DIRECTION OF ARRIVAL ESTIMATION BY USING ALTERNATING DIRECTION METHOD OF MULTIPLIERS IN DISTRIBUTED SENSOR ARRAY NETWORKS

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

BY

EKIN NURBAŞ

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

AUGUST 2022

Approval of the thesis:

DIRECTION OF ARRIVAL ESTIMATION BY USING ALTERNATING DIRECTION METHOD OF MULTIPLIERS IN DISTRIBUTED SENSOR ARRAY NETWORKS

submitted by **EKIN NURBAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Prof. Dr. Temel Engin Tuncer
Supervisor, **Electrical and Electronics Engineering** _____

Dr. Emrah Onat
Co-supervisor, **Siliconally GmbH** _____

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering, METU _____

Prof. Dr. Temel Engin Tuncer
Electrical and Electronics Engineering, METU _____

Prof. Dr. Umut Orguner
Electrical and Electronics Engineering, METU _____

Prof. Dr. Cenk Toker
Electrical and Electronics Engineering, Hacettepe University _____

Assoc. Prof. Dr. Sevinç Figen Öktem
Electrical and Electronics Engineering, METU _____

Date: 26.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ekin NURBAŞ

Signature :

ABSTRACT

DIRECTION OF ARRIVAL ESTIMATION BY USING ALTERNATING DIRECTION METHOD OF MULTIPLIERS IN DISTRIBUTED SENSOR ARRAY NETWORKS

NURBAŞ, Ekin
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Temel Engin Tuncer
Co-Supervisor: Dr. Emrah Onat

August 2022, 73 pages

In recent years, developments in microprocessor and wireless communication technologies have benefited a variety of distributed sensor network applications, including array signal processing. Researchers have been investigating distributed implementations of array signal processing algorithms, such as Direction of Arrival Estimation, for a variety of applications. Performance and practical implementations of those algorithms are affected by a variety of factors, such as inter-array phase and frequency matching and array geometry dependencies. Using the Alternating Direction Method of Multipliers (ADMM) in the Sparse Bayesian Learning (SBL) framework, we provide a new method for Direction of Arrival (DoA) estimation in distributed sensor array networks. Our new method, Collaborative Direction of Arrival Estimation (CDoAE), has a number of advantages over earlier distributed DoA estimate techniques. It does not necessitate any particular array geometry or inter-array frequency and phase matching. To minimize a shared objective function, CDoAE employs the distributed ADMM to update the parameter set extracted by the SBL frameworks in the local arrays. The master-node performs this update procedure, and the result

is sent to the slave nodes. It is demonstrated that the performance of local arrays can be greatly enhanced using this distributed DOA estimation method. Furthermore, we provide a method, CDoAE-TVR, for reducing the amount of parameters broadcast from the local arrays to the master array, which is crucial for networks with restricted bandwidth and energy. Several simulations, including scenarios with coherent sources, have been done. It is demonstrated that the widespread use of ADMM improves the effectiveness of the SBL output. In addition, the suggested CDoAE-TVR technique minimizes the transmitted parameters at the cost of a minor reduction in DoA estimation performance.

Keywords: Direction of Arrival, DoA, Alternating Direction Method of Multipliers, ADMM, Sparse Bayesian Learning, SBL, Distributed Processing, Distributed DoA Estimation

ÖZ

ÇARPANLARIN ALTERNATİF YÖN YÖNTEMİ KULLANILARAK DAĞITIK SENSOR DİZİSİ AĞLARINDA İŞARET GELİŞ AÇISI KESTİRİMİ

NURBAŞ, Ekin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Temel Engin Tuncer
Ortak Tez Yöneticisi: Dr. Emrah Onat

Ağustos 2022 , 73 sayfa

Son yıllarda, mikroişlemci ve kablosuz iletişim teknolojilerindeki gelişmeler, dizi sinyal işleme de dahil olmak üzere çeşitli dağıtılmış sensör ağı uygulamalarına oldukça fayda sağlamıştır. Araştırmacılar geçmişten günümğüze çeşitli uygulamalar için İşaret Geliş Açısı Tahmini gibi dizi sinyal işleme algoritmalarının dağıtılmış uygulamalarını araştırmakta olup bu kapsamda çeşitli çalışmalar ve algoritmalar yer sunmuşlardır. Bu algoritmaların performansı ve pratik uygulamaları, diziler arası faz ve frekans eşleştirme ve dizi geometrisi bağımlılıkları gibi çeşitli faktörlerden etkilenmektedir. Bu çalışmada ise, Seyrek Bayes Öğrenmesi (SBL) çerçevesindeki Çarpanların Alternatif Yönü Yöntemini (ADMM) kullanılarak, dağıtılmış sensör dizisi ağlarında İşaret Geliş Açısı tahmini için yeni bir yöntem sunulmuştur.Yeni yöntemimiz CDoAE, daha önceki dağıtılmış DoA tahmin tekniklerine göre bir takım avantajlara sahiptir. Yönrem, belirli bir dizi geometrisi veya diziler arası frekans ve faz eşleştirmesi gerektirmeksizin çalışabilemkte olup, dağıtık bir amaç fonksiyonunu en aza indirmek ve yerel dizilerdeki SBL parametre setini güncellemek için dağıtılmış ADMM'yi kul-

lanmaktadır. Çalışamda, önerilen dağıtılmış DOA tahmin yönteminin yerel dizilerin performansının büyük ölçüde artırılabileceği gösterilmiş olup, sınırlı bant genişliği ve enerjiye sahip ağlar için de yerel dizilerden ana diziye gönderilen parametre miktarını azaltan CDoAE-TVR yöntemi sunulmuştur. Evreuyumlu kaynaklara sahip senaryolar da dahil olmak üzere çeşitli simülasyonlar yapılmıştır. Önerilen yöntemin yerel dizilerin performansını ve etkinliğini geliştirdiği gösterilmiş olup, benzer yçntemlere de üstünlük kurduğu görülmüştür.

Anahtar Kelimeler: İşaret Geliş Açısı, DoA, Çarpanların Alternatif Yönü Yöntemini, ADMM, Seyrek Bayes Öğrenmesi, SBL, Dağıtık İşleme, Dağıtık İşaret Geliş Açıs Kestirimi

In dedication to my love Kardelen Deveci and my beloved family; Hatice Nurbaş and Prof. Dr. Macid Nurbaş...

ACKNOWLEDGMENTS

I am extremely thankful to my supervisor Prof. Dr. Temel Engin Tuncer for his noble guidance, patience, support with full encouragement and enthusiasm. I am grateful to Dr. Emrah Onat for his valuable suggestions, ever encouraging and motivating. Last but not the least I would also like to thank Kardelen Deveci for her limitless love and support, I would like to express my gratitude to my beloved family Hatice Nurbaş and Prof. Dr. Macid Nurbaş for encouraging and supporting me from the day I was born.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xviii

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation and Problem Definition 1

1.2 Contributions and Novelties 3

1.3 The Outline of the Thesis 4

2 DISTRIBUTED SENSOR NETWORKS 5

2.1 Distributed Sensor Networks 5

2.1.1 Distributed Sensor Network Applications 6

2.1.1.1 Monitoring Applications 7

2.1.1.2 Tracking Applications 8

3 DIRECTION OF ARRIVAL ESTIMATION 9

3.1 Problem Definition 9
3.2 Signal Model 9
3.2.0.1 Far-field Source Assumption 11
3.2.0.2 Narrowband Assumption 11
3.2.0.3 Homogeneous Propagation Medium 12
3.2.0.4 Literature Survey 15
4 DIRECTION OF ARRIVAL ESTIMATION IN DISTRIBUTED SENSOR NETWORKS 17
4.1 Problem Definition and Data Model 17
4.2 Literature Survey 19
5 SINGLE ARRAY DOA ESTIMATION IN SPARSE BAYESIAN LEARNING FRAMEWORK 21
5.1 Single Array DoA Estimation in Sparse Bayesian Learning Framework 21
5.1.1 Sparse Signal Model 21
5.1.2 Parameter Update through EM Algorithm 25
5.1.2.1 Hidden Variable Update 25
5.1.2.2 Hyperparameter Update 26
6 COLLABORATIVE DOA ESTIMATION (CDOAE) 31
6.1 Collaborative DoA Estimation (CDoAE) 31
6.1.1 Distributed ADMM Algorithm 32
6.1.2 Collaborative Variable Update 35
6.1.2.1 Hidden Variable Update 36
6.1.2.2 Hyper Parameter Update 36
6.1.2.3 Weight Function for the Local Estimates 38

6.1.2.4 Global Variable Update 41
6.1.2.5 Dual Variable Update 41
6.1.2.6 Stopping Criteria 41
6.2 Computational Complexity of CDoAE 42
6.3 Parameter Exchange Between Master and Local Nodes 43
6.3.1 Transferred Variable Reduction Method 44
7 SIMULATIONS 45
7.1 Simulations 45
7.1.1 Performance Analysis for Uniform Planar Arrays 46
7.1.1.1 Single Source Scenario 47
7.1.1.2 Multiple Uncorrelated Source Scenario 51
7.1.1.3 Multiple Coherent Source Scenario 53
7.1.2 2D DoA Estimation for Multiple Coherent Sources 56
7.1.3 Performance Analysis with Random Arrays 58
7.1.3.1 Multiple Uncorrelated Source Scenario 61
7.1.3.2 Multiple Coherent Source Scenario 62
8 CONCLUSION 67
8.1 Conclusion 67
REFERENCES 69

LIST OF TABLES

TABLES

Table 6.1 Total number of multiplication per iteration for SBL, ADMM and CDoAE 42

Table 7.1 Simulation parameters for Single Source Scenario 48

Table 7.2 Simulation parameters for Multiple Uncorrelated Sources Scenario . 52

Table 7.3 Simulation parameters for Multiple Coherent Sources Scenario . . . 53

Table 7.4 Simulation parameters for Multiple Coherent Sources Scenario . . . 57

Table 7.5 Simulation parameters for Multiple Uncorrelated Sources Scenario with Randomized Array Geometry 61

Table 7.6 Simulation parameters for Multiple Coherent Sources Scenario with Randomized Array Geometry 63

LIST OF FIGURES

FIGURES

Figure 2.1 Distributed Sensor Network Operation Flow 6

Figure 2.2 Distributed Sensor Networks Applications 7

Figure 3.1 Coordinate system, sensor array and source position for 2D DoA estimation 10

Figure 3.2 Coordinate system, sensor array and source position for 1D DoA Estimation 11

Figure 4.1 Distributed Sensor Array Network 18

Figure 5.1 Possible source directions in azimuth, θ, and elevation, ψ and true source directions represented with red circles. 22

Figure 5.2 Sensor array and possible source directions with complex Gaussian distribution for Sparse Bayesian Learning framework. 23

Figure 5.3 Normalized variance spectrum obtained by SBL method and true DoAs represented with dashed lines 29

Figure 6.1 Sensor nodes and variable exchange diagram for Distributed Alternating Direction Method of Multipliers. 34

Figure 6.2 Sensor Node 1,2 and their braodside and endfire regions. 38

Figure 6.3 Approximate Cramer Rao Bound and the Weighting Function for URA. 40

Figure 7.1 Two sensor nodes with URA with a separation of D distance are shown in addition to source direction represented with a dashed line. . . 47

Figure 7.2 Weight functions for Array 1 and Array 2 49

Figure 7.3 RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL, SBL-AVG and non-coherent CRB for single source. 50

Figure 7.4 Two sensor nodes with URA with a separation of D distance are shown in addition to source directions represented with a dashed line. . 51

Figure 7.5 RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL-AVG and non-coherent CRB for multiple uncorrelated sources. 52

Figure 7.6 RMSE-SNR curves for Decentralized MUSIC, CDoAE, CDoAE-TVR(a), CDoAE-TVR(b) and CRB for multiple coherent sources. . . . 54

Figure 7.7 Percentage of average total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b) . . 55

Figure 7.8 Percentage of total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b) for each iteration 56

Figure 7.9 RMSE-SNR curves for Decentralized MUSIC, CDoAE and non-coherent CRB for 2D DoA estimation. 57

Figure 7.10 Sensor node one is shown and source directions represented with dashed line 59

Figure 7.11 Sensor node two is shown and source directions represented with dashed line 60

Figure 7.12 RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL and CRB for multiple uncorrelated sources. 62

Figure 7.13 RMSE-SNR curves for Decentralized MUSIC, CDoAE, CDoAE-TVR(a), CDoAE-TVR(b) and CRB for multiple coherent sources. . . . 64

Figure 7.14 Percentage of average total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b) . . 65

LIST OF ABBREVIATIONS

1D	1 Dimensional
2D	2 Dimensional
DoA	Direction of Arrival
DF	Direction Finding
RF	Radio Frequency
MUSIC	Multiple Signal Classification
ESPRIT	Estimation of Signal Parameters via Rotational Invariant Techniques
DSN	Distributed Sensor Network
SBL	Sparse Bayesian Learning
EM	Expectation Maximization
ADMM	Alternating Direction Method of Multipliers
DADMMM	Distributed Alternating Direction Method of Multipliers
CDoAE	Collaborative Direction of Arrival Estimation
CDoAE-TVR	Collaborative Direction of Arrival Estimation - Transferred Variable Reduction

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Direction of Arrival (DoA) estimation is a topic that has been studied by researchers for many years and plays an essential role in various fields such as radar, sonar, and communication systems. In the literature there are many studies that investigates variety of problems in DoA estimation. One important problem is the DoA estimation for coherent sources without having any dependency on array geometry. Recently, Sparse Bayesian Learning (SBL) method [1], [2], is presented to solve this problem. SBL method is investigated for different problems in DOA estimation. In [3],[4] gain/phase and mutual coupling problems are considered within the SBL framework. Off-grid DoA estimation by the SBL method is presented in [5]. Most of the previous SBL based approaches consider a single array. In this thesis, the advantages of SBL method are used for a distributed sensor array structure for DoA estimation.

On the other hand, DoA estimation for distributed sensor arrays raised significant interest within the community in the recent years. While there are different possibilities for the distributed sensor array configurations, we will mainly focus on a structure which involves a master and local sensor arrays. Each sensor array can independently estimate DoA and transfer certain parameters to the master array. The function of the master array is to obtain a global solution which is superior to the local results. In the literature there are several studies that covers the DoA estimation in distributed sensor arrays with the configuration defined above. In [6], a synchronization method for resolving the inter array time and frequency offsets as well as jointly estimating the DoA angles and phase offsets is presented. The main disadvantage of this method

is the requirement to know either the signal or a part of the signal for accurate estimation. Also, raw data for each sensor array should be transferred to the master node. In [7] and [8], a method for estimating the complete covariance matrix of the whole array is presented. Raw data is not transferred to the master node but the inter-array synchronization for time, frequency and phase must be done before processing.

In [9], Decentralized MUSIC algorithm is presented with several advantages. In this method, MUSIC algorithm is extended for the distributed sensor array networks by sharing only the covariance matrices of the local nodes to the master node. There is no need to perform gain/phase and frequency matching between the local arrays. Decentralized MUSIC requires Forward-Backward Spatial Smoothing [10] for coherent sources. Hence a special array structure is required for this purpose. Also the algorithm needs to know the sensor positions of each array at the master node. Search-free alternative of Decentralized MUSIC algorithm is proposed in [11].

On the other hand, decentralized implementation of ESPRIT method is proposed in [12], [13] for DoA estimation with multiple sensor arrays. In these methods, the signal subspace of the observed signal is estimated in a distributed manner. Researchers have shown that the Decentralized ESPRIT (D-ESPRIT) can perform accurate DoA estimations in distributed sensor array settings but it is only applicable for the networks which consist of arrays with shift invariant properties. In [14] D-ESPRIT method is extended for the arbitrary array geometries by using array interpolation but it is known that the interpolation error directly effects the performance of the DoA estimation.

In [15], distributed form of Alternating Direction Method of Multipliers(ADMM) is proposed. ADMM can be used to obtain global optimum solution for convex problems. In addition, it is shown that its performance for non-convex problems is good compared to alternative methods [16]. ADMM is used in DoA estimation in different works [17], [18]. In [17], the problem is separated into smaller parts to be able to process them in different processing cores. In [18], DoA cost function is divided into two functions and solved using the ADMM algorithm. But none of these studies covers the DoA estimation in Distributed Sensor Networks, instead they shown that DoA estimation problem for single sensor array can be decomposed into multiple

parts where each part of the problem can be handled in a different processors.

In this thesis, a new method, Collaborative DoA Estimation (CDoAE), in distributed sensor arrays is proposed. CDoAE combines the advantages of SBL and ADMM to obtain an effective method with certain advantages. CDoAE does not require gain/phase and frequency matching for the sensor arrays. There is no need to know the sensor positions and only a set of parameters are transferred from the local arrays to the master array. In addition, CDoAE-Transferred Variable Reduction (CDoAE-TVR) method is presented to reduce the number of transferred parameters from the local arrays to the master array.

1.2 Contributions and Novelties

Contributions in this thesis are listed as follows:

- A new method for collaborative DoA estimation with distributed sensor array network by using SBL framework and ADMM algorithm is presented.
- This new method employs SBL algorithm by modifying its structure with the distributed ADMM framework.
- While estimating the DoA of the incoming signals in a collaborative manner, algorithm does not require a phase and frequency matching between the distributed sensor arrays.
- Also, there is no need to know the sensor positions of the slave nodes in the master node as well as the other slave nodes.
- Lastly, the required total number of variables that need to be transferred in the network is reduced with the proposed Data Transfer Reduction method and when compared to the phase and frequency matching algorithms it requires less amount of data transfer between distributed sensor arrays.

1.3 The Outline of the Thesis

In Chapter 2, Distributed Sensor Network (DSN), DSN types, and DSN applications will be explained using examples from the literature. In Chapter 3, problem definition for the Direction of Arrival(DoA) estimation will be given. Moreover, the signal model for the DoA estimation problem will be presented with the assumptions in detail. Afterwards the literature survey for the DoA estimation algorithms will be given. In Chapter 4, the two concepts given in the previous chapters, DSN and DoA estimation, will be combined to address the problem of DoA estimation in distributed sensor array networks. In addition, the signal model for the DoA estimation in distributed sensor arrays will be introduced as well as the literature survey. In Chapter 5, Sparse Bayesian Learning(SBL) which is one of the key elements for the proposed Collaborative Direction of Arrival Estimation(CDoAE) method will be introduced. In this chapter, SBL method for DoA estimation will be explained only for the case where there exists a single sensor array. Extension of the SBL method for distributed sensor array networks will be proposed in Chapter 6 where the proposed CDoAE algorithm is introduced. In Chapter 6, proposed CDoAE method which uses the distributed Alternating Direction Method of Multipliers(ADMM) approach to modify the variable update procedure of the SBL is introduced as well as the distributed implementation of the ADMM method. Moreover, the proposed CDoAE - Transferred Variable Reduction method(CDoAE-TVR) will be introduced to reduce the number of variables transferred from the local nodes to the master node with a small sacrifice on the achieved performance. In Chapter 7, the performances of the proposed algorithms will be investigated for different scenarios including Uniform Rectangular Array(URA) geometries, Randomized array geometries, single source scenarios, multiple source scenarios, and non-coherent and coherent source scenarios. In Chapter 8, the discussions and possible future works related with the thesis will be given.

CHAPTER 2

DISTRIBUTED SENSOR NETWORKS

This section explains the Distributed Sensor Network (DSN) idea, DSN types, and DSN applications using examples from the literature.

2.1 Distributed Sensor Networks

A distributed sensor network can be defined as a network that is composed of elements called as nodes that are capable of processing and communicating with one another. Alternatively, a distributed sensor network could be described as a network of autonomous pieces that are able to process data and function as a coherent system. According to the given characteristics, DSN possesses the capability of operating as a single system when all of the nodes in the network are cooperating with one another, in addition to the capability of allowing each node in the network to function as an independent system. According to the definitions that have been presented, DSN can be applied in a wide variety of fields where collaborative integration of local observations is required. Since the network nodes are the ones that collect and analyze the local data, DSN makes it possible for the computing effort to be split up and disseminated across the network. Because the information that is obtained at the nodes is shared with one another, it is possible to combine analogous observations from a variety of locations or different kinds of information in order to produce a higher quality outcome. Figure 2.1 is a diagram that illustrates how a DSN performs its operations. During the step named as "Signal Processing," the data that is captured by the nodes is processed independently by each node. In the process that comes after, which is called "Collaborative Signal Processing," the individual results are combined and

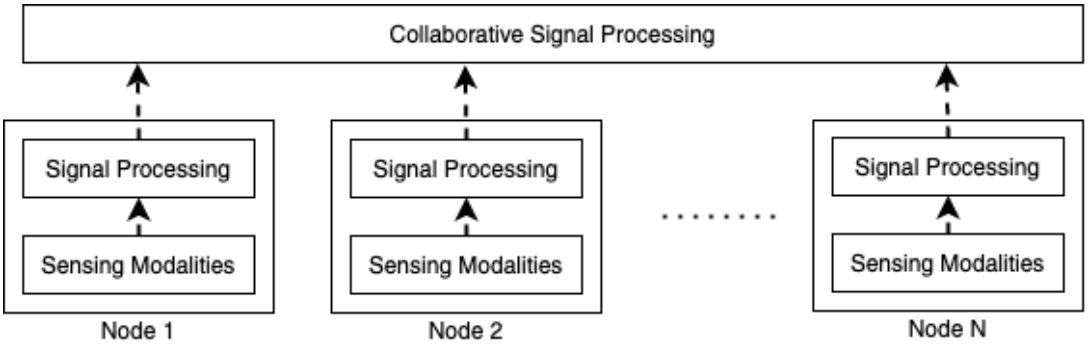


Figure 2.1: Distributed Sensor Network Operation Flow

used to generate the final output.

2.1.1 Distributed Sensor Network Applications

Sensor technology advancements such as MEMS, wireless communications, embedded systems, distributed processing, and wireless sensor applications have lately led to a significant shift in Distributed Sensor Network (DSN). With these improvements, DSNs become remarkably popular in a variety of fields as well as the research literature. The fact that DSNs provide a significant degree of flexibility in terms of resource sharing, scalability, and fault tolerance is one of the primary reasons why they are used by such a diverse variety of applications. Figure 2.2 depicts how these applications could be classified based on the domains that they serve.

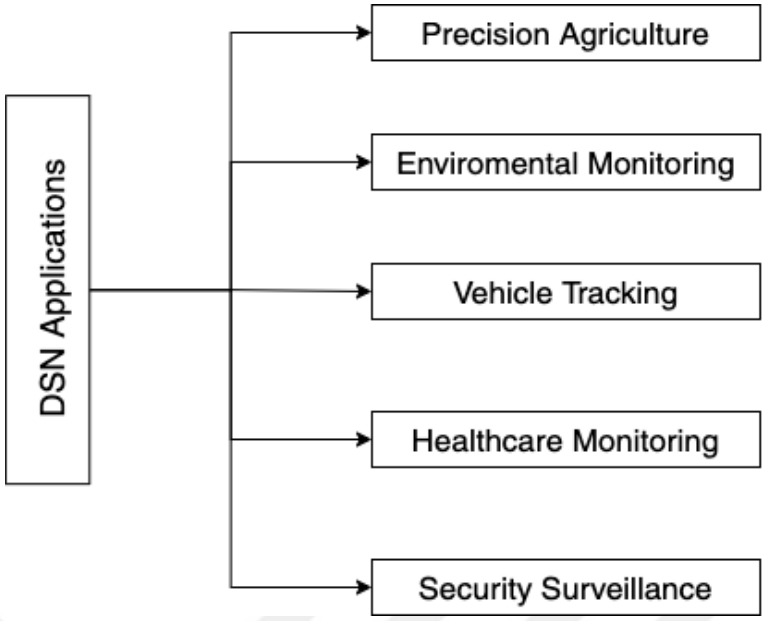


Figure 2.2: Distributed Sensor Networks Applications

It is seen from Figure 2.2, applications in Distributed Sensor Networks can be categorized under two main categories such as monitoring and tracking. In the monitoring category, applications can vary from health monitoring to environmental monitoring. There are many studies which cover power monitoring, factory monitoring, process automation, and seismic monitoring. On the other hand, the tracking category consists of many tracking applications such as animal tracking, vehicle tracking and object tracking [19].

2.1.1.1 Monitoring Applications

With the improvements in the sensor technology as well as the wireless communication, distributed sensor networks become widely used in the medical field. There are many applications for different cases including patient monitoring, disease diagnosis, hospital management, remote patient monitoring [19].

With the help of the cost effective and rapid deployment properties of the distributed sensor networks in addition to its robustness and livability abilities, distributed sensor networks are becoming popular even in the harsh environments such as farms and fields. Distributed Sensor Networks are utilized in the agricultural monitoring area

consisting of many different applications including monitoring environmental conditions, precision agriculture monitoring, detection of forest fires, pollution monitoring, rainfall monitoring, water level monitoring and so on [20], [21], [22],[19].

Since the nodes in the distributed sensor networks are communicating with each other and have the capability to operate as a single system, it is an effective solution to the smart building applications. In those applications, buildings are configured with the sensors and they are linked to each other or a single center so that the information obtained at each node can be gathered to provide as much convenience and comfort for the occupants [23], [19].

2.1.1.2 Tracking Applications

Commercial and military applications require target tracking. Situational awareness on the battlefield requires precise and fast vehicle targeting. Networked sensors are often well-suited for tracking due to their spatial coverage and sensing diversity. By utilizing the geographical and sensory variety of a large number of sensors, the network can reach at a global estimation by combining information from scattered sources [24]. Asset tracking is one of the many potential military and commercial applications for wireless sensor networks. Researchers have described a potential application in [25] for the tracking of shipping containers on both land and sea vessels. Another possible usage is for friendly troops to identify and track themselves [26].

CHAPTER 3

DIRECTION OF ARRIVAL ESTIMATION

3.1 Problem Definition

The process of determining the direction of many electromagnetic waves/sources from the outputs of a number of receiving antennas that constitute a sensor array is referred to as direction-of-arrival estimation (DOA). DOA estimation is a significant topic in array signal processing and has numerous applications in radar, sonar, wireless communications, and other fields [27].

DOA estimation is usually researched as part of the broader area of array processing and that investigates the problem of determining the source direction of the incoming waves. The type of wave of interest could be an electromagnetic wave or a seismic wave, depending on the application [28].

Radio direction finding, also known as estimating the direction of electromagnetic waves that are impinging on a single antenna or an antenna array, occupied a significant portion of the early research that was conducted on this topic. Additionally, extensive research has been done on the topic of acoustic direction of arrival estimation, and this has mostly been done in the context of sonar [28].

3.2 Signal Model

In a typical application, an incoming wave is detected by an array, and the associated signals at different sensors in space are analyzed to determine the direction of arrival (DoA) of the incoming signal. At each sensor the signal of interest is captured with

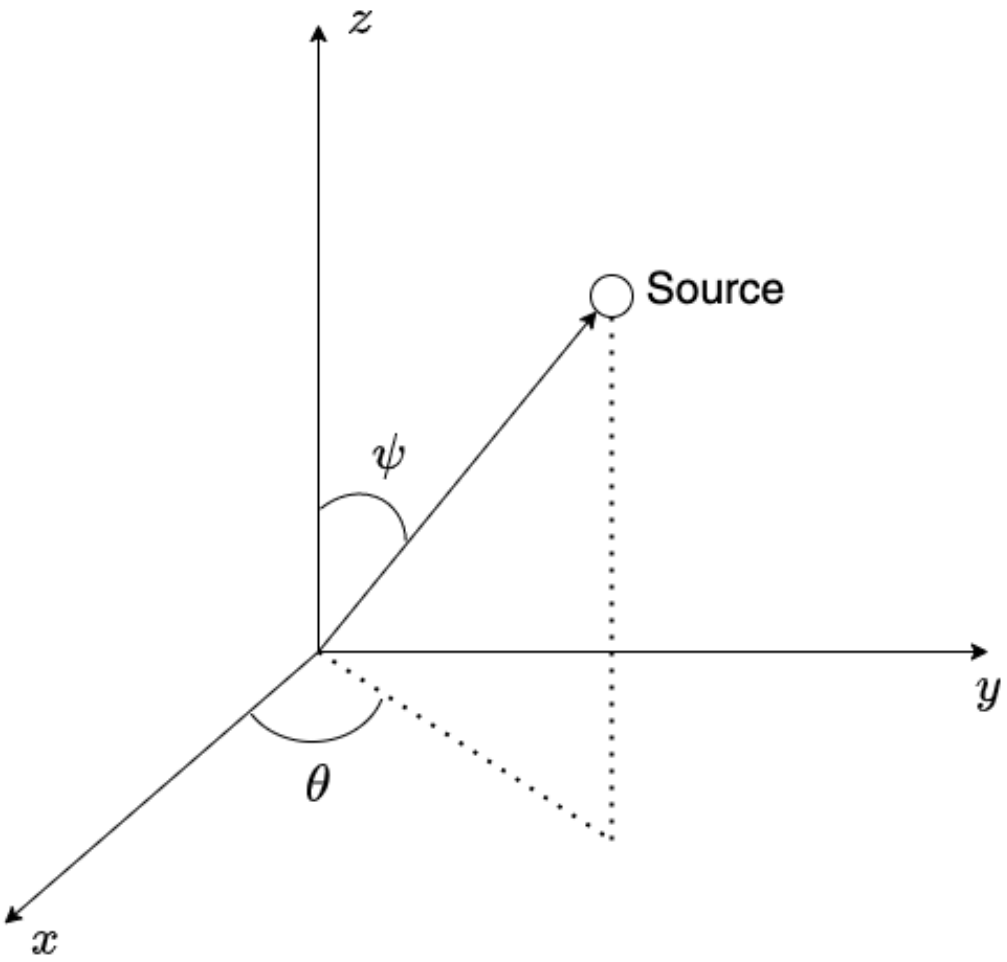


Figure 3.1: Coordinate system, sensor array and source position for 2D DoA estimation

a different delay, τ, caused by the displacement of the sensors in the array. In Figure 3.1, array model for the DoA estimation is summarized.

The given model can be narrowed down to the 1D scenario for simplicity by choosing the elevation angle, θ as 90^o degrees. The simplified form of the given model can be seen in the Figure 3.2.

As it can be seen from the Figure 3.1 and 3.2, a sensor array is composed of multiple sensors called elements and at each element signal in interest is captured with a delay.

To define the delay occurred at the each sensor in terms of sensor positions and source directions several assumptions are made [29].

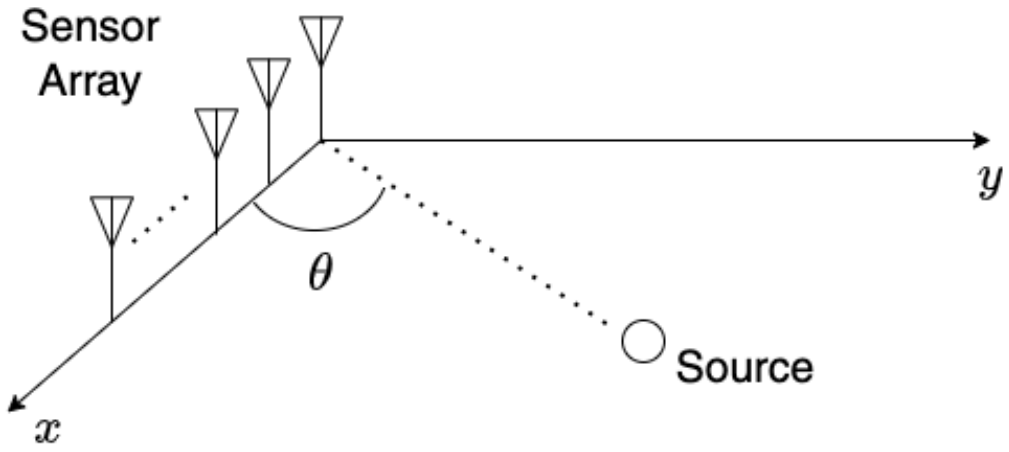


Figure 3.2: Coordinate system, sensor array and source position for 1D DoA Estimation

3.2.0.1 Far-field Source Assumption

Far-field source assumption is the assumption for the distance between the source and the sensor array. In this assumption, a source is considered as a far-field source if the distance between the sensor array and the source itself is greater that the Rayleigh distance which is defined as,

$$d_r = D^2/\lambda \tag{3.1}$$

where D denotes the array aperture and λ denotes the wave-length.

3.2.0.2 Narrowband Assumption

Narrowband assumption assumes that time-bandwidth product is small i.e.,

$$BT_{max} << 1 \tag{3.2}$$

where B denotes the bandwidth of the signal of interest and T_{max} denotes maximum time to travel across the array.

3.2.0.3 Homogeneous Propagation Medium

Signal characteristics may be altered if the transmission medium isn't homogeneous, hence this assumption is critical. As an example, sound waves in the sea differ from signal waves in a homogeneous propagation medium because of its specifics. In addition, the speed of light can be affected by the medium in which it travels. Therefore it is critical to assume that the propagation environment is homogeneous and the signal characteristics are the same at every part of the propagation medium.

With all the given assumptions, signal model for the received signal can be written as follows for the sensor array given in Figure 3.1.

Let's assume that the complex signal of interest is defined as $s(t)$ for a specific sample t and its modulated version, $s_p(t)$ is propagating in the medium.

$$s_p(t) = R(s(t)e^{2\pi f_c t}) \tag{3.3}$$

where f_c denotes the carrier frequency for the modulation and $R(.)$ denotes the real part operation.

At the m^{th} sensor, the propagating signal, $s_p(t)$ is captured with a time delay, τ_m and the received signal at the m^{th} sensor can be written as,

$$s_{p_m}(t) = s_p(t - \tau_m) = R(s(t - \tau_m)e^{2\pi f_c t}e^{-2\pi f_c \tau_m}) \tag{3.4}$$

Note that, since the narrow band assumption is valid, the complex source signal with a time delay, $s(t - \tau_m)$ can be expressed without the time delay, i.e.,

$$s(t - \tau) \approx s(t) \tag{3.5}$$

Thus the demodulated baseband signal at the m^{th} sensor, $y_m(t)$, can be written as,

$$y_m(t) = s(t)e^{-2\pi f_c \tau_m} + n_m(t) \tag{3.6}$$

where $n_m(t)$ denotes the additive Gaussian noise.

Furthermore, we can express the time delay for the m^{th} sensor, τ_m in terms of sensor positions and source direction. First lets define the unit direction vector, $\boldsymbol{g}$ for the source given in Figure 3.1.

$$\boldsymbol{g} = [cos(\theta)sin(\psi), sin(\theta)sin(\psi), cos(\psi)]^T \tag{3.7}$$

where θ denotes the azimuth angle, ψ denotes the elevation angle in Figure 3.1, and $()^T$ denotes the transpose operation.

With the given unit direction vector in (3.7), the time delay in the m^{th} sensor, τ_m, can be written as,

$$\tau_m = \frac{-\boldsymbol{P_m} \odot \boldsymbol{g}}{c} \tag{3.8}$$

where c denotes the velocity of the propagation in medium(m/s), $\odot$ denotes the elements wise multiplication operation and $\boldsymbol{P_m}$ denotes the sensor position which is defined as,

$$\boldsymbol{P_m} = [x_m, y_m, z_m]^T \tag{3.9}$$

Thus, equation (3.8) becomes as follows,

$$\tau_m = \frac{-1}{c}(x_m cos(\theta)sin(\psi) + y_m sin(\theta)sin(\psi) + z_m cos(\psi)) \tag{3.10}$$

Therefore, the time delay vector for the m^{th} sensor, τ_m, can be written as,

$$\tau_m = \frac{-\boldsymbol{P_m}^T \boldsymbol{g}}{c} \tag{3.11}$$

Let the frequency of the narrowband signal be $w = 2\pi f(rad/s)$, and f (frequency in Hz or 1/s)

$$
\begin{aligned}
w\tau_m &= \frac{-w}{c}\boldsymbol{P_m}^T\boldsymbol{g} \\
&= \frac{-2\pi}{\lambda}\boldsymbol{P_m}^T\boldsymbol{g}
\end{aligned}
\tag{3.12}
$$

where the wavelength, $\lambda = \frac{c}{f}$. Thus, the wave vector, $\boldsymbol{k}$ can be written as,

$$
\begin{aligned}
\boldsymbol{k} &= \frac{2\pi}{\lambda}\boldsymbol{g} \\
&= \frac{2\pi}{\lambda}[cos(\theta)sin(\psi), sin(\theta)sin(\psi), cos(\psi)]^T
\end{aligned}
\tag{3.13}
$$

Furthermore we can define the array steering vector which incorporates all the spatial characteristics of the array in terms of the time delays, $\boldsymbol{\tau} = [\tau_1, \tau_2, ..., \tau_M]^T$, associated with each sensor in the array.

$$
\boldsymbol{a}(\theta, \psi) = e^{-jw\boldsymbol{\tau}} = \begin{bmatrix} e^{j2\pi/\lambda(x_1 cos(\theta)sin(\psi)+y_1 sin(\theta)sin(\psi)+z_1 cos(\psi))} \\ e^{j2\pi/\lambda(x_2 cos(\theta)sin(\psi)+y_2 sin(\theta)sin(\psi)+z_2 cos(\psi))} \\ \vdots \\ e^{j2\pi/\lambda(x_M cos(\theta)sin(\psi)+y_s in(\theta)sin(\psi)+z_M cos(\psi))} \end{bmatrix} \tag{3.14}
$$

When there are multiple plane wave sources in $(\theta_1, \psi_1), (\theta_2, \psi_2), ..., (\theta_L, \psi_L)$ directions, array steering matrix $\boldsymbol{A}(\theta, \psi)$ can be constructed from the steering vectors, $\boldsymbol{a}(\theta_i, \psi_i)$ as,

$$
\boldsymbol{A}(\theta, \psi) = [\boldsymbol{a}(\theta_1, \psi_1), \boldsymbol{a}(\theta_2, \psi_2), ..., \boldsymbol{a}(\theta_L, \psi_L)]^T \tag{3.15}
$$

Then the array output , $\boldsymbol{y}(t) = [y_1(t), y_2(t), .., y_M(t)]^T$ can be written as,

$$
\boldsymbol{y}(t) = \boldsymbol{A}(\theta, \psi)\boldsymbol{S}(t) + \boldsymbol{n}(t) \tag{3.16}
$$

where $\boldsymbol{S}(t)$ is the signal matrix which composed of the source signals $s_1(t), s_2(t), ..., s_L(t)$ as

$$\boldsymbol{S}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_L(t) \end{bmatrix} \tag{3.17}$$

and $\boldsymbol{n}(t) = [n_1(t), n_2(t), ..., n_M(t)]$ denotes the noise vector composed of the individual additive noises occurred at the each sensor.

Furthermore, the signal model can be simplified to 1D model as in Figure 3.2 by assuming the elevation angle $\psi = 90$. For the simplified signal model, equation (3.16) becomes,

$$\boldsymbol{y}(t) = \boldsymbol{A}(\theta)\boldsymbol{S}(t) + \boldsymbol{n}(t) \tag{3.18}$$

where $\boldsymbol{A}(\theta) = [\boldsymbol{a}(\theta_1), \boldsymbol{a}(\theta_2), ..., \boldsymbol{a}(\theta_L)]$ and the steering vector, $\boldsymbol{a}(\theta)$ is defined as follows for the simplified signal model.

$$\boldsymbol{a}(\theta) = e^{-jw\boldsymbol{\tau}} = \begin{bmatrix} e^{j2\pi/\lambda(x_1 cos(\theta) + y_1 sin(\theta))} \\ e^{j2\pi/\lambda(x_2 cos(\theta) + y_2 sin(\theta))} \\ \vdots \\ e^{j2\pi/\lambda(x_M cos(\theta) + y_M sin(\theta))} \end{bmatrix} \tag{3.19}$$

3.2.0.4 Literature Survey

In the literature, DoA estimation is covered by many researchers. Starting from the interferometric approach, many direction estimation algorithms have been introduced in the past decades [30]. Although these methods perform significant estimations, their spatial resolution is limited by the beam-width. In addition, super-resolution algorithms based on subspace separation have been introduced. While these algorithms provide superior performance in cases where the resolution of previous algorithms was insufficient, ESPRIT in [31] is bounded by the displacement between elements and MUSIC in [32] requires a special uniform linear array geometry to overcome

the coherent source separation, so the performances of both algorithms strongly depend on the array geometry. Moreover, since these algorithms are based on subspace separation, they require the knowledge of the total number of non-coherent sources to correctly separate the noise and signal subspace. On the other hand, researchers have shown that the maximum likelihood algorithm can separate multiple coherent sources without requiring any special array geometry. However, the ML algorithm has a disadvantage in terms of computational complexity, especially in cases where the number of sources present is high. Since the complexity of the algorithm increases exponentially as the number of sources increases, it is not suitable for cases where interference and multi-path effects occur. With the introduction of compressive sampling theory by Nandes [], the sparse array representation in sensor array signal processing gained attention, and the DoA estimation problem was studied by converting the estimation problem into a norm minimization problem and diving with numerous optimizers such as LASSO and OMP[33]. Although these methods can sufficiently estimate the directions of coherent sources without requiring unique array geometry, their performance can be dramatically affected by the choice of hyper-parameters.

In recent years, the sparse representation of CS has been reformulated from the Bayesian point of view, and the sparse information is used assuming sparse prior probabilities for the signal of interest. In this sense, there have been several approaches to the DoA estimation problem using the SBL approach for various conditions such as mutual coupling, sensor position error, etc. [2], [3], [4],[34]. But most of the previous SBL based approaches consider a single array. In this thesis, the advantages of SBL method are used for a distributed sensor array structure for DoA estimation.

CHAPTER 4

DIRECTION OF ARRIVAL ESTIMATION IN DISTRIBUTED SENSOR NETWORKS

In recent years, DoA estimation for distributed sensor arrays raised significant interest within the community. While there are different possibilities for the distributed sensor array configurations, we will mainly focus on a structure which involves a master and local sensor arrays. Each sensor array can independently estimate DoA and transfer certain parameters to the master array. The function of the master array is to obtain a global solution which is superior to the local results.

4.1 Problem Definition and Data Model

In this section, we explain the details of the problem of DoA estimation in distributed sensor array networks and the data model employed in this process.

We assume that there are N sensor arrays where each of these arrays can be equivalently called as a sensor node. One of these sensor nodes is selected as the master node and the rest are considered as slave nodes. Each slave node is capable to apply signal processing tasks for DoA estimation and transmit the local estimates to the master node as shown in Figure 4.1.

Each sensor node in Figure 4.1 can have its local coordinate system but these local coordinates should be converted to a global coordinate system so that the source DoA is expressed in this global coordinate system by taking North as the reference[35].

The received signal at the n^{th} sensor node, $\boldsymbol{y_n}(t)$, is given as,

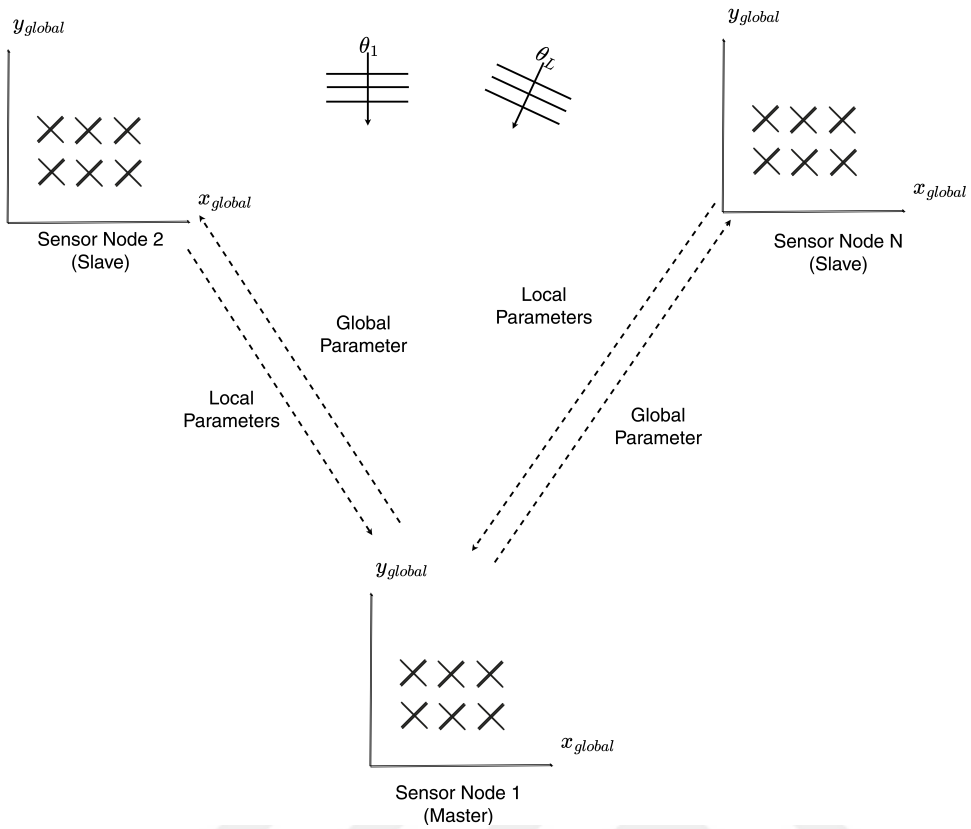


Figure 4.1: Distributed Sensor Array Network

$$\boldsymbol{y_n}(t) = \sum_{i=1}^{L} \boldsymbol{a_n}(\theta_i, \psi_i) e^{j(2\pi f_{d_n} t + \phi_n)} s_i(t) + \boldsymbol{n_n}(t) \tag{4.1}$$

where f_{d_n} and ϕ_n are the constant frequency and phase offset for each array respectively and $\boldsymbol{a_n}(\theta_i, \psi_i)$ denotes the steering vector corresponding to the direction $(\theta_i, \psi_i$ for the n^{th} sensor array.

We can define the received signal matrix for the n^{th} array as,

$$\boldsymbol{Y_n} = [\boldsymbol{y_n}(1), \boldsymbol{y_n}(2), ..., \boldsymbol{y_n}(T)] \tag{4.2}$$

which consists of the observed snapshots $\boldsymbol{y_n}(t),\ t = 1, 2, ..., T$ where T is the total number of snapshots. Also, it is assumed that source and the sensor nodes are in the same plane and only the azimuth estimation is performed for simplicity. This is not a restriction for the proposed method and it can be applied for the more general scenario.

Narrowband assumption is used and the signal matrix $\boldsymbol{S}$ is defined as,

$$\boldsymbol{S} = [\boldsymbol{s_1}, \boldsymbol{s_2}, ..\boldsymbol{s_L}]^T \tag{4.3}$$

where $\boldsymbol{s_i}$ is the vector which consist of N snapshots of i^{th} source. Noise, $\boldsymbol{n_n}(t)$ is uncorrelated zero mean white Gaussian. $\boldsymbol{a_n}(\theta_i, \psi_i)$ is the steering vector for the n^{th} sensor node and for the i^{th} source with the azimuth angle θ_i and elevation angle ψ_i. There are L sources.

It is possible to reduce the computational complexity and improve signal to noise ratio by using singular value decomposition, (SVD), as the number of observations increase. SVD of the array output matrix, $\boldsymbol{Y_n}$ can be written as,

$$\boldsymbol{Y_n} = \boldsymbol{U_{\hat{L}}} \boldsymbol{\Lambda_{\hat{L}}} \boldsymbol{V_{\hat{L}}^H} + \boldsymbol{U_{(N-\hat{L})}} \boldsymbol{\Lambda_{(N-\hat{L})}} \boldsymbol{V_{(N-\hat{L})}^H} \tag{4.4}$$

where $\hat{L} = L$ represents the number of sources. In practice $\hat{L}$ can be overestimated by looking to the eigen values of the covariance matrix of the observed signal $\boldsymbol{Y_n}$, $\hat{L} >= L$ in order to improve the convergence of CDoAE algorithm. The first term in equation (4.4) represents the signal space, and the second term is the noise space. If we multiply $\boldsymbol{Y_n}$ with $V_{\hat{L}}$, we obtain the processed array output matrix, $\bar{Y_n}$, and CDoAE uses $\bar{Y_n}$ to obtain DoA estimates, i.e.,

$$\boldsymbol{\bar{Y_n}} = \boldsymbol{Y_n} \boldsymbol{V_{\hat{L}}} = \boldsymbol{U_{\hat{L}}} \boldsymbol{\Lambda_{\hat{L}}} \tag{4.5}$$

4.2 Literature Survey

In the literature, there are several studies which cover the problem of distributed DoA estimation. In [6] a synchronization method for resolving the inter array time and frequency offsets as well as jointly estimating the DoA angles and phase offsets is presented. The main disadvantage of this method is the requirement to know either the signal or a part of the signal for accurate estimation. Also, raw data for each sensor array should be transferred to the master node. In [7] and [8], a method for estimating the complete covariance matrix of the whole array is presented. Raw data is not

transferred to the master node but the inter-array synchronization for time, frequency and phase must be done before processing.

In [9], Decentralized MUSIC algorithm is presented with several advantages. In this method, MUSIC algorithm is extended for the distributed sensor array networks by sharing only the covariance matrices of the local nodes to the master node. There is no need to perform gain/phase and frequency matching between the local arrays. Decentralized MUSIC requires Forward-Backward Spatial Smoothing [10] for coherent sources. Hence a special array structure is required for this purpose. Also the algorithm needs to know the sensor positions of each array at the master node. Search-free alternative of Decentralized MUSIC algorithm is proposed in [11].

In [15], distributed form of Alternating Direction Method of Multipliers(ADMM) is proposed. ADMM can be used to obtain global optimum solution for convex problems. In addition, it is shown that its performance for non-convex problems is good compared to alternative methods [16]. ADMM is used in DoA estimation in different works [17],[18]. In [17], the problem is separated into smaller parts to be able to process them in different processing cores. In [18], DoA cost function is divided into two functions and solved using the ADMM algorithm. But none of these studies covers the DoA estimation in Distributed Sensor Networks, instead they shown that DoA estimation problem for single sensor array can be decomposed into multiple parts where each part of the problem can be handled in a different processor.

In this thesis, a new method, Collaborative DoA Estimation (CDoAE), in distributed sensor arrays is proposed. CDoAE combines the advantages of SBL and ADMM to obtain an effective method with certain advantages. CDoAE does not require gain/phase and frequency matching for the sensor arrays. There is no need to know the sensor positions and only a set of parameters are transferred from the local arrays to the master array. In addition, CDoAE-Transferred Variable Reduction (CDoAE-TVR) method is presented to reduce the number of transferred parameters from the local arrays to the master array.

CHAPTER 5

SINGLE ARRAY DOA ESTIMATION IN SPARSE BAYESIAN LEARNING FRAMEWORK

5.1 Single Array DoA Estimation in Sparse Bayesian Learning Framework

In this section, we present the procedure for DoA estimation in a single array without any collaborative processing. In this respect, we present the details of the SBL method since some of the steps of the SBL algorithm are modified in the following section to obtain collaborative DoA estimation framework. Furthermore, single array DoA estimation is compared with the collaborative processing to demonstrate the performance improvement. Since we present the details of the SBL for single array, the subscript n is removed in the following equations for simplicity.

SBL [1] is an effective approach with several important advantages in DoA estimation. SBL method can handle coherent sources effectively independent of the array geometry. SBL uses statistical Bayesian framework to achieve this, contrary to subspace techniques [32], [31] which employ a covariance matrix for DoA estimation.

5.1.1 Sparse Signal Model

SBL uses a sparse signal representation for DoA estimation [2]. The source signal vector $\boldsymbol{S}$ is defined as a complex $K \times 1$ vector whose elements are generated by a complex K-dimensional multivariate Gaussian distribution ($K >> L$) where L denotes the total number of sources in the environment.

$$K = K_\theta K_\psi \tag{5.1}$$

where K_θ represents the number of points in the azimuth, θ, spectrum and K_ψ represents the number of points in the elevation, ψ, spectrum.

In Figure 5.1, possible source directions are represented with black solid dots and the true source directions are represented with the red circles. For each point in the given gird, a hypothetical source s_k , $k = 1, 2, 3, ..., K$ is defined. Grid selection can be done uniformly or the grid size, Δ_θ, Δ_ψ, can be determined with respect to the beamwidth of the array so that the number of grids in the regions where the beamwidth is large can be reduced and the number of grids in the regions where the beamwidth is narrow can be increased for better accuracy. In addition, if there exist a prior information about the source direction sector, then the sectors can be sampled intensely while the rest of the spectrum is sampled coarsely. Moreover, the off-grid source problem is not considered within the scope of this thesis but the proposed algorithm can be extended for the off-grid source scenarios by using the methods presented in [5] and [36].

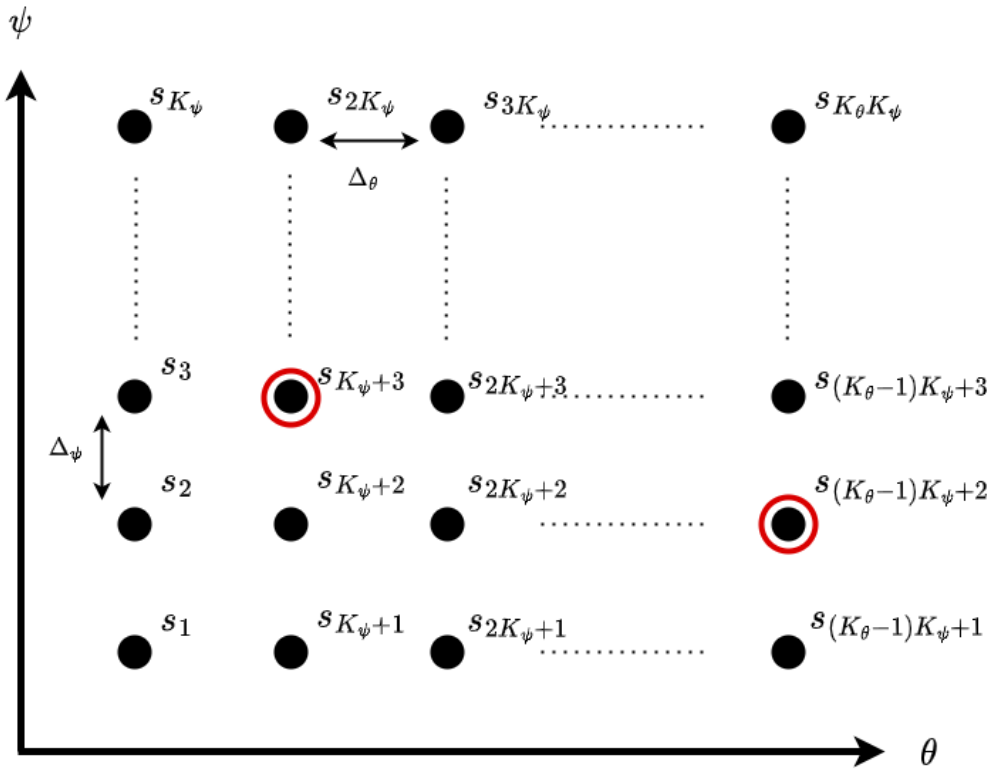


Figure 5.1: Possible source directions in azimuth, θ, and elevation, ψ and true source directions represented with red circles.

Each element of $\boldsymbol{S}$ represents the signal s_k from the direction (θ_i, ψ_j) and the variance of each row corresponds to the power of the transmitted signal. The true sources are identified after the SBL updates according to the estimated inverse source variances. In Figure 5.2, the relationship between the elements of the $\boldsymbol{s}$ vector and the hypothetical sources are shown. Each hypothetical source is represented by its variance, $\sigma^2_{s_k}$, and mean, μ_{s_k}. Let $\gamma_k = 1/\sigma^2_{s_k}$ and assume zero mean sources, $\mu_{s_k} = 0$.

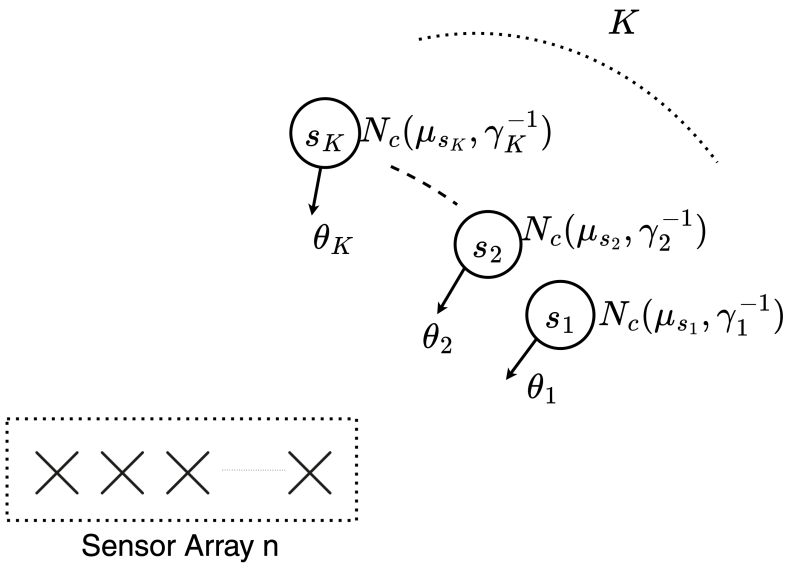


Figure 5.2: Sensor array and possible source directions with complex Gaussian distribution for Sparse Bayesian Learning framework.

Consider $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_K]^T$, and $\boldsymbol{\Delta} = diag(\boldsymbol{\gamma})$, prior distribution of $P(\boldsymbol{S}|\boldsymbol{\gamma})$ can be written as follows,

$$p(\boldsymbol{S}|\boldsymbol{\gamma}) = \prod_{k=1}^{\hat{L}} \mathcal{N}_{\mathcal{C}}(s_k|0, \boldsymbol{\Delta^{-1}}) \tag{5.2}$$

where $\mathcal{N}_{\mathcal{C}}(\mu, \sigma^2)$ denotes a complex Gaussian distribution with mean, μ, and variance, σ^2.

Moreover, the distribution of the γ_k's are modelled as Gamma distribution in order to enforce most of the rows of $\boldsymbol{S}$ to be zero, i.e.,

$$p(\boldsymbol{\gamma}) = \prod_{k=1}^{K} \Gamma(\gamma_k, c, d) \tag{5.3}$$

where c and d are the parameters for the Gamma distribution.

On the other hand, the additive noise terms in $\bar{\boldsymbol{Y}}$, are independent and modelled as complex Gaussian distribution where we define the noise precision as $\beta = \sigma^{-2}$. Therefore, $p(n_k|\beta) = \mathcal{N}_{\mathcal{C}}(n_k|0, \beta^{-1})$, and defining noise vector $\boldsymbol{n} = [n_1, n_2, ..., n_k]^T$ we can model noise as independent and identical complex Gaussian distribution,

$$p(\boldsymbol{n}|\beta) = \prod_{k=1}^{K} \mathcal{N}_{\mathcal{C}}(n_k|0, \beta^{-1}) \tag{5.4}$$

Noise precision β can be assumed to be a Gamma distribution with parameters a and b,

$$p(\beta) = \Gamma(\beta; a, b) \tag{5.5}$$

Given the noise and sparse signal models, the posterior distribution of the received signal can be expressed in terms of $\boldsymbol{s_k}$, $\boldsymbol{\gamma}$ and β as follows,

$$p(\bar{\boldsymbol{Y}}|\boldsymbol{S}, \beta) = \prod_{k=1}^{K} \mathcal{N}_{\mathcal{C}}(y_k|\boldsymbol{A}s_k, \beta\boldsymbol{I}) \tag{5.6}$$

where $\boldsymbol{A}$ is the steering matrix given as,

$$\begin{gathered} \boldsymbol{A} = [\boldsymbol{a}(\theta_1, \psi_1)), ..., \boldsymbol{a}(\theta_i, \psi_j)), ..., \boldsymbol{a}(\theta_{K_\theta}, \psi_{K_\psi}))] \\ i = 1, 2, ..., K_\theta \; j = 1, 2, ..., K_\psi, \end{gathered} \tag{5.7}$$

In the SBL approach, the main objective is to find the variables β and $\boldsymbol{\gamma}$ to estimate the source directions. To achieve this goal, the posterior probability of these parameters $p(\beta, \boldsymbol{\gamma}, \boldsymbol{S}|\bar{\boldsymbol{Y}})$ must be calculated. Since the computation of $p(\beta, \boldsymbol{\gamma}, \boldsymbol{S}|\bar{\boldsymbol{Y}})$ cannot

be done explicitly, the problem can be transformed into the maximization of the following equation using Bayes' theorem,

$$p(\boldsymbol{S}, \boldsymbol{\gamma}, \beta | \bar{\boldsymbol{Y}}) = p(\bar{\boldsymbol{Y}}, \boldsymbol{\gamma}, \beta, \boldsymbol{S}) / p(\bar{\boldsymbol{Y}}) \tag{5.8}$$

Since the term of $p(\bar{\boldsymbol{Y}})$ is just a normalization factor, it can be ignored and the problem can be considered as,

$$\max_{\boldsymbol{\gamma}, \beta} p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \boldsymbol{\gamma}, \beta) \tag{5.9}$$

Since the problem in equation (5.9) has both observable and hidden variables, variable update and estimate procedure can be implemented by using a two stage Expectation Maximization (EM) Algorithm [37] which is outlined in the following part.

5.1.2 Parameter Update through EM Algorithm

EM method is based on the principle of continuously constructing a lower bound for $\langle ln((p(\bar{\boldsymbol{Y}}|\beta, \boldsymbol{\gamma}))) \rangle$ and then optimizing this lower bound.

$$\langle ln((p(\bar{\boldsymbol{Y}}|\beta, \boldsymbol{\gamma}))) \rangle = E_{p(\boldsymbol{S}|\boldsymbol{\gamma}, \beta, \bar{\boldsymbol{Y}})} \{ ln((p(\bar{\boldsymbol{Y}}|\beta, \boldsymbol{\gamma}))) \} \tag{5.10}$$

In the following part, two main steps of the EM algorithm, namely, Hidden Variable Update and Hyperparameter Update are described respectively. In the first step, $\boldsymbol{S}$ is considered as a hidden variable and the lower bound is constructed. After the lower bound is determined, the hyperparameters, β and $\boldsymbol{\gamma}$, are obtained by maximizing the lower bound. This procedure continues until convergence.

5.1.2.1 Hidden Variable Update

In the hidden variable update stage, lower bound for expected value of the $ln(p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \beta, \boldsymbol{\gamma}))$, namely, $\langle ln(p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \beta, \boldsymbol{\gamma})) \rangle$, is constructed to be maximized in the Hyperparameter Update stage. To construct the lower bound, expected value of $ln(p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \beta, \boldsymbol{\gamma}))$ must be calculated over the distribution $p(\boldsymbol{S}|\boldsymbol{\gamma}, \beta, \bar{\boldsymbol{Y}})$ in the following parts. In order

to do this, posterior distribution $p(\boldsymbol{S}|\bar{\boldsymbol{Y}}, \beta, \boldsymbol{\gamma})$ should be written through the Bayesian expression given below,

$$p(\boldsymbol{S}|\boldsymbol{\gamma}, \beta, \bar{\boldsymbol{Y}}) = \frac{p(\bar{\boldsymbol{Y}}|\boldsymbol{S}, \beta)p(\boldsymbol{S}|\boldsymbol{\gamma})}{p(\bar{\boldsymbol{Y}}|\beta, \boldsymbol{\gamma})} \tag{5.11}$$

This expression for the posterior distribution can also be treated as a Gaussian distribution with mean, $\boldsymbol{\mu} = [\mu_1, \mu_2, ..., \mu_K]^T$, and covariance matrix, $\boldsymbol{\Sigma}$, respectively. The $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the conditional distribution can be estimated iteratively using an initial estimate through the following expressions [3],

$$\boldsymbol{\mu_k^{j+1}} = \beta^j \boldsymbol{\Sigma^j A^H y_k}, k = 1, 2, ..., \hat{L} \tag{5.12}$$

$$\boldsymbol{\Sigma^{j+1}} = (\beta^j \boldsymbol{A^H A} + \boldsymbol{\Delta^j})^{\boldsymbol{-1}} \tag{5.13}$$

At each iteration, j, lower bound for $\langle ln(p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \beta, \boldsymbol{\gamma}))\rangle^{j+1}$ is constructed with the estimated mean and variance. Note that $p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \boldsymbol{\gamma}, \beta)$ can be written as,

$$p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \boldsymbol{\gamma}, \beta) = p(\bar{\boldsymbol{Y}}|\boldsymbol{S}, \beta)p(\boldsymbol{S}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})p(\beta) \tag{5.14}$$

Using the Bayesian relations the lower bound for the expected value $\langle ln((p(\bar{\boldsymbol{Y}}|\beta, \boldsymbol{\gamma})))\rangle^{j+1}$ becomes,

$$\langle ln(p(\bar{\boldsymbol{Y}}, \boldsymbol{S}, \beta, \boldsymbol{\gamma}))\rangle = \langle ln(p(\bar{\boldsymbol{Y}}|\boldsymbol{S}, \beta)p(\boldsymbol{S}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})p(\beta))\rangle \tag{5.15}$$

The hyperparameters are estimated by maximizing the lower bound in equation (5.15). After updating the hyperparameters, the mean and variance can be recalculated in the next iteration.

5.1.2.2 Hyperparameter Update

In the first step of the hyper parameter estimation, noise precision β is updated first. This is achieved by ignoring the irrelevant terms and the following expression is obtained [3],

$$\langle ln(p(\bar{\boldsymbol{Y}}|\boldsymbol{S},\beta)p(\beta))\rangle = \langle ln(p(\bar{\boldsymbol{Y}}|\boldsymbol{S},\beta))\rangle + ln(p(\beta)) \tag{5.16}$$

Equation (5.5) can be more explicitly written as,

$$ln(p(\beta)) = Const + (a-1)ln(\beta) - \beta/b \tag{5.17}$$

Then $\langle ln(p(\bar{\boldsymbol{Y}}|\boldsymbol{S},\beta))\rangle$ becomes,

$$\langle ln(p(Y|S,\beta))\rangle = \hat{L}Mln(\beta) - \beta\sum_{k=1}^{\hat{L}} ||\boldsymbol{y_k} - \boldsymbol{A\mu_k}||_2^2 - \beta Ktr\{\boldsymbol{A\Sigma A^H}\} \tag{5.18}$$

Finally, by taking the derivative of equation (5.16) with respect to β, we obtain the following update equation [3],

$$\beta^{j+1} = \frac{\hat{L}M + (a-1)}{b + \sum_{k=1}^{\hat{L}} ||\boldsymbol{y_k} - \boldsymbol{A\mu_k}^j||_2^2 - \hat{L}tr\{\boldsymbol{A\Sigma A^H}\}}, \tag{5.19}$$

In the next step, a similar procedure is followed to obtain the $\boldsymbol{\gamma}$ update expression. Keeping only the $\boldsymbol{\gamma}$ related terms and ignoring the rest in equation (5.15) can be expressed as,

$$\langle ln(p(\boldsymbol{S}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}))\rangle = \langle ln(p(\boldsymbol{S}|\boldsymbol{\gamma})\rangle + ln(p(\boldsymbol{\gamma})) \tag{5.20}$$

Since $\boldsymbol{\gamma}$ has a Gamma distribution, logarithm of its probability density function can be written similar to the β update step with respect to the parameters c and d,

$$ln(p(\boldsymbol{\gamma}|c,d)) = Const + (c-1)ln(\boldsymbol{\gamma}) - \boldsymbol{\gamma}/d \tag{5.21}$$

Then equation (5.20) can be written as,

$$\langle ln(p(\boldsymbol{S}|\boldsymbol{\gamma}))\rangle = \hat{L} ln(|\boldsymbol{\Delta}|) - \sum_{k=1}^{\hat{L}} \langle s_k^H \boldsymbol{\Delta} s_k \rangle \tag{5.22}$$

If we take the derivative of (5.20) with respect to γ_k the following update equation is obtained,

$$\gamma_k^{j+1} = \frac{\hat{L} + c + 1}{d + \sum_{l=1}^{\hat{L}} [\boldsymbol{\Xi_l^{j+1}}]_{kk}} \tag{5.23}$$

where $\boldsymbol{\Xi_l^{j+1}} = \boldsymbol{\mu_l^{j+1}} \boldsymbol{\mu_l^{(j+1)H}} + \boldsymbol{\Sigma^{j+1}}$ and $[.]_{kk}$ represents the (k,k) element of matrix. Note that Gamma distribution parameters, c and d, are positive real terms as well as γ_k^{j+1}. Hence $\boldsymbol{\gamma_k^{j+1}}$ is a real vector.

After obtaining $\boldsymbol{\mu^{j+1}}$,$\boldsymbol{\Sigma^{j+1}}$,$\beta^{j+1}$ and $\boldsymbol{\gamma_l^{j+1}}$, a single iteration of the SBL algorithm is completed. Algorithm continues to update hidden variables, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, with the recently estimated hyper-parameters, β, $\boldsymbol{\gamma}$ to construct the lower bound and optimize this lower bound to obtain the updated hyper-parameters sequentially until the convergence.

After the algorithm converges, spatial spectrum, $P(\theta, \psi)$, is obtained according to the following equation,

$$P(\theta_i, \psi_j) = \frac{1}{\gamma_k},\ k = (i-1)K_\psi + j$$
$$i = 1, 2, ..., K_\theta\ j = 1, 2, ..., K_\psi \tag{5.24}$$

Note that γ_k corresponds to the inverse of the variance for the source coming from the direction (θ_i, ψ_j). After obtaining the spectrum, $P(\theta, \psi)$, DoA estimation can be done by searching for the peaks in the spectrum. Moreover, the spectrum can be normalized to obtain values between $[0, 1]$. In Figure 5.3, an example spectrum obtained by SBL method is given.

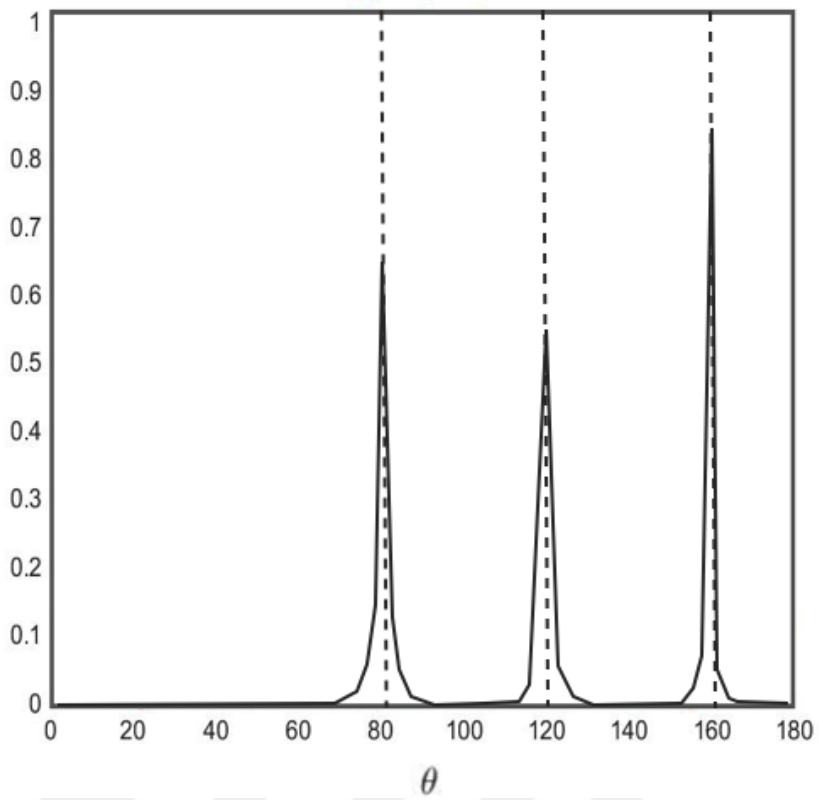


Figure 5.3: Normalized variance spectrum obtained by SBL method and true DoAs represented with dashed lines

The steps of the SBL method [3] for DoA estimation on a single array are given below.

Algorithm 1 Single Array DoA Estimation

Step 1: Initialize parameters $\boldsymbol{\gamma^0}, \boldsymbol{\beta^0}, \boldsymbol{\mu^0}, \boldsymbol{\Sigma^0}, a, b, c, d$

Step 2: Update hidden variable parameters $\boldsymbol{\mu^{j+1}}$ and $\boldsymbol{\Sigma^{j+1}}$ according to equation (5.12) and equation (5.13) respectively.

Step 3: Update hyper-parameter $\boldsymbol{\beta^{j+1}}$ according to equation (5.19).

Step 4: Update hyper-parameter $\boldsymbol{\gamma}^{j+1}$ according to equation (5.23)

Step 5: Check if the algorithm converged.

Step 6: If not converged go back to Step 2 otherwise report DoA angles found by using $\boldsymbol{\gamma}$ equation (5.24)

CHAPTER 6

COLLABORATIVE DOA ESTIMATION (CDOAE)

6.1 Collaborative DoA Estimation (CDoAE)

In this section, we present the collaborative DoA estimation where distributed ADMM approach is used to modify the variable update procedure of the master node and local nodes. This modification consists of the addition of certain constraints imposed by the updated global variable vector transmitted by the master node to the local nodes. In the local nodes, the updated global variable is used in a constrained equation where the original SBL expression is modified so that the solution is forced in a direction pointed by the global variable. In the master node, global variable is obtained by using all the local parameters.

With the combination of the advantages of SBL and distributed ADMM, proposed Collaborative Direction of Arrival Estimation (CDoAE) algorithm can overcome the limitations for the Distribıted Sensor Array Networks. While SBL brings the ability of estimating the coherent source directions without requiring a special array geometry, distributed ADMM brings the ability of sharing limited number of parameters instead of sharing the raw data between the nodes. Moreover, with the distributed ADMM, each node in the network does not require the sensor positions of the other nodes and there is no need for a synchronization step between the nodes in terms of phase and frequency.

Since the proposed method uses the distributed ADMM framework in the defined procedure, we present the details of the distributed ADMM algorithm at first.

6.1.1 Distributed ADMM Algorithm

Distributed ADMM (DADMM) is introduced as a technique which enables the global solution for a distributed sensor array [15]. Distributed ADMM has all the main features of the ADMM algorithm where global optimum is obtained when the problem is convex. It is known that ADMM generates a well behaved solution for non convex problems in general [16]. In the following part, main features of the distributed ADMM are presented which are valid for both local and master nodes.

In Distributed ADMM main goal is to minimize a composite function, $f(\boldsymbol{\gamma})$, which can be defined as the sum of N different convex functions, $f_n(\boldsymbol{\gamma})$, $n = 1, 2, 3, ..., N$, i.e.,

$$\min_{\boldsymbol{\gamma}} f(\boldsymbol{\gamma}) = \sum_{n=1}^{N} f_n(\boldsymbol{\gamma}) \tag{6.1}$$

In the Distributed ADMM approach, it is aimed to obtain a collaborative solution while the each function $f_n(\boldsymbol{\gamma})$ is optimized at its own processor separately. Therefire the problem in 6.1 can be written in terms of both local variables, $\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, ..., \boldsymbol{\gamma_N}$, and global variable, $\boldsymbol{z}$, as,

$$\min_{\boldsymbol{\gamma_n}} \sum_{n=1}^{N} f_n(\boldsymbol{\gamma_n}) \; s.t \; \boldsymbol{\gamma_n} = \boldsymbol{z} \tag{6.2}$$

The global variable $\boldsymbol{z}$ in the given equation is inserted for imposing the following constraint,

$$\boldsymbol{\gamma_1} = \boldsymbol{\gamma_2} = \boldsymbol{\gamma_3} = ... = \boldsymbol{\gamma_N} \tag{6.3}$$

For the defined problem in 6.2, augmented Lagrangian function, $L_p(\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, ..., \boldsymbol{\gamma_N}, u)$ can be written as,

$$L_p(\boldsymbol{\gamma_1}, \boldsymbol{\gamma_2}, ..., \boldsymbol{\gamma_N}, \boldsymbol{z}, \boldsymbol{u}) = \sum_{n=1}^{N} f_n(\boldsymbol{\gamma_n} + u_n^T(\boldsymbol{\gamma_n} - z) + (\rho/2)||\boldsymbol{\gamma_n} - z||_2) \tag{6.4}$$

Given the augmented Lagrangian function in (6.4), the distributed ADMM update equations can be obtained as,

$$\boldsymbol{\gamma_n}^{j+1} = arg\min_{\boldsymbol{\gamma_n}} L_p(\boldsymbol{\gamma_1}^j, \boldsymbol{\gamma_2}^j, .., \boldsymbol{\gamma_n}, ..., \boldsymbol{\gamma_N}^j, \boldsymbol{z}^j, \boldsymbol{u}^j) \tag{6.5}$$

$$\boldsymbol{z}^{j+1} = arg\min_{\boldsymbol{z}} L_p(\boldsymbol{\gamma_1}^{j+1}, \boldsymbol{\gamma_2}^{j+1}, .., \boldsymbol{\gamma_n}^{j+1}, ..., \boldsymbol{\gamma_N}^{j+1}, \boldsymbol{z}, \boldsymbol{u}^j) \tag{6.6}$$

$$\boldsymbol{u_n^{j+1}} = \boldsymbol{u_n^j} + \rho(\boldsymbol{\gamma_n^{j+1}} - \boldsymbol{z^{j+1}}) \tag{6.7}$$

According to equation 6.5,6.6,6.7, following update steps are obtained.

$$\boldsymbol{\gamma_n^{j+1}} = arg\min_{\boldsymbol{\gamma_n}}(f_i(\boldsymbol{\gamma_n}) + \boldsymbol{u^{jT}}(\boldsymbol{\gamma_n} - \boldsymbol{z^j}) + \rho/2||\boldsymbol{\gamma_n} - \boldsymbol{z^j}||_2^2) \tag{6.8}$$

$$\boldsymbol{z^{j+1}} = 1/N \sum_{n=1}^{N} (\boldsymbol{\gamma_n^{j+1}} + 1/\rho \boldsymbol{u_n^j}) \tag{6.9}$$

$$\boldsymbol{u_n^{j+1}} = \boldsymbol{u_n^j} + \rho(\boldsymbol{\gamma_n^{j+1}} - \boldsymbol{z^{j+1}}) \tag{6.10}$$

where ρ is scalar used as a weight value for the update term.

As it can be seen in the equations above, defined update procedure requires the exchange of the local variables, $\boldsymbol{\gamma_n}$, $\boldsymbol{u_n}$, and the global variable, $\boldsymbol{z}$, between the local nodes and the master node. In the step of updating the local variables, global variable obtained in the previous iteration must be known in each node. Also when the global variable is updated, the local variables estimated in each node must be gathered in the master node before the update. In Figure 6.1, this process is shown.

In collaborative DoA estimation, we assume that there are N sensor arrays where one of them is the master node as shown in Figure 6.1. Both master and the local nodes proceed their own iterative processes. The local nodes transmit their parameters to the master node so that it generates a global parameter, $\boldsymbol{z}$, which is then transmitted

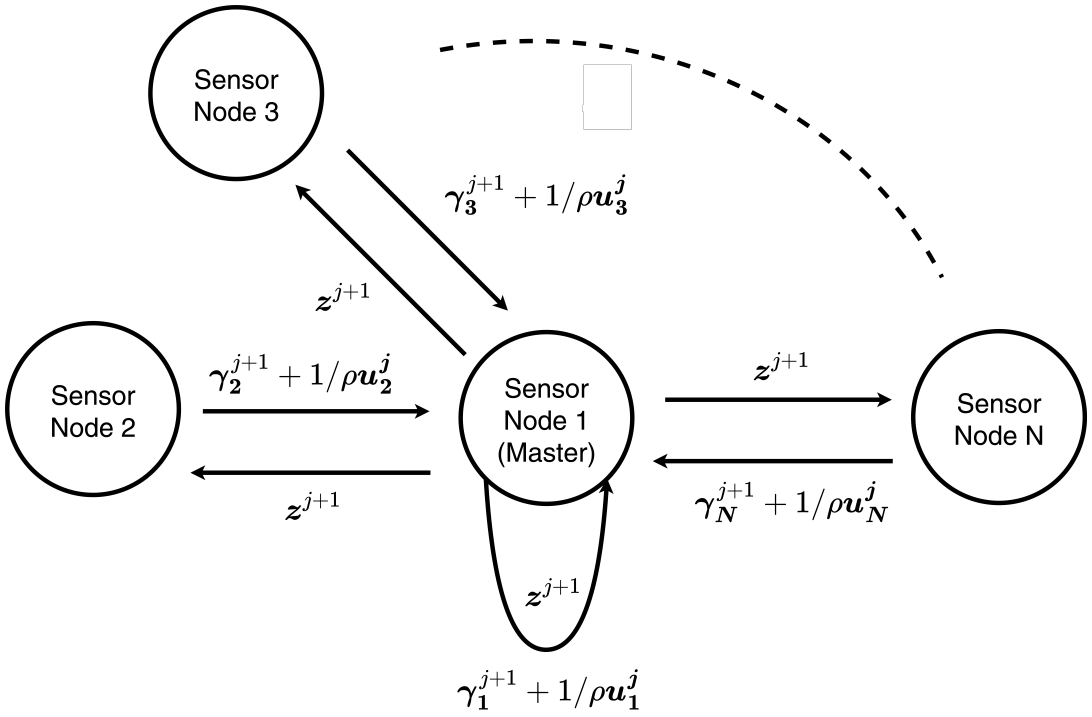


Figure 6.1: Sensor nodes and variable exchange diagram for Distributed Alternating Direction Method of Multipliers.

back to the local nodes. Both the master and local nodes generate their estimates using DADMM framework. To convert the SBL DoA estimation problem into a collaborative optimization, it assumed that the local coordinate system at each node is converted to a global coordinate system so that each sensor array DoA is expressed in the global coordinate system. In DADMM framework, the composite objective function in (6.2) is selected in terms of the negative lower bound expressions in (5.20) at each node n, i.e.,

$$f_n(\boldsymbol{\gamma_n}) = -\langle ln(p(\boldsymbol{S_n}|\boldsymbol{\gamma_n}))\rangle - ln(p(\boldsymbol{\gamma_n})) \tag{6.11}$$

Using (5.21) and (5.22) above expression can be simplified as [3],

$$f_n(\boldsymbol{\gamma_n}) = -(\hat{L} + c + 1)\sum_{k=1}^{K} ln([\boldsymbol{\gamma_n}]_k) + d\sum_{k=1}^{K}[\boldsymbol{\gamma_n}]_k + \sum_{l=1}^{\hat{L}} tr(\Xi_{n_l}\Delta) \tag{6.12}$$

We can write the collaborative optimization problem as follows,

$$\min_{\gamma_n} \sum_{n=1}^{N} -(\hat{L}+c+1)\sum_{k=1}^{K} ln([\boldsymbol{\gamma_n}]_k) + d\sum_{k=1}^{K}[\boldsymbol{\gamma_n}]_k + \sum_{l=1}^{\hat{L}} tr(\Xi_{n_l}\Delta)\ st.\boldsymbol{\gamma_n} = \boldsymbol{z},\ n = 1, 2, .., N \tag{6.13}$$

Thus, the γ update in the previous SBL algorithm can be performed in a collaborative manner by modifying the update steps according to the DADMM approach. In the following section, the proposed method for collaborative variable update process is presented.

6.1.2 Collaborative Variable Update

In this section, we present the collaborative variable update procedure which is essential for the proposed method. In this procedure, SBL variable update equations are modified with the distributed ADMM and the collaborative DoA estimation is achieved.

In the first step of the proposed method, mean and variance, $\boldsymbol{\mu_n}$ and $\boldsymbol{\Sigma_n}$, of the hidden variable $\boldsymbol{S_n}$ are estimated for each array in the network. After the hidden variable estimation step, lower bound is constructed as in equation (5.16) for each node and noise precision β_n is obtained according to equation (5.19). In the third step, the local hyperparameter $\boldsymbol{\gamma_n}$ is estimated over the modified update equation which consists of both EM and ADMM terms. The obtained local solutions are weighted according to a weighting function and passed to the master node. In the master node, the update of the global solution is performed and shared with the local arrays. The remaining ADMM updates are performed separately in each node and the algorithm iterates until convergence by repeating the above steps. The details of the update procedure are given below.

6.1.2.1 Hidden Variable Update

The hidden variable estimation in Section 5.1.2.1 is used for each node. Therefore, in each iteration, the parameters of the hidden variables $\boldsymbol{\mu_n^{j+1}}$ and $\boldsymbol{\Sigma_n^{j+1}}$ are estimated for each node according to equation (6.14) and (6.15) [3], i.e.,

$$\boldsymbol{\mu_{k_n}^{j+1}} = \beta_n^j \boldsymbol{\Sigma_n^j A_n^H y_{k_n}}, k = 1, 2, ..., \hat{L} \tag{6.14}$$

$$\boldsymbol{\Sigma_n^{j+1}} = (\beta_n^j \boldsymbol{A_n^H A_n} + \boldsymbol{\Delta_n^j})^{\boldsymbol{-1}} \tag{6.15}$$

where the subscript n is used for node index, $\boldsymbol{A_n}$ is the array steering matrix, $\boldsymbol{y_{k_n}}$ represents the signal captured by the n^{th} array. Note that equation (6.14) and (6.15) are the same as the SBL updates in (5.12),(5.13) except the index n which indicates that these equations are employed at each sensor node.

6.1.2.2 Hyper Parameter Update

In this part, the update of the hyper-parameters β_n^{j+1} and $\boldsymbol{\gamma_n^{j+1}}$ in the j^{th} iteration is presented.

At each node similar β^{j+1} update equation in (5.19) is applied to obtain the local estimates [3], i.e.,

$$\beta_n^{j+1} = \frac{\hat{L}M + (a-1)}{b + \sum_{k=1}^{\hat{L}} ||\boldsymbol{y_{k_n}} - \boldsymbol{A_n \mu_{k_n}^{j+1}}||_{\boldsymbol{2}}^{\boldsymbol{2}} - \hat{L} tr\{\boldsymbol{A_n \Sigma_n^{j+1} A_n^H}\}}, \tag{6.16}$$

On the other hand, $\boldsymbol{\gamma_n}^{j+1}$ update procedure is changed due to the terms added from the distributed ADMM algorithm. In (6.17), an additional term,$c_n(\boldsymbol{\gamma_n}^j)$, comes from the ADMM update procedure [3], i.e.,

$$\boldsymbol{\gamma_n^{j+1}} = \underset{\boldsymbol{\gamma_n^{j+1}}}{\operatorname{argmin}} f_n(\boldsymbol{\gamma_n}^{j+1}) + c_n(\boldsymbol{\gamma_n}^{j+1}) \tag{6.17}$$

where $c_n(\boldsymbol{\gamma_n}^{j+1}) = \boldsymbol{u_n^{j^T}}(\boldsymbol{\gamma_n}^{j+1} - \boldsymbol{z^j}) + (\rho/2)||\boldsymbol{\gamma_n}^{j+1} - \boldsymbol{z^j}||_2^2)$ and $f_n(\boldsymbol{\gamma_n}^{j+1})$ is the

function in (6.12). Therefore, ${\boldsymbol{\gamma_n}}^{j+1}$ update equation can be obtained by taking the derivative of $c_n({\boldsymbol{\gamma_n}}^{j+1}) + f_n({\boldsymbol{\gamma_n}}^{j+1})$ with respect to $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ and equating to zero. Note that $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ represents the k^{th} element of the ${\boldsymbol{\gamma_n}}^{j+1}$ vector. Hence,

$$\frac{\partial f({\boldsymbol{\gamma_n}}^{j+1}))}{\partial [{\boldsymbol{\gamma_n}}^{j+1}]_k} + \frac{\partial c({\boldsymbol{\gamma_n}}^{j+1}))}{\partial [{\boldsymbol{\gamma_n}}^{j+1}]_k} = 0 \tag{6.18}$$

The derivative of $c_n({\boldsymbol{\gamma_n}}^{j+1})$ with respect to $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ is given as,

$$\frac{\partial c_n({\boldsymbol{\gamma_n}}^{j+1})}{\partial [{\boldsymbol{\gamma_n}}^{j+1}]_k} = [\boldsymbol{u_n^j}]_k + \rho([{\boldsymbol{\gamma_n}}^{j+1}]_k - [\boldsymbol{z^j}]_k) \tag{6.19}$$

Similarly, derivative of $f_n({\boldsymbol{\gamma_n}}^{j+1})$ with respect to $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ is written as,

$$\frac{\partial f_n({\boldsymbol{\gamma_n}}^{j+1})}{\partial [{\boldsymbol{\gamma_n}}^{j+1}]_k} = d + \sum_{l=1}^{\hat{L}} [\Xi_{n_l}]_{kk} - \frac{\hat{L} + c - 1}{[{\boldsymbol{\gamma_n}}^{j+1}]_k} \tag{6.20}$$

where $\boldsymbol{\Xi_{n_l}}$ is the $\boldsymbol{\Xi_l}$ matrix in equation (5.23) obtained at the n^{th} sensor node.
We obtain a second order polynomial for $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ from (6.18) and it is given as,

$$[{\boldsymbol{\gamma_n}}^{j+1}]_k^2 + \alpha_1 [{\boldsymbol{\gamma_n}}^{j+1}]_k + \alpha_2 = 0 \tag{6.21}$$

where the real valued parameters α_1 and α_2, are given as,

$$\alpha_1 = \frac{[\boldsymbol{u^j}]_k + d + \sum_{l=1}^{\hat{L}} [\Xi]_{kk}}{\rho} - [\boldsymbol{z^j}]_k \tag{6.22}$$

$$\alpha_2 = \frac{-K - c + 1}{\rho} \tag{6.23}$$

Since (6.23) is negative, (6.21) has always two real roots, one which is positive and the other is negative. Since $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ is defined as the inverse of the source variance, $\sigma_{s_k}^2$, the desired $[{\boldsymbol{\gamma_n}}^{j+1}]_k$ is positive for a given k^{th} source. Therefore, the root which is positive is selected for each $[{\boldsymbol{\gamma_n}}^{j+1}]_k \ , k = 1, 2, ..., K$. Then, ${\boldsymbol{\gamma_n}}^{j+1}$ vector is constructed as,

$$\boldsymbol{\gamma_n}^{j+1} = [[\boldsymbol{\gamma_n}^{j+1}]_1, [\boldsymbol{\gamma_n}^{j+1}]_2, ..., [\boldsymbol{\gamma_n}^{j+1}]_K]^T \tag{6.24}$$

6.1.2.3 Weight Function for the Local Estimates

In the proposed method, local estimates are weighted by a weighting function before they are sent to the master node. The main idea behind this weighting function is to trust the local estimates obtained at some nodes more than the estimates obtained in the other nodes for the direction (θ_i, ψ_j) because of the non-omnidirectional performance of the nodes.

In Figure 6.2, two uniform linear arrays with different headings are given. As it can be seen from the figure, sensor node 1 has broadside region around 90 degrees and has endfire region around 0 degrees in azimuth. On the other hand, sensor node 2 has broadside region where sensor node 1 has its endfire and sensor node 2 has endfire region where the sensor node 1 has its broadside region.

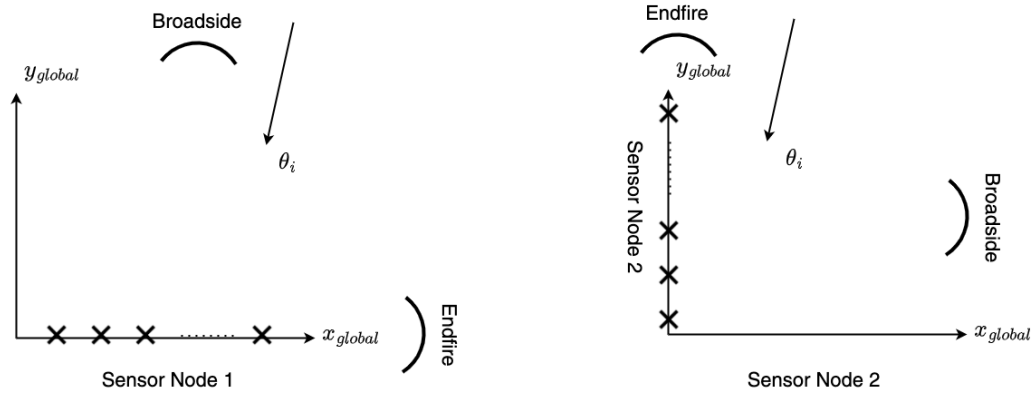


Figure 6.2: Sensor Node 1,2 and their braodside and endfire regions.

Consider a source signal which comes from the azimuth angle, θ_i, as in Figure 6.2. In this scenario, the source signal is closer to the broadside region of sensor node 1 and closer to the endfire region of sensor node 2. Therefore, we would like to trust the estimates obtained at the sensor node 1 more than the sensor node 2 due to the fact that the DoA estimation performances increases as the source signal getting closer to the broadside region of the array and decreases when the source signal becomes closer to the endfire region. With this in consideration we would like to apply a weighting

function to emphasize our trust for each array at each direction in the spectrum in order to improve the collaborative DoA estimation performance.

For simplicity, we assume the same weighting function for all elevation angles, ψ, and hence ψ is selected as $\pi/2$ for weighting function generation. Therefore, for each possible angle, θ, in the spectrum, the weighting function $w_n(\theta)$ is used at each array. We selected the weighting function using the Approximate Cramer Rao Bound (ACRB) [38]. ACRB expresses a approximate lower bound on the variance of the DoA estimation error and for an array with $\lambda/2$ inter-element spacing it is given as,

$$ACRB(\theta) = \frac{M}{2NSNR||\dot{\boldsymbol{a}}(\boldsymbol{\theta})|_2^2|} \tag{6.25}$$

In this study, ignoring the common terms, weight function, $w_n(\theta)$ is selected inversely proportional to the ACRB, i.e.,

$$w_n(\theta) = ||\dot{\boldsymbol{a_n}}(\boldsymbol{\theta})||_2^2 \tag{6.26}$$

where $\dot{\boldsymbol{a_n}}(\boldsymbol{\theta})$ is the derivative of the steering vector with respect to θ at the n^{th} node.

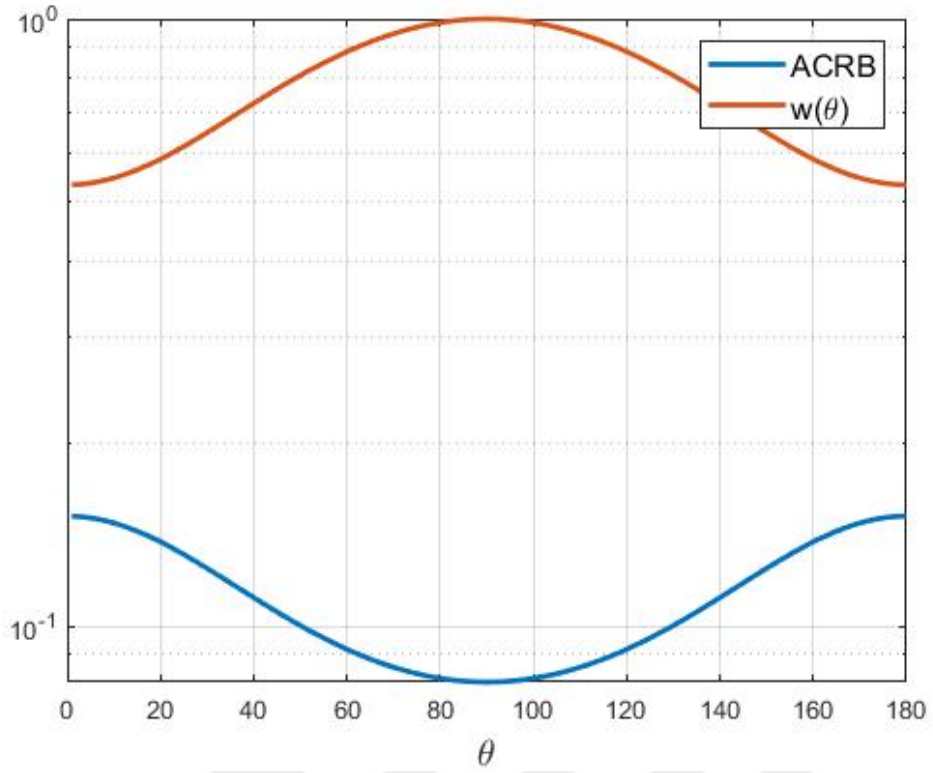


Figure 6.3: Approximate Cramer Rao Bound and the Weighting Function for URA.

In the Figure 6.3, ACRB and the weighting function corresponding to the Uniform Rectangular Array(URA) with $M = 12(3 \times 4)$ elements is given. As it can be seen in Figure 6.3, the performance of URA decreases when the DoA of the incoming signal is closer to the end-fire of the array. Therefore, the DoA estimate obtained for an angle closer to the end-fire can be expected to be less reliable than the angle closer to the broadside.

After we have the weighting functions for each node, local $\boldsymbol{\gamma_n}$ and $\boldsymbol{u_n}$ estimates are shared with the master node after the multiplication with the weighting function as follows,

$$\boldsymbol{\gamma_n'}^{j+1} = \boldsymbol{w_n} \odot \boldsymbol{\gamma_n^{j+1}}, \tag{6.27}$$

$$\boldsymbol{u_n'}^{j+1} = \boldsymbol{w_n} \odot \boldsymbol{u_n^j} \tag{6.28}$$

where $\odot$ is the element-wise multiplication.

6.1.2.4 Global Variable Update

After obtaining the local variables $\boldsymbol{\gamma_n}$, $\boldsymbol{u_n}$ and multiplying them with the weight function, the local variables are shared with the master array. In the master array, the global variable is updated and shared with the local arrays. The update equation for the global variable can be obtained from (6.9) as,

$$\boldsymbol{z^{j+1}} = 1/N \sum_{n=1}^{N} (\boldsymbol{\gamma_n'^{j+1}} + 1/\rho \boldsymbol{u_n'^{j}}) \tag{6.29}$$

Above equation is computed in the master node.

6.1.2.5 Dual Variable Update

Dual variable update is performed at each local node separately. After receiving the global variable $\boldsymbol{z^{j+1}}$, dual variable is updated using (6.10) as,

$$\boldsymbol{u_n^{j+1}} = \boldsymbol{u_n^{j}} + \rho(\boldsymbol{\gamma_n^{j+1}} - \boldsymbol{z^{j+1}}) \tag{6.30}$$

Note that, $\boldsymbol{u_n^j}$ and $\boldsymbol{\gamma_n^{j+1}}$ are not weighted variables different from (6.29). Once the dual variable is updated, algorithm checks if the convergence criteria is satisfied.

6.1.2.6 Stopping Criteria

The exiting criterion for the iterations of the proposed algorithm is determined based on the changes in the global variable $\boldsymbol{z}$ between iterations. The algorithm terminates when the norm of the change in the global variable is less than a given threshold ϵ,

$$||\boldsymbol{z^{j+1}} - \boldsymbol{z^{j}}||_2^2 / ||\boldsymbol{z^{j}}||_2^2 < \epsilon \tag{6.31}$$

The steps of the proposed collaborative DoA estimation approach, CDoAE, are presented below,

Algorithm 2 CDoAE

Step 1: Initialize parameters, $\boldsymbol{\gamma_n^0}, \boldsymbol{\beta_n^0}, \boldsymbol{\mu_n^0}, \boldsymbol{\Sigma_n^0}, a_n, b_n, c_n, d_n$, at each node n,

Step 2: Initialize global variable,$\boldsymbol{z^0}$, in master node

Step 3: Update hidden variable parameters,$\boldsymbol{\mu_n^{j+1}}$ and $\boldsymbol{\Sigma_n^{j+1}}$, according to equation (6.14) and equation (6.15) at each node.

Step 4: Update β_n^{j+1} according to equation (6.16) at each node.

Step 5: Update $\boldsymbol{\gamma_n^{j+1}}$ according to equation (6.24) at each node.

Step 6: Gather the weighted local node parameters, $\boldsymbol{\gamma_n^{j+1}}$ and $\boldsymbol{u_n^j}$, in the master node

Step 7: Update global variable, $\boldsymbol{z^{j+1}}$ according to equation (6.29)

Step 8: Distribute global variable $\boldsymbol{z^{j+1}}$ to local nodes.

Step 9: Update dual ADMM variable, $\boldsymbol{u_n^{j+1}}$ at each node according to equation (6.30)

Step 10: Check if the algorithm satisfies the stopping criteria according to equation (6.31)

Step 11: If not converged go back to Step 3.

6.2 Computational Complexity of CDoAE

In this part, computational complexity of the CDoAE method is investigated. CDoAE method is composed of two main parts, SBL and ADMM respectively. Computational complexity can be expressed in terms of the number of complex multiplications for each of these methods. The complexity of CDoAE is approximately the sum of the complexity of each algorithm given in [39], and [40] and it is given in Table 6.1. K is the number of points used in spatial spectrum, and M_n is the number of array elements in the n^{th} array.

Table 6.1: Total number of multiplication per iteration for SBL, ADMM and CDoAE

	Total number of multiplication per iteration
SBL	$O\{2K^3 + 6K^2M_n\}$
ADMM	$O\{K^3\}$
CDoAE	$O\{3K^3 + 6K^2M_n\}$

6.3 Parameter Exchange Between Master and Local Nodes

In this section, the number of variables exchanged between the master and slave nodes is discussed. It is assumed that, spatial spectrum is divided into K points and there are N sensor arrays with M elements.

In the proposed algorithm, there are two types of variable transfers. One is the parameter transfer to the master node from the local nodes. The other is the global variable transfer to the local nodes.

Local node n transfers $Kx1$ vector $\boldsymbol{v_n^{j+1}}$ to the master node which is given below,

$$\boldsymbol{v_n^{j+1}} = \boldsymbol{w_n} \odot (\boldsymbol{\gamma_n^{j+1}} + 1/\rho \boldsymbol{u_n^j}) \tag{6.32}$$

Then, the total number of variables transferred from the local nodes to the master node, V_{LM}, becomes,

$$V_{LM} = J(N-1)K \tag{6.33}$$

where J is the number of iterations.

After updating the global variable at the master array, it is transferred back to the local nodes. The total number of parameter transfer for this stage, V_{ML}, is,

$$V_{ML} = J(N-1)K \tag{6.34}$$

Therefore, the grand total for the parameter exchange,V_T, can be given as,

$$V_T = V_{LM} + V_{ML} = 2J(N-1)K \tag{6.35}$$

It is possible to decrease the number of local to master variable transfer, V_{LM}, with a small performance degradation. This may be especially important for local nodes which are power constrained like the IoT devices [41]. In the following section, a method for this purpose is presented.

6.3.1 Transferred Variable Reduction Method

Transferred Variable Reduction (CDoAE-TVR), is proposed to reduce the number of variables transferred from the local nodes to the master node with a small sacrifice on the achieved performance. TVR is based on the observation that as the iterations continue, the difference between $\boldsymbol{v_n^{j+1}}$ and $\boldsymbol{v_n^j}$ decreases. Especially the change in some of the elements of $\boldsymbol{v_n}$ become smaller and they can be ignored. Therefore, only the elements with large differences are transferred to the master node.

Hence, local nodes transfer $Rx1$ $(R <= K+1)$ vector, $\boldsymbol{\bar{v_n}^{j+1}}$, instead of $Kx1$ vector, $\boldsymbol{v_n^{j+1}}$. Let g be the scalar which has the encoded element existence information within $\boldsymbol{\bar{v_n}^{j+1}}$. Note that this encoding can be easily done in binary representation. An example is $g = [1, 1, 0, ..., 0, 0, 1]$, which represents that first two elements of $\boldsymbol{\bar{v_n}^{j+1}}$ are transmitted in a eight element vector. g is set as the last element of $\boldsymbol{\bar{v_n}^{j+1}}$. The remaining elements of $\boldsymbol{\bar{v_n}^{j+1}}$ are selected as in equation (6.36), where t_v denotes the predetermined threshold.

$$\bar{v}_{nr}^{\,j+1} = \begin{cases} v_{n_r}^{j+1}, & \left| \dfrac{v_{n_r}^{j+1} - v_{n_r}^{j}}{v_{n_r}^{j}} \right| > t_v \\ do\ not\ transmit,\ v_{n_r}^{j+1} & o.w. \end{cases}, \qquad (6.36)$$

$$r = 1, 2, ...R-1$$

After obtaining the parameter vector, $\boldsymbol{\bar{v_n}^{j+1}}$, it is transferred to the master array for the remaining steps of the proposed method. The proposed method, CDoAE, and the reduced variable transfer method, CDoA-TVR, are compared in the Simulations part both in terms of the performance and the number of transferred variables.

CHAPTER 7

SIMULATIONS

7.1 Simulations

In this section, the performances of the proposed algorithms are investigated. In the first part, performance of the CDoAE and CDoAE-TVR methods are compared with Decentralized MUSIC [9] and non-coherent Cramer Rao Bound (CRB) with uniform rectangular array geometries. The reason that Decentralized MUSIC is chosen as a benchmark algorithm is that, it does not require any synchronization between sensor nodes and it does not transmit the raw data, in the Decentralized MUSIC, the local nodes share only the covariance matrices. On the other hand, there are some limitations that Decentralized MUSIC can not overcome such as array geometry dependency for coherent source DoA estimation and the sensor positions of the local nodes must be known at the master node. Therefore, for the Decentralized MUSIC algorithm, it is assumed that the master node has the sensor position information of the other nodes and uniform rectangular arrays are used in the simulations to satisfy the geometry requirement of the Spatial Smoothing algorithm where Decentralized MUSIC algorithm uses for the coherent source DoA estimation. On the other hand, CDoAE is not dependent on any special array geometry even in case of coherent sources and does not require the sensor position information of the other nodes in the master node.

In the second part, performance of the CDoAE method is investigated for randomly distributed planar arrays. The purpose of this experiment is to show the effectiveness of the proposed CDoAE method for randomly distributed arrays and to prove the non-geometry dependent property of the CDoAE algorithm. To achieve this purpose,

performance comparison is obtained with the randomized array structures. Coherent and Non-coherent source scenarios in the first part is repeated with the new array structure and RMSE-SNR curves are obtained for CDoAE, Decentralized MUSIC and non-coherent Cramer Rao Bound.

7.1.1 Performance Analysis for Uniform Planar Arrays

In this part, CDoAE and CDoAE-TVR, are investigated for different scenarios which include single source, multiple source, coherent and non-coherent sources with URA geometry. Proposed methods are compared with Decentralized MUSIC [9] and non-coherent Cramer Rao Bound (CRB), [42], [43]. Non-coherent CRB [42] used in the simulations can be given as,

$$CRB = \left(\sum_{n=1}^{N} {CRB_n}^{-1} \right)^{-1} \tag{7.1}$$

where CRB_n denotes the CRB obtained for the n^{th} sensor node.

In the scenarios, two uniform rectangular array(URA) are constructed as in Figure 7.1 where local arrays and the direction of incoming signal are shown. Since decentralized MUSIC algorithm requires Spatial Smoothing [10] for coherent sources, two rectangular arrays with different orientations are selected. Note that CDoAE is not dependent on any special array geometry even in case of coherent sources. Decentralized MUSIC needs to know the array element positions for both master and local arrays. CDoAE does not need to know array element positions in the master node. Both methods do not need to know gain/phase mismatches between the arrays. Decentralized MUSIC is not an iterative method and only transfers covariance matrix elements to the master node. In the simulations, each array has a constant gain/phase coefficient different than the other arrays. Note that this does not pose any problem for DoA estimation in the individual arrays but array outputs can not be combined simply due to unknown constant gain/phase factors.

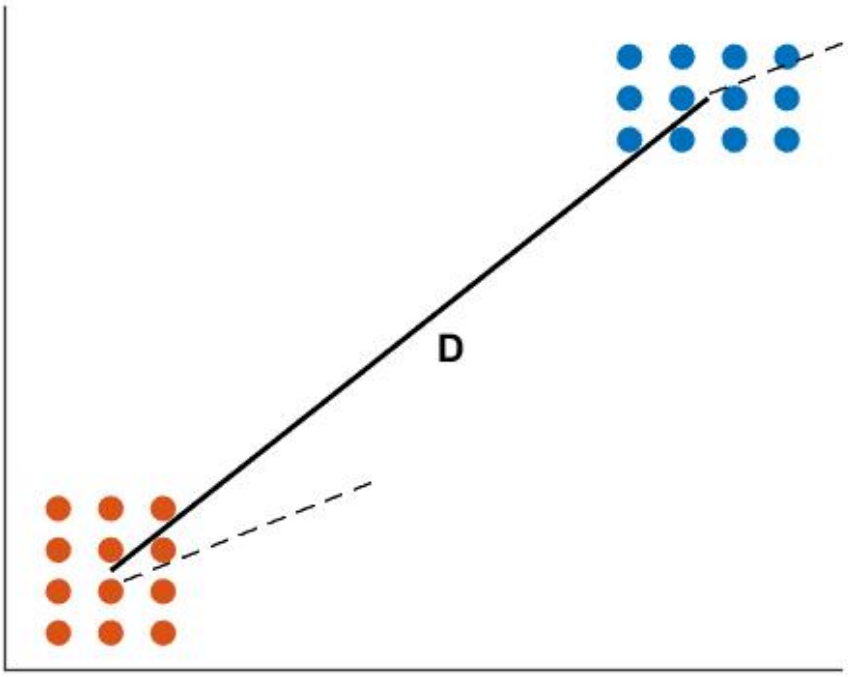


Figure 7.1: Two sensor nodes with URA with a separation of D distance are shown in addition to source direction represented with a dashed line.

In Figure 7.1, sensor arrays are positioned with a distance, D, between them. D can be any value and it need not to be known for both CDoAE and Decentralized MUSIC. Dashed lines show the direction of single source for both arrays. Note that the source direction is the same for both sensor nodes since it is represented in the global coordinate system. The orientation of arrays are different with respect to the source direction. Therefore, DoA performance for each array will be different and CDoAE is expected to enhance single array performances.

7.1.1.1 Single Source Scenario

In this case, the DoA of the signal source is chosen as 26.754 degrees in azimuth and 90 degrees in elevation. Moreover, the arrays are assumed as in the same plane with the source and DoA estimation is performed only in azimuth angle. Performance of the proposed algorithm is compared to the Decentralized MUSIC and SBL algo-

rithms performed in the local arrays as well as the non-coherent CRB in (7.1) over the RMSE-SNR curves. Simulations are conducted with SNR levels ranging from -5dB to 20dB, and 200 iterations are performed at each SNR level to obtain RMSE values.

Table 7.1: Simulation parameters for Single Source Scenario

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	1
Source directions(θ)	26.754 degrees
Source directions(ψ)	90 degrees
K	60
Number of iteration per SNR	200

In Figure 7.2 weight function in equation (6.26) for each array is shown.

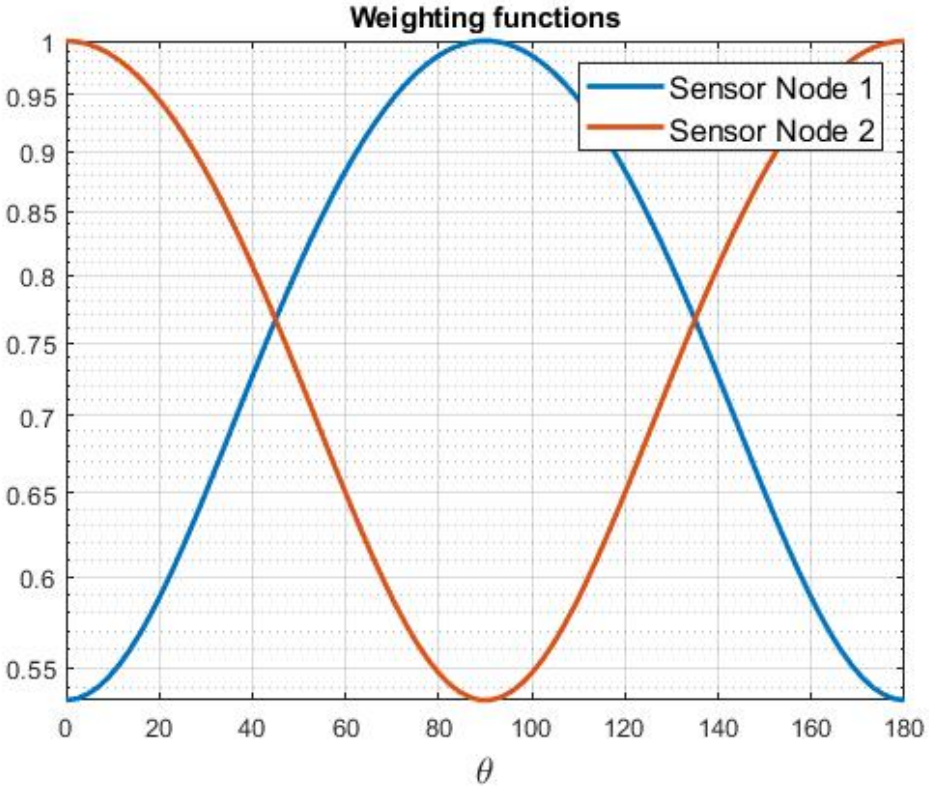


Figure 7.2: Weight functions for Array 1 and Array 2

In Figure 7.3, RMSE-SNR curves for the proposed algorithm and Decentralized MUSIC algorithm as well as the non-coherent Cramer Rao Bound(CRB) [42] are shown. Moreover, CDoAE is also compared with a method which we named as "SBL-AVG", where the local SBL results are averaged in the master node as follows,

$$\boldsymbol{\gamma_{avg}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\gamma_n^{J_n}}, \tag{7.2}$$

where $\boldsymbol{\gamma_n}^{J_n}$ represents the local $\boldsymbol{\gamma_n}$ estimates obtained at the n^{th} node in the J_n^{th} iteration for the local SBL estimations.

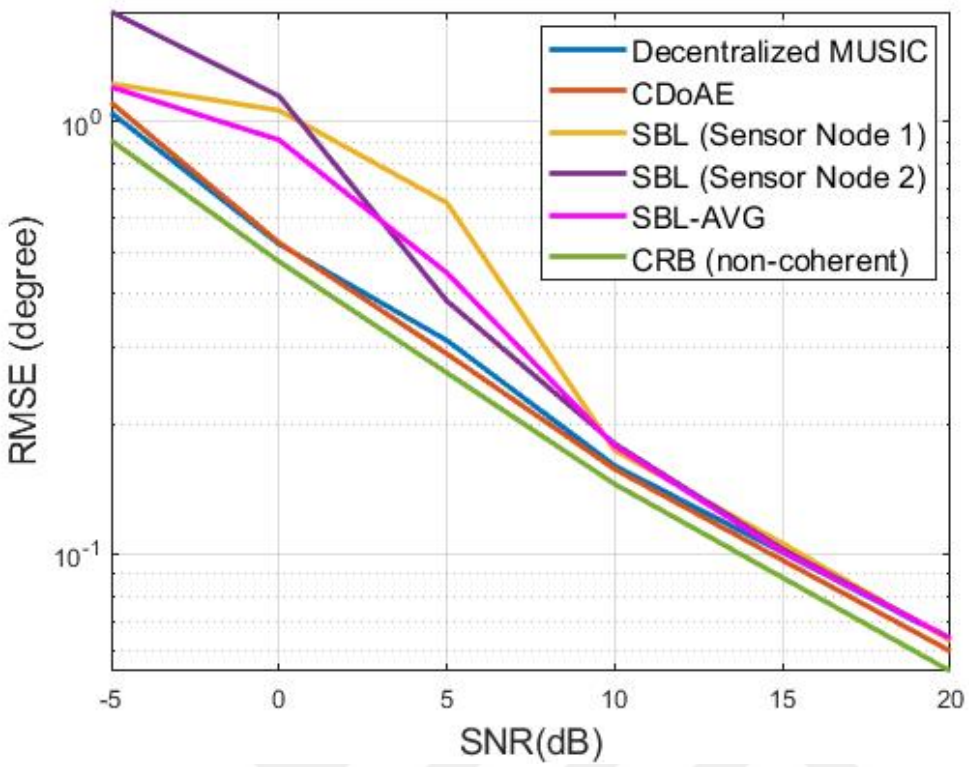


Figure 7.3: RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL, SBL-AVG and non-coherent CRB for single source.

As it can be seen in the figure, performances of both methods improves with the increase in SNR level as expected and approaches to the CRB. The performance of the CDoAE and Decentralized MUSIC algorithms are similar to each other for single source. They perform significantly better than single array SBL result. In addition, performance of the SBL-AVG is worse than CDoAE since CDoAE uses the global parameter, z^j, to update the local γ_n at each iteration which improves the local estimates significantly at Hidden Variable Update step given in Chapter 6.

In the following scenarios, simulations are extended for multiple uncorrelated and multiple coherent source cases by keeping the array structures similar to the single source scenario.

7.1.1.2 Multiple Uncorrelated Source Scenario

In this case, two uncorrelated sources with DoA angles, $\theta_1 = 26.754$ and $\theta_2 = 76.456$ degrees in azimuth and 90 degrees in elevation are considered. Again, DoA estimation is performed in only azimuth direction. In Figure 7.4, source directions with respect to the sensor nodes one and two are given. Dashed lines show the directions of multiple sources for both arrays. In the simulations for multiple uncorrelated source scenario, SNR levels and the number of iterations are the same as in the previous scenario.

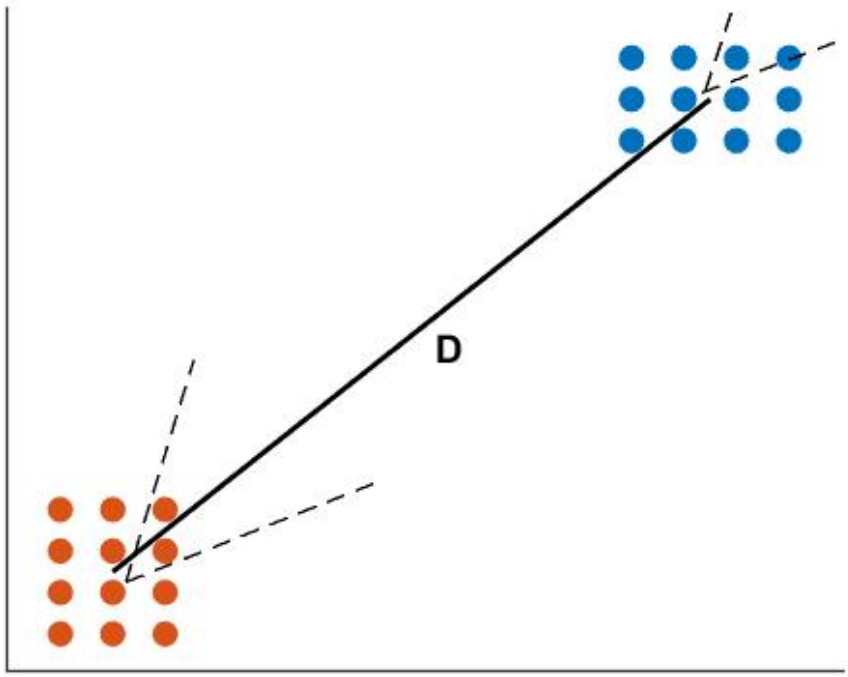


Figure 7.4: Two sensor nodes with URA with a separation of D distance are shown in addition to source directions represented with a dashed line.

In the Figure 7.5, RMSE-SNR curves for both algorithms are given. CDoAE performs better than the Decentralized MUSIC for all SNR levels and it approaches to the non-coherent CRB. The gap between CDoAE and the single array SBL increases for multiple sources.

Table 7.2: Simulation parameters for Multiple Uncorrelated Sources Scenario

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	2
Source directions(θ)	26.754 degrees 76.456 degrees
Source directions(ψ)	90 degrees 90 degrees
Coherent sources	No
K	60
Number of iteration per SNR	200

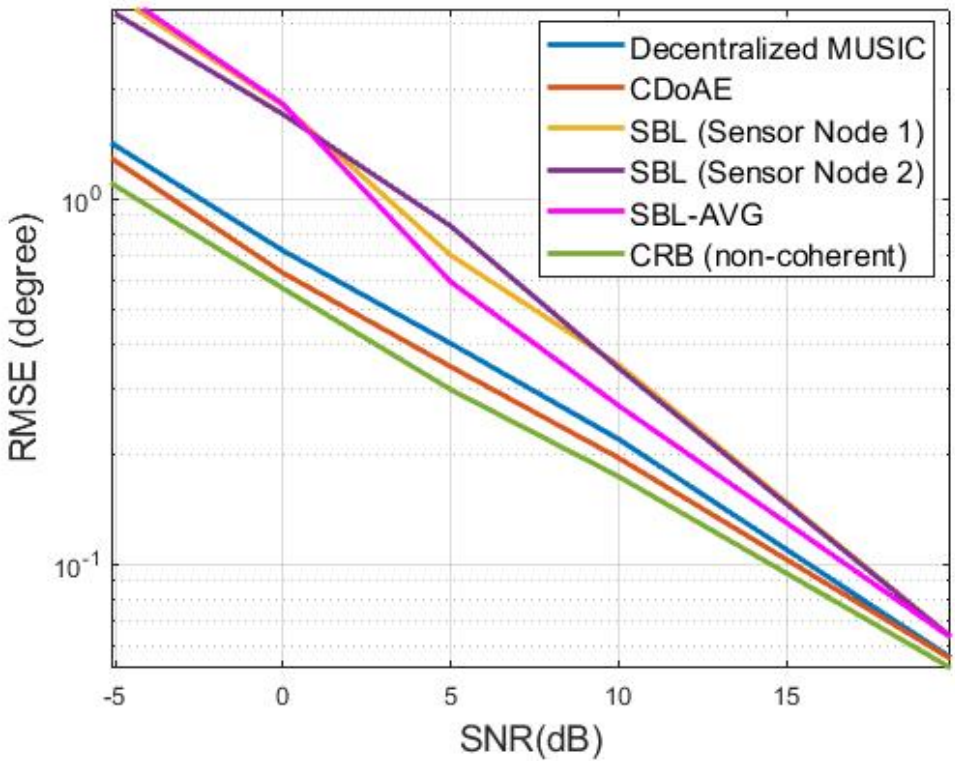


Figure 7.5: RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL-AVG and non-coherent CRB for multiple uncorrelated sources.

7.1.1.3 Multiple Coherent Source Scenario

In this case, the scenario in 7.1.1.2 is modified to have two coherent sources with the same DoA angles as in Figure 7.4. DoA estimation is performed in only azimuth direction. Also, the effect of decreasing the number of transferred parameters in CDoAE performance is investigated. CDoAE-TVR is compared with the original CDoAE where all the parameters are transferred to the master node and Decentralized MUSIC algorithm. For CDoAE-TVR method, two threshold values, t_v, in equation (6.32) are selected as $t_v = 0.01$ and $t_v = 0.02$ and the results are denoted as CDoAE-TVR(a) and CDoAE-TVR(b) respectively. The experiment results given in Figure 7.6 are obtained.

Table 7.3: Simulation parameters for Multiple Coherent Sources Scenario

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	2
Source directions(θ)	26.754 degrees 76.456 degrees
Source directions(ψ)	90 degrees 90 degrees
Coherent sources	Yes
K	60
Number of iteration per SNR	200

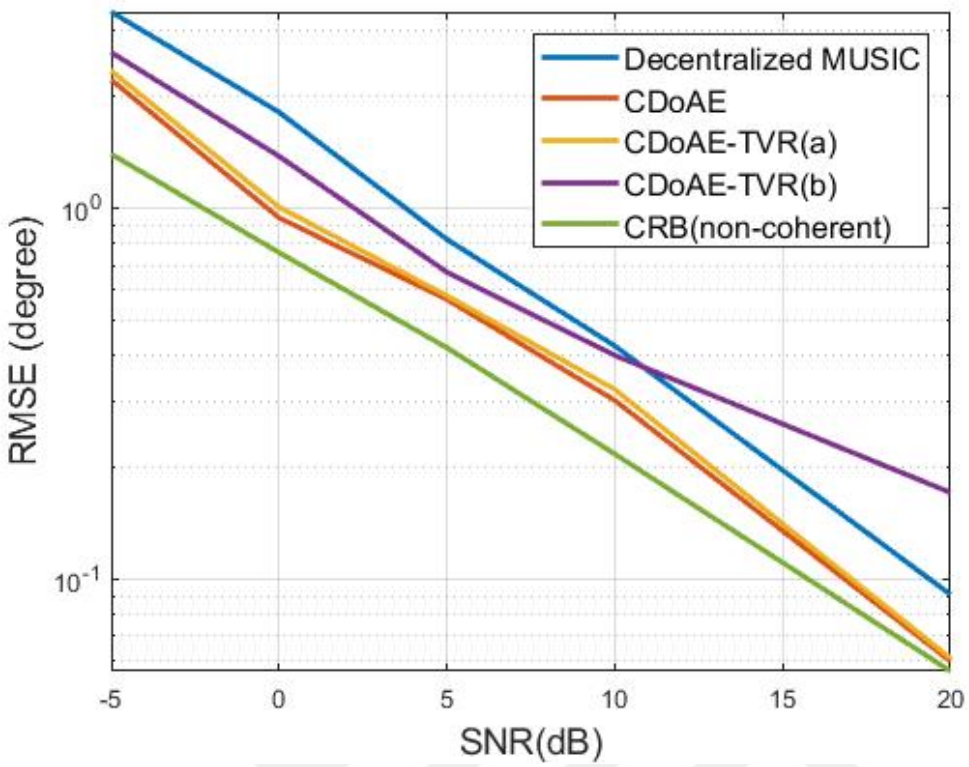


Figure 7.6: RMSE-SNR curves for Decentralized MUSIC, CDoAE, CDoAE-TVR(a), CDoAE-TVR(b) and CRB for multiple coherent sources.

As it is expected, both Decentralized MUSIC and CDoAE methods perform slightly worse than the uncorrelated scenario. However, the difference between Decentralized MUSIC and CDoAE is larger due to the fact that Decentralized MUSIC must use Forward-Backward Spatial Smoothing [10] and has array shrinkage problem whereas CDoAE inherently can solve the case thanks to the characteristics of the SBL method. The performance of the CDoA-TVR(a), where the threshold is chosen as $t_v = 0.01$ is very close to the CDoAE. On the other hand, CDoAE-TVR(b), where we chose the threshold $t_v = 0.02$, has larger errors compared to the CDoAE. While CDoAE-TVR(b) is better than Decentralized MUSIC at low SNR, its performance is degraded at high SNR since decreasing the number of parameters transferred to the master node generates larger error beyond a certain point.

In Figure 7.14, the percentage of the parameters transferred from the local nodes to the master node is shown. It turns out that the number of transferred parameters is almost

independent of the SNR level. This is due to the fact that most of the parameters are transferred to the master node at the beginning of the iterations. This fact is more obvious in Figure 7.8. The percentage of the total transferred variable is decreased approximately $\%15$ for CDoAE-TVR(a), where we set the threshold $t_v = 0.01$ and it is decreased $\%20$ for CDoAE-TVR(b) with the threshold $t_v = 0.02$ in equation (6.36).

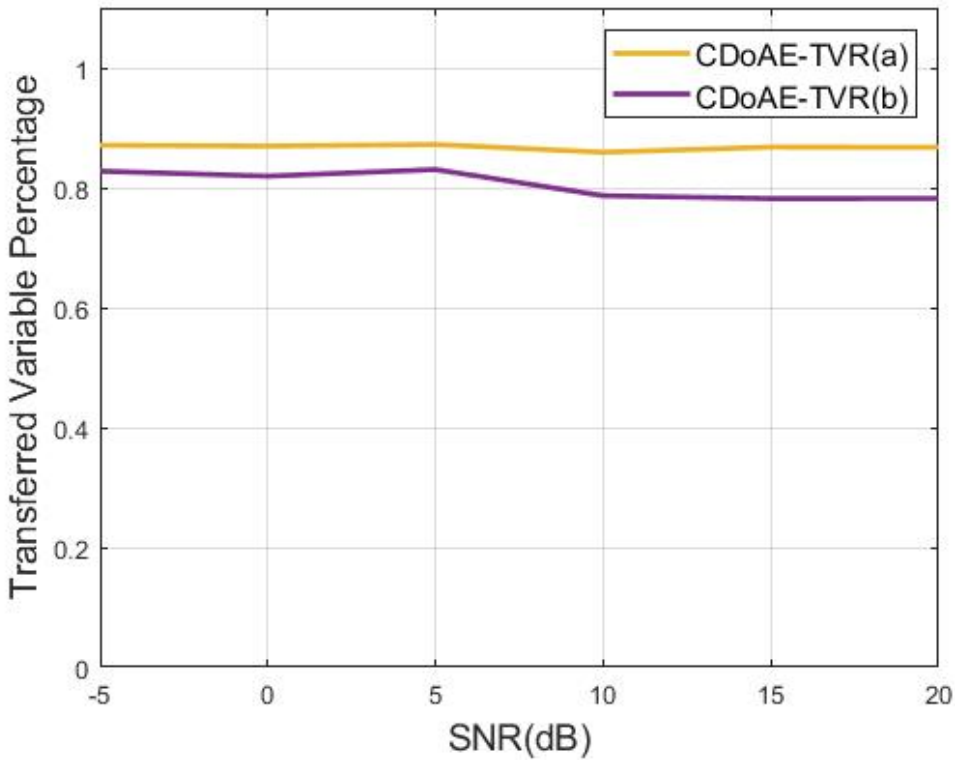


Figure 7.7: Percentage of average total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b)

In Figure 7.8, the number of transferred variables at each iteration is given for both CDoAE-TVR(a) and CDoAE-TVR(b). As it can be seen, the the number of transferred variables generally decreases as the iterations continues due to the convergence of the methods. At the beginning of the iterations, the algorithm generates an almost uniform spatial spectrum where the variance of the parameters decreases. This in term generates a sharp decrease in the number transferred parameters as shown in Figure 7.8. Then algorithm converges to the true spatial spectrum increasing the number of transferred parameters.

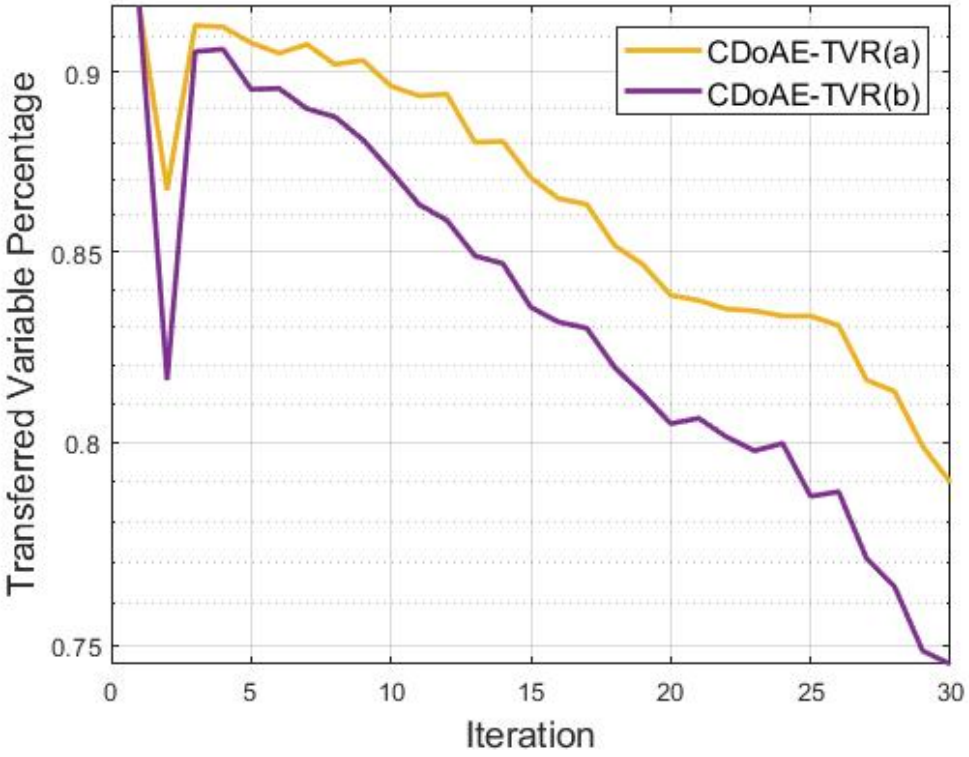


Figure 7.8: Percentage of total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b) for each iteration

7.1.2 2D DoA Estimation for Multiple Coherent Sources

In this case, the scenario in 7.1.1.3 is modified to observe the performances of the CDoAE and Decentralized MUSIC algorithms for 2D DoA estimation problem. In the simulations, azimuth angles of the sources, θ_1 and θ_2 are selected as 26.754 degrees and 76.456 degrees respectively. Moreover, the elevation angles of both sources, ψ_1 and ψ_2, are selected as 80.567 degrees. Performance of CDoAE is compared with Decentralized MUSIC and non-coherent CRB [42] for 2D DoA estimation [44] in both azimuth, θ, and elevation, ψ, angles separately. The experiment results are given in Figure 7.9.

In Figure 7.9, RMSE-SNR performances of the algorithms in azimuth, θ, estimation is represented by solid lines and the elevation, ψ, estimation performances are represented with dashed lines.

Table 7.4: Simulation parameters for Multiple Coherent Sources Scenario

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	2
Source directions(θ)	26.754 degrees 76.456 degrees
Source directions(ψ)	80.576 degrees 80.576 degrees
Coherent sources	Yes
K	200
Number of iteration per SNR	50

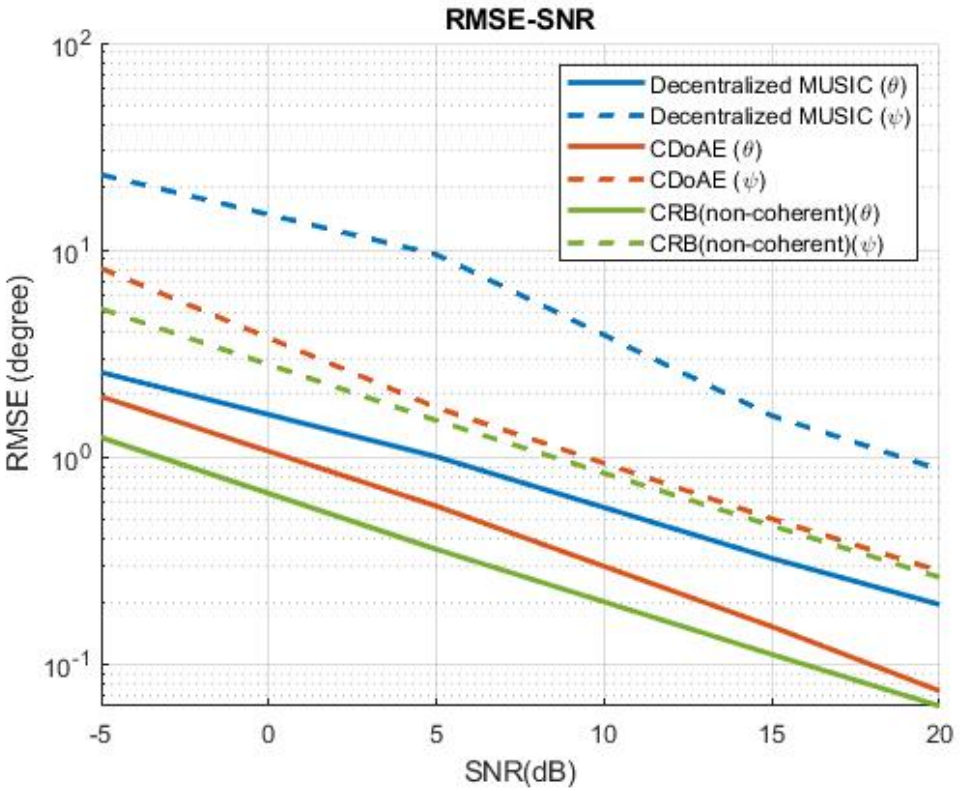


Figure 7.9: RMSE-SNR curves for Decentralized MUSIC, CDoAE and non-coherent CRB for 2D DoA estimation.

As it can be seen in Figure 7.9, similar results to the scenario in 7.1.1.3 are obtained in both azimuth and elevation estimation performances. Decentralized MUSIC suffers from the array shrinkage problem whereas CDoAE inherently can solve the case and approached to the non-coherent CRB in both azimuth and elevation.

7.1.3 Performance Analysis with Random Arrays

In this part, the simulations for multiple source scenarios in the previous part is repeated with random array geometry structures. Since the proposed CDoAE algorithm is not dependent to any special array geometry, simulations are conducted for non-coherent and coherent source scenarios to show the efectiveness of the CDoAE algorithm in random array geometry structure. Proposed CDoAE method is again compared with Decentralized MUSIC [9] and non-coherent Cramer Rao Bound (CRB).

In the scenarios, two random arrays are constructed as in Figure 7.10 and 7.11 where local arrays and the direction of incoming signal are shown. While constructing the random arrays, element positions of the Uniform Rectangular Array is perturbed by random displacements with a Gaussian distribution with mean, $\mu = 0$ and variance, $\sigma^2 = 0.4\lambda/2$ to keep the array aperture similar to the previous scenarios. In the simulations, each array has a constant gain/phase coefficient different than the other arrays. Note that this does not pose any problem for DoA estimation in the individual arrays but array outputs can not be combined simply due to unknown constant gain/phase factors.

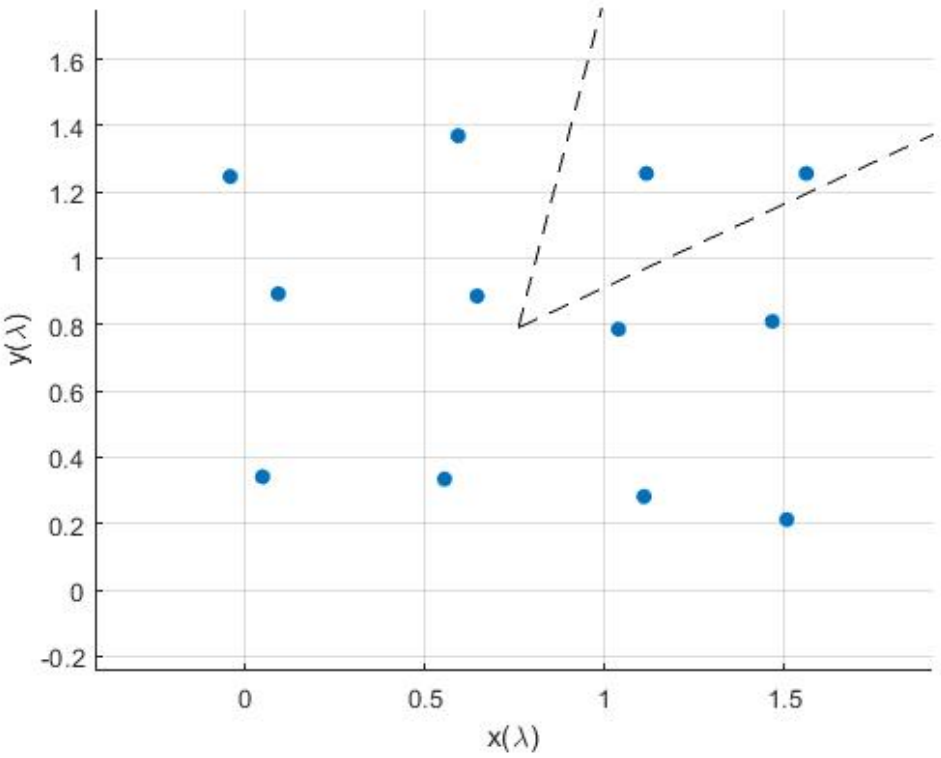


Figure 7.10: Sensor node one is shown and source directions represented with dashed line

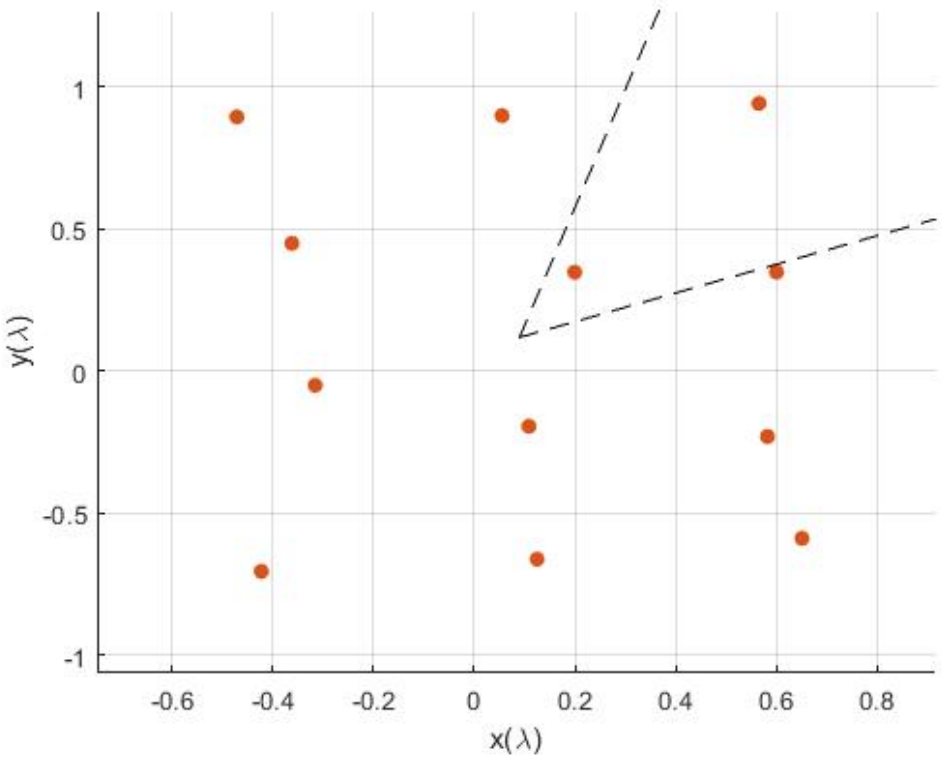


Figure 7.11: Sensor node two is shown and source directions represented with dashed line

7.1.3.1 Multiple Uncorrelated Source Scenario

In this case, two uncorrelated sources with DoA angles, $\theta_1 = 26.754$ and $\theta_2 = 76.456$ degrees are considered. In Figure 7.10 and 7.4, source directions with respect to the sensor nodes one and two are given. Dashed lines show the directions of multiple sources for both arrays. In the simulations for multiple uncorrelated source scenario, SNR levels and the number of iterations are the same as in the previous scenario.

Table 7.5: Simulation parameters for Multiple Uncorrelated Sources Scenario with Randomized Array Geometry

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	2
Source directions(θ)	26.754 degrees 76.456 degrees
Source directions(ψ)	90 degrees 90 degrees
Coherent sources	No
K	60
Number of iteration per SNR	200

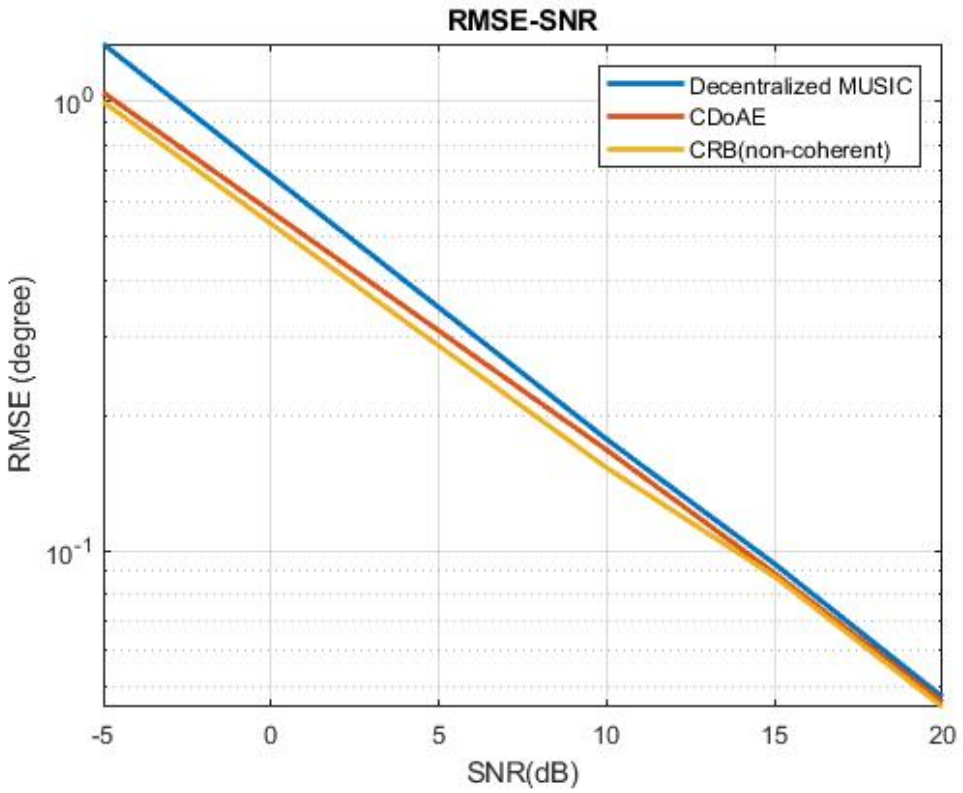


Figure 7.12: RMSE-SNR curves for Decentralized MUSIC, CDoAE, SBL and CRB for multiple uncorrelated sources.

In the Figure 7.12, RMSE-SNR curves for the proposed algorithm and Decentralized MUSIC algorithm as well as the non-coherent Cramer Rao Bound(CRB) [42] are given. As it can be seen in the figure, performances of both methods improves with the increase in SNR level as expected and approaches to the CRB. CDoAE performs better than the Decentralized MUSIC for all SNR levels and it approaches to the non-coherent CRB.

7.1.3.2 Multiple Coherent Source Scenario

In this case, the scenario in 7.1.2.1 is modified to have two coherent sources with the same DoA angles as in Figure 7.10 and 7.11. Also, the effect of decreasing the number of transferred parameters in CDoAE performance is investigated. CDoAE-TVR is compared with the original CDoAE where all the parameters are transferred to the master node and Decentralized MUSIC algorithm. For CDoAE-TVR method,

two threshold values, t_v, in equation (6.32) are selected as $t_v = 0.01$ and $t_v = 0.02$ and the results are denoted as CDoAE-TVR(a) and CDoAE-TVR(b) respectively. The experiment results given in Figure 7.13 are obtained.

Table 7.6: Simulation parameters for Multiple Coherent Sources Scenario with Randomized Array Geometry

Simulation Parameters	
Number of arrays	2
SNR	-5dB to 20dB
Number of sources	2
Source directions(θ)	26.754 degrees 76.456 degrees
Source directions(ψ)	90 degrees 90 degrees
Coherent sources	Yes
K	60
Number of iteration per SNR	200

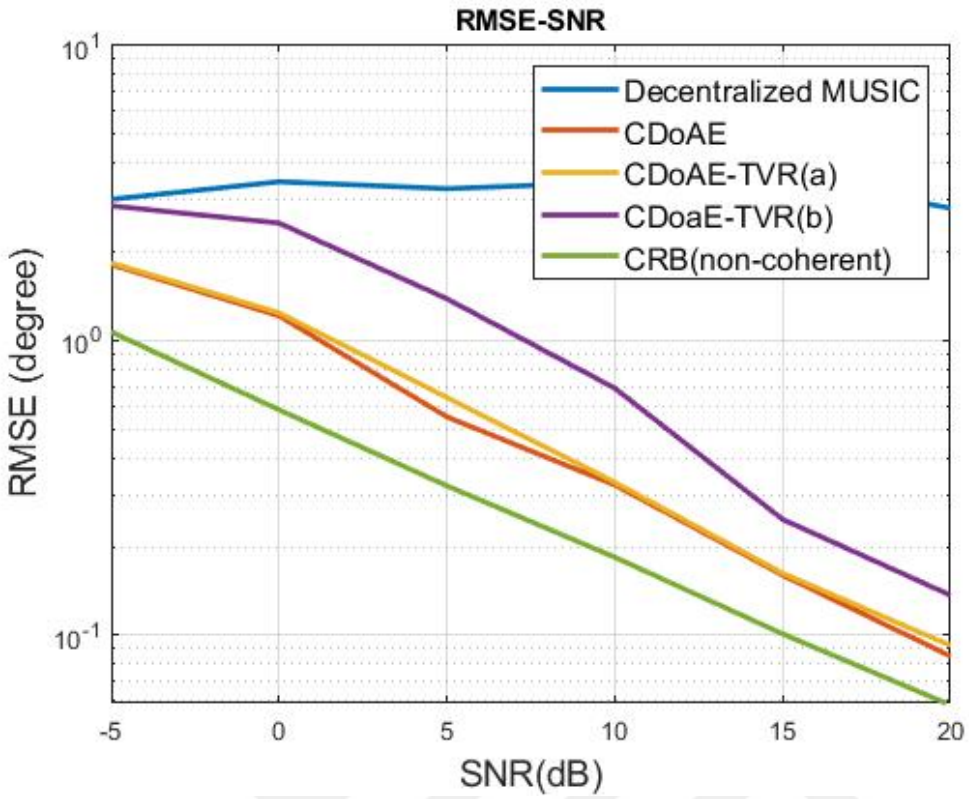


Figure 7.13: RMSE-SNR curves for Decentralized MUSIC, CDoAE, CDoAE-TVR(a), CDoAE-TVR(b) and CRB for multiple coherent sources.

As it is expected, Decentralized MUSIC algorithm performs worse than the Uniform Rectangular Array scenarios due to the fact that the Spatial Smoothing algorithm requiring a special array geometry. On the other hand, CDoAE methods perform better DoA estimations than the Decentralized MUSIC algorithm at all SNR levels. Moreovoer, the performance of the CDoAE methods increases as the SNR level increases as expected. In addition, the performance of the CDoA-TVR(a), where the threshold is chosen as $t_v = 0.01$ is very close to the CDoAE. On the other hand, CDoAE-TVR(b), where we chose the threshold $t_v = 0.02$, has larger errors compared to the CDoAE.

In Figure 7.14, the percentage of the parameters transferred from the local nodes to the master node is shown. It turns out that the number of transferred parameters is almost independent of the SNR level. This is due to the fact that most of the parameters are transferred to the master node at the beginning of the iterations. This fact is more

obvious in Figure 7.8. The percentage of the total transferred variable is decreased approximately $\%15$ for CDoAE-TVR(a), where we set the threshold $t_v = 0.01$ and it is decreased $\%25$ for CDoAE-TVR(b) with the threshold $t_v = 0.02$ in equation (6.36).

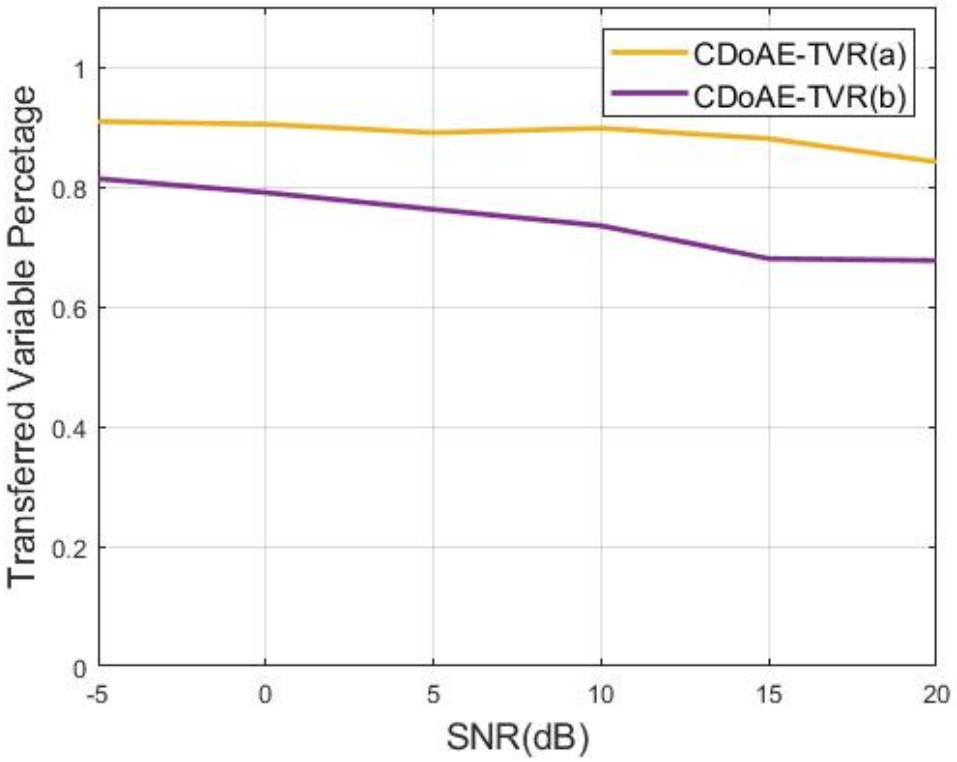


Figure 7.14: Percentage of average total number of shared variables from local nodes to master node with CDoAE-TVR(a) and CDoAE-TVR(b)

CHAPTER 8

CONCLUSION

8.1 Conclusion

This thesis investigates the subject of DoA estimation in distributed sensor array networks. While there are a variety of possible configurations for distributed sensor arrays, this study focuses primarily on a framework consisting of a master and local sensor arrays, each of which is capable of independently evaluating DoA and transferring specific parameters to the master array. It is known that the DoA estimation performances are effected by some limitations for distributed sensor arrays such as phase and frequency mismatch between the arrays, coherent sources and having sensor position information in master array. Therefore, in this thesis, a novel method called Collaborative DoA Estimation (CDoAE), which is aimed for use with distributed sensor arrays is presented.

In the CDoAE algorithm, the local SBL parameters are used in the ADMM framework to obtain a global solution by minimizing a common cost function. This solution is then transmitted back to the slave nodes where the slave node improves its local estimate with this new updated parameter set. This process is repeated in an iterative manner so that the final solution is obtained when the process converges or maximum number of iterations is exceeded. Therefore, CDoAE does not require gain/phase matching and frequency matching. It is not necessary to provide sensor location information from the local arrays to the master array since only a set of parameters are transferred between the master array and local arrays.

Furthermore, the CDoAE-Transmitted Variable Reduction (CDoAE-TVR) approach is presented to reduce the quantity of data that must be transmitted from the local

arrays to the master array. A variable selection algorithm is employed in CDoAE-TVR to decrease the number of transferred variables by only transferring variables with higher value changes than the pre-determined threshold.

Moreover, in the simulations, it is shown that CDoAE is effective for DoA estimation and improves the local performances significantly. Moreover, performance of the CDoAE is compared with the Decentralized MUSIC algorithm and it is shown that the CDoAE performs better DoA estimations than Decentralized MUSIC algorithm especially in coherent source cases. In addition, the proposed CDoAE-TVR, is an alternative method to decrease the number of parameters transferred to the master node performs significant savings with a small performance degradation. This is especially useful for power and bandwidth constrained IoT systems.

As future work, the CDoAE and CDoAE-TVR algorithms can be improved for the cases where there exist gain/phase mismatch between the elements of the local sensor arrays or for the cases if the sensor positions are known with an error. In both cases, the steps of the proposed CDoAE and CDoAE-TVR algorithms can be modified to estimate the gain/phase mismatches as well as the sensor position errors.

Moreover, using a grid based approach may effect the estimation performance negatively, especially if the gird size chosen larger. In those cases, the true DoAs may yield far away from the selected grid points for the algorithms, which is also called as off-grid problem, an it may occur in the real world applications as well. For those cases, signal model used in the SBL method can be extended to include the off-grid angle offset in the Bayesian Model. With this update in the SBL data model, proposed CDoAE and CDoAE-TVR methods can be extended for the off-grid source direction scenarios.

REFERENCES

[1] "Analysis of sparse Bayesian learning, author=Faul, Anita and Tipping, Michael," *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[2] P. Gerstoft, C. F. Mecklenbräuker, and A. Xenaki, "Multi Snapshot Sparse Bayesian Learning for DoA Estimation," *ArXiv*, vol. abs/1602.09120, 2016.

[3] J. Dai, N. Hu, W. Xu, and C. Chang, "Sparse Bayesian Learning for DoA Estimation with Mutual Coupling," *Sensors*, vol. 15, no. 10, pp. 26267–26280, 2015.

[4] X. Xiong, M. Zhang, H. Shi, A. Zhang, and Z. Xu, "SBL-Based 2-D DoA Estimation for L-Shaped Array With Unknown Mutual Coupling," *IEEE Access*, vol. 9, pp. 70071–70079, 2021.

[5] J. Dai, X. Bao, W. Xu, and C. Chang, "Root Sparse Bayesian Learning for Off-Grid DoA Estimation," *IEEE Signal Processing Letters*, vol. 24, no. 1, pp. 46–50, 2017.

[6] N. BniLam, J. Steckel, and M. Weyn, "Synchronization of Multiple Independent Subarray Antennas: An Application for Angle of Arrival Estimation," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 2, pp. 1223–1232, 2019.

[7] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 1722–1726, 2008.

[8] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA," *Signal Processing*, vol. 104, pp. 120–135, 2014.

[9] M. Wax and T. Kailath, "Decentralized processing in sensor arrays," *IEEE*

Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 5, pp. 1123–1129, 1985.

[10] A. Paulraj, V. Reddy, T. Shan, and T. Kailath, "Performance Analysis of the Music Algorithm with Spatial Smoothing in the Presence of Coherent Sources," in *MILCOM 1986 - IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, vol. 3, pp. 41.5.1–41.5.5, 1986.

[11] W. Suleiman and P. Parvazi, "Search-free decentralized direction-of-arrival estimation using common roots for non-coherent partly calibrated arrays," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2292–2296, 2014.

[12] W. Suleiman, M. Pesavento, and A. M. Zoubir, "Performance Analysis of the Decentralized Eigendecomposition and ESPRIT Algorithm," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2375–2386, 2016.

[13] W. Suleiman, P. Parvazi, M. Pesavento, and A. Zoubir, "Decentralized direction finding using Lanczos method," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 9–12, 2014.

[14] W. Suleiman, A. A. Vaheed, M. Pesavento, and A. M. Zoubir, "Decentralized direction-of-arrival estimation for arbitrary array geometries," in *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 1921–1925, 2016.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, 01 2011.

[16] Y. Wang, W. Yin, and J. Zeng, "Global Convergence of ADMM in Nonconvex Nonsmooth Optimization," *Journal of Scientific Computing*, 2019.

[17] Q. Wang, Z. Zhao, and Z. Chen, "Fast compressive sensing DoA estimation via ADMM solver," in *2017 IEEE International Conference on Information and Automation (ICIA)*, pp. 53–57, 2017.

[18] X. Zhang, T. Jiang, Y. Li, and X. Liu, "An Off-Grid DoA Estimation Method using Proximal Splitting and Successive Nonconvex Sparsity Approximation," *IEEE Access*, vol. 7, pp. 66764–66773, 2019.

[19] A. Alkhatib, "A Review of Wireless Sensor Networks Applications," in *The 2011 Conference on Innovations in Computing and Engineering Machinery*, 08 2011.

[20] S. Zhang and H. Zhang, "A review of wireless sensor networks and its applications," in *2012 IEEE International Conference on Automation and Logistics*, pp. 386–389, 2012.

[21] N. Pantazis and D. Vergados, "A survey on power control issues in Wireless Sensor Networks," *Communications Surveys Tutorials, IEEE*, vol. 9, pp. 86 – 107, 03 2008.

[22] E. Lawrence, K. Felix Navarro, D. Hoang, and Y. Lim, "Data Collection, Correlation and Dissemination of Medical Sensor Information in a WSN," pp. 402–408, 01 2009.

[23] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network For structural monitoring," pp. 13–24, 01 2004.

[24] J. Liu, J. Reich, and F. T, "Collaborative In-Network Processing for Target Tracking," *EURASIP journal on advances in signal processing*, 05 2003.

[25] C. Jr, *Wireless Sensor Networks: Architectures and Protocols*. 08 2003.

[26] V. Salatas and N. (U.S, "Object tracking using wireless sensor networks," 09 2005.

[27] Z. Yang, J. Li, P. Stoica, and L. Xie, "Chapter 11 - Sparse methods for direction-of-arrival estimation," in *Academic Press Library in Signal Processing, Volume 7* (R. Chellappa and S. Theodoridis, eds.), pp. 509–581, Academic Press, 2018.

[28] T. E. Tuncer and B. Friedlander, *Classical and modern direction-of-arrival estimation*. Academic Press, 2009.

[29] T. Külbay, "Direction of arrival estimation in sensor arrays with faulty elements," Master's thesis, Middle East Technical University, 2020.

[30] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[31] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.

[32] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[33] A. Kulkarni and T. Mohsenin, "Low Overhead Architectures for OMP Compressive Sensing Reconstruction Algorithm," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 6, pp. 1468–1480, 2017.

[34] H. Wang, X. Wang, M. Huang, C. Cao, and G. Bi, "Off-Grid DoA Estimation in Mutual Coupling via Robust Sparse Bayesian Learning," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, 2018.

[35] M. Grewal, L. Weill, and A. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration, Second Edition*. 02 2007.

[36] J. Dai, A. Liu, and V. Lau, "FDD Massive MIMO Channel Estimation with Arbitrary 2D-Array Geometry," *IEEE Transactions on Signal Processing*, vol. PP, 11 2017.

[37] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[38] D. N. Swingler, "An approximate expression for the Cramer-Rao bound on DoA estimates of closely spaced sources in broadband line-array beamforming," *IEEE transactions on signal processing*, vol. 42, no. 6, pp. 1540–1543, 1994.

[39] X. Xiong, M. Zhang, H. Shi, A. Zhang, and Z. Xu, "SBL-Based 2-D DoA Estimation for L-Shaped Array With Unknown Mutual Coupling," *IEEE Access*, vol. 9, pp. 70071–70079, 2021.

[40] R. Tibshirani, "Lecture notes in Convex Optimization," February 2016.

[41] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, 2015.

[42] T. Söderström and P. Stoica, "Statistical analysis of decentralized MUSIC," *Circuits, Systems and Signal Processing*, vol. 11, no. 4, 1992.

[43] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, 1989.

[44] Z. Hu, Q. Wu, J. Zou, and Q. Wan, "Fast and Efficient Two-Dimensional DoA Estimation for Signals with Known Waveforms using Uniform Circular Array," *Applied Sciences*, vol. 12, p. 4007, 04 2022.