INTERNATIONAL JOURNAL OF ELECTRONICS AND COMMUNICATION ENGINEERING & TECHNOLOGY (IJECET)

ISSN 0976 – 6464(Print)
ISSN 0976 – 6472(Online)
Volume 5, Issue 4, April (2014), pp. 25-35
© IAEME: www.iaeme.com/ijecet.asp
Journal Impact Factor (2014): 7.2836 (Calculated by GISI)
www.jifactor.com

IJECET

©IAEME

INVESTIGATION OF ADAPTIVE BEAMFORMING ALGORITHMS FOR COGNITIVE RADIO ARCHITECTURE

Niranjana.R[1], N.K.Neelakantan[2], P.M.Smriti Menon[3], Saranya Ramanath[4], Vidya.K.A[5], Dr. Sudha.T[6]

[1, 2, 3, 4, 5]Students, [6]Professor, Department of Electronics and Communication Engineering, NSS College of Engineering, Palakkad – 678008, India

ABSTRACT

The present fixed spectrum assignment policy becomes a bottleneck for more efficient spectrum utilization, as majority of the already scarce spectrum goes under-utilized. The concept of Cognitive Radio has been receiving increasing attention as a solution to the aforementioned quandary, since it equips the capability to optimally adapt their operating parameters according to the interactions with the surrounding radio environment. In this paper, the feasibility of a smart antenna to this system is analyzed and the performance of the different beamforming algorithms is compared.

Keywords: Adaptive Beamforming Algorithms, Cognitive Radio, Smart Antennas.

1. INTRODUCTION

With the growth of wireless communication in leaps and bounds, the demand for radio spectrum has seen a never before increase in demand – everyone wants a piece of it. Understandably, with the upward escalation of the number of mobile users, the already limited licensed frequency spectrum is getting crowded. On the other hand, the licensed spectrum bands are under-utilized due to the current inflexible spectrum allocation policy [1]. This point of view is supported by the recent studies of the FCC, which reveal that in some locations or at some times of day, 70% of the allocated spectrum remains idle, which is alarming [2].

In order to solve this conundrum, Cognitive Radio (CR) has been recently proposed as the solution to current low usage of licensed spectrum problem. A cognitive radio allows a cognitive user to access a spectrum hole unoccupied by a primary user and improve the spectrum utilization while reducing the white spaces in the spectrum [1]. Although CR has many advantages, it is plagued with some demerits like complexity and interference [1]. This work aims at eliminating the interference problem of CR.

Specifically, this paper emphasizes the following objectives:

1. To study the behavior of smart antennas and propose them for CR technology
2. To investigate the performances of the different beam forming algorithms like Least Mean Square (LMS), Sample Matrix Inversion (SMI), Recursive Least Square (RLS), Constant Modulus (CMA) and Least Square Constant Modulus (LSCMA) and to determine the best one by comparing them on the basis of computational complexity and radiation patterns.

2. SMART ANTENNAS

The term 'Smart Antennas' refers to any antenna array, terminated in a sophisticated digital signal processor, which can adjust its own beam pattern in order to emphasize signals of interest and to minimize interfering signals [3]. The smart antenna incorporates smartness to a communication system by electronically steering the main beam in a desired direction [4]. As opposed to a 'dumb' switched beam antenna, a smart antenna has the following advantages [3]:

1. It can track multiple targets.
2. It can produce low side lobes.
3. It helps in diminishing co-channel interference to an acceptable level.
4. It increases both range and capacity of the communication system.
5. Using smart antennas results in a significant decrease in the required transmitter power and eliminates multipath effects too.

Smart Antennas are compatible with multiple access schemes like FDMA, TDMA, SDMA and so on.

Since CR is a wireless communication technology, the presence of an antenna is inevitable to establish the wireless link [1]. Also, the elimination of interference between users solely depends on the type of antenna used as the antenna is the one which directs the transmitters' signals into space. Therefore, a smart antenna proves to be an ideal choice for CR architecture as they have the capability to direct beams in different directions [3]. Adding some intelligence to the antenna array aids in keeping a track on the location of different users in a wireless environment. Because of this, as well as the above mentioned merits, smart antennas can be used in CR to mitigate interference [3]. The output of the digital beamformer shown in is Fig 1 at the k^{th} sample is given by:

$$\boldsymbol{y(k) = \sum_{n=1}^{N} W_n^* x_n(k)} \quad \textbf{(1)}$$

$$\boldsymbol{y(k) = \overline{W_H}(k).\overline{X(k)}} \quad \textbf{(2)}$$

where W_n is the weight applied to the n^{th} element.

$x_n(k)$ is the k^{th} sample of the signal received at the n^{th} element.

$y(k)$ is the output of the beamformer corresponding to the k^{th} sampling instant.

The objective of the adaptive element of the smart antenna array system is to find the set of weights w_n such that a very strong beam is produced in the desired direction and nulls are produced in the direction of the interferers/unintended users [3].

If there are M number of users in a particular system, then the total received signal is the sum of $\overline{X_k}$(t), which is the received signal vector of the k^{th} user, and $n(t)$, which represents the noise due to the receiver as well as the background channel noise [4], i.e.,

$$\bar{\boldsymbol{X}}(\boldsymbol{t}) = \sum_{k=1}^{M} \overline{\boldsymbol{X}_k}(\boldsymbol{t}) + \boldsymbol{n}(\boldsymbol{t}) \quad \textbf{(3)}$$

In matrix form, this can be written as:

$$\bar{X}(t) = A\bar{S}(t) + n(t) \quad \textbf{(4)}$$

where,

$A = [a(\theta_1) \quad a(\theta_2) \quad \cdots \quad a(\theta_M)]$

$$\bar{S}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_M(t) \end{bmatrix}$$

$a(\theta_k)$ is the array response vector or array steering vector in the direction of arrival θ_k of the k^{th} user and $s_k(t)$ is the complex baseband signal incident upon the array from the k^{th} user.

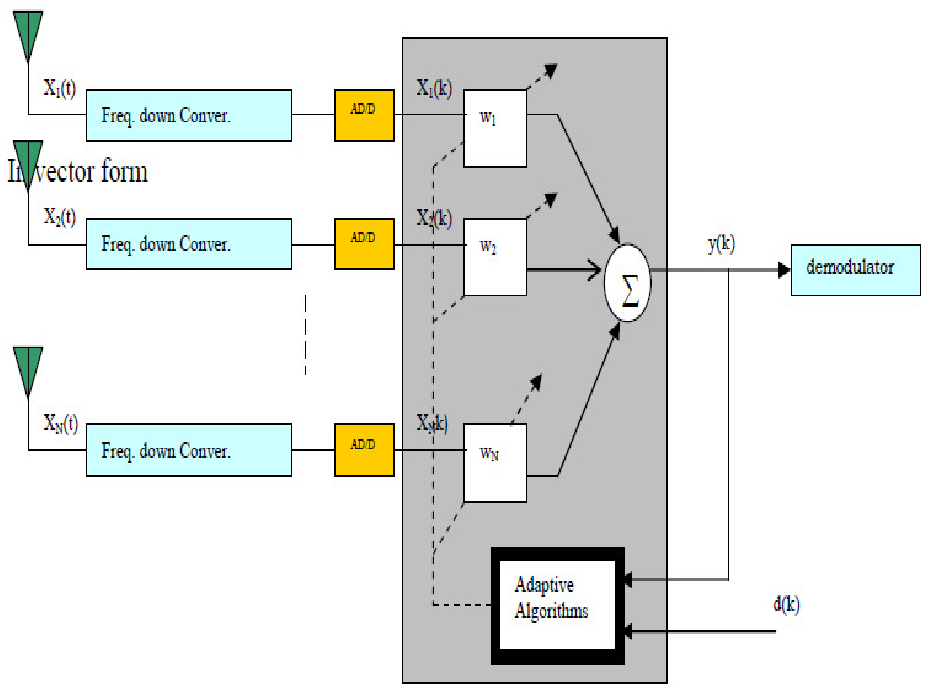

Fig.1: Block diagram of an Adaptive Smart Antenna Array

3. ADAPTIVE BEAMFORMING ALGORITHMS

In Adaptive beamforming, array weights are changed dynamically based on the dynamically changing environments so as to make optimum beam to the direction of the intended user and put nulls to the direction of the interferers/noise. This phenomenon is accomplished by using adaptive beamforming algorithms. There are two major classes of adaptive beamforming algorithms based on their requirements for training signal sequence: Non-Blind and Blind Adaptive Algorithms [4].

Non-Blind Adaptive Algorithms requires statistical knowledge of the transmitted signal in order to optimize the array weights. In other words, to extract the desired user(s) from the surrounding environment (received signals) a training signal sequences which are known both at the receiver and transmitter are transmitted. Then based on the information obtained from the received signal about the channel the array weights are optimized (adjusted) to reduce the error between the received signals sequences and the known transmitted signal sequences at the receiver. Sample Matrix Inversion (SMI), Least Mean Square (LMS) and Recursive Least Square (RLS) are non-blind algorithms.

Unlike non-blind adaptive algorithms, blind algorithms do not require training signal sequences rather they try to estimate information from the received signal. Constant Modulus (CM) and Least Square Constant Modulus (LSCM) are blind adaptive beamforming algorithms.

3.1. Least Mean Square (LMS) Algorithm

Least Mean Square (LMS) algorithm is a non-blind adaptive beamforming algorithm which searches for optimal weights that would minimize the Mean Square Error(MSE). Through a feedback loop, the weights $w_1, w_2 ..., w_n$ are updated by the adaptive processor that has the time sampled error signal, *e(k)* as input, given by:

$$e(k) = d(k) - y(k)$$

or

$$e(k) = d(k) - \bar{w}^H(k)\bar{x}(k) \tag{5}$$

where, *d(k)* is the training sequence, *y(k)* is the output of the adaptive beamformer, $\bar{w}^H(k)$ is the hermitian of the weight matrix and $\bar{x}(k)$ is the input signal vector to the array.

Since signal statistics are not known, estimate of the array correlation matrix, $\bar{R}_{xx}$ and the signal correlation vector, $\bar{r}$ over a range of snapshots or for each time instant, are used by the LMS algorithm [4]. The instantaneous estimates of these values are given as:

$$\hat{R}_{xx}(k) = \bar{x}(k)\bar{x}^H(k) \tag{6}$$

and

$$\hat{r}(k) = d^*(k)\bar{x}(k) \tag{7}$$

The above equation of MSE is called the *performance surface* or *cost function*, $J(\bar{w})$ which forms a quadratic surface in the *M* dimensional space [4]. Since the optimum weights must provide the minimum MSE, the extremum is the minimum of the cost function [5]. The minimum value is obtained by taking the gradient of cost function with respect to the weight vectors and equating it to zero. Solution for the weights is called *optimum Weiner solution* and is given by:

$$\bar{w}_{opt} = \bar{R}_{xx}^{-1}\bar{r} \tag{8}$$

An iterative technique, called the *method of steepest descent* is employed at this stage to approximate the gradient of the cost function in terms of the weights using LMS method [4]. The iterative expression for calculating the optimum weights by LMS Algorithm is given as:

$$\bar{w}(k+1) = \bar{w}(k) + \mu e^*(k)\bar{x}(k) \tag{9}$$

where, μ is the step size parameter. Many iterations are required for the LMS algorithm for convergence of the weight vector. Hence, LMS algorithm converges slowly [6].

3.2. Sample Matrix Inversion (SMI) Algorithm

Sample Matrix Inversion (SMI) algorithm is a non-blind adaptive beamforming algorithm which uses Minimum Mean Squared Error (MMSE) criterion to obtain the optimal array weight vector. This method tries to overcome the slow convergence of LMS algorithm by using direct matrix inversion [4]. The sample matrix is the array correlation matrix, $\hat{R}_{xx}$, defined as:

$$\hat{R}_{xx} = \frac{1}{K}\sum_{k=1}^{K}\bar{x}(k)\bar{x}^{H}(k) \tag{10}$$

and the correlation vector, $\hat{r}$, defined as:

$$\hat{r} = \frac{1}{K}\sum_{k=1}^{K} d^{*}(k)\bar{x}(k) \tag{11}$$

For implementation of the algorithm, $\hat{R}_{xx}$ and $\hat{r}$ are calculated using:

$$\bar{X}_K(k) = \begin{bmatrix} x_1(1+kK) & x_1(2+kK) & \cdots & x_1(K+kK) \\ x_2(1+kK) & x_2(2+kK) & \cdots & x_2(K+kK) \\ \vdots & & & \\ x_M(1+kK) & x_M(2+kK) & \ldots & x_M(K+kK) \end{bmatrix} \tag{12}$$

where k is the block number and K is the block length. The matrix defines $\bar{X}_K(k)$ as the k^{th} block of $\bar{x}$ vectors ranging over K data snapshots. The desired signal vector can be defined by:

$$\bar{d}(k) = [\ d(1+kK) \quad d(2+kK) \quad \cdots \quad d(K+kK)\ ] \tag{13}$$

The estimate of the array correlation matrix and correlation vector respectively is given by:

$$\hat{R}_{xx}(k) = \frac{1}{K}\bar{X}_K(k)\bar{X}_K^{H}(k) \tag{14}$$

$$\hat{r}(k) = \frac{1}{K} d^{*}(k)\bar{X}_K(k) \tag{15}$$

Finally, the optimum weights can be calculated for the k^{th} block of length K as:

$$\bar{w}_{SMI}(k) = \hat{R}_{xx}^{-1}(k)\hat{r}(k) \tag{16}$$

SMI Algorithm, although converges quickly, requires large number of multiplications and additions per iteration. Hence, SMI is found to be computationally complex.

3.3. Recursive Least Square (RLS) Algorithm

Recursive Least Square (RLS) Algorithm is a non-blind adaptive beamforming algorithm that uses the Least Square criterion for optimization. This algorithm tries to achieve the faster convergence of SMI while overcoming its computational complexity and potential singularities by computing the array correlation matrix and correlation vector recursively [4]. The correlation matrix and the correlation vector used in SMI algorithm uses rectangular windows of equal width, which considers all previous time samples equally [4]. Since the signal sources can change or slowly move with time, RLS deemphasizes the earlier data samples and emphasize the most recent ones to compute the weighed estimate of $\hat{R}_{xx}(k)$ and $\hat{r}(k)$ [4]. This reduces the computational complexity. The weighted estimates are calculated by the iterative expressions given by:

$$\hat{R}_{xx}(k) = \alpha\hat{R}_{xx}(k-1) + \bar{x}(k)\bar{x}^H(k) \quad (17)$$

$$\hat{r}(k) = \alpha\bar{r}(k-1) + d^*(k)\bar{x}(k) \quad (18)$$

where, α is the forgetting factor such that $0 \le \alpha \le 1$.

Sherman-Morrison-Woodbury (SMW) equation is used to compute the inverse of $\hat{R}_{xx}(k)$, i.e.

$$\hat{R}_{xx}^{-1}(k) = \alpha^{-1}\hat{R}_{xx}^{-1}(k-1) - \frac{\alpha^{-2}\hat{R}_{xx}^{-1}(k-1)\bar{x}(k)\bar{x}^H(k)\hat{R}_{xx}^{-1}(k-1)}{1+\alpha^{-1}\bar{x}^H(k)\hat{R}_{xx}^{-1}(k-1)\bar{x}(k)} \quad (19)$$

The optimum weights are given by the iterative equation:

$$\bar{w}(k) = \bar{w}(k-1) + \bar{g}(k)[d^*(k) - \bar{x}^H(k)\bar{w}(k-1)] \quad (20)$$

where, the gain vector $\bar{g}(k)$ is defined as:

$$\bar{g}(k) = \frac{\alpha^{-1}\hat{R}_{xx}^{-1}(k-1)\bar{x}(k)}{1+\alpha^{-1}\bar{x}^H(k)\hat{R}_{xx}^{-1}(k-1)\bar{x}(k)} \quad (21)$$

3.4. Constant Modulus Algorithm (CMA)

Constant Modulus Algorithm (CMA) is a blind adaptive beamforming algorithm, which uses the constant modulus property of the transmitted signal to make an optimum beam to the intended direction instead of using a training signal sequence. The receiver restores the envelope of the transmitted signal by equating the received signal to some constant value that corresponds to the envelope of the transmitted signal. This is made possible by continuously updating the weight of the beamformer until the output of the array has the same modulus as that of the original transmitted signal [4]. The cost function J*(k)* used for CMA is given by:

$$J(k) = E[\,|y(k)|^p - |\alpha|^q] \quad (22)$$

where p = 1, 2 or q = 1, 2 and α is the desired signal amplitude at the output of the array. Assuming that $|\alpha|$ = 1, the above equation becomes:

$$J(k) = E[\,|y(k)|^p - 1] \quad (23)$$

The cost function is a positive measure of the average amount of deviation of the output of beamformer, *y(k),* from the unit modulus condition [4]. To choose the weight vector recursively that minimizes *J* and consequently makes *y(k)* as close to a constant modulus signal as possible, an iterative method is used, which is based on the steepest descent method applied in LMS Algorithm [4]. The LMS equation thus becomes:

$$\bar{w}(k+1) = \bar{w}(k) - \frac{1}{2}\mu\nabla(J) \quad (24)$$

where, $\nabla(J)$ is the gradient of the cost function of CMA.

The weights are calculated by:

$$\bar{w}(k+1) = \bar{w}(k) - \mu\bar{X}(k)e^*(k) \quad \textbf{(25)}$$

The above equation is used in CMA for determining the array weights dynamically. The step size is selected as $\mu \ll 1$ for stability [4]. The convergence rate can also be controlled by μ as in LMS algorithm. However, for better convergence behavior, non-linear least square method needs to be used.

3.5. Least Square Constant Modulus Algorithm (LSCMA)

The least-squares method, also known as the Gauss method or LS-CMA algorithm, is a blind algorithm [7]. It is also known as an autoregressive estimator based on a least squares minimization. In the method of least squares, first a cost function is defined as the weighed sum of error squares or the total error energy [7]. The energies are energies of a finite sample set K. The cost function is defined by:

$$C(\bar{w}) = \sum_{k=1}^{K} |\phi_k(\bar{w})|^2 = \|\phi(\bar{w})\|_2^2 \quad \textbf{(26)}$$

The optimum weights are calculated by the iterative expression,

$$\bar{w}(n+1) = [\bar{X}\bar{X}^H]^{-1}\bar{X}\bar{r}^* \quad \textbf{(27)}$$

where,

$$\bar{r}^*(n) = \left[\frac{\bar{w}^H(n)\bar{x}(1)}{|\bar{w}^H(n)\bar{x}(1)|} \;\; \frac{\bar{w}^H(n)\bar{x}(2)}{|\bar{w}^H(n)\bar{x}(2)|} \;\; \cdots \;\; \frac{\bar{w}^H(n)\bar{x}(K)}{|\bar{w}^H(n)\bar{x}(K)|} \right]^H \quad \textbf{(28)}$$

4. RESULTS AND DISCUSSION

Matlab Simulation of various adaptive beam-forming algorithms was performed. A ten element antenna array is simulated and it is assumed that the Primary user is located at 45^0 and interferer at -10^0. The rectangular as well as the polar plots of the radiation pattern corresponding to each algorithm were plotted for comparison. From the radiation pattern, the strength of the side lobe along the direction of the intended user and the ability of the algorithm to produce a null along the direction of the interferer were studied. The normalized array factor was plotted in the rectangular plot.

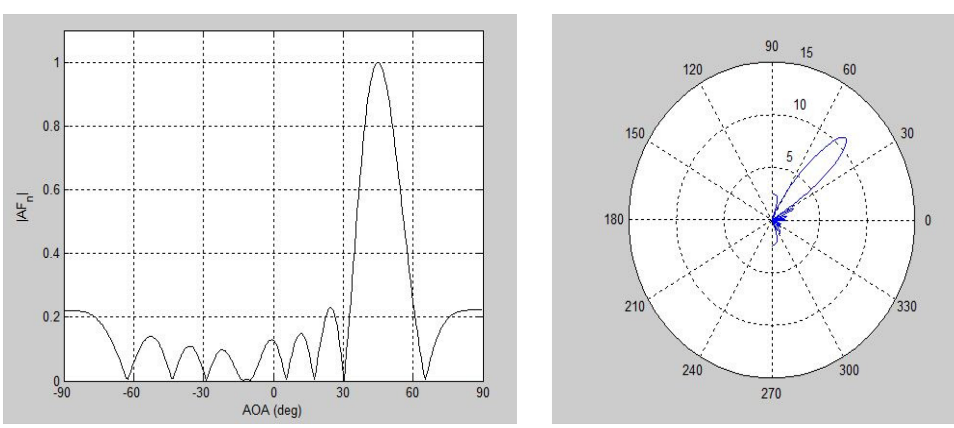

Fig.2: Rectangular plot and Polar plot for LMS algorithm

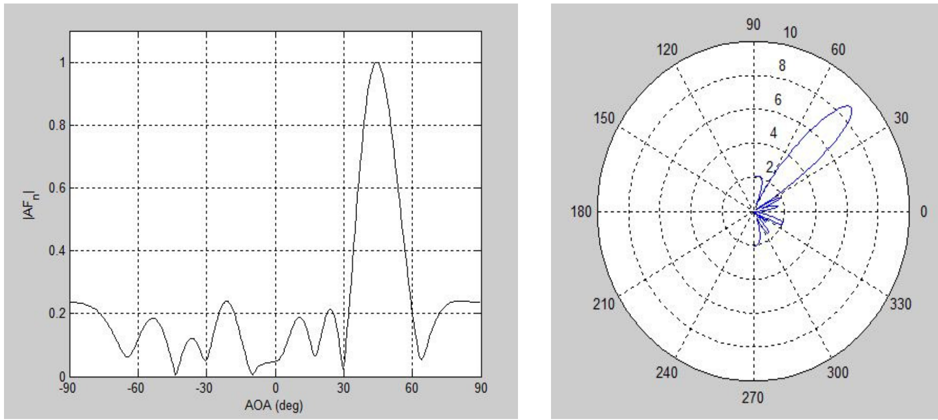

Fig.3: Rectangular plot and Polar plot for SMI algorithm

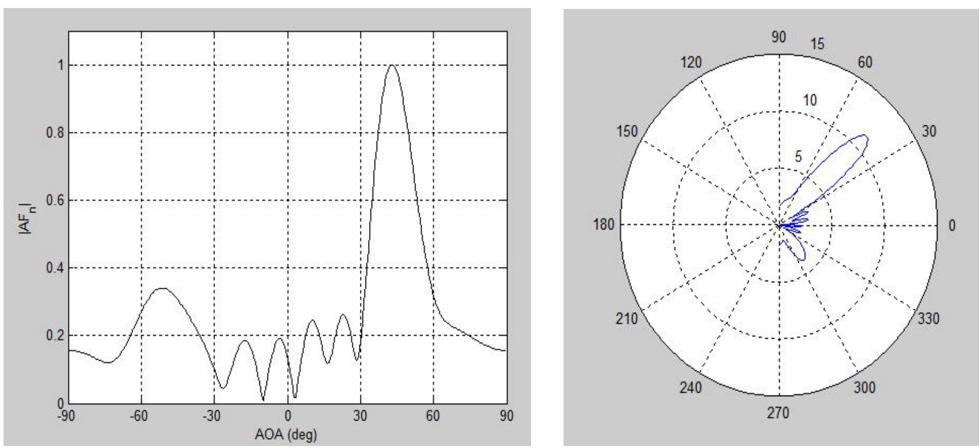

Fig.4: Rectangular plot and Polar plot for RLS algorithm

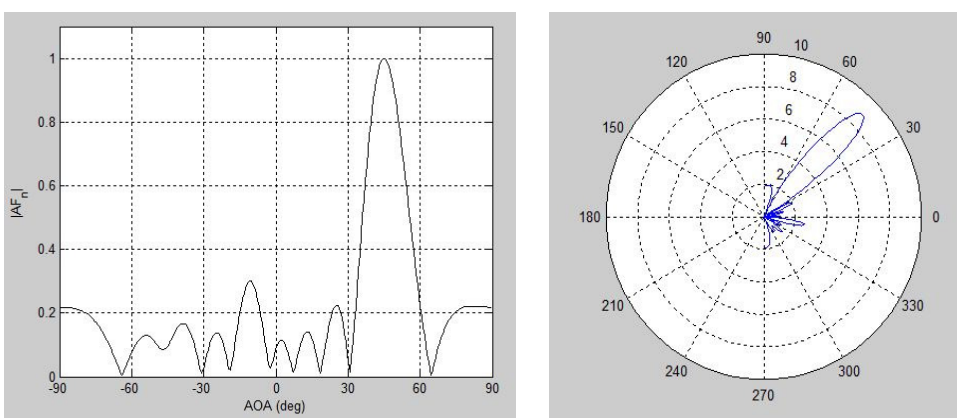

Fig.5: Rectangular plot and Polar plot for CM algorithm

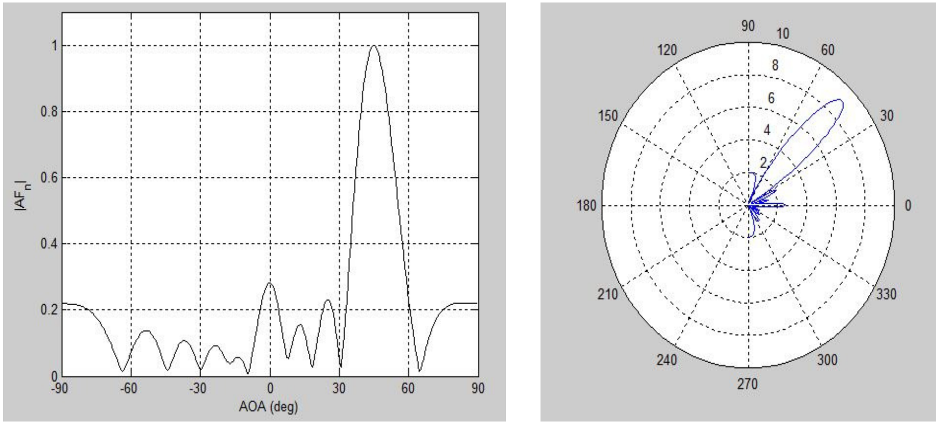

Fig.6: Rectangular plot and Polar plot for LSCM algorithm

4.1. Comparison based on Computational Complexity

In Tables 1-5 corresponding to the various Adaptive Beamforming Algorithms, the number of multiplications and additions required per iteration is expressed for an adaptive beam-forming system with N array elements and length of observable data, K. The mathematical operations needed for procedures such as Auto-correlation Matrix Estimation, Cross-correlation Vector Estimation, Estimation of Weights etc. are considered.

Table 1: Computational Complexity of SMI algorithm

Procedure	Multiplication per iteration	Addition per iteration
Auto-correlation matrix estimation	KN + 1	K + N
Cross-correlation vector estimation	$K^2N + 1$	K
Estimation of Weights	N^2	N^2
Total Operation	$K^2N + KN + N^2 + 2$ + Matrix Inversion	$N^2 + 2K + N$

Table 2: Computational Complexity of LMS algorithm

Procedure	Multiplication per iteration	Addition per iteration
Calculation of Array output signal, y(k)	N	N
Error of Adaptive System	-	1
Array weight optimization by Steepest Descent method	N + 1	N + 1
Total Operation	2N + 1	2N + 2

Table 3: Computational Complexity of RLS algorithm

Procedure	Multiplication per iteration	Addition per iteration
Calculation of Gain vector	$2N^2 + 3N + 1$	$2N^2 + 2N + 1$
Error of Adaptive System	N	N + 1
Array weight optimization	N	N
Calculation of Inverse of Correlation Matrix	$N^2 + 2N + 1$	$N^2 + N + 1$
Total Operation	$3N^2 + 7N + 2$	$3N^2 + 5N + 3$

Table 4: Computational Complexity of CM algorithm

Procedure	Multiplication per iteration	Addition per iteration
Calculation of Beamformer output	N	N
Error of Adaptive System	1	1
Array weight optimization	N + 1	N
Total Operation	2N + 2	2N + 1

Table 5: Computational Complexity of LSCM algorithm

Procedure	Multiplication per iteration	Addition per iteration
Calculation of Output Data Vector	NK	NK
Calculation of Complex-limited Output Data Vector	K	-
Array weight optimization	$N^2K + N^2 + NK$ + Inversion	$N^2K + N^2 + NK$
Total Operation	$N^2K + N^2 + 2NK + K$ + Inversion	$N^2K + N^2 + 2NK$

From the above results, the following order of algorithms in the decreasing order of computational complexity can be established:

SMI > LSCMA > RLS > LMS > CMA

4.2. Comparison based on Half Power Beam Width (HPBW) and Side Lobe Level (SLL)

In Table 6, the simulated values of the Side Lobe Level, SLL (dB) and the Half Power Beam Width, HPBW (^{o}) as computed from Fig.2-6 are tabulated. From the results, it is clear that, LS-CMA has the best SLL ratio and RLS, the worst. The HPBW of SMI algorithm is 14.53^o which makes it best suited for application in Cognitive Radio Technology.

Table 6: Comparison of algorithms based on SLL (dB) and HPBW (^{o})

Algorithm	SLL (dB)	HPBW (^{o})
SMI	-6.2069	14.53
LMS	-6.4111	14.71
RLS	-3.9675	16.77
CMA	-6.5365	14.56
LSCMA	-6.6827	14.58

5. CONCLUSION

The investigation of different adaptive beamforming algorithms for the cognitive radio technology from both blind and non-blind algorithms has shown that the Sample Matrix Inversion (SMI) from the non-blind beamforming family and the Constant Modulus Algorithm (CMA) from the blind beamforming family have better radiation patterns (beam pattern) that suit the cognitive radio technology. The others have low and dying side lobes and using them for detection in CR applications could result in the scanned RF giving wrong information (false alarm) about the white spaces in the frequency spectrum. When overall performance is considered, the SMI is preferred for CR applications as compared to the other adaptive beamforming algorithms. The biggest advantage that SMI has to offer is that it has very fast convergence rate of all adaptive beamforming algorithms studied in this work which is a benediction in a CR system.

REFERENCES

[1] Simon Haykin, *'Cognitive Radio: Brain Empowered Wireless Communications'*, IEEE Journal On Selected Areas In Communications, Vol. 23, No. 2, February 2005.

[2] FCC Report and Order, ET Docket No.03-108, *'Facilitating Opportunities for Flexible, Efficient and Reliable Spectrum Use Employing Cognitive Radio Technology'*, March 11, 2005.

[3] Ioan Nicolaescu, Dan Stoica, *'Smart Antennas for Wireless Communication Systems'*, Vol.25, IEEE Explore 2011.

[4] Frank Gross, *'Smart Antennas for Wireless Communications with MATLAB'*, McGraw Hill Publications, 2005 (Textbook).

[5] C. G. Christodoulou, M. Georgiopoulos, *'Smart Adaptive Array Antennas for Wireless Communications'*, Proceedings of SPIE Vol. 4395 (2001) SPIE - 0277-786X/01.

[6] I.S. Reed, J.D. Mallett, L.E. Brennan, *'Rapid Convergence Rate in Adaptive Arrays'*, IEEE Transactions On Aerospace And Electronic Systems Vol. Aes-10, No. 6 November 1974.

[7] Thomas E. Biedka, William H. Tranter, Jeffrey H. Reed, *'Convergence Analysis Of The Least Squares Constant Modulus Algorithm In Interference Cancellation Applications'*, IEEE Transactions On Communications, Vol. 48, No. 3, March 2000.

[8] Dr. H.V. Kumaraswamy and Vijay B.T, '*Efficient Beamforming Algorithm for MIMO Multicast with Application Layer Coding*', International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 4, Issue 2, 2013, pp. 116 - 128, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.