Direction-of-Arrival Estimation Using Signal Processing on Graphs

Eldridge Alcantara

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Les Atlas, Chair

Shima Abadi, Chair

Linda Bushnell

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Direction-of-Arrival Estimation Using Signal Processing on Graphs

Eldridge Alcantara

Co-Chairs of the Supervisory Committee:
Dr. Les Atlas
Department of Electrical & Computer Engineering

Dr. Shima Abadi
Department of Electrical & Computer Engineering

Direction-of-arrival (DOA) estimation, the ability to find the direction from where a sound source originates, is a long-studied problem in array signal processing. Single-snapshot DOA estimation, in particular, brings a unique set of challenges and opportunities for research, particularly in scenarios where environments vary rapidly or limited computational resources are available. The development of single-snapshot DOA estimations systems has long been directed, however, by conventional signal processing. One limitation with conventional signal processing is that data is assumed to lie on a fixed domain of regularly spaced points, resulting in fixed estimation performance. Current methods for single-snapshot DOA estimation can overcome this limitation, and hence gain more flexibility, if the problem is studied from the point of view of signal processing on graphs (SPG). In SPG, data no longer lies on a fixed uniform grid, but instead on vertices of graph. Our conjecture is that we can improve DOA estimation performance by utilizing the framework of SPG. To support this conjecture, SPG principles and tools were employed to produce a new single-snapshot DOA estimation system, based on an existing estimation method from conventional signal processing, that incorporates graphs into the data processing chain. The system was then evaluated using real and simulated data inputs, and explanations were developed to understand the system's

output behavior. By leveraging the flexibility offered in SPG, this work shows that our approach to single-snapshot DOA estimation can not only produce outputs equivalent to those in conventional signal processing, but also, depending on the choice of graph structure, outputs with more improved DOA estimation accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CBF: Conventional Beamformer

SPG: Signal Processing on Graphs

DFT: Discrete Fourier Transform

GFT: Graph Fourier Transform

ℓ: Frequency bin

c: Sound speed (m/sec)

d: Array inter-element spacing (m)

f_o: Source frequency (Hz)

θ: Direction-of-arrival angle (deg)

N: Number of array elements, or number of graph vertices

$\mathbf{x}$: Measured data vector

$\mathbf{w}$: Additive noise vector

$\mathbf{a}(\theta)$: Steering vector

$\mathbf{A}$: Graph adjacency matrix

ACKNOWLEDGMENTS

The completion of this dissertation would not have been possible without many people in my life. I would first like to thank my co-advisors, Professor Les Atlas and Professor Shima Abadi. Professor Atlas welcomed me into his lab with open arms during my first year when I was searching for a new advisor, while Professor Abadi started me on the right track to research during my second year when she agreed to mentor and eventually co-advise me in working on DOA estimation. Jointly, they have been sources of positivity and encouragement throughout my entire Ph.D. program. Even when I reached several roadblocks in my work, they never wavered and continued to believe in me to find new solutions to the problem. I would also like to acknowledge my following General and Final Exam committee members: Professor Linda Bushnell, Professor Mehran Mesbahi, Professor Azadeh Yazdan, and Professor Marina Meila. Without all these people, I would not have been able to reach this significant milestone in my life and career.

I would like to thank all my generous sponsors during my Ph.D. program. These include the ECE Department for numerous quarters as a a Lead TA and for my most recent IBM PhD Fellowship Award for Autumn 2020, the DoD SMART Scholarship for funding the first five years of my program, and ONR Code 321 for funding me as an RA for several quarters. At ONR, I would like to acknowledge Dr. Keith Davidson and Dr. John Tague for being the first to champion the early stage of my work in signal processing on graphs and for continuing to support the area to this day.

During my Ph.D. program, I have been fortunate enough to wear many hats as a student researcher, a TA, and a group fitness instructor, meeting many great people along the way. I would like to start by thanking my fellow lab-mates over the years: Jessica, Symbol, Renshu,

Elliot, Brad, Tyler, Dave, Scott, Tommy, Ruobai, Cuinn, Jordan, Felix, and John. I would like to thank Jim and Ken at ET&L, the staff at CTL, my co-TAs (Trang, Sang, Thomas, Astrini, Swati, and Mayoore, to name a few) over the years, and my EE 235 students for all helping me become a better teacher. I have to thank Professor Ostendorf for supporting me during my predoctoral lecturing appointment for EE235 and Professor Fazel for letting me volunteer TA for her class when I first entered UW. I also have to single out Professor Bushnell for letting me lead discussion section every quarter she taught EE 235 and giving me the chance to hone my teaching skills over the last few years. I would also like to acknowledge UW Recreation, Annie, Carrie, and Jess for training me and giving me the opportunity to teach Cardio Kickboxing and further enhance my Husky experience.

I would like to also thank my home department in Electrical & Computer Engineering, particularly all the students, faculty, and staff I have had the pleasure to interact with over these last seven years. It has been an honor to be a part of the ECE Department. This would not have been possible without my initial phone call and email exchange with Professor Payman Arabshahi and Professor Sumit Roy in early 2013, so thank you to them for accepting my application to come to UW.

At my full-time job at NIWC Pacific, I would like to thank my supervisor Tom for supporting me throughout my Ph.D. journey, my Ph.D. work mentor Pedro for all our talks about my work and for introducing to the area of signal processing on graphs, and Eric, Guy, Paul, LorRaine, and Emily for helping me with my DoD SMART Scholarship submission and writing all my letter of recommendations for graduate school as well.

Finally, I have to acknowledge my personal support system. Thank you to all my friends in Seattle (Man, Jimmy, Alton, Davie, and Riz, to name a few) for showing me that there is life outside of school. To all my friends and family in San Diego, especially to my mom, dad, and brother, Andrews, thank you for all your love and encouragement during this time. I could not have completed this dissertation without all of you!

Chapter 1

INTRODUCTION

Sensor arrays and array signal processing systems have a long history of enhancing our situational awareness. One way they do this is by recording and processing data in order to find the direction of a detected sound source relative to the array, a technique referred to in array signal processing as direction-of-arrival (DOA) estimation. Methods for DOA estimation typically assume the availability of multiple, or even infinite, snapshots of data in time [1]. Estimating DOA with limited snapshots, or even a single snapshot, however, is just as useful in practice. Single-snapshot DOA estimation (Figure 1.1) is practical for physically-constrained situations such as in sonar processing where sound speed varies and only a limited number of snapshots may be available to process. Processing a single snapshot is also valuable in reducing the computational load for both DOA estimation and the rest of an array processing system [2]. For this reason, and all others described, the focus of this dissertation is on single-snapshot DOA estimation systems.

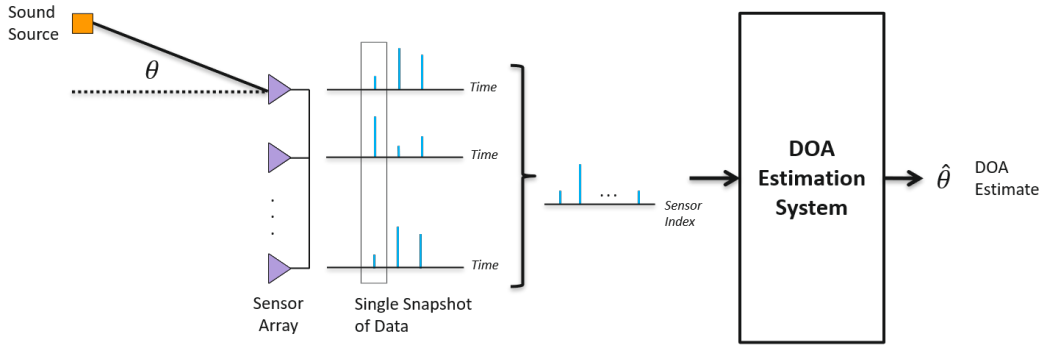


Figure 1.1: Single-snapshot direction-of-arrival (DOA) estimation system.

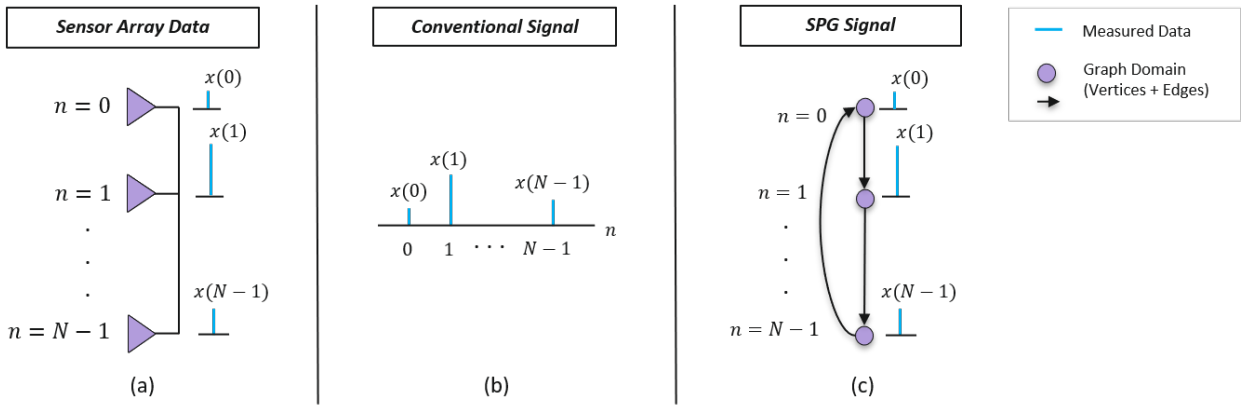


Figure 1.2: Representing sensor array data with conventional signal processing or signal processing on graphs (SPG).

Previous work in this area has long been directed by the framework of conventional signal processing. There are estimation systems that rely on a power spectrum calculation [2], while others rely on a Discrete Fourier Transform (DFT) [3] or principles from compressed sensing [4]. If there is one limitation across all these methods, it is this: unless a pre-processing system is included [5], the performance of these systems is fixed and cannot change. The reason for this, we posit, is because of the assumption made on the data modeling and processing domain. Across all the examples we listed here, it is assumed that the domain on which data lies is restricted to a regular and uniform grid of points (Figure 1.2ab).

This restriction can be overcome if we incorporate principles from another area of work that uses a more flexible domain for modeling and processing data. One viable option is the framework of signal processing on graphs (SPG). In contrast to conventional signal processing, data in SPG is modeled and processed on a different domain (Figure 1.2c) made up of a set of vertices and edges that are jointly referred to as a graph [6]. Representation on a graph is a natural construct for numerous information-collecting structures. These include temperature sensor networks, road traffic networks, brain networks, a 3D point cloud, and a Twitter user's network of followers, to name a few, in which a graph vertex can represent a

network node and graph edges can represent the physical proximity or correlated connection between network node pairs [7]. If SPG can be applied to these scenarios, then there is no reason to believe it cannot handle our information of interest: a single snapshot of data collected from an array of sensors.

Furthermore, the application of SPG to our problem is advantageous. Graphs can be constructed in many ways, which thereby offers us more choices for domains for data modeling and processing. Having more choices is important because the likelihood of finding a domain, or graph structure, with better performance increases. For this reason, and all others described in the last paragraph, the hypothesis for this dissertation is this: ***the performance of single-snapshot DOA estimation systems can be improved by developing an estimation system on a graph domain***.

To support this conjecture, our approach is to create a single-snapshot DOA estimation system that builds upon existing work in DOA estimation with the aid of SPG and then comprehend the behavior of the system we have built by analysis. We expand and justify the details of this approach by summarizing and considering all prior work in Chapter 2. We then proceed with a slice-and-dice method (Figure 1.3) to develop our system by first considering the system input model in Chapter 3 and then the actual estimation system in Chapter 4. The next two chapters will then focus on estimation performance evaluation using real data in Chapter 5 and then simulated input data in Chapter 6. Based on the observations that are made on estimation performance, Chapter 7 focuses on explaining why the system behaves the way it does. Finally, we close this dissertation in Chapter 8 with a summary of our contributions and offer ideas for future work in single-snapshot DOA estimation.

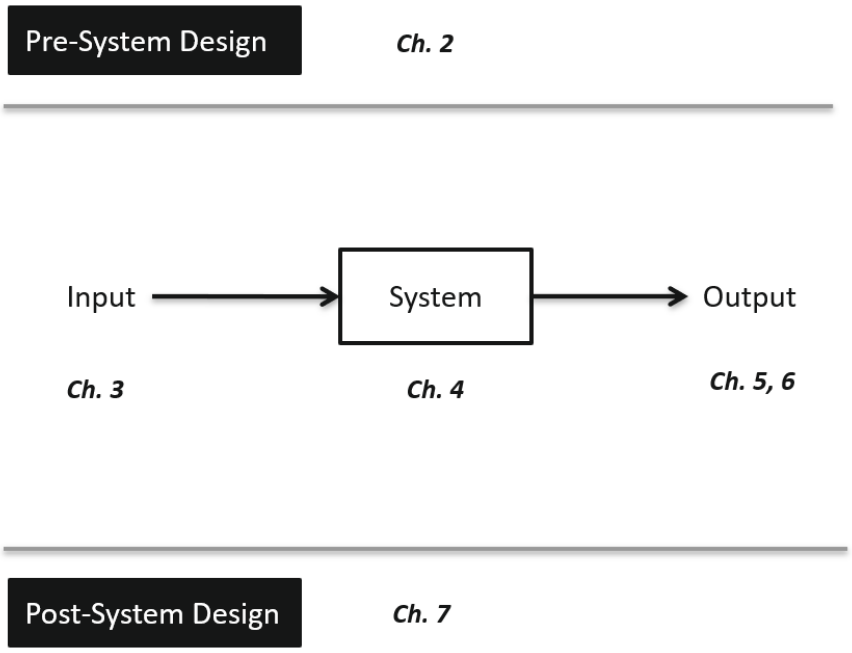


Figure 1.3: Overview of rest of chapters.

Chapter 2
BACKGROUND AND ASSUMPTIONS

In this chapter, the current state of two relevant areas of work is discussed. We start with a summary of the area of single-snapshot DOA estimation using the conventional signal processing framework and then proceed to describe advances in DOA estimation using signal processing on graphs (SPG). By summarizing these areas, the goal is to develop and justify our own approach, with clearly stated assumptions, to the problem of improving the performance of single-snapshot DOA estimation with SPG.

2.1 Current State of Single-Snapshot DOA Estimation

In the current state, single-snapshot DOA estimation remains a problem of interest among scientists and engineers. As one of the earliest papers in the topic noted, estimation performance degrades more from reducing the number of array elements rather than the number of temporal snapshots [8], This area, therefore, is motivated, and this motivation has led to increased curiosity. Interest in the area has grown over the years as single-snapshot DOA estimation has been shown to be relevant in applications such as automotive radar systems [2] and array signal processing for smart cities [9] in which limited snapshots of data are available for processing and estimating DOA. The advancement in knowledge in single-snapshot DOA estimation can be summarized according to which array configuration is considered and which approach was used to develop the actual estimation algorithm.

2.1.1 Uniform Line Array

The majority of contributions in single-snapshot DOA estimation has been for uniform line array (ULA) receivers. Prior work has shown that estimating DOA from a single-snapshot in

time is possible, and numerous methods have been developed using one of two approaches. The first approach is to translate methods from conventional signal processing that use multiple or infinite snapshots to the single-snapshot case. This translation approach was found to show improvements in estimation accuracy, under certain conditions, for existing DOA estimation methods based on the input power spectrum like conventional (Bartlett) beamforming [3] and Capon [10], the input subspace like multiple signal classification (MUSIC) [11] or estimation of signal parameters via rotational invariance (ESPRIT) [12], and singular value decomposition of the input like Matrix Pencil [13].

Another popular approach to single-snapshot DOA estimation is to develop new methods that are grounded in theory from either signal processing, linear algebra, or probability and statistics. Those that have followed this approach have also shown improvement in estimation accuracy, under certain conditions, when existing theory like compressed sensing [4], spatial windowing [14], iterative interpolation [5], and Bayesian statistics [8] were considered.

With numerous methods available, performance comparison studies are natural to conduct. While some have compared multiple DOA estimation methods [2,11], most prior work compare with the estimation method they drew inspiration from. There is one method that is often used in performance comparison studies, and that is the conventional beamformer. It is a method that has been found to either show an improvement in performance in certain conditions [4] or a performance decline [11] based on which other method it is compared to.

To conduct these performance evaluation studies, all prior work related to ULAs assume a far-field plane wave model for the input signal [2]. In this model, data $\mathbf{x} \in \mathbb{C}^N$ recorded by a ULA, assuming a single source, is modeled as a linear combination of a steered source component $\mathbf{s} \in \mathbb{C}^N$ and a noise component $\mathbf{w} \in \mathbb{C}^N$ (Equation 2.1).

$$\begin{aligned} &\mathbf{x} = \mathbf{s} + \mathbf{w} \\ &\mathbf{s} = \mathbf{a}(\theta)s(t_o) \\ &\mathbf{w} = \mathcal{CN}\left(\mathbf{0}, \frac{1}{SNR}\mathbf{I}\right) \\ &\text{where } \mathbf{a}(\theta) = \left[e^{j2\pi f_o \frac{d\sin(\theta)}{c} n}\right]_{n=0}^{N-1} \text{ and } s(t_o) = e^{j2\pi f_o t_o} \end{aligned} \tag{2.1}$$

The source signal $s(t)$ is modeled as a narrowband complex exponential signal with a frequency f_o (in Hz) and is evaluated at a single-snapshot in time t_o (in sec). In the far-field plane wave model, the receiver is assumed to be far enough away from the source that each array element receives a time-delayed version of $s(t)$. Assuming the source is located at angle θ (in deg) relative to the array, and that the data is collected by an N-element ULA with inter-element spacing d (in m), the data recorded by each array element can be modeled as a shift in phase of the source. The phase shift in the source can be characterized by a steering vector $\mathbf{a}(\theta) \in \mathbb{C}^N$ with constant sound speed c (in m/sec). The noise component, on the other hand is modeled as identically distributed, circularly symmetric complex normal with with zero mean and variance $\frac{1}{SNR}$.

2.1.2 Other Array Configurations

Beyond a ULA, a few other contributions have been made with other array configurations. Prior work has found that single-snapshot DOA estimation is possible with configurations like a non-uniform line array [15], a non-uniform rectangular array [16], uniform circular array [17], and randomly placed array elements [18]. For these prior work, new algorithms were developed based on existing theory such as principal singular vector, dual functions, compressed sensing, and Fourier coefficient interpolation. For all these methods, evidence of improved estimation accuracy were found over other existing methods, with some showing an ability to get closer to the theoretical estimation error bounds, particularly at low amounts of noise [15].

2.2 Current State of DOA Estimation Using Signal Processing on Graphs

As shown in the last section, DOA estimation is directed heavily by the framework of conventional signal processing. Approaching the problem from the point of view of SPG is what separates this dissertation from the work that has just been described. While SPG has been shown to be valuable for applications like smart cities [19], brain networks [20], and biological networks [21], to name a few, DOA estimation using SPG has not received as

much attention. Indeed, this topic of DOA estimation using SPG is a recent development with this dissertation being closely related to only a small body of work. This work includes our own earlier contributions to the area, along with contributions from a few others.

Our early contributions to the area demonstrated how SPG can be applied to the problem of DOA estimation. In [22, 23], we addressed how ULA measured data, characterized as a far-field plane wave, can be modeled on a one-dimensional graph with undirected edges. In addition, we attempted to develop a new DOA estimation method by applying two existing operators in SPG: the the Graph Fourier Transform (GFT) and graph signal smoothness. The results from these earlier work uncovered an ambiguity issue when estimating DOA, in which we are not able to resolve DOA angles that were positive from those that were negative in value, an issue known as top-bottom ambiguity.

This dissertation attempts to improve on what we did by considering another approach. Instead of trying to develop a new estimation algorithm, we consider translating an existing DOA estimation algorithm from conventional signal processing to the graph domain. This work is most closely related to three recent publications [24–26] in which they demonstrated that the estimation methods of differential beamforming and MUSIC were translatable to the graph domain and showed improvements in estimation accuracy over these existing techniques. Input data modeling, however, was handled differently from our own prior work. Some developed a model for ULA data using multiple snapshots of time and with a fixed, fully-connected graph configuration across two dimensions [25,26], while another also utilized a two-dimensional graph but considered instead an arbitrary array configuration on a single plane [24]. Therefore, there exists an opportunity for contribution with single-snapshot DOA estimation using a one-dimensional graph structure and another existing technique from conventional signal processing.

2.3 Summary of Assumptions

Taking into account the body of work we have just summarized and the goal of laying a solid foundation of work, we choose to make the following assumptions:

1. **Source**: To narrow the focus of our study with the intention of making extensions later on, we assume a single narrowband and stationary sound source located on the same plane as the receiver array. The source's narrowband frequency will be denoted by f_o (in Hz) and its DOA angle will be denoted by θ (in deg).

2. **Environment**: Consistent with prior work, we assume a homogeneous transmission channel or environment characterized by a constant sound speed c (in m/sec).

3. **Receiver Array**: Consistent with prior work that have developed and introduced new DOA estimation algorithms, we assume a uniform line array (ULA) at the receiver with N array elements and inter-element spacing d (in m).

4. **Source-Receiver Distance**: Consistent with prior work that have developed and introduced new DOA estimation algorithms, we assume a source-to-receiver distance far enough away to assume the received data can be modeled as far-field plane waves

5. **Input Data**: Distinguishing our work from recent prior work in DOA estimation using SPG, we assume a single snapshot of input data that can be represented as a vector $\mathbf{x}$ such that
$$\mathbf{x} = \begin{bmatrix} x(0) & x(1) & \dots & x(N-1) \end{bmatrix}^T \in \mathbb{C}^N$$

6. **Graph Structure**: Distinguishing our work from some prior work in DOA estimation using SPG, we assume a one-dimensional and time-invariant graph structure for data modeling and processing, where the graph structure is denoted by variable G .

Chapter 3
MODELING SYSTEM INPUT

Before we can start developing a single-snapshot direction-of-arrival (DOA) estimation system with signal processing on graphs (SPG), we must first address what directs system development the most: the model for the system input. Without it, there is no way to predict the output or characterize the performance of an estimation system. Rather than creating an entirely new input model, the focus of this chapter is to build upon existing input models from DOA estimation and conventional signal processing. In conventional signal processing, the most common input model in DOA estimation research is the far-field plane wave [27]. The plane wave's equation, assuming a ULA and a single snapshot of time, is well-known and already defined in Chapter 2. The problem with this expression is that it defines the measured data samples of a plane wave on a fixed domain of uniformly spaced grid points, so the concept of a graph is not yet incorporated into the model. Broadening this model so that plane waves are defined on the domain of a graph is the challenge for this chapter. Our conjecture is that **if principles from SPG are applied, then the far-field plane wave can be extended and modeled on the graph domain.**

3.1 Prior Work and Approach

Modeling plane waves with SPG is a topic that has been addressed by a few recent papers, but there are still opportunities for improvement. In [24–26], graph-based input models for the far-field plane wave were developed, but they were made under a different set of assumptions than ours. These assumptions include multiple sound sources, two-dimensional graphs, arbitrary sensor array configurations, and DOA angles defined on planes different from the sensor array. To build a solid foundation of work in SPG-based single-snapshot

DOA estimation, our approach is to express the far-field plane wave with the assumptions we defined in Chapter 2: a single sound source, one-dimensional graph, a ULA, and a DOA angle defined on the same plane as the ULA.

To develop this graph-based input model, SPG literature was reviewed to learn how SPG practitioners model data on a graph. The two most cited papers in SPG [6, 28] were studied first, followed by the recent papers in graph-based DOA estimation that were cited in the last paragraph. Across these papers, the aim was to find a set of well-accepted principles for data modeling that would guide the work in this chapter.

From this review, a clear direction emerged for our input modeling task. SPG practitioners model input data by using the concept of a graph signal. A graph signal is best understood as an extension of signals from conventional signal processing by expanding the mathematical model of a signal to allow for data samples that are defined on a graph. A graph signal accounts for the underlying graph domain by using a specification that consists of two components: a vector and a matrix. Each component represents a unique piece of the data model. As illustrated in Figure 3.1, the graph signal vector is used to mathematically represent the actual measured data (in blue), while the graph signal matrix is used to describe the data's underlying graph domain (in red). This common framework for data modeling breaks down our work into two challenges: finding an expression for the graph signal vector for the far-field plane wave, as well as an expression for plane wave's graph signal matrix. Once these expressions are found, we can then validate them through an implementation in software. The next three sections are used to summarize our findings.

Figure 3.1: Two components of a graph signal.

3.2 Expressing Graph Signal Vector for Far-Field Plane Wave

To develop an expression for the graph signal vector of a far-field plane wave, the SPG framework was studied and applied. The common practice among SPG practitioners is to structure the graph signal vector in the following way: the size of the vector is equal to the number of measured data samples [7]. Therefore, for the case in which there are N samples of data, an N-element vector $\mathbf{x} \in \mathbb{C}^N$ is used, where $\mathbf{x} = \begin{bmatrix} x(0) & x(1) & x(2) & \dots & x(N-1) \end{bmatrix}^T$. The elements in the vector are ordered so that each element $x(n)$ is associated with the data sample on graph vertex n. This means that $x(0)$, for example, is interpreted as the data sample on graph vertex $n = 0$, while $x(N-1)$ specifies the data on vertex $n = N-1$. As long as all vertices on a graph are indexed, all measured data on a graph can be represented naturally by a graph signal vector.

Applying this framework to a far-field plane wave, we found an expression for graph signal vector $\mathbf{x}$ with direct ties to conventional signal processing (Equation 3.1).

$$\begin{aligned} &\mathbf{x} = \mathbf{a}(\theta)s(t_o) + \mathbf{w} \\ &\mathbf{a}(\theta) = \left[e^{j2\pi f_o \frac{d\sin(\theta)}{c} n} \right]_{n=0}^{N-1} \text{ and } s(t_o) = e^{j2\pi f_o t_o} \end{aligned} \tag{3.1}$$

This expression is, in fact, equivalent to the plane wave's equation in conventional signal processing. The same variables are used: $\mathbf{a}(\theta)$ is the steering vector at angle θ, $s(t_o)$ is the narrowband source signal at frequency f_o and time snapshot t_o, and $\mathbf{w}$ is additive complex noise. The only thing that we modified was the definition of index n. In conventional signal processing, n is the array element index, but in SPG, n now represents the graph vertex. We were able to update the definition of index n and still use the equation from conventional signal processing by assuming the following: that the set of graph vertices are ordered in the same way as the sensor array elements. For this dissertation, the array elements will be indexed in increasing order so that the top of the array is the first element and the bottom of the array is the last element. Furthermore, if it is assumed that there are N array elements, the first element will map to graph vertex $n = 0$ and the last element will correspond to graph vertex $n = N-1$ (Figure 3.2). If this order and mapping are followed, then we can

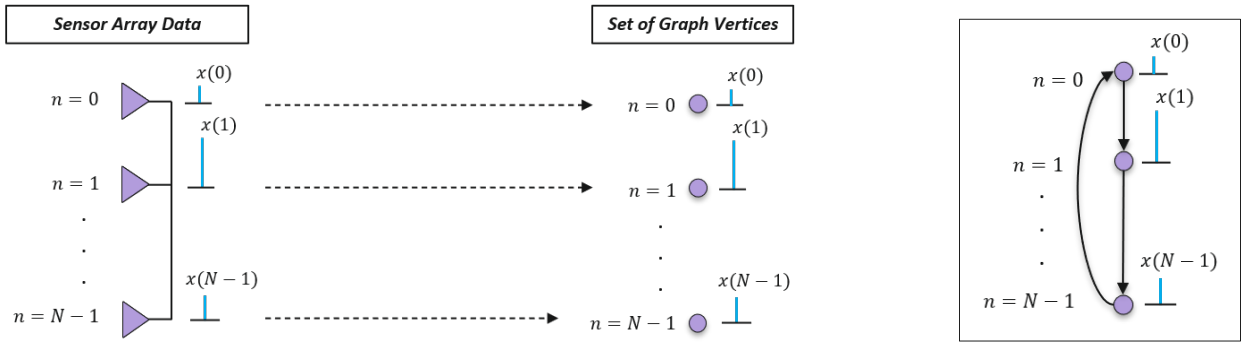


Figure 3.2: (a) One-to-one mapping between array element and graph vertex. (b) Example of a full graph signal with set of graph edges.

set graph signal vector $\mathbf{x}$ for a far-field plane wave equal to the equation already used in conventional signal processing.

This expression indicates an important step for broadening the model for the far-field plane wave because the equation *preserves* information. With this vector $\mathbf{x}$, the information from conventional signal processing is carried over to the graph domain by representing the measured data samples with the exact same equation. The only thing left to do to complete the generalization of this waveform is to define the second component of a graph signal: the graph signal matrix.

3.3 Expressing Graph Signal Matrix for Far-Field Plane Wave

To develop an expression for the graph signal matrix, the SPG literature was again reviewed first to identify a common framework. One challenge that emerged from this review was that, unlike the graph signal vector, SPG practitioners do not follow a single common framework for the graph signal matrix. In fact, there exists several forms of the matrix that are used to describe a graph. Some of these matrix forms include the unweighted and weighted adjacency matrix [28], the unnormalized and normalized Laplacian matrix [6], and the incidence matrix

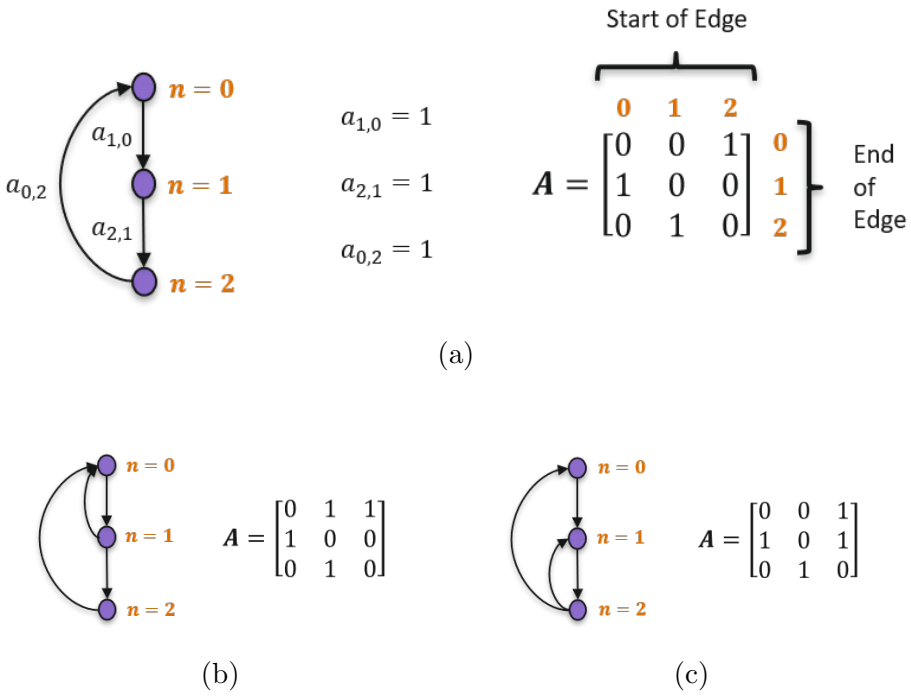


Figure 3.3: Examples of graph structures with their corresponding graph signal matrix **A**.

[24]. Because the Laplacian matrix is a function of the adjacency matrix and because the thrust of this dissertation is to build a solid foundation of work for single-snapshot DOA estimation, we chose to use the unweighted adjacency matrix.

The unweighted adjacency matrix follows a well-accepted structure in SPG. The matrix is a square matrix $\mathbf{A}$ with elements $a_{i,j} \in \mathbb{R}^N$ that specify which set of edges are used to construct the graph. To set these elements, the convention is to follow an existence criterion. This means that matrix element $a_{i,j} = 1$ when there exists a directed edge from vertex j to vertex i. Otherwise, $a_{i,j} = 0$. If matrix $\mathbf{A}$ is structured in this way, then the matrix should paint a clear picture of the structure for the dataset's underlying graph domain. Some examples of graph structures and their corresponding graph signal matrix $\mathbf{A}$ are shown in Figure 3.3.

Applying this framework to a far-field plane wave, we established a general expression

for the graph signal matrix that leaves room for further exploration, but still relates to the configuration of a ULA (Equation 3.2).

$$\mathbf{A} = \begin{bmatrix} 0 & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & 0 & \dots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1,0} & a_{N-1,1} & \dots & 0 \end{bmatrix} \text{ where } a_{i,j} = \begin{cases} 1, & \text{if edge } j \text{ to } i \text{ exists} \\ 0, & \text{if edge } j \text{ to } i \text{ does not exist} \end{cases} \tag{3.2}$$

The ULA contributed one key information to the development of this matrix expression: an N-element array established a matrix $\mathbf{A}$ with dimensions $N \times N$. Furthermore, we assumed no edge would start and end on the same vertex – a type of edge known as a self-loop that is commonly ignored among SPG practitioners [28] – which allowed us to set the main diagonal elements $a_{i,i}$ to 0. The remaining elements were left as variables to allow for studies later on which set of graph edges are needed for DOA estimation. This set of information and choices resulted in a way to develop a general expression for the graph signal matrix of a plane wave.

The combination of this expression matrix $\mathbf{A}$ and the expression that we formulated for vector $\mathbf{x}$ in the previous section gives the far-field plane wave a complete representation as a graph signal. More importantly, this complete representation demonstrates a generalization of the plane wave. This graph signal representation not only preserves, but also adds, new information. With matrix $\mathbf{A}$, the far-field plane wave is given an additional layer of information that describes the plane wave's underlying graph domain. This matrix further generalizes the plane wave by allowing the same vector of data to be represented on more than one domain (Figure 3.4). In SPG, the graph domain can be flexibly altered by simply inserting and deleting edges from the graph, which in turn modifies elements in graph signal matrix $\mathbf{A}$. Because our expression for matrix $\mathbf{A}$ has up to $N(N-1)$ elements that can be modified, and because the value of these elements is restricted to one of two values, a 0 or a 1, the matrix can be filled in up to $2^{N(N-1)}$ different ways. This means that a vector of data can have, depending on the choice of graph structure, a maximum of $2^{N(N-1)}$ representations versus the one representation offered in conventional signal processing. It will be interesting

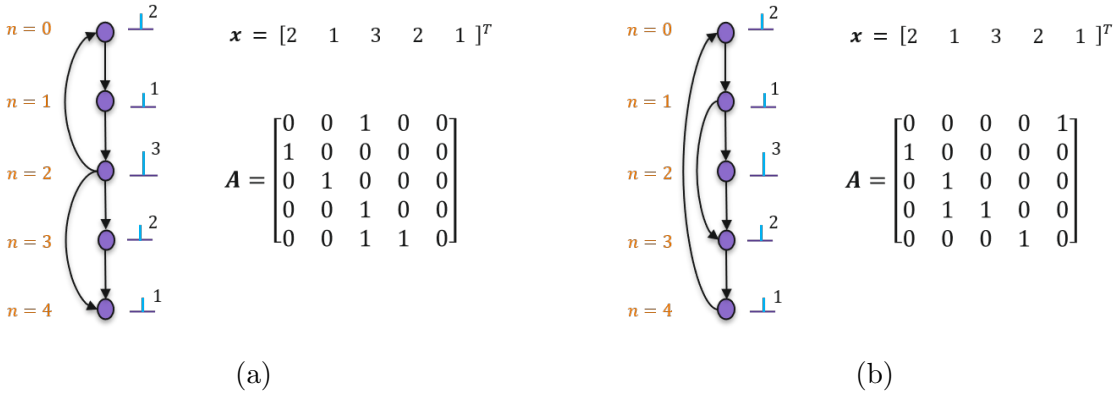


Figure 3.4: Graph signals with same measured data samples but different graph structures.

to see the impact of this generalization when we start developing and analyzing our actual SPG-based estimation system.

3.4 Implementing Model in Software

To confirm the graph signal representation that we developed in the last two sections for a far-field plane wave, we implemented and simulated the plane wave graph signals in software with Python and additional libraries. Using the Numpy library and its tools for creating vectors and matrices, we actualized the graph signal matrix (Equation 3.2) for a fixed graph structure and the graph signal vector (Equation 3.1) for a fixed set of parameters. The graph signal vector's noise component **w** for **x** was generated by using the definition of noise in rectangular form: $w(n) = w_R(n) + jw_I(n)$ in which $w_R(n)$ and $w_I(n)$ are the real and imaginary noise components, respectively. Both components were created by using independent normally distributed random variables with mean 0 and variance 1, e.g. $w_R(n), w_I(n) \sim \mathcal{N}(0, 1)$, and then scaling each component by a factor to achieve the desired SNR for the measured data. This scaling factor was determined by computing the noise variance σ_w^2, given the SNR, and then taking the square root of half the noise variance so that the total variance of complex

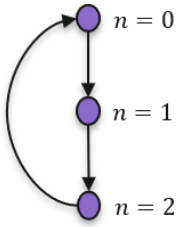
noise is σ_w^2 (Equation 3.3):

$$w(n) = \sqrt{\frac{\sigma_w^2}{2}} w_R(n) + j\sqrt{\frac{\sigma_w^2}{2}} w_I(n) \quad \text{where} \quad \sigma_w^2 = \frac{1}{10^{SNR/10}} \tag{3.3}$$

The numerator for noise variance is 1 because the source signal is assumed to be a unit-amplitude complex-exponential function. With this method for generating noise, we simulated a full graph signal and produced plots of the graph signal vector $\mathbf{x}$ as a function of graph vertex n with the aid of the Matplotlib library. Because the graph signal is complex-valued, plots of the graph signal are divided in two, one for the real part and another for the imaginary part of $\mathbf{x}$.

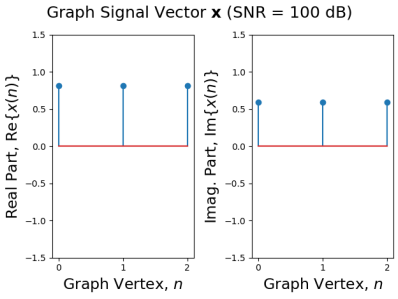
To demonstrate that our graph signal model behaves as expected for a plane wave, we simulated and plotted a collection of graph signals at different DOA angles and SNRs for two different graph structures, each one representing a specific ULA configuration. The first graph structure we simulated consisted of $N = 3$ vertices and represented a ULA with inter-element spacing $d = 0.9$ m, while the second graph structure had $N = 4$ vertices for a ULA with spacing $d = 1.55$ m. For each graph structure, a graph signal vector $\mathbf{x}$ was generated at a fixed SNR and a fixed DOA angle from the following set of values: $SNR = \{100, 10, 5\}$ dB and $\theta = \{0°, 50°\}$. The following values were also for the sound speed and source frequency to create a complete plane wave: $c = 1500$ m/sec and $f_o = 400$ Hz.

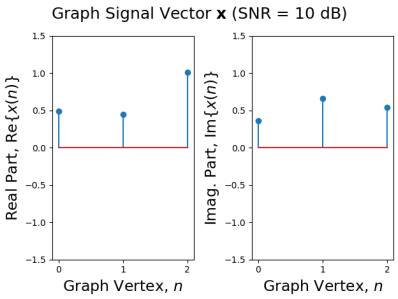
Our collection of graph signals (Figures 3.5 and 3.6) validate key characteristics of the plane wave. In Figure 3.5, the impact of the DOA angle is noticeable at high SNR. When $\theta = 0°$ and $SNR = 100$ dB, each graph vertex has the same value, which makes sense since the source is located directly in front of the array, an angle location referred to as broadside. When the DOA angle moves away from broadside $\theta = 50°$, there is, as expected, more variability in value across graph vertices. Also, in Figure 3.5, the impact of SNR is visible if we focus on a fixed DOA. At $\theta = 0°$, for example, as the SNR decreases, we start to see more variability across graph vertices due to the increasing variability in the noise. These observations from Figure 3.5 all carry over to Figure 3.6 in the case in which we have $N = 4$ graph vertices, further confirming our graph signal representation for a far-field plane wave.

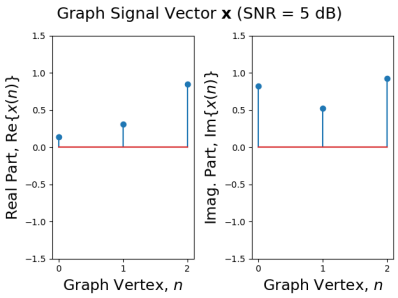



```
In [47]: A
Out[47]:
matrix([[0, 0, 1],
        [1, 0, 0],
        [0, 1, 0]])
```

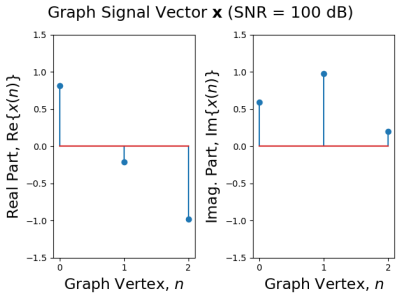
(a) Graph structure and adjacency matrix.

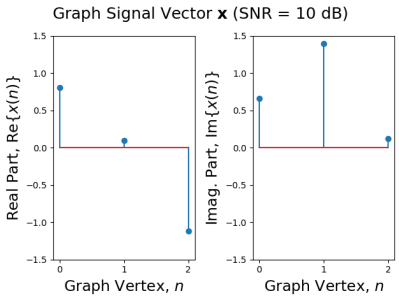


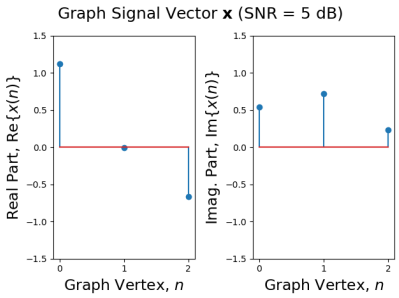




(b) DOA $\theta = 0°$

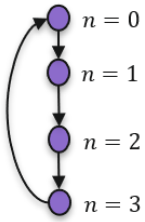






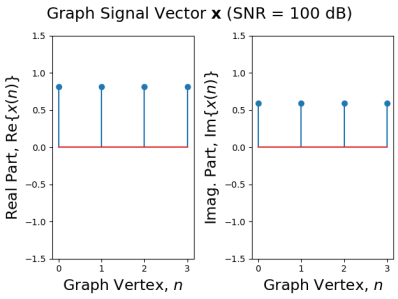
(c) DOA $\theta = 50°$

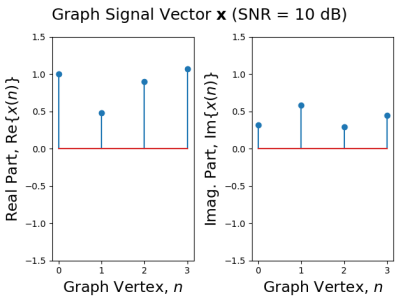
Figure 3.5: Simulated graph signals for $N = 3$ at different DOAs and SNRs.

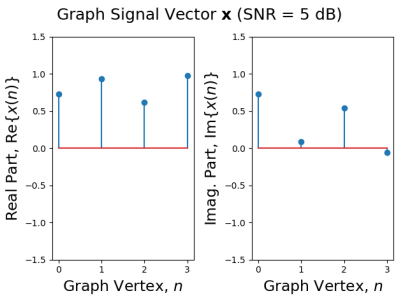



```
In [49]: A
Out[49]:
matrix([[0, 0, 0, 1],
        [1, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, 1, 0]])
```

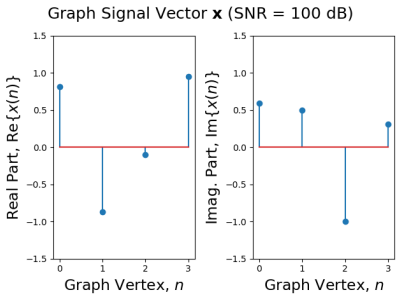
(a) Graph structure and adjacency matrix.

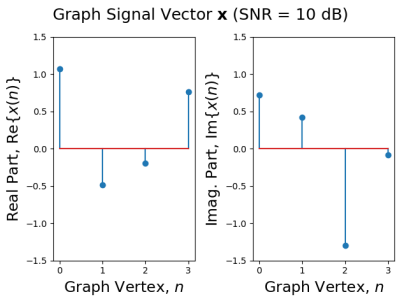


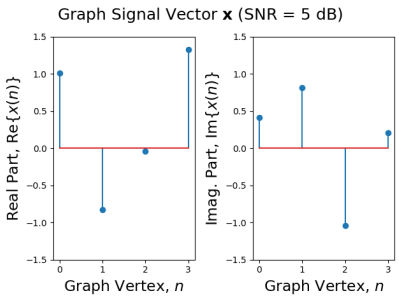




(b) DOA $\theta = 0°$







(c) DOA $\theta = 50°$

Figure 3.6: Simulated graph signals for $N = 4$ at different DOAs and SNRs.

3.5 Summary

In summary, the work in this chapter shows that if we apply the concept of a graph signal, then it is possible to reformulate and implement in software the far-field plane wave on a graph domain. This expression, made up of two components with vector $\mathbf{x}$ and matrix $\mathbf{A}$, demonstrates that not only can plane waves be represented on a graph, but plane waves can also be generalized to up to $2^{N(N-1)}$ representations, depending on the choice of graph structure. This increase in the number of representations ultimately helps with this dissertation's main task of finding performance improvement for single-snapshot DOA estimation. Furthermore, this generalized expression for a plane gives us a way to model the input signal for an estimation system, thereby overcoming the first hurdle in developing a graph-based single-snapshot DOA estimation system.

Chapter 4

DEVELOPING A NEW DOA ESTIMATION SYSTEM

In the last chapter, we laid the ground work for the development of an improved single-snapshot DOA estimation system by formulating a graph signal model for the system's input data. The focus for this chapter will now be on the actual system that will process this signal and produce an estimate of DOA (Figure 4.1) with the aid of SPG. The advantage of SPG is that it can *generalize* – that is, it can take something we know, like an existing estimation system, and broaden it by incorporating new information into the system processing chain. Generalizing an existing DOA estimation system with SPG has been studied in a few recent papers [24–26], but there is one algorithm for DOA estimation, described in Chapter 2, that has yet to be explored: the conventional beamformer. The conventional beamformer is important to examine because it is widely used in practice [29–31] and referenced regularly in the literature as a baseline system [32, 33]. To continue building a solid foundation of work and make progress on using SPG to improve estimation performance, the conventional beamformer needs to be studied. Our conjecture for this chapter is that, **using SPG and the conventional beamformer as inspiration, a new single-snapshot DOA estimation system can be developed that works with inputs defined on a graph domain.**

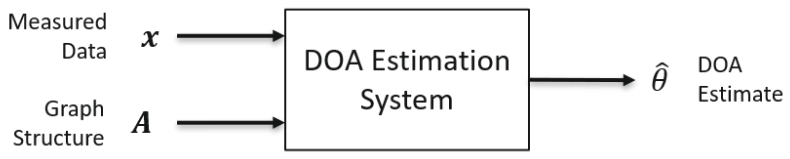


Figure 4.1: General DOA estimation system diagram using input graph signal.

4.1 Building and Distinguishing From Prior Work

The work in this chapter builds on prior work and also charts a new path to using SPG for DOA estimation system development. As in previous work, figuring out how to incorporate a graph structure into the system process remains a priority. However, rather than using an incidence matrix [24] or a weighted adjacency matrix [25], we build on the work from the last chapter and use an unweighted adjacency matrix to help simplify and narrow the focus of our task. Using an unweighted adjacency matrix allows the focus to shift from the existence and strength of graph edges to just the existence of edges. To add to this approach, we also distinguish from some of our work in which we attempted to use SPG tools to develop new DOA estimation methods [22, 23]. The approach now is to leverage the idea of starting with an existing DOA estimation algorithm and then generalizing it with the help of current SPG principles and tools. However, rather than focusing on the differential beamformer [25] or the MUSIC algorithm [26], our attention in this chapter is on the conventional beamformer.

To develop our new DOA estimation system from the conventional beamformer, the algorithm for the DFT-based conventional beamformer, first discussed in [3], was used to guide the work. We started by writing the algorithm as a single formula (Equation 4.1) that

$\mathbf{x}$	Measured data
$\mathbf{f}_\ell$	Linear Fourier weights
c	Sound speed
N	Number of array elements
f_o	Source frequency
d	Inter-element spacing
$\hat{\ell}$	Localized frequency bin
$\hat{\theta}$	Estimated DOA angle

Table 4.1: Summary of variables for the conventional beamformer.

incorporates variables we have already introduces (Table 4.1).

$$\hat{\theta} = \sin^{-1}\left(\frac{\hat{\ell}c}{Nf_o d}\right) \text{ where } \hat{\ell} = \arg\max_{\ell} \frac{\mathbf{f}_\ell \mathbf{x}\mathbf{x}^H \mathbf{f}_\ell^H}{\mathbf{f}_\ell \mathbf{f}_\ell^H} \tag{4.1}$$

To guide our work further, we broke the algorithm and formula down to a sequence of processes. Once these processes were identified, we translated each one to the graph domain by using existing principles and tools from SPG. Once we produced a full system diagram, we then validated that our system can indeed process input graph signals. To perform this validation, we extended the Python software we developed for plane wave signals in the last chapter and included an implementation of our actual system. The following sections are used to present and discuss our results.

4.2 Using SPG to Translate Conventional Beamformer

From the algorithm for the conventional beamformer (Equation 4.1), three distinct processes were identified (Figure 4.2). The first process was the computation of the input data's power spectrum P_ℓ from the linear weights of a DFT. The second process was the localization of maximum power to a frequency bin $\hat{\ell}$. The third and final process was the conversion of the localized frequency bin to an angle $\hat{\theta}$, the conventional beamformer's estimate of DOA. The identification of these processes is important because it divided our work into three clear tasks. The upcoming subsections detail how SPG was used to translate each process.

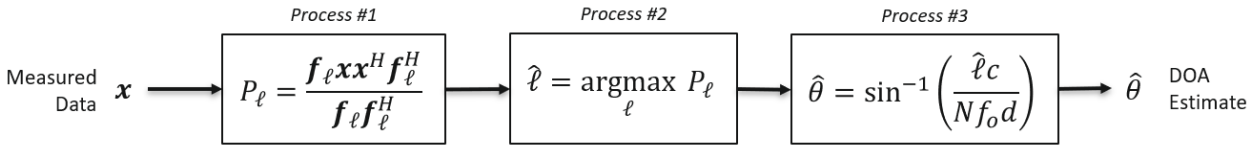


Figure 4.2: System diagram for conventional beamformer.

4.2.1 Process #1

Of the three processes, SPG was used the most to translate process #1, which at its core relies on the DFT. The DFT provides two things that are needed to compute the power spectrum. It provides the linear Fourier weights $\mathbf{f}_\ell$ and the transformation equation to convert the measured data to the frequency domain by way of a projection operation $\mathbf{f}_\ell\mathbf{x}$. The power spectrum is then computed by magnitude-squaring the output of this projection and normalizing it by the linear weights of the DFT. The DFT is indeed central to this process, so updating this whole process to the graph domain, therefore, requires finding and using SPG's equivalent to a DFT.

In SPG, the equivalent to the DFT is the Graph Fourier Transform (GFT) [34]. The GFT is, simply put, an extension of the DFT and provides the same two items that process #1 requires: linear Fourier weights and a transformation equation. The linear Fourier weights for a GFT are found by extracting them from the input signal's graph structure $\mathbf{A}$ with the aid of a procedure called an eigen-decomposition. In an eigen-decomposition, two new matrices, often denoted as $\mathbf{V}$ and $\mathbf{\Lambda}$, are extracted by factorizing matrix $\mathbf{A}$ into three terms $\mathbf{V\Lambda V}^{-1}$. The matrix $\mathbf{V}$ is particularly important because it is inverted to get the transformation matrix $\mathbf{F}$. From this matrix $\mathbf{F}$, the linear Fourier weights at frequency bin ℓ are then obtained by extracting the elements from the ℓth row of $\mathbf{F}$ This entire extraction procedure is summarized in Equation 4.2.

$$\mathbf{A} = \mathbf{V\Lambda V}^{-1} \;\longrightarrow\; \mathbf{F} = \mathbf{V}^{-1} \;\longrightarrow\; \mathbf{f}_\ell = [\mathbf{F}]_\ell \tag{4.2}$$

In addition to linear Fourier weights, the GFT also provides the equation to transform measured data to the frequency domain at frequency bin ℓ (Equation 4.3).

$$\hat{x}_\ell = \mathbf{f}_\ell\mathbf{x} \tag{4.3}$$

This equation is, in fact, equivalent to the DFT but uses a set of weights extracted from the input graph structure rather than a set of weights calculated from a fixed formula.

Using this knowledge of the GFT, process #1 was updated so that it now operates on a full input graph signal (Figure 4.3). Two modifications were required overall. First, the

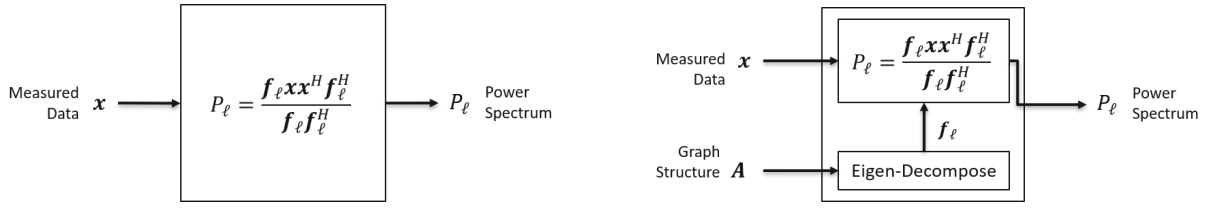


Figure 4.3: Translating the conventional beamformer's first process from (a) conventional signal processing to (b) SPG.

power spectrum block was carried over from the conventional beamformer. Because the power spectrum is calculated from the same transformation formula used in both the DFT and the GFT, it required no change from conventional signal processing. The second change made to process #1 was the inclusion of the eigen-decomposition process to extract, from the input graph structure $\mathbf{A}$, the linear Fourier weights $\mathbf{f}_\ell$ needed to perform a GFT and calculate the power spectrum. This modification was important to both incorporate the input graph structure into the processing chain and handle the fact that the linear weights vary with graph structure. Using all these updates, based on principles and tools from SPG, process #1 was reformulated so that it now worked on the graph domain.

4.2.2 Handling Sorting Issue With Linear Fourier Weights

The one issue that had to be handled with the eigen-decomposition process was that the extracted linear Fourier weights $\mathbf{f}_\ell$ were not yet properly sorted in ascending frequency order. Proper sorting will help make our estimation system's behavior more interpretable, so the process we had now needed to be reconsidered and expanded. To properly sort the linear weights, a procedure was developed (Figure 4.4) that leverages two quantities that are calculated from the eigen-decomposition of input graph adjacency matrix $\mathbf{A}$. These two quantities are the eigenvectors matrix $\mathbf{V}$ and the eigenvalues matrix $\mathbf{\Lambda}$. The process that

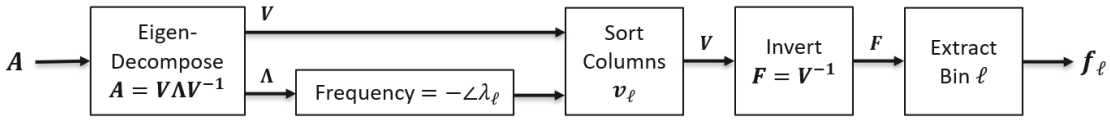


Figure 4.4: Expanded eigen-decomposition process to sort linear Fourier weights in ascending frequency order.

follows the eigen-decomposition takes advantage of two key properties:

1. *Relationship Between* $\mathbf{f}_\ell$ *and* $\mathbf{V}$: The linear Fourier weights $\mathbf{f}_\ell$ are extracted from the ℓth row of transformation matrix $\mathbf{F}$, and this matrix $\mathbf{F}$ is related to matrix $\mathbf{V}$ according to $\mathbf{F} = \mathbf{V}^{-1}$. As a result, the way to sort $\mathbf{f}_\ell$ is by sorting the column vectors in matrix $\mathbf{V}$, often referred to as eigenvectors $\mathbf{v}_\ell$ [35].

2. *Relationship Between* $\mathbf{V}$ *and* $\mathbf{\Lambda}$: Every diagonal element λ_ℓ of matrix $\mathbf{\Lambda}$ is tied to a corresponding column vector $\mathbf{v}_\ell$. In addition, these elements, often referred to as the eigenvalues, can provide a measurement of frequency [35, 36]. Therefore, a way to sort the eigenvectors $\mathbf{v}_\ell$ in ascending frequency order is to use these eigenvalues.

To obtain measurements of frequency from the eigenvalues λ_ℓ, we can use one of the methods that has been discussed in the SPG literature. SPG practitioner sort the eigenvalues and eigenvectors by either using the magnitude of the eigenvalues [35] or by using the phase of the eigenvalues [36]. For our work, the phase is the most appropriate to use because the eigenvalues we will be using are complex-valued. The reason for this is because, according to spectral graph theory, the corresponding adjacency matrix of graph structures with directed edges are not guaranteed to be symmetric. This means the eigenvalues of the adjacency matrix are in general complex-valued [37]. By taking the phase of these eigenvalues, and multiplying it by negative one, we can get a measurement of frequency, in radians, at index

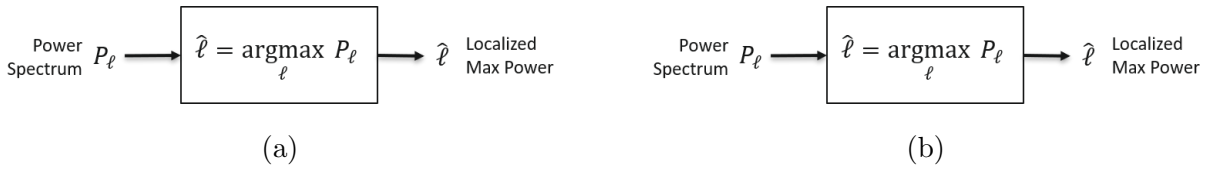


Figure 4.5: Translating the conventional beamformer's second process from (a) conventional signal processing to (b) SPG.

ℓ (Equation 4.4).

$$\text{Amount of Frequency} = -\angle\lambda_\ell \tag{4.4}$$

Using these measurements of frequency from every eigenvalue, we can then sort λ_ℓ and the corresponding eigenvectors $\mathbf{v}_\ell$ in ascending frequency order starting from $-\pi$ all the way to up π. This produces the sorted eigenvectors matrix $\mathbf{V}$ that we desire, and in turn, linear Fourier weights that are properly sorted in frequency as well.

4.2.3 Process #2

Compared to process #1, SPG only had a minor role to play in translating process #2. At the core of process #2 is the concept of a frequency bin ℓ, which the process needs to search over in order to identify peak power. From SPG, we leveraged the fact the concept of a frequency bin extends and is retained in the graph domain through SPG's definition of a GFT. Process #2, as a result, required no modification at all and used the same system block representation from conventional signal processing to localize maximum power (Figure 4.5).

4.2.4 Process #3

The impact of using the SPG framework for translation is felt the most with translating this third and final process in the conventional beamformer to convert a frequency bin to an angle. In the conventional beamformer, this conversion is done by using a fixed equation

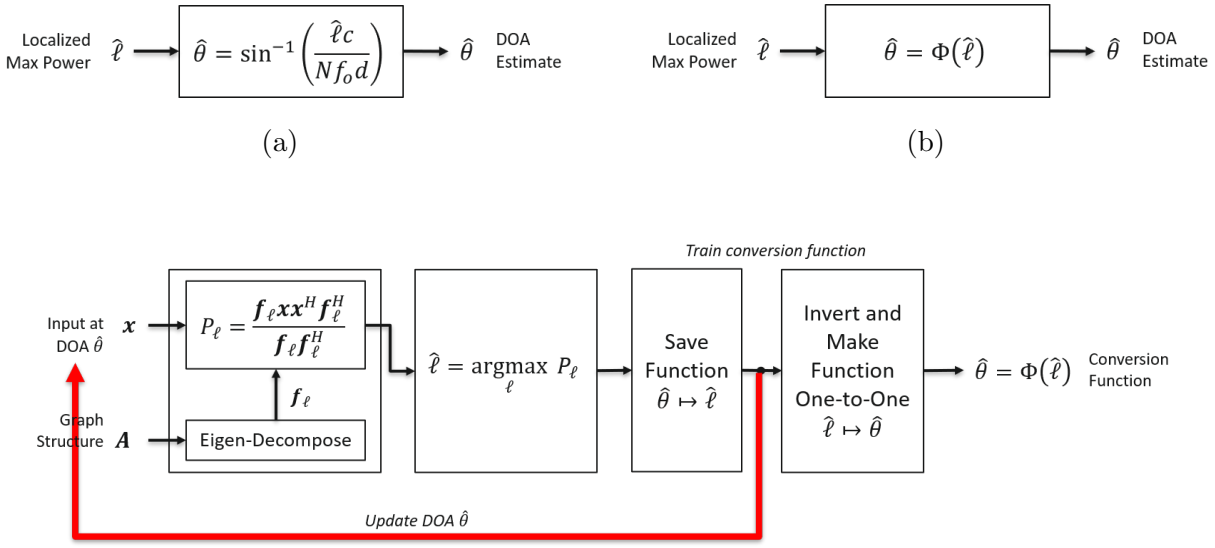
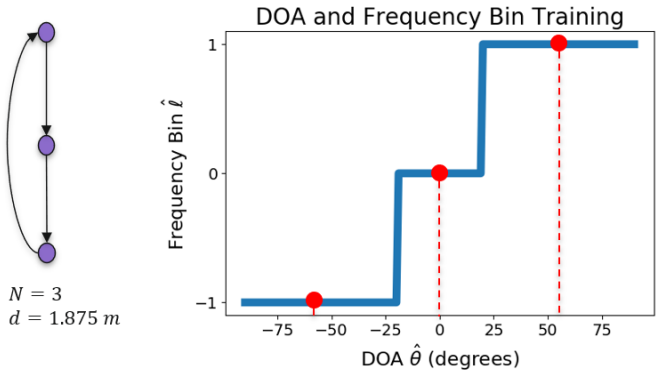


Figure 4.6: Translating the conventional beamformer's third process from (a) conventional signal processing to (b) SPG with a training system in (c) to obtain conversion function.

(Figure 4.6a). This conversion equation is fixed because it is derived from knowledge of the DFT linear weights, which come from a fixed expression as well. As a result of updating process #1 and process #2 to the graph domain, the conventional beamformer's conversion equation can no longer be supported in the graph domain. The linear weights in the graph domain are now derived from a numerical process and a variable input graph structure, so therefore what is needed in place of a fixed equation for process #3 is a conversion function. This conversion function $\hat{\theta} = \Phi(\hat{\ell})$ would adapt to every graph structure $\mathbf{A}$ and can reliable be used to map every frequency bin $\hat{\ell}$ to a DOA angle $\hat{\theta}$ (Figure 4.6b).

To obtain this conversion function, given we know the input graph structure $\mathbf{A}$, a training system was developed (Figure 4.6c). The system is provided values of $\hat{\theta}$ and $\hat{\ell}$ to train the conversion function by inputting a set of noiseless plane waves, simulated at different DOA angles $\hat{\theta}$ and modeled by the formulation we developed in the last chapter, and then

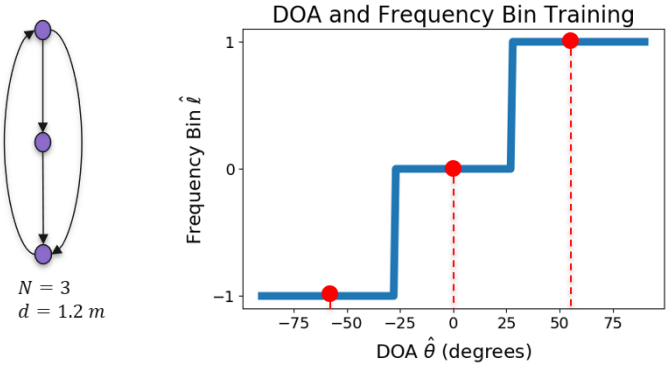
processing them to produce the different frequency bins $\hat{\ell}$. These frequency bins are produced by utilizing the new system blocks we developed for process #1 and process #2 in the last two sections. By running simulated plane waves, with angles from $-90°$ to $90°$, through process #1 and process #2, the first part of our training process outputs a function mapping angle to bin, $\hat{\theta} \mapsto \hat{\ell}$. This function is then inverted to get a function mapping bin to angle, $\hat{\ell} \mapsto \hat{\theta}$. To ensure this function is one-to-one, we performed an averaging procedure in which, for every frequency bin $\hat{\ell}$, the bin is mapped to a single DOA angle $\hat{\theta}$ that we get by averaging all the DOA angles that had mapped to that bin and then rounding that value to the nearest integer. The result of this averaging procedure creates the final desired conversion function $\hat{\theta} = \Phi(\hat{\ell})$. Examples of this functional mapping and inversion process to obtain a conversion function are shown for two different graph structures in Figure 4.7.



$\hat{\theta} = \Phi(\hat{\ell})$

$\hat{\ell}$	$\hat{\theta}$
-1	-55°
0	0°
1	55°

(a)



$\hat{\theta} = \Phi(\hat{\ell})$

$\hat{\ell}$	$\hat{\theta}$
-1	-59°
0	0°
1	59°

(b)

Figure 4.7: Examples of how training system is used to obtain conversion function when $f_o = 400$ Hz and $c = 1500$ m/sec.

This training system we developed to obtain this conversion function was enhanced further by resolving these two special cases:

1. The first case is when N is even, in which the averaging process does not work due to angle ambiguity. In this case (Figure 4.8a), the frequency bin $\hat{\ell} = -N/2$ has ambiguity because it maps to both positive and negative non-zero DOA angles. This ambiguity can be explained by understanding the power spectrum calculation a bit more. Because the numerator of the power spectrum calculation is the quadratic form of a magnitude-square operation, we can expand the equation to a product of two summations:

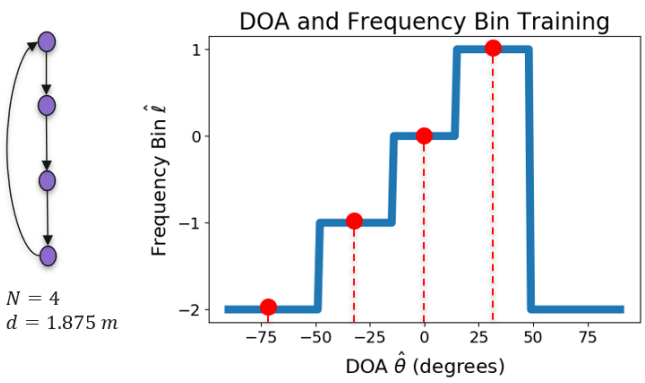
$$\begin{aligned}
\mathbf{f}_\ell \mathbf{x}\mathbf{x}^H \mathbf{f}_\ell &= |\mathbf{f}_\ell \mathbf{x}|^2 = (\mathbf{f}_\ell \mathbf{x})(\mathbf{f}_\ell \mathbf{x})^* \\
&= \left(\sum_n s(t_o) e^{j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell(n) \right) \left(\sum_n s^*(t_o) e^{-j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell^*(n) \right) \\
&= s(t_o)s^*(t_o) \left(\sum_n e^{j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell(n) \right) \left(\sum_n e^{-j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell^*(n) \right) \\
\mathbf{f}_\ell \mathbf{x}\mathbf{x}^H \mathbf{f}_\ell &= \left(\sum_n e^{j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell(n) \right) \left(\sum_n e^{-j2\pi f_o \frac{d\sin(\theta)}{c} n} f_\ell^*(n) \right)
\end{aligned}$$

   From this expression, we can infer that if the linear Fourier weights at bin ℓ are real-valued with $f_\ell(n) = f_\ell^*(n)$, then power spectrum at bin ℓ will be equal for inputs steered to DOA $\theta = \theta_o$ and $\theta = -\theta_o$ since $\sin(-\theta_o) = -\sin(\theta_o)$. It just so happens that the DFT linear weights are indeed real-valued at bin $\ell = -N/2$ with $\mathbf{f}_\ell = [(-1)^n]_{n=0}^{N-1}$. Since the GFT is an extension of the DFT, it can be shown in simulation that the GFT linear weights are also real-valued at bin $\ell = -N/2$ hence the ambiguity in angle at that bin. The impact of this ambiguity is that averaging all angles that map to this particular frequency bin would result in a value of zero, an angle which already maps to bin $\hat{\ell} = 0$. We resolved this case by ignoring angles that only using the negative angles to do the averaging and hence mapping $\hat{\ell} = -N/2$ to a negative angle.

2. In the second case, some array configurations result in a situation in which the maximum power is not localized to all the frequency bins (Figure 4.8b). This means that,

for some bins $\hat{\ell}$, there are no angles to average and map to. This case occurs because the inter-element spacing d is small. Since the calculation of the power spectrum relies on computing a Fourier Transform of values from an array across space, the parameter d determines the overall window of the Fourier Transform. A smaller d results in a smaller window in space, so hence not all frequency bins will be used. To resolve this case, we mapped these frequency bins to the angle of its nearest neighbor. So for example, if frequency bin $\hat{\ell} = -2$ is one of the bins that has no angle to map to, we used the angle that was mapped to its nearest neighbor $\hat{\ell} = -1$. This resolution does extend the domain to which some DOA angles are mapped to, but we found this to be acceptable since this enhancement creates a conversion function that is both one-to-one and fully defined such that every frequency bin is now mapped to a DOA angle.

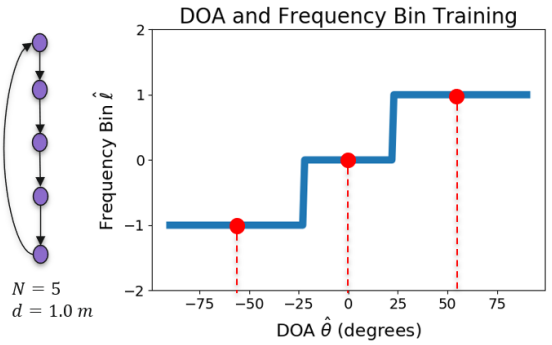
By dealing with these two special cases, our training system becomes more robust in producing effective conversion functions. This increase in effectiveness furthermore makes process #3 more reliable overall and be able to estimate DOA angles from a frequency bin in the graph domain.



$\hat{\theta} = \Phi(\hat{\ell})$

$\hat{\ell}$	$\hat{\theta}$
-2	-69°
-1	-31°
0	0°
1	31°

(a)



$\hat{\ell}$	$\hat{\theta}$
-2	???
-1	-56°
0	0°
1	56°
2	???

$\hat{\theta} = \Phi(\hat{\ell})$

$\hat{\ell}$	$\hat{\theta}$
-2	-56°
-1	-56°
0	0°
1	56°
2	56°

(b)

Figure 4.8: Two special cases handled by training system when $f_o = 400$ Hz and $c = 1500$ m/sec: (a) angle ambiguity when $N =$ even, and (b) frequency bins not all utilized.

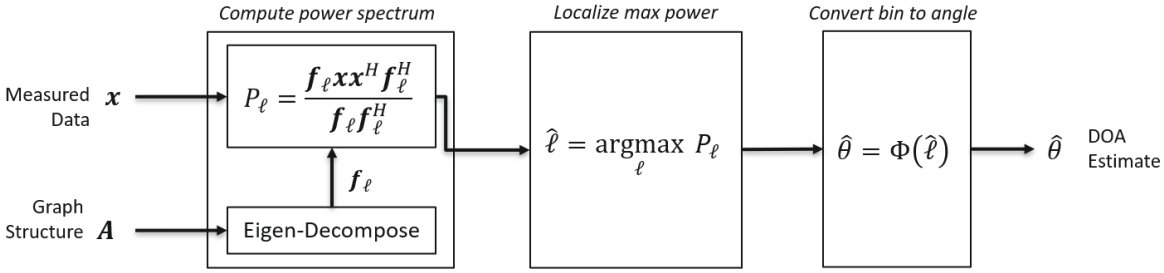
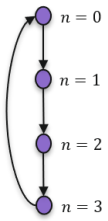


Figure 4.9: System diagram of our new single-snapshot DOA estimation.

4.2.5 Full System Diagram

When all the translated processes are connected together into one, we get the first key outcome of this chapter: the creation of a new single-snapshot DOA estimation system that is fully defined and designed to operation on a graph domain. This system diagram (Figure 4.9) shows a possession of three key differences that distinguish it from its conventional beamformer counterpart: the inclusion of a graph signal at the input, the processing of the graph structure input to produce new linear Fourier weights, and the incorporation of a function to convert frequency bin to angle that will adapt to the input graph structure. By showing how the input graph structure is incorporated and processed by the estimation system, this also demonstrates the system's ability to still produce an estimate of DOA regardless of how the input data's underlying graph domain is structured.

A closer look at this system diagram also reveals and confirms SPG's ability to generalize. The estimation system can indeed by interpreted as a generalized conventional beamformer because, for one, it preserves system behavior from the conventional beamformer. If we incorporate past research from SPG on its connections to conventional signal processing [7], one special graph structure exists and is worthy to study here: the directed cycle graph (Figure 4.10). The directed cycle graph is a graph in which all graph edges are oriented in the same direction and forms a complete cycle, and its adjacency matrix can be characterized as a cyclic shift matrix in which there exists only a single entry of one in each row/column and



$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 4.10: Example of a directed cycle graph with $N = 4$ vertices.

each row/column is nothing more than a shift by one element from the previous row/column. This graph structure is important to mention here because the eigen-decomposition of a directed cycle graph produces exactly the linear Fourier weights from a DFT [38]. Because of this property, the power spectrum, the localized maximum power, and the estimated DOA from our system would be identical to the estimate produced by the conventional beamformer. Therefore, if data is modeled and processed on a directed cycle graph, the conventional beamformer is preserved with our estimation system.

Interpreting our estimation system as a generalized conventional beamformer is further demonstrated by the fact that our system not only preserves system behavior, but also adds new system behaviors. It is able to do that by allowing flexible graph structures as inputs to the system processing chain. Simply by adding an extra edge to the directed cycle graph, the system behaves in a different way. The linear weights would change and so would the conversion function, resulting in a potentially different output estimate for DOA. Therefore, with the translations we made with the aid of SPG, we developed an estimation system that replicates a conventional beamformer and also extends the capability of a conventional beamformer to process a wider class of input signals.

4.3 Implementing New DOA Estimation System in Software

Our conjecture for this chapter – that a working graph-based DOA estimation can be developed – is further supported by a second key outcome: a software implementation of our new single-snpashot DOA estimation system. To implement our estimation system in software,

we took our Python script from the last chapter that implemented plane waves and extended it to include all of the system processing blocks that we developed in the last section.

4.3.1 Implementing Training System

The first part of our estimation system required an implementation of the training system (Figure 4.6c) to obtain the function the estimation system needs to convert frequency bin to angle, a task that added four new sections of code to our Python script. These sections include: the eigen-decomposition process to extract the linear Fourier weights with frequency bin sorting, the computation of the power spectrum, the localization of maximum power, the formation of the angle to bin function mapping, and finally the inversion of this function mapping to produce the final conversion function. All blocks in the training system were implemented as they were outlined and written, without any major changes. As a result, a full training system was created that could provide us with the conversion function required to estimate DOA.

4.3.2 Implementing Estimation System with Input Data

The actual DOA estimation system (Figure 4.9) was implemented by taking advantage of code already written for the training system. Since the estimation system's first two processing blocks were identical to those in the training system, code that was developed in the last section was reused. To implement the third and final processing block, the work from the training system implementation was again leveraged by incorporating the conversion function that was created from training so that the system can use it to estimate DOA. Inserting this conversion function into the software gave us the last piece that was needed to create a fully operational DOA estimation system that could now be tested.

To test our system, simulation experiments were conducted by focusing on plane waves as inputs. These inputs were generated by using the last chapter's graph signal model for the far-field plane wave. To generate the measured data $\mathbf{x}$, we fixed three of the model parameters and varied the remaining three others. We used, as fixed parameters, source

frequency $f_o = 400$ Hz, sound speed $c = 1500$ m/sec, and inter-element spacing $d = 1.875$ m. As variable parameters, we generated inputs representing two different array sizes ($N = 3$ and $N = 32$), two different amounts of noise (SNR = 20 dB and SNR = -5 dB), and three different DOA angles (selected from values in the system's conversion function). To generate the graph structure $\mathbf{A}$, we used two different graphs for every array configuration of size N. The first graph, for each array configuration, was the directed cycle graph, and the second graph was the directed cycle graph with one additional graph edge. For each of these input scenarios, 100 snapshots or realizations of data were created to input into our system and produce an estimate of DOA.

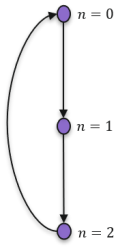
We summarized the results of this simulation experiment with a plot. We plotted the estimated DOA angles $\hat{\theta}$ versus each data snapshot index for every graph structure, array size, SNR, and DOA angle we studied. We grouped the plots according to graph structure, resulting in a total of four graph structures that were studied with this simulation experiment.

4.3.3 Results

Our plots of DOA estimate versus time snapshot (Figures 4.11-4.14 visually confirm that the new DOA estimation system we developed does indeed work. These results, first and foremost, confirm that our system can estimate DOA from input data modeled and processed on a graph domain, regardless of which DOA angle or SNR was used. This is apparent from the first collection of plots from the 3-element ULA we tested using a directed cycle graph (Figure 4.11). Our estimation system was able to produce correct and incorrect DOA estimates, depending on the noise level. When SNR was high at 20 dB (Figure 4.11b), the DOA estimates were, as expected, correct for every snapshot of data that was processed and for all three DOA angles that were simulated. When SNR was lowered to -5 dB (Figure 4.11c), there was an increase in incorrect estimates, as to be expected with an increasing amount of noise.

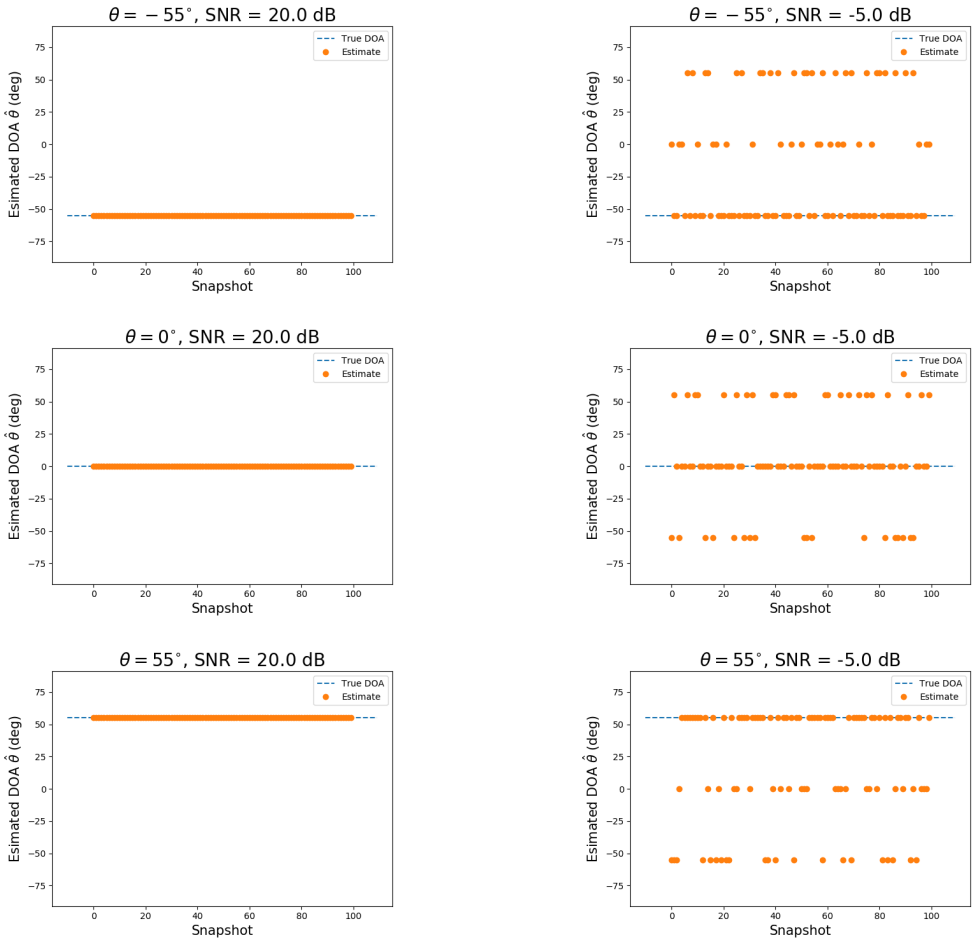
The operation of our system is further confirmed with the other three collection of results. Our results show that our system can not only process data from a directed cycle graph,

but it can also process data from the same array configuration but with a different graph structure. When the 3-element ULA was processed with a graph structure that included an additional graph edge (Figure 4.12), we see the same pattern in the output estimates as we did for the directed cycle graph. As expected, an increase in the amount of noise again led to an increase in the number of incorrect estimates. These expectations and results, however, were not limited to 3-element arrays and carry over to array configurations with more elements. We can see that with our results from a 32-element that were processed with a directed cycle graph (Figure 4.13) and a graph structure with one added edge (Figure 4.14).



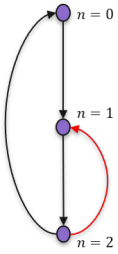
$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(a)



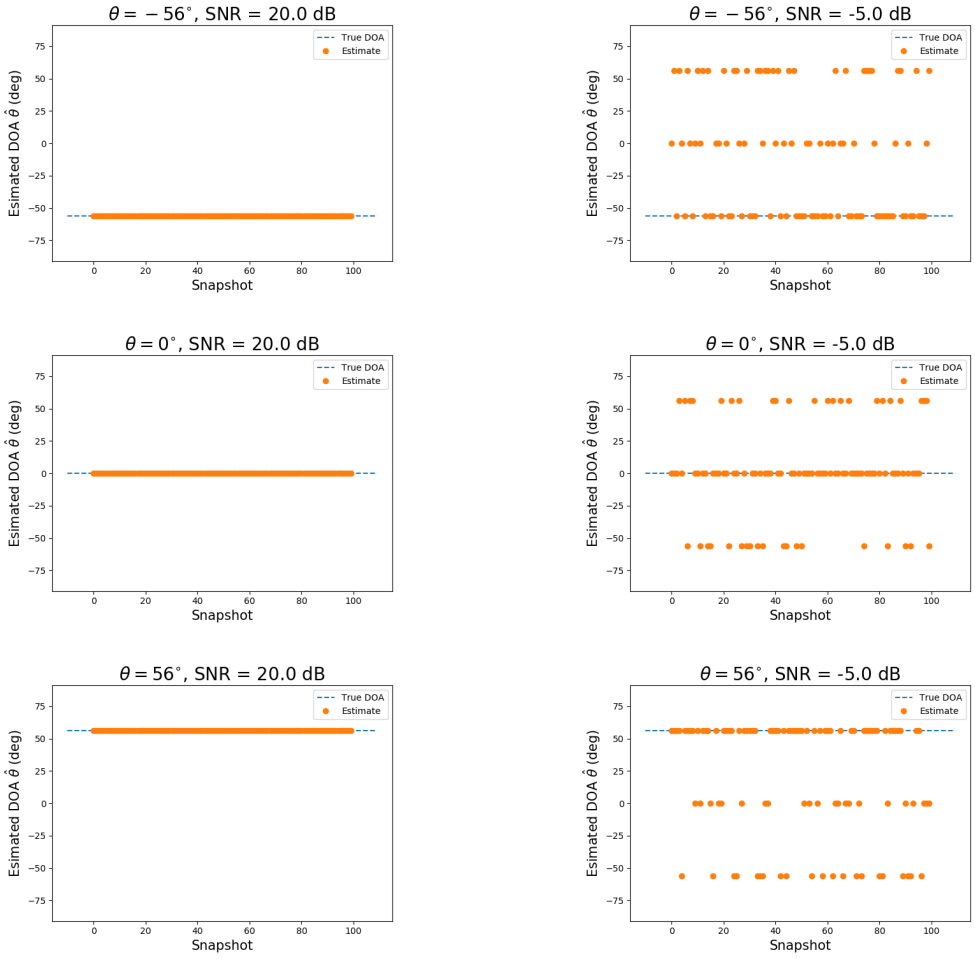
(b) (c)

Figure 4.11: DOA estimation outputs for 100 snapshots of data using graph structure shown in (a) and for the case of (b) low noise and (c) high noise.



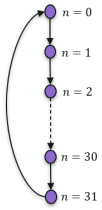
$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

(a)



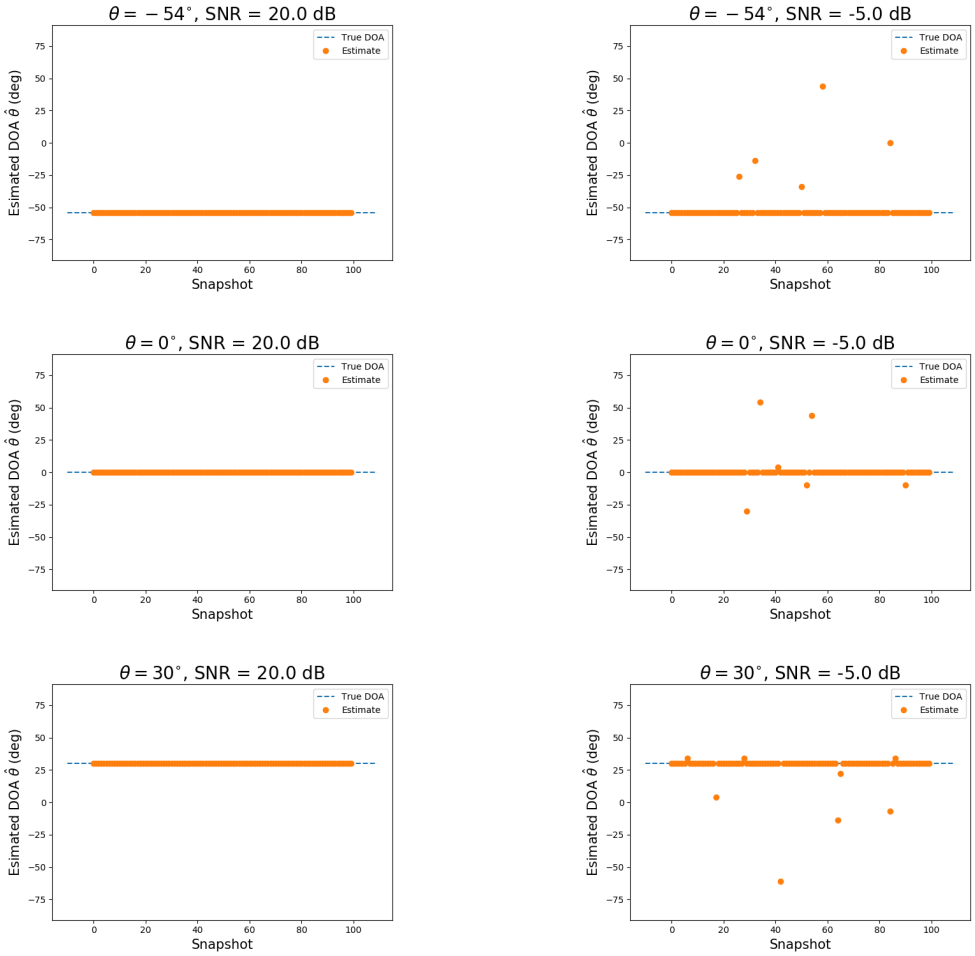
(b) (c)

Figure 4.12: DOA estimation outputs for 100 snapshots of data using graph structure with one added edge shown in (a) and for the case of (b) low noise and (c) high noise.



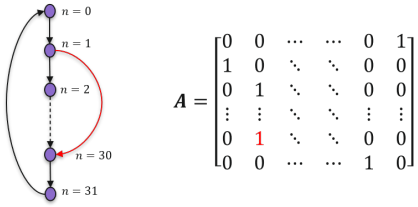
$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & 1 \\ 1 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 1 & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & \cdots & 1 & 0 \end{bmatrix}$$

(a)

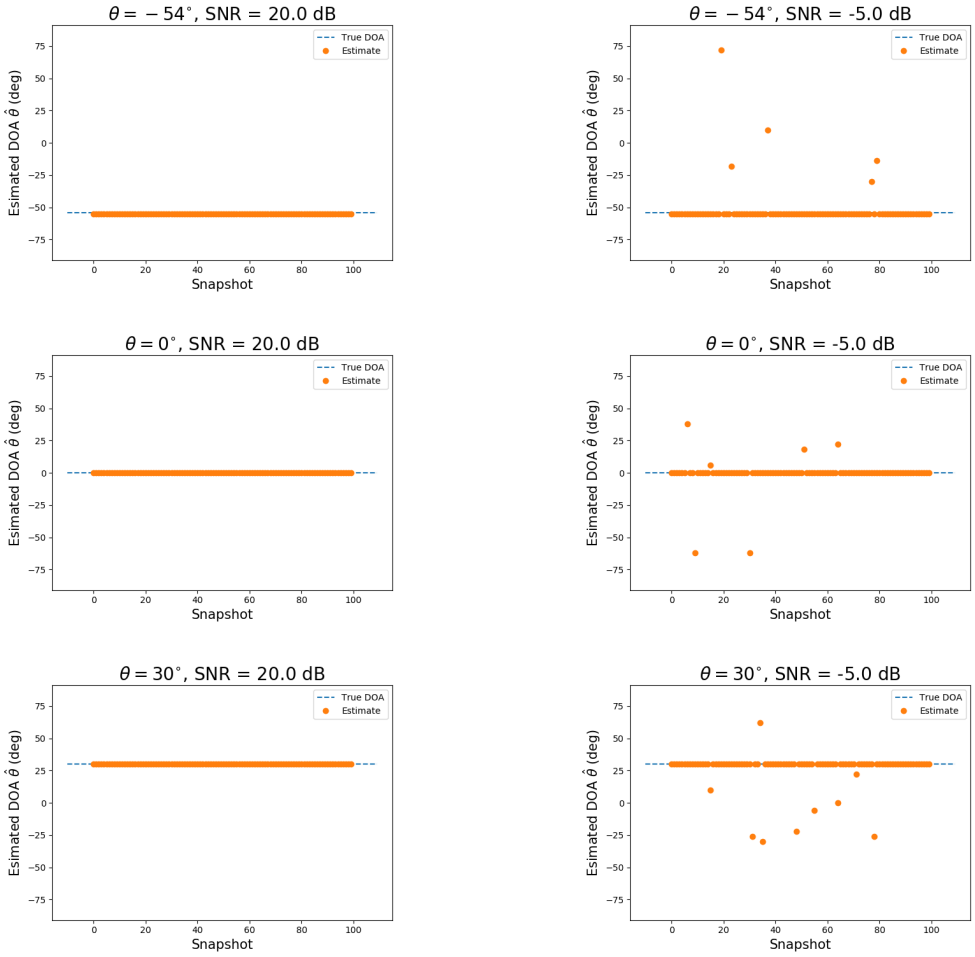


(b) (c)

Figure 4.13: DOA estimation outputs for 100 snapshots of data and a higher-order array using graph structure shown in (a) and for the case of (b) low noise and (c) high noise.



(a)



(b) (c)

Figure 4.14: DOA estimation outputs for 100 snapshots of data and a higher-order array using graph structure with one added edge shown in (a) and for the case of (b) low noise and (c) high noise.

4.4 Summary

In summary, our system diagram and software implementation show that the SPG framework can indeed be used for single-snapshot DOA estimation system development. With SPG, we designed a new estimation system that generalizes the conventional beamformer and makes it far more flexible than its original representation in conventional signal processing. Our results from software, in particular, demonstrate that this new system can work on a wider class of input signals. The work we have done in this chapter inches us another step closer to finding ways to improve performance in single-snapshot DOA estimation system. We now have a system that we can investigate and use as a platform to study performance improvement by taking advantage of SPG's ability to model and process on data on the more flexible graph domain.

Chapter 5

EVALUATING PERFORMANCE WITH REAL DATA

From the formulation of an input model to the design of a graph-based estimation system, the focus so far has been on system development. The last chapter ended with demonstrations showing that our system *does work*, but knowledge on *how well it works* remains unclear, especially when compared to the system that inspired its development, the conventional beamformer (CBF). To continue to argue that SPG is viable for single-snapshot DOA estimation, the attention needs to shift from *system development* to *system performance evaluation*. Performance evaluation in DOA estimation often means measuring estimation accuracy as a function of some input signal parameter with real or simulated data [39] (Figure 5.1). When it comes to number of input parameters that can be investigated, our estimation system has the advantage over the CBF of having additional parameters from the system's second input, a graph structure **A**. Another advantage with our system is that it can be used to replicate the CBF. As shown in the last chapter, the CBF can be replicated as long as a directed cycle graph is used as the input graph structure. If a directed cycle graph causes our system to behave like the CBF, this begs the question: **what if we try other graph structures? Will our system perform better than the conventional beamformer?**

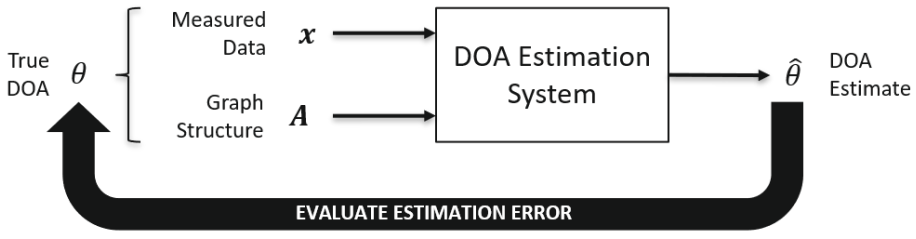


Figure 5.1: Overview of performance evaluation for a DOA estimation system.

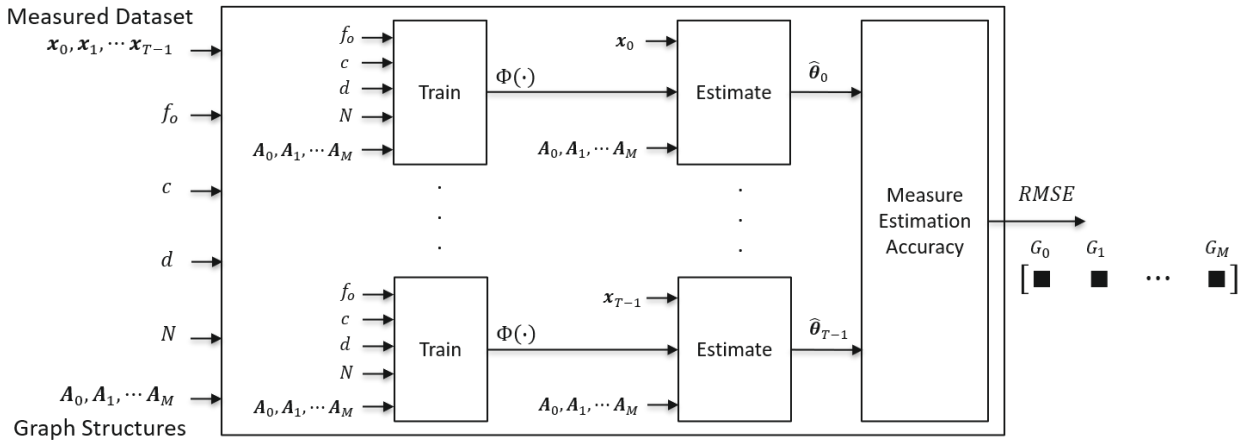


Figure 5.2: Performance evaluation system that measured estimation accuracy for every graph G_m that was to be tested. Multiple realizations of data $\mathbf{x}_t$ were tested for each input graph structure $\mathbf{A}_m$ and with known parameters about the source, environment, and receiver.

5.1 Approach

To answer this question, a test system was devised that would measure and compare estimation performance among different graph structures (Figure 5.2). This test system is designed to accept a set of inputs made up of multiple single snapshots of data, all the graph structures to be evaluated, and information about the source, environment, and ULA receiver. For every snapshot of data that is provided, the data is passed through our DOA estimation system, a process that consists of training and then estimation for every graph structure in our input test set. The training process, as a reminder, is used to produce the bin-to-angle conversion function that the second process needs to estimate DOA. Every snapshot of data that is processed by our testing system will produce a vector of DOA estimates, an estimate for every graph structure. Once all data snapshots are processed, estimation performance is measured for each graph by taking DOA estimates produced by each graph and calculating the average amount of accuracy that was observed. The final output of our test system is a vector of performance measurements, one for every graph structure that was tested.

5.1.1 Metric for Estimation Accuracy

To quantify estimation accuracy, our test system used as a metric a root-mean-square error (RMSE). RMSE was the most common metric that found across all the papers read in single-snapshot DOA estimation. RMSE measures DOA estimation accuracy, in units of degrees, by computing the square root of the average squared deviation between a system's DOA estimates $\hat{\theta}$ and the true DOA angle θ (Equation 5.1) [40]).

$$\text{RMSE} = \sqrt{E\left[(\hat{\theta} - \theta)^2\right]} \tag{5.1}$$

In prior DOA estimation studies, RMSE is calculated by either using a single fixed value for θ [12] or a set of values for θ in which each angle is equally likely to occur [41]. This means that RMSE can be used to quantify performance for both single-source and multi-source scenarios. Other performance metrics that are used in the literature, specifically in the prior work in SPG-based DOA estimation, include the shape of the beampattern, the white noise gain, and the directionality factor [24–26]. These metrics, however, are more relevant for environments with multiple sound sources and do not provide much information for a single source. Since a single source is assumed for this dissertation, we chose to use RMSE as our performance metric.

5.1.2 Input Dataset

For the input dataset (Table 5.1), our test system used data collected from the September 2009 Cooperative Array Performance Experiment (CAPEX09) [42]. The entire experiment was performed in Lake Washington near Seattle, a shallow water environment with a soft lakebed and a variable sound speed profile. To simplify our performance evaluation study, we used the sound speed typical of environments like this [43] and assumed a constant sound speed of $c = 1420$ m/sec. From the CAPEX09 dataset, we used data specific to one of the sound sources that was deployed, a 50 msec chirp signal generated from a linear frequency modulated sinusoidal signal in which frequency was swept from 1.5 kHz to 4 kHz. The source transmitted wave fronts over a direct path and from a direction angle of $\theta = 5^\circ$ relative to the

Source Frequency	$1.5 \leq f_o \leq 4$ kHz
True DOA Angle	$\theta = 5^\circ$
Sound Speed	$c = 1420$ m/sec
No. Array Elements	$N = 32$
Inter-Element Spacing	$d = 0.224$ m
Sampling Rate	$F_s = 25,000$ samples/sec

Table 5.1: Summary of variables from CAPEX09 input dataset.

receiver [44]. Because the water was shallow, data was also received from surface and bottom reflected paths, but those were ignored in our study. In total, two receiver array systems were used in CAPEX09, but the array system we focused on was the ULA of pressure sensors. This array had $N = 32$ elements with data being recorded at a sampling rate of $F_s = 25,000$ samples/sec. The array was also configured with a dense inter-element spacing of $d = 0.224$ m, making the array fully of capable of capturing data from the transmitted chirp signal.

Because the transmitted signal did not come from a narrowband source that our estimation system expects, the input dataset was converted into single snapshots in which each snapshot had a known and fixed narrowband frequency f_o. To convert the CAPEX09 data into this desired form, data snapshots were extracted by using the input data's spectrogram (Figure 5.3). The spectrogram for every $N = 32$ time series were calculated first and then used to estimate a set of narrowband frequencies at different instances of time. In total, we were able to reliably extract $T = 16$ snapshots of data, each with its own narrowband frequency f_o.

5.1.3 Input Graph Structures and Relation to Prior Work

For the input graph structures, our evaluation study concentrated on two kinds of graphs: a directed cycle graph to represent the CBF and variations of the directed cycle graph to rep-

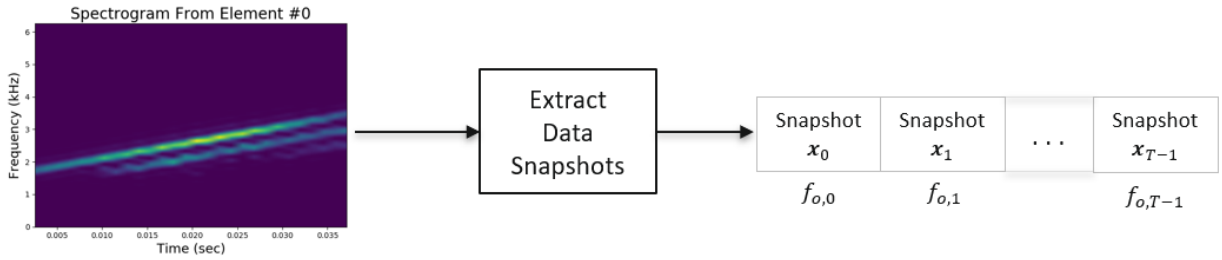


Figure 5.3: Process to extract single snapshots of data, each with their own narrowband frequency $f_{o,t}$, from the input chirp signal using the spectrograms.

resent the new graph structures that were being investigated for performance improvement. To narrow the focus of our study even further, we chose to investigate a class of structures that are based on the directed cycle graph and include one added edge (Figure 5.4). Focusing on just one added edge provided a way to build a solid foundation of work for studying the impact of graph structures on estimation performance. This approach also helped distinguish our work from other SPG-based DOA estimation studies, in which fully-connected graphs were employed [25]. Using one added edge was also useful because it provided our investigation with plenty of graph structures to study. For a graph with N vertices, we were able to measure and compare estimation performance between the directed cycle graph and a total of $M = N(N-2)$ new graph structures (proof shown below).

Proof. Consider a graph with N vertices:

- The corresponding adjacency matrix $\mathbf{A}$ will have N^2 available elements to fill in
- Since a graph with no self-loops is assumed, the main diagonal elements must be set to 0. This means there are only $N^2 - N$ elements left to fill in
- Since all new graphs are based on the directed cycle graph which consists of N directed edges, this leaves only $N^2 - N - N = N(N-2)$ elements left to fill in

Therefore, $M = N(N-2)$ directed cycle graphs with one added edge can be constructed. □

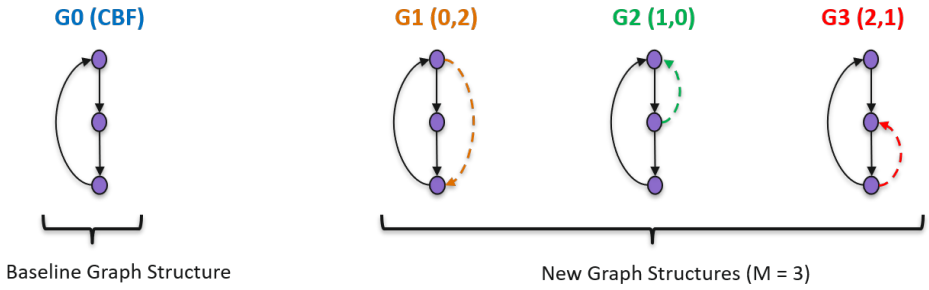


Figure 5.4: Graph structures that are to be tested assuming $N = 3$. Notation (n_1, n_2) is used to indicate in which vertex n_1 an edge starts and in which vertex n_2 an edge ends.

5.1.4 Expanding Breadth of Study

To increase the breadth of our study beyond a fixed DOA angle of $\theta = 5°$ and a ULA of only $N = 32$ elements, the CAPEX09 dataset was modified to enable two additional investigations in performance. The first modification was a change in the direction angle at which the source transmitted. To steer the data by an additional angle of φ degrees, the input dataset was altered in software by performing a phase shift in the frequency domain (Equation 5.2).

$$x_n(t) \longrightarrow X_n(j\omega) \longrightarrow X_{n,steered}(j\omega) = X_n(j\omega)e^{-j\omega\tau_n(\varphi)} \longrightarrow x_{n,steered}(t) \tag{5.2}$$

This procedure leverages the relationship between DOA angle φ, time delay, and phase shift. As discussed in [39], a change in DOA angle induces a constant delay in time in the recordings $x_n(t)$ between array elements, in which the DOA angle and the time delay at array element n are related by $\tau_n(\varphi) = \frac{nd\sin(\varphi)}{c}$. This delay in time can be implemented in the frequency domain using the time shift property of Fourier Transforms and shifting the phase by amount $\phi_n(\varphi) = -\omega\tau_n(\varphi)$. This gives us the updated dataset $X_{n,steered}(j\omega)$ in the frequency domain. The dataset is then converted back to the time domain with an inverse Fourier Transform to give us time series recordings $x_{n,steered}(t)$ in which the source is now steered to a new DOA angle of $\theta = \varphi + 5°$. Because the source is a real-valued sinusoidal, we were able to steer and

process data that have been steered to DOA angles ranging from $\theta = 0°$ to $\theta = 90°$.

The second modification we did to the data was remove the recordings from the last array element. To remove data from the last array element, the estimation system's variable N was updated to 31 and instructing our system to process data only from the first 31 elements rather than all 32. Updating the variable N also instructed our system to form graph structures with $N = 31$ vertices rather than $N = 32$.

These two modifications, on the whole, allowed us to measure and investigate how RMSE varies with DOA angle, as well as how much RMSE is impacted by the number of array elements. The results of all these performance evaluation studies are summarized and discussed in the next section.

5.2 Results

5.2.1 Impact of Using Different Graph Structures

When the unmodified CAPEX09 data was tested with other input graph structures, our estimation system saw no improvement in performance. For all the possible graph structures that were employed, the lowest RMSE that our system could produce was a value of 1°, which was identical to the RMSE for the CBF (Figure 5.5a). Our system produced this RMSE for 687 of the $M = 960$ new graph structures that were tested. All other graphs in our test set resulted in an RMSE greater than 1°. While no improvement was found, these results do indicate that our system can, at best, perform no better than the CBF. These results also continues to highlight the flexibility offered by our system. Different input graph structures can produce a different RMSE, so it is possible for the RMSE to decrease if the source is transmitting from a direction other than $\theta = 5°$.

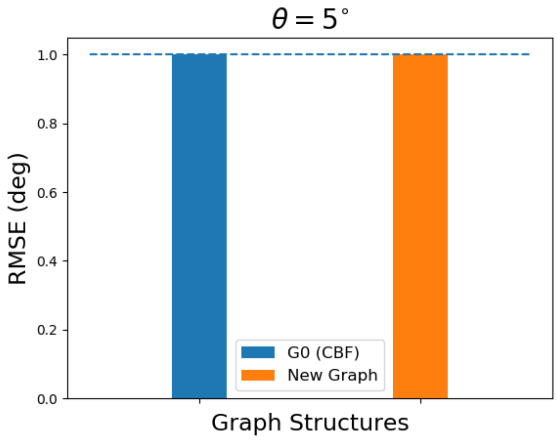
5.2.2 Impact of Using Different DOA Angles

When the CAPEX09 data was resynthesized to emulate transmissions from other direction angle, our estimation system showed, for some directions, an improvement in performance

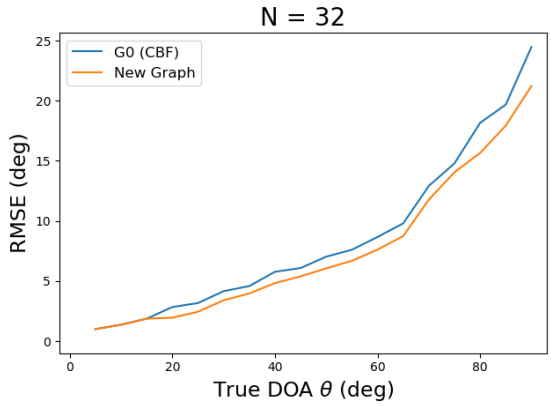
over the CBF. Our system continued to produce values of RMSE that were identical to the CBF, especially when the data was steered around broadside angles from $\theta = 5°$ to $\theta = 15°$ (Figure 5.5b). When the data was steered away from broadside, however, the RMSE started to decrease. At $\theta = 30°$, the RMSE from the best performing graph decreased from 4.16° to 3.40°, an improvement of 18.28%. This improvement trend continued as the DOA was steered towards end-fire angles with improvements of 12.08% and 13.78% at angles $\theta = 55°$ and $\theta = 80°$, respectively (Figure 5.5b). From these results, the impact of using different DOA angles is evident. Our system can perform either the same or better than the CBF, depending on the location of the source relative to the receiver array. This performance improvement is again a result of our system ability to flexibly alter and use its input graph structures.

5.2.3 Impact of Modifying Graph Size

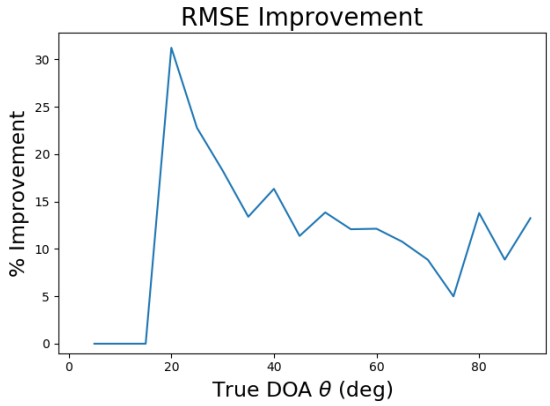
Our system has the ability to change graph configurations by not only adding edges, but also by modifying the graph size by adding or removing vertices. When the system was tested with data from one array element less ($N = 31$ vs. $N = 32$), the amount of performance improvement over the CBF increased in certain regions of DOA (Figure 5.6). Using $N = 31$ elements in our system showed little or no change in percentage improvement in RMSE versus the $N = 32$ configuration, especially when the data was near broadside. When the data was steered away from broadside and toward endfire, however, the change in percentage improvement went up for the $N = 31$ configuration. When $\theta = 30°$, the RMSE percentage improvement changed from 18.28% with $N = 32$ to 23.78% with $N = 31$. The change was 12.08% to 48.16% and 13.78% to 40.34% at angles $\theta = 55°$ and $\theta = 80°$, respectively. This pattern, in which RMSE improvement increased when using $N = 31$ versus $N = 32$ near endfire angles, demonstrates that our system can perform better than the CBF in other experimental scenarios. Modifying the array size has a noticeable impact. This pattern may also indicate that our system has a bias towards graphs structures with an odd number of elements and sources transmitting near endfire rather than near broadside. It would be



(a)



(b)



(c)

Figure 5.5: Results of performance evaluation tests using CAPEX09 data and $N = 32$ array elements. RMSE calculations between CBF and our system are shown for (a) unmodified data and (b) resynthesized data steered to other directions with (c) RMSE percentage improvement over CBF.

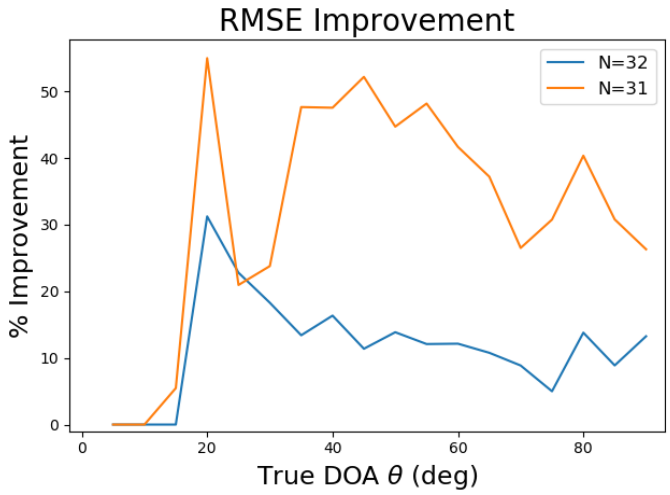


Figure 5.6: Comparing RMSE percentage improvement over the CBF with CAPEX09 data when $N = 32$ and $N = 31$ elements.

interesting to see whether this pattern of behavior holds with other datasets.

5.3 Summary

In summary, our graph-based DOA estimation system was evaluated for estimation accuracy. This performance evaluation was conducted by using real data collected from the CAPEX09 experiment. When this data was processed with graph structures other than the directed cycle graph, we saw that, under certain conditions, our estimation system can perform better than the CBF. For instance, when the data was resynthesized so that the source was transmitting away from broadside and towards end-fire, we found that our system can increase the estimation accuracy over the CBF by altering its choice for the input graph structure. The impact with the choice of graph was even more pronounced when we used one less array element, an array with an odd size of $N = 31$ rather than an even size of $N = 32$, in the system processing chain. This chapter's shift in focus to system performance evaluation, from system development, helps to further support our dissertation's main conjecture by

providing concrete evidence of gains in estimation accuracy. Indeed, the work in this chapter strengthens our argument that developing a single-snapshot DOA estimation system on a graph domain leads to performance improvement.

Chapter 6

EVALUATING EXPECTED PERFORMANCE

Of all the stages performed so far in the development cycle for an improved graph-based DOA estimation system, performance evaluation has been the most compelling with results from the last chapter showing that, under certain conditions, our system had an increase in estimation accuracy over the conventional beamformer. These results, however, were obtained from a single real dataset. To improve our understanding of how well our system operates, we need to go beyond a single dataset and study how we *expect* our system to perform with other input datasets. This task is carried out in prior work by evaluating a metric like estimation accuracy and assuming a input plane wave model [24–26]. The advantage of this model is that it consists of multiple parameters, such as DOA angle, number of array elements, and signal-to-noise ratio (SNR), to study and evaluate estimation performance. Studying how the variation of these parameters impacts estimation accuracy can not only give us more insight on what performance to expect from our system, but it can also provide more evidence for asserting performance improvement with our system. Our conjecture for this chapter's focus on performance evaluation is that **under certain conditions for the input measured data and the input graph structure, our system's *expected* estimation accuracy will improve over the conventional beamformer (CBF)**.

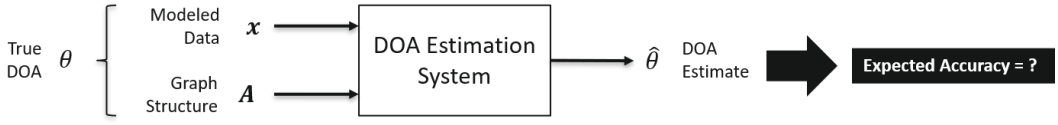


Figure 6.1: Overview of evaluating our DOA estimation system's expected performance.

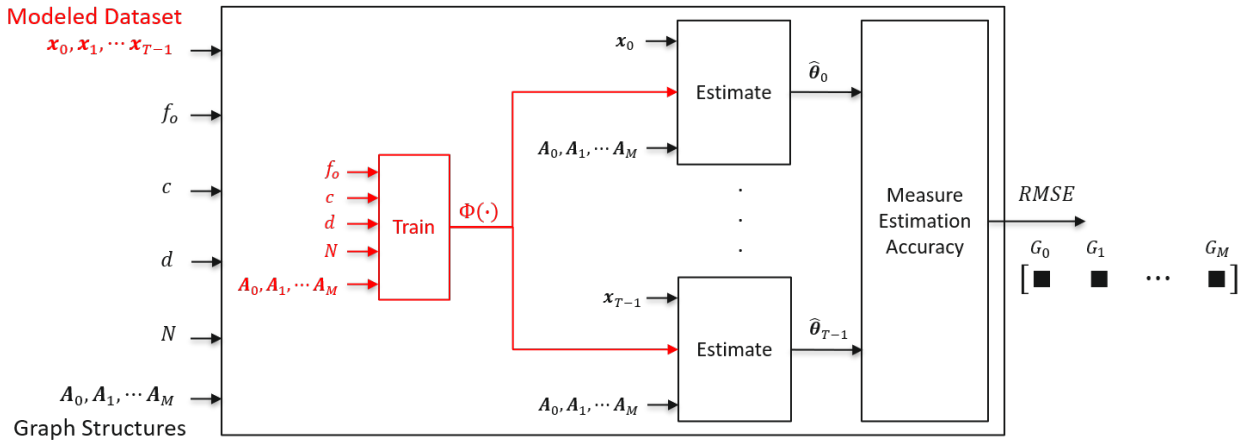


Figure 6.2: Performance evaluation system that was modified (shown in red) from the last chapter for expected estimation accuracy. Multiple realizations of modeled plane wave data $\mathbf{x}_t$ were tested for each input graph structure $\mathbf{A}_m$ and with known parameters about the source, environment, and receiver.

6.1 Approach

To measure our system's expected estimation accuracy, the performance evaluation system from the last chapter was modified (Figure 6.2). Two modifications were made overall. The input measured data was replaced with modeled plane wave data and the training section was reduced to a single system block. These modifications, as well as highlights of the rest of the system and the breadth of the evaluation study are discussed in the upcoming section.

6.1.1 Modified System Components

To generate input plane wave data for the performance evaluation system, we used the graph signal model and software implementation that were developed in Chapter 3. The plane wave model required six parameters to be fully specified. These six parameters include source frequency f_o, DOA angle θ, SNR, sound speed c, array element spacing d, and array size N. For every fixed set of parameters in the plane wave model, a total of $T = 20,000$ realizations

of plane wave data were produced using a fixed snapshot in time t_o and a sampling rate $F_s = 200,000$ samples/sec. These realizations of plane wave data are then processed for estimation accuracy

One impact of using plane waves as an input is that it simplified the evaluation processing chain. Rather than having a separate training block for every source frequency, the training section was reduced to a single system block. This is because the input in the last chapter was a frequency-varying chirp signal. For this chapter, we now have a signal fixed to a single frequency f_o across time. Therefore, only one frequency needed to be used to train and produce the required bin-to-angle conversion function for DOA estimation.

6.1.2 Other System Components

Other than the two changes described in the last section, the rest of the performance evaluation remained the same. For the performance metric, we continued to use a root-mean-square error (RMSE), which was consistent with how estimation accuracy was evaluated in prior work [4]. Continuing to use this metric also allowed for cohesion between this chapter and the last. To provide further cohesion between these two chapters, we evaluated our system with the same set of input graph structures. The directed cycle graph ($\mathbf{A}_0$) was used to represent the CBF, while all other $M = N(N-2)$ graph structures ($\mathbf{A}_m$), made up of the directed cycle graph with one added edge, were used to represent the other candidate domains for data modeling and processing (Figre 6.3). Using these set of graph structures allowed us to continue our study of the impact of graph structure selection on estimation accuracy.

6.1.3 Breadth of Performance Evaluation Study

To study the impact of plane wave parameters on a graph's ability to improve estimation accuracy, we fixed three of the plane wave's parameters and varied the other three. The plane wave was fixed to $f_o = 400$ Hz, $c = 1500$ m/sec, and $d = 1.875$ m. Fixing these parameters to these values allowed us to avoid a case where multiple frequency bins correspond to single DOA angle, a situation known as spatial aliasing [27]. This left us with three remaining

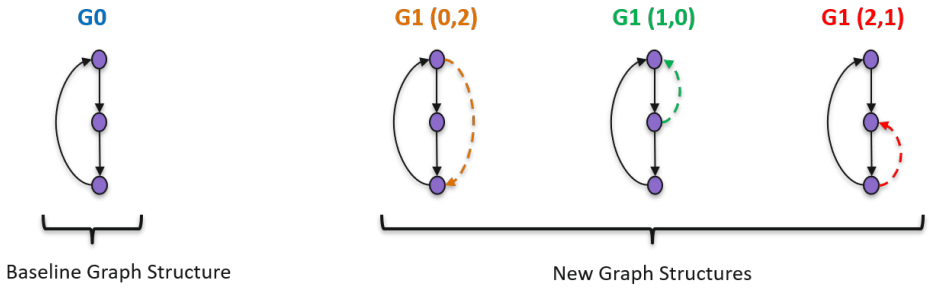


Figure 6.3: Example of set of input graph structures tested if $N = 3$. Notation (n_1, n_2) is used to indicate in which vertex n_1 an edge starts and in which vertex n_2 an edge ends.

parameters, θ, SNR, and N, to vary. These parameters were modified and studied separately when computing the corresponding the RMSE. This allowed us to plot and study how RMSE changed with DOA angle, with SNR, and with array size.

To measure the extent of the improvement, our plot of RMSE versus SNR also includes the Cramer-Rao Lower Bound (CRLB). The CRLB defines a constraint on the RMSE given information about the array size N and the SNR (Equation 6.1) [45].

$$CRLB = \frac{6}{4\pi^2 N(N^2-1)SNR} \tag{6.1}$$

Including the CRLB was important to also maintain consistency with other prior work [2] in studying the impact of SNR on RMSE. The results of this investigation, as well as all others we previously described, are shown in the next section.

6.2 Results

6.2.1 Impact of True DOA Angle

When estimation accuracy was measured for the case of variable θ and fixed N and SNR, we found several cases in which our system's RMSE decreased over the CBF. One example was the case when $N = 5$ and $SNR = 2.5$ dB (Figure 6.4a). For this case, we found one input graph structure $G15(4, 3)$ (Figure 6.4b) that resulted in a decrease in RMSE at DOA angles

near end-fire. At angle $\theta = -65°$, the decrease over the CBF was about 2.42%, while at $\theta = 65°$, it was 3.45%. This pattern of RMSE reduction near end-fire was also discovered in three other system input scenarios we tested. For the case when $N = 15$ and $SNR = -2.5$ dB (Figure 6.4cd), RMSE went down by at most 5% at $\theta = -76°$ and it went down for more than one graph structure. As we increased the array size to values like $N = 31$ and $N = 63$ (Figure 6.5), we found more DOA regions that had an improvement in RMSE over the CBF, but the region near end-fire continued to show reductions. At $N = 31$, our system produced a decline in RMSE of 12.75% at $\theta = -79°$, while at $N = 63$, the decline was 6.53% at $\theta = -81°$. These results demonstrate that, depending on which graph structure is selected and which DOA angle our source transmits from, our system can perform better than the CBF. The improvement in performance is also consistent with the results from the last chapter, showing again that an increase in estimation accuracy is most pronounced when a source transmits at an angle near end-fire.

6.2.2 Impact of SNR

When we fixed the number of array elements and the DOA angle to end-fire, we saw that the amount of RMSE improvement with our system varied with SNR (Figure 6.6). Each array size N used a different range for SNR because some arrays did not start showing error estimates until the amount of noise was increased. Overall, we found that the amount of improvement over the CBF was at its greatest for cases of higher SNR. For example, when $N = 31$, RMSE in our system had a noticeable reduction at SNR = -5 dB and SNR = -2.5 dB. As the amount of noise increased and got closer to -15 dB, the amount of RMSE reduction over the CBF decreased but the value never went above the RMSE of the CBF. This same pattern occurred for all other graph structures we analyzed in Figure 6.6. These results indicate, first and foremost, that the results from the last section were not random. Our results here show that our system can perform the same or better than the CBF at end-fire, depending on the amount of noise in the input data. The improvement in estimation accuracy over the CBF is most pronounced when processing data with higher SNRs.

Possible System Outputs $N = 5$	$\hat{\theta} = \{\pm 65°, \pm 23°, 0°\}$
Possible System Outputs $N = 15$	$\hat{\theta} = \{\pm 76°, \pm 54°, \pm 42°, \pm 32°, \pm 23°, \pm 15°, \pm 7°, 0°\}$

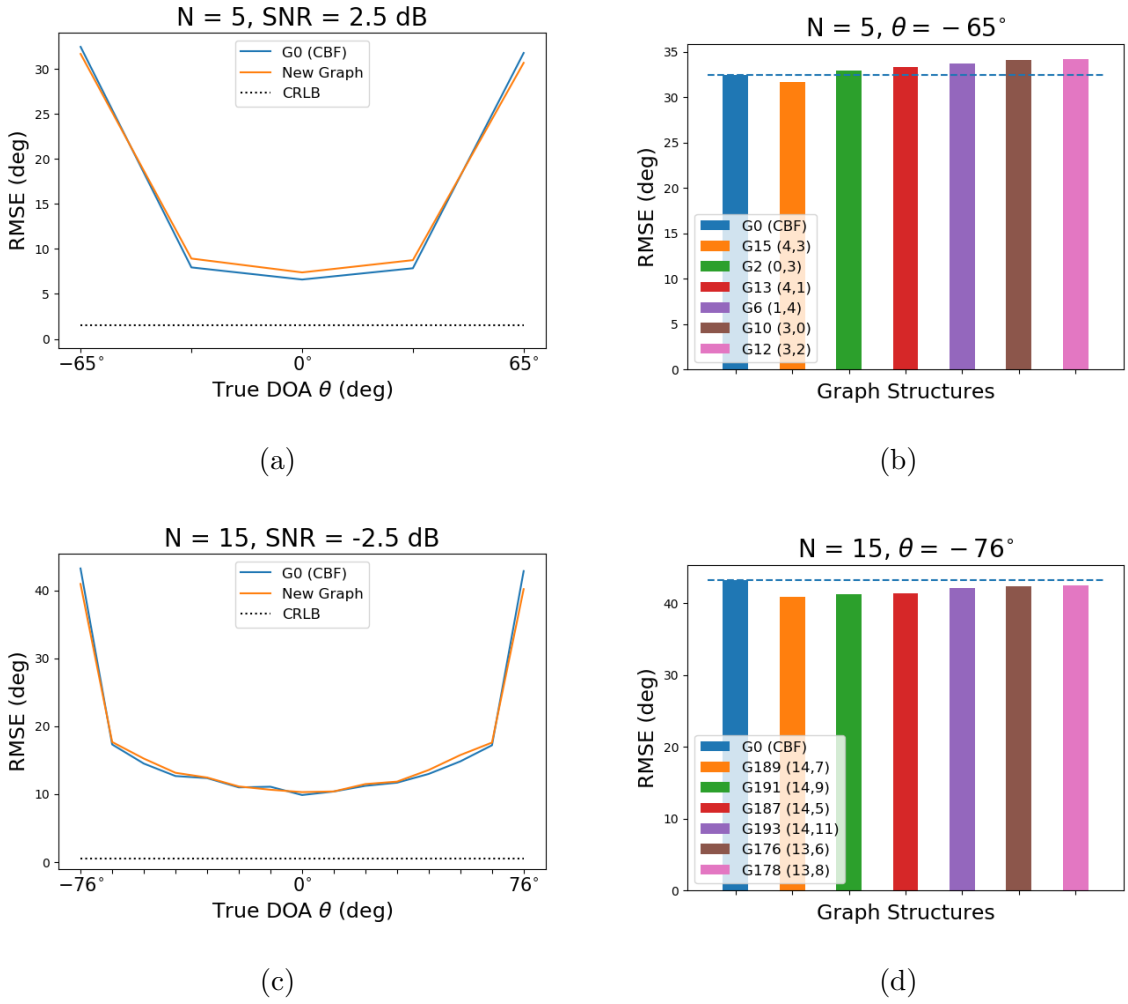
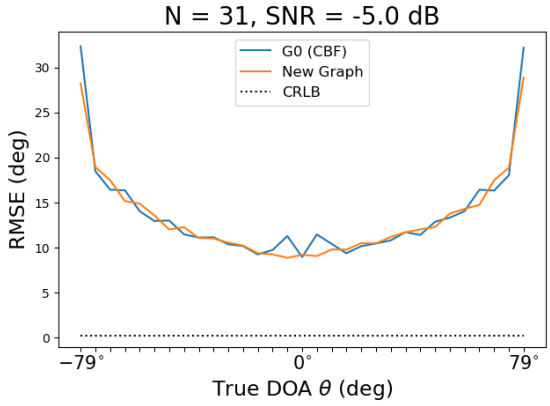
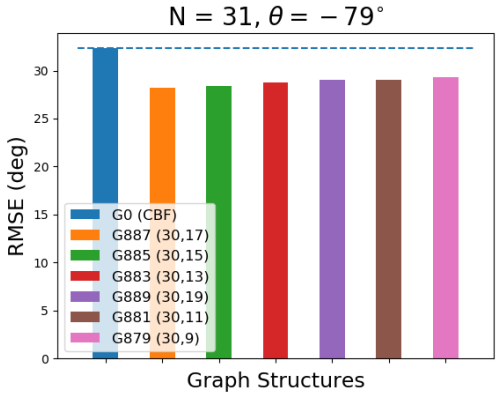


Figure 6.4: Comparing estimation accuracy between CBF and our system with the best performing graph structure(s). The impact of varying DOA angle θ is shown for (a) $N = 5$ and (c) $N = 15$ with the Cramer Rao Lower Bound (CRLB) is shown for reference. The results for just end-fire DOA angles are also plotted to compare the CBF and our system's six best performing graph structures for (b) $N = 5$ and (d) $N = 15$.

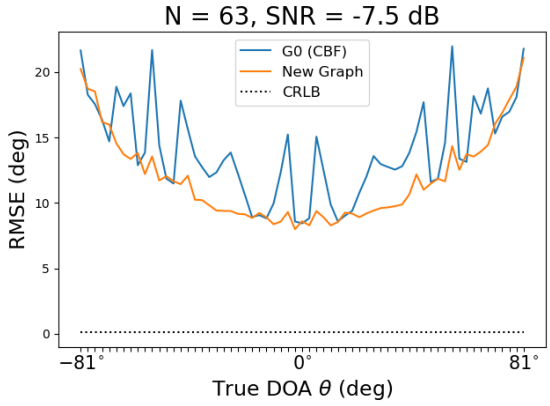
Possible System Outputs $N = 31$	$\hat{\theta} = \{\pm 79°, \pm 65°, \pm 57°, \pm 50°, \pm 45°, \pm 40°, \pm 35°, \pm 31°, \pm 27°, \pm 23°, \pm 19°, \pm 15°, \pm 11°, \pm 7°, \pm 3°, 0°\}$
Possible System Outputs $N = 63$	$\hat{\theta} = \{\pm 81°, \pm 72°, \pm 67°, \pm 63°, \pm 59°, \pm 55°, \pm 52°, \pm 49°, \pm 47°, \pm 44°, \pm 41°, \pm 39°, \pm 37°, \pm 35°, \pm 32°, \pm 30°, \pm 28°, \pm 26°, \pm 24°, \pm 22°, \pm 20°, \pm 18°, \pm 17°, \pm 15°, \pm 13°, \pm 11°, \pm 9°, \pm 7°, \pm 5°, \pm 3°, \pm 2°, 0°\}$



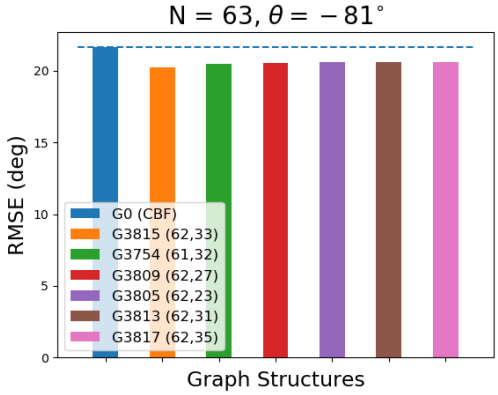
(a)



(b)



(c)



(d)

Figure 6.5: Comparing estimation accuracy as a function DOA angle between CBF and our system with the best performing graph structure(s) for (a,b) $N = 31$ and (c,d) $N = 63$.

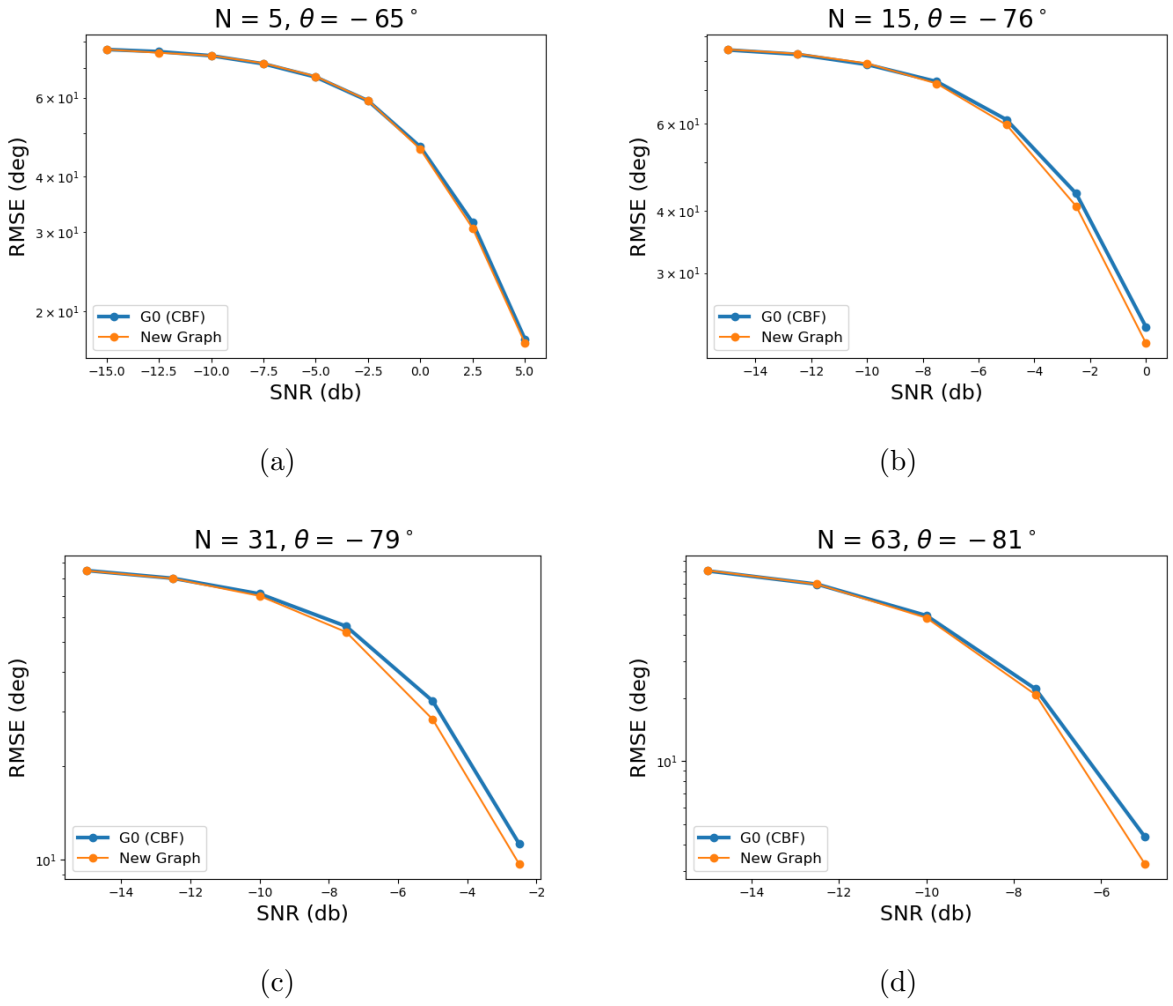


Figure 6.6: Estimation accuracy at end-fire as a function of SNR for (a) $N = 5$, (b) $N = 15$, (c) $N = 31$, and (d) $N = 63$.

6.2.3 Impact of Array Size

When the SNR and DOA angle were fixed, RMSE only decreased at end-fire when the array size N was an odd number. This pattern was noticeable for both lower and higher order array sizes. For the case of $N = 5$ (Figure 6.7a), RMSE decreased over the CBF and that decrease occurred with one graph structure configuration. For the case of $N = 6$ (Figure 6.7b), however, all the best performing new graphs had an increase in RMSE over the CBF. The best performing graph had an RMSE increase of 11.76%, which means all other graph structures we tested had an RMSE increase close to this amount or even higher. This rise in RMSE for all possible new graphs continued with $N = 16$ (Figure 6.7d), $N = 32$ (Figure 6.8b), and $N = 64$ (Figure 6.8d). These results are important because they identify a condition in array size for which our system can produce improvement in performance. These results also confirm the observation that was made in the last chapter: that our system has a bias towards odd-sized arrays when attempting to estimate the DOA from sources directed at end-fire.

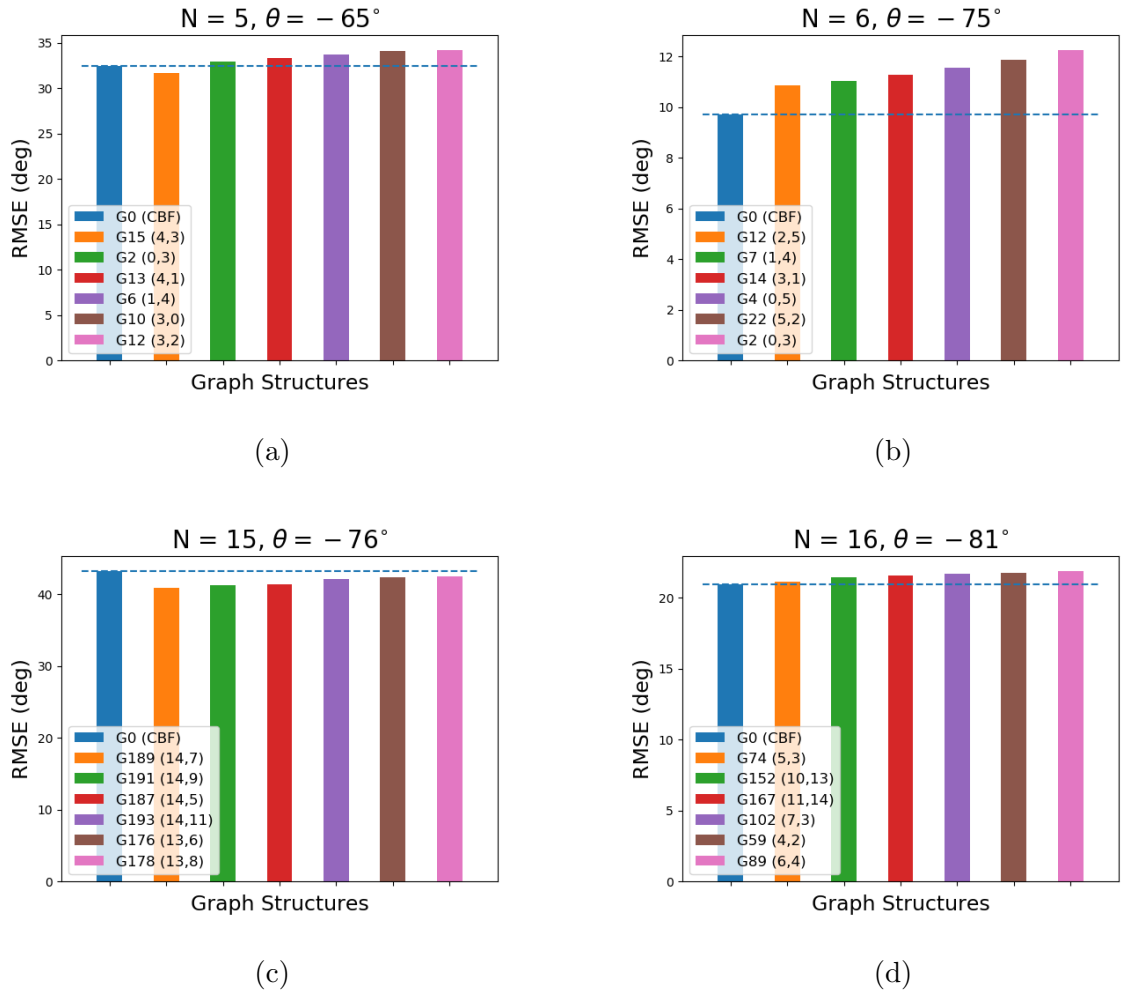


Figure 6.7: Comparing estimation accuracy at end-fire between CBF and our system with the six best performing graph structures for N = odd and N = even. Plots are shown for odd-sized arrays, (a) $N = 5$ and (c) $N = 15$, and even-sized arrays, (b) $N = 6$ and (d) $N = 16$.

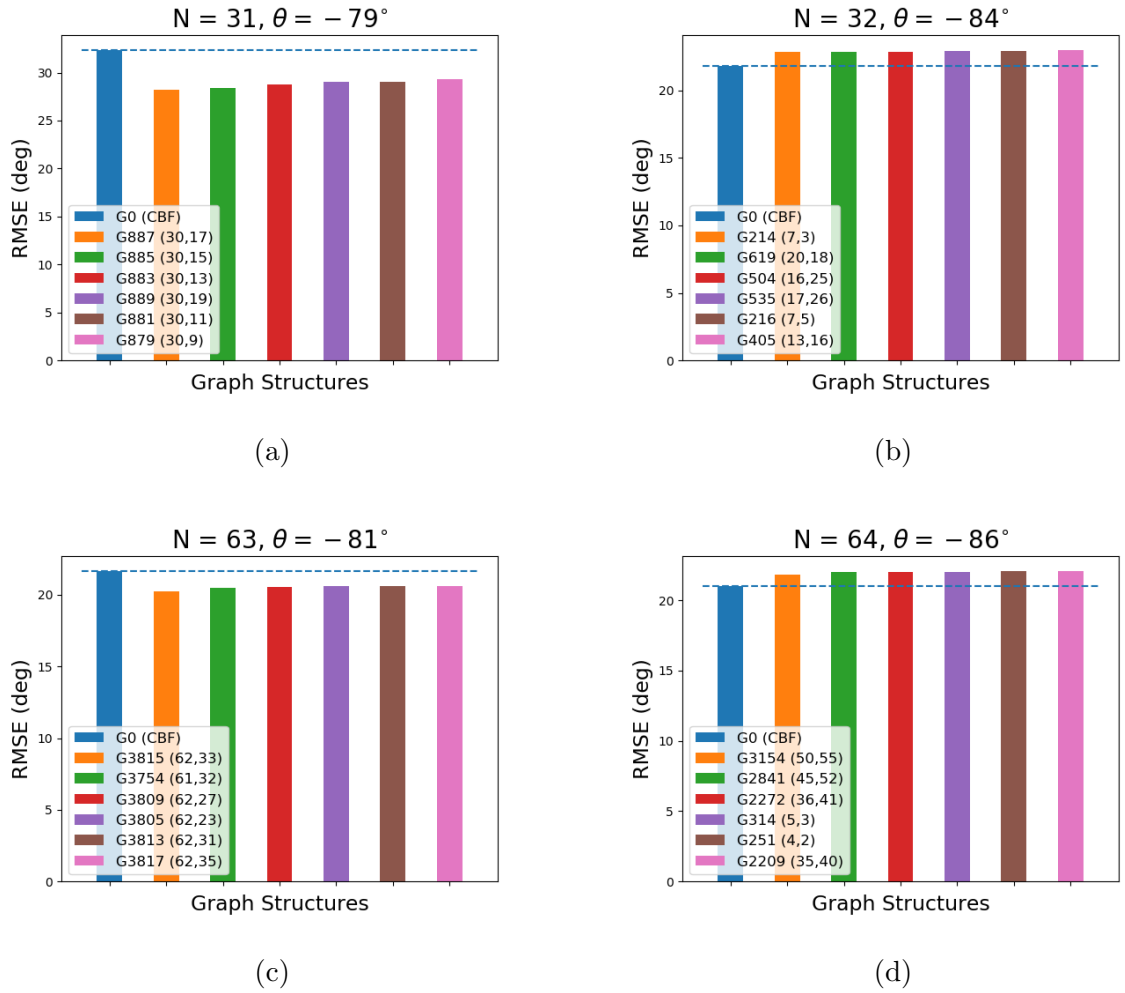


Figure 6.8: Comparing estimation accuracy at end-fire between CBF and our system with the six best performing graph structures for N = odd and N = even. Plots are shown for odd-sized arrays, (a) $N = 31$ and (c) $N = 63$, and even-sized arrays, (b) $N = 32$ and (d) $N = 64$.

6.3 Summary

In summary, the work on estimation performance evaluation was expanded from the last chapter. This time, the focus was on expected performance, and this task was carried out by measuring RMSE with input plane waves that varied in DOA angle θ, noise amount SNR, and array size N. Our collection of RMSE results confirm that our system does indeed perform better than the CBF, under certain input conditions. These conditions include a source transmitting from end-fire DOA angles, data with higher SNRs, and odd-sized arrays. If these conditions are met, we saw that there exists a set of input graph structures that will cause an increase in our system's estimation accuracy. The outcomes obtained from this performance evaluation only help to enhance what we learned previously. In the last chapter, we gained an understanding of how well our system operated with one real dataset. With this chapter, we gained an improved understanding for how our system is expected to operate with other datasets. The combined evidence from the last two chapters only helps to strengthen the argument for using SPG to develop a DOA estimate system on the graph domain.

Chapter 7

EXPLAINING SYSTEM PERFORMANCE BEHAVIOR

Our graph-based single-snapshot DOA estimation system has been the focus of attention over the last several chapters with the most recent results showing that the conventional beamformer (CBF) can be improved upon with our system solution. But just as important as the solution itself is the explainability of the solution. To have additional confidence in our system's design and its output performance, and to produce further insights from them, we need to study *the why*. Why does our estimation system show estimation performance improvement at end-fire? Why does the estimation accuracy increase for odd-sized arrays? Why does the RMSE decrease when using data with higher SNRs? Our conjecture for this chapter is that **all these question can be answered by leveraging supporting material from array signal processing theory, SPG, spectral graph theory, and probability theory and statistics**.

7.1 Preliminaries

Before we start addressing these questions, we begin with a review of relevant material from previous chapters that will be referenced in upcoming sections.

7.1.1 Review of Far-Field Plane Wave Model

The model for the measured input data $\mathbf{x}$ is given by Equation 7.1 in which $s(t_o)$ is the narrowband source signal at snapshot t_o, $\mathbf{a}(\theta_o)$ is the steering vector directed at DOA angle θ, and $\mathbf{w}$ is the additive noise component. The noise component is modeled as a circularly

symmetric complex normal distribution with $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \frac{1}{SNR}\mathbf{I}\right)$.

$$\begin{aligned} \mathbf{x} &= \mathbf{a}(\theta)s(t_o) + \mathbf{w} \\ \mathbf{a}(\theta) &= \left[e^{j2\pi f_o \frac{d\sin(\theta)}{c} n}\right]_{n=0}^{N-1} \text{ and } s(t_o) = e^{j2\pi f_o t_o} \end{aligned} \tag{7.1}$$

7.1.2 Review of Power Spectrum Equation

Calculating the power spectrum is the first processing stage of our estimation system, and it is calculated using Equation 7.2 with GFT linear weights $\mathbf{f}_\ell$.

$$P_\ell = \frac{\mathbf{f}_\ell \mathbf{x}\mathbf{x}^H \mathbf{f}_\ell^H}{\mathbf{f}_\ell \mathbf{f}_\ell^H} \tag{7.2}$$

This formula can be rewritten in two different ways:

- In terms of magnitude and L2-norm operators:

$$P_\ell = \frac{|\mathbf{f}_\ell \mathbf{x}|^2}{\|\mathbf{f}_\ell\|^2} \tag{7.3}$$

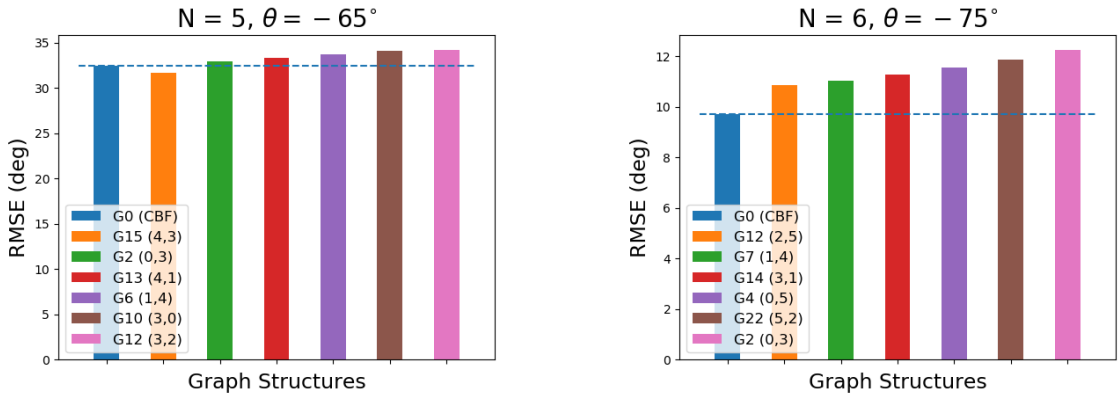
- In terms of the input data's GFT $\hat{x}_\ell$ and its respective real and imaginary components, $\hat{x}_{\ell,R}$ and $\hat{x}_{\ell,I}$

$$P_\ell = \frac{|\hat{x}_\ell|^2}{\|\mathbf{f}_\ell\|^2} = \frac{\hat{x}_{\ell,R}^2}{\|\mathbf{f}_\ell\|^2} + \frac{\hat{x}_{\ell,I}^2}{\|\mathbf{f}_\ell\|^2} = \left(\frac{\hat{x}_{\ell,R}}{\|\mathbf{f}_\ell\|}\right)^2 + \left(\frac{\hat{x}_{\ell,I}}{\|\mathbf{f}_\ell\|}\right)^2 \tag{7.4}$$

7.1.3 Review of a Few Performance Evaluation Experiments

Results from array size $N = 5$ and $N = 6$ using simulated plane wave data are summarized in Figure 7.1.

Possible System Outputs $N = 5$	$\hat{\theta} = \{\pm 65°, \pm 23°, 0°\}$
Possible System Outputs $N = 6$	$\hat{\theta} = \{-75°, \pm 43°, \pm 20°, 0°\}$



(a) (b)

Figure 7.1: Review of estimation performance results at end-fire, for the top six best-performing graphs, for array size (a) $N = 5$ and (b) $N = 6$.

7.2 Improvements at End-Fire

7.2.1 Supporting Theory

To analyze the performance improvement at end-fire, we leveraged an idea from array signal processing that interprets the conventional beamformer (CBF) as a matched (spatial) filter [46, 47]. What this means is that the job of the CBF, through a calculation of the power spectrum P_ℓ in the first stage, is to pass or reject an input signal $\mathbf{x}$ according to which direction θ the source is being transmitted from. The CBF's ability to filter is aided by the set of linear Fourier weights $\mathbf{f}_\ell$ that are used to calculate power spectrum. Every frequency bin ℓ is matched to a single direction angle θ. So if $\theta = \theta_o$, which is known to map to bin $\ell = \ell_o$, the power computed with linear weights $\mathbf{f}_{\ell_o}$ should be higher than the power computed at all other frequency bins. The ideal CBF will possess a set of linear Fourier

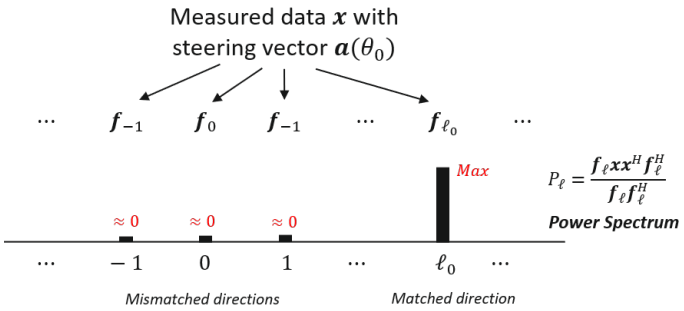


Figure 7.2: Ideal power spectrum behavior for a CBF.

weights such that the power at the matched bin ℓ_o is maximum, while the power at all other mismatched bins $\ell \neq \ell_o$ is zero (Figure 7.2). If a value of zero cannot be obtained, then just as important is how far away the power is at each bin from the maximum value because it indicates how well our system's linear Fourier weights both passes an input at the matched direction and rejects the input a the mismatched direction. The quality of a CBF's performance is therefore dependent on the quality of the Fourier weights.

To assess the quality of Fourier weights, we can measure, in the assumed direction angle θ_o, how parallel or how aligned in phase the input signal's steering vector $\mathbf{a}(\theta_o)$ and the Fourier weights $\mathbf{f}_\ell$ are by using a Hermitian angle. The reason the input's steering vector is important is because, at high SNR, the input data $\mathbf{x}$ and its power spectrum P_ℓ vary according to the steering vector $a(\theta)$ (Equation 7.5).

$$\begin{aligned} \mathbf{x} &= \mathbf{a}(\theta)s(t_o) + \mathbf{w} \approx \mathbf{a}(\theta)s(t_o) \\ P_\ell &= \frac{|\mathbf{f}_\ell \mathbf{x}|^2}{\|\mathbf{f}_\ell\|^2} \approx \frac{|\mathbf{f}_\ell \mathbf{a}(\theta)s(t_o)|^2}{\|\mathbf{f}_\ell\|^2} = \frac{|\mathbf{f}_\ell \mathbf{a}(\theta)|^2 |s(t_o)|^2}{\|\mathbf{f}_\ell\|^2} = \frac{|\mathbf{f}_\ell \mathbf{a}(\theta)|^2}{\|\mathbf{f}_\ell\|^2} \end{aligned} \tag{7.5}$$

This means that the power spectrum attains its maximum when linear weights $\mathbf{f}_\ell$ is parallel or aligned in phase with vector $\mathbf{a}(\theta)$. Phase alignment can be measured by a quantity known as the Hermitian angle [48]. The Hermitian angle is similar to a regular inner product in that it calculates the angle between two vectors. Whereas the inner product calculates the angle between two real-valued vectors, the Hermitian angle is used to measure the angle between

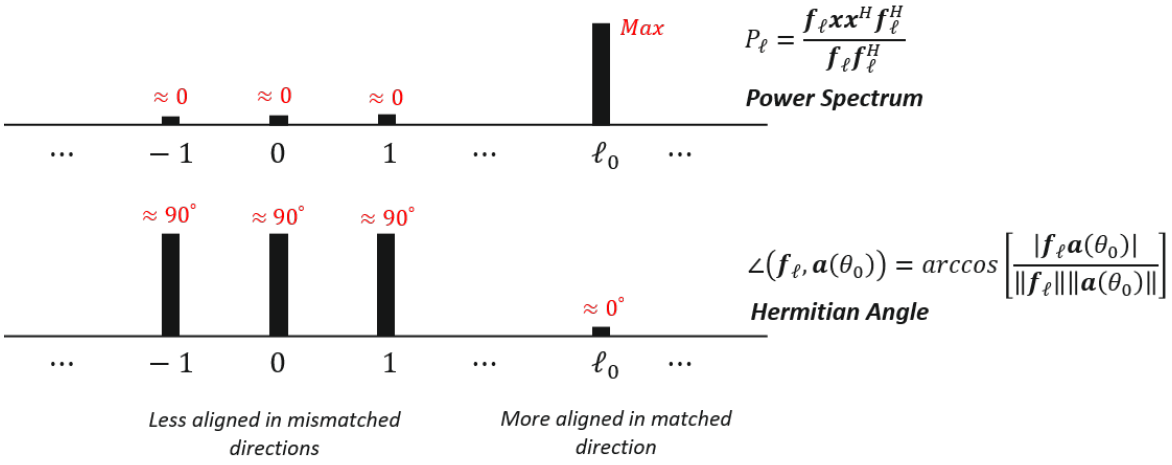


Figure 7.3: Ideal values in the Hermitan angle for a CBF.

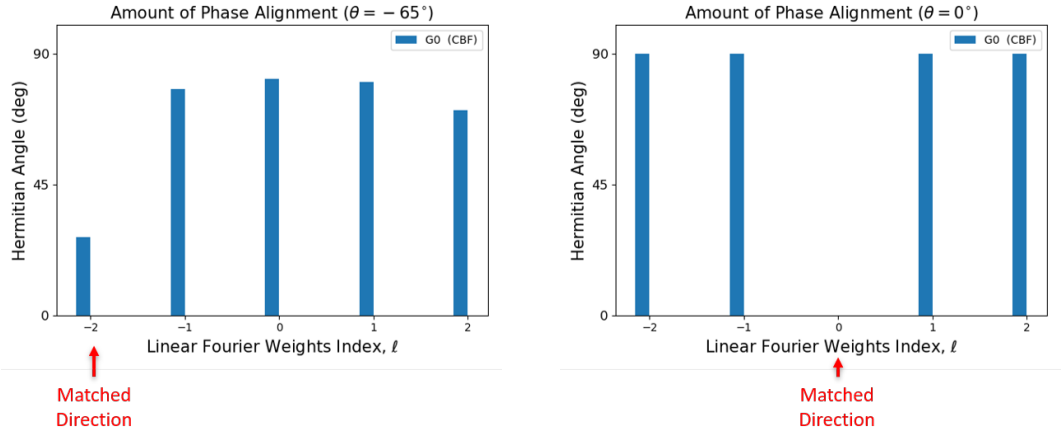
two complex-valued vectors. The Hermitian angle between complex-valued vectors $\mathbf{f}_\ell$ and $\mathbf{a}(\theta)$ is given by Equation 7.6.

$$\angle(\mathbf{f}_\ell, \mathbf{a}(\theta)) = \arccos\left[\frac{|\mathbf{f}_\ell \mathbf{a}(\theta)|}{\|\mathbf{f}_\ell\|\|\mathbf{a}(\theta)\|}\right] \tag{7.6}$$

When the two vectors are parallel, the Hermitian angle will report a value of 0°. When they are perpendicular, a value of 90° is reported. This information is useful because we can use it to inform what we would expect to see in the power spectrum (Figure 7.3). If the values of 0° and 90° are not obtainable, then looking at the difference in angles is just as important. In particular, it is useful to compute the difference of the Hermitian angle at a mismatched direction $\ell \neq \ell_o$ minus the Hermitian angle at the matched direction $\ell = \ell_o$. Like what we already described for the power spectrum, this difference is important because it indicate how well our estimation system operates as a matched filter.

7.2.2 Applying Knowledge to Explain End-Fire Improvement

Using knowledge of Hermitian angles and the interpretation of the CBF as a matched filter, we can identify why the CBF had room for improvement at end-fire. This can be explained



(a) (b)

Figure 7.4: Amount of phase alignment between input steering vector and linear Fourier weights for the CBF. Hermitian angles are calculated for the case when $N = 5$ and when input directed from (a) end-fire and (b) broadside.

$\angle(\mathbf{f}_\ell, \mathbf{a}(\theta)) - \angle(\mathbf{f}_{\ell_o}, \mathbf{a}(\theta))$				
	$\ell = -1$	$\ell = 0$	$\ell = 1$	$\ell = 2$
G0 (CBF)	51.1°	54.5°	53.3°	43.7°

Table 7.1: Difference between Hermitian angles at frequency bin ℓ and the frequency bin at the matched direction $\ell_o = -2$ for graph G0 and end-fire DOA $\theta = -65°$

.

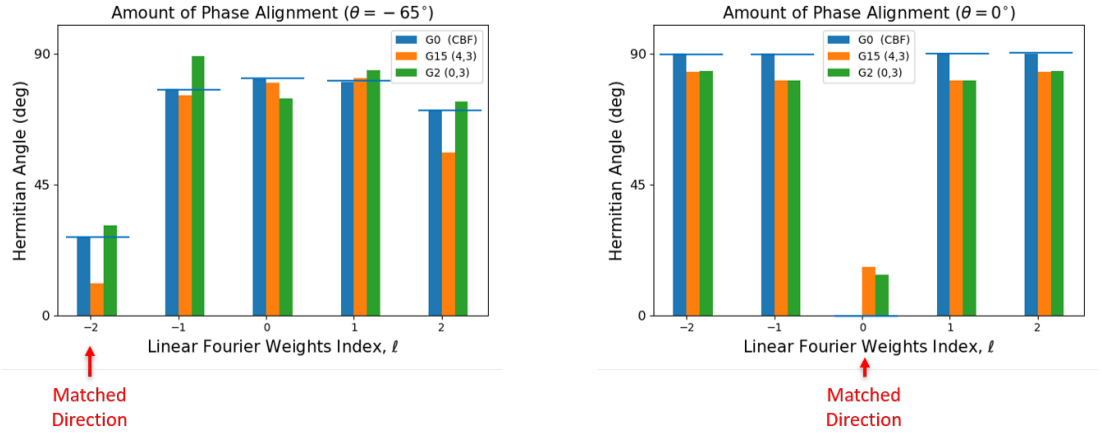
by considering plots of the amount of phase alignment versus frequency bin for input plane waves with no noise and steered in different directions. When the Hermitian angles were computed for end-fire angle $\theta = -65°$ and $N = 5$ array elements (Figure 7.4a), the outputs were not ideal. The Hermitian angle was 26.9° in the matched bin $\ell = -1$, while in the mismatched bins, the hermitian angles ranged from 70.5° to 81.4°. This means that in the matched direction, the amount of how parallel vectors $\mathbf{f}_\ell$ and $\mathbf{a}(\theta)$ are has room to get better and reach the ideal of 0°. The CBF also has room to grow in rejecting input signals in the mismatched directions since the ideal value of 90° was not reached and the difference in Hermitian angles (Table 7.1) were still far away from 90°.

When we plotted the Hermitian angles for the CBF at broadside direction $\theta = 0°$ (Figure 7.4b), the behavior was opposite from what we saw in end-fire. The Hermitian angles obtained reached the ideal value of 0° in the matched bin $\ell = 0$, as well as the ideal value of 90° in the mismatched bins $\ell \neq 0$. The CBF is therefore a perfect matched filter at broadside and has no room for any further improvement.

The only way to modify the phase alignment at end-fire is to modify the linear Fourier weights, and that is exactly what our estimation system has to offer. By introducing an input graph structure into the system processing chain, the Fourier weights $\mathbf{f}_\ell$ can change and potentially increase our system's ability to operate as an ideal matched filter.

When we plotted the amount of phase alignment with our estimation system and compared the top two best-performing graphs with the CBF, we can see why our system did not improve at broadside (Figure 7.5b). The Hermitian angles that were calculated with an input at broadside went in an undesired direction. At the matched bin $\ell = 0$, the Hermitian angles with the two new graph structures increased over the CBF, while at the mismatched bins $\ell \neq= 0$, the Hermitian angles decreased. This implies that processing data from broadside with these new graph structures will not work because the linear Fourier weights are not as well-matched as the weights from the CBF.

When the phase alignment (Figure 7.5a) and the difference in phase alignment (Table 7.2) at end-fire was analyzed, the results varied for different bins and for different graph



(a) (b)

Figure 7.5: Comparing amount of phase alignment, between input steering vector and linear Fourier weights, for the CBF and the top two new graphs. Plots show the case when $N = 5$ and when input was directed from (a) end-fire and (b) broadside.

$\angle(\mathbf{f}_\ell, \mathbf{a}(\theta)) - \angle(\mathbf{f}_{\ell_o}, \mathbf{a}(\theta))$				
	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$
G0 (CBF)	51.1°	54.5°	53.3°	43.7°
G15 (4,3)	**64.9°**	**69.2°**	**70.8°**	**45.1°**
G2 (0,3)	**58.3°**	43.7°	**53.5°**	42.7°

Table 7.2: Comparing difference between Hermitian angles at frequency bin ℓ and the frequency bin at the matched direction $\ell_o = -2$ at end-fire DOA $\theta = -65°$. New graphs that increased in difference over graph G0 are highlighted in bold.

structures. Graph G15 showed a noticeable decrease in Hermitian angle versus Graph G0 at bin $\ell = -2$, but in the mismatched bin, the Hermitian angle only went up in one of the four bins, making it difficult to draw a conclusion. When we analyzed G15's difference in Hermitian angles, the performance increase over graph G0 is much more clear. All the differences in angle went up for G15 over G0, indicating an increased ability to operate as a matched filter and improve in estimation accuracy at end-fire. The differences in angle for graph G2, on the other hand, also explain why it had weaker estimation accuracy. Its differences only increased over G0 in two of the four bins, and so the net effect results in a slight decrease in estimation accuracy at end-fire.

7.3 Improvements With Certain Graph Structures

7.3.1 Supporting Theory

To investigate the impact of adding an edge on the quality of the linear Fourier weights matched to end-fire directions, we leveraged recent studies from SPG that relate directed edges to information flow. In [49], directed edges are interpreted as specifying the direction of signal flow from a large value to a smaller value. In [50], directed edges are used in a similar fashion to represent the brain. The graph is constructed so that edges capture the direction that information flows between different regions in the brain.

We build on this concept of information flow between vertices interconnected on a graph and measure phase between two vertices. Since phase alignment between the input steering vector and the Fourier weights is crucial to obtaining optimal estimation performance, as shown in the last last section, we measured the difference in phase between signal data at vertices connected by a directed edge. If two vertices n_1 and n_2 are inter-connected by an edge directed from vertex n_1 to vertex n_2, the difference in phase of signals $x(n)$ on these vertices is defined as $\Delta\phi(n_1, n_2)$ in Equation 7.7.

$$\Delta\phi(n_1, n_2) = \angle x(n_2) - \angle x(n_1) \tag{7.7}$$

Differences in phase were measured and analyzed for the linear Fourier weights from different

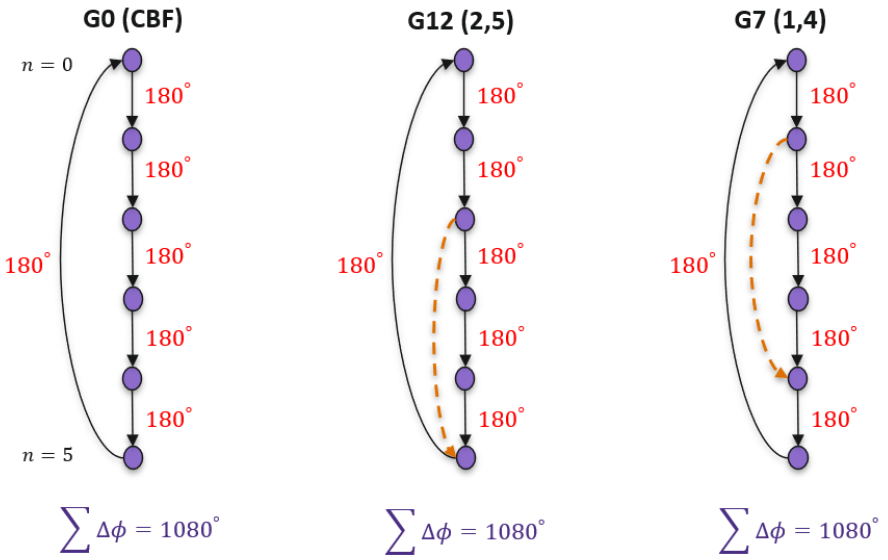
graph structures that are matched to end-fire angles.

To understand why odd-sized graphs improve estimation performance at end-fire over even-sized graphs, we leveraged material from spectral graph theory and SPG on the relationship between eigenvalues, eigenvectors, and linear Fourier weights. In [38], the eigenvalues and eigenvectors for a directed cycle graph described by an adjacency matrix is discussed. It was shown that the eigenvalues of this graph type are the nth roots of unity and can be plotted on a complex plane around the unit circle. It was also mentioned that every eigenvalue λ_ℓ corresponds to an eigenvector $\mathbf{v}_\ell$, which means knowing the location of an eigenvalue gives some indication of the characteristic behavior of an eigenvector. In [34], on the other hand, eigenvectors and linear Fourier weights are related by an inverse matrix, e.g. $\mathbf{F} = \mathbf{V}^{-1}$, which means we can also relate every eigenvalue λ_ℓ to the linear Fourier weights $\mathbf{f}_\ell$. Using this information, we plotted and analyzed the eigenvalues for different graph structures for $N = 5$ and $N = 6$ in order to compare an odd-sized graph with an even-sized graph.

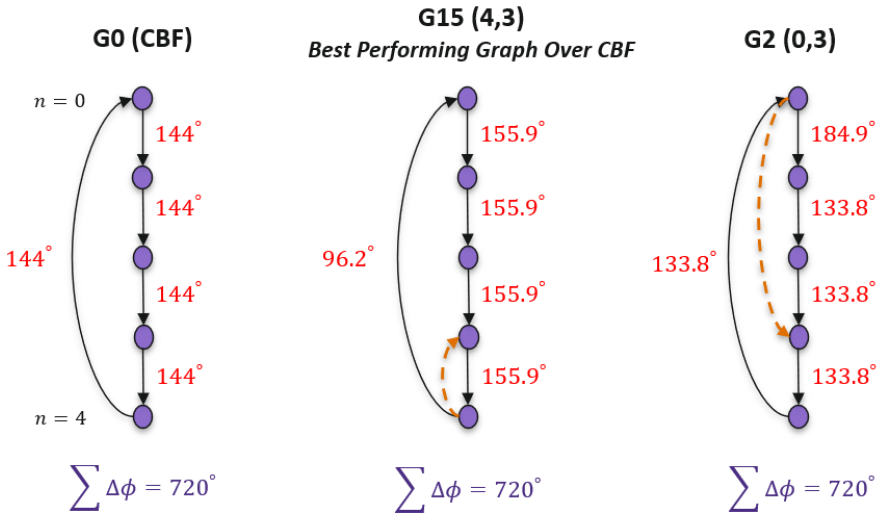
7.3.2 Applying Knowledge To Explain Effect of Graph Size

When we calculated the difference in phase $\Delta\phi(n_1, n_2)$ for the linear Fourier weights matched to end-fire, we observed several patterns that distinguish even and odd sized graphs (Figure 7.6). First, the total in differences in phase was constant despite an added edge. For even-sized graph $N = 6$, the total in differences in phase was 1080°, while for odd-sized graph $N = 5$, the total was lower with 720°. The second observation we made was on the values of $\Delta\phi(n_1, n_2)$ themselves. The values were constant at 180° for all even-sized graphs we analyzed $N = 6$, but the values varied for $N = 5$. When $N = 5$, the differences in phase were only constant-valued for the graph associated with the CBF. The graphs with an added edge, on the other hand, had one unique phase difference while the other four were constant in value. These results indicate a reason why performance improved over the CBF with odd-sized graphs. They had the ability to modify the phase values in their linear weights with the added edge, whereas even-sized graphs maintained their phase values with the CBF.

The effect of the graph size, e.g. even versus odd, can also be explained by plotting

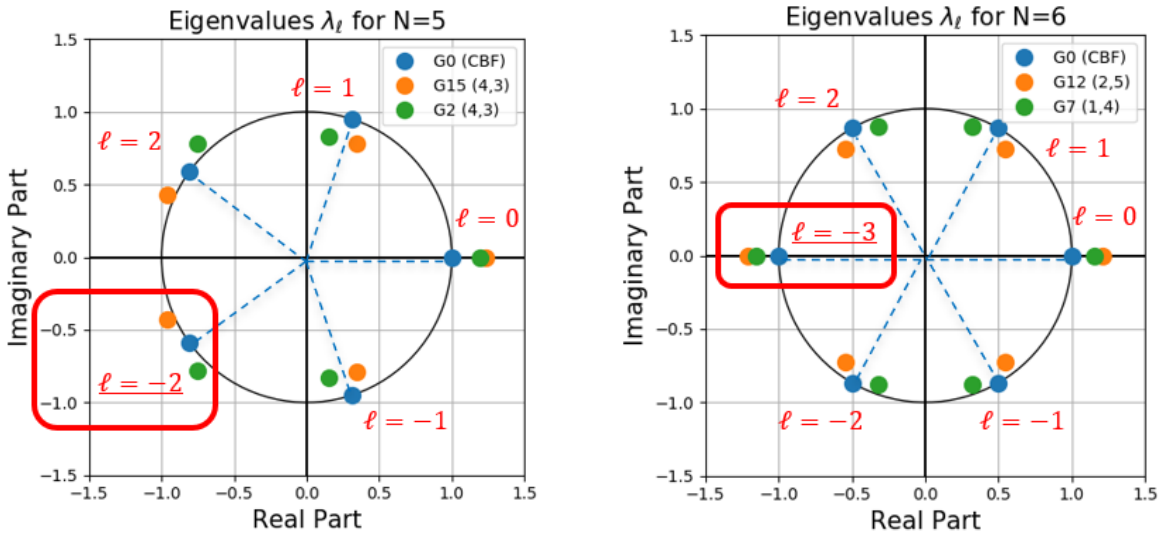


(a) $N = 6, \theta = -75°$



(b) $N = 5, \theta = -65°$

Figure 7.6: Phase difference of linear Fourier weights $\mathbf{f}_\ell$ matched to end-fire direction when (a) $N = 6$ and (b) $N = 5$



(a) $N = 5$ (b) $N = 6$

Figure 7.7: Comparing amount of phase alignment, between input steering vector and linear Fourier weights, for the CBF and the top two best performing new graph structures. Hermitian angles are calculated for the case when $N = 5$ and when input directed from (a) end-fire and (b) broadside.

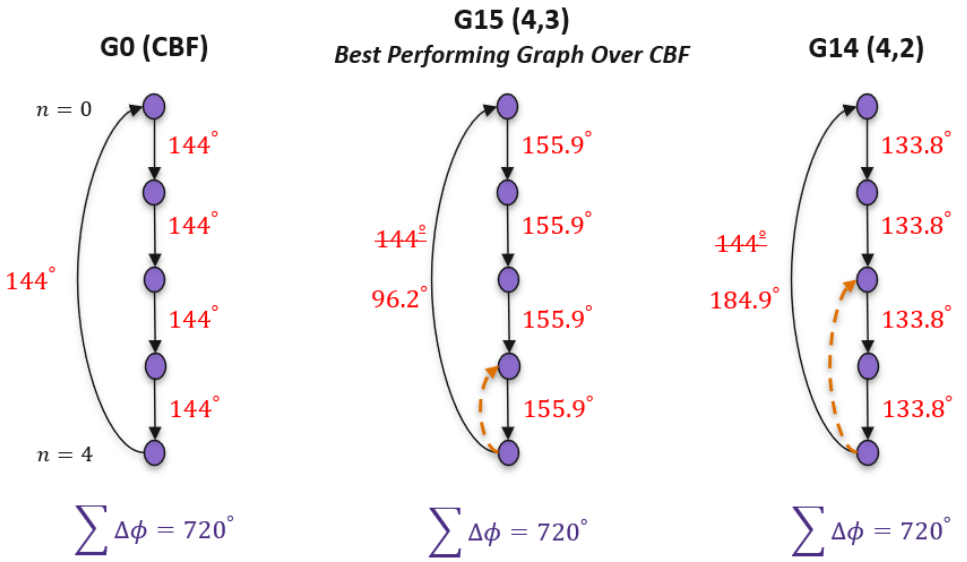
the eigenvalues of the graph's adjacency matrix. When we plotted the eigenvalues for the graph associated with the CBF and the two top-performing graphs for $N = 5$ and $N = 6$ (Figure 7.7), we found properties distinguishing odd from even-sized graphs, similar to what we found with differences in phase. When $N = 5$, only the eigenvalues at $\ell = 0$ are aligned in phase on the positive real axis. The eigevenvalues at the other locations, particularly at the bins matched to end-fire direction like $\ell = -2$, varied in phase. For the case of $N = 6$, on the other hand, we found two frequency bins, instead of one, for which eigenvalues from all graphs are aligned in phase. One is at $\ell = 0$ and the other is at $\ell = 3$, corresponding to the end-fire direction. Because the eigenvalues at $\ell = 3$ for the new graphs did not change in phase over the CBF, and because eigenvalues correspond to linear Fourier weights, this

lack of change in phase explains why even-sized graphs showed no performance improvement over the CBF. The lack of change in phase also aligns with our results in the last paragraph and explains why the differences in phase were constant-valued for new graphs even when an edge was added.

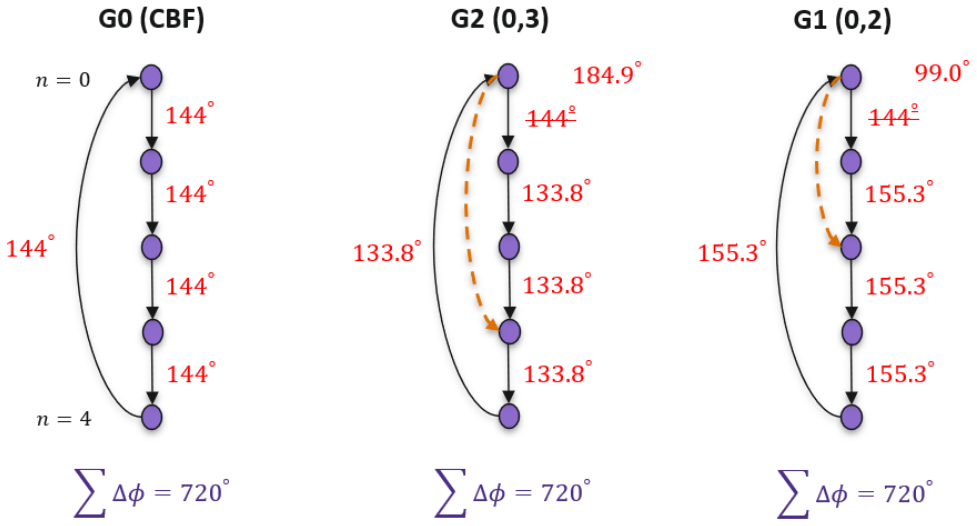
7.3.3 Applying Knowledge To Explain Effect of Added Edge

When we analyzed the difference in phase for linear weights matched to end-fire for odd-sized graphs $N = 5$ (Figure 7.8), we observed that adding an edge redistributes the flow of phase over the graph associated the CBF. This can be seen with a closer look at the added edge from vertex $n = 4$ to $n = 3$ for graph G15. Originally, in graph G0, all the information from vertex $n = 4$ flowed to $n = 0$, but with the added edge from $n = 4$ to $n = 3$ in G15, some of the information that flowed from $n = 0$ now flows to $n = 3$. Hence, the modification in the difference in phase from 144° to 96.2°. This modification affects all the other locations because to maintain the constant total difference in phase, the value must change at all other locations. This pattern was the same for graph G14 when the edge was directed again from $n = 4$ but this time ended on vertex $n = 2$. The change in vertex location again caused the difference in phase from $n = 4$ to $n = 0$ to modify, this time to 184.9°. These results indicate that where to place the added edge is important. One location may cause the phase of the linear weights to be more aligned to the input steering vector, while another location may not.

When we repeated the experiment for edges directed from a vertex other than $n = 4$ and this to $n = 0$ (Figure 7.8b), our observations were consistent with those from the last paragraph. We notice a similar modification in phase information. Because an edge is coming out of vertex $n = 0$ in these examples, the flow that was once directed from $n = 0$ to $n = 1$ in graph G0 undergoes a change. For the case of graph G2, the difference in phase from $n = 0$ to $n = 1$ changed from 144° to 184.9°, while other differences in phase were also altered but they all had a constant value to maintain the same total differences in phase. When the edge was directed to a different location in graph G1, we again saw a change in the difference in



(a)



(b)

Figure 7.8: Changes in phase difference of linear Fourier weights $\mathbf{f}_\ell$ matched to end-fire direction when adding one directed edge for $N = 5$.

phase from $n = 0$ and $n = 1$. This time the difference was 99° rather than 184.9°, as it was for G2. While we have no formula yet to explain how the difference in phase relates to the location of the added edge, the patterns we observed in this section shows for the first time the impact of adding an edge to the linear Fourier weights. These results show that adding an edge impacts the phase components of the linear weights. Furthermore, depending on where the edge is directed to and from, a certain added edge can modify the difference in phase enough so that there is more phase alignment between the linear Fourier weights and the input steering vector.

7.4 Improvements in RMSE at Higher SNRs

7.4.1 Supporting Theory

To explain the improvement in RMSE at end-fire for certain graph structures at higher SNRs, we relied on models, quantities, and theorems from probability theory and statistics. These include:

- Law of Total Expectations [51]:

$$\mathrm{E}[X] = \sum_i \mathrm{E}[X \mid A_i] \Pr\{A_i\}$$

- Normal Distribution [51]:

$$X \sim \mathcal{N}\left(\mu_X, \sigma_X^2\right)$$

- Complex Normal Distribution [51]:

$$X \sim \mathcal{CN}\left(\mu_X, \sigma_X^2\right)$$

- Property for Variance of Linear Combination of Complex Random Variables [51]: Assuming $X = \sum_i a_i X_i$

$$\mathrm{Var}(X) = \sum_i \sum_j a_i a_j^* \, \mathrm{Cov}[X_i, X_j]$$

- Noncentral Chi-Squared Distribution, $\chi_k^{'2}$ [52]: Assuming $(X_1, X_2, \ldots, X_k)$ are k independent normally distributed random variables with unit variance in which $X_i \sim \mathcal{N}(\mu_{X_i}, 1)$

$$X = \sum_{i=1}^{k} X_i^2 \sim \chi_k^{'2}(\lambda)$$
$$\lambda = \sum_{i=1}^{k} \mu_{X_i}^2$$

  in which k represents the degrees of freedom and λ represents the noncentrality parameter

- Generalized Chi-Squared Distribution, Q [53]: Assuming $X_i \sim \chi_{k_i}^{'2}(\lambda_i)$ are independent noncentral chi-squared random variables and random variable X is a linear combination of X_i with $X = \sum_i \alpha_i X_i$:

$$X \sim Q(\lambda_1, \lambda_2, \ldots)$$
$$\mathrm{E}[X] = \sum_i \alpha_i (k_i + \lambda_i)$$
$$\mathrm{Var}(X) = 2 \sum_i \alpha_i^2 (k_i + 2\lambda_i)$$

To set up the explanation in the upcoming section, we started by rewriting the RMSE in terms of the probability of estimating a frequency bin. Suppose a source is transmitted from direction $\theta = \theta_o$, which is matched to frequency bin ℓ_o. Let the bin-to-angle conversion be $\hat{\theta} = \Phi(\hat{\ell})$ and let the probability of estimating a frequency bin be $\Pr(\hat{\ell})$. Using this information, the RMSE for direction angle θ_o can be rewritten as follows:

$$\mathrm{RMSE}_{\theta_o} = \sqrt{\sum_{\hat{\ell} \neq \ell_o} \Big(\Phi(\hat{\ell}) - \theta_o\Big)^2 \Pr(\hat{\ell})} \tag{7.8}$$

Proof. We already know that, at $\theta = \theta_o$, the RMSE is defined as $\mathrm{RMSE}_{\theta_o} = \sqrt{\mathrm{E}[(\hat{\theta} - \theta_o)^2]}$. Applying the law of total expectations, we can rewrite the RMSE as:

$$\mathrm{RMSE}_{\theta_o} = \sqrt{\sum_{\hat{\ell}} \mathrm{E}\left[(\hat{\theta} - \theta_o)^2 \mid \hat{\ell}\right] \Pr(\hat{\ell})}$$

Using the bin-to-angle conversion function for $\hat{\theta}$ and excluding the case $\hat{\ell} = \ell_o$ since that does not contribute to any error, we get:

$$\begin{aligned}
\text{RMSE}_{\theta_o} &= \sqrt{\sum_{\hat{\ell}} \text{E}\left[(\Phi(\hat{\ell}) - \theta_o)^2 \mid \hat{\ell}\right] \Pr(\hat{\ell})} \\
&= \sqrt{\sum_{\hat{\ell}} \left(\Phi(\hat{\ell}) - \theta_o\right)^2 \Pr(\hat{\ell})} \\
&= \sqrt{\sum_{\hat{\ell} \neq \ell_o} \left(\Phi(\hat{\ell}) - \theta_o\right)^2 \Pr(\hat{\ell})}
\end{aligned}$$

If we let the probability of estimation error be defined as $P[E] = \Pr(\hat{\ell} \neq \ell_o)$, then we have:

$$\text{RMSE}_{\theta_o} = \sqrt{\sum_{\hat{\ell} \neq \ell_o} \left(\Phi(\hat{\ell}) - \theta_o\right)^2 P[E]}$$

□

This equation is important because we can see that RMSE is directly affected by the behavior of the error probability $P[E]$. If we can describe the behavior of $P[E]$ stochastically, then we can develop an explanation for the improvement in RMSE.

7.4.2 Applying Knowledge To Describe Behavior Of Error Probability

Step #1: Model for Power Spectrum

When we applied knowledge from probability theory and statistics, we found that the power spectrum P_ℓ can be modeled as a probability distribution. Let the noise variance for our plane wave model be $\sigma_w^2 = \frac{1}{SNR}$. If we scale the power spectrum P_ℓ by half the noise variance $0.5\sigma_w^2$, we can model the scaled power spectrum Z_ℓ as a noncentral chi-squared random variable with degree of freedom $k = 2$ and noncentrality parameter in terms of source power spectrum $P_{\ell,s} = \frac{|\hat{s}_\ell|^2}{\|f_\ell\|^2}$:

$$\begin{aligned}
Z_\ell &= \frac{P_\ell}{0.5\sigma_w^2} \sim \chi_2'^2(\lambda_\ell) \\
\text{where } \lambda_\ell &= \frac{P_{\ell,s}}{0.5\sigma_w^2}
\end{aligned} \tag{7.9}$$

Proof. The scaling factor can be determined by analyzing the distributions of the real and imaginary components of the unscaled power spectrum P_ℓ. From Equation 7.4), we can derive the distributions of $\hat{x}_{\ell,R}$ and $\hat{x}_{\ell,I}$ from the distribution of the complex-valued $\hat{x}_\ell$. Since $\hat{x}_\ell = \mathbf{f}_\ell \mathbf{x}$ and $\mathbf{x} \sim \mathcal{CN}(\mathbf{s}, \sigma_w^2 \mathbf{I})$, we have the following:

$$\begin{aligned}
\hat{x}_\ell &= \mathbf{f}_\ell(\mathbf{s} + \mathbf{w}) \\
\mathrm{E}[\hat{x}_\ell] &= \mathbf{f}_\ell \mathbf{s} = \hat{s}_\ell \\
\mathrm{Var}(\hat{x}_\ell) &= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f_\ell(n) f_\ell^*(m) \,\mathrm{Cov}[f_\ell(n), f_\ell(m)] = \sum_{n=0}^{N-1} |\mathbf{f}_\ell(n)|^2 \sigma_w^2 = \sigma_w^2 \|\mathbf{f}_\ell\|^2 \\
\therefore\ \hat{x}_\ell &\sim \mathcal{CN}(\hat{s}_\ell, \sigma_w^2 \|\mathbf{f}_\ell\|^2)
\end{aligned}$$

The distributions of $\hat{x}_{\ell,R}$ and $\hat{x}_{\ell,I}$ are therefore real-valued normal with the expected values equal to the respected real and imaginary parts of $\hat{s}_\ell$ and variance equal to half the total variance of complex-valued $\hat{x}_\ell$

$$\begin{aligned}
\hat{x}_{\ell,R} &\sim \mathcal{N}(\hat{s}_{\ell,R}, 0.5\sigma_w^2 \|\mathbf{f}_\ell\|^2) \\
\hat{x}_{\ell,I} &\sim \mathcal{N}(\hat{s}_{\ell,I}, 0.5\sigma_w^2 \|\mathbf{f}_\ell\|^2)
\end{aligned}$$

The distributions of $\hat{x}_{\ell,R}/\|\mathbf{f}_\ell\|$ and $\hat{x}_{\ell,I}/\|\mathbf{f}_\ell\|$ are therefore:

$$\begin{aligned}
\frac{\hat{x}_{\ell,R}}{\|\mathbf{f}_\ell\|} &\sim \mathcal{N}\left(\frac{\hat{s}_{\ell,R}}{\|\mathbf{f}_\ell\|}, 0.5\sigma_w^2\right) \\
\frac{\hat{x}_{\ell,I}}{\|\mathbf{f}_\ell\|} &\sim \mathcal{N}\left(\frac{\hat{s}_{\ell,I}}{\|\mathbf{f}_\ell\|}, 0.5\sigma_w^2\right)
\end{aligned}$$

In order to use the noncentral chi-squared distribution, we need to have unit variances. We can obtain unit variances by scaling the distributions above by $\sqrt{0.5\sigma_w^2}$:

$$\begin{aligned}
\frac{\hat{x}_{\ell,R}}{\sqrt{0.5\sigma_w^2}\|\mathbf{f}_\ell\|} &\sim \mathcal{N}\left(\frac{\hat{s}_{\ell,R}}{\sqrt{0.5\sigma_w^2}\|\mathbf{f}_\ell\|}, 1\right) \\
\frac{\hat{x}_{\ell,I}}{\sqrt{0.5\sigma_w^2}\|\mathbf{f}_\ell\|} &\sim \mathcal{N}\left(\frac{\hat{s}_{\ell,I}}{\sqrt{0.5\sigma_w^2}\|\mathbf{f}_\ell\|}, 1\right)
\end{aligned}$$

By squaring and these two random variables, we obtain what we have defined as the scaled power spectrum Z_ℓ, which has a relationship to P_ℓ as follows:

$$Z_\ell = \frac{\hat{x}_{\ell,R}^2}{0.5\sigma_w^2\|f_\ell\|^2} + \frac{\hat{x}_{\ell,I}^2}{0.5\sigma_w^2\|f_\ell\|^2} = \frac{1}{0.5\sigma_w^2}\left(\frac{\hat{x}_{\ell,R}^2}{\|f_\ell\|^2} + \frac{\hat{x}_{\ell,I}^2}{\|f_\ell\|^2}\right) = \frac{P_\ell}{0.5\sigma_w^2}$$

Because Z_ℓ is the sum of two squared normally distributed random variables, we can model it as a noncentral chi-square with $k = 2$ degrees of freedom and the following noncentrality parameter:

$$Z_\ell \sim \chi_2^{'2}(\lambda_\ell)$$
$$\lambda_\ell = \frac{\hat{s}_{\ell,R}^2}{0.5\sigma_w^2\|f_\ell\|^2} + \frac{\hat{s}_{\ell,I}^2}{0.5\sigma_w^2\|f_\ell\|^2} = \frac{|\hat{s}_\ell|^2}{0.5\sigma_w^2\|f_\ell\|^2} = \frac{P_{\ell,s}}{0.5\sigma_w^2}$$

□

Step #2: Estimation Probability In Terms Of Power Spectrum

When we applied knowledge from probability theory and statistics and our results from the last step, we found that the probability of estimating frequency bin $\hat{\ell}$ can be written in terms of the scaled power spectrum Z_ℓ:

$$\Pr(\hat{\ell}) = \Pr\left\{\max_{\ell\neq\hat{\ell}} Z_\ell \leq Z_{\hat{\ell}}\right\} \tag{7.10}$$

Proof. Probability of estimating bin $\hat{\ell}$ is derived from the results of the power spectrum. Our system will estimate frequency bin $\hat{\ell}$ if the power at $P_{\hat{\ell}}$ is greater than the power at all other frequency bins. This can be written mathematically as follows:

$$\Pr(\hat{\ell}) = \Pr\left\{P_{\hat{\ell}} \geq \max_{\ell\neq\hat{\ell}} P_\ell\right\}$$

Using the scaled power spectrum Z_ℓ does not affect the statement of probability:

$$\Pr(\hat{\ell}) = \Pr\left\{Z_{\hat{\ell}} \geq \max_{\ell\neq\hat{\ell}} Z_\ell\right\} = \Pr\left\{\max_{\ell\neq\hat{\ell}} Z_\ell \leq Z_{\hat{\ell}}\right\}$$

□

Step #3: Behavior of Error Probability In Terms Of Noise and Power Spectrum

Let the probability of estimation error be defined as $P[E] = \Pr(\hat{\ell} \neq \ell_o)$. When we applied knowledge from probability theory statistics and our results from the last step, we found

that error probability is upper bounded to the noise variance and the power spectrum of the source at matched bin ℓ_o and the estimated bin $\hat{\ell}$:

$$P[E] \leq \frac{0.5\sigma_w^2}{P_{\ell_o,s} - P_{\hat{\ell},s}} \text{ where } P_{\ell,s} \ = \frac{|\mathbf{f}_\ell \mathbf{a}(\theta_o)|^2}{\|\mathbf{f}_\ell\|^2} \tag{7.11}$$

Proof. Starting with the results from the last section and the law of total probability, we get a statement in terms of an expectation for random variable $Z_{\hat{\ell}}$:

$$\begin{aligned}
\Pr(\hat{\ell}) &= \Pr\left\{\max_{\ell \neq \hat{\ell}} Z_\ell \leq Z_{\hat{\ell}}\right\} = \int_0^\infty \Pr\left\{\max_{\ell \neq \hat{\ell}} Z_\ell \leq z\right\} \Pr\left\{Z_{\hat{\ell}} = z\right\} dz \\
&= \int_0^\infty \Pr\left\{Z_0 \leq z, Z_1 \leq z, \dots\right\} \Pr\left\{Z_{\hat{\ell}} = z\right\} dz \\
&= \int_0^\infty \left[\prod_{\ell \neq \hat{\ell}} \Pr\left\{Z_\ell \leq z\right\}\right] \Pr\left\{Z_{\hat{\ell}} = z\right\} dz \\
&= \int_0^\infty \left[\prod_{\ell \neq \hat{\ell}} F_{Z_\ell}(z)\right] f_{Z_{\hat{\ell}}}(z) dz = \mathrm{E}_{Z_{\hat{\ell}}}\left[\prod_{\ell \neq \hat{\ell}} F_{Z_\ell}(z)\right]
\end{aligned}$$

Since random variables $Z_{\hat{\ell}}$ are independent, we can pull out the component Z_{ℓ_o} that corresponds to the true direction θ_o as separate expectation and upper bound the estimation probability to that expectation:

$$\begin{aligned}
&\Pr(\hat{\ell}) = \mathrm{E}_{Z_{\hat{\ell}}}\left[F_{Z_{\ell_o}}(z) \prod_{\ell \neq \hat{\ell}, \ell \neq \ell_o} F_{Z_\ell}(z)\right] = \mathrm{E}_{Z_{\hat{\ell}}}\left[F_{Z_{\ell_o}}(z)\right] \mathrm{E}_{Z_{\hat{\ell}}}\left[\prod_{\ell \neq \hat{\ell}, \ell \neq \ell_o} F_{Z_\ell}(z)\right] \\
&\Pr(\hat{\ell}) \leq \mathrm{E}_{Z_{\hat{\ell}}}\left[F_{Z_{\ell_o}}(z)\right]
\end{aligned}$$

If we let the probability of estimation error be defined as $P[E] = \Pr(\hat{\ell} \neq \ell_o)$, then $\Pr(\hat{\ell})$ can be replaced with $P[E]$:

$$P[E] \leq \mathrm{E}_{Z_{\hat{\ell}}}\left[F_{Z_{\ell_o}}(z)\right]$$

We can convert this expectation statement back to a probability statement and rewrite it as a linear combination of noncentral chi-squared random variables, which can then be modeled

as a generalized chi-squared random variable Q:

$$P[E] \leq \mathrm{E}_{Z_{\hat{\ell}}}\left[F_{Z_{\ell_o}}(z)\right] = \Pr\left\{Z_{\ell_o} \leq Z_{\hat{\ell}}\right\} = \Pr\left\{Z_{\ell_o} - Z_{\hat{\ell}} \leq 0\right\}$$
$$P[E] \leq F_Q(0) \text{ where } Z_{\ell_o} - Z_{\hat{\ell}} \sim \mathrm{Q}\left(\lambda_{\ell_o}, \lambda_{\hat{\ell}}\right)$$

For a generalized chi-squared random variable Q, the mean is given by the following:

$$\mathrm{E}(Q) = (2 + \lambda_{\ell_o}) - (2 + \lambda_{\hat{\ell}}) = \lambda_{\ell_o} - \lambda_{\hat{\ell}} = \frac{P_{\ell_o,s} - P_{\hat{\ell},s}}{0.5\sigma_w^2}$$
$$\text{where } P_{\ell,s} = \frac{|\mathbf{f}_\ell \mathbf{a}(\theta_o)|^2}{\|f_\ell\|^2}$$

The value of the expectation of random Q, $\mathrm{E}(Q)$, will have an impact on the value of $F_Q(0)$. Since the CDF is a non-decreasing continuous function with range from 0 to 1, then as the mean goes up, this causes the value of $F_Q(0)$ to go down. This means there is an inverse relationship between $F_Q(0)$ and $\mathrm{E}(Q)$. It follows from this that $\Pr(\hat{\ell})$ and $\mathrm{E}(Q)$ are inversely proportional as well:

$$P[E] \leq \frac{1}{\mathrm{E}(Q)}$$
$$\therefore \; P[E] \leq \frac{0.5\sigma_w^2}{P_{\ell_o,s} - P_{\hat{\ell},s}} \text{ where } P_{\ell,s} = \frac{|\mathbf{f}_\ell \mathbf{a}(\theta_o)|^2}{\|f_\ell\|^2}$$

□

7.4.3 Applying Knowledge to Explain RMSE Decrease

Using knowledge of RMSE as a function of estimation error probability (Equation 7.8) and the upper bound of the estimation error probability (Equation 7.11), an explanation for the RMSE decrease at high SNR can be developed. High SNR corresponds to lower noise variance σ_w^2, lowering the upper bound for the error probability $P[E]$. Since RMSE is a function of $P[E]$, it follows that a reduction in $P[E]$ from high SNR, therefore, results in a decrease in RMSE.

This knowledge is consistent with our explanations in the last sections tying phase alignment to performance improvement. The upper bound to estimation error probability indicates that estimation probability can be reduced not only by lowering the amount of noise,

but also by increasing the difference between the source power spectrum at the matched bin ℓ_o and the mismatched bin $\hat{\ell}$. The difference is dependent on the phase alignment behavior between linear weights $\mathbf{f}_\ell$ and steering vector $\mathbf{a}(\theta)$. If the amount of phase alignment, as measured by the Hermitian angle, is 0° at the matched bin ℓ_o and 90° at the mismatched bins, then this difference between source power spectrum is at its maximum, thereby lowering the error probability. The reason why some graph structures have improved estimation accuracy at end-fire is because they have improved phase alignment behavior over the CBF, which decreases the probabilities of estimation errors at the mismatched bins $\hat{\ell} \neq \ell_o$ and hence reduces the RMSE.

7.5 Summary

In summary, explanations were developed to understand our system's increase in estimation accuracy at end-fire, odd-sized arrays, and measured data with high SNR. Our results in this chapter showed that estimation performance improvement is indeed explainable and that estimation performance improvement is tied most directly to phase alignment. As long as a graph structure possesses linear Fourier weights that have improved phase alignment to the input measured data, our system can reduce the RMSE at end-fire over the conventional beamformer. This connection was highlighted consistently over all the explanations we developed using array signal processing, spectral graph theory, and probability theory and statistics. These results, in the end, are important because they give us additional confidence in our system design and the evidence we collected on how it performs.

Chapter 8

CONCLUSION

Signal processing on graphs (SPG) possesses numerous principles and tools that are promising for single-snapshot direction-of-arrival (DOA) estimation. When we applied SPG principles and tools to develop a new estimation system, based on the conventional beamformer (CBF), three key results emerged. First, a new graph-based DOA estimation system was developed that both replicated and generalized the CBF. Second, an evaluation of performance using both real and simulated data provided evidence that, under certain conditions, our system does indeed perform better than the CBF. Third, and finally, explanations were developed showing that both the limitations and advantages of our system are directly tied to the flexibility and higher quality, in phase, of linear Fourier weights that certain graph structures possess over the fixed DFT linear weights used by the CBF. On the whole, these key outcomes address the single question we had for this dissertation: can we improve the performance of single-snapshot DOA estimation systems using SPG? These results studying the CBF show that, yes with certain conditions, SPG can be adopted to increase estimation accuracy. Furthermore, beyond just showing that estimation accuracy is possible, this work made three key contributions to the body of work in single-snapshot DOA estimation.

*Contribution #1: Knowledge of **how** to estimate DOA using input graph signals*

With our work, we now know *how* to estimate DOA using inputs that are represented as a graph signal. The system diagram we developed in Chapter 3 shows us exactly how to take data measured from a uniform line array (ULA) and, with a graph input selection, take that data all the way through an algorithm to an estimate of DOA. From Chapter 7, we even developed knowledge of which graphs to pick to improve performance. While more work

needs to be done to prove the relationship between added edge location and the adjustment of the phase component in the linear weights, our work lays the foundation for that by providing evidence of certain patterns. These patterns include using odd-sized graphs and edges directed out of the last graph vertex.

*Contribution #2: Knowledge of **why** our estimation system and tools from SPG are valuable*

With our work, we now know *why* our system and SPG are valuable as approaches to DOA estimation. The results from Chapter 5 and 6 using real and simulated data, respectively, show that our system approach to DOA estimation should be used. Numerous examples of increased estimation accuracy over the CBF are given, all quantified by a measure of RMSE percentage improvement over the conventional method to add confidence in our graph-based system design. Confidence is further increased with the results from Chapter 7. The explanations developed for performance improvement, based on probability theory and statistics, array signal processing, and graph theory, to name a few, help to alleviate doubts on the applicability of SPG. By showing that SPG can improve the the quality of the linear Fourier weights and how that improvement in quality is tied to gains in estimation accuracy, we have provided a clear reason why SPG should be used for DOA estimation.

*Contribution #3: Knowledge of **when** and **where** to use our estimation system in application*

With our work, we now know *when* and *where* to apply our estimation system. The results from Chapter 5 and 6 show that, because our system performs than the CBF when the input is directed from end-fire and is received with high SNR, our system should be applied in those scenarios. These defined set of conditions are only further supported by the explanations we developed in Chapter 7. But even if these conditions are not met, the flexibility offered by our system diagram in Chapter 3 allows our system to adapt to different scenarios simply by modifying the input graph structure. While more work needs to be done to figure out how to adaptively modify the graph, the outcomes from this dissertation will help to enable this

development. Not only do we have the system structure to change input graph structures, but we also have knowledge of which graph structure to use for different DOA directions.

Opportunities for Future Work

While opportunities for future was not listed as a contribution, one may argue that it can be. This is because, in this work, we assumed a single stationary and narrowband source, a uniform line array, input data from a single snapshot in time, and 1D graphs structures with different edge configurations. Opportunities for future work lie in the extension of these stated assumptions.

We can build on this work by considering other receiver array or graph structure configurations. For instance, instead of a ULA, it would be interesting to see how this work extends to a non-uniform line array. Will our setup using unweighted edges still work? What will the performance be like? In addition to using added edges, this work also lends itself naturally to studying graphs with weighted edges to characterize both existence and strength of a connection between two pairs of vertices. What edge weights should we use? Will certain edge weights lead to performance improvement over the CBF? Some example rules for edge weights from [6, 54] can assist with this investigation.

We can also consider other extensions for the sound source. For instance, we can consider multiple sound sources instead of just one. This assumptions which lends itself naturally to opportunities to study and incorporate other performance metrics like the beampattern [39]. Instead of stationary source, a moving source may also be considered. Our work using single snapshots of data is well-suited for moving sources because we can continue to estimate DOA from a single snapshot, but now we can keep track of the DOA estimates over time as the source moves.

Another possible idea is to try out broadband source signals, which lends itself to interesting extensions of our work. For broadband source signals, we may have to start considering representing an input consisting of multiple snapshots of data. With multiple snapshots, this means the work would have to be broadened beyond 1D graphs to 2D graphs. Which ever

of these opportunities is considered, the work we have done here is significant. It serves as a platform to conduct these extended studies, and our results can also be used as a baseline for performance comparison. Some work in 2D graphs in [55] would be useful to consider.

Finally, for extensions in the theoretical aspects of our work, we offer a few suggestions. To improve explanations on the impact of added edges, the emerging publications in directed graphs like [49] would be worthy to leverage. In addition, recent work in perturbation theory on directed graphs in [56] would interesting to explore to the change from a directed cycle to another graph configuration. Which ever one of these avenues one ultimately chooses to explore, the work in this dissertation lays out a clear plan to make these and other contributions in single-snapshot DOA estimation.

Final Remarks

In conclusion, key results from this dissertation produced three important contributions to the body of work in single-snapshot DOA estimation. We now have knowledge of how, why, when, and where DOA can be estimated with the aid of flexible graphs as an additional system input. Not only does our graph-based system lead to improved estimation accuracy, under certain conditions, but it also lays the foundation to study other performance enhancements in the future. Furthermore, the work in this dissertation also continues to address the larger problem of why sensor arrays are used – to strengthen our understanding of what is around us. Our system enables this enhancement by using the framework of SPG and graphs as a flexible system input. While there is still room for improvement, our work offers a pathway for sensor array systems to continue its history of increasing our situational awareness and make us more informed decisionmakers in the end.

BIBLIOGRAPHY

[1] Hamid Krim and Mats Viberg. Two decades of array signal processing research. *IEEE Signal Processing Magazine*, 1996.

[2] Patrick Häcker and B Yang. Single snapshot DOA estimation. *Advances in Radio Science*, 8, 2010.

[3] Liang Tao and HK Kwan. A novel approach to fast DOA estimation of multiple spatial narrowband signals. In *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, volume 1, pages 1–431. IEEE, 2002.

[4] Stefano Fortunati, Raffaele Grasso, Fulvio Gini, Maria S Greco, and Kevin LePage. Single-snapshot DOA estimation by using compressed sensing. *EURASIP Journal on Advances in Signal Processing*, 2014(1):120, 2014.

[5] Elias Aboutanios, Aboulnasr Hassanien, Moeness G Amin, and Abdelhak M Zoubir. Fast iterative interpolated beamforming for accurate single-snapshot DOA estimation. *IEEE Geoscience and Remote Sensing Letters*, 14(4):574–578, 2017.

[6] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

[7] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[8] Bill M Radich and Kevin M Buckley. Single-snapshot DOA estimation and source number detection. *IEEE Signal processing letters*, 4(4):109–111, 1997.

[9] IEEE. Emerging trends, issues and challenges for array signal processing and its applications in smart city, Jan 2020.

[10] Aboulnasr Hassanien, Moeness G Amin, Yimin D Zhang, and Fauzia Ahmad. Capon-based single-snapshot DOA estimation in monostatic mimo radar. In *Mobile Multimedia/Image Processing, Security, and Applications 2015*, volume 9497, page 94970G. International Society for Optics and Photonics, 2015.

[11] Christoph Degen. On single snapshot direction-of-arrival estimation. In *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pages 92–97. IEEE, 2017.

[12] Han Cui, Tong Liu, and Wenjuan Peng. Single-snapshot DOA estimation for uniform linear array. In *2015 IEEE International Conference on Information and Automation*, pages 2905–2909. IEEE, 2015.

[13] Nuri Yilmazer, Tapan K Sarkar, and Magdalena Salazar-Palma. DOA estimation using matrix pencil and esprit methods using single and multiple snapshots. In *2010 URSI International Symposium on Electromagnetic Theory*, pages 215–218. IEEE, 2010.

[14] Arpita Thakre, Martin Haardt, and Krishnamurthy Giridhar. Single snapshot spatial smoothing with improved effective array aperture. *IEEE Signal Processing Letters*, 16(6):505–508, 2009.

[15] Yanan Ma, Xianbin Cao, and Xiangrong Wang. Off-grid DOA estimation with arbitrary-spaced linear array using single snapshot. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2019.

[16] Yuntao Wu, Xiaobing Pei, and Hing Cheung So. Utilizing principal singular vectors for 2D DOA estimation in single snapshot case with uniform rectangular array. *International Journal of Antennas and Propagation*, 2015, 2015.

[17] Anupama Govinda Raj and James H McClellan. Single snapshot super-resolution DOA estimation for arbitrary array geometries. *IEEE Signal Processing Letters*, 26(1):119–123, 2018.

[18] Ahmet M Elbir and T Engin Tuncer. Single snapshot DOA estimation in the presence of mutual coupling for arbitrary array structures. In *2016 IEEE sensor array and multichannel signal processing workshop (SAM)*, pages 1–5. IEEE, 2016.

[19] Ireneusz Jabłoński. Graph signal processing in applications to sensor networks, smart grids, and smart cities. *IEEE Sensors Journal*, 17(23):7659–7666, 2017.

[20] Weiyu Huang, Leah Goldsberry, Nicholas F Wymbs, Scott T Grafton, Danielle S Bassett, and Alejandro Ribeiro. Graph frequency analysis of brain signals. *J. Sel. Topics Signal Processing*, 10(7):1189–1203, 2016.

[21] Divyanshu Srivastava and Vibhor Kumar. *Graph signal processing based analysis of biological networks*. PhD thesis, IIIT-D, 2018.

[22] Eldridge Alcantara, Les Atlas, and Shima Abadi. Applying concepts and tools from signal processing on graphs (SPG) to problems in array signal processing. *The Journal of the Acoustical Society of America*, 143(3):1852–1852, 2018.

[23] Eldridge Alcantara, Les E. Atlas, and Shima H. Abadi. Graph signal smoothness for direction-of-arrival estimation of source targets using non-uniform line arrays. *Journal of the Acoustical Society of America*, 146:2886–2886, 2019.

[24] Gongping Huang, Jacob Benesty, Israel Cohen, and Jingdong Chen. Differential beamforming on graphs. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:901–913, 2020.

[25] Leandro AS Moreira, António LL Ramos, Marcello LR de Campos, José A Apolinário, and Felipe G Serrenho. A graph signal processing approach to direction of arrival estimation. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[26] Ian K Proudler, Vladimir Stankovic, and Stephan Weiss. Narrowband angle of arrival estimation exploiting graph topology and graph signals. In *International Conference on Sensor Signal Processing for Defence*, 2020.

[27] D.H. Johnson and D.E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Prentice-Hall signal processing series. P T R Prentice Hall, 1993.

[28] Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013.

[29] S Najeem, K Kiran, A Malarkodi, and G Latha. Open lake experiment for direction of arrival estimation using acoustic vector sensor array. *Applied Acoustics*, 119:94–100, 2017.

[30] Jason D Holmes, Edmund J Sullivan, and William M Carey. Passive synthetic aperture processing with an autonomous underwater vehicle towed hydrophone array. In *OCEANS 2006*, pages 1–5. IEEE, 2006.

[31] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[32] Henry Cox. Resolving power and sensitivity to mismatch of optimum array processors. *The Journal of the Acoustical Society of America*, 54(3):771–785, 1973.

[33] MEGD Colin, J Groen, and BAJ Quesson. *Experimental comparison of bearing estimation techniques for short passive towed sonar arrays*, volume 2. IEEE, 2004.

[34] Aliaksei Sandryhaila and Jose MF Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, 2014.

[35] Joya A Deri and José MF Moura. Taxi data in new york city: A network perspective. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pages 1829–1833. IEEE, 2015.

[36] Arun Venkitaraman, Saikat Chatterjee, and P Handel. On hilbert transform of signals on graphs. *Proc. Sampling Theory Appl*, 2015.

[37] Ljubisa Stankovic, Danilo Mandic, Milos Dakovic, Milos Brajovic, Bruno Scalzo, and Anthony G Constantinides. Graph signal processing–part II: Processing and analyzing signals on graphs. *arXiv preprint arXiv:1909.10325*, 2019.

[38] Andries E Brouwer and Willem H Haemers. *Spectra of graphs*. Springer Science & Business Media, 2011.

[39] Harry L Van Trees. *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[40] J.R. Taylor and S.L.L.J.R. Taylor. *Introduction To Error Analysis: The Study of Uncertainties in Physical Measurements*. ASMSU/Spartans.4.Spartans Textbook. University Science Books, 1997.

[41] Shahriar Shirvani-Moghaddam and Sakineh Almasi-Monfared. A comprehensive performance study of narrowband DOA estimation algorithms. *International Journal on Communications Antenna and Propagation (IRECAP)*, 1(4), 2011.

[42] Menglu Xia, Daniel Rouseff, James A Ritcey, Xiang Zou, Chantri Polprasert, and Wen Xu. Underwater acoustic communication in a highly refractive environment using sc–fde. *IEEE journal of oceanic engineering*, 39(3):491–499, 2013.

[43] University of Washington Applied Physics Lab. *APL-UW High-Frequency Ocean Environmental Acoustic Models Handbook*. Defense Technical Information Center, 1994.

[44] Shima H Abadi, Daniel Rouseff, and David R Dowling. Blind deconvolution for robust signal estimation and approximate source localization. *The Journal of the Acoustical Society of America*, 131(4):2599–2610, 2012.

[45] Petre Stoica and Arye Nehorai. Performance study of conditional and unconditional direction-of-arrival estimation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1783–1795, 1990.

[46] Mikael Coldrey and Mats Viberg. Generalization and analysis of the conventional beamformer for localization of spatially distributed sources. In *2006 14th European Signal Processing Conference*, pages 1–5. IEEE, 2006.

[47] Barry D Van Veen and Kevin M Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988.

[48] Klaus Scharnhorst. Angles in complex vector spaces. *Acta Applicandae Mathematica*, 69(1):95–103, 2001.

[49] Antonio G Marques, Santiago Segarra, and Gonzalo Mateos. Signal processing on directed graphs: The role of edge directionality when processing and learning from network data. *IEEE Signal Processing Magazine*, 37(6):99–116, 2020.

[50] Rasoul Shafipour, Ali Khodabakhsh, Gonzalo Mateos, and Evdokia Nikolova. A digraph fourier transform with spread frequency components. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 583–587. IEEE, 2017.

[51] Alberto Leon-Garcia. *Probability and random processes for electrical engineering*. Pearson Education India, 1994.

[52] Norman L Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous univariate distributions, volume 2*, volume 289. John wiley & sons, 1995.

[53] Robert B Davies. Algorithm as 155: The distribution of a linear combination of χ^2 random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333, 1980.

[54] Hilmi E Egilmez and Antonio Ortega. Spectral anomaly detection using graph-based filtering for wireless sensor networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1085–1089. IEEE, 2014.

[55] Andreas Loukas and Damien Foucard. Frequency analysis of temporal graph signals. *arXiv preprint arXiv:1602.04434*, 2016.

[56] Bastian Seifert and Markus Püschel. Digraph signal processing with generalized boundary conditions. *arXiv preprint arXiv:2005.09762*, 2020.

[57] Richard Roy and Thomas Kailath. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on acoustics, speech, and signal processing*, 37(7):984–995, 1989.

[58] William Baxter and Elias Aboutanios. Fast direction-of-arrival estimation in coprime arrays. In *2018 International Conference on Radar (RADAR)*, pages 1–6. IEEE, 2018.

[59] Shahriar Shirvani-Moghaddam and Farida Akbari. A novel ula-based geometry for improving aoa estimation. *EURASIP Journal on Advances in Signal Processing*, 2011(1):39, 2011.

[60] Matthew Hawes, Lyudmila Mihaylova, Francois Septier, and Simon Godsill. A bayesian compressed sensing kalman filter for direction of arrival estimation. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 969–975. IEEE, 2015.

[61] Panayiotis Ioannides and Constantine A Balanis. Uniform circular arrays for smart antennas. *IEEE Antennas and propagation magazine*, 47(4):192–206, 2005.

[62] Abhranil Das, R Calen Walshe, and Wilson S Geisler. A new method to compute classification error. *Journal of Vision*, 19(10):87–87, 2019.

VITA

Eldridge Alcantara was born and raised in San Diego, California. He earned his B.S. in Electrical & Computer Engineering and M.Eng. in Electrical Engineering from the University of California, San Diego. Since 2009, he has been working as an Electronics Engineer at Naval Information Warfare Pacific (NIWC Pacific, formerly SSC Pacific). After four years of full-time work, he returned to graduate school in 2013 to pursue a Ph.D. in Electrical Engineering from the Department of Electrical & Computer Engineering (ECE) at the University of Washington, Seattle. He continued to work at NIWC Pacific during the summer breaks from school. During his Ph.D. program, he supplemented his research work with numerous teaching/instruction duties and activities in the ECE department, the Center for Teaching and Learning (CTL), and UW Recreation. Following graduation, he will return to his full-time job at NIWC Pacific.