Beamforming for wireless communications

Qu, Lin

2007

Qu, L. (2007). Beamforming for wireless communications. Master’ s thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/3524

https://doi.org/10.32657/10356/3524

BEAMFORMING FOR WIRELESS COMMUNICATIONS

QU LIN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2006

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

....

Date QU LIN

Acknowledgements

First of all, I would like to express my deepest gratitude to my advisor Prof. Ser Wee for his guidance and support throughout the duration of this work. Since I started this work, he has always given me motivation and encouragement to excel in research. It has been a pleasure to work under his supervision and I would like to maintain this good relationship with him.

Furthermore I am indebted to Dr. Shao Zhenhai and the National Institute of Information and Communications Technology of Japan for the fellowship that I have received during the time of this research. This work would have been impossible without their financial support.

Thanks also go to the members of the Center for Signal Processing in the Nanyang Technological University for the useful discussions, support and fun that I have received and enjoyed while working together with them.

Next, a big thanks to my friends in Singapore and China who gave me great help and encouragement, and made my off campus life filled with joy. Special gratitude extends to Xu Zhiming for his willingness to lend an ear at all hours, unceasing support and encouraging talks when I was frustrated. He has been a vital source for driving me to where I am today.

Finally, I dedicate this thesis to my beloved parents for their wisdom, support

and sacrifice. They always give me the most support for my academic pursuit and encourage me to do my best. Their understanding and love has helped me through difficult times and challenges in completing this work.

Abstract

This thesis studies the DOA estimation and beam pattern synthesis techniques for wireless communications applications. In particular, three specific problems have been investigated and novel algorithms have been proposed in this thesis.

In wireless communications systems, the information about directions of signal of interests is very important for the subsequent signal processing procedures. Many modulated communication signals exhibit a cyclostationary property, which corresponds to the underlying periodicity arising from carrier frequencies or baud rates. Utilizing this underlying property in the received signals, we propose a new DOA (Direction of Arrival) estimation algorithm for cyclostationary signals. The proposed algorithm is very effective when the desired signals do not share a common cycle frequency. We have also derived the necessary conditions and provided a theoretical analysis on the choice of the parameters for making our proposed algorithm effective in coherent environment.

Next, we come to beam pattern design techniques for antenna arrays. We classify the techniques into two categories: non-iterative beampattern synthesis algorithms and iterative beampattern synthesis algorithms. For non-iterative approaches, we propose an improved beampattern synthesis technique based on the Generalized Sidelobe Canceller (GSC) structure. Constrained optimiza-

tion algorithm is used to control the beampattern from the main beam. We use the cyclostationary property of the transmitting signals to ensure that the blocking matrix spans the interference-plus-noise subspace to get a better control of the beampattern, especially in the interference-limited and time-varying environment and when the interference is close to the look direction. We also consider the case when multipath exists in the desired signal, and put forward an improved structure to solve this problem.

After that we shift to iterative approaches. Firstly, we reveal the relationship between sidelobe level and the number and strength of interferences. Based on this relationship, three iterative beampattern techniques for arbitrary array geometry and arbitrary desired pattern were proposed by Olen [1], Bell [2] and Zhou [3] respectively. However, all of them require the calculation of matrix inverse in each iteration. In order to simplify the algorithms, we make an improvement based on Zhou's algorithm to avoid matrix inverse calculation. Moreover, we consider combining the improved algorithm with Frost beamforming algorithm. Taking advantages of the merits of both techniques, our proposed algorithm has the ability of beam pattern control and adaptive beamforming at the same time. In addition, Householder Transform (HT) is applied to further reduce the computational load by constraining the weight vector update in each iteration in a subspace with reduced dimension. Analysis and comparison on the computational load between our proposed algorithm and several relevant algorithms are provided as well to prove the computational efficiency of the new algorithm.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

Wireless systems are now popular worldwide to help people and machines to communicate with each other irrespectively of their location. So far, using a cellular system is the most common wireless method to access data or to perform voice dialling. But in a near future, we will be surrounded by a number of options to set up a wireless connection over the radio interface. One of the objectives for the fourth generation wireless communication system (4G) is "always best connected", meaning that your wireless equipment should connect to the network or system that at the moment is the "best" for you. Various connection systems range from satellites that provides low bit rates but global coverage and cellular systems with continental coverage to high bit rate local area networks and personal area networks with a maximum range of a few to a hundred meters. If these systems should co-exist, then we would obtain a crowded frequency spectrum, since there are many different "actors" who want

their share of the limited frequency resource. However, due to the limitation of spectrum allocation, the growth of services is expected soon to reach many system's capacity. In addition, several physical limitations such as time-variant and dispersive nature of propagation environments induce many technical challenges for reliable and ubiquitous communications. Moreover, the requirements of small size, light weight, and low power of personal communication devices present additional challenges. Therefore, providing high spectrum efficiency, high quality and robust wireless communications combined with complexity-reduced signal processing algorithms have become prime challenges in university and industrial worldwide research and development centers. The current trend to achieve high spectral efficiency is by utilizing adaptivity in the ever changing radio environment. Adaptivity on the physical layer can be used in all possible dimensions: time, frequency, power and space. Adaptivity can also be used on higher signalling layers to boost performance even further, such as multi-user scheduling.

This thesis is devoted to the physical layer of wireless communication systems and will focus on the algorithms in spatial dimension. As was pointed out in [5], time domain processing techniques have virtually been squeezed down to their last one-tenth of a decibel in terms of performance improvement. Spatial processing is truly the last frontier in terms of the rewards that can be achieved in improving the performance of wireless communications systems. Space utilization is made possible through the use of multiple antennas arranged in an array for transmitting and receiving signals. Or in some cases, a single antenna which has several polarizations or modes is used to obtain polarization or angle diversity. In the mid-1990s, the terms *smart antennas* and *adaptive antennas* were introduced because with signal processing algorithms, the antenna array

can be made adaptive and change its transmission or reception characteristics in a smart manner according to the radio environment.

1.2 Major Contributions

This thesis contains some original contributions on algorithms used in adaptive antenna processor. The main contributions can be classified into two parts– DOA estimation and beam pattern synthesis algorithms. Now we will present a brief summary of the main points of our work for each of these two parts.

DOA Estimation: In wireless communication systems, subscribers are usually spatially separated and the use of antenna arrays makes it possible to locate the direction of arrival of each signal. Therefore, information about the DOA of signals is required and important in most smart antenna techniques where signals are transmitted and received in a directional manner. Various techniques for DOA estimation have been proposed in the past several decades [6–12], among which the most commonly used are subspace-based techniques such as MUltiple SIgnal Classification (MUSIC) [8], Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) [9, 10] and their variations. When the total number of interference and target signals is larger than the number of antenna elements, which is a common situation in current wireless communication environment, these traditional algorithms fail to work. Other limitations with MUSIC algorithm include inability of selective DOA estimation and the spatial characteristics of noise must be known *a priori*, or be transformation invariable, *etc.*. Cyclic MUSIC algorithm [13] was later proposed

to solve some of the above mentioned problems by taking into consideration a property called the *cyclostationarity* which is a common characteristic shared by transmission signals in applications such as radar, sonar or telecommunications. But disadvantages with Cyclic MUSIC exist in situations when the signals of our interests do not share a common cycle frequency. To overcome this problem, we propose a new cyclic MUSIC direction finding algorithm which only requires single operation of direction finding process in such a situation. Furthermore, to make our proposed algorithm compatible with wireless multipath environment, we combine it with spatial smoothing, a preprocessing technique which is very effective in de-correlating the signals. Theoretical analysis and simulation results are provided to show the effectiveness of our proposed algorithm. The result of this work has been submitted to the *IEEE Transaction on Signal Processing*.

Beam Pattern Synthesis: To synthesize array pattern is to find a set of weights such that the array pattern has a desired shape, *e.g.* a maximum at the desired angle with a certain beam-width and a certain sidelobe rolloff. The second objective of this dissertation is to investigate and develop algorithms on how to design weights of array elements to produce the desired pattern, *i.e.* array pattern synthesis technique.

It is commonly believed that there are two types of pattern synthesis algorithms: non-iterative and iterative. For the case of non-iterative beam pattern synthesis algorithms, the array achieve its optimal pattern in one step. Olen, Zhou, and Guo make use of adaptive theory to realize pattern synthesis [1, 3, 14]; Tseng, Ng and Er use quadratic programming [15–17]

as the solution to pattern synthesis problem while Lebert and Wang utilize second-order cone programming [18, 19]. The Generalized sidelobe canceler (GSC) is an important structure for both theoretical and practical reason. As Griffiths and Jim explained in [20], the GSC can be viewed as an alternative implementation and extension of Frost's algorithm [21]. Sim in [4] applied the quadratic programming in the main beam of GSC structure to guarantee a highly directional quiescent pattern with low sidelobe level. However, in determining the blocking matrix, Sim did eigen-decomposition of the covariance matrix of received signal, and chose the blocking matrix as the noise subspace which couldn't reject strong interferences effectively, especially in interference-limited environment. Equipped with the cyclostationary property of transmission signals, we present a beam pattern synthesis algorithm for arbitrary array geometry. Our algorithm determines the blocking matrix in the interference-plus-noise subspace exactly so that deeper nulls can be achieved in the interference directions. Not only can this approach suppress strong interference and raise the output SNR, it can also keep the sidelobe pattern in control, and is very helpful when the interference is close to the look direction of the array. Furthermore, we also make an improvement to enable it to work in wireless multipath environment. The result of this work has been presented at the 3^{rd}*IEEE International Symposium on Wireless Communications Systems, Sept. 2006.*

Compared with non-iterative algorithms, an adaptive array with iterative beam pattern synthesis ability is a more complex system which is able to adapt itself in a time-varying signal environment. However, sidelobe shapes in an adaptive beamformer are usually left uncontrolled and can

become undesirably high. In a receiving array, when an interference signal suddenly switches to the direction of a high sidelobe while the array is in operation, it would interrupt the normal system operation because it takes time for adaptive array to complete the interference nulling process. In a transmission array, high sidelobes cause emissions in unwanted directions that may interference with other communication systems and waste the transmission power. In this thesis, we propose an iterative algorithm, which can achieve beam pattern control and adaptive beamforming at the same time. Our proposed algorithm is computationally efficient because it avoids the calculation of matrix inverse which is often required by conventional beam pattern synthesis algorithms. Besides, the new algorithm is superior in being able to shape the sidelobe region which is left uncontrolled by other adaptive beamforming algorithms. In addition, Householder Transform (HT) is applied to further reduce the computational load. Analysis and comparison on the computational costs of our proposed algorithm along with several relevant algorithms are provided to prove the computational efficiency of the new algorithm. Numerical results show that our proposed algorithm has the ability of nulling out interferences and maintaining a well-shaped beam pattern simultaneously which guarantees an overall best performance among several other algorithms. This work has been presented at 64^{th} *IEEE Vehicular Technology Conference, Sept. 2006* and has been submitted to *IEEE Transaction on Antennas and Propagations*.

1.3 Organization of the Thesis

This thesis is divided into seven chapters, and its organization is as follows.

Chapter 2 overviews fundamentals of adaptive arrays. First, some basic concepts, terminologies and classification of adaptive array are presented. Then the assumptions on array model and transmission environment are described. This assumptions and array model will be used to develop the algorithms throughout the following chapters. Criteria to optimize performance of adaptive arrays and adaptive algorithms to obtain the array optimal weight vector are also summarized here.

Chapter 3 presents the classification of DOA estimation algorithms and briefly reviews some of the well-known algorithms in each category.

Chapter 4 discusses the direction-of-arrival estimation algorithm of cyclostationary coherent signals in wireless communications. We propose a new cyclic MUSIC direction finding algorithm which only requires single operation of direction finding process when the desired signals do not share a common cycle frequency.

In chapter 5, we introduce two non-iterative beam pattern synthesis techniques by forming a quadratically constrained minimization problem. Then the relationship between this two algorithms is explained. Finally, we employ method I in GSC structure to shape the quiescent beam pattern, and use an effective approach to construct the blocking matrix to make sure that nulls can be achieved at the directions of strong interferences and multipath signals.

In chapter 6, we firstly review the Olen, Bell and Zhou's iterative beam pattern shaping algorithms. Then we reformulate Zhou's algorithm to avoid

the computation of matrix inverse in each iteration, and put forward an algorithm which incorporates the reformulated algorithm with Frost beamforming algorithm. The new algorithm can achieve beam pattern control and adaptive beamforming at the same time. Householder transform is applied to further reduce the computational load of our proposed technique.

Chapter 7 concludes this thesis. Some recommendations on future work are discussed in this chapter.

Chapter 2

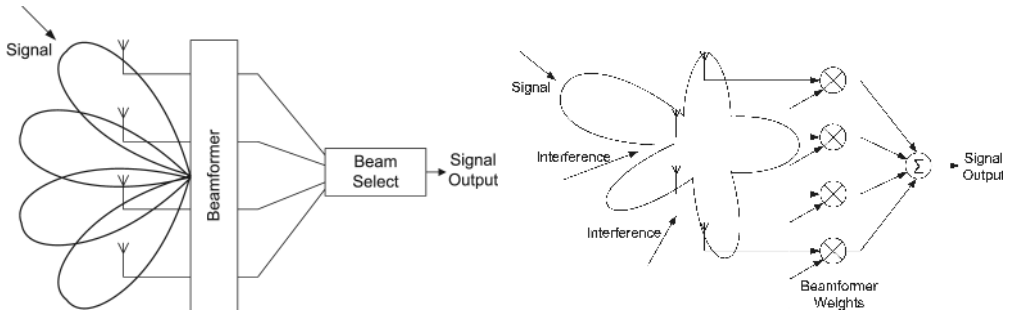
Fundamentals of Smart Antennas

This chapter presents the principle concepts of adaptive arrays. In particular, this chapter covers array signal model, different types of adaptive beamforming, criteria to optimize performance and adaptive signal processing algorithms for adaptive arrays. The benefits of using adaptive arrays in wireless mobile communications are also discussed.

2.1 Overview of Smart Antennas

2.1.1 Two Types of Smart Antennas

A smart antenna [6] is a multi-element antenna array where the signals received at each antenna element are intelligently combined to improve the performance of the wireless system.



(a) Switched multibeam antenna (b) Adaptive array

Figure 2.1: Two basic types of smart antennas

There are two basic types of smart antennas, as shown in Figure 2.1. The first type is the *switched beam antenna array*, which forms multiple fixed beams with heightened sensitivity in particular directions. These antenna systems detect signal strength, choose from one of several predetermined, fixed beams, and switch from one beam to another as the mobile moves in the sector. Instead of shaping the directional antenna pattern with the metallic properties and physical design of a single element (like a sectorized antenna), switched beam systems combine the outputs of multiple antennas in such a way as to form finely sectorized (directional) beams with more spatial selectivity than that can be achieved with conventional, single-element approaches. The second type is defined here as an *adaptive array* as shown in Figure 2.1(b) in which the signals from several antenna elements (not necessarily a linear array), each with similar antenna patterns, are weighted (both in amplitude and phase) and combined to maximize the performance of the output signal. Note that the adaptive array will form a narrow beam in a line-of-sight environment without multipath, but can also optimally suppress interference and provide fading mitigation and gain in a multipath environment. The switched beam antenna array is less complex because it uses simple beam tracking. That is, the beam-selection technique only

need to look at the signal level in each beam every few seconds to determine which beam to use. Similarly, in the case of linear array implementation of the switched beam antenna array, the phase shifts only need to be slowly adjusted to track the change in direction-of-arrival of the received signal. On the other hand, the beamformer weights in the adaptive array need to track the fading of the desired signal. For example, at 2 GHz with 100 km/hr vehicle speeds, the Doppler is about 200 Hz and the complex weights need to be calculated at least 100 times faster for accurate tracking, *i.e.*, the complex weights need to be calculated at a 20 kHz rate. However, although the adaptive-array processing is much more computationally complex, the requirement is well within the capability of current signal processing ICs.

2.1.2 Benefits of Using Smart Antennas

Using multiple antennas in a receiver can reduce the effect of cochannel interference, multipath fading, and background noise. For these reasons, the use of antenna arrays in wireless communications has received a great deal of attention in the literature (see [5–7, 22] and references therein). Using an array of elements to improve a wireless connection is an old technique, even used by Marconi in 1901 to increase the gain of the Atlantic transmissions of Morse codes [23]. Today, antenna arrays in wireless communication systems are used to improve performance in several ways, not just for range improvement. Systems with array antennas in commercial operation have been reported for GSM networks [24], fixed broadband wireless access networks [25] and 3G CDMA networks [26]. A number of field trials with testbed antenna arrays have also been reported, see [27, 28].

This trend in wireless communications has been possible mainly through the advents of signal processing, digital signal processors and high speed analog-digital convertor. The main advantages today, for using multiple antennas when transmitting over a wireless link, are:

Improved signal quality: Due to the use of multiple antennas, adaptive arrays can provide additional antenna gain, which depends on the number of array elements. This, consequently, leads to an improved output SINR. Define the input SNR as $\mathrm{SNR}_{\mathrm{in}}$ and M as the number of array elements. If the number of interferences is smaller than the degree of freedom in the array, the output SINR in a single propagation environment (without multipath fading) can be found as

$$\mathrm{SINR}_{\mathrm{out}} = M \cdot \mathrm{SNR}_{\mathrm{in}}, \tag{2.1}$$

or

$$\mathrm{SINR}_{\mathrm{out}}[\mathrm{dB}] = 10\log_{10} M + \mathrm{SNR}_{\mathrm{in}}[\mathrm{dB}]. \tag{2.2}$$

In multipath fading environment, if signal processing technique is used in both spatial and temporal domains such as the case of the broadband beamformer, more diversity gain could be achieved depending on the number of taps in the employed tap-delay-lines and fading characteristics. Take a simple case of 2-path model as an example. When the two paths are spatially uncorrelated, for example, the preceding and delayed signals coming from 0° and 30°, respectively, the output SINR is estimated as

$$\mathrm{SINR}_{\mathrm{out}}[\mathrm{dB}] = 10\log_{10} M + 10\log_{10} 2 + \mathrm{SNR}_{\mathrm{in}}[\mathrm{dB}]. \tag{2.3}$$

This means that additional 3dB diversity gain has been obtained in 2-path multipath fading environment. The richer the multipath fading environment is, the more diversity gain can be achieved. Figure 2.2 plots the output SNR versus the number of employed array elements.

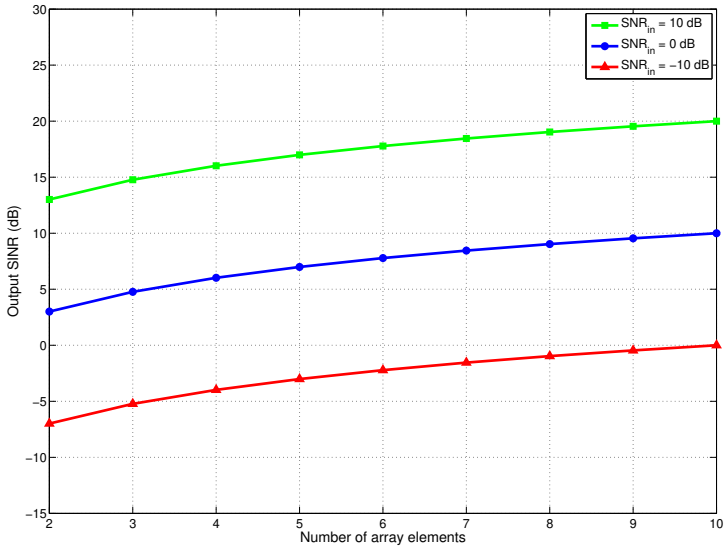


Figure 2.2: Output SNR versus number of array elements

Extended coverage: From (2.2) it is clear the array gain that can be achieved by an adaptive array is

$$G = 10\log_{10} M. \tag{2.4}$$

This additional gain allows to extend the coverage of the basestation. When the angular spread is small and the path loss is modeled with exponent α, the range extension factor (REF) is given by [29]

$$\text{REF} = \frac{r_{\text{array}}}{r_{\text{conv}}} = M^{\frac{1}{\alpha}}, \tag{2.5}$$

where r_{conv} and r_{array} denotes the range covered by the conventional an-

tenna (with single element) and the array antenna (with multiple elements), respectively. The extended area coverage factor (ECF) is [29]

$$\mathrm{ECF} = \left(\frac{r_{\mathrm{array}}}{r_{\mathrm{conv}}}\right)^2 = \mathrm{REF}^2. \tag{2.6}$$

Figure 2.3 shows that with an 6 element array, the coverage area is almost

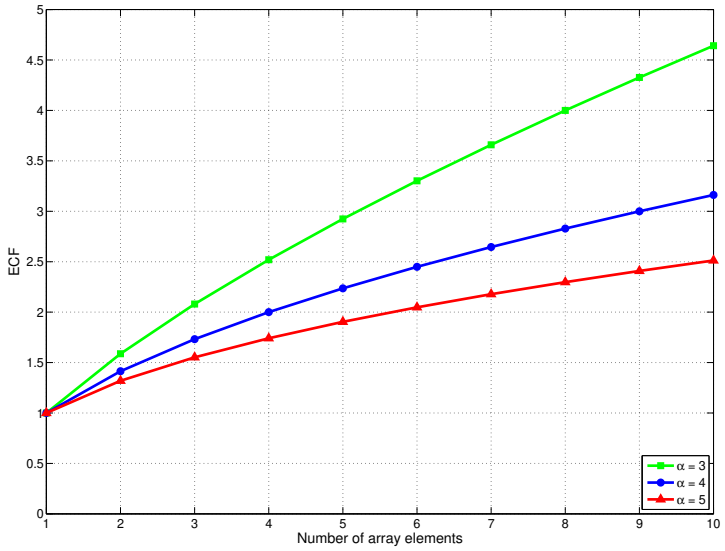


Figure 2.3: Improvement of area coverage by adaptive arrays

doubled compared with single antenna case for $\alpha = 5$. Since the inverse of the ECF represents the reduction factor in number of basestation required to cover the same area using a single antenna, it is clear that using adaptive arrays can significantly reduce the number of base stations. For example, for the above mentioned case with $\alpha = 5$, the number of base stations can be reduced to only one half of the original number. This is useful in remote areas with low population. A larger area can thus be served with less base stations. Alternatively, the transmit power of the mobile units can be reduced due to the increased gain, or sensitivity, of the receiving

base station antenna array.

Reduced transmission power: We have seen in (2.1) that the use of adaptive arrays can provide a large array gain. This gain consequently leads to the reduction in required transmission power in base station. If the required reception sensitivity is kept the same, then the power requirement of a base station employed an M-element array is reduced to M^{-1} and correspondingly the required output power of the base station power amplifier can be reduced to M^{-2}. The reduction in the transmission power is beneficial to user's health and will reduce implementation cost since high frequency power amplifiers are often very expensive.

Interference suppression: By using the spatial dimension provided by multiple antenna elements, it is possible to suppress interfering signals in a way that is not possible with a single antenna. Hence the system can be tuned to be less susceptible to interference and the distance between base stations using the same time/frequency channel can be reduced, which is beneficial in densely populated area. This will definitely lead to system capacity improvement.

Spatial diversity: Multiple antennas can also be used to counteract the channel fading due to multipath propagation. Sufficiently spaced multiple antennas at the receiver give copies of the transmitted signal that has propagated through channels with different fading profiles. The probability that all signal copies are in a deep fading simultaneously is small. Thus, spatial diversity increases the robustness of the wireless link and this can be utilized to obtain a higher data throughput to decrease the transmission power. A link capacity improvement is thus obtained.

Transmitter localization: A receiver array antennas can be used to localize the transmitter, just as we can use our both ears to localize the source of a sound in a room without using our eyes. This has application in positioning services and emergency call localization.

2.1.3 Architecture of Smart Antennas

A typical smart antenna architecture for a base station can be divided into the following functional blocks:

Radio Unit: This unit mainly consists of antenna arrays, down-conversion chains and analog-to-digit convertors.

Beamforming Unit: In beamforming unit, the signals received by antenna array are multiplied by a set of weights to form a beam towards desired direction. By choosing different sets of weights, it is possible to steer the beam pattern towards arbitrary directions. Moreover, with appropriate selected weighting vector, smart antennas can achieve beam steering, adaptive nulling and beam shaping at the same time.

Adaptive Antenna Processor: The function of the adaptive processor unit is to determine the complex weight vector to be used in beamforming unit. The weight vector can be optimized according to different criteria which will be described in detail in section 2.4. In general, this processor can be divided into the following computation processes:

- **Sampling:** Snapshot is taken of the training signals coming from all of the antenna elements.

- **DOA estimation:** The number of incoming wave fronts and their DOAs are estimated.
- **DOA classification:** First, the spatially resolved wave fronts with different DOAs are extracted from the input data. Then, user identification decides whether a DOA belongs to a user or to an interference signal.
- **The optimum weight calculation:** The processor calculates the optimum weights to maximize the SINR for each user. A beamforming algorithm forms an antenna pattern with a main beam steered into the direction of the user, while minimizing the influence of the interference wave fronts.
- **Tracking:** The user DOAs are tracked to increase the reliability of the DOA estimates.

2.2 Array Signal Model

An adaptive array is a system consisting of an array of antenna elements and a real-time adaptive processor which controls the beamforming network to automatically adjust its control weights toward optimization of a certain criterion in accordance with a selected algorithm [30–33]. Sometimes adaptive arrays are also referred to as adaptive antennas or smart antennas. A typical configuration of an adaptive array is illustrated in Figure 2.4.

Antenna elements can be arranged in various geometry configurations of which the most popular are linear, circular and planar (see Figure 2.5). A linear array consists of array elements whose centers are aligned along a straight

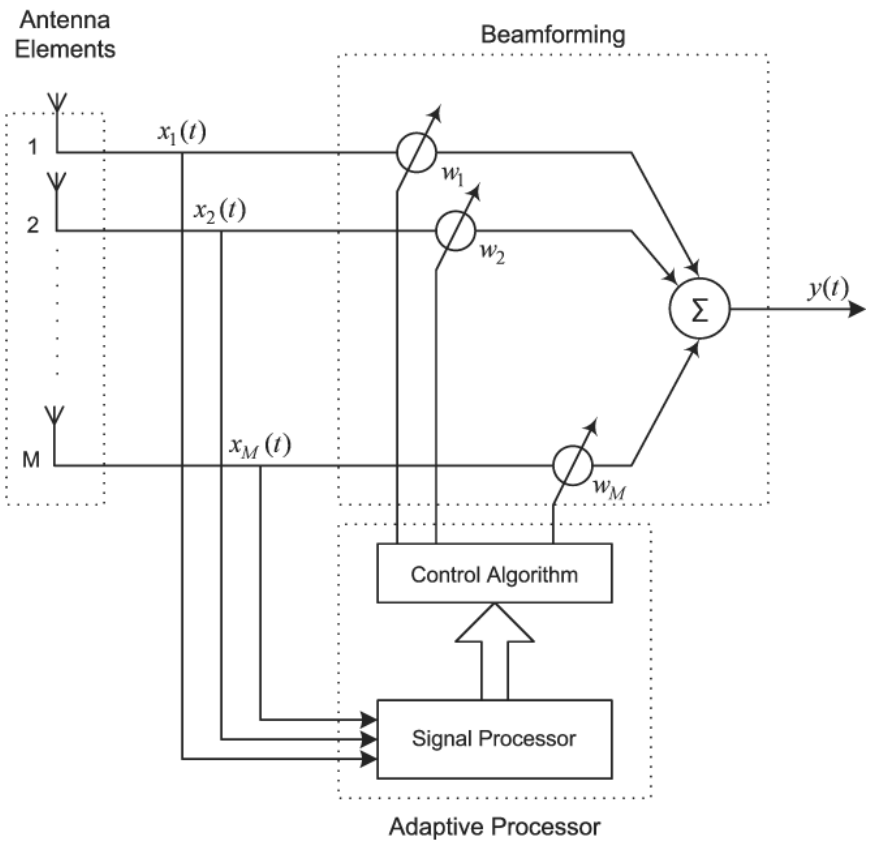
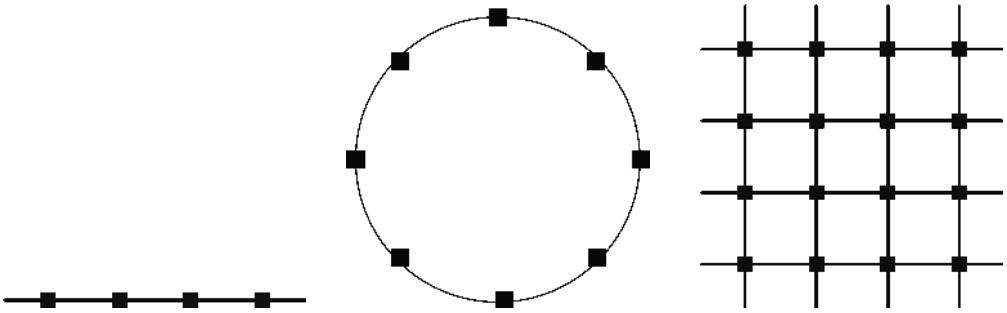


Figure 2.4: An adaptive array with M elements

line. If the spacing between consecutive array elements is equal, it is called a *uniformly-spaced linear array* (ULA). Similarly, a circular array contains array elements whose centers lie on a circle. Finally, a planar array consists of array elements whose centers are placed on a single plane. While both linear and circular arrays can only perform one-dimensional beamforming (horizontal plane), planar arrays can be used for two-dimension (2-D) beamforming (both in vertical and horizontal planes).

Although the geometry configurations are different, the principle of adaptive arrays is the same, and in order to simplify the analysis and synthesis of arrays, the uniformly spaced linear arrays are often considered. The principles and mathematics can be then extended to other array geometries [34]. Throughout



(a) Uniformly spaced linear array (b) Circular array (c) Planar array

Figure 2.5: Different geometry configurations of adaptive arrays

this thesis, therefore, we shall restrict our study to the uniformly spaced linear arrays.

As most of the modern approaches in signal processing are model based, which means they rely on certain assumptions made on the observation data, we describe first the prevailing data model used in this thesis first.

The following scenario is assumed and maintained throughout this thesis:

Isotropic and Linear Transmission Medium: The K transmission signals and an M elements antenna array are in the same plane so that a 1-dimension angle and 2-dimension coordinate pair is sufficient to describe the direction of arrival of the signals and the positions of the sensors respectively. Besides, the transmission medium is assumed to be linear such that the signals received at one antenna element can be modeled as a superposition of K wavefronts. The gain of each sensor element is assume to be identical and equal to 1.

Farfield Assumption: The K signals are located far from the array such that their direction of propagation is equal at each sensor and the wavefronts are planar. Thus, the propagating field within the array aperture consists

of plane waves. This assumption can in general be valid if the dimension of the antenna array is neglectable compared with the distance from the signal sources to the array.

Narrowband Assumption: The K emitted signals have the same carrier frequency f_c, which means their frequency content is concentrated in the vicinity of carrier frequency f_c. Then, the real component of transmission signals can be expressed as

$$s_i(t) = \alpha_i(t)\cos[2\pi f_c t + \beta_i(t)], \qquad \text{for } i = 1, \ldots, K. \tag{2.7}$$

The signals are called *narrowband* if their amplitudes $\alpha_i(t)$ and phases $\beta_i(t)$ vary slowly with respect to the propagation time across the array τ, *i.e.*,

$$\alpha_i(t-\tau) \approx \alpha_i(t) \qquad \text{and} \qquad \beta_i(t-\tau) \approx \beta_i(t).$$

In other words, the narrowband assumption allows the time delay of the signals across the array to be modeled as simply a phase shift of the carrier frequency.

AWGN Channel: The noise is modeled as a complex white gaussian process. The additive noise is taken from a zero mean, spatially uncorrelated random process which is uncorrelated with the signals. The noise have a common variance σ_n^2 at all sensors and is uncorrelated among all sensors.

Under the above assumptions, we now begin to consider a uniformly spaced linear array with M elements as illustrated in Figure 2.6, where d is the distance between adjacent elements.

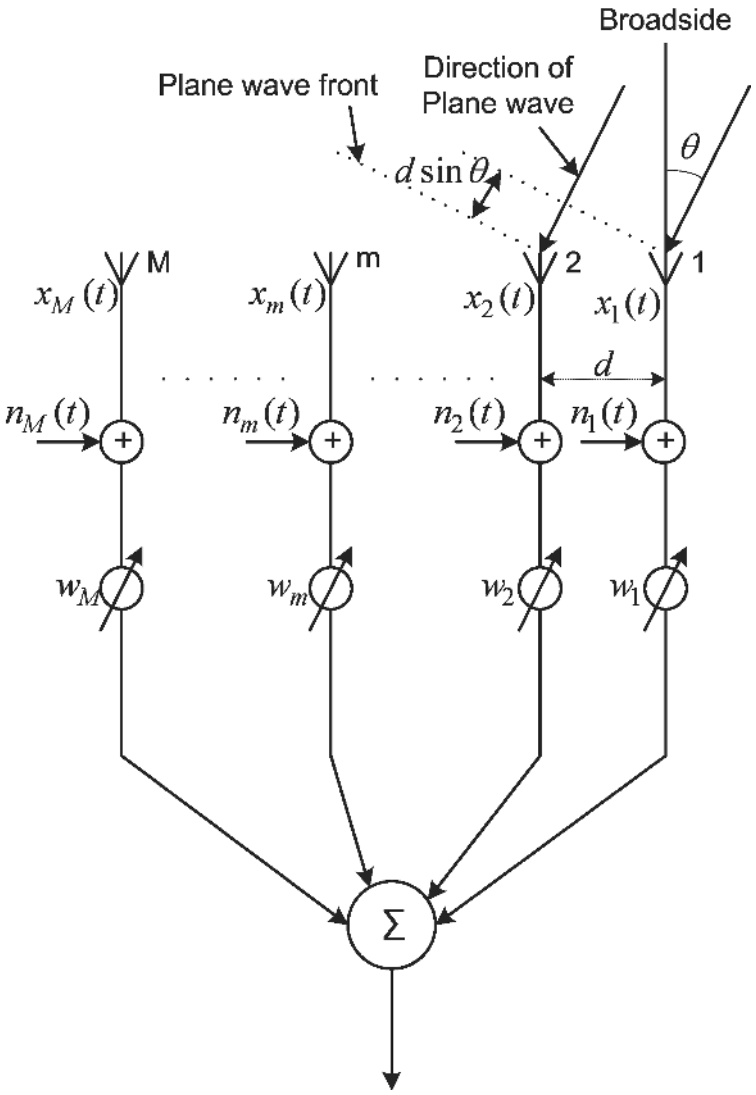


Figure 2.6: An adaptive array with M elements

Assume that a plane wave arrives at the array from a direction θ off the array broadside. The angle θ, measured clockwise from the array broadside, is called the *direction of arrival* (DOA) or *angle of arrival* (AOA) of the received signal. The wavefront at the $(m+1)$th element is later than that at the mth element a differential distance of $d\sin\theta$. Let us take the first element as the reference element and let the signal at the reference element be $s(t)$, then the phase delay of the signal at element m relative to element 1 is $(m-1)kd\sin\theta$,

where $k = \frac{2\pi}{\lambda}$ is the wave number and λ is the wavelength. Consequently, the received signal at the mth element $x_m(t)$ is given as

$$x_m(t) = s(t)\mathrm{e}^{-j\frac{2\pi}{\lambda}(m-1)d\sin\theta}, \tag{2.8}$$

where $j = \sqrt{-1}$ is the imaginary unit and $m = 1, 2, \ldots, M$.

Now let us arrange $x_m(t)$ in a vector form as

$$\mathbf{x}(t) = [x_1(t)\ x_2(t)\ \ldots\ x_M(t)]^T, \tag{2.9}$$

and let

$$\mathbf{a}(\theta) = [1\ \mathrm{e}^{-j\frac{2\pi}{\lambda}d\sin\theta}\ \ldots\ \mathrm{e}^{-j\frac{2\pi}{\lambda}(M-1)d\sin\theta}]^T, \tag{2.10}$$

where $[\cdot]^T$ denotes the vector/matrix transpose operation. Then (2.9) can be expressed by

$$\mathbf{x}(t) = s(t)\mathbf{a}(\theta). \tag{2.11}$$

The vector $\mathbf{x}(t)$ is called the array *input data vector* and $\mathbf{a}(\theta)$ is referred to as the *array response vector* or *steering vector*. The array response vector in this case depends only on the angle of arrival. In general, it may also depend on individual element response, the array geometry, and signal frequency. The set of array response vectors over all directions and frequencies is known as the *array manifold*. For simple arrays such as uniformly spaced linear array considered here, the array manifold can be analytically computed. In practice, however, it is measured as point source responses over various directions and frequencies and this process of obtaining the array manifold is referred to as *array calibration*.

Since there is one-to-one relationship between the spatial frequency $-\pi \leq \mu_i \leq \pi$ and the range of possible DOAs, the maximum range of DOAs is restricted to the interval $-90^\circ \leq \theta_i \leq 90^\circ$. This in turn requires that the sensor spacing satisfy $d \leq \lambda/2$. If the sensor spacing does not satisfy this relationship, the array will be subject to grating lobes and there will be an ambiguity in determining the DOAs. Grating lobes in beam pattern refer to lobes other than main lobe, and they will amplify signals from undesired directions. This is called *spatial aliasing* technically. The above requirement for sensor spacing is analogous to the Nyquist sampling rate for frequency domain analysis.

Now taking local noise effect into consideration, the input data vector becomes

$$\mathbf{x}(t) = s(t)\mathbf{a}(\theta) + \mathbf{n}(t), \tag{2.12}$$

where the noise vector $\mathbf{n}(t)$ is defined as

$$\mathbf{n}(t) = [n_1(t)\ n_2(t)\ \ldots\ n_M(t)]^T. \tag{2.13}$$

It should be noted that (2.8) holds for signals with bandwidth much smaller than the reciprocal of the propagation time across the array. Any signal which satisfies this condition is referred to as the *narrowband*, otherwise it is called *wideband*.

2.3 Adaptive Beamforming

Beamforming is one type of signal processing used to form the array beams toward the desired signal sources while simultaneously create nulls toward in-

terferences. This process of separating desired user from the interferences based on their spatial characteristics is called *spatial filtering*. In the reverse or uplink (from mobile to basestation), the objective of beamforming is to maximize the signal to interference plus noise ratio (SINR) of the received desired signal. Similarly, beamforming is utilized in the forward or downlink (from basestation to mobile) to maximize the transmission power of basestation to a desired mobile, thereby maximizing SINR of the downlink. When beamforming is controlled using adaptive signal processing, it is called *adaptive beamforming*. In some cases, it is desired to steer not only the beams but also array nulls toward a specific location to suppress interferences.

A beamformer is a processor used in conjunction with an array to perform versatile form of spatial filtering [35]. Broadly speaking, there are two types of beamformers, namely, narrowband beamformer and broadband beamformer.

A narrowband beamformer samples input signals in spatial domain and is typically used to process narrowband signals. The configuration of a narrowband beamformer is depicted in Figure 2.7. The output of the narrowband

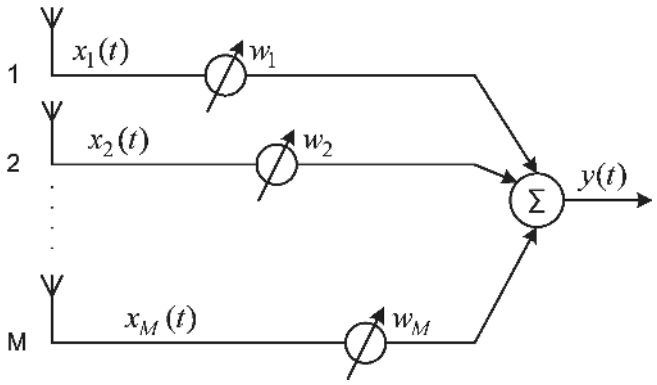


Figure 2.7: Configuration of an adaptive narrowband beamformer

beamformer is a weighted linear combination of received signals at each array

element, and can be given by

$$y(t) = \mathbf{w}^H \mathbf{x}(t), \tag{2.14}$$

where $\mathbf{x}(t)$ is input data vector, $(\cdot)^H$ represents Hermitian (complex conjugate transpose) operation of a vector/matrix, and the complex weight vector $\mathbf{w}$ is defined as

$$\mathbf{w} = [w_1 \; w_2 \; \ldots \; w_M]^T. \tag{2.15}$$

Although our thesis focuses on algorithms dealing with narrowband signals, we take some time to give a brief introduction on broadband beamformer as it is of increasing importance in modern wireless communications. Different from the narrowband beamformer, a broadband beamformer samples input signals in both spatial and temporal domains and is employed to process broadband signals. A broadband beamformer is also called a spatio-temporal processor or spatio-temporal equalizer. The structure of a broadband beamformer often contains in each array element a *tapped delayed line* (TDL) which is also called transversal filters. If the tap spacing is sufficiently long and the number of taps is large, the TDL approximates an ideal filter that allows exact control of gain and phase at each frequency within the band of interest [36]. The TDL is not only useful for providing desired adjustment of gain and phase over the frequency band of interest for wideband signals but also suited for other purposes such as mitigation of multipath fading and compensation for effects of finite array propagation delay and interchannel mismatch [36]. A typical broadband beamformer using TDLs is shown in Figure 2.8.

To model the broadband beamformer, we arrange the signals and complex

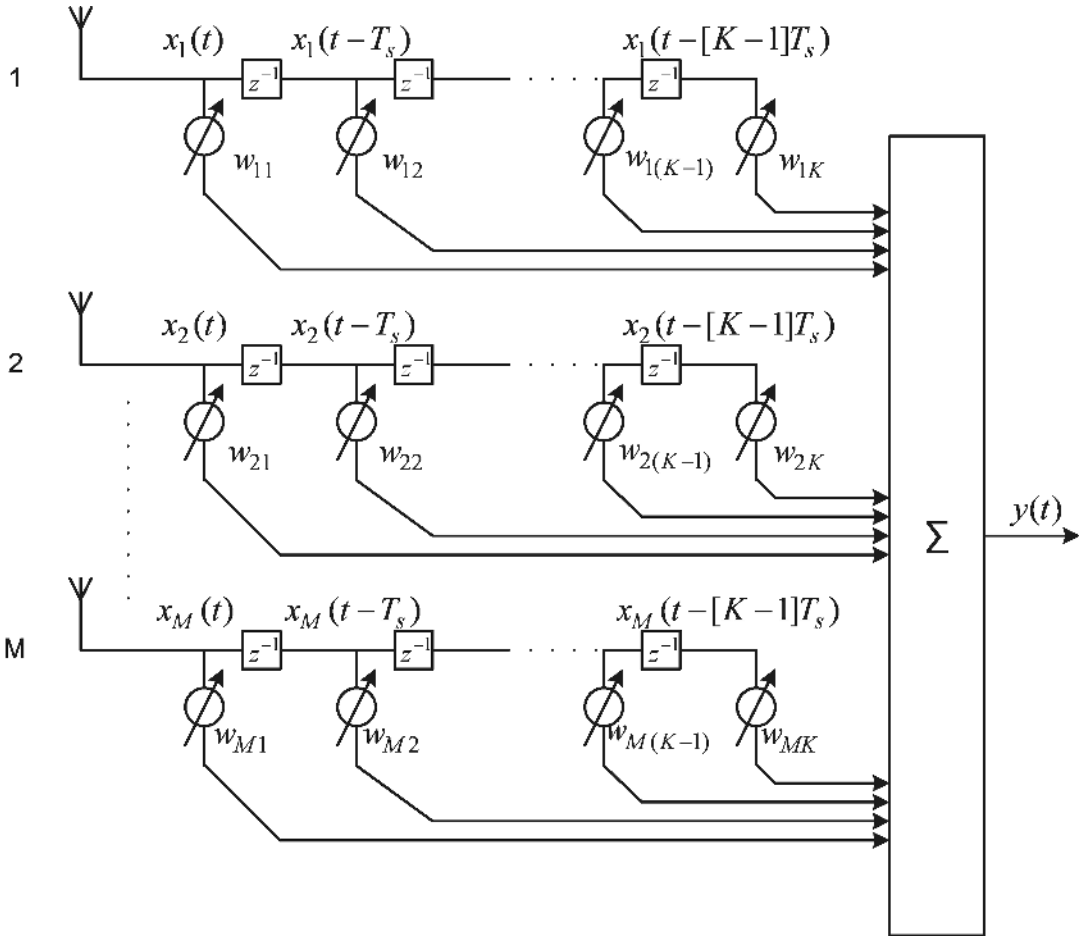


Figure 2.8: Configuration of an adaptive broadband beamformer using tapped delay lines

weights at the K TDL taps of antenna m as

$$\mathbf{x}_m(t) = [x_m(t)\ x_m(t-T_s)\ \ldots\ x_m(t-[K-1]T_s)]^T, \tag{2.16}$$

and

$$\mathbf{w}_m = [w_{m1}\ x_{m2}\ \ldots\ w_{mK}]^T, \tag{2.17}$$

and define

$$\mathbf{x}(t) = [\mathbf{x}_1(t)^T\ \mathbf{x}_2(t)^T\ \ldots\ \mathbf{x}_M(t)^T]^T, \tag{2.18}$$

and

$$\mathbf{w} = [\mathbf{w}_1^T \ \mathbf{w}_2^T \ \ldots \ \mathbf{w}_M^T]^T. \tag{2.19}$$

Now the output of broadband beamformer can be expressed in exactly the same form as that of narrowband beamformer in (2.14), that is,

$$y(t) = \mathbf{w}^H \mathbf{x}(t). \tag{2.20}$$

2.4 Criteria for Performance Optimization

As we have mentioned earlier in this chapter, the adaptive processor controls the beamforming network to optimize the beamforming weights according to a certain criterion. The four most common criteria which are often employed to obtain optimum weights for adaptive arrays in mobile communications are Minimum Mean Square Error (MMSE), Maximum Signal to Interference plus Noise ratio (MSINR), Minimum Variance (MV) and Maximum Likelihood (ML). These optimum criteria will be reviewed below.

2.4.1 Minimum Mean Square Error

The MMSE criterion is first considered by Widrow *et al.* in [30]. The criterion strives to minimize the error between the array output signal $y(t)$ and the desired signal $s(t)$. In practice, the desired signal $s(t)$ is of course not known. However, by using some techniques such as the use of training signal or estimation based on the desired signal characteristics, one can generate a reference signal $r(t)$ that closely approximates the desired signal to a certain extent. Consider the

input signal vector given by

$$\mathbf{x}(t) = s(t)\mathbf{a}(\theta) + \mathbf{n}(t), \tag{2.21}$$

where $\mathbf{a}(\theta)$ is the array response and $\mathbf{n}(t)$ is a vector containing zero mean noise and uncorrelated interferences. For a narrowband adaptive array, the output signal is calculated according to (2.14) as

$$y(t) = \mathbf{w}^H\mathbf{x}(t). \tag{2.22}$$

The error signal is defined as

$$\begin{aligned} \epsilon(t) &= r(t) - y(t) \\ &= r(t) - \mathbf{w}^H\mathbf{x}(t), \end{aligned} \tag{2.23}$$

and the weights are chosen to minimize the mean square error (MSE) of the error signal

$$\mathcal{E}\left\{|\epsilon(t)|^2\right\} = \mathcal{E}\left\{|r(t) - \mathbf{w}^H\mathbf{x}(t)|^2\right\}, \tag{2.24}$$

where $\mathcal{E}\{\cdot\}$ denotes the expectation operation. Expanding (2.24) we can have

$$\begin{aligned} \mathcal{E}\left\{|\epsilon(t)|^2\right\} &= \mathcal{E}\left\{|r(t)|^2\right\} - \mathbf{w}^T\mathcal{E}\left\{\mathbf{x}^*(t)r(t)\right\} - \mathbf{w}^H\mathcal{E}\left\{\mathbf{x}(t)r^*(t)\right\} + \mathbf{w}^H\mathcal{E}\left\{\mathbf{x}(t)\mathbf{x}^H(t)\right\}\mathbf{w} \\ &= \mathcal{E}\left\{|r(t)|^2\right\} - \mathbf{w}^T\mathbf{r}_{xr}^* - \mathbf{w}^H\mathbf{r}_{xr} + \mathbf{w}^H\mathbf{R}_{xx}\mathbf{w}, \end{aligned} \tag{2.25}$$

where $\mathbf{r}_{xr} = \mathcal{E}\left\{\mathbf{x}(t)r^*(t)\right\}$ and $\mathbf{R}_{xx} = \mathcal{E}\left\{\mathbf{x}(t)\mathbf{x}^H(t)\right\}$ are called the *correlation vector* and the *covariance matrix*, respectively. Here $(\cdot)^*$ denotes the complex conjugate. The optimum weight vector can be found by setting the gradient of

(2.25) with respect to $\mathbf{w}$ equal to zero [5], *i.e.*

$$\nabla_{\mathbf{w}}\mathcal{E}\left\{|\epsilon(t)|^2\right\} = -2\mathbf{r}_{xr} + 2\mathbf{R}_{xx} = 0, \tag{2.26}$$

which gives the solution

$$\mathbf{w}_{\text{MMSE}} = \mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xr}. \tag{2.27}$$

Equation (2.27) is often referred to as the Wiener-Hopf equation or the optimum Wiener solution [5]. By substituting (2.27) into (2.25), we have the optimum MMSE as

$$\text{MMSE} = \mathcal{E}\left\{|\epsilon(t)|^2\right\} = \mathcal{E}\left\{|r(t)|^2\right\} - \mathbf{r}_{xr}^H\mathbf{R}_{xx}^{-1}\mathbf{r}_{xr}. \tag{2.28}$$

2.4.2 Maximum Signal to Interference plus Noise Ration (MSINR)

The criterion considered in this subsection is the maximum SINR. With (2.14) and (2.21), the output of the array can be expressed as

$$\begin{aligned} y(t) &= \mathbf{w}^H\mathbf{x}(t) = \mathbf{w}^H\mathbf{s}(t) + \mathbf{w}^H\mathbf{n}(t) \\ &= y_s(t) + y_n(t). \end{aligned} \tag{2.29}$$

The average output SINR is given by

$$\text{SINR} = \mathcal{E}\left\{\frac{|y_s(t)|^2}{|y_n(t)|^2}\right\} = \mathcal{E}\left\{\frac{\mathbf{w}^H\mathbf{s}(t)\mathbf{s}^H(t)\mathbf{w}}{\mathbf{w}^H\mathbf{n}(t)\mathbf{n}^H(t)\mathbf{w}}\right\} = \frac{\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w}}{\mathbf{w}^H\mathbf{R}_{nn}\mathbf{w}}, \tag{2.30}$$

where $\mathbf{R}_{ss} = \mathcal{E}\left\{\mathbf{s}(t)\mathbf{s}^H(t)\right\}$ and $\mathbf{R}_{nn} = \mathcal{E}\left\{\mathbf{n}(t)\mathbf{n}^H(t)\right\}$. Taking the gradient of (2.30) with respect to $\mathbf{w}$, we can get

$$\begin{aligned}\nabla_{\mathbf{w}}\text{SINR} &= \frac{\nabla_{\mathbf{w}}(\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w})(\mathbf{w}^H\mathbf{R}_{nn}\mathbf{w}) - (\mathbf{w}^H\mathbf{R}_{nn}\mathbf{w})\nabla_{\mathbf{w}}(\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w})}{(\mathbf{w}^H\mathbf{R}_{uu}\mathbf{w})^2} \\ &= \frac{2\mathbf{R}_{ss}\mathbf{w}(\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w}) - 2\mathbf{R}_{nn}\mathbf{w}(\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w})}{(\mathbf{w}^H\mathbf{R}_{nn}\mathbf{w})^2}.\end{aligned} \tag{2.31}$$

The optimum weight $\mathbf{w}_{\text{opt}}$ can be found by setting $\nabla_{\mathbf{w}}\text{SINR} = 0$, which leads to

$$\mathbf{R}_{ss}\mathbf{w} = \frac{\mathbf{w}^H\mathbf{R}_{ss}\mathbf{w}}{\mathbf{w}^H\mathbf{R}_{nn}\mathbf{w}}\mathbf{R}_{nn}\mathbf{w} = \text{SINR}\cdot\mathbf{R}_{nn}\mathbf{w}. \tag{2.32}$$

If $\mathbf{R}_{nn}$ is invertible, then (2.32) can be written as

$$\mathbf{R}_{nn}^{-1}\mathbf{R}_{ss}\mathbf{w} = \text{SINR}\cdot\mathbf{w}, \tag{2.33}$$

which is the generalized eigenvalue problem. Note that the value on the right hand side of (2.33) is bounded by the maximum and minimum eigenvalues of $\mathbf{R}_{nn}^{-1}\mathbf{R}_{ss}$. The maximum eigenvalue $\lambda_{\max}$ satisfies the following condition,

$$\mathbf{R}_{nn}^{-1}\mathbf{R}_{ss}\mathbf{w} = \lambda_{\max}\mathbf{w}. \tag{2.34}$$

Comparing (2.33) with (2.34), it is clear that $\lambda_{\max}$ is the optimum value of SINR. The eigenvector associated with $\lambda_{\max}$ is $\mathbf{w}_{\text{opt}}$, and can be written as

$$\begin{aligned}\mathbf{w}_{\text{opt}} &= \frac{\mathbf{R}_{nn}^{-1}\mathbf{R}_{ss}\mathbf{w}_{\text{opt}}}{\text{SINR}} = \frac{\mathbf{R}_{nn}^{-1}\mathcal{E}\left\{s(t)\mathbf{a}(\theta)s^*(t)\mathbf{a}^H(\theta)\right\}\mathbf{w}_{\text{opt}}}{\text{SINR}} \\ &= \frac{\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)\mathbf{a}^H(\theta)\mathbf{w}_{\text{opt}}\mathcal{E}\left\{|s(t)|^2\right\}}{\text{SINR}}.\end{aligned} \tag{2.35}$$

Define

$$\beta = \frac{\mathbf{a}^H(\theta)\mathbf{w}_{\text{opt}}\mathcal{E}\left\{|s(t)|^2\right\}}{\text{SINR}}, \tag{2.36}$$

then the optimum weight vector can be expressed in a similar form of the Wiener-Hopf equation as

$$\mathbf{w}_{\text{SINR}} = \mathbf{w}_{\text{opt}} = \beta \mathbf{R}_{nn}^{-1} \mathbf{a}(\theta). \tag{2.37}$$

2.4.3 Maximum Likelihood (ML)

We rewrite the input signal vector in (2.21),

$$\begin{aligned} \mathbf{x}(t) &= s(t)\mathbf{a}(\theta) + \mathbf{n}(t) \\ &= \mathbf{s}(t) + \mathbf{n}(t), \end{aligned} \tag{2.38}$$

and define the probability density function for $\mathbf{s}(t)$ given $\mathbf{x}(t)$ as $p_{\mathbf{x}(t)|\mathbf{s}(t)}\{\mathbf{x}(t)\}$. Given $\mathbf{x}(t)$, we wish to maximize $p_{\mathbf{x}(t)|\mathbf{s}(t)}\{\mathbf{x}(t)\}$. Since the natural logarithm is a monotonous function, increasing $p_{\mathbf{x}(t)|\mathbf{s}(t)}\{\mathbf{x}(t)\}$ is equivalent to increasing $\ln\left[p_{\mathbf{x}(t)|\mathbf{s}(t)}\{\mathbf{x}(t)\}\right]$. Thus the likelihood function of $\mathbf{x}(t)$ can be defined as

$$\mathcal{L}[\mathbf{x}(t)] = -\ln\left[p_{\mathbf{x}(t)|\mathbf{s}(t)}\{\mathbf{x}(t)\}\right]. \tag{2.39}$$

Assume that the $\mathbf{n}(t)$ is a stationary zero mean Gaussian vector with covariance matrix $\mathbf{R}_{nn}$, and that $\mathbf{x}(t)$ is a Gaussian random vector with mean $s(t)\mathbf{a}(\theta)$. The likelihood function can then be expressed as [36]

$$\mathcal{L}\left[\mathbf{x}(t)\right] = c\left[\mathbf{x}(t) - \mathbf{a}(\theta)s(t)\right]^H \mathbf{R}_{nn}^{-1}\left[\mathbf{x}(t) - \mathbf{a}(\theta)s(t)\right], \tag{2.40}$$

where c is a scalar constant independent of $\mathbf{x}(t)$ and $\mathbf{s}(t)$.

Our objective is to find an estimate $\hat{s}(t)$ of $s(t)$ which minimizes (2.40).

Calculating the partial derivative of $\mathcal{L}[\mathbf{x}(t)]$ with respect to $s(t)$ to zero [36], we get

$$\frac{\partial \mathcal{L}[\mathbf{x}(t)]}{\partial s(t)} = -2\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{x}(t) + 2\hat{s}(t)\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta) = 0, \tag{2.41}$$

and we notice that $\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)$ is a scalar, then it follows that

$$\hat{s}(t) = \frac{\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}}{\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)}\mathbf{x}(t). \tag{2.42}$$

Comparing (2.42) with (2.14), it is easy to realize that the optimum weight vector $\mathbf{w}_{\text{opt}}$ using ML criterion is given by

$$\mathbf{w}_{\text{ML}} = \mathbf{w}_{\text{opt}} = \frac{\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)}. \tag{2.43}$$

Defining

$$\beta = \frac{1}{\mathbf{a}^H(\theta)\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta)}, \tag{2.44}$$

then the optimal weight vector using ML criterion can be expressed in the similar form of the Wiener-Hopf equation as

$$\mathbf{w}_{\text{ML}} = \beta\mathbf{R}_{nn}^{-1}\mathbf{a}(\theta). \tag{2.45}$$

2.4.4 Minimum Variance (MV)

Minimum variance (MV), also known as linear constrained minimum variance (LCMV), is used when the desired signal and its direction are both known *a*

priori. In (2.14), the beamformer output can be written as

$$\begin{aligned} y(t) &= \mathbf{w}^H \mathbf{x}(t) = \mathbf{w}^H \mathbf{s}(t) + \mathbf{w}^H \mathbf{n}(t) \\ &= \mathbf{w}^H \mathbf{a}(\theta) s(t) + \mathbf{w}^H \mathbf{n}(t). \end{aligned} \tag{2.46}$$

In order to obtain the desired signal with a specific gain in a given direction, we can use a constraint [5]

$$\mathbf{w}^H \mathbf{a}(\theta) = g. \tag{2.47}$$

Substituting (2.47) into (2.46), we obtain the array output subject to the constraint as

$$y(t) = g s(t) + \mathbf{w}^H \mathbf{n}(t). \tag{2.48}$$

Since $\mathbf{u}(t)$ is assumed to be uncorrelated and zero mean Gaussian, we have $\mathcal{E}\{y(t)\} = g s(t)$. The variance of the array output then is given by

$$\begin{aligned} \text{var}\{y(t)\} &= \mathcal{E}\{[y(t) - g s(t)][y(t) - g s(t)]^*\} \\ &= \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w}. \end{aligned} \tag{2.49}$$

Now using the method of Lagrange, we have

$$\nabla_{\mathbf{w}} \left\{ \mathbf{w}^H \mathbf{R}_{nn} \mathbf{w} - \beta \left[g - \mathbf{w}^H \mathbf{a}(\theta) \right] \right\} = 0, \tag{2.50}$$

or equivalently,

$$\mathbf{R}_{nn} \mathbf{w} - \beta \mathbf{a}(\theta) = 0, \tag{2.51}$$

where β is used as Lagrange multiplier. If $\mathbf{R}_{nn}$ is invertible the optimum weight vector using MV criterion can be expressed as

$$\mathbf{w}_{\mathrm{MV}} = \beta \mathbf{R}_{nn}^{-1} \mathbf{a}(\theta), \tag{2.52}$$

where

$$\beta = \frac{g}{\mathbf{a}(\theta) \mathbf{R}_{nn}^{-1} \mathbf{a}(\theta)}. \tag{2.53}$$

When $g = 1$, the MV beamformer is often referred to as the minimum variance distortionless response (MVDR) beamformer, or the Capon beamformer.

2.5 Adaptive Algorithms

2.5.1 Least Mean Square

The least mean square (LMS) is the most popular adaptive algorithm for continuous adaptation [37]. The algorithm is based on the steepest-descent method [38], which chooses the weight vector to minimize the ensemble average of the error squares toward the MSE. Using the steepest decent method, the updated weight vector at time $(n+1)$ is given by [37],

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \nabla_{\mathbf{w}} \mathcal{E} \left\{ \epsilon^2(n) \right\}, \tag{2.54}$$

where μ is the step size, and it needs to satisfy the following condition to ensure the convergence characteristics of $\mathbf{w}(n)$,

$$0 < \mu < \frac{1}{\lambda_{\max}}, \tag{2.55}$$

where $\lambda_{\max}$ is the largest eigenvalue of the covariance matrix $\mathbf{R}_{xx}$. From (2.26) we have

$$\nabla_{\mathbf{w}}\mathcal{E}\left\{\epsilon^2(n)\right\} = -2\mathbf{r}_{xr} + 2\mathbf{R}_{xx}\mathbf{w}(n). \tag{2.56}$$

Substituting (2.56) into (2.54), we have

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{r}_{xr} - \mathbf{R}_{xx}\mathbf{w}(n)]. \tag{2.57}$$

In order to update the optimum weight using (2.57), it is necessary to know in advance both $\mathbf{R}_{xx}$ and $\mathbf{r}_{xr}$, and a convenient way is to use their instantaneous values,

$$\mathbf{R}_{xx}(n) = \mathbf{x}(n)\mathbf{x}^H(n), \tag{2.58}$$

$$\mathbf{r}_{xr}(n) = \mathbf{x}(n)r^*(n). \tag{2.59}$$

Thus (2.57) now becomes

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu\mathbf{x}(n)\left[r^*(n) - \mathbf{x}^H(n)\mathbf{w}(n)\right] \\ &= \mathbf{w}(n) + \mu\mathbf{x}(n)\left[r^*(n) - y^*(n)\right] \\ &= \mathbf{w}(n) + \mu\mathbf{x}(n)\epsilon^*(n).\end{aligned} \tag{2.60}$$

It is clear that the convergence rate of the LMS algorithm depends on the step size μ, and as a consequence, on the eigenvalue spread of the covariance matrix $\mathbf{R}_{xx}$.

2.5.2 Sample Matrix Inverse (SMI)

If the desired and reference signals are both known *a priori*, then the optimal weights could be computed using the direct inversion of the covariance matrix $\mathbf{R}_{xx}$ as in (2.27). Since in general the desired and reference signals are unknown in practice, it is possible to use their estimates from the input data vector as [37]

$$\mathbf{R}_{xx}(n) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}(i)\mathbf{x}^H(i), \tag{2.61}$$

$$\mathbf{r}_{xr}(n) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}(i)r^*(i). \tag{2.62}$$

From (2.27), it follows that the updated weight vector using the SMI algorithm is given by

$$\mathbf{w}(n) = \mathbf{R}_{xx}^{-1}(n)\mathbf{r}_{xr}(n). \tag{2.63}$$

SMI is a block-adaptive algorithm and it has been shown to be the fastest algorithm for estimating the optimum weight vector [32]. However, it suffers the problems of increased computational complexity and numerical instability due to the inversion of a large matrix.

2.5.3 Recursive Least Square (RLS)

RLS algorithm determines the weight vector which minimizes a cost function $Q(n)$ which is the sum of the error squares over a time window

$$Q(n) = \sum_{i=1}^{n}\gamma^{n-i}|\epsilon(i)|^2, \tag{2.64}$$

where the error function $\epsilon(i)$ is defined in (2.23) and $0 < \gamma < 1$ is the forgetting factor. Using the least squares method, the covariance matrix and correlation vector are calculated in the following way [5],

$$\mathbf{R}_{xx}(n) = \sum_{i=1}^{n} \gamma^{n-i} \mathbf{x}(i)\mathbf{x}^H(i), \tag{2.65}$$

$$\mathbf{r}_{xr}(n) = \sum_{i=1}^{n} \gamma^{n-i} \mathbf{x}(i)r^*(i). \tag{2.66}$$

Factoring out the terms corresponding to $i = n$, (2.65) and (2.66) become

$$\mathbf{R}_{xx}(n) = \sum_{i=1}^{n-1} \gamma^{(n-1)-i}\gamma \mathbf{x}(i)\mathbf{x}^H(i) + \mathbf{x}(n)\mathbf{x}^H(n) = \gamma \mathbf{R}_{xx}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n), \tag{2.67}$$

$$\mathbf{r}_{xr}(n) = \sum_{i=1}^{n-1} \gamma^{(n-1)-i}\gamma \mathbf{x}(i)r^*(i) + \mathbf{x}(n)r^*(n) = \gamma \mathbf{r}(n-1) + \mathbf{x}(n)r^*(n). \tag{2.68}$$

Apply Woodebury's Identity [39], we can obtain the inversion of the covariance matrix as follows,

$$\mathbf{R}_{xx}^{-1}(n) = \gamma^{-1}\left[\mathbf{R}_{xx}^{-1}(n-1) - \mathbf{q}(n)\mathbf{x}^H(n)\mathbf{R}_{xx}^{-1}(n-1)\right], \tag{2.69}$$

where

$$\mathbf{q}(n) = \frac{\gamma^{-1}\mathbf{R}_{xx}^{-1}(n-1)\mathbf{x}(n)}{1 + \gamma^{-1}\mathbf{x}^H(n)\mathbf{R}_{xx}^{-1}(n)\mathbf{x}(n)}. \tag{2.70}$$

The estimated weight vector can be updated using (2.27) as

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{R}_{xx}^{-1}(n)\mathbf{r}_{xr}(n) \\ &= \gamma^{-1}\left[\mathbf{R}_{xx}^{-1}(n-1) - \mathbf{q}(n)\mathbf{x}^H(n)\mathbf{R}^{-1}(n-1)\right]\left[\gamma \mathbf{r}(n-1) + \mathbf{x}(n)r^*(n)\right],\end{aligned} \tag{2.71}$$

which finally gives us

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{q}(n)\left[r^*(n) - \mathbf{w}^H(n-1)\mathbf{x}(n)\right]. \tag{2.72}$$

Since the RLS algorithm utilizes information from the initial sample to estimate the weight, it is an order of magnitude faster than that of the LMS algorithm [38]. However, this convergence improvement is achieved at the expense of an increased computational complexity.

Chapter 3

DOA Estimation Algorithms

Array processing techniques generally require the spatial properties of the signals impinging on the array of sensors, where the most important issue lies in the number of signals of interest and their directions-of-arrival (DOA). Many methods have been developed in the field of DOA estimation, and they can be broadly classified into *Beamforming techniques*, *Maximum Likelihood techniques* and *Subspace based techniques*. We will first present an overview of these DOA methods in this chapter.

3.1 Beamforming Techniques

The basic idea behind beamforming techniques for DOA estimation [40] is to "steer" the array in one direction at a time and measure the output power. The steering locations which results in maximum power yield the DOA estimates.

Given the knowledge of array steering vector, an array can be steered electronically just as a fixed antenna can be steered mechanically. A weight vector

$\mathbf{w}$ can be used to linearly combine the output signal from the sensors to form a single output signal $y(t)$ as that defined in (2.14). The total output power can be expressed as,

$$\begin{aligned} P(\mathbf{w}) &= \frac{1}{N}\sum_{t=1}^{N}|y(t)|^2 \\ &= \frac{1}{N}\sum_{t=1}^{N}\mathbf{w}^H\mathbf{x}(t)\mathbf{x}^H(t)\mathbf{w} \\ &= \mathbf{w}^H\hat{\mathbf{R}}_{xx}\mathbf{w}, \end{aligned} \tag{3.1}$$

where N is the number of snapshots. Different beamforming techniques were developed by taking different choices of the weighting vector $\mathbf{w}$. Three main techniques of this class are discussed here.

3.1.1 Conventional Beamformer

For any particular direction θ_0, the antenna pattern formed using the weight vector $\mathbf{w}_{\text{cb}} = \mathbf{a}(\theta_0)$ has the highest gain in the direction θ_0 of any possible weight vector of the same magnitude. The reason is $\mathbf{w}_{\text{cb}}$ optimally adjust the amplitudes, and aligns the phases of the signal components arriving from θ_0 at the sensors, therefore causing them to add constructively. In conventional beamforming approach [6], the beam is scanned over the angular region of interest and for each look direction θ, the average power output $P_{\text{cb}}(\theta)$ of the steered array is measured as,

$$P_{\text{cb}}(\theta) = \mathbf{w}^H\hat{\mathbf{R}}_{xx}\mathbf{w}. \tag{3.2}$$

The weight vector which maximizes $P_{\text{cb}}(\theta)$ is

$$\mathbf{w}_{\text{cb}} = \frac{\mathbf{a}(\theta_0)}{\sqrt{\mathbf{a}^H(\theta_0)\mathbf{a}(\theta_0)}}. \tag{3.3}$$

The above weight vector can be interpreted as a spatial filter which has been matched to the impinging signal. Intuitively, the array weight vector equalizes the delays experienced by the signal on various sensors to maximally combine their respective contributions.

A simulation was conducted employing a 6 element ULA with its omnidirectional sensors separated by half wavelength. Two equi-powered uncorrelated signals were made to impinge on the array from 10° and 30°. An SNR of 10 dB was assumed. Figure 3.1 shows the simulation results. It can be seen that the peaks at 10° and 30° are not quite obvious and somewhat "averaged".

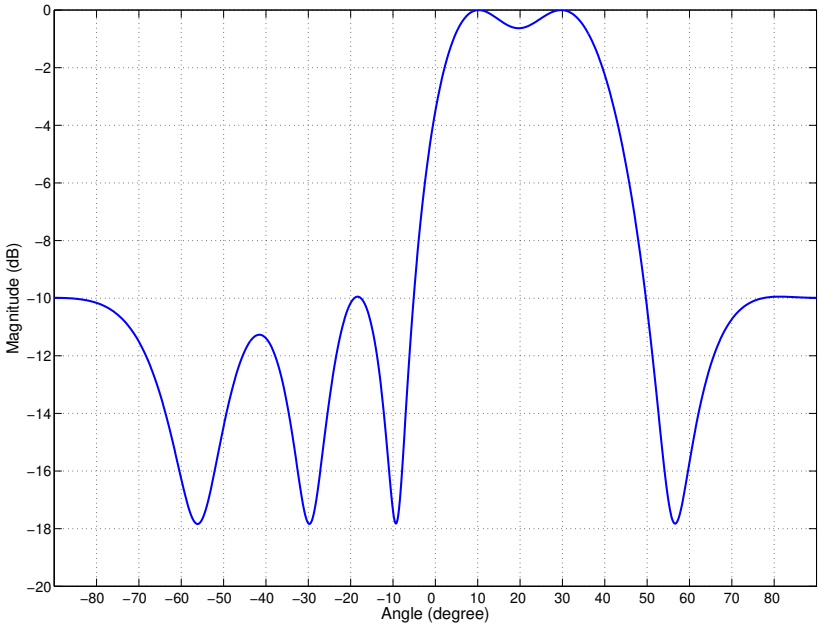


Figure 3.1: Conventional beamformer

Inserting (3.3) into (3.2), the output power as a function of DOA, *i.e.* the spatial spectrum, is obtained as

$$P_{\mathrm{cb}}(\theta) = \frac{\mathbf{a}^H(\theta)\hat{\mathbf{R}}_{xx}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{a}(\theta)}. \tag{3.4}$$

This method has many disadvantages. The width of the beam and the height of the sidelobes limit the effectiveness when signals arriving from multiple directions and sources are present, because the signals over a wide angular region contribute to the measured average power at each look direction. Hence this technique has poor resolution. Although it is possible to increase the resolution by adding more sensor elements, increasing the number of sensors increases the number of receivers and the amount of storage required for the calibration data.

3.1.2 Capon Beamformer

The conventional beamformer works on the premise that pointing the strongest beam in a particular direction yields the best estimate of power arriving in that direction. In other words, all the degrees of freedom (equal in number to one less than the number of sensors) available to the array were used in forming a beam in the required look direction. This works well when there is only one signal present. But when there is more than one signal present, the array output power contains contribution from the desired signal as well as the undesired ones from other directions.

Capon's method [41] overcomes this problem by using some degrees of freedom to form a beam in the look direction and simultaneously using the remaining degrees of freedom to form nulls in other directions in order to reject other signals. In terms of the array processor output power, forming nulls in the directions from which other signals arrive can be accomplished by minimizing the output power and simultaneously constraining a beam (or at least maintaining a unity gain) in the look direction. Thus, for a particular look direction, Capon's method uses all but one of the degrees of the freedom to minimize the

array processor output power while using the remaining degrees of freedom to constrain the gain in the look direction to be unity, that is,

$$\min P(\mathbf{w}) \qquad \text{subject to } \mathbf{w}^H\mathbf{a}(\theta) = 1. \tag{3.5}$$

The weight vector chosen in this way is often referred to as the minimum variance distortion-less response (MVDR) beamformer since for a particular look direction, it minimizes the variance (average power) of the array processor output signal while passing the signal arriving from the look direction with no distortion (unity gain and zero phase shift signal). The resulting weight vector is shown to be given by

$$\mathbf{w}_{\text{cap}} = \frac{\hat{\mathbf{R}}_{xx}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\hat{\mathbf{R}}_{xx}^{-1}\mathbf{a}(\theta)}. \tag{3.6}$$

Substituting the above weight vector into (3.1), the following spatial spectrum is obtained,

$$P_{\text{cap}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\hat{\mathbf{R}}_{xx}^{-1}\mathbf{a}(\theta)}. \tag{3.7}$$

A simulation was conducted in the same situation as that in the previous subsection. Figure 3.2 shows the result. It can be seen that the peaks at 10° and 30° are somewhat well separated compared with that of conventional beamformer.

Though it provides a better resolution compared to conventional beamforming technique, Capon's method suffers from many other disadvantages. Capon's method fails if other signals correlated with the SOI are present because it inevitably uses that correlation to reduce the processor output power without spatially nulling it. In other words, the correlated components may be combined

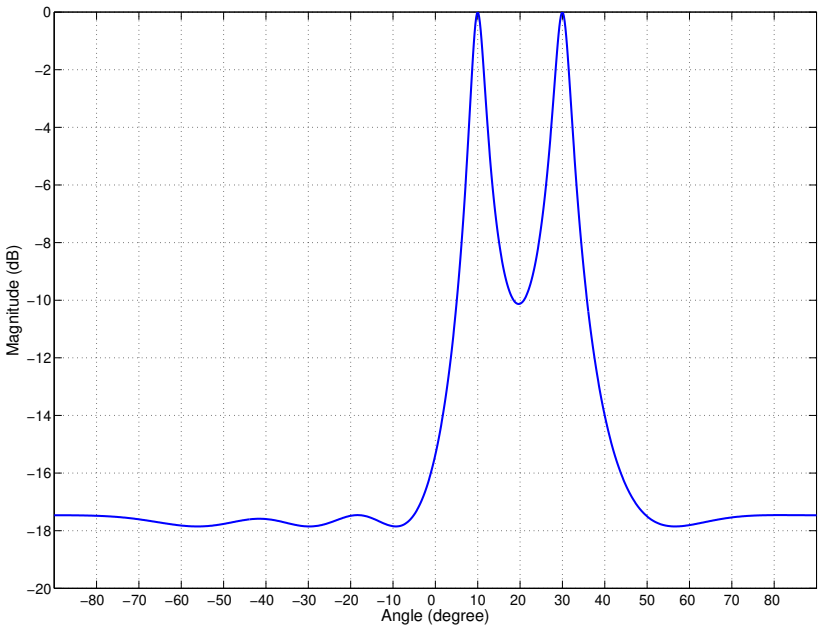


Figure 3.2: Capon's beamformer

destructively in the process of minimizing the output power. Also, Capon's method requires the computation of a matrix inverse, which can be expensive for large array configuration.

3.1.3 Linear Prediction Method

This method estimates the output of one sensor using linear combinations of the remaining sensor outputs and minimizes the mean square prediction error, that is, the error between the estimate and the actual output [42]. Thus it obtains the array weights by minimizing the mean output power of the array subject to the constraint that the weight on the selected sensor is unity. Expressions for

the array weights and the power spectrum are given respectively by

$$\mathbf{w}_{\text{lp}} = \frac{\hat{\mathbf{R}}_{xx}^{-1}\mathbf{u}}{\mathbf{u}^H\hat{\mathbf{R}}_{xx}^{-1}\mathbf{u}} \tag{3.8}$$

and

$$P_{\text{lp}}(\theta) = \frac{\mathbf{u}^H\hat{\mathbf{R}}_{xx}^{-1}\mathbf{u}}{|\mathbf{u}^H\hat{\mathbf{R}}_{xx}^{-1}\mathbf{a}(\theta)|^2} \tag{3.9}$$

where $\mathbf{u}$ is a column vector of all zeros except one element which is equal to 1. The position of this element corresponds to the position of selected element in the array for predicting its output. There is no criterion for proper choice of this element, however, the choice of this element does affect the resolution capability of the estimate, and this effect is dependent on the SNR and the angular separation of the directional sources. The linear prediction method performs well in a moderately low SNR environment and is a good compromise in situations where sources are of approximately equal strength and are nearly coherent.

A simulation was conducted in the same situation as that in the previous 2 methods. Figure 3.3 shows the result. It can be seen that the peaks at 10° and 30° are more prominent and obvious compared to those of conventional beamformer and Capon beamformer.

3.2 Maximum Likelihood Method

Maximum likelihood techniques were some of the first techniques investigated for DOA estimation. This method estimates the DOAs from a given set of array samples by maximizing the log-likelihood function [43]. The likelihood function

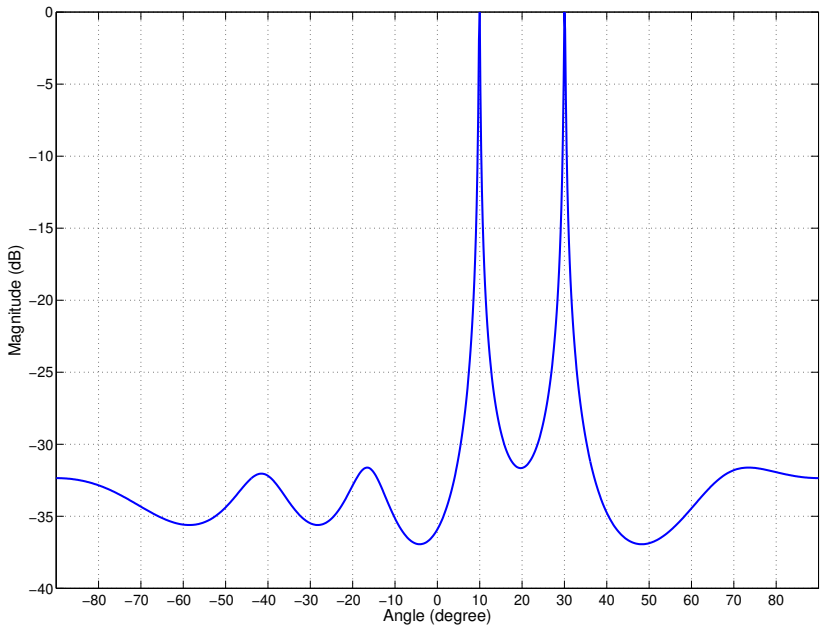


Figure 3.3: Linear prediction beamformer

is the joint probability density function of the sampled data given the DOAs and viewed as a function of the desired variables the DOAs, for this case. The method searches for those directions that maximize the log of this function, the log-likelihood function. The ML criterion signifies that plane waves from these directions are most likely to cause the given samples to occur.

The maximization of the log-likelihood function is a nonlinear optimization problem. In the absence of a closed form solution, it requires iterative schemes for solutions. There are many such schemes available in the literature, such as the well-known gradient descent algorithm, Newton-Raphson method [44], alternating projection method [45] and expectation maximum algorithm [43] *etc.*

The ML method gives a superior performance compared to other methods,

particularly when the SNR is small, the number of samples are small, or the sources are correlated [43], and thus is of practical interest. For the case of estimating the direction of a single source, the results obtained by this method are asymptotically unbiased [46], that is, the expected values of the estimates are equal to their true values. When a large number of samples are available, ML techniques become more and more computationally intensive. In such cases, other computationally more efficient schemes, such as subspace based techniques, may be used with performance almost equal to this method.

3.3 Subspace Based Method

These methods basically rely on the following properties of the covariance matrix $\mathbf{R}_{xx}$ of received signal:

1. The space spanned by its eigenvectors can be partitioned into two subspaces, namely, the *signal subspace* and *noise subspace*.

2. The steering vectors corresponding to the signal subspace is orthogonal to the noise subspace.

It should be observed that the noise subspace is spanned by the eigenvectors associated with the smaller eigenvalues of the covariance matrix and the signal subspace is spanned by the eigenvectors associated with its larger eigenvalues. The popular algorithms like ESPRIT and MUSIC belong to this class. A discussion on the basic concepts of subspace based approach for direction finding is first given in the next subsection [47].

3.3.1 Concept of Subspaces

With a given matrix $\mathbf{X}$ of size $M \times N$ and with complex entries, one may want to know how many columns (rows) of this matrix are non-parallel or independent of each other. If there are $d \leq M \leq N$ independent columns in $\mathbf{X}$, then this matrix is said to have a d dimensional range or column space, which is a subspace of the M-dimensional Euclidean space $\mathbb{C}^M$. The rank of the matrix is the dimension of this subspace. If $d = M$, then the matrix is of full rank, and for $d < M$, it is rank deficient. Now $\mathbb{C}^M$ is spanned by the columns of any unitary matrix in $\mathbb{C}^{M \times M}$, the euclidean space of square, complex valued M dimensional matrices. The same holds for $\mathbb{C}^N$ of which the row space of $\mathbf{X}$ is a d dimensional subspace: the columns of any $N \times N$ unitary matrix in $\mathbb{C}^{N \times N}$ span the vector space $\mathbb{C}^N$. Assuming $d \leq M \leq N$, a unitary matrix $\mathbf{U}$ can be chosen such that the d dimensional column space of $\mathbf{X}$ is spanned by a subset of d columns of $\mathbf{U}$, *e.g.* the first d columns, which together formm a matrix $\mathbf{U}_s$. Let the remaining $M - d$ columns together form a matrix $\mathbf{U}_n$. Then,

$$\mathbf{U} = [\mathbf{U}_s \ \mathbf{U}_n]. \tag{3.10}$$

Since $\mathbf{U}$ is a unitary matrix, it can be observed that

1. From $\mathbf{U}^H\mathbf{U} = \mathbf{I}_M$,
   (a) $\mathbf{U}_s^H\mathbf{U}_s = \mathbf{I}_d$,
   (b) $\mathbf{U}_s^H\mathbf{U}_n = \mathbf{0}$,
   (c) $\mathbf{U}_o^H\mathbf{U}_n = \mathbf{I}_{M-d}$.
2. From $\mathbf{U}\mathbf{U}^H = \mathbf{I}_M$,

(d) $\mathbf{U}_s\mathbf{U}_s^H + \mathbf{U}_n\mathbf{U}_n^H = \mathbf{I}_M$.

where $\mathbf{I}_d$ is the identity matrix of order d, and the same applies for $\mathbf{I}_{M-d}$. Relations (a)–(d) tell us that any vector $\mathbf{u} \in \mathbb{C}^M$ can be decomposed into two mutually orthogonal vectors $\mathbf{u}_s$ and $\mathbf{u}_n$ in the spaces spanned by $\mathbf{U}_s$ and $\mathbf{U}_n$ respectively. These two spaces are d dimensional and $M-d$ dimensional orthogonal subspaces in $\mathbb{C}^M$, and their direct sum is equal to $\mathbb{C}^M$. These two subspaces are orthogonal to each other, which implies that in terms of inner products, the noise free signals and disturbances are independent of one another. The dominant subspace consists of signals of interest and is referred to as *the signal subspace* while its complimentary subspace is referred to as *the noise subspace* [47].

Hence, the observation matrix is decomposed to obtain these two subspaces. Singular value decomposition (SVD) is a mathematical tool which can deocmpose $\mathbf{X}$ as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{U}$ and $\mathbf{V}$ are matrices whose columns span the column and row spaces of $\mathbf{X}$ respectively and where $\mathbf{\Sigma}$ is an invertible $d \times d$ matrix.

3.3.2 Singular Value Decomposition

A mathematical tool used to decompose the range space of the observation matrix into two complimentary subspaces is the singular value decomposition (SVD). The SVD is computationally very robust and allows for high resolution discrimination against noise contamination.

In terms of the above discussion of subspaces, the singular value decomposition gives rise to following decomposition on the $M \times N$ matrix $\mathbf{X}$ of rank

d,

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = [\mathbf{U}_s\ \mathbf{U}_n] \begin{bmatrix} \mathbf{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_n \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^H \\ \mathbf{V}_n^H \end{bmatrix}, \tag{3.11}$$

where $\mathbf{\Sigma}$ is an $M \times N$ diagonal matrix containing the singular values σ_i of $\mathbf{X}$ in descending order,

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d \geq \sigma_{d+1} = \ldots = 0. \tag{3.12}$$

Not that only d singular values are nonzero. The d columns of $\mathbf{U}_s$ corresponding to these nonzero singular values span the column space of $\mathbf{X}$ and are called *left singular vectors*. Similarly the d columns of $\mathbf{V}_s$ are called *right singular vectors* and span the row space of $\mathbf{X}$ (or the column space of $\mathbf{X}^H$). However, only the d largest singular values are of interest to us.

Another way of decomposition is eigenvalue decomposition on the data covariance matrix $\mathbf{X}\mathbf{X}^H$. The main difference between these two approaches is that SVD based algorithms operate directly on observation matrix $\mathbf{X}$ instead of "squared" matrix $\mathbf{X}\mathbf{X}^H$, and thus making it superior in practical applications with finite precision.

In principle, the subspace based methods search for directions such that the steering vectors associated with these directions are orthogonal to the noise subspace and are contained in the signal subspace. Once the signal subspace has been determined, the model parameters can be extracted from it. This insight gives rise to a number of subspace based approaches. In the following subsection, one such technique called MUltiple SIgnal Classification (MUSIC) is discussed.

3.3.3 MUSIC Algorithm

MUSIC (MUltiple SIgnal Classification) is one of the earliest proposed and the most popular methods for super resolution direction finding [48]. In terms of the data model described in section 2.2, the input data covariance matrix $\mathbf{R}_{xx}$ is written as

$$\mathbf{R}_{xx} = \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H + \sigma_n^2\mathbf{I}_M, \tag{3.13}$$

where $\mathbf{R}_{ss}$ is the signal correlation matrix. The eigenvalues of $\mathbf{R}_{xx}$ are values $\{\lambda_1 \ldots \lambda_M\}$ which satisfy the following condition,

$$|\mathbf{R}_{xx} - \lambda_i\mathbf{I}_M| = 0, \tag{3.14}$$

where $\mathbf{I}_M$ is the identity matrix. Substituting (3.13) into (3.14), we can get

$$|\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H - (\lambda_i - \sigma_n^2)\mathbf{I}_M| = 0. \tag{3.15}$$

We denote the eigenvalues of $\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H$ by $\{\nu_1, \nu_2, \ldots, \nu_M\}$, and we can then get the following relationship between ν_i and λ_i,

$$\nu_i = \lambda_i - \sigma_n^2. \tag{3.16}$$

Since $\mathbf{A}$ is composed of steering vectors which are linearly independent, it has full column rank, and the signal correlation matrix $\mathbf{R}_{ss}$ is nonsingular as long as the incident signals are not highly correlated or coherent.

The full column rank of $\mathbf{A}$ and non-singularity of $\mathbf{R}_{ss}$ guarantee that when the number of incident signals d is less than the number of elements M, the matrix $\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H$ is positive semidefinite with rank d. This implies that $M - d$

eigenvalues of $\mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H$ are zero. From (3.16), this means that $M-d$ eigenvalues of $\mathbf{R}_{xx}$ are equal to the noise variance σ_n^2. Therefore,

$$\lambda_{d+1} = \lambda_{d+2} = \ldots = \lambda_M = \lambda_{\min} = \sigma_n^2. \tag{3.17}$$

Once the multiplicity of the smallest eigenvalues is determined as k, an estimate of the number of signals can be obtained from the following relationship,

$$d = M - k. \tag{3.18}$$

In practice, however, when the covariance matrix $\mathbf{R}_{xx}$ is estimated from a finite data sample, all the eigenvectors corresponding to the noise power will not be identical. Instead, they will appear as a closely spaced cluster, and the variance of this spread decreases as the number of samples increases.

We express the eigenvector in association with a particular eigenvalue λ_i as $\mathbf{q}_i$ which satisfies,

$$(\mathbf{R}_{xx} - \lambda_i \mathbf{I}_M)\,\mathbf{q}_i = 0. \tag{3.19}$$

For eigenvectors associated with the $M-d$ smallest eigenvalues, we have,

$$\begin{aligned}(\mathbf{R}_{xx} - \sigma_n^2\mathbf{I}_M)\mathbf{q}_i =& \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H\mathbf{q}_i + \sigma_n^2\mathbf{I}_M\mathbf{q}_i - \sigma_n^2\mathbf{I}_M\mathbf{q}_i = 0 \\ \Rightarrow& \mathbf{A}\mathbf{R}_{ss}\mathbf{A}^H\mathbf{q}_i = 0.\end{aligned} \tag{3.20}$$

Since $\mathbf{A}$ has full rank and $\mathbf{R}_{ss}$ is nonsingular, these imply that

$$\mathbf{A}^H\mathbf{q}_i = \mathbf{0} \qquad \text{for } i = d+1, \ldots, M. \tag{3.21}$$

The above equation means that the eigenvectors associated with the $M - d$ smallest eigenvalues are orthogonal to the d steering vectors that make up matrix $\mathbf{A}$, that is,

$$\{\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_d)\} \perp \{\mathbf{q}_{d+1}, \ldots, \mathbf{q}_M\}. \tag{3.22}$$

This remarkable observation forms the corner stone of almost all of the subspace based methods. It means that one can estimate the steering vectors associated with the received signals by finding the steering vectors that are most nearly orthogonal to the d eigenvectors associated with the eigenvalues of $\mathbf{R}_{xx}$ that are approximately equal to σ_n^2.

This analysis shows that the eigenvectors of the covariance matrix $\mathbf{R}_{xx}$ belong to either of the two orthogonal subspaces, *i.e.* the principal eigen subspace (signal subspace) and the non-principal eigen subspace (noise subspace). The steering vectors corresponding to the DOAs of signals lie in the signal subspace and are hence orthogonal to the noise subspace. By searching through all possible array steering vectors to find those which are perpendicular to the space spanned by the non-principal eigenvectors, the DOAs can thus be determined [6].

To search the noise subspace, a matrix $\mathbf{U}_n$ containing the noise eigenvectors is formed as

$$\mathbf{U}_n = [\mathbf{q}_{d+1}, \ldots, \mathbf{q}_M]. \tag{3.23}$$

Since the steering vectors corresponding to signal components are orthogonal to the noise subspace eigenvectors, we can have

$$\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta) = 0 \tag{3.24}$$

for all the signals and their noncoherent multipath components in the environment. Then the DOAs of the multiple incident signals can be estimated by locating the peaks of a MUSIC spatial spectrum determined by

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta)}. \tag{3.25}$$

Orthogonality between $\mathbf{a}(\theta)$ and $\mathbf{U}_n$ will minimize the denominator and will hence give rise to peaks in the MUSIC spectrum. The d largest peaks in the MUSIC spectrum correspond to the directions of arrival of the signals impinging on the array.

A simulation was conducted employing a 6 elements ULA with omnidirectional sensors separated by half wavelength. Three equip-power and uncorrelated signals arrived at the array from 10°, 20° and 40° respectively. SNR was assumed to be 10dB. 50 trials were taken with each trial averaged over 250 snapshots. Figure 3.4 shows the results.

Now we summarize the MUSIC algorithm as follows:

Step 1: Collect input samples $\mathbf{x}_k$, $k = 1, 2, \ldots, N$ and estimate the input covariance matrix as

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}(k)\mathbf{x}^H(k). \tag{3.26}$$

Step 2: Perform eigen decomposition on $\hat{\mathbf{R}}_{xx}$

$$\hat{\mathbf{R}}_{xx}\mathbf{U} = \mathbf{U}\mathbf{\Lambda}, \tag{3.27}$$

where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_M\}$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$ are the eigenvalues of $\hat{\mathbf{R}}_{xx}$ and $\mathbf{U}$ contains the corresponding eigenvectors of $\hat{\mathbf{R}}_{xx}$.

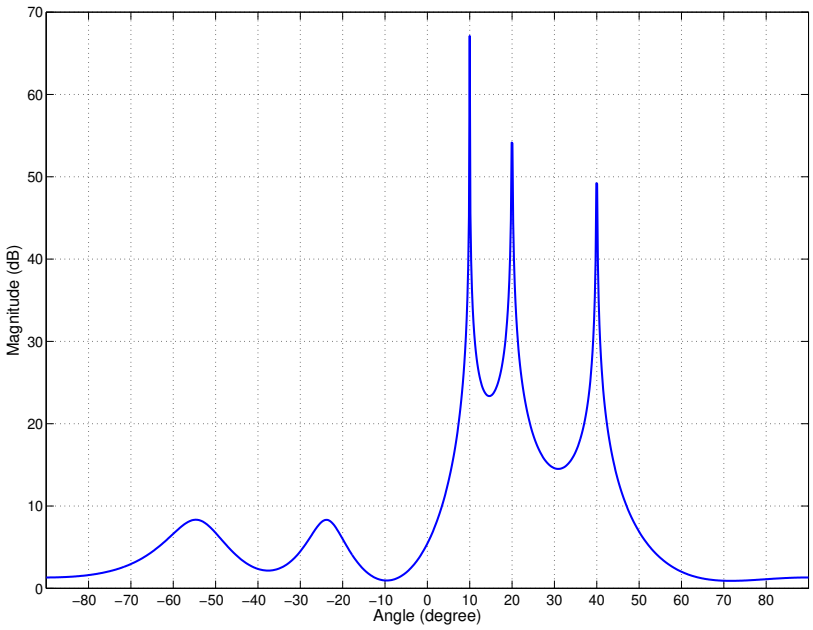


Figure 3.4: MUSIC algorithm

Step 3: Estimate the number of signals d from the multiplicity k of the smallest eigenvalue $\lambda_{\min}$ as $d = M - k$.

Step 4: Compute the MUSIC spectrum

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta)}, \tag{3.28}$$

where

$$\mathbf{U}_n = [\mathbf{q}_{d+1}, \ldots, \mathbf{q}_M] .$$

Step 5: Find the d largest peaks of $P_{\text{MUSIC}}(\theta)$ to obtain estimates of the DOAs.

3.3.4 Minimum Norm Method

Minimum norm method [49] is applicable for linear array, and it finds a DOA estimate by searching for the location of peaks in the spectrum

$$P_{\mathrm{MN}}(\theta) = \frac{1}{|\mathbf{w}^H \mathbf{a}(\theta)|}, \tag{3.29}$$

where the weight $\mathbf{w}$ is chosen with the minimum norm and its first element equal to 1. The solution of the above problem leads to the following expression for the minimum norm spatial spectrum

$$P_{\mathrm{MN}}(\theta) = \frac{1}{|\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{W}\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta)|}, \tag{3.30}$$

where matrix $\mathbf{W} = \mathbf{e}_1\mathbf{e}_1^H$ and $\mathbf{e}_1$ is the first column of an $M \times M$ identity matrix. A simulation was conducted employing a 6 element ULA with its omnidirectional sensors separated by half wavelength. Three equi-power and uncorrelated signals arrived at the array from 5°, 25° and 45°. SNR was 10dB. 50 trials were taken with each trial averaged over 250 snapshots. Figure 3.5 shows the result.

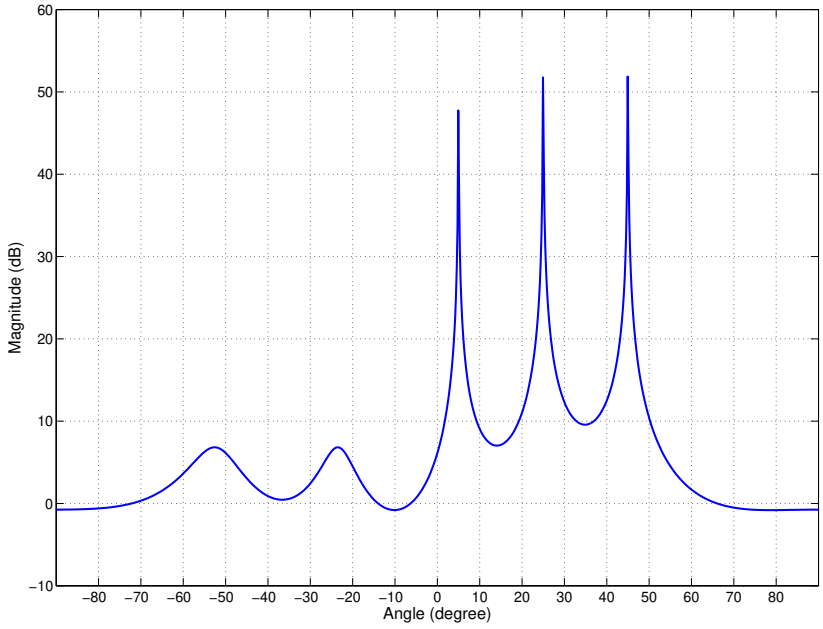


Figure 3.5: Minimum norm beamformer

Chapter 4

Proposed DOA Estimation Algorithms

4.1 Introduction

The ever growing number of subscribers and the demand for next generation data services have made the issues of capacity increase and performance improvement for wireless communication systems more and more crucial. These can be achieved by adding additional carrier frequencies or increasing cell density in the network which are generally extremely expensive. In recent years, interference cancelation through beamforming is especially attractive and has been recognized as one of the most promising and cost-effective techniques to solve the above-mentioned problems in the third generation and future wireless communication systems where capacity, carrier frequency and coverage are the most important issues.

In wireless communication systems, subscribers are usually spatially sepa-

rated and the use of antenna arrays makes it possible to track the direction-of-arrival of each signal and locate the position of a subscriber. Based on the position information, the spatial separation can be exploited through beamforming to multiplex the channel in the spatial dimension as well as in the frequency and time dimensions to receive and transmit signals in a directional manner. In this way, the co-channel interference can be reduced, which will subsequently lead to an improvement in the capacity, carrier efficiency and coverage of a wireless communication system.

This chapter is organized as follows. In the next section, existing algorithms for DOA estimation will be reviewed. A brief study about the cyclostationary property of communication signals followed by the introduction of system model is presented in section 4.3. Our proposed algorithms for noncoherent signal environment and coherent signal environment along with theoretical analysis are described in section 4.4 in detail. Section 4.5 provides some simulation examples to verify the effectiveness of our proposed algorithm.

4.2 Existing Methods

Since the middle of 20^{th} century, many methods have been developed in the field of DOA estimation, *e.g.* classical Fourier analysis, conventional beamforming, minimum-variance method [41] and maximum entropy method [50] *etc.* Over the past few decades, subspace approaches have received much prominence due to their high resolution performance and low computational complexity. Pioneered by Pisarenko [51], different algorithms in this field have been put forward, such as MUSIC by Schmidt [8], Min-Norm linear prediction by Tufts and Ku-

maresan [49], state-space methods by Kung [52], ESPRIT by Paulraj [53] and Matrix Pencil by Ouibrahim [54]. Detailed description of some of the above-mentioned methods can be found in chapter 3. Generally speaking, these subspace based approaches take advantage of certain eigen-structure properties associated with the special structure of array output covariance matrix for planar wavefronts to generate spectral peaks (or nulls) in the DOAs. They have very desirable and attractive properties, such as asymptotically unbiased estimation of the DOAs and low mean-square error which is comparable to Cramer-Rao lower bound in some signal environment. Despite all these advantages, the performance and applications of MUSIC algorithm are limited in practice due to the following reasons. First, the MUSIC algorithm requires that the total number of signals impinging on the array must be less than the number of antennas which cannot always be satisfied in real wireless applications. Second, the spatial characteristics of noise must be known *a priori*, or be transformation invariable. Moreover, MUSIC algorithm resolves all the directions of signals arriving at the array, including both desired signals and undesired ones. This results in both a waste of unnecessary computational efforts in dealing with the unwanted signals and some extra post-processing techniques to distinguish the SOIs. Besides, when an undesired signal comes from the direction that is very close to the desired signals, more antennas and long data samples are required to distinguish them.

One way to overcome the aforementioned problems is to select and process only a signal subset composed of interested signals only. In applications such as radar, sonar or telecommunications, where almost all signals exhibit the cyclostationary characteristics [55], the SOIs may have rich properties that can be exploited to cancel interferences and background noise. So it came the

Cyclic-MUSIC algorithm in [13]. The cyclic MUSIC algorithm can achieve better performance than MUSIC in some environments in which MUSIC operates unsatisfactorily or even fails to work. For instance, in situation where two signals impinge on the array from almost the same direction, cyclic MUSIC needs much less data to give correct DOA estimation than MUSIC if these two signals have different cycle frequencies. In addition, it is impossible for MUSIC to work properly if the number of signals arriving at the array exceeds the number of antennas whereas cyclic MUSIC can still function well as long as the number of signals sharing the same cycle frequency is smaller than the number of antennas. Moreover, substantial savings in computations can be achieved by using cyclic MUSIC because it discards the undesired signals and only involves the processing on SOIs.

Just as every algorithm has its own pros and cons, cyclic MUSIC cannot escape. One of the drawbacks of cyclic MUSIC is a direct consequence of its "signal-selective" advantages. When the SOIs do not have a common cycle frequence, we need to apply cyclic MUSIC repeatedly for each cycle frequency of interest, which results in a repeated computations of cyclic correlation matrix, singular value decompositions, cyclic MUSIC spectrum calculations and localization of the DOAs. To reduce such an extra waste, we in this chaper put forward a new algorithm which can obtain the DOAs of all the SOIs by simply performing the DOA estimation algorithm only once. Therefore, considerable savings in computations can be achieved. Another drawback of cyclic MUSIC is its inability to work in multipath environment. The multipath propagation due to various reflections is often encountered in most wireless communications systems. The radiated signals from a single signal source reach the array from different directions, and the coherence inherent in these signals makes the cyclic

correlation matrix singular. In such a situation, cyclic MUSIC performs as poorly as the ordinary MUSIC algorithms. To reduce multipath effect, we try to incorporate a preprocessing technique–spatial smoothing [56]–with our proposed algorithm to enable it to work with coherent signals. Some other works dealing with cyclostationary signals in the presence of coherent sources are presented in [57] and [58]. Other algorithms such as "cyclic maximum likelihood direction finding" are presented in [55] which can resolve the multipath signals as well. But it requires a search in high dimension which would incur a heavy computational load, and therefore is not considered in this chapter.

4.3 Problem Formulation

4.3.1 Cyclostationary Property

Random signals in many conventional signal processing methods are considered as statistically stationary in which the mean or autocorrelation function of the signal is time invariant. But for most manmade signals encountered in communications, such as radar and sonar systems, their mean and autocorrelation do vary periodically with time [55]. This includes, for example, signals with Amplitude Modulation (AM), Binary Phase Shift Keying (BPSK), Quaternary Phase Shift Keying (QPSK) *etc.* This attribute is referred to as *cyclostationarity*, which enables generation of cyclic spectral lines in the signal spectrum.

Now let us describe cyclostationarity mathematically. Cyclic correlation for a given signal $s(t)$, is defined as the Fourier transform of the correlation of the

signal at a certain frequency, let us say α [59]

$$r_{ss}^{\alpha} = \int \mathcal{E}\left\{s(t+\tau/2)s^*(t-\tau/2)\right\} \mathrm{e}^{-j2\pi\alpha t}\mathrm{d}t. \tag{4.1}$$

We know if $s(t)$ is stationary, then its correlation is independent on t. Therefore the Fourier transform of the correlation of a stationary signal is zero unless when $\alpha = 0$. But if $s(t)$ is cyclostationary, then its correlation is periodic. If we denote the period as T_s, then the Fourier transform of the correlation of the signal will have large values at frequencies of integer multiples of $1/T_s$. These frequencies are referred to as *cycle frequencies*. Similarly, cyclic conjugate correlation of the signal $s(t)$ can be defined as

$$r_{ss^*}^{\alpha} = \int \mathcal{E}\left\{s(t+\tau/2)s(t-\tau/2)\right\} \mathrm{e}^{-j2\pi\alpha t}\mathrm{d}t. \tag{4.2}$$

Signal s(t) is said to be cyclostationary if $r_{ss}^{\alpha}(\tau)$ or $r_{ss^*}^{\alpha}(\tau)$ is not zero at some time delay τ and cycle frequency α. Many man-made communication signals exhibit cyclostationarity due to modulation, periodic gating, etc. They usually have cycle frequencies at twice the carrier frequency or multiples of the baud rate or combinations of these. Moreover, some signals may have both nonzero cyclic correlation and nonzero cyclic conjugate correlation. Note cyclic correlation can be simplified as [55]

$$r_{ss}^{\alpha}(\tau) = \langle s(t+\tau/2)s^*(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty}, \tag{4.3}$$

$$r_{ss^*}^{\alpha}(\tau) = \langle s(t+\tau/2)s(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty}, \tag{4.4}$$

where $\langle\bullet\rangle_{\infty}$ represents infinite time average operation.

For a given vector $\mathbf{x}(t)$, we can calculate the cyclic correlation matrix and

the cyclic conjugate correlation matrix which are defined as

$$\mathbf{R}_{xx}^{\alpha}(\tau) = \langle \mathbf{x}(t+\tau/2)\mathbf{x}^{H}(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty}, \tag{4.5}$$

$$\mathbf{R}_{xx^*}^{\alpha}(\tau) = \langle \mathbf{x}(t+\tau/2)\mathbf{x}^{T}(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty}. \tag{4.6}$$

Instead of infinite time average, in real practice, these above autocorrelation matrices are estimated by the finite time operator.

4.3.2 System Model

We consider an uniform linear array (ULA) with M antennas and element spacing d. K narrow band far-field planar waves $s_k(t)$ with wavelength λ impinge on the array from different angular directions θ_k, $k = 1, 2, \ldots, K$. The steering vector for the k-th signal can be expressed as

$$\mathbf{a}_k = [1, \mathrm{e}^{-j\frac{2\pi}{\lambda}d\sin\theta_k}, \ldots, \mathrm{e}^{-j(M-1)\frac{2\pi}{\lambda}d\sin\theta_k}]^T, \tag{4.7}$$

then the received data vector $\mathbf{x}(t)$ can be written as

$$\begin{aligned} \mathbf{x}(t) &= \sum_{k=1}^{K} s_k(t)\mathbf{a}_k + \mathbf{n}(t), \\ &= \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \end{aligned} \tag{4.8}$$

where $\mathbf{A}$ is the matrix for steering vectors and is defined as

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K], \tag{4.9}$$

and $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_K(t)]^T$ is source signal vector, and $\mathbf{n}(t)$ is additive noise which is uncorrelated with source signals.

Assume that we are interested in the first two signals $s_1(t)$ and $s_2(t)$, and both of them are cyclostationary signals each with cycle frequency α_i for $i = 1$ and 2. Besides, they satisfy the following properties as well,

$$r_{s_1 s_1}^{\alpha_1} \neq 0, \qquad r_{s_1 s_1}^{\alpha_2} = 0, \qquad r_{s_1 s_2}^{\alpha_1} = r_{s_1 s_2}^{\alpha_2} = 0, \tag{4.10}$$

and

$$r_{s_2 s_2}^{\alpha_1} = 0, \qquad r_{s_2 s_2}^{\alpha_2} \neq 0, \qquad r_{s_2 s_1}^{\alpha_1} = r_{s_2 s_1}^{\alpha_2} = 0. \tag{4.11}$$

And we further assume that all the remaining signals $s_k(t)$ for $k = 3, \ldots, K$ and noise are not cyclostationary or do not exhibit cyclostationarity at cycle frequencies α_i for $i = 1$ and 2.

4.4 Proposed DOA Estimation Algorithms

4.4.1 Proposed Algorithm in Noncoherent Environment

Here, we give an extension of the conventional cyclic correlation matrix by introducing a function $\varphi(t)$ as

$$\varphi(t) = \sum_{i=1}^{K_\alpha} \mathrm{e}^{-j2\pi\alpha_i t}, \tag{4.12}$$

where K_α is the number of cycle frequencies among the SOIs. In the environment described above, K_α equals 2. Now the extended cyclic correlation matrix

of received data vector can be written as

$$\mathbf{R}_{xx}(\tau) = \langle \mathbf{x}(t+\tau/2)\mathbf{x}^H(t-\tau/2)\varphi(t)\rangle_\infty. \tag{4.13}$$

Substituting (4.8) into (4.13), we can get

$$\begin{aligned}\mathbf{R}_{xx}(\tau) =& \mathbf{A}\langle \mathbf{s}(t+\tau/2)\mathbf{s}^H(t-\tau/2)\varphi(t)\rangle_\infty \mathbf{A}^H \\ &+ \langle \mathbf{n}(t+\tau/2)\mathbf{n}^H(t-\tau/2)\varphi(t)\rangle_\infty.\end{aligned} \tag{4.14}$$

With (4.10)–(4.11) and the assumptions in section 4.3.2, the above equation can be further written as

$$\mathbf{R}_{xx}(\tau) = \tilde{\mathbf{A}}\mathbf{R}_{ss}(\tau)\tilde{\mathbf{A}}^H, \tag{4.15}$$

where $\tilde{\mathbf{A}} = [\mathbf{a}_1,\ \mathbf{a}_2]$ is the matrix for steering vectors of signals of interests only, and $\mathbf{R}_{ss}(\tau)$ is given by

$$\mathbf{R}_{ss}(\tau) = \begin{bmatrix} r_{s_1s_1}^{\alpha_1} & 0 \\ 0 & r_{s_2s_2}^{\alpha_2} \end{bmatrix}. \tag{4.16}$$

In this case, the problem of DOA estimation returns to the basic cyclic-MUSIC case, so we can say that as long as M is greater than the number of desired signals, our proposed algorithm is able to estimate the directions of SOIs no matter how large the number of total signals is in the system and whether SOIs share a common cycle frequencies or not.

Now we describe the procedures of our proposed algorithm for DOA estimation:

- **Step 1**: Estimate $\mathbf{R}_{xx}(\tau)$ in (4.13) by calculating

$$\hat{\mathbf{R}}_{xx}(\tau) = \sum_{t=1}^{T} \mathbf{x}(t+\tau/2)\mathbf{x}^H(t-\tau/2)\varphi(t), \tag{4.17}$$

  where T is the number of samples.

- **Step 2**: Perform singular value decomposition on $\hat{\mathbf{R}}_{xx}(\tau)$ and obtain

$$\hat{\mathbf{R}}_{xx}(\tau) = \begin{bmatrix} \hat{\mathbf{U}}_s & \hat{\mathbf{U}}_n \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Sigma}}_s & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{\Sigma}}_n \end{bmatrix} \begin{bmatrix} \hat{\mathbf{V}}_s & \hat{\mathbf{V}}_n \end{bmatrix}^H, \tag{4.18}$$

  where $\begin{bmatrix} \hat{\mathbf{U}}_s & \hat{\mathbf{U}}_n \end{bmatrix}$ and $\begin{bmatrix} \hat{\mathbf{V}}_s & \hat{\mathbf{V}}_n \end{bmatrix}$ are matrices for left singular vectors and right singular vectors respectively. The second matrix in the right hand side of (4.18) is a diagonal matrix of nonnegative singular values in descending order. In practice, there are no zero singular values in $\hat{\mathbf{\Sigma}}_n$ but only small singular values which satisfy $T \to \infty$, $\hat{\mathbf{\Sigma}}_n \to \mathbf{0}$. Then the dimension of signal subspace which is spanned by $\hat{\mathbf{U}}_s$ can be estimated by Minimum Description Length (MDL) criterion [60].

- **Step 3**: Calculate the spectrum $P(\theta) = 1/||\hat{\mathbf{U}}_n^H \mathbf{a}(\theta)||^2$ for all the possible impinging directions, where $\mathbf{a}(\theta)$ is the steering vector associated with each direction, and the peaks in the spectrum correspond to the directions of SOIs.

4.4.2 Proposed Algorithm in Coherent Environment

It is frequently encountered in wireless communications that the signal received by the base station from a mobile terminal consists of multiple copies of the

original signal, *i.e.* the so called coherent multipath signals, due to reflection or refraction in the transmission process. To make our proposed algorithm in the previous section applicable in such environments, we make an improvement by incorporating with it a preprocessing technique–spatial smoothing. To elaborate the improved algorithm, we first describe the signal environment considered in this part.

First, consider the same signal environment as that introduced in section 4.3.2 except that SOI-1 and SOI-2 reach the array with several multipath coherent signals. Suppose that we intend to obtain the directions of $s_1(t)$ (SOI-1), $s_2(t)$ (SOI-2) and their multipath signals. Let K_i, $i = 1$ and 2, denote the number of SOI-i and its multipath components, and $\theta_i(k)$, $k = 1,\ 2,\ \ldots,\ K_i$, are the directions of SOI-i and its multipath signals respectively. Assume that SOI-1, SOI-2 and their multipath signals reach the array from different directions. We use $s_{i,k}(t) = s_i(t)\beta_i(k)$, for $k = 1, \ldots, K_i$ and $i = 1, 2$, to denote the k-th multipath signal of SOI-i, and we group the SOI-i along with its multipath signals into a vector $\mathbf{s}_i(t)$ which can be written as

$$\mathbf{s}_i(t) = s_i(t)\boldsymbol{\beta}_i, \qquad \text{for } i = 1 \text{ and } 2, \tag{4.19}$$

where $\boldsymbol{\beta}_i = [\beta_i(1), \ldots, \beta_i(K_i)]^T$ is the vector of multipath coefficients of SOI-i with the k-th nonzero element $\beta_i(k)$ representing the complex attenuation of the k-th multipath signal of SOI-i with respect to the first signal $s_i(t)$ and $\beta_i(1) = 1$.

Now we divide the M-element uniform linear array into L subarrays, each with N antennas, and L and N satisfy the relationship $M = L + N - 1$. Then the $N \times 1$ vector $\bar{\mathbf{x}}_l(t)$ of received signals in the l-th subarray can be expressed

by

$$\bar{\mathbf{x}}_l(t) = \bar{\mathbf{A}}\mathbf{B}^{l-1}\mathbf{s}(t) + \bar{\mathbf{n}}_l(t), \qquad \text{for } l = 1, \ldots, L \tag{4.20}$$

where $\bar{\mathbf{n}}_l(t)$ is the noise vector received by the l-th subarray; $\bar{\mathbf{A}}$ is the matrix containing the steering vectors of length N for SOI-1, SOI-2, their multipath components and other directional interference signals. $\mathbf{B}^l$ denotes the l-th power of diagonal matrix $\mathbf{B}$ which is defined by

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_3 \end{bmatrix}, \tag{4.21}$$

where $\mathbf{B}_i = \text{diag}\{\mathbf{b}_i\}$ with

$$\mathbf{b}_i = \begin{cases} \left[\mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_i(1)} \ldots \mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_i(k)} \ldots \mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_i(K_i)} \right]^T, & \text{for } i = 1 \text{ and } 2, \\ \left[\mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_3} \ldots \mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_k} \ldots \mathrm{e}^{-j\frac{2\pi}{\lambda} d \sin\theta_K} \right]^T, & \text{for } i = 3. \end{cases} \tag{4.22}$$

According to signals from SOI-1, SOI-2 and interferences, we may also divide $\bar{\mathbf{A}}$ into three submatrices $\bar{\mathbf{A}}_i$ for $i = 1, 2, 3$ as

$$\bar{\mathbf{A}} = [\bar{\mathbf{A}}_1 \ \bar{\mathbf{A}}_2 \ \bar{\mathbf{A}}_3], \tag{4.23}$$

and $\mathbf{s}(t)$ into three subvectors $\mathbf{s}_i(t)$ for $i = 1, 2$ and 3 as

$$\mathbf{s}(t) = [\mathbf{s}_1^T(t) \ \mathbf{s}_2^T(t) \ \mathbf{s}_3^T(t)]^T. \tag{4.24}$$

Therefore, matrix $\bar{\mathbf{A}}_i$, $i = 1,\ 2$, can be written as

$$\bar{\mathbf{A}}_i = [\bar{\mathbf{a}}_{i,1},\ \ \ldots,\ \ \bar{\mathbf{a}}_{i,K_i}], \tag{4.25}$$

where $\bar{\mathbf{a}}_{i,k}$ for $k = 1, \ldots, K_i$ and $i = 1, 2$ are steering vectors associated with SOI-i from direction $\theta_i(k)$, and they consist of the first N elements of that defined in (4.7) and can be expressed by

$$\bar{\mathbf{a}}_{i,k} = [1, \mathrm{e}^{-j\frac{2\pi}{\lambda}d\sin\theta_i(k)}, \ldots, \mathrm{e}^{-j(N-1)\frac{2\pi}{\lambda}d\sin\theta_i(k)}]^T, \qquad \text{for } i = 1 \text{ and } 2. \tag{4.26}$$

We combine the $N \times 1$ steering vectors of interference signals into matrix $\bar{\mathbf{A}}_3$. In the same vein, $\mathbf{s}_i(t)$ for $i = 1, 2, 3$ have the following expressions,

$$\mathbf{s}_1(t) = [s_1(t)\beta_1(1),\ \ \ldots,\ \ s_1(t)\beta_1(K_1)]^T = s_1(t)\boldsymbol{\beta_1}, \tag{4.27}$$

$$\mathbf{s}_2(t) = [s_2(t)\beta_2(1),\ \ \ldots,\ \ s_2(t)\beta_2(K_2)]^T = s_2(t)\boldsymbol{\beta_2}, \tag{4.28}$$

$$\mathbf{s}_3(t) = [s_3(t),\ \ s_4(t),\ \ \ldots,\ \ s_K(t)]^T. \tag{4.29}$$

The cyclic autocorrelation matrix for the l-th subarray can therefore be expressed as

$$\begin{aligned}\bar{\mathbf{R}}_{xx,l}(\tau) =& \langle \bar{\mathbf{x}}_l(t+\tau/2)\bar{\mathbf{x}}_l^H(t-\tau/2)\varphi(t)\rangle_\infty \\ =& \bar{\mathbf{A}}\mathbf{B}^{l-1}\mathbf{R}_{ss}(\tau)\mathbf{B}^{-(l-1)}\bar{\mathbf{A}}^H + \langle \mathbf{n}_l(t+\tau/2)\mathbf{n}_l^H(t-\tau/2)\varphi(t)\rangle_\infty,\end{aligned} \tag{4.30}$$

where $\mathbf{R}_{ss}(\tau)$ is the cyclic autocorrelation matrix of the source signals and can

be written by

$$\mathbf{R}_{ss}(\tau) = \begin{bmatrix} r_{s_1s_1}^{\alpha_1}(\tau)\boldsymbol{\beta}_1\boldsymbol{\beta}_1^H & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & r_{s_2s_2}^{\alpha_2}(\tau)\boldsymbol{\beta}_2\boldsymbol{\beta}_2^H & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.31}$$

As $\mathbf{R}_{ss}(\tau)$ has a special structure of being a block diagonal matrix and along with the assumptions that interference signals and noise do not exhibit cyclostationarity at cycle frequencies α_1 and α_2, we can further rewrite (4.30) as

$$\bar{\mathbf{R}}_{xx,l}(\tau) = \tilde{\bar{\mathbf{A}}}\tilde{\mathbf{B}}^{l-1}\tilde{\mathbf{R}}_{ss}(\tau)\tilde{\mathbf{B}}^{-(l-1)}\tilde{\bar{\mathbf{A}}}^H, \tag{4.32}$$

with $\tilde{\bar{\mathbf{A}}} = [\bar{\mathbf{A}}_1\ \bar{\mathbf{A}}_2]$, and $\tilde{\mathbf{B}}$, $\tilde{\mathbf{R}}_{ss}(\tau)$ being the upper left $(K_1+K_2)\times(K_1+K_2)$ submatrices of $\mathbf{B}$ and $\mathbf{R}_{ss}(\tau)$ respectively.

Next, we define the spatial smoothed cyclic autocorrelation matrix as

$$\bar{\mathbf{R}}_{xx}(\tau) = \frac{1}{L}\sum_{l=1}^{L}\bar{\mathbf{R}}_{xx,l}(\tau) = \tilde{\bar{\mathbf{A}}}\bar{\mathbf{R}}_{ss}(\tau)\tilde{\bar{\mathbf{A}}}^H, \tag{4.33}$$

where

$$\begin{aligned} \bar{\mathbf{R}}_{ss}(\tau) =& \frac{1}{L}\sum_{l=1}^{L}\tilde{\mathbf{B}}^{l-1}\tilde{\mathbf{R}}_{ss}(\tau)\tilde{\mathbf{B}}^{-(l-1)} \\ =& \frac{1}{L}\sum_{l=1}^{L}\tilde{\mathbf{B}}^{l-1}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^H\tilde{\mathbf{B}}^{-(l-1)} \\ =& \frac{1}{L}\left[\boldsymbol{\Gamma},\ \tilde{\mathbf{B}}\boldsymbol{\Gamma},\ \ldots,\ \tilde{\mathbf{B}}^{L-1}\boldsymbol{\Gamma}\right]\left[\boldsymbol{\Gamma},\ \tilde{\mathbf{B}}\boldsymbol{\Gamma},\ \ldots,\ \tilde{\mathbf{B}}^{L-1}\boldsymbol{\Gamma}\right]^H \\ =& \frac{1}{L}\mathbf{C}\mathbf{C}^H. \end{aligned} \tag{4.34}$$

In the above equation, matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = [\mathbf{\Gamma},\ \tilde{\mathbf{B}}\mathbf{\Gamma},\ \ \ldots,\ \tilde{\mathbf{B}}^{L-1}\mathbf{\Gamma}], \tag{4.35}$$

and $\mathbf{\Gamma}$ has the following expression

$$\mathbf{\Gamma} = \begin{bmatrix} \boldsymbol{\gamma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\gamma}_2 \end{bmatrix}, \tag{4.36}$$

with $\mathbf{\Gamma}\mathbf{\Gamma}^H = \tilde{\mathbf{R}}_{ss}(\tau)$ and vector $\boldsymbol{\gamma}_i = \sqrt{r_{s_i s_i}^{\alpha_i}(\tau)}\boldsymbol{\beta}_i$ for $i = 1$ and 2.

With the definition in (4.36), (4.35) can be rewritten as

$$\mathbf{C} = \mathbf{D}\mathbf{V}, \tag{4.37}$$

where $\mathbf{D}$ is a $(K_1 + K_2) \times (K_1 + K_2)$ diagonal matrix given by

$$\mathbf{D} = \begin{bmatrix} \text{diag}\{\boldsymbol{\gamma}_1\} & \mathbf{0} \\ \mathbf{0} & \text{diag}\{\boldsymbol{\gamma}_2\} \end{bmatrix}, \tag{4.38}$$

and $\mathbf{V}$ is a block diagonal matrix with the following formula

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{bmatrix}. \tag{4.39}$$

In (4.39), the $K_1 \times L$ matrix $\mathbf{V}_1$ and $K_2 \times L$ matrix $\mathbf{V}_2$ have the following definitions

$$\mathbf{V}_i = \begin{bmatrix} \mathbf{1} & \mathbf{b}_i & \ldots & \mathbf{b}_i^{L-1} \end{bmatrix}, \qquad \text{for } i = 1 \text{ and } 2, \tag{4.40}$$

where $\mathbf{b}_i^l$ is the l-th power of element-wise Hadamard product with the defini-

tion as $(\mathbf{A} \odot \mathbf{B})(i,j) = \mathbf{A}(i,j)\mathbf{B}(i,j)$ provided that $\mathbf{A}$ and $\mathbf{B}$ are of the same dimension. Under the assumption that SOI-1, SOI-2 and their multipath signals come from different directions, it is obvious that $\mathbf{V}_i$ for $i = 1$ and 2 are Vandermonde Matrices.

Now that matrix $\mathbf{D}$ defined in (4.38) is a diagonal matrix with nonzero diagonal elements, we have $\text{rank}(\mathbf{D}) = K_1 + K_2$. For matrix $\mathbf{V}$, because it is a block diagonal matrix, its rank is the sum of the rank of its submatrices $\mathbf{V}_1$ and $\mathbf{V}_2$, *i.e.* $\text{rank}(\mathbf{V}) = \text{rank}(\mathbf{V}_1) + \text{rank}(\mathbf{V}_2)$. To calculate the rank of matrix $\mathbf{V}$, we first prove the following theorem.

Theorem 1. *Rank($\mathbf{V}_i$) = min$\{K_i,\ L\}$, for $i = 1$ and 2.*

Proof. First, we define $P_i = \min\{K_i,\ L\}$. Because $\mathbf{V}_i$ has a dimension of $K_i \times L$, it is easy to get the following inequality,

$$\text{rank}(\mathbf{V}_i) \leq P_i. \tag{4.41}$$

Next, we choose the upper left $P_i \times P_i$ submatrix $\mathbf{V}_{i,sub}$ of $\mathbf{V}_i$, which can then be written as

$$\mathbf{V}_{i,sub} = \begin{bmatrix} 1 & b_i(1) & b_i^2(1) & \dots & b_i^{P_i-1}(1) \\ 1 & b_i(2) & b_i^2(2) & \dots & b_i^{P_i-1}(2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & b_i(P_i) & b_i^2(P_i) & \dots & b_i^{P_i-1}(P_i) \end{bmatrix}. \tag{4.42}$$

Let $\det(\mathbf{V}_{i,sub})$ denote the determinant of $\mathbf{V}_{i,sub}$, and it can be regarded as a polynomial of $b_i(1)$ up to (P_i-1)-th order. Therefore, $\det(\mathbf{V}_{i,sub})$ has P_i-1 roots at $b_i(1) = b_i(k)$ for $k = 2, \dots, P_i$, because when arbitrary two rows in a square

matrix are equivalent, the matrix is singular and its determinant turns out to be zero. As a result, $\det(\mathbf{V}_{i,sub})$ can be expressed by the following equation,

$$\det(\mathbf{V}_{i,sub}) = f(b_i(2), \ldots, b_i(P_i)) \prod_{k=2}^{k=P_i} [b_i(1) - b_i(k)], \tag{4.43}$$

where $f(b_i(2), \ldots, b_i(P_i))$ is a polynomial related with $b_i(2), \ldots, b_i(P_i)$ only. According to the assumption we made previously, we have $b_i(m) \neq b_i(n)$ for $m \neq n$. So, the determinant of $\mathbf{V}_{i,sub}$ is nonzero, and therefore, $\mathbf{V}_{i,sub}$ is nonsingular with its rank equal to P_i. Because the rank of a matrix is greater than or equal to its submatrix, we have

$$\text{rank}(\mathbf{V}_i) \geq P_i. \tag{4.44}$$

Combining (4.41) and (4.44), we can reach the conclusion of $\text{rank}(\mathbf{V}_i) = P_i = \min\{K_i,\ L\}$. □

Now we begin to obtain the conditions on N and L in order to make our proposed algorithm able to resolve the directions of SOI-1, SOI-2 and their multipath signals simultaneously.

Theorem 2. *If N and L satisfy the following conditions,*

$$N \geq K_1 + K_2, \tag{4.45}$$

$$L \geq \max\{K_1, K_2\}, \tag{4.46}$$

our proposed algorithm with spatial smoothing preprocessing technique has the ability to discriminate directions of all the signals of interests and their multipath at the same time.

Proof. In (4.33), as matrix $\tilde{\bar{\mathbf{A}}} = [\bar{\mathbf{A}}_1\ \bar{\mathbf{A}}_1]$ is of dimension $N \times (K_1 + K_2)$, it

is obvious that N need to satisfy (4.45) to make our algorithm effective in multipath environment.

From (4.34), we have $\text{rank}(\mathbf{C}) = \text{rank}(\bar{\mathbf{R}}_{ss}(\tau))$. From (4.37) along with the fact that $\mathbf{D}$ is a diagonal matrix with $\text{rank}(\mathbf{D}) = K_1 + K_2$, we have $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{V})$. With the results in Theorem 1, we can classify the relationship between K_1, K_2 and L into the following 4 cases:

1. $K_1,\ K_2 \leq L$. According to Theorem 1, we have $\text{rank}(\mathbf{V}_i) = K_i$ for $i = 1$ and 2. So $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{V}) = K_1 + K_2$.

2. $K_1,\ K_2 > L$. According to Theorem 1, we have $\text{rank}(\mathbf{V}_i) = L$ for $i = 1$ and 2. So $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{V}) = 2L < K_1 + K_2$.

3. $K_1 \geq L \geq K_2$. According to Theorem 1, we have $\text{rank}(\mathbf{V}_1) = L$ and $\text{rank}(\mathbf{V}_2) = K_2$. So $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{V}) = L + K_2 \leq K_1 + K_2$.

4. $K_2 \geq L \geq K_1$. According to Theorem 1, we have $\text{rank}(\mathbf{V}_1) = K_1$ and $\text{rank}(\mathbf{V}_2) = L$. So $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{V}) = K_1 + L \leq K_1 + K_2$.

Summarizing the 4 cases listed above, we can find that it is only when $L \geq \max\{K_1,\ K_2\}$ can the rank of matrix $\mathbf{C}$ be equal to $K_1 + K_2$, which implies that $\bar{\mathbf{R}}_{ss}(\tau)$ is nonsingular and $\bar{\mathbf{R}}_{xx}(\tau)$ has a rank no less than $K_1 + K_2$.

Moreover, after examining (4.32), and with the definitions made from (4.21)–(4.29), we can also rewrite (4.32) as

$$\begin{aligned}\bar{\mathbf{R}}_{xx}(\tau) =& r_{s_1s_1}^{\alpha_1}(\tau)\left\{\sum_{m=1}^{K_1}\sum_{n=1}^{K_1}\beta_1(m)\beta_1^*(n)\left[\frac{1}{L}\sum_{l=1}^{L}\mathrm{e}^{j(l-1)(\phi_1(n)-\phi_1(m))}\right]\bar{\mathbf{a}}_1(m)\bar{\mathbf{a}}_1^H(n)\right\} \\ &+ r_{s_2s_2}^{\alpha_2}(\tau)\left\{\sum_{m=1}^{K_2}\sum_{n=1}^{K_2}\beta_2(m)\beta_2^*(n)\left[\frac{1}{L}\sum_{l=1}^{L}\mathrm{e}^{j(l-1)(\phi_2(n)-\phi_2(m))}\right]\bar{\mathbf{a}}_2(m)\bar{\mathbf{a}}_2^H(n)\right\}\end{aligned} \tag{4.47}$$

where $\phi_i(m) = \frac{2\pi}{\lambda} d \sin\theta_i(m)$ for $m = 1, \ldots, K_i$ and $i = 1, 2$. In (4.47), the signal components in relation with m-th and n-th signal multipath signal from SOI-1 can be written as

$$\begin{bmatrix} \bar{\mathbf{a}}_1(m) & \bar{\mathbf{a}}_1(n) \end{bmatrix} \begin{bmatrix} \rho_{m,m} & \rho_{m,n} \\ \rho_{n,m} & \rho_{n,n} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{a}}_1(m) & \bar{\mathbf{a}}_1(n) \end{bmatrix}^H, \tag{4.48}$$

where

$$\begin{aligned} \rho_{m,m} &= \frac{1}{L}\beta_1(m)\beta_1^*(m) \sum_{l=1}^{L} \mathrm{e}^{j(l-1)[\phi_1(m)-\phi_1(m)]} \\ &= |\beta_1(m)|^2, \end{aligned} \tag{4.49}$$

$$\begin{aligned} \rho_{n,n} &= \frac{1}{L}\beta_1(n)\beta_1^*(n) \sum_{l=1}^{L} \mathrm{e}^{j(l-1)[\phi_1(n)-\phi_1(n)]} \\ &= |\beta_1(n)|^2, \end{aligned} \tag{4.50}$$

$$\begin{aligned} \rho_{m,n} &= \frac{1}{L}\beta_1(m)\beta_1^*(n) \sum_{l=1}^{L} \mathrm{e}^{j(l-1)[\phi_1(n)-\phi_1(m)]} \\ &= \beta_1(m)\beta_1^*(n) \frac{1 - \mathrm{e}^{jL[\phi_1(n)-\phi_1(m)]}}{L\left\{1 - \mathrm{e}^{j[\phi_1(n)-\phi_1(m)]}\right\}}, \end{aligned} \tag{4.51}$$

$$\begin{aligned} \rho_{n,m} &= \frac{1}{L}\beta_1(n)\beta_1^*(m) \sum_{l=1}^{L} \mathrm{e}^{j(l-1)[\phi_1(m)-\phi_1(n)]} \\ &= \beta_1(n)\beta_1^*(m) \frac{1 - \mathrm{e}^{jL[\phi_1(m)-\phi_1(n)]}}{L\left\{1 - \mathrm{e}^{j[\phi_1(m)-\phi_1(n)]}\right\}}. \end{aligned} \tag{4.52}$$

The determinant of the second matrix in (4.48) can be calculated by

$$\begin{aligned} &\rho_{m,m}\rho_{n,n} - \rho_{m,n}\rho_{n,m} \\ =&|\beta_1(m)|^2|\beta_1(n)|^2 \left\{1 - \frac{\left\{1 - \mathrm{e}^{jL[\phi_1(m)-\phi_1(n)]}\right\}\left\{1 - \mathrm{e}^{-jL[\phi_1(m)-\phi_1(n)]}\right\}}{L^2\left\{1 - \mathrm{e}^{j[\phi_1(m)-\phi_1(n)]}\right\}\left\{1 - \mathrm{e}^{-j[\phi_1(m)-\phi_1(n)]}\right\}}\right\} \\ =&|\beta_1(m)|^2|\beta_1(n)|^2 \left\{1 - \frac{\mathrm{sinc}^2\frac{L}{2}[\phi_1(m) - \phi_1(n)]}{\mathrm{sinc}^2\frac{1}{2}[\phi_1(m) - \phi_1(n)]}\right\}. \end{aligned} \tag{4.53}$$

The component in the square bracket in (4.53) is nonzero for $\phi_1(m) \neq \phi_1(n)$. This implies that the matrix defined in (4.48) has rank 2. Therefore, the m-th and n-th multipath of SOI-1 can be successfully decorrelated with our proposed algorithm. We can get a similar result if we apply the above analysis to SOI-2. Therefore, we can come to the conclusion that under the condition in (4.45) and (4.46), our algorithm is able to fulfill the task in multipath environment of discriminating all of the directions of SOIs with different cycle frequencies simultaneously. □

The procedures for DOA estimation by our proposed algorithm with spatial smoothing is the same as that described in section 4.4.1 except that $\hat{\mathbf{R}}_{xx}(\tau)$ in step 1 is replaced by spatially smoothing cyclic autocorrelation matrix $\hat{\bar{\mathbf{R}}}_{xx}(\tau)$ which is computed according to

$$\begin{aligned}\hat{\bar{\mathbf{R}}}_{xx}(\tau) &= \frac{1}{L}\sum_{l=1}^{L}\hat{\bar{\mathbf{R}}}_{xx,l}(\tau) \\ &= \frac{1}{L}\sum_{l=1}^{L}\left[\sum_{t=1}^{T}\bar{\mathbf{x}}_l(t+\tau/2)\bar{\mathbf{x}}_l^H(t-\tau/2)\varphi(t)\right],\end{aligned} \tag{4.54}$$

where T is the number of samples.

4.5 Simulation Results

In this section, we present some simulation results to compare our proposed algorithm with MUSIC and cyclic MUSIC algorithms to illustrate the effectiveness and superiority of our proposed algorithm.

We consider in this part a uniform linear array with eight antennas spaced by a half wavelength of the incoming signals. Among all the incoming signals, two

of them are signal of interests, and they are assumed to be BPSK cyclostationary signals with signal-to-noise ratio (SNR) 10 dB. The bit rate of the BPSK SOI-1 is 4 Mb/s, and 3Mb/s for BPSK SOI-2. Then we choose the cycle frequencies of SOIs as $\alpha_1 = 4$MHz, $\alpha_2 = 3$MHz, and τ is chosen as $0.125\mu s$. The spectral correlation density function of SOI-1 versus f and α is plotted in Figure 6.1.

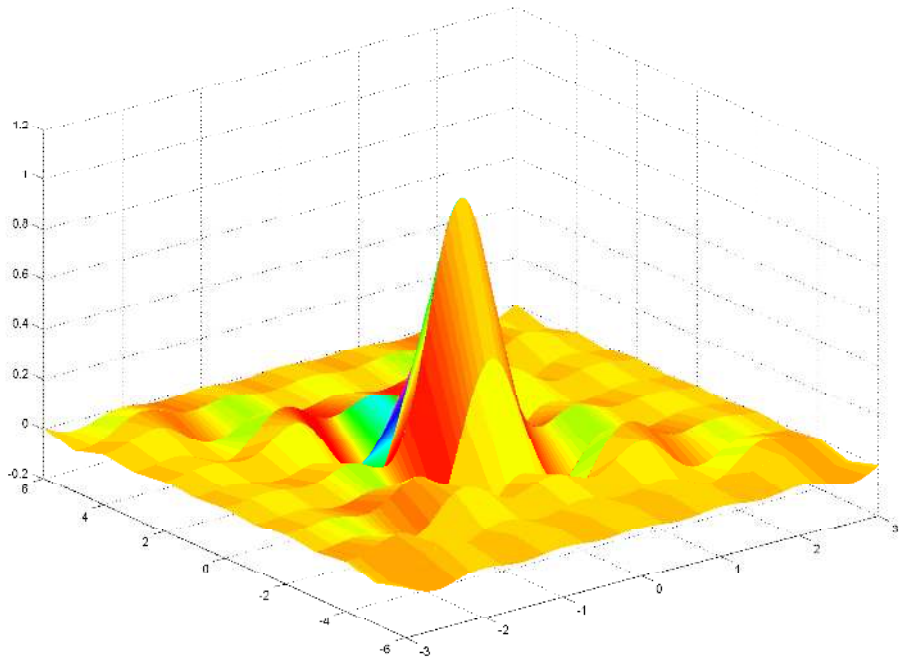


Figure 4.1: Magnitude of the spectral-correlation density for BPSK SOI-1

We further assume that the interferences are stationary and directional sources, and noise is temporally and spatially white Gaussian signals. Suppose both of the interferences and noise are uncorrelated with SOIs. Then, the above assumptions imply that both of the interferences and noises do not have nonzero cycle frequencies, and as a result, their contributions at the two cycle frequencies α_i for $i = 1$ and 2 are theoretically zero.

4.5.1 Case 1: Noncoherent Sources

In this first simulation example, we verify the superior signal selective ability of our proposed algorithm. Consider the same signal environment as that described above, and SOI-1 and SOI-2 arrive at the array from direction 0° and 30° respectively along with omnidirectional white Gaussian noise. A stationary and directional interference with SNR 10dB reaches the array from 35°. We assume the cycle frequency used in cyclic MUSIC algorithm is set to equal α_2. We compare the performance of MUSIC, cyclic MUSIC and our proposed algorithms in this situation. The resulting spatial spectrum is shown in Figure 4.2, and the averaged estimates and the root mean square error (RMSE) over 100 trial runs are compared in Table 4.1.

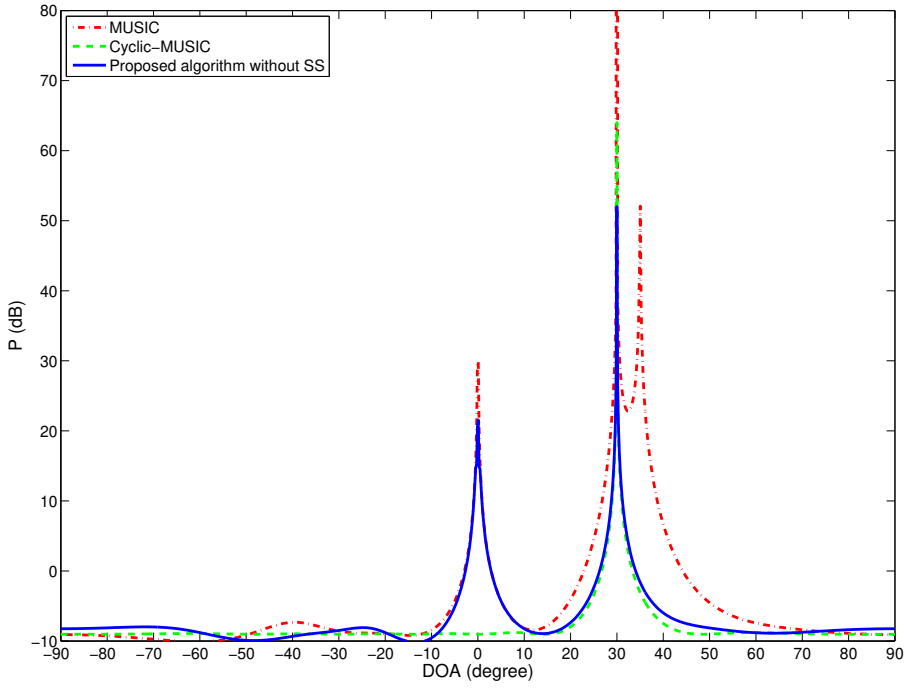


Figure 4.2: Spatial spectrum for case 1 with two cyclostationary SOIs of 10dB from 0° and 30°, and one interference of 10dB from 35°.

Table 4.1: Performance comparison of DOA estimation by MUSIC, cyclic MUSIC and proposed algorithm in case 1

True		$\theta_1 = 0°$	$\theta_1 = 30°$
MUSIC	Mean	-0.102	30.075
	RMSE	0.116	0.088
Cyclic MUSIC	Mean	0.150	30.125
	RMSE	0.164	0.162
Proposed Algorithm	Mean	0.174	30.167
	RMSE	0.201	0.192

The rank of signal subspace estimated by MDL criterion is 3 for MUSIC algorithm, 1 for cyclic MUSIC algorithm and 2 for our proposed algorithm. As depicted in Figure 4.2, the ordinary MUSIC algorithm resolves all the signals present in the array, including both the SOIs and interferences. Although MUSIC algorithm shows its ability of correct discrimination of all the sources, we still need extra efforts to distinguish the SOIs from interferences. As to cyclic MUSIC algorithm, due to its signal selectivity ability, it ignores the interferences. But it cannot estimate the DOAs for both of the SOIs simultaneously. We have to re-apply the cyclic MUSIC algorithm with a different cycle frequency to resolve the other SOI from 0°. Our proposed algorithm overcomes the problems encountered by MUSIC and cyclic MUSIC algorithms, and allows to perfectly select the two SOIs with different cycle frequencies and ignore the interferences by exploiting the two cycle frequencies at the same time.

4.5.2 Case 2: Coherent Sources

Now we begin to consider the case where SOIs reach the array with multipath signals. The signal environment is the same as that in the previous case except

that the SOI-1 arrives at the array from $-15°$ along with a multipath signal from $50°$ and multipath coefficients $\beta_1(2) = 0.7 + j0.2$, while the SOI-2 comes from $0°$ with a multipath signal from $30°$ and multipath coefficient $\beta_2(2) = 0.8 + j0.3$. Interference is stationary and directional signal from $60°$. We use MUSIC, MUSIC with spatial smoothing, cyclic MUSIC, cyclic MUSIC with spatial smoothing and our proposed algorithm with spatial smoothing in this case for comparison. Here the cycle frequency used in cyclic MUSIC and cyclic MUSIC with spatial smoothing is chose as $\alpha_1 = 4$MHz.

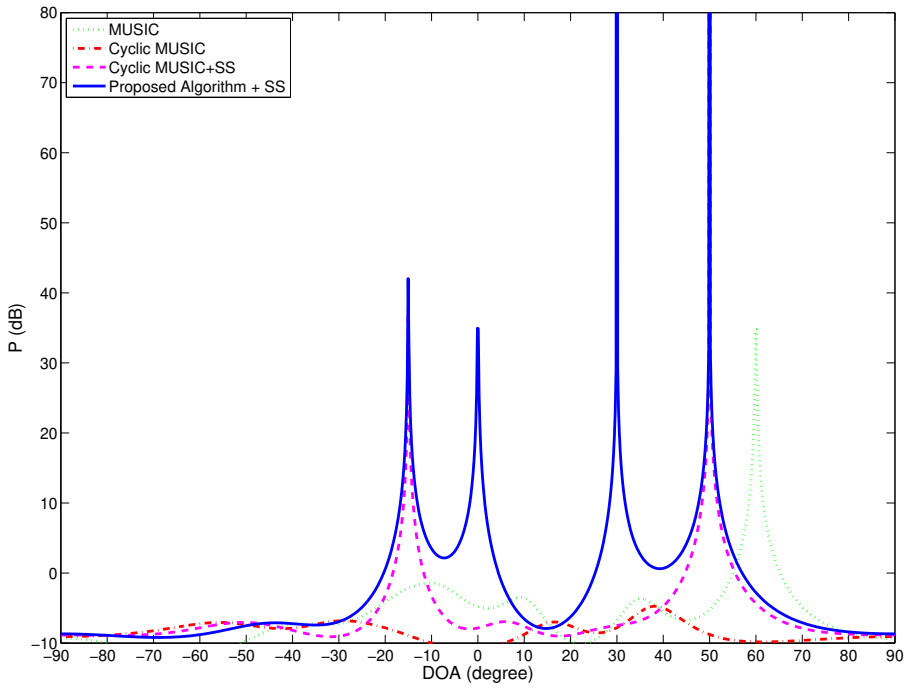


Figure 4.3: Spatial spectrum for case 2 containing two cyclostationary SOIs each with 10dB. SOI-1 comes from $-15°$ with a multipath signal from $50°$ and multipath coefficient $\beta_1(2) = 0.7 + j0.2$. SOI-2 reaches array from $0°$ and one interference with 10dB with a multipath signal from $30°$ and multipath coefficient $\beta_2(2) = 0.8 + j0.3$. The interference is stationary with 10dB and has a DOA of $60°$.

From Figure 4.3 we can see that MUSIC and cyclic MUSIC algorithms fail

Table 4.2: Performance comparison of DOA estimation by MUSIC, cyclic MUSIC and proposed algorithm with spatial smoothing in case 2

		BPSK SOI-1		BPSK SOI-2	
True		$\theta_1(1) = -15°$	$\theta_1(2) = 50°$	$\theta_2(1) = 0°$	$\theta_2(2) = 30°$
MUSIC	Mean	-14.895	50.107	0.116	30.102
	RMSE	0.118	0.125	0.112	0.129
Cyclic MUSIC	Mean	-15.150	50.505	0.165	30.245
	RMSE	0.188	0.484	0.192	0.460
Proposed Algorithm	Mean	-15.166	49.455	-0.172	30.354
	RMSE	0.204	0.693	0.218	0.589

to estimate the DOAs of SOIs and their multipath signals accurately because of the rank deficit of the signal correlation matrix and signal cyclic correlation matrix. When these two algorithms are used with spatial smoothing preprocessing technique, both of them can give us correct DOA estimations of SOIs. For MUSIC, besides the DOAs of two SOIs and their multipath signals, it also provides us with an undesired information from interference. For cyclic MUSIC, it can only give us the directions from SOI-1 and its multipath signal, requiring to repeat the algorithm once more for detecting the SOI-2. Our proposed algorithm with spatial smoothing performs the best among all the aforementioned algorithms in that it provides with us the only directions of our interests correctly and exactly. In Table 4.2, the means and RMSE of the arrival angle estimations are given.

4.5.3 Effect of SNR on Performance of DOA Estimation

Figure 4.4 shows that under signal environment in case 1 , how the input SNR affects the performance of DOA estimation of SOI-1 by evaluating the root mean

square error over 200 independent trials and then making an average of all the results.

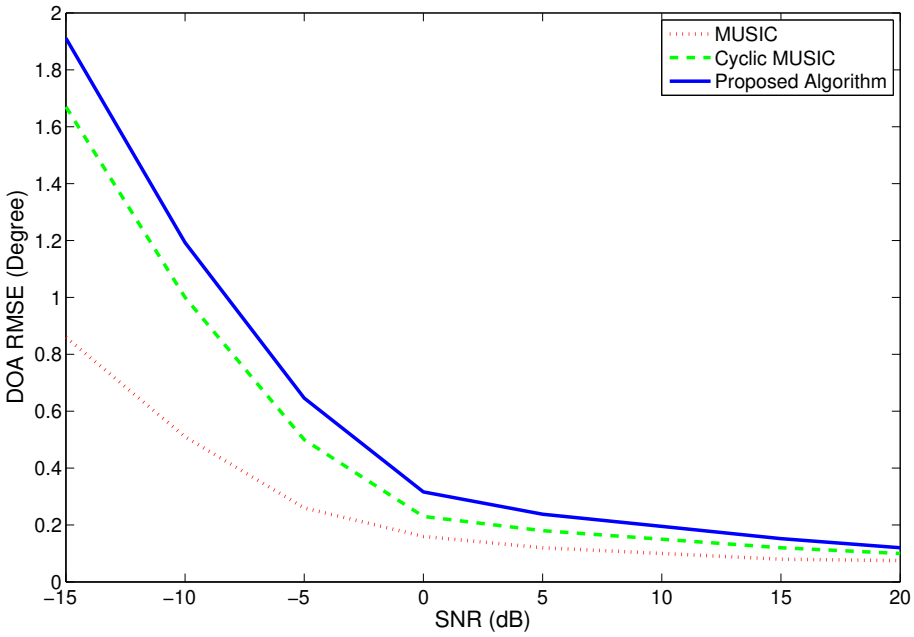


Figure 4.4: Average RMSE versus SNR for environment in case 1

We can find in Figure 4.4 that our proposed algorithm has a similar performance with cyclic MUSIC, but the RMSE obtained by MUSIC algorithm is the lowest among the three algorithms. This is reasonable because MUSIC utilizes the ordinary correlation matrix for direction finding, which is better estimated and contains more of the estimated second-order signal components than that of cyclic correlation matrices used by cyclic MUSIC and our proposed method. This can be easily confirmed if we compare the level of spectral correlation density in Figure 6.1, where the signal strength at $f = 0$ and $\alpha = 0$ is the strongest in the $f - \alpha$ plane. However, as the input SNR increasing, the difference in RMSE between MUSIC and our proposed algorithm is diminishing. So it is preferable to choose our proposed algorithm for DOA estimation in

high SNR environment as it can overcome the problems of non-signal selectivity with MUSIC but yet achieve a comparably satisfactory performance as that with MUSIC.

Chapter 5

Noniterative Beam Pattern Synthesis Techniques

5.1 Introduction

The applications of sensor or antenna arrays can be found in many areas such as radar, sonar and wireless communications. By linearly combining sensor outputs, a receiver array can achieve spatial filtering, that is, an enhancement of signals coming from desired directions and an attenuation of signals coming from undesired directions. A linear-combining network can also be used as a feed network for a transmission array, distributing relatively high amount of power in desired direction and low power in undesired directions. Therefore, the formation of array pattern is important for both the uplink and downlink transmission in wireless communications. To synthesis an array pattern is to find a set of weights such that the array pattern has a desired shape, *e.g.* a maximum at the desired direction with a certain beamwidth and a certain sidelobe rolloff.

This set of weights is sometimes called the array taper or the weight vector.

Generally speaking, there are two types of array pattern synthesis algorithms: non-iterative and iterative, which we will elaborate in chapter 5 and chapter 6 respectively. In this chapter, a brief review about several non-iterative beam pattern synthesis algorithms will be given in the next section. Based on the two algorithms introduced in section 5.2, we, in section 5.3, put forward an improved algorithm with generalized sidelobe canceler (GSC) structure by using the cyclostationary property of the transmitting signals. This will ensure that the blocking matrix spans the interference-plus-noise subspace so that we can get a better control of the beampattern, especially in the interference-limited and time-varying environment and when the interference is close to the look direction. We also consider the case when multipath exists in the desired signal, and put forward an improved structure to solve this problem. Numerical results are provided in section 5.4 to show that the our proposed beampattern synthesis technique is able to reject strong directional interferences as compared to other type of GSC-based structure.

5.2 Existing Methods

5.2.1 Overview

In this section, we consider several non-iterative approaches to the selection of weighting vector for linear arrays. All the methods introduced here reach their optimal solution in one step. However, such an advantage may be at the cost of heavy computational load or the requirement of certain *a priori* knowledge of signal's characteristics.

The algorithms which we are going to introduce are:

Delay-and-sum beamformer: The delay and sum beamformer, also known as conventional beamformer, is based on the idea that if an antenna array is being used, and both of the direction of desired signal and array geometry are known, then the output of each antenna will be the same, except that each one will be delayed by a different amount. So, if the output of each antenna is delayed appropriately then we add all the outputs together the signal that was propagating through the array will reinforce, while noise will tend to cancel.

Minimax design: The array pattern with low sidelobes is often of interest in practice. An important type of pattern is the minimax pattern in which the maximum sidelobe level of the synthesized pattern is minimized. Dolph [61] published a well-known paper in which he identified the array pattern with a Chebyshev polynomial to obtain the current distribution of an uniform linear array. The resulting pattern has uniform sidelobes and is optimal in the sense that the sidelobe levels are the lowest with the main beam width fixed. However, Dolph's approach can only be applied to the ULAs.

Quadratic programming: Ng, Er and Kot developed a non-iterative method to minimize the L_2 norm of the error between the synthesized pattern and a desired pattern using quadratic programming [16]. This method approximates the integral of the squared pattern error, *i.e.* the L_2 norm squared, with a matrix expression that is quadratic in the weight vector, and solves this expression subject to linear constraints. The optimal weight vector can be obtained in one step instead of through an iterative process. Er

in [17] and Sim in [4] also proposed some similar methods based on the idea of quadratic programming, which we will introduce in more detail in the following subsections.

5.2.2 Method I

This method is proposed by Er [17]. Its major idea is to form a quadratically constrained minimization problem by minimizing the mean square error between the array response and the desired response over the mainlobe region while keeping the level of sidelobe region under certain prescribed level.

Now we first define some variables that will be used in this chapter.

Consider a ULA with M antenna elements. The normalized mean-square error ε between the desired response and the response of the array system over certain mainlobe width is given by

$$\varepsilon = \frac{1}{\beta}\int_{\theta_0-\Delta\theta/2}^{\theta_0+\Delta\theta/2} |p_r(\theta) - \mathbf{w}^H\mathbf{a}(\theta)|^2 \mathrm{d}\theta = \mathbf{w}^H\mathbf{Q}_1\mathbf{w} - \mathbf{w}^H\mathbf{p} - \mathbf{p}^H\mathbf{w} + 1, \quad (5.1)$$

where $p_r(\theta)$ is the desired response, θ_0 is the look direction, $\Delta\theta$ is the mainlobe width of interest, and $\mathbf{Q}_1$ is the $M \times M$ dimensional Hermitian matrix given by

$$\mathbf{Q}_1 = \frac{1}{\beta}\int_{\theta_0-\Delta\theta/2}^{\theta_0+\Delta\theta/2} \mathbf{a}(\theta)\mathbf{a}^H(\theta)\mathrm{d}\theta, \quad (5.2)$$

and $\mathbf{p}$ is the M-dimensional vector given by

$$\mathbf{p} = \frac{1}{\beta}\int_{\theta_0-\Delta\theta/2}^{\theta_0+\Delta\theta/2} p_r^*(\theta)\mathbf{a}(\theta)\mathrm{d}\theta, \quad (5.3)$$

where β is a normalization scalar,

$$\beta = \int_{\theta_0-\Delta\theta/2}^{\theta_0+\Delta\theta_2} p_r^*(\theta) p_r(\theta) \mathrm{d}\theta. \tag{5.4}$$

The mean-square value of the array response over the sidelobe regions is given by

$$\rho = \frac{1}{\Delta\theta_1}\left[\int_{-\frac{\pi}{2}}^{-\theta_1} |g(\theta)|^2 \mathrm{d}\theta + \int_{\theta_1}^{\frac{\pi}{2}} |g(\theta)|^2 \mathrm{d}\theta\right] = \mathbf{w}^H \mathbf{Q}_2 \mathbf{w}, \tag{5.5}$$

where $\Delta\theta_1 = \pi/2 - \theta_1$, $g(\theta) = \mathbf{w}^H \mathbf{a}(\theta)$ is array's beam pattern response and $\mathbf{Q}_2$ is the $M \times M$ dimensional Hermitian matrix given by

$$\mathbf{Q}_2 = \frac{1}{\Delta\theta_1}\left[\int_{-\pi/2}^{-\theta_1} \mathbf{a}(\theta)\mathbf{a}^H(\theta)\mathrm{d}\theta + \int_{\theta_1}^{\pi/2} \mathbf{a}(\theta)\mathbf{a}^H(\theta)\mathrm{d}\theta\right]. \tag{5.6}$$

We can rewrite (5.1) as

$$\varepsilon = (\mathbf{w}_0 - \mathbf{w})^H \mathbf{Q}_1 (\mathbf{w}_0 - \mathbf{w}) + \alpha_0, \tag{5.7}$$

where $\alpha_0 = 1 - \mathbf{w}_0^H \mathbf{Q}_1 \mathbf{w}_0$ is a scalar and $\mathbf{w}_0$ is the M-dimensional vector satisfying $\mathbf{Q}_1 \mathbf{w}_0 = \mathbf{p}$.

In method I, the optimal weight vector $\hat{\mathbf{w}}_1$ is the solution to the following constrained optimization problem:

$$\min\ \mathbf{w}_1^H \mathbf{Q}_1 \mathbf{w}_1 - \mathbf{w}_1^H \mathbf{p} - \mathbf{p}^H \mathbf{w}_1 + 1 \qquad \text{subject to } \mathbf{w}_1^H \mathbf{Q}_2 \mathbf{w}_1 \le \xi. \tag{5.8}$$

With the aid of Lagrangian multiplier γ, the optimal weight vector can then be found as:

$$\hat{\mathbf{w}}_1(\gamma) = (\mathbf{Q}_1 + \gamma \mathbf{Q}_2)^{-1} \mathbf{p}, \tag{5.9}$$

and the optimal Lagrangian multiplier $\hat{\gamma}$ is the root of the following equation

$$\mathbf{p}^H(\mathbf{Q}_1 + \gamma\mathbf{Q}_2)^{-1}\mathbf{Q}_2(\mathbf{Q}_1 + \gamma\mathbf{Q}_2)^{-1}\mathbf{p} = \xi. \tag{5.10}$$

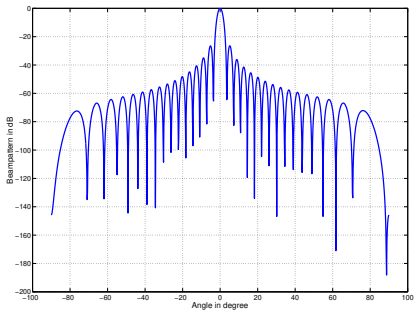
Now, we will give simulation examples to illustrate the performance of method I. Suppose we employ a 32-element ULA with half-wavelength spacing. The desired response $p_r(f_0, \theta)$ has the pattern of linear constrained minimum variance (LCMV) beamformer. The look direction is set to $0°$, the mainlobe of interest is $[-3.0°, \ +3.0°]$ and the sidelobe regions are $[-90°, \ -3.6°]$ and $[+3.6°, \ +90°]$. Figure 5.1 shows the the directional patterns using the quadratic constraint method I with error limit $\xi = 10^{-2}$ and $\xi = 10^{-4}$ respectively. We can see by choosing different values of ξ, a trade-off can be made in terms of beamwidth and sidelobe level in the achieved beam pattern. In this case, the bigger the error limit, the higher the sidelobe level.

5.2.3 Method II

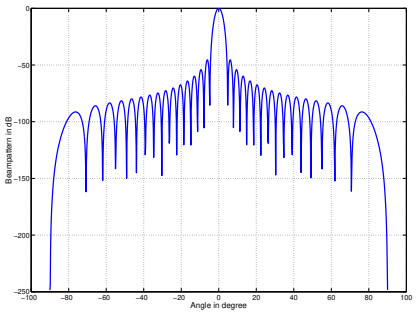
The main idea of another effective pattern control algorithm is to match the array response to a desired response over mainlobe width and minimize the mean square value of the array response over sidelobe regions. The optimum weight vector $\hat{\mathbf{w}}_2$ is the solution to the following constrained optimization problem

$$\min \ \mathbf{w}_2^H\mathbf{Q}_2\mathbf{w}_2 \qquad \text{subject to } \mathbf{w}_2^H\mathbf{Q}_1\mathbf{w}_2 - \mathbf{w}_2^H\mathbf{p} - \mathbf{p}^H\mathbf{w}_2 + 1 \leq \varepsilon, \tag{5.11}$$

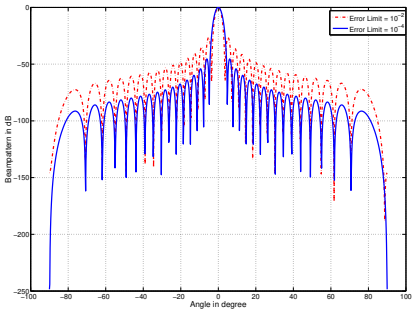
where $0 \leq \varepsilon < 1$ defines a normalized error limit over the mainlobe. From (5.7), we can have $\varepsilon = (\mathbf{w}_0 - \mathbf{w}_2)^H\mathbf{Q}_1(\mathbf{w}_0 - \mathbf{w}_2) + \alpha_0$. We let $\epsilon = \varepsilon - \alpha_0$, then (5.11) can be expressed as $(\mathbf{w}_0 - \mathbf{w}_2)^H\mathbf{Q}_1(\mathbf{w}_0 - \mathbf{w}_2) \leq \epsilon$. With the same approach in



(a) Method I, error limit $\xi = 10^{-2}$



(b) Method I, error limit $\xi = 10^{-4}$



(c) Comparison of method I with different values of error limit

Figure 5.1: Beam patterns of method I

subsection 5.2.2, the optimal weight vector can be written as

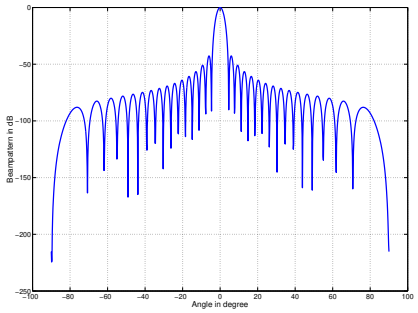
$$\hat{\mathbf{w}}_2(\alpha) = (\mathbf{Q}_1 + 1/\alpha\mathbf{Q}_2)^{-1}\mathbf{p}, \tag{5.12}$$

where the optimal lagrangian multiplier $\hat{\alpha}$ is the root of the following equation,

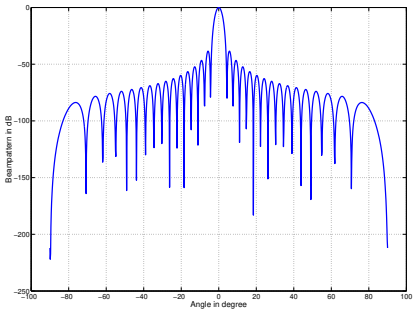
$$\mathbf{w}_0^H\mathbf{Q}_2(\mathbf{Q}_2 + \alpha\mathbf{Q}_1)^{-1}\mathbf{Q}_1(\mathbf{Q}_2 + \alpha\mathbf{Q}_1)^{-1}\mathbf{Q}_2\mathbf{w}_0 = \epsilon. \tag{5.13}$$

Now, we give examples for method II in Figure 5.2. The assumption is the same as that in the previous subsection. We can have the same observation that different error limit results in different sidelobe level and main beamwidth. But

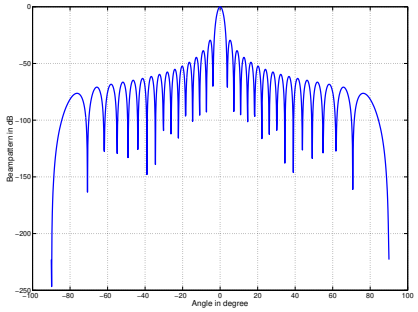
this time, the smaller the error limit, the higher the sidelobe level.



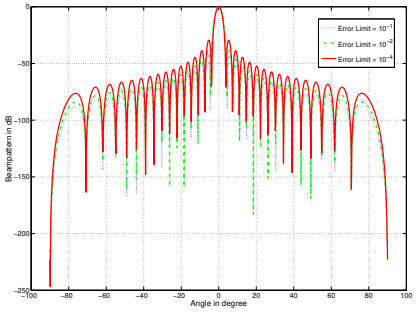
(a) Method II, error limit=10^{-1}



(b) Method II, error limit=10^{-2}



(c) Method II, error limit=10^{-4}



(d) Comparison of method II with different values of error limit

Figure 5.2: Beam patterns of method II

5.2.4 Relationship Between Method I and Method II

Comparing (5.9) with (5.12), it is easy to find out that if $1/\hat{\alpha} = \hat{\gamma}$, then the two optimum weights are identical. Since $\hat{\alpha}$ is the root of (5.13), it is possible to find an ε such that for a given ξ, $1/\hat{\alpha} = \hat{\gamma}$.

Note that (5.13) can also be expressed as

$$1/\hat{\alpha}\mathbf{p}^H \left(\mathbf{Q}_1 + 1/\hat{\alpha}\mathbf{Q}_2\right)^{-1} \mathbf{Q}_2 \left(\mathbf{Q}_1 + 1/\hat{\alpha}\mathbf{Q}_2\right)^{-1} \mathbf{p} + \mathbf{p}^H \left(\mathbf{Q}_1 + 1/\hat{\alpha}\mathbf{Q}_2\right)^{-1} \mathbf{p} = 1 - \varepsilon. \tag{5.14}$$

Substituting $\hat{\gamma} = 1/\hat{\alpha}$ into (5.14) and with some further calculation, we can obtain

$$\varepsilon = 1 - \hat{\gamma}\xi - \mathbf{p}^H \left(\mathbf{Q}_1 + \hat{\gamma}\mathbf{Q}_2\right)^{-1} \mathbf{p}. \tag{5.15}$$

Although the relationship between ξ and ε is not simple, the significance of (5.15) is that for a given design value of ξ, $\hat{\gamma}$ can be determined by (5.10), and (5.15) can be used to calculate the mean square error over the mainlobe of interest. Table 5.2.4 gives a numerical relationship between ξ and ε which

Table 5.1: Relationship between ξ and ε

ξ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
ε	0.00833	0.006537	0.14316	0.60755	0.86383

reveals that a bigger ξ corresponds to a smaller ε. It coincides exactly with what we have discovered in Figure 5.1 and Figure 5.2. In Figure 5.1, a bigger ξ results in a higher sidelobe and wider main beamwidth which can also be achieved with a smaller ε in Figure 5.2.

5.3 Proposed Non-iterative Beam Pattern Synthesis Algorithms

Sim proposed a algorithm for beam pattern synthesis with GSC structure in [4]. Method I was used in the main beam to shape the quiescent beam pattern, and the blocking matrix $\mathbf{B}$ in auxiliary beam was chosen as the eigenvectors corresponding to the n_0 smallest eigenvalues of $\mathbf{R}_{xx}$ which is the correlation matrix of input signals $\mathbf{x}(k)$. Its structure is shown in Figure 5.3. This blocking matrix spans the noise subspace when the interference is much stronger than the desired signal, so it is unable to cancel the effect of strong interference signals

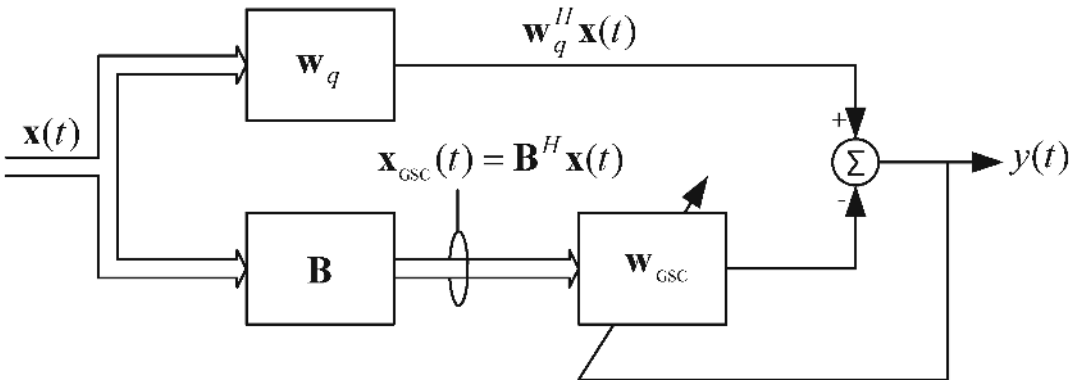


Figure 5.3: GSC Structure

in such circumstances.

5.3.1 Proposed Algorithm for Environment Without Multipath

Consider a ULA that receives the planar narrow band signal from far field, which can be expressed as

$$\mathbf{x}(t) = \mathbf{a}(f_0, \theta_0)s(t) + \mathbf{n}(t), \tag{5.16}$$

where $\mathbf{a}(f_0, \theta_0)$ is steering vector of signal of interest (SOI) from θ_0 at frequency f_0, $s(t)$ is the SOI which exhibits spectral correlation at a particular α, the vector $\mathbf{n}(t)$ is interference and noise vector which has different cycle frequencies or does not exhibit cyclostationarity.

Instead of calculating the correlation matrix of the received signal in the conventional method of choosing $\mathbf{B}$, we calculate the cyclic correlation matrix

which is estimated by

$$\begin{aligned}\mathbf{R}_{xx}^{\alpha}(\tau) =&\langle \mathbf{x}(t+\tau/2)\mathbf{x}^{H}(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty} \\ \approx&\mathbf{a}(f_0,\theta_0)r_{ss}^{\alpha}(\tau)\mathbf{a}^{H}(f_0,\theta_0),\end{aligned} \tag{5.17}$$

and $r_{ss}^{\alpha}(\tau)$ is the cyclic correlation value of the desired signal $s(t)$. Here $\mathbf{n}(t)$ disappears because evaluating the cyclic correlation at α retains only SOI. Applying singular value decomposition (SVD) on $\mathbf{R}_{xx}^{\alpha}(\tau)$, we obtain

$$\mathbf{R}_{xx}^{\alpha}(\tau) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{H} = [\mathbf{U}_s\ \mathbf{U}_n]\begin{bmatrix}\mathbf{\Sigma}_s & 0\\ 0 & \mathbf{\Sigma}_n\end{bmatrix}\begin{bmatrix}\mathbf{V}_s^{H}\\ \mathbf{V}_n^{H}\end{bmatrix}, \tag{5.18}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{\Sigma}$ is a real non-negative diagonal matrix. Since the rank of $\mathbf{R}_{xx}^{\alpha}(\tau))$ is one, we have $\mathbf{a}^{H}(f_0,\theta_0)\mathbf{U}_n = \mathbf{0}^{T}$. So choosing $\mathbf{U}_n$ as blocking matrix can effectively block away SOI while keeping the interference and noise in auxiliary beam regardless of the strength of undesired signals.

However, in multipath environment, this algorithm is unable to distinguish the SOI from its multipath components, and consequently, cannot generate nulls in the multipath direction. In the next part, we will give another algorithm for improvement.

5.3.2 Proposed Algorithm for Environment With Multipath

Let us first note one simple but important property of cyclic correlation.

Lemma 1. *If $x(t)$ is a cyclostationary process with cyclic correlation function*

$r_{xx}^{\alpha}(\tau)$ and $y(t) = x(t+T)$, then $r_{yy}^{\alpha}(\tau) = r_{xx}^{\alpha}(\tau)e^{j2\pi\alpha T}$.

Consider the same situation as before, except that there are K coherent signals. The signal received by the i-th antenna is

$$x_i(t) = \sum_{k=1}^{K} \beta_k s(t - \tau_k + (i-1)\triangle_k) + n_i(t), \tag{5.19}$$

where the β_k and τ_k are the attenuation in amplitude and phase delay associated with the k-th multipath and $\triangle_k$ is inter-element phase shift which can be calculated as $\triangle_k = \frac{d}{\lambda}\sin\theta_k$. With the definition in (4.5), we have the (m,n)-th element of the cyclic correlation matrix $\mathbf{R}_{xx}^{\alpha}(\tau)$ as

$$\begin{aligned} r_{s_m s_n}^{\alpha}(\tau) =& \langle s_m(t+\tau/2)s_n^*(t-\tau/2)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty} \\ =& \langle \sum_{k=1}^{K} \beta_k s(t+\tau/2-\tau_k+(m-1)\triangle_k)\times \\ & \sum_{p=1}^{K} \beta_p s(t-\tau/2-\tau_p+(n-1)\triangle_p)\mathrm{e}^{-j2\pi\alpha t}\rangle_{\infty}. \end{aligned} \tag{5.20}$$

With Lemma 1, we can further simplify the above equation as

$$\begin{aligned} r_{s_m s_n}^{\alpha}(\tau) =& \sum_{k=1}^{K}\sum_{p=1}^{K} \beta_k\beta_p r_s^{\alpha}(\tau+\tau_p-\tau_k+(m-1)\triangle_k-(n-1)\triangle_p)\times \\ & \mathrm{e}^{j\pi\alpha(-\tau_p-\tau_k+(m-1)\triangle_k+(n-1)\triangle_p)} \\ =& \sum_{k=1}^{K} \beta_k \mathrm{e}^{-j\pi\alpha\tau_k}\mathrm{e}^{j\pi\alpha(m-1)\triangle_k} r_s^{\alpha}(\tau) \sum_{p=1}^{K} \beta_p \mathrm{e}^{-j\pi\alpha\tau_p}\mathrm{e}^{j\pi\alpha(n-1)\triangle_p}. \end{aligned} \tag{5.21}$$

We define two vectors

$$\mathbf{z}_i = [\mathrm{e}^{j\pi\alpha(i-1)\triangle_1}, \ \ldots, \ \mathrm{e}^{j\pi\alpha(i-1)\triangle_K}]^T, \tag{5.22}$$

$$\mathbf{q} = \left[\beta_1 \mathrm{e}^{-j\pi\alpha\tau_1}, \ldots, \beta_K \mathrm{e}^{-j\pi\alpha\tau_K}\right]^T, \tag{5.23}$$

then we have

$$r^{\alpha}_{s_m s_n}(\tau) = \mathbf{z}_m^T \mathbf{q} r_s^{\alpha}(\tau) \mathbf{q}^T \mathbf{z}_n = \mathbf{z}_m^T \mathbf{F} \mathbf{z}_n, \tag{5.24}$$

where $\mathbf{F} = \mathbf{q} r_{ss}^{\alpha}(\tau) \mathbf{q}^T$. Combining all the elements to form cyclic correlation matrix $\mathbf{R}_{xx}^{\alpha}(\tau)$, we obtain

$$\mathbf{R}_{xx}^{\alpha}(\tau) = [\mathbf{a}(\alpha,\theta_1) \ \ldots \ \mathbf{a}(\alpha,\theta_K)] \mathbf{F} [\mathbf{a}(\alpha,\theta_1) \ \ldots \ \mathbf{a}(\alpha,\theta_K)]^T = \mathbf{U}_K \mathbf{F} \mathbf{U}_K^T,$$

where $\mathbf{U}_K = [\mathbf{a}(\alpha,\theta_1) \ldots \mathbf{a}(\alpha,\theta_K)]$ is a $M \times K$ matrix. As $\mathrm{rank}(\mathbf{F}) = 1$ in multipath environment, performing SVD will not be able to resolve the multipath component from the direct path component. So spatial smoothing (SS) technique is employed to de-correlate signals first.

We divide the array into L overlapping subarrays, with N antennas in each subarray, and $\min(L, N) \geq K$, so $L + N - 1 = M$. The l-th to $N + l - 1$-th antennas form the l-th subarray, and its received signal is

$$\mathbf{x}^{(l)}(t) = [x_l(t) \ \ldots \ x_{N+l-1}(t)]^T, \tag{5.25}$$

with cyclic correlation matrix as

$$\mathbf{R}_{xx}^{(l)\alpha}(\tau) = \mathbf{U}_K^{(l)} \mathbf{F} \mathbf{U}_K^{(l)T}, \tag{5.26}$$

where $\mathbf{U}_K^{(l)} = \bar{\mathbf{U}}_K \mathbf{D}^{(l-1)}$, and here $\bar{\mathbf{U}}_K$ represents the first N rows in $\mathbf{U}_K$ and $\mathbf{D}$ is a diagonal matrix which is defined as $\mathbf{D} = \text{diag}\{\text{e}^{j\pi\alpha\triangle_1} \ \ldots \ \text{e}^{j\pi\alpha\triangle_K}\}$. Although $\mathbf{F}$ is still a rank one matrix, by adding L cyclic correlation matrix altogether, we obtain the new cyclic correlation matrix after spatial smoothing,

$$\bar{\mathbf{R}}_{xx}^{\alpha}(\tau) = \frac{1}{L}\sum_{l=1}^{L} \mathbf{R}_{xx}^{(l)\alpha}(\tau) = \bar{\mathbf{U}}_K \left(\sum_{l=1}^{L} \mathbf{D}^{(l-1)} \mathbf{F} \mathbf{D}^{(l-1)T} \right) \bar{\mathbf{U}}_K^T. \tag{5.27}$$

As we have the assumption that $\min(L, N) \geq K$, we are sure this time $\bar{\mathbf{R}}_{xx}^{\alpha}(\tau)$ has full rank. Then we can apply SVD on $\bar{\mathbf{R}}_{xx}^{\alpha}(\tau)$ and get the result

$$\bar{\mathbf{R}}_{xx}^{\alpha}(\tau) = \bar{\mathbf{U}}\bar{\mathbf{\Sigma}}\bar{\mathbf{V}}^H = \begin{bmatrix} \bar{\mathbf{U}}_s & \bar{\mathbf{U}}_n \end{bmatrix} \begin{bmatrix} \bar{\mathbf{\Sigma}}_s & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{\Sigma}}_n \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}_s^H \\ \bar{\mathbf{V}}_n^H \end{bmatrix}, \tag{5.28}$$

where $\bar{\mathbf{\Sigma}}_s = \text{diag}\{\eta_1, \ \eta_2, \ \ldots, \ \eta_K\}$ denotes the singular value matrix for the K multipath signals, with $\eta_i \geq \eta_j$ for $i < j$; and $\bar{\mathbf{U}}_s = [\mathbf{u}_{s_1}, \ \mathbf{u}_{s_2}, \ \ldots, \ \mathbf{u}_{s_K}]$ are the corresponding left singular vectors. The signal path with largest singular value holds the most energy, and its direction is our interest. So we choose the blocking matrix as

$$\mathbf{B} = [\mathbf{u}_{s_2}, \ \ldots, \ \mathbf{u}_{s_K}, \bar{\mathbf{U}}_n]. \tag{5.29}$$

This new blocking matrix can effectively screen the SOI and enable the multipath, interference and noise components to pass into the auxiliary beam. The improved beamforming structure is shown in Figure 5.4.

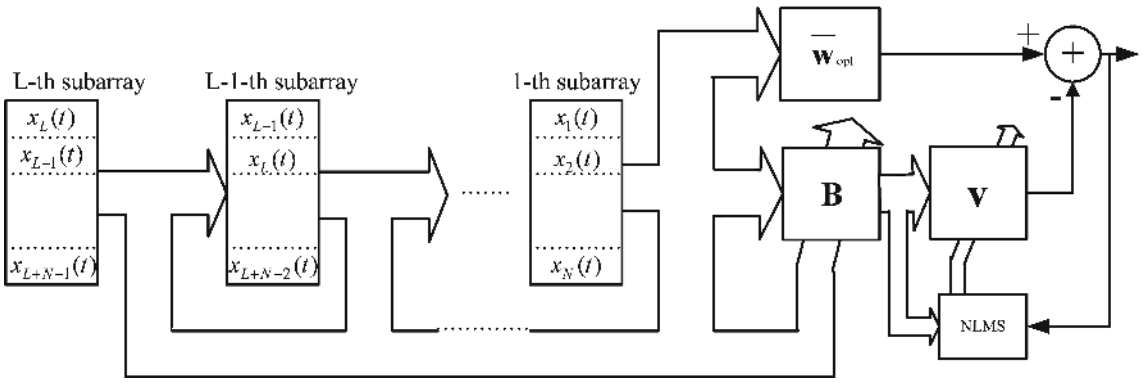
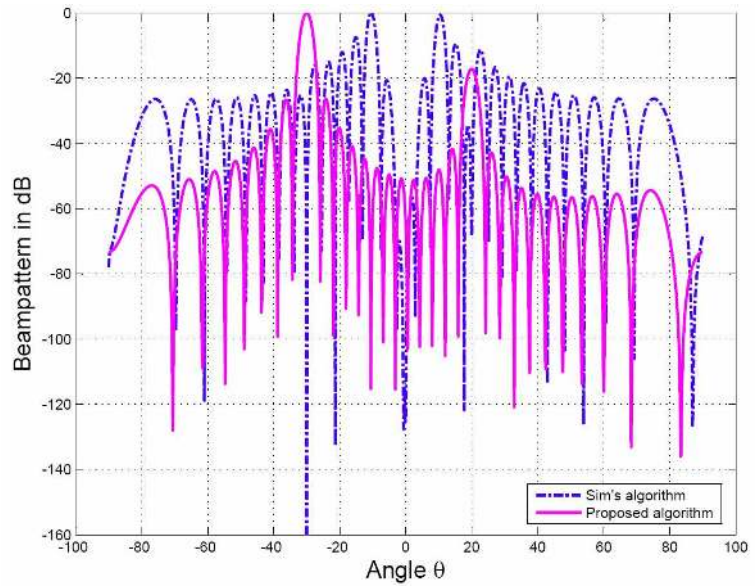


Figure 5.4: Detailed structure for our proposed algorithm

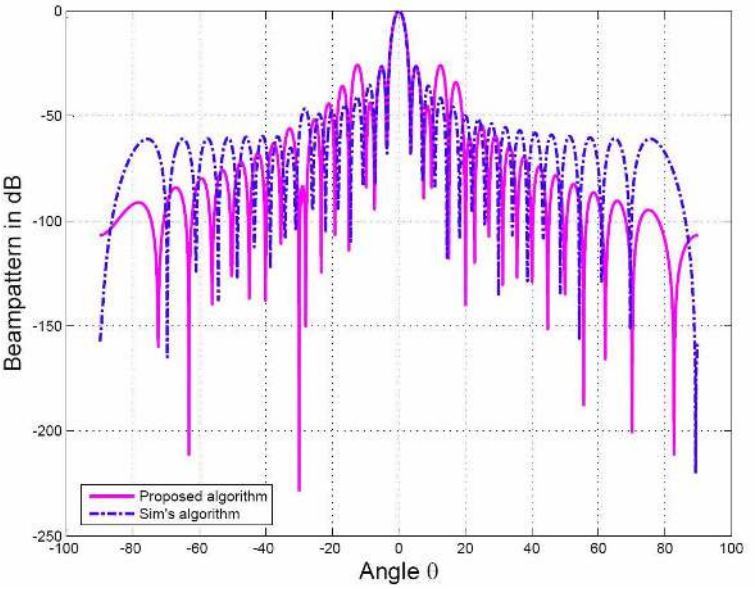
5.4 Simulation Results

Case 1 We employ a 32-element uniform linear array with half-wavelength spacing to illustrate the performance improvement of our algorithm. A desired beampattern response $p_r(f_0, \theta)$ of LCMV beamformer is used. The look direction is set to 0°, the mainlobe of interest is $[-3.0°, \ +3.0°]$ and the sidelobe regions are $[-90°, \ -3.6°]$ and $[+3.6°, \ +90°]$. The environment consists of two uncorrelated 16-QAM directional interferences of power 30 dB with carrier frequency of 100MHz and 200MHz, and from $-30°$ and 20° respectively. The desired BPSK signal of 0 dB, with baud rate= 16kbps, carrier frequency= 0, comes from broadside at 0°. No multipath is present in the system. The signal to noise ratio is 15 dB, p is set to 3 and n_0 is set to 5. As we can see from Figure 5.5(a), there are more interference components at the output of the auxiliary beam for our improved algorithm than for the algorithm in [4]. This implies that more interferences are being subtracted in our proposed algorithm, which results in a much deeper null in the interference direction as shown in Figure 5.5(b).

Case 2 Now, consider the same situation as aforementioned except there



(a) Comparison of auxiliary beam

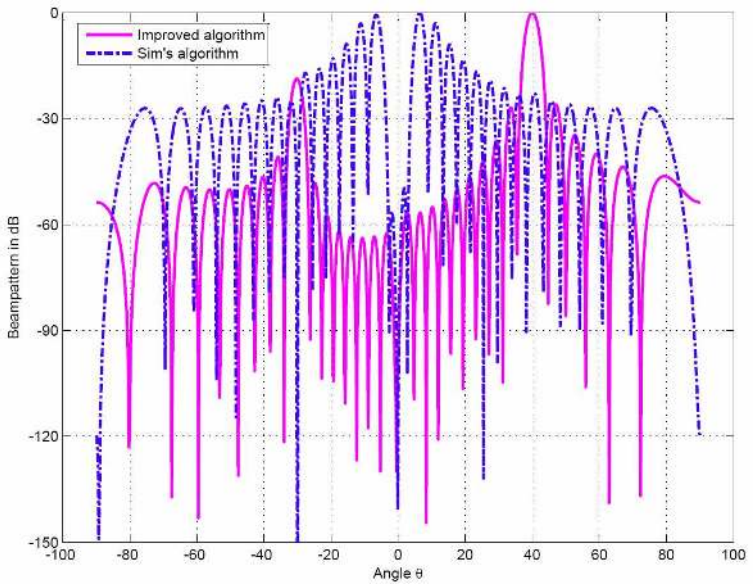


(b) Comparison of final output

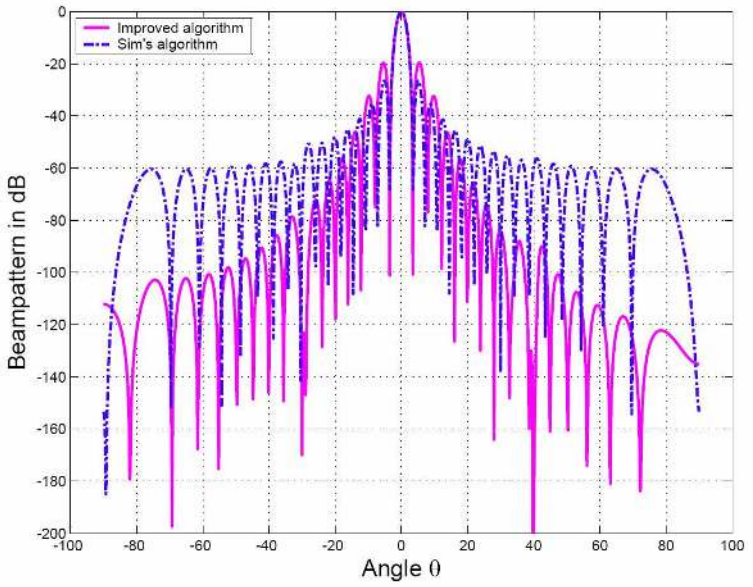
Figure 5.5: Comparison between the algorithm in Sec 5.3.1 and algorithm in [4], error limit=10^{-2}

is one strong interference with power 30 dB, carrier frequency 100 MHz, baud rate 16 kbps, from $-30°$ direction. The desired signal reaches array companied with a multipath component with power -10 dB and direction $40°$. In order

to resolve the multipath component, we form 2 subarrays, each of them has 31 antennas. We can see from Figure 5.6 that in our improved algorithm, nulls are directed at both of the interference and multipath signal directions, which can not be achieved by Sim's algorithm.



(a) Comparison of auxiliary beam



(b) Comparison of final output

Figure 5.6: Comparison between the algorithm in Sec 5.3.2 and algorithm in [4], error limit=10^{-2}

Chapter 6

Iterative Beam Pattern Synthesis Techniques

6.1 Introduction

To implement high performance beamformer, the antenna array requires functions that can acquire the interference statistics before pattern synthesis process. The cost of this procedure could turn out to be impractical for several reasons:

- The interference in the system can be time-varying, which is often the case in communications applications. This would require continuous retransmission of information of the interference parameters from the transmitter to receiver.
- The computational cost of the solution may render it impractical.

Instead of having a solution where all the interference informations are required to be *a priori*, we could choose an implementation which adaptively

estimates the interference, and as time proceeds, incorporates this information into a dynamic pattern synthesis formation. Hence come the more and more popular adaptive beamforming techniques in the past few decades, which can cancel out interferences without knowing their directions beforehand and point its mainbeam in the desired signal's direction. However, in an adaptive beamformer, sidelobes in directions of no interference are usually left uncontrolled and can be undesirably high.

In this chapter, we investigate and develop new adaptive beamforming algorithms that are not only able to perform conventional adaptive beamforming functions, but also able to control the adapted beam patterns including both the sidelobe area and main beam. The techniques from iterative pattern synthesis and adaptive beamforming are merged to provide such capability.

In the following part, Section 6.2 presents in brief about the Frost algorithm; several existing iterative beam pattern synthesis algorithms, especially the algorithm proposed by Zhou, are introduced in section 6.3; in section 6.4, we present our proposed algorithm in detail; computational cost of the above mentioned algorithms are analyzed and compared in section 6.5 to show that our algorithm is superior in saving computational costs over other algorithms; and finally simulation results are given in Section 6.6 to illustrate the effectiveness of our proposed algorithm.

6.2 Existing Methods on Adaptive Beamforming

In the family of adaptive beamforming techniques, the use of Least Mean Square (LMS) algorithm to estimate optimal weights of an array is widespread and its research has been of great interest for some time. The algorithm is referred to as *constrained LMS* algorithm [7, 21] when the weights are updated in accordance with certain constrains which reflect prior knowledge of certain parameters or properties of the problem under consideration, *e.g.* DOA of user signals, whereas it is referred to as *unconstrained LMS* algorithm when the weights are not constrained at each iteration. The latter algorithm is often applied when weights are updated using reference signals and without the knowledge of DOA of desired signal. A significant feature of LMS-type algorithms is their simplicity; moreover, it does not require measurements of the pertinent correlation functions, nor does it require matrix inversion. Despite all these advantages, we can only control the value of certain directions rather than the full range of beam pattern. As a result of this, although the resulting beam pattern satisfies our expectation on certain directions, it may give high output levels on the remaining sidelobe regions, increasing the noise gain. This may be undesirable in certain applications such as mobile communications because unwanted signals may come in the system and interrupt the normal system operation [62]. Next we begin to describe frost adaptive beamforming in brief.

Consider the following optimization problem,

$$\mathbf{w} = \arg\min_{\mathbf{w}} \; J_{\mathbf{w}} \qquad \text{subject to } \mathbf{C}^{\mathbf{H}}\mathbf{w} = \mathbf{f}, \tag{6.1}$$

where $\mathbf{w}$ is weight vector of length M, $\mathbf{C}$ is the $M \times p$ constraint matrix, and $\mathbf{f}$ is the $p \times 1$ constraint vector with p being the number of linear constraints. The most commonly used objective function in literature is the one that uses mean output energy, *i.e.*, $J_{\mathbf{w}} = \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w}$, where $\mathbf{R}_{xx}$ is the $M \times M$ autocorrelation matrix of input signal. We can obtain the solution to (6.1) by using Lagrange multipliers [21], and the result turns out to be

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{xx}^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}_{xx} \mathbf{C})^{-1} \mathbf{f}. \tag{6.2}$$

Frost [21] proposed an algorithm to estimate $\mathbf{w}_{\text{opt}}$ by using steepest descent algorithms, or more precisely, by using the LMS algorithm for updating the weight vector.

The narrowband version of constrained LMS algorithm proposed by Frost [21] uses as an estimate of autocorrelation matrix $\hat{\mathbf{R}}_{xx}$, at time instant k, the outer product of input signal vector $\mathbf{x}(k)$ by itself, *i.e.* $\hat{\mathbf{R}}_{xx} = \mathbf{x}(k)\mathbf{x}^H(k)$. In this case, the coefficient-update equation becomes:

$$\begin{aligned} \mathbf{w}(k+1) =& \mathbf{w}(k) + \mu e(k)[\mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H]\mathbf{x}(k) + \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^H\mathbf{w}(k)] \\ =& \mathbf{P}[\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)] + \tilde{\mathbf{f}}, \end{aligned} \tag{6.3}$$

where $\mathbf{I}$ is the $M \times M$ identity matrix and $\mathbf{P}$ is the projection matrix onto subspace orthogonal to the subspace spanned by constraint matrix $\mathbf{C}$ and is given by

$$\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H; \tag{6.4}$$

$\tilde{\mathbf{f}}$ is quiescent weight vector which can be calculated by

$$\tilde{\mathbf{f}} = \mathbf{C}(\mathbf{C}\mathbf{C}^H)^{-1}\mathbf{f}; \tag{6.5}$$

μ is gradient step size and $e(k)$ is the error signal between reference signal and array output. In case of absence of reference signal, *i.e.* the reference signal is zero for all time instants, $e(k)$ is simply the negative of array output.

We notice from (6.3) that the term multiplied by the projection matrix $\mathbf{P}$, *i.e.* $\mathbf{w}(k)+\mu e(k)\mathbf{x}(k)$, corresponds to the unconstrained LMS solution, and then it is projected onto the homogenous hyperplane $\mathbf{C}^H\mathbf{w} = \mathbf{0}$ and moved back to the constraint hyperplane $\mathbf{C}^H\mathbf{w} = \mathbf{f}$ by adding vector $\tilde{\mathbf{f}}$.

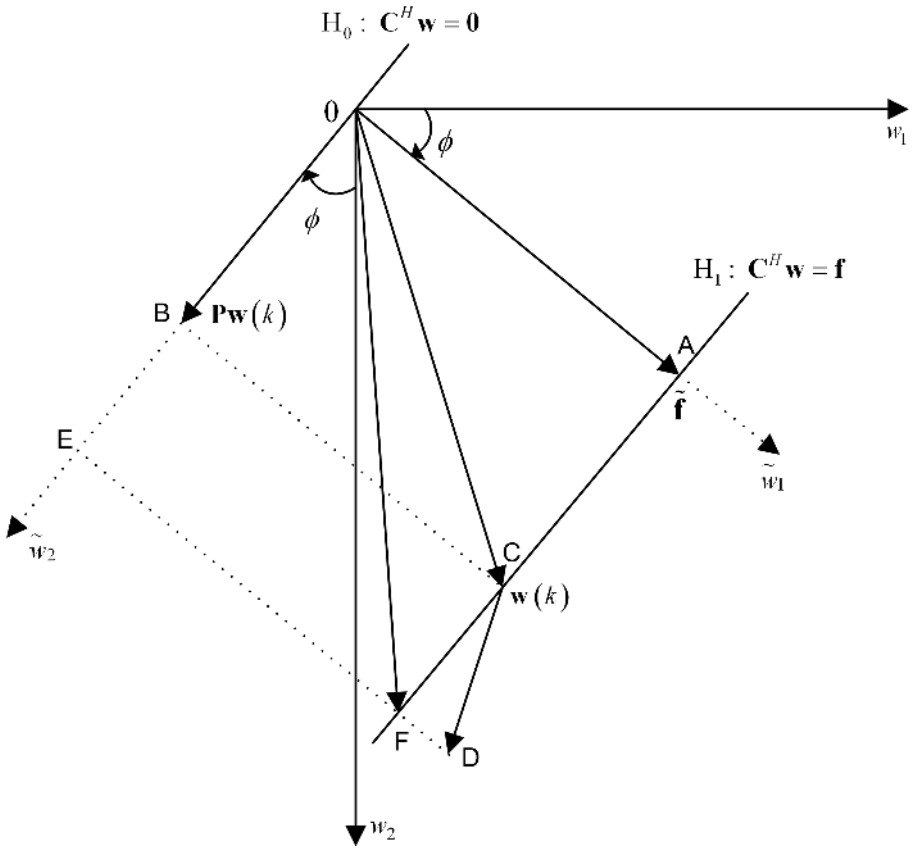


Figure 6.1: Geometrical interpretation of Frost adaptive beamformer

Figure 6.1 shows how weights are updated and how the projection system works by using a vector diagram for a 2-dimensional weight vector system. Point A in the figure indicates the quiescent weight vector $\tilde{\mathbf{f}}$. Point C is the position of the weight vector after the completion of the k-th iteration. It is the cross-section of the constraint hyperplane H_1 and the power surface $\mathbf{w}^H(k)\mathbf{R}_{xx}\mathbf{w}(k)$ (not shown in the figure). Point B denotes the projection of $\mathbf{w}(k)$ onto the orthogonal complementary subspace of constraint subspace. Vector $\overrightarrow{CD}$ is the gradient vector at k-th time instant, so point D denotes $\mathbf{w}_{\text{unc}}(k+1)$ for unconstrained LMS algorithm. For constrained LMS algorithm, the weights are updated by adding the gradient vector and then projecting it onto homogenous hyperplane H_0. This operation is indicated by point E in the figure. Then the quiescent weight vector $\tilde{\mathbf{f}}$ is added to restore the constraint. This action moves the updated weights towards point F. The process continues until the optimal weight vector is achieved.

6.3 Existing Methods on Iterative Beam Pattern Synthesis

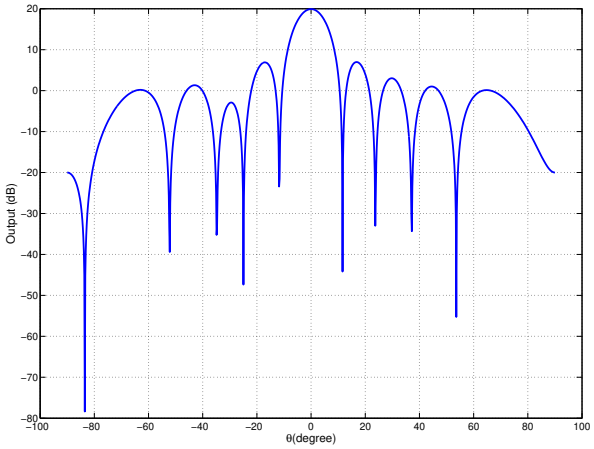
6.3.1 Relationship Between Sidelobe and Interference

To synthesize an array pattern is to find a set of weights such that the array pattern has a desired shape, *e.g.* a maximum at the desired signal angle and a certain sidelobe rolloff. Many algorithms have been developed in this area, and the literature review for this part has been presented in section 1.2. With the algorithms mentioned there, we can obtain a beam pattern which is directed

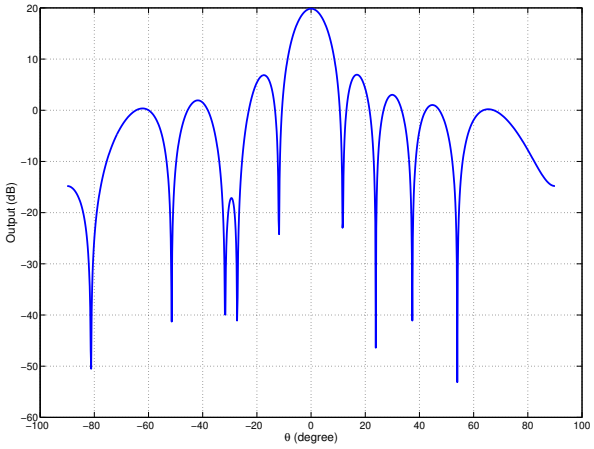
towards the direction of signal of interest with sidelobe equal to or lower than certain predefined level. Its major disadvantage is that on directions of interferences, instead of producing deep nulls, these algorithms can only ensure the output on these directions is no higher than certain level. Besides, calculation of matrix inverse is frequently required by these algorithms, which may give rise to a heavy computational burden. Now, we begin to describe the underlying principles which are frequently used in various beam pattern synthesis algorithms.

The response of an adaptive array pattern to interfering signals depends on the number of interfering signals in relation to the number of degree of freedom in the array. An M-element array has $M-1$ degrees of freedom in the array [33]. One degree of freedom is used to form a pattern maximum on the desired signal. The remaining $M-2$ degrees are available to null interference signals. If $M-2$ or fewer interference signals are incident on the array, the array usually forms a null on each interference direction. However, if more than $M-2$ interference signals are incident, the array does not normally null the individual interference signals but instead forms a compromise pattern that minimizes the total interference power at the array output.

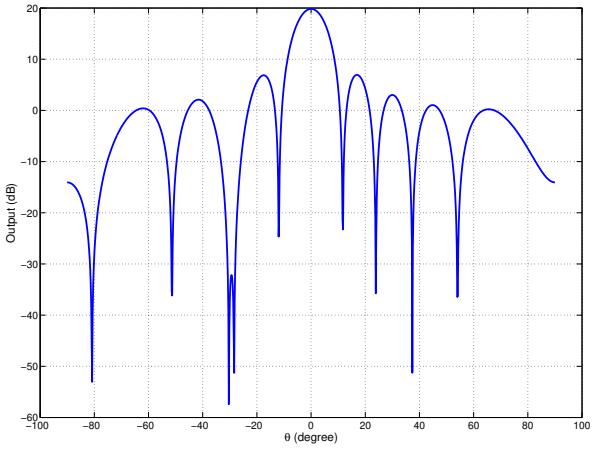
The response of an adaptive array to an interference signal also depends on the interference signal strength. The stronger the interference, the lower the adapted pattern level. Figure 6.2 and Figure 6.3 illustrate this behavior. In Figure 6.2, a single interference signal is incident on a ten-element adaptive array from $\theta_i = -30°$. The elements of the array are isotropic and spaced every half-wavelength; the desired signal is at $\theta_d = 0°$. In Figure 6.2(a), the INR is -10 dB, in Figure 6.2(b), it is 0 dB and in Figure 6.2(c) it is $+10$ dB. Note



(a) INR=-10dB



(b) INR=0dB



(c) INR=10dB

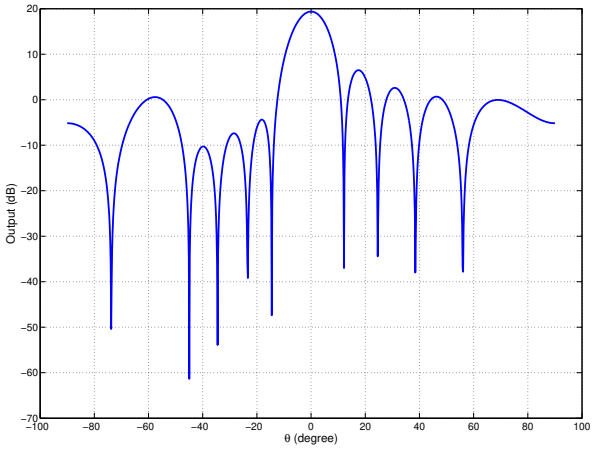
Figure 6.2: Adapted patterns with one interference signal

that the adapted pattern level at $\theta = -30°$ decreases as the interference power is increased.

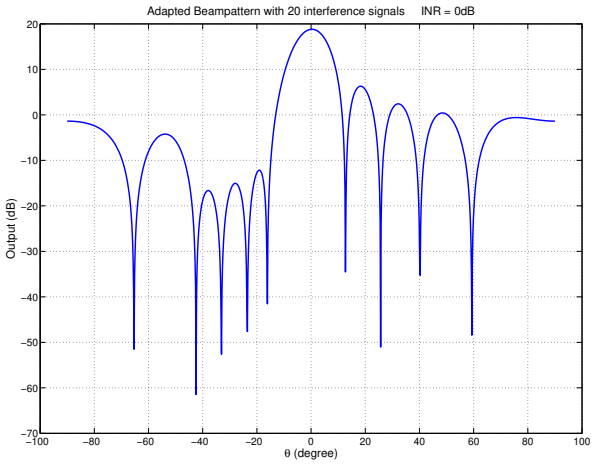
Figure 6.3 shows what happens when there are more interference signals than the number of degrees of freedom in the array. In Figure 6.3, 20 interference signals are incident on the same ten-element array from a 28.5° angular region centered at $\theta = -30°$. The INR (for each interference signal) is again varied from -10 to $+10$ dB. Note again the sidelobe drops as the INR is inceased. The pattern design algorithm described below takes advantage of this behavior to force the sidelobes down.

6.3.2 Olen and Bell's Algorithms

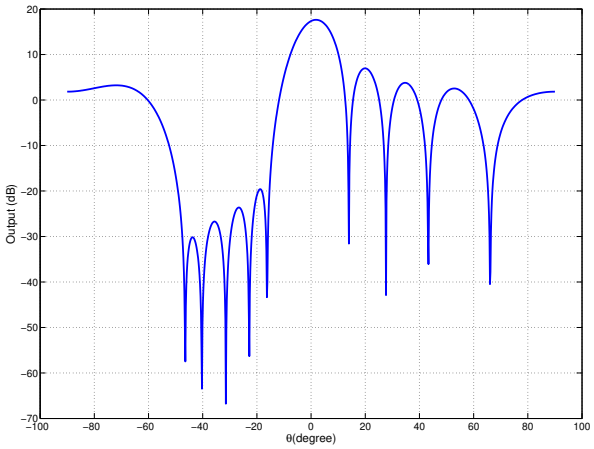
Olen's Algorithm: The underlying idea for this approach is to assume the given array elements are used as elements of an adaptive array [63]. The main beam is steered in the desired direction by choosing the steering vector for that direction. To reduce sidelobe, a large number of interfering signals is assumed to be incident on the array from the sidelobe region. The adapted pattern is then computed and compared with the design objectives. At any angle where the sidelobes are too high or too low, the interference power is increased or decreased accordingly and then the weights are recalculated. This process is repeated iteratively until a suitable final pattern is obtained. For a given set of elements, we can find a set of array coefficients that steer the main beam in a given direction and yield sidelobes meeting a specified criterion, if such a set of array coefficients exists. If the pattern specifications cannot be met with the given elements, the algorithm finds the best attainable pattern. Because this



(a) INR per signal=-10dB



(b) INR per signal=0dB



(c) INR per signal=10dB

Figure 6.3: Adapted patterns with 20 interference signals

approach is a numerical technique, it does not yield analytic solutions for the weights. However, being a numerical technique, it can be used with much more general types of problems than an analytical approach. The method easily handles arrays in which the element patterns of different elements are different and the element locations are arbitrary. It can also be used to obtain patterns whose sidelobe levels vary arbitrarily with angle.

Despite all these advantages, it leaves the mainlobe uncontrolled, and also, in each iteration, it requires large computation of outer product and matrix inverse which makes it difficult to realize in real-time DSP chips. For details of the algorithm, please refer to [1].

Bell's Algorithm: This algorithm takes a similar form with the previous one, and explains the beam pattern adaptation from the perspective of a trade-off between directivity and sidelobe level.

With this algorithm, we can obtain tight sidelobe control by defining a set of small sectors in the sidelobe region and setting the desired beam pattern to zero in these regions. The desired weight vector in each sector is just the all-zero vector. In the limit of infinitesimally small sectors, the pattern error constraints become constraints on the magnitude squared of the beam pattern at every point in the sidelobe region. The allowed deviation can be set to the maximum allowable sidelobe level, and the sidelobe levels can be controlled directly. By choosing wide but relatively small sectors, we can still control sidelobe levels fairly accurately. Furthermore, if we choose to constrain pattern "error" only in the sidelobe region and not in the main beam, the problem can be greatly simplified. So, the optimization problem becomes to minimize the total output power subject

to unit response at look direction and squared magnitude of sidelobe level is less than our pre-determined value. We can add a loading factor for each sector in the solution for the previous optimization problem, we are able to calculate the weight in an iterative manner by adjusting the loading factor according to whether a constraint is exceeded or not in that sector.

Similar with previous algorithm, it is usually necessary to adjust the sectors included in sidelobe region at each iteration. As sidelobes are pushed down, the main beam widens, and some sectors previously in the sidelobe region fall in main beam. The constraints on these sectors must then be dropped. Please refer to [2] for details.

6.3.3 Zhou's Algorithm

To overcome the above problems of uncontrolled mainlobe region, Zhou proposed a new pattern synthesis method with iterative weighted least squares [3].

6.3.3.1 Formulation

Let ε be the sum of the weighted pattern errors over the set of angles $\theta_1, \theta_2, \ldots, \theta_N$,

$$\varepsilon = \sum_{i=1}^{N} f(\theta_i) \left| \mathbf{w}^H \mathbf{a}(\theta_i) - p_r(\theta_i) \right|^2, \tag{6.6}$$

where $f(\theta_i)$ is the weighting function and $p_r(\theta_i)$ is the reference pattern. The problem of pattern synthesis can be formulated as finding the optimal weight vector $\mathbf{w}_{\text{opt}}$ that minimize the error ε which can be expressed as

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{aa}^{-1} \mathbf{r}_d, \tag{6.7}$$

where $\mathbf{R}_{aa}$ is the covariance matrix and $\mathbf{r}_d$ is the cross-correlation vector defined as

$$\mathbf{R}_{aa} = \sum_{i=1}^{N} f(\theta_i)\mathbf{a}(\theta_i)\mathbf{a}^H(\theta_i), \tag{6.8}$$

$$\mathbf{r}_d = \sum_{i=1}^{N} f(\theta_i)p_r(\theta_i)\mathbf{a}(\theta_i). \tag{6.9}$$

The error in (6.6) can be further expressed as [38],

$$\varepsilon = \varepsilon_{\min} + (\mathbf{w} - \mathbf{w}_{\text{opt}})^H \mathbf{R}_{aa}(\mathbf{w} - \mathbf{w}_{\text{opt}}), \tag{6.10}$$

where $\varepsilon_{\min}$ is the minimized error ε and can be calculated as

$$\varepsilon_{\min} = \sum_{i=1}^{N} f(\theta_i)\left|p_r(\theta_i)\right|^2 - \mathbf{w}_{\text{opt}}^H \mathbf{R}_{aa}\mathbf{w}_{\text{opt}}. \tag{6.11}$$

The array response at each angular location depends on the weighting function $f(\theta_i)$. A different value of $f(\theta_i)$ puts a different emphasis on array response at the particular direction θ_i. By changing values of $f(\theta_i)$, various array responses can be achieved, thus a specific or desired array pattern can be obtained.

6.3.3.2 Beam pattern control algorithm

The most common objective for pattern synthesis is to obtain a pattern with the sidelobe lower than or equal to a specified level over some regions while maintaining a certain gain at the look direction θ_d. The reference pattern, $p_r(\theta)$, can have the shape in which all the responses in sidelobe regions are zeros and the mainlobe peak response is certain value A. The main lobe shape is specified by the designer. While it is impractical to have all zero sidelobe levels, we can

induce lower and lower sidelobes by increasing the weighting function values in selected areas. The weighting function $f(\theta_i)$ is adaptively updated through an iteration procedure which leads to a desired array pattern:

$$f_{k+1}(\theta) = \begin{cases} h_k(\theta_n) & \theta_n \text{ in main lobe region,} \\ \max\{f_k(\theta_n) + K_p[g_k(\theta_n) - p_r(\theta_n)], 0\} & \theta_n \text{ in sidelobe region,} \end{cases} \tag{6.12}$$

where $n = 1, 2, \ldots, N$ indexes the directions over which we are interested in controlling the pattern, $g_k(\theta_n) = |\mathbf{w}^H(k)\mathbf{a}(\theta_n)|$ is the synthesized pattern magnitude at the k-th iteration. In (6.12), $h_k(\theta)$ is calculated by

$$h_k(\theta) = \begin{cases} f_k(\theta_n) & \text{if } |g_k(\theta_n) - p_r(\theta_n)| \le \epsilon, \\ f_k(\theta_n) + K_m|g_k(\theta_n) - p_r(\theta_n)| & \text{otherwise,} \end{cases} \tag{6.13}$$

where ϵ is a small number for an error tolerance between the synthesized pattern $g_k(\theta_n)$ and the desired pattern $p_r(\theta_n)$ in main lobe region, and K_m and K_p are iteration constants. The desired pattern $p_r(\theta_n)$ is used for weighting function updating to achieve the pattern convergence.

Next, we use $f_{k+1}(\theta_n)$ to compute the updated weight vector. The covariance matrix and the cross-correlation vector become

$$\mathbf{R}_{aa}(k+1) = \sigma^2\mathbf{I} + \sum_{n=1}^{N} f_{k+1}(\theta_n)\mathbf{a}(\theta_n)\mathbf{a}^H(\theta_n), \tag{6.14}$$

$$\mathbf{r}_d(k+1) = \sum_{n=1}^{N} f_{k+1}(\theta_n)p_r(\theta_n)\mathbf{a}(\theta_n), \tag{6.15}$$

where a small quantity σ^2 is added to each diagonal element of the covariance matrix to prevent it from being ill-conditioned [15]. The weight vector in the

next iteration can be calculated by

$$\mathbf{w}(k+1) = \mathbf{R}_{aa}^{-1}(k+1)\mathbf{r}_d(k+1). \tag{6.16}$$

The iteration stops when the difference between $g_k(\theta_n)$ and $p_r(\theta_n)$ is small enough in the main lobe region and when the sidelobe levels of $g_k(\theta_n)$ are equal to or lower than that of $p_r(\theta_n)$.

The weighting function $f_k(\theta_n)$ updates in the sidelobe region are equivalent to the interference updates in Olen and Compton's synthesis algorithm [1]. The major difference between this two algorithms is that the Zhou's algorithm minimizes the output power of a pattern difference while the Olen and Compton algorithm minimizes the power of the just achieved array pattern. Using a difference allows features like the mainlobe to be easily specified. Therefore, this algorithm takes the advantage of controlling the mainlobe shape according to designer's specification whereas Olen and Compton's algorithm is unable to control the pattern in the mainlobe region. However, the tradeoff is additional computation for processing in the mainlobe region.

If the full iteration version of the algorithm is allowed to continue updating after the maximum sidelobe specification is met, all weighting function values outside of the mainlobe will go to zero except those on the peaks of the sidelobes. This happens because the polarity of the $K_p[g_k(\theta_n) - p_r(\theta_n)]$ term in (6.12) becomes negative for all θ_n in the sidelobes that don't correspond to peak locations. Eventually the left argument of the max function becomes negative, and $f_{k+1}(\theta_n)$ becomes zero. This is illustrated using a 10-element uniform linear array in Figure 6.4. The selected desired pattern has a constant -30 dB sidelobe. Figure 6.4 shows the synthesized pattern along with the weighting

functions at different iteration steps using the aforementioned algorithm. The synthesis process starts with unity values of weighting function. It is clear that the values of weighting function will only exist on the peaks in sidelobe regions as $k \to \infty$. So it may not be necessary to deal with such a large number of processing points every time. We consider an alternative form of the algorithm that uses a greatly reduced number of processing points.

The new covariance matrix can be written as the sum of the current covariance matrix $\mathbf{R}_{aa}(k)$ and a residual covariance matrix $\tilde{\mathbf{R}}_{aa}(k)$ that represents the correction needed at the current step, it is the same with $\mathbf{r}_d(k+1)$, *i.e.*

$$\mathbf{R}_{aa}(k+1) = \mathbf{R}_{aa}(k) + \tilde{\mathbf{R}}_{aa}(k), \tag{6.17}$$

$$\mathbf{r}_d(k+1) = \mathbf{r}_d(k) + \tilde{\mathbf{r}}_d(k), \tag{6.18}$$

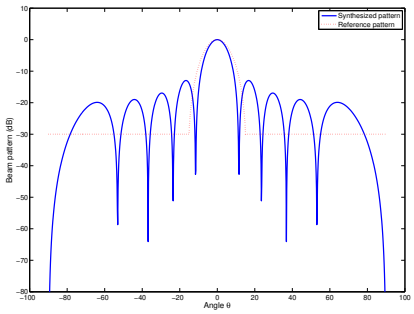
where

$$\tilde{\mathbf{R}}_{aa}(k) = \sum_n r_k(\theta_n)\mathbf{a}(\theta_n)\mathbf{a}^H(\theta_n), \tag{6.19}$$

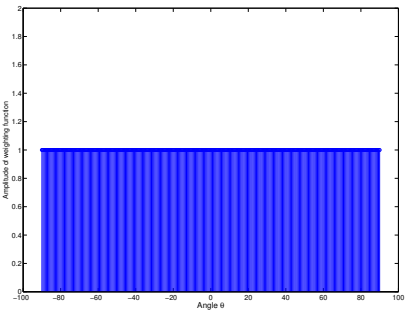
$$\tilde{\mathbf{r}}_d(k) = \sum_n r_k(\theta_n)p_r(\theta_n)\mathbf{a}(\theta_n). \tag{6.20}$$

Here $r_k(\theta_n)$ is a residual weighting function that depends on the difference between the achieved and the desired patterns at the current iteration, and is defined as

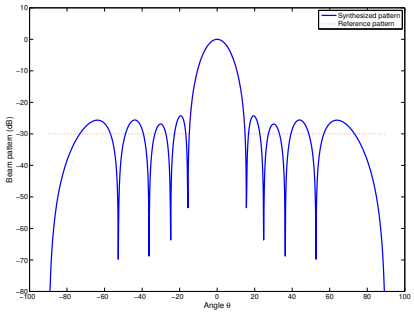
$$r_k(\theta_n) = \begin{cases} s_k(\theta_n) & \theta_n \text{ in mainlobe} \\ \max\{0, K_p[g_k(\theta_n) - p_r(\theta_n)]\} & \theta_n \text{ at sidelobe peak} \end{cases} \tag{6.21}$$



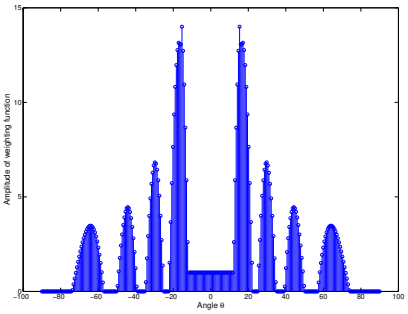
(a) The initial pattern for the uniform linear array



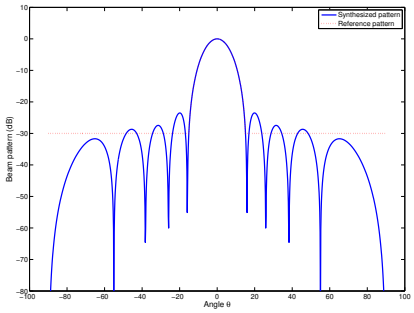
(b) The weighting function for the initial pattern



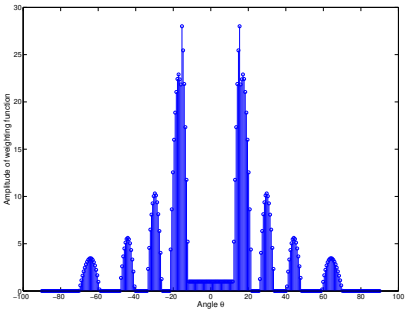
(c) The synthesized pattern at the 12^{th} iteration



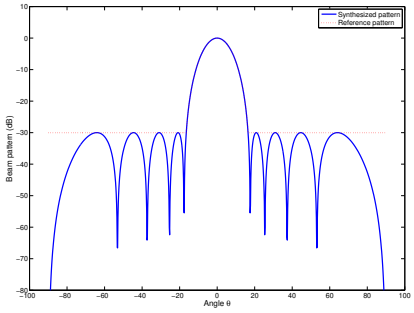
(d) The weighting function at the 12^{th} iteration



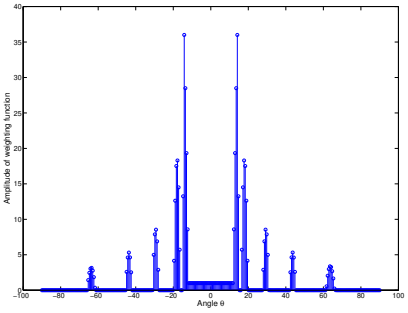
(e) The synthesized pattern at the 120^{th} iteration



(f) The weighting function at the 120^{th} iteration



(g) The synthesized pattern at the 1200^{th} iteration



(h) The weighting function at the 1200^{th} iteration

Figure 6.4: Beam pattern and weighting function evolution for 10-element uniform linear array

where

$$s_k(\theta_n) = \begin{cases} 0 & \text{if } |g_k(\theta_n) - p_r(\theta_n)| \leq \epsilon, \\ K_m \, |g_k(\theta_n) - p_r(\theta_n)| & \text{otherwise.} \end{cases} \tag{6.22}$$

Sidelobe peaks can be found using simple algorithms. The next optimal weight vector is

$$\mathbf{w}(k+1) = \mathbf{R}_{aa}^{-1}(k+1)\mathbf{r}_d(k+1). \tag{6.23}$$

The initial covariance matrix $\mathbf{R}_{aa}(0)$ and cross-correlation vector $\mathbf{r}_d(0)$ can be simply set up as

$$\mathbf{R}_{aa}(0) = \sigma^2\mathbf{I} + \sum_{n=1}^{N} f_0(\theta_n)\mathbf{a}(\theta_n)\mathbf{a}^H(\theta_n), \tag{6.24}$$

$$\mathbf{r}_d(0) = \sum_{n=1}^{N} f_0(\theta_n)p_r(\theta_n)\mathbf{a}(\theta_n), \tag{6.25}$$

where the values of $f_0(\theta_n)$ can be a constant of 1 or smaller. The stopping criterion is the same as before.

6.4 Our Proposed Algorithm

6.4.1 Improvement Based on Zhou's Algorithm

Although the simulation results in Zhou's paper show effectiveness of his algorithm, if we take a closer examination on (6.23), it can be found that calculation of matrix inversion is required for every update of weight vector. This gives rise to a heavy computational load and makes his algorithm computationally inefficient. To overcome this problem, we propose an improved algorithm which uses

constrained optimization technique.

Consider the optimization equation shown below,

$$\mathbf{w} = \arg \min_{\mathbf{w}} J_{\mathbf{w}} \qquad \text{subject to } \mathbf{C}^H\mathbf{w} = \mathbf{f}, \tag{6.26}$$

and this time, we choose (6.6) as the objective function $J_{\mathbf{w}}$. The problem can be solved by using Lagrange multiplier. Let $\boldsymbol{\lambda}$ denote the Lagrangian multiplier vector, then the cost function at the k-th iteration can be written as,

$$\begin{aligned} L(k) =& \mathbf{w}^H(k)\mathbf{R}_{aa}(k)\mathbf{w}(k) + \sum_{i=1}^{N} f_k(\theta_i)|p_r(\theta_i)|^2 - \mathbf{w}^H(k)\mathbf{r}_d(k) - \mathbf{w}^T(k)\mathbf{r}_d^*(k) \\ &+ \boldsymbol{\lambda}^H[\mathbf{C}^H\mathbf{w}(k) - \mathbf{f}] + \boldsymbol{\lambda}^T[\mathbf{C}^T\mathbf{w}^*(k) - \mathbf{f}^*], \end{aligned} \tag{6.27}$$

and the gradient of $L(k)$ with respect to $\mathbf{w}^*(k)$ at the k-th iteration is

$$\nabla_{\mathbf{w}^*} L(k) = \mathbf{R}_{aa}(k)\mathbf{w}(k) - \mathbf{r}_d + \boldsymbol{\lambda}^T\mathbf{C}^T. \tag{6.28}$$

Applying steepest gradient-descent algorithm, and after some mathematical manipulations, we can get the following iterative equation for updating weight vector,

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\nabla_{\mathbf{w}^*} L(k) \\ &= \tilde{\mathbf{f}} + \mathbf{P}[\mathbf{w}(k) - \mu\mathbf{R}_{aa}(k)\mathbf{w}(k) + \mu\mathbf{r}_d(k)], \end{aligned} \tag{6.29}$$

where μ is a constant step size; $\mathbf{R}_{aa}$ and $\mathbf{r}_d$ are calculated by (6.17) and (6.18); $\mathbf{P}$ and $\tilde{\mathbf{f}}$ are defined by (6.4) and (6.5). Substituting (6.8) and (6.9) into (6.29), we can rewrite (6.29) as,

$$\mathbf{w}(k+1) = \tilde{\mathbf{f}} + \mathbf{P}\left[\mathbf{w}(k) - \mu\sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)]\mathbf{a}(\theta_i)\right]. \tag{6.30}$$

Several points in mainlobe region and peak points in sidelobe regions are updated in each iteration. To ensure the equivalence between (6.29) and (6.30), $f_k(\theta_i)$ should be updated by

$$f_{k+1}(\theta_i) = f_k(\theta_i) + r_k(\theta_i), \tag{6.31}$$

where $r_k(\theta_i)$ is defined by (6.21). To ensure the convergence of the algorithm, μ should satisfy the condition $0 < \mu < 2/\sup\{\lambda_{\max}^{(k)}\}$, where $\lambda_{\max}^{(k)}$ is the largest eigenvalue of the matrix $\mathbf{PR}_{aa}(k)\mathbf{P}$.

In (6.30), the weight vector at $k+1$-th iteration is updated by summing the quiescent weight vector $\tilde{\mathbf{f}}$ with the projection of the linear combination of several steering vectors onto the orthogonal complementary constrained subspace $\mathbf{P}$. In this process, no matrix inverse calculation is observed. Because of this, compared with Zhou's algorithm, ours is able to save much computational time per iteration.

If we take a careful examination on (6.3) and (6.30), we can notice that these two equations bear much resemblance with each other: the weight vectors in both of these two algorithms are adaptively updated in a similar manner as the sum of a quiescent weight vector $\tilde{\mathbf{f}}$ and a projection vector onto the orthogonal complementary subspace $\mathbf{P}$. Such characteristic suggests us with a new algorithm which combines these two equations together, we name it composite adaptive beamforming algorithm with pattern control (CABPC). The formula of our proposed algorithm is,

$$\mathbf{w}(k+1) = \tilde{\mathbf{f}} + \mathbf{P}[\mathbf{w}(k) - \mu_1 y(k)\mathbf{x}(k) - \mu_2 \sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)]\mathbf{a}(\theta_i)]. \tag{6.32}$$

Here, we consider the case of no reference signal in Frost adaptive beammforming, which is equivalent to a constant zero reference signal, so we use $-y(k)$ to replace $e(k)$ in (6.3). We can see that on one hand, if we let $\mu_1 = 0$, (6.32) becomes beam pattern synthesis algorithm; on the other hand, if μ_2 is set to zero, (6.32) turns to be Frost adaptive beamforming algorithm. The effect of beam pattern control and adaptive beamforming can be assigned different emphasis by adjusting μ_1 and μ_2 respectively. Moreover, no matrix inversion calculation is involved in (6.32), which makes our algorithm computationally efficient.

6.4.2 Householder Transform

Next, we introduce the Householder Transform [64] and try to apply it to our proposed algorithm to further reduce computational load.

The main idea of Householder Transform (HT) is that it performs a rotation on the second term in (6.32) in order to make sure that the coefficient vector is never perturbed by directions that are not excited by this term. We use an orthogonal rotation matrix $\mathbf{Q}$ to complete this task, and make the following definitions:

$$\bar{\mathbf{w}}(k) = \mathbf{Q}\mathbf{w}(k), \qquad \bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k),$$
$$\bar{\mathbf{a}}(\theta_i) = \mathbf{Q}\mathbf{a}(\theta_i), \qquad \bar{\mathbf{C}} = \mathbf{Q}\mathbf{C},$$

The operation of rotating $\mathbf{w}(k)$ can be visualized in Figure 6.1 by rotating the axis w_1 and w_2 clockwise by ϕ to $\tilde{w}_1$ and $\tilde{w}_2$ which are indicated by dashed lines.

If the transform matrix $\mathbf{Q}$ satisfies the following equations,

$$\mathbf{Q}\mathbf{Q}^H = \mathbf{Q}^H\mathbf{Q} = \mathbf{I}, \tag{6.33}$$

and

$$\bar{\mathbf{C}}(\bar{\mathbf{C}}^H\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^H = \begin{bmatrix} \mathbf{I}_{p\times p} & \mathbf{0}_{p\times(M-p)} \\ \mathbf{0}_{(M-p)\times p} & \mathbf{0}_{(M-p)\times(M-p)} \end{bmatrix}, \tag{6.34}$$

then we could have the following properties

$$\bar{\mathbf{C}}^H\bar{\mathbf{w}}(k) = \mathbf{C}^H\mathbf{Q}^H\mathbf{Q}\mathbf{w}(k) = \mathbf{f}, \tag{6.35}$$

and the transformed projection matrix

$$\bar{\mathbf{P}} \triangleq \mathbf{Q}\mathbf{P}\mathbf{Q}^H = \mathbf{I} - \bar{\mathbf{C}}(\bar{\mathbf{C}}^H\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^H = \begin{bmatrix} \mathbf{0}_{p\times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \tag{6.36}$$

If we choose the initial weight vector as

$$\bar{\mathbf{w}}(0) = \bar{\mathbf{C}}(\bar{\mathbf{C}}^H\bar{\mathbf{C}})^{-1}\mathbf{f} = \mathbf{Q}\mathbf{C}(\mathbf{C}^H\mathbf{Q}^H\mathbf{Q}\mathbf{C})^{-1}\mathbf{f} = \mathbf{Q}\tilde{\mathbf{f}}, \tag{6.37}$$

we notice that the only the first p elements of $\bar{\mathbf{w}}(0)$ are nonzero because

$$\begin{aligned} \bar{\mathbf{w}}(0) &= \bar{\mathbf{C}}(\bar{\mathbf{C}}^H\bar{\mathbf{C}})^{-1}\mathbf{f} = \bar{\mathbf{C}}(\bar{\mathbf{C}}^H\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^H\mathbf{w}(0) \\ &= \begin{bmatrix} \mathbf{I}_{p\times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{w}(0). \end{aligned} \tag{6.38}$$

Then the iterative updating equation of our proposed Householder-Transform CABPC (HT-CABPC) comes out by premultiplying (6.32) by orthogonal rota-

tion matrix $\mathbf{Q}$,

$$\begin{aligned}\bar{\mathbf{w}}(k+1) =& \mathbf{Q}\mathbf{w}(k+1) \\ =& \bar{\mathbf{w}}(0) + \mathbf{Q}\mathbf{P}\mathbf{Q}^H \left\{ \bar{\mathbf{w}}(k) - \mu_1 y(k)\bar{\mathbf{x}}(k) - \mu_2 \sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)]\bar{\mathbf{a}}(\theta_i) \right\}.\end{aligned} \tag{6.39}$$

Substituting (6.36) and (6.38) into (6.39), the above equation can be further written as

$$\begin{aligned}\bar{\mathbf{w}}(k+1) &= \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k+1) \end{bmatrix} \\ &= \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k) \end{bmatrix} - \mu_1 y(k) \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{x}}_L(k) \end{bmatrix} \\ &\quad - \mu_2 \sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)] \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{a}}_L(\theta_i) \end{bmatrix},\end{aligned} \tag{6.40}$$

where $\bar{\mathbf{w}}_U(0)$ denotes the first p elements of $\mathbf{w}(0)$; $\bar{\mathbf{w}}_L(k)$, $\bar{\mathbf{x}}_L(k)$ and $\bar{\mathbf{a}}_L(\theta_i)$ denote the last $M-p$ elements of $\bar{\mathbf{w}}(k)$, $\bar{\mathbf{x}}(k)$ and $\bar{\mathbf{a}}(\theta_i)$ respectively. Therefore, in (6.40), only the last $M-p$ elements in the weight vector require update in each iteration, which can greatly reduce the computational requirement when the number of constraint is large.

Although $\bar{\mathbf{w}}(k)$ is rotated and biased by a transformation matrix $\mathbf{Q}$, both the output signal and beam pattern are unchanged, as

$$y(k) = \bar{\mathbf{w}}^H(k)\bar{\mathbf{x}}(k) = \mathbf{w}^H(k)\mathbf{Q}^H\mathbf{Q}\mathbf{x}(k) = \mathbf{w}^H(k)\mathbf{x}(k), \tag{6.41}$$

$$g(\theta_i) = \bar{\mathbf{w}}^H(k)\bar{\mathbf{a}}(\theta_i) = \mathbf{w}^H(k)\mathbf{Q}^H\mathbf{Q}\mathbf{a}(\theta_i) = \mathbf{w}^H(k)\mathbf{a}(\theta_i). \tag{6.42}$$

Then we are sure that our proposed algorithm minimizes the same objective function as (6.32).

The transform matrix $\mathbf{Q}$ can be calculated with successive Householder Transform applied to each of the p columns of $\mathbf{CK}$, where $\mathbf{K}$ is the square root of $(\mathbf{C}^H\mathbf{C})^{-1}$, *i.e.*, $\mathbf{K}\mathbf{K}^H = (\mathbf{C}^H\mathbf{C})^{-1}$. Details of choice of $\mathbf{Q}$ can be found in [**?**].

6.5 Comparison on Computational Cost

In this section, we give an evaluation of the computations that are required for the above-mentioned algorithms. Admitted that in reality there can be different ways to perform certain calculation, we are simply intended to provide an approximate idea of the computational complexity for each algorithms.

6.5.1 Frost Beamforming

We begin this section with a detailed description of the calculation requirement of Frost adaptive beamforming.

1. Each iteration in (6.3) requires the evaluation of the inner product $y(k) = \mathbf{w}^H(k)\mathbf{x}(k) = -e(k)$, between two vectors of size M each. This calculation requires M complex multiplication and $M-1$ complex additions. Using the fact that one complex multiplication requires four real multiplications and two real additions, while one complex addition requires two real additions, we find that the evaluation of this inner product requires $4M$ real multiplications and $4M-2$ real additions.

2. Evaluation of the product $\mu e(k)$, where we assume μ is a real scalar, requires two real multiplications when the data is complex-valued.

3. The algorithm further requires multiplying the the scalar $\mu e(k)$ by the data vector $\mathbf{x}(k)$. This requires M complex multiplications which correspond to $4M$ real multiplications and $2N$ real additions.

4. Next, the addition of two vectors $\mathbf{w}(k)$ and $\mu e(k)\mathbf{x}(k)$ requires M complex additions, *i.e.*, $2M$ real additions.

5. The obtained vector $[\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)]$ is going to multiply with the projection matrix $\mathbf{P}$. This operation requires M^3 complex multiplications and $M^2(M-1)$ complex additions, so $4M^3$ real multiplications and $4M^3 - 2M^2$ real additions are required to fulfill this multiplication.

6. Finally, vector $\tilde{\mathbf{f}}$ is being added to the result in the last step, and the process of updating weight vector in an iteration is completed. In this step, M complex additions, *i.e.* $2M$ real additions are required.

The results of above analysis are listed in Table 6.1.

Table 6.1: Estimated computational cost of Frost algorithm

Term	$\times$	$+$
$\mathbf{w}^H(k)\mathbf{x}(k)$	$4M$	$4M-2$
$\mu e(k)$	2	
$\mu e(k)\mathbf{x}(k)$	$4M$	$2M$
$\mathbf{w}(k)+\mu e(k)\mathbf{x}(k)$		$2M$
$\mathbf{P}[\mathbf{w}(k)+\mu e(k)\mathbf{x}(k)]$	$4M^3$	$4M^3-2M^2$
$\mathbf{P}[\mathbf{w}(k)+\mu e(k)\mathbf{x}(k)]+\tilde{\mathbf{f}}$		$2M$
Total per iteration	$4M^3+8M+2$	$4M^3-2M^2+10M-2$

6.5.2 Zhou's Algorithm

Now we apply the above analysis approach to Zhou's algorithm which is introduced in section 6.3.3. Computational requirements for each iteration in Zhou's algorithm are summarized in Table 6.2.

Table 6.2: Estimated computational cost of Zhou's algorithm

Term	$\times$	$+$
$g_k(\theta_i), \quad i = 1, 2, \ldots, N$	$4MN$	$(4M-2)N$
$r_k(\theta_i), \quad i = 1, 2, \ldots, N$	$2N$	$2N$
$\mathbf{a}(\theta_i)\mathbf{a}^H(\theta_i), \quad i = 1, 2, \ldots, N$	$4NM^2$	$2NM^2$
$\tilde{\mathbf{R}}_{aa}(k)$	$4NM^2$	$4NM^2 + 2MN - 2M^2$
$\tilde{\mathbf{r}}_d(k)$	$4N(M+1)$	$4MN + 2N - 2M$
$\mathbf{R}_{aa}(k+1)$		$2M^2$
$\mathbf{r}_d(k+1)$		$2M$
$\mathbf{R}_{aa}^{-1}(k+1)$	M^3	M^3
$\mathbf{R}_{aa}^{-1}(k+1)\mathbf{r}_d(k+1)$	$4M^3$	$4M^3 - 2M^2$
Total per iteration	$5M^3 + 5NM^2 + 8MN + 6N$	$5M^3 + 6NM^2 + 10MN$ $-2M^2 + 2N - 2M$

6.5.3 Our Proposed Algorithm CABPC

With the above results, we can write out the computational load for our proposed algorithm CABPC in Table 6.3 with (6.32) as weight vector updating equation.

Comparing the results in Table 6.2 and Table 6.3, we find that our proposed algorithm is superior over Zhou's algorithm when we choose a large number of angles for beam pattern synthesis, *i.e.* N is comparable in size with M.

Table 6.3: Estimated computational cost of our proposed algorithm CABPC

Term	$\times$	$+$
$g_k(\theta_i), \quad i = 1, 2, \ldots, N$	$4MN$	$N(4M-2)$
$g_k(\theta_i) - p_r(\theta_i), \quad i = 1, 2, \ldots, N$		$2N$
$r_k(\theta_i), \quad i = 1, 2, \ldots, N$	$2N$	$2N$
$f_k(\theta_i), \quad i = 1, 2, \ldots, N$		$2N$
$\mu_2 \sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)]\mathbf{a}(\theta_i)$, $i = 1, 2, \ldots, N$	$4MN + 4N + 2M$	$4MN + 2N + 2M$
$\mu_2 \mathbf{P} \sum_i f_k(\theta_i)[g_k(\theta_i) - p_r(\theta_i)]\mathbf{a}(\theta_i)$, $i = 1, 2, \ldots, N$	$4M^3$	$4M^3 - 2M^2$
Frost algorithm	$4M^3 + 8M + 2$	$4M^3 - 2M^2 + 10M - 2$
$\mathbf{w}(k+1)$		$2M$
Total per iteration	$8M^3 + 8MN$ $+10M + 6N + 2$	$8M^3 - 4M^2 + 8MN$ $+6N + 14M - 2$

6.5.4 Our Proposed Algorithm HT-CABPC

Furthermore, by taking advantages of Householder Transform, the computational load of our algorithm can be further reduced greatly since only the last $M-p$ elements in the vectors which require update in each iteration are recalculated. Now we give a brief analysis on this point. Much of the computations in (6.40) comes from the calculation of $\bar{\mathbf{x}}(k)$. This transformation of input signal can be carried out as

$$\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k) = \mathbf{Q}_p \ldots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{x}(k), \tag{6.43}$$

where

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I}_{(i-1)\times(i-1)} & \mathbf{0}_{(i-1)\times(M-i+1)} \\ \mathbf{0}_{(M-i+1)\times(i-1)} & \bar{\mathbf{Q}}_i \end{bmatrix}, \tag{6.44}$$

and matrix $\bar{\mathbf{Q}}_i$ is a $(M-i+1)\times(M-i+1)$ Householder Transform matrix, which is given by

$$\bar{\mathbf{Q}}_i = \mathbf{I} - 2\bar{\boldsymbol{v}}_i\bar{\boldsymbol{v}}_i^H. \tag{6.45}$$

Here we define the vector $\boldsymbol{v}_i$ as

$$\boldsymbol{v}_i = [\mathbf{0}_{i-1}^T \bar{\boldsymbol{v}}_i^T]^T, \tag{6.46}$$

and the matrix $\boldsymbol{\Upsilon}$ as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_p \end{bmatrix}. \tag{6.47}$$

Now, the calculation of $\bar{\mathbf{x}}(k)$ can be carried out according to the following process:

Step 1: Let $\bar{\mathbf{x}}_k = \mathbf{x}(k)$; $i = 1$;

Step 2: For $i \le p$

Step 3: $\bar{\mathbf{x}}_k(i:M) = \bar{\mathbf{x}}_k(i:M) - 2\boldsymbol{\Upsilon}(i:M,i)\times$

$$[\boldsymbol{\Upsilon}^H(i:M,i)\bar{\mathbf{x}}_k(i:M)]$$

Step 4: $i = i+1$

Step 5: Go to Step 2

Step 6: $\bar{\mathbf{x}}(k) = \bar{\mathbf{x}}_k$;

The computational cost for calculating $\bar{\mathbf{x}}(k)$ is described in Table 6.4.

With the above results, we can estimate the computational load for updating weight vector of (6.40) by our proposed algorithm with Householder Transform. The analysis is shown in Table 6.5.

Table 6.4: Estimated computational cost for calculating $\bar{\mathbf{x}}(k)$

Term	$\times$	$+$
$\boldsymbol{\Upsilon}^H(i:M,i)\bar{\mathbf{x}}_k(i:M), \quad i=1,2,\ldots,p$	$2p[2M-(p-1)]$	$2p[2M-(p-1)]-2p$
$2\boldsymbol{\Upsilon}(i:M,i), \quad i=1,2,\ldots,p$	$p[2M-(p-1)]$	
$2\boldsymbol{\Upsilon}(i:M,i)[\boldsymbol{\Upsilon}^H(i:M,i)\bar{\mathbf{x}}_k(i:N)]$, $i=1,2,\ldots,p$	$2p[2M-(p-1)]$	$p[2M-(p-1)]$
$\bar{\mathbf{x}}_k(i:M)-2\boldsymbol{\Upsilon}(i:M,i)[\boldsymbol{\Upsilon}^H(i:M,i)\bar{\mathbf{x}}_k(i:M)]$, $i=1,2,\ldots,p$		$p[2M-(p-1)]$
Total per iteration	$5p[2M-(p-1)]$	$4p[2M-(p-1)]-2p$

In the case of $N > p$, comparing the results in Table 6.1-6.3 with Table 6.5, we can easily find that equipped with Householder Transform, our proposed algorithm can lead to a great saving on computations at about from $O(M^3)$ to $O(MN)$. As the data rates of evolving communications systems increases, the digital signal processors will have less and less time to perform the required calculations. Therefore, this is a very attractive advantage in real communications applications.

6.6 Simulation Results

In this section, we employ a 21-element uniform linear array with half wavelength spacing to illustrate the performance of our proposed algorithm. The desired signal of 10 dB is coming from the boresight of the array. Two interference of 26 dB and 30 dB arrive at the array from 40° and -40° respectively. The noise level is 10 dB. The reference beam pattern has a mainbeam width of 20° and 40 dB uniform sidelobe level. The iteration constants K_m and K_p are 30 and 300 respectively.

Table 6.5: Estimated computational cost of our proposed algorithm HT-CABPC

Term	$\times$	$+$
$\bar{\mathbf{x}}(k)$	$5p[2M-(p-1)]$	$4p[2M-(p-1)]-p$
$\mu_1 y(k)$	2	0
$\mu_1 y(k)\bar{\mathbf{x}}_L(k)$	$4(M-p)$	$2(M-p)$
$g_k(\theta_i), \quad i=1,2,\ldots,N$	$4MN$	$4MN-2N$
$g_k(\theta_i)-p_r(\theta_i), \quad i=1,2,\ldots,N$		$2N$
$r_k(\theta_i), \quad i=1,2,\ldots,N$	$2N$	$2N$
$f_k(\theta_i), \quad i=1,2,\ldots,N$		$2N$
$\mu_2 \sum_i f_k(\theta_i)[g_k(\theta_i)-p_r(\theta_i)]\bar{\mathbf{a}}_L(\theta_i)$, $i=1,2,\ldots,N$	$(4N+2)(M-p+1)$	$(4N-2)(M-p)+2N$
$\mathbf{w}(k+1)$		$4(M-p)$
Total per iteration	$(2M-p)(4N+5p)+$ $O(M,N,p)$	$4(2M-p)(N+p)+$ $O(M,N,p)$

6.6.1 Beam Patterns

Figure 6.5(a) shows the output of beam pattern synthesis algorithm, which is a special case in our proposed formula (6.40) with $\mu_1 = 0$. We can see although it has a uniform sidelobe with mainbeam steered towards the desired signal's direction, it fails to form nulls on interferences directions because in the process of pattern synthesis, the algorithm does not take into account of the real-time input signal $\mathbf{x}(k)$ but only fulfills certain predefined pattern shape requirement. This is a disadvantage especially for some applications such as wireless communications, in which strong interferences may appear in the system from time to time without a priori knowledge. In such case, nulls are more preferable as low sidelobe level is not enough to combat with the effect of strong interferences.

The output of Frost adaptive beamforming algorithm is shown in Figure 6.5(b). Frost adaptive beamforming is also a special case of our propose algorithm if we let $\mu_2 = 0$ in (6.40). In Figure 6.5(b), we observe nulls on interference

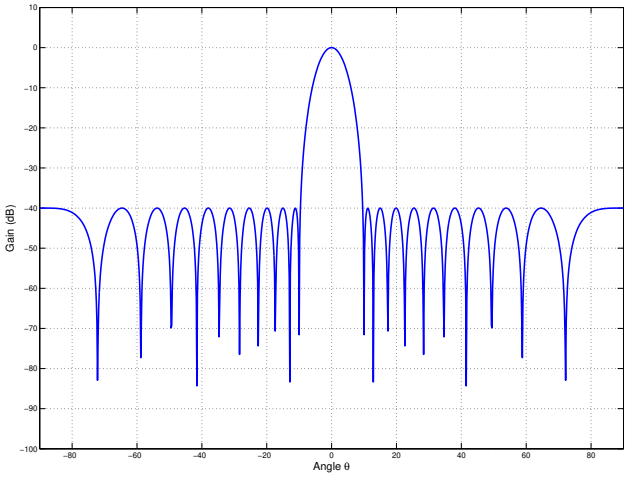
directions. However, the remaining sidelobe region is in relative higher level as compared with Figure 6.5(a) and in irregular shape. Such a beam pattern is much more unfavorable than the one in Figure 6.5(a) especially when a strong interference comes into a mobile communications system suddenly, and happens to come from the direction corresponding to one of the sidelobe peaks.

Figure 6.5(c) shows the result of our proposed algorithm. The two step size μ_1 and μ_2 are set to 1. From Figure 6.5(c) we can notice that besides uniform and well-shaped sidelobe, our algorithm is able to place nulls towards interferences as well. Then, it can be concluded our proposed algorithm has the benefits and features from both adaptive beamforming and pattern synthesis. Moreover, matrix inverse is not required in updating the weight vector, which makes our algorithm a much more competitive candidate in real application.

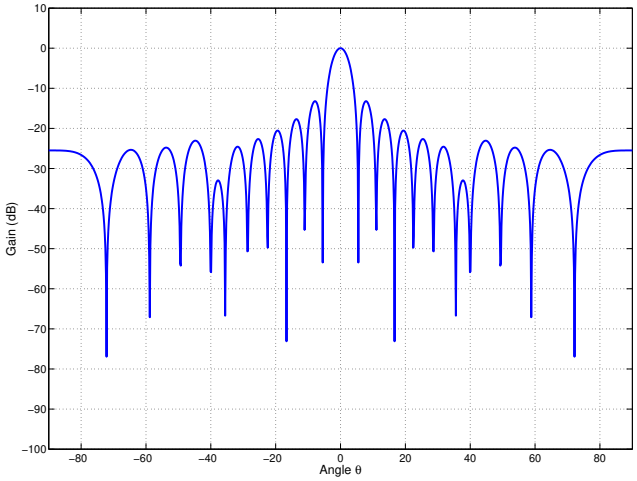
6.6.2 Convergence Behavior

Next, we show in Figure 6.6 the convergence behavior of the above three algorithms. The figure indicates the null depth of the beam pattern at the direction of the interference from 40° versus the number of samples. We can find that our proposed algorithm can achieve the same null depth (even a bit deeper) towards the interference direction as that of Frost beamformer although at a slower convergence rate. This is because that a number of degrees of freedom are used in the beginning to push down the sidelobe level instead of concentrating on interference nulling. As a result, in order to obtain an overall better-balanced beam pattern, a tradeoff of convergence speed has to be traded off.

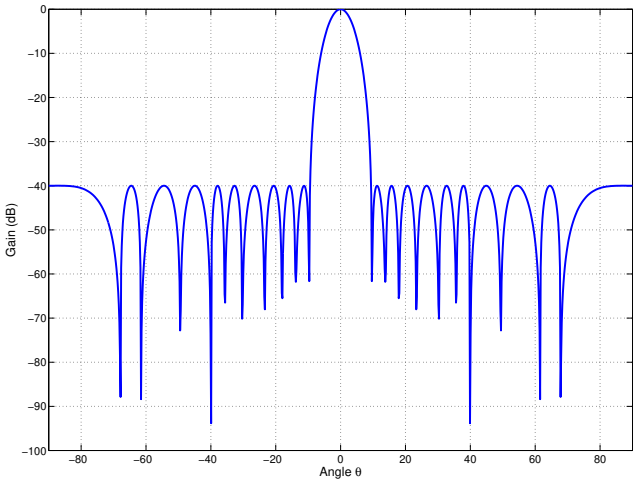
Compared with Zhou's algorithm, despite the relative slow convergence speed, ours can achieve a much deeper null at the interference direction in that our



(a) Output of beam pattern synthesis algorithm



(b) Output of Frost adaptive beamforming



(c) Output of our proposed algorithm

Figure 6.5: Comparison of various beam pattern outputs

proposed beamformer is required to satisfy the nulling constraints at interference directions while Zhou's algorithm only aims at pushing down the whole sidelobe under some level with no directive constraints on certain directions.

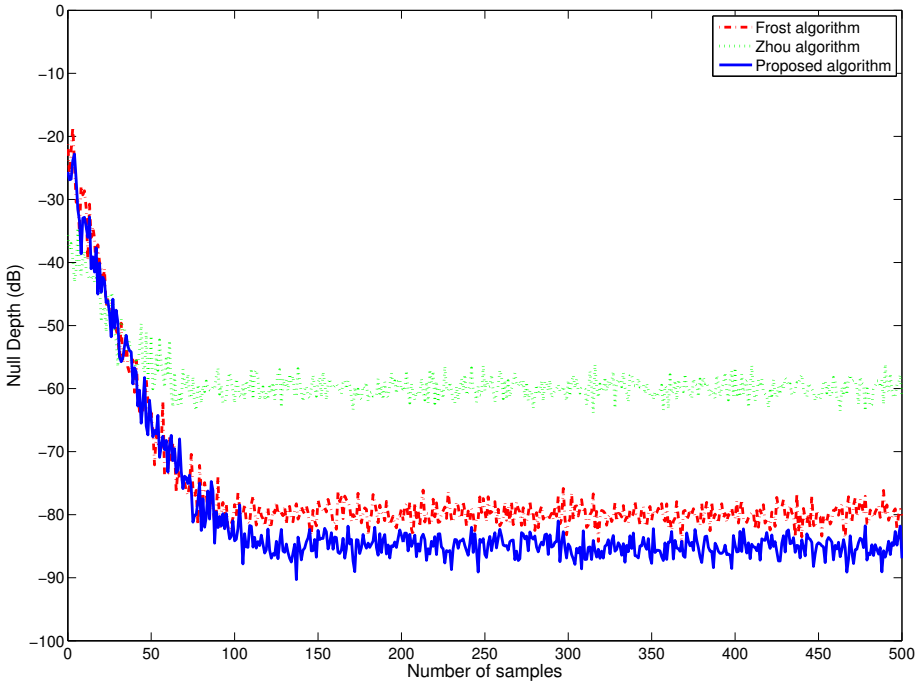


Figure 6.6: Convergence behavior of Frost algorithm, Zhou's algorithm and our proposed algorithm

6.6.3 SINR Performance

In Figure 6.7 we present the average output SINR of the three algorithms along with the optimal output SINR. From this figure, we can see that our proposed algorithm has the best output SINR among these three algorithms. Compared with Frost beamformer, our proposed algorithm wins out because it has a regular and low sidelobe beam pattern which enables it to null out interference and cancel out the noise effect at the same time. Although Frost beamformer produces

nulls towards interference as well, its irregular and comparatively high sidelobe region makes its output SINR inferior to that of our proposed algorithm. Zhou's algorithm simply guarantees a uniform sidelobe and doesn't incorporate the information of interferences into its beam pattern shaping process. Therefore it gives the worst SINR performance, especially in strong interference situations.

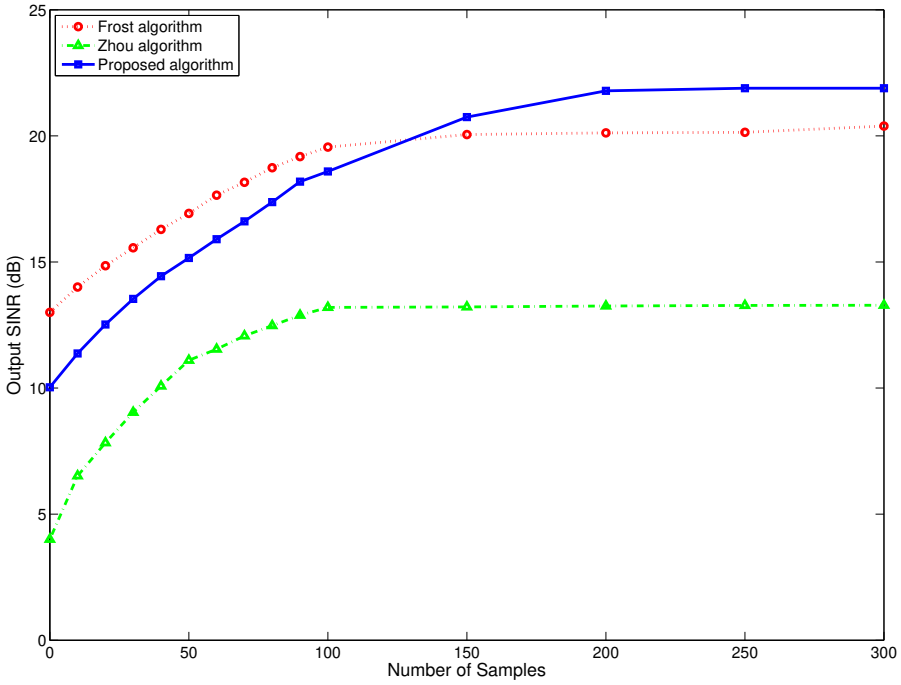


Figure 6.7: Average output SINR of Frost algorithm, Zhou's algorithm and our proposed algorithm

6.6.4 Choice of μ_1 and μ_2

In our proposed algorithm, μ_1 and μ_2 is closely related with the output performance. However as Zhou's algorithm is an ad-hoc approach, it is not easy to establish an overall optimal criterion on how to decide the value of μ_1 and μ_2 to obtain the best performance. Here we consider two examples to illustrate the

effects of these two parameters on the output performance and give a general guideline on the way to choose μ_1 and μ_2.

We consider two situations for this issue. The first one is the same as that described in the beginning of this section, and we name it "High Interference" situation. The second one is the same the "High Interference" situation except that the INR is -3dB instead, which we name as "High Noise" situation. Figure 6.8 shows the ratio of μ_1/μ_2 to the optimal output SINR. We find that in "High Interference" situation, the best performance is achieved when this ratio is set between 1/2 to 4. Output SINR will decrease when μ_1/μ_2 falls below this region in that compared with adaptive interference nulling, sidelobe pattern shaping is gaining more weight to combat with noise effect, which leads to a relative worse performance in such an strong interference environment. If μ_1/μ_2 rises above this region, chances are that μ_1 would not be able to satisfy the requirement for a stable algorithm, which would result in a diverged solution and a dramatically decreasing SINR. When it comes to the "High Noise" situation, a choice of μ_1/μ_2 between 1/4 to 4 will ensure the best performance. This result is reasonable as in noise limited situation, an overall low sidelobe level is more important to obtain a high SINR. So we recommend that μ_1/μ_2 should be between 1/4 and 4. The algorithm would take a risk of becoming instable with a large or small ratio. In case of high interference, more weights should be placed on adaptive interference nulling while in case of high noise, sidelobe shaping would be more emphasized.

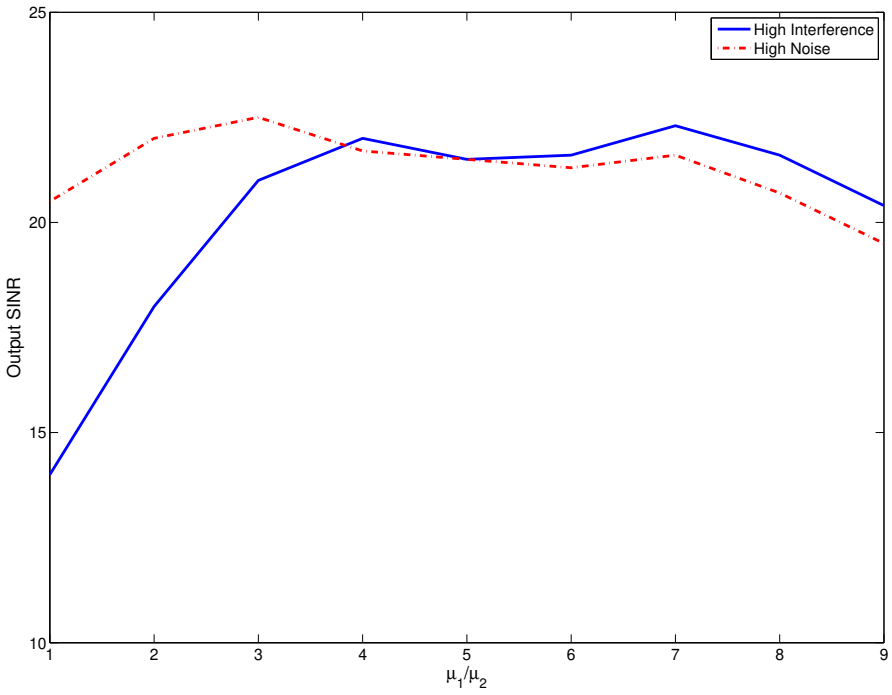


Figure 6.8: Average RMSE versus SNR for environment in case 1

Chapter 7

Conclusion and Future Work

This chapter summarizes the results of this thesis and describes open topics for future research.

7.1 Conclusion

This dissertation has addressed issues regarding DOA estimation and beam pattern synthesis techniques. Various problems are discussed and new algorithms are proposed to overcome the existing problems. Simulation examples and comparison with existing algorithms are given to illustrate the effectiveness of the proposed algorithms. The following is a summary of the contributions:

1. **DOA Estimation**: In this thesis, a new DOA estimation algorithm for cyclostationary signals is presented in chapter 4. In our proposed method, a new operator is introduced by combining all the cycle frequencies of SOIs together. This allows our proposed algorithm to select desired signals with different cyclostationary properties simultaneously and ignore the distur-

bance from undesired signals. In order to select SOIs and eliminate the interferences and noise, our algorithm only requires a moderate *a priori* information of the incoming signals of interests, such as their modulation type, baud rate and carrier frequencies. To make our proposed algorithm able to combat with multipath effects, we incorporate spatial smoothing, a preprocessing technique, with it. Theoretical analysis is presented to derive the necessary conditions on the size of subarray and number of subarrays to make our algorithm effective when it is combined with spatial smoothing. We compare our proposed algorithm with MUSIC and cyclic MUSIC in different signal environments, and the simulation results confirm the improvement and superiority in estimation performance with our proposed algorithm in both of the ordinary and multipath cases.

2. **Beam Pattern Synthesis Techniques**:

   (a) **Non-iterative Beam Pattern Synthesis Techniques**: In chapter 5, we propose a method which can achieve beam pattern shaping as well as interference and multipath nulling without *a priori* knowledge of their directions in a non-iterative manner. The main idea is to use a quadratically constrained minimization problem to control the main beam pattern and take advantage of the cyclostationary property of the transmission signal to ensure that the blocking matrix in the auxiliary beam lies in the interference subspace. Compared with other similar methods, ours can form much deeper nulls in the interference directions, resolve multipath components and keep a well-behaved beam pattern shape especially in environments with strong interference, which can help to alleviate the problem of co-channel

and multipath interference in modern wireless communications. Simulation examples show its effectiveness.

(b) **Iterative Beam Pattern Synthesis Techniques**: In chapter 6, we address a new composite adaptive beamforming algorithm with pattern control ability. The role of pattern control and adaptive beamforming can be emphasized by changing the step size μ_1 and μ_2. Frost adaptive beamformer and pattern synthesis algorithm are included in our algorithm and considered as special cases. Constrained optimization technique is used to overcome the calculation of matrix inverse which is frequently encountered in pattern synthesis algorithms. Householder Transform is also applied to make each weight update iteration operate on the minimum possible dimension. Computational costs are analyzed and compared with various relevant algorithms to show the saving of our proposed algorithm in computational load. Simulation examples confirm the effectiveness of our proposed algorithm. Finally, the effect of the choice of μ_1 and μ_2 on the output performance is briefly discussed and we give a general criterion on how to determine these two parameters.

7.2 Future Work

Regarding the above conclusions, there are several areas of this work that could be extended for future research:

1. The effect of mutual coupling among the array elements can be studied and its effect on beam pattern synthesis and DOA estimation can be

investigated.

2. Most algorithms proposed in this thesis use linear arrays. We can also consider the use of planar arrays to achieve a more directed beam. Such arrays are able to work in 3-D space, that is both azimuth and elevation angles of incoming signals can be estimated and any direction in space can be accurately resolved.

3. Real implementation of pattern synthesis algorithms can be considered based on the result of this work. We can compare the simulation results with the real application results to further improve our proposed algorithms.

4. In chapter 6, we proposed a new beamforming technique with pattern control ability. However, this algorithm was presented for narrowband operation. The technique could be extended to broadband operation for application to interference nulling for wireless communications.

5. The pattern synthesis algorithm that we develop in chapter 6 are applicable to any linear filter design problem, especially those that have mask-type specifications. For the case of digital filter design, we only need to redefine the steering vector in term of time-delayed responses and the other procedures are basically the same. This method can provide an alternative to existing digital filter design algorithms, particularly in cases where shaped frequency responses are needed.

Author's Publications

Journal papers

1. L. Qu and W. Ser, Z. Shao and M. Fujise "An efficient adaptive beamforming technique with pattern control ability," submitted to *IEEE Trans. Antennas & Propagations*

2. L. Qu and W. Ser, Z. Shao and M. Fujise "A New Direction-of-Arrival Estimation Algorithm for Cyclostationary Signals," submitted to *IEEE Trans. Signal Processing*

Conference papers

3. L. Qu, W. Ser, Z. Shao and M. Fujise "An efficient adaptive beamforming technique with pattern control ability," 3^{rd} *IEEE International Symposium on Wireless Communications Systems*, Sept. 2006

4. L. Qu, W. Ser, Z. Shao and M. Fujise "Beam pattern synthesis in the presence of interference and multipath," 64^{th} *IEEE Vehicular Technology Conference*, Sept. 2006

Bibliography

[1] C. A. Olen and R. T. Compton, "A numerical pattern synthesis algorithm for arays," *IEEE Trans. Antennas Propagat.*, vol. 38, pp. 1666 – 1676, Oct. 1990.

[2] K. L. Bell, H. L. Van Trees, and L. J. Griffiths, "Adaptive beampattern control using quadratic constraints for circular array,," *8th Annual Workshop on Adaptive Sensor Array Processing (ASAP2000)*, 2000.

[3] P. Zhou and M. Ingram, "Pattern synthesis for arbitrary arrays using an adaptive method," *IEEE Trans. Antennas Propagat.*, vol. 47, pp. 862 – 869, May 1999.

[4] S. L. Sim and M. H. Er, "An effective quiescent pattern control strategy for GSC structure," *IEEE Signal Processing Lett.*, vol. 3, pp. 236 – 238, Aug. 1996.

[5] J. Litva and T. K. Lo, *Digital Beamforming in Wireless Communications*, Artech, Boston, MA, 1996.

[6] J. C. Liberti and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*, Prentice Hall, Englewood Cliffs, NJ, 1999.

[7] L. C. Godara, "Applications of antenna arrays to mobile communications, Part II: Beamforming and direction-of-arrival considerations," *Proc. IEEE*, vol. 85, no. 8, pp. 1195 – 1245, Aug. 1997.

[8] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. 34, pp. 276 – 280, Mar. 1986.

[9] R. Roy and T. Kailath, "ESPRIT estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Signal Processing*, vol. 37, pp. 984 – 995, Jul. 1989.

[10] G. Xu, S. D. Silverstein, R. H. Roy, and T. Kailath, "Beamspace ESPRIT," *IEEE Trans. Signal Processing*, vol. 42, pp. 349 – 356, Feb. 1994.

[11] M. L. McCloud and L. L. Scharf, "A new subspace identification algorithm for high resolution DOA estimation," *IEEE Trans. Antennas Propagat.*, vol. 50, pp. 1382 – 1390, Oct. 2002.

[12] N. Tayem and H. M. Kwon, "Conjugate ESPRIT," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 2618 – 2624, Oct. 2004.

[13] S. V. Shell, R. A. Calabretta, W. A. Gardner, and B. G. Agee, "Cyclic MUSIC algorithms for signal selective doa estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1989, pp. 2278 – 2281.

[14] Q. Guo, G. Liao, Y. Wu, and J. Li, "Pattern synthesis method for arbitrary arrays based on LCMV criterion," *IEE Electronics Lett.*, vol. 39, pp. 1628 – 1630, Nov. 2003.

[15] C. Tseng and L. J. Griffiths, "A simple algorithm to achieve desired patterns for arbitrary arrays," *IEEE Trans. Signal Processing*, vol. 40, pp. 2737 – 2746, Nov. 1992.

[16] B. P. Ng, M. H. Er, and C. Kot, "A flexible array synthesis method using quadratic programming," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 1541 – 1550, Nov. 1993.

[17] M. H. Er, "Array pattern synthesis with a controlled mean-square sidelobe level," *IEEE Trans. Antennas Propagat.*, vol. 40, pp. 977 – 981, Apr. 1992.

[18] H. Lebret and S. Boyd, "Antenna array pattern synthesis via convex optimization," *IEEE Trans. Signal Processing*, vol. 45, pp. 526 – 532, May 1997.

[19] F. Wang, V. Balakrishnan, P. Zhou, J. Chen, R. Yang, and C. Frank, "Optimal array pattern synthesis using semidefinite programming," *IEEE Trans. Signal Processing*, vol. 51, pp. 1172 – 1183, May 2003.

[20] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. 30, pp. 27 – 34, Jan. 1982.

[21] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, pp. 926 – 935, Aug. 1972.

[22] L. C. Godara, "Applications of antenna arrays to mobile communications, Part I: Performance improvements, feasibility and system considerations," *Proc. IEEE*, vol. 85, no. 7, 1997.

[23] P. K. Bondyopadhyay, "The first application of array antenna," in *Proceedings of IEEE International Conference on Phased Array Systems and Technology*, 2000, pp. 29 – 32.

[24] H. Dam and M. Burg, "Performance evaluation of adaptive antenna base stations in a commercial gsm network," in *Proceedings of Vehicular Technology Conference*, 1999, pp. 47 – 51.

[25] K. Sheikh, D. Gesbert, D. Gore, and A. J. Paulraj, "Smart antennas for broadband wireless access network," *IEEE Signal Processing Mag.*, vol. 37, no. 11, pp. 100 – 105, Nov. 1999.

[26] R. T. Derryberry, S. D. Gray, and D. M. Ionescu, "Transmission diversity in 3G CDMA systems," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 68 – 75, Apr. 2002.

[27] S. Andersson and U. Forssen, "Ericsson GSM field trials with adaptive antennas," in *Proceedings of Vehicular Technology Conference*, 1997, pp. 1587 – 1591.

[28] G. Tsoulos, M. Beach, and J. McGeehan, "Space division multiple access field trials Part 2: Calibration and linearity issues," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 145, no. 1, pp. 79 – 84, Feb. 1998.

[29] G. V. Tsoulos, "Smart antennas for mobile communication systems," *Electronics and Communication Engineering Journal*, vol. 11, pp. 84 – 94, Apr. 1999.

[30] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143 – 2159, Dec. 1967.

[31] W. F. Gabriel, "Adaptive array - an introduction," *Proc. IEEE*, vol. 64, pp. 239 – 272, Feb. 1976.

[32] J. E. Hudson, *Adaptive Array Principle*, Peter Peregrinus Ltd., Stevenage, UK, 1981.

[33] R. T. Compton, *Adaptive Antennas: Concepts and Performance*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[34] P. H. Lehne and M. Pettersen, "An overview of smart antenna technology for mobile communications system," *IEEE Communications Surrey*, vol. 2, pp. 2 – 13, Fourth Quarter 1999.

[35] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Signal Processing Mag.*, vol. 5, pp. 4 – 24, Apr. 1988.

[36] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, John Wiley & Sons, New York, 1980.

[37] H. L. Van Trees, *Optimum Array Processing*, Wiley-Interscience, New York, 2002.

[38] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, N.J., 2002.

[39] G. H. Golub, *Matrix Computations*, Johns Hopkins University Press, 1996.

[40] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Processing Mag.*, vol. 13, pp. 67 – 94, Jul. 1996.

[41] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 85, pp. 1031 – 1060, Jan. 1969.

[42] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561 – 580, Apr. 1975.

[43] M. I. Miller and D. R. Fuhrmann, "Maximum-likelihood narrow-band direction finding and the em algorithm," *IEEE Trans. Signal Processing*, vol. 38, pp. 1560 – 1577, Sept. 1990.

[44] M. Wax and T. Kailath, "Optimum localization of multiple sources by passive arrays," *IEEE Trans. Signal Processing*, vol. 31, pp. 1210 – 1221, Oct. 1983.

[45] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. Signal Processing*, vol. 36, pp. 1553 – 1560, Oct. 1988.

[46] H. Lee and R. Stovall, "Maximum likelihood methods for determining the direction of arrival for a single electromagnetic source with unknown polarization," *IEEE Trans. Signal Processing*, vol. 42, pp. 474 – 479, Feb. 1994.

[47] Van Der Veen, E. F. Deprettere, and A. L. Swindlehurst, "Subspace-based signal analysis using singular value decomposition," *Proc. IEEE*, vol. 81, pp. 1277 – 1308, Sept. 1993.

[48] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. 34, pp. 276 – 280, Mar. 1986.

[49] D. W. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: making linear prediction perfrom like maximum likelihood," *Proc. IEEE*, vol. 70, pp. 975 – 989, Sept. 1982.

[50] J. P. Burg, *Maximum Entropy Spectral Estimation*, Ph.D. Dissertation, Stanford University, Stanford CA, 1975.

[51] V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophys. J. R. Astron. Soc*, vol. 33, pp. 347 – 366, 1973.

[52] S. Y. Kung, K. S. Arun, and B. D. Rao, "New state space and singular value decomposition based approximation methods for harmonic retrieval," *Journal of the Optical Society of America*, vol. 73, pp. 1799 – 1811, 1983.

[53] A. Paulraj, R. H. Roy, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques-ESPRIT," in *Proc. of the 19th Asilomar Conference on Circuits Systems and Computers*, 1985, pp. 83 – 89.

[54] H. Ouibrahim, D. D. Weiner, and T. K. Sarkar, "Matrix pencil approach to angle of arrival estimation," in *Proc. of the 20th Asilomar Conference on Circuits Systems and Computers*, 1986, pp. 203 – 206.

[55] W. A. Gardner, *Cyclostationarity in Communications and Signal Processing*, IEEE Press, New York, 1993.

[56] T. J. Shan, M. Wax, and T. Kailath, "On spatial smoothing for direction-of-arrival estimation of coherent signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 806 – 811, Aug. 1985.

[57] J. Xin, H. Tsuji, Y. Hase, and A. Sane, "Direction-of-arrival estimation of cyclostationary coherent signals in array processing," *IEICE Trans. Fundamentals*, vol. E81-A, pp. 1560 – 1569, Aug. 1998.

[58] Y. T. Lee and J. H. Lee, "Direction-finding methods for cyclostationary signals in the presence of coherent sources," *IEEE Trans. Antennas Propagat.*, vol. 49, pp. 1821 – 1826, Dec. 2001.

[59] W. A. Gardner, *Introduction to Random Processes with Applications to Signals and Systems*, Macmillan Publisher, New York, 1986.

[60] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Signal Processing*, vol. 33, pp. 387 – 392, Apr. 1985.

[61] C. L. Dolph, "A current distribution for broadside arrays which optimizes the relationship between beam width and sidelobe leve," *Proc. IEEE*, vol. 34, pp. 335 – 348, Jun. 1946.

[62] L. J. Griffiths and K. M. Buckley, "Quiescent pattern control in linearly constrained adaptive array," *IEEE Trans. Signal Processing*, vol. 35, pp. 917 – 926, Jul. 1987.

[63] S. P. Applebaum, "Adaptive arrays," *IEEE Trans. Antennas Propagat.*, vol. 24, pp. 585 – 598, Sept. 1976.

[64] M. L. R. de Compos, S. Werner, and J. A. Apolinario, "Constrained adaptation algorithms employing householder tranformation," *IEEE Trans. Signal Processing*, vol. 50, pp. 2187 – 2195, Sept. 2002.