Conjugate Gradient Algorithm Design with RLS Normal Equation

Zhao Shengkui
School of Computer Engineering
Nanyang Technological University
zhao0024@ntu.edu.sg

Man Zhihong
School of Computer Engineering
Nanyang Technological University
aszhman@ntu.edu.sg

Khoo Suiyang
School of Computer Engineering
Nanyang Technological University
khoo0032@ntu.edu.sg

***Abstract*—In this paper, the conjugate gradient (CG) algorithm is modified using the RLS normal equation and new data windowing scheme. It is known that CG algorithm has fast convergence rate and numerical stability. However, the existing CG algorithms still suffer from either slow convergence or high misadjustment compared with the RLS algorithm. In this paper, the parameter β for CG algorithm is redesigned from the RLS normal equation and a general data windowing scheme reusing the data inputs is presented to solve these problems. The optimal property of parameter α is also analyzed using the control Lyapunov function (CLF) of the square deviation of weight error vector. The superior performance of the proposed algorithms over the RLS algorithm and the other existing CG algorithms is demonstrated by computer simulations.**

***Keywords*—Adaptive equalizer, conjugate gradient method, control Liapunov function (CLF), system identification.**

I. INTRODUCTION

Adaptive filtering algorithms have found many applications in adaptive equalization, system identification, linear predication, and noise cancellation etc [1]. For the various applications, the adaptive filtering algorithms are chosen based on the convergence rate, misadjustment, computational complexity and numerical stability. The least-mean square (LMS) adaptive algorithm [2] based on stochastic steepest descent (SD) gradient method is simply implemented and numerically stable, but with slow convergence rate. Another fast adaptive algorithm with independent of the input conditions is recursive least square (RLS) adaptive algorithm [1]. However, it suffers from the problems of numerical stability and highly computational complexity for the requirement of matrix inverse. To overcome these limitations, many fast RLS algorithms have been proposed in the literature. However, the numerical stability problem still tends to occur in these fast RLS algorithms [3]. In recent years, many adaptive algorithms based on the conjugate gradient (CG) method have been reported for adaptive filtering [4] [5] [6] [7]. It is well known that the CG method can solve the equation $\mathbf{Rw} = \mathbf{b}$ efficiently using the orthogonality of the residue directions for symmetric, positive-definite matrix $\mathbf{R}$ [9]. For the adaptive filtering problem, the autocorrelation matrix and cross-correlation vector are estimated in some manner. The advantage of the CG method is that fast convergence rate and numerical stability can be achieved. However, the existing CG algorithms still suffer from either slow convergence rate or high misadjustment compared with the RLS algorithm. For example, the Boray-Srinath method [4] can converge fast, but introducing a trade-off between misadjustment and window size. To achieve comparable misadjustment with the RLS algorithm, a very large window size is required, thereby increasing computational complexity significantly. In the Chang-Willson method [6], the scheme of finite sliding data window introduces the same problem as the Boray-Srinath method, and the scheme of exponentially decaying data window that has reduced the computational cost has slow convergence rate. The CG-CLF method [7] also has slow convergence property.

To overcome above problems, we redesign the parameter β based on the RLS normal equation and introduce a general data window for the CG algorithm. The new design of parameter β can be considered as a new approach which minimizes the norm of the difference between the CG update direction and the RLS update direction. The new data window is considered as a combination of the finite sliding data window and the exponentially decaying data window by data reusing manner. We also show the optimal property of the parameter α by the CLF method [7]. The MSE property is also described.

This paper is organized as follows. In section II, we will introduce the CG method briefly. The new data windowing scheme will be given. Section III derives the parameter β based on the RLS normal equation. Two new modified CG algorithms will be described. The optimal property of the parameter α is then analyzed using the CLF method [10] in section IV. Computer simulations are presented to show the superior performance of the proposed CG algorithms in Section V. Section VI draws conclusions for this paper.

II. CG METHOD AND DATA WINDOWNING SCHEMES

Considering the minimization problem for the following quadratic performance function

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

where $\mathbf{R}$ is $N \times N$ square matrix and positive definite, $\mathbf{b}$ and $\mathbf{w}$ are vectors with dimension N. To solve for the vector $\mathbf{w}$ that minimizes the quadratic performance function in (1), it can be simplified to the following linear equation after taking derivate of (1) with respect to $\mathbf{w}$ and setting to zero

$$\mathbf{Rw} = \mathbf{b}. \tag{2}$$

The direct way to solve (2) is to compute the matrix inverse $\mathbf{R}^{-1}$. However, the inverting $\mathbf{R}$ requires the

computational complexity of $O(M^3)$. The adaptive algorithms can be viewed as an alternative way to iteratively solve for the optimal solution $\mathbf{R}^{-1}\mathbf{b}$. Compared with other algorithms, the algorithms based on CG method have the advantages of no computation of matrix inverse required and fast convergence. Even if $\mathbf{R}$ is not full rank, the CG algorithms can still be employed to search for the best solution. While the RLS method and the Newton method may diverge during computing $\mathbf{R}^{-1}$. We give the basic CG algorithm described in [8] as follows:

$$
\begin{aligned}
&\text{Initialize } \mathbf{w}(0)=0,\ \mathbf{g}(0)=\mathbf{b}-\mathbf{R}\mathbf{w}(0),\ \mathbf{p}(1)=\mathbf{g}(0)\\
&\quad \text{For } k=1,\ 2,\cdots, \text{ until convergence}\\
&\qquad \alpha(k)=\frac{\|\mathbf{g}(k-1)\|^2}{\mathbf{p}(k)^T\mathbf{R}\mathbf{p}(k)}\\
&\qquad \mathbf{w}(k)=\mathbf{w}(k-1)+\alpha(k)\mathbf{p}(k)\\
&\qquad \mathbf{g}(k)=\mathbf{g}(k-1)-\alpha(k)\mathbf{R}\mathbf{p}(k)\\
&\qquad \beta(k)=\frac{\|\mathbf{g}(k)\|^2}{\|\mathbf{g}(k-1)\|^2}\\
&\qquad \mathbf{p}(k+1)=\mathbf{g}(k)+\beta(k)\mathbf{p}(k)\\
&\quad \text{End}
\end{aligned}
\tag{3}
$$

where the parameter $\alpha(k)$ is chosen to minimize the performance function $J(\mathbf{w}(k-1)+\alpha(k)\mathbf{p}(k))$ along the update direction and $\beta(k)$ is chosen to satisfy the current direction vector $\mathbf{p}(k)$ is linearly independent and $\mathbf{R}$-conjugate to all the previous direction vectors $\mathbf{p}(1)$, $\mathbf{p}(2)$, $\cdots$, $\mathbf{p}(k-1)$. Therefore, there is no redundant update to the vector $\mathbf{w}(k)$ in each update direction. The vector $\mathbf{w}(k)$ finally converges to the optimal solution $\mathbf{R}^{-1}\mathbf{b}$ in finite number of iterations [9].

When using the CG algorithm for adaptive filtering problem, the correlation matrix $\mathbf{R}$ and cross-correlation vector $\mathbf{b}$ are often not known and need to be estimated. There are two approaches reported in [6]. One is called shift finite sliding data window given below:

$$\mathbf{R}(k)=\frac{1}{M}\sum_{j=k-M+1}^{k}\mathbf{x}(j)\mathbf{x}(j)^T \tag{4}$$

$$\mathbf{b}(k)=\frac{1}{M}\sum_{j=k-M+1}^{k}d(j)\mathbf{x}(j) \tag{5}$$

The advantage of finite sliding data windowing method is that it allows the CG algorithms to process in block. Therefore, the convergence rate is fast and independent of the eigenvalue spread of $\mathbf{R}$, whereas the disadvantage is that, the misadjustment is dependent on the window size M. Usually a large window size $M > N$ and N iterations per data update is required to achieve comparable misadjustment with the RLS. The result is that it will cost more storage and computation time [5] [6].

Another approach is called exponentially decaying data window which has the same form as used in the RLS algorithm given as follows:

$$\mathbf{R}(k)=\lambda\mathbf{R}(k-1)+\mathbf{x}(k)\mathbf{x}(k)^T \tag{6}$$

$$\mathbf{b}(k)=\lambda\mathbf{b}(k-1)+d(k)\mathbf{x}(k) \tag{7}$$

where λ is the forgetting factor. For sample-by-sample processing, the gradient vector can be computed as

$$
\begin{aligned}
\mathbf{g}(k)&=\mathbf{b}(k)-\mathbf{R}(k)\mathbf{w}(k)\\
&=\lambda\mathbf{g}(k-1)-\alpha(k)\mathbf{R}(k)\mathbf{p}(k)+e(k)\mathbf{x}(k)
\end{aligned}
\tag{8}
$$

where $e(k)$ is priori output error for time index k. Since no past data are needed in the current update, this approach is much more efficient and less memory required. Moreover, the sample-by-sample processing approach reduces the computational cost using only one iteration for each coefficient and data update. However, in this approach, the update conjugate direction $\mathbf{p}(k)$ loses $\mathbf{R}$-orthogonality and the convergence rate becomes slow.

Based on the two data windowing schemes, we introduce a general data windowing scheme which combines both forms of the finite sliding data window and exponentially decaying data window. This new scheme can be considered as a data reusing manner for the estimation of correlation function and cross-correlation function given as follows:

$$\mathbf{R}(k)=\lambda\mathbf{R}(k-1)+\frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\mathbf{x}(k-j)^T \tag{9}$$

$$\mathbf{b}(k)=\lambda\mathbf{b}(k-1)+\frac{1}{M}\sum_{j=0}^{M-1}d(k-j)\mathbf{x}(k-j) \tag{10}$$

where λ is again the forgetting factor and M is the finite length of data samples. From the general form (9) and (10), we can see that it can be modified to finite sliding data window by setting $\lambda=0$ and exponentially decaying data window by setting $M=1$. Therefore, it is very flexible for the user to set the forgetting factor and number of reusing data M according to the input data characteristics and convergence requirement. It will be shown in the simulation results that using the general data window (9) and (10) for the CG algorithms, the superior performance over the RLS and other existing CG algorithms can be achieved for both sample-by-sample processing and block processing with relatively small value of M.

III. CG Algorithms for Adaptive Filtering

Considering the following RLS normal equation in sample-by-sample update case:

$$\mathbf{R}(k-1)\mathbf{w}(k)=\mathbf{b}(k-1) \tag{11}$$

where $\mathbf{w}(k)$ is the estimate of the solution which minimizes the performance function (1) when the estimates of the autocorrelation $\mathbf{R}(k-1)$ and cross-correlation $\mathbf{b}(k-1)$ are available. We have known that the solution of this normal equation for the $k-1$ data inputs is the least square estimate. For the CG algorithm, the step size $\alpha(k)$ can be easily derived from the normal equation (11). Substituting the weight update equation given in (3) into (11), after rearrangement we have

$$\mathbf{b}(k-1)=\mathbf{R}(k-1)\big(\mathbf{w}(k-1)+\alpha(k)\mathbf{p}(k)\big) \tag{12}$$

Using the gradient vector $\mathbf{g}(k-1)=\mathbf{b}(k-1)-\mathbf{R}(k-1)\mathbf{w}(k-1)$ for (12), we yield:

$$\mathbf{p}(k) = \left(\alpha(k)\mathbf{R}(k-1)\right)^{-1}\mathbf{g}(k-1). \quad (13)$$

Equation (13) shows that the optimal vector $\mathbf{p}(k)$ is actually the transformed gradient vector $\mathbf{g}(k-1)$ by the matrix $\left(\alpha(k)\mathbf{R}(k-1)\right)^{-1}$. Premultiplying (13) by $\mathbf{p}(k)^T\mathbf{R}(k-1)^{-1}$ and through simple manipulations, the step size is obtained

$$\alpha(k) = \frac{\mathbf{p}(k)^T\mathbf{g}(k-1)}{\mathbf{p}(k)^T\mathbf{R}(k-1)\mathbf{p}(k)}. \quad (14)$$

Notice that the result of the step size (14) has the same form as given in [6]. Therefore, it minimizes the following quadratic performance function along the searching direction $\mathbf{p}(k)$

$$J\left(\mathbf{w}(k)\right) = \frac{1}{2}\mathbf{w}(k)^T\mathbf{R}(k-1)\mathbf{w}(k) - \mathbf{b}(k-1)^T\mathbf{w}(k). \quad (15)$$

It can be shown that with the knowledge of the direction $\mathbf{p}(k)$ (13), we can reach the least square solution in one step. However, it may not be possible to get this direction without the matrix inverse. What the CG algorithm does is to divide this optimal direction into several orthogonal directions. When sample-by-sample update is performed, the orthogonality property is lost between the update directions. In that case, the direction vector needs to be reset to the true gradient periodically for the convergence of the algorithm, leading to slow convergence. A non-reset method called Polak-Ribiere method can be found in [8]. We now present a new non-reset method. Considering the update direction $\mathbf{p}(k+1)$ that minimizes the norm $\left\|\mathbf{p}(k+1) - \left(\alpha(k)\mathbf{R}(k)\right)^{-1}\mathbf{g}(k)\right\|$. We design the parameter $\beta(k)$ which is given by

$$\beta(k) = \arg\min_{\beta(k)\in\mathbb{R}}\left\|\mathbf{p}(k+1) - \left(\alpha(k)\mathbf{R}(k)\right)^{-1}\mathbf{g}(k)\right\|, \quad (16)$$

where the direction $\mathbf{p}(k+1)$ is computed as:

$$\mathbf{p}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{p}(k). \quad (17)$$

To solve for $\beta(k)$, substituting (17) into (16) and setting the derivative of the norm with respect to $\beta(k)$ to zero, we obtain

$$\beta(k) = \frac{\mathbf{p}(k)^T\mathbf{g}(k) - \alpha(k)\mathbf{g}(k)^T\mathbf{R}(k)\mathbf{p}(k)}{\alpha(k)\mathbf{p}(k)^T\mathbf{R}(k)\mathbf{p}(k)}. \quad (18)$$

The parameter (18) can be considered as a non-reset method since the best approximation of the true direction is always ensured. Now we summarize the new modified sample-by-sample processing CG algorithm (referenced as CG1) with the general data windowing scheme (9) and (10) as follows:

set $\mathbf{w}(0) = 0,\ \mathbf{g}(0) = \mathbf{b}(0) - \mathbf{R}(0)\mathbf{w}(0),\ \mathbf{p}(1) = \mathbf{g}(0)$

$$\alpha(k) = \eta\frac{\mathbf{p}(k)^T\mathbf{g}(k-1)}{\mathbf{p}(k)^T\mathbf{R}(k-1)\mathbf{p}(k)}.$$
$$\mathbf{w}(k) = \mathbf{w}(k-1) + \alpha(k)\mathbf{p}(k)$$
$$\mathbf{g}(k) = \lambda\mathbf{g}(k-1) - \alpha(k)\mathbf{R}(k)\mathbf{p}(k) + \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\varepsilon(k-j)$$
$$\mathbf{R}(k) = \lambda\mathbf{R}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\mathbf{x}(k-j)^T$$
$$\beta(k) = \frac{\mathbf{p}(k)^T\mathbf{g}(k) - \alpha(k)\mathbf{g}(k)^T\mathbf{R}(k)\mathbf{p}(k)}{\alpha(k)\mathbf{p}(k)^T\mathbf{R}(k)\mathbf{p}(k)}$$
$$\mathbf{p}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{p}(k) \quad (19)$$

where η is a small constant with $\lambda - 0.5 \le \eta \le \lambda$ to ensure the convergence as stated in [6], and the error is computed as $\varepsilon(k-j) = d(k-j) - \mathbf{x}^T(k-j)\mathbf{w}(k-1)$. In the CG1, the finite $M-1$ past data samples are required for the estimation of $\mathbf{R}$ and computation of errors $\varepsilon(k-j)$. The convergence rate of CG1 depends on the eigenvalue spread of $\mathbf{R}$ weakly and the dependence can be reduced by increasing the reusing data size M.

The block processing allows the CG algorithms to converge faster than the CG algorithms with sample-by-sample processing. That is because the CG algorithm with block processing can run several iterations per data update. However, the block processing CG algorithms described in [4] and [6] has the output MSE dependent on the data window size. Usually a large length of data window must be used to achieve a misadjustment comparable to the RLS algorithm. The general data windowing scheme allows the CG algorithm to achieve as low misadjustment as the RLS algorithm independent of the data window size. In addition, the fast convergence is also ensured with independence of the eigenvalue spread $\mathbf{R}$. Following the same approach used in [4] and [6], the new modified block processing CG algorithm (referenced as CG2) can be described as follows:

Set initial condition: $\mathbf{w}(0) = 0$.

For each time instant k, compute:

Start:

$$\mathbf{R}(k) = \lambda\mathbf{R}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1}\mathbf{x}(k-j)\mathbf{x}(k-j)^T$$
$$\mathbf{b}(k) = \lambda\mathbf{b}(k-1) + \frac{1}{M}\sum_{j=0}^{M-1}d(k-j)\mathbf{x}(k-j) \quad (20)$$
$$\mathbf{g}(0) = \mathbf{b}(k) - \mathbf{R}(k)\mathbf{w}(k-1),\ \mathbf{p}(1) = \mathbf{g}(0)$$

for $n = 1$ to $\min(k,\ N,\ M)$ do:

$$\alpha(n) = \frac{\mathbf{p}(n)^T\mathbf{g}(n-1)}{\mathbf{p}(n)^T\mathbf{R}(k)\mathbf{p}(n)}.$$
$$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha(n)\mathbf{p}(n)$$
$$\mathbf{g}(n) = \mathbf{g}(n-1) - \alpha(n)\mathbf{R}(k)\mathbf{p}(n)$$
$$\beta(n) = -\frac{\mathbf{g}(n)^T\mathbf{R}(k)\mathbf{p}(n)}{\mathbf{p}(n)^T\mathbf{R}(k)\mathbf{p}(n)}$$
$$\mathbf{p}(n+1) = \mathbf{g}(n) + \beta(n)\mathbf{p}(n)$$

after $\min(k,\ N,\ M)$ iterations, do:

$$\mathbf{w}(k) = \mathbf{w}(n),\ k = k+1$$

goto Start.

where $\mathbf{R}(k)$, $\mathbf{b}(k)$ and $\mathbf{x}(k)$ are assumed to be zero for $k < 0$. Notice that the parameter β simplified in the block processing

CG2 algorithm since $\mathbf{p}(n)^T \mathbf{g}(n) = 0$ has the same form as designed by the control Liapunov function (CLF) method [7]. At the beginning of the process, the number of data samples k is less than $\min(N, M)$, so only k iterations are necessary to run.

IV. Analysis of the CG Algorithms

The optimality property of the step size $\alpha(k)$ can be easily verified using the square deviation of weight error vector as the CLF

$$V_\alpha(k) = \|\mathbf{w}_o - \mathbf{w}(k)\|^2, \tag{21}$$

where $\mathbf{w}_o$ is the true weight vector to be estimated by adaptive filter. The difference is given by

$$\begin{aligned}\Delta V_\alpha(k) &= V_\alpha(k) - V_\alpha(k-1) \\ &= \|\mathbf{w}_o - \mathbf{w}(k)\|^2 - \|\mathbf{w}_o - \mathbf{w}(k-1)\|^2.\end{aligned} \tag{22}$$

Substituting the weight update equation given in (3) into (22), the difference can be rewritten as

$$\Delta V_\alpha(k) = -2\alpha(k)\left(\mathbf{w}_o - \mathbf{w}(k-1)\right)^T \mathbf{p}(k) + \alpha(k)^2 \|\mathbf{p}(k)\|^2. \tag{23}$$

The optimal value of $\alpha(k)$ that makes $V_\alpha(k)$ as negative as possible is determined by calculating the derivative of (23) with respect to $\alpha(k)$ and setting it to zero:

$$\frac{\partial \Delta V_\alpha(k)}{\partial \alpha(k)} = -2\left(\mathbf{w}_o - \mathbf{w}(k-1)\right)^T \mathbf{p}(k) + 2\alpha(k)\|\mathbf{p}(k)\|^2 = 0 \tag{24}$$

From (24), the optimal step size $\alpha(k)$ is then computed as

$$\alpha(k) = \frac{\mathbf{p}(k)^T \left(\mathbf{w}_o - \mathbf{w}(k-1)\right)}{\|\mathbf{p}(k)\|^2}. \tag{25}$$

Inserting the value of $\alpha(k)$ in (25) into (23), we have

$$\Delta V_\alpha(k) = -\frac{\left[\mathbf{p}(k)^T \left(\mathbf{w}_o - \mathbf{w}(k-1)\right)\right]^2}{\|\mathbf{p}(k)\|^2} < 0. \tag{26}$$

That means the square deviation of the weight error vector will finally converge to zero according to the CLF method [10]. In addition, a bound can be found for the deviation of the weight error vector [11].

To show the step size $\alpha(k)$ in (21) approaches to the optimal one in (25), we rewrite (14) as

$$\alpha(k) = \frac{\mathbf{p}(k)^T \left(\mathbf{b}(k-1) - \mathbf{R}(k-1)\mathbf{w}(k-1)\right)}{\mathbf{p}(k)^T \mathbf{R}(k-1)\mathbf{p}(k)}. \tag{27}$$

Assuming that there is no measurement noise, we have

$$\mathbf{b}(k-1) = \mathbf{R}(k-1)\mathbf{w}_o. \tag{28}$$

Substituting (28) into (27) yields

$$\alpha(k) = \frac{\mathbf{p}(k)^T \mathbf{R}(k-1)\left(\mathbf{w}_o - \mathbf{w}(k-1)\right)}{\mathbf{p}(k)^T \mathbf{R}(k-1)\mathbf{p}(k)}. \tag{29}$$

For sample-by-sample processing method, we assume that data input is stationary and the fluctuations in $\mathbf{R}(k-1)$ are small enough from one iteration to the next iteration. Therefore, we can justify approximating $\alpha(k)$ in (29) as

$$\alpha(k) \approx \frac{\mathbf{p}(k)^T \left(\mathbf{w}_o - \mathbf{w}(k-1)\right)}{\|\mathbf{p}(k)\|^2}. \tag{30}$$

It is observed that for the larger time index k is, the step size (14) can be very close to the optimal one (25). The step size in (14) will eventually approach to the optimal step size when the algorithm converges.

For the MSE of the proposed CG algorithms, the correlation and cross-correlation functions eventually approaches to the true values after enough data inputs are collected using the general data window. At that point and onwards, the estimated correlation and cross-correlation functions are varying very slowly. The CG algorithms can easily track the slow changes and a low misadjustment comparable to that of the RLS algorithm can then be achieved.

V. Simulation Results

In this section, we implemented several simulations using the two basic configurations: adaptive equalization and system identification [1]. The performance of the modified CG1 and CG2 algorithms are compared to the CG-CLF algorithm [7], the Chang-Willson method [6] and the RLS method [6] [1]. All the simulation results were obtained by ensemble averaging 200 independent Monte-Carlo simulation runs.

A. Adptive Equalizer

The example used in this section is as described in [1]. The input signal applied to the channel is a random Bernoulli sequence $r(k) = \pm 1$, which has zero-mean and unit variance. The channel is corrupted by a zero mean white Gaussian noise of variance σ_v^2 equal to 0.001. The equalizer with 11 taps was employed. The simulations were performed with the parameter W set to 2.9 and 3.5, which implied that the eigenvalue spread equal to 6.0782 and 46.8216, respectively. The parameters were chosen as $\lambda = \eta = 0.99$. Figure 1 shows the ensemble average MSE behaviors of the proposed CG1, the CG-CLF algorithm, the Chang-Willson method (CG1) and the RLS method. The number of reusing data was chosen such that $M = 11$ for the proposed CG1. In this case, it can be observed that the proposed CG1 algorithm has the transient property better than the RLS method, and significantly improved over the CG-CLF algorithm and the Chang-Willson method (CG1). All the compared CG algorithms give MSE comparable with the RLS algorithm in steady state with 1.7×10^{-3} for $W = 2.9$ and 1.4×10^{-2} for $W = 3.5$. Figure 2 shows the effects of the number of reusing data on the learning curves of the proposed CG1 algorithm. We observed that when no past data is reused, the proposed CG1 uses an equivalent exponential decaying data window. In this case, its performance is comparable with the other CG algorithms. When the number of reusing data M increases, the convergence rate of the proposed CG1 becomes faster. Therefore, it provides the user with flexibility in the choice of the number of reusing data by considering convergence and the added computations. As stated in [4], chosen M in the range

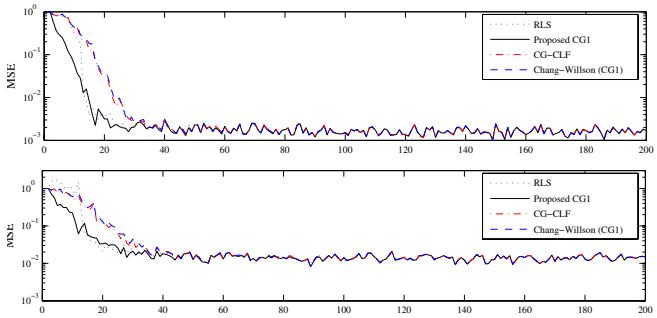

Figure 1. Ensemble-average MSE behaviors of various algorithms for adaptive transversal equalizer. (Top) $W = 2.9$. (Bottom) $W = 3.5$.

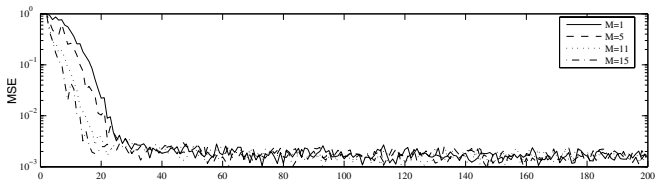

Figure 2. Learning curves of the CG1 algorithm for adaptive transversal equalizer with $W = 2.9$ and $M = 1,\ 5,\ 11,\ 15$.

$1 < M < \sqrt{N}$ seems to provide reasonable convergence with an affordable computational complexity.

B. System Identifcation

An unknown time-invariant plant is to be identified, which has a finite time impulse response of order 20 and the FIR adaptive filters with the same length are designed. The correlated input $x(k)$ is generated by the following random walk model [1]:

$$x(k) = 0.98x(k-1) + f(k)$$

where $f(k)$ is independent Gaussian sequences with zero mean and unity variance. The measurement noise is zero-mean, white Gaussian sequences of variance σ_k^2 equal to 0.01. In this case, the block processing of the CG algorithms were implemented and the data window size was chosen as $M = 10$ for both the proposed CG2 and Chang-Willson CG2. Figure 3 shows the ensemble average MSE behaviors of the compared algorithms. It is observed that the proposed CG2 algorithm achieves the convergence which is faster than the RLS algorithm and Chang-Willson CG2. The proposed CG2 gives misadjustment comparable with the RLS algorithm independent of the window size. The Chang-Willson CG2 has fast convergence rate but with high misadjustment.

VI. CONCLUSIONS

In this paper, we have modified the CG algorithms for adaptive filtering problem using the RLS normal equation and general data windowing scheme. The parameter β has been redesigned from the RLS normal equation. The general data window has been presented which allows the designed CG algorithms to reuse the data inputs. In addition, this general data window can also be extended to other CG algorithms for fast convergence and low misadjustment. The advantage of the proposed CG algorithms can be observed from the simulations that fast convergence property and low misadjustment have been provided.

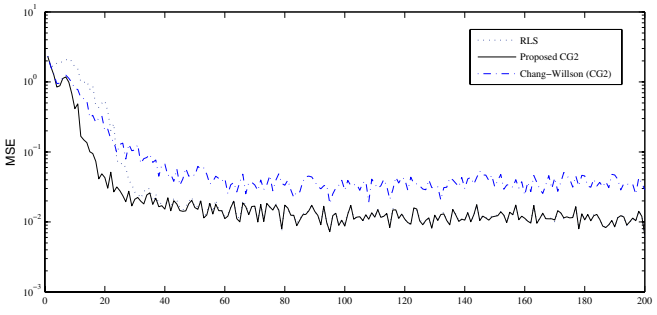

Figure 3. Ensemble-average MSE behaviors of various algorithms for the system identification with SNR=20dB.

REFERENCES

[1] S. Haykin. *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, Nj: Prentice Hall, 2002.

[2] B. Widrow and J. M. McCool, "Stationary and nonstationary leaming characteristics of the LMS filter," *Proc. IEEE,* vol. 64, pp. 1151-1162, Aug. 1976.

[3] S. Haykin. *Adaptive Filter Theory*. Englewood Cliffs. NJ: Prentice Hall. 1986.

[4] G.K Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Trans. Circuits Syst. I*, vol. 39, pp. 1-10, Jan. 1992.

[5] T. Bose and M. Q. Chen, "Conjugate gradient method in adaptive bilinear filtering," *IEEE Trans. Signal Process*., vol. 43, pp. 1503-1508, Jan. 1995.

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

[7] O. Diene and A. Bhaya, "Adaptive filtering algorithms designed using control Liapunov functions," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp.224-227, Apr. 2006.

[8] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.

[9] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Boston, MA: PWS-Kent, 1996.

[10] A. Bhaya and E. Kaszkurewicz, *Control Perspectives on Numerical Algorithms and Matrix Problems*. Philadelphia, PA: SIAM, 2006.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.