Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

MI-NLMS adaptive beamforming algorithm for smart antenna system applications*

MOHAMMAD Tariqul Islam, ZAINOL Abidin Abdul Rashid
(*Department of Electrical, Electronics and System Engineering, Faculty of Engineering, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor D.E., Malaysia*)
E-mail: titareq@yahoo.com; zaar@vlsi.eng.ukm.my
Received June 14, 2006; revision accepted July 12, 2006

Abstract: A Matrix Inversion Normalized Least Mean Square (MI-NLMS) adaptive beamforming algorithm was developed for smart antenna application. The MI-NLMS which combined the individual good aspects of Sample Matrix Inversion (SMI) and the Normalized Least Mean Square (NLMS) algorithms is described. Simulation results showed that the less complexity MI-NLMS yields 15 dB improvements in interference suppression and 5 dB gain enhancement over LMS algorithm, converges from the initial iteration and achieves 24% BER improvements at cochannel interference equal to 5. For the case of 4-element uniform linear array antenna, MI-NLMS achieved 76% BER reduction over LMS algorithm.

Key words: Smart antenna, Beamforming algorithm, Least Mean Square (LMS), Normalized LMS (NLMS), Matrix Inversion NLMS (MI-NLMS)
doi:10.1631/jzus.2006.A1709 **Document code:** A **CLC number:** TN828.6

INTRODUCTION

The demand for mobile communication services is increasing at a rapid pace throughout the globe. The increasing demand for mobile communication services in a limited RF spectrum motivates the need for better techniques to improve spectrum utilization. Smart antenna system was adopted by ITU for the IMT-2000 or the Third Generation (3G) wireless networks due to its capability to improve channel capacity and interference suppression. A smart antenna system combines multiple antenna elements with a signal-processing capability to optimize its radiation pattern automatically in response to the signal environment. Beamforming is a key technology in smart antenna systems so that many different adaptive beamforming algorithms have been the subject of active research (Agee, 1989; Chen *et al.*, 2005; Krim and Viberg, 1996; Liberti and Rappapoert, 2002).

Beamforming is a process in which each user's signal is multiplied by complex weight vectors that adjust the magnitude and phase of the signal from each antenna element. Hence the array forms a transmit beam in the desired direction and minimizes the output in the interferer directions. A beamformer appropriately combines the signals received by different elements of an antenna array to form a single output. Classically, this is achieved by minimizing the mean square error (MSE) between the desired output and the actual array output. This principle has its roots in the traditional beamforming employed in sonar and radar systems. Adaptive implementation of the minimum MSE (MMSE) beamforming solution can be realized using temporal reference techniques (Ganz *et al.*, 1990; Godara, 1997; Griffiths, 1969; Litva and Lo, 1996; Reed *et al.*, 1974; Widrow *et al.*, 1967; Wells, 1996).

* Project supported by the IRPA Secretariat, Ministry of Science, Technology and Environment of Malaysia (No. 04-02-02-0029) and the Zamalah Scheme

Many researchers focused on the development of software algorithm, i.e., adaptive beamforming algorithms in mobile communication systems to determine the optimal weight vectors of array antenna elements dynamically, based on different performance criteria. The weight vectors produce the desired radiation pattern that can be changed dynamically, by considering the position of users and interferers to optimize the signal to noise performance.

Among these algorithms, temporal updating algorithms such as Least Mean Square (LMS) and Recursive Least-Squares (RLS) which determine the optimum weight vectors sample by sample in time domain (Widrow *et al.*, 1967; Griffiths, 1969) take a long time to converge. This situation becomes worse if channel situation varies rapidly in time domain, where in such time variance, weight vectors updating becomes more complicated. To overcome this problem, block adaptation approach such as Sample Matrix Inversion (SMI) is employed. However, due to its discontinuity in updating the weight vectors, adaptive block approach is unsuitable for continuous transmission. A new beamforming algorithm that will be easy to implement with less complexity and having faster convergence speed and accurate tracking capability is extremely crucial and a challenging issue to explore. The individual good aspects of both block adaptive and sample by sample techniques will be employed in this paper to address these issues.

This paper presents a novel adaptive beamforming algorithm, the "MI-NLMS", for smart antenna system which combines the normalized LMS (NLMS) and SMI algorithms to improve the convergence speed with small bit error rate (BER). Section 2 of this paper gives a brief account on adaptive beamforming algorithms for smart antenna system. Section 3 discusses the novel proposed algorithm, the MI-NLMS adaptive beamforming algorithm. Section 4 presents the simulation results, and finally Section 5 concludes the paper.

ADAPTIVE BEAMFORMING ALGORITHMS

The purpose of beamforming is to form multiple beams towards desired users while nulling to the interferers at the same time, through the adjustment of the beamformer's weight vectors. Fig.1 shows a generic adaptive beamforming system which requires a reference signal. The signal $\boldsymbol{x}(n)$ received by multiple antenna elements is multiplied with the coefficients in a weight vector $\boldsymbol{w}$ (series of amplitude and phase coefficients) which adjust the phase and amplitude of the incoming signal accordingly. This weighted signal is summed up, resulting in the array output, $y(n)$. An adaptive algorithm is then employed to minimize the error $e(n)$ between a desired signal $d(n)$ and the array output $y(n)$. For the beamformer, the output at time n, $y(n)$, is given by a linear combination of the data at the K sensors and can be expressed as:

$$y(n)=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(n), \quad (1)$$

where $\boldsymbol{w}=[w_1 \ \dots \ w_K]$ and $\boldsymbol{x}(n)=[x_1(n) \ \dots \ x_K(n)]$, H denotes Hermitian (complex conjugate) transpose. The weight vector $\boldsymbol{w}$ is a complex vector. The process of weighting these complex weights $w_1, \dots, w_K$ adjusted their amplitudes and phases so that when added together they form the desired beam. Typically, the adaptive beamformer weights are computed in order to optimize the performance in terms of a certain criterion.

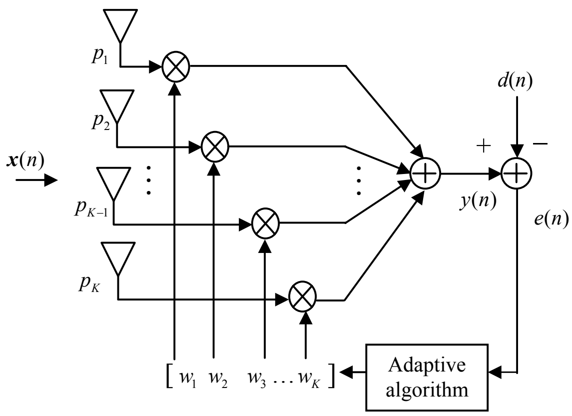

Fig.1 A generic adaptive beamforming system

Most adaptive algorithms are derived by first setting a performance criterion and then generating a set of iterative equations to adjust the weights so that the performance criterion is met. Some of the most frequently used performance criteria include MMSE, maximum signal-to-interference-and-noise ratio (SINR), maximum likelihood (ML), minimum noise variance, minimum output power and maximum gain, etc. (Islam *et al.*, 2003). In order to obtain the optimum weight vector, one needs to know the second order statistics, which are usually unknown and change over time. Adaptive beamforming algorithms estimate them and update the weight vector over time. As the weights are iteratively adjusted, the performance of beamformer approaches the desired criterion. The algorithm is said to be converged when such a performance criterion is met.

There are many types of adaptive beamforming algorithms that exist in the literature. Adaptive beamforming can separate signals transmitted on the same carrier frequency, provided that they are separated in the spatial domain. Most of the adaptive beamforming algorithms can be categorized under two classes according to whether training signal is used or not. These two classes are non-blind adaptive algorithm and blind adaptive algorithm. Non-blind adaptive beamforming algorithm uses a training signal $d(n)$ to update its complex weight vector. This training signal is sent by the transmitter to the receiver during the training period. Beamformer in the receiver uses this information to compute new complex weight. LMS, NLMS, RLS and SMI algorithms are categorized as non-blind algorithm. Blind algorithms on the other hand do not need any training sequence to update its complex vector. Constant Modulus Algorithm (CMA), Spectral self-COherence REstoral (SCORE), and Decision Directed (DD) algorithms are examples of blind adaptive beamforming algorithms. These algorithms use some of the known properties of the desired signal. In blind algorithm, the goal is to retrieve the input signal by using output signal and possibly the statistical information for the input.

MI-NLMS ADAPTIVE BEAMFORMING ALGORITHM

In this section, a novel optimum MI-NLMS algorithm is introduced for adaptive beamforming. In MI-NLMS algorithm, the SMI algorithm is utilized to determine the optimum weight vectors assigned to each of the antenna elements of the array instead of arbitrary value before calculating the final weight vector. The weight is calculated only for the first few samples or for a small block of incoming data. The weight coefficients derived by SMI algorithm are set as initial coefficients and are updated by introducing NLMS algorithm. However, to improve the stability of the system and convergence speed, NLMS method is used instead of LMS. NLMS is LMS, but the step size is changed as the correlation matrix is changed to avoid unstable system (Yasui *et al*., 2002). The weights are chosen to minimize the MSE between the beamformer output and the reference signal as:

$$\boldsymbol{w}=\boldsymbol{R}^{-1}\boldsymbol{r}. \tag{2}$$

It has faster convergence since Eq.(2) employs direct inversion of the covariance matrix $\boldsymbol{R}$ (Litva and Lo, 1996). The initial optimum weight can be calculated by using Eq.(2) where covariance matrix is $\boldsymbol{R}(n)=\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{H}}(n)$ and cross-correlation matrix is $\boldsymbol{r}(n)=d^{*}(n)\boldsymbol{x}(n)$.

In practice, the signals are not known and the signal environment frequently changes. Thus, adaptive processors must continually update the weight vector to meet the new requirements imposed by the varying conditions. Optimal weight vectors can be computed by obtaining an estimation of the covariance matrix $\boldsymbol{R}$ and the cross-correlation matrix $\boldsymbol{r}$ in a finite observation interval and then these estimates are used to obtain the desired vector.

The estimation of both $\boldsymbol{R}$ and $\boldsymbol{r}$ over a block size N_2-N_1 can be evaluated respectively as follows:

$$\boldsymbol{R}=\sum_{i=N_1}^{N_2}\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{H}}(n), \tag{3}$$

$$\boldsymbol{r}=\sum_{i=N_1}^{N_2}d^{*}(n)\boldsymbol{x}(n), \tag{4}$$

where, N_1 and N_2 are the lower and upper limits of observation interval or window and n is the sample index. This limit is taken to be small, to ensure that the effect due to changes in the signal environment during block acquisition does not affect the performance of the algorithm. Also, large limit or block only means more matrix inversions, making the algorithm computationally intensive.

The SMI algorithm requires the calculation of the inverse covariance matrix $\boldsymbol{R}$ and this incurs high computational complexity. The LMS algorithm (Haykin, 1996) avoids matrix inversion operation by using instantaneous gradient vector ∇J to update the weight vector. If $\boldsymbol{w}(n)$ denotes the estimate of the weight vector at the nth iteration and $J(n)$ is the mean square error, the next estimation of the weight vector for the $(n+1)$th iteration, $\boldsymbol{w}(n+1)$ is estimated according to the following simple recursion

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+0.5\mu[-\nabla J(n)], \tag{5}$$

where μ is a small positive constant, called the step size whose value is between 0 and 1. The LMS algorithm is based on the steepest-descent method which recursively computes and updates the weight vector. From LMS algorithm we know that

$$y(n)=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(n), \tag{6}$$

$$e(n)=d(n)-y(n), \tag{7}$$

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu\boldsymbol{x}(n)e^{*}(n). \tag{8}$$

In the LMS algorithm, Eq.(8) shows that the product vector $\mu\boldsymbol{x}(n)e^{*}(n)$ at iteration n, applied to the weight vector $\boldsymbol{w}(n)$, is directly proportional to the input vector $\boldsymbol{x}(n)$. Therefore, the LMS algorithm experiences a gradient noise amplification problem when the input signal $\boldsymbol{x}(n)$ is large, i.e., the product vector $\mu\boldsymbol{x}(n)e^{*}(n)$ is large. This is solved by normalization of the product vector at iteration $n+1$ with the square Euclidean norm of the input vector $\boldsymbol{x}(n)$ at iteration n (Haykin, 1996). The change of the weight vector is given as

$$\delta\boldsymbol{w}(n+1)=\boldsymbol{w}(n+1)-\boldsymbol{w}(n), \tag{9}$$

which is subject to the constraint in order to minimize the square Euclidean norm in weight vector $\boldsymbol{w}(n+1)$ with respect to its previous value $\boldsymbol{w}(n)$. So

$$\boldsymbol{w}^{\mathrm{H}}(n+1)\boldsymbol{x}(n)=d(n), \tag{10}$$

where $d(n)$ is the desired value. The method of Lagrange multipliers is used to solve this constrained optimization problem [details about Lagrange multipliers can be found in (Haykin, 1996)].

Using the complex constraint of Eq.(10) the conjugate Lagrange multiplier becomes

$$\lambda^{*}=\frac{2}{\|\boldsymbol{x}(n)\|^{2}}[d^{*}(n)-\boldsymbol{w}^{\mathrm{T}}(n)\boldsymbol{x}^{*}(n)]. \tag{11}$$

Furthermore, substituting Eqs.(6) and (7) into Eq.(11) yields

$$\lambda^{*}=\frac{2}{\|\boldsymbol{x}(n)\|^{2}}e^{*}(n). \tag{12}$$

After manipulation equivalently, the weight vector can be written as

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\frac{\mu}{\|\boldsymbol{x}(n)\|^{2}}\boldsymbol{x}(n)e^{*}(n). \tag{13}$$

As can be seen from Eq.(13), the algorithm reduces the step size μ to make the changes large. As a result, the step size μ varies adaptively by following the changes in the input signal level. This prevents the update weights from diverging and makes the algorithm more stable and faster converging than when a fixed step size is used. In addition, the NLMS algorithm is used as the MMSE method needs to cope with the large changes in the signal levels of mobile communication systems. By combining the above two algorithms, the novel optimum MI-NLMS algorithm updates the weight vectors according to the following equations:

$$\begin{aligned}
&\boldsymbol{R}=\sum_{i=N_1}^{N_2}\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{H}}(n),\quad \boldsymbol{r}=\sum_{i=N_1}^{N_2}d^{*}(n)\boldsymbol{x}(n),\\
&\boldsymbol{w}=\boldsymbol{R}^{-1}\boldsymbol{r},\quad y(n)=\boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(n),\quad e(n)=d(n)-y(n),\\
&\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu\boldsymbol{x}(n)e^{*}(n)\\
&\qquad=\boldsymbol{w}(n)+\frac{\mu}{\|\boldsymbol{x}(n)\|^{2}}\boldsymbol{x}(n)e^{*}(n)\\
&\qquad=\boldsymbol{w}(n)+\frac{\mu}{\|\boldsymbol{x}(n)\|^{2}}\boldsymbol{x}(n)\left[d(n)-\boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(n)\right]^{*}.
\end{aligned} \tag{14}$$

The final weight vector of the MI-NLMS algorithm is estimated from Eq.(14). In the MI-NLMS algorithm, advantages of both the block adaptive and sample by sample techniques are employed. In this algorithm, the initial weight vector is obtained by matrix inversion through SMI algorithm, only for the first few samples or for a small block of incoming data instead of arbitrary value before calculating the final weight vector. The final weight vector is updated by using the LMS algorithm. The flowchart of the MI-NLMS algorithm is shown in Fig.2.

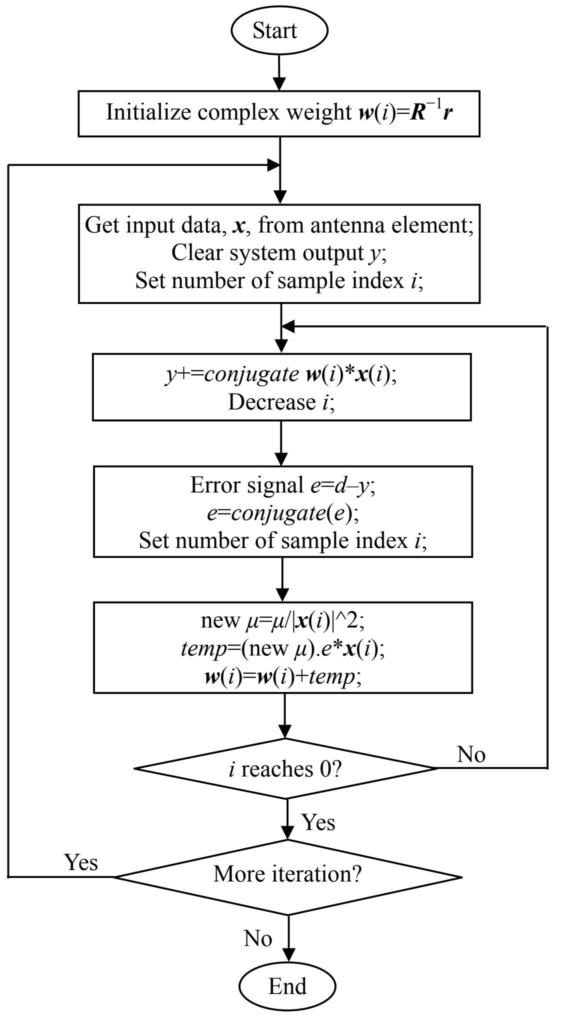

Fig.2 Flow chart of the MI-NLMS algorithm

The above description of the adaptive procedure based on the MI-NLMS algorithm can be given as follows:

Step 1: Initial calculation of weight vector by matrix inversion (first few samples or small block of incoming data).

Step 2: Calculate the error and scale the input vector by the equations of the algorithm.

Step 3: Normalize the weight vector.

Step 4: Update the weight vector by final equation until convergence.

RESULTS

In this work, we consider a simple uniform linear array antenna of 4×1 element with half wavelength spacing between the elements, located at the base station to perform spatial filtering. Data sequences are generated using Binary Phase Shift Keying (BPSK) modulation and for the sake of simplicity the radio channel is assumed to be multipath free and non-dispersive with Additive White Gaussian Noise (AWGN). In the simulation, the angle of arrival of the desired user and the interferer are at 60° and 30° respectively. The signal to noise ratio (SNR) is set at 8 dB and the number of data bit length is 300. Error convergence plot, bit error rate, adaptive array pattern performances and weight convergence of the MI-NLMS are evaluated and compared with LMS algorithm to demonstrate the merits of this new algorithm.

Figs.3 and 4 show the MSE plot or convergence plot for the LMS and the MI-NLMS algorithm respectively.

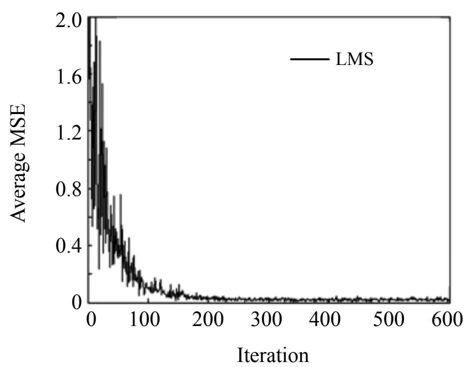

Fig.3 Mean square error plot for the LMS algorithm

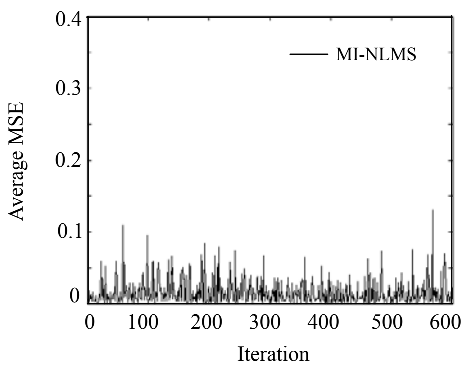

Fig.4 Mean square error plot for the MI-NLMS algorithm

As shown in these figures, for the same adaptation size or iterations, the MI-NLMS algorithm can achieve faster convergence than the LMS algorithm. Also the LMS algorithm starts to converge from the iteration number 200 whereas in the MI-NLMS algorithm it starts to converge from the initial iteration. In this case, the LMS error is almost 0.0979 and the MI-NLMS error is almost 0.0206 at around 100 iterations.

Figs.5 and 6 present the linear and polar plot of the beampattern for the MI-NLMS and LMS algorithm respectively. These figures show that the MI-NLMS generates a deeper null of about −70 dB and LMS generates a null of about −55 dB towards the interferer, giving about 15 dB improvements for the MI-NLMS algorithm in interference suppression with respect to LMS algorithm. Both algorithms have their main beam pointed to the desired user direction. The ratio of main lobe to the first side lobe is 30 dB and 25 dB for MI-NLMS and LMS respectively.

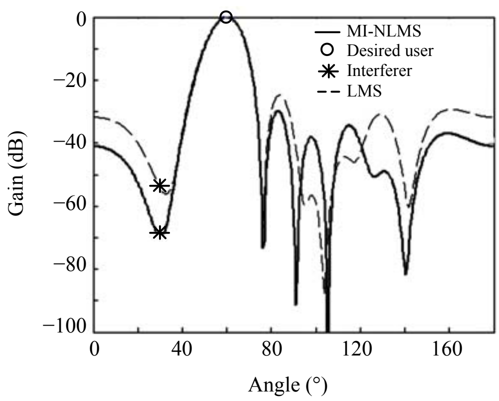

Fig.5 Comparison of linear beampattern of MI-NLMS and LMS algorithms

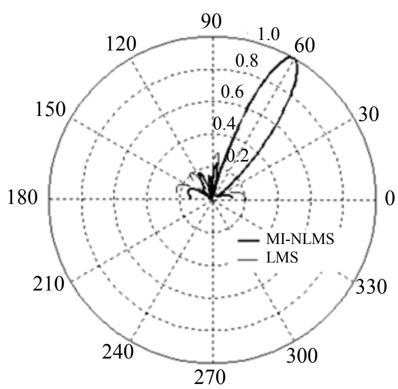

Fig.6 Comparison of polar beampattern of MI-NLMS and LMS algorithms

CoChannel Interference (CCI) is a major factor that limits the capacity of a cellular system. To increase the capacity of a cellular system, frequency is reused, i.e., a frequency band is used in two different cells belonging to different clusters, which are sufficiently separated so that they do not interfere significantly with each other. The BER performances with respect to the number of CCI of the two algorithms are presented in Fig.7. As can be seen from the figure, for CCI equal to 5, BER for the LMS and the MI-NLMS are 4.51×10^{-2} and 3.45×10^{-2} respectively. Fig.7 also shows that BER increases as the number of interferer increases for both algorithms and that MI-NLMS provide about 24% BER improvement at CCI equal to 5 over that of LMS algorithm.

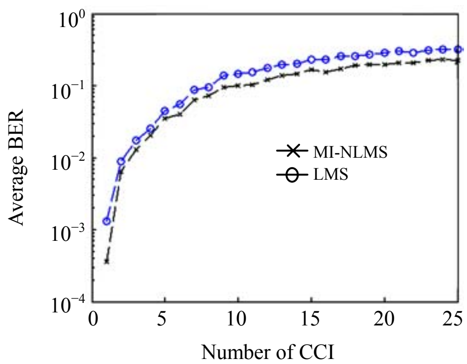

Fig.7 BER performance of the MI-NLMS and LMS algorithms with varying CCI

Fig.8 shows the comparison of BER with 4-element antenna array for MI-NLMS and LMS algorithms. The BER performance is greatly improved in the MI-NLMS algorithm compared to the LMS algorithm. The BER rate of the MI-NLMS and LMS algorithm are 2.9×10^{-2} and 12.3×10^{-2} respectively. The reduction of the BER for MI-NLMS is 76% compared to LMS algorithm.

The magnitude of the complex weights plotted against the number of samples for each antenna element is presented in Fig.9 by using LMS algorithm. The convergence of the weights to their optimum values for LMS algorithm is shown in the figure. The complex weights at the iteration 1200 for LMS algorithm are as follows:

$$w_1=0.2071+0.0819i,\quad w_2=0.0036+0.2068i,$$
$$w_3=-0.1938+0.0193i,\quad w_4=-0.0295-0.2612i.$$

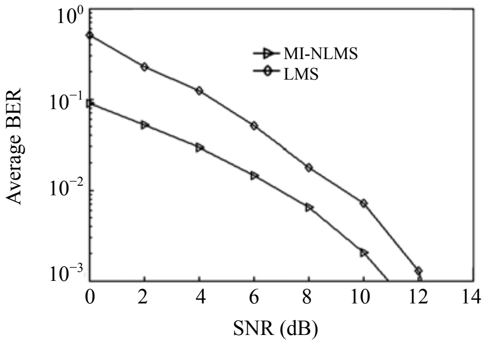

Fig.8 Performance of BER with 4-element antenna array for MI-NLMS and LMS algorithms

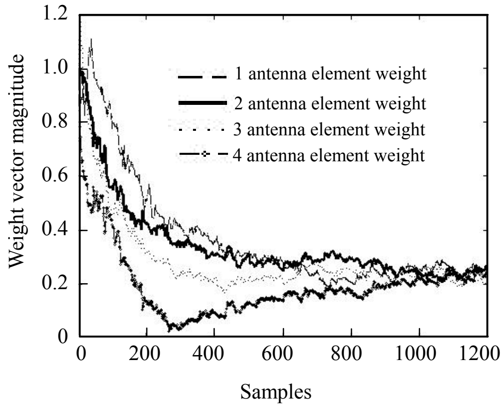

Fig.9 Weight convergence plot of the first 4 antenna elements for LMS algorithm

Fig.10 shows the magnitude of the complex weights plotted against the number of samples for each antenna element by using MI-NLMS algorithm. It is evident that the weight values converge to their optimum values for MI-NLMS algorithm.

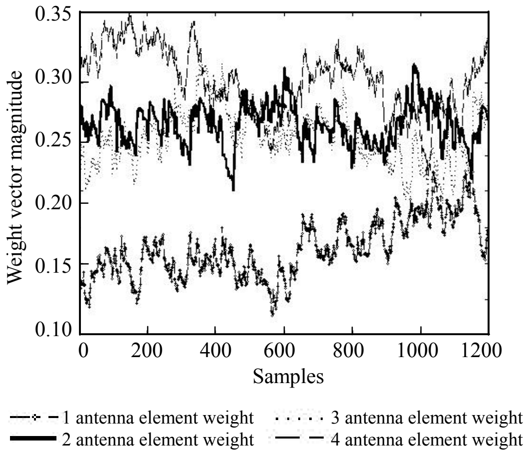

Fig.10 Weight convergence plot of the first 4 antenna elements for MI-NLMS algorithm

The complex weights at the iteration 1200 for MI-NLMS algorithm are as follows:

$$w_1=0.2116+0.0269i,\quad w_2=-0.0105+0.2197i,$$
$$w_3=-0.2199+0.0535i,\quad w_4=-0.0120-0.2805i.$$

As can be observed from Fig.10, the MI-NLMS starts adaptation to optimum weights from the initial weight vector values. On the contrary, LMS algorithm has to converge from the arbitrary weight values (in this case 1) to the optimum weight values. However, before it converges to its optimum values, the interfering directions will change; consequently more iterations are needed to converge.

CONCLUSION

In this paper, we introduced a novel and less complex adaptive beamforming algorithm, the "MI-NLMS", for smart antenna system. MI-NLMS combines the SMI and NLMS algorithms to improve the convergence speed with small BER. In this algorithm individual good aspects of both the sample by sample and block adaptive algorithms are employed. MI-NLMS computes the optimal weight vector based on the SMI algorithm and updates the weight vector by NLMS algorithm. Simulation results showed that the MI-NLMS algorithm provides remarkable improvements in terms of interference suppression, convergence rate and BER performance over that of LMS algorithms. With respect to LMS, MI-NLMS provides: (1) 15 dB improvements in interference suppression, (2) 5 dB gain enhancement, (3) convergence from the initial iteration (whereas LMS convergence from the iteration number 200), and (4) 24% BER improvements at CCI equal to 5. The reduction of the BER for MI-NLMS is 76% compared to LMS algorithm in the case of 4-element antenna array.

References

Agee, B., 1989. Blind separation and capture of communication signals using a multitarget constant modulus beamformer. *Proceedings of the IEEE Military Communications Conference*, **2**:340-346.

Chen, S., Ahmad, N.N., Hanzo, L., 2005. Adaptive minimum bit-error rate beamforming. *IEEE Transactions on Wireless Communications*, **4**(2):341-348. [doi:10.1109/TWC.2004.842981]

Ganz, M.W., Moses, R.L., Wilson, S.L., 1990. Convergence of the SMI and the diagonally loaded SMI algorithms with weak interference (adaptive array). *IEEE Trans. Antennas Propagat.*, **38**(3):394-399. [doi:10.1109/8.52247]

Godara, L.C., 1997. Applications of antenna arrays to mobile communications. Part I: performance improvement, feasibility, and system considerations. *Proc. IEEE*, **85**(7): 1031-1060.

Griffiths, L.J., 1969. A simple adaptive algorithm for real-time processing in antenna arrays. *Proc. IEEE*, **57**:1696-1704.

Haykin, S., 1996. Adaptive Filter Theory (3rd Ed.). Prentice Hall, New York.

Islam, M.T., Ping, C.C., Rashid, Z.A.A., 2003. Performance Evaluation of Adaptive Non-blind Algorithms of a Digital Beamforming System for Linear Array Antenna. The 6th International Conference on Computer & Information Technology. Dhaka, Bangladesh, p.686-691.

Krim, H., Viberg, M., 1996. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, **13**(4):67-94. [doi:10.1109/79.526899]

Liberti, J.C., Rappapoert, T.S., 2002. Smart Antenna for Wireless Communications Is-95 and Third Generation CDMA Applications. Prentice-Hall PTR, New Jersey.

Litva, J., Lo, T.K.Y., 1996. Digital Beamforming in Wireless Communications. Artech, London, U.K.

Reed, I.S., Mallett, J.D., Brennan, L.E., 1974. Rapid convergence rate in adaptive arrays. *IEEE Trans. Aerosp. Electron. Syst.*, **10**:853-863.

Wells, M.C., 1996. Increasing the capacity of GSM cellular radio using adaptive antennas. *IEE Proc. Comm.*, **143**(5): 304-310. [doi:10.1049/ip-com:19960743]

Widrow, B., Mantey, P.E., Griffiths, L.J., Goode, B.B., 1967. Adaptive antenna systems. *Proc. IEEE*, **55**:2143-2159.

Yasui, Y., Kobaysakawa, S., Nakamura, T., 2002. Adaptive array antenna for W-CDMA systems. *FUJITSU Sci. Tech. J.*, **38**(2):192-200.