World Applied Sciences Journal 10 (Special Issue of Computer & Electrical Engineering): 01-09, 2010
ISSN 1818-4952

Tracking Performance of RLS and KAPA Algorithms for a Smart Antenna System

M. Yasin, Pervez Akhtar and M. Junaid Khan

Department of Electronics and Power Engineering,
National University of Sciences and Technology, Islamabad, Pakistan

Abstract: Radio frequency spectrum is always scarce and is considered to be an expensive resource to the users. Its efficient utilization is only possible by using smart/adaptive antenna array systems to exploit capabilities of wireless systems for optimization of service quality and capacity. Smart antenna radiates not only narrow beam towards desired users exploiting the signal processing capability but also places null towards interferers, thus optimizing the signal quality. Recursive least squares (RLS) and Kernel Affine Projection Algorithm (KAPA) are two adaptive algorithms presented in this paper for beamforming and mean square error (MSE) estimation. For the first time, KAPA is presented for noise cancellation but we are using it for adaptive beamforming which is novel in this application. The efficiency of RLS and KAPA is compared on the basis of normalized array factor and MSE for mobile communication. Simulation results verify that KAPA has high resolution for array factor taking small number of samples as compared to RLS. However RLS demonstrates good performance to minimize MSE than KAPA. Therefore, RLS is found the more efficient algorithm to minimize MSE whereas KAPA is found better for beam and null formation.

Key words: Smart Antenna · Adaptive Filtering · Recursive Least Square (RLS) Algorithm and Kernel Affine Projection Algorithm (KAPA)

INTRODUCTION

A comprehensive detail regarding antenna array, its performance, beamforming schemes, its effects on errors, capacity optimization, signal to noise (SNR) improvement, source separation from noise and performance analysis of adaptive algorithms is discussed in [1-3]. The comparison is made between MUltiple SIgnal Classification (MUSIC) and the Estimation of Signal Parameter via Rotational Invariance Techniques (ESPRIT) in [4] for smart antenna. In [5], a performance comparison of two non blind algorithms least mean square (LMS), recursive least square (RLS) and one blind constant modulus algorithm (CMA) is presented. A study regarding issues in smart antenna and its development is conducted in [6, 7] whereas echo cancellation is discuss in [8]. In our research work described in [9-11], a lot of improvement have been shown w.r.t referred papers either employing blind or non blind algorithms like LMS, normalized least mean square (NLMS), CMA, Minimum Variance Distortionless Response (MVDR) etc, respectively. In this paper, KAPA is presented for the first time for adaptive beamforming as discussed earlier and its performance comparison is made with RLS for smart antenna. The radio frequency (RF) spectrum is limited [1], therefore its efficient management is only possible by employing smart/adaptive antenna array systems to exploit capabilities of wireless systems for data and voice communication. The name smart refers to the signal processing capability that forms a vital part of the adaptive antenna system. This controls the antenna pattern by updating a set of antenna weights. Smart antenna, supported by signal processing capability [11-14], points narrow beam towards desired users but at the same time introduces null towards interferers, thus optimizing the service quality and capacity. Consider a smart antenna system with *Ne* elements equally spaced (*d*) and user's signal arrives from an angle Φ as shown in Figure 1 [1].

Corresponding Author: M. Yasin, Department of Electronics and Power Engineering, National University of Sciences and Technology, Islamabad, Pakistan.
E-mail: hamzayaseenkhattak1@yahoo.com, pervez@pnec.edu.pk, contactjunaid@yahoo.com.

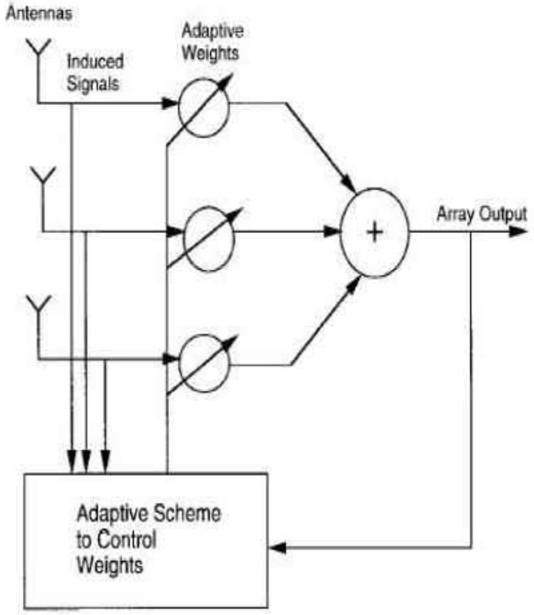

Fig. 1: Smart/adaptive antenna array system

The adaptive beamforming algorithm that is RLS and KAPA is used to control weights adaptively to optimize SNR of the desired signal in look direction Φ_0. The array factor for elements (Ne) equally spaced (d) linear array is given by:

$$AF(\Phi) = \sum_{n=0}^{N-1} A_n . e^{(jn(\frac{2\pi d}{\lambda}\cos\Phi + \alpha))} \tag{1}$$

Where α is the inter element phase shift and is described as:

$$\alpha = \frac{-2\pi d}{\lambda_0}\cos\Phi_0 \tag{2}$$

and Φ_0 is the desired direction of the beam.

When smart antenna is deployed in mobile communication environment using either time division multiple access (TDMA) or code division multiple access (CDMA) techniques. A separate time slot or code is assigned to each user using these techniques in a cell. Smart antenna then radiates beam towards desired users only. Each beam becomes a channel, thus avoiding interference in a cell. Because of these, each coded channel reduces co-channel interference, due to the processing gain of the system [1]. The processing gain (PG) of the CDMA system is described as:

$$PG = 10 \log (B / R_b) \tag{3}$$

Where B is the CDMA channel bandwidth and R_b is the information rate in bits per second.

If a single antenna is used for CDMA system, then this system supports a maximum of 31 users. When an array of five elements is employed instead of single antenna, then capacity of CDMA system can be increased more than four times. It can be further enhanced if array of more elements are used [1-15].

The Rest of the Paper Is Organized as Follows: Section 2 introduces RLS algorithm with simulation results. KAPA with simulation results are presented in section 3. Finally the concluding remarks of this work are provided in section 4.

RLS Algorithm

Introduction: RLS is an adaptive beamforming algorithm which finds the minimum mean square error (MSE) and thus yields the set of optimum weights of the array. The RLS algorithm has already been described in [16, 17] but in our case, the weight update equation is computed by:

$$w(n) = w(n-1) - R^{-1}(n)u(n)\,\varepsilon^*(w(n-1)) \tag{4}$$

Where $\varepsilon(n)$ denotes the error signal and is defined by:

$$\varepsilon(n) = d(n) - y(n) \tag{5}$$

Where $d(n)$ represents the desired signal consisting of original signal plus noise and $y(n)$ denotes the array output. The array output is defined by:

$$y(n) = w^H(n)u(n+1) \tag{6}$$

Where w and H refers to weight vector and complex conjugate transpose – called Hermitian transpose, respectively. $u(n)$ represents the signal array vector, written by:

$$u(n) = [u_1(n), u_2(n), \ldots\ldots, \mu_M(n)]^T \tag{7}$$

This signal array vector can also be written as:

$$u(n) = s_d(n)a(\theta_d) + \sum_{i=1}^{I} s_i(n)a(\theta_i) + N(n) \tag{8}$$

Where s_d & s_i are the desired and interfering signals arriving at the array at an angle θ_d & θ_i respectively. L denotes number of interfering signals and N is the noise at the array elements. $a(\theta_d)$ and $a(\theta_i)$ are the steering vectors for the desired and interfering signals respectively. The steering vector is described as:

$$a(\theta)=[1,e^{-j\phi},\dots\dots,e^{-j(M-1)\phi}] \tag{9}$$

Where $\phi=\frac{2\pi d}{\lambda}\sin\theta$ is the phase shift observed at each sensor due to the angle of arrival of the wavefront and assume d is the uniform distance between array elements. $\lambda=\frac{c}{f}$ where f is in Hertz. Therefore, the steering vector can be written as

$$a(\theta)=[1,e^{-j\frac{2\pi}{\lambda}d\sin(\theta)},\dots\dots,e^{-j\frac{2\pi}{\lambda}d(M-1)\sin(\theta)}] \tag{10}$$

The speed of convergence relies on gain matrix R^{-1} at the *nth* iteration producing weight $w(n)$ update equation as described in (4), where $R(n)$ is given by:

$$\begin{aligned}R(n)&=\delta_0 R(n-1)+u(n)*u^H(n)\\&=\sum_{n=0}^{k}\delta_0^{\,n-k}u(n)u^H(n)\end{aligned} \tag{11}$$

Where δ_0 is known as forgetting factor that represents a real scalar quantity which is small but close to one and is used for exponential weighting of the past data.

The RLS adaptive beamforming algorithm updates the required inverse of $R(n)$ using the previous inverse and the present sample as:

$$R^{-1}(n)=\frac{1}{\delta_0}\left[R^{-1}(n-1)-\frac{R^{-1}(n-1)u(n)*u^H(n)R^{-1}(n-1)}{\delta_0+u^H(n)R^{-1}(n-1)u(n)}\right] \tag{12}$$

The matrix R^{-1} is initialized as:

$$R^{-1}(0)=\frac{1}{\varepsilon_0}I,\varepsilon_0>0 \tag{13}$$

The RLS algorithm minimizes the cumulative square error as:

$$J(n)==\sum_{n=0}^{k}\delta_0^{\,n-k}\left|\varepsilon(n)\right|^2 \tag{14}$$

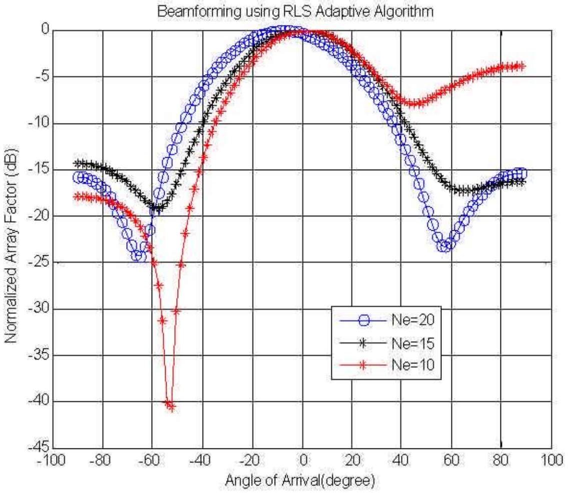

Fig. 2: Normalized array factor plot for RLS algorithm with AOA for desired user is 0 degree and -50 degrees for interferer for constant space of (λ / 20 between elements

Simulation Results: Computer simulation is carried out, to illustrate that how various parameters such as number of elements (*Ne*) and element spacing (*d*), affect the beam formation. The simulations are designed to analyze the properties of RLS. The desired signal is phase modulated with *SNR* = 20 dB, used for simulation purpose. It is given by:

$$S(t)=e^{-j\sin(2*\pi*f*t)} \tag{15}$$

Where f is the frequency in Hertz.

Effect of Number of Elements on Array Factor: Uniform linear array is taken with different number of elements for simulation purpose. The spacing between array elements is taken as (λ / 2).

Angle of Arrival (AOA) for desired user is set at 0 degree and for interferer at -50 degrees as shown in Figure 2 which provides null at -50 degrees.

AOA for desired user is set at 0 degree and provides null at -20 degrees as shown in Figure 3. For space (λ / 2), we got broad beam width as compared to d = λ/8 with reduced sidelobes. The results are summarized in Table 1 when element spacing is kept (λ / 8) as shown.

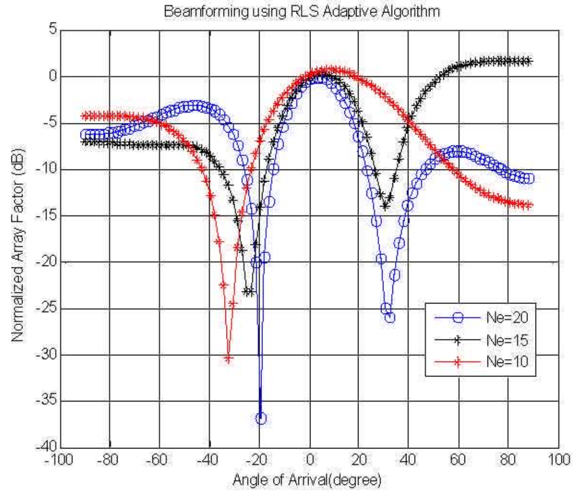

Fig. 3: Normalized array factor plot for RLS algorithm with AOA for desired user is 0 degree and -20 degrees for interferer with constant space of (λ / 8) between elements

Table 1: Effect of number of elements on beam width

N	d	Beam width (degree)
20	0.125	52
15	0.125	70
10	0.125	118

Table 2: Effect of element spacing on beam width

N	d	Beam width (degree)
6	0.5	38
6	0.25	76
6	0.125	102

Effect of Spacing Between Elements on Array Factor: The effect of array spacing for (λ / 2), (λ / 4) and (λ / 8) is shown in Figure 4 for *Ne* = 8.

When number of elements is reduced from 8 to 6, then effect of array spacing is shown in Figures 4 and 5, respectively. Again, narrower beam width is achieved at $d = \lambda/2$. The results are summarized in Table 2 as shown when number of elements is kept constant and elements spacing is different.

It is observed that increasing element spacing produces narrower beams, but this happens at the cost of increasing number of sidelobes. It is also clear, that spacing between elements equal to (λ / 2) gives optimum result for narrower beam.

Effect of Number of Elements on MSE: The effect of number of elements on MSE for constant space $d = \lambda/2$ and $d = \lambda/4$ between elements is shown in Figures 6 and 7, respectively, for same SNR as taken before (SNR = 20).

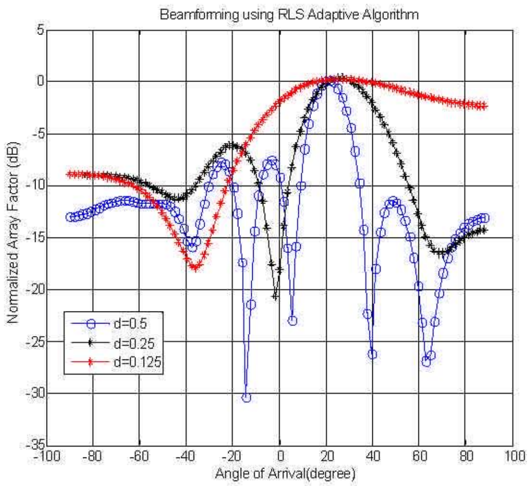

Fig. 4: Normalized array factor plot for RLS algorithm for *Ne* = 8 with interferer – 40 degrees

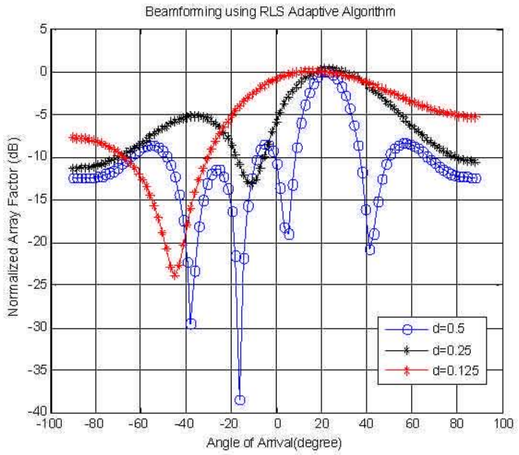

Fig. 5: Normalized array factor plot for RLS algorithm for *Ne* = 6 with interferer – 20 degrees

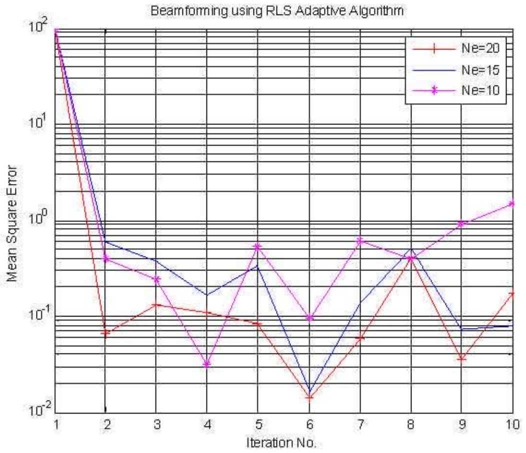

Fig. 6: Mean square error for RLS algorithm for *Ne* = 20, *Ne* = 15, *Ne* = 10 and space $d = \lambda/2$ is kept constant

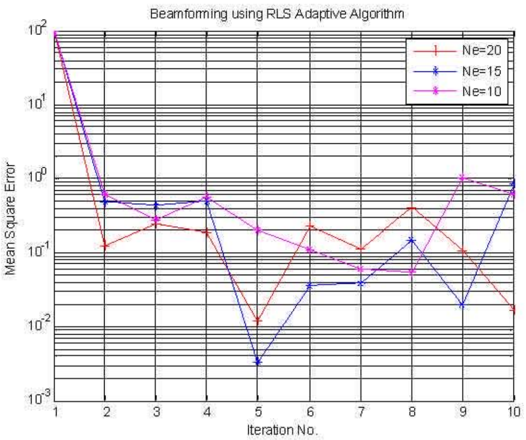

Fig. 7: Mean square error for RLS algorithm for *Ne* = 20, *Ne* = 15, *Ne* = 10 and space $d = \lambda/4$ is kept constant

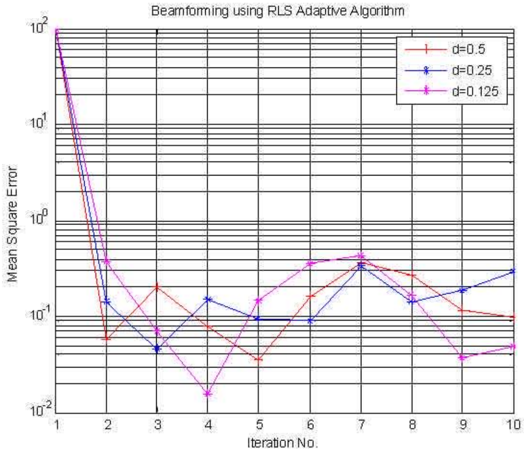

Fig. 8: Mean square error for RLS algorithm for $d = \lambda/2$, $d = \lambda/4$ and $d = \lambda/8$ and number of elements (*Ne* = 15) is kept constant

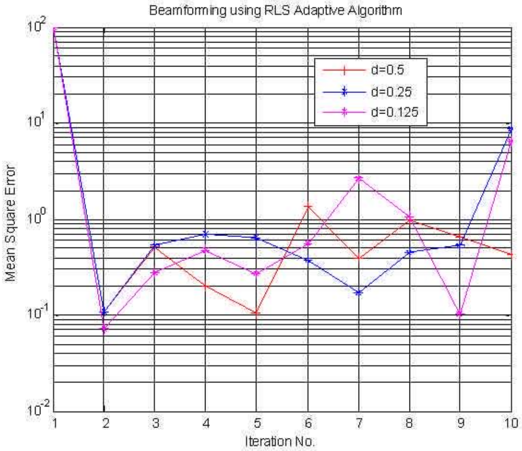

Fig. 9: Mean square error for RLS algorithm for $d = \lambda/2$, $d = \lambda/4$ and $d = \lambda/8$, for number of elements (*Ne* = 10) is kept constant

From these Figures, it is clear that minimum MSE is obtained for $d = \lambda/2$ when same number of elements is applied for comparison.

Effect of Spacing Between Elements on MSE: When space between elements is taken differently for same number of elements (*Ne* = 15,10) then MSE is further reduced as shown in Figures 8 and 9 respectively.

KAPA Algorithm

Introduction: For the first time, KAPA algorithm is presented in [18], for noise cancellation and providing a unifying model for several neural networks techniques. It is the combination of famed kernel trick and affine projection (APA) algorithm. In our case, this algorithm is employed for beamforming as well as for estimation of MSE. In KAPA algorithm, the input signal $u(n)$ is transformed into a high dimensional feature space via a positive definite kernel such that the inner product operation in the feature space can be computed efficiently through the kernel evaluation.

The weight $w(n)$ update equation for the KAPA algorithm is defined as:

$$w(n) = w(n-1) + \eta\varphi(n)\varepsilon(n)$$

$$= \sum_{n=1}^{k-1} a_n(k-1)\varphi(n) + \sum_{n=1}^{K} \eta\varepsilon_n(n)\varphi(n-1+K) \tag{16}$$

Where φ is an eigen function, ε is a regularization factor and η is the step size.

During the iteration, the weight vector in the feature space assumes the following expansion as:

$$w(n) == \sum_{n=1}^{k} a_n(k)\varphi(n)\forall_n > 0 \tag{17}$$

That is, the weight at time n is a linear combination of the previous transformed input.

The error signal is given by:

$$\varepsilon(n) = d(n) - \phi(n)w(n-1) \tag{18}$$

Where $d(n)$ is the desired/reference signal, also known as the prior knowledge of the signal of interest that is needed to improve accuracy and achieve faster convergence.

Simulation Results: The simulations are designed to analyze the properties of KAPA algorithms. The desired signal is phase modulated with *SNR* = 20 dB, used for simulation purpose as given in (15).

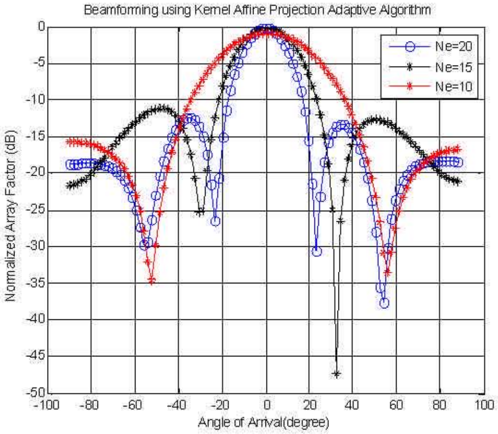

Fig. 10: Normalized array factor plot for KAPA algorithm with AOA for desired user is 0 degree and - 50 degrees for interferer with constant space of (λ/8) between elements

Table 3: Effect of number of elements on beam width

N	d	Beam width (degree)
20	0.125	42
15	0.125	62
10	0.125	114

Table 4: Effect of element spacing on beam width

N	d	Beam width (degree)
10	0.5	22
10	0.25	42
10	0.125	104

Effect of Number of Elements on Array Factor: Uniform linear array is taken for simulation purpose. AOA for desired user is set at 0 degrees and for interferer at -50 degrees as shown in Figure 10. The space (λ/8) is maintained between elements. The narrow beam with side lobes is observed for longer array. The results are summarized as shown in Table 3 when element spacing is kept (λ /8).

Now if number of elements is changed then a broad beam is obtained with reduced sidelobes as shown in Figure 11, for desired user at 20 degrees and for interferer is at -40 degrees.

Effect of Spacing Between Elements on Array Factor: When number of elements is kept constant for different array spacing i. e. $d = \lambda/2$, $d = \lambda/4$ and $d = \lambda/2$, then its effect is shown in Figures 12 and 13 for $Ne = 10$ and $Ne = 8$, respectively. The sharp beam is obtained for $Ne = 10$ for $d = \lambda/2$ as compared to $Ne = 8$. AOA for desired user is set at 0 and -50 degrees for interferer in Figure 12.

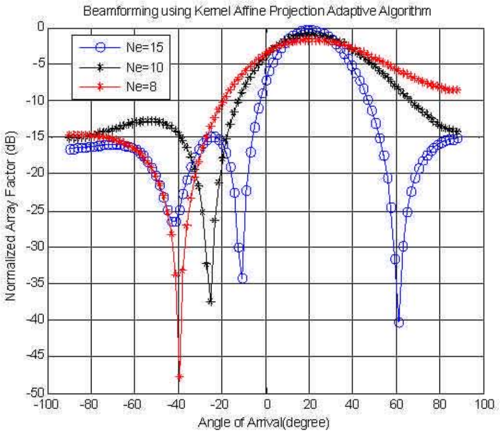

Fig. 11: Normalized array factor plot for KAPA algorithm with AOA for desired user is 20 degrees and - 40 degrees for interferer for constant space (λ /8) between elements

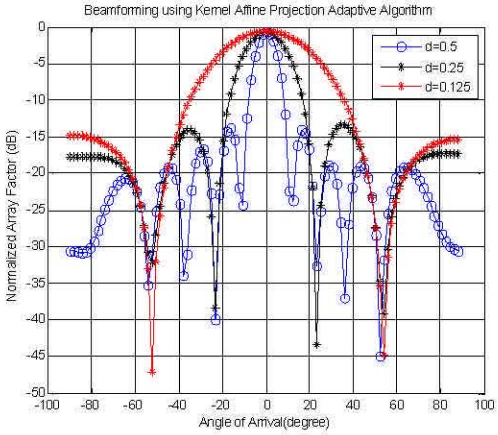

Fig. 12: Normalized array factor plot for KAPA algorithm for $Ne = 10$ with interferer – 50 degrees

The results are summarized as shown in Table 4 when number of elements is kept constant and elements spacing is different.

Similarly AOA for desired user is set at 0 and -30 degrees for interferer in Figure 13.

Effect of Number of Elements on MSE: The MSE is shown in Figure 14 for element spacing of (λ/ 8). Minimum MSE is obtained for $Ne = 20$, instead of $Ne = 15$. When elements spacing is kept as (λ/ 2) instead of (λ / 8), then no significant changes are observed in MSE as shown in Figure 15.

Effect of Spacing Between Elements on MSE: When different spacing between elements are kept for same number of elements as shown in Figures 16 and 17

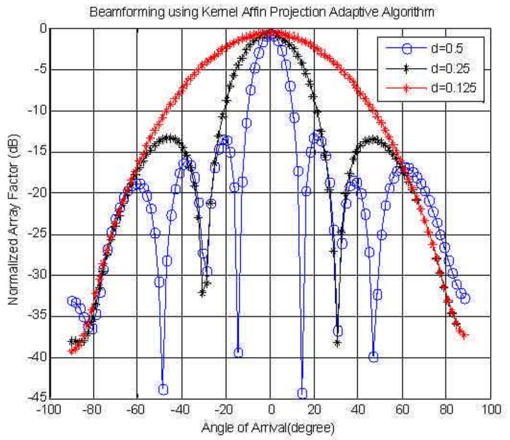

Fig. 13: Normalized Array factor plot for KAPA algorithm for *Ne* = 8 with interferer – 30 degrees

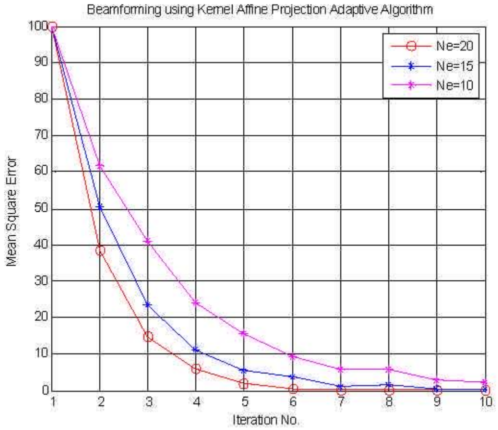

Fig. 14: Mean square error for KAPA algorithm for *Ne* = 20, *Ne* = 15, *Ne* = 10 and space $d = \lambda/8$ is kept constant

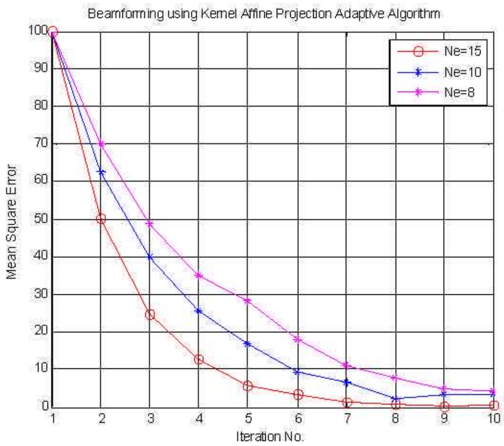

Fig. 15: Mean square error for KAPA algorithm for *Ne* = 15, *Ne* = 10, *Ne* = 8 and space $d = \lambda/2$ is kept constant

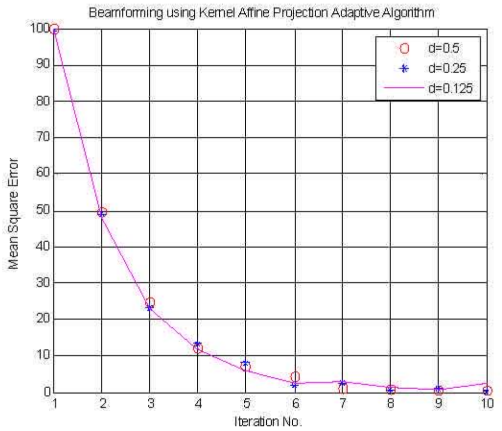

Fig. 16: Mean square error for KAPA algorithm $d = \lambda/2$, $d = \lambda/4$ and $d = \lambda/8$ for number of elements (*Ne* = 15) is kept constant

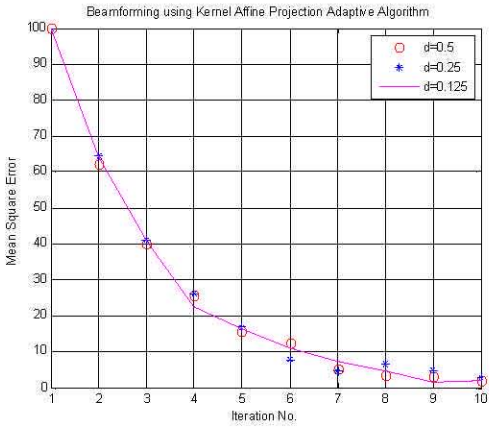

Fig. 17: Mean square error for KAPA algorithm for $d = \lambda/2$, $d = \lambda/4$ and $d = \lambda/8$ and number of elements (*Ne* = 10) is kept constant

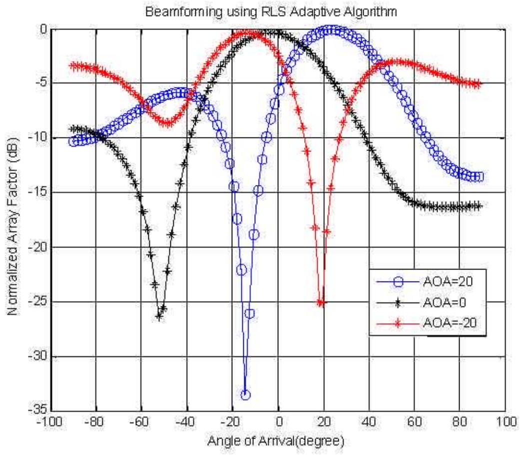

Fig. 18: Normalized array factor plot for RLS algorithm for *Ne* = 10

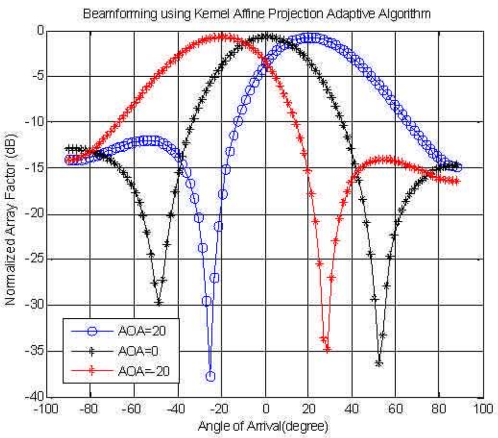

Fig. 19: Normalized array factor plot for KAPA algorithm for *Ne* = 10

for *Ne* = 15, 10 respectively, then it is clear from Figures that minimum MSE is obtained for large number of elements.

Comparison on the Basis of AOA: RLS and KAPA algorithms can also be compared on the basis of AOA as shown in Figures 18 and 19, respectively.

KAPA algorithm has shown best response for beamforming keeping ($\lambda/8$) spacing between elements.

CONCLUSION

In this paper, two adaptive beamforming algorithms RLS and KAPA are discussed. These algorithms are used in smart/adaptive antenna array system in coded form, to enhance mobile communication system performance. It is ascertained from the simulation results that RLS algorithm has shown good performance to minimize MSE taking different number of elements and for different spacing maintained between elements. However, KAPA algorithm has exercised reasonable performance to minimize MSE for same number of iteration. It is also confirmed from the simulation results that narrow beam of smart antenna can be steered towards the desired direction adaptively by changing weights $w(n)$, beam steering angle Φ_0 and number of elements (*Ne*) for both algorithms. However, KAPA algorithm has shown better response towards desired direction and has good capability to place null towards interferer than RLS. But the convergence speeds of both algorithms are approximately equal. Therefore, KAPA is found to be the most efficient algorithm and also simple in computation as compared to RLS. It is also confirmed from the comparison tables 1 to 4. KAPA is, therefore, a better option to implement at base station of mobile communication systems using CDMA environment to reduce interference, enhance capacity and service quality.

REFERENCES

1. LAL, C. Godara, 1997. Senior Member, IEEE, "Applications of Antenna Arrays to Mobile Communications, Part I; Performance Improvement, Feasibility and System Considerations, Proceeding of the IEEE, 85(7): 1031-1060.
2. Jalil Taghia and Mohammad Ali Doostari, 2009. Subband-based Single-channel Source Separation of Instantaneous Audio Mixtures, World Applied Sciences Journal 6 (6): 784-792.
3. Ziad, A.A. Alqadi, Musbah Aqel and Ibrahiem M.M. El Emary, 2008. Performance Analysis and Evaluation of Parallel Matrix Multiplication Algorithms, World Appl. Sci. J., 5(2): 211-214.
4. Md. Bakhar, Vani R.M. and P. V. Hunagund, 2009. Eigen Structure Based Direction of Arrival Estimation Algorithms for Smart Antenna Systems, IJCSNS International J. Computer Science and Network Security, 9(11): 96-100.
5. Shaukat, S.F., Mukhtar Ul Hassan, R. Farooq, H.U. Saeed and Z. Saleem, 2009. Sequetial Studies of Beamforming Algorithms for Smart Antenna Systems, World Appl. Sci. J., 6(6):754-758.
6. Rameshwar Kawitkar, 2008. Issues in Deploying Smart Antennas in Mobile Radio Networks, Proceedings of World Academy of Science, Engineering and Technol., 31: 361-366.
7. Fakoukakis, F.E., S.G. Diamantis, A.P. Orfanides and G.A. kyriacou, 2005. Development of an Adaptive and a Switched Beam Smart Antenna System for Wireless Communications, Progress in Electromagnetics Research Symposium, Hangzhou, China, pp: 1-5, 22-26.
8. Hun Choi and Hyeon-Deok Bae, 2006. Subband Affine Projection Algorithm for Acoustic Echo Cancellation System, EURASIP Journal on Advances in Signal Processing, Article ID 75621, 12: 18.
9. Yasin, M., Pervez Akhtar and Valiuddin, 2010. Performance Analysis of LMS and NLMS Algorithms for a Smart Antenna System, International J. Computer Applications, 4(9): 25-32.
10. Yasin, M., Pervez Akhtar and M. Junaid Khan, 2010. Affine Projection Adaptive Filter is a Better Noise Canceller, IST Transactions of Computer System – Theory and Applications, 1(2): 1-10.

11. Yasin, M. Pervez Akhtar and Junaid M. Khan, 2010. CMA an Optimum Beamformer for a Smart Antenna System, International Journal of Computer Applications, 5(7): 33-40.
12. LAL, C. Godara, 1997. Senior Member, IEEE, "Applications of Antenna Arrays to Mobile Communications, Part II; Beam-Forming and Directional of Arrival Considerations," Proceeding of the IEEE, 85(8): 1195-1245.
13. Frank Gross, 2005. Smart Antennas for Wireless Communications with MATLAB, McGraw-Hill, New York.
14. Prabhakar S. Naidu, 2001. Sensor Array Signal Processing, CRC Press, Washington D.C.
15. Yasin, M., Pervez Akhtar and M. Junaid Khan, 2010. MVDR an Optimum Beamformer for a Smart Antenna System in CDMA Environment, International J. Computer Science and Information Security, 8(4): 99-106, ISSN 1947-5500.
16. Simon Haykin, 1996. Adaptive Filter Theory, third edition (New Jersey, Prentice-Hall Inc.
17. Widrow, B. and S.D. Steams, 1985. Adaptive Signal Processing (New Jersey, Prentice-Hall Inc.
18. Weifeng Liu and Jose C. Principe, 2008. Kernel Affine Projection Algorithms, EURASIP Journal on Advances in Signal Processing, VOL. 2008, Article ID 784292, pp: 12.