf{U}(n)\mathbf{x}(n)(\mathbf{s}^T(n)\mathbf{w}_{\rm opt} + z(n))\boldsymbol{\Delta}^T(n)]$ and $\mathbf{B}_2(n) = 2\boldsymbol{\mu\Omega}E[\mathbf{U}(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)\boldsymbol{\Delta}^T(n)]$. Proceeding as before, i.e., using the "independence assumption" and recalling that $z(n)$ is a zero-mean white Gaussian process, $E[\mathbf{x}(n)\mathbf{s}^T(n)] = (1/2)\mathbf{I}$, $\mathbf{U}(n)\mathbf{w}_{\rm opt} \approx A[-a(n)\Omega\Delta_D(n), 1]^T$, $\mathbf{U}(n)\mathbf{w}(n) = [0, a(n)]^T$, we have

$$\mathbf{B}_1(n) = A\,\boldsymbol{\mu}\,\boldsymbol{\Omega}E\begin{bmatrix} -a(n)\,\Omega\Delta_D^2(n) & * \\ * & \Delta_a(n)\end{bmatrix}$$

$$\mathbf{B}_2(n) = 2\left(\sigma_u^2 + \frac{1}{2}\right)\boldsymbol{\mu}\,\boldsymbol{\Omega}E\begin{bmatrix} 0 & * \\ * & a(n)\Delta_a(n)\end{bmatrix}. \tag{13}$$

Substituting $E[a(n)\Delta_a(n)]$ by $AE[\Delta_a(n)] + E[\Delta_a^2(n)]$, neglecting $A\sigma_u^2 E[\Delta_a(n)]$ in comparison to $(\sigma_u^2 + (1/2))E[\Delta_a^2(n)]$ (since, in the steady state, i.e., for very large n, $E[\Delta_a(n)] \approx 0$) and using the approximations $(\sigma_u^2 + (1/2))E[\Delta_a^2(n)] \approx (1/2)E[\Delta_a^2(n)]$ and $E[a(n)\Delta_D^2(n)] = E[(A + \Delta_a(n))\Delta_D^2(n)] \approx E[A\Delta_D^2(n)]$, we then have

$$\mathbf{B}(n) = 2\boldsymbol{\mu}\begin{bmatrix} -\frac{\Omega^2 A^2}{2}E\left[\Delta_D^2(n)\right] & * \\ * & -\frac{1}{2}E\left[\Delta_a^2(n)\right]\end{bmatrix}. \tag{14}$$

For $\mathbf{A}(n)$, we first make the following definitions:

1) $\alpha(n) = a(n)\mathbf{w}'^T(n)\mathbf{u}(n)$;
2) $\beta(n) = \overline{\mathbf{w}}^T(n)\mathbf{u}(n)$;
3) $\gamma(n) = a(n)\mathbf{w}'^T(n)\mathbf{x}(n) = a(n)\sin(n\pi/2r + \phi - \Omega D(n)) + \alpha(n), r \in Z$;
4) $\gamma'(n) = \overline{\mathbf{w}}^T(n)\mathbf{x}(n) = \cos(n\pi/2r + \phi - \Omega D(n)) + \beta(n), r \in Z$.

Then, we can write

$$\mathbf{A}(n) = 4\,\boldsymbol{\mu}\,\boldsymbol{\Omega}\begin{bmatrix} E\left[\gamma(n)\,e(n)\right]^2 & * \\ * & E\left[\gamma'(n)\,e(n)\right]^2\end{bmatrix}\boldsymbol{\Omega}\,\boldsymbol{\mu}. \tag{15}$$

In the Appendix, it is shown that $A_{11}(n) \approx 4\mu_1^2\Omega^2[(3/8)\Omega^2 A^4 E\{\Delta_D^2(n)\} + (A^2/8)E\{\Delta_a^2(n)\} + (\sigma_z^2 A^2/2) + (\sigma_u^2 A^4/2)]$ and $A_{22}(n) \approx 4\mu_2^2[(1/8)\Omega^2 A^2 E\{\Delta_D^2(n)\} + (3/8)E\{\Delta_a^2(n)\} + (\sigma_z^2/2) + (\sigma_u^2 A^2/2)]$. Defining $\boldsymbol{\delta}(n) = [E\{\Delta_D^2(n)\}, E\{\Delta_a^2(n)\}]^T$, it is then possible to write

$$\boldsymbol{\delta}(n+1) = \mathbf{F}\boldsymbol{\delta}(n) + \mathbf{h} \tag{16}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 - 2\mu_1\Omega^2 A^2 + \frac{3}{2}\mu_1^2\Omega^4 A^4 & \frac{1}{2}\mu_1^2\Omega^2 A^2 \\ \frac{1}{2}\mu_2^2\Omega^2 A^2 & 1 - 2\mu_2 + \frac{3}{2}\mu_2^2\end{bmatrix} \tag{17}$$

and

$$\mathbf{h} = \left(\sigma_z^2 + \sigma_u^2 A^2\right)\begin{bmatrix} 2\mu_1^2\Omega^2 A^2 \\ 2\mu_2^2\end{bmatrix}. \tag{18}$$

For stability, i.e., to maintain $\lim_{n\to\infty}\|\boldsymbol{\delta}(n)\|$ finite, the eigenvalues of $\mathbf{F}$ should have magnitude less than unity [13]. The eigenvalues of $\mathbf{F}$, which are obtained by finding the roots of the characteristic polynomial $(\lambda - F_{11})(\lambda - F_{22}) - F_{12}F_{21}$, are given as

$$\lambda_1 = \frac{(F_{11} + F_{22}) + \sqrt{(F_{11} - F_{22})^2 + 4F_{12}F_{21}}}{2}$$

$$\lambda_2 = \frac{(F_{11} + F_{22}) - \sqrt{(F_{11} - F_{22})^2 + 4F_{12}F_{21}}}{2}$$

where $F_{ij} = [\mathbf{F}]_{i,j}$, $i, j = 1, 2$. Now, it can easily be seen that each F_{ij} is a strictly positive function, and thus, $\lambda_1 = |\lambda_1| > |\lambda_2|$, meaning, for stability, it is enough to have $\lambda_1 < 1$. Substituting the value of each F_{ij} in λ_1, introducing a normalized step size $\mu_1' = \mu_1\Omega^2 A^2$ and recalling from the convergence (in mean) condition that $\mu_1, \mu_2 > 0$, the above, after some calculations, leads to the following condition:

$$f(\mu_1', \mu_2) \equiv 4 - 3\mu_1' - 3\mu_2 + 2\mu_1'\mu_2 < 0. \tag{19}$$

The function $f(\mu_1', \mu_2)$ represents a quadratic surface, as shown in Fig. 2 over the region $0 < \mu_1', \mu_2 < 2$ (i.e., the region of convergence (in mean) of the proposed algorithm). For stability, we need to find out portion of this region where the surface lies above the $\mu_1' = 0$, $\mu_2 = 0$ plane. Alternatively, the equation $f(\mu_1', \mu_2) = 0$, or, equivalently, $\mu_1' = (4 - 3\mu_2)/(3 - 2\mu_2)$ represents a hyperbola with center at (3/2, 3/2), as shown in Fig. 3. The inequality (19) then results in the following stability regions: for $\mu_2 < 3/2$ (i.e., $3 - 2\mu_2 > 0$), $\mu_1' < (4 - 3\mu_2)/(3 - 2\mu_2)$, and for $\mu_2 > 3/2$ (i.e., $3 - 2\mu_2 < 0$), $\mu_1' >$

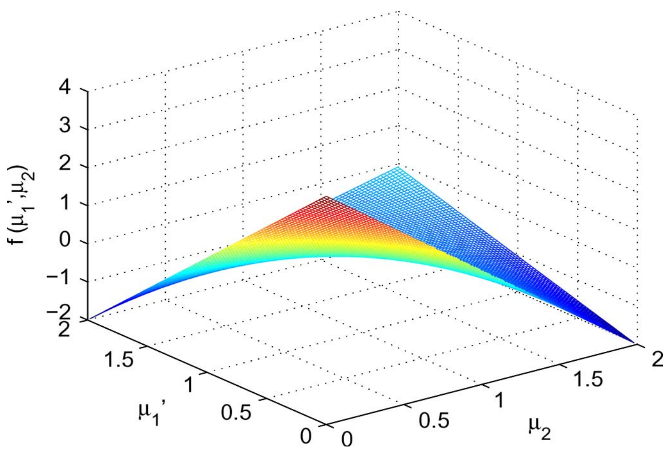

Fig. 2. Quadratic surface $f(\mu_1', \mu_2)$.

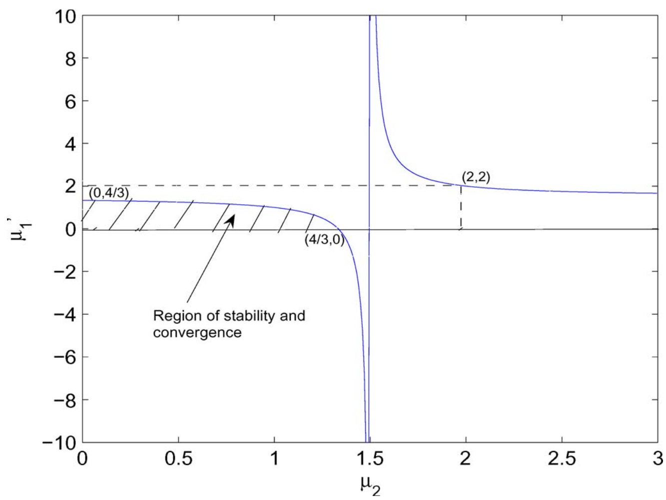

Fig. 3. Stability region in the $\mu_1'-\mu_2$ plane.

$(4-3\mu_2)/(3-2\mu_2)$. For both convergence in mean and stability of the steady-state MSE, one has to choose μ_1' and μ_2 from the intersection of the above region and the rectangle with vertices (0, 0), (0, 2), (2, 0), and (2, 2). This intersection is given by $\{\mu_1', \mu_2 | 0 < \mu_2 < (4/3), 0 < \mu_1' < (4-3\mu_2)/(3-2\mu_2)\}$ and is shown by shaded lines in Fig. 3.

The steady-state $\boldsymbol{\delta}(n)$ for a stable (16) will be given by $(\mathbf{I}-\mathbf{F})^{-1}\mathbf{h}$, which, after some calculations, is obtained as

$$\lim_{n\to\infty} \boldsymbol{\delta}(n) = K \begin{bmatrix} \mu_1(2-\mu_2) \\ \mu_2\,(2-\mu_1\Omega^2 A^2) \end{bmatrix} \equiv K \begin{bmatrix} \mu_1(2-\mu_2) \\ \mu_2\,(2-\mu_1') \end{bmatrix} \quad (20)$$

where

$$K = \frac{2(\sigma_z^2 + \sigma_u^2 A^2)}{4 - 3\mu_1' - 3\mu_2 + 2\mu_1'\mu_2} \equiv \frac{2(\sigma_z^2 + \sigma_u^2 A^2)}{f(\mu_1', \mu_2)}.$$

Note that while (19) ensures that the steady-state values of $E[\Delta_D^2(n)]$ and $E[\Delta_a^2(n)]$ remain bounded, it is important to keep these values small, particularly in view of the fact that the convergence of $E[\Delta_D(n)]$ to D, as proved in Section II-B, is based on the approximation $E[a(n)\Delta_D(n)] \approx E[A\Delta_D(n)]$. This means that μ_1' and μ_2 must not be chosen from areas where $f(\mu_1', \mu_2)$ is close to zero as that leads to division by zero in K. From Fig. 2, it then follows that, for better convergence and stability behavior, one should choose μ_1' and μ_2 from areas close to the origin.

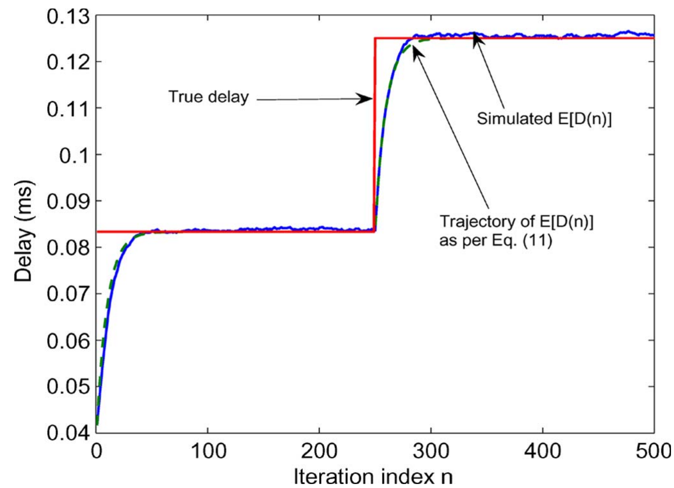

Fig. 4. Mean delay estimate $E[D(n)]$ vis-a-vis the true delay trajectory and the theoretical $E[D(n)]$ curve as per (11) (SNR of 17 dB).

III. Simulation Results

Computer simulations were carried out to evaluate the delay estimation performance of the proposed scheme. For this, a sinusoidal signal of amplitude $A = 0.8$ and frequency $\Omega = 2\pi \times 10^3$ rad $\cdot$ s^{-1} was used, which was sampled with a sampling period of $\tau = 2.5 \times 10^{-4}$ s (meaning the sampling frequency $\Omega_s = 4\ \Omega$). The powers of the two noise processes $u(n)$ and $z(n)$ were initially fixed as 0.01 and 0.0064, respectively, resulting in the same SNR of 17 dB for both the received signals $x(n)$ and $y(n)$. The step-size constants μ_1' and μ_2 were both taken to be 0.1. Observations made are based on averages of 500 independent runs of the proposed algorithm.

Fig. 4 shows the trajectory of the mean delay estimate $E[D(n)]$ for a step change in D, which was held constant at 0.0833 ms during the first 250 iterations and instantaneously changed to 0.125 ms afterward. It is seen from Fig. 4 that the observed $E[D(n)]$ estimates and tracks the piecewise constant time delay accurately, taking about 50 iterations for the given choice of μ_1' and μ_2 to converge to the true delay value. It is also easily seen that the simulated plot of $E[D(n)]$ for the proposed method conforms very well to the dynamics described by (11). A representative plot of the instantaneous delay estimate $D(n)$ is also shown in Fig. 5, which also confirms satisfactory estimation and tracking performance. Similarly, for the aforementioned step variation in D and a constant $A = 0.8$, the mean amplitude estimate $E[a(n)]$ was simulated and is shown in Fig. 6, where it is seen to agree very well with the theoretical calculation of (10). The steady-state value of $E[\Delta_a(n)]$ for this case is seen to lie at around -0.0157, which closely matches the theoretical value of $\epsilon(A) = -2A\sigma_u^2$ $(= -0.016)$. The bias eliminated estimate of A, namely, $(1/(1-2\sigma_u^2))E[a(n)]$, is also plotted in Fig. 6, which is seen to converge on the true A after about 50 iterations. Very good convergence behavior is also noticed for the instantaneous estimate $a(n)$, which,

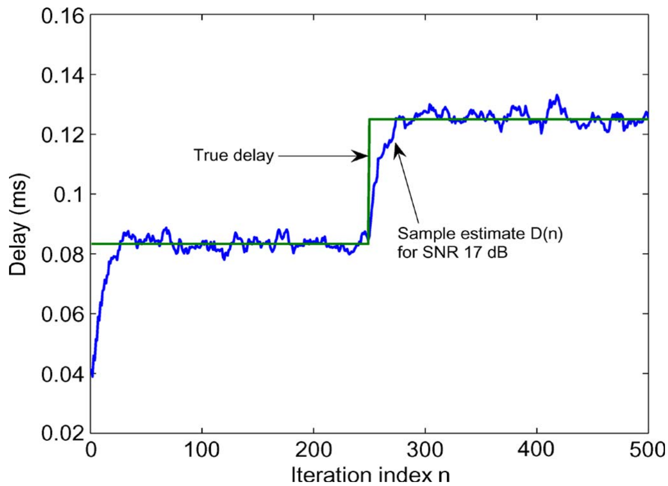

Fig. 5. Delay estimate $D(n)$ based on a single run of the experiment vis-a-vis the true delay (SNR of 17 dB).

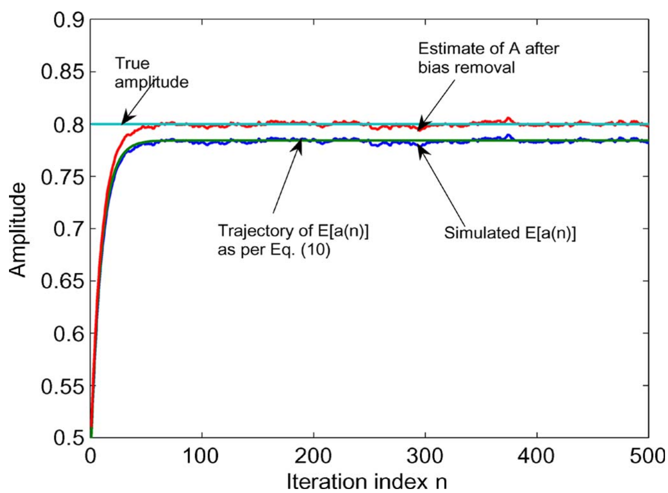

Fig. 6. Mean amplitude estimate $E[a(n)]$ vis-a-vis the true amplitude A and the theoretical $E[a(n)]$ curve as per (10); mean amplitude estimate after bias correction (SNR of 17 dB).

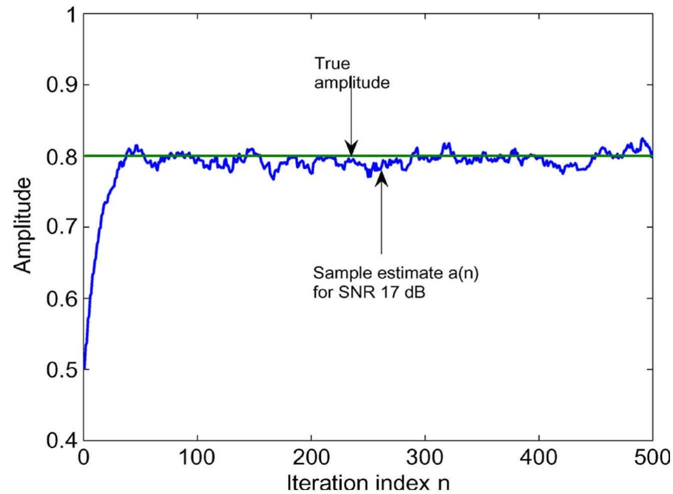

Fig. 7. Amplitude estimate $a(n)$ (after bias removal) based on a single run of the experiment vis-a-vis the true amplitude A (SNR of 17 dB).

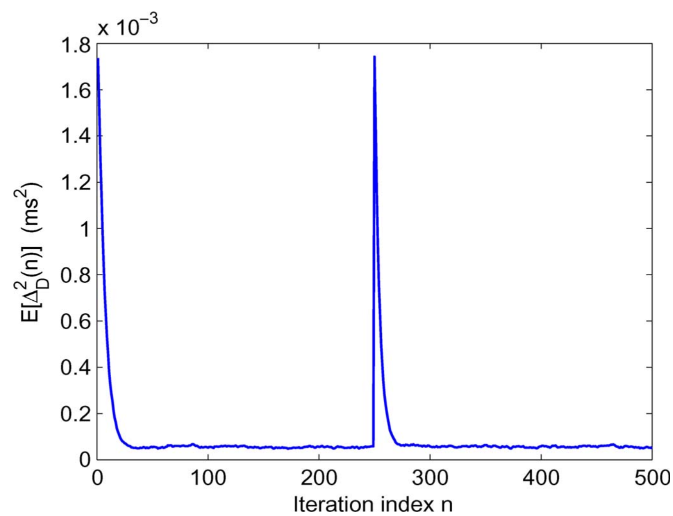

Fig. 8. Mean square delay estimation error $E[\Delta_D^2(n)]$ (SNR of 17 dB).

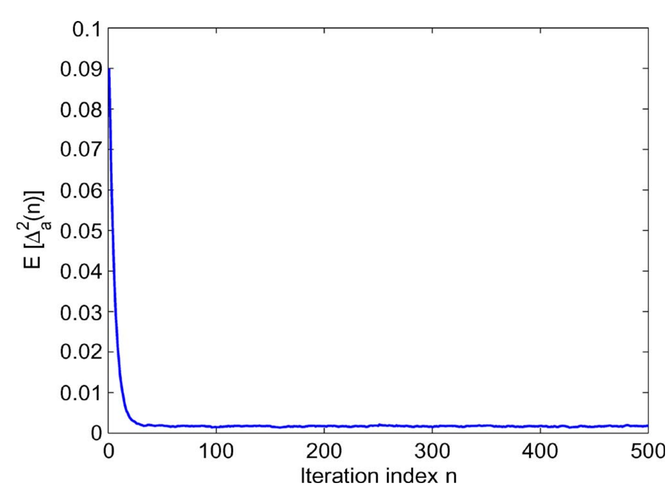

Fig. 9. Mean square amplitude estimation error $E[\Delta_a^2(n)]$ (SNR of 17 dB).

after bias correction, is displayed against the true amplitude in Fig. 7. Next, we measured the mean square delay and amplitude estimation errors, i.e., $E[\Delta_D^2(n)]$ and $E[\Delta_a^2(n)]$, respectively, and the results are plotted in Figs. 8 and 9, respectively. It is seen that both the estimation errors $E[\Delta_D^2(n)]$ and $E[\Delta_a^2(n)]$ rapidly decay to their respective steady-state values. Furthermore, the steady-state values of $E[\Delta_D^2(n)]$ at $D = 0.0833$ ms and $D = 1.25$ ms were measured as 0.054×10^{-3} ms^2 and 0.058×10^{-3} ms^2, respectively, which are close to the theoretical value given by (20), namely, 0.056×10^{-3} ms^2. Again, the steady-state value of $E[\Delta_a^2(n)]$ was measured as 0.0017, which favorably compares with its theoretical value of 0.0014 as per (20).

We next conducted the aforementioned experiment by varying the SNR over a wide range. For this, as before, we kept both the input and output SNRs to be the same. Note that, with the increase in noise power, the factor K in (20) increases, meaning both $E\{\Delta_D^2(n)\}$ and $E\{\Delta_a^2(n)\}$ become larger. This, in turn, affects the dynamics of $E[\Delta_a(n)]$ and $E[\Delta_D(n)]$, causing them to slowly deviate from (10) and (11), respectively, and thus affecting the convergence (note that (10) and (11) have been derived assuming relatively small magnitudes for both $\Delta_a(n)$ and $\Delta_D(n)$). In the experiments conducted, however, no major

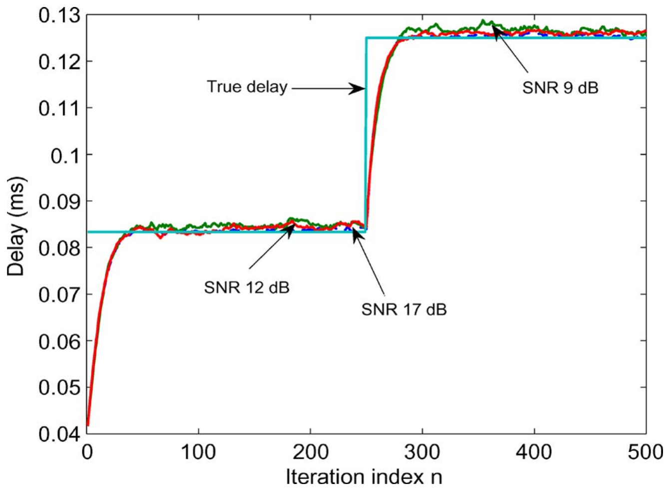

Fig. 10. Mean delay estimate $E[D(n)]$ for SNRs of 17, 12, and 9 dB.

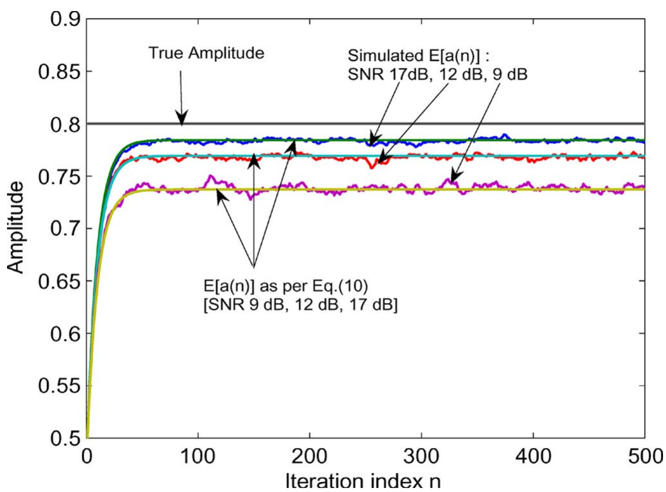

Fig. 11. Mean amplitude estimate $E[a(n)]$ (without bias correction) for SNRs of 17, 12, and 9 dB.

deterioration in convergence behavior was noticed, except for a very low range of the SNR (typically 0–6 dB). In Fig. 10, we show the convergence behavior of the mean delay estimate $E[D(n)]$ for three SNR figures, namely, 17 dB (shown by the dashed line), 12 dB, and 9 dB. The same is shown for the mean amplitude estimate $E[a(n)]$ (without bias correction) in Fig. 11, where we have also plotted the respective trajectories of $E[a(n)]$ as given by (10) (note that the bias term $-2A\sigma_u^2$, different for the three SNR figures, clearly shows up in the respective plots of $E[a(n)]$ in Fig. 11). Both Figs. 10 and 11 confirm that the estimation performance of the proposed algorithm deteriorates as the SNR is brought down from 17 to 9 dB. However, even at 9 dB, the extent of deterioration is not significant and, on the contrary, is very much within the acceptable range.

IV. Discussions and Conclusion

A new adaptive filter is presented, which considers two noisy sinusoids of the same frequency received at two spatially separated sensors and directly estimates the delay and the relative amplitude of one of them taking the other as reference. The algorithm uses the sampling frequency from a specified set that results in a two-tap FIR filter model for the delayed signal. The delay and the amplitude are estimated by identifying the FIR filter, for which a delay variable and an amplitude variable are time updated in an LMS-like manner. Trajectories for the mean delay estimate and mean amplitude estimate are derived, and convergence conditions are established. To ensure convergence, the algorithm needs to employ separate step sizes for the delay and the amplitude update processes. A detailed stability analysis is carried out, and stability regions in the step-size plane are determined that guarantee bounded steady-state MSEs. The proposed algorithm is computationally very simple and is well suited for implementation on CORDIC processors. MATLAB simulations showed satisfactory estimation performance of the proposed method, both in mean and mean square.

Appendix
Evaluation of the Matrix $\mathbf{A}(n)$

First, by expressing $y(n)$ as $y(n) = A\cos((n\pi/2r) + \phi - \Omega D) + z(n)$, $e(n)$ as $e(n) = y(n) - a(n)\gamma'(n)$, and $a(n)\gamma'(n)$ as $a(n)\gamma'(n) = (A + \Delta_a(n))\cos((n\pi/2r) + \phi - \Omega D(n)) + a(n)\beta(n)$, and as before, using the approximation $\sin(\Omega\Delta_D(n)/2) \approx \Omega\Delta_D(n)/2$ for small $\Omega\Delta_D(n)$, we write

$$e(n) \approx z(n) - \Omega\, A\, \Delta_D(n)\, \sin\left(\frac{n\pi}{2r} + \phi - \frac{\Omega\,(D + D(n))}{2}\right) - \Delta_a(n)\cos\left(\frac{n\pi}{2r} + \phi - \Omega\, D(n)\right) - a(n)\beta(n). \quad \text{(A.1)}$$

To evaluate $\mathbf{A}(n)$, we substitute $e(n)$ by the above approximation in $E[\gamma(n)e(n)]^2$ and $E[\gamma'(n)e(n)]^2$ and evaluate the expectation on a term-by-term basis. For both cases, the cross terms involving $z(n)$ take zero value since $z(n)$ is a zero-mean white Gaussian process independent of ϕ and $u(n)$ and also of $D(n)$, as explained in Section II-B. The remaining terms in $E[\gamma(n)e(n)]^2$ and $E[\gamma'(n)e(n)]^2$ are evaluated here.

1) $E[\gamma(n)e(n)]^2$. This consists of the following terms:

A1) $E[z^2(n)\gamma^2(n)]$: Since $z(n)$ is independent of ϕ, $u(n)$, $a(n)$, and $D(n)$, $E[z^2(n)\gamma^2(n)] = \sigma_z^2 E[\gamma^2(n)]$, where $\sigma_z^2 = E[z^2(n)]$. To evaluate $E[\gamma^2(n)]$, we substitute $\gamma(n)$ by $a(n)\sin((n\pi/2r) + \phi - \Omega D(n)) + \alpha(n)$, introduce E_ϕ as the expectation operator w.r.t. ϕ, and make the following general observations: for any Ψ independent of ϕ and for any nonzero integer k, $E_\phi[\sin(k\phi + \Psi)] = E_\phi[\cos(k\phi + \Psi)] = 0$, meaning $E_\phi[\sin^2(k\phi + \Psi)] = (1/2)E_\phi[1 - \cos(2k\phi + 2\Psi)] = 1/2$, and similarly, $E_\phi[\cos^2(k\phi + \Psi)] = 1/2$. From this and using the independence of $D(n)$ and $a(n)$ w.r.t. ϕ and $u(n)$ as per the "independence assumption," $E[\gamma^2(n)]$ is then seen to consist of the following terms: 1) $E[a^2(n)\sin^2((n\pi/2r) + \phi - \Omega D(n))] = E\,[a^2(n)\; E_\phi[\sin^2((n\pi/2r) + \phi - \Omega D(n))]] = (1/2)E[a^2(n)]$; 2) $E\,[2a(n)\sin((n\pi/2r) + \phi - \Omega D(n))\alpha(n)] = E\,[2a\,(n)\,E_\phi[\sin((n\pi\,/\,2r) + \phi - \Omega D(n))]\alpha(n)] = 0$; and 3) $E[\alpha^2(n)] = E[a^2(n)\mathbf{w}'^T(n)E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{w}'(n)] = \sigma_u^2 E[a^2(n)]$. Thus, $E[z^2(n)\gamma^2(n)] = \sigma_z^2(1/2 + \sigma_u^2)E[a^2(n)]$.

Neglecting the fourth-order term $\sigma_u^2\sigma_z^2$, which is very small for high SNRs, we have $E[z^2(n)\gamma^2(n)] \approx (\sigma_z^2/2)E[a^2(n)] \approx (\sigma_z^2 A^2/2) + (\sigma_z^2/2)E[\Delta_a^2(n)]$, where we have neglected the term involving $E[\Delta_a(n)]$, which becomes very small for large n.

B1) $E[a^2(n)\beta^2(n)\gamma^2(n)]$: As before, using the independence of $\mathbf{u}(n)$ w.r.t. ϕ, $a(n)$, and $D(n)$, we observe the following: 1) $E[a^4(n)\sin^2((n\pi/2r) + \phi - \Omega D(n))\beta^2(n)] = E[a^4(n)\sin^2((n\pi/2r) + \phi - \Omega D(n))\overline{\mathbf{w}}^T(n)\, E\{\mathbf{u}(n)\mathbf{u}^T(n)\}\, \overline{\mathbf{w}}(n)] = \sigma_u^2\, E[a^4(n) E_\phi[\sin^2((n\pi/2r) + \phi - \Omega D(n))]] = (\sigma_u^2/2)E[a^4(n)] \approx \sigma_u^2 A^4/2 + 3\sigma_u^2 A^2 E[\Delta_a^2(n)]$, and 2) any cross term involving the first power of $\sin((n\pi/2r) + \phi - \Omega D(n))$ results in zero, since $E_\phi[\sin((n\pi/2r) + \phi - \Omega D(n))] = 0$. The third term, i.e., $E[a^2(n)\alpha^2(n)\beta^2(n)]$, on simplification, results in $E[a^4(n)[(-(1/2)\sin(2\Omega D(n))u^2(n) + (1/2)\sin(2\Omega D(n))u^2(n-1) + \cos(2\Omega D(n))u(n)u(n-1))]^2]$. Since $u(n)$ is a zero-mean white Gaussian process, we have $E[u^3(n)] = 0$ and $E[u^4(n)] = 3\sigma_u^4$. From this and using the independence of $\mathbf{u}(n)$ with $D(n)$ and $a(n)$, we observe that $E[a^2(n)\alpha^2(n)\beta^2(n)]$ consists only of terms involving σ_u^4, which is again negligibly small for high SNR conditions. Thus, $E[a^2(n)\beta^2(n)\gamma^2(n)] \approx (\sigma_u^2 A^4/2) + 3\sigma_u^2 A^2 E[\Delta_a^2(n)]$.

C1) $E[\Omega^2 A^2 \Delta_D^2(n)\sin^2((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\gamma^2(n)]$: Substituting the value of $\gamma(n)$, we observe that this has three terms: 1) a cross term involving the first power of $\alpha(n)$, which becomes zero after expectation since $\mathbf{u}(n)$ is zero mean and is independent of ϕ, $a(n)$, and $D(n)$; 2) $E[a^2(n)\Omega^2 A^2\Delta_D^2(n)\sin^2((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\mathbf{w}'^T(n) \quad E[\mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{w}'(n)] = \Omega^2 A^2\sigma_u^2 E[a^2(n)\Delta_D^2(n)/2\{1 - \cos((n\pi/r) + 2\phi - \Omega(D(n) + D)\}] = (1/2)\Omega^2 A^2\sigma_u^2 E[a^2(n)\Delta_D^2(n)] \approx (1/2)\Omega^2 A^4 \sigma_u^2 E[\Delta_D^2(n)]$; and 3) $\Omega^2 A^2 E[a^2(n)\Delta_D^2(n)\sin^2((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\sin^2((n\pi/2r) + \phi - \Omega D(n))] = (\Omega^2 A^2/4)E[a^2(n)\Delta_D^2(n)E_\phi[\{1 - \cos((n\pi/r) + 2\phi - \Omega(D + D(n)))\}\{1 - \cos((n\pi/r) + 2\phi - 2\Omega D(n))\}]] = (3/8)\Omega^2 A^2 E[a^2(n)\Delta_D^2(n)] \approx (3/8)\Omega^2 A^4 E[\Delta_D^2(n)]$, since $E_\phi[\cos((n\pi/r) + 2\phi - \Omega(D + D(n)))\cos((n\pi/r) + 2\phi - 2\Omega D(n))] = (1/2) E_\phi[\cos((2n\pi/r) + 4\phi - \Omega(D + D(n)) - 2\Omega D(n)) + \cos(\Omega\Delta_D(n))] \approx 1/2$, where we use the approximation $\cos(\Omega\Delta_D(n)) \approx 1$ for small values of $\Omega\Delta_D(n)$. Combining, $E[\Omega^2 A^2\Delta_D^2(n)\sin^2((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\gamma^2(n)] \approx ((1/2)\sigma_u^2 + (3/8))\Omega^2 A^4 E[\Delta_D^2(n)] \approx (3/8)\Omega^2 A^4 E[\Delta_D^2(n)]$ for high input SNR conditions.

D1) $E[\Delta_a^2(n)\cos^2((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)]$: This consists of three terms, out of which the cross term involving the first power of $\alpha(n)$ results in zero for reasons previously explained. The other two terms are as follows: 1) $E[a^2(n)\sin^2((n\pi/2r) + \phi - \Omega D(n))\Delta_a^2(n)\cos^2((n\pi/2r) + \phi - \Omega D(n))] = (1/4)E[a^2(n)\Delta_a^2(n)\sin^2((n\pi/r) + 2\phi - 2\Omega D(n))] \approx (A^2/8)E[\Delta_a^2(n)]$ and 2) $E[\Delta_a^2(n)\cos^2((n\pi/2r) + \phi - \Omega D(n))\alpha^2(n)] = \sigma_u^2 E[\Delta_a^2(n)a^2(n)\cos^2((n\pi/2r) + \phi - \Omega D(n))] \approx (\sigma_u^2 A^2/2)E[\Delta_a^2(n)]$. Combining, $E[\Delta_a^2(n)\cos^2((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)] \approx [\sigma_u^2 + (1/4)](A^2/2)E[\Delta_a^2(n)]$.

E1) $E[2\Omega A\Delta_D(n)\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))a(n)\beta(n)\gamma^2(n)]$: Again, substituting the value of $\gamma(n)$ and using the "independence assumption," we observe that this has the following three terms: 1) a cross term involving the first power of $\beta(n)$ but free of $\alpha(n)$, which is zero, since $E[\beta(n)] = 0$; 2) a cross term involving $\alpha(n)\beta(n)$, which is also zero, since $E[\alpha(n)\beta(n)] = E[a(n)\mathbf{w}'^T(n)E[\mathbf{u}(n)\mathbf{u}^T(n)]\overline{\mathbf{w}}(n)] = 0$ (since $\mathbf{w}'(n)$ and $\overline{\mathbf{w}}(n)$ are mutually orthogonal); and 3) $E[2\Omega A\Delta_D(n)\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))a(n)\beta(n)\alpha^2(n)] = 0$, since $E_\phi[\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))] = 0$. Thus, $E[2\Omega A\Delta_D(n)\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))a(n)\beta(n)\gamma^2(n)] = 0$.

F1) $E[2a(n)\beta(n)\Delta_a(n)\cos((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)]$: Expanding $\gamma^2(n)$, we find that this has three terms: 1) a cross term involving the first power of $\beta(n)$ but free of $\alpha(n)$, which is zero, as explained in E1 above; 2) a cross term involving $\alpha(n)\beta(n)$, which is also zero, again as explained in E1; and 3) $E[2a(n)\beta(n)\Delta_a(n)\alpha^2(n)\cos((n\pi/2r) + \phi - \Omega D(n))] = 0$, since $E_\phi[\cos((n\pi/2r) + \phi - \Omega D(n))] = 0$. Thus, $E[2a(n)\beta(n)\Delta_a(n)\cos((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)] = 0$.

G1) $E[2\Omega A\Delta_D(n)\Delta_a(n)\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\cos((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)]$: As before, substituting the value of $\gamma(n)$, this is seen to consist of three terms: 1) a term involving the first power of $\alpha(n)$, which results in zero, as explained earlier; 2) a term involving $\alpha^2(n) \equiv a^2(n)\mathbf{w}'^T(n)\mathbf{u}(n)\mathbf{u}^T(n)\mathbf{w}'(n)$, which leads to $\sigma_u^2 a^2(n)$ after applying expectation w.r.t. $\mathbf{u}(n)$. Noting that $E_\phi[2\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\cos((n\pi/2r) + \phi - \Omega D(n))] \approx \Omega\Delta_D(n)/2$ and replacing $a(n)$ by $(A + \Delta_a(n))$, this term can then be approximated as $(1/2)\Omega^2\sigma_u^2 A^3 E[\Delta_D^2(n)\Delta_a(n)]$. 3) To evaluate the third term, we first observe that $E_\phi[2\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\cos((n\pi/2r) + \phi - \Omega D(n))\sin^2((n\pi/2r) + \phi - \Omega D(n))] \approx \Omega\Delta_D(n)/8$ under the following approximation: $\cos(\Omega\Delta_D(n)/2) \approx 1$. This term then approximates to $(1/8)\Omega^2 A^3 E[\Delta_D^2(n)\Delta_a(n)]$, and thus, combining, $E[2\Omega A\Delta_D(n)\sin((n\pi/2r) + \phi - (\Omega(D + D(n))/2))\cos((n\pi/2r) + \phi - \Omega D(n))\gamma^2(n)] \approx 1/2\Omega^2 A^3(1/4 + \sigma_u^2)E[\Delta_D^2(n)\Delta_a(n)]$.

Combining the results as obtained under A1–G1 and with some straightforward approximations, we obtain $A_{11}(n) \approx 4\mu_1^2\Omega^2[(3/8)\Omega^2 A^4 E\{\Delta_D^2(n)\} + (A^2/8)E\{\Delta_a^2(n)\} + (\sigma_z^2 A^2/2) + (\sigma_u^2 A^4/2)]$.

2) $E[\gamma'(n)e(n)]^2$. This consists of the following terms:

A2) $E[z^2(n)\gamma'^2(n)]$: From A1, it is easy to check that $E[z^2(n)\gamma'^2(n)] = ((1/2) + \sigma_u^2)\sigma_z^2 \approx (1/2)\sigma_z^2$.

B2) $E[a^2(n)\beta^2(n)\gamma'^2(n)]$: From B1, $E[a^2(n)\beta^2(n)\gamma'^2(n)] \approx (\sigma_u^2/2)E[a^2(n)] \approx (\sigma_u^2 A^2/2) + (\sigma_u^2/2)E[\Delta_a^2(n)]$.

C2) $E[\Omega^2 A^2 \Delta_D^2(n)\sin^2((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\gamma'^2(n)]$: Again, following the steps in C1, it is easy to check that $E[\Omega^2 A^2 \Delta_D^2(n)\sin^2((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\gamma'^2(n)] \approx (1/2)\Omega^2 A^2[\sigma_u^2 + (1/4)]E[\Delta_D^2(n)] \approx (1/8)\Omega^2 A^2 E[\Delta_D^2(n)]$.

D2) $E[\Delta_a^2(n)\cos^2((n\pi/2r)+\phi-\Omega D(n))\gamma'^2(n)]$: As in D1, this consists of three terms: 1) a cross term involving the first power of $\beta(n)$, which is zero; 2) $E[\Delta_a^2(n)\cos^4((n\pi/2r)+\phi-\Omega D(n))] = (1/4)E[\Delta_a^2(n)\{1+\cos((n\pi/r)+2\phi-2\Omega D(n))\}^2] = (3/8)E[\Delta_a^2(n)]$; and 3) $E[\Delta_a^2(n)\cos^2((n\pi/2r)+\phi-\Omega D(n))\beta^2(n)] = (\sigma_u^2/2)E[\Delta_a^2(n)]$. Combining, $E[\Delta_a^2(n)\cos^2((n\pi/2r)+\phi-\Omega D(n))\gamma'^2(n)] = ((\sigma_u^2/2)+(3/8))E[\Delta_a^2(n)] \approx (3/8)E[\Delta_a^2(n)]$.

E2) $E[2\Omega A\Delta_D(n)\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))a(n)\beta(n)\gamma'^2(n)]$: This consists of three terms—one involving the first power of $\beta(n)$, which is zero, and the other involving $\beta^3(n)$, which is also zero, since $E_\phi[\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))] = 0$. For the third term, we replace $E[\beta^2(n)]$ by σ_u^2 as before and note that $E_\phi[4\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\cos((n\pi/2r)+\phi-\Omega D(n))] \approx \Omega\Delta_D(n)$. This leads to $E[2\Omega A\Delta_D(n)\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))a(n)\beta(n)\gamma'^2(n)] \approx \sigma_u^2\Omega^2 AE[a(n)\Delta_D^2(n)] \approx \sigma_u^2\Omega^2 A^2 E[\Delta_D^2(n)]$.

F2) $E[2a(n)\beta(n)\Delta_a(n)\cos((n\pi/2r)+\phi-\Omega D(n))\gamma'^2(n)]$: Following the lines of E2, it is easy to verify that the above is equal to $2\sigma_u^2 E[a(n)\Delta_a(n)] = 2\sigma_u^2 E[(A+\Delta_a(n))\Delta_a(n)] \approx 2\sigma_u^2 E[\Delta_a^2(n)]$, since, for large n, $E[\Delta_a(n) \approx 0$.

G2) $E[2\Omega A\Delta_D(n)\Delta_a(n)\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\cos(n\pi/2r+\phi-\Omega D(n))\gamma'^2(n)]$: As before, after substituting the value of $\gamma'(n)$, we observe that the cross term involving the first power of $\beta(n)$ is zero and the other one involving $\beta^2(n)$ approximates to $(1/2)\Omega^2 A\sigma_u^2 E[\Delta_D^2(n)\Delta_a(n)]$. The third term involves $E_\phi[\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\cos^3((n\pi/2r)+\phi-\Omega D(n))]$. Following the procedure adopted above, this gets approximated to $(3/16)\Omega\Delta_D(n)$, meaning the third term is approximately equal to $(3/8)\Omega^2 AE[\Delta_D^2(n)\Delta_a(n)]$. Combining, $E[2\Omega A\Delta_D(n)\Delta_a(n)\sin((n\pi/2r)+\phi-(\Omega(D+D(n))/2))\cos((n\pi/2r)+\phi-\Omega D(n))\gamma'^2(n)] \approx ((\sigma_u^2/2)+(3/8))\Omega^2 AE[\Delta_D^2(n)\Delta_a(n)]$.

Combining the results as obtained under A2–G2 and with some straightforward approximations, we obtain $A_{22}(n) \approx 4\mu_2^2[(1/8)\Omega^2 A^2 E\{\Delta_D^2(n)\} + (3/8)E\{\Delta_a^2(n)\} + (\sigma_z^2/2) + (\sigma_u^2 A^2/2)]$.

Acknowledgment

The author would like to thank Prof. H. C. So, Department of Electronic Engineering, City University of Hong Kong, for some useful suggestions in the context of this work.

References

[1] G. C. Carter, *Coherence and Time Delay Estimation: An Applied Tutorial for Research, Development, Test, and Evaluation Engineers*. Piscataway, NJ: IEEE Press, 1993.
[2] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.
[3] H. C. So, "A comparative study of two discrete-time phase delay estimators," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2501–2504, Dec. 2005.
[4] D. L. Maskell and G. S. Woods, "The discrete-time quadrature sub-sample estimation of delay," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 1, pp. 133–137, Feb. 2002.
[5] A. K. Nandi, "On the subsample time-delay estimation of narrow-band ultrasonic echoes," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 42, no. 11, pp. 993–1001, Nov. 1995.
[6] R. A. Reed, P. L. Feintuch, and N. J. Bershad, "Time delay estimation using the LMS adaptive filter—Static behaviour," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 3, pp. 561–571, Jun. 1981.
[7] H. C. So, P. C. Ching, and Y. T. Chan, "A new algorithm for explicit adaptation of time delay," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1816–1820, Jul. 1994.
[8] S. R. Dooley and A. K. Nandi, "Adaptive subsample time delay estimation using Lagrange interpolators," *IEEE Signal Process. Lett.*, vol. 6, no. 3, pp. 65–67, Mar. 1999.
[9] Z. Cheng and T. T. Tjhung, "A new time delay estimator based on ETDE," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1859–1869, Jul. 2003.
[10] D. L. Maskell and G. S. Woods, "Adaptive subsample delay estimation using a modified quadrature phase detector," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 10, pp. 669–674, Oct. 2005.
[11] M. Chakraborty, H. C. So, and J. Zheng, "A new adaptive algorithm for delay estimation of sinusoidal signals," *IEEE Signal Process. Lett.*, vol. 14, no. 12, pp. 984–987, Dec. 2007.
[12] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[14] Y. H. Hu, "CORDIC-based VLSI architecture for digital signal processing," *IEEE Signal Process. Mag.*, vol. 9, no. 3, pp. 16–35, Jul. 1992.
[15] M. Chakraborty, A. S. Dhar, and M. H. Lee, "A trigonometric formulation of the LMS algorithm for realization on pipelined CORDIC," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 9, pp. 530–534, Sep. 2005.

Mrityunjoy Chakraborty (M'94–SM'99) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 1983 and the M.Tech. and Ph.D. degrees in electrical engineering from the Indian Institute of Technology (IIT) at Kanpur and New Delhi, in 1985 and 1994, respectively

He joined IIT, Kharagpur, as a Faculty Member in 1994, where he currently holds the position of a Professor of electronics and electrical communication engineering. His teaching and research interests are in digital and adaptive signal processing, including algorithm, architecture and implementation, VLSI signal processing, and DSP applications in wireless communications. In these areas, he has supervised several graduate theses, has carried out independent research, and has several well-cited publications.

Prof. Chakraborty is a Fellow of the Indian National Academy of Engineers. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II during 2008–2009 and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I during 2006–2007 and 2004–2005. He has also been a Guest Editor for a recent Special Issue on Distributed Space–Time Systems of the *EURASIP Journal on Advances in Signal Processing*, a Founding Member of the Asia Pacific Signal and Information Processing Association, and has been in the technical committee of many top-ranking international conferences.