On the Convergence of the CORDIC Adaptive Lattice Filtering (CALF) Algorithm

Yu Hen Hu, *Senior Member, IEEE*

***Abstract*—In this paper, the convergence of a recently proposed CORDIC adaptive lattice filtering (CALF) algorithm is proved. It is shown that the update of the rotation angle (which is equivalent to the reflection coefficient) can be modeled by the state transition of a regular Markov chain, with each rotation angle being a state. The convergence of the CALF algorithm then is established as this Markov chain converges from an initial state probability distribution to its limiting state probability distribution. Formulae that enable explicit calculation of the limiting state distribution are derived. Moreover, it is shown that the algorithm has an exponential convergence rate.**

***Index Terms*—Adaptive filters, CALF, convergence, CORDIC, lattice filters.**

I. Introduction

THE CORDIC adaptive lattice filter (CALF) [11] is an efficient implementation of the stochastic gradient based adaptive lattice filter algorithm using a special arithmetic unit called CORDIC [12], [20], [21]. A distinct feature of CALF is that it requires only a simple up/down binary counter per lattice stage to realize the stochastic gradient (SG)-based adaptation algorithm. This compares very favorably, in terms of cost and ease of design, to the microprocessor-based architecture used in conventional implementation of adaptive lattice algorithms [4], [8]–[10], [14], [18]. The simplicity and regularity of the CALF architecture makes it an excellent candidate for many adaptive lattice filter applications such as echo cancellation or channel equalization.

In an earlier empirical study [11], we observed that the CALF algorithm demonstrates good convergence properties that are roughly comparable with those employing more complicated adaptation schemes. However, no theoretical results were developed to prove that the CALF algorithm will indeed converge. The main objective of this paper is to develop a novel approach to prove the convergence of the CALF algorithm theoretically. By taking advantage of the fact that the parameters (rotation angles) in the CALF algorithm may take only a finite number of distinct values, we established the convergence of the CALF algorithm as the convergence of a regular Markov chain toward its limiting state probability distribution. This convergence definition allows us to establish the asymptotic convergence rate of the CALF algorithm. Moreover, the method we develop is very general and can be applied to other gradient-based adaptation algorithms with large step size and a finite number of different parameter values.

In the rest of this paper, some basic notations and definitions will be reviewed in Section II. The CALF algorithm will be derived in Section III. The convergence proof and discussion of convergence rate is derived in Section IV.

Manuscript received October 15, 1995; revised January 12, 1998. The associate editor coordinating the review of this paper and approving it for publication was Dr. Akihiko Sugiyama.

The author is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706-1691 USA.

Publisher Item Identifier S 1053-587X(98)04421-3.

II. Preliminaries

A. The CORDIC Algorithm

The coordinate rotation digital computer (CORDIC) is an iterative arithmetic algorithm capable of efficient evaluation of elementary functions. With the CORDIC algorithm, all arithmetic operations are evaluated via the rotation of a 2×1 real vector in one of three possible coordinate systems: linear, circular, or hyperbolic. The rotation angle θ is decomposed into a sequence of *elementary rotation angles*. Each elementary rotation angle is chosen such that it can be evaluated using simple shift-and-add method. Furthermore, these elementary operations can be realized with an array of pipelined CORDIC processors [1], [2], [5], [7], [12], [13], [15], [16]. Thus, for applications that demand high throughput rate, such as video signal processing, the CORDIC processor implementation of lattice filter sections would be desirable.

B. CORDIC Implementation of Lattice Filters

A lattice filter is formed by cascading successive basic (normalized) lattice sections (cf., Fig. 1), which can be described by[1]

$$\begin{bmatrix} f_p(t) \\ b_p(t) \end{bmatrix} = \frac{1}{\sqrt{1-\kappa_p^2}} \begin{bmatrix} 1 & -\kappa_p \\ -\kappa_p & 1 \end{bmatrix} \begin{bmatrix} f_{p-1}(t) \\ b_{p-1}(t-1) \end{bmatrix} = \begin{bmatrix} \cosh\theta_p & -\sinh\theta_p \\ -\sinh\theta_p & \cosh\theta_p \end{bmatrix} \begin{bmatrix} f_{p-1}(t) \\ b_{p-1}(t-1) \end{bmatrix} \quad (1)$$

where $f_p(t)$ and $b_p(t)$ are called *forward* and *backward* prediction error of the pth lattice section at time instant t, and[2] $\kappa_p(t) = \tanh\theta_p(t)$. Obviously, $|\kappa_p(t)| < 1$.

The operation performed in each normalized lattice section corresponds to a *hyperbolic rotation* of the 2×1 vector

[1] Although only normalized lattice section formulation is discussed in this paper, the un-normalized version result can be derived using a similar approach.

[2] $\kappa_p(t)$ is *not* the same as the *reflection coefficient* of an *un-normalized* lattice filter, which is often denoted by $k_p(t)$.

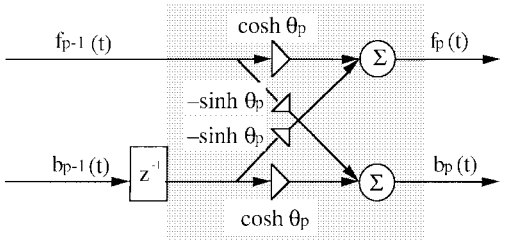

Fig. 1. pth stage of normalized lattice filter.

$[f_{p-1}(t)\ \ b_{p-1}(t-1)]^T$. Thus, if the rotation angle θ_p is represented by

$$\sum_{i=0}^{n-1} \nu_i 2^{-i-1} \tag{2}$$

with $\nu_i = \pm 1$, $0 \le i \le n-1$, the hyperbolic rotation in (1) can be computed using the following iterative formulation.

Initiation: Given $x(0) = f_{p-1}(t)$, $y(0) = b_{p-1}(t-1)$, and $\{\nu_i; 0 \le i \le n-1\}$.

For $i = 0$ **to** $n-1$ **Do** /* CORDIC iterations */

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\nu_i 2^{-i-1} \\ -\nu_i 2^{-i-1} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \tag{3}$$

End /* For-i Loop */.

In general, to obtain $f_p(t)$ and $b_p(t)$, we must multiply both $x(n)$ and $y(n)$ by a scaling factor to compensate the change of magnitude of $f_p(t)$ and $b_p(t)$ during the CORDIC iterations. However, applying the same scaling factor on both f_p and b_p will not affect the computation of $\kappa_p(t)$ or $\theta_p(t)$, which is desired. Thus, the scaling operation can be skipped.

C. SG Adaptive Lattice Algorithms

Stochastic gradient lattice algorithms are derived by minimizing an instantaneous prediction error

$$J_p(t) = f_p^2(t) + b_p^2(t). \tag{4}$$

The reflection coefficient is updated using a stochastic gradient formula

$$\kappa_p(t+1) = \kappa_p(t) - \mu \nabla_{\kappa_p} J_p(t). \tag{5}$$

After simplification, we have

$$\nabla_{\kappa_p} J_p(t) = \frac{-4 f_p(t) \cdot b_p(t)}{(1-\kappa_p^2)^2}. \tag{6}$$

Direct computation of the gradient, obviously, will require several multiplication and division operations. Alternately, a sign-sign update formula

$$\kappa_p(t+1) = \kappa_p(t) + \mu \cdot \mathrm{sgn}(f_p(t)) \cdot \mathrm{sgn}(b_p(t)) \tag{7}$$

will require only an addition operation to update $\kappa_p(t)$. Equations (5) and (7) are the stochastic gradient adaptive lattice algorithm for normalized lattice filters. In the literature, there are also two types of stochastic gradient lattice algorithms for *un-normalized* lattice filters [10]. Regardless of which of these adaptive algorithms are used, however, none of them admit practical realization using a CORDIC-based arithmetic unit. This is because these adaptation algorithms update $\kappa_p(t)$ rather than the CORDIC rotation angle $\theta_p(t)$. As such, when a CORDIC processor is used to compute the lattice operation, at every time step, $\kappa_p(t)$ will need to be converted to $\theta_p(t)$, and then, $\{\nu_i\}$, before the CORDIC processor can evaluate the lattice operation with the new parameter. These conversions constitute tremendous overheads that makes such a naive CORDIC implementation of SG adaptive lattice algorithm impractical. Clearly, the remedy would be to directly update the rotation angle $\theta_p(t)$ or, better yet, the set of parameters $\{\nu_i; 0 \le i \le n-1\}$, rather than $\kappa_p(t)$.

III. CALF: CORDIC Adaptive Lattice Filter

In this section, we will derive the CALF algorithm. Although a preliminary formulation of CALF has been reported earlier [11], the derivation below provides much details which are needed to facilitate the convergence proof.

A. Updating Rotation Angles

In order to derive an updating formula for the rotation angle θ, we take the gradient of $J_p(t)$, which is defined in (4), with respect to θ_p. Based on (1), it is not difficult to verify that

$$\frac{\partial J_p(t)}{\partial \theta_p} = 2\frac{(f_{p-1}^2(t) + b_{p-1}^2(t-1))\,(1+\kappa_p^2(t-1))}{(1-\kappa_p^2(t-1))} \cdot [\tanh 2\theta_p(t) - k_p^*(t)] = -2 f_p(t) \cdot b_p(t) \tag{8}$$

where

$$k_p^*(t) = \frac{2 f_{p-1}(t) \cdot b_{p-1}(t-1)}{f_{p-1}^2(t) + b_{p-1}^2(t-1)} \tag{9}$$

is an instantaneous estimate of the *reflection coefficient* of an *un-normalized* lattice filter according to Burg's formula [3]

$$\hat{k}_p = \frac{\mathrm{E}\{2 f_{p-1}(t) \cdot b_{p-1}(t-1)\}}{\mathrm{E}\{f_{p-1}^2(t) + b_{p-1}^2(t-1)\}}. \tag{10}$$

From (8), it is easy to prove the following lemma.

Lemma 1: Define the instantaneous estimate of the rotation angle:

$$\theta_p^*(t) = \frac{1}{2}\tanh^{-1}\left\{\frac{2 f_{p-1}(t) b_{p-1}(t-1)}{f_{p-1}^2(t) + b_{p-1}^2(t-1)}\right\} = \frac{1}{2}\tanh^{-1} k_p^*(t). \tag{11}$$

Then, $\theta_p^*(t) > \theta_p(t)$, if and only if $f_p(t) \cdot b_p(t) > 0$.

Proof: This lemma follows from the definition of $\theta_p^*(t)$ and (8). ■

Now, we would like to find a gradient-based updating formulation for $\theta_p(t)$

$$\theta_p(t+1) = \theta_p(t) - \mu \frac{\partial J_p(t)}{\partial \theta_p(t)} = \theta_p(t) + \mu\{f_p(t) b_p(t)\}. \tag{12}$$

To avoid the multiplication operation, a sign-sign updating algorithm similar to (7) can be derived from (12)

$$\theta_p(t+1) = \theta_p(t) + \mu \cdot \mathrm{sgn}[f_p(t)] \cdot \mathrm{sgn}[b_p(t)]. \tag{13}$$

Compared with (12), (13) requires only a simple addition for each update of $\theta_p(t)$.

However, neither (12) nor (13) will be suitable if the lattice sections are to be realized with a CORDIC processor. This is because the rotation angle $\theta_p(t)$ must be converted to the corresponding set of $\{\nu_i(t);\ 0 \le i \le n-1\}$ before the CORDIC iteration may proceed. This conversion will incur significant overhead in terms of additional hardware and delay. The CALF algorithm is designed to alleviate such a problem by updating the implicit representation of θ, i.e. $\{\nu_i(t)\}$ directly.

B. The CALF Algorithm

From (2), it is clear that the set of so called *shift sequence* $\{\nu_i;\ 0 \le i \le n-1\}$ can be regarded as a representation of the rotation angle θ using the set of elementary rotation angles $\{a(i); 0 \le i \le n-1\}$ as the basis. Since that there are only n elementary rotation angles and that the coefficient ν_i is restricted to $\{+1, -1\}$, there are exactly 2^n distinct values that θ can take. In other words, using the CORDIC algorithm, θ will be *quantized* into one of the 2^n predetermined values.

Let us define a n-bit binary integer $\boldsymbol{d} = d_0\ d_1 \cdots d_{n-1}$ such that $d_i \in \{0,1\}$ and that $d_i = (1+\nu_i)/2$. Here, we let d_0 be the most significant bit and d_{n-1} be the least significant bit of $\boldsymbol{d}$. Then, the binary number $\boldsymbol{d}$ and the rotation angle $\theta = \Sigma_{i=0}^{n-1}\ \nu_i a(i)$ have a one-to-one correspondence. Moreover, between θ and $\boldsymbol{d}$, the magnitude precedence relation is maintained.

Lemma 2: Let $\boldsymbol{d}_a$ and $\boldsymbol{d}_b$ correspond to θ_a and θ_b, respectively. $\theta_a \le \theta_b$ if and only if $\boldsymbol{d}_a \le \boldsymbol{d}_b$.

Proof: We make use an important property of the inverse hyperbolic tangent function

$$a(i^*) = \tanh^{-1} 2^{-i^*-1} > \sum_{i=i^*+1}^{n-1} a(i) = \sum_{i=i^*+1}^{n-1} \tanh^{-1} 2^{-i-1}. \quad (14)$$

Now, denote $\Delta_i = (\nu_{ia} - \nu_{ib})/2 = d_{ia} - d_{ib} \in \{-1, 0, 1\}$. Then, $\theta_a \le \theta_b$ implies

$$\theta_a - \theta_b = \sum_{i=0}^{n-1} (\nu_{ia} - \nu_{ib}) \cdot a(i) = 2 \sum_{i=0}^{n-1} \Delta_i \cdot a(i) \le 0. \quad (15)$$

Using (14), we may deduce that (15) is true if and only if there exists an i^* such that

$$\Delta_i = \begin{cases} 0 & i < i^* \\ -1 & i = i^*. \end{cases} \quad (16)$$

However, (16) is true if and only if

$$\boldsymbol{d}_a - \boldsymbol{d}_b = \sum_{i=0}^{n-1} (d_{ia} - d_{ib}) \cdot 2^{-i-1} = \sum_{i=0}^{n-1} \Delta_i \cdot 2^{-i-1} \le 0. \quad (17)$$

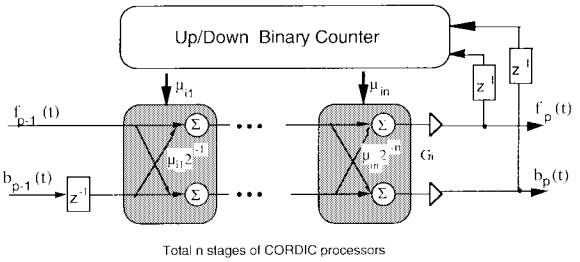

Fig. 2. CALF implementation of pth stage lattice filter.

The last inequality follows from the fact that $2^{-i^*-1} > \Sigma_{i=i^*+1}^{n-1} 2^{-i-1}$. ■

With μ being a positive integer, Lemma 2 guarantees that the binary number $\boldsymbol{d}$ can be updated using

$$\boldsymbol{d}(t+1) = \boldsymbol{d}(t) + \mu \cdot \mathrm{sgn}[f_p(t)] \cdot \mathrm{sgn}[b_p(t)]. \quad (18)$$

Combining the CORDIC-implemented lattice filter operation (1) with (18), we have an implementation of the CALF algorithm as

Initiation : $f_p(0) = b_p(0) = 0, 0 \le p \le P_L$ (P_L is the number of lattice sections); select $\boldsymbol{d}_p(0), 1 \le p \le P_L$; μ: a positive integer; $t = 1$; and terminated $=$ false.
While terminated $=$ false,
  $f_0(t) = b_0(t) = x(t)$;
  For $p = 1$ **to** P_L,
    $x(0) = f_{p-1}(t), y(0) = b_{p-1}(t-1)$;
    For $i = 0$ **to** $n-1$ **Do** /* CORDIC iterations */
      $\nu_i = 2d_i(p) - 1$;
      $\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\nu_i 2^{-i-1} \\ -\nu_i 2^{-i-1} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$;
  End /* For-i Loop */
    $f_{p+1}(t) = x(n)$, $b_{p+1}(t) = y(n)$;
    If $0 < \boldsymbol{d}_p(t) < 2^n - 1$,
      $\boldsymbol{d}_p(t+1) = \boldsymbol{d}_p(t) + \mu \cdot \mathrm{sgn}[x(n)] \cdot \mathrm{sgn}[y(n)]$;
    Elseif $\boldsymbol{d}_p(t) = 0$ or $\boldsymbol{d}_p(t) = 2^n - 1$,
      $\boldsymbol{d}_p(t+1) = \boldsymbol{d}_p(t)$
    End
  End /* of p loop */
  $t = t + 1$;
  If $t > t_{\max}$, **then** terminated $=$ true;
End /* of while not (terminated) loop */.

One important observation is that (18) can be realized with a simple binary up/down counter. A block diagram of a single stage lattice filter implemented using CALF is depicted in Fig. 2.

We also observe that the actual step size, in terms of rotation angle, is *not* constant in CALF. This is due to the nonlinearity of the inverse tanh function. Specifically, according to (14),

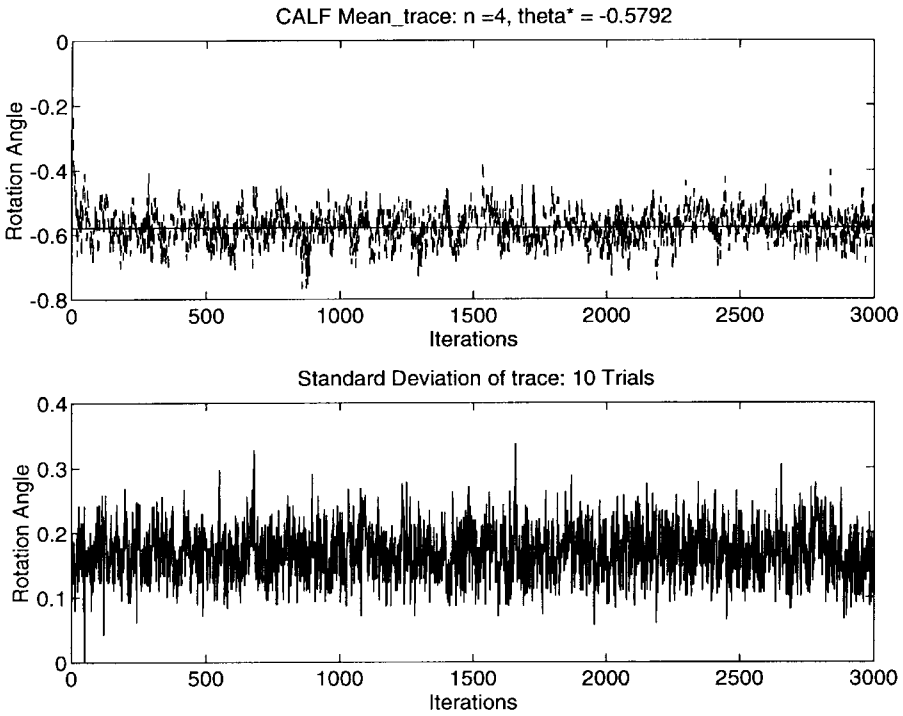

Fig. 3. Average trace of ten runs of CALF $\theta_1^* = -.5792$, $n = 4$.

the largest step size

$$\delta_{\max} = 2\left(a(0) - \sum_{i=1}^{n-1} a(i)\right) \geq 0.862858 \qquad (19)$$

regardless of how large the value of n may be. These fixed step size obviously will impact on the accuracy of the parameter estimate $\theta(t)$ of the CALF algorithm. More details and possible solutions will be discussed in Section IV-D.

Example 1: We generated a second-order AR process with a spectral shaping filter $A(z) = 1 - 1.6z^{-1} + 0.95z^{-2}$. This corresponds to $\theta_1^* = -.5792$ and $\theta_2^* = 0.916$. The excitation process is a white, normal, wide-sense stationary random process with zero mean, and $\sigma^2 = 0.01$. We used three different values of n, namely, $n = 4$, $n = 8$, and $n = 12$ for experimentation and fix $\mu = 1$ for all trials. For each value of n, ten trials on independently generated AR process, with 10 000 samples each, are performed. The traces of the parameter estimate of θ_1^*, $\hat{\theta}_1(t)$ are recorded for each trial. The mean and standard deviation over these ten trials on each time step are shown in Figs. 3–5. From these results, we observe that when n is small, the convergence is faster, although it suffers from larger variations. As n increases so that the step size decreases, the convergence is slower and with smaller variations. In what follows, we will derive the theory that explains these observations.

IV. Convergence Analysis of CALF

In this section, we will show that if the input to the adaptive lattice filter is a stationary random process, then the rotation angle of each lattice section, updated using the CALF algorithm, comprises a *regular* Markov chain. This Markov chain will contain 2^n states, each corresponding to a distinct value of $\boldsymbol{d}$ and, hence, a distinct rotation angle. The CALF angle adaptation process thus corresponds to the state transition within this Markov chain. By showing that this Markov chain is *regular*, we invoke the asymptotic convergence theory of a regular Markov chain to establish the convergence of the rotation angle of the CALF algorithm to a limiting probability distribution. Furthermore, by showing that this CALF-induced Markov chain is reversible, we relate the convergence rate in terms of a time constant of the convergence curve to the second largest eigenvalue (in magnitude) of the state probability transition matrix. As such, we are able to establish the exponential convergence rate of the CALF algorithm. We note that previously, the Markov chain-based method has been used to analyze the convergence of sign-sign adaptive lattice algorithms [14], which uses conventional arithmetic unit. The application of this approach to a CORDIC-based lattice filter algorithm appears to be new.

A. Structure of the Markov Chain

For convenience, let us denote θ_i as the rotation angle corresponding to $\boldsymbol{d}_p = i$ and $S = \{\theta_i;\ 0 \leq i \leq 2^n - 1\}$ to be the set of all possible θ_i's. By Lemma 2, $\theta_p(t) = \theta_i$ if and only if $\boldsymbol{d}_p(t) = i$. If $\mu = 1$, then S will be the set of rotation angles reachable through the adaptation process of the CALF algorithm. If $\mu > 1$, then only a subset of S will be realized by the CALF algorithm. In this paper, we will assume $\mu = 1$ in the following discussion. For convenience, we will use θ_i and its index i interchangeably.

The CALF direct angle updating formula (18) dictates the state transition probability from state i at time instance t.

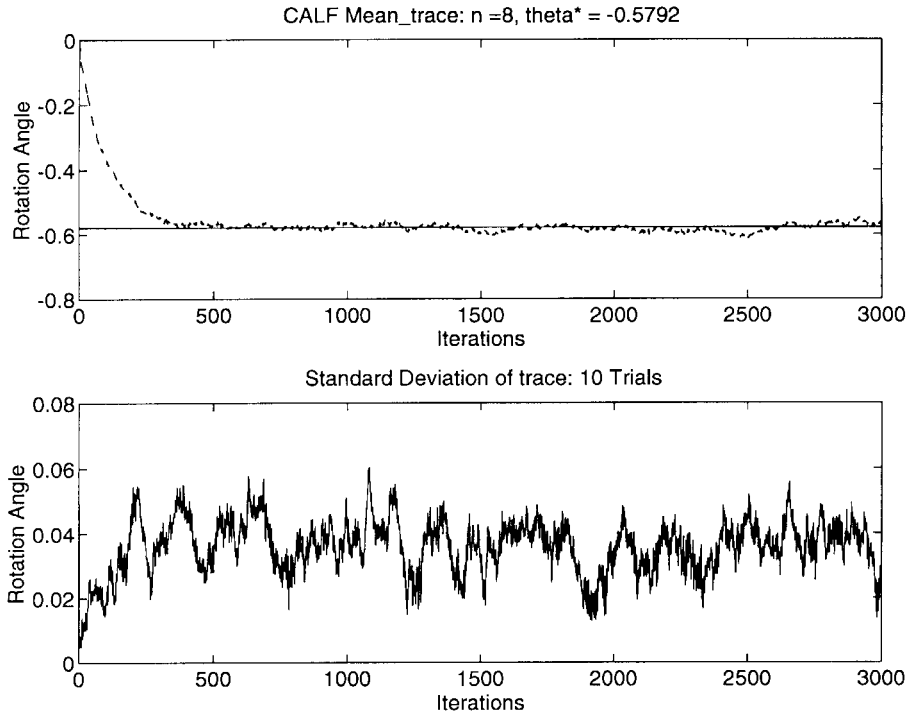

Fig. 4. Average trace of ten runs of CALF $\theta_1^* = -.5792$, $n = 8$.

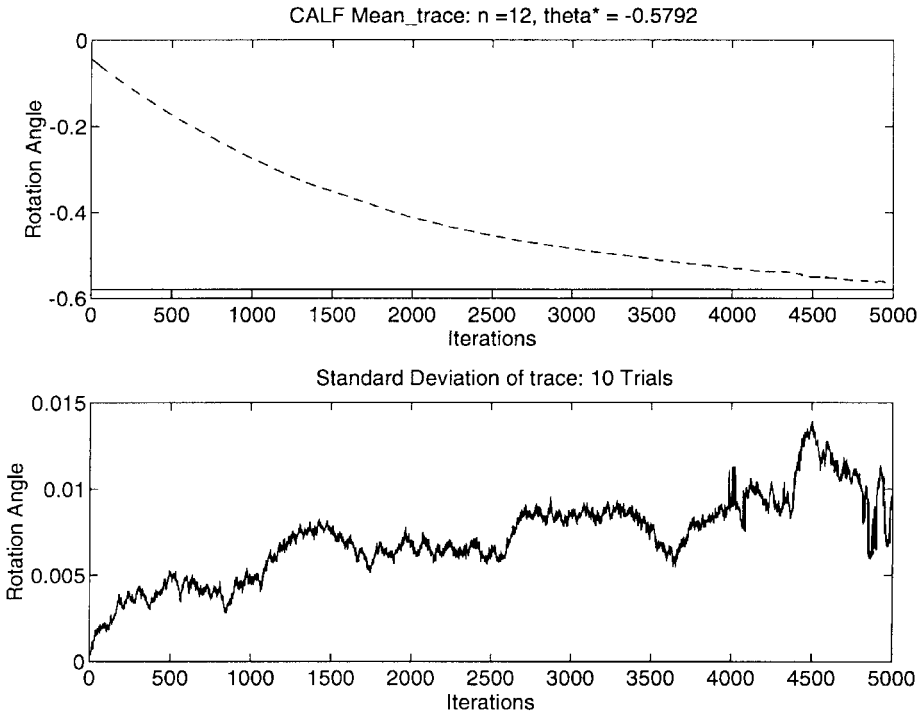

Fig. 5. Average trace of ten runs of CALF, $\theta_1^* = -.5792$, $n = 12$.

In particular, for $0 \le i, j \le 2^n - 1$

$$\Pr\{\theta_p(t+1) = \theta_j | \theta_p(t) = \theta_i\} = \begin{cases} \Pr\{f_p(t)b_p(t) > 0 | \theta_p(t) = \theta_i\} & j = i+1 \ge 1 \\ \Pr\{f_p(t)b_p(t) < 0 | \theta_p(t) = \theta_i\} & 2^n - 2 \ge j = i-1 \\ 0 & |j-i| \ne 1. \end{cases} \tag{20}$$

Lemma 3: If $[f_{p-1}(t)\ b_{p-1}(t-1)]^T$ is a jointly normal distributed, wide-sense stationary random process, then the one-step state transition probability will be independent of the time index. That is

$$\Pr\{\theta_p(t+1) = \theta_j | \theta_p(t) = \theta_i\} = \Pr(j|i).$$

Proof: At state i, the rotation angle θ_i is a constant. Therefore, according to (1), both $f_p(t)$ and $b_p(t)$ will be linear combinations of $f_{p-1}(t)$ and $b_{p-1}(t-1)$. That is to say, the output of the pth stage lattice $[f_p(t)\ b_p(t)]^T$ will also be a jointly normal distributed, wide-sense stationary random process. Thus, the probability distribution of the product $f_p(t)\cdot b_p(t)$ will be independent of time index t. Hence, the lemma is proved.

Lemma 3 confirms the fact that the underlying Markov process is a time homogeneous Markov chain because the state transition probability is independent of time indices. At the boundary states, where $i = 0$ or $2^n - 1$, we require $\boldsymbol{d}_p(t)$ to remain at state 0 even if $f_p(t)b_p(t) < 0$, and at state $2^n - 1$ even if $f_p(t)b_p(t) > 0$. That is, we force $P(0|0) = \Pr\{f_p(t)b_p(t) < 0 | \theta_p(t) = \theta_0\}\}$ and $P(2^n - 1|2^n - 1) = \Pr\{f_p(t)b_p(t) > 0 | \theta_p(t) = \theta_{2^n-1}\}$.

Lemma 4: Let f and b be two jointly distributed normal random variables with zero mean and variance σ_f^2 and σ_b^2, respectively, and correlation coefficient $r_{fb} = \mathrm{E}\{f\cdot b\}/[\sigma_f\cdot\sigma_b]$. Define a random variable $Z = 2f\cdot b/[f^2 + b^2]$, and $F_z(z) = \Pr\{Z \le z\}$. Then

$$F_z(z) = \begin{cases} 1 - F_u(g(z)) + F_u(h(z)) & 0 < z \le 1 \\ F_u(h(z)) - F_u(g(z)) & -1 \le z \le 0 \end{cases} \tag{21}$$

where the roots of the equation $x^2 - (2/z)x + 1 = 0$ are

$$g(z) = [1 + \sqrt{1 - z^2}]/z, \quad h(z) = [1 - \sqrt{1 - z^2}]/z \tag{22}$$

and $U = f/b$ is a random variable with a *Cauchy distribution:*

$$F_u(u) = \Pr\{f/b \le u\} = \frac{1}{2} + \frac{1}{\pi}\tan^{-1}\frac{\sigma_b u - r_{fb}\sigma_f}{\sigma_f\sqrt{1 - r_{fb}^2}}. \tag{23}$$

Proof: The distribution function of the random variable z can be found as

$$\begin{aligned} F_z(z) &= \Pr\left\{\frac{2f\cdot b}{f^2 + b^2} \le z\right\} \\ &= \Pr\left\{z\cdot\left(\frac{f}{b} - g(z)\right)\cdot\left(\frac{f}{b} - h(z)\right) \ge 0\right\} \end{aligned} \tag{24}$$

where $g(z)$ and $h(z)$ are defined in (22). It can easily be shown that $g(-1) = h(-1) = -1$, $g(1) = h(1) = 1$, and $|g(z)| \ge |h(z)|$ for $|z| \le 1$. In particular, $h(0) = \lim_{z\to 0} h(z) = 0$. Hence, for $-1 < z \le 0$, we must have $g(z) < f/b < h(z) < 0$. That is to say, $F_z(z) = \Pr\{g(z) < f/b < h(z)\}$. On the other hand, for $0 < z < 1$, we must have $F_z(z) = \Pr\{f/b > g(z)\} + \Pr\{f/b < h(z)\}$. These observations lead to (21). Finally, we note that the random variable u has a *Cauchy density* centered at $r_{fb}\sigma_f/\sigma_b$, as shown in (23) [19]. ■

Based on Lemmas 1 and 4, the probability of state transition can be found according to the following theorem.

Theorem 1: Let $f_{p-1}(t)$ and $b_{p-1}(t-1)$ have a joint normal distribution with zero mean, variances σ_f^2 and σ_b^2, and correlation coefficient r_{fb}. If $\boldsymbol{d}_p(t) = i$ is the current state (i.e., $\theta_p(t) = \theta_i$), then

$$P(i+1|i) = \Pr\{\theta_{i+1}|\theta_i\} = 1 - F_z(\tanh 2\theta_i). \tag{25}$$

Proof: Denote $Z = \tanh 2\theta_p^*(t)$, and from Lemma 1

$$\begin{aligned} P\{i+1|i\} &= \Pr\{\boldsymbol{d}_p(t+1) = i+1 | \boldsymbol{d}_p(t) = i\} \\ &= \Pr\{f_p(t)\cdot b_p(t) > 0 | \theta_p(t) = \theta_i\} \\ &= \Pr\{\theta_p^*(t) > \theta_i\} = \Pr\{\tanh 2\theta_p^*(t) > \tanh 2\theta_i\} \\ &= \Pr\{Z > \tanh 2\theta_i\} = 1 - F_z(\tanh 2\theta_i). \end{aligned}$$ ■

Corollary 1: For $0 < i < 2^n - 1$

$$P(i-1|i) = 1 - P(i+1|i) = F_z(\tanh 2\theta_i). \tag{26}$$

Corollary 2: Let

$$\theta^* = \frac{1}{2}\tanh^{-1}\frac{\mathrm{E}\{2f_{p-1}(t)\cdot b_{p-1}(t-1)\}}{\mathrm{E}\{f_{p-1}^2(t) + b_{p-1}^2(t-1)\}}. \tag{27}$$

Then

$$P(i+1|i) > \tfrac{1}{2} > P(i-1|i) \quad \text{if } \theta_i < \theta^* \tag{28}$$

$$P(i+1|i) < \tfrac{1}{2} < P(i-1|i) \quad \text{if } \theta_i > \theta^*. \tag{29}$$

Proof: Recall the trigonometric identity

$$\tan^{-1} A - \tan^{-1} B = \tan^{-1}\frac{A - B}{1 + AB} \tag{30}$$

if the result falls within $\{-\pi/2, \pi/2\}$. Otherwise, the right-hand side must be corrected by adding or subtracting π. Using (23)

$$\begin{aligned} &F_u(h(z)) - F_u(g(z)) \\ &\quad = \frac{1}{\pi}\cdot\tan^{-1} \\ &\qquad \cdot\frac{\sigma_b[h(z) - g(z)]\cdot(\sigma_f\sqrt{1 - r_{fb}^2})}{\sigma_f^2\cdot(1 - r_{fb}^2) + (\sigma_b h(z) - r_{fb}\sigma_f)(\sigma_b g(z) - r_{fb}\sigma_f).} \end{aligned} \tag{31}$$

Using the relation $h(z)\cdot g(z) = 1$ and $h(z) + g(z) = 2/z$

$$\begin{aligned} &(\sigma_b h(z) - r_{fb}\sigma_f)(\sigma_b g(z) - r_{fb}\sigma_f) \\ &\quad = \sigma_b^2 + \sigma_f^2 r_{fb}^2 - \sigma_f\sigma_b r_{fb}\cdot(2/z). \end{aligned}$$

Therefore

$$\sigma_f^2\cdot(1 - (\sigma_b g(z) - r_{fb}\sigma_f) = [\sigma_f^2 + \sigma_b^2]\cdot(1 - \rho/z) \tag{32}$$

where $\rho = \mathrm{E}\{2f\cdot b\}/\mathrm{E}\{f^2 + b^2\} = 2r_{fb}\cdot\sigma_f\sigma_b/(\sigma_f^2 + \sigma_b^2) = \tanh 2\theta^*$ is the Burg's estimate of the reflection coefficient [3]. Note that the denominator can be either greater or less than 0, depending on whether ρ/z is greater or less than 1.

Next, because $h(z) - g(z) = -2\sqrt{1 - z^2}/z$, the numerator of (31) can be simplified to $(-2\sqrt{1 - z^2}/z)\cdot\sigma_f\sigma_b\sqrt{1 - r_{fb}^2}$. Substituting (32) and the above expression into (31), we have

$$\begin{aligned} &F_u(h(z)) - F_u(g(z)) \\ &\qquad = \frac{1}{\pi}\cdot\tan^{-1}\frac{\rho\sqrt{1 - z^2}\sqrt{1 - r_{fb}^2}}{r_{fb}\cdot[z - \rho].} \end{aligned} \tag{33}$$

This expression is valid for the range $-1 < z \le 0$ and $z > \rho$. If $z < \rho$, the right-hand side become negative, but the left-hand side remains positive (because $h(z) > g(z)$ for $-1 < z \le 0$). This implies the inverse tangent portion on the right-hand side

needs to be corrected by adding π. Hence, if $-1 < z \leq 0$ and $z < \rho$, we have

$$F_u(h(z)) - F_u(g(z)) = 1 + \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [z-\rho]}. \quad (34)$$

The same argument can be applied to the range $0 < z < 1$, where $F_u(z) = 1 + F_u(h(z)) - F_u(g(z))$. Specifically

$$1 + F_u(h(z)) - F_u(g(z)) = \begin{cases} \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [z-\rho]} & z > \rho \\ 1 + \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [z-\rho]} & z < \rho. \end{cases} \quad (35)$$

Combining (33)–(35), we have

$$F_z(z) = \begin{cases} 1 + \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [z-\rho]} & z < \rho. \\ \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [z-\rho]} & z > \rho. \end{cases} \quad (36)$$

Moreover,

$$F_z(\rho) = \lim_{z \to \rho} \left\{ \frac{1}{\pi} \cdot \tan^{-1} \frac{\rho\sqrt{1-z^2}\sqrt{1-r_{fb}^2}}{r_{fb} \cdot [\rho - z]} \right\} = 1/2. \quad (37)$$

Hence, from Corollary 1

$$P(i-1|i) = 1 - P(i+1|i) = F_z(\tanh 2\theta_i) < F_z(\tanh 2\theta^*) = 1/2. \quad (38)$$

Finally, (29) can be proved similarly. ■

Note that $\rho/r_{fb} = 2\sigma_f\sigma_b/(\sigma_f^2+\sigma_b^2) \leq 1$, with the maximum reached when $\sigma_f = \sigma_b$.

Corollary 3: If $\theta_i < \theta_{i+1}$, then $P(i-1|i) < P(i|i+1)$.

Proof: From (38), $P(i-1|i) = F_z(\tanh 2\theta_i) < P(i|i+1) = F_z(\tanh 2\theta_{i+1})$. ■

B. Asymptotic Convergence

In this section, we apply results derived in the previous section to prove the asymptotic convergence of the CALF algorithm. Our approach is to show that the adaptation steps in the CALF algorithm effect a regular, finite-state Markov chain. Then, we apply the convergence theorem of a regular Markov chain to establish the asymptotic convergence toward a limiting state probability distribution of the CALF algorithm.

Definition 1 [17]: A Markov chain is said to be *regular* if after sufficient number of state transitions, all elements of the state probability vector are positive, regardless of the initial state probability vector π_0.

Theorem 2: Denote $\boldsymbol{S} = \{\theta_i; 0 \leq i \leq 2^n - 1\}$ to be the set of states and $\boldsymbol{P}$ to be the state transition probability matrix such that its i,jth element $p_{ij} = P(j|i)$, as defined in Theorem 1; then, the corresponding Markov chain $\{\boldsymbol{S}, \boldsymbol{P}\}$ is a regular Markov chain.

Proof: From (20), it is clear that the corresponding state transition matrix $\boldsymbol{P}$ has a banded-matrix structure such that its i,jth element $p_{ij} = P(j|i)$ is

$$p_{ij} = \begin{cases} F_z(\tanh 2\theta_i) & j = i-1 \\ 1 - F_z(\tanh 2\theta_i) & j = i+1 \\ F_z(\tanh 2\theta_0) & j = i = 0 \\ 1 - F_z(\tanh 2\theta_{2^n-1}) & j = i = 2^n - 1 \\ 0 & \text{otherwise.} \end{cases} \quad (39)$$

Thus, $\boldsymbol{P}$ is a banded matrix with a *bandwidth*[3] equal to 3. Furthermore, all elements on the upper and lower subdiagonal of the $\boldsymbol{P}$ matrix are positive. It is well known that the bandwidth of the product of two banded matrices is the sum of the bandwidths of these two matrices minus one. Hence, the nth step probability transition matrix $\boldsymbol{P}^n$ will be a full matrix consisting of positive elements. This implies that the probability of making a state transition from a state i to any other state j in n steps must be greater than zero. This observation, together with Theorem 1 and Corollary 1, lead to the conclusion that the corresponding Markov chain is a regular Markov chain.

Given a regular Markov chain, it will converge to a unique limiting state probability distribution vector α, regardless of its initial state distribution $\pi(0)$. For convenience, we include the results from [17] as Theorem 3 below without proof.

Theorem 3: If $\boldsymbol{P}$ is a regular transition matrix, then we have the following.

1) The powers P^k approach a probability matrix Π.
2) Each row of Π is the same probability vector $\pi(\infty)$.
3) The components of $\pi(\infty)$ are positive.

Theorem 3 establishes the asymptotic convergence of the probability transition matrix of a regular Markov chain to a probability matrix Π. Consequently, for any initial state probability (row) vector $\pi(0)$, we have

$$\lim_{k\to\infty} \pi(k) = \lim_{k\to\infty} \pi(0)\boldsymbol{P}^k = \pi(0) \lim_{k\to\infty} \boldsymbol{P}^k = \pi(0)\Pi = \pi(\infty). \quad (40)$$

In other words, any initial state probability distribution will converge to a unique limit state probability distribution. Interpreted in terms of the CALF algorithm, the above results imply that the rotation angle $\theta_p(t)$ converges to a *random variable* $\theta_p(\infty)$ that has a unique probability distribution $\pi(\infty)$.

Theorem 4: Let $\theta_p(t)$ and $\pi(\infty)$ be defined as above. Then

$$\lim_{t\to\infty} \Pr\{\theta_p(t) = \theta_i\} = \pi_i$$

where π_i is the ith component of $\pi(\infty)$.

[3] Here, the numerical linear algebra term *bandwidth* refers to the number of nonzero main and subdiagonals of the matrix P and has nothing to do with the rise in bandwidth frequency analysis in signal processing.

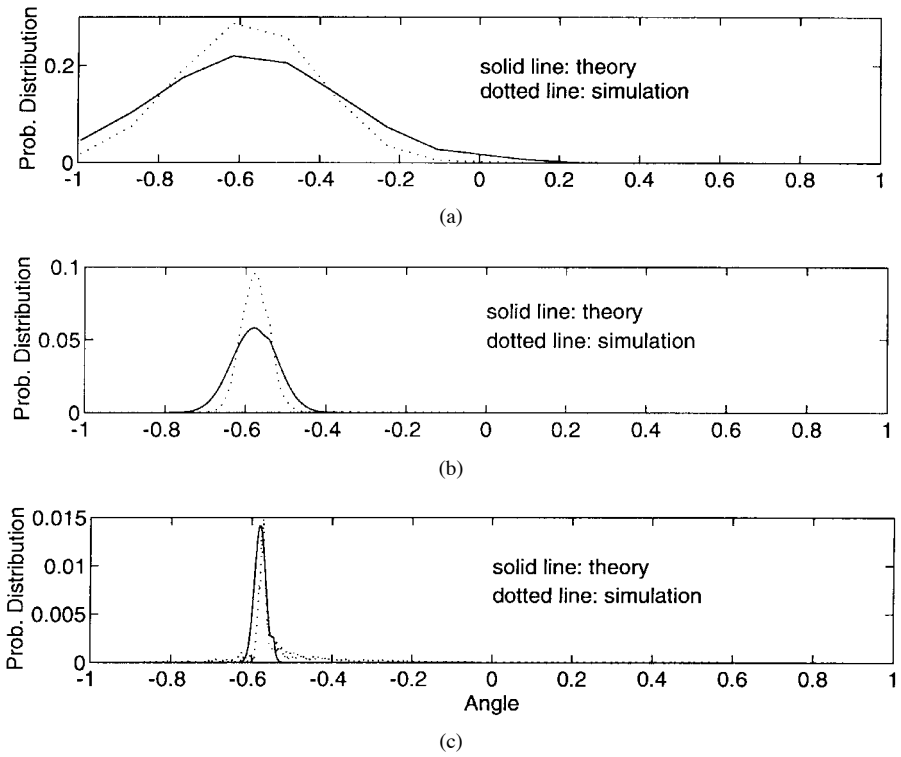

Fig. 6. Theoretical versus empirical distribution of $\hat{\theta}_1(t)$, $n = 4, 8$, and 12.

Example 2: Based on the same model as described in Example 1, consider the case where $n = 2$. The one-step state transition probability matrix is

$$\boldsymbol{P} = \begin{bmatrix} 0.3611 & 0.6389 & 0 & 0 \\ 0.6725 & 0 & 0.3275 & 0 \\ 0 & 0.8901 & 0 & 0.1099 \\ 0 & 0 & 0.9601 & 0.0399 \end{bmatrix}.$$

The banded structure of the $\boldsymbol{P}$ matrix is quite clear. The limiting probability distribution vector corresponding to $\theta_i = [-0.8047 \ -0.2939 \ 0.2939 \ 0.8047]$ is

$$\pi(\infty) = [0.4274 \quad 0.4061 \quad 0.1494 \quad 0.0171].$$

An explicit formulation for the limiting state probability distribution is generally too complicated for any practical problem. However, we may still deduce some of the basic properties. Our plan is to show that the limit probability distribution usually peaked around the true parameter value θ_p^*, regardless of the number of states.

Corollary 4: $\pi_i > \pi_j$ if and only if $\theta_j < \theta_i < \theta^*$ or $\theta^* < \theta_i < \theta_j$.

Proof: From Corollary 2, if $\theta_i < \theta_{i+1} < \theta^*$, we have $P(i+1|i) > P(i+2|i+1) = 1 - P(i|i+1) > \frac{1}{2}$. Thus, $P(i+1|i) > \frac{1}{2} > P(i|i+1)$. On the convergence of a regular Markov chain, we have

$$P(i+1|i)\pi_i = P(i|i+1)\pi_{i+1} \tag{41}$$

for each i. Therefore, $\pi_i < \pi_{i+1}$. By repeated application of this argument, we can easily prove that if $\theta_i < \theta_j$, then $\pi_i < \pi_j$. The same argument applies when $\theta_i > \theta^*$. Finally, by induction, this theorem is proved.

Corollary 4 guarantees that the state probability is larger for states closer to the optimal rotation angle θ^*.

Example 3: With the same experiment performed in Example 1, we plot the empirical rotation angle distribution of the ten trials and compare it with the theoretically computed results based on theories developed in this section. We assume that $\rho = r_{fb} = 1$ (i.e. $\sigma_f = \sigma_b$). The empirical distribution is generated by counting the histogram of rotation angles out of the ten trial runs. In each trial, the initial value of $\boldsymbol{d}_p(0)$ is selected randomly between 0 and 2^{n-1}, and the algorithm runs 100 000 time steps to ensure that the convergence is achieved. Then, the distribution of all the rotation angles in each time step are computed. These angle distributions for all three cases of $n = 4$, $n = 8$, and $n = 12$ are plotted in Fig. 6. We note that there are some discrepancies between the solid curve, which is the predicted angle distribution from the theorems developed in this paper, and the dotted lines, which are obtained in this experiment. These differences are mainly due to statistical variations because only ten trials are used.

C. Empirical Analysis of Convergence Rate

From Example 1 earlier, we observe that the convergence of CALF is faster when n is small and slower when n is large. To quantify these different convergence rates, we need to analyze the eigenvalue distribution of the probability transition matrix $\boldsymbol{P}$. Toward this goal, we first review a definition called *reversible Markov chain*.

Definition 2: A Markov chain is *reversible* if

$$\pi_i \cdot p_{ij} = \pi_j \cdot p_{ji} \tag{42}$$

where π_i and π_j, respectively, are the ith and jth element of the limiting state probability vector π, and $p_{ij} = P(j|i)$ is the i, j element of the probability transition matrix $\boldsymbol{P}$.

Lemma 5: The Markov chain induced by the CALF algorithm is a reversible Markov chain.

Proof: From (41) in Corollary 4, (42) is verified for the case of $|j-i| = 1$. Now that $P(j|i) = p_{ij} = 0$ for $|j-i| > 1$, as shown in (39), clearly, (42) should also be satisfied for $|j-i| > 1$, as both sides are zeros. ■

A reversible Markov chain has a very desirable property regarding the eigenvalues and eigenvectors of its probability transition matrix.

Theorem 5 [6]: Let $\boldsymbol{P}$ be the probability transition matrix of a reversible Markov chain; then, it has real eigenvalues and a set of linearly independent eigenvectors.

Proof: Define $D = \text{diag}\{\sqrt{\pi_1}, \sqrt{\pi_2}, \cdots, \sqrt{\pi_N}\}$, where $N = 2^n$ is the total number of states. The i, jth element of the matrix DPD^{-1} can be found as

$$\sqrt{\frac{\pi_i}{\pi_j}} p_{ij} = \sqrt{\frac{\pi_j}{\pi_i}} \frac{\pi_i}{\pi_j} p_{ij} = \sqrt{\frac{\pi_j}{\pi_i}} p_{ji}. \tag{43}$$

Hence, DPD^{-1} is a real-valued, symmetric matrix, which implies that it must have a set of real eigenvalues with linearly independent, real-valued eigenvectors. Let $DPD^{-1} = V_D \Lambda V_D^T$ be its eigenvalue decomposition. Clearly, P has an eigenvalue decomposition as

$$\boldsymbol{P} = D^{-1} V_D \Lambda V_D^T D = V \Lambda U = \sum_{i=1}^{N} \lambda_i v_i u_i^T \tag{44}$$

where $U^T = V^{-1}$ and the eigenvalues $\{\lambda_i; 1 \le i \le 2^n = N\}$ are arranged in descending orders with $1 = \lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_N \ge -1$. Moreover, we can easily show that $\Pi = \lambda_1 v_1 u_1^T = v_1 u_1^T$. ■

The convergence of the state probability vector in the CALF algorithm can be measured by the L_2 norm of the difference between the present state probability vector $\pi(k)$ and the state probability vector at convergence $\pi(\infty)$. That is, $e(k) = \|\pi^*(k) - \pi(\infty)\|$.

Theorem 6: Let $\boldsymbol{P}$ be the state transition probability matrix and $\pi(k)$ be the state probability matrix at time k. Denote $\{\lambda_i; 1 \le i \le 2^n\}$ to be the set of eigenvalues of $\boldsymbol{P}$ in descending order of their magnitudes such that $|\lambda_i| \ge |\lambda_j|$ if $i \le j$. Define $e(k) = \|\pi^*(k) - \pi(\infty)\|$. Then, as $k \to \infty$

$$e(k) \to |\pi(0)| \cdot |\lambda_2|^k = |\pi(0)| \cdot \exp[-k/\tau] \tag{45}$$

where $\tau = -1/\ln|\lambda_2|$ is the *time constant* of the convergence curve.

Proof: First, note that

$$\begin{aligned} e(k) &= \|\pi^*(k) - \pi(\infty)\| = \|\pi(0)\{\boldsymbol{P}^k - \Pi\}\| \\ &\le |\pi(0)| \cdot \|\{\boldsymbol{P}^k - \Pi\}\| \\ &= |\pi(0)| \cdot \left\| \sum_{i=1}^{N} \lambda_i v_i v_i^T - \lambda_1 v_1 v_1^T \right\| \\ &= |\pi(0)| \sqrt{\sum_{i=2}^{N} \lambda_i^{2k}} = |\lambda_2|^k \pi(0) \sqrt{1 + \sum_{i=3}^{N} (\lambda_i/\lambda_2)^{2k}}. \end{aligned}$$

Then, take limits on both sides of the above expression, and note that $\lim_{k\to\infty} |\lambda_i/\lambda_2|^k = 0$ for $i > 2$. This leads to (45).

Note that as $\lambda_2 \to 1$, $\tau \to \infty$. Thus, the convergence rate slows down rapidly as λ_2 increases. Using Matlab, we computed the λ_2 based on several different values of n and θ^* and plot the results in Fig. 7. From this figure, it is clear that as n increases and as $|\theta^*|$ decreases, λ_2 approaches 1, which implies slower convergence. These trends are quite evident from the simulation results depicted in Figs. 3–5. In practice, to expedite convergence, we may start by using a smaller value of n in the CALF algorithm. Later, as convergence is detected, the user may opt to increase the value of n to further enhance accuracy.

D. Variance of $\theta(t)$

The limiting distribution of $\theta(t)$, as illustrated in Fig. 6, has a peak located around θ^*, which is the ideal value of $\theta(t)$. This peak becomes narrower as n increases. Let us consider the variance of $\theta(t)$ when θ^* falls within an interval $[\theta_{i*-1}\theta_{i*}]$, with $\theta_{i*} - \theta_{i*-1} = \delta$. We may assume that $\pi_i \approx 0$ for $\theta_i \gg \theta_{i*}$ and $\theta_i \ll \theta_{i*-1}$. Thus

$$\begin{aligned} \text{Var}\{\theta(t)\} &= \sum_{i=0}^{N-1} \pi_i |\theta_i - \theta^*|^2 \\ &\approx \pi_*(\delta - \alpha)^2 + (1 - \pi_*)\alpha^2 \\ &= (\alpha - \pi_*\delta)^2 + \pi_*(1 - \pi_*)\delta^2 \ge \pi_*(1 - \pi_*)\delta^2 \end{aligned} \tag{46}$$

where $\pi_* = \Sigma_{i=0}^{i*-1} \pi_i$,, and $\alpha = \theta_{i*} - \theta^*$. Therefore, the variance of $\theta(t)$ is roughly proportional to δ^2. Obviously, when $\delta = \delta_{max} \approx 0.862$ according to (19), the variance can be considerably large.

The classical solution for this uneven step-size problem, according to Walther [21], is to allow the duplication of several elements in the shift sequences $\{\nu_i\}$ or, equivalently, duplicate specific elementary rotation angles $a(i)$. For $n < 13$, only $a(4)$ needs to be repeated.[4] When applied to CALF, this remedy will lead to a nonmonotonic relation between $\boldsymbol{d}$ and θ, which violates Lemma 2. This is illustrated in Fig. 8. Fortunately, as shown in the figure, the case where $\theta_i > \theta_j$ for $i < j$ only occurs at isolated points. These points are such that (15) no longer implies (16). For example, if $n < 13$, then these pairs of points can be found as $i = 2^{n-5} - 1 + 16k$ and $i = 2^{n-5} + 16k$, for $0 \le k \le 14$. Thus, when the CALF algorithm is applied to such a modified sequence of elementary rotation angles, the convergence proof given in this section is still applicable.

E. Mean Square Prediction Error

The CALF algorithm is derived by minimizing an instantaneous estimate of the (single stage) output prediction error $J = f_p^2(t) + b_p^2(t)$. If $\theta(t) \to \theta^*$, by (4), we see that $J_p(t)$ is minimized. Similarly, it is not difficult to show that the mean

[4] Other angles to be repeated are $a(13)$, $a(40)$, and so on.

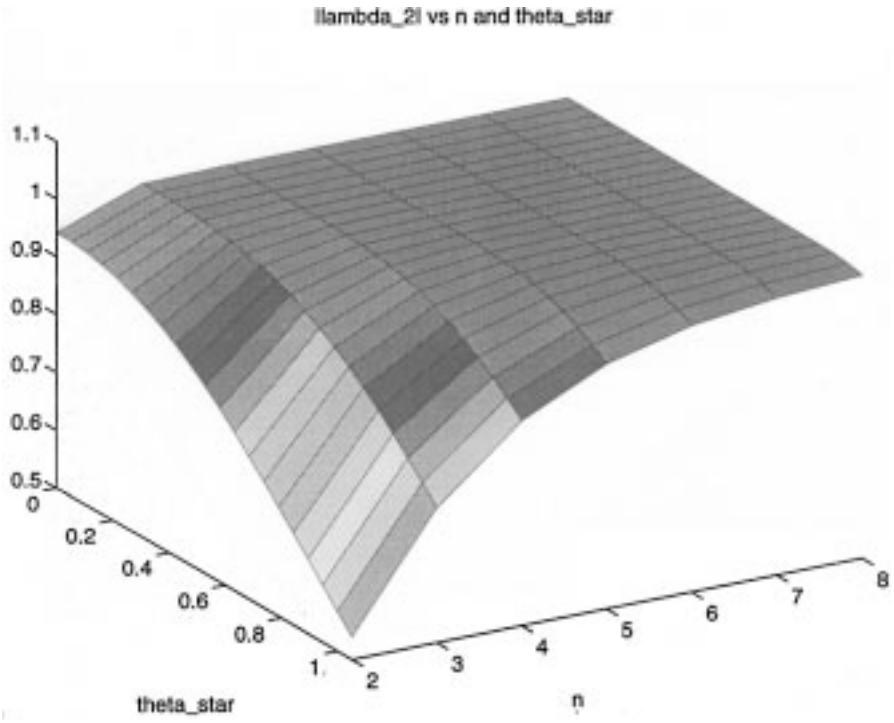

Fig. 7. λ_2 versus n and θ^*.

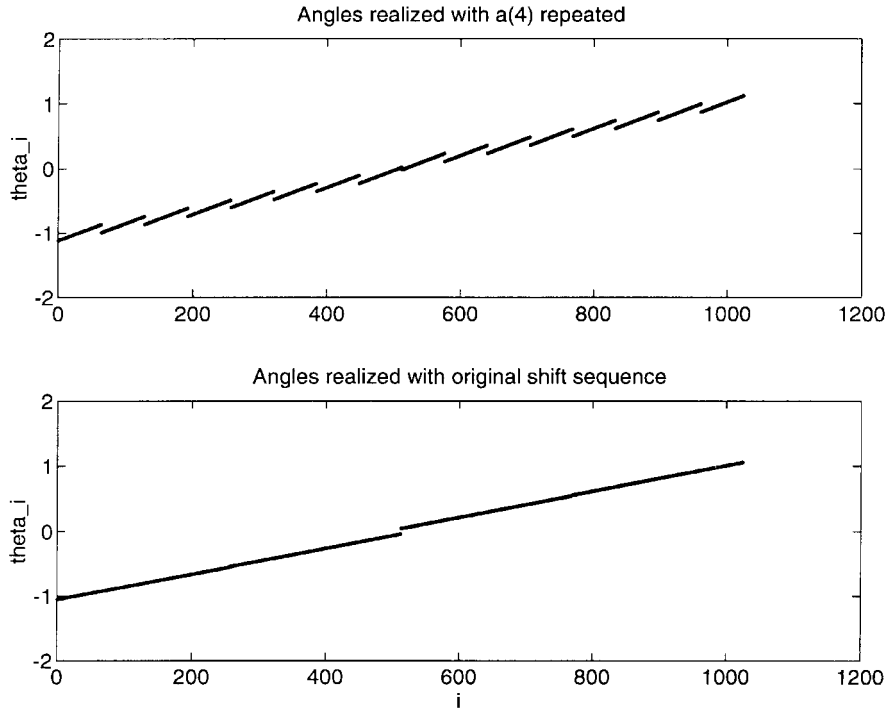

Fig. 8. Comparison between original and modified elementary angle sequences ($n = 10$).

square output prediction error $\mathrm{E}\{J\}$ is minimized if [cf., (32)]

$$\tanh 2\theta_p = \tanh 2\theta^* = \frac{\mathrm{E}\{2f_{p-1}(t)\cdot b_{p-1}(t-1)\}}{\mathrm{E}\{f_{p-1}^2(t)+b_{p-1}^2(t-1)\}} = \hat{k}_p = \rho.$$

Furthermore, we have argued in Corollary 4 that $\Pr\{\theta(t) = \theta^*\} > \Pr\{\theta(t) \neq \theta^*\}$. Thus, with the CALF algorithm, on convergence, the angle $\theta(t)$ is more likely to be at the minimum mean square solution θ^* than other rotation angles. Finally, we note that the mean square output prediction error, on convergence of the CALF algorithm, can be expressed as

$$\mathrm{E}\{J\} = \sum_{i=0}^{N-1} \pi_i \mathrm{E}\{J|\theta_p = \theta_i\} \geq \mathrm{E}\{J|\theta_p = \theta^*\} = J_{ms}.$$

The amount $J_{ex} = \mathrm{E}\{J\} - J_{ms}$ is the *excessive mean square error*, which leads to the definition of *misadjustment* as J_{ex}/J_{ms}.

V. Conclusion

Under the assumption that the input is a wide-sense stationary normal random process, we established the convergence of the CALF algorithm as the convergence of the state probability distribution of a regular Markov chain toward its equilibrium probability distribution. We link the asymptotic convergence rate to the magnitude of the second largest eigenvalue of the state probability transition matrix. These results coincide very well to those obtained empirically using simulations. Future research will focus on studying the convergence properties when the modified shift sequence is used, the effects of adaptively changing step size to expedite convergence, and a comparison of the performance of CALF with other adaptive lattice filtering algorithms not based on a CORDIC processor implementation.

References

[1] J. D. Bruguera, E. Antelo, and E. L. Zapata, "Design of a pipelined radix-4 CORDIC processor," *J. Parallel Comput.*, vol. 19, no. 7, pp. 729–744, 1993.

[2] J. Bu, E. F. A. Deprettere, and F. De-Lange, "On the optimization of pipelined silicon CORDIC algorithm," in *Proc. EUSIPCO Signal Process. III: Theories Applicat.*, The Hague, The Netherlands, 1986, pp. 1227–1230.

[3] J. P. Burg, *Maximum Entropy Spectral Analysis*, Ph.D. dissertation, Dept. Geophys., Stanford Univ., Stanford, CA, 1975.

[4] S. G. Chen and J.-F. Lin, "Efficient implementation of the normalized recursive least square lattice fitler," in *Proc. ICASSP*, Toronto, Ont., Canada, 1991, pp. 1565–1568.

[5] E. F. Deprettere, "Synthesis and fixed point implementation of pipelined true orthogonal filters," in *Proc. ICASSP*, Boston, MA, 1983, pp. 217–220.

[6] M. P. Desai and V. B. Rao, "On the convergence of reversible Markov chains," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 4, pp. 950–966, 1993.

[7] P. Dewilde, E. F. Deprettere, and R. Nouta, "Parallel and pipelined VLSI Implementation of signal processing algorithms," in *VLSI and Modern Signal Processing*, S. Y. Kung *et al.*, Ed. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1985.

[8] B. Friedlander, "Lattice filtering for adaptive signal processing," *Proc. IEEE*, vol. 70, pp. 829–867, 1982.

[9] L. J. Griffiths, "A continuously adaptive filter implemented as a lattice structure," in *Proc. ICASSP*, 1977, pp. 683–686.

[10] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Hingham, MA: Kluwer, 1984.

[11] Y. H. Hu and H. E. Liao, "CALF: A CORDIC adaptive lattice filter I," *IEEE Trans. Signal Processing*, vol. 40, pp. 990–993, 1992.

[12] Y. H. Hu, "CORDIC-based VLSI architecture for digital signal processing," *IEEE Signal Processing Mag.*, vol. 9, pp. 16–35, July 1992.

[13] ———, "A forward angle recoding CORDIC algorithm and pipelined processor array structure for digital signal processing," *Dig. Signal Process.—Rev. J.*, vol. 3, no. 1, pp. 2–15, 1993.

[14] Y. Iiguni, H. Sakai, and H. Tokumaru, "Convergence properties of simplified gradient adaptive lattice algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1427–1434, 1985.

[15] I. Jou, T. Sung, Y. Hu, and T. Parng, "A CORDIC implementation of pipelined Toeplitz system solver," in *Proc. Int. Symp. Circuits Syst.*, Kyoto, Japan, 1985.

[16] I. C. Jou, Y. H. Hu, and W. S. Feng, "A novel implementation of pipelined Toeplitz system solver," *Proc. IEEE*, vol. 74, pp. 1463–1464, 1986.

[17] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. New York: Springer-Verlag, 1976.

[18] J. Makhoul and R. Viswanathan, "Adaptive lattice methods for linear prediction," in *Proc. ICASSP*, 1978, pp. 83–86.

[19] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1965.

[20] J. E. Volder, "The CORDIC Trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, no. 3, pp. 330–334, 1959.

[21] J. W. Walther, "Unified algorithm for elementary functions," in *Proc. Spring Joint Comput. Conf.*, 1971, pp. 379–385.

Yu Hen Hu (SM'87) received the B.S.E.E. degree from National Taiwan University, Taipei, in 1976. He received the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1980 and 1982, respectively.

From 1983 to 1987, he was an Assistant Professor with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX. From 1987 to 1989, he was an Assistant Professor with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison. He is currently an Associate Professor. His research interests include multimedia signal processing, artificial neural networks, fast algorithms and design methodology for application specific micro-architectures, as well as computer-aided design tools for VLSI using artificial intelligence. He has published more than 150 journal and conference papers in these areas.

He was an Associate Editor of the IEEE Transactions on Acoustics, Speech, and Signal Processing in the areas of system identification and fast algorithms from 1988 to 1990. He is currently Associate Editor of the *Journal of VLSI Signal Processing*. He is a founding member of the neural network signal processing technical committee of the IEEE Signal Processing Society. He served as Chair of that committee from 1993 to 1996. He is a former member of VLSI signal processing technical committee of the Signal Processing Society. Currently, he serves as the Secretary of the same society.