Tracking Performance of the MMax Conjugate Gradient Algorithm

Bei Xie and Tamal Bose

Wireless@VT
Bradley Dept. of Electrical and Computer Engineering
Virginia Tech

Outline

- Motivation
- Background
  - Conjugate Gradient (CG) Algorithm
  - Partial Update Methods
- Partial Update CG Algorithm
  - Tracking Performance Analysis
  - Simulations
- Summary

Motivation

- How to reduce the computational complexity of an adaptive filter?

  Solutions: Using partial update (PU) methods.
- What are the tracking performance by using partial update methods to CG?

Conjugate Gradient Algorithm

Solve system with form $\mathbf{Rw} = \mathbf{b}$

Equivalent to find a **w** to minimize the cost function

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

The gradient of the cost function is:

$$\nabla_{\mathbf{w}} J(\mathbf{w}(n)) = \mathbf{R}\mathbf{w}(n) - \mathbf{b}$$

The residual vector is defined as:

$$\mathbf{g}(n) = -\nabla_{\mathbf{w}} J(\mathbf{w}(n)) = \mathbf{b} - \mathbf{R}\mathbf{w}(n)$$

A line search method for minimizing the cost function has the form:

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\alpha\mathbf{p}(n)$$

p(n) is the direction vector

CG chooses the direction is conjugately orthogonal to the previous directions

$$\mathbf{p}^T(n)\mathbf{R}\mathbf{p}(m)=0, n>m$$

Now we find α to minimize $J\left(\mathbf{w}(n-1)+\alpha\mathbf{p}(n)\right)$

$$\nabla_{\alpha} J\left(\mathbf{w}(n-1)+\alpha(n)\mathbf{p}(n)\right)$$

$$=\mathbf{p}^{T}(n)\mathbf{R}\left(\mathbf{w}(n-1)+\alpha(n)\mathbf{p}(n)\right)-\mathbf{p}^{T}(n)\mathbf{b}=0$$

$$\alpha(n)=\frac{\mathbf{p}^{T}(n)\left(\mathbf{b}-\mathbf{R}\mathbf{w}(n-1)\right)}{\mathbf{p}^{T}(n)\mathbf{R}\mathbf{p}(n)}=\frac{\mathbf{p}^{T}(n)\mathbf{g}(n-1)}{\mathbf{p}^{T}(n)\mathbf{R}\mathbf{p}(n)}$$

The residual vector is also equal to

$$\mathbf{g}(n)=\mathbf{b}-\mathbf{R}\mathrm{w}(n)=\mathbf{b}-\mathbf{R}\left(\mathrm{w}(n-1)+\alpha(n)\mathbf{p}(n)\right)$$

$$=\mathbf{g}(n-1)-\alpha(n)\mathbf{R}\mathbf{p}(n)$$

The residual vector is orthogonal to the previous direction vectors,

$$\mathbf{g}^{T}(n)\mathbf{p}(m)=0, n>m$$

CG chooses the direction in the form of

$$\mathbf{p}(n+1) = \mathbf{g}(n) + \beta(n)\mathbf{p}(n)$$

Use Polak-Ribière (PR) method,

$$\beta(n) = \frac{\left(\mathbf{g}(n) - \mathbf{g}(n-1)\right)^T \mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}$$

PR method is chosen because it is a non-reset method and performs better for non-constant matrix **R**

CG with PR method usually converges faster than Fletcher-Reeves (FR) method

CG Algorithm in adaptive filter system

A basic adaptive filter system model is

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

d(n) is the desired signal

$\mathbf{x}(n)=[x(n),x(n-1),\ldots,x(n-N+1)]^T$ is the input data vector of an unknown system

$\mathbf{w}^*=[w_1^*,w_2^*,\ldots,w_N^*]^T$ is the impulse response vector of the unknown system

$\mathbf{w}^*$ is constant for time-invariant system

$\mathbf{w}^*$ changes for time-varying sytem

v(n) is a white noise

To estimate the **R** and **b** in $\mathbf{R}\mathbf{w} = \mathbf{b}$

The exponentially decaying data window is used

$$\mathbf{R}(n) = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$$

$$\mathbf{b}(n) = \sum_{i=0}^{n} \lambda^{n-i} d(i)\mathbf{x}(i) = \lambda \mathbf{b}(n-1) + d(n)\mathbf{x}(n)$$

λ is the forgetting factor

The CG algorithm in an adaptive filter system is summarized as:

Initial conditions:

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

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

$$\alpha(n)=\eta\frac{\mathbf{p}^T(n)\mathbf{g}(n-1)}{\mathbf{p}^T(n)\mathbf{R}(n)\mathbf{p}(n)},\quad \lambda-0.5\le\eta\le\lambda$$

η is used to guarantee convergence

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\alpha\mathbf{p}(n)$$

$$\mathbf{g}(n)=\mathbf{b}(n)-\mathbf{R}(n)\mathbf{w}(n)=\lambda\mathbf{g}(n-1)-\alpha(n)\mathbf{R}(n)\mathbf{p}(n)+\mathbf{x}(n)\left(d(n)-\mathbf{x}(n)^T\mathbf{w}(n-1)\right)$$

$$\beta(n)=\frac{\left(\mathbf{g}(n)-\mathbf{g}(n-1)\right)^T\mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}$$

$$\mathbf{p}(n+1)=\mathbf{g}(n)+\beta(n)\mathbf{p}(n)$$

Partial Update (PU) Methods

- Update part of the weights to save the computational complexity
- Each update step, update M<N coefficients
- Basic PU methods include periodic, sequential, stochastic, and MMax methods
  - The periodic method: update the weights at every S^{th} iteration and copy the weights at the other iterations, where $S = \left\lceil \frac{N}{M} \right\rceil$

- The sequential method: choose the subset of the weights in a round-robin fashion.
- The stochastic method: is a randomized version of the sequential method. Usually a uniformly distributed random process will be applied.
- The MMax method: the elements of the weight **w** are updated according to the position of the M largest elements of the input vector **x**(n).

Partial Update CG Algorithm

The partial update CG algorithm in an adaptive filter system is summarized as:

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

$$\alpha(n) = \eta\frac{\mathbf{p}^T(n)\mathbf{g}(n-1)}{\mathbf{p}^T(n)\hat{\mathbf{R}}(n)\mathbf{p}(n)}, \quad \lambda - 0.5 \le \eta \le \lambda$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha\mathbf{p}(n)$$

$$\mathbf{g}(n) = \lambda\mathbf{g}(n-1) - \alpha(n)\hat{\mathbf{R}}(n)\mathbf{p}(n) + \hat{\mathbf{x}}(n)\left(d(n) - \mathbf{x}(n)^T\mathbf{w}(n-1)\right)$$

$$\beta(n) = \frac{\left(\mathbf{g}(n) - \mathbf{g}(n-1)\right)^T\mathbf{g}(n)}{\mathbf{g}^T(n-1)\mathbf{g}(n-1)}$$

$$\mathbf{p}(n+1) = \mathbf{g}(n) + \beta(n)\mathbf{p}(n)$$

$$\hat{\mathbf{x}}(n) = \mathbf{I}_M(n)\mathbf{x}(n)$$

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

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

The number of multiplications of CG is $3N^2+10N+3$ per sample

The number of multiplications of PU CG is $2N^2+M^2+9N+M+3$ per sample

The MMax method: the elements of the input **x** are chosen according to the position of the M largest elements of the input vector **x**(n)

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

SORTLINE method is used for comparison
Mmax CG needs 2 + $2\log_2 N$ comparisons

Tracking Performance Analysis of PU CG

Desired signal becomes:

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

Time-varying system $\mathbf{w}^*(n)$ uses a first-order Markov model

$$\mathrm{w}^*(n) = \gamma \mathrm{w}^*(n-1) + \eta(n)$$

γ is very close to unity
$\eta(n)$ is process noise

Assumptions:

- The coefficient error **w**(n)-**w***(n) is small and independent of the input signal **x**(n) at steady state
- White noise v(n) is independent of the input signal **x**(n) and is independent of process noise η(n)
- Input signal **x**(n) is independent of both v(n) and η(n)

At steady state, the MSE of PU CG with correlated input is

$$E\left\{ \left|e(n)\right|^2 \right\} = E\left\{ \left|d(n) - \mathbf{x}^T(n)\mathbf{w}(n)\right|^2 \right\}$$

$$\approx \sigma_v^2 + tr\left(\mathbf{R}\left(\frac{1-\lambda}{1+\lambda}\sigma_v^2 \tilde{\mathbf{R}}^{-1}\hat{\mathbf{R}}\tilde{\mathbf{R}}^{-T} + \frac{\lambda^2}{1-\lambda^2}\mathbf{R}_\eta \right)\right)$$

$$\mathbf{R} = E\left\{\mathbf{x}(n)\mathbf{x}^T(n)\right\}$$

$$\tilde{\mathbf{R}} = E\left\{\hat{\mathbf{x}}(n)\mathbf{x}^T(n)\right\} \approx (1-\lambda)\tilde{\mathbf{R}}(n)$$

$$\tilde{\mathbf{R}}(n) = \lambda\tilde{\mathbf{R}}(n-1) + \hat{\mathbf{x}}(n)\mathbf{x}^T(n)$$

$$\hat{\mathbf{R}} = E\left\{\hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\right\}$$

At steady state, the MSE of PU CG with white input is

$$E\left\{ |e(n)|^2 \right\} \approx \sigma_v^2 + \frac{N(1-\lambda)}{1+\lambda}\sigma_v^2 \sigma_x^2 \sigma_{\hat{x}}^2 \sigma_{\tilde{x}}^{-4} + \frac{\lambda^2}{1-\lambda^2}\sigma_x^2 tr(\mathbf{R}_\eta)$$

$$\sigma_x^2 = tr(\mathbf{R})$$

$$\sigma_{\hat{x}}^2 = tr(\hat{\mathbf{R}})$$

$$\sigma_{\tilde{x}}^2 = tr(\tilde{\mathbf{R}})$$

$$\sigma_v^2 = E\left\{ v^2(n) \right\}$$ Variance of noise

For MMax method and white input,

$$\sigma_{\hat{x}}^2 \approx \kappa\sigma_x^2, \quad \sigma_{\tilde{x}}^2 \approx \kappa\sigma_x^2$$

κ <1, κ is close to 1

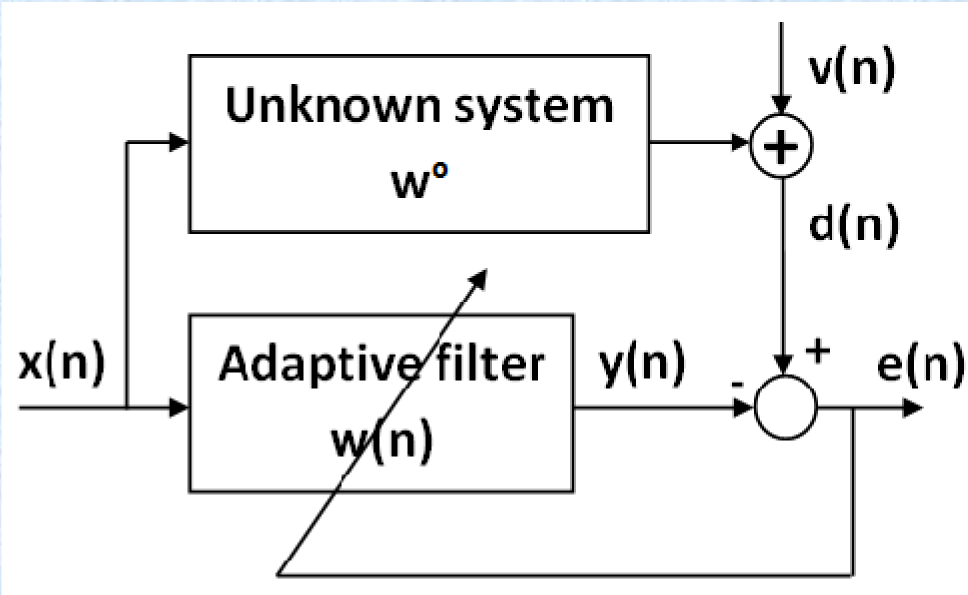
$$E\left\{ |e(n)|^2 \right\} \approx \sigma_v^2 + \frac{N(1-\lambda)}{(1+\lambda)\kappa}\sigma_v^2 + \frac{\lambda^2}{1-\lambda^2}\sigma_x^2 tr(\mathbf{R}_\eta)$$

For white process noise η(n)

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

Simulations

System identification model



The initial impulse response of unknown system is 16-order (N=16) FIR filter

$w^*(n)=[0.01,0.02,-0.04,-0.08,0.15,-0.3,0.45, 0.6, 0.6,0.45,-0.3,0.15,-0.08,-0.04,0.02,0.01]^T$

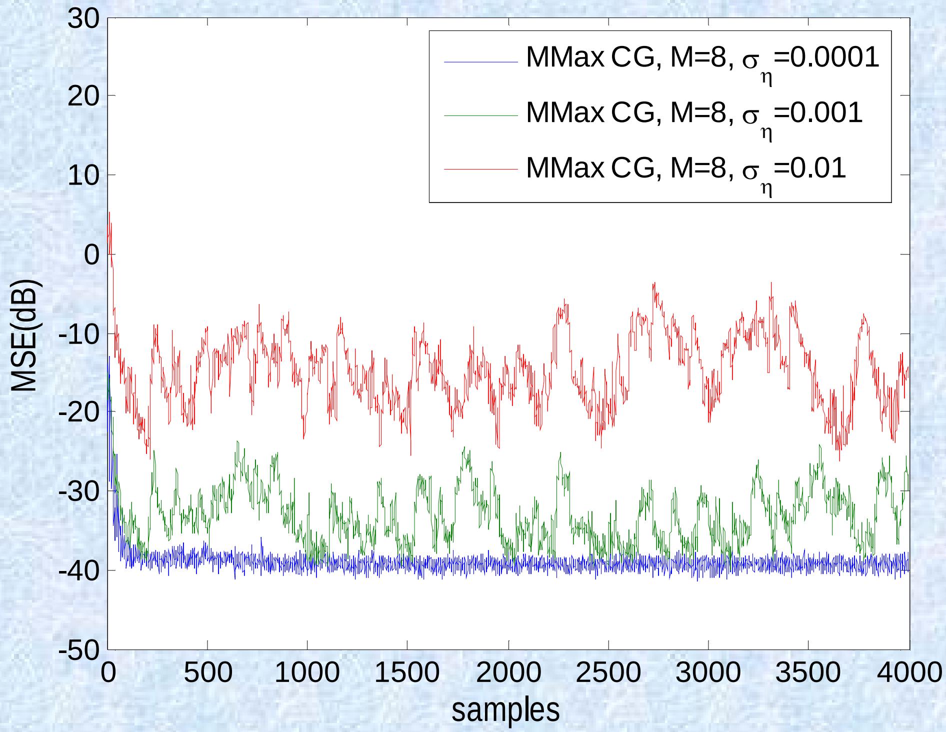
The variance of the input noise $\sigma_v^2=0.0001$

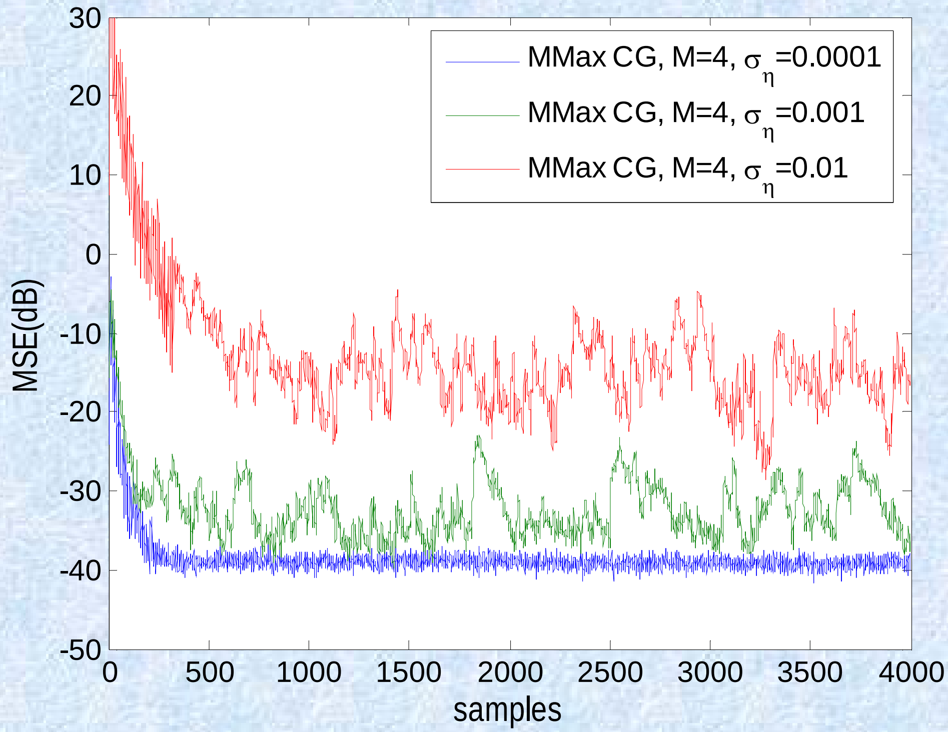
Parameter $\lambda=0.9$ and $\eta=0.6$

White input, variance is 1

γ in Markov model is 0.9998

Tracking Performance of MMax CG





Comparison of MSE of MMax CG for varying process noise η, M=8

Comparison of MSE of MMax CG for varying process noise η, M=4

VirginiaTech Invent the Future

Wireless @ Virginia Tech

- The MSE of MMax CG increases when the process noise increases.
- The variance of the MSE increases when the process noise increases.
- The partial update length does not have much effect on the MSE results.
- The partial update length only affects the convergence rate. The convergence rate decreases as the partial update length decreases.

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

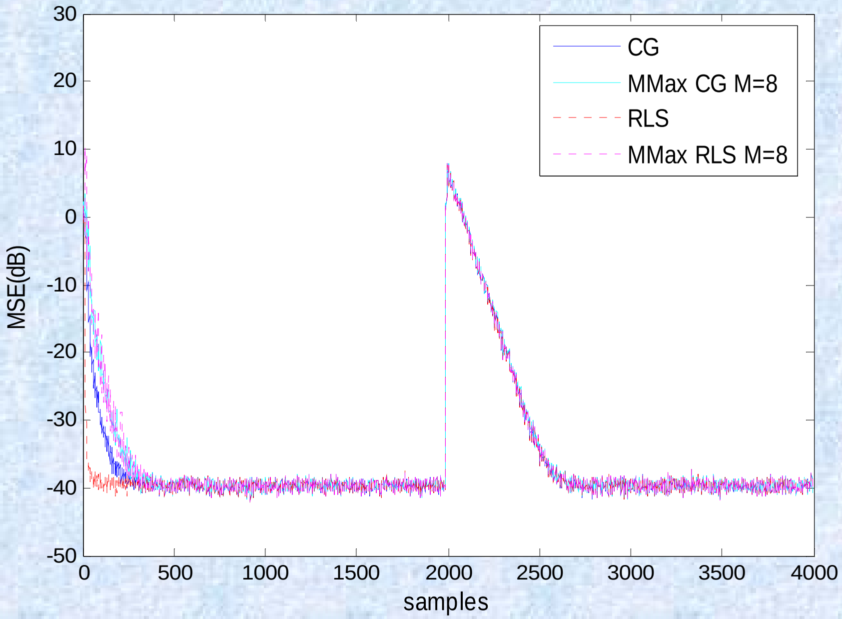
Process noise σ_η	Simulated MSE (dB)	Theoretical MSE (dB)
0.0001	-39.2381	-39.3584
0.001	-32.9019	-32.9019
0.01	-13.4403	-11.0287

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

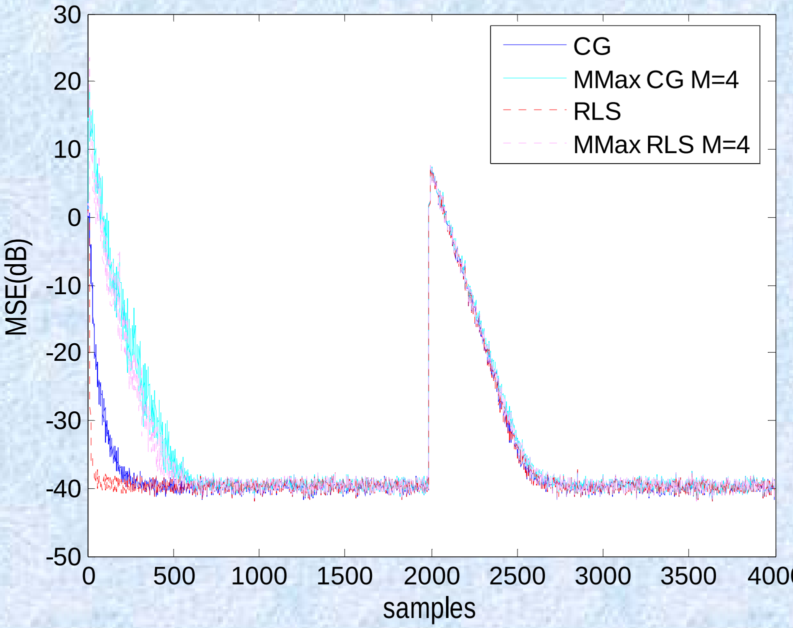
Process noise σ_η	Simulated MSE (dB)	Theoretical MSE (dB)
0.0001	-38.9965	-39.0672
0.001	-31.2768	-30.4378
0.01	-11.6397	-11.0282

The theoretical results match the simulated results.

Performance comparison between MMax CG and MMax RLS



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



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

After 2000 samples/iterations pass, the unknown system is changed by multiplying all coefficients by -1.

The partial update length only affects the convergence rate at the beginning in this case.

Virginia Tech Invent the Future

Wireless @ Virginia Tech

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 (N=8)	731	10
MMax CG (N=4)	679	10
RLS (N=16)	3721	--
MMax RLS (N=8)	825	10
MMax RLS (N=4)	693	10

Summary

- The tracking performance of the MMax CG is analyzed
- Theoretical mean-square performance is derived for white and correlated inputs
- The tracking performance of MMax CG is compared with CG, RLS, MMax RLS by using computer simulations
  - The MMax CG algorithm can achieve similar performance to the full-update CG while reducing computational complexity significantly
  - The MMax CG algorithm can achieve similar performance to the MMax RLS while having lower computational complexity