OPTIMIZATION OF THE ARRAY GEOMETRY FOR DIRECTION FINDING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SEVAL ÖZAYDIN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences

_____
Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____
Prof. Dr.Mübeccel DEMİREKLER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____	_____
Prof. Dr. Yalçın TANIK Co-Supervisor	Assoc. Prof. Dr. Sencer KOÇ Supervisor

Examining Committee Members

Prof. Dr. Rüyal ERGÜL _____

Prof. Dr. Yalçın TANIK _____

Assoc. Prof. Dr.Sencer KOÇ _____

Assist. Prof. Dr. Arzu TUNCAY KOÇ _____

Gülgün KADERLİ _____

ABSTRACT

OPTIMIZATION OF THE ARRAY GEOMETRY FOR DIRECTION FINDING

ÖZAYDIN, Seval

MSc. , Department of Electrical and Electronic Engineering

Supervisor: Assoc. Prof. Dr. Sencer KOÇ

Co-Supervisor: Prof. Dr. Yalçın TANIK

December 2003, 86 pages

In this thesis, optimization of the geometry of non-uniform arrays for direction finding yielding unambiguous results is studied. A measure of similarity between the array response vectors is defined. In this measure, the effects of antenna array geometry, source placements and antenna gains are included as variable parameters. Then, assuming that the antenna gains are known and constant, constraints on the similarity function are developed and described to result in unambiguous configurations and maximum resolution. The problem stated is solved with two different methods, the MATLAB optimization toolbox, and genetic algorithm in which different genetic codings are also studied.

The performance of the MUSIC algorithm with the optimized array geometries

are investigated through computer simulations. The direction of arrival estimates are obtained using the optimized array geometry on the MUSIC algorithm along with the effects of different parameters. Statistics of the true and probable erroneous arrival angles and the probability of gross error are obtained as a measure of performance. It is observed that the proposed optimization process for the array geometry gave rise to unambiguous results for direction finding.

Keywords : direction finding, MUSIC algorithm, array optimization, genetic algorithm

ÖZ

YÖN BULMADA ANTEN DİZİLERİNİN GEOMETRİ OPTİMİZASYONU

ÖZAYDIN, Seval

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Sencer KOÇ

Ortak Tez Yöneticisi: Prof. Dr. Yalçın TANIK

Aralık 2003, 86 sayfa

Bu araştırmada, eş aralıklı olmayan anten dizilerinin, yön bulmada belirsizlik içermeyen sonuçlar verebilmesi için, geometri optimizasyonları üzerine çalışılmıştır. Anten dizisi tepki vektörlerinin arasında bir benzerlik ölçüsü tanımlanmıştır. Bu ölçüye, anten dizisi geometrisi, kaynak yerleşimleri ve anten kazançları değişken parametreler olarak dahil edilmiştir. Daha sonra, anten kazançlarının bilindiği ve sabit olduğu varsayılarak, yön bulmada en yüksek çözünürlüğe sahip ve belirsizlik içermeyen sonuçlar elde etmek üzere, benzerlik fonksiyonu üzerine getirilecek kısıtlamalar belirlenmiş ve anlatılmıştır. Tanımlanan problem, MATLAB'ın optimizasyon işlevi ve farklı genetik kodlamaların da kullanıldığı genetik algoritma yöntemleriyle çözülmüştür.

MUSIC algoritmasının optimum anten geometrileriyle performansı bilgisayar simülasyonlarıyla gözlemlenmiştir. Geliş açısı tahminleri, farklı parametrelerin etkisi de göz önünde bulundurularak, optimum anten geometrilerinin MUSIC algoritmasında kullanılmasıyla elde edilmiştir. Performans ölçüsü olarak, gerçek ve muhtemel yanlış geliş açılarının istatistikleri ve karışıklık oluşması ihtimali hesaplanmıştır. Anten dizisi geometrilerinin optimizasyonu için öngörülen yaklaşımın, yön bulmada belirsizlik içermeyen sonuçlar verdiği gözlemlenmiştir.

Anahtar Kelimeler: yön bulma, anten dizisi optimizasyonu, MUSIC algoritması, genetik algoritma

ACKNOWLEDGEMENTS

I appreciate to Assoc. Prof. Dr. Sencer Koç and Prof. Dr. Yalçın Tanık for their valuable supervision during the development and the improvement stages of this thesis.

I also wish to thank to ASELSAN Inc. for the facilities provided for the completion of this thesis.

Thanks a lot to my parents, Ülker and Hadi Özaydın; my brother, Salim Özaydın; and my friends, Levent Atılgan, Selva Muratoğlu, Yelda Altuğ, Dilara Güler, Güzin Kurnaz, for their trust, great encouragement and continuous morale support.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

DF	Direction Finding
DOA	Direction of Arrival
MUSIC	Multiple Signal Classification
SNR	Signal-to-Noise Ratio
UCA	Uniform Circular Array
ULA	Uniform Linear Array
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
MIN-NORM	Minimum-Norm

CHAPTER 1

INTRODUCTION

1.1. Direction Finding

Estimation problems in theoretical as well as applied statistics have long been of great research interest given their importance in a great variety of applications. Parameter estimation has particularly been an area of focus by applied statisticians and engineers as problems required ever improving performance. Many techniques were the result of an attempt by researchers to go beyond the classical Fourier-limit.

As applications extended, the interest in accurately estimating relevant temporal as well as spatial parameters grew. Sensor array signal processing emerged as an active area of research and was centered on the ability to fuse data collected at an array of judiciously placed antenna sensors in space in the field of interest. The received signal is assumed to be generated by a finite number of emitters, and contains information about signal parameters characterizing the emitters. The goal is to extract useful characteristics of the received signal (e.g., the number of sources, their direction in azimuth and elevation, complex envelopes, location, signature, speed of propagation, center frequency, range, etc., associated with each signal). One of the most important parameters to be extracted from the

sensor array is the direction of the signals impinging on the array, the problem named as '*Direction Finding'* (DF), [1].

The sources under consideration, may be uncorrelated (i.e., independent from each other), correlated (i.e., dependent by some amount designated by the correlation coefficient), or coherently related (i.e., correlation coefficient is 1, these sources are identical) to each other.

There are some important factors to be considered in the DF problem such as the choice of the sensors, the receiver structure, and the optimum algorithm. One other important factor to be considered in the DF problem is the choice of the optimum array geometry, which is the focus of this thesis.

1.2. Array Geometry

In DF systems, a variety of different array geometries are used depending on the application of interest. The array geometry affects various aspects of the DF system such as resolution and sensitivity to system errors.

The most commonly used configuration is the linear array, in which the sensors (all of a common type) are spaced along a straight line. If sensor spacings are uniform, this is called a uniform linear array (ULA). Another common configuration is a planar array, in which the sensors may form a rectangular grid or lie on concentric circles. If the sensors are placed on a single circle with uniform spacings, this is called a uniform circular array (UCA), [2].

The widely studied uniform linear array (ULA) can provide estimates of source bearings relative to the array axis. However, a planar array is required if estimates

of both source azimuth and elevation are required. The following properties of uniform circular arrays (UCA's) make them attractive in the context of DOA estimation. UCA's provide 360° azimuthal coverage and also provide information on source elevation angles. ULA's, in contrast, provide only 180° coverage and have left-right ambiguity. They do not provide uniform resolution over the entire space.

Conventionally, the uniform linear array is implemented with inter-element spacing less than or equal to $\lambda/2$, where λ is the wavelength at the center frequency. The value of $\lambda/2$ is derived from the sampling theorem in order to cover angles from $-\pi/2$ to $\pi/2$ in physical space, measured from the boresight of the array, and to eliminate the problem of ambiguity in angles of arrival.

In some cases, to increase the accuracy of measurement, one has to maximize the aperture or the length of the array for a given number of sensors, which will increase the inter-sensor spacing such that the average spacing will be larger than $\lambda/2$. This will introduce ambiguities if ULA is employed. To avoid this problem, array structures in which the element spacing is unequal, called non-uniform array, are used.

1.3. Direction Finding Algorithms

There exist a vast number of algorithms in the literature for estimating the direction of arrival (DOA) from the measured output of a sensor array. These algorithms can be classified into two main categories, namely *spectral-based* and *parametric* approaches. In the former, one forms some spectrum-like function of the

parameter(s) of interest, e.g., the DOA. The locations of the highest (separated) peaks of the function in question are recorded as the DOA estimates. Parametric techniques, on the other hand, require a simultaneous search for all parameters of interest. The latter approach often results in more accurate estimates, at the expense of an increased computational complexity.

1.3.1. Spectral-Based Methods

1.3.1.1. Beamforming Techniques [1]

The first attempt to automatically localize signal sources using antenna arrays was through beamforming techniques. The idea is to 'steer' the array pattern in one direction at a time and measure the output power. The steering locations which result in maximum power yield the DOA estimates.

1.3.1.2. Subspace-Based Methods [2]

The subspace based methods exploit the underlying structure of the array covariance matrix:

$$\underline{\underline{R}} = E\left\{\underline{y}(t_i)\underline{y}^H(t_i)\right\} = \underline{\underline{A}}(\underline{\Theta})\underline{\underline{R}}_x\underline{\underline{A}}^H(\underline{\Theta}) + \sigma^2\underline{\underline{I}}, \tag{1.1}$$

where

$\underline{\underline{A}}(\underline{\Theta}) = \left[\underline{a}(\underline{\Theta}_1) \quad . \quad . \quad . \quad \underline{a}(\underline{\Theta}_L)\right]$: array manifold matrix,

$\underline{a}(\underline{\Theta})$: array steering vector,

$\underline{\underline{R}}_x = E\left\{\underline{x}(t_i)\underline{x}^H(t_i)\right\}$: signal covariance matrix,

σ^2 : power of the additive white Gaussian noise.

$\underline{x}(t_i)$: complex envelope of the emitter signal at the i^{th} time instant

$\underline{y}(t_i)$: complex envelope of the sensor output at the i^{th} time instant

These methods are based on the fact that the signal part of the array output vectors lies in the so-called signal subspace, which is a lower dimensional subspace of the array manifold.

The subspace based methods use either the signal subspace or the noise subspace for the estimation of the signal parameters. The issue is then the estimation of either signal or the noise subspace. The estimates of these subspaces are commonly found either from the eigendecomposition of the sample correlation matrix, or equivalently, from the singular value decomposition of the data matrix itself, although there are other approaches which do not use eigendecomposition techniques in an attempt to reduce the computational complexity at the cost of performance, [2].

The signal and noise subspaces can be consistently estimated from the eigendecomposition of a consistent estimate of the array covariance matrix, where the most popular estimate for $\underline{R}$ is

$$\hat{\underline{R}} = \frac{1}{N}\sum_i \underline{y}(t_i)\underline{y}^H(t_i) \tag{1.2}$$

with N as the number of samples.

Specific Algorithms and Discussions

The high resolution capability and lower computational complexity (as compared to parametric approaches) of subspace based methods have led to a vast number of algorithms based on the same approach.

MUSIC [1]

In the engineering literature, Pisarenko's work in harmonic retrieval was among the first to be published for subspace based methods. However, the tremendous interest in the subspace approach is mainly due to the introduction of the MUSIC (MUltiple SIgnal Classification) algorithm. It is interesting to note that while earlier works were mostly derived in the context of time series analysis and later applied to the sensor array problem, MUSIC was indeed originally presented as a DOA estimator.

In this algorithm, DOA estimates are obtained using the fact that any vector in the noise subspace is in the null space of the signal subspace, i.e., the signal and noise subspaces are orthogonal.

MUSIC algorithm will be explained in detail in Chapter 4.

MIN-NORM [1]

This algorithm is a result of an attempt to improve/overcome some of MUSIC's shortcomings in various specific scenarios. The method is the unifying theme of *weighted MUSIC* in which the denominator of the MUSIC spectrum is weighted by a matrix W. For different weighting matrices W, the method is particularized to various algorithms.

The weighting matrix W is introduced to take into account the influence of each of the eigenvectors. It is clear that a uniform weighting of the eigenvectors, i.e., W= I, results in the original MUSIC method. This is indeed the optimal weighting in terms of yielding estimates of minimal asymptotic variance. However, in difficult scenarios involving small number of samples, low SNR and highly correlated signals, a carefully chosen non-uniform weighting may improve the resolution capability of the estimator without seriously increasing the variance.

One particularly useful choice of weighting is given by

$W=e_1\ e_1^T$

where e_1 is the first column of the LxL identity matrix, L representing the number of sensors. This corresponds to the Min-Norm algorithm, which is originally proposed for uniform linear arrays but also extended to arbitrary array geometries.

The Min-Norm algorithm exhibits a lower bias and hence a better resolution than the original MUSIC algorithm, at least when applied to ULA's.

ESPRIT

Among the algorithms developed, ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) has been greatly exploited because of its computational efficiency. The reduction in the computational cost is achieved by requiring that the sensor array possess displacement invariance, i.e., sensors occur in matched pairs with identical displacement vectors. Thus, there exist two identical subarrays, one of which is shifted by a known amount Δ relative to the other. This method, different from MUSIC, does not require the array manifold, i.e., the actual

positions, patterns; etc., of the array elements; only the shift vector needs to be known, [2].

State-Space Realization (TAM)

The Toeplitz Approximation Method (TAM), is based on a state-space representation of the sensor outputs. TAM estimates an LxL state transition matrix, where L is the number of sensors, and obtains estimates of the DOA's from the eigenvalues of this matrix, [2].

1.3.2. Parametric Methods [1]

While the spectral-based methods presented in the previous sections are computationally attractive, they do not always yield sufficient accuracy. In particular, for scenarios involving highly correlated (or even coherent) signals, the performance of spectral-based methods may be insufficient. An alternative is to more fully exploit the underlying data model, leading the so-called *parametric* array processing methods. In these methods, the problem is the maximization of the *likelihood function* of the observed data. The set of parameters that maximizes the likelihood function is the set of parameters that makes the observed data most probable. Depending on the model assumption on the signal waveforms there exist two of these algorithms. They are presented shortly below:

Deterministic Maximum Likelihood

In this method, noise is modeled as a stationary Gaussian white random process whereas the signal waveforms are deterministic (arbitrary) and unknown.

The deterministic signal model is more appropriate in certain applications, such as radar and radio communication.

Stochastic Maximum Likelihood

This model is obtained by modeling the signal waveforms as stationary, Gaussian random processes. This method is reasonable, for instance, if the measurements are obtained by filtering wideband signals using a narrow bandpass filter.

1.4. Earlier Approaches Towards Array Geometry Optimization

The optimal array geometry problem is discussed in some papers, both in relation with beamforming and the DF problem. The most common structures are the linear and circular arrays.

Moffet studied on non-uniform linear arrays and found that there exists a class of non-uniform linear arrays, called the minimum redundancy array. The minimum redundancy array is a linear array that contains the least possible number of sensors such that all the required sensor spacings are present, hence the redundancy in the structure is minimized, [2].

The effect of the linear array geometry on DOA estimation for a single source is studied in [3]. A comparison on the performances of uniform and non-uniform linear array structures is made and in the case of single targets in additive white Gaussian noise, non-uniform arrays are found to provide significant improvement over the uniform arrays of the same number of elements.

The linear array design problem for optimum DOA performance with regard to the maximum likelihood estimation is studied in [4]. The non-uniform array structure proposed in this study has an improved performance as compared to uniform linear arrays and the minimum redundancy arrays.

The design of super-resolution direction finding arrays that satisfy prespecified performance levels, such as detection-resolution thresholds and Cramer-Rao bounds on error variance is also a different problem and has been addressed by Dowlut and Manikas, [5]. The sensor placement problem is formulated in the framework of subspace-based DF techniques. The core of the design approach is the sensor locator polynomial (SLP), which is constructed using the manifold curvatures and whose roots yield the normalized sensor locations of the desired array.

The problem of DOA ambiguity of arbitrary array for subspace-based DF methods in a wide frequency band is studied in [6]. The ambiguities are divided into different orders and first-order ambiguity is intensively discussed. Especially, the ULA and UCA are examined and a general conclusion on first-order ambiguity versus array configuration is reached.

One of the most specific works on optimizing the array geometry for ambiguity resolution in direction finding is that of Gavish and Weiss, [7]. They introduced a measure of similarity between array response vectors and derived a tight lower bound of the similarity measure. The array geometry associated with the highest lower bound performs better than other arrays with the same aperture and the same number of sensors. Therefore, this lower bound is used for selecting the

best thinned array configuration from a set of candidate geometries by computing the bound for each configuration in the set. It is shown that for wideband arrays, the optimal array selection should be performed only once at the highest frequency of operation. The approach proposed in this study is applicable to any array configuration.

1.5. The Motivation and Purpose of This Work

Although sensor-array applications such as direction finding, interference cancellation, communications, etc., have been worked on a lot, there is not much literature on the design of the array geometry. In this thesis, the problem of optimizing the array geometry to minimize probability of large errors in direction of arrival (DOA) estimation is studied.

The array manifold gives the array response in the presence of a single signal. For different direction of arrivals (DOA's), the manifold should be different. If, for a given set of widely separated DOA's the array response is similar, large errors, usually referred to as ambiguity errors, are likely to occur. A measure of similarity between the array response vectors is introduced for the solution of this problem. Appropriate array geometries, yielding unambiguous results, are found by putting some constraints on the evaluated similarity function.

The particular problem considered in this work is the estimation of DOA's, in azimuth, of L incident plane waves, at a known wavelength λ, by using N data samples taken from a non-uniform linear array with M antennas whose locations are determined by the constrained optimization process implemented on the evaluated similarity function, using genetic algorithm. The MUSIC algorithm is applied to

the resultant array geometry. A direction estimation covering (0°) – (180°) in azimuth is performed since a linear array is used.

The performance of the proposed optimization process is then investigated through computer simulations, for which the variable parameters are chosen carefully for verifying the array geometry in any case. The probability of ambiguity is found for the optimized geometries. It is observed that the proposed optimization process resulted in unambiguous array geometries.

1.6. Outline of the Thesis

This thesis is organized as follows: In Chap. 2, the problem is stated and formulated; an approach on optimizing the array geometry in DF, by Motti Gavish and Anthony J. Weiss, is presented, and the details for the proposed constraints on the similarity function are developed and described.

In Chap. 3, the optimization methods are told. A brief summary on Genetic Algorithm is provided. The details of the array optimization, using the constraints developed in Chap.2, are given. Also, the simulation results for the optimum array geometry, depending on the constraints on the similarity function, are presented.

In Chap. 4, typical results of the computer simulations, using the optimized array geometry on the MUSIC algorithm, are presented, along with the discussions related to effects of different parameters. To investigate the performance of the proposed geometries, the statistics of the true and probable ambiguous peaks are given. The probability of ambiguity is derived for the performance measure.

Finally, Chap.5 includes some concluding remarks.

CHAPTER 2

THE OPTIMIZATION APPROACH

In this study, the problem of optimizing the antenna array geometry for unambiguous estimation of the direction of arrivals (DOA's) of plane waves is investigated. Aim of this study is to find the optimum array geometry, for the estimation of the DOA's of L narrowband plane waves from the measurements taken by an array consisting of M sensors ($\underline{y}_{Mx1}(t_i)$ for $i = 0,...,N-1$).

The problem is formulated in terms of the steering vector concept described later, and a measure of similarity of the steering vectors is developed. In the first part of this chapter, the study by Motti Gavish and Anthony J. Weiss [7], is given. In their study, a measure of similarity between the array response vectors is derived and lower bounded by a discrete function that can be easily computed. According to the proposed design criterion, the best array geometry is found using this lower bound.

In the second part, the optimization criteria for the array geometry used in this thesis are described. The same measure of similarity between the array response vectors is used as in the work by Motti Gavish and Anthony J. Weiss, [7]. But, the proposed optimization criteria are novel. Detailed derivations for the optimization criteria are also provided.

2.1. Data Model

The data model used in this chapter assumes M sensors located in an arbitrary geometry and L plane waves impinging on the array. When each sensor output is modeled as the response of a linear time-invariant system, the output of the k^{th} sensor can be written as a superposition

$$\tilde{y}_k(t) = \sum_{i=1}^{L} h_{ki}(t) * \tilde{x}_i(t - \tau_{ki}) + \tilde{n}_k(t), \tag{2.1}$$

where $h_{ki}(t)$ denotes the impulse response of the k^{th} sensor to the i^{th} signal $\tilde{x}_i(t)$ impinging on the array, τ_{ki} denotes the time delay of the i^{th} signal at the k^{th} sensor, relative to some fixed reference point, (*) denotes convolution, and $\tilde{n}_k(t)$ is an additive noise term. Denoting the DOA of the i^{th} signal as θ_i, and under the narrowband assumption, i.e., time delay between any two elements of the array is small compared to the time variations and phase modulations of the carrier frequency, the *complex envelope* of the sensor output can be written as

$$y_k(t) = \sum_{i=1}^{L} H_k(\theta_i) e^{-j\omega\tau_k(\theta_i)} x_i(t) + n_k(t) \ , \tag{2.2}$$

where the k^{th} sensor's response and the time-delay of propagation for the i^{th} signal are denoted by $H_k(\theta_i)$ and $\tau_k(\theta_i)$, respectively. $x_i(t)$ and $n_k(t)$ are the complex envelopes of the i^{th} emitter signal and the k^{th} sensor noise, respectively. Adopting a vector notation, the above equation can be written in the following form:

$$\begin{aligned} \boldsymbol{y}(t) &= \left[\boldsymbol{a}(\theta_1) \ldots \boldsymbol{a}(\theta_L)\right] \left[x_1(t) \ldots x_L(t)\right]^T + \boldsymbol{n}(t), \\ &= \mathbf{A}(\boldsymbol{\theta})\, \mathbf{x}(t) + \boldsymbol{n}(t), \end{aligned} \tag{2.3}$$

where the response of the k^{th} sensor to the i^{th} signal is $a_k(\theta_i) = H_k(\theta_i)e^{-j\omega\tau_k(\theta_i)}$.

The vector $\boldsymbol{y}$ (t) belongs to an *M*-dimensional complex vector space.

The *array response vector* $\mathbf{a}(\theta_i)$ is an element of the complex *M*-dimensional vector space, and the *array manifold* is defined as the collection of all array response vectors over the parameter range of interest, which for the special case studied in this thesis, is only the signal's DOA as measured relative to a reference axis.

2.2. An Approach for Optimization of Array Geometry by Motti Gavish and Anthony J. Weiss [7]

The array manifold gives the array response in the presence of a single signal, as explained before. For different direction of arrivals (DOA's), the array response vectors should be different. If, for widely separated DOA's, the array responses are close to each other, this will yield ambiguity errors. In [7], a measure of similarity between array response vectors is introduced. A tight lower bound for the similarity measure is derived. The array geometry which has the highest lower bound has a smaller probability of making a gross error than the other arrays with the same aperture and the same number of antennas. Therefore, this bound can be used to select the best array geometry from a set of given geometries, by computing the bound for each geometry in this selected set of geometries.

The notation used in the following sections is as follows:

$(.)^T$: Transpose

$(.)^*$: Complex conjugate

$(.)^H$: Hermitian (complex conjugate) transpose

$E\{.\}$: Expected value

$\| . \|$: Euclidean norm

I_M : MxM identity matrix

tr{X}: Trace of matrix X

2.2.1. Linear Arrays

Consider a linear array of M sensors. The sensor gains are assumed to be constant. The array steering vector (the array response vector) is given by

$$\mathbf{a}(\theta) = \left[g_1, g_2 e^{j2\pi d_2 \sin\theta}, g_3 e^{j2\pi d_3 \sin\theta}, ..., g_M e^{j2\pi d_M \sin\theta}\right]^T \tag{2.4}$$

where d_m is the mth sensor distance from the first sensor (in wavelength units), θ is the off-broadside signal angle, and g_m is the gain of the mth sensor.

Obviously, the steering vectors of two closely spaced angles are not much different, and therefore, the array spatial resolution is limited. In this work, the concentration is on the similarity of steering vectors associated with widely separated angles, which are the probable candidates for ambiguity errors.

It is assumed that the array is intended to intercept signals from a given field of view defined by $\Theta = [\theta_{min}, \theta_{max}]$. It is desired that steering vectors associated with $\theta_1 \in \Theta$ and $\theta_2 \in \Theta$ where $\theta_1 \neq \theta_2$ would be as different as possible. Each steering vector is characterized by the vector of phases of the last M-1 elements relative to the first element

$$\mathbf{p}(\theta)=\left[2\pi d_2 \sin\theta, 2\pi d_3 \sin\theta, \ldots, 2\pi d_M \sin\theta\right]^T \tag{2.5}$$

Corresponding to the DOA's θ_1 and θ_2, a very good measure of the similarity or difference between the two steering vectors, is the squared Euclidean norm

$$Q(\theta_1, \theta_2) \triangleq \frac{1}{4\pi^2} \| \mathbf{G}((\mathbf{p}(\theta_2) - \mathbf{p}(\theta_1)))_{2\pi} \|^2 \tag{2.6}$$

where $((\mathbf{z}))_{2\pi}$ is used for representing the evaluation of each of the elements of the vector $\mathbf{z}$ modulo 2π, and the modulo is defined in the interval $(-\pi, \pi]$. The coefficient $1/4\pi^2$ is used to reduce the computational complexity for the mathematical operations. The matrix $\mathbf{G}$ in Eq. (2.6) is a diagonal weighting matrix defined by

$$\mathbf{G} \triangleq 1/g_1 \, \mathrm{diag}\{g_2, g_3, \ldots, g_M\}. \tag{2.7}$$

In (2.7) g_1 must be nonzero since the case $g_1 \equiv 0$ is meaningless.

Using the notation $x \triangleq \sin\theta_2 - \sin\theta_1$, and $\mathbf{d} \triangleq [d_2, d_3, \ldots, d_M]^T$, (2.6) becomes

$$Q(x) = \frac{1}{4\pi^2} \left\| \mathbf{G}((2\pi \mathbf{d} x))_{2\pi} \right\|^2 \tag{2.8}$$

In Eq. (2.6), the introduced measure of similarity seems to be intuitively satisfying, since it is closely related to the probability of error in choosing between θ_1 and θ_2 using a finite number of noisy samples of the array output. This approach is based on the classical detection theory. During the probability of error calculations, it is assumed that the signal and noise are mutually uncorrelated zero-mean Gaussian processes with known covariance.

Equation (2.8) can be written as

$$Q(x) = \| \ \mathbf{G}(\mathbf{d}x\text{-}\mathbf{k}) \ \|^2 \tag{2.9}$$

where $\mathbf{k} \underline{\underline{\Delta}} \ [k_2, k_3, \ldots, k_M]^T$ is a vector of integers selected so that each element of the vector ($\mathbf{d}x$-$\mathbf{k}$) is in the interval (-1/2,1/2].

As a result of its definition, x is limited to the interval $\mathbf{X}= [x_{min}, x_{max}]$ which is induced by Θ. According to the value of **k,** the interval **X** can be divided into subintervals. Within each subinterval, the value of **k** is fixed, and is different for different subintervals.

The identification of the integer vectors **k** associated with a given array geometry is an important step in the evaluation of the array score. Starting with linear arrays, and recalling that the vector $\mathbf{k} \underline{\underline{\Delta}} \ [k_2, k_3, \ldots, k_M]^T$ is a vector of integers selected to limit the elements of the vector ($\mathbf{d}x$-$\mathbf{k}$) to the interval (-1/2,1/2], the valid interval for x can be found

$$-1/2 < d_i x - k_i \leq 1/2 \qquad\qquad i = 2,3,\ldots,M$$

or

$$\frac{-1/2 + k_i}{d_i} < x \leq \frac{1/2 + k_i}{d_i} \qquad i = 2,3,\ldots,M \tag{2.10}$$

The above equation identifies intervals of x in which a given k_i value is used. Since $k_i = 0,1,2,\ldots, k_i^{max}$, it is easy to identify all the valid intervals of x in which k_i is fixed. The combination of the intervals for $i = 2,3,\ldots,M$ can be used to

identify the set of vectors **k** associated with a given array. The vector **k** is fixed within each interval and is different in different intervals.

For three-dimensional arrays composed of orthogonal linear subarrays, this approach can still be used. For an arbitrary array geometry, the vectors **k** can be found by using a dense enough grid of azimuth and elevation angles within the spatial sector of interest.

After the computation of integers **k,** the minimization of Q(x) with respect to x for a given vector **k** is required. Using the first derivative of Q(x), the minimizer is found as

$$\tilde{x} = (\mathbf{d}^T\mathbf{G}^2\mathbf{d})^{-1}\mathbf{d}^T\mathbf{G}^2\mathbf{k} \tag{2.11}$$

substituting $\tilde{x}$ back to Eq. (2.9) gives the lower bound

$$Q(x) \geq \| \mathbf{Dk} \|^2 \tag{2.12}$$

where **D** is the matrix

$$\mathbf{D} \triangleq [\mathbf{Gd}(\mathbf{d}^T\mathbf{G}^2\mathbf{d})^{-1}\mathbf{d}^T\mathbf{G}\text{-}\mathbf{I}_{M-1}]\mathbf{G} \tag{2.13}$$

For any array configuration, the function $Q(x)$ and its lower bound can be evaluated. As expected, for x=0 the distance between the steering vectors is zero, and the function $Q(x)$ takes the value zero. This is also the same for the case x=2 where the steering vectors are equal for any array whose sensor spacings are multiples of λ/2, where λ is the wavelength of the incoming signal, and reflects the difficulty of such arrays to distinguish between directions (close to) 90° and -90°!

The lower bound found for $Q(x)$ is tight and $Q(x)$ reaches the bound at certain points.

To evaluate the array, it is useful to calculate the bound $\|\mathbf{Dk}\|^2$ for all the vectors **k** except for the vector associated with the first interval which is a vector of zeros. The number of vectors **k** is finite and is a function of the array aperture and the number of sensors. For the values of x associated with the minima of $Q(x)$, $Q(x)$ is equal to the bound. Because of this fact, the examination of the bound reveals the combinations of DOA's corresponding to the poorest array performance.

The set of all the **k** vectors induced by the vector **d** and the interval **X** is denoted by **K ,** not including **k**=0. Then, a scalar S is defined as

$$S = \min_{\mathbf{k}\in\mathbf{K}} \|\mathbf{Dk}\|^2 \tag{2.14}$$

Here, S characterizes the worst-case ambiguity behaviour of the array and can be used as a performance index of a given array geometry. Better arrays will be associated with larger S, using a worst-case approach. From all the candidate geometries, the one with the highest score S will be chosen. This strategy yields to the selection of array having the best worst-case performance among the considered configurations.

The evaluation of S requires finding the smallest number out of a finite set of numbers.

Some remarks about the criterion are as follows: First, in Eq. (2.6), the phase vector $\mathbf{p}(\theta)$ which is linear in $\sin\theta$ is used, and it is normalized by the relative gains of the matrix **G** rather than using directly the steering vector $\mathbf{a}(\theta)$. This

selection was done to have a similarity function $Q(x)$ which can be easily bounded, to simplify the approach. Second, the gains of the antennas, g_m should be known and are assumed constant (but not necessarily identical) within the angular sector of interest. Small gain changes are not likely to affect the score function significantly.

A more generalized form of the approach is presented below, derived for the case of most practical arrays required to operate over a wide-frequency band.

The elements of the vector **d** are the spacings between the first sensor and the *m*th sensor in wavelength units. So, the vector **d** is different for each frequency. Letting $\mathbf{d}_l$ to be the vector **d** corresponding to a frequency associated with wavelength λ_l , and $\mathbf{d}_h$ to be the vector **d** evaluated at the highest frequency of operation, associated with wavelength λ_h, there exists the relation

$$\mathbf{d}_l = \mathbf{d}_h \frac{\lambda_h}{\lambda_l} \tag{2.15}$$

Since the vector $\mathbf{d}_l$ is a scaled version of the vector $\mathbf{d}_h$, the matrix **D**, defined in (2.13), is the same for all frequencies. Also, the set of integer vectors $\mathbf{K}_l$ associated with $\mathbf{d}_l$ is a subset of the set $\mathbf{K}_h$ associated with $\mathbf{d}_h$.

Hence, the score S_h , associated with the highest frequency is always less than or equal to the score S_l associated with any other frequency

$$S_h \leq S_l \ . \tag{2.16}$$

This result shows that the higher the frequency the higher is the probability of ambiguity. If a worst-case design criterion is used, the geometry selection should be performed only once at the highest frequency of operation. This approach yields to

the selection of the array having the best worst-case performance (over all frequency range and all directions of interest) among the considered configurations.

2.2.2. General Arrays

For a general array of sensors, in which the sensor gains are assumed constant within the spatial sector of interest, the array steering vector is given by

$$\mathbf{a}(\mathbf{u}) = \left[g_1, g_2 e^{j2\pi \mathbf{r}_2^T \mathbf{u}}, g_3 e^{j2\pi \mathbf{r}_3^T \mathbf{u}}, \ldots, g_M e^{j2\pi \mathbf{r}_M^T \mathbf{u}} \right]^T \tag{2.17}$$

where

$$\mathbf{r}_m \triangleq [\, x_m, y_m, z_m \,]^T \qquad m=2,3,\ldots,M \tag{2.18}$$

$$\mathbf{u} \triangleq [\cos\theta \cos\phi\,,\ \sin\theta \cos\phi\,,\ \sin\phi\,]^T \tag{2.19}$$

In Eq. (2.18), x_m , y_m , and z_m are the cartesian coordinates of the mth sensor, and $\mathbf{u}$ defined in (2.19) is a unit vector pointing to the source. The angles θ , ϕ stand for azimuth and elevation, respectively. The first sensor coincides with the origin of the coordinate system.

In this case, the vector of phases is given by

$$\mathbf{p}(\mathbf{u})=2\pi\mathbf{R}\mathbf{u} \tag{2.20}$$

where $\mathbf{R} \triangleq [\mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_M]^T$.

In this case, the measure of similarity is

$$Q(\mathbf{u_1}, \mathbf{u_2}) \triangleq 1/4\pi^2 \,\| \mathbf{G}((\mathbf{p}(\mathbf{u_2}) - \mathbf{p}(\mathbf{u_1})))_{2\pi} \|^2 \tag{2.21}$$

for which, the matrix **G** is as defined in (2.7).

If, (2.20) is substituted in (2.21), the similarity measure becomes

$$Q(\bar{x}) = \| \ \mathbf{G}(\mathbf{R}\bar{x} - \mathbf{k}) \ \|^2 \tag{2.22}$$

where $\bar{x} \triangleq \mathbf{u}_2 - \mathbf{u}_1$ (2.23)

and $\mathbf{k}$ is a vector of integers, selected so that each element of the vector $(\mathbf{R}\bar{x} - \mathbf{k})$ is in the interval (-1/2,1/2].

The function $Q(\bar{x})$ can be bounded, using the same technique as for linear arrays as follows:

$$Q(\bar{x}) \geq \| \ \mathbf{Tk} \ \|^2 \tag{2.24}$$

where

$$\mathbf{T} \triangleq [\mathbf{GR}(\mathbf{R}^T\mathbf{G}^2\mathbf{R})^{-1}\mathbf{R}^T\mathbf{G} - \mathbf{I}_{M-1}]\mathbf{G}. \tag{2.25}$$

The corresponding score for a given array geometry becomes

$$S = \min_{k \in K} \| \ \mathbf{Tk} \ \|^2 \tag{2.26}$$

where the set of integers **K** is obtained by removing the vector **k**=0 from the set of all the **k** vectors induced by the matrix **R** and the corresponding interval **X** of $\bar{x}$.

So, the technique that is used to evaluate a linear array can be used for arrays with a general geometry.

2.2.3. Conclusion

In the study by M. Gavish and A. J. Weiss [7], that is presented above, a measure of similarity is introduced and lower bounded by a tight bound which

serves as a score function for optimization of array configurations. The proposed design procedure can be used with any array geometry.

2.3. The Proposed Optimization Approach for Unambiguous Array Design

As described in the previous section, the array manifold gives the array response in the presence of a single signal.

In this study, a linear array of M sensors is considered. The sensor gains are assumed constant. The the array response vector is as given in Eq. (2.4), i.e.,

$$\mathbf{a}(\theta) = \left[g_1, g_2 e^{j2\pi d_2 \sin\theta}, g_3 e^{j2\pi d_3 \sin\theta}, ..., g_M e^{j2\pi d_M \sin\theta}\right]^T \tag{2.27}$$

where d_i is the ith sensor distance from the first sensor (in wavelength units), θ is the signal angle measured from the broadside direction, and g_i is the gain of the ith sensor.

Throughout this section, lowercase boldface letters are used to denote vectors and uppercase boldface letters are used to denote matrices.

Letting $\mathbf{a}(\theta)$ to be the array response vector, define

$$Q(\theta_1, \theta_2) \underline{\Delta} 1/4 \parallel \mathbf{a}(\theta_1) - \mathbf{a}(\theta_2) \parallel^2 \tag{2.28}$$

as the similarity function. This function is a measure of how similar the array steering vectors for different angles θ_1 and θ_2.

Substituting (2.27) in (2.28), the following are obtained:

$$Q(\theta_1, \theta_2) = 1/4 \parallel \mathbf{a}(\theta_1) - \mathbf{a}(\theta_2) \parallel^2$$

$$= \frac{1}{4}\sum_{i=2}^{M} g_i^2 \left| \left[e^{j2\pi d_i \sin\theta_1} - e^{j2\pi d_i \sin\theta_2} \right] \right|^2$$

$$= \frac{1}{4}\sum_{i=2}^{M} g_i^2 \left| e^{j2\pi d_i \frac{\sin\theta_1+\sin\theta_2}{2}} \left(e^{j2\pi d_i \frac{\sin\theta_1-\sin\theta_2}{2}} - e^{-j2\pi d_i \frac{\sin\theta_1-\sin\theta_2}{2}} \right) \right|^2$$

$$= \frac{1}{4}\sum_{i=2}^{M} g_i^2 \left| e^{j2\pi d_i \frac{\sin\theta_1+\sin\theta_2}{2}} 2j \sin\left[2\pi d_i \frac{\sin\theta_1 - \sin\theta_2}{2} \right] \right|^2$$

$$= \frac{1}{4}\sum_{i=2}^{M} g_i^2 \left| e^{j2\pi d_i \frac{\sin\theta_1+\sin\theta_2}{2}} \right|^2 \left| 2j \sin\left[2\pi d_i \frac{\sin\theta_1 - \sin\theta_2}{2} \right] \right|^2$$

$$= \frac{1}{4}\sum_{i=2}^{M} g_i^2 \; 4\sin^2(\pi d_i(\sin\theta_1 - \sin\theta_2))$$

$$= \sum_{i=2}^{M} g_i^2 \sin^2(\pi d_i x) \qquad (2.29)$$

where $x \triangleq \sin\theta_1 - \sin\theta_2$ (2.30)

So, the similarity function is reduced to

$$Q(\mathrm{x}) = \sum_{i=2}^{M} g_i^2 \sin^2(\pi d_i x)$$

Recall that in the study by M.Gavish and A.J.Weiss [7], the proposed similarity function for general array geometry was defined as

$$Q(\bar{x}) = \left\| \mathbf{G}(\mathbf{R}\bar{x} - \mathbf{k}) \right\|^2$$

where $\bar{x} \triangleq \mathbf{u}_2 - \mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_1$ representing the unit vectors pointing to the sources.

As expected, for x=0 the distance between the steering vectors is zero, and $Q(\bar{x})$= 0. The similarity function reaches the tight lower bound at multiple points in this approach. The details of the computations are provided in 2.2.

From Eq. (2.29), it is seen that the similarity function is reduced to a function of a single variable x, where $-2 \leq x \leq 2$.

Q(x) is an even function of x, i.e.,

$$Q(x) = Q(-x)$$

so, the range of x can be restricted to [0,2].

Obviously, Q(0)=0, since the array responses of two angles which are equal, are identical.

For the array not to have ambiguities, we require

$$Q(x) \neq 0 \quad \text{for} \quad x \neq 0. \tag{2.31}$$

To search for the condition of ambiguity occurrence, the following equation has to be solved:

$$Q(x) = \sum_{i=2}^{M} g_i^2 \sin^2(\pi d_i x) = 0 \tag{2.32}$$

This yields the equation $\sin(\pi d_i x) = 0 \qquad i=2,3,\ldots,M$ (2.33)

The values of x satisfying this equation are for

$$\pi d_i x = n\pi \text{ or } d_i x = n \qquad i=2,3,\ldots,M \text{ and } n \text{ is an integer} \tag{2.34}$$

It can also be concluded from Eq. 2.34 that, d_2 has to be greater than 1/2 for an

ambiguity to occur since x can be 2 at its maximum, and if d_2 is less than 1/2, $\pi d_i x$ can not be an integer multiple of π. Hence, if $d_2 < 1/2$, the array will have no ambiguity.

On the other hand, if $Q(x)$ gets very small for $x \neq 0$, a small perturbation in the signals due to noise will make it impossible to distinguish between θ_1 and θ_2, resulting in a gross error. Hence, in the design of the array geometry, the relevant $Q(x)$ is required to be as large as possible for $x \neq 0$.

For a given array geometry, as seen in Fig. 2.1, S($\boldsymbol{d}$) is defined as the minimum of the local minima of $Q(x)$, $\boldsymbol{d}$ and s_0 denoting the optimized array geometry and the lower bound (which will be explained later), respectively.

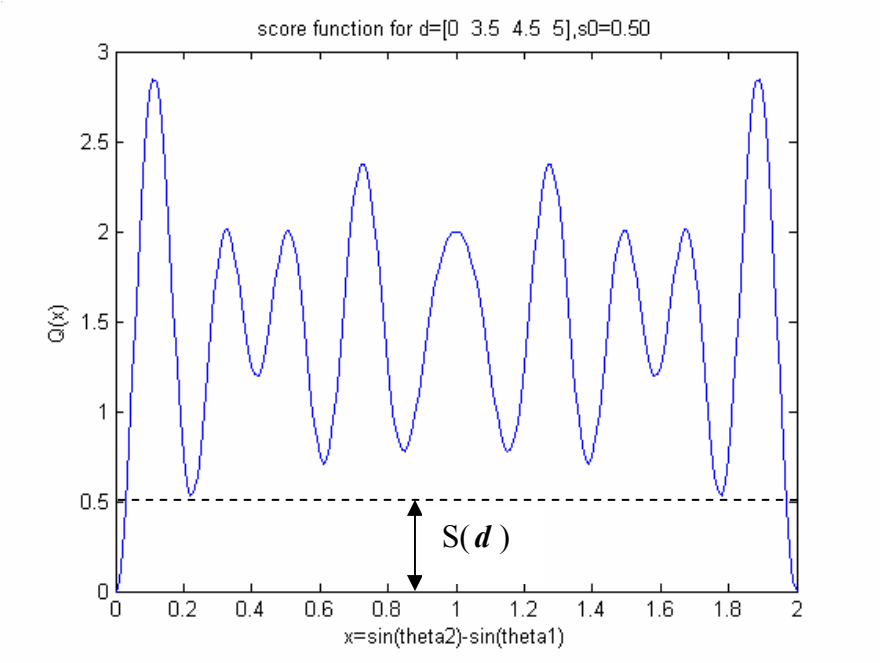


Fig. 2.1: $Q(x)$ as a function of x for an arbitrary array geometry

Thus, one approach could be to maximize S($\boldsymbol{d}$) as a function of $\boldsymbol{d}$ and among such geometries to select the one which has the smallest curvature at x=0 as a means of optimizing the array resolution for the sources close in space.

To investigate the behaviour of $Q(x)$ in the neighbourhood of x=0, we expand $Q(x)$ into Taylor series,

$$Q(x) \cong \sum_{i=2}^{M} \left\{ \; g_i^2 [\pi d_i x]^2 + \text{h.o.t} \; \right\}$$

$$= \sum_{i=2}^{M} \pi^2 g_i^2 d_i^2 x^2 + \text{h.o.t} \qquad (2.35)$$

Since, the resolution measure carries information only around the angles near in magnitude, i.e., $x \cong 0$, higher order terms can be extracted during the evaluations made for resolution.

Hence, a measure of the array resolution around x=0 can be written as

$$R(\boldsymbol{d}) = \sum_{i=2}^{M} g_i^2 d_i^2 \qquad (2.36)$$

So, various optimization approaches can be developed depending on parameters R($\boldsymbol{d}$) and S($\boldsymbol{d}$).

Remark: If two signals arriving from θ_1 and θ_2 have different amplitudes, the measure of similarity should be considered as

$$Q(\theta_1, \theta_2) = \left\| \; \mathbf{a}(\theta_1) - b\mathbf{a}(\theta_2) \; \right\|^2 \qquad (2.37)$$

where the constant b represents the ratio of the amplitude of the second signal to the first one. In this study, however, the case of ambiguity for two signals is not considered. This remark is provided for the sake of completeness.

If the approach is to maximize S($\boldsymbol{d}$), the local minima of Q(x) should be determined first. This is done by taking the first derivative and equating it to zero:

Since $$Q(\mathrm{x}) = \sum_{i=2}^{M} g_i^2 \sin^2(\pi d_i x)$$

$$\frac{dQ(x)}{dx} = \sum_{i=2}^{M} g_i^2 \pi d_i 2 \sin(\pi d_i x)\cos(\pi d_i x) = 0$$

$$= \sum_{i=2}^{M} g_i^2 \pi d_i \sin(2\pi d_i x) = 0 \qquad (2.38)$$

Obviously, there is a finite number of solutions (let this number be N) to this equation. Let us denote these solutions by x_j , j=1,2,…,N.

However, some of these solutions are local maxima and some of them are local minima. To choose the local minima out of these N solutions, second derivatives should be evaluated to decide whether the points are on a decreasing or increasing point of the curve.

As evaluated previously, the first derivative of the similarity function is

$$\frac{dQ(x)}{dx} = \sum_{i=2}^{M} g_i^2 \pi d_i \sin(2\pi d_i x)$$

for which the second derivative is derived as

$$\frac{d^2Q(x)}{dx^2}\Big|_{x=x_j} = \sum_{i=2}^{M} 2g_i^2\pi^2 d_i^2 \cos(2\pi d_i x_j) \tag{2.39}$$

To ensure that x_j corresponds to a local minimum, the second derivative must be greater than zero.

Actually, local minima and maxima will be interleaved since $Q(x)$ is a continuous function, and determination of the first local extrema (x_1) will be enough. The rest can be determined in a switching order. In any case, one of the optimization parameters, S($\boldsymbol{d}$), can be expressed as

$$S(\boldsymbol{d}) = \min_{j\in[1,N+1]} \{Q(x_j)\} \tag{2.40}$$

where x_{N+1} stands for the case of $x=\sin\theta_1 - \sin\theta_2=2$, i.e., $x_{N+1}=2$.

Note: The first derivative of $Q(x)$ was obtained as

$$\frac{dQ(x)}{dx} = \sum_{i=2}^{M} g_i^2 \pi d_i \sin(2\pi d_i x)$$

As seen in the formula, the equation includes a summation in terms of the placements of the array elements. Hence, to maximize S($\boldsymbol{d}$), derivatives of S($\boldsymbol{d}$) wrt various d_i are needed. Let Eq (2.38) have the solutions x_j ($\boldsymbol{d}$) when equated to 0. To simplify the notation

$$Q_j(\boldsymbol{d}) \triangleq Q(x_j(\boldsymbol{d})) \tag{2.41}$$

is defined. Then,

$$\frac{\partial Q_j(\boldsymbol{d})}{\partial(d_l)} = \sum_{i=2}^{M} g_i^2 \frac{\partial}{\partial(d_l)} \sin^2(\pi d_i x_j)$$

$$= \sum_{\substack{i=2 \\ i\neq l}}^{M} g_i^2 \sin(2\pi d_i x_j)\pi d_i \frac{\partial x_j}{\partial(d_l)} + g_l^2 \sin(2\pi d_l x_j)\pi\left[d_l \frac{\partial x_j}{\partial(d_l)} + x_j \right]$$

$$= \left[\sum_{i=2}^{M} \pi d_i g_i^2 \sin(2\pi d_i x_j) \right] \frac{\partial x_j}{\partial(d_l)} + \pi x_j g_l^2 \, sin(2\pi d_l x_j)$$

$$= \pi x_j g_l^2 \, sin(2\pi d_l x_j) \tag{2.42}$$

Among the solutions of this equation when it is equated to zero, the one that minimizes $Q(x)$, yields the optimum array geometry, i.e., the optimum array distances are the values satisfying the equation

$$\frac{\partial S(\boldsymbol{d})}{\partial(d_l)} = \frac{\partial}{\partial(d_l)} [\min_{\mathrm{j}} Q_{\mathrm{j}}(\boldsymbol{d})] \tag{2.43}$$

However, $\mathrm{R}(\boldsymbol{d})$ and $\mathrm{S}(\boldsymbol{d})$ cannot be maximized simultaneously. An array with the possible maximum $\mathrm{S}(\boldsymbol{d})$ cannot possess the smallest curvature at x=0. Hence, a cost function of the form

$$C(\boldsymbol{d}) = w_S \, \mathrm{S}(\boldsymbol{d}) + w_R \, \mathrm{R}(\boldsymbol{d}) \tag{2.44}$$

can be defined and optimization can be realized using this cost function. Another approach is to maximize $\mathrm{R}(\boldsymbol{d})$ while keeping $\mathrm{S}(\boldsymbol{d})$ greater than a specified tight lower bound, denoted by s_0. Keeping $\mathrm{S}(\boldsymbol{d})$ over a lower bound guarentees that gross error would not occur. Among such geometries, the one with the highest resolution is selected by this second approach, which is used in this study.

The design procedure outlined in this chapter is for linear arrays for which the gains of each of the elements are assumed to be known and constant. However,

small gain deviations of the antennas are not likely to affect the score function significantly.

In the following chapter, the optimization algorithm used for solving the problem of maximizing R($\boldsymbol{d}$) constrained to S($\boldsymbol{d}$) to be greater than a specified value will be derived.

CHAPTER 3

OPTIMIZATION METHODS

In this chapter, the problem of optimizing the antenna array geometry for unambiguous estimation of the direction of arrivals (DOAs) of plane waves, according to the constraints developed in Chap.2, is investigated.

First, the structure of the array, used for the problem formulation is stated. Then, the specific case of optimization using genetic programming is discussed. The genetic algorithm developed in this study is based on two different approaches, for which the details of the algorithm are provided. Optimization using MATLAB optimization toolbox is provided next. A discussion on the effects of the specified parameters on the performance of the optimization processes is given, along with the optimized array geometries and the corresponding score functions for changing lower bounds. Finally, a comparison of the performances of the three proposed approaches is provided.

3.1. Array Structure

In this study, specifically, optimization of linear arrays is studied. The model and related representations used for the antenna placement are as shown in Fig. 3.1.

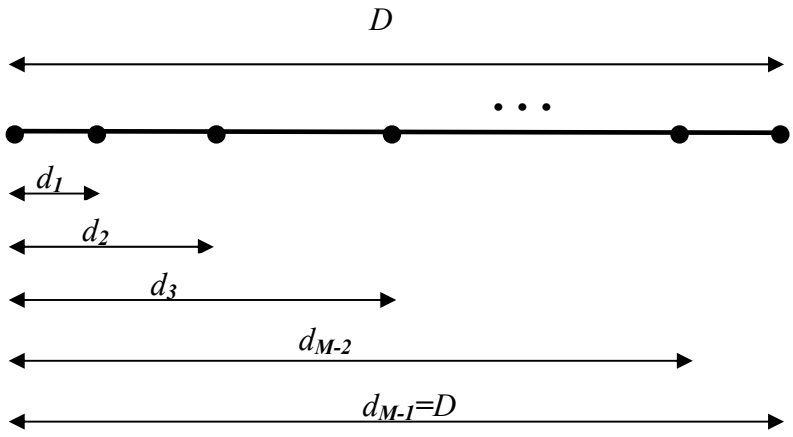


Figure 3.1: Graphical representation of the structure of the array

As expressed before, a linear array of *M* sensors, located on a linear platform of length *D,* is considered. Two of the array elements are constrained to be at the end points of the line segment of length *D* as shown in Fig. 3.1. In between these two, the remaining elements are placed, locations of which are to be optimized. Since, the placements of two of the array elements are fixed, the array structure is represented by *M-2* numbers that are the distances of array elements from the first element which is used as reference. So, the placement vector reduces to

$$\boldsymbol{d} = [d_1, d_2, ..., d_{M-2}] \tag{3.1}$$

In the following sections, different approaches for the optimization of this placement vector are given.

3.2. Genetic Algorithm [8]

In nature, variety is defined as variation in the chromosomes of the entities in the population. The variation in both the structure and the behaviour of the entities

in their environment is a result of this variety. Differences in the rate of survival and reproduction reflect this variation in structure and behaviour. Some entities are better able to perform tasks in their environment, survive and reproduce at a higher rate, whereas less fit entities survive and reproduce at a lower rate. Over a period of time and many generations, the population comes to contain more individuals whose chromosomes are translated into structures and behaviours that enable those individuals to better perform their tasks in their environment and to survive and reproduce. So, as time passes the structure of individuals in the population changes as a result of natural selection. When these visible and measurable differences in structure that arose from differences in fitness is observed, it is concluded that the population has evolved. In this process, structure arises from fitness.

When there exists a population of entities, the existence of some variability having some differential effect on the rate of survivability, is almost inevitable. So, for an entity, to have the ability to reproduce itself, is the most important condition for starting the evolutionary process.

Evolutionary process can be applied to artificial systems. Any problem in adaptation can be formulated in genetic terms. Once formulated in those terms, such a problem can be solved by ‘genetic algorithm’. The genetic algorithm simulates Darwinian evolutionary processes and naturally occuring genetic operations on chromosomes.

Genetic algorithm is a highly parallel mathematical algorithm that transforms a set (population) of individual mathematical objects (typically fixed-length

character strings patterned after chromosome strings), each with an associated *fitness* value, into a new population (i.e., the next *generation*) using operations patterned after the Darwinian principle of reproduction and survival of the fittest after naturally occuring genetic operations.

3.3. The Representation Problem for the Genetic Algorithms

Since genetic algorithms directly manipulate a coded representation of the problem and because the representation scheme can severly limit the window by which a system observes its world, representation is a key issue in genetic algorithm work. The conventional genetic algorithm operating on fixed-length character strings is capable of solving a great many problems. Nevertheless, the use of fixed-length character strings leaves many issues unsettled. For many problems, the most natural representation for a solution is a hierarchical computer program rather than a fixed-length character string. The hierarchies in representing the tasks and subtasks (that is, programs and subroutines) that are needed to solve complex problems have a central importance in genetic coding. The hierachical computer program should have the potential of changing its size and shape since the size and shape of this program that will solve a given problem are generally not known in advance [8].

Virtually any programming language is capable of expressing and executing the general, hierarchical computer programs. In the method provided in this thesis, MATLAB is used as the computer program, due to its ability to perform alternative computations conditioned on the outcome of intermediate calculations, to perform operations in a hierarchical way, to perform computations on variables of many

different types and since the problem involved requires iteration, due to its recursion and dynamic variability.

3.4. Detailed Description of the Proposed Genetic Program

For the conventional genetic algorithm and genetic programming, the structures undergoing adaptation are a *population* of individual points from the search space, rather than a single point. Genetic methods differ from most other search techniques in that they simultaneously involve a parallel search involving hundreds or thousands of points in the search space.

In the proposed genetic program, the structure chosen to undergo adaptation is the antenna array geometry, i.e., the placement of the antennas in space assumed to lie on a linear platform with a predetermined length, D. The number of elements in the array, M, is an input to the program. A population consists of individuals which represent the array structures by $M-2$ numbers as in Eq. (3.1), that are the distances of array elements from the first element used as the reference. The generation of each individual, i.e, $\boldsymbol{d}$ in Eq. (3.1), in the initial population is done by randomly generating $M-2$ numbers arranged in an increasing order. Although the numbers are generated randomly, there is a restriction on the differences of successive numbers to be at least a predetermined resolution, which is determined by the physical constraints for the minimum distance between the antennas, and given as an input to the program.

As described before, fitness of an individual is the probability that it survives to the age of reproduction and reproduces. The most common approach to

measuring fitness is to create an explicit fitness measure for each individual in the population. Each individual in the population is assigned a scalar fitness value by means of some well-defined explicit evaluative procedure. In this work, the fitness value assigned to the array geometries is R($\boldsymbol{d}$), provided that S($\boldsymbol{d}$) is greater than a specified value, s_0. This selection is due to the proposed optimization criterion provided in Chapter 2. The array geometry with the highest fitness value among all the geometries satisfying S($\boldsymbol{d}$) > s_0, is said to be the best array geometry.

To modify the structures undergoing adaptation in genetic programming, an operation known as 'crossover' is applied to the individuals, namely, array geometries in this thesis. This operation creates variation in the population by producing new offsprings that consist of parts taken from each parent. The crossover operation starts with two parental individuals and produces two offspring individuals. The parents are chosen from the part of the population which is composed of a predetermined number of individuals with the highest fitness values, satisfying S($\boldsymbol{d}$) > s_0, as will be explained in detail later.

Two different methods for the representation of array distances in $\boldsymbol{d}$, are implemented. For both of the methods, the number of antennas that will be used in the array, the length of the linear platform used, namely, the last distance that an antenna can be placed, the number of individuals to be created in the population, and the number of generations to be created are input to the program. Also, the minimum distance required between the array elements arising from the physical constraints of the platform and of the antennas used, is provided as an input. In the

first method, the placements of the array elements are represented as decimal numbers, in individuals, $\boldsymbol{d}$. The initial generation is created by assigning random numbers in an increasing order, to the antenna distances in $\boldsymbol{d}$, as expressed previously. Then, the operation begins by reducing the population to the individuals satisfying $S(\boldsymbol{d}) > s_0$. Next, by independently selecting, using a uniform probability distribution, one random point in each parent in the population created in the previous step, to be the crossover point, offsprings are produced by deleting the crossover fragment of the first parent from the first parent and then inserting the crossover fragment of the second parent at the crossover point of the first parent. During the insert operation, array placements are selected in an increasing order in their decimal values. Number of offsprings to be produced are restricted to complete the number of individuals to the predetermined population. This operation is done to increase the probability of every individual in the population to overcome the lower bound restriction. For all the individuals in this population, fitness values, i.e., $R(\boldsymbol{d})$, are evaluated, and individuals are put in order according to increasing fitness values. Among all the individuals, ordered with respect to their fitnesses in this population, 10% of them with the highest fitness values are selected to form a part of a new generation, created to improve quality in the sense of higher $R(\boldsymbol{d})$. The remaining individuals of this new generation are formed by crossover operation implemented on the prior 10%. By this way, the probability of maximizing the fitness value of every individual in the population is increased. This provides better initial populations with possible highest fitness values and adequate lower bounds, for next generations, thus increasing the quality for every next generation.

In addition to the two primary genetic operations of reproduction and crossover in genetic programming, there are five optional secondary operations, known as mutation, permutation, editing, encapsulation and decimation. In this work, only mutation is used which introduces random changes in structures in the population.

The number of the mutant individuals is determined at the beginning of the program. The mutation operation removes whatever is currently at the selected point of an indivudual $\boldsymbol{d}$, and inserts a randomly generated decimal value representing the distance of the array at that point. By this way, random changes are introduced in structures, and transferred to the next generations.

In the second approach, antenna placements are elements of the vector $\boldsymbol{d}$, representing each individual as in the first approach, but they are represented by binary fixed point numbers for which the number of bits is provided as an input to the program, say *K*. Thus, every individual in the population is a *Kx(M-2)* bit array, *M* representing the number of sensors, as in the first case. The structure of an individual can be modeled as shown in Fig. 3.2.

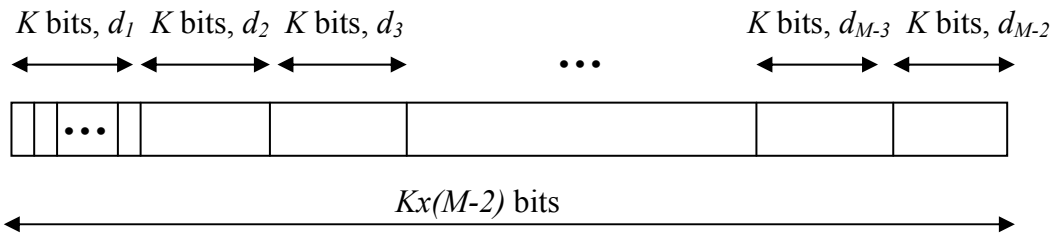


Figure 3.2: Schematical representation of the structure of an invidual

As the number of bits increases, the placement accuracy of the antennas increases. But the properties of the platform used must also be considered, since it may not always be possible to place the antennas at very precise points.

As in the first case, in an individual, the binary numbers representing the antenna placements, are generated in an increasing order. The operation begins by reducing the population to the individuals satisfying $S(\boldsymbol{d}) > s_0$. Then, those individuals which satisfy the lower bound constraint, are put in order according to the value of their fitnesses. A new generation with an improved quality in the sense of higher $R(\boldsymbol{d})$ is generated, then 10% of which is constructed using the individuals with the highest fitnesses in the ordered sequence, generated in the last step. The remaining 90% of this new generation are formed by crossover operation implemented on the prior 10%.

The critical advantage of this approach is realized in the crossover operation. The method is as follows: the individuals in the upper 10%, with the highest fitnesses, are rearranged in an other matrix, so as to group the most significant bits of each antenna distance. This operation is repeated till the least significant bit. At the end of this, the new matrix, say *MSB*, have individuals with *Kx(M-2)* bits ordered as groups according to their significance in the binary numbers, representing the antenna distances. A schematic representation is shown in Fig. 3.3.

The crossover operation is then applied to the individuals in *MSB*. This is done to improve the efficiency of the crossover operation, by increasing the probability of most significant bits to be preserved. Although there is still the

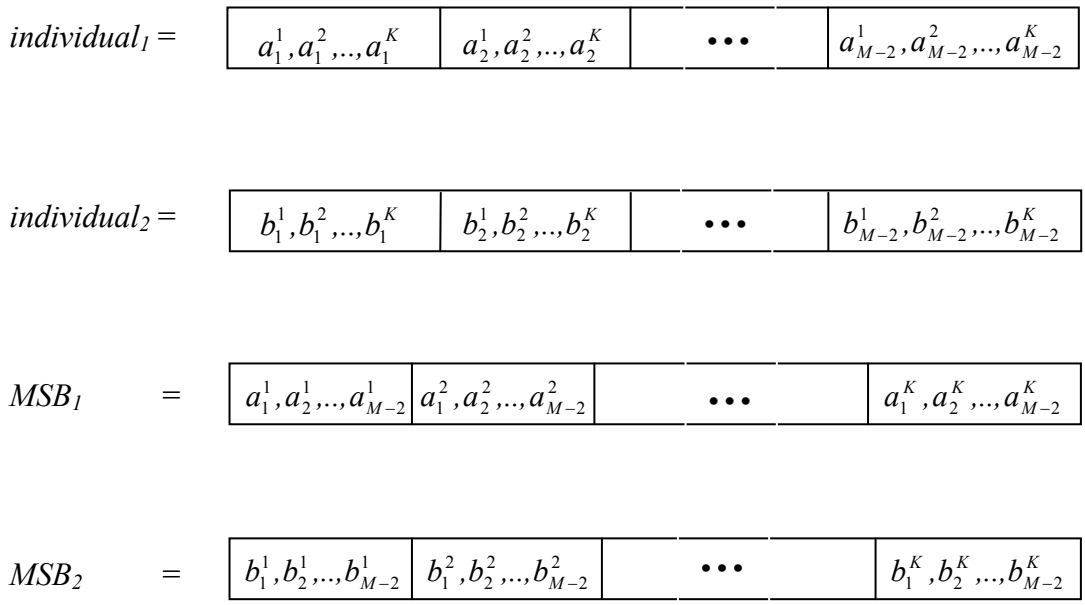


Figure 3.3. Schematic representation of conserving MSB

probability for the crossover point to be in between the most significant bits, and separate them, this probability can be considered negligible compared to the remaining alternative points in the individuals. After the crossover operations, the individuals are transformed to their original structure, and 90% of the new generation is filled with them. By this way, crossover operation is realized between the parents which provide the condition for S($\boldsymbol{d}$) to be greater than s_0, and have the highest fitness values, while at the same time preserving the values at their most significant bits. Mutation operation is also applied here, by inserting a random binary number to a randomly selected point of a randomly selected individual. This insertion, for which the amount is determined as input to the program, is made to introduce random changes in structures in the population. Using this method by binary representations, better initial populations regarding R($\boldsymbol{d}$) are provided to

next generations, and the best geometry is likely to be obtained after among less generations than in the first approach.

The performance of the two methods are nearly the same with a small number of sensors. The optimization results of these two approaches will be presented in the following sections. A more detailed performance comparison for the approaches will be presented later. The results of the genetic optimization algorithm that are used in simulations with MUSIC are given in Chapter4.

3.5. Optimization Using Optimization Toolbox in MATLAB

Other than using genetic algorithm, optimization is realized also with the optimization toolbox of Matlab. In this approach, 'fmincon' function of the toolbox is used. The structure is

[d,fval,exitflag,output]=fmincon(Fun,d_0,a,b,Aeq,Beq,lb,ub,nonlcon,options)

The function 'fmincon' finds a constrained minimum of a function of several variables. In our case, the function for which the minimum is found, is -R($\boldsymbol{d}$). This is the same as finding the maximum value for R($\boldsymbol{d}$). This function subjects the minimization to the constraints defined in 'nonlcon'. For the specific problem in this thesis, 'nonlcon' is defined as S($\boldsymbol{d}$) to be greater than a predetermined value. A set of lower and upper bounds on the design variables $\boldsymbol{d}$ are defined, so that a solution is found in the range lb <= d_i <= ub, where d_i are the elements of vector $\boldsymbol{d}$. During the search, the 'fmincon' function starts at $\boldsymbol{d}_0$ and finds $\boldsymbol{d}$ that minimize the function -R($\boldsymbol{d}$), that is close to the starting point $\boldsymbol{d}_0$. To enlarge the search

space, the optimization is repeated many times with different initial estimates, $\boldsymbol{d}_0$. This is done by creating a set of different initial points of search to be used for the function. Initial points are created starting from the least possible distances for d_1 and padding numbers in an increasing order, for the remaining distances, considering a predetermined resolution in between them. This is done for all the possible values of d_1, restricted by the physical constraints of the platform for the placement of the remaining elements. Optimization results for this approach will also be presented in the next sections.

3.6. Optimization Results for the Array Geometry

In this section, the studies on array geometry optimization are presented. Various scenarios were considered with *design related* parameters such as the length of the linear platform used, the number of antennas used, the resolution for the antenna placement, the number of generations and the initial population created during the genetic programming, and the value of the lower bound, i.e., the constrain for the value of $\mathrm{S}(\boldsymbol{d})$, all varied to some extent.

The optimization processes resulted in non-uniform linear arrays, i.e., arrays with unequal inter-sensor spacings. This is in a sense, a confirmation of the study in [3], in which the performance of uniform and different nonuniform linear array stuructures are compared, and in the event of single targets in additive white Gaussian noise, the nonuniform arrays are found to provide significant improvement over uniform arrays of the same number of elements.

Simulation results for the three methods described in this chapter are not presented separately. For a given constrain for the lower bound of S($\boldsymbol{d}$), all three programs are run to find the optimum array geometry in itself. Among the solutions that are found to be the best in each method, which are very close to each other except for small variations resulting from the resolution differences, the one with the highest R($\boldsymbol{d}$) value is selected as the optimum array geometry. It is observed from the results that, genetic algorithm with the array placements represented as binary numbers, provides the most accurate results using the shortest run time, but all three methods result in nearly the same optimum geometry in different run times.

Since simultaneous illustration of the effects of all the *design related* parameters on the performance of the algorithm is a difficult task, the basic approach is to vary one parameter at a time.

3.6.1. Effect of the Length of the Linear Platform

As expressed previously, the array elements are located on a linear platform of length *D*. All of the optimization methods presented in this study, are designed to optimize the arrays with any length. However, the run time increases with larger array dimensions. In genetic algorithm, the optimum result is obtained in a later generation for example. Also, since the initial generations are created randomly, program has to be run several times to produce different initial generations, from which the optimum geometry will be found.

In this study, the simulation results for the last distance to be 5 times the wavelength, that is, 5λ, will be presented, 5λ selected as an example.

3.6.2. Effect of the Number of Antennas

The optimization process does not include the optimization of the number of antennas that will be used. The process begins with the assumption that the number of antennas to be used is known. The programs are capable of optimizing the geometry with any number of sensors that do not violate the resolution requirements on the linear platform, with the given length. The run time and the number of iterations required to reach the optimum geometry increases with increasing number of antennas used.

In this study, the simulations performed by using four antennas are presented. But the algorithms are designed to optimize the arrays with different number of antennas.

3.6.3. Effect of the Resolution for Antenna Placement

Uncertainity concerning sensor locations can degrade the ability of an array to estimate the location of radiating sources. Then, array calibration becomes an important issue. In this study, a measure of resolution for the inter-element spacings is introduced which represents the minimum distance required between the sensors. This resolution must be kept at a sensible value to ascertain the possibility of that placement practically.

In this study, the minimum distance required between the sensors is selected as 0.05λ.

3.6.4. Effect of the Number of Generations

The maximum number of generations to be produced is an input parameter in the genetic programming codes. During the simulations, it is observed that, as the array length and the number of antennas increase, the optimum geometry is found at later generations. In case of keeping the number of generations less than enough, the algorithm results in an intermediate geometry that is not optimum. So, it is necessary to keep this parameter high enough to assure that the optimum geometry is achieved at that much generation. In the special example of 5 sensors, and 5λ as the array length, the optimum geometry is mostly found after among the 4^{th} or 5^{th} generations.

3.6.5. Effect of the Value of the Lower Bound

The optimization process is realized with different lower bounds, to see the effect of this bound on the direction finding performance. A detailed discussion of the effect of s_0 on the probability of gross error will be presented in Chap.4.

The array structure considered in this chapter assumes 4 sensors, first located at the origin, last located at the end of the linear platform, and the positions of the remaining two are optimized to reach the best performance in direction finding, according to the proposed constraints. As expressed before, the length is selected to be 5 times the wavelength, that is, 5λ, as an example.

To observe the effect of s_0 on the performance of the algorithm, eight different values of s_0, namely, s_0 = 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55,

were considered. In each case, the score function for that special s_0 is also demonstrated (see Figures 3.4-3.11).

It is convenient to remind the case of ambiguity for x=2 here. In Chap. 2, during the search for ambiguity case, it was shown in Eq. 2.34 that, the condition for ambiguity is

$$\pi d_i x = n\pi \quad \text{or} \quad d_i x = n \qquad i=2,3,\ldots,M \text{ and } n \text{ is an integer}$$

The placements of the optimized arrays $\boldsymbol{d}$ =[0 3.5 4.5 5], for s_0=0.50, and $\boldsymbol{d}$ =[0 3.5 4 5], for s_0=0.55, satisfy this condition, and this fact is observed from their score functions with 0 values, at x=2. Since the case for x=2 is valid for θ_1=90 and θ_2= -90 degrees, an inherent ambiguity for the linear array geometry occurs [9], which is out of the scope of this thesis.

- s_0 = 0.20 , Optimum array geometry = [0 4.05 4.8 5], with resulting global minimum point = 0.2202, and R($\boldsymbol{d}$) = 64.4425.

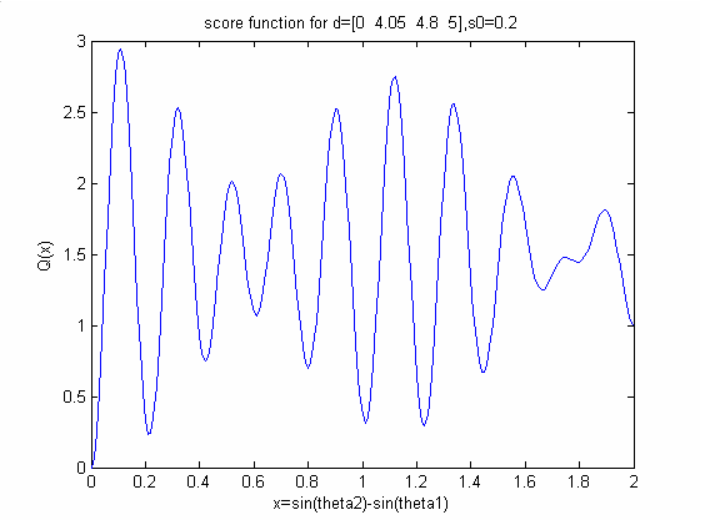


Figure 3.4: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.2.

- s_0 = 0.25 , Optimum array geometry = [0 3.95 4.75 5], with resulting global minimum point = 0.2665, and R($\boldsymbol{d}$) = 63.165

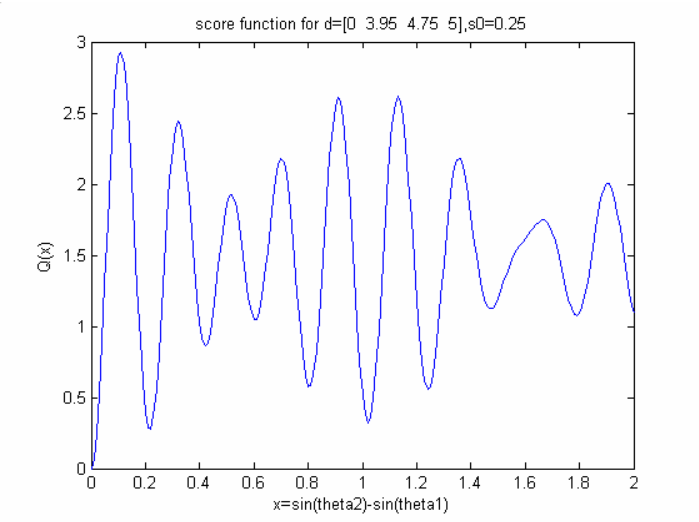


Figure 3.5: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.25.

- s_0 = 0.30 , Optimum array geometry = [0 3.85 4.75 5], with resulting global minimum point = 0.3231, and R($\boldsymbol{d}$) = 62.385

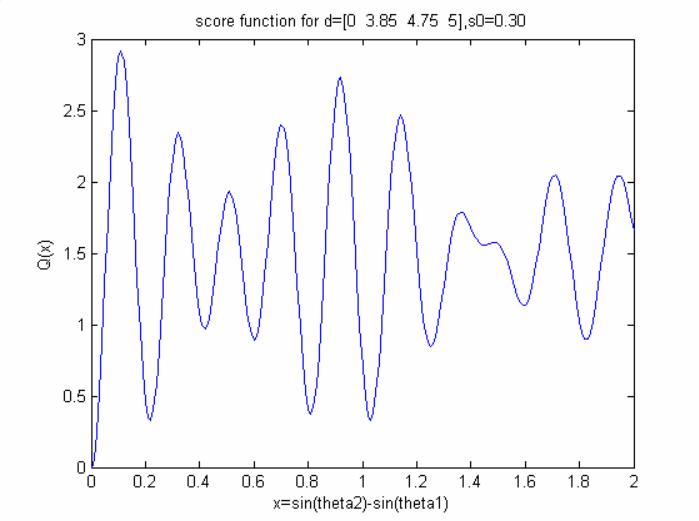


Figure 3.6: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.30.

- s_0 = 0.35 , Optimum array geometry = [0 3.75 4.6 5], with resulting global minimum point = 0.3699, and R($\boldsymbol{d}$) = 60.2225

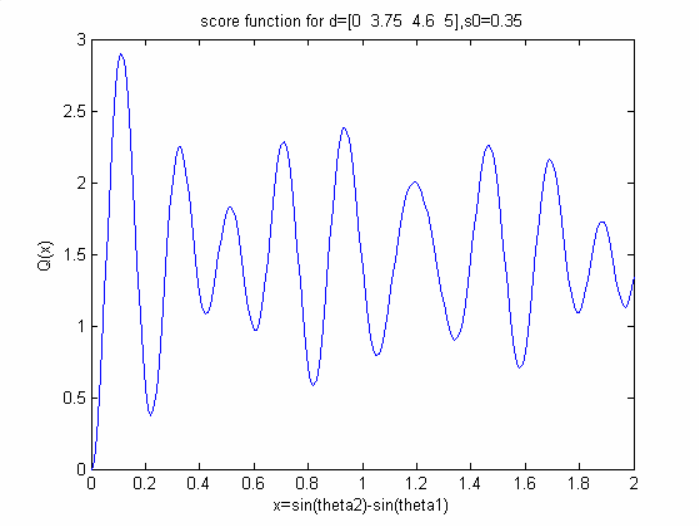


Figure 3.7: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.35.

- s_0 = 0.40 , Optimum array geometry = [0 3.7 4.65 5], with resulting global minimum point = 0.4048, and R($\boldsymbol{d}$) = 60.3125

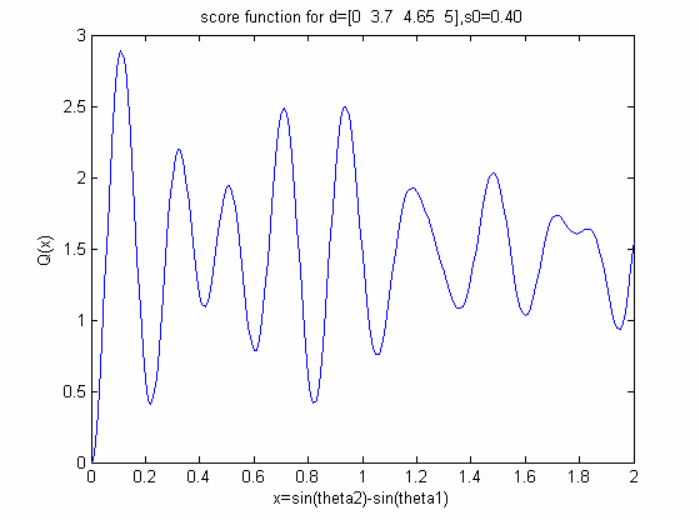


Figure 3.8: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.40.

- s_0 = 0.45 , Optimum array geometry = [0 3.45 4.6 5], with resulting global minimum point = 0.4534, and R($\boldsymbol{d}$) = 58.0625

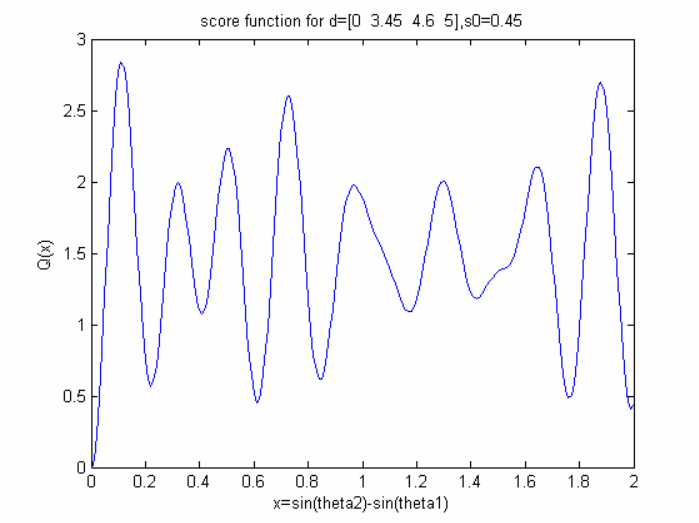


Figure 3.9: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.45.

- s_0 = 0.50 , Optimum array geometry = [0 3.5 4.5 5], with resulting global minimum point = 0.5298, and R($\boldsymbol{d}$) = 57.5

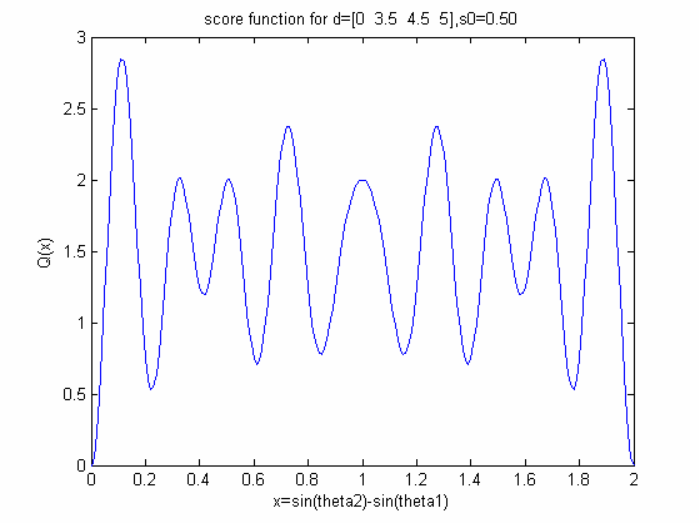


Figure 3.10: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.50.

- s_0 = 0.55 , Optimum array geometry = [0 3.5 4 5], with resulting global minimum point = 0.5879, and R($\boldsymbol{d}$) = 53.25

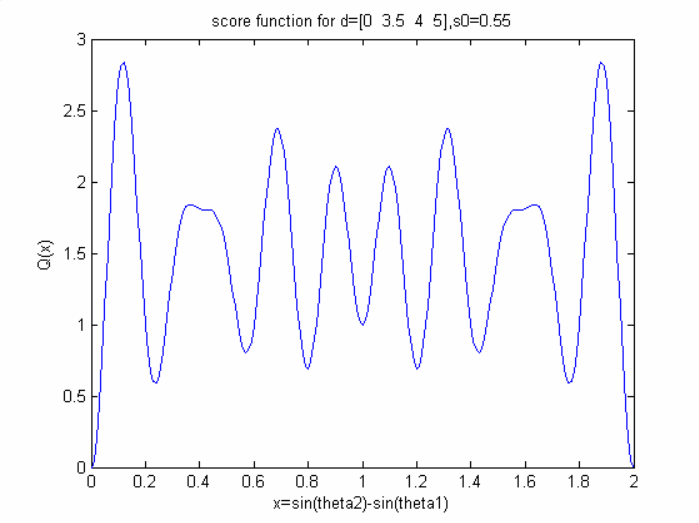


Figure 3.11: Score function as a function of the difference of the sines of the incoming source angles for s_0 = 0.55.

3.7. Comparison of the Performances of Proposed Optimization Approaches

In the description of the different optimization approaches, it is mentioned that, the performance of the methods are nearly the same with a small number of sensors. Although all of the three optimization methods proposed in this study are capable of optimizing the array geometry with any length and any number of sensors, differences arise in their performance concerning run times. To observe this effect in practice, an array of M=40 sensors, placed on a linear platform of length D=20λ, is considered. Placements of the sensors are optimized using the three proposed methods. The optimum array geometries are given in Appendix.A.

Related parameters for the optimization operations are observed as follows:

i) Genetic algorithm with array placements represented as decimal numbers

Optimum array geometry is obtained at the end of the 2nd generation, with about 700 sec. run time. R($\boldsymbol{d}$) is obtained as 1.4155e+004.

ii) Genetic algorithm with array placements represented as binary numbers

Optimum array geometry is obtained at the end of the 2nd generation, with about 450 sec. run time. R($\boldsymbol{d}$) is obtained as 1.4151e+004.

iii) MATLAB optimization toolbox

If the initial distances for all the array elements are tried in an exhaustive manner, run time increases drastically. But, in case of selecting a proper initial condition, for the elements in $\boldsymbol{d}$, optimization results in R($\boldsymbol{d}$) = 1.4156e+004 with a run time of about 75 seconds. Simulations are performed on a computer configured with P4 2.00 GHz CPU, 256 MB RAM, and Windows XP Professional operating system.

CHAPTER 4

SIMULATIONS with MUSIC

In the first part of this chapter, the MUSIC algorithm is described in detail. Then, optimum array geometries obtained in Chap.3 are employed in direction finding with the MUSIC algorithm. The simulations are performed for a specified number of iterations, and after many iterations, the statistical properties of the MUSIC spectrum for a given array geometry, are obtained. These properties are then used to compute the probability of gross error in finding the direction of the arriving signal as well as the resolution in the vicinity of the true DOA. It would be useful to remind here that, the proposed optimization approach for the sensor placements is applicable to any direction finding algorithm. MUSIC is selected in this study as an example algorithm due to its low computational complexity.

Simulations are carried out with MATLAB, in order to observe the effects of several parameters for MUSIC. Those parameters are the change in the statistical properties of the signal during observation duration, number of samples, number of antennas, number of incoming signals, number of iterations, antenna placement, and signal-to-noise ratio (SNR). The simulation results provided here however, include only the effects of antenna placements which are found to be optimum, and SNR, keeping the remaining parameters at fixed values, since the effects of those

parameters on performance are out of the scope of this study. In the following sections, the SNR for each signal is defined as:

$$SNR = 10\log\left(SignalPower \Big/ \sigma^2 \right), \tag{4.1}$$

where σ^2 is the power of the additive white Gaussian noise corrupting the measurements.

Since the simultaneous illustration of the effects of these two parameters on the performance is difficult, the simulations are carried out by changing one parameter at a time.

4.1. MUSIC (MUltiple SIgnal Classification) Algorithm [12]

The MUSIC algorithm is one of the most widely used DF algorithms in the literature due to its interesting breakdown of the principal components of the input signals. The input signals provide information about the DOA of the received plane-waves as well as the noise received at each element. In our case, only one signal is assumed to impinge on the array.

The core concept of the MUSIC algorithm is that the space of the signal covariance matrix can be divided into two subspaces, the signal subspace and the noise subspace. Due to the eigenvector orthogonality, the noise subspace is orthogonal to the signal subspace and, consequently, to the array response vectors corresponding to the direction of arrival. In other words, the product of the array response vector and the eigenvector corresponding to noise should be relatively small when the angle is close to the direction of signal arrival. Therefore, the

direction of arrival can be estimated as the peak of the MUSIC spatial spectrum given by the inverse of the magnitude of the product between the array response vector and the noise subspace matrix.

The MUSIC algorithm makes use of the equation

$$a^H(\theta)E_n E_n^H a(\theta) = 0, \tag{4.2}$$

where $a(\theta)$ is the array response vector, E_n represents the noise subspace and θ represents the direction of arrival of the signal impinging on the array.

Since, in practice a finite number of noisy data vectors are available, an estimate of the array covariance matrix, which is called the sample covariance matrix is found first. Once the eigendecomposition of the sample covariance matrix is obtained, the DOA estimate can be found by the peak of the function

$$P(\theta) = \frac{1}{a^H(\theta)\hat{E}_n \hat{E}_n^H a(\theta)}, \tag{4.3}$$

which is called the MUSIC spectrum.

Although $P(\theta)$ is not a true spectrum in any sense (it is only the distance between two subspaces), it exhibits a peak in the neighbourhood of the true DOA. In the following sections, however, the reciprocal of $P(\theta)$, called the null-spectrum, will be used to estimate the DOA in the MUSIC algorithm. This will yield a minimum point which represent the DOA. This method is preferred to achieve a known distribution function, which is shown to be Gaussian [11], for the differences of main null and local minima, for the sake of gross error probability calculations.

4.2. Simulation Results

As expressed before, simulations are carried out to see the effects of scenario related parameters by changing one parameter at a time while keeping the others at the values specified for that case.

The first section is mainly a study on the direction finding performance of MUSIC using the array geometry found to be optimum. The optimum array geometries considered are those obtained in Chap.3 and the individual peformances of all these geometries are observed for the MUSIC algorithm for different SNR's, number of snapshots and iterations.

Since single snapshot processing is difficult to solve the DOA of the incoming plane wave satisfactorily due to its inability to integrate the signal in time, i.e., needs increasing the minimum signal to noise ratio for reliable operation, multiple snapshot processing approach, i.e., temporal averaging is used to improve the estimation of the DOA of the incident plane wave. So, a 0dB SNR value along with 100 snapshots, integrates the signal and results in 20dB; similarly, it integrates to 30dB if 1000 snaphots are used. The SNR values and number of snapshots used during the simulations in this chapter are by no means the only ones that satisfy the accurate DOA estimates.

It is observed that other than the main null, there exists other points which seem to be candidates to create ambiguity. In the second sections of each optimization approach subtitle, the behaviour of the null-spectrum in the immediate vicinity of the true DOA is studied. The MUSIC null spectrum is observed on the

same graph for 10000 trials to have an intuition on the behaviour of these probable ambiguous points. For the derivation of the probability of ambiguity, the histograms of the difference of these two probable ambiguous points and the main null are presented. Next, correlation coefficients and probability of ambiguity are computed.

4.2.1. MUSIC Null-Spectrum with Optimized Array Geometries

A detailed examination of Figures 4.1-4.3, which are the MUSIC null spectra for changing s_0, reveals the fact that the difference of the local minima and the main null increases with increasing s_0, meaning less probability of ambiguity. It is also observed that, the algorithm tends to equiripple the local minima with increasing s_0, which is a realization of the general behaviour of minimax problems so to equiripple local minima.

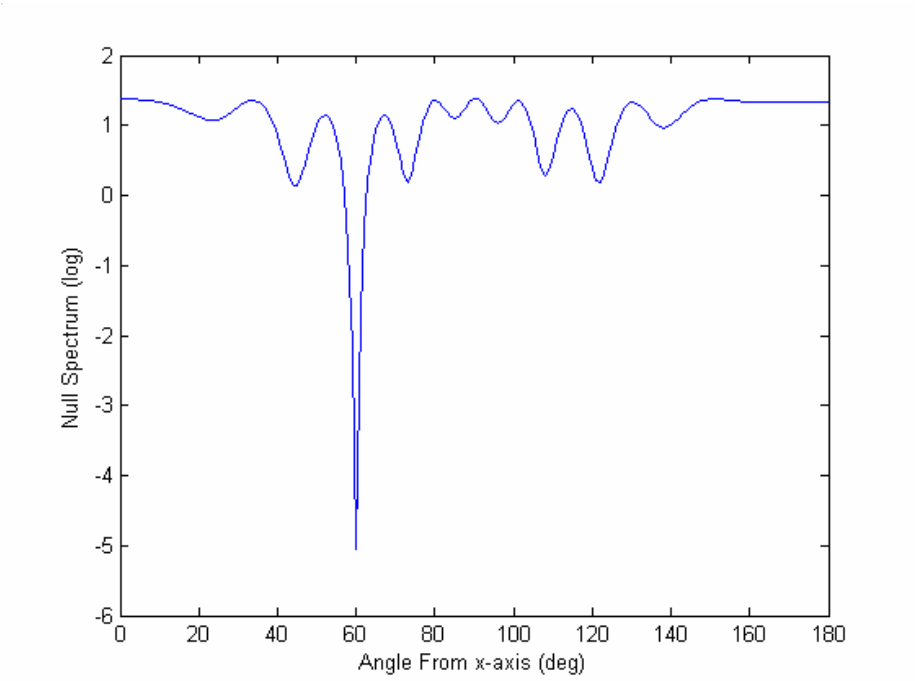


Figure 4.1: Null spectrum for SNR=30dB (10dB with 100 snapshots), no of antennas=4, θ =60, dlast=5, s_0=0.30, optimized geometry=[0 3.85 4.75 5]

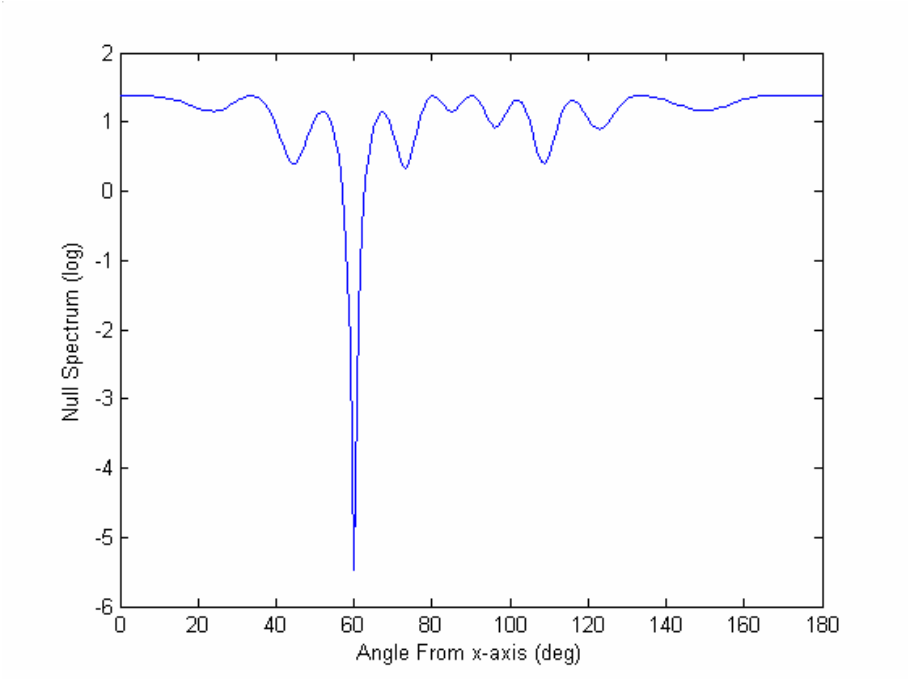


Figure 4.2: Null spectrum for SNR=30dB (10dB with 100 snapshots), no of antennas=4, θ =60, dlast=5, s_0=0.40, optimized geometry=[0 3.7 4.65 5]

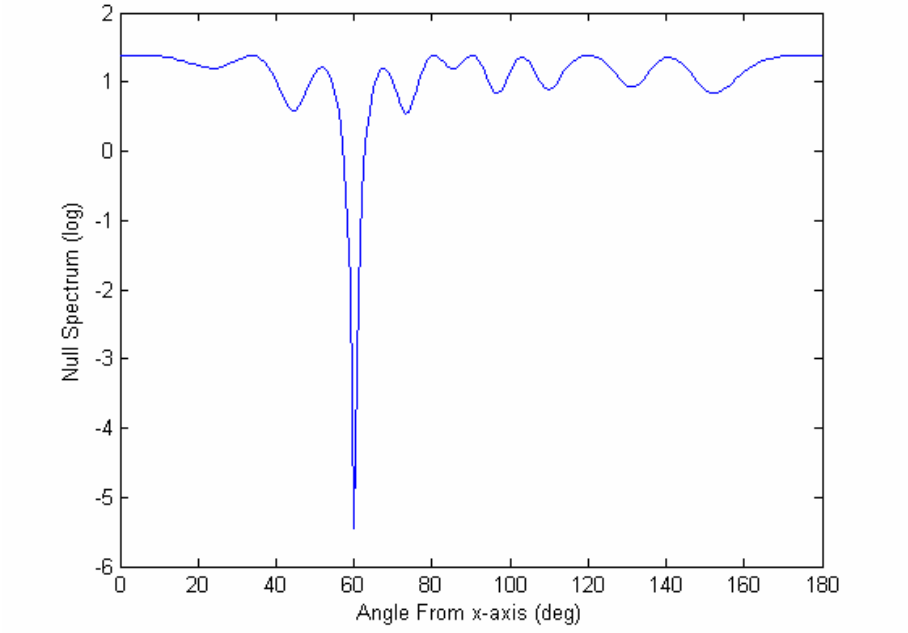


Figure 4.3: Null spectrum for SNR=30dB (10dB with 100 snapshots), no of antennas=4, θ =60, dlast=5, s_0=0.50, optimized geometry=[0 3.5 4.5 5]

4.2.2. Gross Error Probability Calculations

4.2.2.1. Graphical Representation of Statistical Values for Probability of Ambiguity

In this part of the simulations, $\boldsymbol{d}$ =[0 3.7 4.65 5] is used as the optimized array geometry as an example. This geometry was found to be the optimum for s_0 being 0.4, as shown in Chap.3. For this geometry, the MUSIC algorithm is run for 10000 iterations and it is seen that other than the main null point, i.e., the true DOA, there exist other probable null points, for which the probability to be less than that of the main null, is derived. These derivations are made by using two different SNR values to observe the effects of this factor on the probability of ambiguity. All the simulations are made assuming one source at θ=60 degrees. The histograms for the difference of the probable ambiguity points and the main null are provided (see Figures 4.4-4.11). The computation of the probabilities are left to the next section.

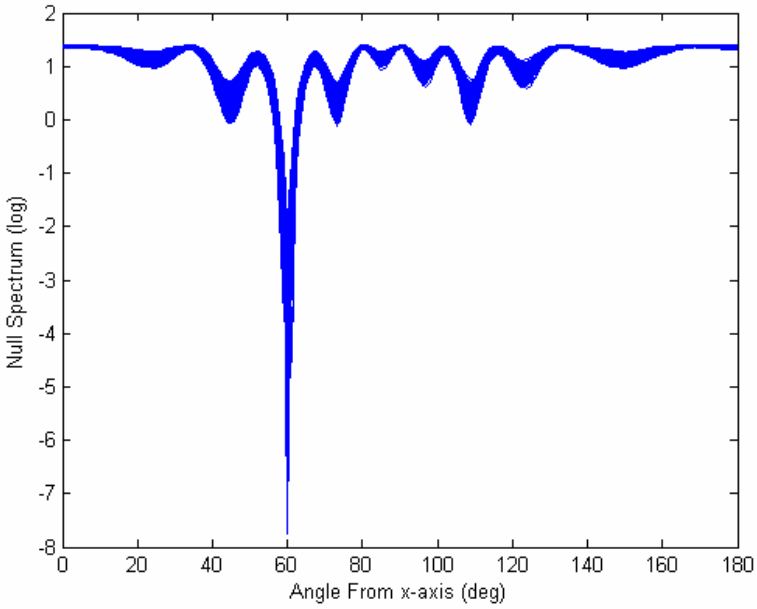


Figure 4.4: Null spectrum for SNR=30dB (10dB with 100 snapshots), no of iterations=10000, θ=60, dlast=5, s_0=0.40, optimized geometry=[0 3.7 4.65 5]

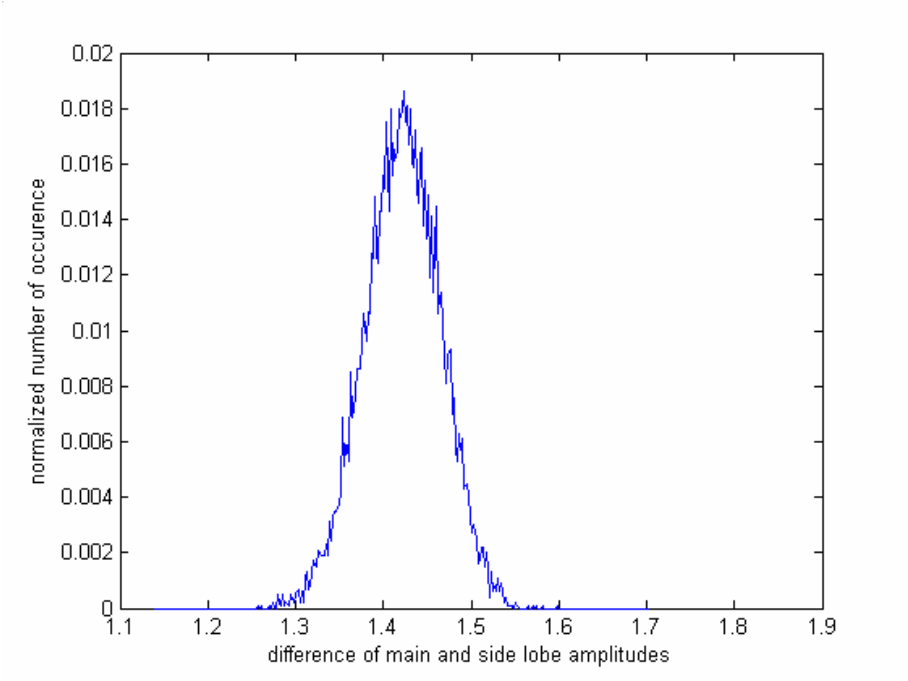


Figure 4.5: Histogram for the difference of the first null (around 44 degrees) and the main null at SNR=30dB (10dB with 100 snapshots), 10000 iterations

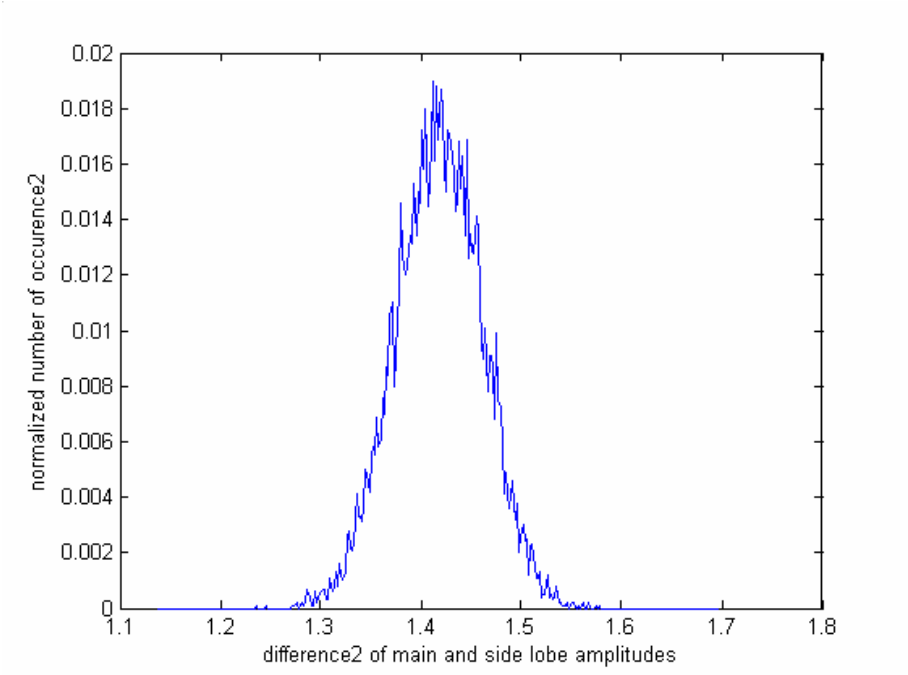


Figure 4.6: Histogram for the difference of the second null (around 73 degrees) and the main null at SNR=30dB (10dB with 100 snapshots), 10000 iterations

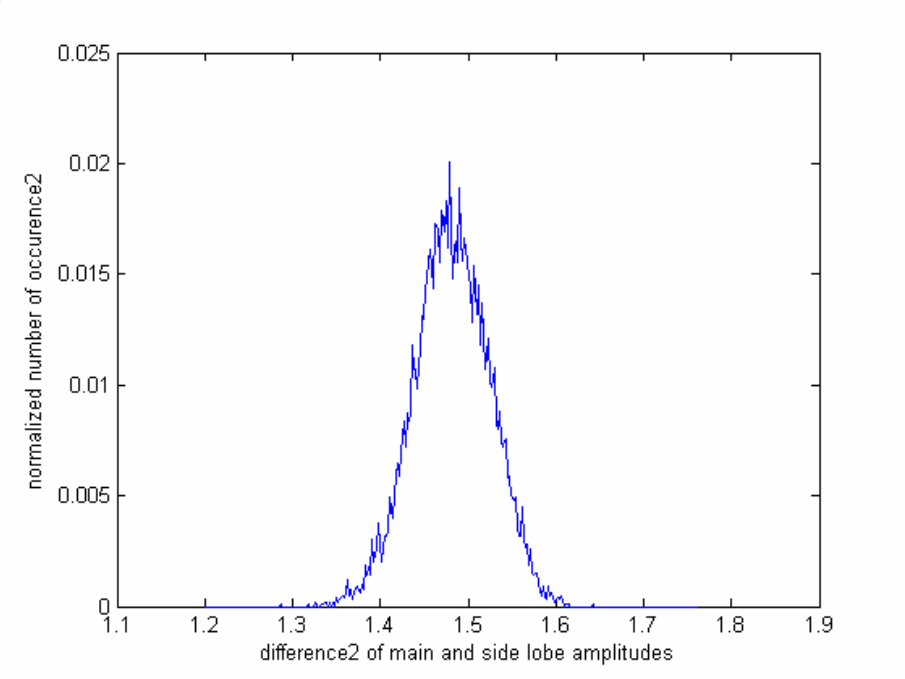


Figure 4.7: Histogram for the difference of the third null (around 109 degrees) and the main null at SNR=30dB (10dB with 100 snapshots), 10000 iterations

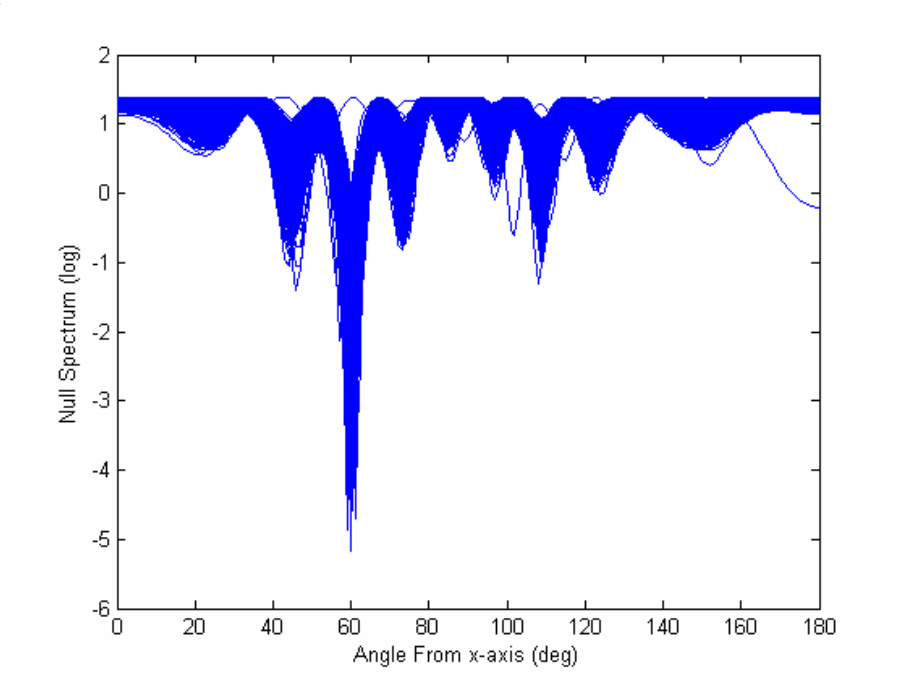


Figure 4.8: Null spectrum for SNR=14dB (-6dB with 100 snapshots), no of iterations=10000, θ =60, dlast=5, s_0=0.40, optimized geometry=[0 3.7 4.65 5]

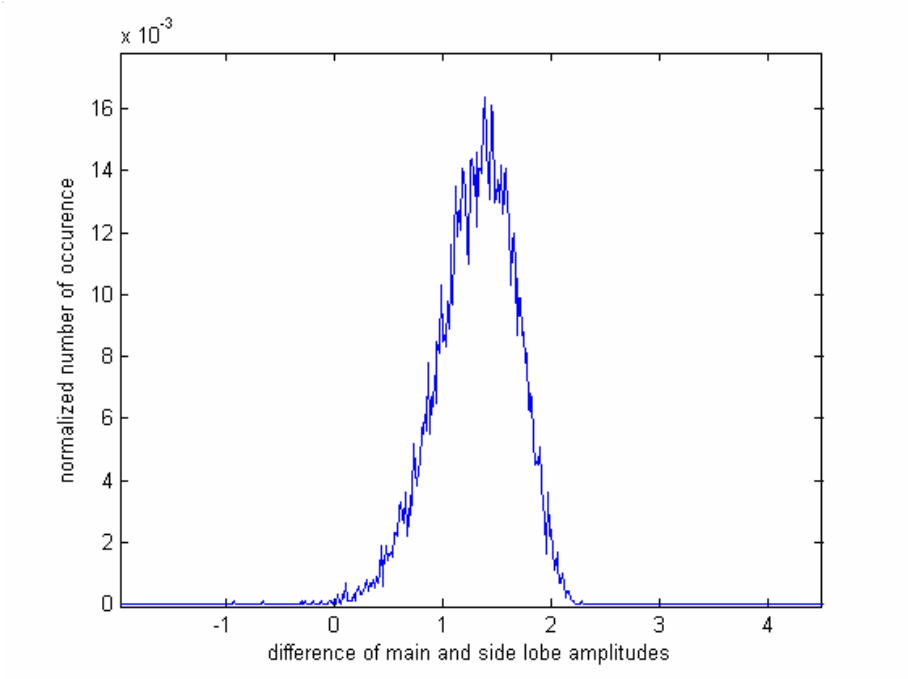


Figure 4.9: Histogram for the difference of the first null (around 44 degrees) and the main null at SNR=14dB (-6dB with 100 snapshots), 10000 iterations

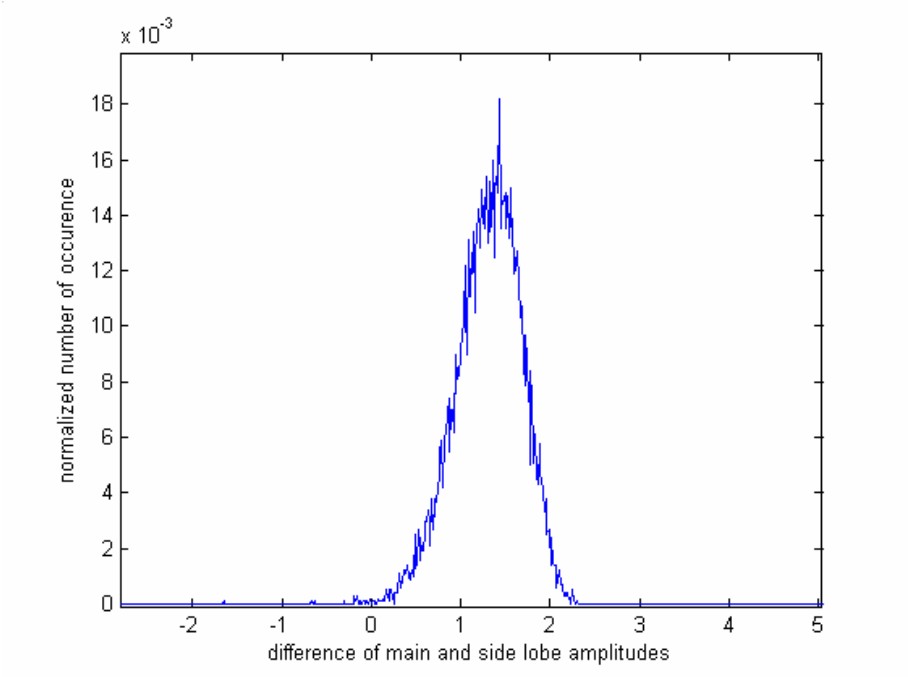


Figure 4.10: Histogram for the difference of the second null (around 73 degrees) and the main null at SNR=14dB (-6dB with 100 snapshots), 10000 iterations

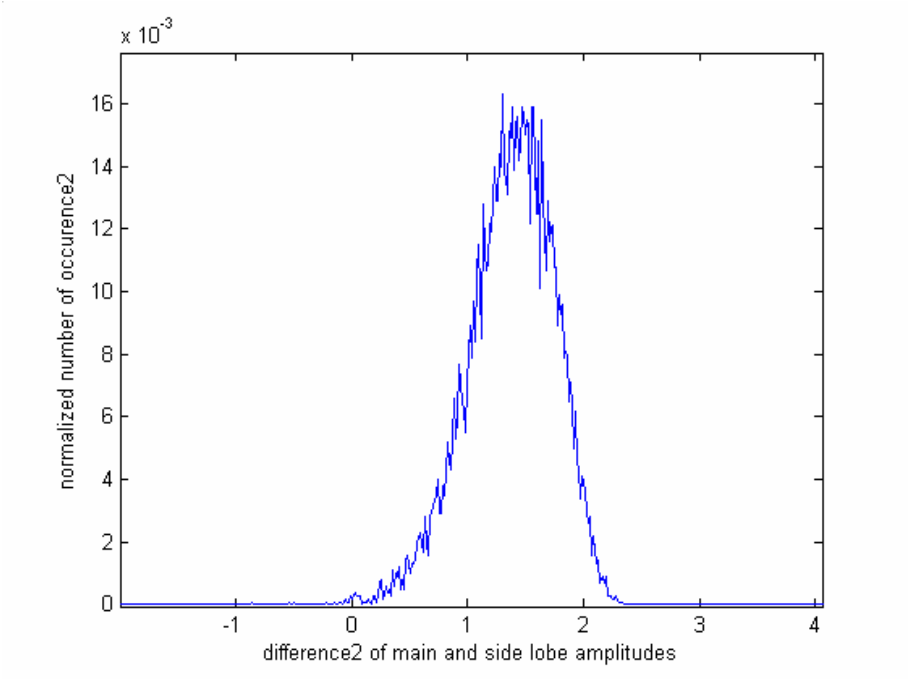


Figure 4.11: Histogram for the difference of the third null (around 109 degrees) and the main null at SNR=14dB (-6dB with 100 snapshots), 10000 iterations

4.2.2.2. Computation and Comparison of Gross Error Probabilities

To obtain the probability of gross error with a given array geometry and SNR, it is necessary to model the distribution of the difference of the main null and the local minima. Modeling of this difference is necessary, because local minima are candidates for gross error with their probability to fall below the main null, so to be detected as the true DOA.

Figure 4.4 through Figure 4.7, for instance, show the MUSIC null spectrum and difference of main null and local minima histograms obtained as a result of 10000 trials using the optimum array geometry found for the lower bound s_0 as 0.40, in a scenario where SNR= 30dB. The simulations for this special case imply

that the spectrum of the difference of the global and local minima have Gaussian distributions with a large amount of iterations, [11]. The approach in calculating the gross error probability is to derive the probability of error individually for each probable gross error region, i.e., the local minima that seem to be canditates to create gross error, first. Then, the correlation among all the pairs is found. The simulation results show that it is not always possible to assume the pairs of these random varibles as uncorrelated. So, a worst case calculation is done for the probability of gross error by adding all individual error probabilities.

Next, similar calculations are carried out for the same array geometry for a lower SNR value, 14dB, to observe the effect of SNR on the gross error probability. From the null spectrum, it is observed that, the values of the local minima in the null spectrum are closer to the global minimum than in the case of 30dB. This fact is realized in the probability of error calculations with a resulting error probability higher than that in the high SNR case.

For both cases, the procedure followed in calculating the probability of error is to first find the means and variances of the difference curves for 10000 iterations, and then to calculate the correlation coefficient [10], between these random variables, to reach a statistical description. As expressed before, the simulations are performed pairwise for the probable gross error points, i.e., if there exist 3 local minima, for instance, that have the probability to cause gross error, two among these three points are selected, and the correlation among these two is calculated first. This is applied for the other pairs in the same way. Then, the functions, for

which the mean, variance and correlation coefficients are calculated, are used in the probability of error derivations.

Remark: Before the probability of gross error calculations for the optimized geometries in detail, it is convenient to provide a short theoretical background for Gaussian random variables and error pobability derivations with such a distribution, here.

Let *x* denote a Gaussian distributed random variable of mean μ_x and variance σ_x^2. The probability density function of such a variable is defined by

$$f_x(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}}, \qquad -\infty < x < \infty \tag{4.4}$$

A random variable is said to be normally distributed if its density function is a Gaussian curve.

The corresponding distribution function is given by

$$F_x(x) = \int_{-\infty}^{x} f(y)dy = \frac{1}{2} + erf\frac{x-\mu_x}{\sigma_x} \tag{4.5}$$

where the definition of error function is given as

$$erf(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2} dy \tag{4.6}$$

Throughout the error probability calculations, the function in (4.5) is used, with the upper bound of the integral being 0, pointing to the search for the region where the amplitude of secondary minima fall below the main null, yielding gross error.

Some notations for the MUSIC null spectrum are defined in Fig. 4.12, which are going to be used in the remaining of this chapter for error probability calculations.

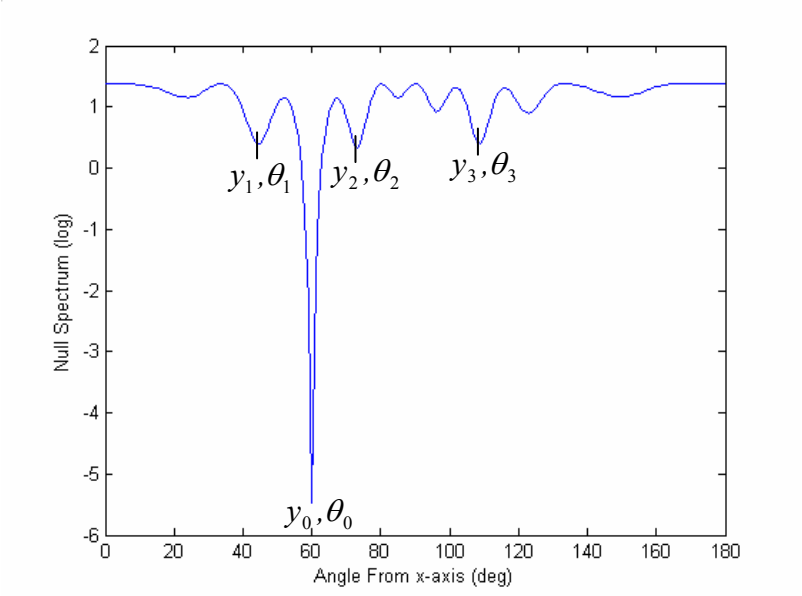


Figure 4.12: An example null spectrum for notation settings

As shown in Fig. 4.12, the local minima that seem to have a probability to fall below the main null intuitively, are given names representing both their amplitudes in the *y-axis,* i.e., null spectrum values, and their placements on the *x-axis,* representing the DOA's corresponding to those minima. All the local minima are not taken into consideration, but the ones which are dominant considering their null spectrum values are employed in the gross error probability calculations. The number of candidate local minima to be employed may change related to s_0 and SNR. Different cases will be dealt with in the next sections.

Since a gross error occurs in case of one of the local minima falling below the main null, the parameters to be considered in the calculations are the differences of

the local and the global minima. Looking at the behaviour of these values around 0, the decision on whether there exists a gross error or not, can be made. Let $d_1, d_2, d_3, \ldots, d_P$ denote the differences, where P is the number of candidate local minima to create gross error. For the example spectrum in Fig. 4.12, these parameters are defined as

$$d_1 = y_1 - y_0 \tag{4.7}$$

$$d_2 = y_2 - y_0 \tag{4.8}$$

$$d_3 = y_3 - y_0 \tag{4.9}$$

To have a more general intuition on the behaviour of the MUSIC null spectrum, 10000 iterations are made. The distributions of the differences are obtained from the result of these iterations, using MATLAB. So, y_i and d_i (i=1,2,…,P) for the iterative case, represent the mean of the 10000 values of the null spectrum, and the differences of these means respectively.

4.2.2.2.1. Probability of Error for SNR = 30dB

It can be seen in Fig.4.4 that, for 10000 iterations, the global minimum is in the neighbourhood of 60 degrees, which is the angle of arrival of the incoming signal. Other than this minumum point, there exist accumulations around three other regions. These regions are the probable gross error angles which are then used in error probability calculations. It can be observed that these local accumulation areas are around 44±1 degrees, 73±1 degrees, and 109±1 degrees. The approach, as explained above, is to deal with the difference of the values of the MUSIC spectrum around these local minima and the global minimum around 60 degrees. According

to the notations introduced previously, gross error probabilities for d_1, d_2, and d_3 have to be calculated. For this example, d_1 corresponds to the difference of y_1 around θ_1 (44±1) and y_0 around the DOA, i.e., θ_0 (60±1 degrees), similarly, d_2 to the difference of y_2 around θ_2 (73±1) and y_0 around θ_0, and d_3 to the difference of y_3 around θ_3 (109±1) and y_0 around θ_0.

i) Gross error probability for d_1

Fig. 4.5 shows the distribution of the difference of the values of the MUSIC spectrum for y_1 around θ_1 (44±1) and y_0 around θ_0 (60±1), i.e., d_1, for 10000 iterations. This curve is obtained from the null spectrum by first setting the values of the regions other than θ_1 and θ_0 to adequately high values, to overcome the interferences out of the region of interest, i.e., to eliminate the probability of the values of these regions to fall below the main null. By this way, the gross error only around θ_1 is taken into account in this step. So, this probability has the mathematical interpretation as $P(d_1 < 0 \mid d_2, d_3 > 0)$ since during the individual calculation of error probability for d_1, d_2 and d_3 are assumed to be free of gross error because of the reason explained above. Error would likely to occur where this Gaussian distribution curve falls below zero, meaning that local minimum at θ_1 reached the global minimum at θ_0 and fell below it. From Figure 4.5, it is observed that, the Gaussian curve is always above zero point at this SNR, which yields a zero probability of error intuitively. Using the mean and variance of the resulting curve, the above probability is calculated with Eq. 4.11 as 5.2991e-5.

ii) Gross error probability for d_2

In a similar manner as for d_1, the error probability for d_2 is calculated. Fig. 4.6 shows the distribution of the difference of the values of the MUSIC spectrum for y_2 around θ_2 (73±1) and y_0 around θ_0(60±1), i.e., d_2, for 10000 iterations. $P(d_2 < 0 \mid d_1, d_3 > 0)$ is found to be 5.2790 e-5, using Eq. 4.11.

iii) Gross error probability for d_3

In Fig. 4.7, the distribution of the difference of the values of the MUSIC spectrum for y_3 around θ_3 (109±1) and y_0 around θ_0(60±1), i.e., d_3, for 10000 iterations, is shown. Using the same method in **i**, $P(d_3 < 0 \mid d_1, d_2 > 0)$ is evaluated using Eq. 4.11 and found as 5.2575e-5.

The idea in calculating the overall gross error probability, is to obtain an intuition about the correlation of the local areas first. Correlation coefficients calculated among the pairs are as follows:

$$\rho_{d_1,d_2} = -0.6925, \quad \rho_{d_1,d_3} = -0.0640, \quad \rho_{d_2,d_3} = -0.0857.$$

Although, it is observed for all cases that the mean of the product of the pair under concern is nearly equal to the product of their means, this is not enough to conclude that these events are independent looking at their correlation coefficients. However, to have an idea about the general behaviour of the algorithm, a worst case calculation is carried out by adding all the individual error probabilities as

$$P_{gross\_error} = P(d_1 < 0 \mid d_2, d_3 > 0) + P(d_2 < 0 \mid d_1, d_3 > 0) + P(d_3 < 0 \mid d_1, d_2 > 0) \tag{4.10}$$

For the specific case of 10dB SNR, using 100 snaphots, meaning 30dB SNR with the integration effect of the multiple snapshot approach, the addition operation results in 0.016% probability of error, which is very close to zero, using the values of gross error probabilities calculated individually for each region.

4.2.2.2.2. Probability of Error for SNR = 14dB

As can be seen from Fig. 4.8, for 10000 iterations, the DF algorithm performance degrades as SNR is decreased, as compared to Fig. 4.4. The global minimum seems to appear around neighbourhood of 60 degrees, most of the time. Other accumulations around various regions are more scattered than in the case of 30dB SNR. These probable gross error areas also seem to have values closer to the global minimum than in the high SNR case. At first sight, these observations create an intuition about the error probability to be higher than the previous example at 30dB SNR. This is also observed at the end of the theoretical probability calculations below.

The approach will be similar to the one used for the high SNR case, i.e., the regions that seem to be candidates for global error, are investigated separately. To reduce the computational complexity, three regions, which seem to be the most probable ambiguity regions, are taken into account in the probability calculations. It can be observed that these local accumulation areas are around 44±2 degrees, 73±2 degrees, and 109±2 degrees. Here, d_1 corresponds to the difference of y_1 around θ_1 (44±2) and y_0 around the DOA, i.e., θ_0 (60±2 degrees), similarly d_2 to the

difference of y_2 around θ_2 (73±2) and y_0 around θ_0, and d_3 to the difference of y_3 around θ_3 (109±2) and y_0 around θ_0.

i) Gross error probability for d_1

Calculations are carried out in a different manner from those for 30dB SNR case, because of the slight asymmetric behaviour of the distribution in Fig.4.9. A careful observation on the simulation results for 14dB SNR case (see Figures.4.9-4.11) reveals that the distributions are not exactly Gaussian, so using error function for this case would result in slight deviations from the true gross error probabilities. Instead, the approach here is to use the ratio of number of negative samples on the distribution function to the total number of samples as a measure of gross error probability. As a result, $P(d_1 < 0 \mid d_2, d_3 > 0)$ is found to be 0.32% (0.0032).

ii) Gross error probability for d_2

Similarly, the distribution function provided in Fig. 4.10 results in 0.33% (0.0033) gross error probability represented by $P(d_2 < 0 \mid d_1, d_3 > 0)$.

iii) Gross error probability for d_3

Similarly, using Fig.11, gross error probability, $P(d_3 < 0 \mid d_1, d_2 > 0)$, is 0.21%.

Correlation coefficients among the pairs for this case are calculated as follows:

$$\rho_{d_1,d_2} = -0.6310, \quad \rho_{d_1,d_3} = -0.0306, \quad \rho_{d_2,d_3} = -0.0140.$$

Using a worst case approximation method as in the high SNR case, total gross error probability

$$P_{gross\_error} = P(d_1 < 0 \mid \ d_2, d_3 > 0) + P(d_2 < 0 \mid \ d_1, d_3 > 0) + P(d_3 < 0 \mid \ d_1, d_2 > 0)$$

is calculated as nearly 0.9% (0.009) for the specific case of -6dB SNR, using 100 snaphots, meaning 14dB SNR with the integration effect of the multiple snapshot approach.

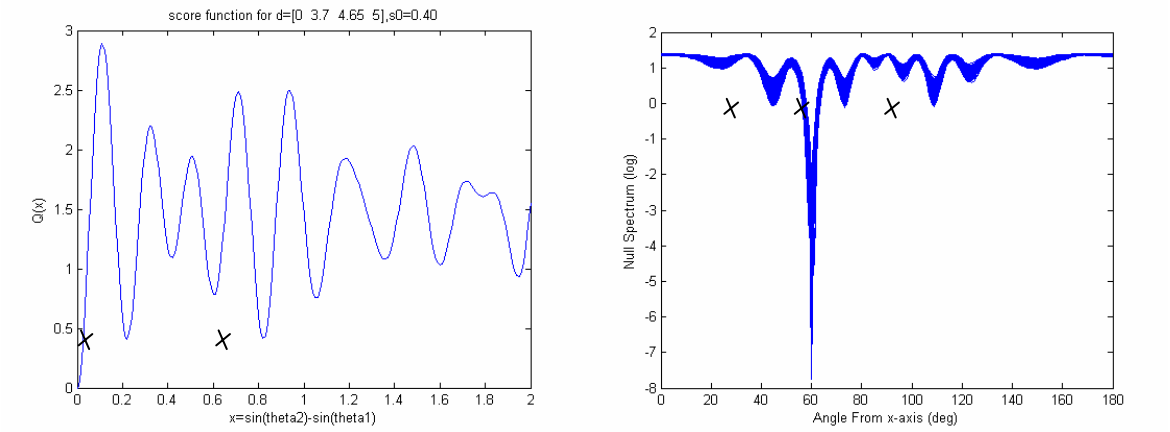
This increase in the error probability is an expected result of decreasing the SNR. It was also observed at the MUSIC null spectrum that, as the signal power decreases with respect to noise power, the local minima are more likely to fall below the main null, so to have a higher probability to cause gross error.

4.2.3. Relation Between the Minima of the Similarity Function and of the Null Spectrum

To investigate the relation between the angles corresponding to the local minima in the null spectrum and the positions of the local minima in the similarity function, some calculations are presented below. As an example, the geometry optimized for s_0=0.40 is selected. It was found to be $\boldsymbol{d}$ =[0 3.7 4.65 5] in Chap.3. This geometry results in the null spectrum provided in Fig.4.4 which is also given in Figure 4.13, for SNR=30 dB (10 dB with 100 snapshots), 10000 iterations and θ_0=60. Related to the convention chosen during MUSIC formulations, angles included in the *x-axis* of the null spectrum and those in the *x-axis* of the similarity function complete each other to 90 degrees, i.e., cosine of one is equal to the sine of the other. This will be reflected to the angle notations in the following calculations.

To provide consistency in the notations, let θ_0 represent the angles around

60±1 degrees, θ_1 around 44±1 degrees, θ_2 around 73±1 degrees and θ_3 around 109±1 degrees in the null spectrum. The approach is then, to find the angles corresponding to the local minima in the similarity function and to check if these results are consistent with the values θ_1, θ_2 and θ_3.



(a) **(b)**

Figure 4.13: (a) Similarity function for s_0=0.40, and **(b)** corresponding null spectrum

The two significant local minima are marked in Fig. 4.13 (a). These two are selected because they give the intuition to cause gross error more than the other candidates and are enough to see the relation between the minima of two functions, which is actually the aim of this section. Processing all the local minima however, would not be practical.

The corresponding *x-axis* values, i.e., the difference of the sines of two incoming signals for these minima of the similarity function, are 0.22 and 0.82. The problem then reduces to solving the following equations:

i) $sin(90-\theta_1^{'})-sin(90-\theta_0)=0.22$, or (4.11)

ii) $sin(90-\theta_0)-sin(90-\theta_2^{'})=0.22$, or (4.12)

iii) $sin(90-\theta_0)-sin(90-\theta_3^{'})=0.82$ (4.13)

The equations written above are the ones that have possible solutions for the given values of difference of sines. For example, the fourth equation that could be written here with the expression $sin(90-\theta_3^{'})-sin(90-\theta_0)=0.82$, has no real solution. θ_0 represents the angle of arrival, 60 degrees here, because the focus is on the gross error, related to the difference of a local minimum and the value at θ_0, which is the main null corresponding to the DOA.

Solution of equation *i* yields 43.94 degrees for $\theta_1^{'}$, equation *ii* yields 73.73 degrees for $\theta_2^{'}$, and equation *iii* yields 108.66 degrees for $\theta_3^{'}$.

It is clearly seen that the solutions for $\theta_1^{'}$, $\theta_2^{'}$ and $\theta_3^{'}$ correspond to the angles θ_1, θ_2 and θ_3 that are the candidate angles to cause gross error. This fact reveals the close relation between the similarity function and the probable angles for gross error. So, once the similarity function for an arbitrary array geometry is obtained, enough data is collected for estimation of the localizations of angles that may cause gross error.

It is also observed here that, the less the value of the local minima in the similarity function, the high the probability of corresponding angle to cause gross error, which is consistent with the proposed optimization approach.

4.2.4. Effect of the Lower Bound on the Probability of Gross Error

To investigate the effect of the lower bound on the gross error probability, four different array geometries, optimized with different lower bound constraints are used and error probabilities for all of them are calculated. The lower bounds chosen as example and the corresponding optimum array geometries are as follows:

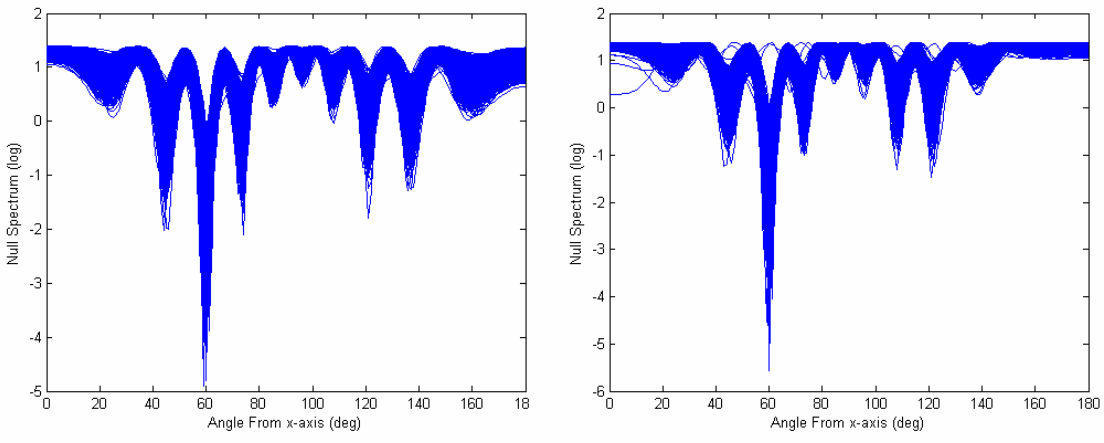
$s_0 = 0.20$, Optimum array geometry = [0 4.05 4.8 5]

$s_0 = 0.30$, Optimum array geometry = [0 3.85 4.75 5]

$s_0 = 0.40$, Optimum array geometry = [0 3.7 4.65 5]

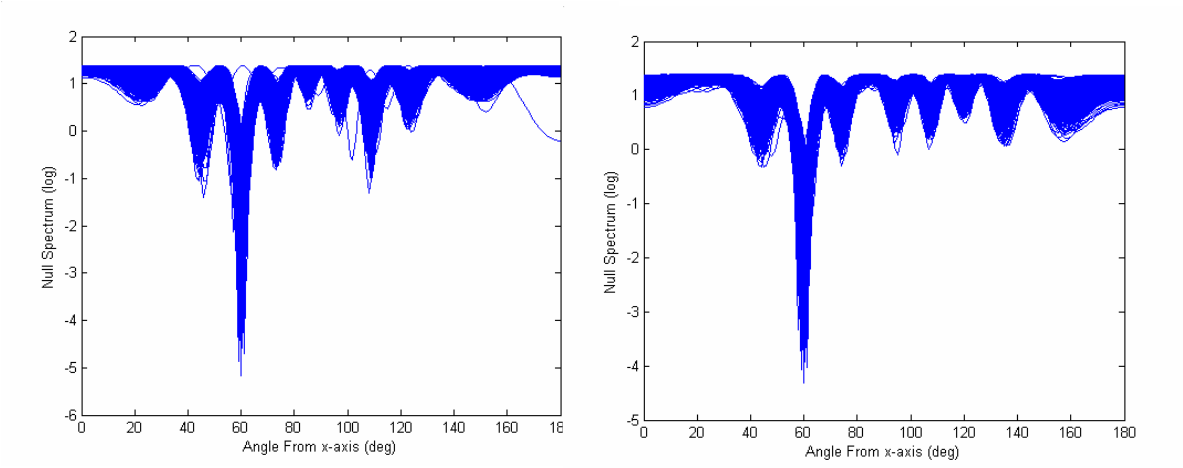
$s_0 = 0.55$, Optimum array geometry = [0 3.5 4 5].

For each of these geometries, MUSIC null spectrum is obtained with 3000 iterations at SNR=14 dB (-6 dB with 100 snapshots). It is observed that, with increasing s_0, the algorithm tends to converge the values of the probable ambiguity regions in the *y-axis,* to each other as seen in Figures 4.14 and 4.15.



(a) **(b)**

Figure 4.14: Null spectrum for **(a)** $s_0 = 0.20$ and **(b)** $s_0 = 0.30$



(a) **(b)**

Figure 4.15: Null spectrum for **(a)** $s_0 = 0.40$ and **(b)** $s_0 = 0.55$

These four examples are investigated in a similar manner as in section 4.2.3.3, and the error probabilities are calculated as 1.82 % for $s_0 = 0.20$, 1.12 % for $s_0 = 0.30$, 0.9 % for $s_0 = 0.40$ and nearly zero for $s_0 = 0.55$ (see Figure 4.16).

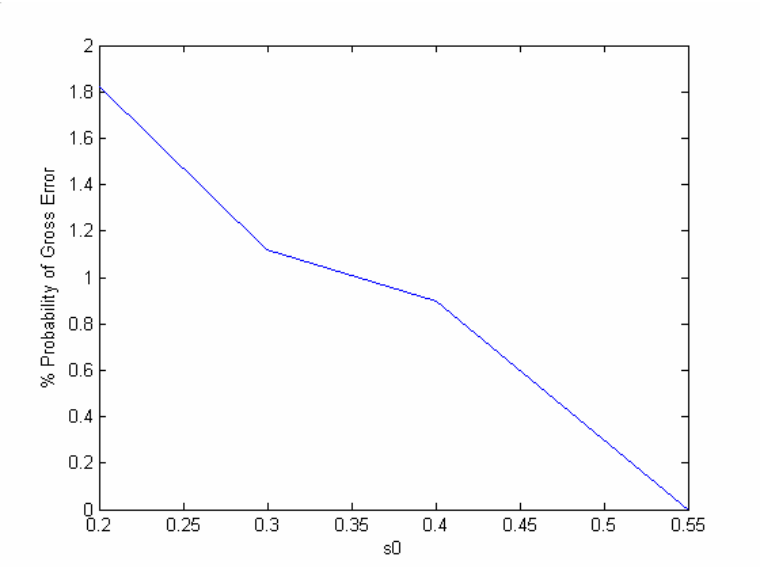


Figure 4.16: Variation of gross error probability with respect to s_0, at SNR=14dB.

It can be seen from the obtained performance curve that, with increasing lower bound, the performance of the algorithm increases in the sense of decreasing probability of gross error. Similar observations can be made using various other optimum geometries.

4.2.5. Resolution Around the True DOA

The optimization problem was defined as follows: maximize R($\boldsymbol{d}$) while keeping S($\boldsymbol{d}$) greater than a specified tight lower bound, s_0. Up to now, the effect of keeping the lower bound greater than s_0 on performance, is studied. Maximizing R($\boldsymbol{d}$) is proposed to provide maximum resolution around the DOA. Here, it is aimed to see this effect through computer simulations. The approach is to select random array geometries with almost equal lower bounds and different R($\boldsymbol{d}$) values first, then to investigate the change in resolution for changing R($\boldsymbol{d}$).

The array geometries selected for this purpose and corresponding s_0 and R($\boldsymbol{d}$) values calculated with Eq. 2.37, are as follows:

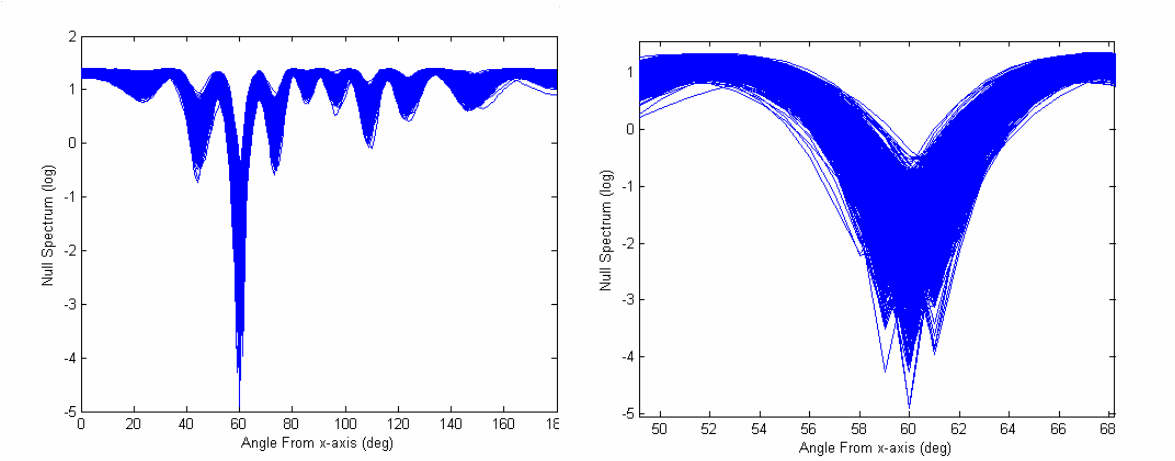
$\boldsymbol{d}_1$=[0 3.75 4.6 5], s_{01}= 0.3699, R($\boldsymbol{d}_1$) = 60.2225

$\boldsymbol{d}_2$=[0 2.1 4.45 5], s_{02}= 0.3661, R($\boldsymbol{d}_2$) = 49.2125

$\boldsymbol{d}_3$=[0 0.9 4.48 5], s_{03}= 0.3684, R($\boldsymbol{d}_3$) = 45.8804

$\boldsymbol{d}_4$=[0 0.94 3.5 5], s_{04}= 0.3660, R($\boldsymbol{d}_4$) = 38.1336

MUSIC spectrum corresponding to $\boldsymbol{d}_1$, as an example, at an SNR of 15 dB (-5 dB with 100 snapshots) with 2000 iterations and the zoomed in part around the DOA are provided in Figure 4.17.



(a) **(b)**

Figure 4.17: (a) Null spectrum at 15 dB SNR, **(b)** zoomed in part of **(a)** around DOA

To gain an intiution for the resolution, rms error around the global minimum of the null spectrum, corresponding to the true DOA, have to be examined for all cases. However, to reach realistic statistics, a huge amount of iterations have to be used during simulations, which caused the computer that is used for simulations, run out of memory. To gain an insight only, simulations are performed with 2000 iterations. Mean values around the global minimum show that the estimates are unbiased. Resulting rms errors are 0.4910 for $\boldsymbol{d}_1$, 0.5122 for $\boldsymbol{d}_2$, 0.5154 for $\boldsymbol{d}_3$, 0.5207 for $\boldsymbol{d}_4$, which are consistent with the proposed optimization constraint, since an increase in rms error means a decrease in resolution. But, as expressed before, for more reliable results, these simulations have to be carried out with quite high number of iterations.

CHAPTER 5

CONCLUSIONS

In this thesis, a criterion for optimizing the geometry of a sensor array for unambiguous DOA estimation is developed, with particular emphasis on the issues of reducing the gross error probability and of improving the resolution performance.

The formulation of the problem is based on a measure of similarity between array response vectors. The similarity measure is formed by finding the difference of the array response vectors for two different angles of arrival. This measure thus is closely related to the probability of error in choosing between two arrival angles using a finite number of noisy samples of the array output. Optimization criteria are extracted from this measure, to satisfy unambiguous DOA estimates by restricting the global minima of the similarity function to s_0, as well as a high resolution capability for the sources close in space, by maximizing $R(\boldsymbol{d})$. Although the design procedure proposed for optimization is applicable to geometry selection for any real array, linear array geometry is chosen due to its lower computational complexity. Performance of the proposed method, using other array geometries, should be analyzed in more detail, which can be the subject of another study.

The proposed optimization methods are based on the suggested optimization

criteria on the similarity function. Optimization is realized by genetic coding or by MATLAB optimization toolbox. Since, the second is a tool for local extrema search, a proper initial condition, for which the results of genetic coding may be used, should be supplied to it, to converge to the optimum geometry. Optimization with genetic algorithm also uses two different genetic coding approaches. The first approach searches for the optimum geometry, with the individuals represented as decimal numbers, whereas the second does the same with the individuals represented as binary fixed point numbers, which is proposed to preserve the data in the most significant bits. Although with a small population, time required for optimization is less for the second approach, when the selected population is large enough, the performances of the two are similar.

The performance of the proposed optimization approach and algorithms developed for optimization are studied through computer simulations. Behaviour of the null-spectrum for MUSIC, using the optimized array geometries are presented, at various SNR. Computations on the gross error probability clearly reveal the influence of various parameters on the estimation error level. It is observed that the algorithm performance depends on the choice of s_0 and SNR. It was found that for a non-uniform linear array of antennas with optimized geometry, the gross error probability increases with decreasing s_0. The minimal error, thus, occurs at highest s_0 value at which the geometry can be optimized. It was demonstrated that the gross error probability also lessens with increasing SNR.

It was also demonstrated that an analogy exists between the resolution of the

array and R($\boldsymbol{d}$), when different array geometries satisfying the same lower bound constraint, are considered. Local accuracy around DOA increases with higher R($\boldsymbol{d}$), i.e., an improvement on system's ability to differentiate between any two closely spaced angles is achieved, pointing out the effectiveness of the proposed optimization approach.

Although the approach is applicable to any geometry, this study focused on linear arrays. Optimization of sensor placements for a 2-D array to maximize the performance of the direction finder may be studied as a future work.

REFERENCES

[1] Hamid Krim and Mats Viberg, "Two Decades of Array Signal Processing Research", IEEE Signal Processing Magazine, pp. 67-94, July 1996.

[2] Arzu Tuncay Koç, "Direction Finding with a Uniform Circular Array via Single Snapshot Processing", Ph.D. thesis, Middle East Technical Univ., 1996.

[3] E.J. Vertatschitsch and S. Haykin, "Impact of Linear Array Geometry on Direction of Arrival Estimation for a Single Source", IEEE Transactions on Antennas and Propagation, Vol. 39, No. 5, pp. 576-584, May 1991.

[4] X. Huang, J.P. Reilly, and M. Wong, "Optimal Design of Linear Array of Sensors", in Proc. ICASSP 1991, pp. 1405-1408.

[5] Naushad Dowlut and Athanassios Manikas, "A Polynomial Rooting Approach to Super-Resolution Array Design", IEEE Transactions on Signal Processing, Vol. 48, No. 6, pp. 1559-1569, June 2000.

[6] Ding Qui and Xiao Xianci, "DOA Ambiguity vs. Array Configuration for Subspace-based DF Methods", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No. 7, pp. 488-491, July 1998.

[7] Motti Gavish and Anthony J. Weiss, “Array Geometry for Ambiguity Resolution in Direction Finding”, IEEE Transactions on Antennas and Propagation, Vol. 44, No. 6, pp. 889-895, June 1996.

[8] John R. Koza, *Genetic Programming,* MIT Press, 1992.

[9] Kah-Chye Tan and Zenton Goh, “A Construction of Arrays Free of Rank Ambiguities” IEEE Transactions on Signal Processing, 6, pp.545-548, 1994.

[10] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes,* McGRAW-HILL, 1968.

[11] Bülent Baygün, “Performance Analysis of the MUSIC Algorithm in Direction Finding Systems”, Master’s Thesis, Middle East Technical Univ., July 1988.

[12] Metin Şengül, “Implementation of the MUSIC Algorithm on a Multiprocessor Architecture”, Master’s Thesis, Middle East Technical Univ., December 2002.

APPENDIX A

OPTIMUM ARRAY GEOMETRIES

i) Genetic algorithm with array placements represented as decimal numbers:

d = [0 17.8000 18.1500 18.2000 18.2500 18.3000 18.3500 18.4000 18.4500 18.5000 18.5500 18.6000 18.6500 18.7000 18.7500 18.8000 18.8500 18.9000 18.9500 19.0000 19.0500 19.1000 19.1500 19.2000 19.2500 19.3000 19.3500 19.4000 19.4500 19.5000 19.5500 19.6000 19.6500 19.7000 19.7500 19.8000 19.8500 19.9000 19.9500 20.0000]

Elapsed_time =700.4 seconds

Best fit value of the generation = 1.4155e+004

ii) Genetic algorithm with array placements represented as binary numbers:

d =[0 17.7973 18.1490 18.1978 18.2466 18.2955 18.3492 18.3980 18.4469 18.4957 18.5495 18.5983 18.6471 18.6960 18.7497 18.7985 18.8474 18.8962 18.9499 18.9988 19.0476 19.0965 19.1453 19.1990 19.2479 19.2967 19.3455 19.3993 19.4481 19.4969 19.5458 19.5995 19.6484 19.6972 19.7460 19.7998 19.8486 19.8974 19.9463 20.0000]

Elapsed_time = 452 seconds

Best fit value of the generation = 1.4155e+004

iii) Optimization using MATLAB Optimization Toolbox:

d = [0 17.8240 18.1500 18.2000 18.2500 18.3000 18.3500 18.4000 18.4500 18.5000 18.5500 18.6000 18.6500 18.7000 18.7500 18.8000 18.8500 18.9000 18.9500 19.0000 19.0500 19.1000 19.1500 19.2000 19.2500 19.3000 19.3500 19.4000 19.4500 19.5000 19.5500 19.6000 19.6500 19.7000 19.7500 19.8000 19.8500 19.9000 19.9500 20.0000]

Elapsed_time = 75 seconds

Best fit value of the generation = 1.4156e+004