World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:4, No:5, 2010

A Novel Convergence Accelerator for the LMS Adaptive Algorithm

Jeng-Shin Sheu, Jenn-Kaie Lain, Tai-Kuo Woo, and Jyh-Horng Wen

***Abstract*—The least mean square (LMS) algorithm is one of the most well-known algorithms for mobile communication systems due to its implementation simplicity. However, the main limitation is its relatively slow convergence rate. In this paper, a booster using the concept of Markov chains is proposed to speed up the convergence rate of LMS algorithms. The nature of Markov chains makes it possible to exploit the past information in the updating process. Moreover, since the transition matrix has a smaller variance than that of the weight itself by the central limit theorem, the weight transition matrix converges faster than the weight itself. Accordingly, the proposed Markov-chain based booster thus has the ability to track variations in signal characteristics, and meanwhile, it can accelerate the rate of convergence for LMS algorithms. Simulation results show that the LMS algorithm can effectively increase the convergence rate and meantime further approach the Wiener solution, if the Markov-chain based booster is applied. The mean square error is also remarkably reduced, while the convergence rate is improved.**

***Keywords*—**LMS, Markov chain, convergence rate, accelerator.

I. Introduction

IN the field of adaptive signal processing, the least-mean square (LMS) is an extensively explored algorithm due to its implementation simplicity [1]-[2]. The LMS algorithm has been widely used in mobile communications. However, convergence rate is its main limitation [3]-[4]. That is, the LMS algorithm has conflicting requirements of small step-size parameter to reduce mis-adjustment and large step-size parameter to achieve fast convergence.

Methods used to address the issue of convergence rate can be classified into two categories: time-domain and transform-domain methods. In the time domain, researchers have constantly looked for new methods of selecting step sizes to improve convergence rate. The most commonly used scheme is the gear shifting approach [3] that utilizes a large step size during the transient state and then shifts to a smaller one during the steady state. Other important time-domain approaches are variable step-size LMS algorithms [5]-[6]. The idea is to make step-size data-dependent. However, when the input signal is highly colored, the convergence rate of an LMS algorithm tends to be slow. The literatures in [7]-[8] alleviate the correlation of the input signal by pre-whitening it using one from a number of transforms. Among these transform-domain methods, the frequency-domain versions are more often used. Frequency-domain LMS algorithms can increase the speed of convergence for broadband signals [9]-[10].

It is obvious that the existing methods mentioned above do not exploit the statistics of the *past information* in the updating process. In this work, a novel convergence accelerator based on the concept of Markov chains is proposed to speed up the process of reaching the steady-state LMS weights. In the meantime, the mean square error (MSE) can be further improved during the steady-state epoch. The nature of Markov chains makes it possible to exploit the *successive updates* of each LMS weight to improve the convergence rate and MSE. The excess mean square error (EMSE) of an LMS algorithm is proportional to the value of step size [3]-[4]. For a specific step size, the weights of an LMS algorithm will eventually get into the steady-state period, that is, each weight of the LMS algorithm stochastically fluctuates within a certain range. We thus refer this phenomenon to *fluctuation steady state* (FSS). An important feature of the proposed boosting scheme is that it utilizes the weight information when the LMS algorithm gets into the FSS. Based on the concept of Markov chains, we divide the fluctuating range of each weight into segments and let each segment represent a state. Then the transition matrix of each weight can be constructed separately through successive updates of the weight value. Since the transition matrix has a smaller variance than that of the weight itself by the central limit theorem (CLT), the weight transition matrix converges faster than the weight itself. Therefore we are motivated to take advantage of the Markov-chain concept to speed up the convergence rate through finding the probability vector of a weight from its transition matrix.

The rest of this paper is organized as follows. In Section II, a brief review of the LMS algorithm is presented, and subsequently we explain how the concept of Markov chains can be applied to the LMS algorithm to speed up the convergence rate. Then the Markov-chain based convergence booster is proposed in Section III. The related implementation techniques

Jeng-Shin Sheu is with the Department of Computer Science and Information Engineering, National Yunlin University of Science & Technology, Yunlin, Taiwan (corresponding author to provide phone: 886-5-5342601; fax: 886-5-5312170; e-mail: jssheu@yuntech.edu.tw).

Jenn-Kaie Lain is with the Department of Electronic Engineering, National Yunlin University of Science & Technology, Yunlin, Taiwan (e-mail: lainjk@yuntech.edu.tw).

Tai-Kuo Woo is now with the Department of Information Management, National Defense University, Taiwan. (e-mail: w13464@yahoo.com.tw).

Jyh-Horng Wen is now with the Department of Electrical Engineering, Tunghai University , Taiwan. (e-mail: jhwen@thu.edu.tw).

are presented in Section IV. Finally simulation results and conclusions are given in Sections V and VI, respectively.

II. The LMS Algorithm and Applications of Markov Chains

A. The LMS Algorithm

Adaptive filtering algorithms are used to estimate a set of parameters of a system model. The most popular system model is the linear model with a transversal structure. Let the set $\{c_i(n), i = 0,1,\ldots,M-1\}$ denotes the M weight values of the filter at discrete time n. The tap inputs are assumed wide-sense stationary (WSS) and denoted by $x(n)$, $x(n-1)$,…, and $x(n-M-1)$. Viewing the tap input $x(n)$ as the current value of the filter input, the remaining $M-1$ tap inputs represent the past values of the filter input. The output $y(n)$ at some discrete time n is used to provide an estimate of the desired response at discrete time n denoted by $d(n)$.

The filter output at discrete time n is described by the following equation:

$$y(n) = \sum_{i=0}^{M-1} c_i(n)\, x(n-i). \tag{1}$$

The above equation can be written in vector form as

$$y(n) = \mathbf{x}^T(n)\mathbf{c}(n), \tag{2}$$

where

$$\mathbf{x}(n) = \left[x(n), x(n-1), \ldots, x(n-M-1)\right]^T, \tag{3}$$

$$\mathbf{c}(n) = \left[c_0(n), c_1(n), \ldots, c_{M-1}(n)\right]^T, \tag{4}$$

and the superscript ^{T} signifies the matrix transpose.

The *prediction error* $e(n)$ is expressed as the difference between the desired response and the filter output:

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{c}(n). \tag{5}$$

Then the weight vector of an LMS algorithm is updated according to the following iterative formula:

$$\mathbf{c}(n+1) = \mathbf{c}(n) + \mu\mathbf{x}(n)e(n), \tag{6}$$

where μ is the step-size parameter.

B. Applications of Markov Chains

The Markov chain is a stochastic process where the transition of an object can be classified into a number of states. The object may switch from the current state i to another state j in a fixed interval of time with a transition probability P_{ij}. The probability of occurrence of the object in each state from the previous time to the current time can be established through a *transition matrix*. During the FSS period, the transition matrix between successive updates of a weight is constructed by

$$\mathbf{A} = \begin{bmatrix} P_{00} & P_{01} & \cdots & P_{0\,S-1} \\ P_{10} & P_{11} & \cdots & P_{1\,S-1} \\ \vdots & & & \vdots \\ P_{S-1\,0} & P_{S-1\,1} & \cdots & P_{S-1\,S-1} \end{bmatrix}, \tag{7}$$

where S denotes the number of states of the Markov chain. The transition probability P_{ij} is obtained by counting the number of occurrences of the object according to the following formula:

$$P_{ij} = f_{ij} \Big/ \sum_s f_{is}, \tag{8}$$

where f_{ij} is the number of occurrences of the weight from state i to state j.

The problem with an LMS algorithm is that the statistical behavior of the past information is not used during the updating process. An important feature of the proposed accelerating scheme is that it exploits the successive weight information when the LMS algorithm gets into the FSS epoch. During the FSS period, each weight moves about within a certain range. For each weight of an LMS algorithm, we divide the range into segments of equal length and let each segment represent a state. Then the transition matrix of each weight can be established through its successive updates individually and independently. Consider a certain weight of an LMS algorithm, and assume that the population, with finite mean μ and standard deviation σ, describes the stochastic behavior of successive updates of the weight during the FSS period.

The fundamental idea of the proposed scheme is based on the observation that the transition probability P_{ij} reaches the steady state faster than that of the weight itself. This can be observed from the CLT, which simply states that if n samples are drawn from the population, then the sample mean can be approximated by a random variable having a normal distribution with mean μ and *sample standard deviation* $\sigma_n = \sigma/\sqrt{n}$. Here the standard deviation σ represents the fluctuation extent of successive updates of the weight itself during the FSS period.

In (6), the step size parameter μ is a constant that governs the stability and convergence rate. A large step size speeds up the rate of convergence, but at the same time the standard deviation σ of the population is increased, together with the *EMSE*. The *EMSE* is defined by

$$E\left[|e(n)|^2\right] - J_{\min} \tag{9}$$

where $E[\cdot]$ stands for expectation, and $J_{\min}$ is the minimum mean square error (*MMSE*) for the prediction error. We can further obtain the *EMSE* as follows [3]:

$$EMSE = \mathrm{tr}\left[\mathbf{K}\cdot\mathbf{R}\right] \approx \mu\,\mathrm{tr}\left[\mathbf{R}\right] J_{\min}, \tag{10}$$

where $\mathrm{tr}[\cdot]$ denotes the trace of a square matrix, $\mathbf{R}$ is the $M \times M$ tap input autocorrelation matrix and $\mathbf{K}$ is the $M \times M$ tap error covariance matrix, defined by

$$\mathbf{K} = E\left[\left(\mathbf{c}-\mathbf{c}^*\right)\left(\mathbf{c}-\mathbf{c}^*\right)^T\right]. \tag{11}$$

Here $\mathbf{c}$ is the $M \times 1$ weight vector defined in (6) and $\mathbf{c}^*$ is the corresponding vector of the Wiener solution. It is observed that the weight vector equals the corresponding Wiener vector $\mathbf{c}^*$, when the value of *EMSE* is zero. Therefore the Wiener vector can be reached only when the LMS algorithm employs an infinitesimally small step size μ. However, such an infinitesimally small step size is infeasible in the real application. Therefore eventually, a weight will stochastically fluctuate around the corresponding Wiener solution with an extent proportional to the value of step size μ.

III. The Markov-Chain Based Convergence Booster

The well-known rule of the LMS algorithm [3] states that the LMS algorithm can converge in the mean square error (*MSE*), if the state-size parameter μ satisfies the following condition:

$$0 < \mu < 2/\mathrm{tr}[\mathbf{R}]. \tag{12}$$

The proposed boosting scheme is applied at the moment when the LMS algorithm gets into the FSS. Consider a certain LMS tap, and assume that its successive weight values are restricted between $R_{\min}$ and $R_{\max}$ during the tap updating process in the FSS epoch. In next Section we will discuss the issues of how to determine both the instant of time when the FSS arrives and the associated fluctuation range $[R_{\min}, R_{\max}]$. We divide the range into S segments of equal length, and each of which represents a state. Therefore the state quantum q is:

$$q = R_{\max} - R_{\min}/S. \tag{13}$$

The number of states, S, for the Markov chain is determined by the required precision of the state quantum. Let d_k and d_{k+1} denote lower bound and upper bound of the kth state, respectively. The state boundaries are seen to be

$$d_k = R_{\min} + k \cdot q, \quad k = 0,1,\ldots,S-1. \tag{14}$$

The number of occurrences of the weight from state i to state j is updated at each iteration of the LMS algorithm. At the end of the training period, the weight transition matrix **A** is constructed according to (7).

Let $\Phi = [\varphi_0, \varphi_1, \ldots, \varphi_{S-1}]^T$ denote the state-value vector, in which φ_k is the representative value of state k. The probability vector **p** of the weight can be obtained from the established transition matrix **A**. Then the booster speeds up the convergence rate by forcing this weight to be the following booster value

$$\hat{c} = \mathbf{p}^T \Phi = \sum_{k=0}^{S_\mu - 1} p_k \varphi_k, \tag{15}$$

where p_k, the kth element of **p**, is the steady state probability of state k. The convergence booster can be incorporated with the existing LMS algorithms [5]-[6] mentioned early. Thus the computation time is saved. The flow diagram of the LMS algorithm intermixed with the booster is given in Fig. 1.

It is worth mentioning that the booster value in (15) is a weighted sum of past weight updates. Therefore the Markov-chain based booster has the ability to track variations in signal characteristics. In other words, the proposed booster can speed up convergence rate of an LMS algorithm.

IV. Implementation Issues of the Markov-Chain Based Booster

A. The Mean Square Successive Difference Estimator

The proposed booster scheme is employed at the moment when the LMS algorithm gets into the FSS. Therefore it is necessary to devise a method to determine this moment. We utilize a simple but effective estimator to track the trend of weight variance during the successive updates. The estimator is the mean square successive difference estimator (MSSDE) [11].

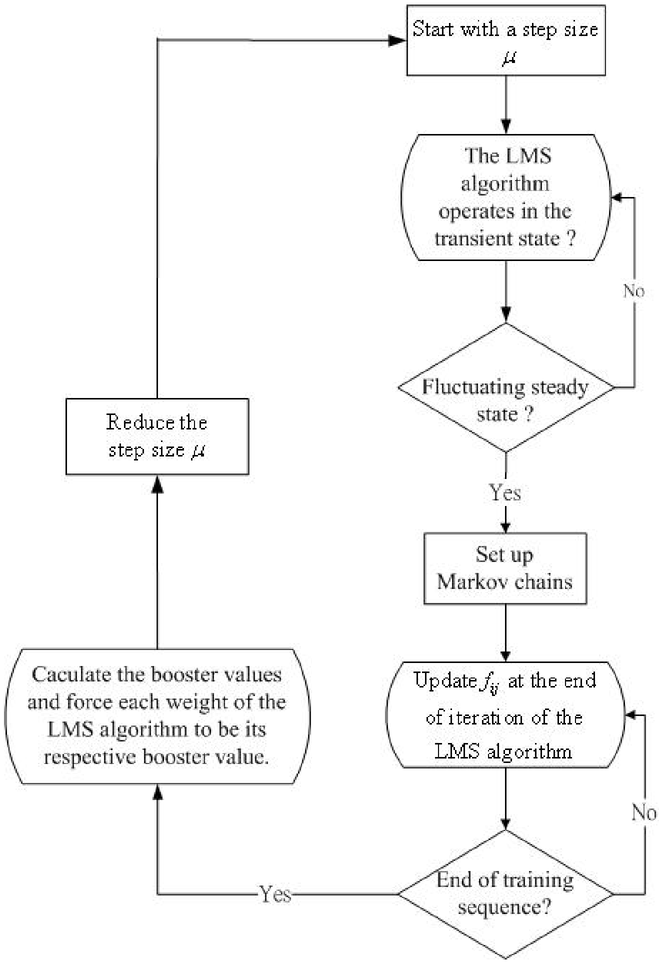

Fig. 1 The flow chart of the Markov-chain based LMS algorithms

B. The Fluctuation Range

Let $\tilde{c}_i$ denote the value of c_i at the instant while the LMS algorithm just gets into the FSS. Then the lower bound and upper bound of the fluctuation range can be respectively determined as

$$R_{\min} = \tilde{c}_i - m\sigma_{c_i} \text{ and } R_{\max} = \tilde{c}_i + m\sigma_{c_i}, \tag{22}$$

where m is a positive constant. By the Chebychev's inequality, a universal bound on the deviation $|c_i - \tilde{c}_i|$ in terms of σ_{c_i} is :

$$P\left(|c_i - \tilde{c}_i| \geq m\sigma_{c_i}\right) \leq 1/m^2, \tag{23}$$

where $P(\cdot)$ stands for probability. For $m = 2$ and 3, we have

$$P\left(|c_i - \tilde{c}_i| \geq 2\sigma_{c_i}\right) \leq 0.25, \text{ and } P\left(|c_i - \tilde{c}_i| \geq 3\sigma_{c_i}\right) \leq 0.11, \tag{24}$$

so there is at least 75% and 89% chance that the weight c_i will be within $2\sigma_{c_i}$ and $3\sigma_{c_i}$ of $\tilde{c}_i$ during the FSS period, respectively. In [12], Gosh and Meeden shown that Chebychev's inequality is very conservative. Thus it is enough to set the value of m to be 3.

V. SIMULATION

A. Model

We consider a first-order autoregressive (AR) process $f(n)$, described by the difference equation:

$$f(n) = af(n-1) + w(n), \tag{25}$$

where a is the parameter, which is also the Wiener solution of the LMS algorithm, and $w(n)$ is a zero-mean Gaussian white process with variance σ_w^2. Assume that the AR process $f(n)$ with the variance of σ_f^2 is independent of the noise process $w(n)$. Then we have $\sigma_w^2 = (1-a^2)\sigma_f^2$. The LMS algorithm to estimate the parameter a is given by

$$c(n+1) = c(n) + \mu f(n-1) e_f(n), \tag{26}$$

where ❍ is the step-size parameter and $e_f(n)$ is the *prediction error* of defined as

$$e_f(n) = f(n) - c(n) f(n-1). \tag{27}$$

We further define the *estimation error* at iteration n as the difference between the parameter a and tap value at iteration n:

$$e_a(n) = a - c(n). \tag{28}$$

Invoking the independence assumption, the ensemble average of the squared prediction error at iteration n for the AR process $f(n)$ is given by [3]:

$$J(n) = \left(\sigma_f^2 - \sigma_w^2\left(1+0.5\mu\sigma_f^2\right)\right)\left(1-\mu\sigma_f^2\right)^{2n} + \sigma_w^2\left(1+0.5\mu\sigma_f^2\right), \tag{29}$$

and the corresponding MMSE for the prediction error is obtained as follows:

$$J_{\min} = \sigma_w^2. \tag{30}$$

According to (21), the weight variance is approximated as

$$\sigma_c^2 \approx \mu\sigma_w^2. \tag{31}$$

Thus the lower bound and upper bound of the fluctuation range are determined as

$$R_{\min} = \tilde{c} - m\sqrt{\mu}\sigma_w \text{ and } R_{\max} = \tilde{c} + m\sqrt{\mu}\sigma_w, \tag{32}$$

where $\tilde{c}$ is the value of the tap c at the instant while the LMS algorithm just gets into the FSS.

B. Results

We evaluate the proposed Markov-chain based booster for the LMS algorithm by computer simulations. For the first-order AR process, the parameter a is 0.99 and unless otherwise stated, the noise variance is assumed 0.1. The step-size parameter μ for the LMS algorithm is 0.05. The number of states for the Markov chain is 6. The following results are obtained by ensemble averaging over 100 independent realizations.

Now, we show both the convergence rate and the MSE improved by the proposed booster, assuming that the value of N_d is 10. We start to establish the weight transition matrix for the proposed convergence accelerator when the LMS just gets into the FSS epoch. Every 100 iterations, the booster speeds up the convergence rate of the weight by forcing this weight to be the booster value calculated according to (15). Fig. 3 shows the weight updating process for the LMS algorithm with and without the proposed convergence booster. It is apparent that the LMS algorithm can effectively improve the convergence rate and meantime further approach the Wiener solution, if the Markov-chain based booster is applied. In Fig. 4, we show the MSE of the estimation error for the parameter a as a function of noise variance. As expected from the previous result of Fig. 3, the MSE is also remarkably reduced, while the convergence rate is improved.

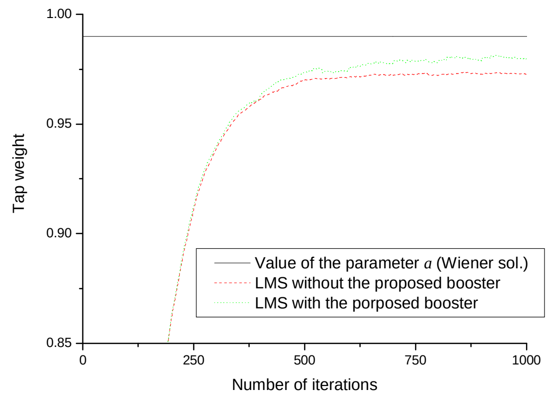

Fig. 2 Rate of convergence improved by the proposed booster

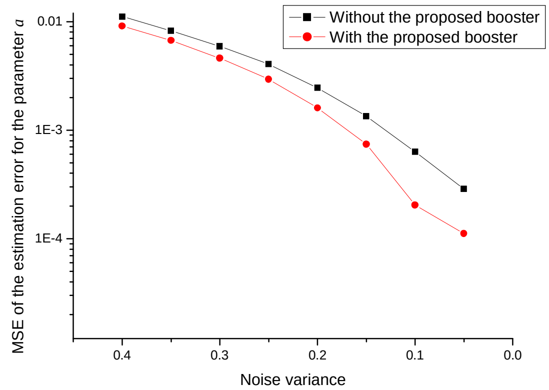

Fig. 3 MSE of the estimation error improved by the proposed booster

VI. CONCLUSIONS

We have proposed a convergence booster based on the concept of Markov chains to accelerate the rate of convergence for LMS algorithms, and meantime approach the Wiener solution. The booster is employed at the moment when the LMS algorithm gets into the FSS. An MSSDE estimator has been proposed to determine the instant when the LMS algorithm enters the FSS. We also formulated the associated fluctuation range. Exploiting the past information in the updating process, and the fact that the transition matrix has a smaller variance, the proposed booster enables the LMS algorithm to track variations in signal characteristics, and improve the convergence rate. Simulation results showed that the proposed convergence accelerator effectively improves both the convergence rate and *MSE* of the estimation error for the LMS algorithm.

REFERENCES

[1] S. Gazor, "Predictions in LMS-type adaptive algorithms for smoothly time-varying environments," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 1735–1739, Jun. 1999.

[2] D. G. Manolakis, V. K. Ingle, and S. M.Kogan, *Statistical and Adaptive Signal Processing*. New York: McGraw-Hill Int. Editions, 2000.

[3] Haykin, S.: 'Adaptive Filter Theory' (Prentice Hall, 1995.)

[4] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[5] Evans, J. B., Xue, P. and Liu, B.: 'Analysis and implementation of variable step size adaptive algorithm', *IEEE Trans. Signal Processing*, 1993, vol. 41, pp. 2517-2535.

[6] Aboulnasr, T. and Mayyas, K.: 'A robust variable step-size LMS-type algorithm: Analysis and simulations', *IEEE Trans. Signal Processing*, 1997, vol. 45, pp. 631-639.

[7] Hosur, S. and Tewfik, A. H., "Wavelet transform domain LMS algorithm," *Proc. ICASSP*, April 1993, Minneapolis, Minnesota, USA, pp. 508-510.

[8] Erdol, N. and Basbug, F., "Performance of wavelet transform based adaptive filters'. Proc. ICASSP, April 1993, Minneapolis, Minnesota, USA, pp. 500-503.

[9] Narayan, S. S., Peterson, A. M. and Narashima, M. J., "Transform domain LMS algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, June. 1983, vol. 31, pp. 4609-615.

[10] Widrow, B., "Fundamental relations between the LMS algorithm and the DFT," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 814-819.

[11] Von Neumann, J., Kent, R. H., Bellinson, H. R. and Habt, B. I., "The mean square successive difference," *Ann. Math. Statist.* Vol. 12, 1941, pp. 153-162.

[12] Ghosh, M. and Meeden, G.: 'On the non-attainability of Chebychev bounds', *American Statistician*, 1977, 31, pp. 35-36.