Title
Homomorphic Directional Beamforming and Near-Field Localization with Analog True Time Delay Arrays

ISBN
9798190934786

Author
Pehlivan, Ibrahim

Publication Date
2026-09-11

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Homomorphic Directional Beamforming and Near-Field Localization with Analog True Time Delay Arrays

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Ibrahim Pehlivan

2026

ABSTRACT OF THE THESIS

Homomorphic Directional Beamforming and Near-Field Localization with Analog True Time Delay Arrays

by

Ibrahim Pehlivan
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2026
Professor Danijela Cabric, Chair

Future wireless applications continue the trend of increasingly demanding bandwidth, data rates, and connectivity, forcing systems to operate at higher frequencies to exploit the abundant bandwidth and with more antennas to exploit array gain and beamforming capabilities. However, the increasing array aperture and bandwidth start to challenge the established channel and array response assumptions, and these changing assumptions require both new array architectures and new algorithms suited to the changing channel characteristics. First, the frequency dependency of the array can no longer be ignored, requiring true-time-delay (TTD)-based array architectures to enable low-cost frequency-dependent control. Furthermore, as the array aperture grows, more users fall into the near-field region, complicating the beamforming design, since the user channel now depends on both angle and distance. This also complicates localization and beam training, as the search must now be performed over both distance and angle. TTD arrays, originally proposed to overcome the frequency dependency caused by the fixed antenna spacing, have also been utilized to realize split beampatterns which map subbands to different spatial regions, allowing the serving of

multiple users at different angles with only a single RF chain. However, current algorithms for split beampattern generation either require high computational complexity or memory, or cannot operate under frequency dependency, while heuristic models are restricted and reduce usability. In Chapter 2, we propose a fast and efficient split beampattern synthesis algorithm based on the mathematical structure of the split-beam synthesis problem, which requires only a fixed generator beampattern dictionary, resulting in a low-cost alternative that also provides fairness. However, the parameters of the proposed algorithm are not optimized for the average received power. In Chapter 3, we propose a data-driven parameter optimization approach for the proposed algorithm to maximize the average received power, and show that the optimized algorithm matches a low-complexity state-of-the-art benchmark algorithm while providing a fairer power distribution among subbands. Likewise, the near-field beam training problem is addressed in Chapter 4, where we show that, by utilizing TTD arrays and a special configuration called rainbow beams, we can virtually partition the array into far-field-operating sub-arrays, recover each sub-array's signal in post-processing, determine each sub-array's angle, and localize the user with triangulation. Therefore, this thesis provides strong solutions to the rising challenges of next-generation wireless systems by addressing both of the changing assumptions.

The thesis of Ibrahim Pehlivan is approved.

Christina Fragouli

Ian Roberts

Danijela Cabric, Committee Chair

University of California, Los Angeles

2026

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Chapters 2 and 3 are supported by NSF awards 1955672 and 2224322, and Chapter 4 is supported by NSF award 2224322 and Research Council of Finland grants 357730 and 359095.

Chapter 2 is a version of I. Pehlivan and D. Cabric, "Homomorphic Directional Beamforming with Analog True Time Delay Arrays," arXiv:2605.06930 [eess.SP], May 2026, in preparation for publication.

Chapter 3 is a version of I. Pehlivan and D. Cabric, "Homomorphic Directional Beamforming for Analog True-Time-Delay Arrays with Parameter Learning," to appear in *Proc. 29th International Workshop on Smart Antennas (WSA)*, Barcelona, Spain, Oct. 2026.

Chapter 4 is a version of I. Pehlivan, M. C. Ilter, M. Valkama, and D. Cabric, "Near-Field Beam Training with Analog Extremely Large True-Time-Delay Arrays (Invited Paper)," in *Proc. 58th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, USA, 2024, pp. 858–864, doi: 10.1109/IEEECONF60004.2024.10942936.

I would like to thank Aditya Wadaskar and Ding Zhao for their discussions on Chapter 2, Rushabha Balaji and Benjamin W. Domae for their discussions on Chapter 3, my co-authors Mehmet Cagri Ilter and Prof. Mikko Valkama for their contributions to Chapter 4, namely collaboration on idea development, help with its introduction, and feedback on the manuscript, my PI Danijela Cabric for her guidance, and CORES Lab members for their discussions and friendship.

I would like to thank my thesis committee, the chair Prof. Danijela Cabric, Prof. Ian Roberts, and Prof. Christina Fragouli, for their guidance. I also would like to thank the Vice Chair of the ECE Department, Prof. Sudhakar Pamarti, for his guidance and support.

Most importantly, I would like to thank my family, my wife Yuksel Ozle Pehlivan and my daughter Lidya Pehlivan, who will soon join us, for their support and love.

CHAPTER 1

Introduction

1.1 Motivation

The next generation of consumer wireless will need to support applications such as Virtual Reality [1], Immersive Extended Reality (XR) [2, 3], and ubiquitous connectivity [1], each of which demands higher data rates, lower latency, and greater connection density. Two responses dominate current research: moving to higher carrier frequencies for their abundant bandwidth, and scaling multi-antenna arrays to compensate for path loss while increasing spatial resources. Combining these two trends, however, changes both the channel characteristics and the frequency-dependent behavior of the array, invalidating assumptions that previously held. Realizing the envisioned applications in a feasible manner, therefore, requires both an analysis of these new channel characteristics and the development of new array architectures.

1.2 Background

Current multi-antenna systems operate in the narrowband regime with a small number of antennas, resulting in a frequency-independent channel response and, under the far-field assumption, an angle-dependent array response. However, as both bandwidth and the number of antennas increase, both of these assumptions begin to break down. First, with fixed antenna spacing, the beam direction deviates from the intended angle as the frequency moves away from the carrier, an effect known as beam squint [4]. Second, as the antenna aperture grows with the number of antennas, the Rayleigh distance, which demarcates the far-field region, also grows, so user equipments (UEs) increasingly fall into the near-field region, and the channel response becomes both direction and distance-dependent [5]. These effects motivate new antenna array architectures together with new algorithms and protocols to address the changing channel and system characteristics.

Multi-antenna systems can achieve frequency-dependent beamforming in baseband by utilizing digital arrays with a dedicated RF chain per antenna. However, this antenna architecture is infeasible for future applications that require massive numbers of antennas, resulting in excessive hardware cost and power consumption [6]. Analog arrays are therefore proposed as a low-cost alternative, using a single RF chain connected to the antennas through frequency-independent phase shifters [6]. However, the use of a single RF chain together with frequency-independent phase shifters limits the array to frequency-independent beamforming and reduces wideband beamforming gain as well as multiplexing capabilities. Recently, true-time-delay (TTD) arrays have been proposed to enable frequency-dependent analog beamforming, by inserting TTD elements between the single RF chain and the antennas [7]. These TTD elements introduce a frequency-dependent phase shift, which enables frequency-dependent beamforming.

The proposed TTD-based array architecture, with its frequency-dependent beamforming, can eliminate beam squint and increase the wideband array gain in far-field channels [4]. In addition, instead of eliminating beam squint, the TTD array can intentionally introduce controlled beam squint to generate far-field rainbow beams, probing multiple angles with unique subcarriers and enabling efficient far-field beam training [8]. Furthermore, by carefully choosing the TTD array parameters, different subbands can be steered towards different directions, enabling subband-to-direction multiplexing for multi-user communication with a single RF chain [9]. However, these applications are designed for far-field channels and the near-field regime requires separate application solutions.

In the near-field, the channel between the UE and the array antennas depends on both direction and radial distance. This new dependency enables spatial separation across location, enhancing spatial resources, reducing interference, and enabling UE localization [5,10]. However, since effective beamforming in the near-field now requires knowledge of the UE's full location rather than only direction, beam training is significantly more complex [11], motivating smart algorithms and protocols.

1.3 Thesis Contributions

Split beampatterns have been utilized to enhance uplink coverage [12], improve edge throughput [13], and meet low-latency requirements through flexible scheduling [14]. Although the wideband subband-to-direction assignment problem, known as split-beam synthesis, has been addressed by optimization-based methods [9,14] and by closed-form heuristics restricted in user direction and number [15, 16], existing solutions remain either too computationally expensive, too memory-intensive for flexible deployment, restricted to specific user configurations, or unable to handle beam squint. In Chapter 2, we develop a fast, efficient, and flexible far-field split-beam synthesis algorithm. Specifically, by exploiting the homomorphism between TTD array configuration matrices and their corresponding beampatterns, we develop the Homomorphic Directional Beamforming (HDB) algorithm that uses a single dictionary of simple two-direction split beampatterns to construct arbitrary multi-direction split beampatterns, in a divide-and-conquer manner through simple manipulation of the dictionary entries. Our results show that the proposed algorithm achieves average spectral efficiency (ASE) close to baselines such as [14] without ignoring beam squint, while also providing improved fairness across UEs. Chapter 2 is based on [17]. However, the proposed algorithm does not optimize the linear shift and scale parameters that act on the dictionary for the maximum average received power and utilizes fixed values; this limitation is addressed in Chapter 3.

The HDB algorithm utilizes the dictionary entries in a fixed manner, based on the assumption that the power across subcarriers for each dictionary entry is constant. Furthermore, we did not optimize the dictionary elements with respect to the average received power or other objectives. However, due to the TTD approximation of the dictionary entries, the power varies across subcarriers. Based on the observation that changing the scale and shift parameters of the dictionary entries can change the power distribution, we optimize these parameters over a desired dataset in a data-driven manner. The results show that by optimizing the shift and scale parameters, we can achieve the same average received power as [14] while providing fairness across subbands. Chapter 3 is to appear in IEEE WSA 2026 [18].

Another aspect of the new array architecture is the near-field channel characteristics. As opposed to the far-field channel which requires the direction of the user, the near-field channel requires both direction and radial distance of the user for effective beamforming [5, 10]. This further complicates the already complicated beam training process, expanding the search space from direction only to a 2D search over user locations [11]. In Chapter 4 we provide a TTD-based solution, the Virtual Sub-array Sparse Rainbow (VSSR) algorithm, by using the exact delay and phase shift values that far-field rainbow beams utilize [8]. Specifically, we show that we can virtually partition the array and recover each sub-array's signal by utilizing the near-orthogonality of each partition's signal via sparse recovery, and localize the user by triangulation. By utilizing a single orthogonal frequency-division multiplexing (OFDM) symbol we can achieve 95% of the beamforming gain after training. Chapter 4 was presented at the Asilomar 2024 conference [19].

1.4 Notation and Conventions

In this thesis, italic letters, t, denote scalars, bold lowercase letters, $\mathbf{y}$, denote vectors, bold uppercase letters, $\mathbf{H}$, denote matrices, and calligraphic uppercase letters, $\mathcal{T}$, denote sets or spaces. $[\mathbf{X}]_{(m,n)}$ denotes the m-th row and n-th column of the matrix. The symbols $\mathbb{R}, \mathbb{C}, \mathbb{Z}$ denote real numbers, complex numbers, and integers, respectively. The operators

$(\cdot)^H, (\cdot)^T, (\cdot)^\dagger$ are Hermitian transpose, transpose, and Moore-Penrose inverse, respectively. $\|\cdot\|$ denotes the vector norm and $\mathcal{F}\{\cdot\}$ denotes the Fourier transform, further defined in the respective chapter. $\mathcal{CN}(0, \sigma^2)$ is the complex circular Gaussian distribution with zero mean and variance σ^2, and $1{:}N$ means indexing $1, \ldots, N$.

1.5 Thesis Organization

The thesis is organized as follows: Chapter 2 introduces the efficient split-beam generation HDB algorithm; Chapter 3 demonstrates the HDB parameter learning; Chapter 4 shows the efficient beam training VSSR algorithm for near-field channels; and finally Chapter 5 concludes the thesis.

CHAPTER 2

Homomorphic Directional Beamforming with Analog True Time Delay Arrays

2.1 Introduction

Future wireless systems are expected to enable various wireless services ranging from virtual reality (VR) to ubiquitous connectivity, with stringent data rate, latency, and connectivity requirements [2, 3]. Enabling technologies are expected to utilize millimeter wave and sub-THz frequency bands to exploit the abundant bandwidth. However, these frequency bands exhibit high attenuation loss and require beamforming gain of the multi-antenna systems to enable viable communication links. Considering stringent power and cost limitations, analog antenna array architecture with frequency-independent phase shifters has been considered a predominant candidate in millimeter-wave systems [6]. However, analog arrays can only create a single frequency-independent beam per given time slot, limiting the scheduling capabilities of the wireless systems. Furthermore, as the bandwidth and number of antennas increase, analog phased arrays suffer from the frequency-dependent array response [20], referred to as beam squint.

Recently, analog TTD circuit elements have been proposed to enable analog frequency-dependent beamforming capabilities [21, 22], where introduced delay in the time domain provides the ability to create frequency-dependent phase shift in the frequency domain. This frequency-dependent phase-shifting capability has been utilized to eliminate the frequency dependency of the array response by re-aligning all frequencies towards the same direction

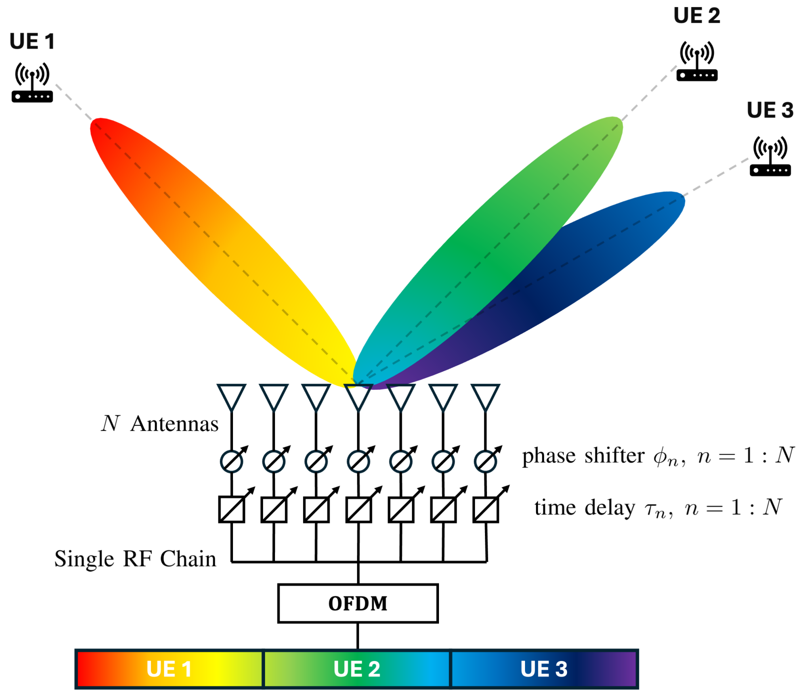


Figure 2.1: Frequency-dependent beamforming with TTD array: Split beampatterns that serve 3 UEs simultaneously with a single RF chain by subband-to-UE (direction) assignment.

[4, 7, 21]. Instead of compensating for the beam squinting effect, TTD elements can also increase the dispersion of the array response across different frequencies, creating so-called *rainbow beampattern*. Rainbow beams have been proposed to reduce the beam training overhead by probing all possible UE directions simultaneously for uniform linear array (ULA) [7, 8, 23–26] and for uniform planar array (UPA) [27]. The array response dispersion is also utilized to facilitate low latency massive connectivity [28] and coverage extension with beam spreading [29].

In addition to dispersive beams, analog TTD arrays have also been utilized to simultaneously serve multiple UEs at different directions based on subband-to-direction multiplexing with *split beampatterns* [9], as shown in Fig. 2.1. The flexible spectrum utilization of split beams is currently being considered as a candidate paradigm for future 6G standards [2] and has been shown to enhance uplink coverage [12], improve edge user throughput [13] and enable low-latency operation [14]. However, the aforementioned applications and increased degrees of freedom introduced by time delay elements bring additional algorithm design challenges that require careful analysis.

Several optimization-based beampattern generation algorithms have been proposed to enable flexible subcarrier-to-direction assignment that is capable of beam-squint elimination, rainbow beamforming, and split beam realization. In [9,14], authors optimized the sum power of the OFDM symbol for different subcarriers for ULA arrays, proposing both exhaustive search-based and convex approximation-based solutions. [14] proposes a closed-form solution for split beampatterns, where partitions of subcarriers are assigned to different directions while ignoring the beam-squint effect. The over-the-air (OTA) verifications of the aforementioned optimization-based algorithms are provided for 28 GHz [30] and sub-6 GHz [31] systems. Furthermore, for UPA arrays, optimization of the max-min power, in addition to sum power, is developed in [32] to provide fairness among UEs. However, optimization-based schemes either require prohibitive time complexity, memory [9], or ignore the beam-squint effect [14]; making the online deployment for wideband systems impractical.

Several heuristic algorithms are proposed for computationally fast split beam generation; in [15], authors propose a closed-form configuration to serve 2 UEs by assigning each UE half of the frequency band and [16] proposed a structured method to serve multiple UEs that are uniformly distributed in direction domain. However, these closed-form methods focus on specific direction-subcarrier assignments and have limitations for practical applications with diverse requirements on the number of users and their spatial directions.

In this chapter, we introduce a novel mathematical framework for computationally fast and memory-efficient split beampattern synthesis using analog TTD arrays. We first show the existence of a homomorphism between the array configuration matrix, i.e., time delay and phase shifter values per antenna and corresponding beampatterns. Utilizing this mathematical structure, we show that hard-to-approximate beampatterns, such as split beampatterns serving multiple UEs, can be written in terms of simple-to-approximate generator beampatterns, allowing fast and efficient split beam synthesis in a divide-and-conquer manner. Our contributions are summarized as follows:

- We rigorously analyze the mathematical structure of the TTD beampattern synthesis and show the homomorphism between TTD array configuration matrices and corresponding beampatterns.
- We propose a novel low-memory and low-complexity algorithm that approximates split beampatterns using a single generator beampattern dictionary.
- We demonstrate the effectiveness of the proposed method with extensive simulations and provide a detailed comparison with the state-of-the-art algorithms.

The remainder of the chapter is organized as follows: We introduce the system model and mathematical definitions in Section 2.2, define the optimization problem in Section 2.3, and analyze the mathematical structure of the beampattern synthesis operation in Section 2.4. Then, we propose a split beampattern synthesis algorithm in Section 2.5 and present the extensive performance analysis in Section 2.6. Finally, we discuss the properties of the proposed algorithm and future directions in Section 2.7 and Section 2.8 concludes the chapter.

2.2 System Model

We consider a wireless system comprising a base station with a single RF chain and N antennas in ULA configuration with half wavelength $\frac{\lambda_c}{2}$ antenna spacing at the carrier frequency f_c. Each antenna element employs a phase shifter with phase shifter value ϕ_n and a TTD unit with a time delay value t_n [9,14]. An M-subcarrier OFDM system with bandwidth BW

is employed for the data communication where each subcarrier $m = 1 : M$ has the frequency $f_m = f_c + (m-1)(BW/(M-1)) - (BW/2)$.

The resulting frequency-dependent precoding vector $\mathbf{v}_m \in \mathbb{C}^{N\times 1}$ for the subcarrier $m \in 1 : M$ and for time delay t_n and phase shift ϕ_n per antenna $n \in 1 : N$ can be defined as follows:

$$[\mathbf{v}_m]_n = \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} \in \mathbb{C}^N,\ m \in 1 : M \tag{2.1}$$

and the precoding matrix is defined as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_M \end{bmatrix} \in \mathbb{C}^{N\times M} \tag{2.2}$$

where the corresponding array response for a given t_n and phase shift ϕ_n, per antenna $n \in 1 : N$ and for the subcarrier $m \in 1 : M$ is given as [9]:

$$\begin{aligned}
P_m(\Psi) &= \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} e^{-jn\pi\Psi\frac{f_m}{f_c}} \\
P_m(\Psi) &= \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} e^{-jn\Omega_m} \\
&= \mathcal{F}\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)}\}_{(\Omega_m)} \\
&:= \mathcal{F}_m\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)}\}
\end{aligned}$$

where $\Psi = \sin(\theta)$ and $\mathcal{F}_m$ is the Discrete Time Fourier Transform (DTFT) with the frequency variable $\Omega_m = \pi\Psi f_m/f_c$, i.e., scaled version of DTFT with the frequency variable $\pi\Psi$.

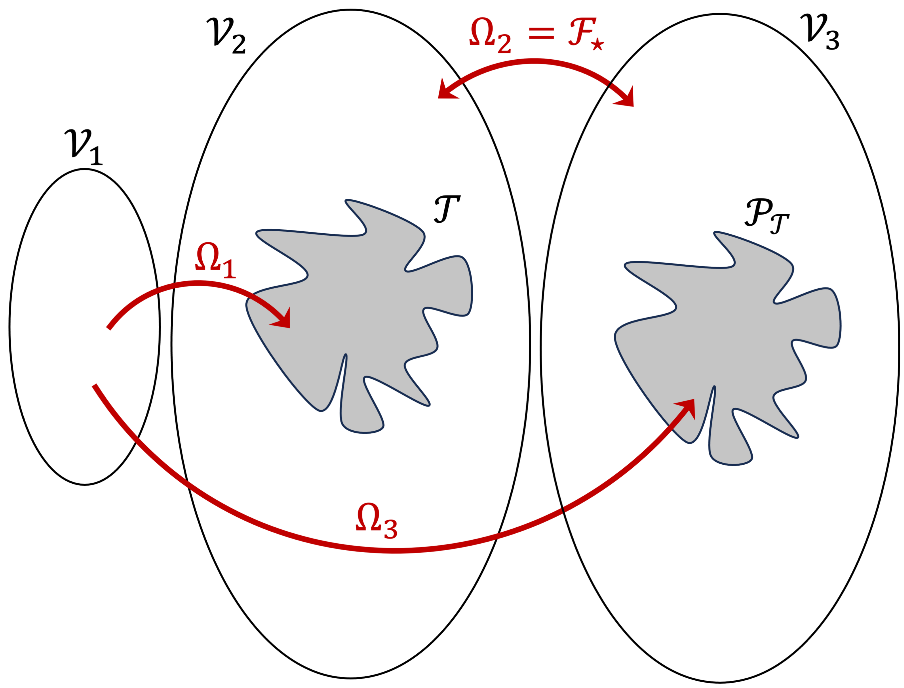


Figure 2.2: Beampattern generation process: sets and corresponding mappings. $\mathcal{V}_1$ is the set of array configuration matrices, $\mathcal{V}_2$ is the set of beamforming vectors and $\mathcal{V}_3$ is the set of beampatterns.

2.2.1 TTD Array and Beampattern Set Mappings

We start by defining the necessary mathematical tools to analyze the TTD beamforming process. Denote the frequency vector $\mathbf{f} \in \mathbb{R}^M$ as $[\mathbf{f}]_{(m)} = f_m, m \in 1 : M$, time delay and phase shifter vectors as $[\mathbf{t}]_{(n)} = t_n$, $[\boldsymbol{\phi}]_{(n)} = \phi_n, n \in 1 : N$ respectively and array configuration matrix as $\mathbf{\Phi} = \begin{bmatrix} \mathbf{t} & \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^{N \times 2}$ and define the following operators:

$$\Omega_1(\mathbf{\Phi}) : \mathcal{V}_1 \to \mathcal{V}_2 := \frac{1}{\sqrt{N}} e^{j(-2\pi \mathbf{t}\mathbf{f}^T + \boldsymbol{\phi}\mathbf{1}_M^T)} := \mathbf{V}_{\mathbf{\Phi}} \tag{2.3a}$$

$$\begin{aligned}\Omega_2(\mathbf{V}) = \mathcal{F}_\star(\mathbf{V}) &:= \mathcal{V}_2 \to \mathcal{V}_3 \\ &= \mathcal{F}_m\{\mathbf{V}(:,m)\} = P_m(\Psi) := [\mathbf{P}]_{(\Psi,m)},\ \forall m\end{aligned} \tag{2.3b}$$

$$\Omega_3(\mathbf{\Phi}) : \mathcal{V}_1 \to \mathcal{V}_3 := \Omega_2(\Omega_1(\mathbf{\Phi})) \tag{2.3c}$$

Where $\mathcal{V}_1 =< \mathbb{R}^{N\times 2}, +, . >$ is array configuration matrix vector space, $\mathcal{V}_2 =< \mathbb{C}^{N\times M}, +, . >$ is beamforming matrix vector space, and $\mathcal{V}_3 =< \mathbb{C}^{[-1,1]}, +, . >$ is beampattern [14] vector space with respect to standard vector addition and multiplication. $\Omega_2 := \mathcal{F}_\star$ is the column-wise, per subcarrier m, $\mathcal{F}_m$ operation defined in the equation (2.3), Ω_1 is the TTD array generating function and Ω_3 is the beampattern generation function from array configuration matrix. $\mathcal{T} := \Omega_1(\mathcal{V}_1) \subset \mathcal{V}_2$ is the set of TTD beamformers i.e. image of the $\mathcal{V}_1$ over the operator Ω_1. $\mathcal{P}_\mathcal{T} := \Omega_2(\mathcal{T})$ is the set of TTD beampatterns generated by TTD Arrays $\mathcal{T}$. Observe that $\mathcal{T}$ and $\mathcal{P}_\mathcal{T}$ are non-convex from the properties of the Ω_1 and the linearity of the Ω_2 operators. The sets and mappings are illustrated in Fig. 2.2.

2.3 Problem Definition and State-of-the-art

The goal of this chapter is to find the array configuration matrix $\mathbf{\Phi} = \begin{bmatrix}\mathbf{t} & \boldsymbol{\phi}\end{bmatrix} \in \mathbb{R}^{N\times 2}$ to generate the beampattern $\mathbf{P}_{\mathbf{\Phi}}$ that minimizes the following multi-objective optimization problem [33]:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{N\times 2}} \oint \|[\mathbf{P}_{\mathbf{\Phi}}]_{(\Psi,m)} - [\mathbf{P}_{target}]_{(\Psi,m)}\|^2 d\Psi,\ \forall m \tag{2.4}$$

where $\mathbf{P}_{target}$ is the desired beampattern. This problem is non-convex and intractable as the feasible set of TTD beampatterns $\mathcal{P}_\mathcal{T}$ is non-convex. Observe that the problem (2.4) can be optimally solved for digital arrays (for relaxed problem) as digital arrays can design beamforming vectors of subcarriers independently. We continue with sum-objective based near-optimal solution strategies [9, 14]:

2.3.1 Sum-objective optimization over TTD Beampatterns

The first sum-objective based solution strategy for the problem (2.4) optimizes the following optimization problem over vector space $\mathcal{V}_3$ [14]:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{N\times 2}} \sum_{m=1}^{M} \oint \|[\mathbf{P}_{\mathbf{\Phi}}]_{(\Psi,m)} - [\mathbf{P}_{target}]_{(\Psi,m)}\|^2 d\Psi \tag{2.5}$$

Although the multi-objective problem is reduced to a single objective problem, the problem (2.5) is still non-convex as the feasible set remains the same, and obtaining an optimal solution is still challenging. An exhaustive-search-based solution for the optimization problem (2.5) is proposed over the relaxed feasible set, and a low-complexity approximate solution is proposed for the approximate Ω_2 [14]. However, the proposed algorithms ignore the beam-squint effect [20] i.e., assumes $\mathcal{F}_m\{.\}$, defined in equation (2.3) is $\mathcal{F}_m\{.\} \approx \mathcal{F}\{.\}$.

2.3.2 Weighted sum-objective optimization over TTD Beamformers

Alternatively, [9] solves the following optimization problem over the vector space $\mathcal{V}_2$:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{N\times 2}} \sum_{m=1}^{M} \sum_{n=1}^{N} |[\mathbf{V}_{\mathbf{\Phi}}]_{(n,m)} - [\mathbf{V}_{target}]_{(n,m)}|^2 \tag{2.6}$$

The problem (2.6) is equivalent to the weighted sum-objective solution of the (2.4) by generalized Parseval's equality [34], where weights are introduced due to the changing signal period per subcarrier m from scaled DTFT definition in equation (2.3). The problem (2.6) is non-convex and NP-hard [9]; therefore, alternating minimization-based solution methodology is proposed in [9] by either employing exhaustive search or by linearizing the TTD array generating function Ω_1.

The aforementioned sum-optimization-based solution methods either require exhaustive search or involve some way of convex relaxation on feasible sets $\mathcal{T}$ or $\mathcal{P}_{\mathcal{T}}$. We want to avoid this route; Instead, we show that the set of the vector space $\mathcal{V}_3$ forms a group structure with an appropriate addition operation, and the beampattern synthesis operation in (2.3c) forms a homomorphism. Hence, we can approximate and generate complicated beampatterns from known and simpler beampatterns to simplify the beampattern synthesis process, following the general wisdom of signal processing. We continue with a rigorous understanding of the beampattern generation process.

2.4 Homomorphism over TTD Beampatterns

We start by showing the function Ω_3 defines a homomorphism [35] between $< \mathcal{V}_1, + >$ and $< \mathcal{P}_{\mathcal{T}} \subset \mathcal{P}, \star >$ where $\star$ is the scaled column-wise circular convolution operation defined as:

$$\begin{aligned}
&\text{Let } \mathbf{P}^{'}, \mathbf{P}^{''} \in \mathcal{P}, \\
&\star : \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{P} := \\
&\qquad [\mathbf{P}^{'} \star \mathbf{P}^{''}]_{(:,m)} = \sqrt{N} \left([\mathbf{P}^{'}]_{(:,m)} \circledast [\mathbf{P}^{''}]_{(:,m)} \right), \forall\, m
\end{aligned} \tag{2.7a}$$

where the operation $\circledast$ is the circular convolution over one period of Ω_m, normalized by the period.

Theorem 2.1. Ω_3 *is a homomorphism [35] between* $< \mathcal{V}_1, + >$ *and* $< \mathcal{P}_{\mathcal{T}}, \star >$ *:*

$$\begin{aligned}
&\forall\, \mathbf{\Phi}_1, \mathbf{\Phi}_2 \in \mathcal{V}_1 = \mathbb{R}^{N \times 2} \\
&\quad \Omega_3 \left(\mathbf{\Phi}_1 + \mathbf{\Phi}_2 \right) = \Omega_3 \left(\mathbf{\Phi}_1 \right) \star \Omega_3 \left(\mathbf{\Phi}_2 \right),
\end{aligned}$$

Proof.

$$\text{Let, } \boldsymbol{\Phi_1} = \begin{bmatrix} \mathbf{t_1} & \boldsymbol{\phi_1} \end{bmatrix}, \boldsymbol{\Phi_2} = \begin{bmatrix} \mathbf{t_2} & \boldsymbol{\phi_2} \end{bmatrix} \in \mathcal{V}_1 = \mathbb{R}^{N\times 2}, \tag{2.9a}$$

$$[\Omega_3(\boldsymbol{\Phi_1} + \boldsymbol{\Phi_2})]_{(:,m)} = \mathcal{F}_m\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m(\mathbf{t_1}+\mathbf{t_2})+(\boldsymbol{\phi_1}+\boldsymbol{\phi_2}))}\}, \quad \forall m \tag{2.9b}$$

$$= \sqrt{N}\mathcal{F}_m\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m \mathbf{t_1}+\boldsymbol{\phi_1})} \frac{1}{\sqrt{N}} e^{j(-2\pi f_m \mathbf{t_2}+\boldsymbol{\phi_2})}\} \tag{2.9c}$$

$$= \sqrt{N}(\mathcal{F}_m\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m \mathbf{t_1}+\boldsymbol{\phi_1})}\} \circledast \mathcal{F}_m\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m \mathbf{t_2}+\boldsymbol{\phi_2})}\}) \tag{2.9d}$$

$$= \sqrt{N}\left([\Omega_3(\boldsymbol{\Phi_1})]_{(:,m)} \circledast [\Omega_3(\boldsymbol{\Phi_2})]_{(:,m)}\right) \tag{2.9e}$$

$$\iff \Omega_3(\boldsymbol{\Phi_1} + \boldsymbol{\Phi_2}) = \Omega_3(\boldsymbol{\Phi_1}) \star \Omega_3(\boldsymbol{\Phi_2}) \tag{2.9f}$$

□

Where (2.9b, 2.9e) is from the definition (2.3c), (2.9c) is from properties of complex numbers, (2.9d) is from convolution property of discrete Fourier transform and (2.9f) is from the definition of $\star$ from (2.7a).

Corollary 2.1.

$$\forall \boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \in \mathcal{V}_1 = \mathbb{R}^{N\times 2},\ \mathbf{P}_{\boldsymbol{\Phi}_1+\boldsymbol{\Phi}_2} = \mathbf{P}_{\boldsymbol{\Phi}_1} \star \mathbf{P}_{\boldsymbol{\Phi}_2}$$

Proof. Proof follows from Theorem 2.1 and Ω_3 definition in (2.3c). □

Convolution and its algebraic properties have been previously discussed under generalized linearity for homomorphic filtering in [36], where homomorphic transformation is intentionally introduced for domain transformation (followed by the inverse transformation). For TTD arrays, however, we observed and showed that homomorphism is the inherent property of the beampattern synthesis process.

$< \mathcal{P}_{\mathcal{T}}, \star >$ defines a group structure with respect to $\star$ operator [36]. However, not every beampattern $\mathbf{P} \in \mathcal{P}$ has an inverse with respect to the $\star$ operator; hence, we cannot extend the group structure to the $< \mathcal{P}, \star >$. To ensure invertibility and group structure, we restrict the set of beampatterns $\mathcal{P}$ to $\mathcal{P}_{nz}$ where the patterns are generated by the beamforming vectors with non-zero gain for every antenna $n \in 1 : N$ and for every subcarrier $m \in 1 : M$. This is true for most of our desired applications, including TTD analog arrays, and we denote $\mathcal{P}_{nz} = \mathcal{P}$ for the rest of the chapter for conciseness. We continue with demonstrating how a TTD beampattern can be synthesized by utilizing the observed homomorphism.

2.4.1 Homomorphic TTD Beampattern Synthesis

Let us consider a TTD beampattern $\mathbf{P}_{\mathbf{\Phi}'} \in \mathcal{P}_{\mathcal{T}}$ and assume that it can be represented with G TTD beampatterns over $\star$ operation such as:

$$\mathbf{P}_{\mathbf{\Phi}'} = \mathbf{P}_{\mathbf{\Phi_1}} \star \cdots \star \mathbf{P}_{\mathbf{\Phi}_{\boldsymbol{G}}} \tag{2.10}$$

Then, the Corollary 2.1 implies that we can write the corresponding array configuration matrices as:

$$\mathbf{\Phi}' = \sum_{g=1}^{G} \mathbf{\Phi}_{\boldsymbol{g}} \tag{2.11}$$

and can directly find time delay $\mathbf{t}'$ and phase shifter values $\boldsymbol{\phi'}$ as:

$$\mathbf{t}' = \sum_{g=1}^{G} \mathbf{t_g}, \quad \boldsymbol{\phi'} = \sum_{g=1}^{G} \boldsymbol{\phi_g} \tag{2.12}$$

Therefore, finding a representation of a TTD beampattern over $\star$ operation is sufficient to determine its array configuration matrix. Although a representation of a beampattern $\mathbf{P} \in \mathcal{P}$ that is not a TTD beampattern, i.e., $\mathbf{P} \notin \mathcal{P}_{\mathcal{T}}$ can also be found from the group structure of $< \mathcal{P}, \star >$; utilization of the homomorphism and equation (2.12) requires the approximation of the $\mathbf{P}$ or its representation by TTD beampatterns in $\mathcal{P}_{\mathcal{T}}$.

2.4.2 Beampattern Approximation: Divide-and-Conquer

We denote the beampatterns that represent $\mathbf{P}$ as *generator* beampatterns. Representing a beampattern $\mathbf{P} \in <\mathcal{P}, \star>$ over $\star$ operation resembles the basis representation of a vector. Hence, instead of working on a hard-to-approximate beampattern, our intuitive approach is to work on corresponding easy-to-approximate *generator* beampatterns and utilize the equations (2.11),(2.12) to synthesize the approximated TTD beampattern, in a divide-and-conquer manner. Unfortunately, representing an arbitrary beampattern in $<\mathcal{P}, \star>$ and determining *generator* beampatterns is not a trivial task and requires more advanced mathematical tools beyond the utilized group structure. We leave this task to future research and instead focus on beampatterns that can be represented by easy-to-approximate *generator* beampatterns such as split beampatterns.

2.5 Split Beampattern Approximation

We continue with the definition of split beampatterns and show that split beampatterns can be represented with easy-to-approximate *generator* beampatterns.

2.5.1 Split Beampattern Definition

A split beampattern with G subbands $\mathbf{P}(\boldsymbol{\varphi}) \in \mathcal{P} \subset \mathcal{P}_{\mathcal{V}_2}$, assigns directions to subbands according to subband-to-direction mapping vector $\boldsymbol{\varphi} \in [-1,1]^{G\times 1}$. Therefore, $\mathbf{P}(\boldsymbol{\varphi}) \in \mathcal{P} \subset \mathcal{P}_{\mathcal{V}_2}$ is defined to be generated by the following far-field beamforming matrix $\mathbf{V}(\boldsymbol{\varphi}) \in \mathcal{V}_2$ [9]:

$$
\begin{aligned}
&[\mathbf{V}(\boldsymbol{\varphi})]_{(:,m)} = \frac{1}{\sqrt{N}} e^{j\pi n[\psi]_m \frac{f_m}{f_c}} \in \mathbb{C}^{N\times 1},\ m \in 1:M \\
&\boldsymbol{\psi} = \Big[\underbrace{\varphi_1 \ \ \dots \ \ \varphi_1}_{M'} \ \ \cdots \ \ \underbrace{\varphi_G \ \ \dots \ \ \varphi_G}_{M'}\Big]^T
\end{aligned}
\tag{2.13}
$$

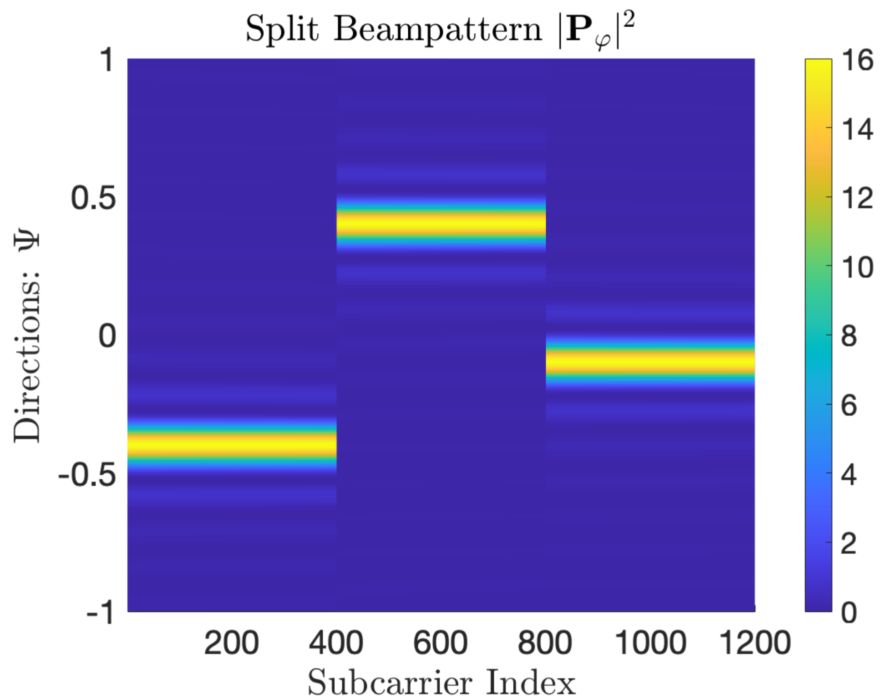


Figure 2.3: Power of a split beampattern with $G = 3$ subbands where each subband is assigned to a direction with the subband-to-direction mapping vector: $\boldsymbol{\varphi} = \begin{bmatrix} -0.4 & 0.4 & -0.1 \end{bmatrix}^T$.

where the subcarriers are partitioned into G subbands each containing $M' = \frac{M}{G}$ subcarriers and subcarrier m is directed towards the direction $[\boldsymbol{\psi}]_m$. Then, the corresponding beampattern $\mathbf{P}(\boldsymbol{\varphi}) \in \mathcal{P}_{\mathcal{V}_2}$ is obtained as follows:

$$\begin{aligned}[\mathbf{P}(\boldsymbol{\varphi})]_{(:,m)} &= [\Omega_2(\mathbf{V}(\boldsymbol{\varphi}))]_{(:,m)}, \\ &= \frac{1}{\sqrt{N}} \Xi_{N,m}(\Psi - \psi_m), \ \forall m \end{aligned} \tag{2.14a}$$

where $\Xi_{N,m}(x) = e^{-j(N-1)\pi x f_m/2f_c} \frac{\sin(Nx\pi f_m/2f_c)}{\sin(x\pi f_m/2f_c)}$ is the Dirichlet sinc function for N antennas and f_m [4] and a split beampattern example is given in Fig. 2.3. The following Lemma 2.1

shows an important property of the split beampatterns, i.e., when the $\star$ operation is applied to two split beampatterns, their corresponding direction mapping vectors are added:

Lemma 2.1. *Let* $\mathbf{P}(\boldsymbol{\varphi}_1), \mathbf{P}(\boldsymbol{\varphi}_2) \in \mathcal{P}_{\mathcal{V}_2}$*, then*

$$\mathbf{P}(\boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2) = \mathbf{P}(\boldsymbol{\varphi}_1) \star \mathbf{P}(\boldsymbol{\varphi}_2)$$

Proof.

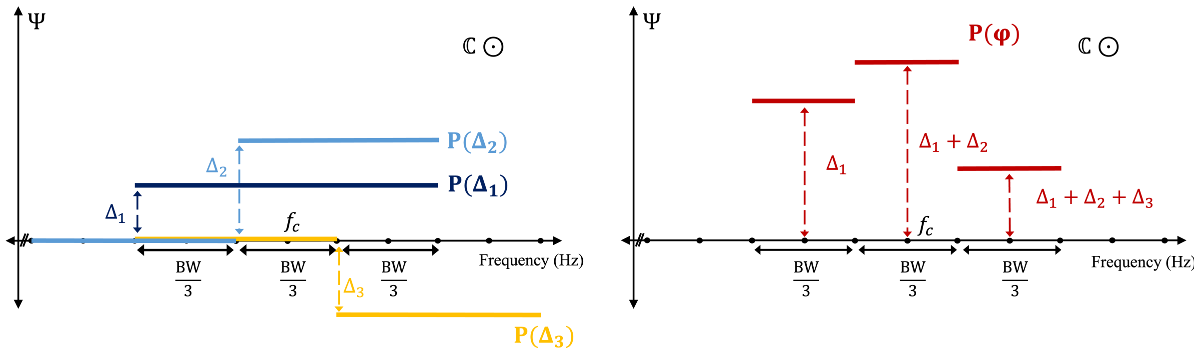
$$\begin{aligned}
&\text{Let, } \mathbf{P}(\boldsymbol{\varphi}_1), \mathbf{P}(\boldsymbol{\varphi}_2) \in \mathcal{P}_{\mathcal{V}_2}, \\
&[\mathbf{P}(\boldsymbol{\varphi}_1) \star \mathbf{P}(\boldsymbol{\varphi}_2)]_{(:,m)} \\
&\quad = \sqrt{N}\Big(\frac{1}{\sqrt{N}}\Xi_{N,m}(\Psi - [\boldsymbol{\psi}_{\mathbf{1}}]_m) \circledast \frac{1}{\sqrt{N}}\Xi_{N,m}(\Psi - [\boldsymbol{\psi}_{\mathbf{2}}]_m)\Big) && \text{(2.15a)} \\
&\quad = \frac{1}{\sqrt{N}}\Xi_{N,m}(\Psi - ([\boldsymbol{\psi}_{\mathbf{1}}]_m + [\boldsymbol{\psi}_{\mathbf{2}}]_m)) && \text{(2.15b)} \\
&\quad = [\mathbf{P}(\boldsymbol{\varphi}_1 + \boldsymbol{\varphi}_2)]_{(:,m)}, \ \forall m && \text{(2.15c)}
\end{aligned}$$

□

where equations (2.15a),(2.15c) follow from the definition in equation (2.14a), and equation (2.15b) follows from the properties of the Dirichlet sinc function. The Lemma 2.1 is the main tool for obtaining generator beampatterns that represent the split beampatterns. We continue with the construction of the generator beampatterns of split beampatterns. Then, we discuss the properties of the generators that make them easy to approximate.

2.5.2 Split Beampattern Generators

We start our demonstration with the split beampatterns with $G = 3$ partitions. Consider an arbitrary split beampattern $\mathbf{P}(\boldsymbol{\varphi})$ with subband-to-direction mapping vector $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \varphi_3]^T \in \mathbb{R}^{3\times 1}$ and beampatterns $\mathbf{P}(\boldsymbol{\Delta}_{\mathbf{1}}), \mathbf{P}(\boldsymbol{\Delta}_{\mathbf{2}}), \mathbf{P}(\boldsymbol{\Delta}_{\mathbf{3}})$ with the following direction



(a) (b)

Figure 2.4: Ideal split beampattern generators and beampattern generation process. (a) Ideal split beampattern generators created from shifted and scaled 2 direction split beampatterns (b) Beampattern obtained by taking $\star$ operation of the generator beampatterns.

mapping vectors:

$$
\begin{aligned}
\boldsymbol{\Delta_1} &= [\Delta_1, \Delta_1, \Delta_1]^T \in \mathbb{R}^{3\times 1} \\
\boldsymbol{\Delta_2} &= [0, \Delta_2, \Delta_2]^T \in \mathbb{R}^{3\times 1} \\
\boldsymbol{\Delta_3} &= [0, 0, \Delta_3]^T \in \mathbb{R}^{3\times 1}
\end{aligned}
\tag{2.16}
$$

Let $\mathbf{P}(\boldsymbol{\varphi}) = \mathbf{P}(\boldsymbol{\Delta_1}) \star \mathbf{P}(\boldsymbol{\Delta_2}) \star \mathbf{P}(\boldsymbol{\Delta_3})$ which is illustrated in Fig. 2.4a, ignoring the out-of-bandwidth components which we discuss in Section 2.5.3. Then, from Lemma 2.1:

$$\mathbf{P}(\boldsymbol{\varphi}) = \mathbf{P}(\boldsymbol{\Delta_1}) \star \mathbf{P}(\boldsymbol{\Delta_2}) \star \mathbf{P}(\boldsymbol{\Delta_3}) = \mathbf{P}(\boldsymbol{\Delta_1} + \boldsymbol{\Delta_2} + \boldsymbol{\Delta_3})$$

which implies that,

$$[\varphi_1, \varphi_2, \varphi_3]^T = [\Delta_1, \Delta_1 + \Delta_2, \Delta_1 + \Delta_2 + \Delta_3]^T \tag{2.17}$$

Therefore, we can generate an arbitrary split beampattern with $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \varphi_3]^T \in \mathbb{R}^{3\times 1}$ from 3 generators, $\mathbf{P}(\boldsymbol{\Delta_1}), \mathbf{P}(\boldsymbol{\Delta_2}), \mathbf{P}(\boldsymbol{\Delta_3})$, by solving the following simple linear equation for a given $\boldsymbol{\varphi}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix} \tag{2.18}$$

This reduces the beampattern design process to a simple linear equation with 3 unknowns, which can be solved as follows:

$$\begin{aligned} \Delta_1 &= \varphi_1 \\ \Delta_2 &= \varphi_2 - \varphi_1 \\ \Delta_3 &= \varphi_3 - \varphi_2 \end{aligned} \tag{2.19}$$

From construction, generator beampattern directions need to satisfy $\forall g \geq 2, |\Delta_g| \leq 1$ to avoid wrapping. Therefore, if $\exists g \geq 2, |\Delta_g| > 1$ we revise the generator beampatterns as follows:

$$\begin{aligned} \boldsymbol{\Delta_g} &\leftarrow \boldsymbol{\Delta_g} - \lfloor \frac{\Delta_g}{2} \rfloor \\ \boldsymbol{\Delta_1} &\leftarrow \boldsymbol{\Delta_1} + \lfloor \frac{\Delta_g}{2} \rfloor \end{aligned} \tag{2.20}$$

which does not change the resulting split beampattern from the equation (2.17). The above process can be generalized to $G > 3$ partitions by defining the generator beampatterns, $\mathbf{P}(\boldsymbol{\Delta_g}),\ g \in\ [1, G]$ with the following subband-direction mapping vector:

$$\boldsymbol{\Delta_g} = [\underbrace{0, \ldots, 0}_{g-1}, \underbrace{\Delta_g, \ldots, \Delta_g}_{G-g+1}]\ g \in\ [1, G] \tag{2.21}$$

and solving the following linear equation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \ddots & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_G \end{bmatrix} = \begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_G \end{bmatrix} \tag{2.22}$$

The solution of (2.22) is:

$$\begin{aligned} &\Delta_1 = \varphi_1 \\ &\Delta_g = \varphi_g - \varphi_{g-1};\ g \in [2, G] \end{aligned} \tag{2.23}$$

The corresponding generator beampatterns are revised as in equation (2.20).

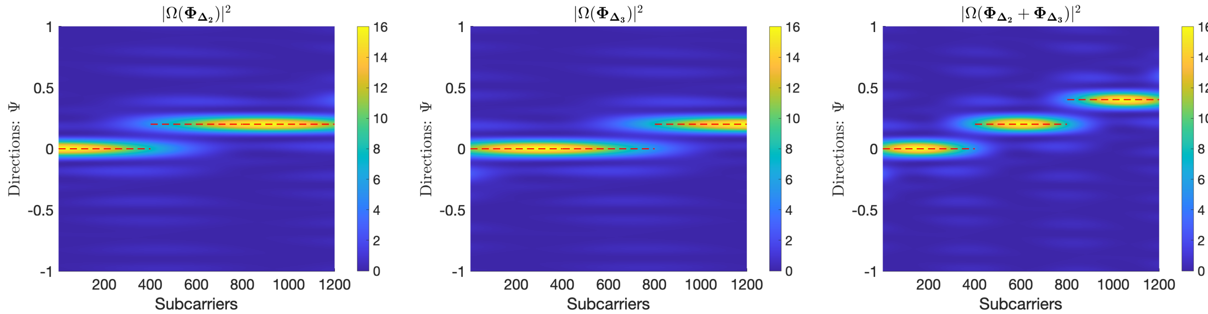
We have shown that a split beampattern $\mathbf{P}(\varphi)$ can be represented over $\star$ operation with split beampattern generators $\mathbf{P}(\boldsymbol{\Delta_g}),\ g \in\ [1, G]$ as follows:

$$\mathbf{P}(\boldsymbol{\varphi}) = \mathbf{P}(\boldsymbol{\Delta_1}) \star \cdots \star \mathbf{P}(\boldsymbol{\Delta_G}) \tag{2.24}$$

where $\Delta_g,\ g \in\ [1, G]$ are obtained from equations (2.23),(2.20). Split beampattern generators for $g > 1$, like the desired split beampattern, cannot be realized by TTD arrays due to the discontinuity of directions between different subbands [9]. However, we have successfully represented the desired split beampattern with split beampattern generators; if split beampattern generators are easy-to-approximate, we can utilize the divide-and-conquer strategy discussed in Section 2.4.2. We continue with the properties of the split beampattern generators and show that such generators can be efficiently approximated with TTD arrays.

2.5.3 Approximating Split Beampattern Generators

Split beampattern generators $\mathbf{P}(\boldsymbol{\Delta_g}),\ g \in\ [1, G]$ can be directly approximated with TTD arrays $\mathbf{P}(\boldsymbol{\Delta_g}) \approx \mathbf{P}_{\Phi_{\Delta_g}}(\boldsymbol{\Delta_g}) \in \mathcal{P}_{\mathcal{T}}$ with the optimization based algorithms [9, 14] discussed in Section 2.3. However, optimization problem-based approaches do not provide any computational advantage as approximating generators have the same computational complexity



(a) $\mathbf{P}_{\mathbf{\Phi}_{\mathbf{\Delta_2}}, \mathbf{\Delta_2}} \approx [0, 0.2, 0.2]^T$ (b) $\mathbf{P}_{\mathbf{\Phi}_{\mathbf{\Delta_3}}, \mathbf{\Delta_3}} \approx [0, 0, 0.2]^T$ (c) $\mathbf{P}_{\mathbf{\Phi}_{\mathbf{\Delta_2}+\mathbf{\Delta_3}}, \mathbf{\Delta_2}+\mathbf{\Delta_3}} \approx [0, 0.2, 0.4]^T$

Figure 2.5: Generator Beampatterns approximated by the mmFlexible math [14] algorithm (2.5a, 2.5b) and resulting beampattern after adding corresponding time delay and phase shifts (2.5c). Dashed red lines indicate the target directions.

as approximating the desired split beampattern. Fortunately, we can utilize the structure of the generator beampatterns to simplify the approximation process significantly.

Similar to Section 2.5.2, we first demonstrate the structure of the split beampattern generators for $G = 3$ subbands; which are illustrated in Fig. 2.4a. We can observe that $\mathbf{P}(\mathbf{\Delta_1}) = \mathbf{P}_{\mathbf{\Phi}_{\mathbf{\Delta_1}}}(\mathbf{\Delta_1})$ has the same direction assignment for all subbands. Therefore, it can be realized by TTD arrays, in closed form, with the following array configuration matrix [7]:

$$\begin{aligned} \mathbf{\Phi}_{\mathbf{\Delta_1}} &= \begin{bmatrix} \frac{-\Delta_1 \mathbf{n}}{2f_c} & \mathbf{0} \end{bmatrix} \in \mathcal{V}_1 \\ \mathbf{n} &= \begin{bmatrix} 0 & \dots & N-1 \end{bmatrix}^T \end{aligned} \tag{2.25}$$

Furthermore, $\mathbf{P}(\mathbf{\Delta_2})$ can be viewed as a split beampattern with $G = 2$ subbands for center frequency $f_c^{\Delta_2} = f_c - \frac{BW}{2} + \frac{BW}{3}$ and with enlarged bandwidth of $BW^{\Delta_2} = \frac{4BW}{3}$. Therefore, it can be approximated with low complexity $G = 2$ subband split beampattern approximation algorithms [9, 14–16] for appropriate center frequency and bandwidth. This approach does not affect the desired split beampattern as subcarriers beyond the assigned bandwidth can be ignored. Similarly, split beam generators for $G > 3$ subbands can be approximated as

$G = 2$ split beampatterns with the following center frequency and bandwidth:

$$\begin{aligned} f_c^{\Delta_g} &= f_c - \frac{BW}{2} + \frac{(g-1)BW}{G} \\ BW^{\Delta_g} &= 2BW\frac{G-1}{G} \end{aligned} \tag{2.26}$$

The split beampattern generator approximation can further be simplified by utilizing the following observation: changing center frequency and bandwidth of a $G = 2$ split beampattern to f^{Δ_g} and BW^{Δ_g} can be accomplished by solving the following linear equation [15]:

$$\begin{aligned} \forall m:\ & -2\pi f_m \mathbf{t} + \boldsymbol{\phi} = -2\pi f_m^{\Delta_g} \mathbf{t}^{\Delta_g} + \boldsymbol{\phi}^{\Delta_g} \\ \forall m:\ & -2\pi(f_c + \frac{(m-1)BW}{M-1} - \frac{BW}{2})\mathbf{t} + \boldsymbol{\phi} \\ & \quad = -2\pi(f_c^{\Delta_g} + \frac{(m-1)BW^{\Delta_g}}{M-1} - \frac{BW^{\Delta_g}}{2})\mathbf{t}^{\Delta_g} + \boldsymbol{\phi}^{\Delta_g} \end{aligned} \tag{2.27}$$

which is obtained from precoder matrix definition in equation (2.3a) and can be solved as:

$$\begin{aligned} \mathbf{t}^{\Delta_g} &= \frac{1}{\alpha_g}\mathbf{t} \\ \boldsymbol{\phi}^{\Delta_g} &= \boldsymbol{\phi} - 2\pi f_c \mathbf{t} + \frac{1}{\alpha_g} 2\pi f_c^{\Delta_g} \mathbf{t} \\ \alpha_g &= \frac{BW^{\Delta_g}}{BW} \end{aligned} \tag{2.28}$$

Therefore, we can approximate split beampattern generators $\mathbf{P}(\boldsymbol{\Delta_g}), \forall g$ from the approximation of split beampatterns with $G = 2$ subbands with a **linear transformation** over corresponding array configuration matrices, as in equation (2.28). Hence, If we define a dictionary of array configuration matrices of split beampatterns with $G = 2$ subbands, we can approximate any split beampattern generator with a simple linear transformation. We continue with the construction of the $G = 2$ subband split beampattern dictionary, which we call *generator dictionary*.

2.5.4 Generator Dictionary

We construct the generator dictionary, i.e., dictionary of $G = 2$ partition split beampatterns $\boldsymbol{\Upsilon} \in \mathbb{R}^{N\times 2\times(2A-1)}$ for A grid directions in two steps. First, we obtain the array configuration matrices $\boldsymbol{\Phi}_{\underline{\Delta}}$ of approximate split beampatterns with subband-to-direction mapping $\underline{\boldsymbol{\Delta}} = [\Delta_1, \Delta_2],\ \Delta_1, \Delta_2 \in [-1, 1]$ with JPTA approximation algorithm [9] or with equation (2.25) if $\Delta_1 = \Delta_2$. The fidelity of the approximated split beampatterns is directly related to the fidelity of the approximated generator beampatterns. As illustrated in Fig. 2.5a and 2.5b, gain across subcarriers of each subband is not constant, and the maximum is not necessarily attained at the center of the subband. Hence, as a second step, we post-process the split beampatterns such that the maximum gain is attained at the center of each subband by adjusting the bandwidth as $BW^{\underline{\Delta}} = BW\frac{M}{2|\arg\max|[\mathbf{P}_{\boldsymbol{\Phi}_{\underline{\Delta}}}]_{(\Delta_2,:)}|-\arg\max|[\mathbf{P}_{\boldsymbol{\Phi}_{\underline{\Delta}}}]_{(\Delta_1,:)}||}$ with the equation (2.28). We empirically observe that this post-processing results in uniform gain across subcarriers of the approximated split beampatterns.

Then, utilizing the dictionary $\boldsymbol{\Upsilon}$, we approximate any generator beampattern $\mathbf{P}(\boldsymbol{\Delta_g})$ by simply reading the dictionary for the corresponding directions and scaling its center frequency and bandwidth with equations (2.26) and (2.28).

2.5.5 Homomorphic Directional Beamforming Algorithm

We have shown that split beam generators can be efficiently approximated simply by reading the corresponding TTD array configuration from the dictionary $\boldsymbol{\Upsilon}$ followed by a trivial linear transformation defined in equation (2.28). Since the desired split beampattern is represented by generator beampatterns as in equation (2.24) and approximated generator beampatterns are in TTD beampatterns group, we can utilize the observed homomorphism and obtain array configuration matrix of the desired split beampattern from the equation (2.11). Therefore, instead of approximating the desired split beampattern directly, we efficiently approximated its generators and obtained the corresponding array configuration matrix from the observed homomorphism.

This basically reduces the split beampattern generation process to solving simple linear equations (2.22) and adding the time delay and phase shifter values of the corresponding generators as in equation (2.12). HDB algorithm is summarized in Algorithm 1. Fig. 2.5a and 2.5b illustrate the approximated generators $\mathbf{P}(\boldsymbol{\Delta_2})$ and $\mathbf{P}(\boldsymbol{\Delta_3})$ for $G = 3$ with mmFlexible algorithm [14] and Fig. 2.5c illustrates the resulting beampattern $\mathbf{P}(\boldsymbol{\Delta_2} + \boldsymbol{\Delta_3})$ after adding the array configuration matrices of the approximated generators.

Algorithm 1: HDB Algorithm

Input: Desired split beampattern $\mathbf{P}(\boldsymbol{\varphi})$ with $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_G]^T \in \mathbb{R}^{G \times 1}$

Output: Array configuration matrix $\boldsymbol{\Phi}_{\boldsymbol{\varphi}}$ such that $\Omega_3(\boldsymbol{\Phi}_{\boldsymbol{\varphi}}) = \mathbf{P}_{\boldsymbol{\Phi}_{\boldsymbol{\varphi}}} \approx \mathbf{P}(\boldsymbol{\varphi})$

1 Solve the equation (2.23) and obtain $\Delta_g,\ g \in [1, G]$ from the equation (2.20)

2 Obtain $\boldsymbol{\Phi}_{\boldsymbol{\Delta_g}}$, $\forall g$ that approximates the generator $\mathbf{P}_{\boldsymbol{\Delta_g}}$ from the dictionary $\boldsymbol{\Upsilon}$ with equation (2.26) and (2.28)

3 Add time delay and phase shifter vectors, equation (2.12):
$\mathbf{t} = \sum_{g=1}^{G} \mathbf{t}^{\Delta_g}, \boldsymbol{\phi} = \sum_{g=1}^{G} \boldsymbol{\phi}^{\Delta_g}$

4 **return** $\boldsymbol{\Phi}_{\boldsymbol{\varphi}} = \begin{bmatrix} \mathbf{t} & \boldsymbol{\phi} \end{bmatrix}$,

---

2.5.6 Effect of Generator Approximation

As discussed, the performance of the HDB algorithm depends on the fidelity of the approximated generators. As the HDB algorithm effectively relies on the addition of the subband directions of each split beampattern generator, any direction error inherited from dictionary construction, such as errors from the JPTA approximation algorithm [9] due to the increasing number of antennas or bandwidth, can accumulate. We investigate the effect of error accumulation in Section 2.6 through extensive simulations. Furthermore, the approximation of generator beampatterns is empirically observed [9] and systematically analyzed [16] to result in the weighted addition of two directional beams as shown in Fig. 2.5a and 2.5b, notably

Table 2.1: Simulation Parameters

Parameter	**Value**
Center frequency f_c	28 GHz
Bandwidth BW	1–10 GHz
Pre-beamforming SNR	10 dB
Number of subcarriers M	1200
Number of ULA antennas N	$\{8, 16, 32, 64, 128\}$
Number of UEs G	$\{3, 4, 5, 6, 7, 8\}$
UE direction grid numbers A	499
UE directions $\varphi_d,\ d \in [1, G]$	$\sim \mathcal{U}_A[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$
UE direction sector	$[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$
Number of Monte Carlo Locations	5000
JPTA [9] Approximation Maximum Delay	$\frac{M}{BW}$
JPTA [9] Approximation iterations I_{it}	30

observable at the transition between subbands. We design the generator beampatterns to avoid coinciding transition regions of different generator beampatterns so that approximated generators preserve the split beampattern generator's behavior.

2.6 Performance Analysis

This section demonstrates a detailed analysis of the performance of HDB algorithm compared to the benchmark algorithms. Numerical simulations are conducted for an OFDM system with a ULA comprising $N = 16$ antennas, operating at a carrier frequency of $f_c = 28$ GHz with bandwidth $BW = 3$ GHz. The system has $M = 1200$ subcarriers and serves $G = 3$ UEs with pre-beamforming SNR of 10 dB unless otherwise stated. Monte Carlo simulations are conducted over 5000 different UE direction configurations, where UE directions are drawn from a discrete uniform distribution over the uniform $A = 499$-point grid on $[-1, 1]$ restricted to the sector $[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$ i.e., $\varphi_d \sim \mathcal{U}_A[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$ for every UE $d \in [1, G]$. The system configuration and benchmark algorithm parameters are summarized in Table 2.1, and the following benchmark schemes are discussed for comparison:

- **mmFlexible [14] Math**: Convex approximation based solution of the problem (2.5), ignores beam-squint. The algorithm provides a closed-form solution with time complexity $\mathcal{O}(NG)$ where G is the number of UEs.
- **FSDA [14]**: Exhaustive search-based solution of the problem (2.5), ignores beam-squint.
- **JPTA [9] Line Search**: Exhaustive search-based solution of the problem (2.4).
- **JPTA [9] Approximation**: Convex approximation based solution of the problem (2.4). The algorithm iteratively solves the approximated problem with a $\mathcal{O}(MN)$ complexity in each iteration.

The *Spectral Efficiency (SE)* is considered as a performance metric which is defined as follows for the target split beampattern with subband-to-direction mapping vector $\boldsymbol{\varphi} \in \mathbb{R}^{G\times 1}$:

$$\mathrm{SE}(m, \boldsymbol{\varphi}) = \log_2(1 + \frac{|[\overline{\mathbf{P}_{\varphi}}]_{([\boldsymbol{\varphi}]_{\lceil \frac{m}{M'} \rceil}, m)}|^2}{\sigma^2}) \tag{2.29}$$

where $\overline{\mathbf{P}_{\varphi}}$ is the synthesized beampattern and σ^2 is the variance of the noise, assuming no pathloss, $\sigma^2 = \frac{1}{\mathrm{SNR}}$.

2.6.1 Computational Complexity and Memory Analysis

In this section, we investigate the computational complexity and memory requirements of HDB and benchmark algorithms. The proposed HDB algorithm has the complexity of

Table 2.2: Practical Deployment Requirements of the Proposed HDB and Benchmark Algorithms.

Algorithms	Computational Complexity	Dict. Size	Dict. Memory for 3 UE	Average Runtime for 3 UE
Proposed HDB	$\mathcal{O}(NG)$	$\mathbb{R}^{N\times 2\times(2A-1)}$	$\approx$ 250 KB	2.4×10^{-8} s
JPTA Approximation **[Online]** [9]	$\mathcal{O}(MNI_{\text{it}})$	0	0	4.7×10^{-2} s
mmFlexible Math **[Online]** [14]	$\mathcal{O}(NG)$	0	0	3.7×10^{-8} s
JPTA Approximation **[Dictionary]** [9]	$\mathcal{O}(1)$	$\mathbb{R}^{N\times 2\times A^G}$	$\approx$ 32 GB	NR
JPTA Line Search **[Dictionary]** [9]	$\mathcal{O}(1)$	$\mathbb{R}^{N\times 2\times A^G}$	$\approx$ 32 GB	NR
FSDA **[Dictionary]** [14]	$\mathcal{O}(1)$	$\mathbb{R}^{N\times 2\times A^G}$	$\approx$ 32 GB	NR

$\mathcal{O}(NG)$ and requires a dictionary of size $\mathbb{R}^{N\times 2\times(2A-1)}$, where the $(2A-1)$ entries correspond to all possible direction differences. We discuss two different deployment scenarios of the benchmark algorithms: *online* and *dictionary-based*. The *online* deployment refers to the case where algorithms calculate array configuration matrices in real-time, and *dictionary-based* deployment refers to the case where the array configuration matrices are obtained from the pre-calculated configuration matrix dictionary of all possible UE configurations. FSDA [14] and JPTA line search [9] algorithms are exhaustive search algorithms and are impractical for *online* deployment; therefore, we only discuss *dictionary-based* implementation of these algorithms in the following sections and Table 2.2.

The *dictionary-based* deployment of the JPTA [9] and mmFlexible [14] algorithms require the storage of the array configuration matrix of all possible UE direction configurations, amounting to A^G possible cases. This corresponds to a matrix of size $\mathbb{R}^{N\times 2\times A^G}$. Even for a small number of UEs, such as for $G = 3$ and for system configuration at Table 2.1, the required dictionary size is $\approx$ 32 GB for 64 bit floating-point variables, and it increases **exponentially** with the number of UEs, making *dictionary-based* deployment impractical. On the contrary, the HDB algorithm only requires a negligible $\approx$ 250 KB for the same system configuration, and the size of the required memory is **independent** of the number of UEs. The dictionary size and memory requirements are summarized in Table 2.2.

As the *dictionary-based* deployment is shown to be impractical, we continue with *online* deployment cases of the benchmark algorithms. In such a scenario, the computational complexity of the JPTA Approximation algorithm [9] is $\mathcal{O}(NMI_{\text{it}})$ and the computational complexity of both HDB and mmFlexible math algorithms is $\mathcal{O}(NG)$. Since the number of subcarriers is much larger than the number of UEs, i.e., $M >> G$, the computational complexity of the JPTA Approximation [9] is expected to be prohibitively higher.

To further assess the computational requirements, we analyze the average runtime of the HDB and *online* deployment cases of the benchmark algorithms. For simulations, we utilize MATLAB *version 23.2.0.2428915 (R2023b) Update 4* on the *Linux Ubuntu* operating system comprising *AMD Ryzen Threadripper 3970X 32-Core* Processor. Fig. 2.6 shows the average runtime for different numbers of UEs, and Table 2.2 demonstrates the average runtime for $G = 3$ UEs. Consistent with the complexity analysis above, the JPTA Approximation [9] algorithm, even for a single iteration, is orders of magnitude more computationally demanding than mmFlexible math [14] and the proposed HDB algorithms. Both HDB and mmFlexible math [14] algorithms perform similarly with very low runtime, making both algorithms low-complexity and practical alternatives to the JPTA Approximation algorithm [9].

2.6.2 Performance Analysis Over Subcarriers

We continue our analysis with the evaluation of the *spectral efficiency* given in the equation (2.29) of the HDB and *online* deployment of the benchmark algorithms for various parameters, including the number of UEs, antennas, and bandwidth.

Fig. 2.7 shows the ASE over different subbands for 3 UEs of 5000 Monte Carlo realizations. We observe that all algorithms perform very close to the maximum ASE at subbands 1 and 3 where JPTA approximation [9], mmFlexible math [14], and HDB algorithms achieve approximately $95\%, 93\%, 90\%$ of the maximum ASE, respectively. However, in the subband 2, JPTA approximation algorithm [9] with $I_{\text{it}} = 30$ iterations and mmFlexible math algorithm [14] achieves 83.35% and 84.53% of the maximum ASE, providing unfair ASE compared to the users assigned to subbands 1 and 3. Additionally, we observe that the JPTA approximation algorithm [9] has improved performance at subband 2 with the increasing number

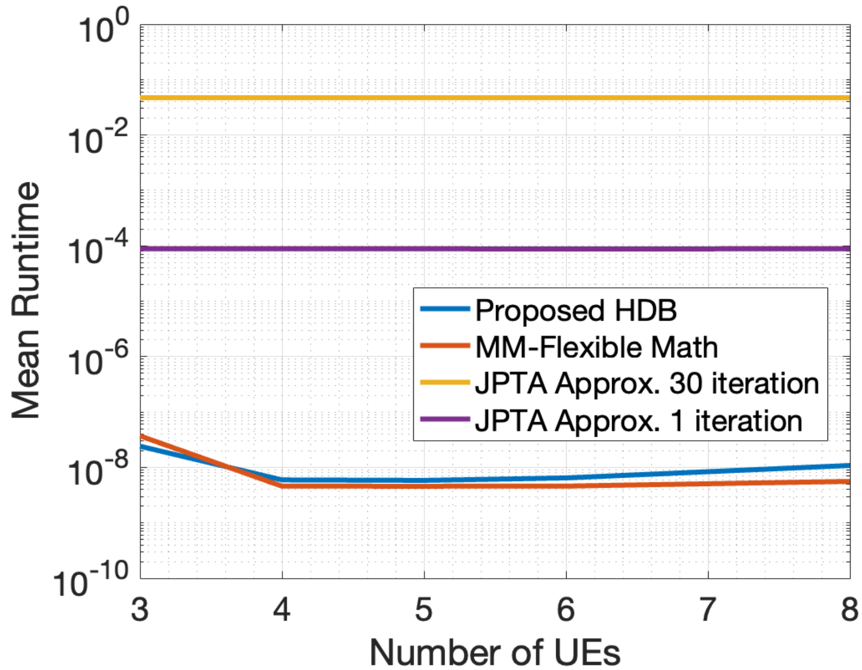


Figure 2.6: Average runtime of the proposed HDB algorithm and benchmark algorithms for different numbers of UEs, G, for $N = 16$ antennas and bandwidth $BW = 3$ GHz. Exhaustive search algorithms are not included, and single iteration, $I_{\text{it}} = 1$, JPTA Approximation algorithm [9] is included for comparison.

of iterations. On the other hand, HDB algorithm achieves 89.55% of the maximum ASE at the subband 2, providing similar and near-optimal performance across different subbands.

Fig. 2.8 demonstrates the ASE over different subcarriers for 3 UEs. We observe that subcarriers close to the center of each subband have higher spectral efficiency compared to subcarriers at the edge of each subband. JPTA approximation algorithm [9] has a slightly off-center maximum at each subband and has a significant difference between ASE across subcarriers: the maximum SE is 25% and 28% higher than the minimum SE for $I_{\text{it}} = 30$ and $I_{\text{it}} = 1$ iterations, respectively. Similarly, the mmFlexible math [14] algorithm achieves the maximum at the slightly off-center of each subband, and the maximum SE is 19% higher than the minimum SE. On the contrary, the proposed HDB algorithm achieves its maximum

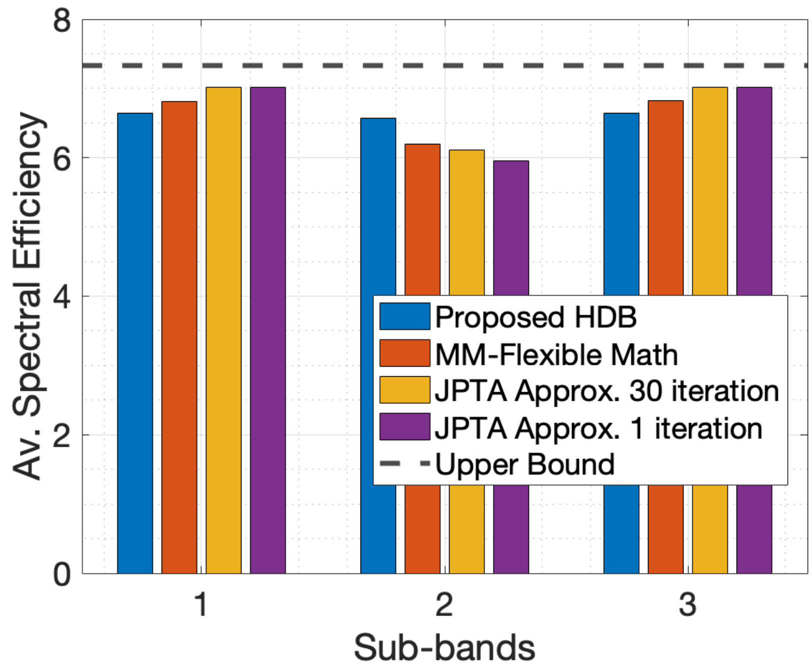


Figure 2.7: ASE of the proposed HDB algorithm and benchmark algorithms across subbands for $G = 3$ UEs, $N = 16$ antennas and bandwidth of $BW = 3$ GHz. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

close to the center for all subbands, with the maximum SE being only 7% higher than the minimum SE. Results indicate that the proposed HDB algorithm provides a more uniform ASE distribution across subcarriers.

Fig. 2.9 shows the empirical cdf of the ASE across different subcarriers and UE locations for 3 UEs. Across all realizations, i.e., subcarriers and UE locations, HDB algorithm provides higher minimum ASE and less variance compared to benchmark algorithms. JPTA approximation $I_{\text{it}} = 30$ iteration and mmFlexible math algorithms achieve less than 6 bps/Hz spectral efficiency for 13% and 10% of the realizations, while HDB achieves less than 6 bps/Hz spectral efficiency for 1% of the realizations. Results demonstrate that HDB algorithm provides consistent spectral efficiency across all realizations, making it a reliable alternative to the benchmark algorithms.

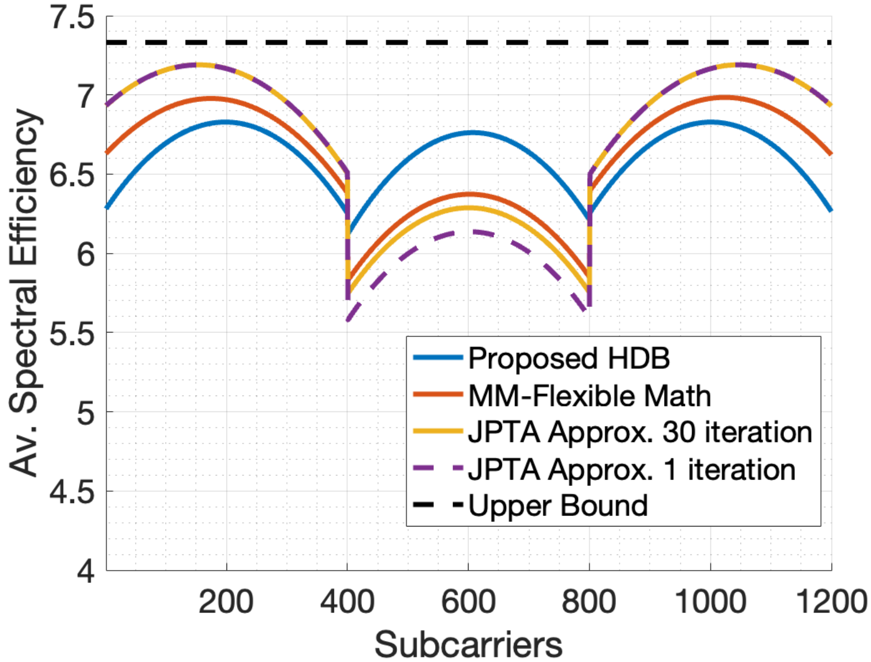


Figure 2.8: ASE of the proposed HDB algorithm and benchmark algorithms across different subcarriers for $G = 3$ UEs, $N = 16$ antennas and bandwidth of $BW = 3$ GHz. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

2.6.3 Impact of Array Parameters

In this section, we assess the HDB and benchmark algorithms for different array parameters and number of UEs.

Fig. 2.10 shows the ASE over different BW values for $G = 3$ UEs and $N = 16$ antennas. We can observe that the JPTA approximation algorithm [9] performs the best across all bandwidth values and is minimally affected by the increased bandwidth. While the mmFlexible algorithm's [14] performance degrades significantly with increased bandwidth, reducing from 90.98% of the maximum ASE at 1 GHz to 78.16% of the maximum ASE at 10 GHz bandwidth as a result of the beamsquint, which increases with bandwidth. HDB algorithm performs almost identically to the benchmark algorithms for $BW < 3$ GHz and gets less

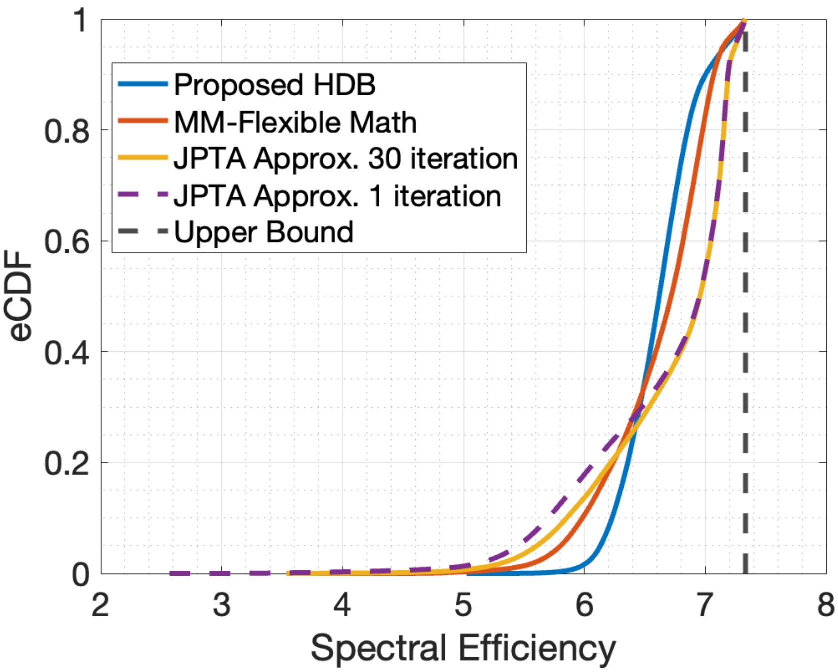


Figure 2.9: Empirical cumulative distribution of the spectral efficiency across different subcarriers and UE direction configurations for the proposed HDB algorithm and benchmark algorithms for 3 UEs for $G = 3$ UEs, $N = 16$ antennas and bandwidth of $BW = 3$ GHz. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

affected by the increased bandwidth compared to the mmFlexible algorithm [14]. This reduction results from the accumulation of generator errors, where the JPTA approximation [9] algorithm is utilized for generator dictionary creation. Results indicate that HDB algorithm outperforms mmFlexible math [14] algorithm for $BW > 3$ GHz and is less affected by the increased bandwidth.

Fig. 2.11 demonstrates the ASE over different numbers of antennas for $G = 3$ UEs and $BW = 3$ GHz bandwidth. HDB algorithm and benchmark algorithms perform almost identically for $N \leq 16$ antennas. We observe that the performance of the JPTA approximation algorithm [9] scales almost linearly with the number of antennas N, deviating slightly for $N \geq 64$. In contrast, mmFlexible math algorithm [14] and HDB algorithms plateau and

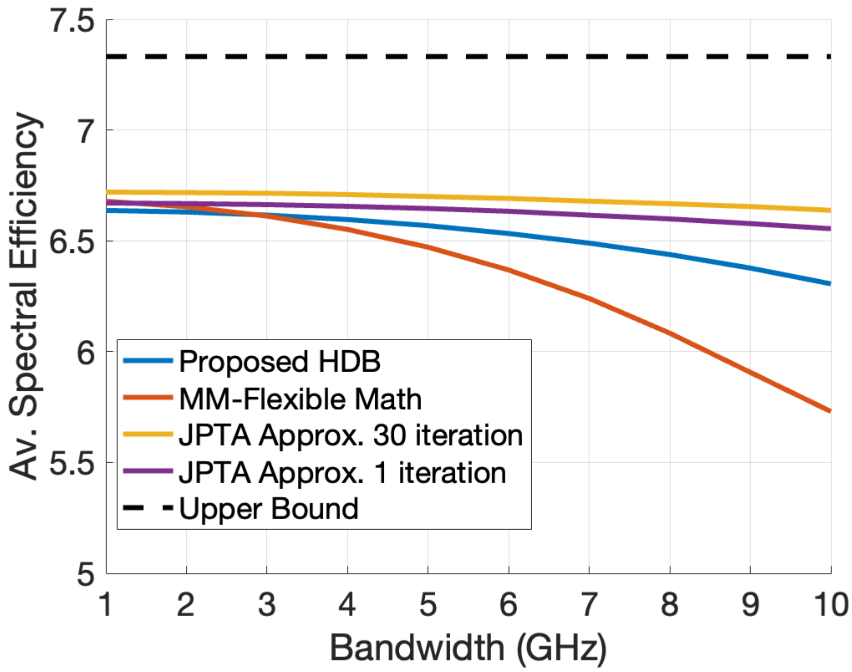


Figure 2.10: ASE of the proposed HDB algorithm and benchmark algorithms for different bandwidths BW for $G = 3$ UEs, $N = 16$ antennas. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

degrade after $N = 32$ and $N = 64$ antennas, respectively. As the number of antennas increases, the corresponding beamwidth decreases, and algorithms become more sensitive to errors, i.e., beam squint as in mmFlexible [14] and generator error accumulation as in HDB algorithm. We observe that HDB algorithm outperforms the mmFlexible math algorithm for $N > 32$ antennas.

Fig. 2.12 shows the ASE over different numbers of UEs for $N = 16$ antennas and $BW = 3$ GHz bandwidth. We observe that HDB and benchmark algorithms perform similarly for $G \leq 4$ UEs and ASE decreases with the number of UEs for all algorithms. Results indicate that utilization of split beampatterns is more suitable for a small number of UE deployments.

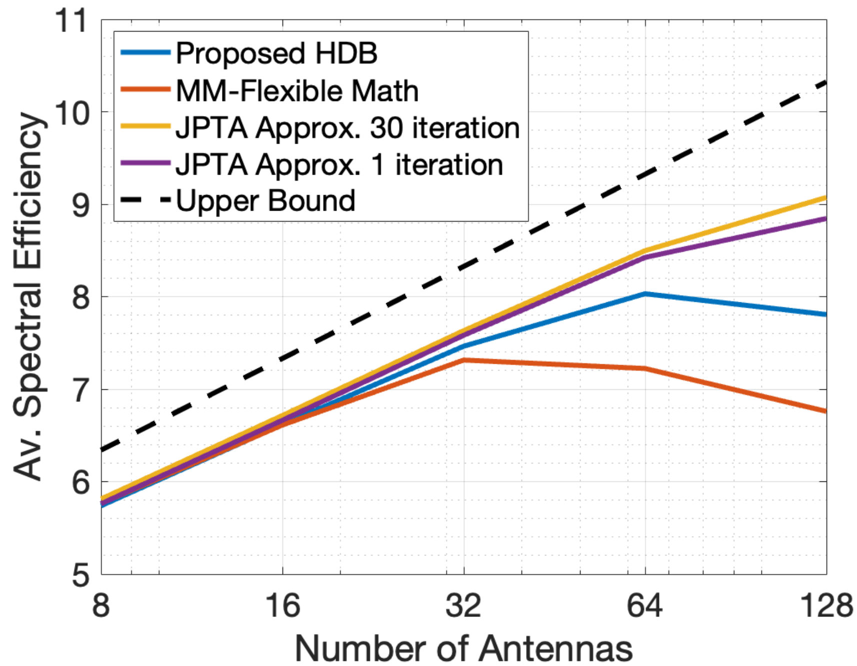


Figure 2.11: ASE of the proposed HDB algorithm and benchmark algorithms for different numbers of antennas N for $G = 3$ UEs and $BW = 3$ GHz bandwidth. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

2.7 Discussion and Future Directions

The HDB algorithm is a low complexity and low-memory algorithm for split beampattern synthesis that is suitable for practical deployment. We showed that the proposed HDB algorithm is less affected by the increased bandwidth and number of antennas compared to the mmFlexible algorithm [14] while having similar computational complexity and negligible memory requirements. In addition, we empirically observe that HDB algorithm provides higher minimum gain and fairness among users due to the post-processing of the generator beampattern dictionary discussed in Section 2.5.4. This property not only makes the HDB more attractive to the systems that are aiming for fair service across UEs but can also reduce the complexity of the upper layer scheduling and assignment algorithms.

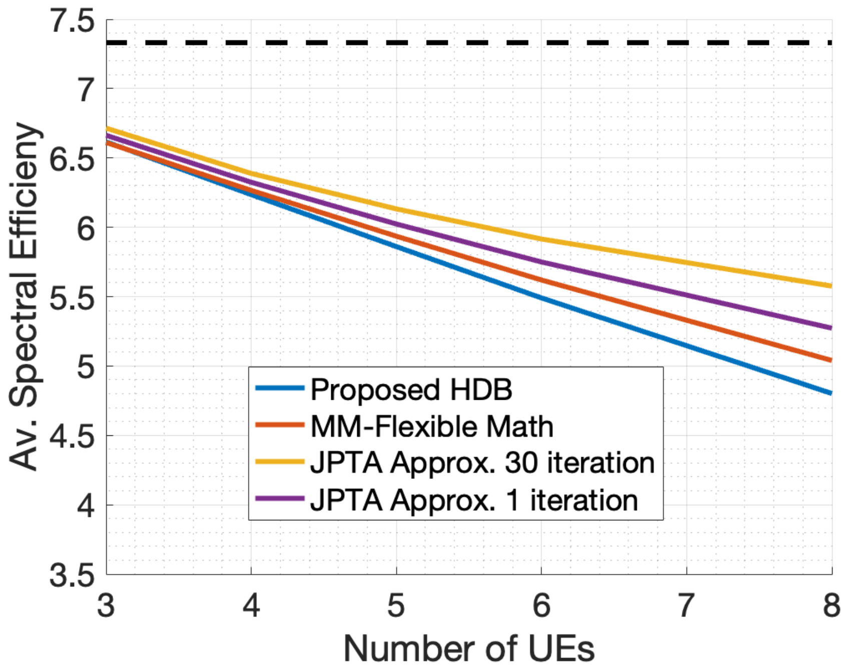


Figure 2.12: ASE of the proposed HDB algorithm and benchmark algorithms for different numbers of UEs G for $N = 16$ antennas and $BW = 3$ GHz bandwidth. The *Upper Bound* refers to the ASE achieved with the maximum beamforming gain.

In the future, we plan to optimize the shift and scale amount in equation (2.26) for various objective functions over the desired UE configurations, adapting the HDB algorithm for different deployment scenarios. The optimization of the generator beampattern dictionary for different objective functions is also an open problem. Lastly, we believe that the investigation of the extension of the observed mathematical structure to hybrid TTD arrays is an interesting research direction.

2.8 Conclusion

In this chapter, we present a low-complexity and low-memory split beampattern approximation algorithm utilizing the mathematical structure of the beampattern synthesis operation.

We first demonstrate the homomorphism between array configuration matrices and corresponding TTD beampatterns. Then, we represent the desired hard-to-approximate split beampattern in terms of easy-to-approximate generator beampatterns. Finally, by utilizing a single generator beampattern dictionary, we approximate the desired split beampattern with the observed homomorphism in a divide-and-conquer manner. The proposed algorithm achieves comparable performance to the high-complexity or high-memory JPTA algorithms [9] with orders of magnitude smaller memory requirements, without ignoring the beam squint as the mmFlexible algorithm [14]. Furthermore, with extensive simulations, we show that the proposed algorithm achieves higher minimum power and fairness across different UEs compared to benchmark algorithms.

CHAPTER 3

Homomorphic Directional Beamforming for Analog True-Time-Delay Arrays with Parameter Learning

3.1 Introduction

The use of millimeter-wave (mmWave) and sub-THz frequency bands has emerged as a key enabler for next-generation wireless systems due to the abundance of available bandwidth. To efficiently exploit these frequencies, large-scale antenna arrays are employed to provide high beamforming gains that compensate for severe path loss. In this context, analog beamforming architectures based on frequency-independent phase shifters are particularly attractive due to their low hardware complexity, deployment cost, and energy consumption [6]. However, such analog arrays cannot independently map different subcarriers or subbands to different directions, which limits scheduling flexibility. Moreover, as both the bandwidth and the number of antennas increase, these arrays exhibit a frequency-dependent response [20], commonly referred to as beam squint.

To address these limitations, analog true-time-delay (TTD) circuit elements have been proposed to enable TTD-based beamforming, providing explicit control over frequency-dependent beampatterns [21,22]. Leveraging this capability, analog TTD arrays can simultaneously serve multiple UEs in different spatial directions via subband-to-direction multiplexing, resulting in split beampatterns [9], as illustrated in Fig. 3.1. Several optimization-based methods have been developed to synthesize such beampatterns, enabling flexible subcarrier-to-direction assignments and supporting functionalities such as beam-squint compensation, rainbow beamforming, and split beampattern realization. However, these approaches either incur prohibitive computational complexity or memory requirements as in [9], or ignore the frequency dependency of the array response [14], limiting their practicality for real-time deployment in wideband systems. Heuristic algorithms have been proposed as lower-complexity alternatives [15, 16], but they typically rely on restrictive subcarrier-to-direction mappings and lack flexibility under practical deployment constraints, such as a limited number of users or constrained spatial distributions.

In this paper, we build on the HDB algorithm [17], which exploits the mathematical structure of TTD split beampatterns to synthesize them from a single low-memory dictionary, providing a lightweight alternative. However, the parameters of HDB are not adapted to user direction assignments, resulting in reduced average received power. To address this limitation, we propose a data-driven parameter learning framework for HDB to maximize the average received power. We analyze the proposed method with respect to average received power and fairness across subbands and quantify its performance relative to the benchmarks [9, 14] using Monte Carlo simulations. In addition, we analyze its implementation cost with respect to computational complexity and memory requirements to demonstrate its feasibility in practical deployments.

3.2 System Model and Problem Formulation

We consider a wireless system in which a base station, equipped with a single RF chain, employs an N-element ULA with half-wavelength antenna spacing $\frac{\lambda_c}{2}$ at the carrier frequency f_c. Each antenna element is equipped with a phase shifter, characterized by phase ϕ_n, and a TTD unit with delay t_n [9,14]. Data transmission is carried out using an M-subcarrier OFDM system with total bandwidth BW, where the frequency of subcarrier $m = 0, \ldots, M-1$ is given by $f_m = f_c + m\frac{BW}{M-1} - \frac{BW}{2}$.

The resulting frequency-dependent beamforming vector $\mathbf{v}_m \in \mathbb{C}^{N\times 1}$ for subcarrier $m =$

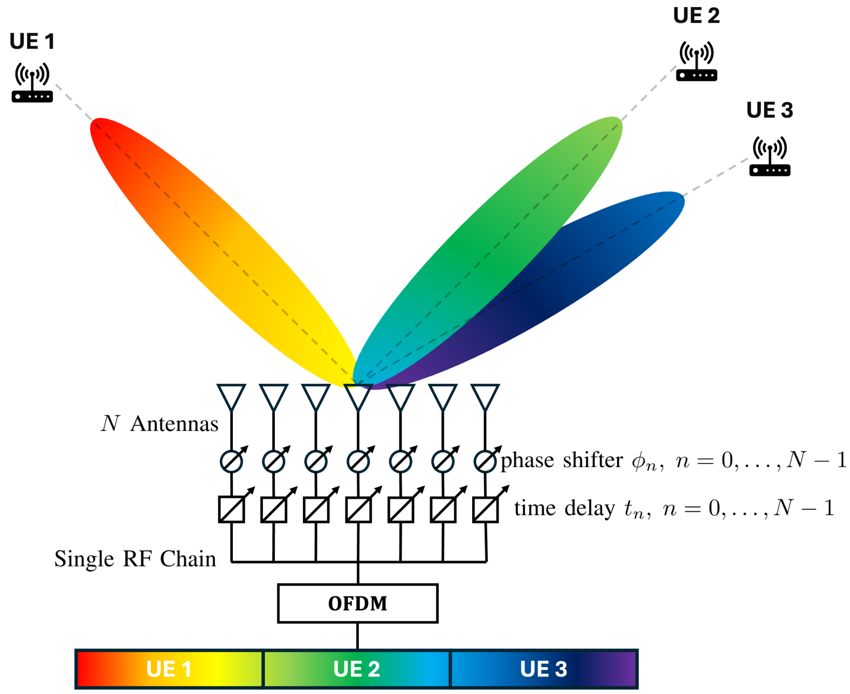


Figure 3.1: Split beampatterns simultaneously serving 3 UEs with a single RF chain by subband-to-UE (direction) assignment with TTD arrays [17].

$0, \ldots, M-1$ and for time delay t_n and phase shift ϕ_n per antenna $n = 0, \ldots, N-1$ can be defined as follows:

$$[\mathbf{v}_m]_n = \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} \in \mathbb{C},\ \forall m, n. \tag{3.1}$$

The beamforming matrix composed of beamforming vectors across all subcarriers is defined as:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_0 & \ldots & \mathbf{v}_{M-1} \end{bmatrix} \in \mathbb{C}^{N \times M}. \tag{3.2}$$

The corresponding array response for given time delays t_n and phase shifts ϕ_n, $n = 0, \ldots, N-1$, at subcarrier m is [9]:

$$a_m(\Psi) = \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} e^{-jn\pi\Psi\frac{f_m}{f_c}} \tag{3.3a}$$

$$= \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)} e^{-jn\Omega_m} \tag{3.3b}$$

$$= \mathcal{F}\{\frac{1}{\sqrt{N}} e^{j(-2\pi f_m t_n + \phi_n)}\}, \tag{3.3c}$$

where θ is the angle of departure measured from the array broadside, $\Psi = \sin(\theta) \in [-1, 1]$ is the corresponding direction variable, and $\mathcal{F}$ is the DTFT with the frequency variable $\Omega_m = \pi\Psi f_m/f_c$. Then, we define the array configuration matrix as $\mathbf{\Phi} = \begin{bmatrix} \mathbf{t} & \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^{N\times 2}$, where $[\mathbf{t}]_n = t_n,\ [\boldsymbol{\phi}]_n = \phi_n,\ n = 0, \ldots, N-1$ are the time delay and phase shifter vectors. The TTD beampattern $\mathbf{P_\Phi}$ is then given by $[\mathbf{P_\Phi}]_{(\Psi,m)} = a_m(\Psi)$, where each column $[\mathbf{P_\Phi}]_{(:,m)}$, $m = 0, \ldots, M-1$, is a continuous function of $\Psi \in [-1, 1]$. In the practical implementation, we sample $\Psi \in [-1, 1]$ on a uniform grid of A points, so that $\mathbf{P_\Phi}$ is represented by a matrix in $\mathbb{C}^{A\times M}$.

Our objective is to find the array configuration matrix that generates the TTD beamforming matrix $\mathbf{V_\Phi} \in \mathbb{C}^{N\times M}$ according to (3.1)–(3.2) and minimizes the following cost function:

$$\min_{\mathbf{\Phi}\in\mathbb{R}^{N\times 2}} \sum_m \|[\mathbf{V_\Phi}]_{(:,m)} - [\mathbf{V}_{target}]_{(:,m)}\|^2, \tag{3.4}$$

where $\mathbf{V}_{target} = \mathcal{F}^{-1}\{\mathbf{P}_{target}\}$ is the desired beamforming matrix obtained from the desired beampattern $\mathbf{P}_{target}$. The feasible set of TTD beamforming matrices in problem (3.4) is nonconvex and the problem is intractable in general.

In this paper, we focus on the approximation of split beampatterns, which are defined as beampatterns over a frequency band partitioned into G subbands, where the g-th subband is mapped to the direction ψ_g. The subband-to-direction mapping is specified by the vector $\boldsymbol{\psi} \in [-1, 1]^{G\times 1}$. Since split beampatterns are parametrized by $\boldsymbol{\psi}$, we denote them as $\mathbf{P}_{target} = \mathbf{P}(\boldsymbol{\psi})$. $\mathbf{P}(\boldsymbol{\psi})$ is defined as the beampattern generated by the following far-field beamforming matrix $\mathbf{V}(\boldsymbol{\psi}) \in \mathbb{C}^{N\times M}$ [9]:

$$
\begin{aligned}
[\mathbf{V}_{target}]_{(n,m)} &= [\mathbf{V}(\boldsymbol{\psi})]_{(n,m)} = \frac{1}{\sqrt{N}} e^{j\pi n \overline{\psi}_m \frac{f_m}{f_c}},\ \forall m, n \\
\overline{\boldsymbol{\psi}} &= \Big[\underbrace{\psi_1 \quad \ldots \quad \psi_1}_{M'} \quad \cdots \quad \underbrace{\psi_G \quad \ldots \quad \psi_G}_{M'}\Big]^T \in [-1, 1]^{M\times 1},
\end{aligned}
\tag{3.5}
$$

where the subcarriers are partitioned into G subbands, each containing $M' = \frac{M}{G}$ subcarriers, and subcarrier m is directed toward direction $\overline{\psi}_m = [\overline{\boldsymbol{\psi}}]_m$. The corresponding target beampattern $\mathbf{P}(\boldsymbol{\psi})$ is obtained as follows:

$$
[\mathbf{P}(\boldsymbol{\psi})]_{(\Psi,m)} = \frac{1}{\sqrt{N}} e^{-j\frac{(N-1)\pi f_m(\Psi - \overline{\psi}_m)}{2f_c}}\, \Xi_{N,m}(\Psi - \overline{\psi}_m), \tag{3.6}
$$

where $\Xi_{N,m}(x) = \frac{\sin(\frac{Nx\pi f_m}{2f_c})}{\sin(\frac{x\pi f_m}{2f_c})}$ is the Dirichlet sinc function for N antennas and f_m [4]. Split beampatterns are not realizable by TTD arrays in general [9], and approximating one requires solving the intractable optimization problem (3.4). Yet, by exploiting the mathematical properties of TTD beampatterns (Section 3.3) and split beampatterns (Section 3.4), we can find a low-complexity and structured alternative.

3.3 Homomorphism over TTD Beampatterns

We continue by showing that the beampattern generation map $\boldsymbol{\Phi} \mapsto \mathbf{P}_{\boldsymbol{\Phi}}$ is a homomorphism, i.e., an operation-preserving map, between the array configuration matrices under standard addition, $\langle \mathbb{R}^{N\times 2}, + \rangle$, and the beampatterns realizable with TTD arrays under the scaled column-wise circular convolution operation $\star$ defined below, $\langle \mathcal{P}_{\mathcal{T}}, \star \rangle$, where $\mathcal{P}$ is the set of beampatterns generated by beamforming matrices with non-zero entries, which correspond to non-zero gain at every antenna and for every subcarrier [17], and $\mathcal{P}_{\mathcal{T}} \subset \mathcal{P}$ is the subset generated by the TTD beamformers $\mathcal{T} := \{\mathbf{V}_{\boldsymbol{\Phi}} : \boldsymbol{\Phi} \in \mathbb{R}^{N\times 2}\}$.

$$
\begin{aligned}
&\text{Let } \mathbf{P}', \mathbf{P}'' \in \mathcal{P}, \\
&\star : \mathcal{P} \times \mathcal{P} \to \mathcal{P}, \\
&[\mathbf{P}' \star \mathbf{P}'']_{(:,m)} := \sqrt{N} \left([\mathbf{P}']_{(:,m)} \circledast [\mathbf{P}'']_{(:,m)} \right), \forall m,
\end{aligned} \tag{3.7}
$$

where the operation $\circledast$ is the circular convolution over one period of Ω_m, normalized by the period. Then, we can observe that:

Theorem 3.1 ([17]). *The mapping $\mathbf{\Phi} \mapsto \mathbf{P}_{\mathbf{\Phi}}$ is a homomorphism between $\langle \mathbb{R}^{N\times 2}, + \rangle$ and $\langle \mathcal{P}_{\mathcal{T}}, \star \rangle$: $\forall \mathbf{\Phi}_1, \mathbf{\Phi}_2 \in \mathbb{R}^{N\times 2}, \mathbf{P}_{\mathbf{\Phi}_1 + \mathbf{\Phi}_2} = \mathbf{P}_{\mathbf{\Phi}_1} \star \mathbf{P}_{\mathbf{\Phi}_2}$.*

Proof. The proof follows from the multiplication property of the DTFT; see [17]. □

This theorem provides a powerful tool to synthesize TTD beampatterns. Let us consider a TTD beampattern $\mathbf{P}_{\mathbf{\Phi}'} \in \mathcal{P}_{\mathcal{T}}$ and assume that it can be represented with G TTD beampatterns over the $\star$ operation as follows:

$$
\mathbf{P}_{\mathbf{\Phi}'} = \mathbf{P}_{\mathbf{\Phi}_1} \star \cdots \star \mathbf{P}_{\mathbf{\Phi}_G}. \tag{3.8}
$$

Then, Theorem 3.1 implies that a corresponding array configuration matrix can be obtained as:

$$
\mathbf{\Phi}' = \sum_{g=1}^{G} \mathbf{\Phi}_g \tag{3.9}
$$

and the time delay $\mathbf{t}'$ and phase shifter values $\boldsymbol{\phi}'$ follow directly as:

$$
\mathbf{t}' = \sum_{g=1}^{G} \mathbf{t}_g, \quad \boldsymbol{\phi}' = \sum_{g=1}^{G} \boldsymbol{\phi}_g. \tag{3.10}
$$

Therefore, finding a representation of a TTD beampattern over the $\star$ operation is sufficient to determine an array configuration matrix that generates it. As discussed, split beampatterns can be outside $\mathcal{P}_{\mathcal{T}}$ [9]; hence, Theorem 3.1 is not directly applicable. However, due to the properties of split beampatterns, we can decompose the desired split beampattern into simpler beampatterns, each of which can be deterministically approximated by TTD beampatterns. This allows us to utilize Theorem 3.1 to synthesize split beampatterns without solving problem (3.4).

3.4 Properties of Split Beampatterns

Split beampatterns exhibit an important property that enables us to represent them with G simpler beampatterns, which can be deterministically approximated in an efficient way. Under the $\star$ operation (which acts as an addition on $\mathcal{P}$), the subband-to-direction mapping vectors of split beampatterns are added, as the following lemma shows.

Lemma 3.1 ([17]). *Let $\mathbf{P}(\boldsymbol{\psi}_1), \mathbf{P}(\boldsymbol{\psi}_2)$ be split beampatterns. Then*

$$\mathbf{P}(\boldsymbol{\psi}_1 + \boldsymbol{\psi}_2) = \mathbf{P}(\boldsymbol{\psi}_1) \star \mathbf{P}(\boldsymbol{\psi}_2).$$

Proof. The proof follows from the properties of the Dirichlet sinc function; see [17]. $\square$

3.4.1 Split Beampattern Generators

We now define the split beampattern generators. The generators are split beampatterns, $\mathbf{P}(\boldsymbol{\psi} = \boldsymbol{\Delta}_g),\ g = 1, \ldots, G$, defined with the following subband-to-direction mapping vector:

$$\boldsymbol{\Delta}_g = [\underbrace{0, \ldots, 0}_{g-1}, \underbrace{\Delta_g, \ldots, \Delta_g}_{G-g+1}]^T,\ g = 1, \ldots, G. \tag{3.11}$$

We define a separate notation $\boldsymbol{\Delta}$ for the split beampattern subband-to-direction mapping, which is of the same type as $\boldsymbol{\psi}$, for ease of exposition. The values Δ_g are selected based on the target directions ψ_g according to the following relation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & \ddots & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \vdots \\ \Delta_G \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_G \end{bmatrix} \tag{3.12}$$

such that they satisfy $\boldsymbol{\psi} = \boldsymbol{\Delta}_1 + \cdots + \boldsymbol{\Delta}_G$; the solution of (3.12) is [17]:

$$\begin{aligned} \Delta_1 &= \psi_1 \\ \Delta_g &= \psi_g - \psi_{g-1},\ g = 2, \ldots, G. \end{aligned} \tag{3.13}$$

Since Δ_g represents a direction, it is periodic with period 2 at f_c and must satisfy $|\Delta_g| \leq 1$ for $g \geq 2$ to avoid periodic wrapping. If wrapping occurs, both Δ_g and Δ_1 are updated; the implementation details can be found in [17].

By Lemma 3.1, any split beampattern $\mathbf{P}(\boldsymbol{\psi})$ can be decomposed into the split beampattern generators $\mathbf{P}(\boldsymbol{\Delta}_g),\ g = 1, \ldots, G$ over the $\star$ operation as follows:

$$\mathbf{P}(\boldsymbol{\psi}) = \mathbf{P}(\boldsymbol{\Delta}_1 + \cdots + \boldsymbol{\Delta}_G) = \mathbf{P}(\boldsymbol{\Delta}_1) \star \cdots \star \mathbf{P}(\boldsymbol{\Delta}_G), \tag{3.14}$$

where $\Delta_g,\ g = 1, \ldots, G$ are obtained from (3.13). The illustration of (3.14) for $G = 3$ can be found in Fig. 3.2.

As discussed, split beampattern generators for $g \geq 2$ with $\Delta_g \neq 0$ cannot be realized by TTD arrays due to the discontinuity of directions between different subbands [9]. We instead approximate each such generator with a TTD array using existing algorithms. Since the approximated beampatterns are in $\mathcal{P}_{\mathcal{T}}$, Theorem 3.1 applies, and we sum the resulting array configuration matrices to obtain an approximation of the split beampattern. This divide-and-conquer strategy circumvents the need to approximate the full split beampattern directly, and allows us to exploit both the properties of split beampatterns (Section 3.4) and of TTD beampatterns (Section 3.3), yielding a low-complexity and structured alternative to (3.4).

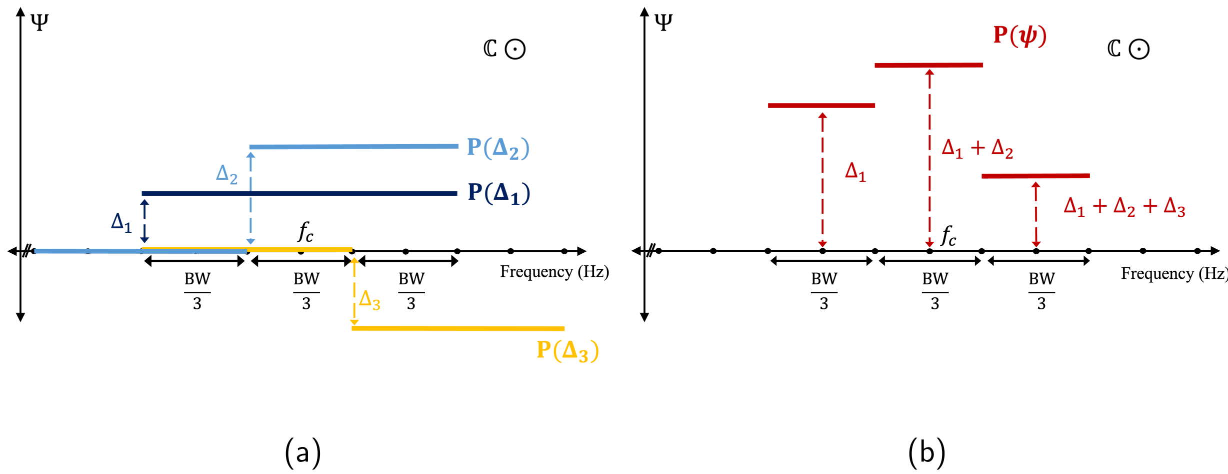


Figure 3.2: (a) Ideal split beampattern generators for $G = 3$ UEs; the generators for $g \geq 2$ are created from frequency-shifted and bandwidth-scaled two-direction split beampatterns. (b) Beampattern obtained by applying the $\star$ operation to the generator beampatterns [17].

3.5 Generator Approximation and HDB Algorithm

We continue with the construction of the generator approximations. The generator $\mathbf{P}(\boldsymbol{\Delta}_1)$ has the same direction assignment across all subbands, and can therefore be realized in closed form by TTD arrays with the following array configuration matrix [7]:

$$\boldsymbol{\Phi}_{\boldsymbol{\Delta}_1} = \begin{bmatrix} \frac{-\Delta_1 \mathbf{n}}{2 f_c} & \mathbf{0} \end{bmatrix}, \; \mathbf{n} = \begin{bmatrix} 0 & \dots & N-1 \end{bmatrix}^T. \tag{3.15}$$

Split beampattern generators for $g \geq 2$ contain at most two distinct directions. Therefore, the g-th generator can be approximated as a $G = 2$ split beampattern with the following center frequency $f_c^{\Delta_g}$ and bandwidth BW^{Δ_g} [17]:

$$\begin{gathered} \alpha_g = 2\frac{G-1}{G}, \qquad\qquad \beta_g = -\frac{1}{2} + \frac{g-1}{G}, \\ BW^{\Delta_g} = \alpha_g \, BW, \quad f_c^{\Delta_g}(\beta_g) = f_c + \beta_g \, BW, \end{gathered} \tag{3.16}$$

where α_g is the bandwidth scaling and β_g is the frequency offset, which are fixed values

Algorithm 2: HDB Algorithm [17]

Input: Desired split beampattern $\mathbf{P}(\boldsymbol{\psi})$ with $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_G]^T \in [-1, 1]^{G\times 1}$

Output: Array configuration matrix $\boldsymbol{\Phi}_{\boldsymbol{\psi}}$ such that $\mathbf{P}_{\boldsymbol{\Phi}_{\boldsymbol{\psi}}} \approx \mathbf{P}(\boldsymbol{\psi})$

Solve (3.12) with (3.13) and obtain $\Delta_g,\ g = 1, \ldots, G$

Obtain $\boldsymbol{\Phi}_{\boldsymbol{\Delta}_g}$ for $g = 2, \ldots, G$ from the dictionary $\boldsymbol{\Upsilon}$ with (3.16) and (3.17), and $\boldsymbol{\Phi}_{\boldsymbol{\Delta}_1}$ with (3.15)

Add the time delay and phase shifter vectors as in (3.10): $\mathbf{t} = \sum_{g=1}^{G} \mathbf{t}_g, \boldsymbol{\phi} = \sum_{g=1}^{G} \boldsymbol{\phi}_g$

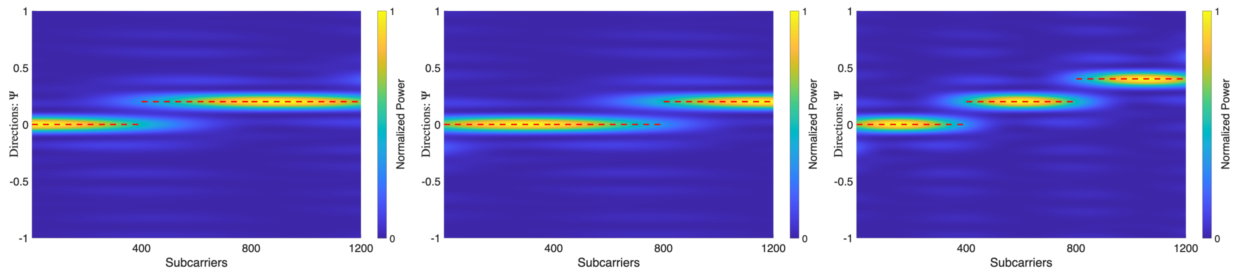
return $\boldsymbol{\Phi}_{\boldsymbol{\psi}} = \begin{bmatrix}\mathbf{t} & \boldsymbol{\phi}\end{bmatrix}$

determined by the structure of the generator beampatterns. Since the g-th generator can be viewed as a shifted (in center frequency) and scaled (in bandwidth) version of a $G = 2$ split beampattern with subband-to-direction mapping $[0, \Delta_g]^T$ and array configuration matrix $\tilde{\boldsymbol{\Phi}}$, we can store it at f_c with bandwidth BW. The desired generator can then be approximated by shifting its center frequency to $f_c^{\Delta_g}(\beta_g)$ and rescaling its bandwidth to BW^{Δ_g}, which corresponds to the following linear transformation over the array configuration matrices [15, 17]:

$$\boldsymbol{\Phi}_{\boldsymbol{\Delta}_g} = \tilde{\boldsymbol{\Phi}} \begin{bmatrix} \frac{1}{\alpha_g} & -2\pi \frac{\alpha_g f_c - f_c^{\Delta_g}(\beta_g)}{\alpha_g} \\ 0 & 1 \end{bmatrix} = \tilde{\boldsymbol{\Phi}} \mathbf{S}(\alpha_g, \beta_g),\ g \geq 2, \tag{3.17}$$

where $\boldsymbol{\Phi}_{\boldsymbol{\Delta}_g}$ is the array configuration matrix that approximates the g-th generator beampattern $\mathbf{P}(\boldsymbol{\Delta}_g)$ defined in (3.11). Equations (3.16)–(3.17) approximate each generator for $g \geq 2$ as a $G = 2$ split beampattern with modified center frequency and bandwidth. Then, we can pre-calculate the $G = 2$ configurations only once, offline, and reuse them for any G. The dictionary size is negligible, as shown in Table 3.1.

The dictionary of $G = 2$ split beampatterns, which we call the *generator dictionary* $\boldsymbol{\Upsilon} \in \mathbb{R}^{N\times 2\times(2A-1)}$, is constructed as follows. For each of the $(2A - 1)$ possible generator direction assignments, which correspond to all possible direction differences, the dictionary



(a) $|\mathbf{P}_{\boldsymbol{\Phi}_{\boldsymbol{\Delta}_2}}|^2$ (b) $|\mathbf{P}_{\boldsymbol{\Phi}_{\boldsymbol{\Delta}_3}}|^2$ (c) $|\mathbf{P}_{\boldsymbol{\Phi}_{\boldsymbol{\Delta}_2}+\boldsymbol{\Phi}_{\boldsymbol{\Delta}_3}}|^2$

Figure 3.3: Subfigures (a) and (b) show the generator beampatterns approximated by the mmFlexible [14] algorithm with $\boldsymbol{\Delta}_2 \approx [0, 0.2, 0.2]^T$ and $\boldsymbol{\Delta}_3 \approx [0, 0, 0.2]^T$, respectively, and (c) shows the resulting beampattern with $\boldsymbol{\Delta}_2 + \boldsymbol{\Delta}_3 \approx [0, 0.2, 0.4]^T$ after summing the corresponding array configuration matrices [17]. Dashed red lines indicate the target directions. Here, the generators are created with the mmFlexible algorithm for illustrative purposes and the colorbars show normalized power over the same range $[0, 1]$.

entry is the array configuration matrix $\boldsymbol{\Phi}_{\underline{\boldsymbol{\Delta}}}$ that approximates the split beampattern with subband-to-direction mapping $\underline{\boldsymbol{\Delta}} = [\underline{\Delta}_1, \underline{\Delta}_2]^T$, $\underline{\Delta}_1, \underline{\Delta}_2 \in [-1, 1]$; in particular, the entry for $\underline{\boldsymbol{\Delta}} = [0, \Delta_g]^T$ is the matrix $\tilde{\boldsymbol{\Phi}}$ used in (3.17) for the g-th generator. When $\underline{\Delta}_1 \neq \underline{\Delta}_2$, $\boldsymbol{\Phi}_{\underline{\boldsymbol{\Delta}}}$ is obtained with the joint phase-time array (JPTA) approximation algorithm [9]; otherwise, with (3.15). The fidelity of the approximated split beampatterns is directly dependent on the fidelity of the approximated generator beampatterns. For $g \geq 2$ with $\Delta_g \neq 0$, the per-subcarrier power within each subband is non-uniform, and the maximum power is not necessarily located at the subband center [17]. Therefore, we need to post-process the split beampatterns such that the maximum gain is attained at the center of each subband [17]. Fig. 3.3(c) illustrates the effect of adding the array configuration matrices of two approximated generators.

The overall process is illustrated in Fig. 3.2 for $G = 3$ UEs served in 3 subbands. To approximate the generator beampattern $\mathbf{P}(\boldsymbol{\Delta}_g)$, we retrieve the array configuration matrix for the corresponding directions from the dictionary $\boldsymbol{\Upsilon}$ and shift its center frequency and scale its bandwidth with (3.16) and (3.17). Since the desired split beampattern is represented

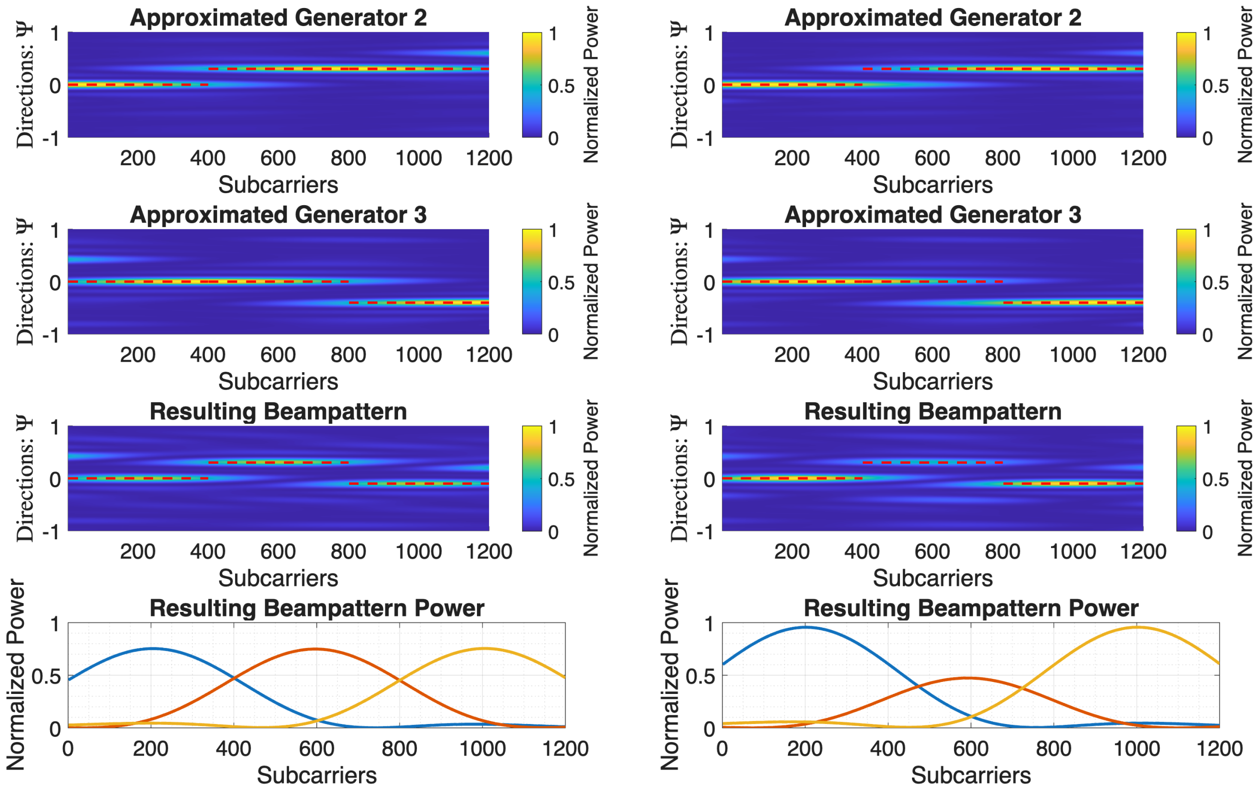


Figure 3.4: Effect of β_g on the resulting received power across different directions. The generator beampatterns are obtained from the JPTA-based dictionary with $\mathbf{\Delta}_2 \approx [0, 0.3, 0.3]^T$ and $\mathbf{\Delta}_3 \approx [0, 0, -0.4]^T$, and the resulting beampattern approximates $\mathbf{P}(\mathbf{\Delta}_2 + \mathbf{\Delta}_3)$ with mapping $\mathbf{\Delta}_2 + \mathbf{\Delta}_3 \approx [0, 0.3, -0.1]^T$. The left column shows HDB with fixed α_g, β_g values, whereas in the right column β_g is varied as $\beta_2 + 0.1$ for generator 2 and $\beta_3 - 0.1$ for generator 3. Dashed red lines indicate the target directions. The power distribution can be controlled by changing β_g.

by generator beampatterns as in (3.14), we can utilize the homomorphism of Theorem 3.1 and obtain the array configuration matrix approximating the desired split beampattern from (3.9). The HDB split beampattern synthesis method is summarized in Algorithm 2.

3.6 HDB Algorithm with Parameter Learning

The HDB algorithm [17] provides a lightweight framework for split beampattern synthesis but operates with a fixed $\mathbf{S}(\alpha_g, \beta_g)$ determined by the structure of the generator beampatterns and assumes an ideal and uniform power distribution. However, the resulting split beampatterns exhibit non-uniform power across subcarriers, and we observe that, for $g \geq 2$, varying α_g and β_g significantly affects the per-subcarrier power distribution, which is illustrated for β_g in Fig. 3.4. We therefore parametrize the generator array configuration matrices as

$$\begin{aligned}
\mathbf{\Phi}_{\mathbf{\Delta}_g}(\hat{\alpha}_g, \hat{\beta}_g) &= \tilde{\mathbf{\Phi}} \begin{bmatrix} \dfrac{1}{\hat{\alpha}_g} & -2\pi \dfrac{\hat{\alpha}_g f_c - f_c^{\Delta_g}(\hat{\beta}_g)}{\hat{\alpha}_g} \\ 0 & 1 \end{bmatrix},\ g \geq 2 \\
&= \tilde{\mathbf{\Phi}} \mathbf{S}(\hat{\alpha}_g, \hat{\beta}_g),\ g \geq 2 && \text{(3.18a)} \\
\hat{\mathbf{\Phi}}_{\boldsymbol{\psi}}(\{\hat{\alpha}_g, \hat{\beta}_g\}_{g=2}^{G}) &= \mathbf{\Phi}_{\mathbf{\Delta}_1} + \sum_{g=2}^{G} \mathbf{\Phi}_{\mathbf{\Delta}_g}(\hat{\alpha}_g, \hat{\beta}_g), && \text{(3.18b)}
\end{aligned}$$

where (3.18b) follows from line 3 of Algorithm 2. We then learn the locally optimal parameters $\hat{\alpha}_g, \hat{\beta}_g$ for $g = 2, \ldots, G$ by solving (3.19) over the training UE configurations. Specifically, for each value of G, we learn a single set of $\{\hat{\alpha}_g, \hat{\beta}_g\}_{g=2}^{G}$ that generalizes across subband-to-direction mappings:

$$\max_{\{\hat{\alpha}_g, \hat{\beta}_g\}_{g=2}^{G}} \sum_{u=1}^{T} \sum_{m} |[\mathbf{V}_{\hat{\mathbf{\Phi}}_{\boldsymbol{\psi}_u}(\{\hat{\alpha}_g, \hat{\beta}_g\}_{g=2}^{G})}]_{(:,m)}^{H} [\mathbf{V}(\boldsymbol{\psi}_u)]_{(:,m)}|^2, \tag{3.19}$$

where T is the number of training UE configurations and the test set contains the same number of configurations. The vector $\boldsymbol{\psi}_u$ denotes the subband-to-direction mapping of the u-th configuration. Both the training and test configurations $\boldsymbol{\psi}_u$ are generated by drawing their entries independently from the discrete uniform distribution over $\overline{\mathcal{A}}$, i.e., $[\boldsymbol{\psi}_u]_g \overset{\text{i.i.d.}}{\sim} \mathcal{U}(\overline{\mathcal{A}})$, $g = 1, \ldots, G$, $u = 1, \ldots, T$, where $\mathcal{A}$ is the uniform A-point grid over the direction range $[-1, 1]$ and $\overline{\mathcal{A}} = \mathcal{A} \cap [-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$ is its restriction to the considered sector. The

Table 3.1: Practical Deployment Requirements of HDB and Benchmark Algorithms

Algorithms	Complexity	Dict. Size	Dict. Mem.
HDB	$\mathcal{O}(NG)$	$\mathbb{R}^{N\times 2\times(2A-1)\dagger}$	≈ 250 KB
JPTA Approx. [9]	$\mathcal{O}(MNI_{\mathrm{it}})$	0	0
mmFlexible [14]	$\mathcal{O}(NG)$	0	0

†HDB ML additionally stores $\mathbb{R}^{(G-1)\times 2}$ learned parameters $\{\hat{\alpha}_g, \hat{\beta}_g\}_{g=2}^{G}$ with negligible memory cost.

I_{it}: number of iterations of the JPTA approximation algorithm [9].

Table 3.2: Training Hyperparameters

Parameter	**Value**
Optimizer	Adam
Training and test set size T	10^6
Learning rate	10^{-2}
Epochs	30
Batch size (gradient accumulation)	5000
Step scheduler step size	15
Step scheduler decay factor	10^{-1}

learnable parameters are initialized at the structural values $\hat{\alpha}_g = \alpha_g,\ \hat{\beta}_g = \beta_g$ from (3.16), $g = 2, \ldots, G$. The optimization is implemented in PyTorch, and we use the Adam optimizer [37] to determine the locally optimal $\hat{\alpha}_g, \hat{\beta}_g$ values over the training set. For each G, only $2(G-1)$ scalar parameters are optimized; this procedure is performed once, offline, with details in Table 3.2.

3.7 Performance Analysis

We compare the HDB algorithms against benchmark algorithms using Monte Carlo simulations with user directions restricted to the sector $[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$. The first 5000 UE configurations of the test set of size $T = 10^6$ defined in Section 3.6 are used, which are independent of the training set. System and simulation parameters are summarized in Table 3.3.

Table 3.3: Simulation Parameters

Parameter	**Value**
Center frequency f_c	28 GHz
Bandwidth BW	1 GHz
Number of subcarriers M	1200
Number of ULA antennas N	16
Number of UEs G	$\{3, 4, 5, 6\}$
Direction grid size A	499
UE direction sector	$[-\sin(\frac{\pi}{3}), \sin(\frac{\pi}{3})]$
Number of Monte Carlo trials	5000
JPTA [9] approximation maximum delay	$M/BW = 1.2\ \mu$s
JPTA [9] approximation iterations I_{it}	30

We compare the following algorithms:

- **HDB Original** [17]: HDB algorithm with fixed parameters as in Section 3.5.
- **HDB ML** (machine learning): HDB algorithm with learned parameters as in Section 3.6.
- **mmFlexible** [14]: Closed-form solution to the average received power maximization problem that ignores the frequency dependency of the array response.
- **JPTA Approximation** [9]: Iterative solution to the average received power maximization problem with $\mathcal{O}(MNI_{\text{it}})$ complexity, where I_{it} is the number of iterations of the JPTA approximation algorithm.

The HDB algorithm has the same $\mathcal{O}(NG)$ complexity as mmFlexible, with negligible dictionary memory, and lower complexity than the JPTA approximation algorithm since $MI_{\text{it}} \gg G$. Lookup-dictionary-based utilization of the JPTA approximation algorithm to avoid its computational complexity would be impractical, as a dictionary for all possible directions would require $\mathcal{O}(NA^G)$ elements. The generator dictionary is constructed offline only once, with the JPTA approximation algorithm applied to each dictionary entry at a computational complexity of $\mathcal{O}(MNI_{\text{it}})$. Table 3.1 summarizes the practical deployment requirements.

Fig. 3.5 shows that HDB ML matches mmFlexible across the considered numbers of UEs, while HDB Original (without parameter optimization) provides slightly lower average power. The JPTA approximation algorithm achieves the highest average power but at higher complexity than the other algorithms.

Fig. 3.6 shows that HDB Original provides the smallest max-to-min ratio of average powers across subbands, and HDB ML remains consistently below both the JPTA approximation algorithm and mmFlexible. The HDB algorithms thus provide a more uniform received power distribution across subbands than the benchmark algorithms. Overall, HDB ML matches mmFlexible in average power while delivering a fairer per-subband distribution.

Limitations: In this paper, delays and phases are modeled as unconstrained real numbers. Negative delays can be addressed by adding a common constant delay offset to all elements, which does not affect the received power [14]. Finite resolution and bounded delay range of practical TTD elements [14] are left for future work.

3.8 Conclusion

In this paper, we extend the structured split beampattern synthesis algorithm of [17], which is based on a generator dictionary and the homomorphism between TTD array configurations and beampatterns. To maximize the average received power across subcarriers, we optimize the shift and scale parameters of the algorithm using a data-driven approach. The proposed HDB ML algorithm achieves average received power comparable to that of the state-of-the-art low-complexity mmFlexible [14] algorithm, with the same computational complexity and negligible memory, while providing a more uniform power distribution across subbands. Future work includes extending HDB ML to hybrid arrays and near-field channels.

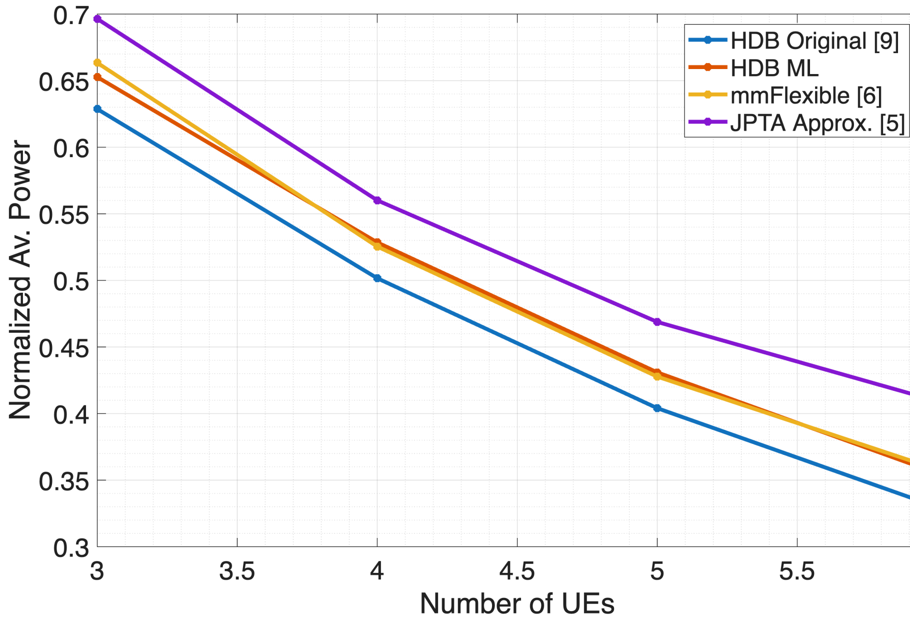


Figure 3.5: Normalized average received power versus number of UEs, averaged over 5000 Monte Carlo trials.

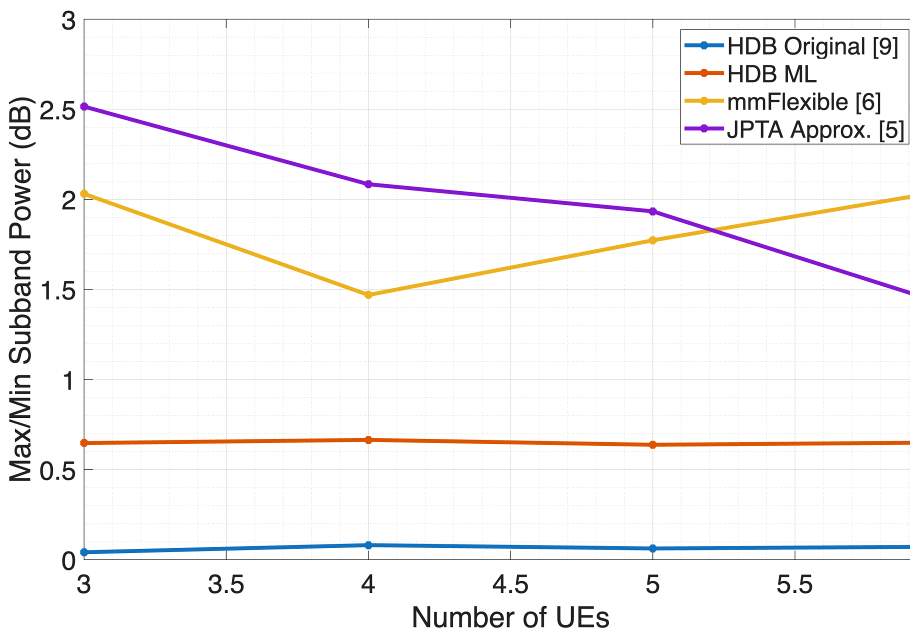


Figure 3.6: Ratio of maximum to minimum per-subband average received power across the G subbands (in dB), averaged over 5000 Monte Carlo trials, versus number of UEs.

CHAPTER 4

Near-Field Beam Training with Analog Extremely Large True-Time-Delay Arrays

4.1 Introduction

To address the diverse demands of next-generation mobile networks–spanning massive communications, ubiquitous connectivity, and integrated sensing and communication (ISAC)–mmWave frequencies and beyond are anticipated to be explored to leverage their abundant bandwidth. As these desired frequencies exhibit prohibitive path loss, extremely large aperture arrays (ELAAs) emerged as a compelling solution featuring massive number of antennas and high beamforming gain [38]. However, as the number of antennas increases, conventional assumptions on channel propagation characteristics may not hold [5].

In conventional multi-antenna systems, UE is assumed to be in far-field region with planar electromagnetic propagation characteristics. Therefore, the knowledge of the UE's direction is sufficient to properly form the high gain beam towards the UE. The boundary separating the far-field and near-field (NF) regions is determined by Rayleigh distance where any UEs beyond this distance are assumed to be in the far-field region. As the Rayleigh distance is proportional to the square of the array aperture [5], the NF region expands with the growing number of antennas. So, the electromagnetic waves have to be accurately modeled as spherical waves. For example, for a uniform linear array (ULA) with 255 antennas operating at 100 GHz, the Rayleigh distance is ~ 96 meters on the boresight. Due to spherical propagation characteristics, the estimation of UE's both direction and distance is required for efficient beam alignment [10], [39]. Consequently, beam training for NF UEs may involve substantial pilot and computational overhead.

Recently, several low overhead NF beam training algorithms are proposed. In [40], authors propose a polar codebook-based compressive channel estimation algorithm using multiple OFDM pilot symbols. In [41], a two-stage hierarchical beam training method is proposed where the coarse UE direction is first estimated with conventional far-field hierarchical beam training followed by distance and direction refinement over multiple OFDM pilot symbols. Whereas, [42] estimates the direction and distance of the UE with a neural network from the output of the far-field DFT codebook. Although these methods can reduce the beam training overhead, they employ arrays with only frequency-independent phase shifters and can only create a single beam per OFDM symbol, requiring multiple OFDM pilot symbols for beam training.

TTD arrays enable low-cost frequency-dependent beamforming capabilities and can simultaneously probe multiple directions with a single OFDM symbol. In [8], a far-field TTD based rainbow beamforming was proposed to perform beam training using a single OFDM pilot. This approach has been adapted to NF channels to probe multiple directions with different frequency measurements over a given path [43] or a region [44], thereby reducing the required number of pilots. Recently, sub-array based TTD architecture has been proposed [45] such that only a partition of the entire array is active per one OFDM pilot measurement. Thus, a UE that is in NF for entire array falls into the far-field region of each sub-array. Hence, beam training problem reduces to utilizing far-field rainbow beams [8] over different sub-arrays with multiple OFDM pilot symbols.

In this work, we propose a novel NF beam training algorithm using a TTD array that requires only a single OFDM pilot symbol. In the proposed method, we virtually partition the TTD array into smaller sub-arrays that effectively operate in far-field [45] while utilizing all sub-arrays simultaneously. We then show that by utilizing simple and uniform time delay and phase values as in Rainbow beamforming [8], the received signals from different sub-

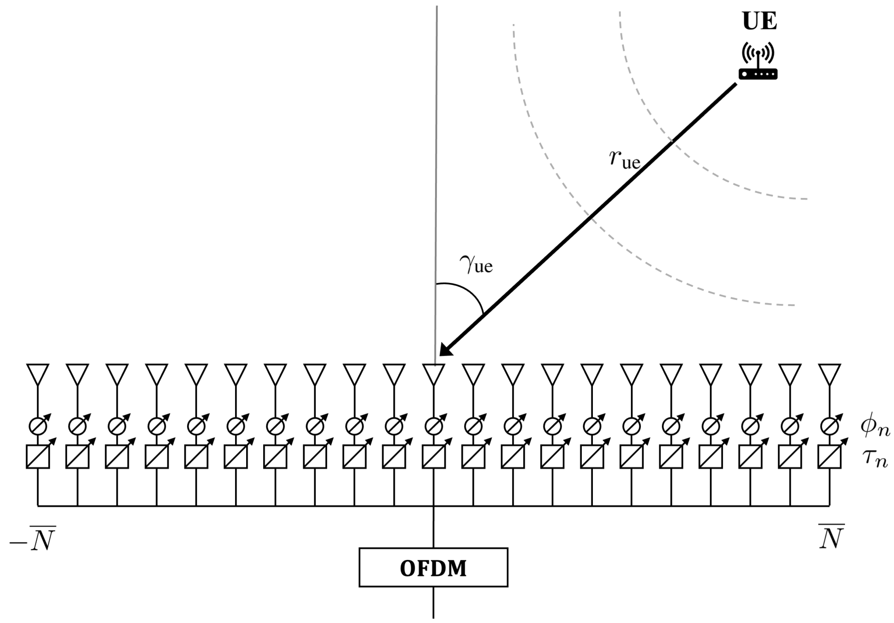


Figure 4.1: Illustration of a $2\overline{N}+1$ antenna-equipped base station (BS) with a TTD array and a single antenna UE.

arrays become near-orthogonal. As a result, UE direction referenced to each sub-array can be recovered with only a single OFDM pilot symbol. Then, by utilizing the known array structure and per sub-array UE directions, the UE's location can be angulated to enable effective beam alignment [45].

4.2 ELAA System Model

We consider an uplink OFDM system with a BS comprising ELAA and a single antenna UE. The system utilizes OFDM waveform operating at center frequency f_c with the total bandwidth W and $M = 2\overline{M}+1$ subcarriers. The BS centered at the origin is equipped with analog uniform linear TTD array with $N = 2\overline{N}+1$ antennas and single RF chain where each antenna $n \in [-\overline{N}, \overline{N}]$ is connected to a phase shifter, ϕ_n, and time delay element τ_n as illustrated in Fig. 4.1.

The UE's location is defined as $(r_{\text{ue}}, \gamma_{\text{ue}})$ where r_{ue} is the radial distance between the UE and the center of the antenna array and $\gamma_{\text{ue}} = \sin(\alpha_{\text{ue}})$ is the corresponding direction for the UE angle α_{ue}. For LOS scenario and ignoring large-scale fading, the NF channel corresponding to m-th subcarrier $\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}} \in \mathbb{C}^{N\times 1}$ is defined as [43]

$$[\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}]_n = e^{-jk_m r_{\text{ue}}} e^{-jk_m(-nd\gamma_{\text{ue}} + n^2 d^2 \frac{1-\gamma_{\text{ue}}^2}{2r_{\text{ue}}})} \tag{4.1}$$

where $k_m = \frac{2\pi f_m}{c}$ is the wave number, f_m denotes the frequency of the m-th subcarrier, $f_m = f_c + W\frac{m}{M}$, and $d = \lambda_c/2$ where λ_c is the wavelength at f_c. If the UE is farther than the Rayleigh distance i.e. $r_{\text{ue}} > \frac{(N-1)^2\lambda_c}{2}$, UE's channel can be approximated as a far-field channel [5] $\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}} \approx \boldsymbol{h}_{m,\gamma_{\text{ue}}} \in \mathbb{C}^{N\times 1}$ where

$$[\boldsymbol{h}_{m,\gamma_{\text{ue}}}]_n = e^{-jk_m r_{\text{ue}}} e^{jk_m nd\gamma_{\text{ue}}} \tag{4.2}$$

where $e^{-jk_m r_{\text{ue}}}$ term is generally ignored [43]. The TTD beamforming vector for the m-th subcarrier is formulated as [45]

$$[\boldsymbol{v}_m]_n = \frac{1}{\sqrt{N}} e^{-j(2\pi f_m \tau_n - \phi_n)} \tag{4.3}$$

Unlike analog arrays using only frequency-independent phase shifters, the TTD array delay units enable frequency-dependent beamforming. Using conventional OFDM processing, the received signal after the combining $\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}} \in \mathbb{C}^{M\times 1}$ is given by

$$\begin{aligned} [\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}]_m &= \boldsymbol{v}_m^H(\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}} + \boldsymbol{n}_m) \\ &= \sum_{n=-\overline{N}}^{\overline{N}} [\boldsymbol{v}_m]_n^* ([\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}]_n s_m + [\boldsymbol{n}_m]_n) \end{aligned} \tag{4.4}$$

where $\boldsymbol{n}_m \sim \mathcal{CN}(0, \sigma^2 \boldsymbol{I}_N) \in \mathbb{C}^{N\times 1}$ is a complex zero-mean Gaussian noise with variance σ^2 mutually independent for each m and s_m is the m-th subcarrier pilot symbol.

The objective of the beam training algorithm is to design a TTD beamforming vector $\boldsymbol{v}_m$ that is aligned with the channel vector $\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}$ to maximize the received power $|\boldsymbol{v}_m^H \boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}|^2 |s_m|^2$ for all m. If the UE's location $(r_{\text{ue}}, \gamma_{\text{ue}})$ is known at the BS, alignment and maximum power can be achieved by configuring time delays as $\tau_n = \frac{1}{c}(-nd\gamma_{\text{ue}} + n^2 d^2 \frac{1-\gamma_{\text{ue}}^2}{2r_{\text{ue}}})$ [43] which results in:

$$\begin{aligned} |\boldsymbol{v}_m^H \boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}|^2 |s_m|^2 &= \frac{1}{N} |\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}^H \boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}|^2 |s_m|^2 \\ &= N|s_m|^2 \end{aligned} \tag{4.5}$$

In order to maximize the beamforming gain towards the UE, beam training algorithm needs to estimate the UE's location $(r_{\text{ue}}, \gamma_{\text{ue}})$ from the received signal $\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}$.

4.3 Beam training with TTD Arrays

Beam training in NF channels can be achieved by aligning TTD beamforming vector towards different locations over different OFDM symbols, and probing all possible UE locations for maximum alignment. Similarly, all possible directions can be probed over different OFDM symbols for far-field channels. However, these approaches introduce considerable training overhead and do not exploit the frequency-dependent beamforming capabilities of the TTD arrays.

4.3.1 Far-field TTD Beam training

For far-field channels, it is shown that TTD arrays can probe multiple directions over different subcarriers with a single OFDM symbol, reducing the beam training overhead significantly [8]. After configuring TTD beamforming vector, as given in (4.3), with the following parameters

$$\tau_n = \frac{n}{W},\ \phi_n = \frac{2\pi f_c n}{W}. \tag{4.6}$$

the received signal $\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}$ can be obtained as [8]

$$[\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}]_m = \frac{e^{-jk_m r_{\text{ue}}}}{\sqrt{N}} \frac{\sin(\frac{\pi}{2} N z_{\gamma_{\text{ue}},m})}{\sin(\frac{\pi}{2} z_{\gamma_{\text{ue}},m})} \tag{4.7}$$

where $z_{\gamma_{\text{ue}},m} = \gamma_{\text{ue}} + m(\gamma_{\text{ue}} \frac{W}{Mf_c} + \frac{2}{M})$. It can be shown that the maximum value of $\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}$ is attained when $\gamma_{\text{ue}} = -2mf_c/(Mf_m) + 2z,\ z \in \mathbb{Z}$. Thus, the direction of the UE can be estimated as

$$\begin{aligned} \hat{m} &= \arg\max |\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}| \\ \hat{\gamma}_{\text{ue}} &= \mod\left(1 - 2\frac{\hat{m} f_c}{M f_m}, 2\right) - 1 \end{aligned} \tag{4.8}$$

with only a single OFDM symbol.

However, this approach cannot be directly applied to NF channels as TTD array configuration in (4.6) does not result in the received signal in (4.7), making straightforward subcarrier to direction mapping in (4.8) inapplicable.

4.3.2 Near-field sub-array TTD Beam training

Recently, sub-array based beam training method is proposed to utilize far-field rainbow TTD configuration for the NF channels [45]. As Rayleigh distance is a function of number of antennas and array aperture, the entire array can be partitioned into $S = 2\overline{S} + 1$ sub-arrays with $N_S = \frac{N}{S}$ antennas as shown in Fig. 4.2 such that UE falls into far-field region of every sub-array. Then by utilizing only one sub-array per OFDM symbol, each sub-array's direction γ_k can be estimated by the far-field TTD configuration described in section 4.3.1, over multiple OFDM symbols.

Upon obtaining direction γ_k of each sub-array and utilizing known sub-array center locations in Cartesian coordinates:

$$\boldsymbol{p} = \begin{bmatrix} -\overline{S}N_S & 0 \\ \vdots & \\ \overline{S}N_S & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_x & \boldsymbol{p}_{\boldsymbol{y}} \end{bmatrix} \in \mathbb{C}^{S \times 2} \tag{4.9}$$

UE's location in Cartesian coordinates $(x_{\text{ue}}, y_{\text{ue}})$ can be related to sub-array center locations as:

$$\boldsymbol{p}_x = \underbrace{\begin{bmatrix} 1 & -\tan\arcsin\gamma_{-\overline{S}} \\ & \vdots \\ 1 & -\tan\arcsin\gamma_{\overline{S}} \end{bmatrix}}_{\boldsymbol{R}} \begin{bmatrix} x_{\text{ue}} \\ y_{\text{ue}} \end{bmatrix} \tag{4.10}$$

and the UE's location can be estimated by following equations [45] where this procedure is termed as angulation:

$$\begin{bmatrix} x_{\text{ue}} \\ y_{\text{ue}} \end{bmatrix} = \boldsymbol{R}^{\dagger}\boldsymbol{p}_x,\ \gamma_{\text{ue}} = \sin\arctan\frac{x_{\text{ue}}}{y_{\text{ue}}},\ r_{\text{ue}} = \sqrt{x_{\text{ue}}^2 + y_{\text{ue}}^2} \tag{4.11}$$

where $\boldsymbol{R}^{\dagger}$ is the pseudo-inverse of $\boldsymbol{R}$ and $\boldsymbol{R}^{\dagger} = (\boldsymbol{R}^H\boldsymbol{R})^{-1}\boldsymbol{R}^H$ if $\text{rank}(\boldsymbol{R}) = 2$.

Using this sequential sub-array activation and collecting measurements using rainbow beams from each sub-array, the beam training can be achieved with S OFDM symbols. In the next section, we show that there is no need to separate measurements from each sub-array and utilize multiple OFDM symbols. In fact, by simply utilizing rainbow beam TTD configuration given in (4.6) direction of each sub-array can be recovered with only a single OFDM symbol.

4.4 Virtual Sub-array Sparse Rainbow (VSSR) Algorithm

In this section, we show that the direction of each sub-array can be recovered with a single OFDM symbol by formulating a sparse recovery problem. As utilizing entire array is equivalent to using all sub-arrays concurrently, we first decompose the received signal from the entire array as a sum of received signals of each sub-array. By exploiting mutual near-orthogonality of sub-array signals for $S = 3$, we recover the direction of each sub-array by Orthogonal Matching Pursuit (OMP) algorithm [46]. Then, angulation method in (4.11) is

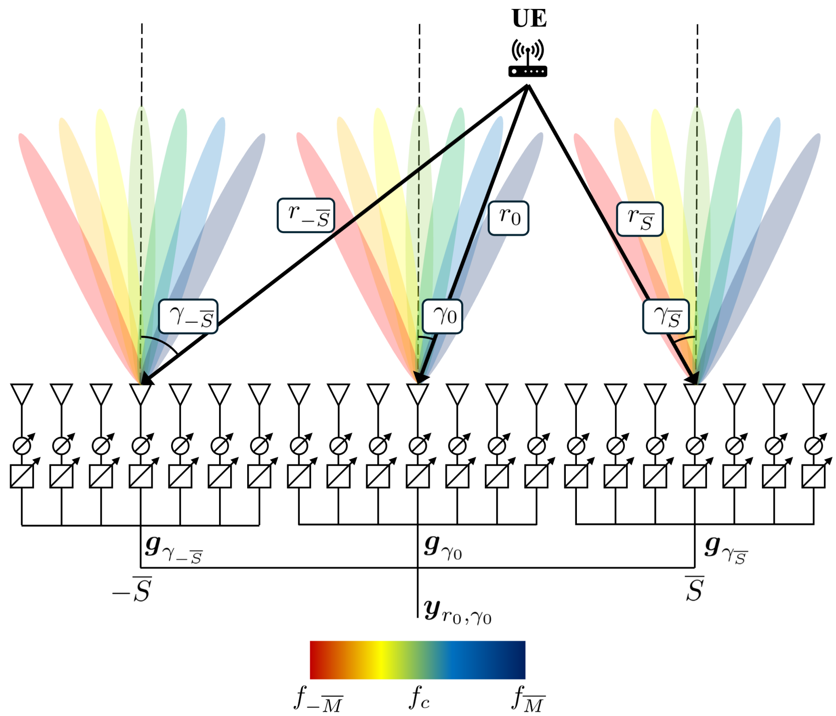
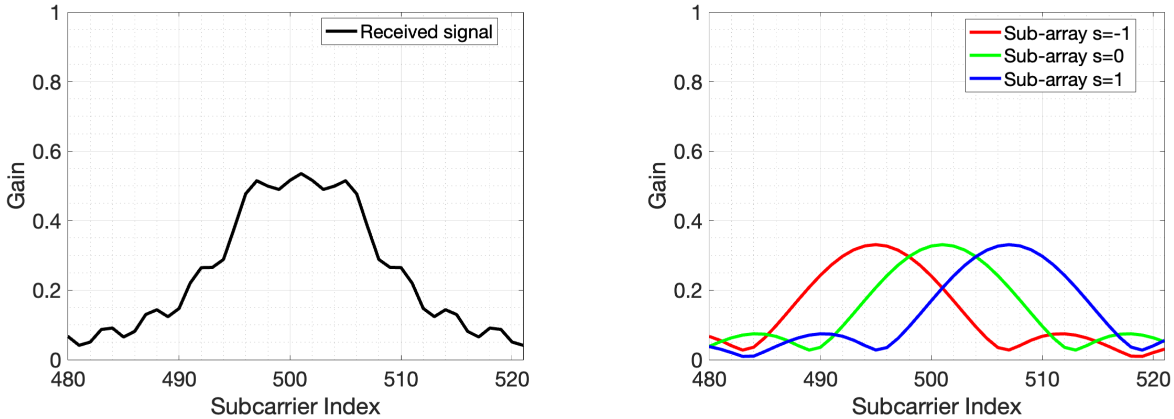


Figure 4.2: Virtual partitioning of the TTD array for $S = 3$ and the VSSR algorithm rainbow beams.

utilized to estimate the UE's location.

4.4.1 Virtual Sub-array Signals

By exploiting the linear operations of signal combining, and inspired by the approach in Section 4.3.2, we can mathematically partition the array into $S = 2\overline{S}+1$ uniform sub-arrays with $N_S = 2\overline{N}_S + 1$ antennas. We refer to this operation as virtual partitioning. Then, the received signal for entire array can be written as:



(a) Received signal gain $|\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}|$ (b) Intermediate signal gain $|\boldsymbol{g}_{\gamma_s}|$, $\forall s$

Figure 4.3: Gain of the received signal, in (a), and intermediate signals, in (b), for parameters $N = 255$, $S = 3$, $M = A = 1001$ and the UE is located at 10.57 meters on boresight of the ELAA.

$$\begin{aligned}
[\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}]_m &= \sum_{n=-\overline{N}}^{\overline{N}} [\boldsymbol{v}_m]_n^* [\boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}]_n \\
&= \sum_{s=-\overline{S}}^{\overline{S}} \underbrace{\sum_{n'=-\overline{N}_S}^{\overline{N}_S} [\boldsymbol{v}_m]_{n'-sN_S}^* [\boldsymbol{h}_{m,r_s,\gamma_s}]_{n'}}_{\text{Signal of sub-array } s}
\end{aligned} \tag{4.12}$$

where $\boldsymbol{h}_{m,r_s,\gamma_s}$ is the NF channel between the UE and sub-array $s \in [-\overline{S}, \overline{S}]$, (r_s, γ_s) is the distance and direction of the UE with respect to sub-array center as illustrated in Fig. 4.2.

Assuming $S = 3$ and the UEs in the far-field region for all virtual-sub-arrays, i.e., $\min_s(r_s) \geq \frac{(N_S-1)\lambda_c}{2}$, the received signal can further be simplified as

$$[\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}]_m = \sum_{s=-\overline{S}}^{\overline{S}} \sum_{n'=-\overline{N}_S}^{\overline{N}_S} [\boldsymbol{v}_m^*]_{n'-sN_S} \underbrace{e^{-j(k_m r_s - k_m n d \gamma_s)}}_{[\boldsymbol{h}_{m,\gamma_s}]_{n'}} \tag{4.13}$$

where $\boldsymbol{h}_{m,\gamma_s}$ is the far-field channel between the UE and the s-th sub-array. From known array geometry and trigonometric identities, we can see that $r_s = \sqrt{r_{s-1}^2 + N_S^2 d^2 - 2\gamma_s r_{s-1} N_S d}$.

From far field assumption, $d(N_S - 1) \approx dN_S$ is small compared to r_s, $\forall s$ [43]; thus $r_s \approx r_{s-1} + dN_S\gamma_s$. Then, the received signal for $S = 3$ is given as:

$$[\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}}]_m = e^{-jk_m r_{\mathrm{ue}}} \sum_{s=-\overline{S}}^{\overline{S}} \underbrace{\sum_{n'=-\overline{N}_S}^{\overline{N}_S} [\boldsymbol{v}_m]^*_{n'-sN_S} e^{jk_m(n'-sN_S)d\gamma_s}}_{[\boldsymbol{g}^s_{\gamma_s}]_m} \tag{4.14}$$

where $\boldsymbol{g}^s_{\gamma_s} \in \mathbb{C}^{M\times 1}$ is the intermediate signal, i.e., the received signal of the sub-array s, depending only on γ_s.

Utilizing TTD array configuration in (4.6), intermediate signal $\boldsymbol{g}^s_{\gamma_s}$ can be calculated as follows:

$$[\boldsymbol{g}^s_{\gamma_s}]_m = \frac{1}{\sqrt{N_S}} \sum_{n'=-\overline{N}_S}^{\overline{N}_S} e^{j\pi(n'-sN_S)(\gamma_s + \frac{2m}{M} + \frac{mW}{Mf_c}\gamma_s)} \tag{4.15a}$$

$$= \frac{1}{\sqrt{N_S}} \frac{\sin(\frac{\pi}{2}N_S z_{s,m})}{\sin(\frac{\pi}{2}z_{s,m})} e^{-j\pi s N_S z_{s,m}} \tag{4.15b}$$

where

$$z_{s,m} = \gamma_s + m(\gamma_s \frac{W}{Mf_c} + \frac{2}{M})$$

Then, the received signal $\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}}$ can be written as

$$\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}} = e^{-jk_m r_{\mathrm{ue}}} \sum_{s=-\overline{S}}^{\overline{S}} \boldsymbol{g}^s_{\gamma_s} \tag{4.16}$$

Observe that the intermediate signal given in (4.15b) is similar to the received signal given in (4.7) and also has the same structure as the received signal of the sub-array s of the method described in Section 4.3.2. The only difference is the sub-array-dependent phase shift introduced by the distance between sub-array centers. Therefore, if we can recover the intermediate signal of each sub-array from the received signal $\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}}$, we can estimate the direction γ_s of each sub-array. An example of received signal and intermediate signals for parameters $N = 255$, $S = 3$, $M = A = 1001$ and a UE at 10.57 meters in the boresight is given in Fig. 4.3. Our goal is to recover intermediate signals illustrated in Fig. 4.3b from the received signal shown in Fig.4.3a.

Next we formulate a sparse recovery problem to recover intermediate signals and corresponding directions. To support the approach of compressive estimation, we show that the measurements from different sub-arrays are nearly-orthogonal.

4.4.2 Sparse Recovery Problem Formulation

The intermediate signal for sub-array s, $\boldsymbol{g}^s_{\gamma_i}$, is uniquely parametrized by sub-array's direction γ_s. To estimate this direction, we define a dictionary of possible directions as follows:

$$\boldsymbol{G}_s = [\boldsymbol{g}^s_{\overline{\gamma}_1}, \boldsymbol{g}^s_{\overline{\gamma}_2}, \ldots, \boldsymbol{g}^s_{\overline{\gamma}_A}] \in \mathbb{C}^{M \times A} \tag{4.17}$$

where for fair comparison with S-NFBT algorithm UE directions are defined as $\overline{\gamma}_a = 2\frac{af_c}{Af_m}$, $a \in [-\overline{A}, \overline{A}]$ and $A = 2\overline{A} + 1 = M$ is the number of grid directions. We estimate each sub-array's direction based on this grid. The received sub-array signal $\boldsymbol{g}^s_{\gamma_i}$ for directions on this grid can be written as:

$$\boldsymbol{g}^s_{\gamma_i} = \boldsymbol{G}_s \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \boldsymbol{G}_s \boldsymbol{x}_s \in \mathbb{C}^{M \times 1} \tag{4.18}$$

where $\boldsymbol{x}_s \in \mathbb{C}^{A \times 1}$ is the sparse selection vector with $\|\boldsymbol{x}_s\|_0 = 1$, $\forall s$. Therefore, we can directly obtain sub-array's direction γ_s, if we know (or have estimate of) the sparse selection vector $\boldsymbol{x}_s$ as follows

$$\begin{aligned} \hat{a} &= \arg\max |\boldsymbol{x}_s| \\ \hat{\gamma}_s &= \mathrm{mod}\,(1 + \overline{\gamma}_{\hat{a}}, 2) - 1 \end{aligned} \tag{4.19}$$

Utilizing dictionary representation of intermediate signals, the received signal $\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}}$ can be written as:

$$[\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}}]_m = e^{-jk_m r_{\mathrm{ue}}} \sum_{s=-\overline{S}}^{\overline{S}} \boldsymbol{G}_s \boldsymbol{x}_s, \quad \|\boldsymbol{x}_s\|_0 = 1,\ \forall s \tag{4.20}$$

Hence, we can solve the following Least Squares (LS) problem to recover sparse selection vector $\boldsymbol{x}_s$ of each sub-array [46].

$$\begin{aligned} &\min_{\boldsymbol{x}_s,\,\forall s} \|\boldsymbol{y}_{r_{\mathrm{ue}},\gamma_{\mathrm{ue}}} - \sum_{s=-\overline{S}}^{\overline{S}} \boldsymbol{G}_s \boldsymbol{x}_s\|_2 \\ &s.t. \|\boldsymbol{x}_{\boldsymbol{s}}\|_0 = 1,\ \forall s \end{aligned} \tag{4.21}$$

However, obtaining the solution of (4.21) depends on the properties of the dictionaries $\boldsymbol{G}_s$, requiring further analysis.

4.4.2.1 Near-orthogonality of Sub-array Signals and Cross Coherence

We continue by showing that the intermediate signals $\boldsymbol{g}^s_{\gamma_s}$ for different sub-arrays with $S = 3$ are nearly orthogonal if the systems bandwidth W is negligible compared to f_c, making dictionaries mutually near-orthogonal.

Lemma 4.1. *If a system's bandwidth is negligible compared to its central frequency f_c, i.e., $W \ll f_c$ and the number of sub-arrays is $S = 3$, the intermediate signals of different sub-arrays $k \in \{-1, 0, 1\}$ and $s \neq k \in \{-1, 0, 1\}$ are orthogonal for any direction $\gamma_i, \gamma_j \in [-1, 1]$: $\boldsymbol{g}^k_{\gamma_i} \perp \boldsymbol{g}^s_{\gamma_j},\ \forall k, s \neq k \quad \&\ \forall \gamma_i, \gamma_j \in [-1, 1]$*

Proof. Let us assume that fractional bandwidth of the system is very small $W << f_c$ and number of sub-arrays is $S = 3$. Then, we can approximate $z_{s,m}$ as $z_{s,m} \approx \frac{2}{M}(m + \frac{\gamma_i M}{2})$. We want to prove that the inner product between signals from different sub-arrays are orthogonal for any direction.

The discrete time Fourier transform (DTFT) of the intermediate signal $[\boldsymbol{g}^s_{\gamma_s}]_m$ can be

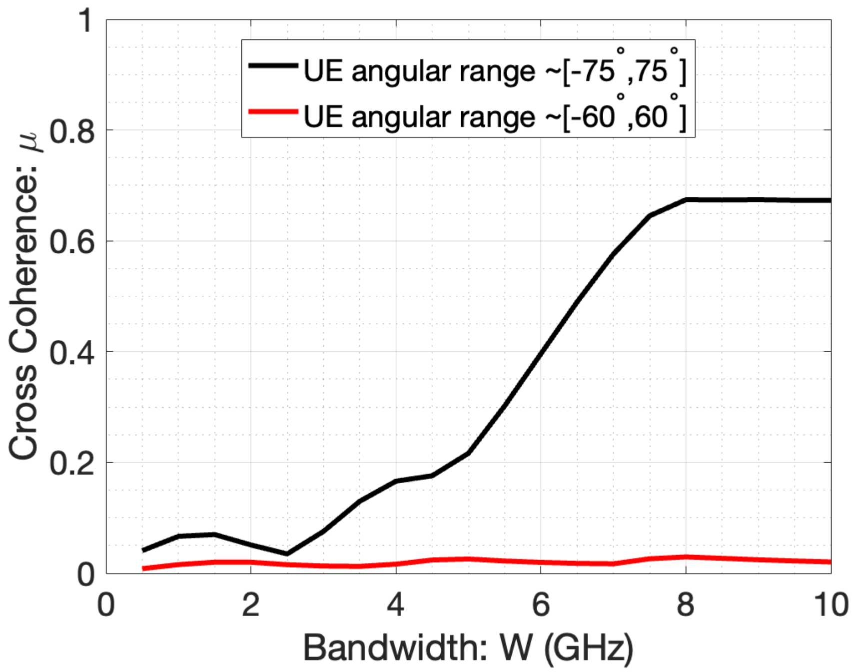


Figure 4.4: The cross coherence values from (4.25) for two different angles of interest regions, $[-75°, 75°]$ and $[-60°, 60°]$, respectively when $W \in [1, 10]$ GHz.

calculated as:

$$\mathcal{F}\{[\boldsymbol{g}^s_{\gamma_s}]_m\} = \Gamma_s(\Omega) = \Pi_{N_S}\left(\Omega + \frac{2\pi s N_S}{M}\right) e^{+j\Omega \frac{\gamma_s M}{2}} \tag{4.22}$$

along with

$$\Pi_{N_S}(\Omega) := \frac{2\pi}{\sqrt{N_S}} \sum_{m'=-\overline{N}_S}^{\overline{N}_S} \delta(\Omega - 2\pi \frac{m'}{M}) \tag{4.23}$$

Then, the inner product of intermediate signal of sub-array k and $s \neq k$ for respective directions γ_i, γ_j can be calculated as:

$$< \boldsymbol{g}^k_{\gamma_i}, \boldsymbol{g}^s_{\gamma_j} >= \sum_{m=-\overline{M}}^{\overline{M}} [\boldsymbol{g}^k_{\gamma_i}]^*_m [\boldsymbol{g}^s_{\gamma_j}]_m \tag{4.24a}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \underbrace{\Pi_{N_S}(\Omega + 2\pi \frac{kN_S}{M})^* \Pi_{N_S}(\Omega + 2\pi \frac{sN_S}{M})}_{0} e^{j\Omega \frac{(\gamma_j - \gamma_i)M}{2}} d\Omega \tag{4.24b}$$

$$= 0 \implies \boldsymbol{g}^k_{\gamma_i} \perp \boldsymbol{g}^s_{\gamma_j} \tag{4.24c}$$

where (4.24b) is the result of the generalized Parseval's equality [47] and (4.24c) is obtained from the properties of the $\Pi_{N_S}(\Omega)$ function defined in (4.23). $\square$

However, if the system has higher bandwidth W, intermediate signals are not orthogonal and inner product between intermediate signals of different sub-arrays become direction dependent. In order to quantify the coherence between intermediate signals of different sub-arrays, cross coherence metric is used [48]. This metric calculates the maximum possible coherence μ between sub-arrays out of all possible direction values in desired region, formulated as

$$\mu = \max_{\gamma_i, \gamma_j, k \neq s} |(\boldsymbol{g}^s_{\gamma_i})^H \boldsymbol{g}^k_{\gamma_j}| \tag{4.25}$$

Fig. 4.4 shows the cross coherence with respect to bandwidth for a system with $f_c = 100$ GHz, $N = 255$, $M = 1001$ and number of sub-arrays $S = 3$ for uniformly sampled directions. It can be seen that near-orthogonality is preserved across different bandwidths if UE's angular range is limited to $[-60°, 60°]$, meanwhile the orthogonality disappears for $[-75°, 75°]$ as W increases. This constraint to use directions between $[-60°, 60°]$ is common in communications systems [49] and intermediate signals can be realistically assumed to be nearly-orthogonal.

Algorithm 3: VSSR algorithm

Input: Received signal, $\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}} \in \mathbb{C}^{M\times 1}$ obtained by TTD array with configuration (4.6),maximum UE location, $r_{\max}$

Output: UE's location, $(r_{\text{ue}}, \gamma_{\text{ue}})$

1 Set $\Lambda = \emptyset,\ \boldsymbol{y} = \frac{\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}}{\|\boldsymbol{y}_{r_{\text{ue}},\gamma_{\text{ue}}}\|}$

2 **for** $s = -1 : 1$ **do**

3   Create dictionary $\boldsymbol{G}_s$ as defined in (4.17) and normalize columns

4   Obtain sparse selection vector $\hat{\boldsymbol{x}}_s = \boldsymbol{G}_s^H \boldsymbol{y}$

5   Obtain direction $\hat{\gamma}_s$ from $\hat{\boldsymbol{x}}_s$ with (4.19)

6     $\Lambda \leftarrow \Lambda \cup \{\boldsymbol{g}^s_{\hat{\gamma}_s}\}$

7     $\boldsymbol{y} \leftarrow \boldsymbol{P}_{\Lambda^\perp} \boldsymbol{y}$

8     $\boldsymbol{y} \leftarrow \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}$

9 **end**

10 Obtain UE's location from (4.11)

11 **if** $r_{ue} < 1e - 16$ **then**

12     $(r_{\text{ue}}, \gamma_{\text{ue}}) \leftarrow (r_{\max}, \hat{\gamma}_0)$

13 **end**

4.4.3 UE Location Estimation and Proposed Algorithm

As intermediate signals are nearly orthogonal for the region of UE directions of interest, we can utilize simplified orthogonal matching pursuit (OMP) algorithm [46] by exploiting the observed structure and solve the problem (4.21). Then, direction of each sub-array γ_s can be obtained from the estimated sparse selection vectors with the equation (4.19). Upon determining sub-array directions, UE's location can be estimated by angulation as described in (4.11). The proposed method is summarized in Algorithm 3 where steps [11-13] are added to address the angulation failure resulting from off-grid directions and noise. In such cases, we assume that the UE is in far-field, and direct a beam based on detected direction of center sub-array and maximum allowed UE distance, $r_{\max}$.

4.5 Numerical Results

In this section, we demonstrate the performance of the proposed VSSR algorithm with extensive simulations where used parameters are listed in the Table 4.1. We calculate the normalized array gain $G(r_{\text{ue}}, \gamma_{\text{ue}})$ as a performance metric defined as follows:

$$\begin{aligned} G(r_{\text{ue}}, \gamma_{\text{ue}}) =& \frac{1}{M\sqrt{N}} \sum_{m=-\overline{M}}^{\overline{M}} |\boldsymbol{v}_m \boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}| \\ =& \frac{1}{M\sqrt{N}} \sum_{m=-\overline{M}}^{\overline{M}} |\boldsymbol{h}_{m,\hat{r}_{\text{ue}},\hat{\gamma}_{\text{ue}}} \boldsymbol{h}_{m,r_{\text{ue}},\gamma_{\text{ue}}}| \end{aligned} \tag{4.26}$$

where $(r_{\text{ue}}, \gamma_{\text{ue}})$ is the actual and $(\hat{r}_{\text{ue}}, \hat{\gamma}_{\text{ue}})$ is the estimated UE location and $\boldsymbol{v}_m$ is configured as in (4.5) and pre-beamforming noise variance is $\sigma^2 = \frac{1}{\text{SNR}}$.

The following benchmark beam training schemes are utilized in the subsequent performance comparison:

S-NFBT algorithm [45]

This is the sequential sub-array rainbow algorithm described in Section 4.3.2. The required number of OFDM symbols for this algorithm is $S = 3$.

Dictionary-based S-NFBT (DS-NFBT) algorithm

To provide a fair comparison, we extend the S-NFBT algorithm [45] with a dictionary based approach and estimate each sub-array's direction with (4.18) and (4.19) rather than (4.8). Note that the required number of the OFDM symbols is the same as in S-NFBT benchmark.

Fig. 4.5 shows the empirical CDF of the normalized array gains with respect to different UE locations for the noise-free scenario. Results indicate that the VSSR algorithm performs

Table 4.1: ELAA Simulation Parameters

Parameter	**Value**
Center frequency f_c	100 GHz
Bandwidth W	3 GHz
Number of subcarriers M	1001
Number of ULA antenna N	255
Number of sub-arrays S	3
Rayleigh distance (Boresight)	96.7 meters
Rayleigh distance of each sub-array (Boresight)	10.57 meters
UE direction grid numbers (A)	1001
UE Distances	4 : 0.5 : 30 meters
UE directions	$-60° : 0.5° : 60°$
SNR Values	$-10 : 10 : 30$ dB

identical to the DS-NFBT algorithm and to benchmark S-NFBT algorithm, verifying that the intermediate signal of each sub-array can be recovered with only a single OFDM symbol.

Similarly, Fig. 4.6 shows the empirical CDF of the normalized array gains of different UE locations when pre-beamforming SNR is -10 dB. Even under this channel condition, VSSR and DS-NFBT perform near-optimal, providing at least 93% of the optimal array gain for 85% and 98% of the locations, respectively. The presented results solidify that VSSR can achieve reliable beam training with a single OFDM symbol even under low SNR range. Both dictionary based algorithms outperform S-NFBT as the projection over dictionary vectors improves SNR.

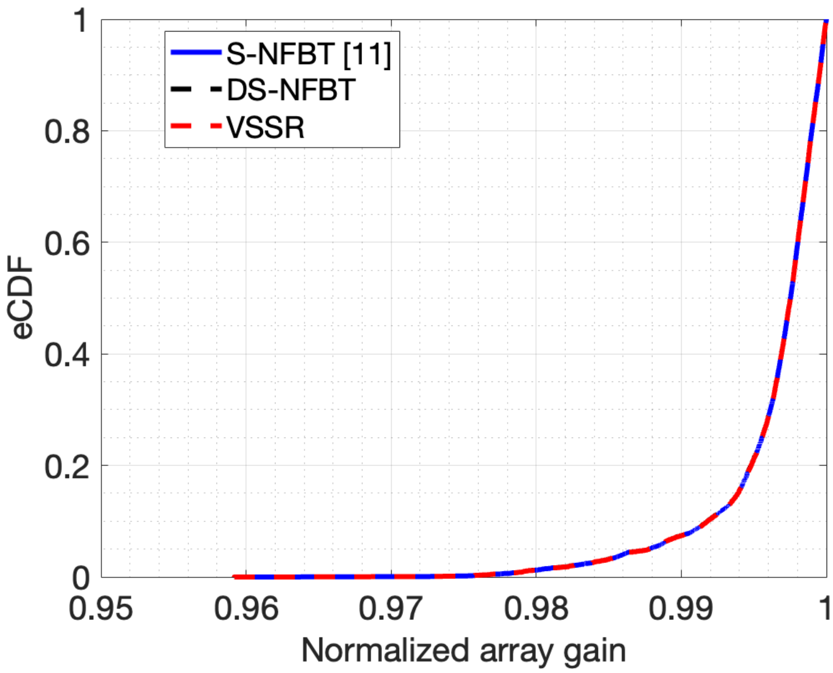


Figure 4.5: The empirical CDF comparison of S-NFBT [45], dictionary-based S-NFBT (DS-NFBT) and VSSR algorithms in terms of the normalized array gain when the noise-free cases are considered.

Table 4.2 demonstrates the angulation failure percentage for different SNR values and different tolerances. Due to noise and finite direction grid resolution, all algorithms can detect same or unrealistic directions for different sub-arrays, making algorithms fail to angulate and return UE location as $(0, 0)$. We treat these cases as if UE is in far-field and if $\hat{r}_{\text{ue}} < tolerance$ we assign UE the location $(r_{\max}, \hat{\gamma}_{\text{ue}})$ where $\hat{\gamma}_0$ is the estimated direction of the center sub-array and $r_{\max}$ is the maximum possible UE distance. The presented results demonstrate that both VSSR and DS-NFBT is resilient to such errors due to additional gain from projection to dictionary vectors. Failure percentage can further be reduced by increasing the number of directiongrid points M for S-NFBT and A for VSSR and DS-NFBT, respectively.

Fig. 4.7 shows the average normalized array gain over different set of UE locations for different SNR ranges. Across all SNR conditions, VSSR and DS-NFBT achieve near-optimal

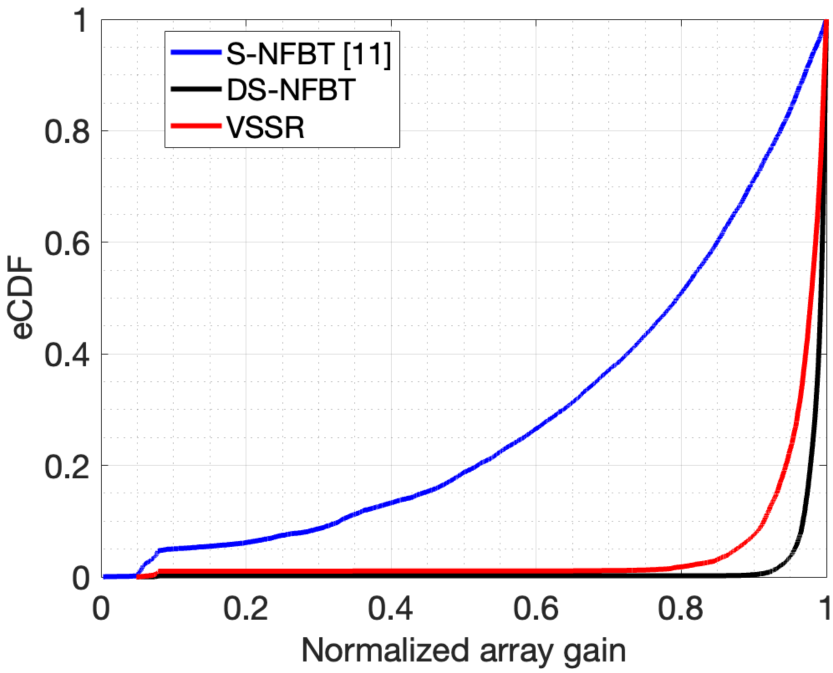


Figure 4.6: The empirical CDF comparison of the normalized array gain between S-NFBT [45], dictionary-based S-NFBT (DS-NFBT) and VSSR algorithms when pre-beamforming SNR is set to -10 dB.

beam training performance, providing an average of more than 95% of the optimal array gain. Furthermore, VSSR improves the averagenormalized array gain of the S-NFBT by $\sim 33\%$ at SNR$= -10$ dB while requiring only a single OFDM symbol.

4.6 Conclusion

In this work, we proposed a single OFDM pilot symbol NF beam training algorithm utilizing rainbow beams with analog TTD array. We showed that by utilizing far-field rainbow beam configuration in NF and virtually partitioning the array, the received signal can be expressed as a superposition of near-orthogonal signals from each sub-array. Based on this observation, we developed an algorithm to recover each sub-array's direction by formulat-

Table 4.2: Percentage of angulation failure for different SNR and sensitivity conditions for S-NFBT [45], DS-NFBT and VSSR algorithms.

Detection of $r_{\mathbf{ue}} < 1e-16$			
SNR (dB)	S-NFBT [45]	DS-NFBT	VSSR
-10	0.5533%	0.3336%	0.5858%
0	0.7404%	0.0244%	0.0651%
10	0.5858%	0%	0%
20	0.2034%	0%	0%
30	0.0325%	0%	0%
∞	0%	0%	0%
Detection of $r_{\mathbf{ue}} < 1e-1$			
SNR (dB)	S-NFBT [45]	DS-NFBT	VSSR
-10	4.7270%	0.5533%	1.6435%
0	2.8720%	0.0244%	0.0814%
10	1.3750%	0%	0%
20	0.2766%	0%	0%
30	0.0325%	0%	0%
∞	0%	0%	0%

ing a sparse recovery problem, and angulate the UE's location for effective beam alignment. Our simulation results demonstrated that the proposed method achieves near-optimal beamforming gain performance even under low SNR conditions. In the future, we are planning to investigate the multi-user scenarios and the design of different virtual partitioning of the XL TTD, and utilization of machine learning (ML) for localization and tracking in NF region.

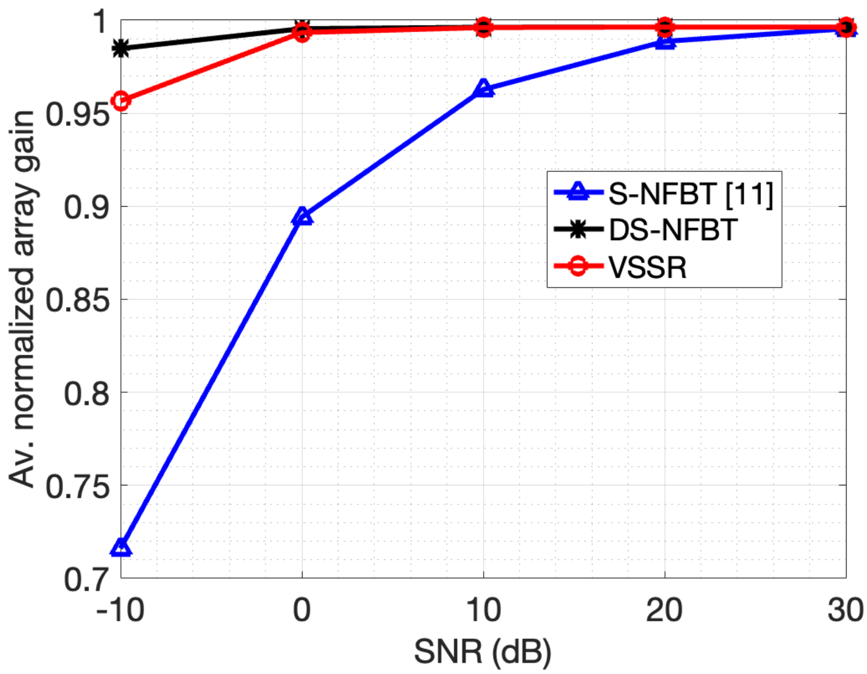


Figure 4.7: Average normalized array gain after beam training with S-NFBT [45], DS-NFBT and VSSR algorithms for different SNR values.

CHAPTER 5

Conclusion

5.1 Summary of Contributions

In Chapter 2, we have demonstrated a low-cost and low-memory split beampattern generation algorithm, the HDB algorithm, based on the homomorphism between TTD array configuration matrices and TTD beampatterns. With extensive simulations, we showed that the proposed algorithm achieves performance close to the high-complexity or high-memory JPTA algorithms [9] with negligible memory requirements, without ignoring the beam squint as the mmFlexible algorithm [14]. Furthermore, we showed that the proposed algorithm achieves better fairness across different UEs.

In Chapter 3, we continued the analysis of the HDB algorithm. First, we showed that changing the shift and scale parameters of the HDB algorithm can change the resulting sub-carrier power allocation of the split beampattern. Then, by data-driven optimization, we found the shift and scale parameters per sub-band that maximize the average received power across sub-bands. The proposed algorithm provides average received power similar to the mmFlexible algorithm [14] while providing a fairer power distribution across sub-bands.

In Chapter 4, we shifted our focus to another challenge of large antenna arrays: near-field channels. We proposed a single OFDM symbol NF beam training algorithm that uses the same time delay and phase shifter values as far-field rainbow beams. In the proposed algorithm, we first showed the near-orthogonality of the virtual sub-arrays' signals and used sparse recovery algorithms to recover each sub-array's signal and localize the user. With extensive simulations, we showed that, even under low SNR conditions, the proposed method provides near-optimal beamforming gain.

5.2 Limitations

We have used ULAs across this thesis, which can limit the feasibility of antenna deployment. Furthermore, throughout the chapters we assumed idealized hardware where TTD elements and phase shifters are ideal, without quantization, gain and phase errors, or noise. Furthermore, Chapter 2 and Chapter 3 are susceptible to error accumulation, where errors in the dictionary can accumulate due to the additive nature of the homomorphism-based algorithm. Furthermore, Chapter 4 requires users to be in a region where the partitions are enough for the far-field assumption per sub-array to hold.

5.3 Future Directions

As an extension, hardware non-idealities of the HDB algorithms in Chapters 2 and 3, such as quantization and phase and time delay errors, can be investigated. Furthermore, multi-user scenarios for the VSSR algorithm in Chapter 4 can be proposed, as well as lower-complexity algorithms for the recovery of each virtual sub-array's signal. Furthermore, extension of the HDB algorithm to low-complexity and low-memory near-field split beampatterns as in [50] is an interesting direction. Finally, the HDB algorithm for hybrid arrays is still an open problem.

REFERENCES

[1] Z. Wang, J. Zhang, H. Du, W. E. I. Sha, B. Ai, D. Niyato, and M. Debbah, "Extremely large-scale MIMO: Fundamentals, challenges, solutions, and future directions," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 117–124, 2024.

[2] Samsung Research, "AI-Native & Sustainable Communication: Samsung 6G Vision 2025," Samsung Research, Tech. Rep., Feb. 2025, white Paper. [Online]. Available: https://cdn.codeground.org/nsr/downloads/researchareas/Samsung_6G_Vision_2025.pdf

[3] R. Singh, A. Kaushik, W. Shin, M. D. Renzo, V. Sciancalepore, D. Lee, H. Sasaki, A. Shojaeifard, and O. A. Dobre, "Towards 6g evolution: Three enhancements, three innovations, and three major challenges," *IEEE Network*, pp. 1–1, 2025.

[4] L. Dai, J. Tan, Z. Chen, and H. V. Poor, "Delay-phase precoding for wideband thz massive mimo," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7271–7286, 2022.

[5] Y. Liu, Z. Wang, J. Xu, C. Ouyang, X. Mu, and R. Schober, "Near-field communications: A tutorial review," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1999–2049, 2023.

[6] E. Björnson, Y. C. Eldar, E. G. Larsson, A. Lozano, and H. V. Poor, "Twenty-five years of signal processing advances for multiantenna communications: From theory to mainstream technology," *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 107–117, 2023.

[7] C.-C. Lin, C. Puglisi, V. Boljanovic, H. Yan, E. Ghaderi, J. Gaddis, Q. Xu, S. Poolakkal, D. Cabric, and S. Gupta, "Multi-mode spatial signal processor with rainbow-like fast beam training and wideband communications using true-time-delay arrays," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3348–3360, 2022.

[8] H. Yan, V. Boljanovic, and D. Cabric, "Wideband millimeter-wave beam training with true-time-delay array architecture," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1447–1452.

[9] V. V. Ratnam, J. Mo, A. Alammouri, B. L. Ng, J. Zhang, and A. F. Molisch, "Joint Phase-Time Arrays: A Paradigm for Frequency-Dependent Analog Beamforming in 6G," *IEEE Access*, vol. 10, pp. 73 364–73 377, 2022, conference Name: IEEE Access.

[10] Z. Wu and L. Dai, "Multiple access for near-field communications: SDMA or LDMA?" *IEEE J. Select. Areas Commun.*, vol. 41, no. 6, pp. 1918–1935, 2023.

[11] M. Cui and L. Dai, "Near-field wideband beamforming for extremely large antenna arrays," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 13 110–13 124, 2024.

[12] A. Alammouri, J. Mo, V. V. Ratnam, B. L. Ng, R. W. Heath, J. Lee, and J. Zhang, "Extending Uplink Coverage of mmWave and Terahertz Systems Through Joint Phase-Time Arrays," *IEEE Access*, vol. 10, pp. 88 872–88 884, 2022, conference Name: IEEE Access.

[13] Y.-H. Nam, A. AlAmmouri, J. Mo, and J. C. Zhang, "Joint phase time array: Opportunities, challenges and system design considerations," 2024. [Online]. Available: https://arxiv.org/abs/2412.01714

[14] I. K. Jain, R. R. Vennam, R. Subbaraman, and D. Bharadia, "mmFlexible: Flexible Directional Frequency Multiplexing for Multi-user mmWave Networks," Jan. 2023, arXiv:2301.10950 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2301.10950

[15] D. Zhao, I. Pehlivan, A. Wadaskar, and D. Cabric, "Fast frequency-direction mapping design for data communication with true-time-delay array architecture," in *2024 International Conference on Computing, Networking and Communications (ICNC)*, 2024, pp. 1071–1076.

[16] A. Wadaskar, D. Zhao, I. Pehlivan, and D. Cabric, "Structured two-stage true-time-delay array code book design for multi - user data communication," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 3378–3384.

[17] I. Pehlivan and D. Cabric, "Homomorphic Directional Beamforming with Analog True Time Delay Arrays," May 2026, arXiv:2605.06930 [eess.SP]. [Online]. Available: http://arxiv.org/abs/2605.06930

[18] ——, "Homomorphic directional beamforming for analog true-time-delay arrays with parameter learning," in *Proc. IEEE Workshop on Smart Antennas (WSA)*, 2026, to appear.

[19] I. Pehlivan, M. C. Ilter, M. Valkama, and D. Cabric, "Near-field beam training with analog extremely large true-time-delay arrays: (invited paper)," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 2024, pp. 858–864.

[20] B. Wang, F. Gao, S. Jin, H. Lin, G. Y. Li, S. Sun, and T. S. Rappaport, "Spatial-wideband effect in massive mimo with application in mmwave systems," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 134–141, 2018.

[21] M. Longbrake, "True time-delay beamsteering for radar," in *2012 IEEE National Aerospace and Electronics Conference (NAECON)*, 2012, pp. 246–249.

[22] R. Rotman, M. Tur, and L. Yaron, "True time delay in phased arrays," *Proceedings of the IEEE*, vol. 104, no. 3, pp. 504–518, 2016.

[23] C. Jans, X. Song, W. Rave, and G. Fettweis, "Fast beam alignment through simultaneous beam steering and power spectrum estimation using a frequency scanning array," in *WSA 2020; 24th International ITG Workshop on Smart Antennas*, 2020, pp. 1–6.

[24] V. Boljanovic, H. Yan, E. Ghaderi, D. Heo, S. Gupta, and D. Cabric, "Design of Millimeter-Wave Single-Shot Beam Training for True-Time-Delay Array," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, May 2020, pp. 1–5, iSSN: 1948-3252.

[25] C.-C. Lin, C. Puglisi, E. Ghaderi, S. Mohapatra, D. Heo, S. Gupta, H. Yan, V. Boljanovic, and D. Cabric, "A 4-element 800mhz-bw 29mw true-time-delay spatial signal processor enabling fast beam-training with data communications," in *ESSCIRC 2021 - IEEE 47th European Solid State Circuits Conference (ESSCIRC)*, 2021, pp. 287–290.

[26] C. Jans, X. Song, W. Rave, and G. Fettweis, "Frequency-selective analog beam probing for millimeter wave communication systems," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[27] A. Wadaskar, V. Boljanovic, H. Yan, and D. Cabric, "3d rainbow beam design for fast beam training with true-time-delay arrays in wideband millimeter-wave systems," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 85–92.

[28] R. Li, H. Yan, and D. Cabric, "Rainbow-link: Beam-alignment-free and grant-free mmw multiple access using true-time-delay array," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1692–1705, 2022.

[29] B. Zhai, Y. Zhu, A. Tang, and X. Wang, "Thzprism: Frequency-based beam spreading for terahertz communication systems," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 897–900, 2020.

[30] J. Mo, A. AlAmmouri, S. Dong, Y. Nam, W.-S. Choi, G. Xu, and J. C. Zhang, "Beamforming with joint phase and time array: System design, prototyping and performance," in *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 2024, pp. 875–881.

[31] I. K. Jain, Y. Sun, S. Cao, and D. Bharadia, "Flexlink demo: Flexible frequency-dependent multi-beamforming with delay-phased array," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1814–1816. [Online]. Available: https://doi.org/10.1145/3636534.3698869

[32] O. Yildiz, A. AlAmmouri, J. Mo, Y. Nam, E. Erkip, and J. C. Zhang, "3d beamforming through joint phase-time arrays," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*, 2024, pp. 1–7.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004. [Online]. Available: https://web.stanford.edu/~boyd/cvxbook/

[34] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing*. Cambridge: Cambridge University Press, 2014.

[35] J. B. Fraleigh, *A First Course in Abstract Algebra*, 7th ed. Pearson, 2003.

[36] R. W. Schafer, "Echo removal by discrete generalized linear filtering." MIT Research Laboratory of Electronics, Tech. Rep., 1969.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[38] S. Ye, M. Xiao, M.-W. Kwan, Z. Ma, Y. Huang, G. Karagiannidis, and P. Fan, "Extremely large aperture array (ELAA) communications: Foundations, research advances and challenges," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7075–7120, 2024.

[39] S. Hu, H. Wang, and M. C. Ilter, "Design of near-field beamforming for large intelligent surfaces," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 762–774, 2024.

[40] M. Cui and L. Dai, "Channel estimation for extremely large-scale MIMO: Far-field or near-field?" *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2663–2677, 2022.

[41] C. Wu, C. You, Y. Liu, L. Chen, and S. Shi, "Two-stage hierarchical beam training for near-field communications," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2032–2044, 2024.

[42] W. Liu, H. Ren, C. Pan, and J. Wang, "Deep learning based beam training for extremely large-scale massive MIMO in near-field domain," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 170–174, 2023.

[43] M. Cui, L. Dai, Z. Wang, S. Zhou, and N. Ge, "Near-field rainbow: Wideband beam training for XL-MIMO," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3899–3912, 2023.

[44] T. Zheng, M. Cui, Z. Wu, and L. Dai, "Near-field wideband beam training based on distance-dependent beam split," 2024. [Online]. Available: https://arxiv.org/abs/2406.07989

[45] S. Shin, J. Moon, S. Kim, and B. Shim, "Subarray-based near-field beam training for 6G terahertz communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Denver, USA,2024, pp. 3189–3194.

[46] C. Herzet, A. Drémeau, and C. Soussen, "Relaxed recovery conditions for omp/ols by exploiting both coherence and decay," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 459–470, 2016.

[47] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of signal processing.* Cambridge University Press, 2014.

[48] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. C. Eldar, "C-hilasso: A collaborative hierarchical sparse modeling framework," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4183–4198, 2011.

[49] "3GPP TS 38.213: NR; Physical layer procedures for control," 3rd Generation Partnership Project (3GPP), Tech. Rep., Sep. 2024, version 17.3.0, Release 17. [Online]. Available: https://www.3gpp.org/DynaReport/38213.htm

[50] Y. Cai, M. Tao, J. Mo, and S. Sun, "Hybrid near/far-field frequency-dependent beamforming via joint phase-time arrays," Jan. 2025. [Online]. Available: https://arxiv.org/abs/2501.15207