[2] B. Liu, "Effect of finite word length on the accuracy of digital filters—A Review," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 670–677, Nov. 1971.
[3] M. J. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, pp. 434–441, June 1986.
[4] B. Usevitch and M. Orchard, "Smooth wavelets, transform coding, and Markov-1 processes," *IEEE Trans. Signal Processing*, vol. 42, pp. 2561–2569, Nov. 1995.
[5] C. W. Barnes, B. N. Tran, and S. H. Leung, "On the statistics of fixed-point roundoff error," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, pp. 595–606, June 1985.

On the Properties of the Reduction-by-Composition LMS Algorithm

Sau-Gee Chen, Yung-An Kao, and Ching-Yeu Chen

***Abstract*— The recently proposed low-complexity reduction-by-composition least-mean-square (LMS) algorithm (RCLMS) costs only half multiplications compared to that of the conventional direct-form LMS algorithm (DLMS). This work intends to characterize its properties and conditions for mean and mean-square convergence. Closed-form mean-square error (MSE) as a function of the LMS step-size μ and an extra compensation step-size α are derived, which are slightly larger than that of the DLMS algorithm. It is shown, when μ is small enough and α is properly chosen, the RCLMS algorithm has comparable performance to that of the DLMS algorithm. Simple working rules and ranges for α and μ to make such comparability are provided. For the algorithm to converge, a tight bound for α is also derived. The derived properties and conditions are verified by simulations.**

***Index Terms*— Adaptive signal processing, convergence, LMS algorithm.**

I. Introduction

The direct-form least-mean-square (DLMS) algorithm is the most popular temporal-domain adaptive filtering algorithm due to its simplicity and robustness. Regarding the temporal-domain approaches, there exist many least-mean-square (LMS) variants in reducing the coefficient update complexities such as the well-known sign-error, sign-input, and zero-forcing algorithms.

However, few improvements were done in reducing its filtering complexities. Recently, a so-called fast exact LMS (FELMS) algorithm [4] was proposed to retain the same convergence properties as those of DLMS, while reducing the multiplication complexities of both filtering and updating complexities by as much as 25%, with a small increase in the number of additions.

More recently, Chen *et al.* proposed a new reduction-by-composition LMS (RCLMS) adaptive filtering algorithm [1]. The algorithm was simulated to have comparable performance to that of the DLMS algorithm, while costing 50% fewer multiplications at the expense of 50% more additions than the DLMS algorithm. The algorithm can be combined with the FELMS algorithm in reducing its coefficient update complexity. However, so far, the algorithm's properties have not been fully addressed.

Here, the properties of the convergence, both in the mean and in the mean square, are investigated in detail, verified by simulations. It is shown, when the common step-size μ is very small and an extra compensation step-size α is properly chosen, the RCLMS algorithm has comparable performance to that of the DLMS algorithm. Due to the extra step constant α, the excess mean-square error (MSE) is shown to be slightly higher than that of the DLMS algorithm for zero-mean input signal. The excess MSE is proportional to α. Also, it is shown that the allowable bound for the step-size α is a function of the step-size μ. Specifically, the larger the step-size μ is, the narrower the bound is for the step-size α.

The paper is organized as follows. In Section II, the RCLMS algorithm will be reviewed, followed by its stability analysis in the third section. Section III covers the issues of weight convergence in mean and mean-square senses, convergence bound for α, and excess MSE. This section also suggests simple working rules for RCLMS algorithm, leading to a comparable performance to the DLMS algorithm. The derived properties, bounds as well as working rules, are verified with the simulations in the Section IV. The final section draws a conclusion.

II. The RCLMS Algorithm

For real-number systems, given an adaptive filter with input sequence $\mathbf{x}(n)$ and coefficients $w_k(n)$'s, the RCLMS algorithm is described below. For the filtering part

$$\begin{aligned} y(n) &= \sum_{k=0}^{N-1} w_k(n)x(n-k) \\ &= \sum_{k=0}^{N/2-1} \{[x(n-2k)+w_{2k+1}(n)] \\ &\quad \cdot [x(n-2k-1)+w_{2k}(n)]\} - C(n) - P(n) \end{aligned} \tag{1}$$

where

$$C(n) = \sum_{k=0}^{N/2-1} w_{2k}(n)w_{2k+1}(n) \tag{2}$$

$$\begin{aligned} P(n) &= \sum_{k=0}^{N/2-1} x(n-2k)x(n-2k-1) \\ &= P(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1). \end{aligned} \tag{3}$$

N is an even number, and $x(n) = 0$, $P(n) = 0$ for $n < 0$. Note that $P(n)$ only costs one multiplication and two additions. The time-varying complicated $C(n)$ can be replaced by a simpler scalar $h_N(n)$ as follows, which costs only one extra multiplication, as depicted in (6). Therefore, for the filtering part

$$\begin{aligned} y'(n) &= \sum_{k=0}^{N/2-1} \{[x(n-2k)+w_{2k+1}(n)] \\ &\quad \cdot [x(n-2k-1)+w_{2k}(n)]\} - h_N(n) - P(n) \\ &= y(n) - [h_N(n) - C(n)]. \end{aligned} \tag{4}$$

For the weight update part

$$w_k(n+1) = w_k(n) + \mu e'(n)x(n-k), \quad k = 0, 1, \cdots, N-1 \tag{5}$$

Manuscript received December 10, 1998; revised August 1999. This work was supported by the National Science Council, Republic of China, under Grant NSC84-2213-E009-083. This paper was recommended by Associate Editor P. Diniz.

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier S 1057-7130(99)09219-8.

$$h_N(n+1) = h_N(n) - \alpha e'(n) \tag{6}$$

where the error signal $e'(n)$ is

$$\begin{aligned} e'(n) &= d(n) - y'(n) \\ &= d(n) - y(n) + [h_N(n) - C(n)] \\ &= e(n) + [h_N(n) - C(n)] \end{aligned} \tag{7}$$

and $d(n)$ is the desired signal.

III. Stability Analysis of the Algorithm

To discuss the stability of an adaptive algorithm, there are two key considerations here [2]: 1) ***Convergence in the mean***, which means that the expectations of the weight vector $\mathbf{w}(n)$ and $h_N(n)$ approach the optimal (Wiener) solutions $\mathbf{w}^*$ and h_N^*, respectively, as the number n of iterations approaches infinity and 2) ***Convergence in the mean square***, which means that the final (steady-state) MSE is finite.

For convenience, some definitions and notations are defined as follows:

$$\begin{aligned} \mathbf{w}(n) &= [w_0(n),\, w_1(n),\, \cdots,\, w_{N-2}(n),\, w_{N-1}(n)]^T_{1\times N} \\ \mathbf{w}^* &= [w_0^*,\, w_1^*,\, \cdots,\, w_{N-1}^*]^T_{1\times N} \\ \xi_{\mathbf{w}}(n) &= \mathbf{w}(n) - \mathbf{w}^* \\ \mathbf{x}(n) &= [x(n),\, x(n-1),\, \cdots,\, x(n-N+2)x(n-N+1)]^T_{1\times N} \end{aligned} \tag{8}$$

where $\mathbf{w}(n) =$ the tap weight vector, $\mathbf{x}(n) =$ the tap input vector.

Here, both $x(n)$ and $d(n)$ are assumed zero-mean and wide-sense stationary. It is also emphasized that the following independence assumption similar to those of [2] and [3] is made.

1) The input vectors $\mathbf{x}(1),\, \mathbf{x}(2),\, \cdots,\, \mathbf{x}(n)$ constitute a sequence of statistically independent vectors.
2) $\mathbf{x}(n)$ is statistically independent of all previous samples of the desired response, namely $d(1),\, d(2),\, \cdots,\, d(n-1)$.
3) $d(n)$ is dependent on the corresponding $\mathbf{x}(n)$, but statistically independent of all previous samples of the desired response.
4) $\mathbf{x}(n)$ and $d(n)$ consist of mutually Gaussian-distributed random variables for all n.

Based on the assumption, the conditions hold: $E[\mathbf{x}(n)d(k)] = \mathbf{0}$, $E[\mathbf{x}(n)\mathbf{x}^T(k)] = \mathbf{0},\, k = 0,\, 1,\, \cdots,\, n-1$. Similarly, $h_N(n+1),\, \mathbf{w}(n+1)$ can be easily shown to be independent of both $\mathbf{x}(n+1)$ and $d(n+1)$, but $\mathbf{x}(n),\, \mathbf{x}(n-1), \cdots, \mathbf{x}(1),\, d(n),\, d(n-1),\, \cdots,\, d(1),\, \mathbf{w}(0)$ and $h_N(0)$.

1) The Convergence of $E[w(n)]$: By subtracting $\mathbf{w}^*$ from both sides of (5), followed by taking its expectation, we have

$$\begin{aligned} E[\xi_{\mathbf{w}}(n+1)] &= E[(\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))\xi_{\mathbf{w}}(n) \\ &\quad + \mu\mathbf{x}(n)e_{\mathrm{opt}}(n) + \mu\mathbf{x}(n)(h_N(n) - C(n))] \\ &= (\mathbf{I} - \mu\mathbf{R})E[\xi_{\mathbf{w}}(n)] \end{aligned} \tag{9}$$

where $e_{\mathrm{opt}}(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}^*$, and $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$. Here, the simplification is achieved by applying the independence assumption and orthogonality principle, i.e., $E[\mathbf{x}(n)e_{\mathrm{opt}}(n)] = \mathbf{0}$. Since $\mathbf{w}(n)$ [and hence $\xi_{\mathbf{w}}(n)$] and $h_N(n)$ [and accordingly $h_N(n) - C_(n)$] are independent of $\mathbf{x}(n)$ and $d(n)$, $E[x(n)(h_N(n) - C(n))] = E[x(n)]E[h_N(n) - C(n)] = 0$. Equation (9) is the same as that for DLMS algorithm, so is the required constraint for step size μ, i.e., $0 < \mu < 2/\lambda_{\max}$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}$.

2) The Convergence of $E[h_N(n)]$: First, we can easily show that the optimal MSE $h_N^*(n)$ of $h_N(n)$ is $h_N^*(n) = E[C(n)] = C_{\mathrm{opt}} \equiv \sum_{k=0}^{N/2-1} w_{2k}^* w_{2k+1}^*$, by setting the derivative of $J'(n) \equiv E[e'^2(n)]$ with respect to $h_N(n)$ to zero. To discuss the convergence of $E[h_N(n)]$, we take the expectation of (6) as follows:

$$\begin{aligned} E[h_N(n+1)] &= E[h_N(n)] - \alpha E[e'(n)] \\ &= E[h_N(n)] - \alpha E[h_N(n) - C(n)]. \end{aligned} \tag{10}$$

As n approaches infinity

$$E[h_N(\infty)] = E[h_N(\infty)] - \alpha E[h_N(\infty) - C(\infty)]. \tag{11}$$

Consequently, $E[h_N(\infty)] = E[C(\infty)] \equiv h_N^*$.

In a more rigorous way, it can be also shown that $E[h_N(\infty)] = E[C(\infty)]$. By subtracting $C(n+1)$ from both sides of (6), and assuming that $C(n+1) \approx C(n)$ for a very small μ and correspondingly a slowly varying $C(n)$, one has

$$\begin{aligned} &h_N(n+1) - C(n+1) \\ &\quad \approx h_N(n) - C(n) - \alpha[d(n) - \mathbf{w}^T(n)\mathbf{x}(n) + h_N(n) - C(n)]. \end{aligned} \tag{12}$$

By defining $\xi_{\mathrm{hc}}(n) \equiv h_N(n) - C(n)$, the expectation of (12) can be reduced to

$$\begin{aligned} E[\xi_{\mathrm{hc}}(n+1)] &= E[\xi_{\mathrm{hc}}(n)] - \alpha E[d(n) - \mathbf{w}^T(n)\mathbf{x}(n) + \xi_{\mathrm{hc}}(n)] \\ &= (1-\alpha)E[\xi_{\mathrm{hc}}(n)] = (1-\alpha)^{n+1}E[\xi_{\mathrm{hc}}(0)]. \end{aligned} \tag{13}$$

Therefore, under the condition $|1-\alpha| < 1$, $E[\xi_{\mathrm{hc}}(n)] = 0$, and $E[h_N(n)] \approx E[C(n)]$ as $n \to \infty$.

3) The Convergence in the Mean Square: The MSE $J'(n)$ can be shown to be

$$\begin{aligned} J'(n) &= E[(e'(n))^2] \\ &= E[e^2(n)] + E[\xi_{\mathrm{hc}}^2(n)] + 2E[(d(n) - \mathbf{w}^T(n)\mathbf{x}(n))\xi_{\mathrm{hc}}(n)] \\ &= J(n) + E[\xi_{\mathrm{hc}}^2(n)] \end{aligned} \tag{14}$$

where $E[e^2(n)]$ is equal to $J(n)$ of the DLMS algorithm. Equation (14) is simplified by applying (7), the independence assumption, and the conditions $E[x(n)] = 0$ and $E[d(n)] = 0$.

To solve $J'(n)$, let us consider $J(n)$ first. $J(n)$ has been shown in [2] to be $J(n) = J_{\min} + J_{ex}(n)$, where $J_{\min} \equiv E[e_{\mathrm{opt}}^2(n)]$, $J_{ex}(n) \equiv E[\xi_{\mathbf{w}}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\xi_{\mathbf{w}}(n)] = \mathrm{tr}[\mathbf{R}\mathbf{K}(n)]$, $K(n) \equiv E[\xi_{\mathbf{w}}(n)\xi_{\mathbf{w}}^T(n)]$ [2]. In computing $\mathbf{K}(n+1)$, various cross terms arise as a result of the multiplication. Many of them can be discarded, which includes

$$\begin{aligned} &E[\mathbf{x}(n)e_{\mathrm{opt}}(n)\xi_{\mathbf{w}}^T(n)(\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))] \\ &\quad = E[(\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))\xi_{\mathbf{w}}(n)(\mathbf{x}^T(n)e_{\mathrm{opt}}(n))] = \mathbf{0} \end{aligned} \tag{15}$$

$$\begin{aligned} &E[\mathbf{x}(n)e_{\mathrm{opt}}(n)(\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n))^T] \\ &\quad = E[\mu e_{\mathrm{opt}}(n)\mathbf{x}(n)\mathbf{x}^T(n)]E[\xi_{\mathrm{hc}}(n)] = \mathbf{0} \end{aligned} \tag{16a}$$

$$E[\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n)(\mathbf{x}(n)e_{\mathrm{opt}}(n))^T] = \mathbf{0} \tag{16b}$$

$$\begin{aligned} &E[\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n)((\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))\xi_{\mathbf{w}}(n))^T] \\ &\quad = E[(\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))\xi_{\mathbf{w}}(n)(\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n))^T] \approx \mathbf{0}. \end{aligned} \tag{17}$$

The reasons leading to (15) can be found in [2]. Equations (16a) and (16b) are due to the independence assumption and $E[\xi_{\mathrm{hc}}(n)] = E\{h_N(n) - C_{\mathrm{opt}} - [C(n) - C_{\mathrm{opt}}]\} = 0$, while (17) is assumed negligible when α and μ are small enough, i.e., $0 < \alpha \ll 1/(2N)$ and $0 < \mu \ll 1/(2N\sigma_x^2)$. Appendix A details the derivation steps leading to the constraints. As a result (with the help of the Gaussian moment factoring theorem [2])

$$\begin{aligned} \mathbf{K}(n+1) = {} & (\mathbf{I} - \mu\mathbf{R})\mathbf{K}(n)(\mathbf{I} - \mu\mathbf{R}) + \mu^2\mathbf{R}\,\mathrm{tr}[\mathbf{R}\mathbf{K}(n)] \\ & + \mu^2\mathbf{R}\mathbf{K}(n)\mathbf{R} + \mu^2\mathbf{R}J_{\min} + \mu^2\mathbf{R}E[\xi_{\mathrm{hc}}^2(n)]. \end{aligned} \tag{18}$$

Furthermore, let $\mathbf{U}(n)$ be defined as $\mathbf{U}(n) \equiv \mathbf{Q}^{\boldsymbol{T}}\mathbf{K}(n)\mathbf{Q}$, then $\mathbf{K}(n) = \mathbf{Q}\mathbf{U}(n)Q^{\boldsymbol{T}}$, where $\mathbf{Q}$ is the eigenvector matrix of $\mathbf{R}$, $Q^{\boldsymbol{T}}\mathbf{R}\mathbf{Q} = \Lambda$ and Λ is a diagonal matrix consisting of the eigenvalues λ_i of $\mathbf{R}$. Consequently, one may rewrite (18) as

$$\begin{aligned}\mathbf{U}(n+1) = {} & (\mathbf{I}-\mu\Lambda)\mathbf{U}(n)(\mathbf{I}-\mu\Lambda) + \mu^2\Lambda\,\mathrm{tr}[\Lambda\mathbf{U}(n)] \\ & + \mu^2\Lambda\mathbf{U}(n)\Lambda + \mu^2 J_{\min}\Lambda + \mu^2\Lambda E[\xi_{\mathrm{hc}}^2(n)]. \end{aligned} \tag{19}$$

First, let us consider the convergence of the off-diagonal element $u_{ij}(n)$ of $\mathbf{U}(n)$ where $i \neq j$. From (19)

$$\begin{aligned} u_{ij}(n+1) &= [(1-\mu\lambda_i)(1-\mu\lambda_j) + \mu^2\lambda_i\lambda_j]u_{ij}(n) \\ &= [(1-\mu\lambda_i)(1-\mu\lambda_j) + \mu^2\lambda_i\lambda_j]^{n+1}u_{ij}(0). \end{aligned} \tag{20}$$

Equation (20) will approach to zero as $n \to \infty$ under the condition $(1-\mu\lambda_i)(1-\mu\lambda_j) + \mu^2\lambda_i\lambda_j < 1$, i.e., $0 < \mu < (1/\lambda_i + 1/\lambda_j)/2$. The worst case is $0 < \mu < (1/\lambda_{\max} + 1/\lambda_{\max\,2})/2$, where $\lambda_{\max\,2}$ is the second largest eigenvalue of $\mathbf{R}$. In practice, since $2/\mathrm{tr}[\mathbf{R}] < 2/(\lambda_{\max} + \lambda_{\max\,2}) \leq (1/\lambda_{\max} + 1/\lambda_{\max\,2})/2$, $0 < \mu < 2/\mathrm{tr}[\mathbf{R}]$ is a more feasible bound.

Next, let us discuss the convergence of $E[\xi_{\mathrm{hc}}^2(n)]$ and the diagonal elements $u_i(n)$ of $U(n)$, $i = 0, 1, \cdots, N-1$. It was shown in [2] that $J_{ex}(n) = \sum_{i=0}^{N-1}\lambda_i u_i(n)$. From (12) and the independence assumption, we have

$$\begin{aligned} & E[\xi_{\mathrm{hc}}^2(n+1)] \\ & \quad = E[((1-\alpha)\xi_{\mathrm{hc}}(n) - \alpha(d(n) - \mathbf{w}(n)^T\mathbf{x}(n)))^2] \\ & \quad = (1-\alpha)^2 E[\xi_{\mathrm{hc}}^2(n)] + \alpha^2\left[J_{\min} + \sum_{i=0}^{N-1}\lambda_i u_i(n)\right]. \end{aligned} \tag{21}$$

One may combine (19) and (21) as $\mathbf{v}(n+1) = \mathbf{A}\mathbf{v}(n) + \mathbf{b}J_{\min}$, where

$$\begin{aligned} \mathbf{v}(n) &= [u_0(n),\, u_1(n),\, \cdots,\, u_{N-1}(n),\, E[\xi_{\mathrm{hc}}^2(n)]]^T_{1x(N+1)} \\ \mathbf{b} &= \mu^2[\lambda_0,\, \lambda_1,\, \cdots,\, \lambda_{N-1},\, \alpha^2/\mu^2]^T_{1x(N+1)} \end{aligned}$$

and $\mathbf{A}$ is an $(N+1)\times(N+1)$ matrix with elements

$$a_{ij} = \begin{cases} (1-\mu\lambda_i)^2 + 2\mu^2\lambda_i^2, & i = j = 0, \cdots, N-1 \\ \mu^2\lambda_i\lambda_j, & i \neq j,\ i = 0, \cdots, N-1 \\ & j = 0, \cdots, N-1 \\ \mu^2\lambda_i, & j = N,\ i = 0, \cdots, N-1 \\ \alpha^2\lambda_j & i = N,\ j = 0, \cdots, N-1 \\ (1-\alpha)^2, & i = j = N. \end{cases}$$

Therefore, $\mathbf{v}(n+1) = \mathbf{A}^{n+1}\mathbf{v}(0) + \mathbf{b}J_{\min}\sum_{k=0}^{n}\mathbf{A}^k$.

It guarantees the convergence of $\mathbf{v}(n+1)$ when all the magnitudes of the eigenvalues of $\mathbf{A}$ are smaller than 1. Let c and $\mathbf{g} = [g_0, \cdots, g_N]$ be an eigenvalue and eigenvector pair of $\mathbf{A}$, then $\mathbf{A}\mathbf{g} = c\mathbf{g}$ or equivalently $\sum_{j=0}^{N} a_{ij}g_i = cg_i$. Consequently

$$\begin{aligned} [(1-\mu\lambda_i)^2 + \mu^2\lambda_i^2]g_i + \mu^2\lambda_i\sum_{j=0}^{N-1}\lambda_j g_j + \mu^2\lambda_i g_N \\ = cg_i,\ i = 0, \cdots, N-1 \end{aligned} \tag{22}$$

$$\alpha^2\sum_{j=0}^{N-1}\lambda_j g_j + (1-\alpha)^2 g_N = cg_N. \tag{23}$$

Hence, g_i can be solved as

$$\begin{aligned} g_i = {} & \left(\sum_{j=0}^{N-1}\lambda_j g_j\right)\mu^2\lambda_i(c-1+2\alpha)\Big/ \\ & \Big[(c-(1-\alpha)^2)(c-(1-\mu\lambda_i)^2 - \mu^2\lambda_i^2)\Big], \\ & i = 0, \cdots, N-1. \end{aligned} \tag{24}$$

Moreover, g_i can be eliminated by multiplying both sides of (24) with λ_i and then summing over all i from zero to $N-1$. As a result

$$\frac{\mu^2(c-1+2\alpha)}{c-(1-\alpha)^2}\sum_{i=0}^{N-1}\frac{\lambda_i^2}{c-(1-\mu\lambda_i)^2-\mu^2\lambda_i^2} = 1. \tag{25}$$

Under the critical condition that $c = 1$, one can get the inequality (26), which is the required convergence condition in mean square

$$\frac{1}{2-\alpha}\sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i} < 1. \tag{26}$$

A special case is when $\alpha = 0$, which reduces to DLMS algorithm with the well-known condition $\sum_{i=0}^{N-1}\mu\lambda_i/(1-\mu\lambda_i) < 2$. From (26), one has the constraint (27) for α

$$0 < \alpha < 2 - \sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i}. \tag{27}$$

In summary, the convergence condition for μ is $0 < \mu < (1/\lambda_{\max} + 1/\lambda_{\max\,2})/2$ [5] for both RCLMS and DLMS algorithms, while α of RCLMS algorithm should satisfy (27).

4) The Steady-State MSE: The steady-state component $E[\xi_{\mathrm{hc}}^2(\infty)]$ can be shown to be $E[\xi_{\mathrm{hc}}^2(\infty)] = \alpha J(\infty)/(2-\alpha)$by letting $n \to \infty$ in (21). One then can solve $u_i(\infty)$ by plugging this specific $E[\xi_{\mathrm{hc}}^2(\infty)]$ into (19) as

$$u_i(\infty) = \mu J(\infty)/[(1-\mu\lambda_i)(2-\alpha)]. \tag{28}$$

Equation (28) can be further extended to

$$\begin{aligned} \sum_{i=0}^{N-1}\lambda_i u_i(\infty) &= \sum_{i=0}^{N-1}\mu\lambda_i J(\infty)/[(2-\alpha)(1-\mu\lambda_i)] \\ &= J_{ex}(\infty) \\ &= J(\infty) - J_{\min}. \end{aligned} \tag{29}$$

Hence

$$J(\infty) = J_{\min}\Big/\left\{1 - \sum_{i=0}^{N-1}\mu\lambda_i\Big/[(2-\alpha)(1-\mu\lambda_i)]\right\}. \tag{30}$$

Finally

$$\begin{aligned} J'(\infty) &= J(\infty) + E[\xi_{\mathrm{hc}}^2(\infty)] \\ &= 2J(\infty)/(2-\alpha) \\ &= 2J_{\min}\Big/\left[(2-\alpha) - \sum_{i=0}^{N-1}\mu\lambda_i\Big/(1-\mu\lambda_i)\right]. \end{aligned} \tag{31}$$

Note that when α is sufficiently small, MSE of the RCLMS algorithm is equal to $J(\infty)$ of the DLMS algorithm [2]. On the other hand, for sufficiently small μ, one has $\sum_{i=0}^{N-1}\mu\lambda_i/(1-\mu\lambda_i) \approx 0$, and $J'(\infty) \approx 2J_{\min}/(2-\alpha)$.

5) The Misadjustment: The misadjustment is defined as follows:

$$\begin{aligned} M &\equiv J'_{\mathrm{ex}}(\infty)/J_{\min} \\ &= \left(\alpha + \sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i}\right)\Big/\left(2 - \alpha - \sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i}\right) \end{aligned} \tag{32}$$

where the excess MSE $J'_{ex}(n)$ is defined as $J'_{ex}(n) \equiv J'(n) - J_{\min}$. One may rewrite (32) as

$$\alpha = \frac{2M}{1+M} - \sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i}. \tag{33}$$

Note that $2M/(1+M) < 2$. In practice, M is first prescribed, then α can be confined to a more conservative upper bound than (27), as shown below

$$0 < \alpha < \frac{2M}{1+M} - \sum_{i=0}^{N-1} \frac{\mu\lambda_i}{1-\mu\lambda_i}. \tag{34}$$

When α is sufficiently small, (32) reduces to a form similar to that of the DLMS algorithm [2]. On the other hand, when μ is sufficiently small, $M \approx \alpha/(2-\alpha)$.

6) Simple Working Rules: Based on the derived properties, one can conclude the following simple working rules that result in a comparable performance of RCLMS algorithm to that of the DLMS algorithm (in terms of speed and MSE).

1) Pick μ subject to the following constraint:

$$\begin{aligned} 0 < \mu &< \min\left\{\frac{2}{\text{tr}[\mathbf{R}]}, \frac{2M}{\text{tr}[\mathbf{R}](1+M)}, \frac{1}{10\text{tr}[\mathbf{R}]}\right\} \\ &= \min\left\{\frac{2M}{\text{tr}[\mathbf{R}](1+M)}, \frac{1}{10\text{tr}[\mathbf{R}]}\right\}. \end{aligned} \tag{35}$$

This constraint is for the stability requirement of RCLMS algorithm. The term $2/\text{tr}[\mathbf{R}]$ follows directly from the well-known bound [2], [3] for DLMS algorithm. The term $2M/[\text{tr}[\mathbf{R}](1+M)]$ follows from the requirement that $1-\mu\lambda_i \approx 1$, and for example $\mu\lambda_i < 0.1$. The condition, in turn, implies the third constraint of $\mu < 1/(10\text{tr}[\mathbf{R}]) < 1/(10\lambda_{\max}) \le 1/(10\lambda_i)$.

2) Pick α subject to the following constraint:

$$\frac{\mu\cdot\text{tr}[\mathbf{R}]}{N} < \alpha < \min\left\{\frac{2M}{(1+M)} - \mu\cdot\text{tr}[\mathbf{R}],\ 2 - \frac{\mu\cdot\text{tr}[\mathbf{R}]}{N}\right\} \tag{36}$$

where the lower bound $\mu\text{tr}[\mathbf{R}]/N$ and one of the upper bounds $2-\mu\text{tr}[\mathbf{R}]/N$ follow from the obvious reason that RCLMS algorithm should be tuned to converge as fast as the DLMS algorithm. More precisely, the constraint $|1-\alpha| < |1-\mu\lambda_{\min}| < 1$ should be satisfied. Since $\mu\lambda_{\min} < 1$ according to (35), the speed constraint reduces to $\mu\lambda_{\min} - 1 < 1-\alpha < 1-\mu\lambda_{\min} < 1$, which leads to $0 < \mu\lambda_{\min} < \alpha < 2-\mu\lambda_{\min} < 2$. Or, more conservatively

$$0 < \mu\lambda_{\min} < \frac{\mu\cdot\text{tr}[\mathbf{R}]}{N} < \alpha < 2 - \frac{\mu\cdot\text{tr}[\mathbf{R}]}{N} < 2-\mu\lambda_{\min} < 2. \tag{37}$$

The other upper bound $2M/(1+M) - \mu\ \text{tr}[\mathbf{R}]$ of (36) is resulted from (34) under the condition of a desired M smaller than the prescribed one and a small μ. Specifically, when (35) holds, (34) can be reduced to more conservative bounds gradually, as follows:

$$\begin{aligned} \alpha &< \frac{2M}{(1+M)} - \mu\cdot\text{tr}[\mathbf{R}] \\ &= \frac{2M}{(1+M)} - \mu\sum_{i=0}^{N-1}\lambda_i \\ &< \frac{2M}{1+M} - \sum_{i=0}^{N-1}\frac{\mu\lambda_i}{1-\mu\lambda_i}. \end{aligned} \tag{38}$$

IV. Simulations

This section verifies the derived properties and simple working rules by simulations. Here equalizer design is tried. The assumed impulse response of the channel is $\boldsymbol{c} = [-1/2.1,\ 1,\ 1/2.1]$ with a white Gaussian, zero-mean input signal of variance = 1. The tap number of the adaptive equalizer is prescribed to 4.

Fig. 1 shows the steady-state MSE's (average of 500 runs) as a function of α considering small $\mu = 0.0001$ and large $\mu = 0.03$. As shown, the simulation curves are very close to the derived theoretical MSE curves of (31), especially for the cases of small μ.

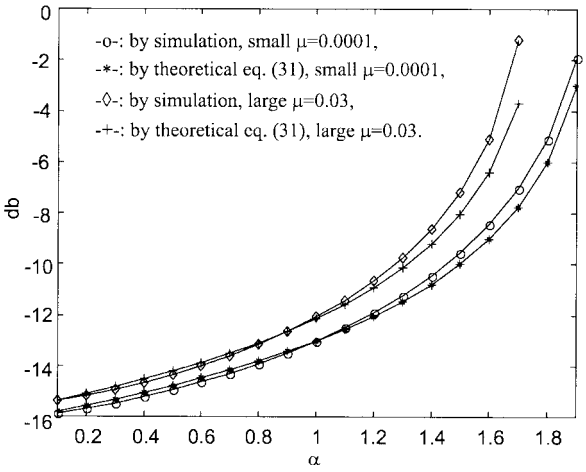

Fig. 1. The steady-state MSE's $J'(\infty)$ versus α, with small $\mu = 0.0001$ and large $\mu = 0.03$.

TABLE I
KEY VALUES OF THE DLMS AND RCLMS ALGORITHMS CORRESPONDING TO $M = 0.09$: MSE'S, THE THEORETICAL BOUNDS AND PICKED VALUES FOR α AND μ

prescribed M	upper bound of (35) for μ	upper bound of (36) for α	lower bound of (36) for α	picked α	picked μ	MSE (db) DLMS	MSE (db) (RCLMS)
0.09	0.0172	0.1070	0.0145	0.1	0.01	-15.89	-15.73

In the large μ case, although there is a more noticeable deviation between the theoretical and simulation results (as expected) than the small μ, the theoretical curve still follows the simulation curve closely. The simulated convergence bound for α exceeds 1.99 (with $\mu = 0.0001$), which is extremely close to the theoretical upper bound 1.999 predicted by (27). On the other hand, the simulations diverged when $\alpha \ge 1.8$ (with $\mu = 0.03$), which is still very close to theoretical value of $\alpha = 1.8174$ from (27). In practice, it is suggested that (36) instead of (27) be used for the upper bound of α.

Next is the verification of the simple working rules suggested in the previous section. With the same equalizer design problem as before and a prescribed $M = 0.09$, Table I lists the values of theoretical upper bound and lower bound for α and μ that could make up a RCLMS algorithm comparable (in speed and MSE) to that of the DLMS algorithm. In the following simulations, assuming an in-bound $\mu = 0.01$, various values of α in and out of the bound are simulated and compared.

Fig. 2 shows the MSE's (average of 500 runs) of DLMS algorithm and the RCLMS algorithm with $\mu = 0.01$ for various α values. As expected, the convergence rate of RCLMS algorithm for $\alpha = 0.001$ is slower than that of the DLMS algorithm, because α is smaller than the lower bound of (36) in Table I. However, also as expected, its steady-state MSE is close to that of the DLMS algorithm. On the other hand, the curves of $\alpha = 0.1$ and the DLMS algorithm have the same convergence speed and comparable MSE's of -15.73 and -15.89 dB, respectively, as shown in Fig. 2 and Table I. This is because both α and μ are in the working ranges listed in Table I. In cases of larger $\alpha = 1$ and 1.5, they are outside the working range. They are also outside the upper bound of α derived in Appendix A for (17) to be negligible. As predicted, these α values result in larger MSE's than those of the DLMS algorithm. One can notice these somewhat peculiar learning curves. They first drop to their minimum values like the DLMS algorithm and then continue

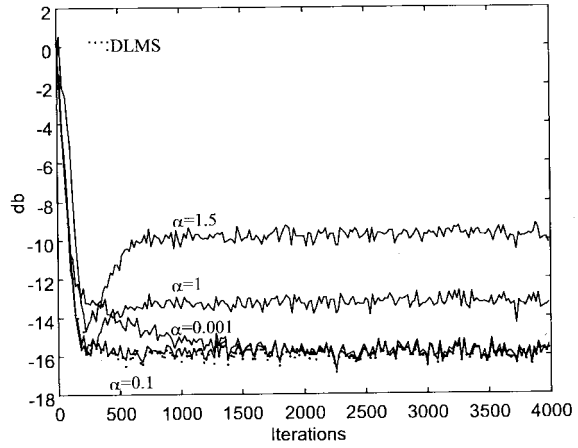

Fig. 2. The MSE's of DLMS and RCLMS algorithm for various α with $\mu = 0.01$.

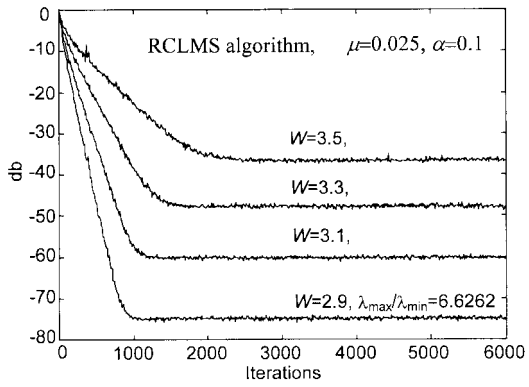

Fig. 3. The MSE curves of the RCLMS algorithm under different eigenvalue spreads.

to raise to higher steady-state MSE's. For these two α's, many neglected terms [especially (17)] in the derivation steps are no longer negligible.

The accuracy of the derived properties were confirmed by the mentioned simulations with some small disparities (particularly for large α) as shown in Figs. 1 and 2. One of the reasons for those deviations could be explained by the so called *shifting properties* [2] as follows. Note that all the previously derived theories are based on the independence assumption. However, the *shifting property* of input data could introduce statistically dependent results [2]. It might result in $E[\mathbf{x}(n)(h_N(n) - C(n))] \neq \mathbf{0}$, even when $\mathbf{x}(n)$ is a zero-mean signal. That means $E[\mathbf{x}(n)(h_N(n) - C(n))]$ would converge to a dc vector in the long run. Accordingly, each converged weight is equal to the sum of its Wiener solution and a dc bias. The magnitude of dc bias is found to directly proportion to α from the simulation results. The dc bias could be an another reason why there are peculiar MSE curves as in Fig. 2, especially when α is large and outside the working range and/or near the upper bound of stability. All the derivations are under the conditions that α and/or μ are small. Therefore, in cases of large α and/or μ, one may expect noticeable (but minor) deviation between theory and the simulation results as shown in Fig. 1.

The last example demonstrates the effects due to eigenvalue spreads. Here, an equalizer is designed to compensate a raised-cosine channel $h(n)$, where $h(n) = (1 + \cos(2\pi(n-2)/W))/2$, $n = 1, 2, 3$. The input signal is white Gaussian, zero-mean with variance $= 1$. The parameter W controls the eigenvalue spread. In the example, the tap number is set to 12, and $\mu = 0.025$, $\alpha = 0.1$. Fig. 3 shows the MSE curves of RCLMS algorithm under various eigenvalue spreads. As expected, a larger eigenvalue spread results in a slower convergence and larger MSE, and vice versa. Since the curves of DLMS virtually overlap with those of RCLMS algorithm, we do not include them for comparison. Regarding practical application examples of RCLMS algorithm, it has been successfully applied to HDSL equalizer [6] and cable modem [7] designs.

V. Conclusion

The properties of RCLMS algorithm have been characterized in the paper. The simulation results match the derived properties closely. Simple working rules and proper bounds for α and μ are also given to facilitate the new algorithm's practical usage. In the theoretical analysis, small μ is assumed. The future work is to analyze its properties under large μ and α. In addition, practical applications of RCLMS algorithm deserve further investigation.

Appendix A
Derivation of the Conditions $0 < \alpha \ll 1/(2N)$ and $0 < \mu \ll 1/(2N\sigma_x^2)$ for (17) to be Negligible

From independence assumption, (17) can be further reduced to

$$\begin{aligned} &E[(\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^T(n))\xi_{\mathbf{w}}(n)(\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n))^T] \\ &\quad = E[(\xi_{\mathbf{w}}(n)(\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n))^T] \\ &\qquad - E[\mu\mathbf{x}(n)\mathbf{x}^T(n)\xi_{\mathbf{w}}(n)(\mu\mathbf{x}(n)\xi_{\mathrm{hc}}(n))^T] \\ &\quad = \mu E[(\xi_{\mathbf{w}}(n)\xi_{\mathrm{hc}}(n)]E[(\mathbf{x}^T(n)] \\ &\qquad - \mu^2 E[\xi_{\mathrm{hc}}(n)\mathbf{x}(n)\mathbf{x}^T(n)\xi_{\mathbf{w}}(n)\mathbf{x}^T(n)] \\ &\quad = -\mu^2 E[\xi_{\mathrm{hc}}(n)\mathbf{x}(n)\mathbf{x}^T(n)\xi_{\mathbf{w}}(n)\mathbf{x}^T(n)]. \end{aligned}$$

On convergence one can reasonably assume that the weight differences from their optimal values are roughly equal to the weight correction terms, i.e.,

$$\xi_{\mathbf{w}}(n) = \mathbf{w}(n) - \mathbf{w}^* \approx \mathbf{w}(n) - \mathbf{w}(n-1) = \mu e'(n-1)x(n-1)$$

and similarly $\xi_{\mathrm{hc}}(n) \approx -\alpha e'(n-1)$. Hence

$$\begin{aligned} &-\mu^2 E[\xi_{\mathrm{hc}}(n)\mathbf{x}(n)\mathbf{x}^T(n)\xi_{\mathbf{w}}(n)\mathbf{x}^T(n)] \\ &\quad \approx \mu^3\alpha E\{[e'(n-1)]^2\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{x}(n-1)\mathbf{x}^T(n)]\}. \end{aligned}$$

It is easily seen that

$$\begin{aligned} &\mu^3\alpha E\{[e'(n-1)]^2\}\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{x}(n-1)\mathbf{x}^T(n)] \\ &\quad \le \mu^3\alpha E\{\mathbf{x}(n)\mathbf{x}^T(n)\}E\{[e'(n-1)]^2\}E\{\mathbf{x}^T(n)\mathbf{x}(n)\} \\ &\quad = \mu^3\alpha\mathbf{R}J'(n)\sum_{i=0}^{N-1}\lambda_i. \end{aligned}$$

Note that this is a very pessimistic bound, because we have replaced $\mathbf{x}^T(n)\mathbf{x}(n-1)$ by $\mathbf{x}^T(n)\mathbf{x}(n)$ in this inequality. For zero-mean input sequence, especially a white input signal $\mathbf{x}^T(n)\mathbf{x}(n-1) \approx 0$. Next, let us transform $\mathbf{R}$ and $\mathbf{K}(n)$ to $\Lambda = \mathbf{Q^T}\mathbf{R}\mathbf{Q}$ and $\mathbf{U}(n) = \mathbf{Q^T}\mathbf{K}(n)\mathbf{Q}$, respectively. As such, $\mu^3\alpha\mathbf{R}J'(n)\sum_{i=0}^{N-1}\lambda_i$ is transformed to $\mu^3\alpha\Lambda J'(n)\sum_{i=0}^{N-1}\lambda_i$, which has a maximum

element of $\mu^3\alpha\lambda_{\max}J'(n)\sum_{i=0}^{N-1}\lambda_i$. To decide the constraints for α and μ for (17) to be negligible, we can then compare the maximum element with the maximum elements of all the other matrices in (19) including $(\mathbf{I}-\mu\Lambda)\mathbf{U}(n)(\mathbf{I}-\mu\Lambda)$, $\mu^2\Lambda\mathrm{tr}[\Lambda\mathbf{U}(\infty)]$, $\mu^2\Lambda\mathbf{U}(\infty)\Lambda$, $\mu^2J_{\min}\Lambda$, and $\mu^2\Lambda E[\xi_{\mathrm{hc}}^2(\infty)]$. Obviously, $(\mathbf{I}-\mu\Lambda)\mathbf{U}(n)(\mathbf{I}-\mu\Lambda)$ is the most significant term; therefore, we only need to compare the remaining terms. Note that we have proved that all the off-diagonal elements of $\mathbf{U}(n)$ converge to zero [assuming a negligible (17)]. Therefore, we can ignore all the off-diagonal elements in the following transformed matrices. Now, let us take a look at the maximum diagonal elements of all the other terms in (19) assuming that (17) is negligible and then derive the required conditions for α and μ.

1) $\mu^2\Lambda\,\mathrm{tr}[\Lambda\mathbf{U}(\infty)]$ has a maximum element of

$$\begin{aligned}&\mu^2\max\{\Lambda\,\mathrm{tr}[\Lambda\mathbf{U}(\infty)]\}\\&\quad=\mu^2\max\{\Lambda\,\mathrm{tr}[\mathbf{R}\mathbf{K}(\infty)]\}=\mu^2\max\{\Lambda J_{ex}(\infty)\}\\&\quad=\mu^2[J(\infty)-J_{\min}]\lambda_{\max}.\end{aligned}$$

2) $\mu^2\Lambda U(\infty)\Lambda$ has a maximum element of

$$\begin{aligned}\mu^2\max\{\Lambda U(\infty\Lambda)\}&=\mu^2\max\{\lambda_i^2u_i(\infty)\}\\&=\mu^3\lambda_{\max}^2J(\infty)/[(1-\mu\lambda_{\max})(2-\alpha)]\end{aligned}$$

3) $\mu^2J_{\min}\Lambda$ has a maximum element of $\mu^2J_{\min}\max\{\lambda_i\}=\mu^2J_{\min}\lambda_{\max}$.
4) $\mu^2\Lambda E[\xi_{\mathrm{hc}}^2(\infty)]$ has a maximum element of $\mu^2\lambda_{\max}E[\xi_{\mathrm{hc}}^2(\infty)]=\mu^2\lambda_{\max}\alpha J(\infty)/(2-\alpha)$.

For $\mu\ll 1/\lambda_{\max}$ and $\alpha\ll 2$, (30) reduces to $J(\infty)\approx J_{\min}/(1-\mu/2\sum_{i=0}^{N-1}\lambda_i)$. As such, the maximum terms of 1–4 above reduce to $\mu^3J_{\min}\lambda_{\max}\sum_{i=0}^{N-1}\lambda_i/(2-\mu\sum_{i=0}^{N-1}\lambda_i)$, $\mu^3J_{\min}\lambda_{\max}^2/(2-\mu\sum_{i=0}^{N-1}\lambda_i)$, $\mu^2J_{\min}\lambda_{\max}$, and $\mu^2\alpha J_{\min}\lambda_{\max}/(2-\mu\sum_{i=0}^{N-1}\lambda_i)$, respectively. And utilizing (31), $\mu^3\alpha\lambda_{\max}J'(\infty)\sum_{i=0}^{N-1}\lambda_i$ can be reduced to $2\mu^3\alpha J_{\min}\lambda_{\max}\sum_{i=0}^{N-1}\lambda_i/(2-\mu\sum_{i=0}^{N-1}\lambda_i)$. Moreover, assume that $\mu\ll 1/\sum_{i=0}^{N-1}\lambda_i=1/\mathrm{tr}[\mathbf{R}]=1/(N\sigma_x^2)$ (as often adopted in practice), then these maximum terms are further reduced to $\mu^3J_{\min}\lambda_{\max}\sum_{i=0}^{N-1}\lambda_i/2$, $\mu^3J_{\min}\lambda_{\max}^2/2$, $\mu^2J_{\min}\lambda_{\max}$, $\mu^2\alpha J_{\min}\lambda_{\max}/2$, and $\mu^3\alpha J_{\min}\lambda_{\max}\sum_{i=0}^{N-1}\lambda_i$. It is easier to check relative magnitude by dividing $\mu^3\alpha J_{\min}\lambda_{\max}\sum_{i=0}^{N-1}\lambda_i$ by all the other terms to obtain 2α, $2\alpha\sum_{i=0}^{N-1}\lambda_i/\lambda_{\max}$, $\mu\alpha\sum_{i=0}^{N-1}\lambda_i$, and $2\mu\sum_{i=0}^{N-1}\lambda_i$. Furthermore, if $\alpha\ll 1/2$, then $2\alpha\ll 1$ and if $\alpha\ll 1/(2N)$, then $2\alpha\sum_{i=0}^{N-1}\lambda_i/\lambda_{\max}\ll 1$. In summary, if the conditions $0<\alpha\ll\min\{1/2,1/(2N)\}=1/(2N)$, and $0<\mu\ll 1/\sum_{i=0}^{N-1}2\lambda_i=1/2\,\mathrm{tr}[\mathbf{R}]=1/(2N\sigma_x^2)$ are met, then $\mu\alpha\sum_{i=0}^{N-1}\lambda_i\ll 1$, $2\mu\sum_{i=0}^{N-1}\lambda_i\ll 1$, and (17) is negligible. Note that they are very conservative bounds. In our derivation, we have assumed the worst cases that rarely happen. Intuitively, on convergence, the randomness of zero-mean $\xi_{\mathrm{hc}}(n)$ and $\xi_{\mathbf{w}}(n)$ is very likely to make a much smaller (17) than the other terms in (18), which is confirmed by simulations.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for insightful comments and helpful critiques on the manuscript.

REFERENCES

[1] S. G. Chen, Y. A. Kao, and C. Y. Chen, "A new efficient LMS adaptive filtering algorithm," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 372–378, May 1996.
[2] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[3] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.
[4] J. Benesty and P. Duhamel, "A fast exact least mean square adaptive algorithm," *IEEE Trans. Signal Processing*, vol. 40, pp. 2904–2920, Dec. 1992.
[5] O. Macchi, *Adaptive Processing: The Least Mean Squares Approach with Applications in Transmission.* New York: Wiley, 1995.
[6] C. I. Hwang, T. C. Tang, D. W. Lin, and S. G. Chen, "An efficient FSE/DFE-based HDSL equalizer with new adaptive algorithms," in *Proc. 1994 IEEE Int. Conf. Communications*, New Orleans, LA, pp. 288–292.
[7] C. I. Hwang and D. W. Lin, "Joint low-complexity blind equalization, carrier recovery and timing recovery with applications to cable modem transmission," *IEICE Trans. Commun.*, vol. E82-B, no. 1, Jan. 1999.

Comments on "Chaotic Digital Encoding: An Approach to Secure Communication"

W. G. Chambers

***Abstract*—A proposal for using a chaotic system with finite wordlength as a means for encrypting data is criticized on the grounds that some of the proposed schemes are readily attacked in the "chosen plaintext" scenario. Moreover, the proposed schemes do not appear to have any advantages over more conventional cryptographic methods as far as security is concerned.**

***Index Terms*—Chaos, cryptography, sequences.**

I. INTRODUCTION

A method of using chaotic systems for secure communications has been proposed recently.[1] The purpose of this note is to point out that the method is simply a proposal for a cryptographic scheme whose advantages over other cryptographic schemes are debatable. Genuinely chaotic systems operate over infinitely large fields, usually the field of real numbers. Finite digital representations turn such systems into finite-state machines, whose autonomous behavior is bound to be ultimately periodic. The question is then whether, in designing a system for providing data security, one should start with a finite representation of a chaotic system, or whether one should choose conventional cryptology. (A good reference for cryptographic techniques is [1].)

The proposal[1] specifies a list of properties of a finite-state machine which are to be regarded as designating a quasichaotic system. A typical property is the following: "The zero input response has a broad noiselike spectrum for almost all choices of initial conditions. Under

Manuscript received June 13, 1995; revised November 4, 1995. This paper was recommended by Associate Editor R. G. Shenoy.

The author is with the Department of Electronic and Electrical Engineering, King's College London, Strand, London WC2R 2LS, U.K.

[1]D. R. Frey, *IEEE Trans. Circuits Syst. II*, vol. 40, no. 10, pp. 660–666, Oct. 1993.

Publisher Item Identifier S 1057-7130(99)07720-4.