Performance Analysis of Array Signal Processing Algorithms for Adaptive Beamforming

Saurav Ganguly*(1), Jayanta Ghosh (1), and Puli Kishore Kumar (2)
(1) National Institute of Technology, Patna, India, 800005, http://www.nitp.ac.in
(2) National Institute of Technology, Andhra Pradesh, India

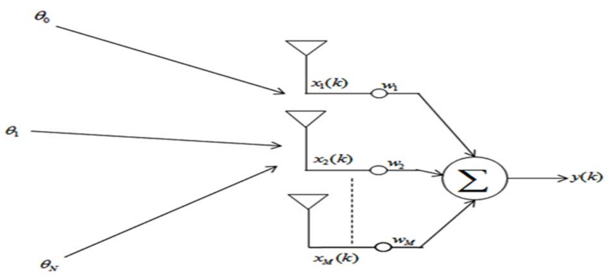
Abstract:

In this paper, performance analysis in terms of beamforming, null steering and rate of convergence is implemented on blind and non-blind adaptive algorithms for smart antenna system. In non-blind category, Least Mean Square (LMS) and its variant, the Normalized Least Mean Square (NLMS) algorithms are discussed and analysed while in blind category, Constant Modulus Algorithm (CMA) based on Steepest Descent method and Least Square (LS-CMA) method are discussed and analysed.

1. Introduction

A spatial filter [1], most commonly denotes an array of antennas, interfaced with an adaptive signal processor, which can adjust or adapt its own beam pattern so that to emphasize the Signal-of-Interest (SOI) and to minimize interfering signals. Smart antenna system, also known as adaptive array antennas takes the advantage of diversity effect at the source (transmitter), the destination (receiver) or both.

Spatial filtering or beamforming [2] is realized by creating radiation pattern of antenna arrays by constructively adding and combining the amplitude and phase in the direction of SOI and minimizing the interfering signals. This can be achieved by using an array of antenna combined with weights, as shown in Fig.1.



gure1. A Spatial Filter

But if the SOI direction changes with time, the array weights must be adapted for better optimization. This adaptation is carried out by a signal processing algorithm at the receiver that must allow for the continuous variation to a changing channel environment. Optimization of the array weights with some adaptive signal processing is known as adaptive beamforming.

Adaptive beamforming algorithms can be classified [3] as: (1) blind, and (2) non-blind. Blind algorithms do not require any training data or reference signal for updating of weights. Constant Modulus Algorithm (CMA) is an example of blind adaptive algorithm. On the other hand, non-blind algorithms use a reference or desired signal to modify the array weights iteratively, so that at the end of each iteration, the output is compared with the desired signal and the generated error signal is used in the algorithms to adapt the weights. A prominent example of this type of algorithms is Least Mean Square (LMS) algorithm.

In this paper, section 2 is devoted in discussing the mathematical aspects of LMS, NLMS, CMA and LS-CMA algorithms. In section 3, simulations of the above algorithms are performed based on MATLAB. The results are analysed and comparison is accomplished in terms of beamforming, null steering and rate of convergence. Discussions and conclusions are drawn in section 4 of this paper.

2. Adaptive Algorithms

A. LMS algorithm

The LMS algorithm is a gradient based approach [3], where an iterative technique called the method of steepest descent is employed to approximate the gradient of the Mean Square Error (MSE), also known as the cost function given by [4] [5]:

$$J(n) = E[e^2(n)] \tag{1}$$

where $J(n)$ is the cost function (or MSE) to be minimized
$E[\cdot]$ is the statistical expectation operator
$e(n)$ is the error signal

The iteration is performed by moving the weights in the negative direction of the gradient by a small amount, which is known as the step size [3] [5]. The array weights are updated as per the equations given below:

$$\bar{w}(n+1) = \bar{w}(n) + \mu e^*(n)\bar{x}(n) \tag{2}$$

$$e(n) = d(n) - y(n) = d(n) - \bar{w}^H(n)x(n) \tag{3}$$

where

n is the iteration number
μ is the step size
$d(n)$ is the desired response
$\bar{x}\ (n)$ is the received signal vector and
$\bar{w}^H\ (n)$ denotes the Hermitian transpose of the weight vector $\bar{w}\ (n)$

The convergence of the LMS algorithm is directly proportional to the step-size factor μ. When the step size is within the ranges that assure convergence, the weights estimated by iteration leads to optimal weights. For stability, the following condition must be satisfied:

$$0 \leq \mu \leq \frac{1}{2\lambda_{\max}} \qquad (4)$$

where $\lambda_{\max}$ is the largest eigen value of the array correlation matrix $\bar{R}_{xx}$. LMS algorithm is based on instantaneous estimate of $\bar{R}_{xx}$, given by:

$$\hat{R}_{xx}(n) \approx \bar{x}(n)\,\bar{x}^H(n) \qquad (5)$$

B. NLMS algorithm

Normalised Least Mean Square (NLMS) algorithm is a variant of the LMS method discussed above whose structures are identical. In LMS, the adaptation of weight vector at iteration $n+1$ is the product of estimation error$e$, tap input vector $x(n)$ and step size parameter μ [3]. When $x(n)$ is large, the LMS filter suffers from gradient noise amplification. Also, the choice of μ is difficult that dictates the convergence of the algorithm [6] [7]. To overcome these problems, the weight vector update equation is normalized by the squared Euclidian norm of the input vector at iteration n.The array weights are updated as per equation (6) given below [8]:

$$\bar{w}(n+1) = \bar{w}(n) + \frac{\mu e^*(n)\bar{x}(n)}{\bar{x}^H(n)\bar{x}(n)} \qquad (6)$$

NLMS differs from LMS algorithm only in terms of weight updating mechanism. In NLMS coefficient updating equation is normalized by the factor which is equal to the product of input vector $x(n)$ and its transpose.

B. CM algorithm

Constant Modulus Algorithm (CMA) is used in adaptive beamforming for blind adaptation. This algorithm is derived keeping in view the constant complex envelope (amplitude) property of the signal. CMA does not require a reference signal to compare with the output. [9][10][11].If the arriving signal has constant amplitude, then this algorithm maintains and restores the amplitude of desired signal. It updates weights by minimizing the cost function given by:

$$J(p,q) = E[(|y(n)|^p - 1)^q] \qquad (7)$$

where $y(n)$ denotes the array output
$|\cdot|$ denotes the modulus function
p and q are non-negative integers

The cases where p=1 or 2 with q=2 are referred to as Constant Modulus Algorithm. The p=1 case converges more rapidly than p=2 case [1]. We consider the p=1 with q=2 note in this paper. The corresponding cost function of consideration becomes:

$$J(1,2) = J(\bar{w}) = E[(|y(n)| - 1)^2] = E[(|\bar{w}^H(n)x(n)| - 1)^2] \qquad (8)$$

The weight update equation thus becomes:

$$\bar{w}(n+1) = \bar{w}(n) - \mu\bar{x}(n)\left(y(n) - \frac{y(n)}{|y(n)|}\right) \qquad (9)$$

Equation (9) is similar to (2), of LMS algorithm, with the exception of error update as $\left(y(n) - \frac{y(n)}{|y(n)|}\right)$.

CMA doesn't require any desired signal or reference to generate the error signal. It routinely selects one or several of the multipath as the desired signal. When array vector is updated it does not need to know the arrival timings of the incident waves. It does not need to synchronously sample the received signal with the clock timing.

C. LS-CM algorithm

The CMA technique discussed above is based on the method of steepest descent, where the gradient of the cost function, which is defined in (7), is executed. An alternate process based on the method of nonlinear least-squares is known as the LS-CM algorithm [12]. This method is also known as autoregressive estimator [1], based on a least square minimization. It is a block update or a batch processing approach, where the cost function is defined as:

$$J(\bar{w}) = \sum_{k=1}^{K} |e_k(\bar{w})|^2 = \|\bar{\epsilon}(\bar{w})\|_2^2 \qquad (10)$$

where $e_k(\bar{w})$ = the error at the k^{th} data sample
$\bar{\varepsilon}(\bar{w}) = [\bar{e}_1(\bar{w}) \quad \bar{e}_2(\bar{w}) \ldots\ldots\ldots \bar{e}_K(\bar{w})]^T$
K= the number of data samples in each block

In dynamic LS-CMA, to maintain latest adaptation in a changing signal environment, the weights are updated as:

$$\bar{w}(n+1) = [\bar{X}(n)\bar{X}^H(n)]^{-1}\bar{X}(n)\bar{r}^*(n) \qquad (11)$$

where $\bar{X}(n)$ is the dynamic block of data at the array output before applying weights, which is given as:

$$\bar{X}(n) = [\bar{x}(1+nK) \;\; \bar{x}(2+nK) \ldots\ldots\ldots \bar{x}(K+nK)] \qquad (12)$$

The weighted array output for the n^{th} is given as:

$$\bar{y}(n) = [y(1+nK) \;\; y(2+nK) \ldots\ldots\ldots y(K+nK)]^T \qquad (13)$$

or, $\bar{y}(n) = [\bar{w}^H(n)\bar{X}(n)]$ (14)

$\bar{r}(n)$ is complex limited output data vector defined as:

$$\bar{r}(n) = \left[\frac{y(1+nK)}{|y(1+nK)|} \;\; \frac{y(2+nK)}{|y(2+nK)|} \cdots \frac{y(K+nK)}{|y(K+nK)|}\right] \qquad (15)$$

3. Simulation Results

The algorithms discussed in the previous section are simulated in MATLAB and the results are examined in this section. A Uniform Linear Array (ULA) with dipole antennas operating at 1 GHz is selected for simulation. The spacing between the elements is fixed at $d = 0.5\lambda$, where λ is the wavelength of the signal. This is the optimum separation distance between the antenna elements. The Direction of Arrival (DOA) of the SOI is chosen at 60^0 and the interference signal direction is selected at -30^0. The factors used for our simulation are given in Table 1.

TABLE I. SIMULATION PARAMETERS

Parameter Name	Parameter Value
Type of antenna	Dipole
Type of array	ULA
Number of array elements (N)	[8 10 15]
Element spacing	0.5λ
Noise Variance (σ^2)	0.001
Total number of simulations	100
Block length (K) for LS-CMA	20
Variability	$-90^0 \leq \theta \leq 90^0$
Frequency of operation	1 GHz
DOA of the desired signal	60^0
DOA of the interference signal	-30^0

The number of elements (N) in the array is arbitrarily chosen as 8, 10 and 15. The performance characteristics based on beamforming, null steering and convergence capabilities are compared by varying N in the array. Two multipath signals are considered for CMA and LS-CMA acting as interference in the directions of -30^0 and 0^0, with 30% and 10% of the direct path amplitude, which is at 60^0.

A. *Study of beamforming and null-steering*

Fig.2 shows the simulation results of amplitude pattern of normalized Array Factor (AF) due to different number of array elements (N) for LMS, NLMS, CMA and LS-CMA algorithms.
Table II, III, IV and V gives their beamforming study.

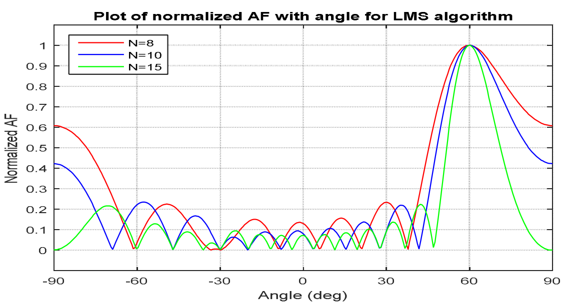


Figure 2(a). Normalized Array Factor plot for LMS algorithm due to N=8, 10 and 15 array elements

TABLE II. ANALYSIS OF BEAMFORMING FOR LMS ALGORITHM

No. of Array Elements (N)	Half Power Beam Width (HPBW) at 60^0	Maximum Side Lobe Level (normalized)	Null depth at -30^0 (normalized)
8	29.79	0.61	0.00
10	22.34	0.42	0.00
15	13.96	0.22	0.00

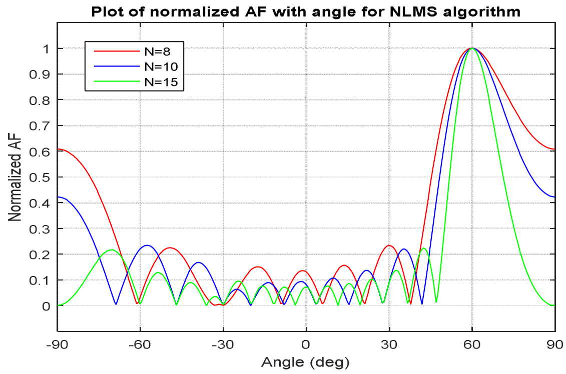


Figure 2(b). Normalized Array Factor plot for NLMS algorithm due to N=8, 10 and 15 array elements

TABLE III. ANALYSIS OF BEAMFORMING FOR NLMS ALGORITHM

No. of Array Elements (N)	Half Power Beam Width (HPBW) at 60^0	Maximum Side Lobe Level (normalized)	Null depth at -30^0 (normalized)
8	29.64	0.59	0.00
10	22.12	0.41	0.00
15	13.50	0.21	0.00

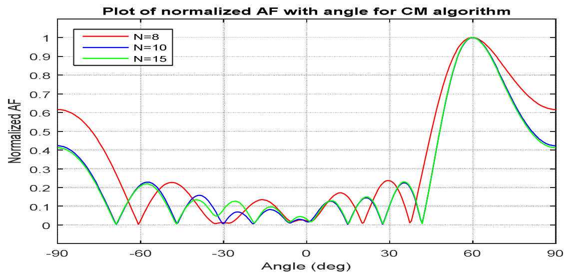


Figure 2(c). Normalized Array Factor plot for CM algorithm due to N=8, 10 and 15 array element

TABLE IV. ANALYSIS OF BEAMFORMING FOR CM ALGORITHM

No. of Array Elements (N)	Half Power Beam Width (HPBW) at 60^0	Maximum Side Lobe Level (normalized)	Null depth at -30^0 (normalized)	Null depth at 0^0 (normalized)
8	32.11	0.64	0.008	0.02
10	24.15	0.43	0.001	0.01
15	21.66	0.42	0.008	0.02

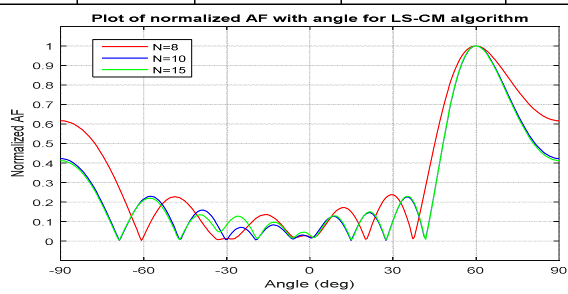


Figure 2(d). Normalized Array Factor plot for LS-CM algorithm due to N=8, 10 and 15 array element

TABLE V. ANALYSIS OF BEAMFORMING FOR LS-CM ALGORITHM

No. of Array Elements (N)	Half Power Beam Width (HPBW) at 60^0	Maximum Side Lobe Level (normalized)	Null depth at -30^0 (normalized)	Null depth at 0^0 (normalized)
8	32.11	0.64	0.008	0.02
10	24.15	0.43	0.001	0.01
15	21.66	0.42	0.008	0.02

The above simulated results in the figures and tables clearly depict the fact that as the number of array elements increases, the beamwidth decreases at the angle of arrival of the SOI. Hence the array antenna becomes more directional at 60^0 in this case. At N=15, the directivity is high in all algorithms but the performance rather degrades due to the existence of higher number of side lobes. For N=8, the number of side lobes are less but the normalized maximum SLL is high in all cases. Also, the beamwidth is reasonably higher. Comparing between LMS and NLMS, we find that both provides almost identical results in terms of beamforming and null-steering, as illustrated in Fig.2(a), 2(b) and Table II and III. Furthermore, we find CMA and LS-CMA gives similar results, as shown in Fig.2(c), 2(d) and Table IV and V. We also find that null depth is not zero for the blind algorithms at -30^0 or at 0^0. LMS and NLMS provides better capabilities in terms of beamforming and null-steering as shown in Fig.2 Table II, III, IV and V, but in the expense of being non-blind.

B. Study on rate of convergence

Total number of iterations chosen in each case is, $n = 100$. Also, $d = 0.5\lambda$ is kept fixed. Fig.3 shows the variation of MSE with number of iterations for LMS and NLMS algorithms with N=8, 10 and 15 respectively. Fig.4 shows the variation of absolute error with the number of iterations for CMA and LS-CM algorithms with N=8, 10 and 15 respectively.

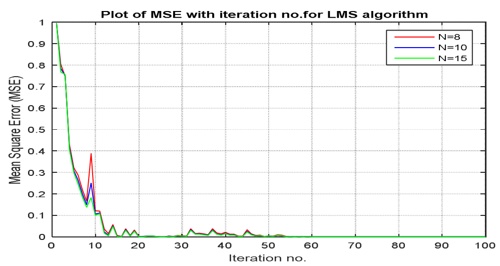


Figure 3(a). A plot of variation of MSE with iteration number for LMS algorithm with N=8, 15 and 30 array elements.

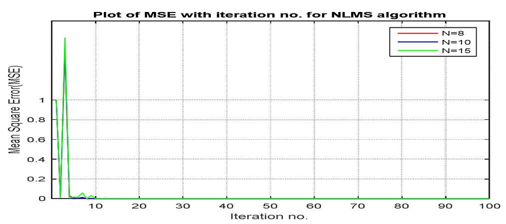


Figure 3(b). A plot of variation of MSE with iteration number for NLMS algorithm with N=8, 15 and 30 array elements.

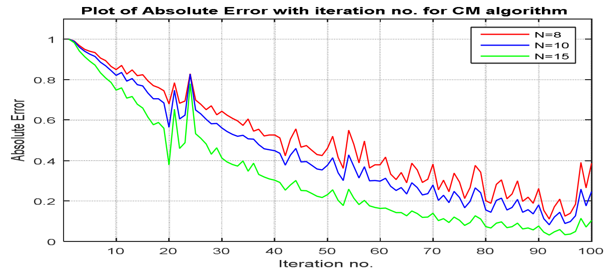


Figure 4(a). A plot of variation of Absolute Error with iteration number for CM algorithm with N=8, 15 and 30 array elements.

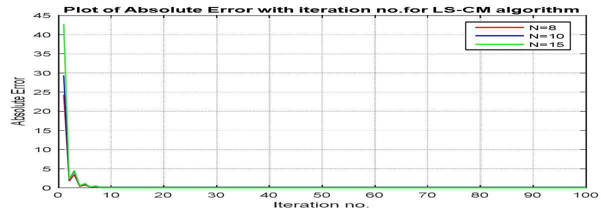


Figure 4(b). A plot of variation of Absolute Error with iteration number for LS-CM algorithm with N=8, 15 and 30 array elements.

Fig.3(a) plot offers a notion on the rate of convergence of the LMS algorithm. It can be observed that the MSE reduces to 0 in between 50 to 60 numbers of iterations for all values of N. Hence, from the plot, we can conclude that the weights converge to their optimum solution at almost 60 iterations. This means about 50% of the iterations are used for convergence. This is relatively slow adaptation. For fast changing input signal characteristics, the LMS algorithm may not be able to track the signal due to slow convergence rate. To counter the slow convergence of LMS algorithm, the NLMS algorithm can be used. Fig.3(b) shows that the MSE converges at around 10 iterations only. This is a significant improvement from the LMS algorithm.

Fig.4(a) provides an idea of constancy of absolute error with the iteration number for CM algorithm. Its convergence is extremely slow. During the simulation it was found that CM algorithm is quiet unstable. This slow convergence rate and instability limits the usefulness of this algorithm in a rapidly changing algorithm. The convergence rate can be significantly improved by block adaptation as we observe in Fig.4(b) with block length K=20 in LS-CM algorithm. It is observed that for values of N, the algorithm converges at around 10 iterations.

4. Acknowledgement

The first author would like to acknowledge the Ministry of Electronics and Information Technology (MeitY), Government of India for supporting and financial assistance during research work through “Visvesvaraya Ph.D. Scheme for Electronics and IT performances as that of LMS, with much faster convergence rate.

References

[1] F. Gross, in *Smart Antennas for WirelessCommunications*, McGraw-Hill, 2005, pp. 227-249.

[2] L. C. Godara, “Applications of Antenna Arrays to Mobile Communocations, Part-II: Beamforming and Direction-of-Arrival Considerations,” *Proceedings of IEEE,* vol. 85, no. 8, pp. 1195-1245, August 1997.

[3] S. Haykin, Adaptive Filter Theory, New York: Prentice Hall, 2007.

[4] W. Y. Leong, “Angle-of-Arrival Estimation: Beamforming Based Smart Antennas,” in *3rd IEEE Conference on Industrial Electronics and Applications*, June 2008.

[5] P. Saxena and A. Kothari, “Performance Analysis of Adaptive Beamforming Algorithms for Smart Antennas,” in *International Conference on Future Information Engineering*, 2014.

[6] V. Veen, D. Barry and K. M. Buckley, “Beamforming: A Versatile Approach to Spatial Filtering,” *ASSP Magazine, IEEE,* pp. 4-24, 1988.

[7] J. Homer, P. J. Kootsookos and V. Selvaraja, “Enhanced NLMS Adaptve Array via Direction-of-Arrival Detection,” *IET Communications,* vol. 1, no. 1, pp. 19-26, 2007.

[8] J. Dhiman, S. Ahmed and K. Gulia, “Comparison Between Adaptive Filter Algorithms (LMS, NLMS and RLS),” *International Journal of Science, Engineering and Technological Research,* vol. 2, no. 8, May 2013.

[9] A. Sharma and S. Mathur, “Performance Analysis of of Adaptive Array Signal Processing Algorithms,” *IETE Technical Review,* vol. 33, no. 5, September 2016.

[10] J. Treichler and B. Agee, “A New Approach to Multipath Correction of Constant Modulus Signals,” *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 31, no. 2, pp. 459-472, 1983.

[11] Y. Chen, T. Le-Ngoc, B. Champagne and C. Xu, “Recursive Least Square Constant Modulus Algorithm for Blind Adaptive Array,” *IEEE Transactions on Signal Processing,* vol. 52, no. 5, May 2004.

[12] B. Agee, “The Least-Squares CMA, A New Approach to Multipath Correction of Constant Modulus Signals,” in *IEEE International Conference on ICASSP' 83*, April 1983.

[13] A. Aziz, M. A. Qureshi, M. J. Iqbal, S. Zeeshan, U. Farooq, U. Ahmad and A. Zaidi, “Performance and Quality Analysis of Adaptive Beamforming Algorithms (LMS, CMA, RLS and CGM) for Smart Antennas,” in *3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)*, 2010.

[14] B. S. Yashoda and K. R. Nataraj, “Performance Analysis of Excisting Beamforming methods for Various Antenna Elements and Interference Sources,” *International Journal of Computer Engineering In Research Trends,* vol. 4, no. 5, pp. 142-149, April 2017.