A New Efficient LMS Adaptive Filtering Algorithm

Sau-Gee Chen, Yung-An Kao, and Kwung-Yee Tsai

***Abstract*—A new efficient LMS adaptive filtering algorithm is proposed. The algorithm has comparable performance to that of the direct-form LMS algorithm (DLMS), while costs $N/2 - 1$ fewer multiplications at the expense of $N/2 + 5$ more additions than the DLMS algorithm, where N is the number of filter taps. The new algorithm has one more parameter adaptation than the DLMS algorithm. Further, the algorithm was combined with sign LMS algorithm (SA), signed regressor algorithm (SRA) and zero forcing (ZFA) algorithm for more complexity reduction. Simulation results showed that the new combined algorithms converge as fast as the direct-form SA, SRA and ZFA algorithms, meanwhile still maintain comparable performances.**

I. Introduction

THERE have been widespread interests in adaptive signal processing. Typical application examples [1] of adaptive filter include channel equalization, acoustic echo cancellation, interference cancellation, system identification and so on which in many cases require hundreds of taps. As such, low-complexity adaptive filtering algorithms are highly desirable.

The most popular adaptive filtering algorithm is the direct-form LMS algorithm (DLMS), due to its simplicity and robustness. There are the sign (error) LMS algorithm (SA), signed regressor (i.e., signed input) LMS algorithm (SRA) [2], and the simplest zero forcing (i.e., both signed error and signed input) algorithm (ZFA) [3], which are simplified versions of the DLMS algorithm. All the mentioned algorithms differ in the ways they adapt coefficients, regardless of how they execute convolution operations.

In order to speed up convolution operation of an adaptive filter, many frequency domain and block based adaptive algorithms [4]–[8] were developed which take advantages of FFT. However, generally speaking, frequency domain and block based algorithms are more hardware demanding than the direct-form LMS type of algorithms. They are most applicable to the cases of very large-tap filterings.

Recently, Benesty and Duhamel [9] proposed an effective temporal domain adaptive algorithm, which has comparable performance to that of the DLMS algorithm, and at the same time reduces the number of multiplications by about 25% with a little increase in the number of additions required of the convolution operation. This algorithm takes advantage of the fast convolution algorithm based on the decimation and decomposition of the input signals and the filter [10].

Alternatively, the fast convolution algorithms described in [11], [12] reduce the number of multiplication by close to 50%, at the expense of 50% more additions. The algorithms decompose convolution equation into three parts. The first part is a function of both input signals and filter coefficients, which accounts for almost all the required computation of the algorithm. The second and the third parts depend solely on the input signals and the filter coefficients respectively. The algorithm is well applied to deterministic digital filtering applications, where filter coefficients are fixed and therefore the third part costs no computation. On the other hand, the second part can be realized by a second-order recursive equation, which costs only one multiplication. As such, the total number of multiplications amounts to only $N/2 + 1$, instead of N for the DLMS algorithm. However, there is no saving in multiplication counts when it is applied to adaptive filtering. In such conditions, since the third part is not fixed, it has to be computed in each iteration with $N/2$ more multiplications.

In order to take advantage of the fast convolution algorithm and eliminate the computation burden incurred for the third part when it is applied to adaptive filtering, we introduce an extra parameter adaptation for the third part estimation. Meanwhile, conventional LMS adaptive algorithm for the filter coefficient updates is still used. Theoretical analysis and simulations results showed that the new adaptive algorithm has comparable performance to the DLMS algorithm. Moreover, the algorithm is combined with the SA, SRA and ZFA LMS algorithms for more complexity reduction. Simulations for those algorithms were shown to converge as fast as the DLMS algorithms.

This paper is organized as follows. In Section II, the fast convolution algorithm in [11], [12] is reviewed. Based on this algorithm, a new adaptive algorithm is proposed. In Section III, properties of this algorithm are analyzed and discussed. Several design examples are demonstrated in Section IV, where the new algorithm and its SA, SRA, and ZFA variants were simulated. Finally, Section V draws the conclusion and some of the further works to be done.

II. The New LMS Adaptive Algorithm

A. An Efficient Convolution Algorithm

The new adaptive algorithm is based on the fast convolution algorithm [11], [12] as follows:

$$\begin{aligned} y(n) &= \sum_{k=0}^{N-1} h(k)x(n-k) \\ &= \sum_{k=0}^{N/2-1} \left[x(n-2k) + h(2k+1)\right] \end{aligned}$$

Manuscript received April 18, 1994; revised November 9, 1994. This paper was recommended by Associate Editor B. Kim.

The authors are with the Department of Electronics Engineering and the Institute of Electronics, National Chiao Tung University, Hsinchu 30050, Taiwan, Republic of China.

Publisher Item Identifier S 1057-7130(96)02324-5.

$$\times \left[x(n-2k-1)+h(2k)\right] - \sum_{k=0}^{N/2-1} h(2k)h(2k+1) - \sum_{k=0}^{N/2-1} x(n-2k)x(n-2k-1) \quad (1)$$

and

$$\sum_{k=0}^{N/2-1} x(n-2k)x(n-2k-1) \equiv P(n) = P(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1) \quad (2)$$

$$\sum_{k=0}^{N/2-1} h(2k)h(2k+1) = \text{constant} \quad (3)$$

$$x(k) = 0 \text{ for } k < 0, \quad \text{and} \quad p(k) = 0 \text{ for } k < 0.$$

Equation (2) can be computed with one multiplication and two additions, and (3) is a fixed constant once the filter is designed. As a result, the overall complexity for the convolution is $N/2+1$ multiplications and $3N/2+3$ additions.

B. The New Adaptive Algorithm

Given an adaptive filter with its input sequence $x(n)$ and coefficients $h_k(n)$'s to be adapted, (1) can be applied to the adaptive filter as follows:

$$y(n) = \sum_{k=0}^{N-1} x(n-k)h_k(n) = \sum_{k=0}^{N/2-1} \left[x(n-2k)+h_{2k+1}(n)\right] \times \left[x(n-2k-1)+h_{2k}(n)\right] - C(n) - \sum_{k=0}^{N/2-1} x(n-2k)x(n-2k-1) \quad (4)$$

where

$$C(n) = \sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n). \quad (5)$$

Since $h_k(n)$'s are time varying, $C(n)$ is also time varying which has to be computed for every adaptive iteration. Consequently, there is no advantage in doing such decomposition for adaptive convolution. However, we can introduce an extra coefficient $h_N(n)$ to be adapted, which in the long run is expected to cancel $C(n)$. On the other hand, the conventional LMS algorithm is still applied to filter tap updates. Equation (6) describes the new adaptive algorithm including both the filtering part and the coefficient update part.

$$y'(n) = \sum_{k=0}^{N/2-1} \left[x(n-2k)+h_{2k+1}(n)\right] \times \left[x(n-2k-1)+h_{2k}(n)\right] - P(n) - h_N(n) = y(n) - [h_N(n) - C(n)] \quad (6a)$$

$$h_j(n+1) = h_j(n) + 2\mu e'(n)x(n-j), \quad j = 0, 1, \ldots, N-1 \quad (6b)$$

$$h_N(n+1) = h_N(n) - \alpha e'(n) \quad (6c)$$

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

$$e'(n) = d(n) - y'(n) = d(n) - \sum_{k=0}^{N/2-1} \left[x(n-2k)+h_{2k+1}(n)\right] \times \left[x(n-2k-1)+h_{2k}(n)\right] + P(n) + h_N(n) = d(n) - y(n) + [h_N(n) - C(n)] = e(n) + [h_N(n) - C(n)] \quad (6d)$$

$d(n) =$ the desired signal.

As shown, the computation of $C(n)$ is replaced by its estimate $h_N(n)$, which costs only one extra multiplication. Therefore, the multiplication saving in the adaptive filtering operations is maintained like in the fixed filtering cases. As can be seen, the new algorithm is similar to the DLMS algorithm. However, it has two major differences from the latter one: 1) it needs one more parameter estimation, and 2) it has a slightly different error signal. The algorithm can be incorporated with LMS SA, SRA and ZFA for further complexity reduction. Tables I and II summarize the complexities for the new algorithm, the new algorithm combined with SA, the conventional DLMS algorithm and the SA DLMS algorithm. In these tables, we assume that the step sizes α and μ are integer powers of 2, which contribute no multiplication.

TABLE I
THE MULTIPLICATION COMPLEXITIES OF THE NEW ALGORITHMS AND DLMS ALGORITHMS

	Filter part	Update part	Total
DLMS	N	N	2N
New algorithm	0.5N+1	N	1.5N+1
Sign DLMS	N	0	N
New algorithm with SA	0.5N+1	0	0.5N+1

TABLE II
THE ADDITION COMPLEXITIES OF THE NEW ALGORITHMS AND DLMS ALGORITHMS

	Filter part	Update part	Total
DLMS	N-1	N	2N-1
New algorithm	1.5N+3	N+1	2.5N+4
Sign DLMS	N-1	N	2N-1
New algorithm with SA	1.5N+3	N+1	2.5N+4

Despite of the multiplication advantage of the new algorithm over the DLMS algorithm, the combined addition and multiplication complexity of the new algorithm is about the same as that of the DLMS algorithm. Therefore, when implemented with general-purpose DSP's (with a single-cycle multiply-accumulate operation), the new algorithm does not present speed advantage over the DLMS algorithm. However, for applications with high speed and small area requirements, ASIC realizations of the algorithm is much more effective than its DSP realizations. Still, there are memory overheads required for storing $P(n-2)$ and the product terms $x(n-1)x(n-2)$ through $x(n-N)x(n-N-1)$ for later use by (2), which has to be taken into account when implementing the algorithm.

III. Derivation and Properties of the New Algorithm

The new algorithm is the same as the DLMS algorithm except that an additional parameter $C(n)$ needs to be estimated, and instead of using $e(n) = d(n) - y(n), e'(n)$ as shown in (6d) is used. The key point for the algorithm to function properly is the convergence of $h_N(n)$ to $C(n)$. Under such condition, $e'(n)$ can follow $e(n)$ closely and eventually converge to $e(n)$ as shown in (6d). A formal discussion and justification leads to the condition is stated as follows.

First of all, let's consider the error signal $e''(n)$ instead of $e(n)$ or $e'(n)$ as the error signal to be applied to the DLMS coefficient updates,

$$\begin{aligned} e''(n) &= d(n) - \left[y(n) + \sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)\right] \\ &= e(n) - C(n). \end{aligned} \tag{7}$$

Since $x(n)$ and $h_k(n)$ are uncorrelated [1], [14], the expectation of $y(n)$ is zero if $x(n)$ is a zero-mean signal. In addition, the expectation of $e(n)$ is zero if $d(n)$ and $x(n)$ are zero-mean signals. When step size μ is small, it is reasonable to assume that $C(n)$ is slowly time varying. Small step sizes are commonly used to assure convergence, small residue error, stability and uncorrelatedness of parameters. Under such conditions, coefficient adaptations can be considered as a DLMS adaptation, but with the desired signal $d(n)$ being biased with a relatively "constant" $C(n)$. If we plug this error signal into coefficient update equations and take the expectations of both sides as follows:

$$\begin{aligned} E\{h_j(n+1)\} &= E\{h_j(n) + 2\mu e''(n)x(n-j)\}, \\ &\qquad j = 0, 1, \ldots, N-1 \\ &= E\{h_j(n) + 2\mu e(n)x(n-j)\} - 2\mu \\ &\quad \times E\left\{\left[\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)\right] x(n-j)\right\} \end{aligned} \tag{8}$$

and assume that $C(n)$ is uncorrelated with $x(n)$, and the mean of $x(n)$ is zero, then (8) is reduced to (9). Note that the assumption $E\{x(n)\} = 0$ is frequently encountered in practical applications, and the assumption of uncorrelatedness is based on the condition that $C(n)$ is slowly varying in comparison with $x(n)$ as is confirmed by later simulations.

$$\begin{aligned} E\{h_j(n+1)\} &= E\{h_j(n) + 2\mu e(n)x(n-j)\} - 2\mu \\ &\quad \times E\left\{\left[\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)\right]\right\} \\ &\quad \times E\{x(n-j)\} \\ &= E\{h_j(n) + 2\mu e(n)x(n-j)\}. \end{aligned} \tag{9}$$

Since by (7) $e''(n)$ is a linear combination of zero mean signal $e(n)$ and the slowly varying parameter $C(n)$, $C(n)$ can be estimated by averaging $e''(n)$'s within a window close to the time instant n as follows:

$$\begin{aligned} h_N(n) &= -\frac{1}{M}\sum_{k=0}^{M-1} e''(n-k) \\ &= h_N(n-1) - \frac{1}{M}[e''(n) - e''(n-M)]. \end{aligned} \tag{10}$$

This update needs 2 additions, one multiplication and M registers for error signal storage. An alternative for $h_N(n)$ adaptation is by using the popular exponential smoothing algorithm as shown below,

$$h_N(n) = (1-\alpha)h_N(n-1) - \alpha e''(n-1) \quad \text{where } 0 < \alpha < 1. \tag{11}$$

And from (11), it can be shown that $E\{e''(n)\} = -Eh_N(n)$.

In the adaptation, step size α is a critical factor for the convergence of $h_N(n)$ to a desired final constant value. By (7), the expectation value of $e''(n)$ can be shown to reflect $C(n)$ as shown in (12).

$$\begin{aligned} E\{e''(n)\} &= E\{e(n)\} - E\left\{\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)\right\} \\ &= -E\left\{\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)\right\}. \end{aligned} \tag{12}$$

Therefore, it is desirable to pick a large α for better tracking of $C(n)$. Comparable performances as those of the DLMS algorithm were obtained when α is in the vicinity of 0.5. Note that α is preferably to be much larger than the step size μ.

From (7), (11), and (12), we expect that $e''(n)$ converges to $-C(n)$. Hence, if $h_N(n)$ is added to (7), we have the error signal $e'(n)$ that is used by the new adaptive algorithm described in (6c), and

$$\begin{aligned} e'(n) &= d(n) - \sum_{k=0}^{N-1} h_k(n)x(n-k) \\ &\quad - \sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n) + h_N(n) \\ &= e(n) - \sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n) + h_N(n) \\ &= e''(n) + h_N(n) \end{aligned} \tag{13}$$

where $e'(n)$ satisfies the following required condition

$$\begin{aligned} E\{e'(n)\} &= E\{e(n)\} \\ &\quad - E\left\{\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n) - h_N(n)\right\} \\ &= 0. \end{aligned} \tag{14}$$

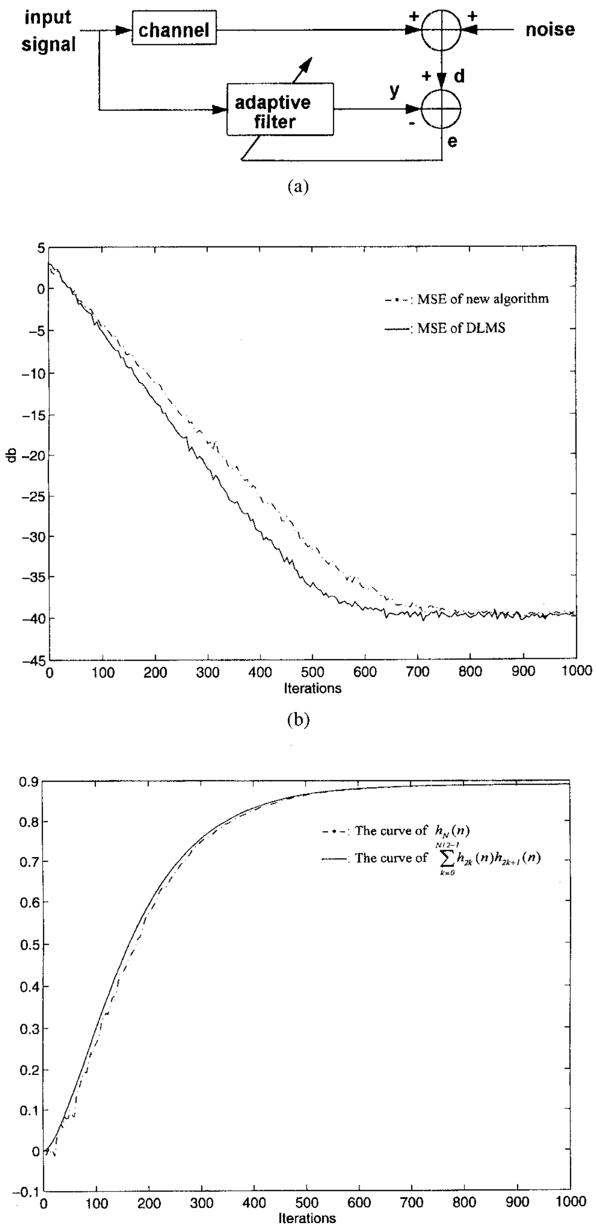


Fig. 1. (a) Block diagram for system identification. (b) Mean square errors of the new algorithm and DLMS algorithm for Example 1. (c) Learning curves of $h_N(n)$ and $\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)$ for Example 1.

From (13), we can express (7) in terms of the actual error signal $e'(n)$ as follows:

$$\begin{aligned} h_N(n) &= (1-\alpha)h_N(n-1) - \alpha[e'(n-1) - h_N(n-1)] \\ &= h_N(n-1) - \alpha e'(n-1) \end{aligned} \tag{15}$$

which ends up with (6c).

Next, we discuss the conditions for the convergence of the new algorithm. By taking the expectation of (6b), we have

$$\begin{aligned} E\{h_j(n+1)\} &= E\{h_j(n)\} + 2\mu E\{e'(n)x(n-j)\} \\ &= E\{h_j(n)\} + 2\mu \\ &\quad \times E\left\{\left[d(n) - \sum_{k=0}^{N-1} h_k(n)x(n-k) - C(n) + h_N(n)\right]x(n-j)\right\}. \end{aligned} \tag{16}$$

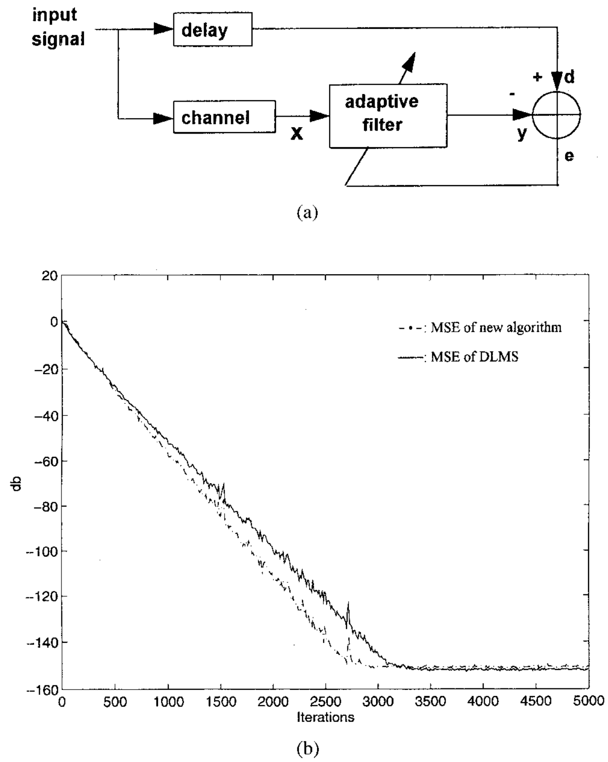


Fig. 2. (a) Block diagram for inverse system modeling. (b) Mean square errors of the new algorithm and DLMS algorithm for Example 2.

For (16) to converge, we require that $h_N(n) - C(n)$ be uncorrelated with $x(n-j)$ and $E[x(n)] = 0$ or $E\{h_N(n) - C(n)\} = 0$. Under such condition, (16) is reduced to the same form as that of the DLMS algorithm as follows:

$$\begin{aligned} E\{h_j(n+1)\} &= E\{h_j(n)\} + 2\mu \\ &\quad \times E\left\{\left[d(n) - \sum_{k=0}^{N-1} h_k(n)x(n-k)\right]x(n-j)\right\}. \end{aligned} \tag{17}$$

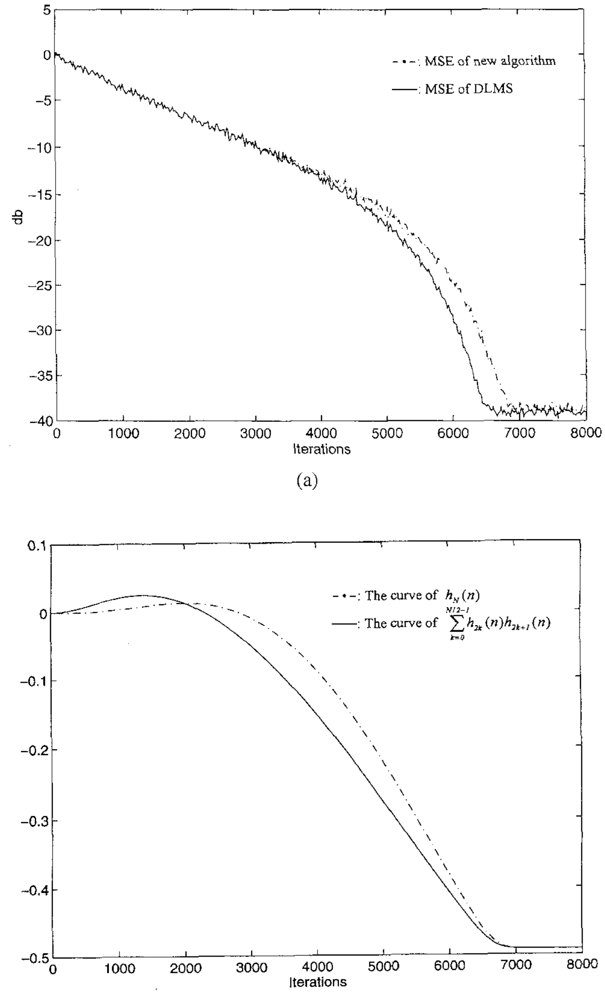
Therefore, for the new algorithm to converge, the required constraint of stepsize μ is the same as that of the DLMS algorithm

$$\frac{1}{\lambda_{\max}} > \mu > 0 \tag{18}$$

where $\lambda_{\max}$ is the largest eigenvalue of system correlation matrix. According to (11), for the convergence of $h_N(n)$ the stepsize α must be confined to the range $1 > \alpha > 0$.

IV. SIMULATION EXAMPLES

In this section we simulate the new algorithm and its SA, SRA and ZFA versions by applying them to system identification and inverse system modeling examples. The results are



(a)

(b)

Fig. 3. (a) Mean square errors of the new algorithm and DLMS algorithm for Example 3, both combined with SA. (b) Learning curves of $h_N(n)$ and $\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)$for Example 3, combined with SA.

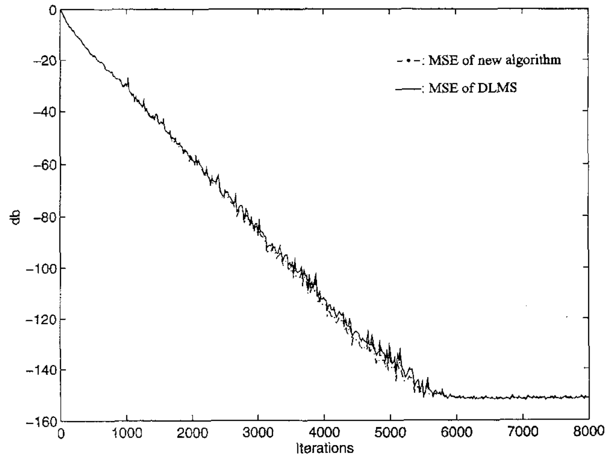


Fig. 4. Mean square errors of the new algorithm and DLMS algorithm for Example 4, both combined with SRA.

compared with those of their counter DLMS algorithms. All the simulation curves are averages of 500 runs.

The condition of zero-mean $x(n)$ is not a necessary condition for the new algorithm to converge as mentioned before and verified in the later simulations. The convergence rate and mean square error of the new algorithm is dependent on stepsizes α and μ. In some examples (α is very close to 0.5), the convergence rates of the new algorithm are a little faster than those of the DLMS in the beginning of adaptations, but the final mean square errors are a little worse than those of DLMS for the same stepsize μ. Generally, the convergence rate of this algorithm is very close to that of DLMS algorithm.

Example 1. System Identification: Fig. 1(a) shows a block diagram for the system identification, with an 8-tap adaptive filter. The channel impulse response $g(n) = 0.33$, 0.67, 1.0, 0.67, 0.33 to be identified is taken from [13]. The input samples are assumed statistically independent binary numbers having values 1 or -1 with equal probability, and the additive noise is a white Gaussian, zero-mean sequence with variance $d = 0.0001$. The stepsizes are $\mu = 0.005$ and $\alpha = 0.1$. Fig. 1(b) shows that although the mean square error (MSE) of the new algorithm is slower than that of the DLMS algorithm, it closely follows the latter one. Fig. 1(c) shows that the $h_N(n)$ converges to $C(n)$ rapidly as expected.

Example 2. Inverse System Modeling: The block diagram for the inverse system modeling is shown in Fig. 2(a). Its input sequence is assumed a white Gaussian, zero-mean noise with variance $d = 1$. The impulse response of channel is simply described by the raised cosine model given below.

$$\begin{aligned} h(n) &= \{1+\cos[2\pi(\mathrm{n}-2)/3.1]\}/2 \quad && \text{for } n = 1,2,3 \\ &= 0 && \text{otherwise.} \end{aligned}$$

Here we use 30 taps for both the DLMS algorithm and new algorithm, with the stepsizes $\mu = 0.01$ and $\alpha = 0.2$. Surprisingly, Fig. 2(b) shows that the MSE curve of the new algorithm converges faster than that of the DLMS algorithm. Since $h_N(n)$ curve almost coincides $C(n)$ curve, they are not shown here.

Example 3. The New Algorithm Combined with SA for Example 2: The simulation conditions are the same as example 2, except that the coefficients are updated with SA for both the new algorithm and DLMS algorithm. The stepsizes are assumed $\mu = 0.000\,25$ and $\alpha = 0.0005$. Fig. 3(a) shows a slower (but closely following) MSE curve of the new algorithm than that of the DLMS algorithm. Fig. 3(b) shows that there is a noticeable deviation between $C(n)$ and $h_N(n)$ curves. However, both curves almost converge at the same instant.

Example 4. The New Algorithm Combined with SRA Algorithm for Example 2: The simulation conditions are the same as example 2, except that the coefficients are updated with SRA. The stepsizes are assumed $\mu = 0.01$ and $\alpha = 0.05$. Fig. 4 shows that both MSE learning curves overlap. Similarly, $C(n)$ and $h_N(n)$ almost coincide.

Example 5. The New Algorithm Combined with ZFA for Example 2: The stepsizes are assumed $\mu = 0.000\,25$ and $\alpha = 0.0005$. Fig. 5(a) shows two very close MSE curves. Similar to example 4, there is a noticeable deviation between $C(n)$

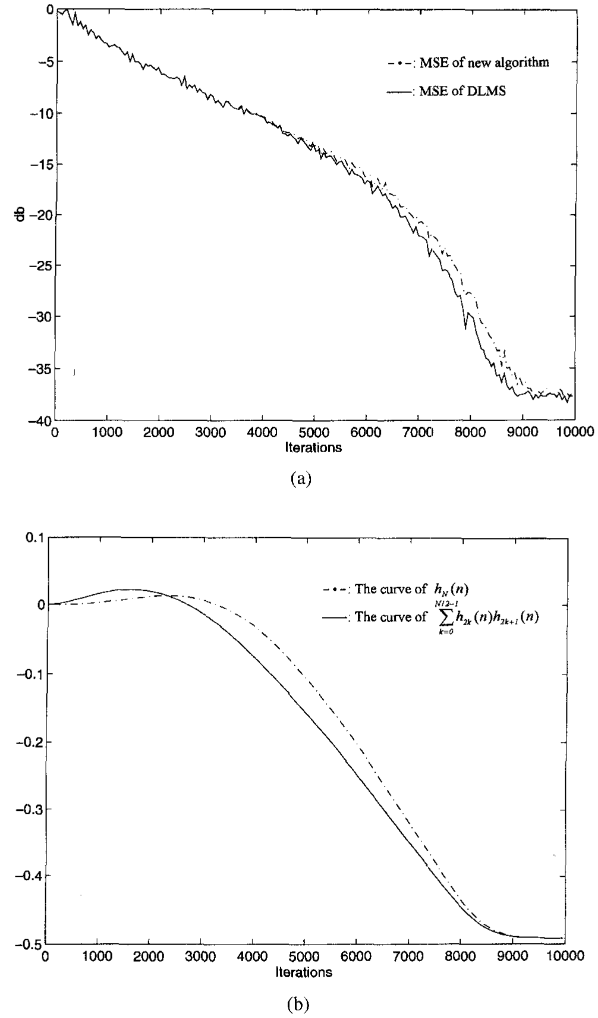


Fig. 5. (a) Mean square errors of the new algorithm and DLMS algorithm for Example 5, both combined with ZFA. (b) Learning curves of $h_N(n)$ and $\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)$for Example 5, combined with ZFA.

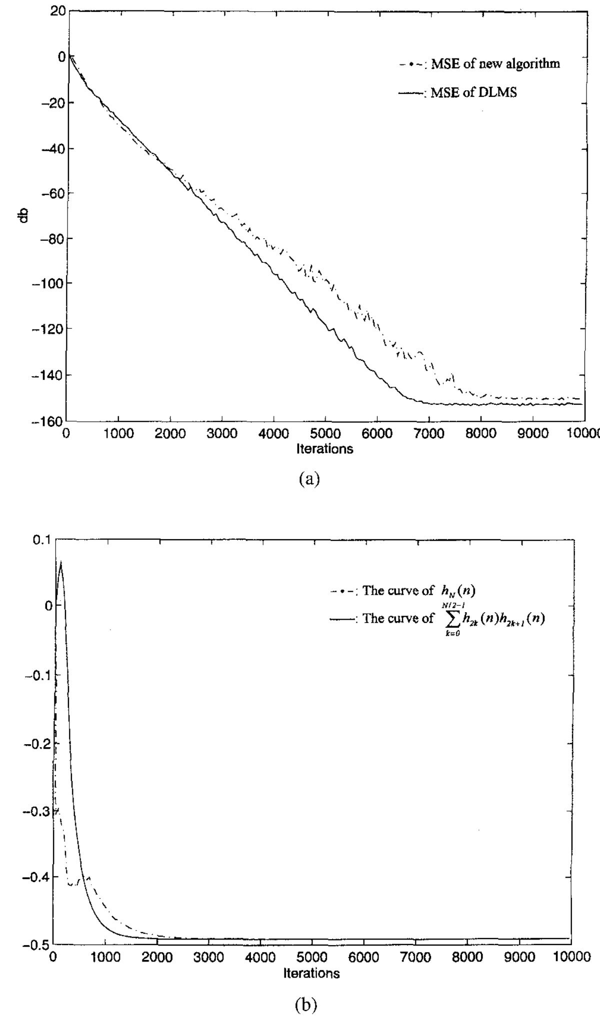


Fig. 6. (a) Mean square errors of the new algorithm and DLMS algorithm for Example 6, with nonzero mean inputs. (b) Learning curves of $h_N(n)$ and $\sum_{k=0}^{N/2-1} h_{2k}(n)h_{2k+1}(n)$ for Example 6, with nonzero mean inputs.

TABLE III
THE EXCESS MEAN SQUARE ERRORS (IN dB) AND THE CORRESPONDING RELATIVE dB DIFFERENCES (IN %) OF THE NEW ALGORITHM AND CONVENTIONAL ALGORITHMS

	Conventional algorithms (dB)	New algorithm (dB)	(DLMS-New algorithm)/DLMS (%)
Example 1	-39.8439	-39.6039	0.6%
Example 2	-151.8568	-150.5929	0.83%
Example 3	-39.0729	-38.6736	1.02%
Example 4	-151.6378	-151.3126	0.21%
Example 5	-37.7465	-37.3535	1.04%
Example 6	-152.4538	-150.0987	1.54%

and $h_N(n)$ curves as shown in Fig. 5(b). However, both MSE curves almost settle at the same time instant.

Example 6. The New Algorithm with Nonzero-Mean Input for Example 2: This is a worse case than the mentioned examples. Here, the input signal is assumed to be a normal distribution random signal, with an expected value of 0.5 and variance of 1. The stepsizes are assumed $\mu = 0.005$ and $\alpha = 0.4$. Very similar results to the previous two examples are obtained as shown in Fig. 6(a) and (b). However, there is larger excess mean square error for the new algorithm than those of the previous examples.

Table III summarizes the excess mean square errors (in dB), and the corresponding relative dB differences (in %) for the new and conventional algorithms. As can be seen, all the relative dB differences are around 1%. In addition, the new algorithm has been successfully applied to the HDSL equalization [15].

V. CONCLUSION

Due to its nonlinear nature, the MSE analysis of the new algorithm remains to be precisely characterized with more efforts. Meanwhile, although the introduced compensation parameter is considered to converge faster than the direct-form coefficients in some examples and therefore has little impact on the algorithm convergence rate, its properties also need to be characterized. The algorithm works well when the input signal has zero mean, as expected. When inputs are not zero mean, the algorithm also converges to optimal solutions, except in some rare occasions. This phenomenon is also left to be further investigated. Moreover, theoretical analysis of the effect of stepsizes to the convergence of the new algorithm especially to its SA, SRA, and ZFA versions is required further investigation. On the other hand, the mentioned fast convolution algorithm [11] can be directly combined with the block LMS algorithms. Doing this way, there is no need introducing $h_N(n)$, and we can directly compute $C(n)$ once for an entire block processing. Therefore, low computation complexity is still maintained. Finally, the new algorithm can also be applied to 2-D adaptive filterings [16].

REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1985.

[2] E. Eweda, "Analysis and design of a signed regressor LMS algorithm for stationary and nonstationary adaptive filtering with correlated gaussian data," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1367–1374, Nov. 1990.

[3] R. W. Lucky, "Technique for adaptive equalization of digital communications systems," *Bell Syst. Tech. J.*, vol. 45, pp. 255–286, Feb. 1966.

[4] J. C. Lee and C. K. Un, "Block realization of multirate adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, pp. 105–117, Feb. 1986.

[5] E. R. Ferrara, "Fast implementation of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, pp. 474–475, Aug. 1980.

[6] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, pp. 1073–1082, Oct. 1983.

[7] ——, "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 584–592, June 1981.

[8] J. J. Shynk, "Frequency domain and multirate adaptive filtering," *IEEE Signal Process. Mag.*, pp. 14–37, Jan. 1992.

[9] J. Benesty and P. Duhamel, "Fast exact LMS adaptive algorithm," *IEEE. Trans. Signal Process.*, vol. 40, pp. 2904–2920, Dec. 1992.

[10] Z. J. Mou and P. Duhamel, "Fast FIR filtering: Algorithms and implementations," *Signal Process.*, pp. 377–384, Dec. 1987.

[11] S. G. Chen and R. J. Tsay, "A novel fast filtering algorithm and its implementation," in *Proc. EUSIPCO'92*, Brussels, Aug. 1992, pp. 945–948.

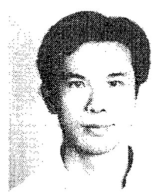
[12] ——, "A class of novel fast FIR filtering algorithms and their realizations," *IEEE Trans. Signal Process.*, Sept. 1992.

[13] F. R. Magee and J. G. Proakis, "Adaptive maximum likelihood sequence estimation for digital signaling in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 120–124, Jan. 1973.

[14] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

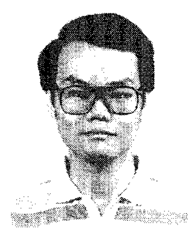
[15] C. I. Hwang, T. C. Tang, D. W. Lin, and S. G. Chen, "An efficient DFE/FSE-based HDSL equalizer with new adaptive algorithms," in *Proc. 1994 IEEE Int. Conf. Commun.*, New Orleans, LA, May 1994, pp. 288–294.

[16] S. G. Chen and Y. A. Kao, "A new efficient 2-D LMS adaptive filtering algorithm," in *Proc. ISCAS'94*, London, England, May 1994, pp. 2233–2236.



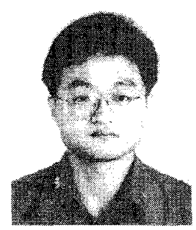
Sau-Gee Chen received the B.S. degree in nuclear engineering from the National Tsing Hua University, Taiwan, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York at Buffalo, in 1984 and 1988, respectively.

He currently holds a joint appointment as an Associate Professor of the Microelectronics and Information System Research Center, and the Department of Electronics Engineering, National Chiao Tung University, Taiwan. His research interests include computer arithmetic, digital signal processing, and VLSI circuit design.



Yung-An Kao was born in Taipei, Taiwan. He received the B.S.E.E. degree from the Tam Kang University, Taipei, Taiwan, in 1989 and the M.S.E.E. degree from the Tatung Institute of Technology, Taipei, Taiwan, in 1991.

He is currently working toward the Ph.D. degree at the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan. His research interests include adaptive filter, digital signal processing, and filter design.



Kwung-Yee Tsai was born in Chia-Yi, Taiwan, in 1969. He received the B.S.E.E. degree from the Chung Yuan University, Chung-Lee, Taiwan, in 1991 and the M.S.E.E. degree from the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, in 1993.

Since 1995, he has been an Associate Tech. Staff with the Peripheral Products Design Division of Acer Laboratories Inc., Taiwan, where he is engaged in the research on VLSI implementation of telecommunications.