PARTIAL UPDATE CONJUGATE GRADIENT ALGORITHMS FOR ADAPTIVE FILTERING

Bei Xie and Tamal Bose
Wireless @ Virginia Tech, Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

Keywords: Conjugate gradient adaptive filter, Partial update, Recursive algorithms.

Abstract: In practice, computational complexity is an important consideration of an adaptive signal processing system. A well-known approach to controlling computational complexity is applying partial update (PU) adaptive filters. In this paper, a partial update conjugate gradient (CG) algorithm is employed. Theoretical analyses of mean and mean-square performance are presented. The simulation results of different PU CG algorithms are shown. The performance of PU CG algorithms are also compared with PU recursive least squares (RLS) and PU Euclidean direction search (EDS) algorithms.

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 RLS, EDS (Bose, 2004), and CG, have higher computational complexity and give better convergence performance than the steepest-descent algorithms. Therefore, a tradeoff must be made between computational complexity and performance. One option is to use partial update techniques (Doğançay, 2008) to reduce the computational complexity. 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 filters, 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 (Douglas, 1995), (Douglas, 1997), (Godavarti and Hero III, 2005), (Mayyas, 2005), (Khong and Naylor, 2007), (Wu and Doroslovacki, 2007), (Doğançay, 2008). There are some analyses for least squares algorithms. In (Naylor and Khong, 2004), the mean and mean-square performance of the MMax RLS has been analyzed for white inputs. In (Khong and Naylor, 2007), the tracking performance has been analyzed for MMax RLS. In (Xie and Bose, 2010), the mean and mean-square performance of PU EDS are studied.

In this paper, partial update techniques are applied to the CG algorithm. CG solves the same cost function as the RLS algorithm. It has a fast convergence rate and can achieve the same mean-square performance as RLS at steady state. It has lower computational complexity when compared with the RLS algorithm. The EDS algorithm is a simplified CG algorithm, and it has lower computational complexity than the CG algorithm. The basic partial update methods such as periodic PU, sequential PU, stochastic PU, and MMax update method, are applied to the CG algorithm. The mean and mean-square performance of different PU CG are analyzed, and compared with the full-update CG algorithm. The goal of this paper is to find one or more PU CG algorithms which can reduce the computational complexity while maintaining good performance. In Section 2, different PU CG algorithms are developed. Theoretical mean and mean-square analyses of PU CG for white input are given in Section 3. In Section 4, computer simulation results are shown. The performance of different PU CG algorithms are compared. The performance of PU CG, PU RLS, and PU EDS are also compared.

Xie B. and Bose T. (2011).
PARTIAL UPDATE CONJUGATE GRADIENT ALGORITHMS FOR ADAPTIVE FILTERING.
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems*, pages 317-323
DOI: 10.5220/0003364103170323

2 CG AND PARTIAL UPDATE CG

The basic adaptive filter system model can be written as:

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

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, $\mathbf{w}^*(n) = [w_1^*, w_2^*, ..., w_N^*]^T$ is the impulse response vector of the unknown system, and $v(n)$ is zero-mean white noise, which is independent of any other signal.

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

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

Since the CG algorithm with reset method needs higher computational complexity than the non-reset method, we consider only the CG with non-reset Polak-Ribière (PR) method. The CG with PR method (Chang and Willson, 2000) is summarized as follows:

Initial conditions:
$\mathbf{w}(0) = \mathbf{0}$, $\mathbf{R}(n) = \mathbf{0}$, $\mathbf{p}(1) = \mathbf{g}(0)$.

$$\begin{aligned}
e(n) &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1), && (4)\\
\mathbf{R}(n) &= \lambda\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n), && (5)\\
\alpha(n) &= \eta\frac{\mathbf{p}^T(n)\mathbf{g}(n-1)}{\mathbf{p}^T(n)\mathbf{R}(n)\mathbf{p}(n)}, && (6)\\
\mathbf{w}(n) &= \mathbf{w}(n-1) + \alpha(n)\mathbf{p}(n), && (7)\\
\mathbf{g}(n) &= \lambda\mathbf{g}(n-1) - \alpha(n)\mathbf{R}(n)\mathbf{p}(n)\\
&\quad + \mathbf{x}(n)e(n), && (8)\\
\beta(n) &= \frac{(\mathbf{g}(n) - \mathbf{g}(n-1))^T\mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}, && (9)\\
\mathbf{p}(n+1) &= \mathbf{g}(n) + \beta(n)\mathbf{p}(n), && (10)
\end{aligned}$$

where $\mathbf{R}$ is the time-average correlation matrix of $\mathbf{x}$, $\mathbf{p}$ is the search direction, and $\mathbf{g}$ is the residue vector which is also equal to $\mathbf{b}(n) - \mathbf{R}(n)\mathbf{w}(n)$, where $\mathbf{b}(n) = \lambda\mathbf{b}(n-1) + \mathbf{x}(n)d(n)$ is the estimated crosscorrelation of $\mathbf{x}$ and d. The choice of $\mathbf{g}(0)$ can be $d(1)\mathbf{x}(1)$ or satisfies $\mathbf{g}^T(0)\mathbf{g}(0) = 1$. λ is the forgetting factor and the constant parameter η satisfies $\lambda - 0.5 \leq \eta \leq \lambda$.

The partial update method aims to reduce the computational cost of the adaptive filters. Instead of updating all the $N \times 1$ coefficients, it usually only updates $M \times 1$ coefficients, where $M < N$. Basic partial update methods include periodic PU, sequential PU, stochastic PU, and MMax update method, etc. These methods will be applied to the CG algorithm. For the CG algorithm, the calculation of $\mathbf{R}$ needs high computational cost. To reduce the computational complexity, the subselected tap-input vector $\hat{\mathbf{x}} = \mathbf{I}_M\mathbf{x}$ is used.

The partial update CG algorithm is summarized as follows:

$$\begin{aligned}
e(n) &= d(n) - \mathbf{x}^T(n)\mathbf{w}(n-1), && (11)\\
\hat{\mathbf{R}}(n) &= \lambda\hat{\mathbf{R}}(n-1) + \hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n), && (12)\\
\alpha(n) &= \eta\frac{\mathbf{p}^T(n)\mathbf{g}(n-1)}{\mathbf{p}^T(n)\hat{\mathbf{R}}(n)\mathbf{p}(n)}, && (13)\\
\mathbf{w}(n) &= \mathbf{w}(n-1) + \alpha(n)\mathbf{p}(n), && (14)\\
\mathbf{g}(n) &= \lambda\mathbf{g}(n-1) - \alpha(n)\hat{\mathbf{R}}(n)\mathbf{p}(n)\\
&\quad + \hat{\mathbf{x}}(n)e(n), && (15)\\
\beta(n) &= \frac{(\mathbf{g}(n) - \mathbf{g}(n-1))^T\mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}, && (16)\\
\mathbf{p}(n+1) &= \mathbf{g}(n) + \beta(n)\mathbf{p}(n), && (17)
\end{aligned}$$

where

$$\hat{\mathbf{x}} = \mathbf{I}_M\mathbf{x}, \tag{18}$$

and

$$\mathbf{I}_M(n) = \begin{bmatrix} i_1(n) & 0 & \dots & 0 \\ 0 & i_2(n) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & i_N(n) \end{bmatrix}, \tag{19}$$

$$\sum_{k=1}^{N} i_k(n) = M, \quad i_k(n) \in \{0, 1\}, \tag{20}$$

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. The total number of multiplications of full-update CG is $3N^2 + 10N + 3$ per sample. The total number of multiplications of partial-update CG is reduced to $2N^2 + M^2 + 9N + M + 3$ per sample. The computational complexity of different PU methods is not considered here.

2.1 Periodic Partial Update CG

The periodic partial update method (Doğançay, 2008) only updates the coefficients at every S^{th} iteration and copies the coefficients at the other iterations. The periodic PU CG updates the weights at every S^{th} iteration. The update equation for periodic PU CG can be written as:

$$\mathbf{w}(nS) = \mathbf{w}((n-1)S) + \alpha(nS)\mathbf{p}(nS), \tag{21}$$

where $S = \lceil \frac{N}{M} \rceil$, which is the ceiling of $\frac{N}{M}$. Since the periodic PU CG still uses the whole input vector, the steady-state performance will be the same as the full-update CG. However, the convergence rate of the periodic PU CG will be S times slower than the full-update CG.

2.2 Sequential Partial Update CG

The sequential partial update method (Doğançay, 2008) designs $i_k(n)$ as:

$$i_k(n) = \begin{cases} 1 & \text{if } k \in K_{n \bmod S+1} \\ 0 & \text{otherwise} \end{cases}, \tag{22}$$

where $K_1 = \{1,2,\ldots,M\}$, $K_2 = \{M+1, M+2,\ldots,2M\}$, ... , $K_S = \{(S-1)M+1,(S-1)M+2,\ldots,N\}$.

2.3 Stochastic Partial Update CG

The stochastic partial update CG chooses input vector subsets randomly. The $i_k(n)$ becomes

$$i_k(n) = \begin{cases} 1 & \text{if } k \in K_{m(n)} \\ 0 & \text{otherwise} \end{cases}, \tag{23}$$

where $m(n)$ is a random process with probability mass function (Doğançay, 2008):

$$Pr\{m(n) = i\} = p_i, \quad i = 1,\ldots,S, \quad \sum_{i=1}^{S} p_i = 1. \tag{24}$$

Usually a uniformly distributed random process will be applied. Therefore, for each iteration, M of N input elements will be chosen with probability $p_i = 1/S$.

2.4 MMax CG

The MMax CG selects the input vector according to the first M max elements of the input $\mathbf{x}$. The condition of $i_k(n)$ (Doğançay, 2008) becomes

$$i_k(n) = \begin{cases} 1 & \text{if } |\mathbf{x}_k(n)| \in \max_{1\le l\le N}\{|\mathbf{x}_l(n)|, M\} \\ 0 & \text{otherwise} \end{cases}. \tag{25}$$

The sorting of the input $\mathbf{x}$ increases the computational complexity. It can be achieved efficiently using the SORTLINE or Short-sort methods (Chang and Willson, 2000). The SORTLINE method needs $2 + 2log_2 N$ multiplications.

3 PERFORMANCE ANALYSIS OF PARTIAL UPDATE CG

The normal equation of the partial update CG algorithm can be represented as:

$$\mathbf{X}_M^T(n)\Lambda(n)\mathbf{X}(n)\mathbf{w}(n) = \mathbf{X}_M^T(n)\Lambda(n)\mathbf{d}(n), \tag{26}$$

where $\mathbf{d}(n) = [d(n), d(n-1),\ldots,d(1)]^T$,

$$\mathbf{X}_M(n) = \begin{bmatrix} \hat{\mathbf{x}}_M^T(n) \\ \hat{\mathbf{x}}_M^T(n-1) \\ \vdots \\ \hat{\mathbf{x}}_M^T(1) \end{bmatrix}, \tag{27}$$

and

$$\Lambda(n) = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & \lambda & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \lambda^n \end{bmatrix}. \tag{28}$$

Therefore, the residue vector $\mathbf{g}$ can also be written as

$$\mathbf{g} = \hat{\mathbf{b}}(n) - \widetilde{\mathbf{R}}(n)\mathbf{w}(n), \tag{29}$$

where

$$\widetilde{\mathbf{R}}(n) = \lambda\widetilde{\mathbf{R}}(n-1) + \hat{\mathbf{x}}(n)\mathbf{x}^T(n), \tag{30}$$

$$\hat{\mathbf{b}}(n) = \lambda\hat{\mathbf{b}}(n-1) + \hat{\mathbf{x}}(n)d(n). \tag{31}$$

To simplify the analysis, we assume that the input signal is wide-sense stationary and ergodic, and $\alpha(n)$, $\beta(n)$, $\widetilde{\mathbf{R}}(n)$, and $\mathbf{w}(n)$ are uncorrelated to each other (Chang and Willson, 2000). Apply the expectation operator to (14), (15), and (17). Define $E\{\alpha(n)\} = \bar{\alpha}$, $E\{\beta(n)\} = \beta$, $E\{\hat{\mathbf{b}}(n)\} = \hat{\mathbf{b}}$, and $E\{\widetilde{\mathbf{R}}(n)\} = \widetilde{\mathbf{R}}$. The system can be viewed as linear and time invariant at steady state. Therefore, the $\mathscr{Z}$-transform can be applied to the system. Define $\mathbf{W}(z) = \mathscr{Z}\{E\{\mathbf{w}(n)\}\}$, $\mathbf{G}(z) = \mathscr{Z}\{E\{\mathbf{g}(n)\}\}$, and $\mathbf{P}(z) = \mathscr{Z}\{E\{\mathbf{p}(n)\}\}$. Equations (14), (15), and (17) become

$$\mathbf{W}(z) = \mathbf{W}(z)z^{-1} + \bar{\alpha}\mathbf{P}(z), \tag{32}$$

$$\mathbf{G}(z) = \frac{\hat{\mathbf{b}}z}{z-1} - \widetilde{\mathbf{R}}\mathbf{W}(z), \tag{33}$$

$$z\mathbf{P}(z) = \mathbf{G}(z) + \bar{\beta}\mathbf{P}(z). \tag{34}$$

Therefore,

$$\mathbf{W}(z) = [(z-1)(z-\beta)\mathbf{I} + \alpha\widetilde{\mathbf{R}}z]^{-1}\frac{\alpha\hat{\mathbf{b}}z^2}{z-1}. \tag{35}$$

Since the system is causal and $n \ge 0$, the $\mathscr{Z}$-transform is one-sided and $\mathbf{W}(z) = \mathbf{W}^+(z)$. At steady state, the mean of weights converge to

$$\lim_{n\to\infty} E\{\mathbf{w}(n)\} = \lim_{z\to 1}\mathbf{W}^+(z) = \widetilde{\mathbf{R}}^{-1}\hat{\mathbf{b}}. \tag{36}$$

For the causal system to be stable, all the poles must be inside the unit circle. Therefore, the conditions for the stability are $|\bar{\beta}| < 1$ and $0 \le \bar{\alpha} \le \frac{2\bar{\beta}+2}{\lambda_{max}}$, where

λ_{max} is the maximal eigenvalue of $\widetilde{\mathbf{R}}$. For the sequential and stochastic methods, the partial update correlation matrix $\hat{\mathbf{R}}$ may become ill-conditioned, especially when M becomes smaller, and the algorithm may suffer convergence difficulty. More sophisticated analysis will be provided in a future paper.

Since the input noise $v(n)$ is assumed to be zero mean white noise and independent of the input signal $\mathbf{x}(n)$, the MSE equation of the PU CG algorithm becomes

$$E\{|e(n)|^2\} = \sigma_v^2 + tr(\mathbf{R}E\{\varepsilon(n)\varepsilon^T(n)\}), \qquad (37)$$

where $\sigma_v^2 = E\{v^2(n)\}$ is the variance of the input noise, and $\varepsilon(n) = \mathbf{w}^* - \mathbf{w}(n)$ is the weight error vector. To simplify the analysis, it is also assumed that the weight error $\varepsilon(n)$ is independent of the input signal $\mathbf{x}(n)$ at steady state, and the input signal is white. At steady state,

$$\mathbf{w}(n) \approx \widetilde{\mathbf{R}}^{-1}(n)\hat{\mathbf{b}}(n). \qquad (38)$$

Using (30), (31), and (1), $\mathbf{w}(n)$ can be written as

$$\mathbf{w}(n) \approx \mathbf{w}^* + \widetilde{\mathbf{R}}^{-1}(n)\sum_{i=1}^{n}\lambda^{n-i}\hat{\mathbf{x}}(i)v(i). \qquad (39)$$

Define the weight error correlation matrix as

$$\begin{aligned}\mathbf{K}(n) &= E\{\varepsilon(n)\varepsilon^T(n)\}\\ &= E\{(\mathbf{w}^* - \mathbf{w}(n))(\mathbf{w}^* - \mathbf{w}(n))^T\}. \qquad (40)\end{aligned}$$

Substituting (39) into (40) and applying the assumptions, we get

$$\begin{aligned}\mathbf{K}(n) \approx E\{\widetilde{\mathbf{R}}^{-1}(n)\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda^{n-i}\lambda^{n-j}\hat{\mathbf{x}}(i)\hat{\mathbf{x}}^T(j)\widetilde{\mathbf{R}}^{-T}(n)\}\\ E\{v(i)v(j)\}. \qquad (41)\end{aligned}$$

Since the input noise is white,

$$E\{v(i)v(j)\} = \begin{cases}\sigma_v^2 & \text{for } i=j\\ 0 & \text{otherwise}\end{cases}. \qquad (42)$$

Therefore, $\mathbf{K}(n)$ becomes

$$\begin{aligned}\mathbf{K}(n) &\approx \sigma_v^2 E\{\widetilde{\mathbf{R}}^{-1}(n)\sum_{i=1}^{n}\lambda^{2(n-i)}\hat{\mathbf{x}}(i)\hat{\mathbf{x}}^T(j)\widetilde{\mathbf{R}}^{-T}(n)\}\\ &= \sigma_v^2\widetilde{\mathbf{R}}^{-1}\hat{\hat{\mathbf{R}}}\widetilde{\mathbf{R}}^{-T}, \qquad (43)\end{aligned}$$

where $\hat{\hat{\mathbf{R}}} = E\{\sum_{i=1}^{n}\lambda^{2(n-i)}\hat{\mathbf{x}}(i)\hat{\mathbf{x}}^T(j)\}$.

The MSE equation becomes

$$E\{|e(n)|^2\} \approx \sigma_v^2 + \sigma_x^2\sigma_v^2 tr(\widetilde{\mathbf{R}}^{-1}\hat{\hat{\mathbf{R}}}\widetilde{\mathbf{R}}^{-T}), \qquad (44)$$

where $tr(\cdot)$ is the trace operator, and $\sigma_x^2 = tr(\mathbf{R})$ is the variance of the white input signal.

4 SIMULATIONS

4.1 Performance of Different PU CG Algorithms

The convergence performance and mean-square performance of different partial update CG algorithms are compared in a system identification application. The system identification model is shown in Figure 1 and is taken from (Zhang et al., 2006). The unknown system (Mayyas, 2005) is a 16-order FIR filter (N=16), with impulse response

$$\begin{aligned}\mathbf{w}^* = [&0.01, 0.02, -0.04, -0.08, 0.15, -0.3, 0.45, 0.6,\\ &0.6, 0.45, -0.3, 0.15, -0.08, -0.04, 0.02, 0.01]^T.\end{aligned}$$

In our simulations, the length of the partial update filter is M=8. The variance of the input noise $v(n)$ is $\varepsilon_{min} = 0.0001$. The initial weights are $\mathbf{w} = \mathbf{0}$. The parameters λ and η of CG are equal to 0.9 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.

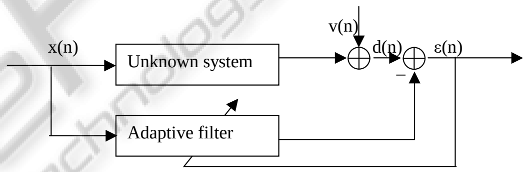

Figure 1: System identification model.

The correlated input of the system (Zhang et al., 2006) has the following form

$$x(n) = 0.8x(n-1) + \zeta(n), \qquad (45)$$

where $\zeta(n)$ is zero-mean white Gaussian noise with unit variance.

Figure 2 and Figure 3 show the mean-square error (MSE) performance of the partial update CG for the correlated input and white input, respectively. From the figures, the steady-state MSE of the periodic PU CG is the same as that of CG. The convergence rate of the periodic PU CG algorithm is about $N/M = 2$ times slower than the full-update CG algorithm. The MMax CG converges a little faster than the periodic CG in this case. The steady-state MSE of MMax CG is close to the full-update CG. The sequential and stochastic PU CG have higher MSE than the full-update CG at steady state. Their convergence rates are slow.

Figure 4 shows the mean convergence of the weights at steady state for MMax CG. The PU length is 8, and the input signal is white. The theoretical results are calculated from (36). We can see that

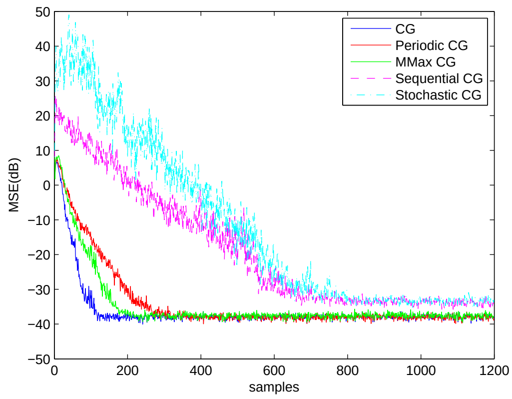

Figure 2: Comparison of MSE of PU CG with correlated input.

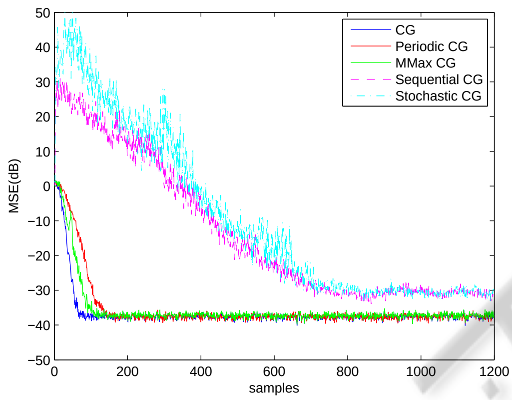

Figure 3: Comparison of MSE of PU CG with white input.

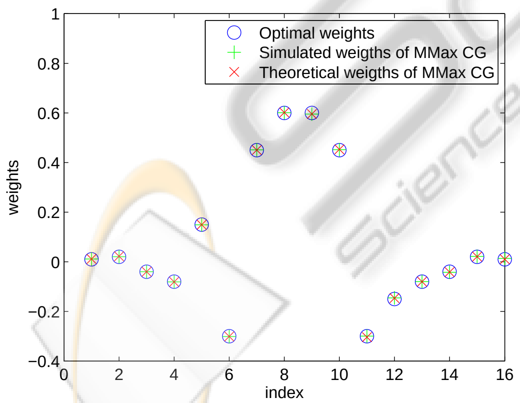

Figure 4: The mean convergence of the weights at steady state for MMax CG.

the theoretical results match the simulated results. The weights of MMax CG are close to the optimal weights.

Table 1 shows the simulated MSE and theoretical MSE of PU CG algorithms at steady state for white input. The theoretical results are calculated from (44). The partial-update lengths are $M = 12$, $M = 8$, and $M = 4$. For sequential and stochastic PU CG, the algorithms do not converge when PU length is 4. When PU length reduces, the MSE of MMax CG increases slowly, while the MSE of sequential and stochastic increase rapidly.

Table 1: The simulated MSE and theoretical MSE of PU CG algorithms.

Algorithms	Simulated MSE (dB)	Theoretical MSE (dB)
MMax CG (M=12)	-37.3869	-37.6732
MMax CG (M=8)	-37.1362	-36.7467
MMax CG (M=4)	-35.4003	-35.7671
Sequential CG (M=12)	-36.2177	-36.3857
Sequential CG (M=8)	-30.3635	-30.2404
Sequential CG (M=4)	–	–
Stochastic CG (M=12)	-36.3456	-35.9052
Stochastic CG (M=8)	-30.8199	-30.1339
Stochastic CG (M=4)	–	–

4.2 Performance Comparison of PU CG with PU RLS and PU EDS

The performance of PU CG is also compared with PU RLS and PU EDS. The comparison uses the MMax method because the MMax method has fast convergence rate and low MSE. Figure 5 shows the MSE results among CG, MMax CG, RLS, MMax RLS, EDS, and MMax EDS. The same system identification model is used. The full-update length is 16 and the partial-update length is 8. Although the full-update CG algorithm has a lower convergence rate than the full-update RLS, the MMax CG has the same convergence rate as the MMax RLS algorithm. Both MMax CG and MMax RLS can achieve the similar MSE as the full-update CG and RLS at steady state. If we use SORTLINE sorting method for both MMax CG and MMax RLS, the total number of multiplications of MMax CG and RLS are $2N^2 + M^2 + 9N + M + 5 + 2log_2N$ and $2N^2 + 2NM + 3N + M + 3 + 2log_2N$, respectively. In this case, $N = 16$ and $M = 8$. Therefore, the MMax CG needs 741 multiplications and the MMax RLS needs 833 multiplications per sample. The MMax CG needs less number of multiplications than the MMax RLS to achieve the same steady-state MSE. The full-update EDS has similar convergence rate and steady-state MSE as the full-update CG. However, the MMax EDS does not perform as well as the MMax CG. It has much higher steady-state MSE than the MMax CG.

Channel equalization performance is also examined among PU CG, PU RLS, and PU EDS algo-

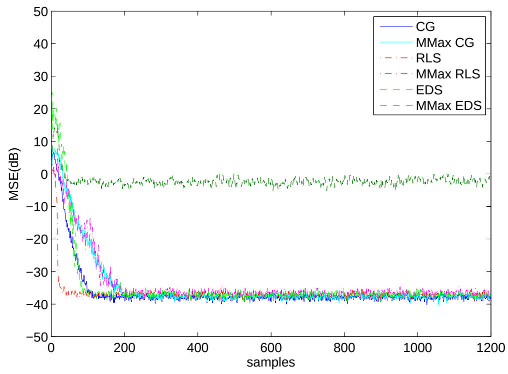

Figure 5: Comparison of MSE of PU CG with PU RLS and PU EDS.

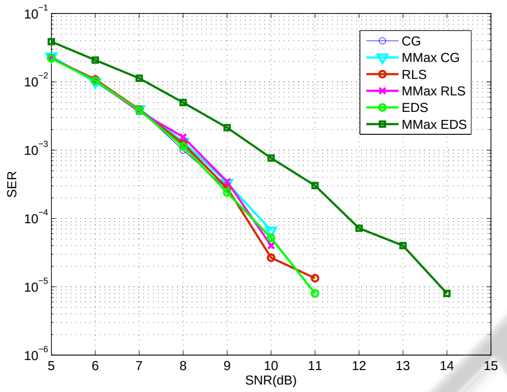

Figure 6: Comparison of SER of PU CG with PU RLS and PU EDS.

rithms. We use a simple short FIR channel (Sayed, 2003)

$$C(z) = 0.5 + 1.2z^{-1} + 1.5z^{-2} - z^{-3}. \quad (46)$$

We assume the full length of the equalizer is 30 and the PU length is 15. The input sequence is 4-QAM. The results are obtained by averaging 50 independent runs. Figure 6 illustrates the symbol-error-rate (SER) in log-scale among CG, MMax CG, RLS, MMax RLS, EDS, and MMax EDS algorithms. The SER performance of these algorithms are still related to the MSE performance shown in Figure 5. The full-update CG, RLS, and EDS have similar SER performance. The MMax CG and MMax RLS have similar performance, and their performance are also close to the full-update algorithms. The MMax CG needs 2325 multiplications while the MMax RLS needs 2819 multiplications per symbol. We can see that the MMax CG can achieve similar SER performance as the MMax RLS, with lower computational cost. The MMax EDS does not perform as well as the other algorithms.

5 CONCLUSIONS

In this paper, different PU CG algorithms are developed. Theoretical mean and mean-square performance are derived for white input. The performance of different PU CG algorithms are compared by using computer simulations. The theoretical results match the simulated results. The performance of PU CG is also compared with PU RLS and PU EDS. We can conclude that the MMax CG algorithm can achieve comparable performance as the full-update CG while having lower computational complexity than the full-update CG. In the future, the performance will be further analyzed for correlated input. Other performance such as stability and tracking performance, will also be analyzed.

ACKNOWLEDGEMENTS

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

REFERENCES

Bose, T. (2004). *Digital Signal and Image Processing*. Wiley, New York.

Chang, P. S. and Willson, A. N. (2000). Analysis of conjugate gradient algorithms for adaptive filtering. *IEEE Trans. Signal Processing*, 48(2):409–418.

Doğançay, K. (2008). *Partial-Update Adaptive Signal Processing*. Academic Press.

Douglas, S. C. (1995). Analysis and implementation of the max-nlms adaptive filter. In *Proc. Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, pages 659–663.

Douglas, S. C. (1997). Adaptive filters employing partial updates. *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, 44(3):209–216.

Godavarti, M. and Hero III, A. O. (2005). Partial update lms algorithms. *IEEE Trans. Signal Processing*, 53(7):2382–2399.

Khong, A. W. H. and Naylor, P. A. (2007). Selective-tap adaptive filtering with performance analysis for identification of time-varying systems. *IEEE Trans. Audio, Speech, and Language Processing*, 15(5):1681–1695.

Mayyas, K. (2005). Performance analysis of the deficient length lms adaptive algorithm. *IEEE Trans. Signal Processing*, 53(8):2727–2734.

Naylor, P. A. and Khong, A. W. H. (2004). Affine projection and recursive least squares adaptive filters employing partial updates. In *Proc. Thirty-Eighth Asilomar Conference on Signal, System & Computers*, pages 950–954.

Sayed, A. H. (2003). *Fundamentals of Adaptive Filtering*. Wiley, New York.

Wu, J. and Doroslovacki, M. (2007). A mean convergence analysis for partial update nlms algorithms. In *Proc. CISS*, pages 31–34.

Xie, B. and Bose, T. (2010). Partial update EDS algorithms for adaptive filtering. In *Proc. ICASSP*.

Zhang, Z. K., Bose, T., Xiao, L., and Thamvichai, R. (2006). Performance analysis of the deficient length EDS adaptive algorithms. In *Proc. APCCAS*, pages 222–226.