TRACKING PERFORMANCE OF THE MMAX CONJUGATE GRADIENT ALGORITHM

Bei Xie and Tamal Bose

Wireless @ Virginia Tech
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061 USA

ABSTRACT

Partial update (PU) conjugate gradient (CG) algorithms have been developed to reduce the computational complexity of the full-update CG. Among the basic partial update methods, the MMax CG can achieve convergence performance and steady-state mean-square-error (MSE) performance comparable to the full-update CG in a time-invariant system while significantly reducing the computational complexity. In this paper, the tracking performance of the MMax CG for a time-varying system is studied. Theoretical MSE results of the PU CG are derived for the steady state. The performance of the MMax CG is also compared with the MMax recursive least squares (RLS) algorithm. Computer simulations are presented to support the theoretical analyses.

1. INTRODUCTION

Adaptive filters play an important role in fields related to digital signal processing such as system identification, noise cancellation, and channel equalization. In the real world, the computational complexity of an adaptive filter is an important consideration for applications which need long filters. Usually, least squares algorithms, such as recursive least squares (RLS), Euclidean direction search (EDS) [1], and CG, have higher computational complexity and give better convergence performance than steepest-descent algorithms. Therefore, a tradeoff must be made between computational complexity and performance. To reduce the computational complexity, one option is to use partial update techniques [2]. The partial update adaptive filter only updates part of the coefficient vector instead of updating the entire vector. The theoretical results on the full-update case may not apply to the partial update case. Therefore, performance analysis of the partial update adaptive filter is very meaningful. In the literature, partial update methods have been applied to several adaptive filter algorithms, such as Least Mean Square (LMS), Normalized Least Mean Square (NLMS), RLS, EDS, Affine Projection (AP), Normalized Constant Modulus Algorithm (NCMA), etc. Most analyses are based on LMS and its variants [2]-[8]. There are some analyses for least squares algorithms. In [9], the mean and mean-square performance of the MMax RLS has been analyzed for white inputs. In [7], the tracking performance has been analyzed for MMax RLS. In [10], the mean and mean-square performance of PU EDS is studied. In [11], partial update techniques have been applied to the CG algorithm. The mean and mean-square performance of different PU CG algorithms are analyzed in a time-invariant system. Among the basic partial update methods, the MMax CG can achieve convergence performance and steady-state mean-square-error (MSE) performance comparable to the full-update CG.

In this paper, the tracking performance of the MMax CG for a time-varying system is studied. Theoretical MSE results are derived for the PU CG at steady state. Computer simulation results are also presented to show the tracking performance of the MMax CG. The performance of the MMax CG is also compared with the full-update CG, full-update RLS, and MMax RLS. Analysis of time-varying systems is necessary because the unknown systems in system identification, echo cancellation, and channel equalization are often time-varying in real world applications. This paper is organized as follows. In Section 2, PU CG algorithms are reviewed. The MSE results of the PU CG for a time-varying system are derived in Section 3. In Section 4, computer simulation results are shown.

2. PARTIAL UPDATE CG

The partial update CG is briefly reviewed in this section. A system identification model is shown in Fig. 1. It can be written as:

$$d(n) = \mathbf{x}^T(n)\mathbf{w}^o + v(n), \tag{1}$$

where $d(n)$ is the desired signal, $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ is the input data vector of the unknown system, $\mathbf{w}^o = [w_1^o, w_2^o, ..., w_N^o]^T$ is the impulse response vector of the unknown system, and $v(n)$ is zero-mean white

This work was supported in part by NASA grant #NNG06GE95G and the Institute for Critical Technology and Applied Science.

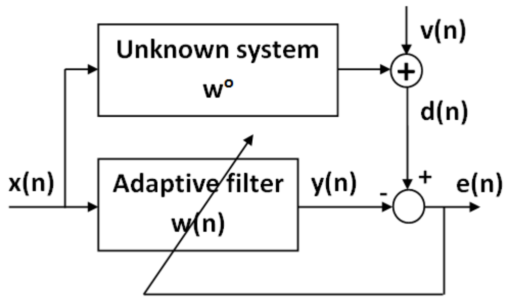

Fig. 1. System identification model.

noise, which is independent of any other signal. In a stationary environment, $\mathbf{w}^o$ is time-invariant. In a non-stationary environment, $\mathbf{w}^o$ is time-varying.

Let $\mathbf{w}$ be the coefficient vector of an adaptive filter. The estimated signal $y(n)$ is defined as

$$y(n) = \mathbf{x}^T(n)\mathbf{w}(n-1), \tag{2}$$

and the output signal error is defined as

$$e(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1). \tag{3}$$

The CG algorithm solves the same least-squares cost function as the RLS. It aims to minimize the the cost function

$$J(n) = \frac{1}{2}\mathbf{w}^T(n)\mathbf{R}\mathbf{w}(n) - \mathbf{b}^T\mathbf{w}(n), \tag{4}$$

where $\mathbf{R}$ is the autocorrelation matrix of the input data vector $\mathbf{x}(n)$ and $\mathbf{b}$ is the cross-correlation vector between the input data vector $\mathbf{x}(n)$ and the desired signal $d(n)$. Unlike RLS, the CG minimizes the cost function using the line search method to avoid matrix inversion. The CG algorithm has two basic implementation methods, reset method and non-reset method. In this paper, we consider only the modified CG algorithm with the non-reset Polak-Ribière (PR) method [12]. The main advantages of this method include the fact that no reset and termination steps are needed and lower computational complexity is needed compared to the reset method. The partial update method aims to reduce the computational cost of the adaptive filters. Instead of updating all of the $N \times 1$ coefficients, it usually only updates $M \times 1$ coefficients, where $M < N$. For the CG algorithm, the calculation of $\mathbf{R}$ results in high computational cost. To reduce the computational complexity, the sub-selected tap-input vector $\hat{\mathbf{x}} = \mathbf{I}_M\mathbf{x}$ is used.

The partial update CG algorithm in an adaptive filter system is summarized as follows [11]:

$$\begin{aligned}
e(n) &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1), && (5)\\
\hat{\mathbf{R}}(n) &= \lambda\hat{\mathbf{R}}(n-1) + \hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n), && (6)\\
\alpha(n) &= \eta\frac{\mathbf{p}^T(n)\mathbf{g}(n-1)}{\mathbf{p}^T(n)\hat{\mathbf{R}}(n)\mathbf{p}(n)}, && (7)\\
\mathbf{w}(n) &= \mathbf{w}(n-1) + \alpha(n)\mathbf{p}(n), && (8)\\
\mathbf{g}(n) &= \lambda\mathbf{g}(n-1) - \alpha(n)\hat{\mathbf{R}}(n)\mathbf{p}(n) \\
&\quad +\hat{\mathbf{x}}(n)e(n), && (9)\\
\beta(n) &= \frac{(\mathbf{g}(n)-\mathbf{g}(n-1))^T\mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}, && (10)\\
\mathbf{p}(n+1) &= \mathbf{g}(n) + \beta(n)\mathbf{p}(n), && (11)
\end{aligned}$$

where

$$\hat{\mathbf{x}} = \mathbf{I}_M\mathbf{x}, \tag{12}$$

and

$$\mathbf{I}_M(n) = \begin{bmatrix} i_1(n) & 0 & \dots & 0 \\ 0 & i_2(n) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & i_N(n) \end{bmatrix}, \tag{13}$$

$$\sum_{k=1}^{N} i_k(n) = M, \quad i_k(n) \in \{0,1\}, \tag{14}$$

For each iteration, only M elements of the input vector are used to update the weights. Note, the calculation of output signal error still uses the the whole input vector, not the subselected input vector. Basic partial update methods include sequential PU, stochastic PU, MMax, etc. According to [11], the MMax CG has the fastest convergence rate and can achieve similar steady-state MSE to the full-update CG in a time-invariant system. Therefore, only the MMax CG is studied for a time-varying system in this paper. The MMax CG selects the input vector according to the first M max elements of the input $\mathbf{x}$. The condition of $i_k(n)$ [2] becomes

$$i_k(n) = \begin{cases} 1 & \text{if } |\mathbf{x}_k(n)| \in \max_{1\le l\le N}\{|\mathbf{x}_l(n)|, M\} \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

The sorting of the input $\mathbf{x}$ increases the computational complexity. The sorting result can be achieved more efficiently by using SORTLINE or Short-sort methods [12]. If the SORTLINE method is used, the MMax CG needs $2N^2 + M^2 + 9N + M + 3$ multiplications and $2 + 2log_2N$ comparisons.

3. TRACKING PERFORMANCE OF PARTIAL UPDATE CG

In a non-stationary environment, the unknown system is time-varying. The desired signal can be rewritten as

$$d(n) = \mathbf{x}^T(n)\mathbf{w}^o(n) + v(n). \tag{16}$$

A first-order Markov model [13] is used for the time-varying unknown system. It has the form as follows:

$$\mathbf{w}^o(n) = \gamma \mathbf{w}^o(n-1) + \eta(n), \tag{17}$$

where γ is a fixed parameter of the model and is assumed to be very close to unity. $\eta(n)$ is the process noise vector with zero mean and correlation matrix $\mathbf{R}_\eta$.

The coefficient error vector is defined as

$$\mathbf{z}(n) = \mathbf{w}(n) - \mathbf{w}^o(n). \tag{18}$$

To determine the tracking performance of partial update CG, three more assumptions are needed: (1) Noise $v(n)$ has zero mean and variance σ_v^2, and is independent of the noise $\eta(n)$; (2) The input signal $\mathbf{x}(n)$ is independent of both noise $v(n)$ and noise $\eta(n)$; (3) At steady state, the coefficient error vector $\mathbf{z}(n)$ is very small and is independent of the input signal $\mathbf{x}(n)$.

Using these assumptions, the MSE equation of the PU CG algorithm at steady state becomes

$$E\{|e(n)|^2\} = \sigma_v^2 + tr(\mathbf{R}E\{\mathbf{z}(n)\mathbf{z}^T(n)\}), \tag{19}$$

where $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$ is the autocorrelation matrix of the input $\mathbf{x}$. At steady state, the coefficient vector is approximate to [11]

$$\mathbf{w}(n) \approx \widetilde{\mathbf{R}}^{-1}(n)\hat{\mathbf{b}}(n), \tag{20}$$

where

$$\begin{aligned} \widetilde{\mathbf{R}}(n) &= \lambda\widetilde{\mathbf{R}}(n-1) + \hat{\mathbf{x}}(n)\mathbf{x}^T(n) \\ &= \sum_{i=1}^{n} \lambda^{n-i}\hat{\mathbf{x}}(i)\mathbf{x}^T(i), \qquad (21) \\ \hat{\mathbf{b}}(n) &= \lambda\hat{\mathbf{b}}(n-1) + \hat{\mathbf{x}}(n)d(n). \qquad (22) \end{aligned}$$

Take the expectation of (21),

$$\begin{aligned} E\{\widetilde{\mathbf{R}}(n)\} &= \sum_{i=1}^{n} \lambda^{n-i}E\{\hat{\mathbf{x}}(i)\mathbf{x}^T(i)\} \\ &= \sum_{i=1}^{n} \lambda^{n-i}\widetilde{\mathbf{R}} \\ &= \frac{\widetilde{\mathbf{R}}}{1-\lambda} \qquad n \to \infty, \end{aligned} \tag{23}$$

where $\widetilde{\mathbf{R}} = E\{\hat{\mathbf{x}}(n)\mathbf{x}^T(n)\}$. Assuming a slow adaptive process (λ is very close to unity), $\widetilde{\mathbf{R}}(n)$ becomes [13]

$$\widetilde{\mathbf{R}}(n) \approx \frac{\widetilde{\mathbf{R}}}{1-\lambda} \qquad n \to \infty. \tag{24}$$

The coefficient vector at steady state is further approximated to

$$\begin{aligned} \mathbf{w}(n) &\approx (1-\lambda)\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{b}}(n) \\ &= (1-\lambda)\widetilde{\mathbf{R}}^{-1}(\lambda\hat{\mathbf{b}}(n-1) + \hat{\mathbf{x}}(n)d(n)) \\ &= \lambda\mathbf{w}(n-1) + (1-\lambda)\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{x}}(n)\mathbf{x}(n)\mathbf{w}^o(n) \\ &+ (1-\lambda)\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{x}}(n)\mathbf{v}(n). \end{aligned} \tag{25}$$

Subtracting $\mathbf{w}^o(n)$ from both sides of (25), using (17) and (18), using the direct-averaging method [13], and applying the assumption that γ in (17) is very close to unity, we get

$$\mathbf{z}(n) \approx \lambda\mathbf{z}(n-1) - \lambda\eta(n) + (1-\lambda)\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{x}}(n)\mathbf{v}(n). \tag{26}$$

Note, the term $(1-\lambda)\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{x}}(n)\mathbf{x}(n)\mathbf{w}^o(n)$ in (25) becomes $(1-\lambda)\widetilde{\mathbf{R}}^{-1}E\{\hat{\mathbf{x}}(n)\mathbf{x}(n)\}\mathbf{w}^o(n) = (1-\lambda)\mathbf{w}^o(n)$ after using the direct-averaging method. Define the weight error correlation matrix as

$$\mathbf{K}(n) = E\{\mathbf{z}(n)\mathbf{z}^T(n)\}. \tag{27}$$

Since the input noise is white,

$$E\{v(i)v(j)\} = \begin{cases} \sigma_v^2 & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}. \tag{28}$$

Using these assumptions, $\mathbf{K}(n)$ becomes

$$\begin{aligned} \mathbf{K}(n) &\approx \lambda^2\mathbf{K}(n-1) + \lambda^2\mathbf{R}_\eta \\ &+ \sigma_v^2(1-\lambda)^2E\{\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\widetilde{\mathbf{R}}^{-T}\} \end{aligned} \tag{29}$$

At steady state $\mathbf{K}(n) \approx \mathbf{K}(n-1)$, therefore $\mathbf{K}(n)$ becomes

$$\begin{aligned} \mathbf{K}(n) &\approx \frac{1-\lambda}{1+\lambda}\sigma_v^2\widetilde{\mathbf{R}}^{-1}E\{\hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\}\widetilde{\mathbf{R}}^{-T} \\ &+ \frac{\lambda^2}{1-\lambda^2}\mathbf{R}_\eta. \end{aligned} \tag{30}$$

The MSE equation becomes

$$\begin{aligned} E\{|e(n)|^2\} &\approx \sigma_v^2 + tr(\mathbf{R}(\frac{1-\lambda}{1+\lambda}\sigma_v^2\widetilde{\mathbf{R}}^{-1}\hat{\mathbf{R}}\widetilde{\mathbf{R}}^{-T} \\ &+ \frac{\lambda^2}{1-\lambda^2}\mathbf{R}_\eta)), \end{aligned} \tag{31}$$

where $tr(\cdot)$ is the trace operator and $\hat{\mathbf{R}} = E\{\hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\}$.

For a white input signal with variance σ_x^2, the MSE can be simplified as

$$\begin{aligned} E\{|e(n)|^2\} &\approx \sigma_v^2 + \frac{N(1-\lambda)}{1+\lambda}\sigma_v^2\sigma_x^2\sigma_{\hat{x}}^2\sigma_{\tilde{x}}^{-4} \\ &+ \frac{\lambda^2}{1-\lambda^2}\sigma_x^2 tr(\mathbf{R}_\eta), \end{aligned} \tag{32}$$

where $\sigma_{\hat{x}}^2\mathbf{I} = E\{\hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\}$ and $\sigma_{\tilde{x}}^{-2}\mathbf{I} = \widetilde{\mathbf{R}}^{-1}$.

For the MMax method and a white input signal, $\sigma_{\hat{x}}^2 \approx \kappa\sigma_x^2$ and $\sigma_{\tilde{x}}^2 \approx \kappa\sigma_x^2$, where κ is smaller than 1, but is close to 1. Therefore, the MSE can be further simplified as

$$E\{|e(n)|^2\} \approx \sigma_v^2 + \frac{N(1-\lambda)}{(1+\lambda)\kappa}\sigma_v^2 + \frac{\lambda^2}{1-\lambda^2}\sigma_x^2 tr(\mathbf{R}_\eta). \tag{33}$$

Assume the process noise is white with variance σ_η^2. Then, the MSE of MMax CG can be further simplified as

$$E\{|e(n)|^2\} \approx \sigma_v^2 + \frac{N(1-\lambda)}{(1+\lambda)\kappa}\sigma_v^2 + \frac{N\lambda^2}{1-\lambda^2}\sigma_x^2\sigma_\eta^2. \tag{34}$$

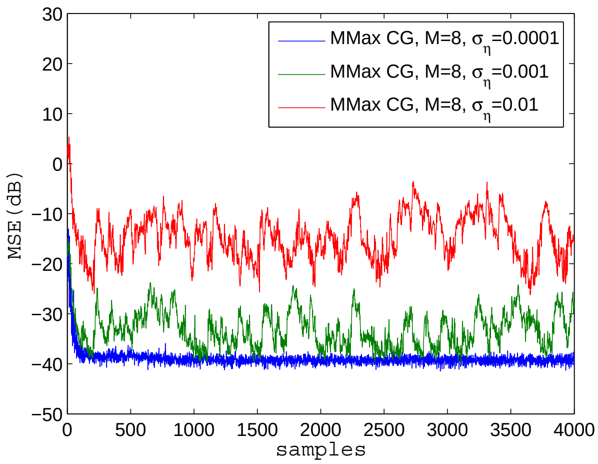

Fig. 2. Comparison of MSE of MMax CG for varying process noise η, $M = 8$.

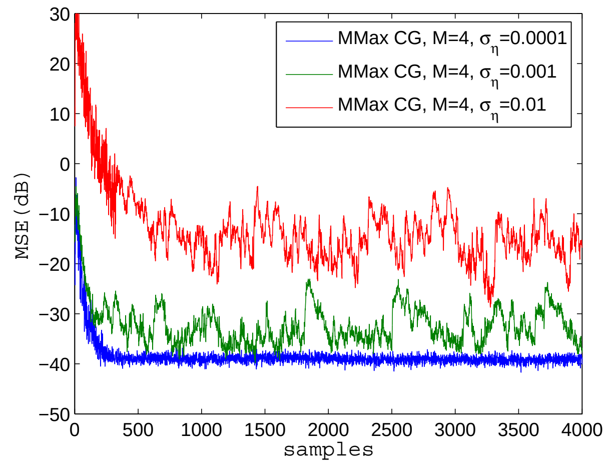

Fig. 3. Comparison of MSE of MMax CG for varying process noise η, $M = 4$.

4. SIMULATIONS

4.1. Tracking performance of the MMax CG using the first-order Markov model

The system identification model is shown in Fig. 1. The first-order Markov model (17) is used for the time-varying impulse response. It is a 16-order FIR filter (N=16). The initial state of the impulse response [6] is

$$\mathbf{w}^o(0) = [0.01, 0.02, -0.04, -0.08, 0.15, -0.3, 0.45, 0.6, \\ 0.6, 0.45, -0.3, 0.15, -0.08, -0.04, 0.02, 0.01]^T. \quad (35)$$

In our simulations, the lengths of the partial update filter are M=8 and M=4. The variance of the input noise $v(n)$ is 0.0001. The initial weights of the CG are $\mathbf{w} = \mathbf{0}$ and the initial autocorrelation matrix $\mathbf{R}(0) = \mathbf{0}$. The parameters λ and η of the CG are equal to 0.99 and 0.6, respectively. The initial residue vector is set to be $\mathbf{g}(0) = d(1)\mathbf{x}(1)$. The results are obtained by averaging 100 independent runs.

Fig. 2 and Fig. 3 show the tracking performance of the MMax CG with different process noise η for $M = 8$ and $M = 4$, respectively. The parameter γ in the first Markov model is 0.9998. The white input signal with unity variance is used. The white process noise is used with difference variances. We can see that the MSE of MMax CG increases when the process noise increases. The variance of the MSE also increases when the process noise increases. The same situation also happens to the full-update CG. However, the partial update length does not have much effect on the MSE results in this case. The partial update length only affects the convergence rate. The convergence rate decreases as the partial update length decreases.

Table 1 and Table 2 show the simulated MSE and theoretical MSE of MMax CG algorithms at steady state for white input. The simulated results are obtained by taking the time average over the last 1000 samples. The theoretical results are calculated from (34). The partial-update lengths are $M = 8$ and $M = 4$. We can see that the theoretical results match the simulated results.

Table 1. The simulated MSE and theoretical MSE of MMax CG for varying process noise η, $M = 8$.

Process noise σ_η	Simulated MSE (dB)	Theoretical MSE (dB)
0.0001	-39.2381	-39.3584
0.001	-32.9019	-30.4766
0.01	-13.4403	-11.0287

Table 2. The simulated MSE and theoretical MSE of MMax CG for varying process noise η, $M = 4$.

Process noise σ_η	Simulated MSE (dB)	Theoretical MSE (dB)
0.0001	-38.9965	-39.0672
0.001	-31.2768	-30.4378
0.01	-11.6397	-11.0282

4.2. Performance comparison of the MMax CG with the CG, RLS, and MMax RLS

The tracking performance of the MMax CG is also compared with the full-update CG, full-update RLS, and MMax RLS. The same system identification model is used. After 2000 samples/iterations pass, the unknown system in (35) is changed by multiplying all coefficients by -1. Fig. 4 and Fig. 5 show the MSE results among CG, MMax CG, RLS, and

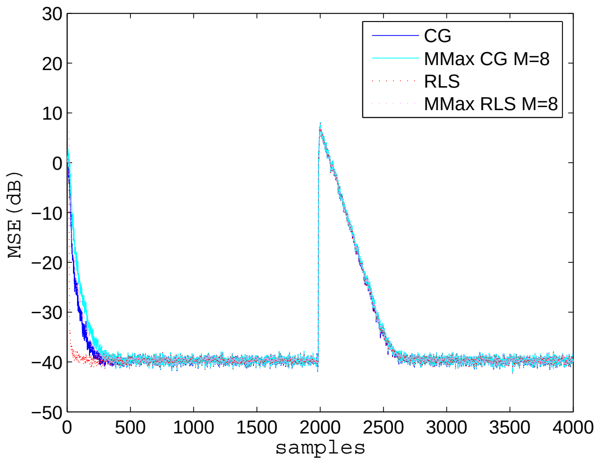

Fig. 4. Comparison of MSE of MMax CG with CG, RLS, MMax RLS for white input, N=16, M=8.

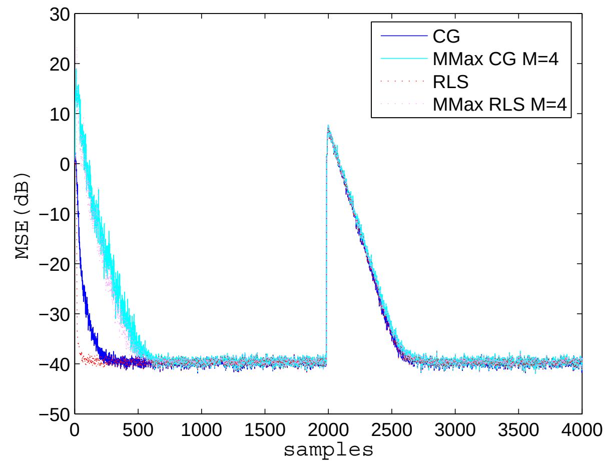

Fig. 5. Comparison of MSE of MMax CG with CG, RLS, MMax RLS for white input, N=16, M=4.

MMax RLS, when $M = 8$ and $M = 4$, respectively. White input is used. The results show that the four algorithms have a similar convergence rate after the unknown system changes. It is also shown that the MMax CG and MMax RLS with $M = 4$ can have a similar convergence rate to the MMax CG and MMax RLS with $M = 8$ after the unknown system change. The partial update length only affects the convergence rate at the beginning in this case. This is because $\mathbf{R}(n)$ gives a worse estimation of the real autocorrelation matrix $\mathbf{R}$ at the beginning when partial update length decreases. If the SORTLINE sorting method is used for both MMax CG and MMax RLS, the total number of multiplications of MMax CG and RLS are $2N^2 + M^2 + 9N + M + 3$ and $2N^2 + 2NM + 3N + M + 1$, respectively. In this case, the full update length N is 16. The partial update length M is 8 and 4, respectively. The detailed computational complexities of the four algorithms are shown in Table 3. The results show that the MMax CG with M=4 can achieve similar tracking performance to the full-update RLS or CG while reducing the computational complexity significantly.

Table 3. The computational complexities of CG, MMax CG, RLS, and MMax RLS.

Algorithms	Number of multiplications per symbol	Number of comparisons per symbol
CG (N=16)	3003	–
MMax CG (M=8)	731	10
MMax CG (M=4)	679	10
RLS (N=16)	3721	–
MMax RLS (M=8)	825	10
MMax RLS (M=4)	693	10

5. CONCLUSION

In this paper, the tracking performance of the MMax CG is analyzed. The MSE expression of the partial update CG is derived for a time-varying system. The MSE of MMax CG is further simplified with white inputs. Simulation results agree with the derived theoretical results in steady state. The tracking performance of the MMax CG is also compared with the full-update CG, full-update RLS, and MMax RLS. The results show that the MMax CG can achieve similar tracking performance to the full-update CG and full-update RLS, while reducing the computational complexity significantly.

6. REFERENCES

[1] T. Bose, *Digital Signal and Image Processing*. New York: Wiley, 2004.

[2] K. Doğançay, *Partial-Update Adaptive Signal Processing*. Academic Press, 2008.

[3] S. C. Douglas, "Analysis and implementation of the max-NLMS adaptive filter," in *Proc. Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, 1995, pp. 659-663.

[4] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209-216, Mar. 1997.

[5] M. Godavarti and A. O. Hero III, "Partial update LMS algorithms," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2382-2399, 2005.

[6] K. Mayyas, "Performance analysis of the deficient length LMS adaptive algorithm," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2727-2734, Aug. 2005.

[7] A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive filtering with performance analysis for identification of time-varying

systems," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1681-1695, July 2007.

[8] J. Wu and M. Doroslovacki, "A mean Convergence analysis for partial update NLMS algorithms," in *Proc. CISS*, 2007, pp. 31-34.

[9] P. A. Naylor and A. W. H. Khong, "Affine projection and recursive least squares adaptive filters employing partial updates," in *Proc. Thirty-Eighth Asilomar Conference on Signal, System & Computers*, 2004, pp. 950-954.

[10] B. Xie and T. Bose, "Partial update EDS algorithms for adaptive filtering," in *Proc. ICASSP*, 2010.

[11] B. Xie and T. Bose, "Partial update conjugate gradient algorithms for adaptive filtering," in *Proc. PECCS*, 2011.

[12] P. S. Chang and A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Trans. Signal Processing*, vol. 48, no. 2, pp. 409-418, Feb. 2000.

[13] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice Hall, 2002.