Constrained Adaptive Natural Gradient Algorithms for Adaptive Array Processing

By

Ijteba-ul-Hasnain Shah

In the fulfilment of the requirement for the degree of

Doctor of Philosophy

Centre for excellence in Signal and Image processing

Department of Electronic and Electrical Engineering

University of Strathclyde

Dedicated to my beloved parents Anwar Hussain Shah and Naseem Firdoos

Declaration and Copyright

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree. The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Ijteba-ul-Hasnain Shah

Acknowledgement

I am heartily thankful to my PhD supervisor, Professor Tariq s Durrani, whose encouragement, guidance and support from the initial to the final level enabled me to gain an insight into the research area of adaptive array processing. This thesis would not have been possible without his prestigious guidance and motivation. It is an honour for me to be your student sir!

I am very grateful to Dr Stephan Weiss for his provision of supportive material and very careful review in the final stages of my thesis. I am also thankful to Professor Peter Grant for his valuable recommendations for improving the quality of my thesis.

I would like to show my gratitude to Ms Fiona Buggy and especially Ms Sheila Forbes for their incredible administrative support. I am also thankful to all staff and members of CeSIP for helping me in number of ways.

I am indebted to many of my friends and colleagues; Shahzad, Zeng, Asif, Sameer, Zeshan and Adel for creating such an exquisite and caring environment in the lab.

I extend my sincere gratitude towards Dr. Mansoor Beg, Mr. A J Ghouri and NESCOM Pakistan for their generous financial support and help.

I do not find words to thank my parents, elder brothers and sisters back home for their unparallel love and support such that I always felt at-home despite living thousands miles away. I was always refreshed talking to them especially with my nephew Ali and niece Aliza.

Finally, I acknowledge my inability in thanking Almighty God who blessed me with intellect and chose me among the knowledge seekers.

Abstract

In a constrained optimisation problem an algorithm has to constantly observe the imposed constraints while maximising or minimising the cost function. The realistic minimization of these cost function problems is not trivial and generally involves manifolds with non linear surfaces. Working with a search space that carries the nonlinear manifolds introduces certain challenges in implementation of the algorithm, such as the choice of step size and the appropriate direction towards the optimal solution. The most popular optimisation technique used, perhaps due to its simplicity is the gradient descent technique. However convergence speed of a gradient adaptation can be slow when the slope of the cost function varies widely for the small changes in the adjusted parameters. Any optimisation scheme that exploits the given structure of the underlying space does not encounter such phenomenon and is deemed successful with better convergence and accuracy. Since the procedure of finding optimisers is exactly a search based on the geometric information of both the constraints and the cost function, it is very important to develop search techniques using the intrinsic geometry properties of both functions. Natural gradient modifies the search direction according to the Riemannian structure of the optimisation space. Natural gradient overcomes many of the limitations of the Newton's method, by assuming the surface of the cost function being minimised to be locally Euclidean.

In this thesis, novel Constrained Adaptive Natural Gradient Algorithms (CANA) have been developed exploiting the natural gradient technique, and have been applied to problems in array processing. The algorithms are capable of rapidly adjusting the response of an array of sensors to a signal coming from Direction of Interest (DoI) and suppressing signals incident on the array from other directions. Constrained optimisation techniques have been extended to enhance the look direction signal in the presence of interference using adaptive natural gradient techniques. The algorithms provide more uniform performance than those based upon conventional tangential gradients, yet are simple to implement. Along with the ability of natural gradient algorithm to follow the exact surface of the optimisation space, self correcting feature of constrained optimisation makes the algorithms rapidly converge to steady feasible points. Numerical simulations confirm algorithms' suitability for beamforming, null steering and mitigating hostile interferences.

Riemannian metric, an essential component of CANA, is updated at each iteration with the update in weight vector,and is thus the major contributor in algorithm computational complexity. So in order to reduce the number of arithmatic operations at each iteration, a more appropriate non adaptive Riemannian metric is proposed. Inclusion of this fixed metric significantly reduces the complexity of technique developed earlier in the thesis. The technique has been further extended to the adaptive step size algorithm.

Next broadband adaptive array processing has been developed through a wave-number filtering based approach. The signal after filtering is summed by an array processor according to a wave-number constraint. The present technique is also simple in the sense that spatial and temporal information has been mapped to only one dimension in the wave-number domain. Mathematical analysis and numerical simulations confirm the algorithm's ability of operating over the full bandwidth of 0 to 2π. The technique has also found use in restoring the capability of cancellation of total number of hostile interferers equal to M-1 for an M sensor array. It eliminates the need for the calculation of tap spacing or the determination of the optimum sampling, in deriving the temporal information from the signals, as the algorithm directly provides the coefficients of the wave-number domain filters. Despite defined at the midpoint of the bandwidth, algorithm prevents from causing main lobe to be contracted at the higher frequencies and expanded at lower frequencies. Beam steering is embedded in the developed scenario, thus the algorithm does not undergo the side-lobe deterioration as in the case of TDL. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, so the side-lobe one-to-one correspondence remains conserved. Extensive computed results are provided to verify the performance of the proposed algorithms.

Table of Contents

List of Figures

List of Tables

List of Symbols

$s(t)$	Signal at time t
$\mathbf{x}(t)$	Received signal vector at time t
$y(t)$	Array output at time t
$d(t)$	Desired response at time t
$e(t)$	Difference in the desired response and array output at time t
$\boldsymbol{w}$	Weight vector
$f(\boldsymbol{w})$	General optimisation surface
$J_{(\boldsymbol{w})}$	Unconstrained cost function
$J_{c(\boldsymbol{w})}$	Constrained cost function
A_o	Signal coefficient
θ	Angle of arrival of the signal
T_s	Sampling time
τ	Time delay
$\tau(\theta)$	Spatial time delay
$a(\theta)$	Array response vector
$\boldsymbol{f}$	Set of constraints
$\boldsymbol{f}_a$	Set of constraints for multi beam antenna
f_c	Scalar constraint
$\mathbf{A}(\theta)$	Array steering matrix for multi beam antenna
$\boldsymbol{c}$	Constraint vector
$\mathbf{v}(t)$	Sensor noise vector at time t
μ	Step size

μ_o	New step size for NLMS
α	Step size for the adaptive step-size algorithm
λ	Wavelength, wherever used as Lagrange multiplier is defined thereof
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers
θ_d	Desired direction of interest
σ_v^2	Noise covariance
$\boldsymbol{p}$	Cross correlation between desired signal and the received data vector
$\nabla J_{(\boldsymbol{w})}$	Gradient of the cost function $J_{(\boldsymbol{w})}$
$\tilde{\nabla} J_{(\boldsymbol{w})}$	Natural gradient of the cost function $J_{(\boldsymbol{w})}$
$\mathbf{C}$	Constraint matrix
$\boldsymbol{w}_{opt}, \boldsymbol{w}^o$	Optimum weight vector
$\frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}, \frac{\Delta f(\boldsymbol{w})}{\Delta \boldsymbol{w}}$	Gradient of the general optimisation surface w.r.t. weight vector $\boldsymbol{w}$
$\mathbf{G}(\boldsymbol{w})$	Riemannian metric over the space defined by vector $\boldsymbol{w}$
$\frac{dJ}{dr, d\theta}$	Polar gradient of the cost function J
$\frac{dJ_{NG}}{dr, d\theta}$	Polar natural gradient of the cost function J
k	Wave-number
$d\varphi(f)$	Infinitesimal component of signal spectrum defined over the frequency f
$d\varphi(k)$	Infinitesimal component of signal spectrum in Wave-number k
$z(t)$	Output of the wave-number k based broadband filter at time t
$\boldsymbol{h}(k)$	Broadband filter weight vector in wave-number domain
$\boldsymbol{h}_o(k)$	Optimum broadband filter weight vector in wave-number domain
$\Phi_0(k)$	Broadband signal spectrum in wave-number domain

$\mathbf{\Phi}(k)$	Covariance of the signal in wave-number domain
$\mathbf{\Phi}_v(k)$	Noise covariance matrix in wave-number domain
$\mathbf{A}(k_o, \theta)$	Array steering matrix defined over the midpoint of wave-number spectrum
$\boldsymbol{\psi}(k)$	Covariance matrix of all signal sources in wave-number domain

List of Acronyms

AWGN	Additive White Gaussian Noise
Bro CANA	Broadband Constrained Adaptive Natural Gradient Algorithm
CANA	Constrained Adaptive Natural Gradient Algorithm
CANA-NAM	Constrained Adaptive Natural Gradient Algorithm with Non Adaptive Metric
CLMS	Constrained Least Mean Square
CG	Conventional Gradient
DoA	Direction of Arrival
DoI	Direction of Interest
dB	Deci Bell
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
FDFIB	Frequency Domain Frequency Independent Beamforming
GPS	Global Positioning System
HPBW	Half Power Beam Width
INR	Interference to Noise Ratio
LMS	Least Mean Square
LCMV	Linearly Constrained Minimum Variance
MVDR	Minimum Variance Distortion less Response
MMSE	Minimum Mean Square Error
NG	Natural Gradient
NLMS	Normalised Least Means Square Algorithm
RLS	Recursive Least Square
R&D	Research and Development
SAR	Synthetic Aperture RADAR
SNR	Signal to Noise Ratio

SINR	Signal to Interference plus Noise Ratio
SMI	Sample Matrix Inversion
SRV	Spatial Response Variation
SDL	Sensor Delay Line
TDL	Time Delay Line
THEA	Thousand Element Array
TDFIB	Time Domain Frequency Independent Beamforming
ULA	Uniform Linear Array

Chapter 1
Introduction

This chapter presents an introduction to the research problem and overview of the thesis structure. The thesis structure follows bottom-up style, such that individual base elements of the array processing and constrained optimisation have been first specified in detail and then progressively built up to construct a conclusive outcome. Thesis contribution and list of publications has also been provided at the end of this chapter.

1.1 Introduction

Humans have always sought to mimic nature [1]. In this pursuit we have reached such impressive levels that we are able to incorporate some of the human thinking process in computers by mimicking human characteristics such as making decisions and operating autonomously. From flying planes to the submarines and from fixed to adaptive systems, almost every human invention can be traced back to the surrounding environment or the essential systems within himself. By adapting mechanisms and capabilities from nature, scientific approaches have helped humans understand the related phenomena and the associated principles in order to engineer novel devices and improve their capability [1]. On the similar note, the receptors of our taste buds, natural sounds and the cosmic systems in the universe, exhibit array paradigms for the concentration, convergence and conveying of the energy to the response processing systems.

Having found way in as early as 1905, array processing [2-3] has become one of the major areas of signal processing [4-5]. It has been studied extensively [2, 4, 6-7] due to its wide applications in various areas ranging from radar, sonar, microphone arrays, radio, astronomy, seismology, medical diagnostics, commercial communications and satellite communications [3-5, 7-13]. In radar and communication systems, antenna arrays are traditionally used to meet the challenges of increased coverage, target identification, trajectory tracking and faster data rates. An array is a composite antenna formed from two or more, generally omni directional radiators. However the elements forming an array could be dipoles, dish reflectors, slots in a wave guide, or any other type of radiator. The basic performance of an

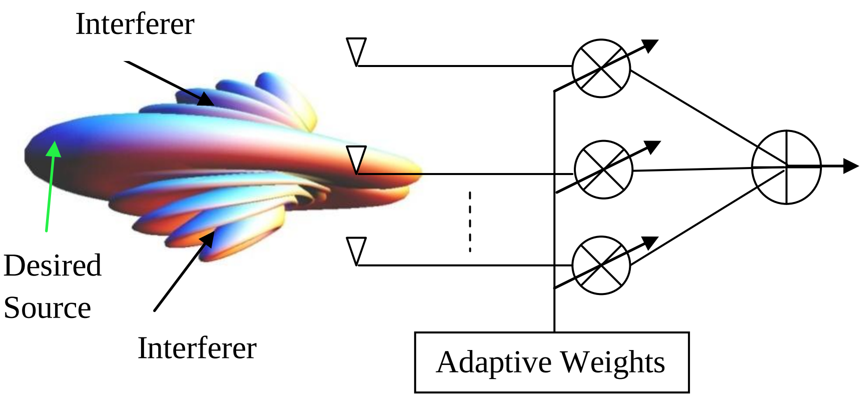

Figure 1.1: Constrained adaptive array processing

antenna array is to synthesise narrow directive beams that may be steered, mechanically or electronically, literally in any direction [9]. Constrained processing for passive arrays is the method used to create the beam pattern of the antenna array by adding constructively the phases of the signals incident on the array from a direction of interest, and nulling the pattern of the sources that are undesired. For narrowband sensors this can be done by attaching a weight network analogous to simple FIR tapped delay line filter [14]. These weights may also be changed adaptively, and used to provide optimal beamforming, by reducing the Minimum Mean Square Error (MMSE) between the desired and resultant beam pattern formed, called adaptive beamforming.

Figure 1.1 is a description of a generic constrained adaptive processor in which a desired source is at broadside of the antenna array while two hostile interferers are incident from arbitrary unknown directions. In addition to the interference, the sensor output also consists of sensor noise which is assumed to be Additive White Gaussian Noise (AWGN). Constrained adaptive array processing aka adaptive beamforming requires that the outputs of array antennas should be passed through a processor comprising filters with coefficients or weights so as to synthesize a desired beam pattern. Therefore the goal is to find a set of weights for adjusting the outputs of sensors of an array, in order to produce a far field pattern that optimizes the reception of a target signal along a direction of interest and suppresses unwanted signals from any direction along with beam steering capability.

In this thesis the problem has been solved through the development of the novel constrained adaptive natural gradient algorithms based upon the concepts from differential geometry. The natural gradient technique provides more uniform performance than those based upon ordinary stochastic gradients, yet simple to implement [15-17]. Along with the ability of natural gradient algorithm to follow the exact surface of the optimisation space, self correcting feature of constrained optimisation makes the algorithms rapidly converge to steady optimum state for range of applications.

1.2 Thesis Structure

This thesis has been arranged in bottom-up style. Individual base elements of the array processing and constrained optimisation technique have been first described in detail. These elements have been linked together to form basic structure of array processing, which then in turn have been linked, to from a complete solution in natural gradient based constrained optimisation.

First chapter has already set the scene by describing the problem of interest and the brief overview of the its importance for the diverse field of array signal processing, followed by the thesis structure and the thesis contribution.

Chapter 2 provides background and prologue to some elementary systems in constrained adaptive array processing. Introduction to the antenna arrays, different geometric formulations and conventional constrained adaptive techniques have been described therein. Formulation of the constraint for narrowband and broadband processors has been outlined followed by the discussions and conclusion.

Chapter 3 gives detailed description of constrained optimisation based upon natural gradient algorithm. The technique has been described by explaining and applying concepts of Riemannian metrics and differential geometry, on which the algorithm is based. A mathematical development of Constrained Natural Gradient Algorithm (CANA) is given along with numerical design example, results and critical analysis. This chapter also presents a critical extension to the Constrained Adaptive Natural Gradient Algorithm (CANA), through inclusion of a non adaptive metric in place of adaptive Riemannian metric. Inclusion of this fixed metric significantly reduces the complexity of the technique developed earlier. The technique has been further extended to formulate an adaptive step size algorithm. This algorithm enhances the convergence properties of a fixed step fixed Riemannian metric algorithm. Mathematical analysis and realistic MATLAB simulations are developed to underpin the fast convergence of the adaptive weights, and the suitability of algorithm for operations in adverse environments. Discussions on the numerical results are provided under the simulation examples followed by the conclusion at the end.

Chapter 4 outlines broadband constrained array processing in wave-number domain. It comprises of sections on introduction, mathematical development and simulation of the proposed methodology. This chapter provides a novel closed form solution and iterative solution based upon natural gradient technique where an array of uniformly spaced sensors is constrained to produce optimum desired response. Analysis and discussion on the simulations and concepts derived therein are provided in detail. Further a section on steering of the broad band arrays is provided followed by conclusion at the end.

Chapter 5 concludes this research monograph with evidences of conformance of the natural gradient based constrained adaptive techniques. Some limitations and drawbacks of the developed algorithms are discussed. This chapter also outlines some future applications and extensions in the subject research area, followed by the bibliography.

1.3 Thesis Contribution

This thesis is a research monograph on natural gradient based constrained optimisation. Working with an optimisation or search space that carries the structure of a non linear manifold, introduces certain challenges in the algorithm implementation. In the typical search methods, iterative algorithms rely heavily on approximating first and second order derivatives of the cost function in Euclidean spaces. A new iteration is proposed which is generated by adding an update increment to the previous update and the process is repeated till a steady state is reached. In order to define algorithms on manifolds, these operations must be translated into differential geometry. Natural gradient modifies the slope according to the actual structure of the optimisation space and thus provides a better approximation to the steepest descent direction.

Based on natural gradient, several optimisation algorithms, addressing the issues of narrow as well as broadband beamforming, have been developed in this thesis. Following is the brief synopsis of these techniques, with detailed description given in subsequent chapters.

1.3.1 Constrained Adaptive Natural Gradient Algorithm (CANA)

Novel Constrained Adaptive Natural Gradient Algorithm (CANA) is capable of adjusting the response of an array of sensors to a signal coming from Direction of Interest (DoI) and suppressing interferers coming from other directions. Constrained optimisation techniques have been developed to enhance the look direction signal in the presence of additive noise and interference using adaptive natural gradient techniques. Mathematical analysis and realistic MATLAB simulations confirm the suitability of algorithm in calculating iterative tap weights of the processor attached with the sensor array. This technique outperforms ordinary gradient based methods, in terms of convergence, yet is simple and computationally efficient. The self correcting feature of the constrained optimising technique complements the natural gradient algorithm's ability of following the exact surface of the optimisation space, resulting in fast convergence to the feasible point.

1.3.2 Constrained Adaptive Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM)

This algorithm is a critical extension to the previously developed Constrained Adaptive Natural Gradient Algorithm (CANA), through inclusion of a non adaptive metric in place of adaptive Riemannian metric. Inclusion of this fixed metric significantly reduces the complexity of previously developed technique. Complemented with constrained optimisation techniques the algorithm 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. Mathematical analysis and realistic MATLAB simulations are developed to underpin the fast convergence of the adaptive weights, and the suitability of algorithm for operations in adverse environments.

1.3.3 Adaptive Step-Size Algorithm

Generally, the optimising algorithm for solving the linear and nonlinear equations is also very sensitive to the search strategy and the choice of the variables value [18]. In case of nonlinear equations and because of the complex search strategy, it needs too long time to run these algorithms. On the other hand, if the choice of the variable value is not reasonable, the results would be restricted to some parts of the extreme values, thus it is difficult to get the high precision solutions [18-19]. Therefore the algorithm should also be able to change the step size according to the optimisation space. To incorporate this adaptability in existing solution, an adaptive step-size scheme has also been developed to complement the adaptability of CANA-NAM. The resultant algorithm is fast in attaining the optimum steady state as compared to the fixed step-size algorithm.

1.3.4 Constrained Broadband Array Processing in Wave-Number Domain

This technique is developed for the broadband array processing in wave-number domain. The approach uses wave-number domain filters associated with each sensor. An optimal array processor is developed which exploits a wave-number constraint. The present technique is also simple in the sense that spatial and temporal information is mapped to only one dimension of wave-number domain. Mathematical analysis and numerical simulations confirm the algorithm's ability of performing across the full bandwidth of 0 to 2π. Along with addressing the entire bandwidth, the technique has been found useful in cancelling a total number of hostile interferers equal to M-1,where M is the number of sensors in the array.

The proposed technique eliminates the need for the calculation of tap spacing or determination of the optimum sampling, in deriving the temporal information from the signal, as the algorithm directly yields the coefficients of the wave-number domain filters. Despite the fact that the array is defined at the midpoint of the wave-number spectrum the algorithm prevents the main lobe from contracting at the higher frequencies and enlarging at lower frequencies. Results show that the more powerful interferers, the deeper the nulls established in the array processor.

Beam steering is embedded within the scenario as no additional arrangement for pre-steering or post- steering [19-20] is required in order to adjust the beam response towards the desired direction. Moreover the processor does not suffer side-lobe deterioration as in the case of TDL based broadband adaptive beam steering. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, the one-to-one correspondence in the side-lobes remains conserved.

1.3.5 List of Publications

The work has been published in a number of international conferences and is being prepared to be presented in renowned journals and transactions. Below is the detail of the publications based on some parts of this thesis.

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Constrained Adaptive Natural Gradient Algorithm (CANA) for Adaptive Array Processing, 17th European Signal Processing Conference (EUSIPCO 2009) Glasgow, Scotland, August 24-28, 2009.

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Broadband Constrained Natural Gradient Based Adaptive Processing, 2009 IEEE Workshop on Statistical Signal Processing, Cardiff, Wales ,UK, Aug 31, 2009 - Sep 03, 2009.

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Constrained Natural Gradient Algorithm with Non Adaptive Metric, IEEE International Symposium on Signal Processing and Information Technology, December 15-18, 2010 - Luxor – Egypt.

Chapter 2

Array Processing

This chapter presents an overview of the array signal processing with an emphasis on constrained array processing. Signals having the same frequency components and originating from different locations can be differentiated through an array of antennas in conjunction with a smart processor called beamformer. Antenna arrays perform spatial filtering in order to discriminate useful signals from interferers in the presence of noise. For many decades, arrays have been traditionally used for radar, sonar, petroleum exploration, astrophysical exploration, biomedical imaging and communications. Smart and simple algorithms are generally required to perform these critical tasks. Constrained array processing is used when direction of the signal of interest is known a priori. In constrained array processing, the output of an array is constrained to adapt in a way so as to suppress all the signals except the one originating from the direction of interest. Similarly, in broadband array processing the arrays are constrained so as to implement a known frequency response for certain signals.

2.1 Introduction

An antenna array is a set of two or more antennas arranged in a particular geometrical configuration. The signals from several antennas are combined or processed in order to achieve enhanced performance over that of a single antenna [21]. Depending upon the application, antenna arrays are traditionally used for

- increasing the overall gain
- providing diversity
- cancelling out interference originating from particular directions
- processing the array output so that it is most/least sensitive in a specific direction
- localisation of the signal sources

- maximising the Signal to Interference plus Noise Ratio (SINR).

An isotropic antenna [4] is a hypothetical antenna with equal radiation pattern in each direction. Antenna radiators having omni-directional [2] characteristics are actually referred to as isotropic radiators. Many applications as in cellular communications and defence imply directional radiation patterns for selective transmission or reception of the signals. An antenna array may consist of a group of isotropic (omni-directional) radiators, or, the array may consist of a group of non isotropic radiators i.e. directional but these are usually identical. In a group of identical radiators all elements share the same orientation in space. The same orientation results in the desired effect of reinforcement or cancellation of the electric field intensity and ensures polarisation in the same direction in space.

2.2 Antenna Basics

Before we further elaborate the antenna arrays it is imperative to define some of the key terms relevant to antenna for completeness and their use in later analysis.

Beam Pattern

Relative distribution of the received power in space as a function of direction is the beam pattern of an antenna. In broadband antenna arrays beam pattern can have another dimension of frequency or wave-number along with direction in space.

Pattern Null

It is the angle at which the received power is zero.

Main Lobe

The main lobe of the beam pattern is the lobe containing the direction of maximum received power.

Side Lobe

Additional lobes in the beam pattern of the antenna other than the main lobe are called side lobes. Ideally no or minimum side lobes are required in antenna applications.

Half Power Beam Width

The beam width of the antenna is the angular width of the main lobe in its far field beam pattern. Half power beam width is also referred as 3dB beam width in the antenna literature.

First Null Beam Width

It is the ability of an antenna to reject the interference. The angle spanned by the main lobe is First Null Beam Width (FNBW).

Side Lobe Level

It is the power of highest side lobe with respect to the main lobe/beam.

Directivity

It is the directive gain in the angular direction of maximum radiation intensity.

Antenna Gain

Antenna gain is defined as the ratio of the radiated/received power to the power of an isotropic antenna.

Grating Lobes

The additional lobes formed in the beam pattern due to inter element spacing larger than half wavelength. Spatial aliasing is the phenomenon behind the formation of grating lobes.

2.3 Array Processing

For completeness, a brief review of array processing techniques is included here.

Arrays and beamformers offer a versatile and effective way of spatial filtering. Applications like phased array radar, air traffic control, synthetic aperture radar, source localisation and classification in sonar, satellite communications and imaging, earth mapping and astrophysical exploration and biomedical applications, incorporate discrete arrays [2, 4, 7, 21-23]. The applications in communications and defence often require narrow beams in order to achieve transmit and receive selectivity. A single dipole has a Half Power Beam Width (HPBW) of $\pi/2$ [2, 21]. The half power beam width can be decreased down to 50^o by

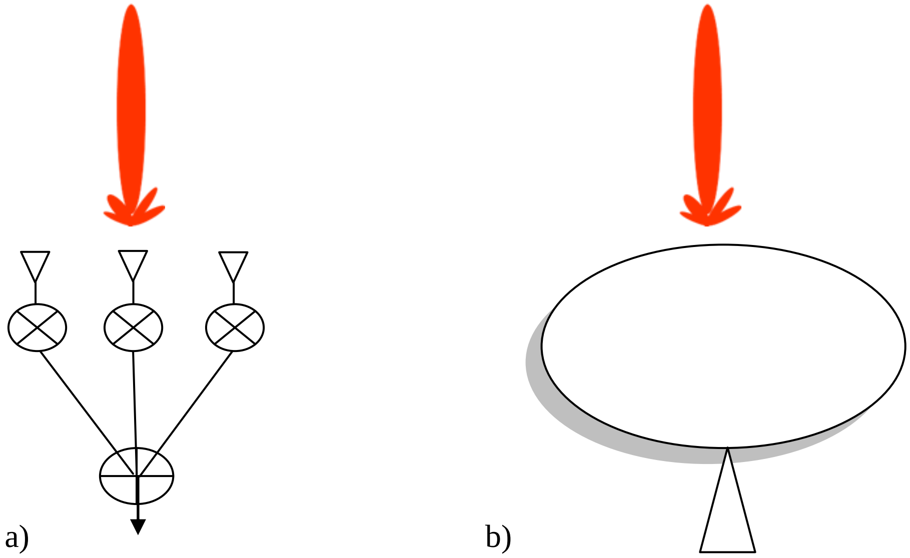

Figure 2.1: Comparison of multiple antenna array with a single directive antenna

increasing dipole length to 1.1 λ. A further increase in length will cause multi lobe beam pattern and loss of useful energy. Therefore alternative approaches need to be exercised to get the directional beam patterns of an antenna. Antenna array is defined as spatial configuration of a couple or more elements. In this arrangement the signal from all antennas is combined spatially to enhance overall signal in a particular direction. So by arranging antenna elements in several geometric arrangements complemented with adaptive processing, it is possible to design and develop intelligent and highly directive antennas.

The information contained in a spatially propagating signal may be either the location of its source or the content of the signal itself [3]. If we are interested in obtaining this information, we generally must deal with the presence of other, undesired signals. Much as a frequency selective filter strengthens signals at a particular frequency, we can choose to focus on signals from the direction of interest [10, 24]. However, this task can be accomplished by using a single sensor, if it has the ability to pass signals from certain directions while rejecting others. Such a single-sensor system, shown in Figure 2.1(b), is commonly found in communications and radar applications in which the signals are collected over a continuous spatial aperture using a parabolic dish. The signals are reflected to the antenna in such a way that signals from the direction in which the dish is pointed, are emphasized. Directivity of an antenna array depends upon its physical characteristics and geometric structure [2, 9]. However, the single-sensor based system has several drawbacks. Since the sensor relies on mechanical pointing for directivity, it can extract and track signals from only one direction.

Thus a slow antenna can miss the high speed or critical targets [7, 11]. The other critical issue with the sole sensor such as in the case of fixed beamformer too, is that, it cannot cope with the interferers which are arbitrary changing their locations. The solution of this very problem is a prime advantage of adaptive beamformers [4] which will be treated in detail later on. However in both the cases of Figure 2.1, the response is designed to emphasize signals from a certain direction through spatial filtering, either continuous or discrete.

An array of sensors has the ability to overcome the shortcomings of a single sensor. Figure 2.1 (a) illustrates the use of a sensor array. The sensor array signals are combined in such a way that a particular direction is emphasized. However, the direction in which the array is focused or pointed is almost independent of the orientation of the array [4]. Therefore, the sensors can be combined in distinct, separate ways so as to emphasize different directions, all of which may contain signals of interest. Since various weighted summations of the sensors simply amount to processing the same data in different ways, these multiple sources can be extracted simultaneously [25].

2.3.1 Array Signal Model

In their most general form, spatial signals represent wave fronts that propagate through space. These signals originate from a source, travel through a propagation medium, say, air or water, and arrive at an array of sensors that spatially samples the wave front. A processor can then take the data collected by the sensor array and attempt to extract information about the source, based on certain characteristics of the propagating wave.

Consider a signal received by the Uniform Linear Array (ULA) from an angle θ as in Figure 2.2. Each sensor receives the spatially propagating signal and transforms into the voltage signal. This voltage signal is then part of the receiver channel of Figure 2.1a. In addition, the receiver contains noise due to internal circuitry attached with the sensors, known as thermal noise.

A signal is a time-varying quantity having single or multiple frequencies and can be accordingly termed as narrowband or broadband. In array processing a signal is said to be broadband when all its frequency components not only undergo phase change but also experience a change in magnitude while travelling into an array of sensors, whereas a narrowband signal only experiences a phase shift.

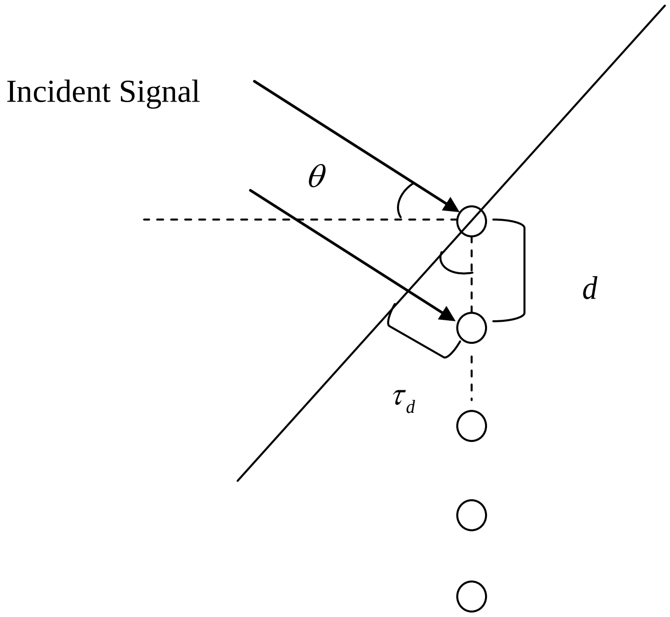

Figure 2.2: Spatial signal representation of a Uniform Linear Array (ULA)

A signal can generally be represented as

$$s(t) = \int e^{j2\pi ft} d\varphi(f) \tag{2.1}$$

Where $d\varphi(f)$ is the infinitesimal component of the frequency spectrum defined over the frequency range of interest. If narrowband assumption is made then we can write

$$d\varphi(f) = A_o \delta(f - f_o) df$$

Where A_o represents the amplitude of the signal and f_o the signal frequency, then equation 2.1 can be written as

$$s(t) = \int e^{j2\pi ft} A_o \delta(f - f_o) df \tag{2.2}$$

This implies that for $f = f_o$

$$s(t) = A_o e^{j2\pi ft} \tag{2.3}$$

The signal defined thus is termed as a narrowband signal. Let us consider an array of uniformly spaced *M* sensors. A single signal source is impinging signal on the array from the direction θ. Then the signal received at the 1st sensor from a direction θ can be written as

$$x_o(t) = s(t) + v(t) \tag{2.4}$$

Where $s(t)$ is defined by Eqs. 2.1, 2.3 and $v(t)$ is the Additive White Gaussian Noise (AWGN) contribution due to thermal characteristics of an antenna element. Then the signal received at the next sensor is a delayed version of the signal received at the 1st sensor as depicted in Figure 2.2 and is given by

$$x_1(t) = x_o(t - \tau_1(\theta)) \tag{2.5}$$

Where we have assumed that the delay at the 1st sensor $\tau_o(\theta) = 0$. Following the same approach the signal received by the third sensor is

$$x_2(t) = x_o(t - \tau_2(\theta)) \tag{2.6}$$

and finally the radiation received at the last sensor of the array is thus written as

$$x_{M-1}(t) = x_o(t - \tau_{M-1}(\theta)) \tag{2.7}$$

Since all the elements are equally spaced, the spatial signal has a difference in propagation paths between any two successive sensors of $d\sin(\theta)$ that result in a time delay of

$$\tau(\theta) = d\sin(\theta)/c$$

Therefore for the *m*th array element the time delay can be defined as

$$\tau_m(\theta) = md\sin(\theta)/c \qquad\qquad 0 \le m \le M-1$$

Where *d* is the distance between the adjacent antenna elements of Figure 2.2 in terms of wavelength and *c* is the velocity of the incident radiation. Using the definition of $s(t)$, Eq. 2.7 can be formulated for *m*th element as

$$x_m(t) = A_o e^{j2\pi f(t-\tau_m(\theta))} + v_m(t) \qquad\qquad \text{or}$$

$$x_m(t) = s(t)e^{-j2\pi f\tau_m(\theta)} + v_m(t) \tag{2.8}$$

The discrete time signals from a ULA may be written as a vector containing the individual sensor signals

$$\boldsymbol{x}(t)=\left[x_o(t) \quad x_1(t) \quad x_2(t) \quad x_3(t) \quad ...x_{M-1}(t)\right]^T \tag{2.9}$$

As sensor signals also constitute thermal noises therefore it can be written as.

$$\boldsymbol{x}(t)=\overset{\wedge}{\boldsymbol{x}}(t)+\boldsymbol{v}(t) \tag{2.10}$$

$$\overset{\wedge}{\boldsymbol{x}}(t)=s(t)\boldsymbol{a}(\theta) \tag{2.11}$$

Where $\boldsymbol{v}(t)=\left[v_o(t) \quad v_1(t) \quad v_2(t) \quad v_3(t) \quad ...v_{M-1}(t)\right]^T$ and

$$\boldsymbol{a}(\theta)=[1 \quad e^{-j2\pi f(d/c)\sin\theta_p} \quad e^{-j2\pi fq(d/c)\sin\theta_p} \quad e^{-j2\pi fM-1(d/c)\sin\theta_p}]^T \tag{2.12}$$

The vector $\boldsymbol{a}(\theta)$ is called the array response, where M is the total number of sensors. A single observation or measurement of this signal vector is known as an array snapshot. If all the signals from a single source developed at an array of antenna depicted in Figure 2.1 and Figure 2.2 are added up, then the array output $y(t)$ can be written as

$$y(t)=s(t)\sum_{m=0}^{M-1}e^{-j2\pi fm(d/c)\sin\theta} \tag{2.13}$$

For no noise, the Array Factor (AF) is defined as the ratio of the output to the input of an array

$$AF=y(t)/s(t) \tag{2.14}$$

Thus the array factor using the definitions above can be derived as

$$AF=\sum_{m=0}^{M-1}e^{-j2\pi fm(d/c)\sin\theta} \tag{2.15}$$

The above summation can also be written as a geometric series given below

$$AF=1+e^{-j\beta}+e^{-j2\beta}+e^{-j3\beta}+..............e^{-jM-1\beta}$$

Where $\beta=2\pi f\,(d/c)\sin\theta$ therefore

$$AF = \left[\frac{1 - e^{-jM\beta}}{1 - e^{-j\beta}} \right] \tag{2.16}$$

Rearranging Eq. 2.16

$$AF = e^{-j(M-1/2)\beta} \left[\frac{e^{j(M/2)\beta} - e^{-i(M/2)\beta}}{e^{j(1/2)\beta} - e^{-j(1/2)\beta}} \right] \tag{2.17}$$

$$AF = e^{-j(M-1/2)\beta} \left[\frac{\sin\left(M\beta/2\right)}{\sin\left(\beta/2\right)} \right] \tag{2.18}$$

If the reference point is taken as lying at the physical centre of the array, the array factor simplifies to

$$AF = \left[\frac{\sin\left(N\beta/2\right)}{\sin\left(\beta/2\right)} \right] \tag{2.19}$$

Here we have assumed that the antenna elements are essentially omni-directional sensors. If however they possess any directivity then the beam pattern is modified by the antenna individual gain [25]. It is imperative to note here that the above definition of the array factor holds valid for both transmitting and receiving arrays. Similarly the beam pattern of an array can be achieved by multiplying its array response vector or steering vector in Eq. 2.12 with set of weights $\boldsymbol{w}$, approximated by an adaptive or fixed array processor. We normally plot this beam pattern to get insight into the array performance for the range of angles as desired.

$$\text{Beam Pattern} = \left| \boldsymbol{w}^H \boldsymbol{a}\left(f, \theta\right) \right| \tag{2.20}$$

Here weight vector $\boldsymbol{w}$ is given by

$$\boldsymbol{w} = \left[w_o \quad w_1 \quad w_2 \quad w_3 \cdots w_{M-1} \right]^H \tag{2.21}$$

For a signal with multiple sources impinging on the array including interference is given by

$$x_m\left(t\right) = \sum_{p=1}^{N-1} A_p e^{j2\pi f\left(t - \tau_m\left(\theta_p\right)\right)} + v_m\left(t\right) \tag{2.22}$$

Where N is the total number of sources impinging on the ULA. Since the model is assumed to be linear, the extension to multiple signals, including interference sources, is straightforward and is given ample space in Chapter 3 and Chapter 4.

2.3.2 Spatial Sampling Theorem

Similar to the phenomenon of temporal sampling, the sensor array provides spatially sampled data that can be used for several applications. If two elements of a passive sensor array are placed very close to each other, the EM field incident on one antenna element couples with the EM field incident on the other antenna element thus erasing any possible phase shift between two waves being received on the antenna elements. The phenomenon is called mutual coupling responsible for spatial aliasing in array processing. Theorems applied to the FIR filter in the time domain may sometimes also be applied to a ULA in the spatial domain because of similarity between an FIR filter and a ULA [4]. In the time/frequency domain, the Nyquist sampling theorem states that for a band-limited signal with highest frequency *f*, the signal is uniquely determined by its discrete time samples if the sampling rate is equal to twice the frequency [26]. If the sampling rate is less than twice the frequency, there will be aliasing. Similarly to avoid spatial aliasing, the beamformer must satisfy the following criterion

$$d < \lambda / 2 \tag{2.23}$$

Where d is the distance between adjacent array elements and λ is the wavelength of incident radiation. Eq. 2.23 is known as the Nyquist sampling theorem in the spatial domain [14, 16]. Therefore to perform beamforming without spatial aliasing, the element spacing of the array must be less than half of the carrier wavelength. However the element spacing cannot be made arbitrarily small because of mutual coupling effects between elements [27].

2.3.3 Array Covariance Matrix

The signal received at an arbitrary sensor is composed of the essential three components; desired signal, interference and an independent thermal noise due to the sensor and internal circuitry attached to it such as front-end amplifier etc. Thus the signal irrespective of its bandwidth can be written as

$$x_m(t) = x_m^d(t) + x_m^i(t) + v_m(t) \tag{2.24}$$

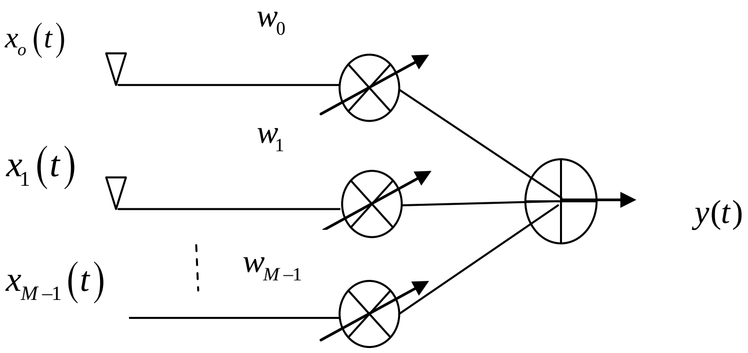

Figure 2.3: General beamforming array

Where $x_m^d(t)$, $x_m^i(t)$ and $v_m(t)$ are the desired, interference and the noise components on the mth sensor. Similarly the resultant received vector can also be written in a split form [28] as below.

$$\boldsymbol{x}(t)=\boldsymbol{x}^d(t)+\boldsymbol{x}^i(t)+\boldsymbol{v}(t) \qquad (2.25)$$

Thus the array covariance is the outer product of the received vector.

$$\mathbf{R}=E\left[\boldsymbol{x}(t)\boldsymbol{x}(t)^H\right] \qquad (2.26)$$

Where E is the expectation and H represents the complex conjugate transpose operator. Thus the covariance matrix can be devised as

$$\mathbf{R}=\mathbf{R}^d+\mathbf{R}^i+\mathbf{R}^v \qquad (2.27)$$

Where $\mathbf{R}^d=E\left[\boldsymbol{x}^d(t)\boldsymbol{x}^d(t)^H\right]$ is the desired signal covariance, $\mathbf{R}^i=E\left[\boldsymbol{x}^i(t)\boldsymbol{x}^i(t)^H\right]$ is the covariance of interfering sources and $\mathbf{R}^v=E\left[\boldsymbol{v}(t)\boldsymbol{v}(t)^H\right]=\sigma_v^2\mathbf{I}$ is the noise covariance contribution by the sensor thermal noise. Where due to the fact that signal, interference and noise are mutually uncorrelated and zero mean, we can assume that all the cross products are zero i.e.; $E\left[\boldsymbol{x}^d(t)\boldsymbol{x}^i(t)^H\right]=0$, $E\left[\boldsymbol{v}(t)\boldsymbol{x}^i(t)^H\right]=0$ and $E\left[\boldsymbol{v}(t)\boldsymbol{x}^d(t)^H\right]=0$ etc.

2.4 Beamforming

There are three major areas of array processing, detecting the presence of impinging signals and their classification, finding DoAs of the impinging signals and beamforming. In general, beamforming is a process of estimating the signal arriving from desired direction in the presence of noise and interference [24, 29]. This is accomplished by spatially sampling the signal with the help of an array of sensors and summing it via a fixed or adaptive processor. In its most general form, a beamformer produces its output $y(t)$ by forming a weighted combination of signals from the *M* elements of the sensor array, that is

$$y(t)=\boldsymbol{w}^H\boldsymbol{x}(t) \tag{2.28}$$

Where $\boldsymbol{w}$ is the vector of complex weights attached to each sensor and $\boldsymbol{x}(t)$ is the data vector composed of signals from all sources including interference and noise at the time instance *t*. Beamforming is generally divided into two forms;

a) Narrowband Beamforming b) Broadband Beamforming.

2.4.1 Narrowband Beamforming

In this type of array processing the bandwidth of the impinging signal is assumed to be narrow enough to ensure that the signal received at the both ends of the array are still correlated with each other [14]. Discrete signal samples are collected at spatially separated sensors, which then are processed to attenuate interference and subsequently extract the desired signal in the presence of noise. Figure 2.3 is a narrowband beamformer where the object of the array processor is to find the weight vector in order to craft the desired beam pattern in the look direction. The delay between the signals arriving at different sensors due to narrowband can be approximated by the phase difference due to position of sensors in the space. The output of a narrowband beamformer [30] is given as

$$y(t)=\sum_{k=0}^{M-1}w_k x_k(t) \tag{2.29}$$

2.4.2 Broadband Beamforming

If the bandwidth of signal increases the beamformer in Eq. 2.29 is no longer valid. This is because the delay cannot be approximated by a single phase shift between the sensor elements due to the angle dispersion and de-correlation of the signal across the array [28]. Tapped delay line with adaptive array processing behind each antenna is an approach to compensate for the dispersion arising from the wideband signal [31]. The decline in performance can be restored by de-correlating each antenna channel with the tapped delay. The output of a broadband beamformer shown in Figure 2.4 can be derived as

$$y(t)=\sum_{k=0}^{M-1}\sum_{l=0}^{P-1}w_{k,l}x_k\left(t-lT_s\right) \tag{2.30}$$

Where T_s is the delay between adjacent taps of the Tapped Delay Line (TDL). T_s is also the temporal sampling period and according to Nyquist sampling theorem should not be greater than half time period of the highest frequency component. Sensor Delay Line (SDL) processing [28] shown in Figure 2.5, is another extension to the time delay processing in which time delays have been materialised by putting extra sensors.

A detailed treatment of the broadband adaptive processing and a brief review of the previous techniques in the subject area have been provided in Chapter 4.

2.5 Adaptive Array Processor

Essentially it is required that the signal from desired direction should be enhanced and the nulls are inserted at the bearing/positions of interfering/jamming signals. As the interference can appear from any direction so a flexible structure for the array processor, which could adapt its output beam pattern as per the changing environment is required. Figure 2.6 depicts an adaptive processor, where the weights are adapted through the error feedback.

As a general case let us consider that each sensor in Figure 2.6 is connected to a single weight multiplier or a transversal filter or TDL as depicted in Figure 2.4. By altering the weights of an adaptive processor beamforming in any direction in the permissible range of $\pi/22$ to – $\pi/2$ is possible in case of ULA. The purpose is well served by employing Least Mean Square (LMS) for updating the weights in order to minimise the Mean Square Error (MSE) between the output and reference desired signal. For narrowband operations and specific

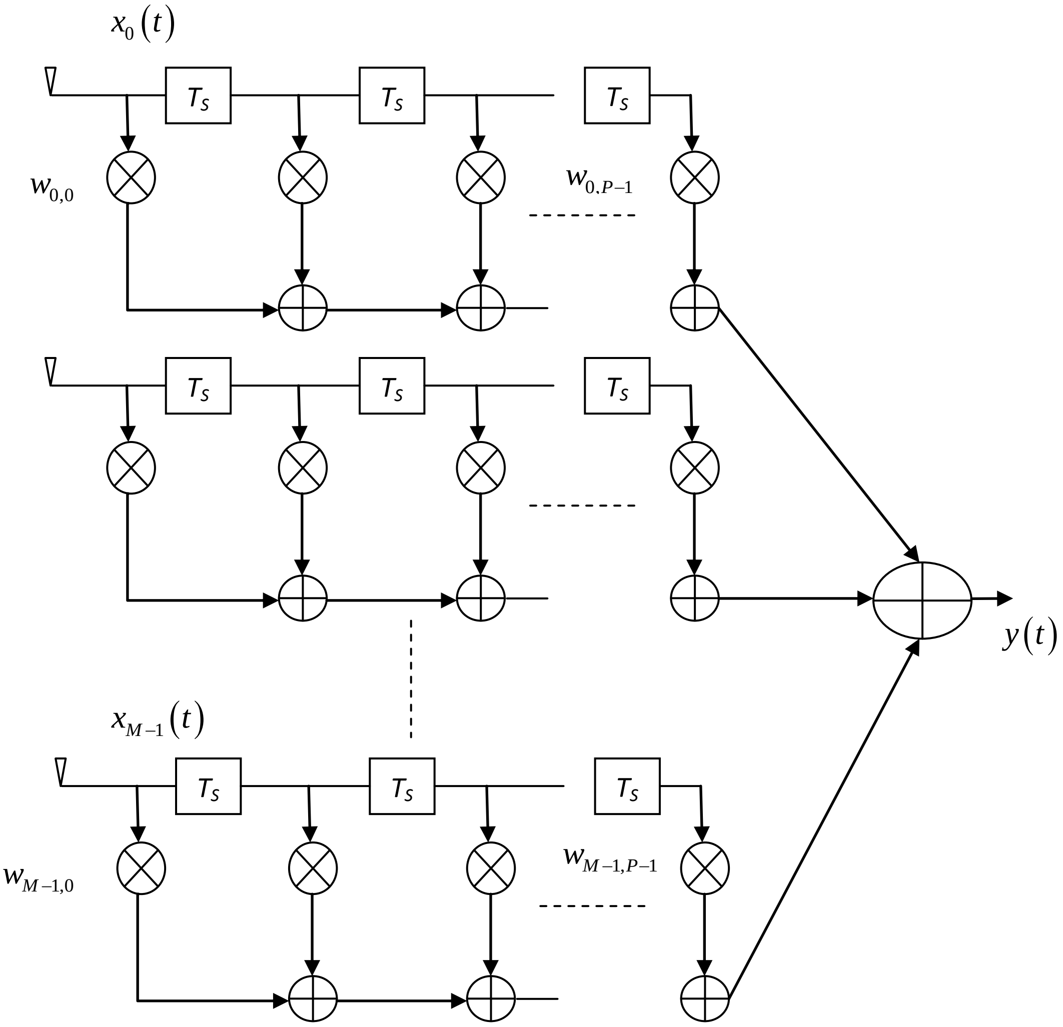

Figure 2.4: TDL based broadband beamforming

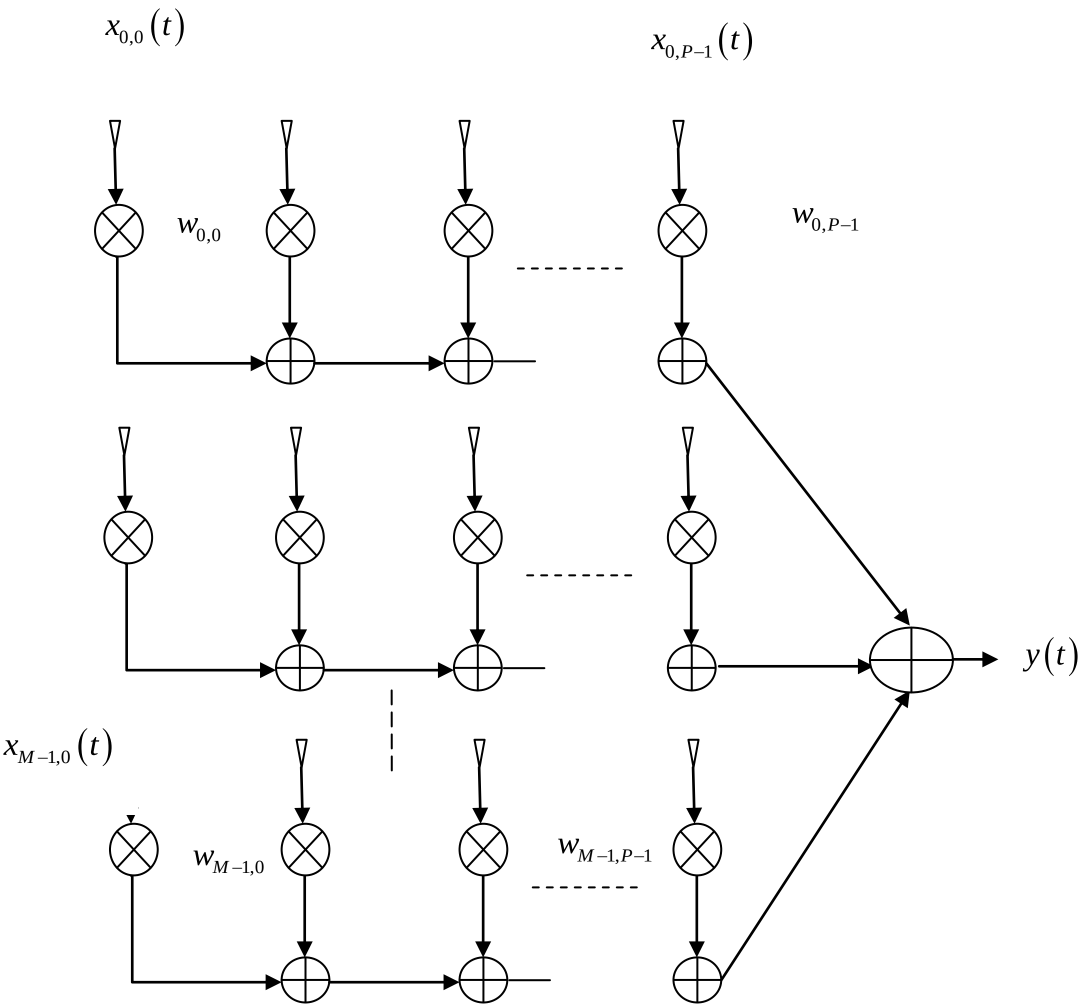

Figure 2.5: SDL based broadband beamforming

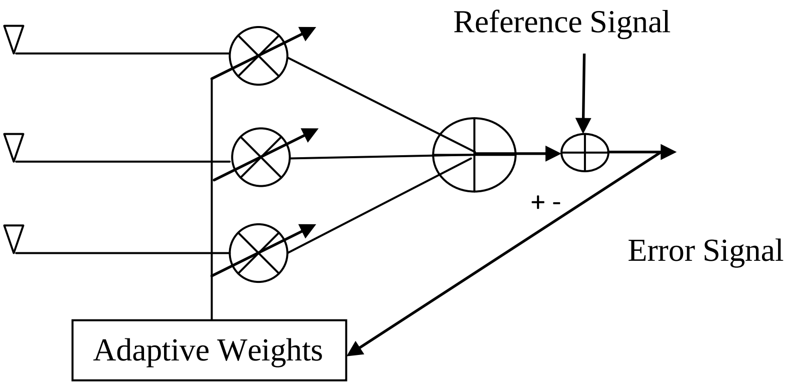

Figure 2.6: Adaptive array processor frequencies a sinusoidal pilot signal can well serve the purpose though it is not feasible for all especially broadband array operations. Generally reference based adaptive processors operate in two modes a) Learning mode and b) Linear combiner mode. In the learning mode the weights are adjusted until they converge with respect to the reference or desired signal and then switch to the second mode where output $y(t)$ is devised using the best linear estimate of the signal received at the array from the direction of interest-the so called "look direction". To compensate for any non-stationarity caused by the inclusion of new or moving interferer, the process is repeated at appropriate intervals.

If the sensor data vector $\boldsymbol{x}(t)$ is given by Eq. 2.21 and corresponding weights by Eq. 2.9 Then the output of the overall array processor can be expressed as

$$y(t) = \boldsymbol{w}^H \boldsymbol{x}(t) \tag{2.31}$$

The availability of reference signal as shown in Figure 2.6 is used to determine the error, which in turn can be used to devise an optimum response of the array processor. The error $e(t)$ is the difference between the output of an array and the desired or reference signal.

$$e(t) = d(t) - y(t) \tag{2.32}$$

This error signal is used according to some criterion for adjusting weight sets attached to the single or multichannel arrays. The criterion is to minimise weighted sum of squares or Mean Square Error (MSE).

Least Mean Square Algorithm (LMS) is stochastic gradient technique based on the particular geometry of the error or the cost function to be minimized. The Mean Square Error (MSE) criterion can be formulated by estimating the squared of the error.

$$J_{(\boldsymbol{w})} = E\left[e(t)e(t)^*\right] \tag{2.33}$$

$$J_{(\boldsymbol{w})} = E\left[\left\{d(t) - y(t)\right\}\left\{d(t) - y(t)\right\}^*\right] \tag{2.34}$$

$$J_{(\boldsymbol{w})} = E\left[\left\{d(t) - \boldsymbol{w}^H \boldsymbol{x}(t)\right\}\left\{d(t) - \boldsymbol{w}^{\mathrm{H}} \boldsymbol{x}(t)\right\}^*\right] \tag{2.35}$$

$$J_{(\boldsymbol{w})} = \sigma_d^2 - \boldsymbol{w}^{\mathrm{H}} \boldsymbol{p} - \boldsymbol{p}^H \boldsymbol{w} + \boldsymbol{w}^{\mathrm{H}} \mathbf{R} \boldsymbol{w} \tag{2.36}$$

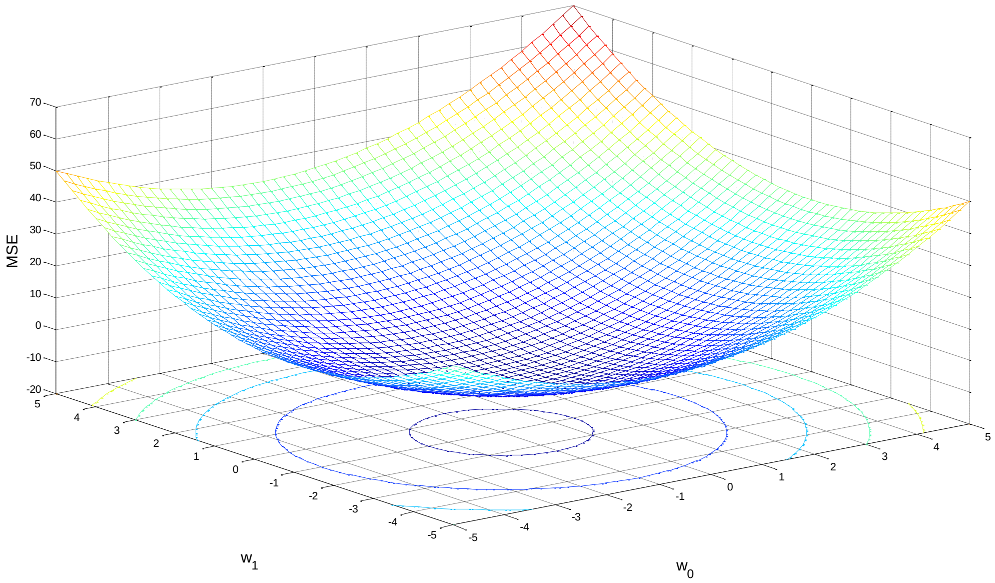

Figure 2.7: Cost function surface based on two element weight vector

Where $\sigma_d^2 = E\left[d(t)d(t)^*\right]$ is the covariance of the desired signal, $\boldsymbol{p} = E\left[\boldsymbol{x}(t)d(t)^*\right]$ is the cross correlation between desired signal and observation and $\mathbf{R} = E\left[\boldsymbol{x}(t)\boldsymbol{x}(t)^H\right]$ is the covariance matrix of the sensor data respectively.

Finding the weight vector that minimises the MSE is a standard adaptive filtering problem. In this case the cost function $J_{(\boldsymbol{w})}$ is a quadratic function of $\boldsymbol{w}$ and has a shape of a paraboloid for two element weight vector case as depicted in Figure 2.7.

Minimum point of the MSE surface is that where the gradient of the cost function with respect to weight vector $\boldsymbol{w}$ vanishes i.e.

$$\frac{\partial J_{(\boldsymbol{w})}}{\partial \boldsymbol{w}} = 0 \tag{2.37}$$

Hence

$$\frac{\partial J_{(\boldsymbol{w})}}{\partial \boldsymbol{w}} = -\boldsymbol{p} + \mathbf{R}\boldsymbol{w} = 0 \tag{2.38}$$

Therefore the optimum weight vector which yields minima is given as

$$\boldsymbol{w}_{opt} = \mathbf{R}^{-1}\boldsymbol{p} \tag{2.39}$$

The solution in Eq. 2.39 is the Wiener Hopf [25] solution for the adaptive filters. In most of the applications the parameters of quadratic search surface are unknown. The location of points on the surface, however, can be measured or estimated by averaging the squared error over a period of time. The problem is to develop systematic procedures or algorithms capable of searching the performance surface and finding the optimal solution.

2.6 Adaptive Algorithms

Cost function minimization can be performed through adaptive algorithms, the most simple of which is Least Mean Square (LMS) adaptive algorithm [32]. However an ideal adaptive algorithm

- Must converge to the optimum filter (Wiener filter)
- Should have fast convergence rate
- Must be able to track changes in non-stationary conditions
- Not too dependent on the input signals , be robust
- Be numerical robust
- Require minimum memory and computational resources.

LMS finds minimum or feasible point by an update rule involving successive corrections of the weight vector $\boldsymbol{w}$ in the direction of the negative gradient of the performance surface. Starting from the initial weight vector the process can be expressed as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \nabla J_{(\boldsymbol{w})} \tag{2.40}$$

Where $\nabla J_{(\boldsymbol{w})} = \dfrac{\partial J_{(\boldsymbol{w})}}{\partial \boldsymbol{w}}$ is the gradient of the cost function of Eq.2.36, therefore

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu\left(-\boldsymbol{p} + \mathbf{R}\boldsymbol{w}_n\right) \tag{2.41}$$

Using instantaneous values of the covariance matrix and cross correlation vector the steepest descent algorithm [32] can be written as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \mu e_n^* \boldsymbol{x}_n \tag{2.42}$$

The above equation is the well known LMS algorithm. Where the time index *n* is related to *t* such that $t=nT_s$, here T_s is the sampling time, thus

$$\boldsymbol{x}_n = \boldsymbol{x}[n] = \boldsymbol{x}[\frac{t}{T_s}]$$

Please note that we will alternatively use these notations throughout the thesis wherever appropriate. The solution requires availability of reference signal for the calculation of error, though it has avoided the requirement of covariance matrix **R**. The convergence of LMS is highly dependent upon the eigenvalue spread of the **R** matrix. If the eigenvalues are widely spread the convergence is slow and vice versa. The convergence and stability also depends upon the correct choice of the step size [32]. A choice of small and larger step size can lead to slow but accurate and fast but inaccurate solution respectively. Thus a trade-off exists between convergence and steady state mean square error. Therefore for the stability of algorithm step size must satisfy the inequality given by

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{2.43}$$

Where $\lambda_{\max}$ is the maximum eigenvalue of the covariance matrix **R**. The step size can be normalised to ensure an approximately constant rate of adaptation to avoid slow convergence in non-stationary environments.

$$\mu = \frac{\mu_o}{\boldsymbol{x}_n{}^H \boldsymbol{x}_n} \tag{2.44}$$

Inserting this new value of step size in Eq. 2.42 yields

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \frac{\mu_o e_n^* \boldsymbol{x}_n}{\boldsymbol{x}_n{}^H \boldsymbol{x}_n} \tag{2.45}$$

The Eq. 2.45 is the Normalised Least Mean Square (NLMS) [33] algorithm for adaptive minimization of the cost function described in Eq. 2.36, where μ_o is the new step size. The critical drawback of LMS is that it is sensitive to the scaling of its input, which makes it hard to choose step size which can guarantee the stability of the algorithms. NLMS addresses this through normalised step size and provides improved power control. LMS has also limitations over it dynamic range [34].

The Recursive Least Square (RLS) [35] algorithm is another class of adaptive algorithms where the cost function to be minimized is sum of squared errors. The algorithm is faster than the LMS provided that the SNR is high, but has forgotten factor which is very much dependent on channel fading rate. The complexity of RLS is higher due to requirement of matrix inversion involved in optimisation procedure. However, the complexity of inversion of a matrix can be reduced through Woodbury Identity [36].

Sample Matrix Inversion (SMI) algorithm is a block adaptive technique [37] which has very fast convergence but at the cost of increase computational complexity. Since SMI employs direct matrix inversion the convergence of this algorithm is much faster compared to the LMS algorithm. However, huge matrix inversions lead to computational complexities that cannot be easily overcome. It is also numerically unstable at times.

There is plethora of adaptive algorithms [2-3, 12, 25, 37-40] available for the weight vector adaptation but most commonly they require desired signal a priori, which is rarely known. To address this issue Griffiths [41] proposed a modification in the LMS array weight vector adaptation of Eq.2.41 such that it can be written as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \mu(\boldsymbol{p} - \boldsymbol{x}_n \boldsymbol{x}_n{}^H \boldsymbol{w}_n) \tag{2.46}$$

This can also be written as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \mu(\boldsymbol{p} - y_n^* \boldsymbol{x}_n) \tag{2.47}$$

Where $y_n{}^* = \boldsymbol{x}_n^H \boldsymbol{w}_n$

This modification has eased the restriction of prior knowledge for the direction and magnitude of the desired signal. Although it no more requires reference signal but it is assumed that the cross correlation is known or can be estimated. Another issue with the adaptive algorithms is that while minimising the error their output power decreases in the desired direction. To prevent this to happen it is required to place constraint on the output of the array towards the desired direction to be maximum while defining the array beam pattern.

2.7 Constraint Algorithms

The techniques discussed above require significant prior knowledge about the desired signal while in reality most practical applications do not have this type of information. However if we know the Direction of Arrival (DoA) of the desired signal and the desired frequency response we can impose some constraints on the array coefficients to adaptively minimise the array output in all direction but the direction of arrival. This leads to well known Linearly Constrained Minimum Variance(LCMV) beamformer proposed by Frost [24, 29]. The only prior information the algorithm requires is the direction and frequency band of interest. Consequently, the algorithm minimises total noise power at the array output while maintaining a chosen frequency response in the "look direction". Let us consider Eq. 2.10 and Eq. 2.11 again to gain more insight into the look direction

$$\boldsymbol{x}(t)=s(t)\boldsymbol{a}(\theta)+\boldsymbol{v}(t)$$

The individual elements of the array contain replicas of the signal $s(t)$ with different phase shifts corresponding to differences in propagation times between elements. Ideally, the signals from the M array sensors are added coherently, which requires that each of the relative phases be zero at the point of summation; that is, we add $s(t)$ with a perfect replica of itself. Thus, we need a set of complex weights that result in a perfect phase alignment of all the sensor signals. The resultant weight vector that phase-aligns a signal from direction θ_d at different array elements is simply the array response vector in that direction [2]. This is a kind of spatial match filtering since the steering vector is matched to the array response of signals impinging on the array from an angle θ_d. As a result, θ_d is known as the "look direction". Thus a spatial matched filter maximises the signal in "look direction" or implements the desired frequency response known priori. Let us consider the broadband array in Figure 2.4, if the "look direction" is chosen as broadside to the line of sensors, then radiating waveforms impinging from the look direction are incident in-phase on all the array elements. Therefore as far as the "look direction" is concerned the processor for the broadband array of Figure 2.4 equates a single tapped delay line where each tap weight of the delay line is the sum of the delay weights in corresponding column of the filter. Figure 2.8 depicts an equivalent processor for the look direction signal. If the look direction is not at $\theta=0^o$ then the array response will be forced either electronically or mechanically such that

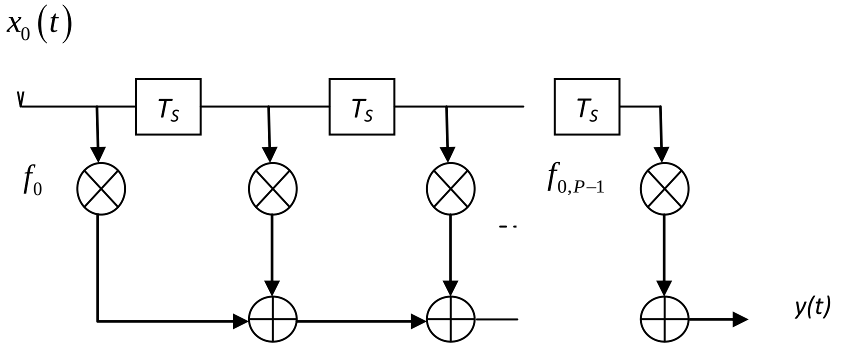

Figure 2.8: Equivalent processor for look direction signal

the signals incident on the array from the directions of interest other than the broadside appear as replicas of one another at the output of the array. However we will show in the broadband wave-number solution proposed in this thesis that by intelligently devising the required frequency response the pre-steering delays can be avoided. Thus the weight vector formulation for the look direction signal can be casted as

$$\mathbf{C}^T \boldsymbol{w} = \boldsymbol{f} \tag{2.48}$$

Eq. 2.48 is called the constraint equation. It imposes linearly independent constraints on the weight vector $\boldsymbol{w}$. Where $\boldsymbol{f}$ is the response vector set of constraints and is given by

$$\boldsymbol{f} = [\,f_o \quad f_1 \quad f_2 .... \quad f_{P-1}]^T \tag{2.49}$$

A single element of the above response vector is the sum of corresponding weight column and is given by

$$f_j = \sum_{m=0}^{M-1} w_{m,j} \tag{2.50}$$

The *MP*x*P* matrix **C** has a special form and is given by

$$\mathbf{C} = \begin{bmatrix} \boldsymbol{c}_d & & \boldsymbol{0} \\ & \ddots & \\ \boldsymbol{0} & & \boldsymbol{c}_d \end{bmatrix} \tag{2.51}$$

where *M*x*1* vector $\boldsymbol{c}_d = [1 \quad 1 \quad 1.... \quad 1]^T$ is the vector of “1s”. For Minimum Variance

Distortion-less Response (MVDR), the response vector in Eq. 2.49 becomes all zeros except one element equal to 1. The LCMV problem can be formulated as;

Minimise output power of the array subject to the constraint of Eq. 2.48 i.e.

Minimise $\boldsymbol{w}^H\mathbf{R}\boldsymbol{w}$ subject to $\mathbf{C}^T\boldsymbol{w} = \boldsymbol{f}$ (2.52)

The solution can be formulated with the help of Lagrange multipliers [42] as

$$J_{c(w)} = \boldsymbol{w}^H\mathbf{R}\boldsymbol{w} + \boldsymbol{\lambda}^T\left(\mathbf{C}^T\boldsymbol{w} - \boldsymbol{f}\right) \tag{2.53}$$

Where $J_{c(w)}$ is the cost function associated with constraint and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. Differentiating Eq. 2.53 w.r.t $\boldsymbol{w}$.

$$\frac{\partial J_{c(w)}}{\partial \boldsymbol{w}} = \mathbf{R}\boldsymbol{w} + \mathbf{C}\boldsymbol{\lambda} \tag{2.54}$$

The solution can be obtained by putting Eq. 2.54 equal to zero, hence

$$\boldsymbol{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}\boldsymbol{\lambda} \tag{2.55}$$

Using the constraint Eq. 2.48, the Eq. 2.55 can be written as the optimal solution

$$\boldsymbol{w}_{opt} = \mathbf{R}^{-1}\mathbf{C}\left(\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)\boldsymbol{f} \tag{2.56}$$

Eq. 2.56 is the optimal solution to the LCMV problem in Eq. 2.53. If we replace the gradient in Eq. 2.40 with the one obtained in Eq. 2.54, after incorporating weight constraint, we can formulate an iterative solution for the weight update.

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu\left[\mathbf{R}\boldsymbol{w}_n + \mathbf{C}\boldsymbol{\lambda}\right] \tag{2.57}$$

Since $\boldsymbol{w}_{n+1}$ must satisfy the constraint in Eq. 2.48 hence we can write

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu\left(\mathbf{I} - \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\right)\mathbf{R}\boldsymbol{w}_n + \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\left(\boldsymbol{f} - \mathbf{C}^T\boldsymbol{w}_n\right) \tag{2.58}$$

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\mathbf{I} - \mu\mathbf{R}\right]\boldsymbol{w}_n + \boldsymbol{k}_c \tag{2.59}$$

The Eq. 2.59 is the Constrained Least Mean Square (CLMS) algorithm for adaptive array processing. Where $\boldsymbol{k}_c = \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\boldsymbol{f}$ and matrix $\mathbf{P}$ is the projection operator and is given by

$$\mathbf{P} = \left(\mathbf{I} - \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T\right) \tag{2.60}$$

A stochastic version of the above algorithm proposed by Frost [24] is given as

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\boldsymbol{w}_n - \mu y_n^* \boldsymbol{x}_n\right] + \boldsymbol{k}_c \tag{2.61}$$

The constrained adaptive processing algorithm presents a flexible solution where the output response is directly controlled using constraint vector and a set of linear constraints. The calculation of optimum weight vector has been generally considered as a challenge in the subject area, due to attached requirement of inversion of the covariance matrix. Although the stochastic gradient approach eliminates the need of inversion but still it requires either the desired response or the cross correlation vector to calculate the optimum weight vector.

2.8 Conclusion

Sensor arrays are an important and indispensable component of signal processing. Complemented with a fixed or adaptive processor, antenna array can discriminate between wanted and unwanted signals in the presence of noise. The constrained adaptive processor optimises the array response according to defined set of spatial constraints such that the output contains the minimal contribution of noise and signals originating from undesired directions. The essence of an adaptive processor is the algorithm comprising of logical finite set of mathematical operations. There is plethora of adaptive algorithms available for the weight vector adaptation but most commonly they require desired signal a priori, which is rarely known. However if the Direction of Arrival (DoA) of the desired signal and the desired frequency response is chosen a priori, we can impose some constraints on the array coefficients to adaptively minimize the array output in all directions but the direction of arrival of the desired signal. The only a priori information the algorithm required is the direction and frequency band of interest. Consequently, the algorithm minimises total noise power at the array output while maintaining a chosen frequency response in the “look direction”.

In this chapter we have reviewed basic adaptive processing techniques. In next Chapter 3 we will formulate the novel natural gradient based algorithms for the constrained adaptive array processing problem discussed herein.

Chapter 3

Constrained Adaptive Natural Gradient Algorithm (CANA) for Adaptive Array Processing

In previous chapters we have established the research question. Some of the pioneering techniques were also reproduced in order to address the subject issue. This chapter presents, novel natural gradient based algorithms for the constrained adaptive array processing problem. Novel Constrained Adaptive Natural Gradient Algorithm (CANA) is capable of adjusting the response of an array of sensors to a signal coming from Direction of Interest (DoI) and suppressing interferers coming from other directions. Constrained optimisation techniques have been developed to enhance the look direction signal in the presence of additive noise and interference using adaptive natural gradient techniques. The algorithms have been described by elaborating concepts of Riemannian metrics and differential geometry-on which the algorithms are based. Mathematical development of the techniques is given along with numerical design examples, results and critical analysis. Mathematical analysis and realistic MATLAB simulations confirm the suitability of algorithms in calculating iterative sensor weights for adaptive beamforming.

3.1 Introduction

Constrained optimisation has found many applications in signal processing as well as for solving control engineering problems involving minimization of a cost function [43]. The realistic minimization of these cost function problems is not trivial and generally involves manifolds with non linear optimisation surfaces [15]. Working with a search space that carries the nonlinear manifolds, introduces certain challenges such as choice of step size and appropriate direction towards the feasible point. Numerical computations also require that the solution set consists only of the isolated points in optimisation domain. The purpose is well served by imposing constraints [10]. The field of constraint analysis has been well established over the last three decades with contributions from a diverse community of researchers in artificial intelligence, databases and programming languages, operations research, management science, and applied mathematics [10, 29]. In a constrained optimisation problem the algorithm has to constantly observe the imposed constraints while maximising or minimising the cost function.

Parameter estimation is central to several scientific, engineering and financial applications, and the most popular technique used, perhaps due to its simplicity is the gradient descent technique. Although its ability to converge quickly and efficiently is limited by certain factors, yet the method has numerous applications in signal processing and control applications. The technique yields optimum performance when a) the cost function has a single minimum and b) the gradients of the cost function are isotropic in magnitude with respect to any direction away from the minimum [16].

On the other hand the applications in signal processing, mechanics and control theory quite often involve optimisation of the manifolds endowed with a metric structure, i.e. Riemannian. In such constrained problems, the super-linear convergence speed of the classical conjugate gradient and Newton algorithms is lost as the structure of the surface is ignored. The convergence speed of a gradient adaptation can be slow when the slope of the cost function varies widely for the small changes in the adjusted parameters. Conventional gradient based approaches undergo this phenomenon hence attain slow convergence. Any optimisation scheme that exploits the given structure of the underlying space does not encounter such phenomenon and is deemed successful with better convergence and accuracy [15].

Natural gradient changes the search direction according to the Riemannian structure of the parameter space. Natural gradient assumes the surface to be locally Euclidean and when used with statistically optimum cost functions, it overcomes many of the limitations of the Newton’s method, which assumes that cost function being minimized is approximately quadratic.

The major drawback of natural gradient is the knowledge required to determine the Riemannian structure of the parameter space, although it can be quite simple for certain applications [15], as described in CANA-NAM.

3.2 Standard Gradient Adaptation

Let us consider a scalar cost function $f(\boldsymbol{w})$ defined by a set of parameters given by

$$\boldsymbol{w} = \left[w_0 w_2 \cdots w_{M-1}\right]^H \tag{3.1}$$

If the cost function is smooth i.e. twice differentiable with respect to w_i then there exists a solution vector $\boldsymbol{w}^o$ given by

$$\boldsymbol{w}^o = \left[w_0^o w_1^o w_2^o \cdots w_{M-1}^o\right]^H \tag{3.2}$$

For which the partial derivative of the cost function with respect to the vector defined in Eq.3.1 is zero, that is

$$\frac{\partial f\left(\boldsymbol{w}^o\right)}{\partial \boldsymbol{w}} = 0 \tag{3.3}$$

and if the cost function has a positive definite Hessian matrix-the square matrix of its second order partial derivatives, then the weight vector represents local minimum of the parametric space $f(\boldsymbol{w})$. The gradient in Eq. 3.3 can be defined as below.

$$\frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}} = \left[\frac{\partial f(\boldsymbol{w})}{\partial w_0} \frac{\partial f(\boldsymbol{w})}{\partial w_1} \quad \frac{\partial f(\boldsymbol{w})}{\partial w_2} \cdots \quad \frac{\partial f(\boldsymbol{w})}{\partial w_{M-1}}\right]^H \tag{3.4}$$

It may be noted that a gradient is a vector field that for a given point of weight vector $\boldsymbol{w}$, points in the direction of the greatest increase of the cost function $f(\boldsymbol{w})$. Iterative procedures are based upon using the scaled version of this component to be subtracted from the previously calculated weight vector.

The standard gradient or steepest descent is a method of iterative minimization of the cost function $f(\boldsymbol{w})$.

The method can be formulated as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \nabla f\left(\boldsymbol{w}\right) \tag{3.5}$$

Where $\nabla f\left(\boldsymbol{w}\right) = \frac{\partial f\left(\boldsymbol{w}\right)}{\partial \boldsymbol{w}}$ is the tangential gradient of the cost function with respect to the weight vector ***w*** while μ is the step size. To start the process, the technique needs a predefined initial value as a starting point. At each iteration the algorithm subtracts a scaled value of tangential gradient from the calculated weight vector to search out the new value. This process is repeated until the tangential gradient vanishes or reaches a very small but stable value. When this steady state is reached the algorithm is said to have converged. At convergence the updated weight vector represents the optimum weight vector and hence represents a local minimum. It is important to note here that the convergence of the steepest descent technique is critically dependent upon the choice of step size. A large value of μ can lead to quick but imprecise results while a small value can drastically reduce the convergence speed though can lead to an accurate solution. Generally it is difficult or impossible to choose step size in order to provide fast convergence from all initial values of the weight vector ***w***. This is due to the fact that the individual components of gradient vector inconsistently vary in magnitude in different directions from the optimum value. So it suffers from poor convergence properties and may lead to inaccuracy at times.

3.3 Natural Gradient Adaptation

Natural gradient is formed by augmenting conventional tangential gradient with the Riemannian metric, calculated over the surface to be optimised. Riemannian metric in contrast to conventional metrics being abstraction of the notion of distance, is an inner product on continuous tangent spaces. There is however a link since any Riemannian metric induces a distance, the Riemannian distance [15, 44]. Nevertheless the metric tensor is used to calculate the actual distance on the Riemannian surface. The rest of the geometrical calculations arise from the curvature of the surface to be optimised. A curvature vector indicates the direction in which a certain curve under inspection is turning. Optimisation i.e. finding minimum or maximum point on the manifold involves displacing the tangential vector from its base to the new point and then subtracting the normal component. At each point of the manifold of Figure 3.1a and 3.1b, we can find a tangent vector in each direction. The set of such vectors constitute

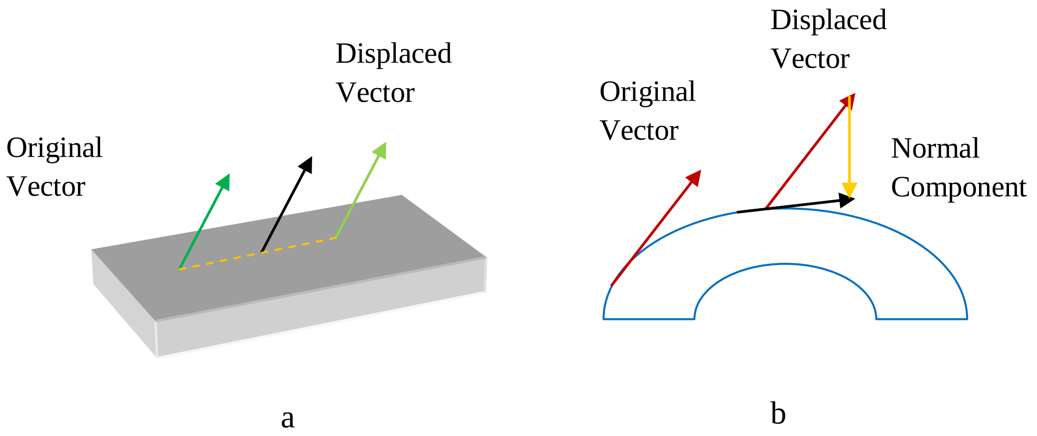

Figure 3.1: Parallel transpositions along a) plane and b) curved manifolds

a tangent bundle. In Euclidean spaces, tangents are naturally isometric hence ideal for conventional gradient based optimisation. As depicted in Figure 3.1a, moving one vector from a point to another does not require more than the original point and end point of the displacement. There is a flat connection between the consecutive movements. Due to curvature such phenomenon does not hold true on the Riemannian manifolds. In this situation, moving on these surfaces require the normal component of the tangent vector to be subtracted from the new tangent vector at each move. The operation of creating the new tangent vector at the new point is referred to as parallel transposition as depicted in Figure 3.1a and 3.1b.

For every Riemannian manifold there exists a unique and intrinsic connection “Levi-Civita” defined by “Christoffel coefficients” in terms of the metric tensor elements, g_{ij} [44]. Description of Christoffel symbols and their calculation is beyond the scope of this thesis. Nevertheless Christoffel symbols/coefficients carry all the information about the curvature of the manifold and thus enable movements on the tangential space. Through these Levi-Civita connections, geodesics-the curves of minimum length [45], are formed in order to perform movements over the surface [46] and consequently the optimisation. Thus on a Riemannian manifold, for motion of a vector $\boldsymbol{w}$ to $\boldsymbol{w}+\Delta\boldsymbol{w}$, where $\Delta\boldsymbol{w}$ is an incremental vector of very small magnitude, the distance metric can be defined as [17, 47]

$$d_w\left(\boldsymbol{w},\boldsymbol{w}+\Delta\boldsymbol{w}\right)=\sqrt{\Delta\boldsymbol{w}^H\mathbf{G}\left(\boldsymbol{w}\right)\Delta\boldsymbol{w}} \tag{3.6}$$

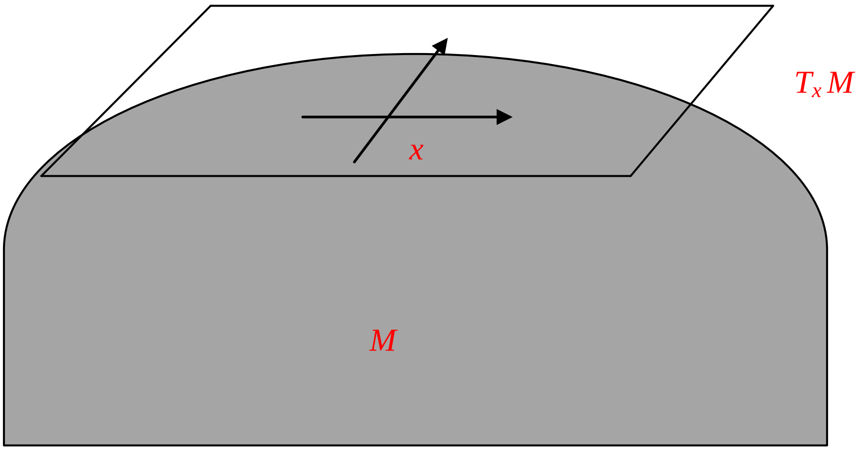

Figure 3.2: A manifold *M* with typical tangent space T_xM at point *x*

Wherein $\mathbf{G}(\boldsymbol{w})$ is the Riemannian metric. Essentially a positive definite matrix, the Riemannian metric carries the intrinsic curvature of a particular manifold. If $\boldsymbol{w}$ is the vector of parameters defining the cost function $f(\boldsymbol{w})$ then any change from $\boldsymbol{w}$ to $\boldsymbol{w}+\Delta\boldsymbol{w}$ reflect the change in cost function from $f(\boldsymbol{w})$ to $f(\boldsymbol{w}+\Delta\boldsymbol{w})$ and hence its gradient with respect to the parameter of the surface. With the assumption of the distance metric to be very small the direction vector can be defined as

$$\Delta\boldsymbol{w}=\mathbf{G}^{-1}(\boldsymbol{w})\nabla f(\boldsymbol{w}) \tag{3.7}$$

Where $\nabla f(\boldsymbol{w})=\frac{\Delta f(\boldsymbol{w})}{\Delta\boldsymbol{w}}=\frac{\partial f(\boldsymbol{w})}{\partial\boldsymbol{w}}$ is the conventional tangential gradient and $\mathbf{G}(\boldsymbol{w})$ is the

Riemannian metric. Wherein a manifold as depicted in Figure 3.2, is a topological surface which is locally Euclidian [48]. If for every point in a manifold , an inner product is defined on its tangent space; a space carrying all possible tangent vectors at that point, then the Riemannian metric is a collection of all these inner products [49]. As described earlier the iterative minimization of the cost function $f(\boldsymbol{w})$ can be defined as

$$\boldsymbol{w}_{n+1}=\boldsymbol{w}_n-\mu\Delta\boldsymbol{w}_n \tag{3.8}$$

Using values of $\Delta\boldsymbol{w}$ from Eq. 3.7 the, natural gradient based adaptation in the descent direction is defined as

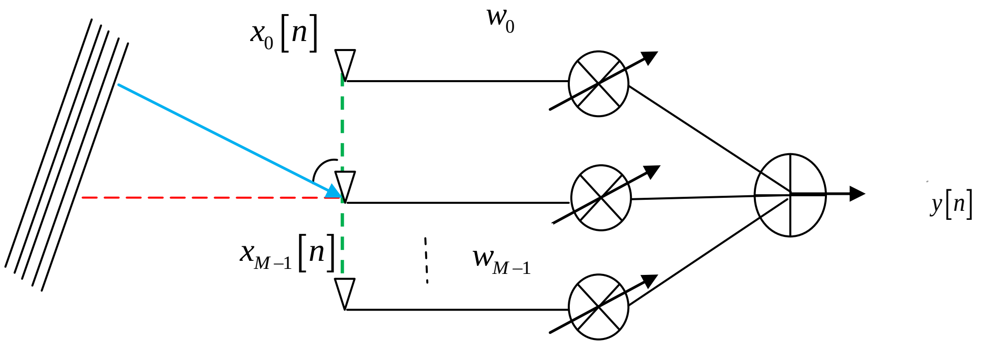

Figure 3.3: Uniform linear array

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \mathbf{G}^{-1}(\boldsymbol{w}) \nabla f(\boldsymbol{w}) \tag{3.9}$$

So for the curved spaces the natural gradient based search method as described above uses explicit knowledge of Riemannian distance structure hence provides better and faster approximation to the steepest descent direction. It may be noted that for flat spaces the Riemannian metric is Identity matrix, in that case, the distance metric of Eq. 3.6 represents a Euclidian metric i.e.

$$d_w(\boldsymbol{w}, \boldsymbol{w} + \Delta\boldsymbol{w}) = \sqrt{\Delta\boldsymbol{w}^H \Delta\boldsymbol{w}}$$

Hence $\Delta\boldsymbol{w} = \nabla f(\boldsymbol{w})$, causing Eq. 3.9 to reduce to the standard gradient case of Eq. 3.5. Another variation can be obtained if we replace Riemannian metric with the Hessian [16] of the cost function $f(\boldsymbol{w})$.

If $\nabla f(\boldsymbol{w}) = \dfrac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}$ is the gradient of the cost function $f(\boldsymbol{w})$ then its Hessian matrix can be defined as

$$\mathbf{H}(\boldsymbol{w}) = \nabla f(\boldsymbol{w}) \nabla f(\boldsymbol{w})^H \tag{3.10}$$

Inserting negative of this value in the update Eq. 3.8 an already known algorithm emerges called Newton method [50] and is given as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \mathbf{H}^{-1}(\boldsymbol{w}) \nabla f(\boldsymbol{w}) \tag{3.11}$$

The algorithm developed thus looks similar to the one based upon natural gradient but due to the fact that Hessian matrix is not always positive definite the technique suffers from poor convergence properties and at times exhibits wrong approximations. This is due to the fact that if the Hessian is not positive definite the direction vector based upon this value would not point towards the increase of the cost function. A negative of this vector would not point towards minimum and will result in wrong approximation or fluctuation around a certain value.

3.4 Constrained Adaptive Natural Gradient Algorithm (CANA)

In array processing, constrained optimisation technique is aimed at minimising the total noise power at the array output while maintaining a chosen frequency response in the Direction of Interest (DoI) [2]. The only a priori information the algorithm requires is the desired Direction of Interest (DoI). With the help of concepts derived in previous sections we will formulate the optimisation technique based upon natural gradient algorithm. We call this technique Constrained Adaptive Natural Gradient Algorithm (CANA).

Consider a conventional narrowband adaptive array processor as depicted in Figure 3.3 with the array output y_n, given by the following weight relationship.

$$y_n = \boldsymbol{w}_n^H \boldsymbol{x}_n \tag{3.12}$$

Where $\boldsymbol{w}$ is the vector of complex weights attached behind each sensor and $\boldsymbol{x}_n$ is so called data vector composed of signals from all sources including interference and noise at time *n*. It is required that the array processor should optimally respond to the signal coming from a desired direction while discriminating against the interferers coming from the other directions. The objective here is to minimise the average output power $E\left[\left|y_n{}^2\right|\right]$ with some look direction constraint; that the processor should produce maximum response in the look direction only, while rejecting interferers from all other directions. The look direction constraint can be formulated as [2, 24, 28]

$$\mathbf{c}^T \boldsymbol{w}_n = f_c \tag{3.13}$$

Where $\mathbf{c}$ is the constraint vector and f_c is scalar constraint to be observed by the processor. For the narrowband array, if we let the look direction be incident broadside to the array then the constraint vector $\mathbf{c}$ becomes a unit vector.

The minimization of average output power in Eq. 3.12 is to minimise the cost function $J_{(\boldsymbol{w})}$ subject to the constraint of Eq. 3.13. The unconstrained cost function is defined as

$$J_{(\boldsymbol{w})} = \ E\left[\boldsymbol{w}_n^H \boldsymbol{x}_n \boldsymbol{x}_n^H \boldsymbol{w}_n\right] \tag{3.14}$$

The Lagrange method can be used to incorporate the constraint of Eq. 3.13 into the cost function of Eq. 3.14 which has to be minimised with respect to the weight vector $\boldsymbol{w}_n$. Hence Eq. 3.14 takes the form

$$J_{c(\boldsymbol{w})} = \boldsymbol{w}_n^H \mathbf{R} \boldsymbol{w}_n + \lambda\left(\mathbf{c}^T \boldsymbol{w}_n - f_c\right) \tag{3.15}$$

Where $\mathbf{R}$ is the correlation matrix of the input data vector $\boldsymbol{x}_n$ and λ is the Lagrange multiplier. Minimising of the cost function requires its gradient to be set to zero. As derived above and also proposed by Amari et. al [15], the natural gradient of a cost function $J_{(\boldsymbol{w})}$ can be written as

$$\tilde{\nabla} J_{(\boldsymbol{w})} = \mathbf{G}^{-1}\left(\boldsymbol{w}\right) \nabla J_{(\boldsymbol{w})} \tag{3.16}$$

Where $\tilde{\nabla} J_{(\boldsymbol{w})}$ is the natural gradient and $\nabla J_{(\boldsymbol{w})}$ is the conventional gradient of the cost function and $\mathbf{G}\left(\boldsymbol{w}\right)$ is the Riemannian metric respectively. The conventional gradient for the problem under discussion has already been defined in Eq. 2.55 as

$$\nabla J_{c(\boldsymbol{w})} = \mathbf{R} \boldsymbol{w}_n + \lambda \mathbf{c} \tag{3.17}$$

While formulating 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 optimised. As described earlier, since flat surfaces and optimisation spaces, carry no curvature, therefore in those cases Riemannian metric is the identity matrix. From Eq. 3.15 the constrained cost function can be derived again as

$$J_{c(\boldsymbol{w})} = J_{(\boldsymbol{w})} + \lambda\left(\boldsymbol{c}^T\boldsymbol{w} - f_c\right) \tag{3.18}$$

Naturally it can be construed that

$$J_{c(\boldsymbol{w}+\Delta\boldsymbol{w})} = J_{c(\boldsymbol{w})} + \Delta J_{c(\boldsymbol{w})} \tag{3.19}$$

Where $J_{c(\boldsymbol{w})}$ is the constrained cost function, as described in Eq. 3.18, if we let $\Delta J_{c(\boldsymbol{w})} = \varepsilon\left\|\Delta\boldsymbol{w}\right\|^2$ the Eq. 3.19 takes the form

$$J_{c(\boldsymbol{w}+\Delta\boldsymbol{w})} = J_{c(\boldsymbol{w})} + \varepsilon\left\|\Delta\boldsymbol{w}\right\|^2 \tag{3.20}$$

but

$$J_{c(\boldsymbol{w}+\Delta\boldsymbol{w})} = J_{(\boldsymbol{w}+\Delta\boldsymbol{w})} + \lambda\left[\boldsymbol{c}^T\left(\boldsymbol{w}+\Delta\boldsymbol{w}\right) - f_c\right]$$

Therefore

$$J_{(\boldsymbol{w}+\Delta\boldsymbol{w})} + \lambda\left[\boldsymbol{c}^T\left(\boldsymbol{w}+\Delta\boldsymbol{w}\right) - f_c\right] = J_{(\boldsymbol{w})} + \lambda\left(\boldsymbol{c}^T\boldsymbol{w} - f_c\right) + \Delta J_{c(\boldsymbol{w})}$$

or

$$J_{(\boldsymbol{w}+\Delta\boldsymbol{w})} + \lambda\left[\boldsymbol{c}^T\Delta\boldsymbol{w}\right] = J_{(\boldsymbol{w})} + \Delta J_{c(\boldsymbol{w})} \tag{3.21}$$

$$J_{(\boldsymbol{w}+\Delta\boldsymbol{w})} - J_{(\boldsymbol{w})} + \lambda\left[\boldsymbol{c}^T\Delta\boldsymbol{w}\right] = \Delta J_{c(\boldsymbol{w})} = \varepsilon\left\|\Delta\boldsymbol{w}\right\|^2$$

or

$$\Delta J_{(\boldsymbol{w})} + \lambda\boldsymbol{c}^T\Delta\boldsymbol{w} = \varepsilon\left(\Delta\boldsymbol{w}\right)^H \mathbf{G}\left(\boldsymbol{w}\right)\left(\Delta\boldsymbol{w}\right) \tag{3.22}$$

Where $\left\|\Delta\boldsymbol{w}\right\|^2 = \left(\Delta\boldsymbol{w}\right)^H \mathbf{G}\left(\boldsymbol{w}\right)\left(\Delta\boldsymbol{w}\right)$ and ε is an arbitrary small number which can be absorbed in the step size μ . From Eq. 3.22, therefore

$$\frac{\Delta J_{(\boldsymbol{w})}}{\Delta\boldsymbol{w}} = -\lambda\boldsymbol{c} + \mathbf{G}\left(\boldsymbol{w}\right)\Delta\boldsymbol{w} \tag{3.23}$$

Hence $\Delta \boldsymbol{w} = \mathbf{G}^{-1}(\boldsymbol{w})(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}}) + \lambda \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}$ (3.24)

Then the iterative weight update equation can be reproduced again

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \Delta \boldsymbol{w}_n \tag{3.25}$$

Inserting the value of direction vector $\Delta \boldsymbol{w}$ the weight update can be formulated as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \left(\mathbf{G}^{-1}(\boldsymbol{w}) \left(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}_n} \right) + \lambda \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c} \right)$$

or it can also be written as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu \left(\mathbf{G}^{-1}(\boldsymbol{w}) \left(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}_n} \right) \right) - \mu \lambda \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c} \tag{3.26}$$

However we know that the constraint is given as

$$\boldsymbol{c}^T \boldsymbol{w}_{n+1} = f_c$$

Pre-multiplying Eq. 3.26 with $\boldsymbol{c}^T$ leads

$$f_c = \boldsymbol{c}^T \boldsymbol{w}_{n+1} = \boldsymbol{c}^T \boldsymbol{w}_n - \mu \boldsymbol{c}^T \mathbf{G}^{-1}(\boldsymbol{w}) \left(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}_n} \right) - \mu \lambda \left[\boldsymbol{c}^T \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c} \right]$$

$$-\mu \lambda = \left[f_c - \boldsymbol{c}^T \boldsymbol{w}_n + \mu \boldsymbol{c}^T \mathbf{G}^{-1}(\boldsymbol{w}) \left(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}_n} \right) \right] / \left[\boldsymbol{c}^T \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c} \right] \tag{3.27}$$

Substituting this value in Eq. 3.26 we get

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

or

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

$$\boldsymbol{w}_{n+1} = \left[\boldsymbol{w}_n - \mu\left(\mathbf{G}^{-1}(\boldsymbol{w})\left(\frac{\Delta J_{(w)}}{\Delta \boldsymbol{w}_n}\right)\right)\right] - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\left[\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right]}\left[\boldsymbol{w}_n - \mu\mathbf{G}^{-1}(\boldsymbol{w})\left(\frac{\Delta J_{(w)}}{\Delta \boldsymbol{w}_n}\right)\right] + \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}f_c}{\left[\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right]}$$

or

$$\boldsymbol{w}_{n+1} = \left[\mathrm{I} - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}}\right]\left[\boldsymbol{w}_n - \mu\mathbf{G}^{-1}(\boldsymbol{w})(\frac{\Delta J_{(w)}}{\Delta \boldsymbol{w}_n}))\right] + \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}f_c}{\left[\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right]}$$

or

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\boldsymbol{w}_n - \mu\mathbf{G}^{-1}(\boldsymbol{w})\left(\frac{\Delta J_{(w)}}{\Delta \boldsymbol{w}_n}\right)\right] + \boldsymbol{k} \tag{3.28}$$

Wherein

$$\boldsymbol{k} = \mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\left[\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right]^{-1} f_c \quad , \qquad \mathbf{P} = \left[\mathbf{I} - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}}\right]$$

Now let us check the idempotent property of projection matrix by projecting it to itself that is by multiplying it with itself. Hence

$$\mathbf{PP} = \left[\mathbf{I} - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}}\right]\left[\mathbf{I} - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}}\right]$$

$$= \left[\mathbf{I} - 2\frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}} + \left(\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T\right)\frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\left(\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right)^2}\right]$$

$$\left[\mathbf{I} - 2\frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}} + \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\left(\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right)^2}\right]$$

It follows that projection matrix is given by

$$\mathbf{P} = \left[\mathbf{I} - \frac{\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\boldsymbol{c}^T}{\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}}\right]$$

The idempotent property of matrix $\mathbf{P}$ as derived above, confirms the suitability of matrix $\mathbf{P}$ to be the true projection operator for the developed Constrained Adaptive Natural Gradient

Algorithm (CANA). Where, $\mathbf{G}(\boldsymbol{w})$ the Riemannian metric, as described earlier, is a collection of inner products on continuous tangent spaces. The metric can also be realised from the parallel transposition on the curved surface defined by parameter *w*. As depicted in Figure 3.1b if tangential gradient of the surface is represented by $\nabla f(\boldsymbol{w})$ then natural tangential gradient is the resultant component if the normal component of $\nabla f(\boldsymbol{w})$ is subtracted from the displaced tangent vector at any point of the surface defined by the parameter *w*. Hence

$$\tilde{\nabla} J_{(\boldsymbol{w})} = \nabla f(\boldsymbol{w}) - \boldsymbol{w}\boldsymbol{w}^H \nabla f(\boldsymbol{w})$$

or

$$\tilde{\nabla} J_{(\boldsymbol{w})} = \left[\mathbf{I} - \boldsymbol{w}\boldsymbol{w}^H\right] \nabla f(\boldsymbol{w}) \tag{3.29}$$

Equating this with the relationship in Eq. 3.16 the inverse of Riemannian metric is given by

$$\mathbf{G}^{-1}(\boldsymbol{w}) = \left[\mathbf{I} \text{ - } \boldsymbol{w}\boldsymbol{w}^H\right] \tag{3.30}$$

To further exploit the convergence characteristics of the algorithm we have slightly modified the metric, resulting;

$$\mathbf{G}^{-1}(\boldsymbol{w}) = \left[\varepsilon\mathbf{I} \text{ - } \upsilon\boldsymbol{w}\boldsymbol{w}^H\right] \tag{3.31}$$

Wherein $\upsilon = [1-\varepsilon]$, we can also use $\upsilon = \left[\varepsilon / 1 + \varepsilon \mathbf{w}^H \boldsymbol{w}\right]$ ensuring more power control. The ε can be categorised as regularisation parameter with archetypal optimisation values. The inclusion of ε also avoids the rank deficiency. Thus with slight apparent increase in computational complexity, convergence performance and accuracy, which far exceeds the performance of the conventional beamformer, make the proposed algorithm very attractive in terms of implementations.

3.4.1 Convergence Analysis

On the basis of Eq. 3.28 we can define a bias vector $\boldsymbol{e}$ as follows

$$\boldsymbol{e}_{n+1} = \boldsymbol{w}_{n+1} - \boldsymbol{w}^o \tag{3.32}$$

Where $\boldsymbol{w}^o$ is optimum weight vector as given by Frost [24, 28] in his famous solution for linearly constrained adaptive array processing. From the Eq. 3.32 we can deduce that

$$\boldsymbol{w}_n = \boldsymbol{e}_n + \boldsymbol{w}^o \tag{3.33}$$

So for the particular cost function of Eq. 3.15, the Eq. 3.28 can be re-written as

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\mathbf{I} - \mu \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{w}_n + \boldsymbol{k} \tag{3.34}$$

At the steady state $\boldsymbol{w}_{n+1}$ and $\boldsymbol{w}_n$ approach to optimum weight vector $\boldsymbol{w}^o$ hence Eq. 3.34 can be written as

$$\boldsymbol{w}^o = \mathbf{P}\left[\mathbf{I} - \mu \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{w}^o + \boldsymbol{k} \tag{3.35}$$

Gradient part also vanishes once the optimum weight vector has been found hence

$$\boldsymbol{k} = \left[\mathbf{I} - \mathbf{P}\right]\boldsymbol{w}^o \tag{3.36}$$

From (3.33), (3.34) and (3.36)

$$\boldsymbol{e}_{n+1} = \mathbf{P}\left[\mathbf{I} - \mu \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{e}_n + \mathbf{P}\left[\mathbf{I} - \mu \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{w}^o + \boldsymbol{k} - \boldsymbol{w}^o \tag{3.37}$$

But for optimal solution

$$\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\boldsymbol{w}^o = 0$$

Therefore Eq. 3.37 reduces to

$$\boldsymbol{e}_{n+1} = \mathbf{P}\left[\mathbf{I} - \mu \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{e}_n \tag{3.38}$$

The idempotent property of projection matrix allows us to manipulate Eq. 3.38 by pre-multiplying it with the matrix $\mathbf{P}$. By doing so we get the same left hand side of Eq. 3.38, which means

$$\mathbf{P}\boldsymbol{e}_n = \boldsymbol{e}_n \quad \text{and} \quad \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\boldsymbol{e}_n = \mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}\boldsymbol{e}_n$$

With these realisations we can write Eq. 3.38 as

$$\boldsymbol{e}_{n+1} = \mathbf{P}\left[\mathbf{P} - \mu\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}\right]\boldsymbol{e}_n$$

$$\boldsymbol{e}_{n+1} = \left[\mathbf{I} - \mu\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}\right]^{n+1}\boldsymbol{e}_o \tag{3.39}$$

Where $\boldsymbol{e}_o$ is the initial value of the bias vector. The convergence of weight vector to the optimum weight vector along any eigenvector of $\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}$ is therefore governed by the convergence condition [43] as

$$-\frac{1}{\lambda_{\max}} \leq \mu \leq 0 \tag{3.40}$$

Where $\lambda_{\max}$ is the largest eigenvalue of the square matrix $\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}$. It may be noted that the trace value, $tr\left[\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}\right]$ is always more than the $M\lambda_{\max}$, where M is the dimension of matrix $\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}$.

In this scenario a suitable choice can be

$$-\frac{M}{tr\left[\mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\mathbf{P}\right]} \leq \mu \leq 0 \tag{3.41}$$

3.4.2 Computational Complexity Analysis

To analyze the efficiency of an algorithm, one way is to determine the amount of resources such as number of Multipliers and Accumulators (MACs) required to execute the algorithm. In order to measure the computational complexity of the CANA we will first analyse computational complexity of CLMS and then compare both of the complexities in terms of number of MACs. Thus final weight update equation for CLMS algorithm defined in Section 2.7 can be re-written as

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\mathbf{I} - \mu\mathbf{R}\right]\boldsymbol{w}_n + \boldsymbol{k}_c \tag{3.42}$$

For the length of weight vector $\boldsymbol{w}$ equal p and sizes of matrices **R** and **P** equal to *pxp*, the computational efforts required to execute each step in getting the final algorithm of Eq.3.42 are described in Table 3.1. Thus the total computational effort required for CLMS in terms of

Constrained Adaptive Algorithm	Computational Complexity[MACs]
CLMS	$\left[\mathbf{I}/\mu - \mathbf{R}\right]$ requires p MACs $\mu\left[\mathbf{I}/\mu - \mathbf{R}\right]\boldsymbol{w}$ requires p^2+p MACs $\mathbf{P}\left[\mathbf{I} - \mu\mathbf{R}\right]\boldsymbol{w}$ requires p^2 MACs $\mathbf{P}\left[\mathbf{I} - \mu\mathbf{R}\right]\boldsymbol{w}+\boldsymbol{k}_c$ requires p additions Total computational effort required = $2p^2$+$3p$
CANA	$\mathbf{G}^{-1}(\boldsymbol{w})=\left[\mathbf{I} - \boldsymbol{w}_n\boldsymbol{w}_n^H\right]$ requires p^2 MACs $\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}$ requires p^2+p MACs $\boldsymbol{k}=\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\left[\boldsymbol{c}^T\mathbf{G}^{-1}(\boldsymbol{w})\boldsymbol{c}\right]^{-1}$ requires p^2+p MACs $\mathbf{P} = \mathbf{I} - \boldsymbol{kc}^T$ requires p^2 MACs $\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\boldsymbol{w}$ requires $2p^2$ MACs $\left[\mathbf{I} - \mu\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{w}$ requires p MACs $\mathbf{P}\left[\mathbf{I} - \mu\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\right]\boldsymbol{w}+\boldsymbol{k}$ requires p^2+p MACs Total computational effort required = $7p^2$+$4p$

Table 3.1: Complexity analysis of constrained adaptive algorithms

MACs is $2p^2$+$3p$, where p is the number of array sensors. The CANA algorithm in addition to the similar computations as in CLMS also requires matrices **P** and $\mathbf{G}^{-1}(\boldsymbol{w})$ to be updated at every iteration, along with the vector $\boldsymbol{k}$. Thus the total computational effort required for the execution of CANA algorithm is $7p^2$+$4p$ as described in Table 3.1. Hence CANA is computationally more expensive than the CLMS algorithm. Table 3.1summarises the comparative computational complexity of both CANA and CLMS.

As p is the number of sensors hence for larger arrays the computational complexity of CANA increases sharply as compared to the computational complexity of the CLMS algorithm. Figure 3.4 represents complexities of two algorithms versus number of sensors, based on Table 3.1. But if we calculate the overall complexity in finding the steady state solution, say,

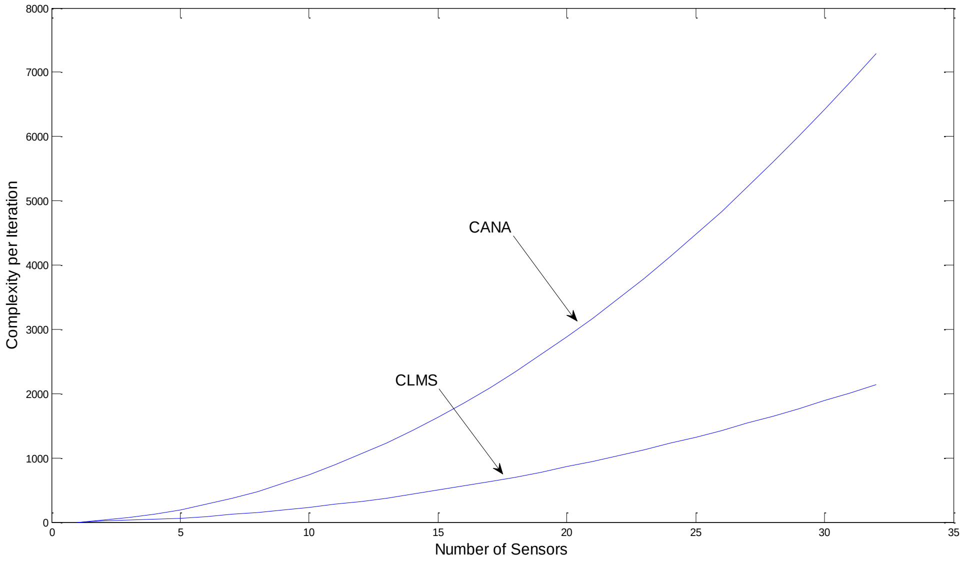

Figure 3.4 : Per iteration complexity comparison of CANA and CLMS

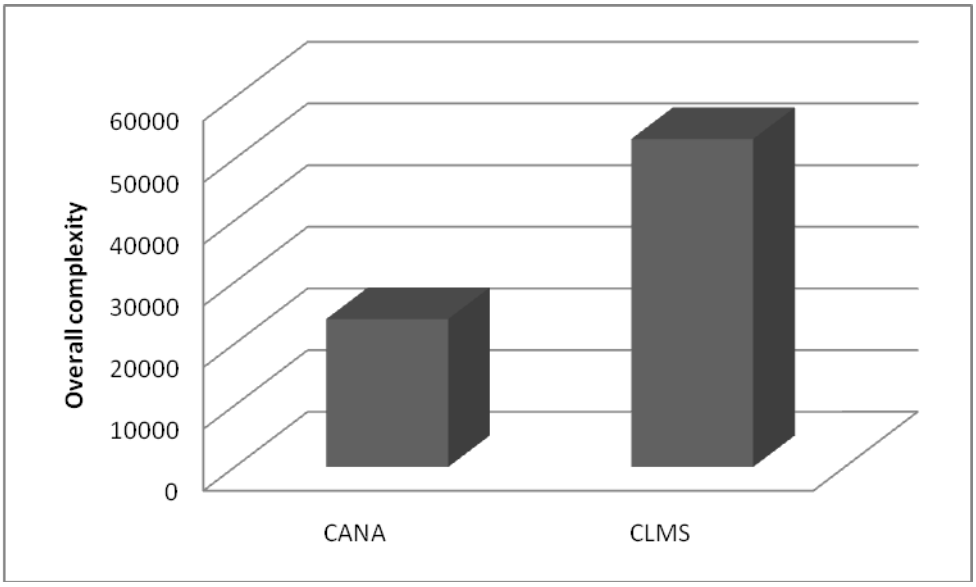

Figure 3.5 : Overall complexity comparison between CANA and CLMS, for 8 sensors at steady state

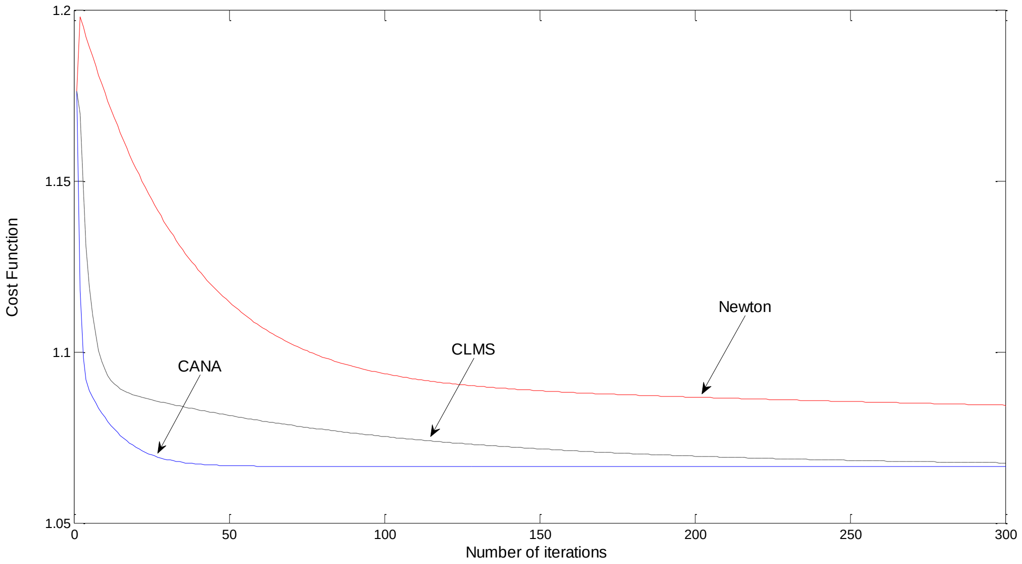

Figure 3.6: Convergence characteristics of CANA, CLMS and constrained Newton algorithms

for 8 sensors the statistics give a different picture. Figure 3.5 is a comparison between overall complexities of CANA and CLMS, where we have used the number of iteration for CANA to converge to optimum state as 50 and 350 for CLMS, for an 8 sensor array. From Figure 3.5, it is evident that the overall computational complexity of CANA is much less than that of CLMS.

3.4.3 Numerical Simulations

In order to validate the concepts developed here, CANA algorithm needs to be tested in a realistic scenario. Let us consider a uniformly spaced array of 8 sensors as depicted in Figure 3.3, where the sensors are placed at half wavelength apart. The two interferers are present at angles 15^{o} and 75^{o} respectively. Assuming direction of arrival of the source of interest to be broadside to the array, with frequency of operation at 300 MHz, the wavelength is equal to 1 meter. Hence the elements are placed at 0.5 meters apart. It is required that the algorithm should iteratively calculate optimum weighing coefficients to be multiplied with received antenna voltages so as to get the desired beam pattern in the presence of interference and additive noise. All the three algorithms CANA, CLMS and constrained Newton algorithm were tested for the calculation of coefficients. The coefficients found thus were used with MATLAB to give insight into the convergence properties, beam pattern and other relevant

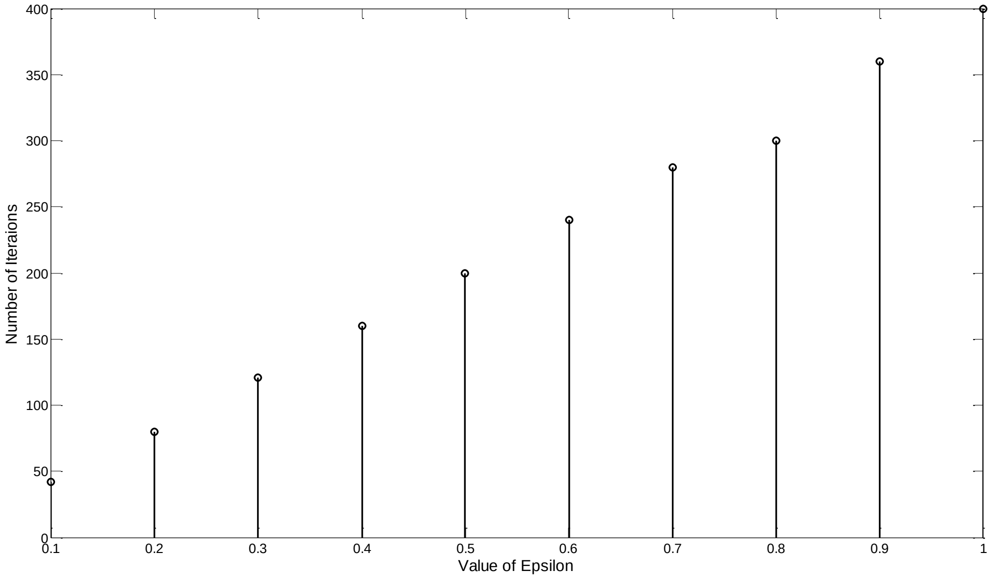

Figure 3.7: Relationship between convergence rate of CANA and the value of Epsilon

measurements. Figure 3.6 shows the convergence properties of three algorithms CANA, CLMS and constrained Newton algorithm for the cost function defined in Eq. 3.14 and Eq. 3.15. For these set of simulations, we chose the convergence step size μ=0.01 and regularisation parameter ε =0.1. Markedly CANA is very quick to attain the optimum value as compared to the rest. While CLMS is notably slow to attain the optimum value, interestingly the Newton algorithm even did not converge in the provided window of iterations. This is because; for certain cost functions the Hessian matrix is not always positive definite. Hence Newton method can get trapped in spurious local minima and thus may not converge as seen in this case. However for the functions that are not exactly quadratic, Hessian matrix can be forced to remain positive definite throughout the optimisation process, but this destroys the usefulness and demands additional computing resources. It may be noted that Constrained Adaptive Natural Gradient Algorithm (CANA) is very sensitive to the values of ε. If we gradually increase the value of ε, algorithm takes longer time to converge to the optimum value. At value of ε equal to 1, the Riemannian metric of Eq. 3.31 is close to Identity matrix, hence the CLMS and CANA agree each other at this point. Interestingly values of ε less than 0.1 are out of bound for this particular case depicted in Figure 3.7 and Figure 3.8, as system becomes unstable for the values less than 0.1. This is in fact governed by Eq. 3.40 which describes the lower bound on the step size. In Figure 3.6 apparently CLMS

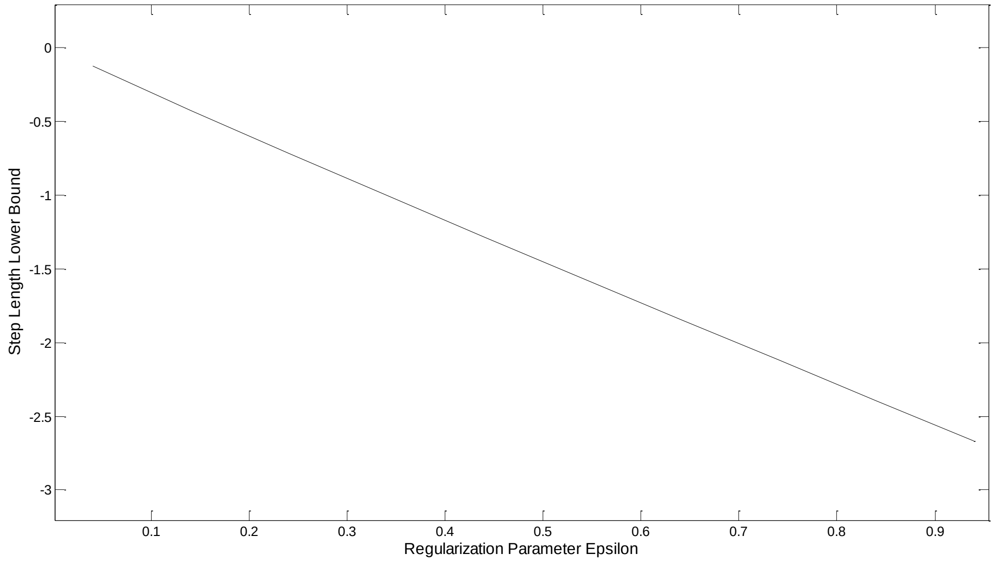

Figure 3.8: Step length lower bound for CANA

Element Number	Optimum Weight	CLMS Weight	CANA Weight
1	0.10718	0.13652	0.10718
2	0.12485	0.09797	0.12485
3	0.12755	0.13293	0.12755
4	0.12801	0.12331	0.12801
5	0.12809	0.15755	0.12809
6	0.12811	0.14669	0.12811
7	0.12811	0.12095	0.12811
8	0.12811	0.08408	0.12811

Table 3.2: Comparison of weights calculated by CANA (NG) and CLMS (CG) algorithms after 350 iterations

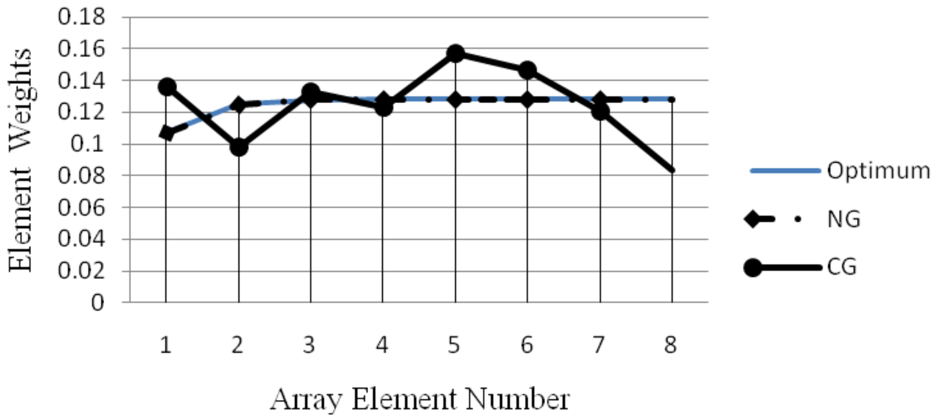

Figure 3.9: Comparison of weights calculated by natural (NG) and conventional gradient based (CLMS) algorithms after 350 iterations

approaches the CANA results, but a close observation and the weights in Table 3.2, reveal that this is not the case and actullay CLMS takes a longer time, way beyond 600 iterations to equate to the Wiener results. This can be shown by examining the weights calculated at 350 iterations, where the CLMS algorithm has apparently converged to the optimum values as shown in Figure 3.6. Table 3.2 summarises the element wise weight comparison between CANA, CLMS and optimum weights after 350 iterations. Please note that cofficients calculated by Newton weight update method have not been included in the comparison presented in Table 3.2 and Figure 3.9, as at this stage they were not even close to convergence as depicted in Figure 3.6. Figure 3.10 gives further insight into this phenomenon, wherein we have steered the main lobe towards -30^o inorder to see the sidelobe structure. Obviously the main lobe is towards -30^o and deep nulls at the designated places of interfrences at 15^o and 75^o but rest of the side lobe structure is not uniform and wider grating lobes are seen, wasting a fair amount of valuable energy. The appearance of such grating lobe structure is due to the discrepency in weight determination by CLMS which is evident in Figure 3.10. It is quite obvious from Figure 3.10 that the CLMS algorithm is following the hard constraints of pointing maxima towards the direction of interest and nulls towards the hostile interferers at 15^o and 75^o respectively but the side lobe structure is not as uniform as required by the Wiener solution. Neverthless after 600 iterations both algorithms conform to each other and sidelobe structures coincide with each other and leads to the optimum Wiener solution. It may also be

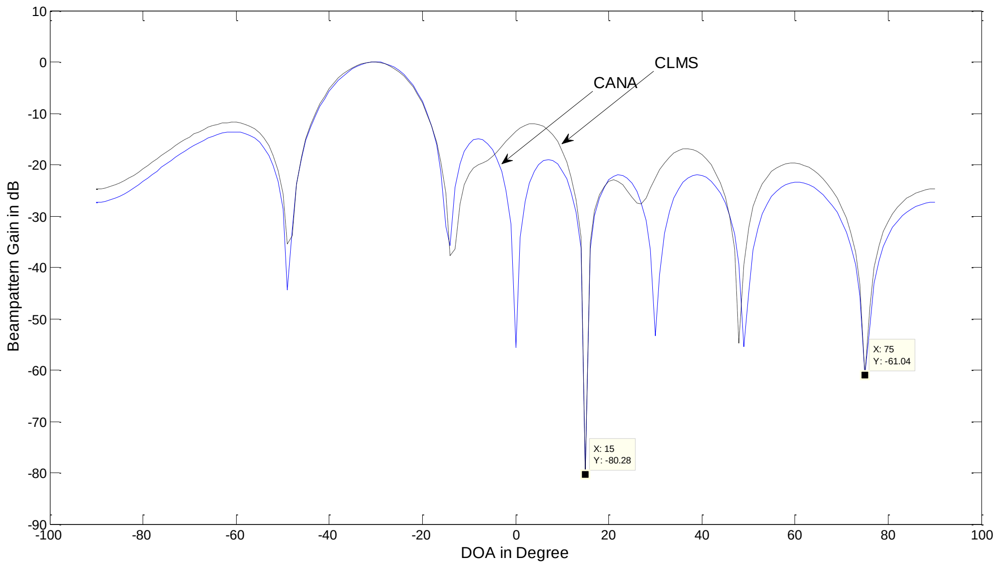

Figure 3.10: A converged beam pattern comparison between CANA and CLMS with weights as in Table 3.2

noted that this phenomenon has been observed even the algorithm has reached the steady state because of the residual error or misadjustment of the algorithm.The CANA algorithm is least affected by the misadjustment however some combinations of step size and regularisation parameter can lead to such discrepency. Thus for better results CANA must observe the step length bound of Eq. 3.41 pictorially depicted in Figure 3.8.

3.5 Multiple Interference Cancellation

Multiple Interference Cancellation (MIC) is another important and crucial area of adaptive array processing [26]. In fact one of the prime advantange of constrained adaptive procesoring over the conventional array processiong or fixed beamforming is that they are able to cancel out interference originating for variable locations [31, 51]. In constrained adaptive array processing the direction of source of interest is generally known but the direction of interfering sources may not be known. If the number of interferer and their locations is variable then coneventional fixed beamformers are unable to cope with it and fail to perform effective interference cancellation [22]. To observe the phenomenon of multiple interference cancellation, two cases of multiple intereference impinging from aribtrary directioms have been studied a) when the interferer is very close to the main-lobe and

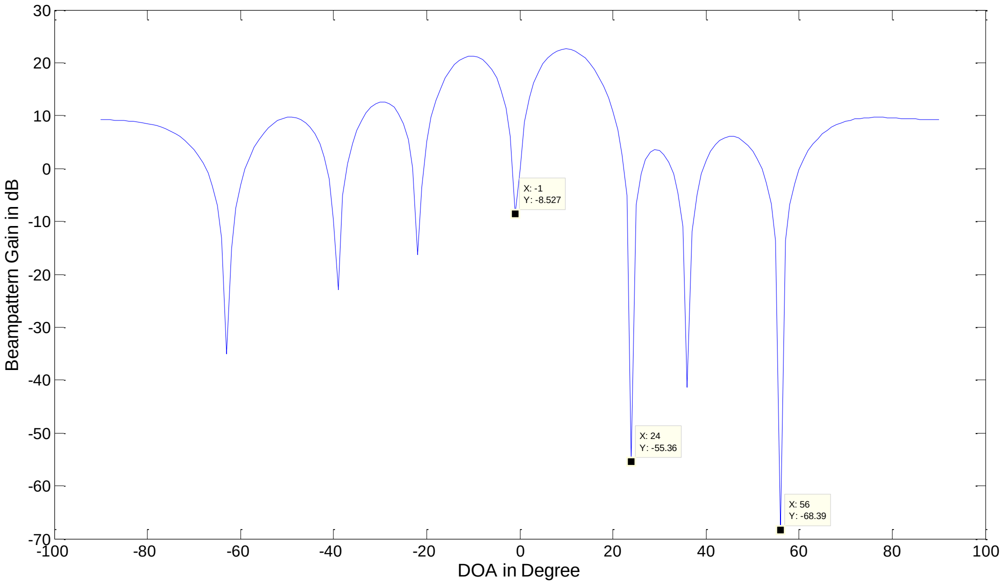

Figure 3.11: Interference cancellation of main-lobe interference present at -1^{o} along with two other interferers at 24^{o} and 56^{o}

b) when when all the interferers have different interference-to-noise ratio. The results have been presented in Tables and Figures. Figure 3.11 represents interference cancellation of three interferers when one of the interferers is present in the main-lobe at -1^{o}. CANA effectively cancels the main-lobe interference and the other interferers present at 24^{o} and 56^{o}. Figure 3.12 illustrates the cancellation of 6 interfering sources when all the signal sources are of unequal Interference-to-Noise(I/N) ratio. The input interference -to-noise ratio of source at 43^{o} is 43.0103 dB, at 30^{o} is 20dB, at 15^{o} is 40 dB , at -15^{o} is 20 dB, at -30^{o} is 20 dB, and that of at -43^{o} is 44.7 dB. The corresponding null depth/power can be seen at the prescribed positions in Figure 3.12. It is also observed that higher power interferers get deeper nulls and lower power interferers get less deep nulls for a fixed noise floor. This asserts that the power of interfering source and the depth of spatial null offered by the adaptive array processor at the position of interferer/jammer have a certain relationship. To explore this, a set of observations has been taken in order to fully understand the phenomenon of interference suppression. For the following set of observations ,it is assumed that a single interferer is at the 24^{o} while the source of interest is at the broadside of the antenna array. Both the sensor noise power and the signal power are 0 dB while the observed null depth at the spatial location of interferer against the input interference-to-noise ratio is listed in Table 3.3. It can be seen that with the increase

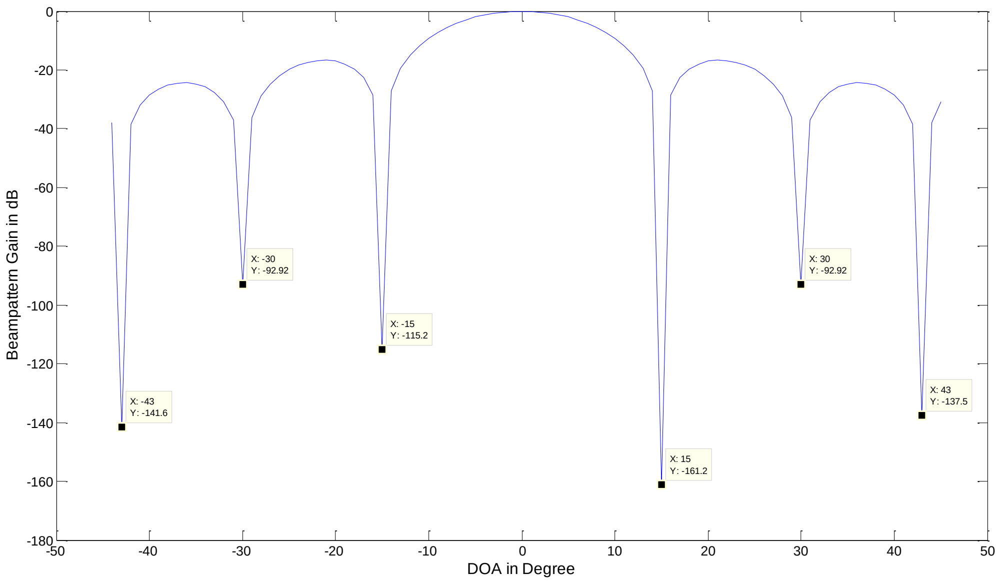

Figure 3.12: Interference cancellation of unequal power signal sources

S/No.	Input I/N ratio dB	Null power at the position of Interferer dB
1	0	-57.87
2	10	-60.0
3	20	-82.92
4	23.0103	-89.85
5	24.7712	-93.9
6	26.0206	-96.77
7	26.98	-99.00
8	27.7815	-100.8
9	28.4510	-102.4
10	29.0309	-103.7
11	29.5424	-104.9
12	30.00	-105.9
13	40.00	-129
14	43.0103	-135
15	44.7712	-139.9
16	50.00	-152.0

Table 3.3: Relationship between I/N ratio and null power/depth

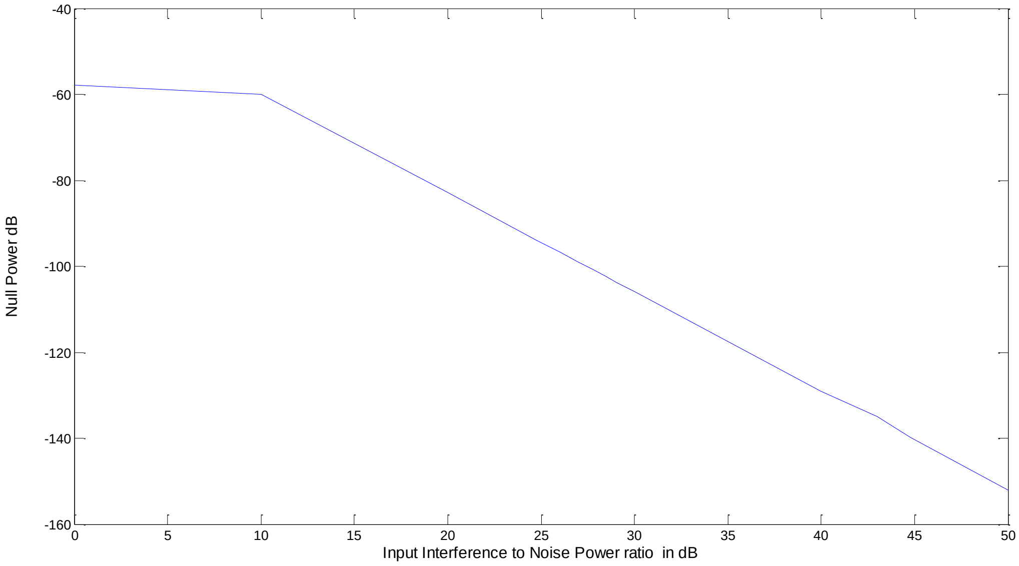

Figure 3.13: Relationship between I/N ratio and null power/depth

in input interference-to-noise ratio the null depth also increases. A direct correlation between the interference-to-noise ratio and the null depth is observed. Figure 3.13 is the pictorial depiction of the same relationship between the input interference-to-noise ratio and the null power/depth. The x-axis represnts null power/depth while interference-to-noise ratio has been plotted at y-axis. Table 3.4 is another set of observations with only difference that the input Interference-to-Noise ratio has been started from -10 dB, while the sensor noise power and the signal power both are kept at 0 dB. As long as the input Interference-to-Noise ratio is lower than the noise floor adaptive processor is unable to insert the null at the spatial location of the interferer. Soon as the input interference-to-noise ratio increased and is equal to or greater than the noise floor/sensor noise the processor is able to offer null at the spatial position of that interferer. Rest of the values for this set of observations are same as in Table 3.2. Both the cases in Table 3.2 and Table 3.3 highlight a trend in Interference suppression which can be observed in Figure 3.11. Input interference-to-noise power in dB at x-axis has been plotted against the null depth in dB at y-axis. It can be seen that by increasing the input input Interference-to-Noise power the null depth increases and vice versa.

S/No.	Input I/N ratio dB	Null power at the position of Interferer dB
1	-10	Null is not at the Required Position
2	0	-57.87
3	10	-60.0
4	20	-82.92
5	23.0103	-89.85
6	24.7712	-93.9
7	26.0206	-96.77
8	26.98	-99.00
9	27.7815	-100.8
10	28.4510	-102.4
11	29.0309	-103.7
12	29.5424	-104.9
13	30.00	-105.9
14	40.00	-129
15	43.0103	-135
16	44.7712	-139.9
17	50.00	-152.9

Table 3.4: Relationship between I/N ratio and null depth at 0 dB noise

3.6 Multi Beam Antenna

A number of applications require that system should be capable of transmitting or receiving multiple beams especially in satellite communication, where number of ground links are attached to single satellite station [52]. Multiuser communications [53],and simultaneous detection of multiple targets are also the primitive applications both in military and air traffic control applications [54]. As development of active phased arrays for radar started over couple of decades ago and is atill being pursued [7]. Only a handful of active phased array radars are operational and much work is still on going to improve the technologies of radiating elements, transmit receive modules and beamformers. Active antennas for telecommunications partially benefit from this R&D, but differences in requirements-low cost, multiple beam operation, re-use of frequencies, and restrictions attached to military developments limit the technology transfer possibilities. Let us consider an array of M sensors placed at half wavelength of centre frequency from each other. There is an incident signal from the direction θ , which is the source of interest for the time being. Note that the rest of the signals received needed to be treated as interference. Thus the recieved signal at the reference element can be written as

$$x_o = e^{j\omega t} \tag{3.43}$$

$$x_1 = x_o e^{-j\omega\tau(\theta)} = e^{j\omega t} e^{-j\omega\tau(\theta)} \tag{3.44}$$

Where $\tau(\theta)$ the time delay is introduced by the signal source at the direction theta with respect to the reference antenna element and can be formulated as

$$\tau(\theta) = d\sin(\theta)/c \tag{3.45}$$

Where *d* is the distance between the adjacent antenna elements in terms of wavelength.With

$$\begin{aligned} c &= f\lambda \\ \omega &= 2\pi f \\ \kappa &= 2\pi / \lambda \end{aligned} \tag{3.46}$$

Therefore Eq. 3.44 can be written as

$$x_1 = e^{j\omega t} e^{-j\kappa d\sin(\theta)} \tag{3.47}$$

And finally the generalised form for the *m*th sensor with respect to reference element is

$$x_m = e^{j\omega t} e^{-jm\kappa d\sin(\theta)} \tag{3.48}$$

The signal recieved by the whole array is the signal of all sensors combined. Thus

$$\boldsymbol{x}[n] = e^{j\omega n} \sum_{m=0}^{M-1} e^{-jm\kappa d\sin(\theta)} \tag{3.49}$$

Where $t = nT_s$, please note that the different notation for time index has been used to avoid confusion between the time index and sensor number.

Eq. 3.49 can be written in vector form as

$$\boldsymbol{x}[n] = e^{j\omega n} \boldsymbol{a}(\theta) \tag{3.50}$$

Where

$$\boldsymbol{a}(\theta) = \begin{bmatrix} 1 & e^{-j\kappa d\sin(\theta)} & e^{-j2\kappa d\sin(\theta)} \ldots & e^{-j(M-1)\kappa d\sin(\theta)} \end{bmatrix}^T \tag{3.51}$$

$\boldsymbol{a}(\theta)$ is called the steering vector or response vector. Now if there are *N* multiple sources impinging on an array , or if the array is a component of multiuser communication system then the signal due to all these sources at the reference sensor can be defined as

$$x_o[n] = \sum_{p=1}^{N} A_p e^{jwn} \tag{3.52}$$

The signal recieved at the m^{th} element with respect to the reference element due to all sources can be written as

$$x_m[n] = \sum_{p=1}^{N} A_p e^{j\left(wn - mkd\sin\left(\theta_p\right)\right)} \tag{3.53}$$

And signal from all sources and all sensors

$$\boldsymbol{x}[n] = \sum_{m=0}^{M-1}\sum_{p=1}^{N} A_p e^{j\left(wn - mkd\sin\left(\theta_p\right)\right)} \tag{3.54}$$

This can be formulated in Vector-Matrix form as

$$\boldsymbol{x}[n] = Ae^{jwn}\mathbf{A}_a(\theta) \tag{3.55}$$

Where $\mathbf{A}_a(\theta)$ is the *M*x*N* steering matrix and is given by

$$\mathbf{A}_a(\theta) = \left[\boldsymbol{a}(\theta_1) \quad \boldsymbol{a}(\theta_2) \quad \boldsymbol{a}(\theta_3)\ldots \quad \boldsymbol{a}(\theta_N)\right] \tag{3.56}$$

and

$$\boldsymbol{a}\left(\theta_p\right) = \left[1 \quad e^{-j\kappa d\sin\left(\theta_p\right)} \quad e^{-j2\kappa d\sin\left(\theta_p\right)}\ldots \quad e^{-j(M-1)\kappa d\sin\left(\theta_p\right)}\right]^T \tag{3.57}$$

The multi beam antenna constraint can be formulated as

$$\mathbf{A}_a(\theta)^T \boldsymbol{w} = \boldsymbol{f}_a \tag{3.58}$$

Where $\boldsymbol{f}_a = [1 \quad 1 \quad 1\ldots. \quad 1]^T$ is *N*x1 vector of constraints. This constraint vector serves to produce N multiple beams in distinct directions, hence every element should be “1” or some defined power level with respect to each look direction or distinct user.

Producing multiple beams in distincts directions requires minimizing the output power subject to the consraint in Eq. 3.58. Thus following the similar mathematics as developed in earlier formulation of CANA in Section 3.4, leads gradient of the cost function to be

$$\partial J_{(w)} \big/ \partial \boldsymbol{w} = \partial \big/ \partial \boldsymbol{w} \left[\boldsymbol{w}^H \mathbf{R} \boldsymbol{w} + \boldsymbol{\lambda}^T \left(\mathbf{A}_a(\theta)^T \boldsymbol{w} - \boldsymbol{f}_a \right) \right]$$

$$= \left[\mathbf{R}\boldsymbol{w} + \mathbf{A}_a(\theta) \boldsymbol{\lambda} \right] \tag{3.59}$$

Further using the analogous procedure as in Section 3.4, the CANA solution for multi beam antenna can be obtained as

$$\boldsymbol{w}[n+1] = \mathbf{P} \left[\boldsymbol{w}[n] - \mu \mathbf{G}^{-1}(\boldsymbol{w}) \left(\frac{\Delta J_{(w)}}{\Delta \boldsymbol{w}_{[n]}} \right) \right] + \boldsymbol{k}_a \tag{3.60}$$

Where

$$\mathbf{P} = \left[\mathbf{I} - \mathbf{A}_a(\theta) \mathbf{A}_a(\theta)^T \left[\mathbf{A}_a(\theta)^T \mathbf{A}_a(\theta) \right]^{-1} \right] \tag{3.61}$$

and

$$\boldsymbol{k}_a = \mathbf{A}_a(\theta)^T \left[\mathbf{A}_a(\theta)^T \mathbf{A}_a(\theta) \right]^{-1} \boldsymbol{f}_a \tag{3.62}$$

Let the number of array elements be 8, placed at half wavelength distance appart from each other. Supose there are two desired sources at 20^o and 45^o and an interferer at 0^o. Keeping the remaining settings of the step size and regularisation parameters intact as in previous numerical simulations the obtained graphical results are given as follows. Figure 3.14 is the ploar graph of multi beam antenna where two beams have been pointed towards the required direction of 20^o and 45^o while interferer is at 0^o. Figure 3.15 and 3.16 are the dB and normalised plot of the same simulation wherein -71.46 dB null has been inserted at the interferer position of 0^o. Number and the power of beams at the defined locations can be manged through the constraint vector.

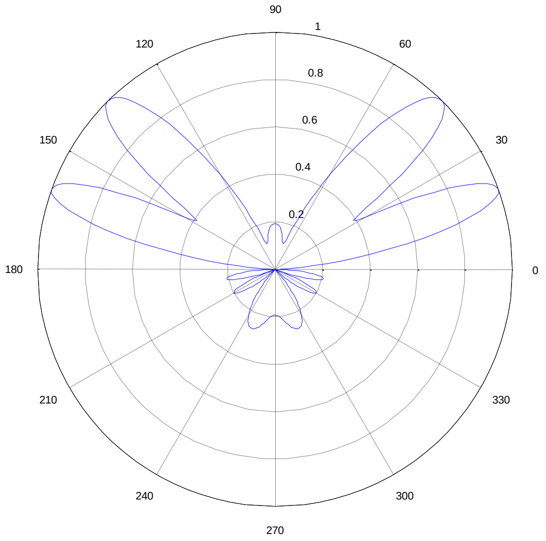

Figure 3.14: Polar plot of multi beam antenna with beams at 20^o and 45^o and interferer at 0^o

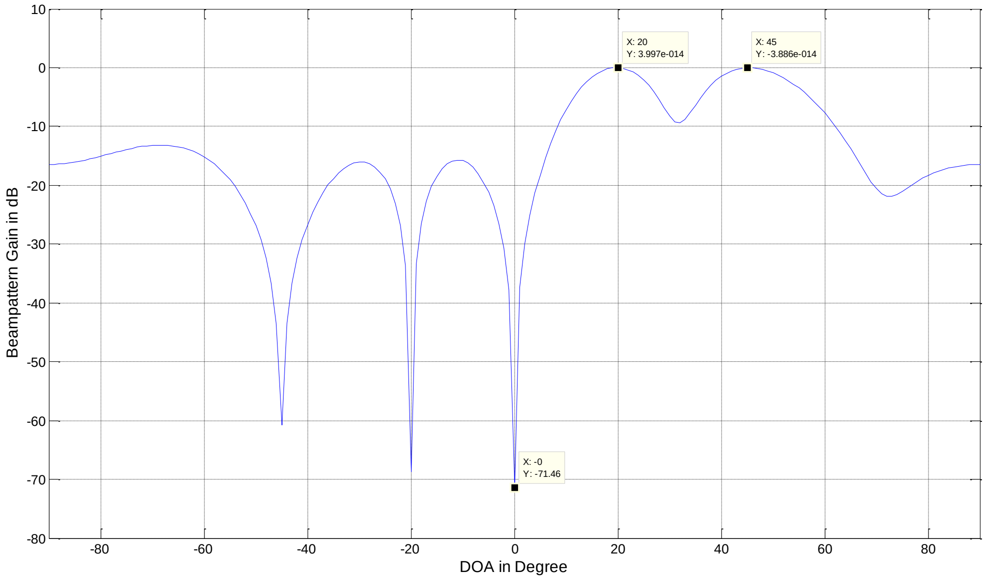

Figure 3.15: Multi beam antenna with beams at 20^o and 45^o and interferer at 0^o

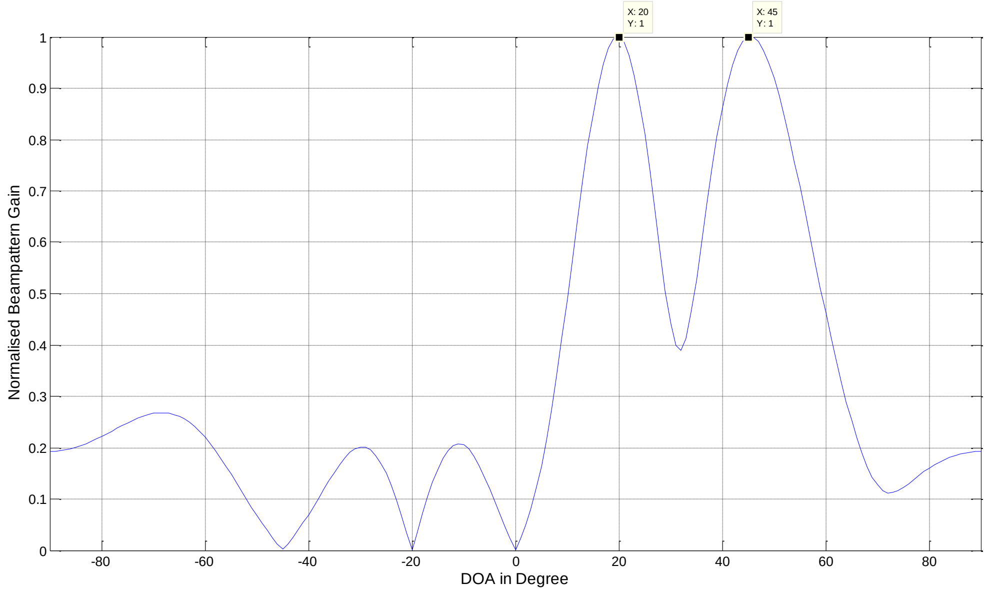

Figure 3.16: Normalised response of multi beam antenna with beams at 20^o and 45^o and interferer at 0^o

3.7 Constrained Adaptive Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM)

Many applications like Thousand Element Array (THEA) and similar processes involving multi functions of navigation, tracking, guidance, and surveillance require the number of elements to be very large. Similarly the size of weight vector $\boldsymbol{w}$ is directly propotional to the number of elements for narrowband processing. But if the system involves broadband processing the size of weight vector becomes MxL. Where M is the number of sensors and L is the order of the filter or number of taps associated with each antenna for Time Delay Line (TDL) type broadband adaptive processing. As explained and derived earlier, the Riemannian metric is a function of weight vector and is updated at each iteration with the update in w, thus is the major contributor in algorithm computational complexity.

Therefore for the purpose of reducing the number of arithmetic operations at each iteration,it would be more appropriate to have a non adaptive approximation to the Riemannian metric. On a similar note, we may not be able to find a single non adaptive metric suitable for all arrays, as every antenna array itself exhibit a Riemannian manifold. Due to intrinsic curvature information of an array manifold, the non adaptive metric must be unique. Thus in order to fully exploit the geometric properties of the objective function it is always apposite to have an adaptive version of the Riemannian metric unless the optimisation surface is known exactly. Consider a conventional narrowband adaptive array processor depicted in Figure 3.1. 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 y_n.

$$y_n = \boldsymbol{w}_n^H \boldsymbol{x}_n \tag{3.63}$$

Where $\boldsymbol{w}_n$ is the complex weight vector of coefficients attached to the antenna elements and $\boldsymbol{x}_n$ is the incoming data vector, and the time instant is represented by the subscript *n*. Now the average output $E\left[\left|{y_n}^2\right|\right]$ has to be minimized while observing look direction constraint. The look direction constraint can be formulated as

$$\mathbf{c}^T \boldsymbol{w}_n = f_c \tag{3.64}$$

Where $\boldsymbol{c}$ is the constraint vector and f_c is the scalar constraint for the narrowband case. We will assume this constraint for a single source of interest to be equal to “1”. $\boldsymbol{c}$ is an *M*x1 vector with all elements equal to one for broadside look direction and is equal to steering vector otherwise. With the look direction broadside to the line of sensors the constraint Eq. 3.64 can be realised as

$$\sum_{m=1}^{M} w_m = 1 \tag{3.65}$$

From Eq. (3.63),(3.64) and (3.65) the cost function to be minimised can be formulated as

$$J_{(\boldsymbol{w})} = E\left[\boldsymbol{w}_n^H \boldsymbol{x}_n \boldsymbol{x}_n^H \boldsymbol{w}_n\right] \tag{3.66}$$

Subject to the constraint given in Eq. (3.64). The Lagrange method, as shown previously can be used to incorporate the constraint in the cost function above to be minimized with respect to the weight vector $\boldsymbol{w}_n$. Hence Eq.3.66 takes the form as

$$J_{c(\boldsymbol{w})} = \boldsymbol{w}_n^H \mathbf{R} \boldsymbol{w}_n + \lambda\left(\mathbf{c}^T \boldsymbol{w}_n - f_c\right) \tag{3.67}$$

Where $\mathbf{R}$ is the correlation matrix of the input data vector $\boldsymbol{x}_n$ and λ is the Lagrange multiplier. To minimise the cost function its gradient is set equal to zero. As described earlier in Section 3.4, the natural gradient of a cost function $J_{(\boldsymbol{w})}$ can be written as.

$$\tilde{\nabla} J_{(\boldsymbol{w})} = \mathbf{G}^{-1}(\boldsymbol{w}) J_{(\boldsymbol{w})} \tag{3.68}$$

Where $\nabla J_{(\boldsymbol{w})}$ and $\mathbf{G}(\boldsymbol{w})$ are the conventional gradient of the cost function and the Riemannian metric respectively. Similarly the gradient of the constrained cost function defined in Eq. 3.67 can be obtained as

$$\nabla J_{c(\boldsymbol{w})} = \mathbf{R}\boldsymbol{w}_n + \lambda \mathbf{c} \tag{3.69}$$

Following a derivation similar to that given in Eq. 3.28 the weight vector update may be expressed as

$$\boldsymbol{w}_{n+1} = \mathbf{P}\left[\boldsymbol{w}_n - \mu \mathbf{G}^{-1}(\boldsymbol{w})\left(\frac{\Delta J_{(\boldsymbol{w})}}{\Delta \boldsymbol{w}_n}\right)\right] + \boldsymbol{k} \tag{3.70}$$

Where the Riemannian metric is given by

$$\mathbf{G}^{-1}(\boldsymbol{w}) = \left[\mathbf{I} - \boldsymbol{w}_n \boldsymbol{w}_n^H\right] \tag{3.71}$$

However as the Riemannian metric is a function of weight vector i.e. of the optimisation surface, it has to be updated at each iteration and this therefore adds to computational complexity. If we could replace it with some fixed matrix then significant computational complexity could be reduced. The Riemannian metric, we used in the formulations and numerical simulations in CANA and broadband constrained natural gradient based adaptive processing [43] gives an intuition to finding a fixed metric for the natural gradient adaptation. The Riemannian metric depicted in Figure 3.17 is the one which was used for the cost function optimisation space of 8 element uniformally spaced linear array, which has proven convergence properties. A bird's eye view reveals that it is a diagonally dominant matrix. So if we replace it with some diagonally dominant fixed matrix we should achieve approximately the same convergence results depending upon our approximation of the metric. However this may not always the be case and the metric has to be updated at every iteration. It may also be noted that Identity matrix is the pure diagonally dominant matrix at hand. We also know that if

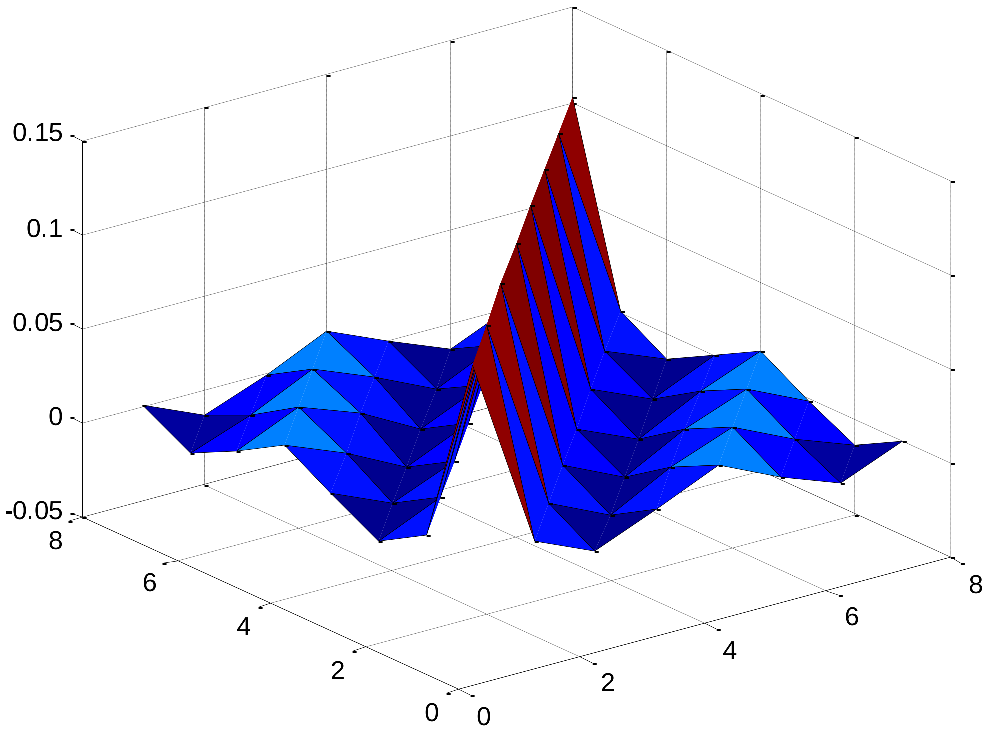

Figure 3.17: Diagonally dominant Riemannian metric

we replace Riemannian metric $\mathbf{G}(\boldsymbol{w})$ with Identity matrix in Eq. 3.70 it becomes the Constrained Least Mean Square (CLMS) algorithm, originally proposed by Frost [24]. But if we let $\mathbf{G}=\xi\mathbf{I}$ where ξ is a regularisation parameter which can be adjusted to achieve the desired convergence similar to an adaptive Riemannian metric. Further it can be explained that the CANA-NAM algorithm observes the same convergence condition as described in convergence analysis of CANA and is given by

$$-\frac{1}{\lambda_{\max}}\leq\mu\leq 0 \tag{3.72}$$

Wherein $\lambda_{\max}$ is the largest eigenvalue of the square matrix $\mathbf{PG}^{-1}(\boldsymbol{w})\mathbf{RP}$ as in Eq. 3.38. It may be noted that the trace value, $tr\left[\mathbf{PG}^{-1}(\boldsymbol{w})\mathbf{RP}\right]$ is always more than $M\lambda_{\max}$, where M is the dimension of matrix $\mathbf{PG}^{-1}(\boldsymbol{w})\mathbf{RP}$. Having this, the convergence condition given by Eq. 3.72 can lead to the choice of step size μ as

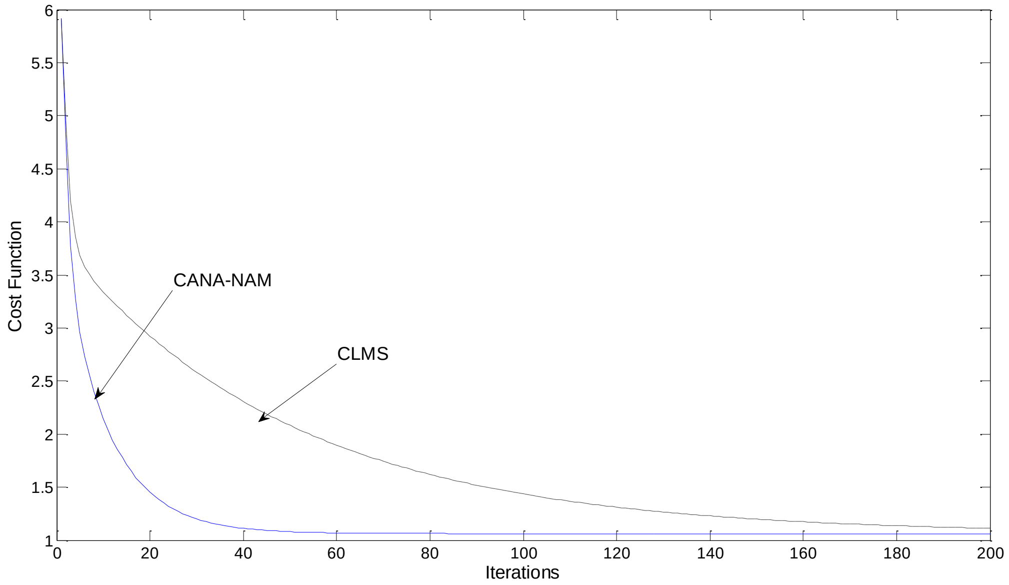

Figure 3.18: Convergence comparison of CANA-NAM with CLMS

$$-\frac{M}{tr\left[\mathbf{PG}^{-1}\left(\boldsymbol{w}\right)\mathbf{RP}\right]}\leq\mu\leq0 \tag{3.73}$$

3.7.1 Numerical Simulations

Let us consider a Uniformly spaced Linear Array(ULA) of 16 antenna elements spaced at half wavelength of the frequency of operation. Let there be two interferers at 45^o and -50^o with source of interest at broadside side of the array. It is required that the array processor should adopt a beampattern such as to present maxima towards the Direction of Interest (DoI) and suppress any interference thereof. The direction of interferers and their strengths are not known to the adaptive processor. However for simulation purposes we will add some interferers in order to create a realistic environment for the algorithm.With known desired response, DoI and the inputs as stated above, Constrained Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM) was used for the calucation of optimum weight vector. For comparison Constrained Least Mean Square algorithm (CLMS) was also simulated with the same sttings. We chose the convergence step size $\mu=0.05$ and regularisation parameter $\xi=0.1$. Figure 3.18 shows the convergence statistics of both the Constrained Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM) and Constrained Least Mean Square algorithm (CLMS). Constrained natural gradient with non adaptive metric is very quick to attain the optimum value and converges in fewer steps than the CLMS, taking as

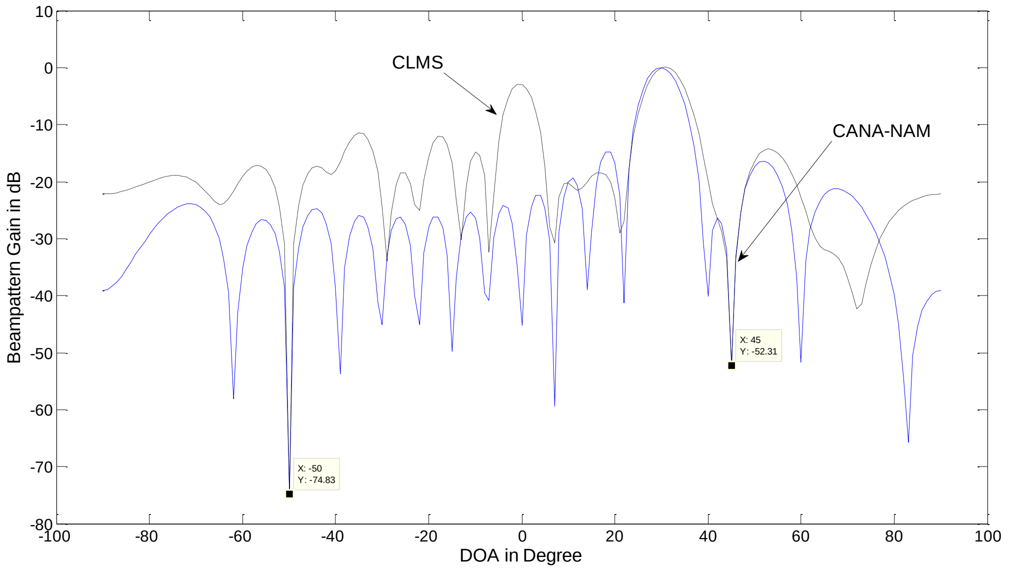

Figure 3.19: A converged beam pattern comparison between CANA-NAM and CLMS after 200 iterations

many as 200 steps to converge to the optimum value. The optimum value for this particular set of simulations was found to be 0.1 resulting algorithm to converge to optimum value far quick than the counterpart. Figure 3.19 is a very good demonstration of efficiency and optimality of proposed algorithm. It is always required that the side lobe level should remain as small as possible, so that the maximum energy can be directed towards the look direction. The side lobe level efficiency of CLMS is much inferior to CANA-NAM as depicted in Figure 3.19. This also shows the accuracy of the developed technique. It may be noted that the value of regularisation parameter ξ is unique and depends upon the number of array antenas as well as number of signal sources. Convergence characteristics of Constrained Adaptive Natural Gradient algorithm with Non Adaptive Metric (CANA-NAM), shown in Figure 3.18, have some interesting relationship with the value of ξ. As shown in Figure 3.20 changing the value of ξ significantly changes the rate of convergence. If we gradually increase the value of ξ, from 0.1 to 1 the the convergence rate of CANA-NAM slows down and it progressively meets with the CLMS solution. At ξ equal to 1, **G** become identity matrix and both of the solutions meet each. With values higher than 1, algorithm takes longer time to converge to the optimum value. Figure 3.21 shows algorithm's capability of rejecting multiple interferers impinging

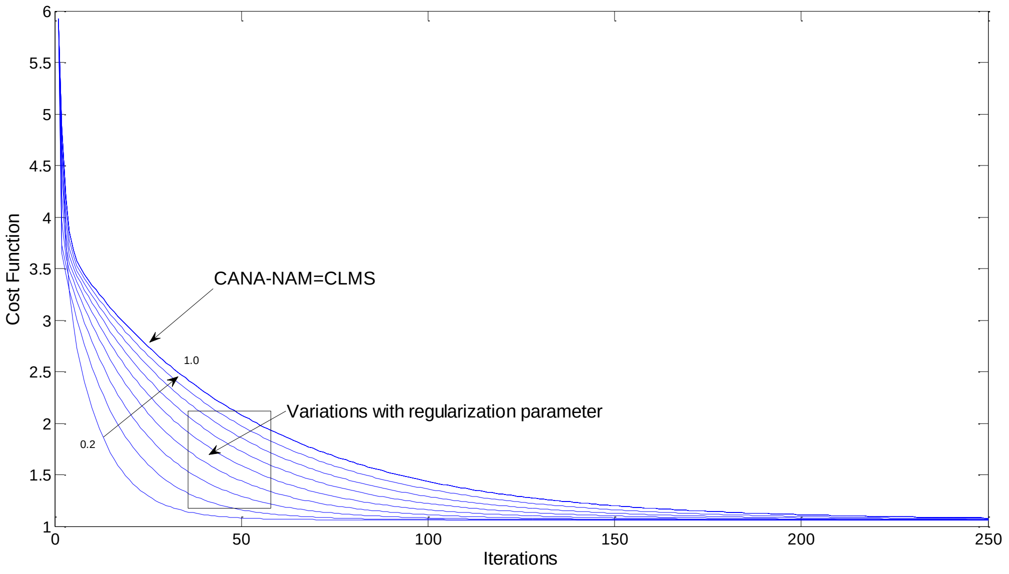

Figure 3.20: Effect of regularisation parameter zeta on convergence of CANA-NAM

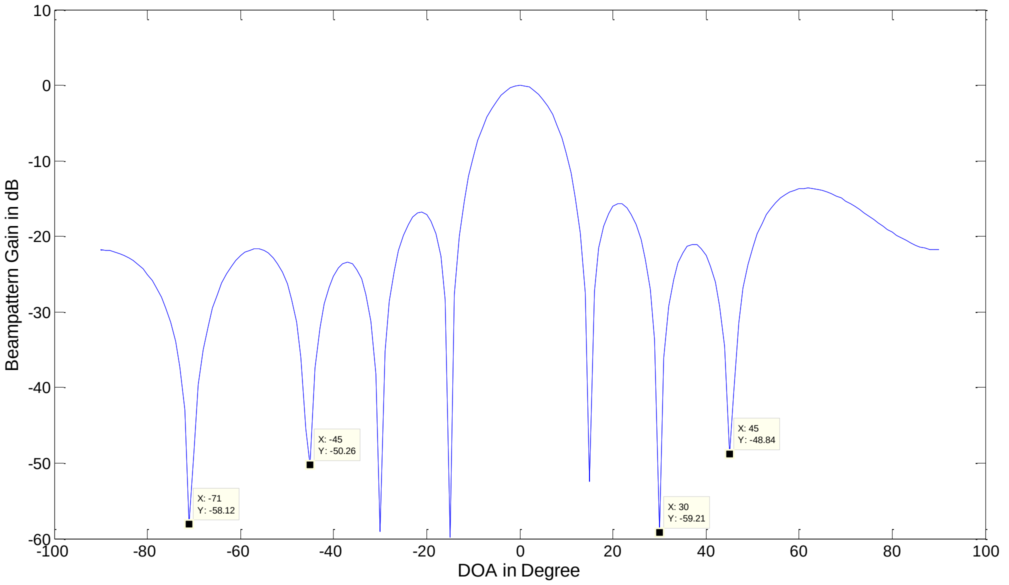

Figure 3.21: Multiple interference cancellation by CANA-NAM with interferers at 30^o, 45^o, -30^o and -7^o

from four different directions of 30^o, 45^o, -30^o and -71^o. The main lobe is directed towards the the broadside source of interest and the deeper nulls are presented. It is also observed that the algorithm produces deeper nulls as the power of interferer is increased and vice versa. However it is imperative to note that the non adaptive metric used here effectively only changes the step size since the metric is a scaled identity. Similarly with the change in number of sources and or the number of antennas, the same metric might not work and can lead to inaccurate results. Since the CANA-NAM algorithm highly depends upon prior knoweldge of the Riemannian metric for approximating the non adaptive metric. Thus a hybrid approach can be followed in which for some initial iterations the CANA algorithm operates in normal mode and after approximating the Riemannian metric, shifts to the non adaptive metric mode. Nevertheless the formulation of this hybrid approach can be left as a future work.

3.8 Adaptive Step-size Algorithm

Convergence rate of the gradient based search methods is highly dependent on the conditioning of autocorrelation matrix of its inputs [55]. When inputs are highly correlated, convergence rate degrades radically. Variable step-size is applied, with the intention of decreasing misadjustment and to maximise convergence rate of LMS algorithms [18].To design an appropriate variable step size method, one should intuitively use a larger step size when the estimate is far from the optimum and a smaller step size as it approaches the optimum [7, 8]. Generally, the optimising algorithm for solving the linear and nonlinear equations is also very sensitive to the search strategy and the choice of the variables value [7]. In case of nonlinear equations and because of the complex search strategy, it needs too long time to run these algorithms.

On the other hand, if the choice of the variables value is not reasonable, the results would be restricted to some parts of the extreme values, thus it is difficult to get the high precision solutions. Thus the algorithm should also be able to change the step size according to the optimisation space. To incorporate this adaptability in existing solution, we note that Constrained Natural Gradient Algorithm (CANA) can further be explained as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}'_{n+1} + \alpha_n \boldsymbol{d}_n \tag{3.74}$$

Where $\boldsymbol{w}'_{n+1} = \mathbf{P}\boldsymbol{w}_n + \boldsymbol{k}$ and $\boldsymbol{d}_n = \mathbf{P}\mathbf{G}^{-1}(\boldsymbol{w})\mathbf{R}\boldsymbol{w}_n$

Now to determine the optimal value of step size α we set

$$\frac{\partial J_{(\boldsymbol{w})}}{\partial \alpha_n} = 0$$

where

$$\frac{\partial J_{(\boldsymbol{w})}}{\partial \alpha_n} \overset{\Delta}{=} \left(\frac{\partial J_{(\boldsymbol{w})}}{\partial \boldsymbol{w}_{n+1}}\right)^T \left(\frac{\partial \boldsymbol{w}_{n+1}}{\partial \alpha_n}\right) \tag{3.75}$$

From Eq. 3.74 we get

$$\left(\frac{\partial \boldsymbol{w}_{n+1}}{\partial \alpha_n}\right) = \boldsymbol{d}_n \tag{3.76}$$

Further since

$$J_{(\boldsymbol{w})} = \frac{1}{2}\boldsymbol{w}_{n+1}^H \mathbf{R}\boldsymbol{w}_{n+1}$$

So $\left(\frac{\partial J_{(\boldsymbol{w})}}{\partial \boldsymbol{w}_{n+1}}\right) = \mathbf{R}\boldsymbol{w}_{n+1}$ (3.77)

Which yields

$$\frac{\partial J_{(\boldsymbol{w})}}{\partial \alpha_n} = \boldsymbol{d}_n^T \mathbf{R}\left[\boldsymbol{w}'_{n+1} + \alpha_n \boldsymbol{d}_n\right] \tag{3.78}$$

putting above Eq. 3.78 equal to zero yields adaptive step size scheme for CANA

$$\alpha_n = -\frac{\boldsymbol{d}_n{}^T \mathbf{R}\boldsymbol{w}'_{n+1}}{\boldsymbol{d}_n{}^T \boldsymbol{d}_n} \tag{3.79}$$

This adaptive step size can be calculated iteratively. Note that for fixed step size the analysis has been provided in earlier section on convergence analysis, which gives us the optimum

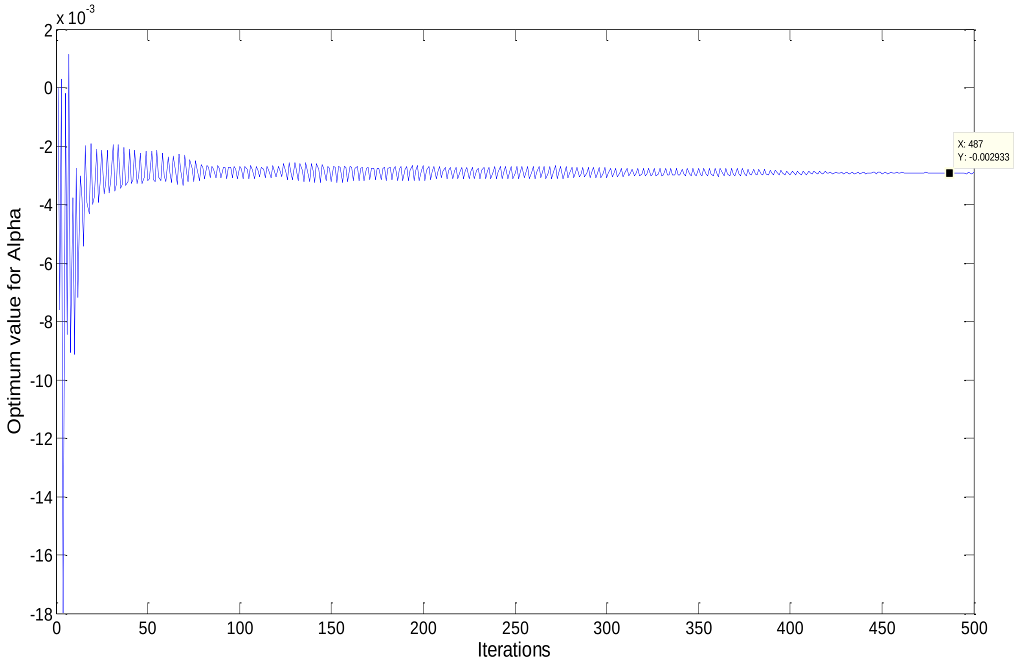

Figure 3.22: Optimum value of step size alpha for adaptive step-size algorithm

range of step size. The same approach as in Section 3.4.1 can be followed to prove that the suitable choice for the step size α can be derived from the following relationship.

$$-\frac{M}{tr\left[\mathbf{P}\tilde{\mathbf{G}}^{-1}\mathbf{RP}\right]} \leq \alpha \leq 0 \tag{3.80}$$

Wherein λ is the eigenvalue of the square matrix $\mathbf{P}\tilde{\mathbf{G}}^{-1}\mathbf{RP}$. Eq. 3.80 has been plotted in Figure 3.22. The algorithm converges to optmimum value as described by the above shown relationship. The step size obtained in 400 iterations is equal to -0.002933. It may be noted that the steady state value of the step size is highly dependent upon the initial weight vector, intial step size and the value of ξ. Figure 3.23 shows the convergence comparison between adaptive step size algorithm and the fixed step size algorithm. The adaptive step size algorithm quickly acquires the steady state weights. Figure 3.24 and 3.25 show the beam patterns in dB and polar forms wherein the desired user is at broadside of the array and an interferer at 24^{o}. Thus the scheme improves the convergence properties by exploting the varaible stepsize while searching the performance surface.

Note that two interferers were used in the simulation that resulted in Figure 3.6, while there were four interferers used in Figure 3.18. One interferer was present for the results plotted in

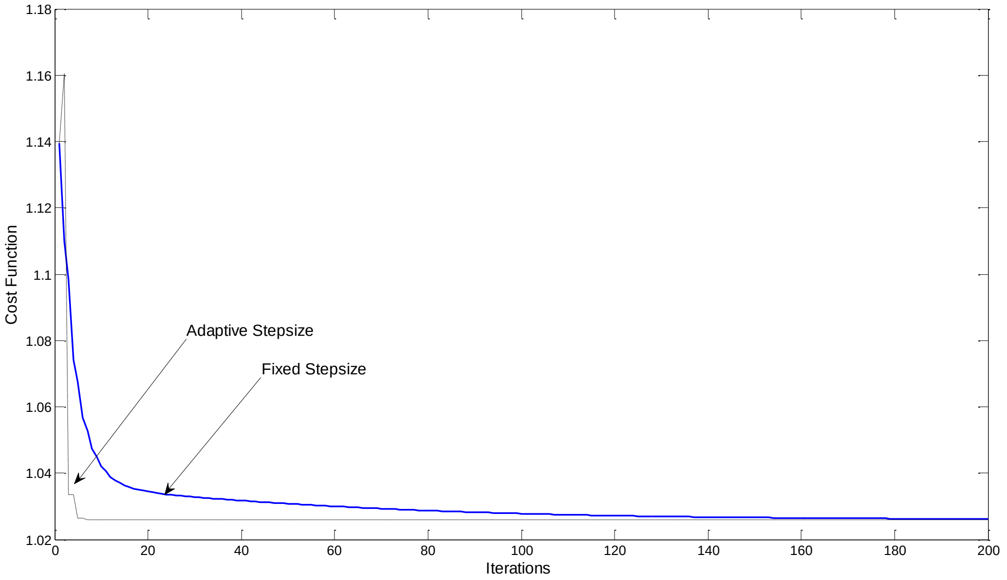

Figure 3.23: Convergence comparison of adaptive step-size algorithm with fixed step-size CANA-NAM

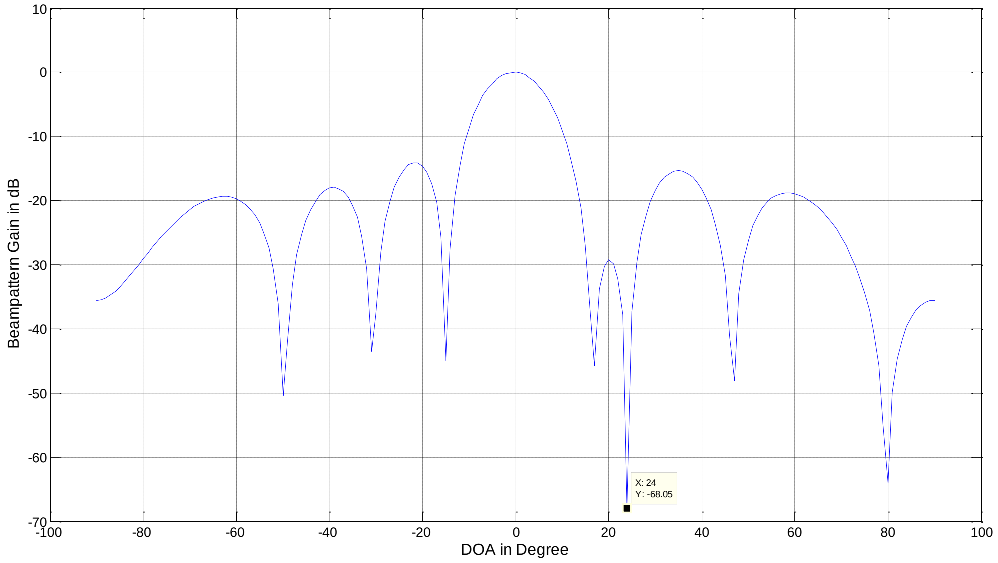

Figure 3.24: Broadside beamforming with adaptive step-size algorithm

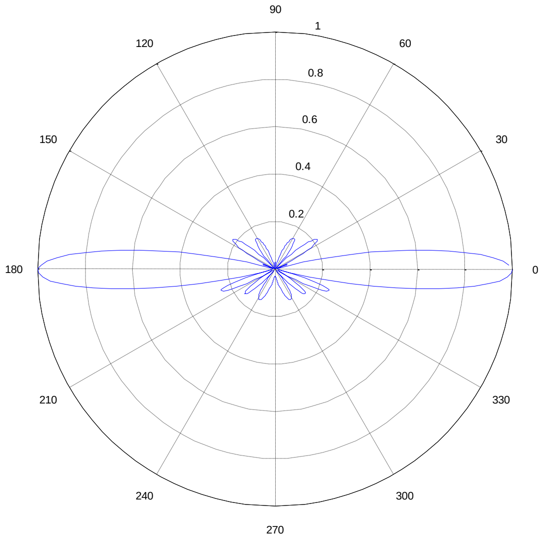

Figure 3.25: Polar plot of broadside beamforming with adaptive step-size algorithm

Figure 3.23. The difference in number of contributing signal sources has lead to the difference in maximum values of the respective cost functions, plotted in Figure 3.6, Figure 3.18 and Figure 3.23.

3.9 CANA in polar space

CANA can also be represented in Polar coordinate system for having further insight into the angular characteristics of the method. If we consider a two element system then the constrained cost function of Eq. 3.15 can be written as

$$J_{(w)} = [w_1 \quad w_2]\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}[w_1 \quad w_2]^H + \lambda\left([c_1 \quad c_2]^T[w_1 \quad w_2] - f_c\right) \tag{3.81}$$

If $w_1 = r\cos\theta, w_2 = r\sin\theta$ and $f_c = 1$, then Eq. 3.81 can be represented in polar coordinate system as

$$J(r,\theta) = [r\cos\theta \quad r\sin\theta]\begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}[r\cos\theta \quad r\sin\theta]^H + \lambda\left([c_1 \quad c_2]^T[r\cos\theta \quad r\sin\theta] - 1\right) \tag{3.82}$$

Differentiating with respect to polar coordinates, r,θ Eq. 3.82 leads

$$\frac{dJ}{dr,d\theta}(r,\theta)=\boldsymbol{h}+\lambda\mathbf{Q}\begin{bmatrix}c_1 & c_2\end{bmatrix}^T \tag{3.83}$$

Where $\frac{dJ}{dr,d\theta}(r,\theta)$ is the gradient in polar coordinate system, ***h*** and **Q** are given by

$$\boldsymbol{h}=\Big[2r\left\{\cos\theta\sin\theta\left(R_{12}+R_{21}\right)+\cos^2\theta R_{11}+\sin^2\theta R_{22}\right\};$$
$$r^2\left\{\left(\cos^2\theta-\sin^2\theta\right)\left(R_{12}+R_{21}\right)-2\cos\theta\sin\theta\left(R_{11}-R_{22}\right)\right\}\Big]$$

$$\mathbf{Q}=\begin{bmatrix}\cos\theta & \sin\theta\\ -r\sin\theta & r\cos\theta\end{bmatrix}$$

Thus the weight update can be written as.

$$\boldsymbol{w}_{\mathrm{n}+1}=\boldsymbol{w}_{\mathrm{n}}-\mu\frac{dJ}{dr,d\theta}(r,\theta) \tag{3.84}$$

Inserting the value of gradient leads

$$\boldsymbol{w}_{\mathrm{n}+1}=\boldsymbol{w}_{\mathrm{n}}-\mu\left[\boldsymbol{h}+\lambda\mathbf{Q}\begin{bmatrix}c_1 & c_2\end{bmatrix}^T\right] \tag{3.85}$$

Following the same procedure as in Eq. 3.27 and inserting the value of λ, we get the weight update equation as

$$\boldsymbol{w}_{n+1}=\mathbf{P}\left[\boldsymbol{w}_n-\mu\boldsymbol{h}\right]+\boldsymbol{b} \tag{3.86}$$

Where $\mathbf{P}=[\mathbf{I}-\boldsymbol{b}\boldsymbol{c}^T]$ is the projection matrix and $\boldsymbol{b}=\mathbf{Q}\boldsymbol{c}\left[\boldsymbol{c}^T\mathbf{Q}\boldsymbol{c}\right]^{-1}$

The natural gradient in polar space is defined as

$$\frac{dJ_{NG}}{dr,d\theta}(r,\theta)=\mathbf{G}^{-1}\frac{dJ}{dr,d\theta}(r,\theta) \tag{3.87}$$

Where **G** is the Riemannian metric in polar space and is given by

$$\mathbf{G}=\begin{bmatrix}1 & 0\\ 0 & r^2\end{bmatrix} \tag{3.88}$$

Applying on the constrained cost function the Polar Natural Gradient (PNG) is given as

$$\frac{dJ_{NG}}{dr,d\theta}(r,\theta)=\boldsymbol{h}_1+\lambda\mathbf{Q}_1\begin{bmatrix}c_1 & c_2\end{bmatrix}^T \tag{3.89}$$

Where

$$\boldsymbol{h}_1=\Big[2r\left\{\cos\theta\sin\theta\left(R_{12}+R_{21}\right)+\cos^2\theta R_{11}+\sin^2\theta R_{22}\right\};$$
$$\left\{\left(\cos^2\theta-\sin^2\theta\right)\left(R_{12}+R_{21}\right)-2\cos\theta\sin\theta\left(R_{11}-R_{22}\right)\right\}\Big]$$

$$\mathbf{Q}_1=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta/r & \cos\theta/r\end{bmatrix}$$

And hence the constrained weight updating equation is given as

$$\boldsymbol{w}_{n+1}=\mathbf{P}\left[\boldsymbol{w}_n-\mu\mathbf{h}_1\right]+\boldsymbol{b}_1 \tag{3.90}$$

Where $\mathbf{P}=\left[\mathbf{I}-\boldsymbol{b}_1\boldsymbol{c}^T\right]$ is the projection matrix and $\boldsymbol{b}_1=\mathbf{Q}_1\boldsymbol{c}\left[\boldsymbol{c}^T\mathbf{Q}_1\boldsymbol{c}\right]^{-1}$

Eq. 3.90 is the natural gradient based weight update equation in polar space. It is worth noting that the natural and Euclidean gradients only vary in the angle update and the magnitude calculation remains same. This formulation can be extended in spherical as well as to the *N* dimensions using different realisations of Riemannian metrics.

3.10 Conclusion

This chapter has presented a novel technique for constrained adaptive algorithm based upon the natural gradient. This technique provides more uniform performance than those based upon ordinary stochastic gradients, yet simple to implement. Along with the ability of natural gradient algorithm to follow the exact surface of the optimisation space, self correcting feature of constrained optimisation makes the algorithm rapidly converge to steady optimum state for variety of statistically optimum functions. So, for a reasonable range of step size the drawback of ordinary gradient of escaping from the cost function surface can be avoided, still maintaining the improved converging properties. This chapter has also presented a non adaptive approximation to the Riemannian metric which was used to develop a new version of

Constrained Adaptive Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM). The developed technique outperforms the conventional algorithm in convergence, yet computationally efficient. The numerical simulations confirm its suitability for adaptive beamforming, adaptive null steering and multiple interference rejection. Further, to aviod or minimise misadjustment or residual error due to fixed step size, a variable step size scheme has also been developed. This adaptive or variable step size algorithm is quicker than the fixed step algorithm. Thus if we are able to approximate the Riemannian metric as a fixed matrix than the technique can further be improved through adaptive stepsize.

However the non adaptive metric used for the simulations in CANA-NAM effectively only changes the step size since the metric is a scaled identity. Similarly with the change in number of sources and or the number of antennas, the same metric might not work and can lead to inaccurate results. Since the algorithm highly depends upon prior knoweldge of the Riemannian metric for approximating the non adaptive metric. Thus a hybrid approach can be followed in which for some initial iterations the CANA algorithm operates in normal mode and after approximating the Riemannian metric shifts to the non adaptive metric mode. Nevertheless the formulation of this hybrid approach can be left as a future work.

The polar space representation of both CLMS and CANA provides further insight into to the algorithm formulation. It is worth noting that the natural and Euclidean gradients only vary in the angle update and the magnitude calculation remains same. This formulation can be extended in spherical as well as to the N dimensions using different realisations of Riemannian metrics.

Chapter 4

Constrained Broadband Array Processing in Wave-Number Domain

Chapter 3 has established the formulation and characteristics of constrained natural gradient based techniques. The convergence properties and complexity of the algorithm has been studied with firm understanding of the technique. However examples and case studies in the previous chapter mostly presented its applications in narrowband domain. This chapter is a detailed description of the broadband constrained array processing in wave-number domain. It comprises of sections on introduction, mathematical development and simulation of the proposed methodology. This chapter provides a closed form and iterative natural gradient based solution whereas an array of uniformly distributed sensors is constrained to produce optimum desired response. The developed technique eliminates the need of the calculation of tap spacing or determination of the optimum sampling, in deriving the temporal information from the signal, as the algorithm directly gives the coefficients of the filters represented in wave-number domain. The technique has been found useful in restoring the capability of cancellation of total number of hostile interferers equal to M-1. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, so the side-lobe one-to-one correspondence remains conserved. Analysis and discussion on the simulations and concepts derived therein are provided in detail followed by conclusion at the end.

4.1 Introduction

Broadband array processing is performed through mainly two types of techniques; frequency based and time based techniques. The time based adaptive array processing incorporates Taped Delay Line (TDL) [56] filter whose order is directly proportional to the fractional bandwidth of the signal. The frequency domain broadband beam forming [14] is achieved by taking Fast Fourier Transform (FFT) of each sensor signal, and then each frequency bin is processed by a narrowband independent processor. Several offshoots of these main techniques like Frequency Domain Frequency Independent Beamforming (FDFIB) and Time Domain Frequency Independent Beamforming (TDFIB) [57] etc., have been reported. Both of these techniques have benefits and disadvantages which make them unsuitable at times. FFT approach is constrained by its inherent time delay, while technique based on time domain processing, is capped by the order of the TDL filter directly proportional to the fractional bandwidth [13, 58].

Another issue with time delay processing is that it loses at least one degree of freedom in mitigating the number of hostile interferers. It has been reported [31] that TDL based broadband beamforming can cancel a maximum of M-2 interferers. Although TDL structure is similar to an FIR, and the tap spacing/temporal sampling observes the Nyquist criteria [59], but it is not always trivial to achieve right delays in order to craft a desired beam pattern.

Sensor Delay Line (SDL) processing is another extension to the time delay processing in which time delays have been introduced by placing extra sensors behind each element. In this scheme a wide band linear array looks like a narrowband planar array [60]. The method is simple to implement as only one coefficient is required for each array element. It also enhances the coverage by expanding the field of view of an array of antenna. Despite this, sensor delay line technique has some obvious disadvantages, since a considerably higher number of sensors would be required with significant calibration efforts. Also for the implementation of very short delays, the sensors are to be placed very near to each other putting further constraints on the proposed technique.

To surmount the adversities in calculations of tap coefficients and determining optimum sampling frequency in implementing FIR type TDL structure, we have developed a wave-number filtering based approach. The technique uses filters represented in wave-number domain associated with each sensor. The signal after filtering is summed by an array processor according to the wave-number constraint. The present technique is also simple, in the sense that spatial and temporal information has been mapped to only one dimension in the wave-number domain. Mathematical analysis and numerical simulations confirm the algorithm’s ability of operating over the full bandwidth of 0 to 2π. The technique has also been found useful in restoring the capability of cancellation of a total number of hostile interferers equal to M-1 for an M-length array. A detailed description of the broadband wave-number processing right from ab initio is given in the following sections.

4.2 Broadband Array Processing in Wave-Number domain

A signal is a time-varying quantity [61] having single or multiple frequencies and can be accordingly termed as narrowband or broadband. In array processing a signal is said to be broadband when all its frequency components not only undergo phase change but also experience a change in magnitude while travelling through an array of sensors, whereas a narrowband signal only experiences a phase shift. A broadband signal integrated over multiple frequencies can be represented as.

$$s(t)=\int e^{j2\pi ft}d\varphi(f) \tag{4.1}$$

Where $d\varphi(f)$ is infinitesimal component of the frequency spectrum defined over the frequency range *f*.

If $d\varphi(f)=A_o\delta(f-f_o)df$

Then the Eq. 4.1 can be written as

$$s(t)=\int e^{j2\pi ft}A_o\delta(f-f_o)df \tag{4.2}$$

This implies that

$$s(t)=A_o e^{j2\pi f_o t} \tag{4.3}$$

Where f_o is the frequency of the source signal; the signal defined thus is termed as the narrowband signal. Since we are concerned with the broadband signals, in this manner the definition of signal in Eq. 4.1 will hold true for the rest of the analysis. Let us consider an array of uniformly spaced *M* sensors, as in Figure 4.1. For a signal source impinging on the array from the direction θ, the signal received at the 1st sensor from a direction θ can be expressed as

$$x_1(t)=s_1(t)+v_1(t) \tag{4.4}$$

Where $s_1(t)$ is the signal received and $v_1(t)$ is the additive noise at the 1st sensor. Similarly

$$x_2(t)=s_1\left(t-\tau_1(\theta)\right)+v_2(t) \tag{4.5}$$

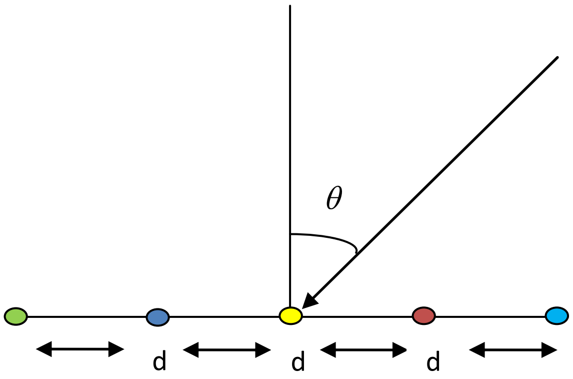

Figure 4.1: Signal impinging on an ULA from an arbitrary direction θ

Finally the radiation at the last sensor of the array is thus written as

$$x_M(t) = s_1\left(t - \tau_{M-1}(\theta)\right) + v_M(t) \tag{4.6}$$

Where $\tau_m(\theta) = md\sin(\theta)/c$ for $1 \le m \le M-1$ (4.7)

Whereas *d* is the distance between the adjacent antenna elements in terms of wavelength and *c* is the velocity of the incident radiation.

Using the definition of $s(t)$ Eq. 4.6 can be formulated for *m*th element as

$$x_m(t) = \int e^{j2\pi f\left(t-\tau_m(\theta)\right)} d\varphi(f) + v_m(t) \tag{4.8}$$

Where $v_m(t)$ is the additive noise at the *m*th sensor. Now $2\pi f\tau_m(\theta) = m2\pi fd\sin(\theta)/c$, but $c = f\lambda$. This implies that $2\pi f\tau_m(\theta) = md\sin(\theta)[2\pi/\lambda]$ or $2\pi f\tau_m(\theta) = mkd\sin(\theta)$, where *k* is the wave-number. Using the above defined quantities the Eq. 4.8 can be written in wave-number domain as

$$x_m(t) = \int e^{jk(ct - md\sin(\theta))} d\varphi(kc/2\pi) + v_m(t) \tag{4.9}$$

Where $k = 2\pi/\lambda$ and $kc = 2\pi f$ without loss of generality $d\varphi(kc/2\pi)$ can be written as $d\varphi(k)$.

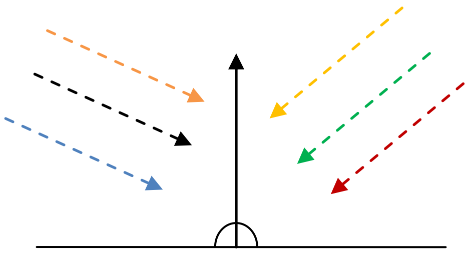

Figure 4.2: Multiple broadband sources impinging on an ULA

Further to representation in wave-number domain, Eq. 4.9 can also be written as

$$x_m(t) = \int e^{jk(ct - md\sin(\theta))} d\varphi(k) + v_m(t) \tag{4.10}$$

Now if there are N broadband sources incident on the array from direction $\left[\theta_p, p = 1, 2..N\right]$ as depicted in Figure 4.2, then the signal received at the 1st sensor is the sum of signals from all the sources and can be written as.

$$x_1(t) = \sum_{p=1}^{N} s_p(t) + v_1(t) \tag{4.11}$$

Where $s_p(t) = \int e^{jkct} d\varphi_p(k)$ is broadband signal for p^{th} source as given in Eq. 4.1. Whereas $d\varphi_p(k)$ is infinitesimal component of the wave-number spectrum of p^{th} source at time *t*, and *N* is the total number of sources. Then the signal received at the m^{th} element of the array due to all the sources combined can be formulated as

$$x_m(t) = \sum_{p=1}^{N} \int e^{jk(ct - md\sin(\theta_p))} d\varphi_p(k) + v_m(t) \tag{4.12}$$

4.2.1 Wave-Number Processing

Let us consider that several broadband signals as depicted in Figure 4.2 are impinging on the uniform linear array of Figure 4.3. The sensors are placed at a distance of half wavelength defined at the midpoint of wave-number bandwidth, k_o. Let there be *M* sensors and *N*

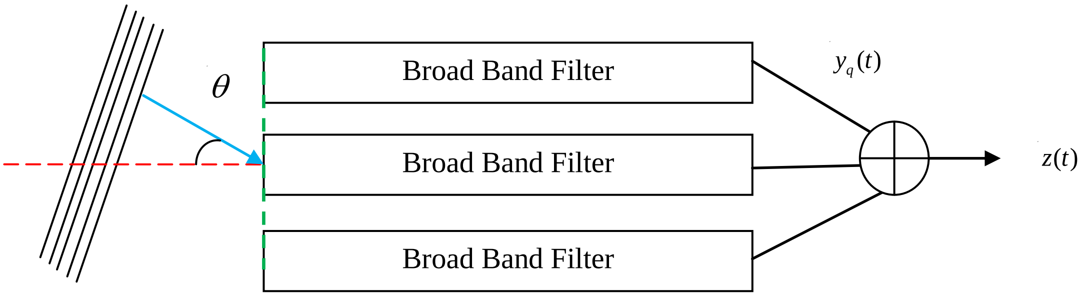

Figure 4.3: Broadband filters attached with each sensor

broadband sources impinging on the array from several directions. The signal developed at each antenna comprising signals from all sources, can be processed by inserting broadband wave-number filters at each sensor. Weights of these filters can be adjusted according to the desired spatial and temporal frequency response stated in wave-number. The algorithm we have developed not only crafts the spatial pattern but also gives the desired optimum response in wave-number, hence eliminating the need for separately finding the number of delays and the optimum spacing between the taps for each sensor channel as in case of conventional time delay beamforming. Output from each of the wave-number filters can be derived as

$$y_q(t) = \int h_q(t-\tau)x_q(\tau)d\tau \tag{4.13}$$

Where $y_q(t)$ is the output of the filter/channel of the q^{th} sensor. Using the definition developed in Eq. 4.12

$$x_q(\tau) = \sum_{p=1}^{N} \int e^{jkc\tau - mk_o d\sin(\theta_p)} d\varphi_p(k) + v_q(\tau) \tag{4.14}$$

$$y_q(t) = \int h_q(t-\tau) \sum_{p=1}^{N} \int e^{jkc\tau - mk_o d\sin(\theta_p)} d\varphi_p(k) \tag{4.15}$$

Now the cumulative output of all sensor channels can be written as

$$z(t) = \sum_{q=1}^{M} y_q(t) \tag{4.16}$$

Using the values from Eq. 4.15

$$z(t)=\sum_{q=1}^{M}\int h_q(t-\tau)\sum_{p=1}^{N}\int e^{jkc\tau-mk_o d\sin(\theta_p)}d\varphi_p(k) \qquad (4.17)$$

Intuitively it can be written as

$$z(t)=\sum_{q=1}^{M}\sum_{p=1}^{N}\int\int h_q(t-\tau)e^{jkc\tau-mk_o d\sin(\theta_p)}d\varphi_p(k)d\tau \qquad (4.18)$$

To underpin the concepts developed above, the algorithm can be further illustrated; for example if we have 3 sensors and 2 sources the above expression can be reformulated as a vector matrix product

$$e^{jkc\tau}d\tau\begin{bmatrix}h_1(t-\tau) & h_2(t-\tau) & h_3(t-\tau)\end{bmatrix}\begin{bmatrix}e^{-jk_o d\sin(\theta_1)} & e^{-jk_o d\sin(\theta_2)}\\ e^{-j2k_o d\sin(\theta_1)} & e^{-j2k_o d\sin(\theta_2)}\\ e^{-j3k_o d\sin(\theta_1)} & e^{-j3k_o d\sin(\theta_2)}\end{bmatrix}\begin{bmatrix}d\varphi_1(k)\\ d\varphi_2(k)\end{bmatrix} \qquad (4.19)$$

This can also be written as

$$e^{jkc\tau}d\tau\begin{bmatrix}\sum_{q=1}^{3}h_q(t-\tau)e^{-jk_o qd\sin(\theta_1)} & \sum_{q=1}^{3}h_q(t-\tau)e^{-jk_o qd\sin(\theta_2)}\end{bmatrix}\begin{bmatrix}d\varphi_1(k)\\ d\varphi_2(k)\end{bmatrix} \qquad (4.20)$$

or

$$\sum_{p=1}^{2}\sum_{q=1}^{3}h_q(t-\tau)e^{-jk_o qd\sin(\theta_p)}d\varphi_p(k)e^{jkc\tau}d\tau \qquad (4.21)$$

Therefore $z(t)$ may be recast as

$$z(t)=\iint e^{jkc\tau}d\tau\boldsymbol{h}^H(t-\tau)\mathbf{A}(k_o,\theta)d\boldsymbol{\varphi}(k) \qquad (4.22)$$

Where

$$\boldsymbol{h}(t-\tau)=\begin{bmatrix}h_1(t-\tau) & h_2(t-\tau)\ldots & h_M(t-\tau)\end{bmatrix}^H \qquad (4.23)$$

$$d\boldsymbol{\varphi}(k)=\begin{bmatrix}d\varphi_1(k) & d\varphi_2(k)\ldots & d\varphi_N(k)\end{bmatrix}^T \qquad (4.24)$$

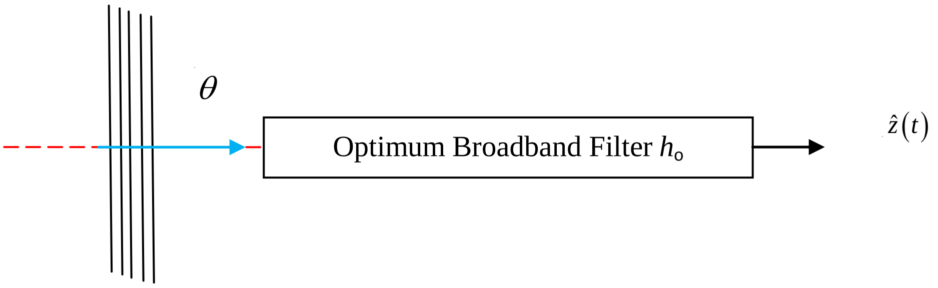

Figure 4.4: Pictorial depiction of broadband constraint for broadside look direction incidence

and

$$\mathbf{A}(k_o,\theta)=\begin{bmatrix} e^{-jk_od\sin(\theta_1)} & e^{-jk_od\sin(\theta_2)} & & e^{-jk_od\sin(\theta_N)} \\ e^{-j2k_od\sin(\theta_1)} & e^{-j2k_od\sin(\theta_2)} & ..... & e^{-j2k_od\sin(\theta_N)} \\ e^{-jMk_od\sin(\theta_1)} & e^{-jMk_od\sin(\theta_2)} & & e^{-jMk_od\sin(\theta_N)} \end{bmatrix} \tag{4.25}$$

4.2.2 Constraint Formulation

It is required that the array processor should optimally respond to the signal coming from a desired direction and discriminate against the interferers coming from the other directions. In other words, for the look direction signal the array processor comprised of all the wave-number filters should behave like a single optimum filter whose response is known a priori and which should present nulls to hostile interferers originating from rest of the directions. If we let look direction to be perpendicular to the line of sensors the constraint can be depicted as in Figure 4.4. In case the look direction is other than the broadside to the line of sensors, the array can be steered to the desired source; this will be explained in a separate section on steering of wave-number filters.

Equivalently the constraint formulation can be expressed in mathematical form as below.

$$\hat{z}(t)=\int h_o(t-\tau)s_o(\tau)d\tau \tag{4.26}$$

Where $\hat{z}(t)$ is the optimum output of the array processor for look direction signal $s_o(\tau)$. For the broadside incidence, it is required that all wave-number filters associated with each of the sensor, should implement the above response hence; the optimum output can be expressed as

$$\hat{z}(t)=\sum_{p=1}^{M}\int h_p(t-\tau)s_o(\tau)d\tau \tag{4.27}$$

$$\hat{z}(t)=\int \boldsymbol{c}^T\boldsymbol{h}(t-\tau)s_o(\tau)d\tau \tag{4.28}$$

where $\boldsymbol{c}^T = [1 \quad 1 \quad 1 \ldots \quad 1]$ (4.29)

So the constraint in its most basic form will appear as

$$\boldsymbol{c}^T \boldsymbol{h}(t-\tau) = h_o(t-\tau) \tag{4.30}$$

4.2.3 Solution for the Optimum Weights

Once the constraint has been realised we can solve for the optimum weights for the array processor to implement that constraint. The optimum weights can be achieved by minimising the average output power while maintaining the look direction frequency response. The total output of the array already derived can be reproduced as follows

$$z(t) = \sum_{q=1}^{M} y_q(t) \tag{4.31}$$

Expanding the above relationship as the sum of individual filters attached to each sensor

$$z(t) = \sum_{q=1}^{M} \int h_q(t-\tau) x_p(\tau) d\tau \tag{4.32}$$

The above expression can also be realised as the vector product as below.

$$z(t) = \int \boldsymbol{h}(t-\tau)^H \boldsymbol{x}(\tau) d\tau \tag{4.33}$$

Where $\boldsymbol{x}(\tau) = \left[x_1(\tau) \quad x_2(\tau) \quad x_3(\tau) \quad \ldots x_M(\tau) \right]^T$ and

$$\boldsymbol{h}(\tau) = \left[h_1(\tau) \quad h_2(\tau) \quad h_3(\tau) \quad \ldots h_M(\tau) \right]^H$$

In wave-number spectrum $z(t)$ can be defined as

$$z(t) \hat{=} \int z(k) e^{jkct} dk \tag{4.34}$$

Using the concepts in Eq. 4.33 and Eq. 4.34 the output of array processor in wave-number representation can be written as below

$$z(k) = \boldsymbol{h}^H(k) \boldsymbol{x}(k) \tag{4.35}$$

Where $\boldsymbol{h}(k)$ is the broadband filter stated in wave-number domain, therefore Eq. 4.33 can be written as

$$z(t)=\int \boldsymbol{h}(k)\boldsymbol{x}(k)e^{jkct}dk \tag{4.36}$$

So the mean output power of the wave-number domain array processor is

$$E\left[z^2(t)\right]=E\int\int dk_1dk_2\boldsymbol{h}(k_1)\boldsymbol{x}(k_1)\boldsymbol{x}^H(k_2)\boldsymbol{h}^H(k_2)e^{jk_1ct-k_2ct} \tag{4.37}$$

Please note that the auto correlation is only defined at $\delta(k_1-k_2)$ so that

$$E\left[\boldsymbol{x}(k_1)\boldsymbol{x}^H(k_2)\right]=\boldsymbol{\Psi}(k)\delta(k_1-k_2) \tag{4.38}$$

where

$$\boldsymbol{\Psi}(k)=E[\mathbf{A}^T\boldsymbol{\Phi}(k)\mathbf{A}]+\boldsymbol{\Phi}_v(k) \tag{4.39}$$

Where $\boldsymbol{x}(k)=\mathbf{A}^T(k_o,\theta)\boldsymbol{s}(k)+\boldsymbol{v}(k)$. The $\boldsymbol{\Phi}_v(k)=E\left[\boldsymbol{v}(k)\boldsymbol{v}^H(k)\right]$ is noise while $\boldsymbol{\Phi}(k)=E\left[\boldsymbol{s}(k)\boldsymbol{s}(k)^H\right]$ is a diagonal matrix of source wave-number spectra. Where $\boldsymbol{v}(k)=\left[v_1(k)\quad v_2(k)\ldots\quad v_M(k)\right]^T$ and $\boldsymbol{s}(k)=\left[s_1(k)\quad s_2(k)\ldots\quad s_M(k)\right]^T$. A single element of the matrix **A** can be read as $A_{p,q}(k_o,\theta)=e^{-jkpq\sin(\theta_p)}$.

Consequently the Eq. 4.37 can be written as

$$E\left[z^2(t)\right]=\int dk\boldsymbol{h}(k)\boldsymbol{\Psi}(k)\boldsymbol{h}^H(k) \tag{4.40}$$

In wave-number domain the constraint Eq. of 4.30 takes the form as below

$$\boldsymbol{c}^T\boldsymbol{h}(k)=h_o(k) \tag{4.41}$$

$$\boldsymbol{c}^T\boldsymbol{h}(k)\text{-}h_o(k)=0 \tag{4.42}$$

Thus to minimise output power of Eq. 4.40 subject to the constraint as in 4.42 the cost function, using Lagrange multipliers can be formulated as

$$J=E\left[z^2(t)\right]+\lambda(k)\int dk\left[h_o(k)-\boldsymbol{c}^T\boldsymbol{h}(k)\right] \tag{4.43}$$

$$J = \int dk\boldsymbol{h}(k)\boldsymbol{\Psi}(k)\boldsymbol{h}^H(k) + \lambda(k)\int dk\left[h_o(k) - \boldsymbol{c}^T\boldsymbol{h}(k)\right] \tag{4.44}$$

Differentiating above Eq. 4.44 with respect to $\boldsymbol{h}(k)$ and putting it equal to zero will give the optimum weights in wave-number domain leading to

$$\boldsymbol{h}(k) = \boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\lambda(k) \tag{4.45}$$

But from Eq. 4.39 $h_o(k) = \boldsymbol{c}^T\boldsymbol{h}(k)$, using Eq. 4.44, this condition leads to

$$h_o(k) = \boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\lambda(k) \tag{4.46}$$

or

$$\lambda(k) = \left[\boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\right]^{-1} h_o(k) \tag{4.47}$$

Putting these values back in Eq. 4.44, the optimum filter in wave-number domain can be derived as

$$\boldsymbol{h}_0(k) = \boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\left[\boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\right]^{-1} h_o(k) \tag{4.48}$$

Inserting the value of optimum filter in the minimization equation above, the output of the optimum processor comes out to be

$$E\left[z^2(t)\right] = \int dk\boldsymbol{h}_o^H(k)\boldsymbol{h}_o(k)\left[\boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\right]^{-1}$$

or

$$E[z^2(t)] = \int dk\left|\boldsymbol{h}_o(k)\right|^2\left[\boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\right]^{-1} \tag{4.49}$$

The optimum filter in Eq. 4.48 can be iteratively approximated using the constrained natural gradient algorithm in wave-number domain. By following the similar arithmetic as adopted for Eq. 3.28, it can be shown that optimum weight is given as

$$\boldsymbol{h}_{n+1}(k) = \mathbf{P}[\mathbf{I} - \mu\mathbf{G}^{-1}\boldsymbol{\Psi}(k)]\boldsymbol{h}_n(k) + \boldsymbol{b}(k) \tag{4.50}$$

Where $\boldsymbol{b}(k) = \mathbf{G}^{-1}(k)\boldsymbol{c}\left[\boldsymbol{c}^T\mathbf{G}^{-1}(k)\boldsymbol{c}\right]^{-1} h_o(k)$ and $\mathbf{P} = \left[\mathbf{I} - {}^{\mathbf{G}^{-1}(k)\boldsymbol{c}\boldsymbol{c}^T}\!\big/_{\boldsymbol{c}^T\mathbf{G}^{-1}(k)\boldsymbol{c}}\right]$

The Riemannian metric **G** in this case can be written as

$$\mathbf{G}^{-1}(k)=\left[\mathbf{I}-\boldsymbol{h}(k)\boldsymbol{h}^{H}(k)\right].$$

4.2.4 Numerical Simulations

Let there be 8 sensor and 3 broadband sources impinging on an array of Figure 4.3. As depicted, let each sensor is equipped with a wave-number based broadband filter. The signal after filtering operation is summed to give optimum response, while maintaining the constraint as defined by Eq. 4.40.

Consider three sources incident upon the array having wave number spectra given respectively as

$$\Phi_0(k)=\frac{A_0}{1+a_0{}^2-2a_0\sin\left(k+2k_2\right)} \tag{4.51}$$

$$\Phi_1(k)=\frac{A_1}{1+a_1{}^2-2a_1\cos\left(k\right)} \tag{4.52}$$

$$\Phi_2(k)=\frac{A_2}{1+a_2{}^2-2a_2\cos\left(k-k_2\right)} \tag{4.53}$$

Where let

$A_0=1, A_1=0.15, A_2=0.15$

$a_0=0.8, a_1=0.75, a_2=0.45$

Let k lie in the range $(0\le k\le 2\pi), k_2=\pi/4$

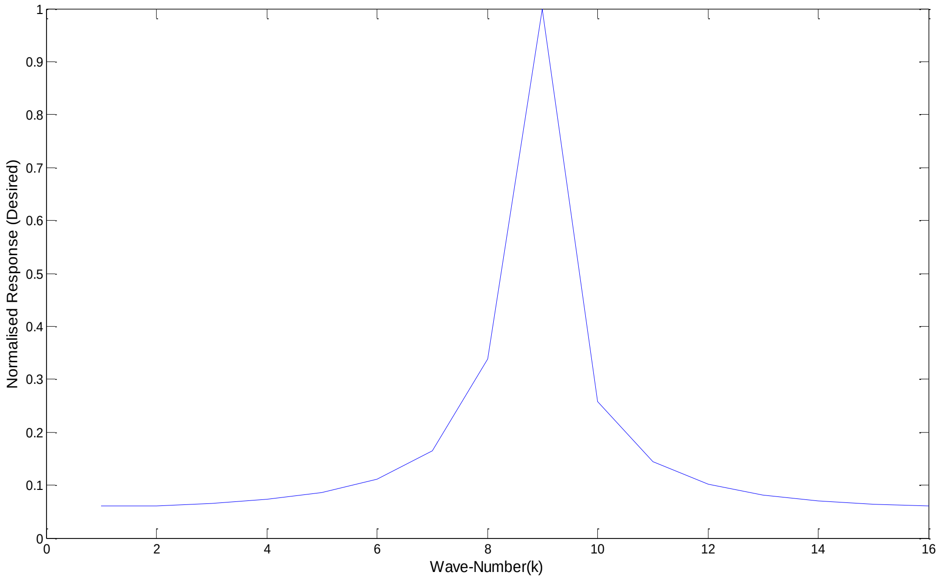

Figure 4.5: Normalised desired response in the look direction

Here we have used normalised values of the frequency and *c*, such that *k* lies in the range (0, 2*π*). Let the desired signal $\Phi_0(k)$ be the incident broadside at an angle $\theta = 0^{\mathrm{o}}$ with respect to the line perpendicular to the line of sensors. Other sources $\Phi_1(k)$ and $\Phi_2(k)$ are at angles 29^{o} and -42^{o} respectively and are the interferers or unwanted signals. These signals need to be suppressed while maintaining the desired frequency response for the signal source $\Phi_0(k)$.Let the desired filter response in wave number domain, known a priori, shown in Figure 4.5 is given as

$$h^{opt}(k) = \frac{1}{1 + a^{opt}\left(\mathrm{e}^{jk}\right)} \tag{4.54}$$

Where $a^{opt} = 0.9$ and $(0 \le k \le 2\pi)$. For *N* sources and *M* sensors, the optimum wave number filters in Eq. 4.48 can be reproduced here for further elaboration, thus

$$\boldsymbol{h}_0(k) = \boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\left[\boldsymbol{c}^T\boldsymbol{\Psi}^{-1}(k)\boldsymbol{c}\right]^{-1} h_o(k)$$

Where $\boldsymbol{h}_0(k)$ is the vector of wave number filters attached to the sensors.

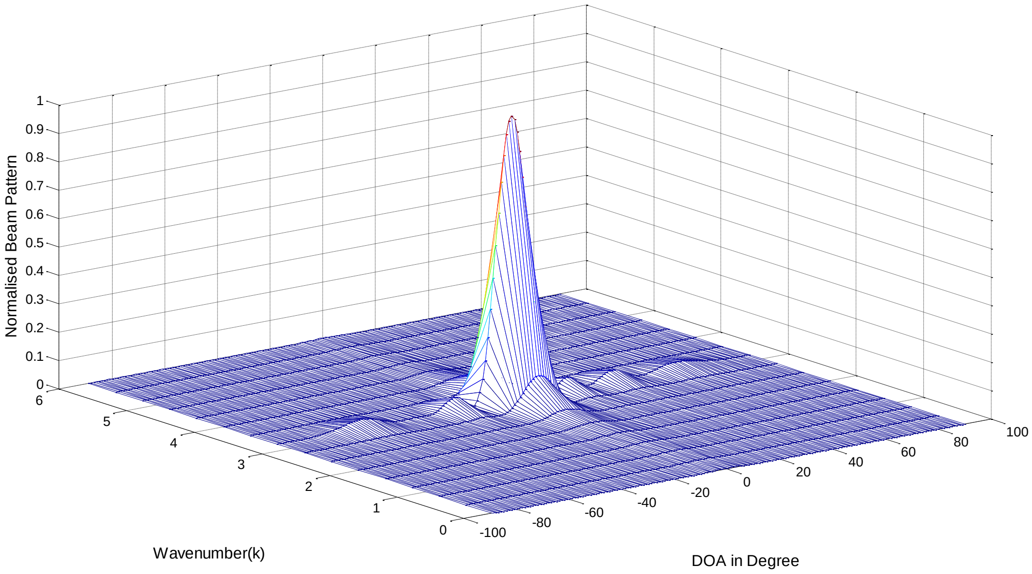

Figure 4.6: Normalised beam pattern of the broadband array

$$\boldsymbol{h}_0(k) = [h_1(k) \quad h_2(k) \quad h_p(k) \qquad h_M(k)]^H \tag{4.55}$$

Here $h_1(k)$ is the wave number response of the filter attached to sensor 1, $h_2(k)$ is attached to sensor 2, etc. The matrix $\boldsymbol{\Psi}(k)$ here is the cumulative correlation matrix corresponding to all signal sources. To materialise this correlation matrix the above expressions can be re-stated in the following manner: Matrix **A** can also be written as row vector of steering vectors corresponding to the sources from different directions as

$$\mathbf{A} = \left[\boldsymbol{a}(k_o, \theta_1) \quad \boldsymbol{a}(k_o, \theta_2) \quad \boldsymbol{a}(k_o, \theta_p) \qquad \boldsymbol{a}(k_o, \theta_N)\right] \tag{4.56}$$

Where any $\boldsymbol{a}(k_o, \theta_p)$ is Mx1 steering or direction vector corresponding to source at θ_{p} and is given by

$$\boldsymbol{a}(k_o, \theta_p) = \left[e^{-jk_o d \sin\theta_p} \qquad e^{-j2k_o d \sin\theta_p} \qquad e^{-jqk_o d \sin\theta_p} \qquad e^{-jMk_o d \sin\theta_p}\right]^T \tag{4.57}$$

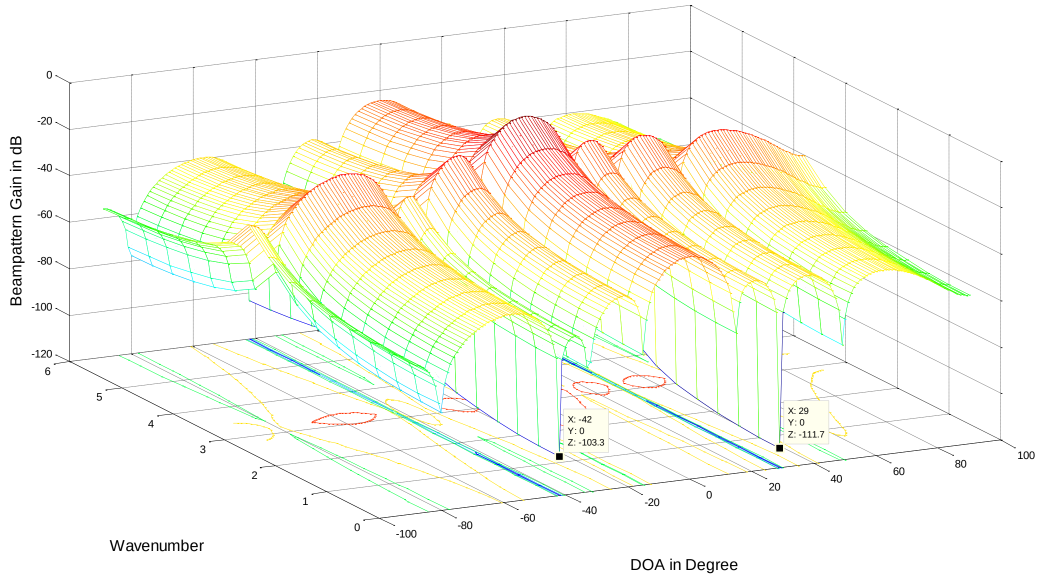

Figure 4.7 Broadside beam pattern gain in dB with interferers at 29^{o} and -42^{o}

d is the fixed uniform distance between sensors defined on the midpoint k_o of the array bandwidth represented in wave-number domain. Using this definition of **A**, we can derive an expression for the correlations of all signal sources.

$$\mathbf{\Psi}(k)=\sum_{p=1}^{N}\mathbf{\Phi}_p(k)\boldsymbol{a}\left(k_o,\theta_p\right)\boldsymbol{a}^*\left(k_o,\theta_p\right) \qquad (4.58)$$

The normalised desired response outlined in Eq. 4.54, has been plotted against wave-number in Figure 4.6. As the desired response is known or to set a priori, so we can force any response and see whether our algorithm behaves accordingly or not. We will explain latter how this response can be manoeuvred to give a more general response like broadband MVDR [38]. Figure 4.6 is the normalised broadband beam pattern plotted against wave-numbers in the range of $(0\leq k\leq 2\pi)$ and direction of arrival in the range of -90^{o} to 90^{o}. As constrained by the desired response, the array pattern is exactly in conformance with the pattern given in Figure 4.5. Maxima can be clearly seen at the desired angle of arrival and at the desired wave-number as specified earlier. Frequency nulls/minima deemed necessary are evident from the normalised plot. Since the interferers also constitute the same bandwidth, though slightly shifted by k_2 so they must be suppressed through the spatial constraints imposed on the array output. However the type of plot in Figure 4.6 is unable to give spatial description

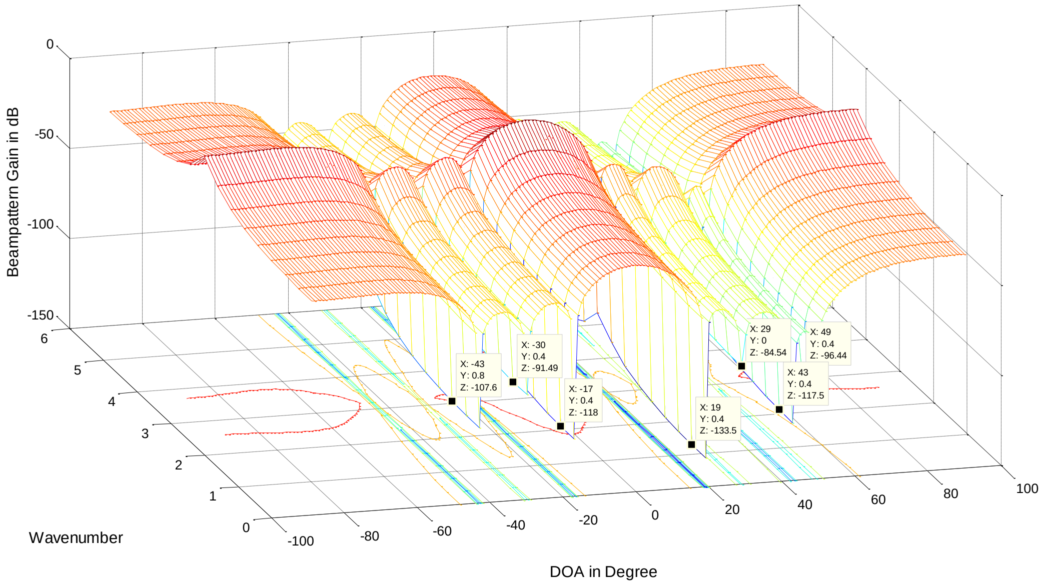

Figure 4.8: 7 Broadband interferers rejected by 8 element broadband array, interferers are present at 19^{o} ,29^{o} ,43 ^{o} ,49^{o} ,-17^{o} ,-30^{o} and -43^{o}

of interferers. A beam pattern plotted in dB will reveal more information as to how the interferers are treated by the algorithm.

Figure 4.7 is the array beam pattern plotted in dB. Clearly the nulls can be seen at the position of broadband interferers while maintaining the desired frequency response at $\theta = 0$. The interferers are suppressed for the whole frequency range of $(0 \leq k \leq 2\pi)$ in the wave-number domain. It may be noted that the array response at the null positions of broadband interferers at 29^{o} and -42^{o} is 100 dB below than the main lobe of the desired filter response from all wave-numbers. The similar paradigm of inserting deeper nulls at the spatial position of the stronger or the one possessing higher interference-to-noise ratio is observed here as well.

The wave-number domain filtering also imparts a very critical improvement in the array processing paradigm by quantitatively enhancing the number of interferes to be cancelled. The algorithm is capable of rejecting *M*-1 number of interferers as opposed to *M*-2 capability of TDL based arrays with space-time processing. This 1 degree of improvement can be used for placing an additional constraint e.g. side-lobe level etc. Figure 4.8 demonstrates rejection of 7 broadband interferers by an 8 element uniform linear array. The nulls, on average are 100 dB below than the main response. It may be noted that the power of nulls can be controlled specifying further constraints on the solution. The trade-off is that we lose extra degrees of freedom. Also it is only possible if the position of interferers is known a priori.

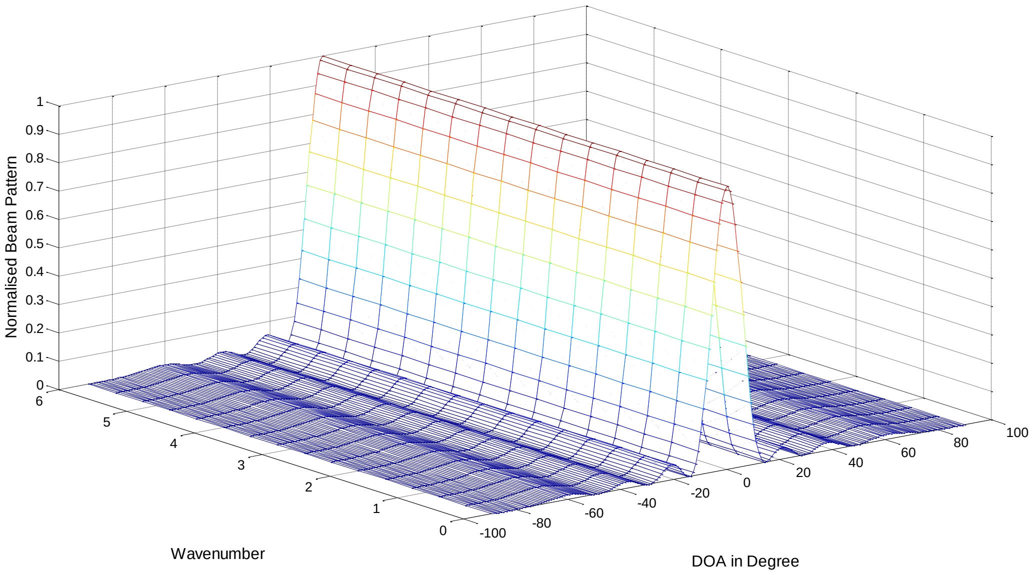

Figure 4.9: The constant beam-width main-lobe response for wave-number based beamforming

4.3 Constant Main-lobe Response

Beamformer with constant main lobe response over the frequency of interest is desirable in many applications such as in underwater acoustics, ultrasonic acoustic imaging and communications etc [13, 58]. The arrays designed at a fixed middle frequency tend to expand main-lobe at lower frequencies and contract at higher frequencies. Several solutions have been provided to solve the issue. But the proposed solutions are either constrained by the fractional bandwidth or sacrifice extra degree of freedom to achieve the said objectives. With the wave-number approach we can not only attain this without sacrificing the extra degrees of freedom but can also maintain the condition of constant main lobe for entire range of bandwidth. The optimum response as given by Eq. 4.48 can be constrained to produce constant main-lobe response over the entire bandwidth by changing required optimum response equal to 1 over the entire bandwidth. The normalised response with constant beam-width is shown in Figure 4.9. The order of the response vector is equal to order of the wave-number filter i.e. for each selection/column of wave-number filters the cumulative response should be 1 for the signal source from the direction of interest. This can be realised as multiple narrowband filters tuned at particular wave-number (k), each implementing the spatial constraint as well. The dB beam pattern for this case is shown in Figure 4.10.

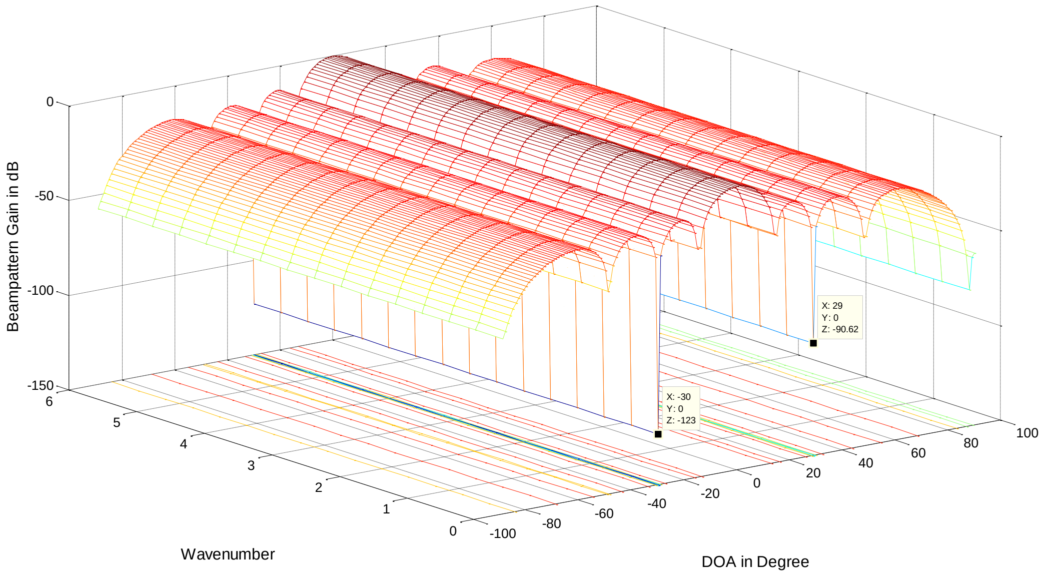

Figure 4.10: The constant beam-width main-lobe response in dB, with interferers at 29^o and -30^o

A constant beam response of 0 dB for all wave-numbers is evident along with nulls towards the direction of two interferers at 29^o and -30^o.

4.4 Narrowband Realisation

The wave-number domain narrowband algorithm can be derived as a special case by placing the required response in Eq. 4.53 equal to unity or some scalar value for a particular wave-number *k*. For the narrowband case the frequency of operation is either defined by the physical separation of sensors or the separation between elements is set according to the frequency of operation. Figure 4.11 and 4.12 show the dB and normalised pattern of the narrowband array. Maxima in the direction of desired signal of interest and two nulls can be seen at the designated places of 29^o and -42^o.

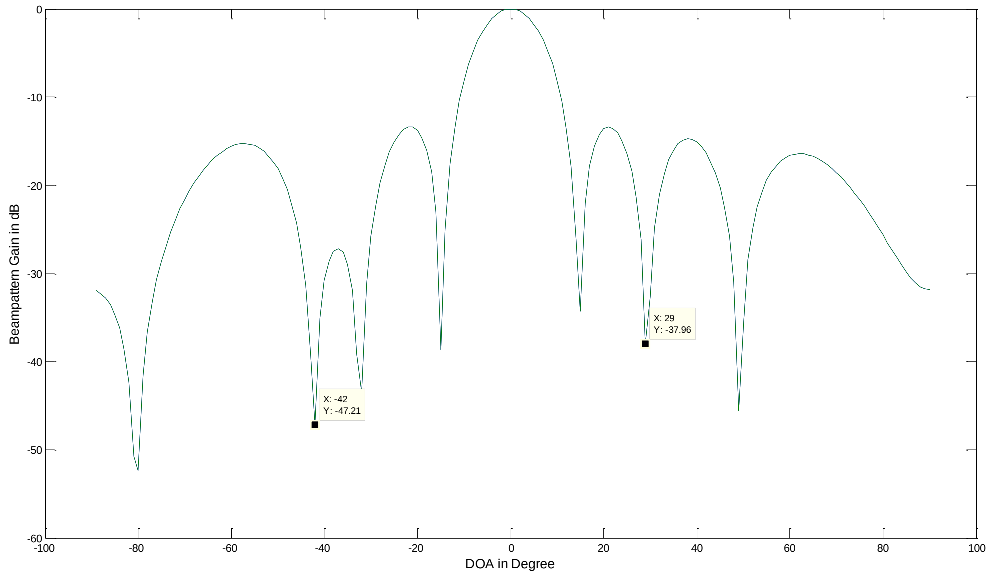

Figure 4.11: Beam pattern gain in dB for narrowband signals at 0^{o} with interferers at 29^{o} and -42^{o}

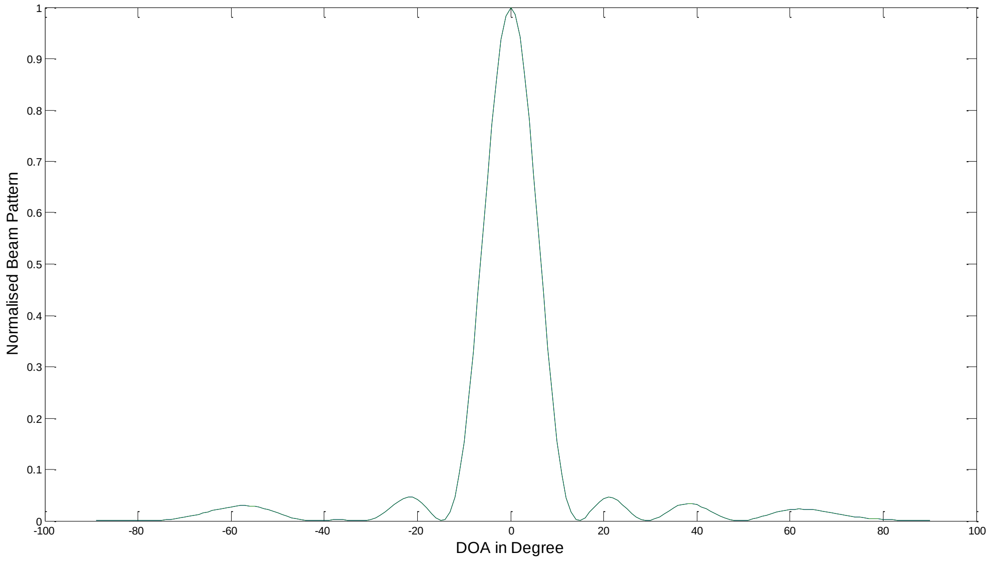

Figure 4.12: Narrowband normalised beam pattern in the desired look direction at 0^{o}

4.5 Steering of the wave-number filters

Beam steering represents the paradigm of changing the direction of the main lobe of a beam pattern towards the direction of interest [24, 29]. The same phenomenon is used for null steering if the position of the jammer/interferer is known a priori. In radio systems, beam steering may be accomplished by mechanically or electronically switching antenna elements or by changing the relative phases of the RF signals driving the elements. If the system is narrowband, relationship between the steered response and the original one is simple and directly related; the former is a circularly shifted version of the later i.e. the side lobe shifted out from one side is simply shifted back from the other side. But steering is a daunting task when it comes to broadband. It was observed that for broadband, the case is not simple and there is no one-to-one correspondence between the original and steered beam response [30].

Beam steering is embedded in our wave-number based technique, thus the beam pattern does not undergo side lobe deterioration after the array response is steered towards a particular direction. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, so the side-lobe one-to-one correspondence is conserved. The phenomenon is underpinned by the following mathematical realisation and MATLAB simulations based upon this. Eq. 4.48 can be rearranged to present embedded capability of steering. Replacing constraint vector $\boldsymbol{c}$ by $\boldsymbol{a}(k_o, \theta_p)$ the optimum weight formula becomes

$$\boldsymbol{h}_0(k) = \boldsymbol{\Psi}^{-1}(k)\boldsymbol{a}(k_o, \theta_p)\left[\boldsymbol{a}(k_o, \theta_p)^T \boldsymbol{\Psi}^{-1}(k)\boldsymbol{a}(k_o, \theta_p)\right]^{-1} h_o(k) \tag{4.59}$$

In this way the filter represented by Eq. 4.59 produces optimum response in the desired direction governed by the steering vector $\boldsymbol{a}(k_o, \theta_p)$. Figure 4.13 demonstrates the steering of the desired response towards an angle -30° while Figure 4.14 is the original un-steered beam pattern towards the direction perpendicular to the line of sensors. The steered response is clearly able to conserve the side lobe symmetry and is further confirmed by steering the response at an angle of +30° as depicted in Figure 4.15.

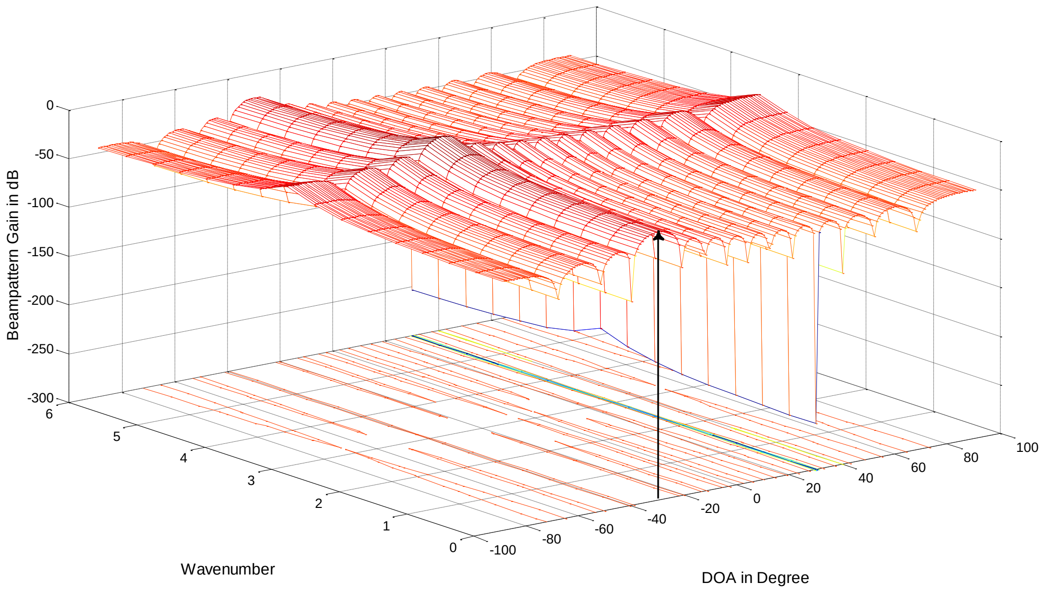

Figure 4.13: Beam pattern steered towards -30^{o} with interferer at 30^{o}

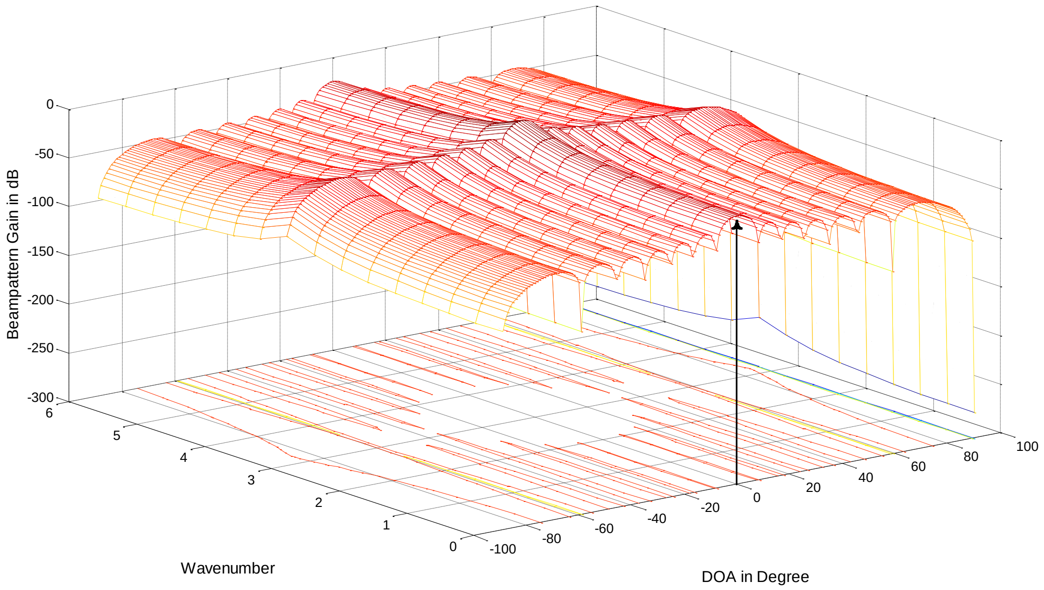

Figure 4.14: Beam pattern at 0^{o}

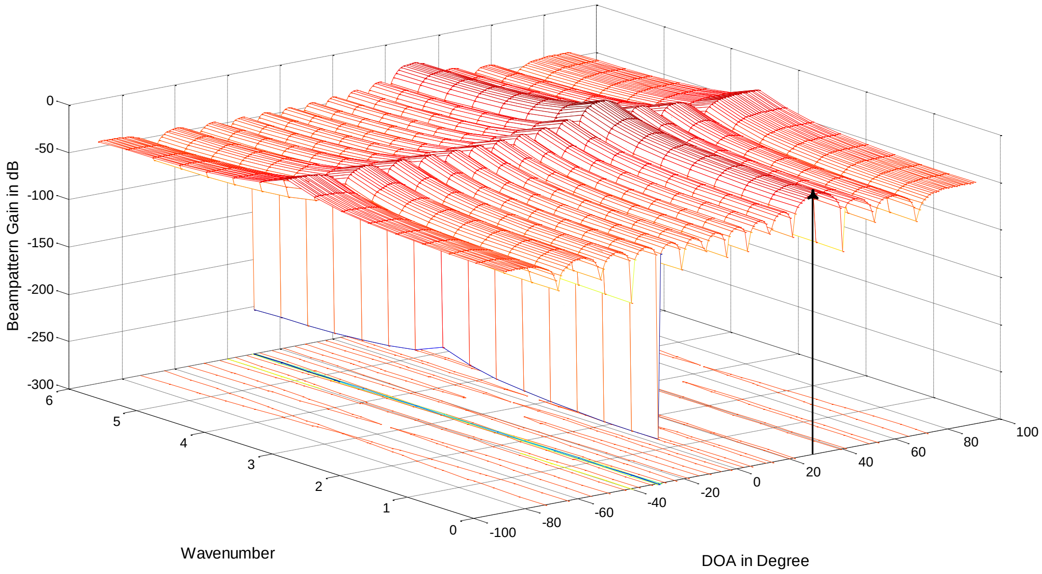

Figure 4.15: Beam pattern steered towards 30^o with interferer at -30^o

4.6 Signal conditioning prior to wave-number processing

In the simulations we used broadband sources with wave-number spectrum. However in practice the signal data received at the sensor array is required to be converted into wave-number domain before it is processed by the broadband wave-number filters. Thus signals are first transformed into wave-number domain through Scaled Fourier Transform and then processed to incorporate the spatial information of the array geometry.

For instance the signal in Eq. 4.1 received at the m^{th} sensor can be defined in wave-number as

$$s_m(k)=\frac{2\pi}{c}\int s_m(t)e^{-jkct}dt$$

or

$$s_m(k)=\frac{2\pi}{c}\left\{S_m(f)\Big|_{f=kc/2\pi}\right\} \tag{4.60}$$

Where $S(f)=\int s(t)e^{-j2\pi ft}dt$ is the FT of the received signal. Since beamforming is same as

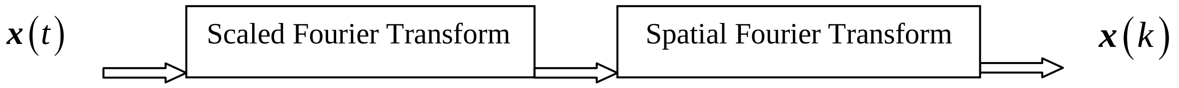

Figure 4.16: Signal conditioning prior to wave-number processing

the Spatial Fourier Transform hence taking the Spatial Transform of the signal after conversion to wave-number would give us the data vector $\mathbf{x}(k)$ in wave-number. Where the Spatial Fourier Transform is represented by the operator matrix **A** given in Eq. 4.25. The process flow diagram for conversion of data vector $\mathbf{x}(t)$ to $\mathbf{x}(k)$ before further processing is shown in the Figure 4.16.

In a real implementation, the FT would be replaced by a DFT processor. Whereby some techniques such as overlap-add or overlap-save [62] may be required to compensate for approximation effects when going from FT to DFT.

Since the hardware implementation is out of the scope of this thesis hence supplementary implementation details are left for future as natural extension to the current research.

4.6 Conclusion

The wave-number filtering technique developed in this chapter eliminates the need of the calculation of tap spacing or determination of the optimum sampling, in deriving the temporal information from the signal, as the algorithm directly gives the coefficients of the wave-number domain filters. The procedure is simple in the sense that spatial and temporal information has been mapped to only one dimension of wave-number domain. Along with operating over the entire bandwidth, the technique has been found useful in restoring the capability of cancellation of total number of hostile interferers equal to *M*-1 for array of *M* sensors.

Despite the fact that the elements in uniform linear array are arranged at half wave length defined at the midpoint of the wave-number spectrum the algorithm prevents from causing main lobe to be contracted at the higher frequencies and expanded at lower frequencies. The power of interferer and null power is inversely proportional to each other, as more powerful interferers get deeper nulls.

Beam steering is embedded in the scenario and the beam pattern does not undergo the side lobe deterioration as in the case of conventional TDL. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, so the side-lobe one-to-one correspondence remains conserved.

Chapter 5

Conclusions and Future Work

Nothing is invented and perfected at the same time

- **John Ray** (November 29, 1627 – January 17, 1705)

The aim of this research has been to develop novel constrained adaptive array algorithms with swift convergence characteristics. The aim has been achieved through the development of the natural gradient based adaptive algorithms for narrowband and broadband array processing, detailed in Chapter 3 and Chapter 4. The choice of natural gradient algorithm was made to exploit its capability in operating on the optimisation surface of the error function.

In order to achieve these objectives this thesis has reviewed basic concepts involved in arrays and the constrained adaptive processing. Important concepts of antennas and some of the primary solutions for narrowband and broadband adaptive array processing have also been examined in detail. Since there is a plethora of adaptive algorithms in constrained adaptive processing domain so it has not been possible to review all of them, however, pioneering techniques in the subject area have been reviewed with emphasis on LMS based solutions for LCMV and MVDR.

This research has lead to the development of following constrained algorithms based upon natural gradient adaptation techniques.

- Constrained Adaptive Natural Gradient Algorithm (CANA)
- Constrained Adaptive Natural Gradient Algorithm with Non Adaptive Metric (CANA-NAM)
- Adaptive Step-size Algorithm
- Constrained Broadband Array Processing in Wave-Number Domain

Part of the research work has been published in a number of international conferences and is being prepared for submission in international journals and transactions. The list of publications based upon some parts of this thesis is given below;

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Constrained Adaptive Natural Gradient Algorithm (Cana) For Adaptive Array Processing, 17th European Signal Processing Conference (EUSIPCO 2009) Glasgow, Scotland, August 24-28, 2009.

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Broadband Constrained Natural Gradient Based Adaptive Processing, 2009 IEEE Workshop on Statistical Signal Processing, Cardiff, Wales ,UK, Aug 31, 2009 - Sep 03, 2009.

- Ijteba-ul-Hasnain Shah and Tariq S Durrani, Constrained Natural Gradient Algorithm with Non Adaptive Metric, IEEE International Symposium on Signal Processing and Information Technology, December 15-18, 2010 - Luxor - Egypt

This thesis is a research monograph on natural gradient based constrained optimisation. Constrained optimisation typically involves minimization of a cost function surface, generally an error surface. The realistic minimization of these cost function problems is not trivial and involves manifolds with non linear optimisation surfaces. Working with such search space that carries the nonlinear manifold introduces certain challenges in the algorithm implementation. In the classic search methods, iterative algorithms rely heavily on approximating first and second order derivatives of the cost function in Euclidean spaces. A new iteration is generated by adding an update increment to the previous update and the process is repeated till a steady state is reached. In order to define algorithms on manifolds, these operations must be translated into differential geometry. Natural gradient modifies the slope according to the actual structure of the optimisation space, thus provides a better approximation to the steepest descent direction.

While parameter estimation is central to several scientific and engineering [3] applications, the most popular, perhaps due to its simplicity is the gradient descent technique. Although its ability to converge quickly and efficiently is limited by certain factors, yet it has numerous applications in signal processing and control applications. The technique gives optimum performance when

- The cost function has single minimum and
- The gradients of the cost function are isotropic in magnitude with respect to any direction away from the minimum.

On the other hand the Riemannian metric structure is observed in optimisation surfaces of applications in signal processing, mechanics and control theory. In such constrained problems, the super-linear convergence speed of the classical conjugate gradient and Newton algorithms is lost because these techniques ignore the structure of the optimisation surface.

The slow convergence of the conventional gradient based approaches is because they are unable to adapt quickly when the slope of the cost function varies widely for the small changes in the adjusted parameters. But if an optimisation scheme exploits the given structure of the underlying space then it does not encounter such phenomenon and is deemed successful with better convergence and accuracy.

Natural gradient modifies the search direction according to the Riemannian structure of the parameter space. Also when used with statistically optimum cost functions, natural gradient overcomes many of the limitations of the Newton's method, which assumes that cost function being minimized is approximately quadratic. The major drawback of natural gradient is considered to be the knowledge required to determine the Riemannian structure of the parameter space, although it can be quite simple for certain applications as observed in CANA-NAM case. Also since many signal and array processing tasks have been firmly defined over years of research, and most of the optimisation surfaces are either known or can be approximated in certain cases hence the Riemannian metric can be approximated to formulate natural gradient algorithm.

Thus based on natural gradient several optimisation algorithms addressing the issues of narrow as well as broadband beamforming have been developed in this thesis.

Complemented with a fixed or adaptive processor, antenna array can discriminate between wanted and unwanted signals in the presence of noise. The constrained adaptive processor optimises the array response according to defined set of spatial constraints such that the output contains the minimal contribution of noise and signals originating from undesired directions. The essence of an adaptive processor is the algorithm comprising of logical finite set of mathematical operations. There are quite a number of adaptive algorithms available for determining the weight vector but most commonly they require desired signal a priori, which is rarely known. However if we know the Direction of Arrival (DoA) of the desired signal and the desired frequency response we can impose some constraints on the array coefficients to adaptively minimize the array output in all directions but the direction of arrival of the desired signal. The only a priori information the algorithm requires is the direction and frequency band of interest. Consequently, the algorithm minimizes total noise power at the array output while maintaining a chosen frequency response in the “look direction”. This thesis has presented several techniques in order to address the constrained array problem.

Constrained Adaptive Natural Gradient Algorithm (CANA) developed in this thesis is a novel technique for constrained adaptive processing based upon the natural gradient. This technique provides more uniform performance than those based upon ordinary gradients, yet simple to implement. Along with the ability of natural gradient algorithm to follow the exact surface of the optimisation space, self correcting feature of constrained optimisation makes the algorithm rapidly converge to steady optimum state for variety of statistically optimum functions. So for a reasonable range of step size the drawback of ordinary gradient of escaping from the cost function surface can be avoided, still maintaining the improved convergence properties.

Many applications such as Thousand Element Array (THEA) [63] and similar processes involving multi functions require the number of elements to be very large. The length of weight vector $\boldsymbol{w}$ is directly propotional to the number of elements for narrowband processing and is MxL for broadband ,whereas M is the number of sensors and L is the order of the filter attached with each sensor. As explained and derived in Chapter 3, Section 3.5, Riemannian metric is a function of weight vector and is updated at each iteration with the update in $\boldsymbol{w}$, thus is the major contributor in algorithm computational complexity.

Hence in these scenarios it would be more appropriate to have a non adaptive Riemannian metric inorder to reduce the number of arithmatic operations at each iteration. On the similar note, we may not be able to find a single non adaptive metric suitable for all arrays, as every antenna array itself exhibits a Riemannian manifold. Due to intrinsic curvature information of an array manifold , the non adaptive metric must be unique. So it would be more appropriate to have an adaptive version of the Riemannian metric to fully exploit the geometric properties of the objective function.

CANA-NAM given in Section 3.5 is a critical extension to the previously developed Constrained Adaptive Natural Gradient Algorithm (CANA), by incorporating a non adaptive metric in place of adaptive Riemannian metric. Inclusion of this fixed metric significantly reduces the complexity of previously developed technique. Complemented with constrained optimisation techniques the algorithm is capable of rapidly adjusting the response of an array of sensors to a signal coming from direction of interest. Numerical simulations confirm its suitability for adaptive beamforming, adaptive null steering and multiple interference rejection.

Further, to aviod or at least minimise the misadjustment or residual error due to fixed step size, a variable step size scheme has also been developed in Chapter 3, Section 3.6. This adaptive or variable step size algorithm is found quicker than the fixed step algorithm. Thus if we are able to approximate the Riemannian metric as a fixed matrix than the technique can further be improved through adaptive stepsize.

As maintained previously, there are mainly two techniques of broadband array processing, frequency based and time based techniques. The time based adaptive array processing incorporates Taped Delay Line (TDL) filter whose order is directly proportional to the fractional bandwidth of the signal. The frequency domain broadband beamforming is done by taking Fast Fourier Transform (FFT) of each sensor signal and then each frequency bin is processed by narrowband independent processor. Several offshoots like Frequency Domain Frequency Independent Beamforming (FDFIB), Time Domain Frequency Independent Beam forming (TDFIB) of these main techniques have been reported. Both of these techniques have plus and minuses which make them unsuitable at times. FFT approach is constrained by its inherent time delay, while time domain is capped by the order of the TDL filter directly proportional to the fractional bandwidth.

Another issue with time delay processing is that it loses at least one degree of freedom in mitigating the number of hostile interferers. It has been reported that TDL based broadband beam forming can cancel a maximum of M-2 broadband interferers [31]. Although TDL structure is similar to an FIR and the tap spacing/temporal sampling observes the Nyquist criteria, but it is not always trivial to achieve right delays in order to craft a desired radiation pattern.

Sensor Delay Line (SDL) processing is another extension to the time delay processing in which time delays have been materialised by putting extra sensors behind each element. In this scheme a wide band linear array looks like a narrow band planar array. The method is simple to implement as only one coefficient is required for each array element. It also enhances the coverage by expanding field of view of an array of antenna. Despite this, sensor delay line technique has some obvious disadvantages, since a considerably higher number of sensors with significant calibration efforts are required. Also for the very short delays, the sensors are to be placed very closely putting further constraints on the proposed technique.

To address these critical issues and lessen the adversities in calculation of tap coefficients and determination of optimum sampling frequency in implementing FIR type TDL structure, we have developed a wave-number filtering based approach. The technique, given in Chapter 4, uses wave-number domain filters behind each sensor. The signal after filtering is summed by an array processor according to the wave-number constraint. The present technique is also simple in the sense that spatial and temporal information has been mapped to only one dimension of wave-number domain. Mathematical analysis and numerical simulations confirm the algorithm's ability of operating over the full bandwidth of 0 to 2π. The technique has been found useful in restoring the capability of cancellation of total number of hostile interferers equal to M-1.

This technique eliminates the need of the calculation of tap spacing or determination of the optimum sampling, in deriving the temporal information from the signal, as the algorithm directly gives the coefficients of the wave-number domain filters. The procedure is simple in the sense that spatial and temporal information has been mapped to only one dimension of wave-number domain.

The issue of array response deterioration with the change in bandwidth/frequency is very critical. Antenna arrays have the tendency of forming wider lobes at lower frequencies, while narrower lobes are formed at higher frequencies. This is due to the fact that arrays are generally of a fixed structure, and are defined at a single wavelength for operating in both narrowband and wideband scenario. Despite the fact that the elements in uniform linear array are arranged at half wave length (defined at the midpoint of the wave-number spectrum), the algorithm developed in Chapter 4 prevents the main lobe from contracting at the higher frequencies and expanding at lower frequencies.

Beam steering is embedded in the scenario and steering the array response from one angle to other does not cause non-uniform formulation of side lobes as this happens in the case of TDL. Since the whole filter is steered towards the direction of interest, rather than the individual elements of the TDL/FIR filters, so the side-lobe one-to-one correspondence remains conserved.

Thus through these techniques this thesis has been able to solve the constrained adaptive array problem with demonstrated improved convergence, accuracy and simplicity. The numerical simulations and mathematical results warrant the techniques' usefulness for range of applications in commercial as well as the defence. Besides, the thesis gives ample thought to the formulation of the Riemannian metric for a certain problem, as this is the main component on which whole concept of the natural gradient is build upon.

5.1 Future work

The techniques developed in this thesis are computationally efficient in addition to their good convergence characteristics. Both of these properties make the algorithms very suitable for the real time implementation with DSP or FPGA based platforms.

The CANA-NAM algorithm highly depends upon prior knoweldge of the Riemannian metric for approximating the non adaptive metric. Thus a hybrid approach can be followed in which for some initial iterations the CANA algorithm operates in normal mode and after approximating the Riemannian metric shifts to the non adaptive metric mode. The formulation of this hybrid approach can be carried as a future work.

In this thesis all the analysis was done on the basis of linear array of sensors. However practically there are a number of geometric arrangements possible for antenna arrays, so the analysis needs to be extended incorporating different geometries. An interesting application in this regard would be to use the natural algorithms for conformal arrays [64].

The work can be potentially expanded to a number of scenarios incorporating different types of constraint. For example there are several types of soft and hard constraints that can be imposed to formulate the natural gradient constrained adaptive techniques. These constraints are

- Correlation Constraint
- Eigen Vector Constraint
- Derivative Constraints
- SRV Constraints
- Norm Constraints etc.

We also know that the convergence characteristics of gradient based techniques heavily depend upon the eigenvalue spread of the input correlation matrix. So it would be useful to formulate and analyse constrained natural gradient in the eigenvalue/eigenvector domain.

Most of the time we take the “Far Field” assumption such that a wave impinging on an antenna array is considered as a plane wave. How the algorithm will be transformed if this assumption is no longer valid? The situation can arise where non-identical sensors or devices’ network such as Body Area Networks [65] where all the equipment; mobile phone, IPod, laptop etc are required to be connected with each other, the far field assumption is invalid. Also if we are using multiple devices in closets such as cars and boats we will encounter the near field phenomenon. Thus the current work needs to be extended to incorporate near field assumptions.

5.1.1 Multipath Problem

The current research can be extended to multipath problem especially if the algorithms are to be used for the urban communications where there is an abundance of the delayed replicas of the signals as well as interfering sources. For example, we consider the Linearly Constrained Minimum Variance (LCMV) beamformer. We have seen that in an LCMV problem where output power $\boldsymbol{w}^H\mathbf{R}\boldsymbol{w}$ is minimised subject to $\mathbf{c}^T\boldsymbol{w} = \boldsymbol{f}$ where $\mathbf{R}$ is the covariance matrix of observed array data $\mathbf{c}$ is the constraint vector/matrix and $\boldsymbol{f}$ is the response vector or set of constraints. The constraint will ensure that beamformer will have the desired response set out by the constraint equation. We have observed that if the interferers are uncorrelated then the minimization will only suppress them, but if there are interferences which are correlated or the delayed replicas of the desired signal itself then the conventional minimization approach will either cancel part of the signal or can even result in complete cancellation of the desired signal. The wave-number broadband solution with multiple beams, developed in Chapter 4 can be extended to cope with the multipath problem.

5.1.2 Cognitive Radio

There are number of open research problems in cognitive radio which can be solved through the help of the techniques developed in this thesis. For example the issue of power control in cognitive radios is very critical. The power control problem in cognitive radios can be formulated as [66]

“Make the spectrum holes available for employment by secondary users efficiently, subject to the constraint that the received power in each spectrum hole does not exceed a prescribed limit.”

Once the issues of defining the criterion of how to decide when to put the power constraint on the second user willing to use the available empty hole and issues such as the first user having to decide what would be the power level of the second user, multi beam narrowband solution can be used to serve the purpose. In case where the users are at different frequencies then broadband constrained wave-number techniques can be used to implement cognitive power control. In this way the technique can permit, two way controls

- Direction of arrival power control
- Frequency power control

We can also define primary users, secondary users, tertiary users and their power level through adjusting the optimum response matrix, whose columns are the broadband response vectors with respect to each user.

5.1.3 GPS Interference Cancellation

In GPS applications generally the signal is very weak, and can be easily interfered or jammed. We have observed that constrained broadband array processing in wave-number domain can suppress the jamming not only in the wave-number/temporal domain, but also in the spatial domain. Therefore the algorithm can be applied for jamming mitigation in GPS. For this we need to simulate space-time array GPS data with high fidelity and the software GPS receiver. Therefore a focused research can be initiated which will mainly present a simulation method for space-time GPS data, as well as the simulation of some typical kinds of jammers. The wave-number domain algorithms developed in this thesis can then be applied to achieve the required interference free communication.

Bibliography

1. Bar-Cohen, Y., *Biomimetics : biologically inspired technologies*. 2006, Boca Raton, FL: CRC/Taylor & Francis. xviii, 527 p.32.
2. Agarwal, V., *Reviews and abstracts - Introduction to adaptive arrays.* Antennas and Propagation Society Newsletter, IEEE, 1981. **23**(4): p. 37-38.
3. Baldwin, P.J., *Book review: Introduction to Adaptive Arrays.* IEE Proceedings For Communications, Radar and Signal Processing, 1981. **128**(3): p. 188.
4. Krim, H. and M. Viberg, *Two decades of array signal processing research: the parametric approach.* Signal Processing Magazine, IEEE, 1996. **13**(4): p. 67-94.
5. Russell, M.E. *Future of RF Technology and Radars*. in *IEEE Radar Conference,* . 2007.
6. Gupta, I.J., J.A. Ulrey, and E.H. Newman. *Antenna element bandwidth and adaptive array performance*. in *IEEE Antennas and Propagation Society International Symposium,*. 2005.
7. Brookner, E. *Phased array radars-past, present and future*. in *RADAR 2002*. 2002.
8. Hussain, M.G.M. and M.A.R. Kourah. *Space-time array processing based on ultrawideband throb signal*. in *IEEE Radar Conference,* . 2010.
9. Alexiou, A. and M. Haardt, *Smart antenna technologies for future wireless systems: trends and challenges.* Communications Magazine, IEEE, 2004. **42**(9): p. 90-97.
10. Applebaum, S. and D. Chapman, *Adaptive arrays with main beam constraints.* IEEE Transactions on Antennas and Propagation, 1976. **24**(5): p. 650-662.
11. Brookner, E. *Phased-Array and Radar Breakthroughs*. in *IEEE Radar Conference,* . 2007.
12. Roberts, W., et al., *Iterative Adaptive Approaches to MIMO Radar Imaging.* IEEE Journal of Selected Topics in Signal Processing,, 2010. **4**(1): p. 5-20.
13. Shefeng, Y., M. Yuanliang, and Y. Kunde. *Optimal array pattern synthesis with desired magnitude response*. in *Proceedings of MTS/IEEE OCEANS,* . 2005.
14. Compton, R.T., Jr., *The relationship between tapped delay-line and FFT processing in adaptive arrays.* IEEE Transactions onAntennas and Propagation, 1988. **36**(1): p. 15-26.
15. Amari, S. and S.C. Douglas. *Why natural gradient?* in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,*. 1998.
16. Amari, S.-i., *Natural Gradient Works Efficiently in Learning.* Neural Computation, 1998. **10**(2): p. 251-276.
17. Douglas, S.C. and M. Gupta. *Scaled Natural Gradient Algorithms for Instantaneous and Convolutive Blind Source Separation*. in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP.* 2007.
18. Zhou, Y.-h., J.-h. Lin, and J.-x. Zhang. *A Variable Step-Size Optimization Algorithm Based on a Special Numerical Method to Solve the System of Nonlinear Equations*. in *International Workshop on Modelling, Simulation and Optimization, WMSO '08,* . 2008.
19. Shutao, Z. and I.L.-J. Thng, *Pre-steering derivative constraints for robust broadband antenna arrays*, in *Proceedings of the EEE International Conference on Acoustics, Speech, and Signal Processing - Volume 05*. 2001, IEEE Computer Society. p. 2913-2916.
20. Nickel, U., *Performance analysis of space-time-adaptive monopulse.* Signal Processing, 2004. **84**(9): p. 1561-1579.
21. *IEE Colloquium on `Antenna and Propagation Problems of Ultrawideband Radar' (Digest No.004)*. 1993.
22. Klouche-Djedid, A. and M. Fujita, *Adaptive array sensor processing applications for mobile telephone communications.* IEEE Transactions on Vehicular Technology,, 1996. **45**(3): p. 405-416.
23. Van Veen, B.D. and K.M. Buckley, *Beamforming: a versatile approach to spatial filtering.* ASSP Magazine, IEEE, 1988. **5**(2): p. 4-24.

24. Frost, O.L., III, *An algorithm for linearly constrained adaptive array processing.* Proceedings of the IEEE, 1972. **60**(8): p. 926-935.
25. Godara, L.C., *Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations.* Proceedings of the IEEE, 1997. **85**(8): p. 1195-1245.
26. Au, W., et al. *Simulation study of wideband interference rejection using STAP*. in *IEEE Aerospace Conference, .* 2006.
27. Ishide, A. and R. Compton, Jr., *On grating nulls in adaptive arrays.* IEEE Transactions on Antennas and Propagation,, 1980. **28**(4): p. 467-475.
28. Lin, N., W. Liu, and R.J. Langley, *Performance analysis of an adaptive broadband beamformer based on a two-element linear array with sensor delay-line processing.* Signal Process., 2010. **90**(1): p. 269-281.
29. Boon Poh, N. and D. Huiping, *Designing Amplitude/Phase Response of the Frost Beamformer.* IEEE Transactions on Signal Processing, , 2007. **55**(5): p. 1944-1949.
30. Liu, W. and S. Weiss, *Beam steering for wideband arrays.* Signal Processing, 2009. **89**(5): p. 941-945.
31. Materum, L.Y. and J.S. Marciano, Jr. *Wideband nulling capability estimate of a tapped delay line beamformer*. in *IEEE Topical Conference on Wireless Communication Technology,*. 2003.
32. Widrow, B., et al., *Adaptive antenna systems.* Proceedings of the IEEE, 1967. **55**(12): p. 2143-2159.
33. Valin, J.M. and I.B. Collings, *Interference-Normalized Least Mean Square Algorithm.* Signal Processing Letters,IEEE ,, 2007. **14**(12): p. 988-991.
34. Ko, C.C., *A simple, fast adaptive algorithm for broad-band null steering arrays.* IEEE Transactions on Antennas and Propagation,, 1991. **39**(1): p. 122-125.
35. Slock, D.T.M. and T. Kailath. *Numerically stable fast recursive least-squares transversal filters*. in *International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88., .* 1988.
36. Tylavsky, D.J. and G.R.L. Sohie, *Generalization of the matrix inversion lemma.* Proceedings of the IEEE, 1986. **74**(7): p. 1050-1052.
37. Hara, Y. *Weight convergence analysis of adaptive array antennas based on SMI algorithm*. in *International Zurich Seminar on Broadband Communications, Access, Transmission, Networking ,*. 2002.
38. Cantoni, A. and L.C. Godara, *Fast Algorithms for Time Domain Broadband Adaptive Array Processing.* IEEE Transactions on Aerospace and Electronic Systems, , 1982. **AES-18**(5): p. 682-699.
39. Godara, L.C., *Application of the fast Fourier transform to broadband beamforming.* The Journal of the Acoustical Society of America, 1995. **98**(1): p. 230-240.
40. Hudson, J.E., *Introduction to Adaptive Arrays.* Electronics and Power, 1981. **27**(6): p. 491.
41. Hudson, J.E., *Adaptive array principles*. 1981, Stevenage, UK ; New York: Peregrinus on behalf of the Institution of Electrical Engineers. xiv, 253 p.
42. Shah, I.U. and T.S. Durrani. *Broadband Constraint Natural Gradient based Adaptive processing*. in *IEEE/SP 15th Workshop on Statistical Signal Processing. SSP '09.* . 2009.
43. Shah, I.U. and T.S. Durrani. *Broadband Constraint Natural Gradient based Adaptive processing*. in *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*. 2009.
44. Mahoney, R.E. and R.C. Williamson. *Riemannian structure of some new gradient descent learning algorithms*. in *The IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium . AS-SPCC.* . 2000.
45. Candela, A.M. and M. Sánchez, *Geodesic connectedness in Gödel type space-times.* Differential Geometry and its Applications, 2000. **12**(2): p. 105-120.
46. Cheng, E., et al., *Distance formula and shortest paths for the (n,k)-star graphs.* Information Sciences, 2010. **180**(9): p. 1671-1680.

47. Amari, S. and S.C. Douglas. *Why natural gradient?* in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. 1998.
48. Alekseevsky, D.V., V. Cortés, and C. Devchand, *Special complex manifolds.* Journal of Geometry and Physics, 2002. **42**(1-2): p. 85-105.
49. *6 Covariant 2-Tensors and Metric Structures*, in *Pure and Applied Mathematics*, W.D. Curtis and F.R. Miller, Editors. 1985, Elsevier. p. 72-93.
50. Polyak, B.T., *Newton's method and its use in optimization.* European Journal of Operational Research, 2007. **181**(3): p. 1086-1096.
51. Lee, A., et al. *Simulation Study of Wideband Interference Rejection using Adaptive Array Antenna*. in *IEEE Aerospace Conference, .* 2005.
52. Nakamura, T., et al., *Development and performance evaluation of point to multipoint communication control scheme via satellite.* Performance Evaluation, 1991. **12**(3): p. 169-179.
53. Zhang, X., et al., *Blind multiuser detection for MC-CDMA with antenna array.* Computers & Electrical Engineering, 2010. **36**(1): p. 160-168.
54. Hansen, J.V., *Genetic search methods in air traffic control.* Computers & Operations Research, 2004. **31**(3): p. 445-459.
55. Xie, L., Y. Liu, and H. Yang, *Gradient based and least squares based iterative algorithms for matrix equations AXB + CXTD = F.* Applied Mathematics and Computation, 2010. **217**(5): p. 2191-2199.
56. Rodgers, W.E. and R.T. Compton, *Adaptive Array Bandwidth with Tapped Delay-Line Processing.* IEEE Transactions on Aerospace and Electronic Systems,, 1979. **AES-15**(1): p. 21-28.
57. Liu, W., et al., *Frequency invariant beamforming for two-dimensional and three-dimensional arrays.* Signal Processing, 2007. **87**(11): p. 2535-2543.
58. Yan, S., et al., *Convex optimization based time-domain broadband beamforming with sidelobe control.* The Journal of the Acoustical Society of America, 2007. **121**(1): p. 46-49.
59. Panayirci, E., Tug, and N. bay, *Optimum design of finite duration nyquist signals.* Signal Processing, 1984. **7**(1): p. 57-64.
60. Liu, W., *Adaptive wideband beamforming with sensor delay-lines.* Signal Processing, 2009. **89**(5): p. 876-882.
61. Twigg, C. and P. Hasler, *Configurable analog signal processing.* Digital Signal Processing, 2009. **19**(6): p. 904-922.
62. Crochiere, R.E. and L.R. Rabiner, *Multirate digital signal processing*. 1983, Englewood Cliffs, N.J.: Prentice-Hall. xx, 411 p.
63. Smolders, A.B. and G.W. Kant. *THousand Element Array (THEA)*. in *IEEE Antennas and Propagation Society International Symposium,*. 2000.
64. Hersey, R.K., et al. *Adaptive conformal array radar*. in *Proceedings of the IEEE Radar Conference,* . 2004.
65. Koutitas, G., *Multiple Human Effects in Body Area Networks.* Antennas and Wireless Propagation Letters, IEEE, 2010. **9**: p. 938-941.
66. Haykin, S., *Cognitive radio: brain-empowered wireless communications.* IEEE Journal on Selected Areas in Communications, , 2005. **23**(2): p. 201-220.