CONSTRAINT ADAPTIVE NATURAL GRADIENT ALGORITHM (CANA) FOR ADAPTIVE ARRAY PROCESSING

I U Shah *T S Durrani, Fellow, IEEE*

Centre of Excellence in Signal and Image Processing
EEE Department, Royal College, Strathclyde University Glasgow, UK
Phone: 00441415482686, Fax: 00441415482684, email: ijteba@eee.strath.ac.uk, t.durrani@eee.strath.ac.uk

ABSTRACT

A novel Constraint Adaptive Natural Gradient Algorithm (CANA) is proposed based on the Natural Gradient technique, which is capable of rapidly adjusting the response of an array of sensors to a signal coming from Direction of Interest (DoI) and suppressing noises coming from other directions. Constrained optimization techniques have been extended to enhance the look direction signal in the presence of interference using adaptive natural gradient techniques. Mathematical analysis and realistic MATLAB simulations are developed to confirm the fast convergence of the adaptive weights, and the suitability of algorithm for operations in adverse environments.

1. INTRODUCTION

This paper introduces a novel algorithm for Adaptive Array Processing. Most often adaptive processing involves minimization of specific cost functions [1,2]. The practical minimization of these cost functions is not trivial as they are generally associated with manifolds with non linear optimization surfaces [3]. Working with search spaces that carry nonlinear manifolds introduces certain challenges in the algorithm implementation [3,4].Natural gradient has been found useful for optimization on non linear surfaces [5]. Also numerical computations require that the solution set consists only of the isolated points in optimization domain. This can be served by imposing constraints [3]. In a constraint optimization problem the algorithm has to constantly meat the imposed conditions while maximizing or minimizing the cost function [6]. Beamforming incorporating sensor arrays has led to important applications in radio communications, sonar, radar, acoustics, and many other areas [1].As an instance, in a cellular system, the desired and the interfering signals originate from different spatial locations. This spatial separation is exploited by a beam former that can be regarded as a spatial filter separating the desired signal from the interference [6]. By incorporating adaptive algorithms, the beam former can suppress interferences coming from the directions that are different from the desired direction [2]. Capon and Frost beam former are among the basic applications of constraint beam forming [8]. The demand of ever increasing truly personal communications relying upon the smart antenna systems and advent of cognitive antenna [9] has increased the need of fast and accurate adaptive algorithms capable of meeting the diverse challenges. To cope with the needs of access throughput and always on requirement, the spatial dimension provided by Multi-Antenna terminals gives rise to multiple additional network resources enabled by the versatile efficient adaptive systems [10]. Similarly, the antenna arrays are widely used in modern radars to meet the challenges of increased range coverage, target identification, trajectory, faster data rates and multi beam applications [11]. Due to increased potential of co-channel interference, the popular user area applications such as Pico cells demand efficient beamforming systems. The reaction time is also a compelling factor in development of the more effective and efficient systems. In a similar manner, the ever increasing air traffic, demands accurate beamformers with good convergence characteristics and less reaction time to direct flights.

The proposed algorithm offers an answer to these challenges. In the following section, the structural development of the proposed algorithm is provided, complemented by convergence analysis and realistic MATLAB simulations to verify the performance of the algorithm.

2. CONSTRAINT ADAPTIVE NATURAL GRADIENT ALGORITHM (CANA)

Constraint optimization technique is aimed at minimizing the total noise power at the array output while maintaining a chosen frequency response in the Direction of Interest [2]. Knowledge of Desired Direction of Interest (DoI) is the basic information required by the algorithm. Optimization based upon natural gradient algorithm is employed to develop the technique we call the constraint adaptive natural gradient algorithm.

Consider a conventional narrowband adaptive array processor depicted in Figure 1. The signal from an incident source impinges on every element of the array. This is subsequently multiplied with relative weights and summed as the antenna array output. The array output y_n can be given by the following relationship:

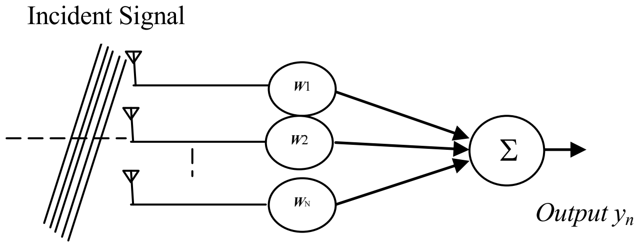

Figure 1: Generic adaptive array processing scheme

$$y_n = \mathbf{w}_n^T \mathbf{x}_n \quad (1)$$

with $$\mathbf{w}_n = [w_1 w_2 w_3 \cdots w_N]^T \quad (2)$$

and $$\mathbf{x}_n = [x_1 x_2 x_3 \cdots x_N]^T \quad (3)$$

Where $\mathbf{w}$ is the complex weight vector and $\mathbf{x}$ is the incoming data vector. To solve the constrained optimisation problem, we have to minimize the average output $E[y_n{}^2]$ with look direction constraint. The look direction constraint can be formulated as in [2]

$$\mathbf{c}^T \mathbf{w}_n = \mathbf{f} \quad (4)$$

Where $\mathbf{c}$ is the constraint vector and $\mathbf{f}$ the look direction constraint. For a single look direction constraint, **f** takes the scalar value of 1. From equation (1) and (4) the cost problem can be formulated as

$$J_{(\mathbf{w})} = \text{Minimize } \mathrm{E}\left[\mathbf{w}_n^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{w}_n\right] \quad (5)$$

Subject to the constraint given in equation (4). By using Lagrangian method the constraint cost function can then be defined as

$$J_{c(\mathbf{w})} = \mathbf{w}_n^T \mathbf{R}_n \mathbf{w}_n + \lambda(\mathbf{c}\mathbf{w}_n^T - 1) \quad (6)$$

Where $\mathbf{R}$ is the correlation matrix of the input data vector $\mathbf{x}$ and λ is the Lagrange multiplier. Minimizing of the cost function is to set its gradient equal to zero. Amari et al. proposed that the natural gradient of a cost function J can be written as in [5].

$$\tilde{\nabla} J_{(\mathbf{w})} = \mathbf{G}^{-1} \nabla J_{(\mathbf{w})} \quad (7)$$

Where ∇J_w and $\mathbf{G}$ are the conventional gradient of the cost function and the Riemannian metric respectively. The conventional gradient for the constraint problem under discussion is described in [2] and is given by

$$\nabla J_{c(\mathbf{w})} = \mathbf{R}_n \mathbf{w}_n + \lambda \mathbf{c} \quad (8)$$

While formulating a natural gradient algorithm on abstract Riemannian space the first and the foremost challenge is the calculation of Riemannian metric tensor. This metric tensor is basically the continuous dot product on the tangent space to the abstract Riemannian surface under observation and carries the curvature information of the surface to be optimized. For flat surfaces and optimization spaces this metric **G=I**, the identity matrix. From equation (6) the constraint cost function can be derived as follows.

$$J_{c(\mathbf{w})} = J_{(\mathbf{w})} + \lambda(\mathbf{c}^T \mathbf{w} - 1) \quad (9)$$

$$J_{c(\mathbf{w}+\Delta\mathbf{w})} = J_{c(\mathbf{w})} + \Delta J_{c(\mathbf{w})} \quad (10)$$

Where $J_{c(\mathbf{w})}$ is the constraint cost function, as defined in equation (6). The natural gradient is related to the $\mathbf{L_2}$ norm of the increment in the weight values, hence

$$J_{c(\mathbf{w}+\Delta\mathbf{w})} = J_{c(\mathbf{w})} + \varepsilon \|\Delta\mathbf{w}\|^2 \quad (11)$$

But $$J_{c(\mathbf{w}+\Delta\mathbf{w})} = J_{(\mathbf{w}+\Delta\mathbf{w})} + \lambda[\mathbf{c}^T(\mathbf{w}+\Delta\mathbf{w}) - 1] \quad (12)$$

Therefore equation (10) transforms to

$$J_{(\mathbf{w}+\Delta\mathbf{w})} + \lambda[\mathbf{c}^T(\mathbf{w}+\Delta\mathbf{w}) - 1] = J_{(\mathbf{w})} + \lambda(\mathbf{c}^T\mathbf{w} - 1) + \Delta J_{c(\mathbf{w})} \quad (13)$$

This implies that

$$J_{(\mathbf{w}+\Delta\mathbf{w})} + \lambda[\mathbf{c}^T \Delta\mathbf{w}] = J_{(\mathbf{w})} + \Delta J_{c(\mathbf{w})} \quad (14)$$

Since we have set $\Delta J_{c(\mathbf{w})} = \varepsilon \|\Delta\mathbf{w}\|^2$ then

$$J_{(\mathbf{w}+\Delta\mathbf{w})} - J_{(\mathbf{w})} + \lambda(\mathbf{c}^T \Delta\mathbf{w}) = \Delta J_{c(\mathbf{w})} = \varepsilon \|\Delta\mathbf{w}\|^2 \quad (15)$$

or

$$\Delta J_{(\mathbf{w})} + \lambda \mathbf{c}^T(\Delta\mathbf{w}) = \varepsilon(\Delta\mathbf{w})^T \mathbf{G}(\Delta\mathbf{w}) \quad (16)$$

Where **ε**, an arbitrary small number, can be absorbed in the step size μ. From equation (16) we deduce that

$$\frac{\Delta J_{(\mathbf{w})}}{\Delta\mathbf{w}} = -\lambda\mathbf{c} + \mathbf{G}\Delta\mathbf{w} \quad (17)$$

This implies that

$$\Delta\mathbf{w} = \mathbf{G}^{-1}\left(\frac{\Delta J_{(\mathbf{w})}}{\Delta\mathbf{w}}\right) + \lambda\mathbf{G}^{-1}\mathbf{c} \quad (18)$$

From Eq. (18), the iterative weight update equation can be established as

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\Delta\mathbf{w}_n \quad (19)$$

Using update rule formulated equation in (19) and employing the mathematical operations (given in the Appendix) we lead to following novel Constraint Adaptive Natural Gradient Algorithm (CANA) for adaptive array processing.

$$\mathbf{w}_{n+1} = \mathbf{P}[\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathbf{w})}}{\Delta\mathbf{w}_n})] + \mathbf{k} \quad (20)$$

where $$\mathbf{P} = [\mathbf{I} - \frac{\mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T}{\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}}]$$

and $$\mathbf{k} = \frac{\mathbf{G}^{-1}\mathbf{c}}{\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}}$$

The idempotent property of matrix **P** as given in appendix, confirms the suitability of matrix **P** to be the true projection operator for our proposed Constraint Adaptive Natural Gradient Algorithm (CANA). Note that the Riemannian metrics should not be confused with linear metrics. A metric is an abstraction of the notion of distance, whereas a Riemannian metric is an inner product on continuous tangent spaces. However a Riemannian metric induces a distance, the Riemannian distance [4, 5].The Riemannian metric for this type

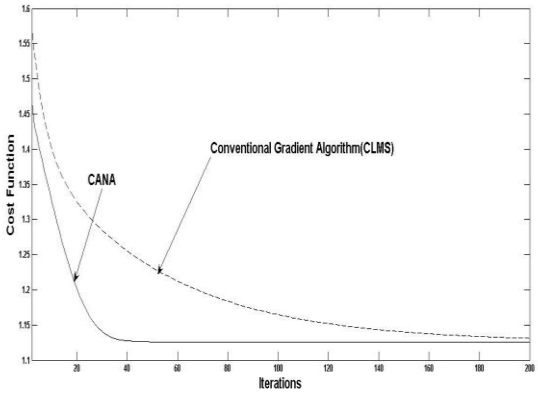

Figure 2: Comparison of convergence characteristics between CANA and conventional gradient based algorithm

of cost function has been proposed in literature [5]. But for our computations, we have slightly modified the metric to introduce the regularisation parameter ν. Thus we use the metric

$$\mathbf{G} = [v\mathbf{I} + (1-v)\mathbf{w}_n\mathbf{w}_n^T] \tag{21}$$

The functionality of ***v*** is to improve the rank deficiency of the metric, due to the generic structure of $\mathbf{w}_n\mathbf{w}_n^T$. Equation (7) reduces to conventional beamformer for **G=I**. Note that according to Woodbury Identity, inverse of **G** does not necessitate matrix inversion and it is given as

$$\mathbf{G}^{-1} = [\mathbf{I} - \frac{v\mathbf{w}_n\mathbf{w}_n^T}{1+v\mathbf{w}_n^T\mathbf{w}_n}] \tag{22}$$

This implies that computing inverse of **G** requires O (n^2) to compute $\mathbf{w}_n\mathbf{w}_n^T$ at each stage. Thus while the implementation of equation (20) is computationally expensive the convergence performance and accuracy, which far exceeds the performance of the conventional beamformer , makes the proposed algorithm very attractive in terms of implementations.

3. CONVERGENCE ANALYSIS

On the basis of equation (20) we can define a bias vector as follows

$$\mathbf{e}_{n+1} = \mathbf{w}_{n+1} - \mathbf{w}^o \tag{23}$$

Where $\mathbf{w}^o$ is optimum weight vector as given by Frost [2] in his famous solution for Linearly Constrained Adaptive Array Processing. From the equation (23) we can deduce that

$$\mathbf{w}_n = \mathbf{e}_n + \mathbf{w}^o \tag{24}$$

For the particular cost function of (6), the equation (17) can be written as

$$\mathbf{w}_{n+1} = \mathbf{P}[\mathbf{I} + \mu\mathbf{G}^{-1}\mathbf{R})]\mathbf{w}_n + \mathbf{k} \tag{25}$$

The steady state solution of equation (25) after the gradient part has vanished, corresponds to

$$\mathbf{k} = [\mathbf{I} - \mathbf{P}]\mathbf{w}^o \tag{26}$$

From equations (23-26)

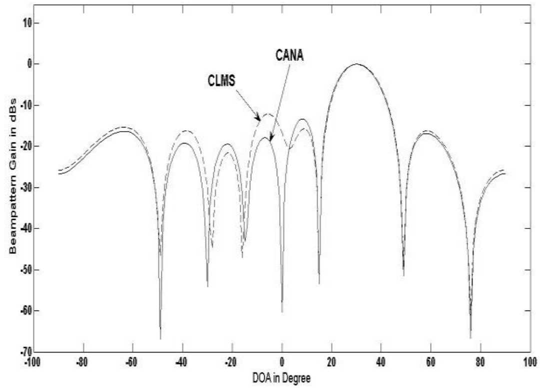

Figure 3: CANA and CLMS Array responses to look direction of 30^o signal. Note nulls in two interfering directions of 15^o and 75^o. Sensor Noise Power= 10 Interference Power = look direction Signal Power=1

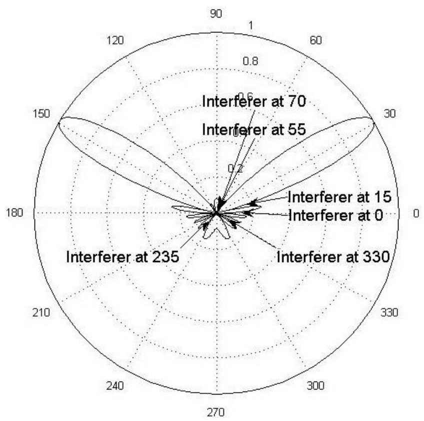

Figure 4: CANA Array response to look direction of 30^o with 6 interferers at 0^o,15^o , 55^o, 70^o , 235^o and 330^o

For optimal solution the gradient term must be zero. Therefore,

$$\mathbf{P}\mathbf{G}^{-1}\mathbf{R}\mathbf{w}^o = 0 \tag{28}$$

Therefore equation (24) reduces to

$$\mathbf{e}_{n+1} = \mathbf{P}[\mathbf{I} + \mu\mathbf{G}^{-1}\mathbf{R}]\mathbf{e}_n \tag{29}$$

The idempotent property of projection matrix allows us to manipulate equation (29) by pre-multiplying it by **P**. We get the same left hand side of equation (29), which means that

$$\mathbf{P}\mathbf{e}_n = \mathbf{e}_n \,\&\, \mathbf{G}^{-1}\mathbf{R}\mathbf{e}_n = \mathbf{G}^{-1}\mathbf{R}\mathbf{P}\mathbf{e}_n$$

With these realizations we can write equation (29) as

$$\mathbf{e}_{n+1} = \mathbf{P}[\mathbf{I} + \mu\mathbf{P}\mathbf{G}^{-1}\mathbf{R}\mathbf{P}]\mathbf{e}_n \text{ or}$$

$$\mathbf{e}_{n+1} = [\mathbf{I} + \mu\mathbf{P}\mathbf{G}^{-1}\mathbf{R}\mathbf{P}]^{n+1}\mathbf{e}_o \tag{30}$$

Where $\mathbf{e}_o$ is the initial value. Therefore the convergence condition is given by

$$-\frac{1}{\lambda} \le \mu \le 0 \tag{31}$$

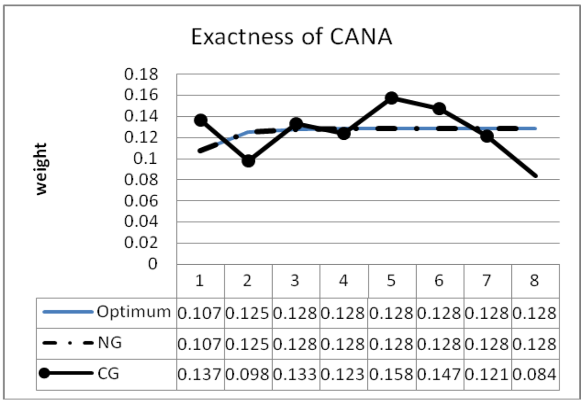

Figure 5: Comparison of weights calculated by Natural and conventional gradient based algorithms

In equation (31) $\lambda_{\max}$ is the largest Eigen value of the Square matrix $\mathbf{PG}^{-1}\mathbf{RP}$. It may be noted that the trace $[\mathbf{PG}^{-1}\mathbf{RP}] \succ M\lambda_{\max}$ where M is the dimension of matrix $\mathbf{PG}^{-1}\mathbf{RP}$. In this scenario a suitable choice can be

$$-\frac{M}{Trace[\mathbf{PG}^{-1}\mathbf{RP}]} \le \mu \le 0 \qquad (32)$$

4. SIMULATIONS

An 8 element uniformly spaced antenna array was simulated with CANA. The iterative constraint adaptive algorithm using conventional gradient (CLMS) was also simulated for comparison. We chose the convergence step size μ=0.01 and regularization parameter *v* =0.09. Input correlation matrix was taken to be of Toeplitz structure. Figure 2 shows the convergence of the cost function to the optimum value for this particular case. The Constraint natural gradient attains the optimum value very rapidly, and converges in about forty steps, while the conventional gradient based constraint optimization commonly known as Constrained LMS (CLMS) takes more than 200 steps to converge to the optimum value. Figure 3 describes CANA and CLMS array responses to look direction (30^o) signal. Note that CANA has deeper nulls in two interfering directions of 15^o and 75^o.The natural nulls of CANA and side lobe structure is better than the CLMS by at least 7dB. Figure 4 depicts the hostile environment of 6 interfering sources managed by CANA by providing nulls in the direction of interferers. Figure 5 is a comparison between the array weights for each eight sensors calculated by CANA and CLMS algorithms in steady state. The CANA weights for each of the sensor are exactly in agreement with the optimal weights obtained as the Wiener solution, while CLMS weights for each sensor vary widely. Along with quick convergence to the optimum values, it's a representation of how accurate is the proposed algorithm (CANA). The accuracy of CANA up to the three decimal places is phenomenal depiction of exactitude and correctness. Figure 6 shows interesting relationship between

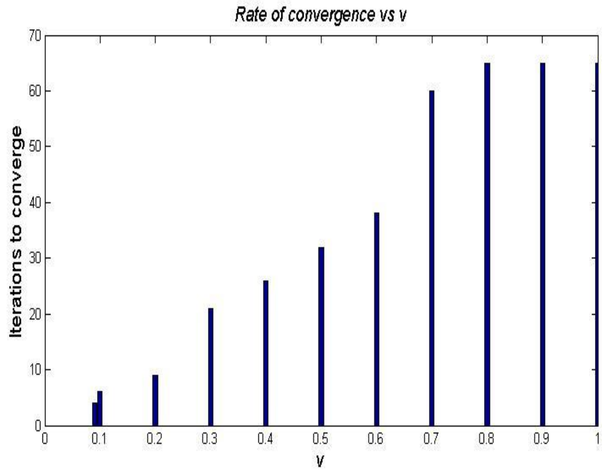

Figure 6: Relation between rate of convergence and regularisation parameter *v*

convergence rate and the value of *v* .The optimum value for this particular set of simulations was found to be 0.09 resulting in the algorithm converging to the optimum value in very few iterations. As we increase the value *v*, the algorithm takes a longer time to converge to the optimum value. At *v* =1 the value of $\mathbf{G}$ reduces to Identity matrix and the algorithm becomes the constrained LMS (CLMS) solution. Interestingly values of *v* less than 0.09 are out of bound for this particular case as the system becomes unstable for lower values. Figure 7 depicts the relationship between values of *v* and the step size μ, obtained by using equation (32)**.** Interpretation of this relationship is significant for the performance of CANA. The figure gives an account of limitation of choice of step size and its relationship to the particular regularization parameter *v*. In fact it puts a lower bound on the values of step size μ as represented in the Figure 6.

5. CONCLUSION

This paper has presented a novel technique for constraint adaptive algorithm based upon the natural gradient. This technique provides more uniform performance than those based upon ordinary stochastic gradients, yet is simple to implement. It may be noted that the rate of convergence of

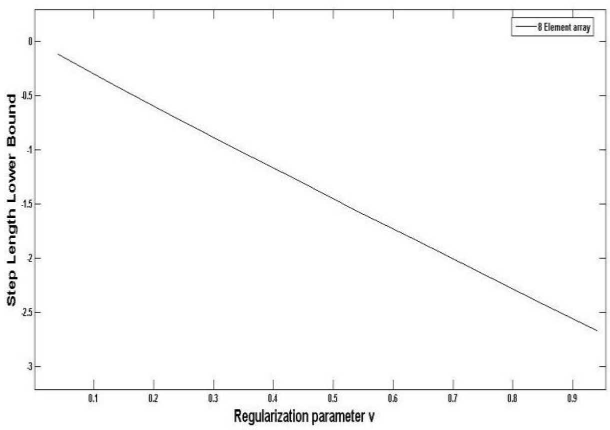

Figure 7 Step lengths vs. regularization parameter *v*

constraint adaptive natural gradient algorithm (CANA) is very sensitive to the value of v. In fact there are very few choices of v along with particular combination of the step size μ. Along with the ability of natural gradient algorithm to follow the exact surface of the optimization space, self correcting feature of constraint optimization ensures that the algorithm rapidly converges to steady optimum state, even for a quadratic function for which ordinary gradient based techniques e.g. LMS are considered as bench mark. So for a reasonable range of step size and regularization parameter the CANA algorithm outperforms other techniques.

6. APPENDIX

Following the procedure given equation (19)

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu(\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n}) + \lambda \mathbf{G}^{-1}\mathbf{c}) \qquad \text{(I)}$$

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu(\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})) + \mu\lambda \mathbf{G}^{-1}\mathbf{c} \qquad \text{(II)}$$

However we know that from the constraint equation (4) that $\mathbf{c}^T\mathbf{w}_{n+1} = 1$, thus pre Multiplying equation (II) with $\mathbf{c}^T$ leads

$$1 = \mathbf{c}^T\mathbf{w}_{n+1} = \mathbf{c}^T\mathbf{w}_n + \mu\mathbf{c}^T\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n}) + \mu\lambda[\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] \qquad \text{(III)}$$

Substituting this value in equation (II) we get

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n}) + [1 - \mathbf{c}^T\mathbf{w}_n + \mu\mathbf{c}^T\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})]\mathbf{G}^{-1}\mathbf{c} \Big/ [\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}]$$

or (IV)

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu G^{-1}(\frac{\Delta J(w)}{\Delta \mathbf{w}_n}) - \mathbf{c}^T[\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})]\mathbf{G}^{-1}\mathbf{c} \Big/ [\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] + \mathbf{G}^{-1}\mathbf{c} \Big/ [\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] \qquad \text{(V)}$$

$$\mathbf{w}_{n+1} = [\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})] - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ [\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] [\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})] + \mathbf{G}^{-1}\mathbf{c} \Big/ [\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] \qquad \text{(VI)}$$

$$\mathbf{w}_{n+1} = [\mathbf{I} - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}][\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})] + \mathbf{G}^{-1}\mathbf{c} \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c} \qquad \text{(VII)}$$

$$\mathbf{w}_{n+1} = \mathbf{P}[\mathbf{w}_n + \mu\mathbf{G}^{-1}(\frac{\Delta J_{(\mathrm{w})}}{\Delta \mathbf{w}_n})] + \mathbf{k} \qquad \text{(VIII)}$$

While $\mathbf{k} = \mathbf{G}^{-1}\mathbf{c} \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}$ and

$$\mathbf{P} = [\mathbf{I} - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}]$$

Now let's check the idempotent property of projection matrix by projecting it onto itself, which is by multiplying it with itself. Hence

$$\mathbf{PP} = [\mathbf{I} - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}][\mathbf{I} - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] \qquad \text{(IX)}$$

This further implies left hand side equal to

$$= [\mathbf{I} - 2\,\mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c} + \mathbf{G}^{-1}\mathbf{c}(\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c})\mathbf{c}^T \Big/ (\mathbf{c}^T\mathbf{G}^{-1}\mathbf{c})^2] \qquad \text{(X)}$$

and thus $\mathbf{P}$ is given as under

$$\mathbf{P} = [\mathbf{I} - \mathbf{G}^{-1}\mathbf{c}\mathbf{c}^T \Big/ \mathbf{c}^T\mathbf{G}^{-1}\mathbf{c}] \qquad \text{(XI)}$$

REFERENCES

[1] L C Godara, *Applications of Antenna Arrays to Mobile Communications, Part I.* CRC press, USA, 1997.

[2] Otis L. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing" *IEE*, Vol.60, August1972, pp926-935.

[3] P.-A. Absil, R.Mahony & R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton Uni. Press, USA, 2008.

[4] Stefan Waner, *Introduction to Differential Geometry and General Relativity*, (Lecture Notes) [online] http://people.hofstra.edu/Stefan_Waner/diff_geom/tc.html

[5] Shun-ichi Amari, "Natural Gradient Works Efficiently in Learning," *Neural Computations*, February 15, 1998, Vol. 10, No.2, pp251-276.

[6] IU Shah, "*Neural Network Assisted Real Time Beam forming*" MS Thesis, MAJU Pakistan, 2006.

[7] Rina Dechter, *Constraint Processing*, Morgan Kauf mann, USA, 2003.

[8] Jian lie and Petre Stoica, *Robust Adaptive Beamforming*, Wiley-Interscience, USA, 2006.

[9] Enzo Baccarelli, MauroBiagi, Cristian Pelizzoni, Nicola Cordeschi: "Multi-antenna cognitive radio for broadband access in 4G-WLANs," *MOBIWAC 2007*,pp66-73

[10] A.J.Paulraj, D.A.Gore, R.U.Nabar, H.Bolcskei,"An Overview of MIMO Communications: A Key to Giga bit Wireless", *Proc. of IEEE*, Feb, 2004, pp198-218.

[11] Alan J. Fenn, *Adaptive Antennas and Phased Arrays for RADAR and communications*, Artech House, 2008.