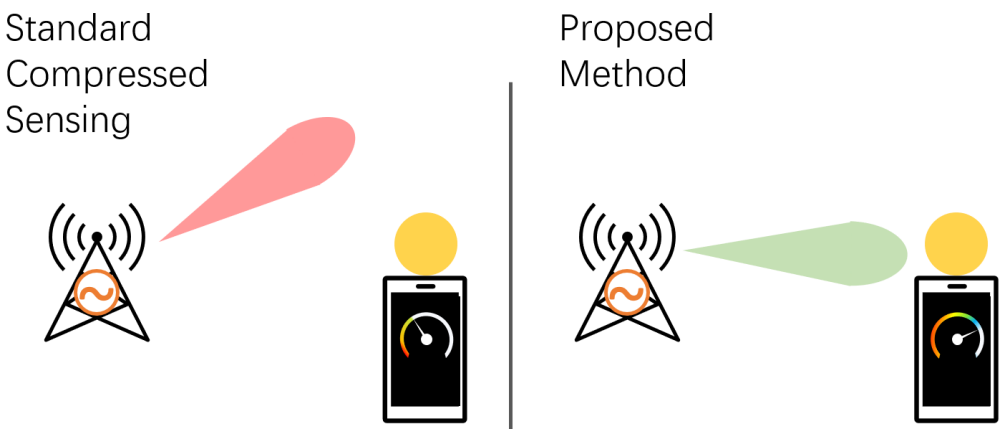
Signal Processing Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

SPS-2023-00

M.Sc. Thesis

Sparse Millimeter Wave Channel Estimation From Partially Coherent Measurements

Weijia Yi B.Sc.



Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Sparse Millimeter Wave Channel Estimation From Partially Coherent Measurements

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Weijia Yi B.Sc.
born in Dali, China

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Sparse Millimeter Wave Channel Estimation From Partially Coherent Measurements"** by **Weijia Yi B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 27\11\2023

Chairman:

prof.dr.ir. A.J. van der Veen

Advisors:

dr. NJ. Myers

dr. G. Joseph

Committee Members:

dr. P. Mohajerin Esfahani

Abstract

This project develops a channel estimation technique for millimeter wave (mmWave) communication systems. Our method exploits the sparse structure in mmWave channels for low training overhead and accounts for the phase errors in the channel measurements due to phase noise at the oscillator. Specifically, in IEEE 802.11ad/ay-based mmWave systems, the phase errors within a beam refinement protocol packet are almost the same, while the errors across different packets are substantially different.

We show that standard compressed sensing algorithms that treat phase noise as a constant fail when channel measurements are acquired over multiple beam refinement protocol packets. Most of the methods that have addressed this problem treat phase noise as purely random, missing the inherent structure within the measurement packets. We present a novel algorithm called partially coherent matching pursuit for sparse channel estimation under practical phase noise perturbations. The proposed approach leverages this partially coherent structure in the phase errors across multiple packets. Our algorithm iteratively detects the support of sparse signal and employs alternating minimization to jointly estimate the signal and the phase errors.

We numerically show that our algorithm can reconstruct the channel accurately at a lower complexity than the benchmarks, and derive a preliminary support detection bound as a performance guarantee.

Acknowledgments

I am deeply grateful for the enriching and transformative experience that my time at TU Delft. Throughout this academic journey, I have had the privilege of working with exceptional individuals and being part of a supportive community. I would like to express my sincere appreciation to those who have played a significant role in this chapter of my life.

I extend my heartfelt gratitude to my daily supervisors, dr. NJ. Myers and dr. G. Joseph, whose unwavering support, meticulous responses to my queries, and continual inspiration, encouragement, and guidance have been instrumental throughout the entire project. Their dedication and willingness to invest substantial time in providing me with valuable advice underscore their commitment to my academic and personal growth. Their exceptional scholarly expertise and insightful perspectives have not only fueled my inspiration but also broadened my intellectual horizons. I extend my appreciation to prof.dr.ir. A.J. van der Veen and dr. P. Mohajerin Esfahanifor graciously accepting the invitation to join the thesis committee.

I would like to express my special appreciation to my parents, whose unwavering support has been my anchor, providing both emotional and physical assistance. I am grateful for the sacrifices they made to enable my pursuit of knowledge. My heartfelt thanks also go to my friends in Delft, who have enriched my journey with shared moments of joy and camaraderie. I will forever memorize the Wifi in 97 kitchen, the move from Cyber-punk 2077, the snow in Poland, and the appointments with an INTJ. Thanks to the Netherlands for its cool summer providing a mosquito-free environment.

Finally, I want to express my sincere thanks to Yuan. His steadfast encouragement and understanding have added immense value to my academic journey. I wholeheartedly wish him outstanding success in his academic pursuits. Looking forward to our shared adventures in the future.

The knowledge acquired at TU Delft will undoubtedly accompany me throughout my life, serving as a constant source of inspiration and growth. As this journey at TU Delft concludes, it marks another departure in my life.

Weijia Yi B.Sc.
Delft, The Netherlands
27\11\2023

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Nowadays, millimeter wave (mmWave) finds extensive applications in diverse fields owing to its enhanced wireless link capacity and smaller size of antennas. Benefiting from the high carrier frequency, the mmWave systems are capable of achieving Gbps data rates by effective beamforming over wide bandwidths, leading to IEEE 802.11ad multiple gigabit wireless system at 60 GHz and 5G cellular telecommunications from 24 GHz to 39 GHz. Besides high data rates, the dense deployment of base stations in 5G is ideal for autonomous vehicle communications. The 60 GHz band is widely accessible for unlicensed use in numerous regions globally, prompting the exploration and development of various technologies centered around this frequency.

Compared to 2.4 and 5 GHz bands, signal propagation at 60 GHz faces challenges such as severe path loss and high attenuation due to human blockage and atmospheric absorption. To compensate for these losses, using phased antenna arrays with precomputed antenna weights becomes a viable option to amplify signal power [3]. The shorter wavelength at 60 GHz allows for the integration of more antenna elements within a single antenna, enhancing spatial resolution and enabling the generation of narrower beams [4].

In a narrow beam wireless system, beam alignment is necessary when the transmitter and receiver need to establish a reliable communication link. It is crucial in scenarios where the spatial orientation of the devices, or the directionality of the antennas, is not fixed or may change over time. When devices are mobile, or when there are environmental factors causing signal blockages or reflections, beam alignment becomes essential for maintaining a stable connection. Beam misalignment can lead to higher path loss and signal attenuation due to a longer indirect path, potentially causing issues with data quality and link budget. To effectively direct the beams at the transmitter (TX), the beam pattern should align with the channel state information (CSI) to maximize the received signal power. In a typical system, the TX transmits a series of training data selected from a known codebook, enabling the receiver (RX) to estimate the current CSI and relay it back to the transmitter. Subsequently, the TX adjusts its beamforming based on this feedback. Our focus lies in the estimation of CSI for beamforming purposes.

1.2 Beam Alignment Strategies

Achieving precise beam alignment is critical for reliable, high-speed data transmission. Here, we introduce four common strategies employed to align the transmission and reception beams in order to maximize the signal strength at the RX, optimizing signal strength, minimizing interference, and ensuring stable connectivity.

1.2.1 Exhaustive Search Beam

Exhaustive search beam stands as a widely adopted technique in wireless communications for precise beam alignment. As illustrated in Figure 1.1, in exhaustive search beam training, the TX meticulously scans narrow beams across all possible beamforming directions. The RX determines the power of signals received from each direction, and the TX identifies the direction with the strongest signal, indicating optimal beam alignment [3]. In the context of exhaustive search, the phase errors in the measurements become insignificant since only the amplitude of measurements is acquired. Concentrating transmission power in a single direction makes exhaustive search robust in intricate propagation environments with high path loss. By thoroughly exploring all potential paths, it ensures reliable communication even in challenging conditions. However, the overhead of this approach increases linearly as the number of beams in the codebook increases. For instance, the number of elements in the DFT codebook is N when the TX has N antennas. In this case, the TX needs to sweep the total N directions, which results in a substantial overhead for a large N.

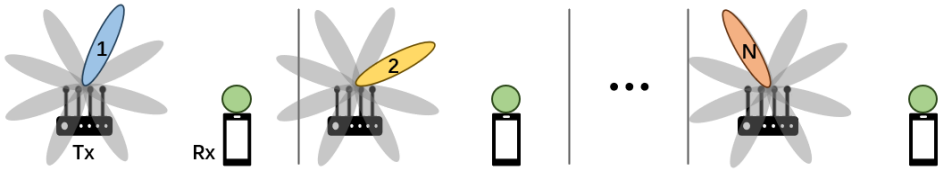


Figure 1.1: An example of exhaustive search with a DFT-based beam codebook. The TX sequentially tries N different beams and the RX determines the transmit beam that results in the highest received power.

1.2.2 Hierarchical Beam Training

Another beam alignment strategy is hierarchical beam training. Unlike exhaustive search, hierarchical beam training employs a tiered approach to explore different beamformers and maximize the beamforming gain. In hierarchical beam training shown in Figure 1.2, the transmitter (TX) initiates by scanning broad sectors,

identifying general directions with potential for optimal alignment. Based on the feedback received from these sectors, the TX refines its focus, narrowing down the beamforming angles to subsectors within the promising directions. This iterative refinement continues until a narrow directional beam results in a high received power [5]. However, as feedback is required for each subsection decision, the overhead will increase when multiple users exist. Another drawback is that at an early training stage before forming the narrow beam, the low SNR environment with noise and interference may lead to misestimation of the section, resulting in a wrong path decision [6].

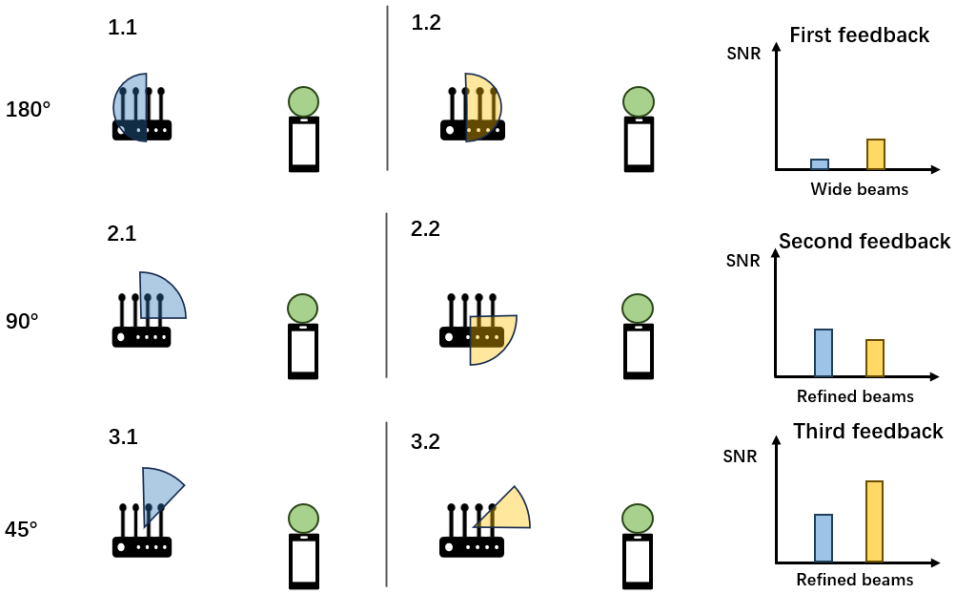


Figure 1.2: An example of hierarchical beam training refining from 180° to 45°. The TX starts sweeping sections from the wide beam, each time the RX evaluates the highest received power and sends the feedback. The TX then refines the beams according to the feedback, until forming a directional narrow beam.

1.2.3 Standard Channel Estimation-based Beam Training

Channel estimation methods rely on detailed knowledge of the CSI, encompassing factors such as path losses, reflections, and interference within the wireless channel. Unlike exhaustive search and hierarchical training, channel estimation techniques focus on understanding the unique characteristics of the channel and aligning beams based on this specific knowledge. Standard channel estimation algorithms, such as least squares (LS) and linear minimum mean squared error (LMMSE) methods [7], necessitate an equal or greater number of measurements than the number of antennas. However, these methods are only suitable for slowly

time-varying channels due to the substantial measurement requirement and high computation overhead [8]. Moreover, the phase corruption has an impact on the estimation results. A general training process for standard channel estimation is shown in Figure 1.3.

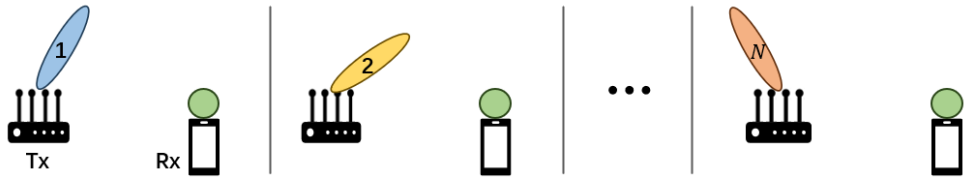


Figure 1.3: An example of standard channel estimation-based beam training. The TX forms different beam patterns, and the RX acquires at least N measurements to estimate the channel, where N is the number of antennas.

- **Compressive Channel Estimation-based Beam Training**

Researchers, in search of training protocols to control overhead and enhance performance, have explored the sparse nature of mmWave channels, leading to the proposal of compressive channel estimation-based beam training techniques [9]. Due to the high scattering properties of mmWave channels, signal power is only along specific arrival directions [10]. Consequently, the wireless channel between the TX and RX exhibits approximate sparsity in the angular domain. In compressive beam training, the TX usually applies random beam patterns from a codebook, and the RX collects measurements for several beam patterns and then employs compressed sensing (CS) algorithms for channel estimation. Compared to standard channel estimation methods, this approach demands fewer measurements as illustrated in Figure 1.4, thereby curtailing training overhead [11].

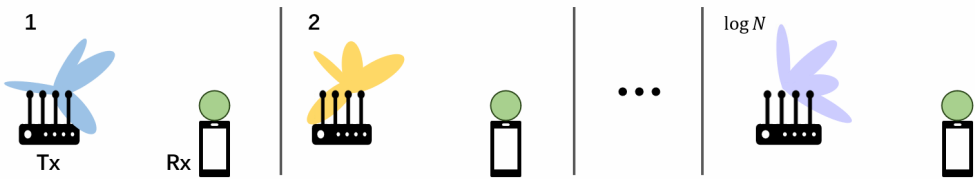


Figure 1.4: An example of compressive channel estimation-based beam training. The TX forms random beam patterns from the codebook, and then the RX acquires M ($M < N$) measurements and estimates the channel using CS-technique.

We will evaluate the necessity of phase information for each strategy and compare them based on both overhead and robustness to additive noise in the measurements. A concise comparison of these strategies is outlined in Table 1.1. For the table, we

denote T_{FB} as the feedback time of the hierarchical approach, and abbreviate the robustness to additive noise as robustness.

Training Strategy	Phase Information	Overhead	Robustness
Exhaustive Search	Not required	$\mathcal{O}(N)$	Very high
Hierarchical Approach	Not required	$\mathcal{O}(T_{FB} \cdot \log(N))$	Low
Standard Channel Estimation	Required	$\mathcal{O}(N)$	Very low
Compressive Approach	Required	$\mathcal{O}(\log(N))$	Very low

Table 1.1: Comparison of the typical beam training strategies from phase information requirement, execution overhead, and robustness to the noise. In this table, we assume that the number of sparse components in the channel is $\mathcal{O}(1)$

In summary, both exhaustive search and the hierarchical approach operate by detecting the direction with the strongest receiving power, creating a directional beam pattern without relying on phase information. However, exhaustive search introduces substantial overhead, while the hierarchical approach is susceptible to additive noise in the measurements. Although conventional channel estimation methods come with high overhead, the compressive approach, capitalizing on the channel's sparse nature, achieves low overhead during the training process. However, the robustness of channel estimation-based training is often lower compared to exhaustive search due to the reliance on accurate CSI. Channel estimation-based methods heavily depend on the ability to accurately estimate the channel characteristics, and any noise or inaccuracies in the estimation process can lead to significant performance degradation. Moreover, standard CS algorithms are sensitive to phase noise in the measurements. In this research, we solve this challenge by exploring the feature of phase noise and jointly estimating the phase noise and the CSI from compressive measurements.

1.3 IEEE 11.ad Beam Refinement Protocol

As the focus of this work is on sparsity-aware channel estimation under phase noise, we first study the protocol employed by commercial IEEE 802.11ad devices to acquire channel measurements during beam training. Additionally, the structure of a training unit and the impact of phase noise on the acquired data are discussed in our model outlined in Chapter 3. Consequently, we will delve into a detailed introduction to this data unit structure and the procedure of beamforming defined in IEEE 11.ad.

1.3.1 Protocol Data Unit Structure

IEEE 802.11ad has formulated a local area network specification that operates within the 60 GHz mmWave band, enabling multigigabit data transmission. This standard introduces a multi-stage beamforming framework specifically crafted to establish dependable connections between pairs of stations (STAs) for subsequent communication. The protocol data unit (PPDU) structure, tailored for beam refinement protocol (BRP), is illustrated in Figure 1.5.

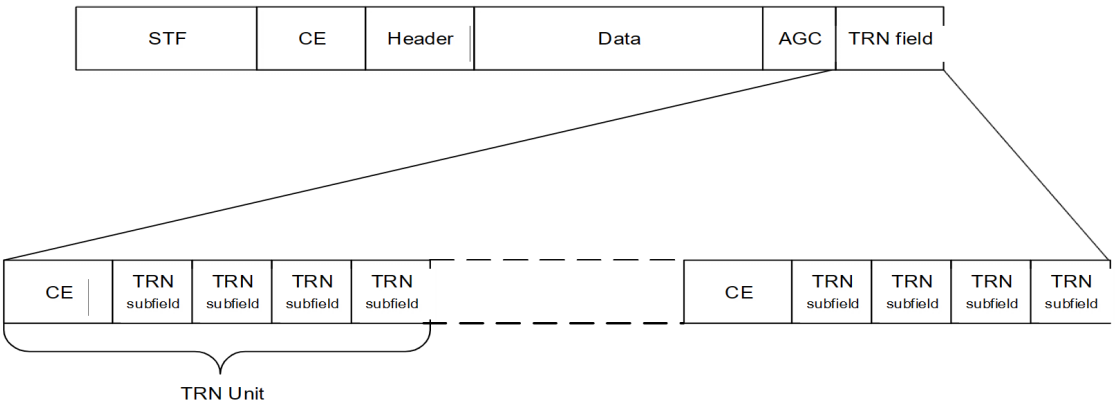


Figure 1.5: Structure of protocol data unit in IEEE 802.11ad for beam refinement protocol. [1]

In a PPDU, the first frame is the short training field (STF) followed by the channel estimation (CE) frame. These two frames enable the receiver to synchronize with the transmitted signal and compensate for channel effects. The Header consists of several fields that define the details of the PPDU to be transmitted, including the identity of the BRP PPDU and the length of the training (TRN) field. The Data field carries the meaningful payload of the communication and must meet the minimum length requirements. The automatic gain control (AGC) field ensures that the received signal's strength is optimized for demodulation. Simultaneously, the TRN field provides training sequences crucial for channel estimation and beam refinement, thereby enhancing the receiver's accuracy in decoding the transmitted data. Two kinds of TRN subfields exist, with TRN-R designated for RX beamforming and TRN-T for TX beamforming. Notably, except for the TRN field, the antenna setting for other fields should remain constant.

1.3.2 Beamforming Procedure

The training process primarily consists of two steps: sector-level sweep (SLS) and beam refinement. The purpose of SLS is to establish a communication link between the initiating station (STA1) and the responding station (STA2). During the SLS

stage, both the initiator and the responder perform sector sweeps. SLS is deemed successful for the initiator if feedback is received from the sweeping sector, and similarly, it is successful for the responder if an ACK signal is received from the sweeping sector. Following the SLS stage, BRP is implemented to further optimize antennas.

In this project, our focus is specifically on this BRP stage. Pilot signals for BRP are encapsulated within each TRN subfield. Due to the planned switching of antennas for compressive antenna training, the time interval between transmitting two subfields is defined as the long beamforming interframe space (LBIFS). This interval ensures efficient handling of antenna switches so that there is enough time for the switches to settle. An example of the beamforming training procedure is illustrated in Figure 1.6.

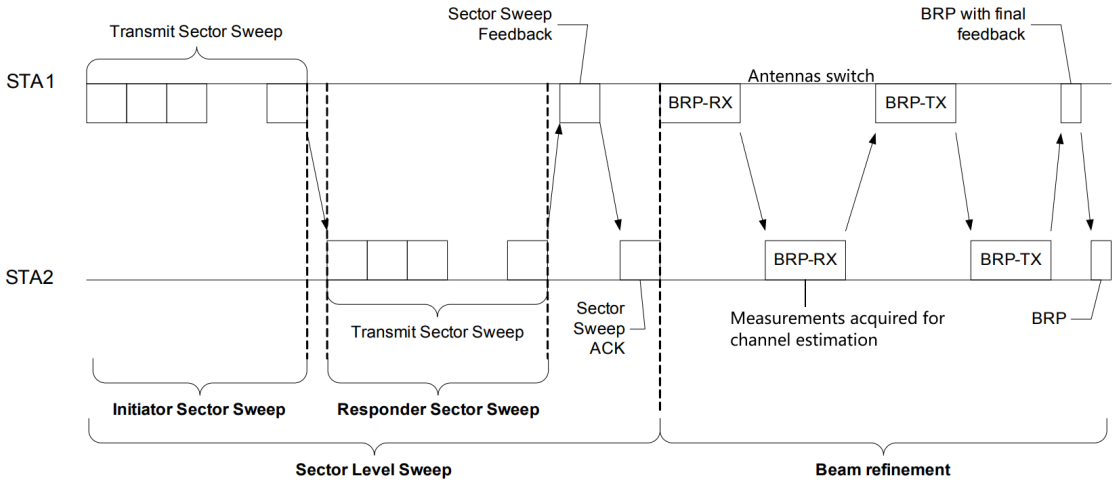


Figure 1.6: An example of beamforming procedure defined in IEEE 802.11.ad between two stations. [1]

1.4 Motivation

In practical scenarios, the deployment of large-scale antennas necessitates a substantial amount of cost-effective and power-efficient baseband hardware. However, due to inherent hardware limitations, several impairments, including residual carrier frequency offset (CFO) and random phase noise at the oscillator [12], introduce perturbations in the phase of compressed channel measurements acquired with typical IEEE 802.11ad/ay hardware. These perturbations create phase ambiguity. Consequently, accurately estimating the channel becomes significantly challenging, especially in fast-changing environments.

It's worth noting that CFO and phase noise are particularly pronounced at mmWave frequencies compared to lower carrier frequencies. This heightened impact results from the rapid increase in phase variance associated with higher frequencies [2]. Considering these perturbations adds complexity to the task of estimating the sparse channel.

We model the phase noise as a Wiener process, i.e. $\phi_m|_{\phi_{m-1}} \sim \mathcal{N}(\phi_{m-1}, \tau^2)$, where τ is $2\pi f_c\sqrt{c(T_m - T_{m-1})}$ [2]. Here, f_c is the carrier frequency, T_m is the time stamp associated with the m^{th} measurement, and c is an oscillator-dependent constant. Some of the practical values of c and associated τ are listed in Table 1.2. From the distribution, we noticed that the variance of phase noise can be accumulated over time, as shown in Figure 1.7.

Oscillator parameter c $((\text{rad}\cdot\text{Hz})^{-1})$	9.4×10^{-19}	4.7×10^{-18}	2.35×10^{-17}	4.7×10^{-17}
Standard deviation τ (rad)	0.1303	0.2924	0.6513	0.9210

Table 1.2: The typical oscillator parameters and the corresponding standard variance of phase noise for 802.11a. [2]

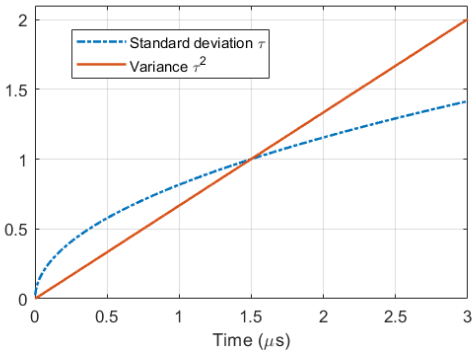
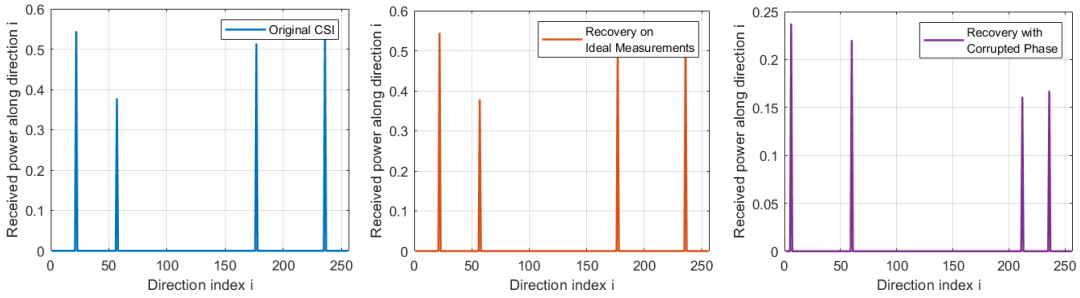


Figure 1.7: The standard deviation and variance of phase noise over 3 μs with carrier frequency $f_c = 60\,\text{GHz}$, $T_m - T_{m-1} = 128\,\text{ns}$.

The phase errors induced by phase noise result in a model mismatch between the standard CS model and the observed measurement model. Unfortunately, this mismatch causes conventional CS-based channel estimation methods [13–15], which are unaware of such phase errors, to fail, as illustrated in Figure 1.8. Therefore, we are motivated to estimate the sparse mmWave channel, leveraging its sparse structure while addressing the challenges posed by phase errors in the measurements.



(a) (b) (c)

Figure 1.8: CS-based channel estimation method (e.g. Orthogonal Matching Pursuit) breaks down (a) Original sparse channel vector. (b) Sparse vector recovered without phase perturbations. (c) Sparse vector recovered with phase perturbations.

We aim to develop a fast algorithm that jointly estimates phase noise and sparse signals by capitalizing on the unique characteristics of phase errors within practical transmission protocols. Several compelling reasons drive this objective. Firstly, a swift training process is essential for real-time applications that demand rapid beamforming adaptation. This speed is pivotal for ensuring stable connectivity and seamless communication in dynamic wireless channels. Additionally, a fast algorithm with low complexity requires fewer computational resources. The reduced overhead provides more time for actual data transmission, enhancing overall system efficiency.

1.5 Outline

The arrangement of this report is as follows.

In Chapter 2, we will show how prior works estimate the sparse channel from a CS perspective, and discuss their advantages and drawbacks. Furthermore, our main contributions are also presented in this chapter.

In Chapter 3, the sparse representation of the channel and the system model will be given. We will further introduce the frame structure of transmitted pilots and the partially coherent model for our algorithm.

In Chapter 4, we will introduce three typical algorithms as well as our benchmarks for comparison, and analyze their computation complexity.

In Chapter 5, we will give details of our proposed partially coherent matching pursuit (PC-MP) method, and give a weak bound for performance guarantee. Additionally, the computation complexity of PC-MP will be analyzed step by step.

In Chapter 6, the simulation settings and metrics for comparison will be explained. With the settings and the metrics, we will show and analyze the simulation results.

In Chapter 7, we will summarize the whole project, draw conclusions based on the simulation results, and further leave some suggestions for future work.

Prior Work Addressing Phase Noise

2

In this chapter, we will review existing sparse mmWave channel estimation methods that are robust to phase errors. We will then discuss our contributions in contrast to prior work on phase error robust channel estimation.

2.1 Prior Works

Before reviewing prior works, we first define the mathematical terms for illustration. A more detailed description of how we derive this model will be deduced in Chapter 3.

A sparse signal refers to a signal in which the majority of its elements are zero, and only a small subset of elements has non-zero values. Mathematically, the sparse level of a vector $\mathbf{x}$ can be represented as

$$\|\mathbf{x}\|_0 = \text{Number of non-zero elements in } \mathbf{x} \tag{2.1}$$

Consider the sparse representation of the channel vector as $\mathbf{x} \in \mathbb{C}^N$, phase-noise free CS measurements can be denoted as

$$\bar{\mathbf{y}} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{2.2}$$

where $\mathbf{w}$ is Gaussian white noise. $\mathbf{A} \in \mathbb{C}^{M \times N}$ is the CS-matrix with $M \ll N$, which means the number of measurements is much less than the cardinality of the beam codebook. The measurements are disturbed by phase noise $\mathbf{p} \in \mathbb{C}^M$. The disturbed measurements $\mathbf{y} \in \mathbb{C}^M$ are given by

$$\mathbf{y} = \text{diag}(\mathbf{p})\mathbf{A}\mathbf{x} + \mathbf{w}, \tag{2.3}$$

where

$$\mathbf{p} = \left[e^{\mathsf{j}\phi_1}, e^{\mathsf{j}\phi_2}, \ldots, e^{\mathsf{j}\phi_M}\right]^{\mathsf{T}}.$$

$\mathbf{y}$ and $\mathbf{A}$ are known while $\mathbf{p}$ and $\mathbf{x}$ are unknown and need to be estimated.

2.1.1 Standard Sparse Channel Estimation

Standard sparse channel estimations assume a long-term phase coherence condition, that is the phase offset is almost invariant and can be approximated as a constant complex number. Thus consider the model as

$$\mathbf{y} = e^{\mathrm{j}\phi}\mathbf{A}\mathbf{x} + \mathbf{w}, \tag{2.4}$$

where $e^{\mathrm{j}\phi}$ is a constant phase offset. Since for beamforming, we do not really care about the phase offset, channel estimation only needs to achieve a global phase stage, thus we can consider the standard CS model (2.2). As vector $\mathbf{x}$ is sparse, with only some of the support non-zero, the estimation of $\mathbf{x}$ can be formed as a non-convex optimization as

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0 \\ \text{s.t. } & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leqslant \varepsilon. \end{aligned} \tag{2.5}$$

ℓ_0 norm minimization in (2.5) is a non-convex NP-hard problem. Greedy pursuit is a choice that iteratively selects the most promising elements of $\mathbf{x}$ and delivers the currently optimal value based on the available measurements, gradually converging towards an accurate sparse representation. For instance, the orthogonal matching pursuit (OMP) first identifies the component with the most contribution to $\mathbf{y}$, estimates it then subtracts it from $\mathbf{y}$. After repeating the above process, $\mathbf{x}$ can be successfully recovered. We can also transform (2.5) as

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1 \\ \text{s.t. } & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leqslant \varepsilon. \end{aligned} \tag{2.6}$$

to relax ℓ_0 norm minimization to a convex ℓ_1 norm minimization. ℓ_1 norm can partly exploit the sparse feature of $\mathbf{x}$ and lead to a sparse solution for the dual problem with high probability. It is noticeable that the global solution of (2.6) might be a local minimal of (2.5). On the other hand, to reach a sparse solution, there is an extra constraint for the selection of matrix $\mathbf{A}$. Another approximation

$$\begin{aligned} \hat{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_p \quad (0 < p < 1) \\ \text{s.t. } & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leqslant \varepsilon. \end{aligned} \tag{2.7}$$

has been proved with better performance compared to ℓ_1 norm relaxation [16]. The minimization of $\|\mathbf{x}\|_p$ $((0 < p < 1))$ can be solved by computing the proximity operator, e.g. a Newton's method-based algorithm proposed in [17].

In practice, the assumption of long-term coherent phase offset is quite challenging with existing hardware due to the high frequency of mmWave. As discussed in Chapter 1, the existence of phase noise leads to a breakdown in the performance of the above CS approaches.

2.1.2 Prior Works on Solving Phase Noise

Considering that CFO and random phase errors can corrupt the phase of measurements over time, a series of research assuming non-coherence measurements assigns independent phase errors in vector $\mathbf{p}$. Consider the m^{th} measurement,

$$y[m] = e^{\mathrm{j}\phi_m}\mathbf{A}(m,:)\mathbf{x} + w[m].$$

When ϕ_m is fully random, e.g. $\phi_m \sim \mathcal{U}(0, 2\pi)$, the phase information of $\mathbf{A}(m,:)\mathbf{x}$ is completely lost. In such a case, it suffices to consider just the magnitude of $\mathbf{y}$ to estimate $\mathbf{x}$, i.e.

$$|\mathbf{y}| = |\mathbf{A}\mathbf{x} + \mathbf{w}|. \tag{2.8}$$

Several studies have delved into the model (2.8), where independent random phase offsets are assigned to each training slot, creating a joint phase retrieval and sparse recovery problem. An early study, exemplified by [18], combined a CS-based algorithm with a Kalman filter to track phase noise in a narrowband channel. Initially, it utilized matching pursuit to estimate the sparse signal using model (2.2), followed by tracing the phase noise with (2.3). However, the accumulation of phase noise variance posed a significant challenge for accurate prediction, especially when dealing with a long series of training data. Another approach employed hard-thresholding gradient descent for (2.5) to estimate the sparse vector, with an initialized support set. However, the phase retrieval step, required anchor measurements with known phase errors [19]. Obtaining specific known phase errors in practical scenarios proved to be challenging. In a different vein, [20] proposed a method that jointly estimated phase errors and the channel by constructing high-dimensional sparse tensors composed of signal and phase error components. This approach transformed the original problem into a standard sparse recovery scheme. However, the introduction of this complex structure led to high computational complexity. To mitigate this complexity, a sparse bipartite graph code-based phaseless decoding scheme was introduced in [21]. The channel estimation process was formulated as a binary hypothesis test problem based on its coding scheme. In this scheme, the sensing matrix exhibited a specific structure according to the bipartite graph, simplifying the computational demands of the channel estimation process.

The algorithms mentioned above, as presented in [19–21], do not fully exploit the structure in the phase errors. However, due to the short packet signaling utilized for beam training in IEEE 802.11ad/ay standards, a specific structure of phase errors can be effectively utilized. To handle this structure, [22] extended the compressive phase retrieval (CPR) framework to the *partially coherent estimation model*. Here, partially coherent indicates that the phase errors fluctuate slightly during a short time slot, yet remain relatively independent across packets. Within this model, [22] defined alternating descent steps for both signal and phase estimation, employing

a hard-thresholding scheme. Similarly, [23] developed a message-passing-based algorithm for partially coherent sparse recovery, adopting a statistical perspective and utilizing the Expectation-Maximization (EM) algorithm.

2.1.3 Literature Gaps

The prior works on sparse channel estimation from a compressive beam training perspective have all exploited the sparse characteristic of mmWave channel, and then designed algorithms based on different model assumptions. We can conclude that the long-term coherence of the phase is unrealistic, while the non-coherent model did not utilize the information from the phase corruptions. According to the existing protocol, the partial coherence assumption relaxes the phase error structure to a more tractable one. In later chapters, we will show the feasibility of this approximation as well. Therefore, we adopt the partially coherent measurement model in our research and then design the algorithm based on it.

In the context of device mobility and changing propagation environments, the swift establishment of new connections requires a low-complexity algorithm to estimate the current channel state. Previous methods, such as the one introduced in [20], involve the estimation of a low-rank higher-order sparse matrix. However, this approach significantly increases the sparsity level and the size of the sparse vector and sensing matrix, leading to heightened computational and storage complexity. Additionally, the method in [20] fails to deal with complex phase error, which will be analyzed in Chapter 6. Other algorithms, including those in [19] and [22], utilize gradient descent with fixed step sizes, which often require numerous iterations to achieve convergence. In Chapter 6 we will analyze the computation complexity and operation time of the method proposed in [22]. Considering that ℓ_p optimization with $0 < p \leq 1$ can yield highly accurate recovery but is time-consuming compared to greedy methods, we have developed a novel algorithm based on greedy schemes, specifically matching pursuit, to address these challenges efficiently.

2.2 Our Contributions

We develop a compressive greedy algorithm for joint sparse channel and phase noise estimation with a partially coherent measurement model. The proposed method is with improved accuracy, low complexity, and no additional prior knowledge. The partially coherent measurement model is motivated by the signaling structure used in IEEE 802.11ad/ay standards, wherein a spatial channel measurement is acquired using the TRN subfield within a beam refinement protocol packet [24]. With a typical phase noise process, phase coherence is preserved within a packet, but lost across different packets.

Our main contribution is the application of a joint support detection rule across packets, combining alternating minimization and matching pursuit techniques to estimate both the channel and phase noise for channel estimation. To assess the algorithm's performance, we also derive guarantees on partial support recovery. By assuming that the phase noise remains constant within a BRP packet but varies over packets, we substantially reduced the number of parameters to be estimated. Also, matching pursuit in our method results in a lower computational complexity than other benchmarks. Finally, we show by simulations that our algorithm outperforms orthogonal matching pursuit (OMP) and Sparse-Lift [20] in the presence of phase noise.

Partially Coherent Channel Measurement Model

3

In this chapter, we aim to derive the model for partially coherent measurements. We begin by introducing the narrowband assumption applicable to multi-path wireless channels. Based on this assumption, we establish a sparse representation of the wireless channel, forming the foundation for our system model. Subsequently, we introduce our assumption regarding phase noise and proceed to derive the model for partially coherent measurements.

3.1 Narrowband Assumption

The narrowband assumption in the context of wireless communication systems is derived based on the relative magnitudes of the signal bandwidth and the delay spread of the channel. The delay spread is a measure of the spread of time delays of multipath components in a channel.

To derive the narrowband assumption from a passband signal, let us consider a wireless communication system with single-carrier modulation. The baseband signal $z(t)$ can be modulated before transmission as a passband signal

$$s(t) = \mathcal{R}\left\{z(t)e^{\mathrm{j}2\pi f_c t}\right\}. \tag{3.1}$$

Thinking of a multi-path channel between an antenna at TX and an antenna at RX as shown in Figure 3.1, the TX transmits the passband $s(t)$, the received signal from the k^{th} path delayed by τ_k is

$$s_{\tau_k}(t) = s(t-\tau_k) = \mathcal{R}\left\{z(t-\tau_k)e^{-\mathrm{j}2\pi f_c \tau_k}e^{\mathrm{j}2\pi f_c t}\right\}, \tag{3.2}$$

where f_c is the carrier frequency.

The root mean square (RMS) delay spread is a measure of the spread of delay values in a wireless communication channel and can be derived from the power delay profile of the channel. It is calculated as the square root of the second central moment of the power delay profile. Mathematically, it is expressed as

$$\Delta\tau = \sqrt{\frac{\sum_{k=1}^{K} P_k \left(\tau_k - \bar{\tau}\right)^2}{\sum_{k=1}^{K} P_k}}, \tag{3.3}$$

where P_k is the power of the k^{th} path, and $\bar{\tau}$ is the mean delay defined as

$$\bar{\tau} = \frac{\sum_{k=1}^{K} P_k \tau_k}{\sum_{k=1}^{K} P_k}. \tag{3.4}$$

The narrowband assumption is valid when the transmission bandwidth $1/T_s$ is much smaller than the reciprocal of the delay spread, i.e.

$$\frac{1}{T_s} \ll \frac{1}{\Delta\tau}.$$

This condition ensures that the channel variations over the signal bandwidth are negligible, allowing us to treat the channel as effectively constant over that bandwidth.

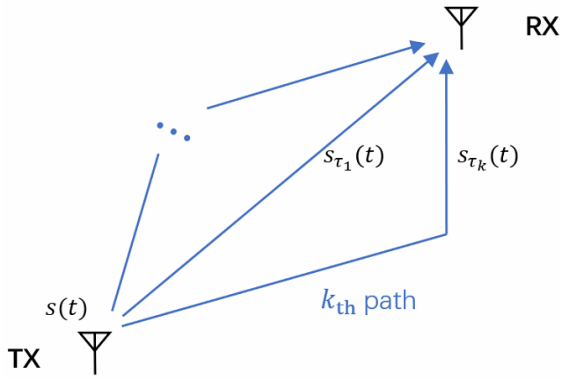


Figure 3.1: An illustration of the delay spread of the multi-path channel for a pair of antennas.

3.2 Channel Model

We consider a narrowband system and model the multiple input single output (MISO) channel between the TX and the RX as a vector $\mathbf{h} \in \mathbb{C}^N$ with multiple paths. Considering TX as a uniform linear array as shown in Figure 3.2, let Δ_T denote the antenna spacing, θ_k the direction of departure (DoD), and c_w the wave speed, then the delay between adjacent antennas is

$$\frac{\Delta_T \sin\theta_k}{c_w}. \tag{3.5}$$

Further denote carrier wavelength

$$\lambda_c = c_w / f_c \,,$$

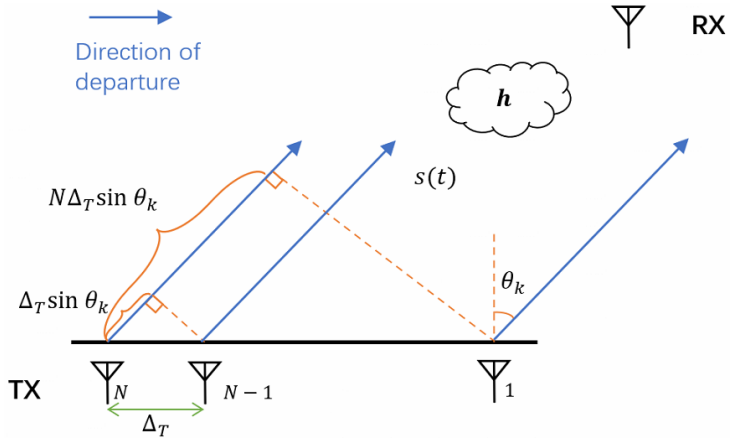


Figure 3.2: An illustration of the MISO system.

then (3.5) is equivalent to

$$\frac{1}{f_c} \cdot \frac{\Delta_T \sin\theta_k}{\lambda_c} \ . \tag{3.6}$$

Denote the attenuation of the k^{th} path as $\bar{h}_k$ and the number of antennas at TX as N, the continuous-time impulse response $\bar{\mathbf{h}}(t)$ is given by

$$\bar{\mathbf{h}}(t) = \sum_{k=1}^{K} \bar{h}_k \begin{bmatrix} \delta\left(t - \frac{d_k}{c_w}\right) \\ \delta\left(t - \frac{d_k + \Delta_T \sin\theta_k}{c_w}\right) \\ \delta\left(t - \frac{d_k + 2\Delta_T \sin\theta_k}{c_w}\right) \\ \vdots \\ \delta\left(t - \frac{d_k + (N-1)\Delta_T \sin\theta_k}{c_w}\right), \end{bmatrix} \tag{3.7}$$

where d_k is the distance of the k^{th} path. In a narrowband channel and a linear time-invariant system, where the wireless channel exhibits certain characteristics, we can draw a connection between time and spatial domains akin to a Fourier transform pair, i.e.

$$\mathbf{h} = \mathcal{F}(\bar{\mathbf{h}}(t)),$$

where $\mathcal{F}$ denotes the Fourier transform operation. Therefore, the spatial response

can be represented as

$$\mathbf{h} = \sum_{k=1}^{K} \bar{h}_k e^{\left(-\mathrm{j}2\pi f_c \frac{d_k}{c_w}\right)} \begin{bmatrix} 1 \\ e^{\left(-\mathrm{j}2\pi f_c \frac{\Delta_T \sin\theta_k}{c_w}\right)} \\ e^{\left(-\mathrm{j}2\pi f_c \frac{2\Delta_T \sin\theta_k}{c_w}\right)} \\ \vdots \\ e^{\left(-\mathrm{j}2\pi f_c \frac{(N-1)\Delta_T \sin\theta_k}{c_w}\right)}. \end{bmatrix} \tag{3.8}$$

Let

$$h_k = \bar{h}_k e^{\left(-\mathrm{j}2\pi f_c \frac{d_k}{c_w}\right)} \tag{3.9}$$

as the complex path gain, again denote carrier wavelength $\lambda_c = c_w/f_c$ and take a half-wavelength spacing $\Delta_T = \lambda_c/2$, then the channel can be written as

$$\mathbf{h} = \sum_{k=1}^{K} h_k \mathbf{a}(\theta_k). \tag{3.10}$$

Here, $\mathbf{a}(\theta_k)$ is the transmit array response vector given by

$$\mathbf{a}(\theta_k) = \left[1, e^{-\mathrm{j}\pi \sin\theta_k}, \cdots, e^{-\mathrm{j}\pi(N-1)\sin\theta_k}\right]^\top . \tag{3.11}$$

the complex gain is assumed to follow circularly symmetrix complex Gaussian distribution, i.e. $h_k \sim \mathcal{CN}(0,1)$. θ is assumed to be uniformly distributed in $(-\pi, \pi)$.

Due to the high scattering observed at mmWave frequencies, only a small number of the directions of arrival (DoA) are effective. This characteristic implicit that in the angle domain, the channel vector is sparse. To acquire the sparse expression, we use the discrete Fourier transform dictionary to sample and obtain the angle-domain representation of the channel,

$$\mathbf{x} = \mathbf{U}_N \mathbf{h}, \tag{3.12}$$

where $\mathbf{U}_N$ is the $N \times N$ unitary discrete Fourier transform matrix denoting as

$$\mathbf{U}_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-\mathrm{j}\frac{2\pi}{N}} & \cdots & e^{-\mathrm{j}\frac{2\pi}{N}(N-1)} \\ : & \vdots & : & \vdots \\ 1 & e^{-\mathrm{j}\frac{2\pi(N-1)}{N}} & \cdots & e^{-\mathrm{j}\frac{2\pi(N-1)}{N}(N-1)} \end{bmatrix} .$$

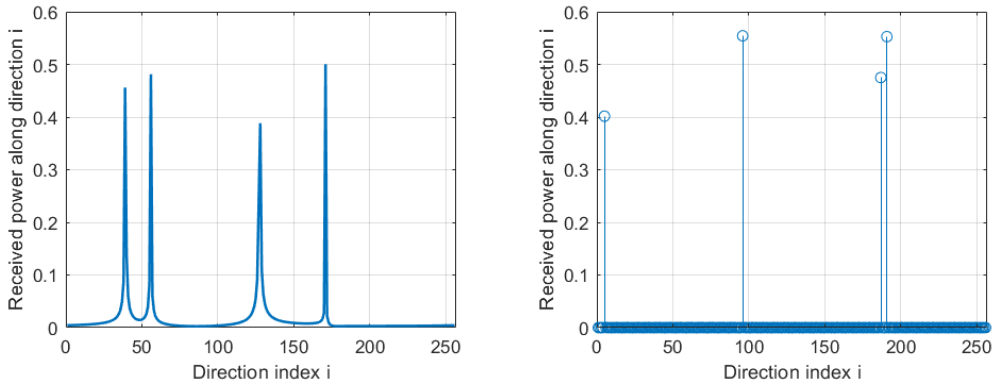
Due to high scattering at mmWave carrier frequencies, the vector $\mathbf{x}$ in (3.12) is approximately sparse with high probability. Spectrum leakage occurs due to the

period mismatch of the on-grid spatial sampling. For simplicity, we assume a perfectly sparse channel vector in this project. This assumption is valid when

$$\pi \sin(\theta_k) = Q \cdot \frac{2\pi}{N} , \tag{3.13}$$

where $k \in \{1, 2, \ldots, K\}, Q \in \mathbb{N}_+$.

Examples of the approximately sparse channel and the perfectly sparse channel are given in Figure 3.3.



(a) (b)

Figure 3.3: Sparse representation of the channel with Number of Effective DoA = 4. (a) is a mmWave channel that is approximately sparse due to leakage effects with the DFT; (b) is the perfectly sparse channel when the DoAs match the angular period defined in (3.13).

3.3 System Model

Before introducing the system model, we first describe the frame structure used to obtain measurements for channel estimation. To this end, we consider the IEEE 802.11ad frame structure shown in Figure 3.4. The TX applies a distinct beamformer for the RX to acquire a measurement. The RX acquires channel measurements within the channel coherence time over P different packets. We define M as the number of beamformers applied in each packet. With IEEE 802.11ad, M can be at most 128 [1]; however, it can often be smaller (e.g. 16) to acquire redundant measurements and enhance the spreading gain. In that case, numerous beam refinement protocol (BRP) packets can be used to acquire enough spatial measurements for channel estimation.

We consider a mmWave system with an analog antenna array comprising N antennas at the transmitter (TX) and a single antenna receiver (RX) as shown in

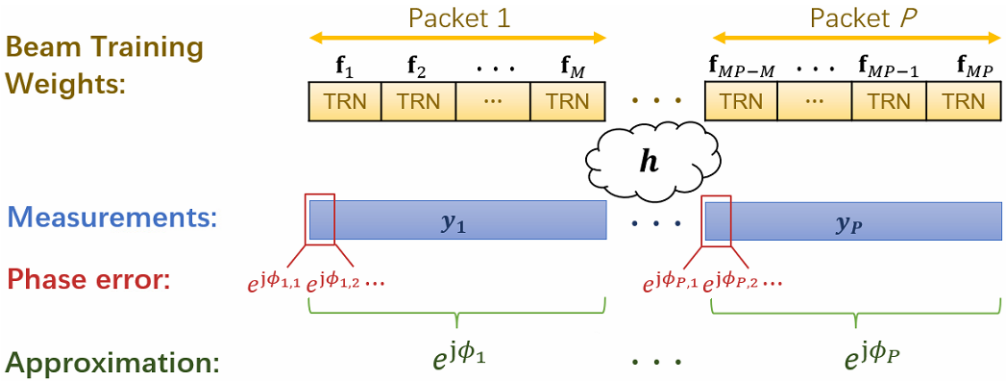


Figure 3.4: Partially coherent measurement model where the phase errors in the channel measurements are assumed to be constant within each packet.

Figure 3.5. The focus of this project is on transmit beam alignment through channel estimation. Although we assume a single antenna at the RX for ease of exposition, our approach can be extended to multi-antenna receivers using an appropriate array response vector.

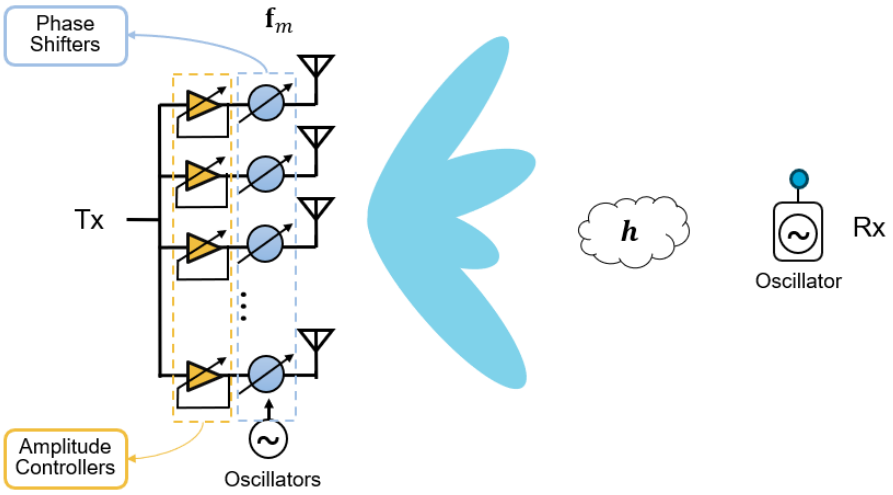


Figure 3.5: An mmWave MISO system with an analog antenna array at the TX and a single antenna at the RX.

Let $\mathbf{f}_m \in \mathbb{C}^N$ denote the m^{th} beam training weights, at the TX and $s[m]$ denote the single m^{th} pilot symbol for beam training, then the codeword applied to the TX array is

$$\mathbf{f}_m \cdot s[m],$$

thus the ideal measurement of the signals received at RX is

$$\mathbf{f}_m^* \mathbf{h} s[m]$$

Note that $\mathbf{f}_m^*$ is the conjugate transpose of $\mathbf{f}_m$. This measurement is perturbed by phase noise ϕ_m and additive white Gaussian noise $w[m]$ of variance σ^2. Let $y[m]$ denote the m^{th} received channel measurement,

$$y[m] = e^{\mathrm{j}\phi_m}\mathbf{f}_m^*\mathbf{h}s[m] + w[m] \tag{3.14}$$

Here, the phase noise ϕ_m can be modeled as a Wiener process

$$\phi_m|_{\phi_{m-1}} \sim \mathcal{N}(\phi_{m-1}, \tau^2),$$

where

$$\tau = 2\pi f_c\sqrt{c(T_m - T_{m-1})}.$$

Here, c is an oscillator-dependent constant, and T_m is the time stamp associated with the m^{th} measurement. In the training packets, $s[m]$ is a known pilot symbol, here we set $s[m] = 1$. Using (3.12), (3.14) can be further written as

$$y[m] = e^{\mathrm{j}\phi_m}\mathbf{f}_m^*\mathbf{U}_N^*\mathbf{x} + w[m], \tag{3.15}$$

Under these modeling assumptions, we aim to estimate the sparse channel $\mathbf{x}$ using the measurements from (3.15) with the knowledge of $\mathbf{f}_m$, for $m = 1, 2, \ldots, M$.

3.4 Partially Coherent CS model

To formulate the CS model, we notice that the time difference between successive measurements in a BRP packet, i.e. $T_m - T_{m-1}$, is 128 ns in IEEE 802.11ad. In contrast, the difference between the successive packets can range from 3 μs to 44 μs [25]. Therefore, a high variance phase offset is introduced in the measurements when switching to a new packet. Figure 3.6 shows a realization of ϕ_m when $M = 16$ measurements are acquired in each of the $P = 4$ packets.

To develop a tractable algorithm, we ignore the phase variations within each packet and only consider phase offsets across different packets. The robustness of our algorithm to phase variations within the packets is studied in Chapter 6.

Under the above approximation assumption, the measurements can be expressed as a partially coherent CS model [22]. We define e^{ϕ_p} as the phase error in the measurements acquired within the p^{th} packet. The vector of MP channel measurements, acquired over P packets, can be expressed in terms of the CS matrix $\mathbf{A} \in \mathbb{C}^{MP\times N}$ and the phase errors $\{e^{\phi_1}, e^{\phi_2}, \ldots e^{\phi_P}\}$. Based on (3.14), we define the m^{th} row of the CS matrix as

$$\mathbf{A}(m,:) = \mathbf{f}_m^*\mathbf{U}_N^*, \tag{3.16}$$

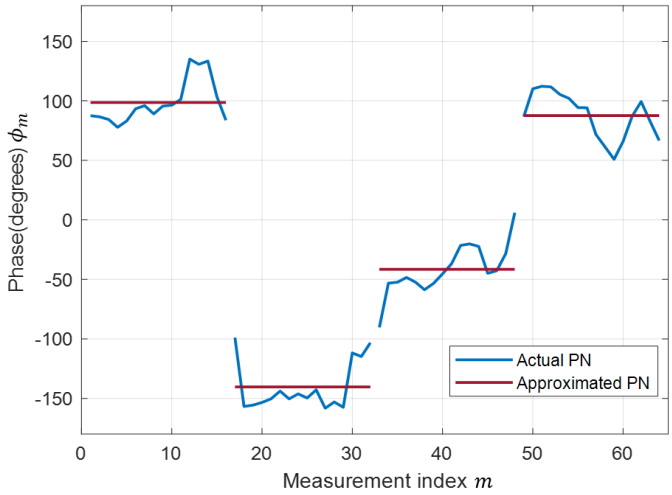


Figure 3.6: A realization of the phase noise process with the measurement index. Our model assumes that the phase error within each packet is the same.

and a diagonal matrix containing the phase errors as

$$\mathbf{\Phi} = \begin{bmatrix} e^{\mathrm{j}\phi_1}\mathbf{I}_M & & & \\ & e^{\mathrm{j}\phi_2}\mathbf{I}_M & & \\ & & \ddots & \\ & & & e^{\mathrm{j}\phi_P}\mathbf{I}_M \end{bmatrix}_{MP\times MP} \tag{3.17}$$

where $\mathbf{I}_M$ is an identity matrix of size $M \times M$. The vector version of (3.14) is then

$$\mathbf{y} = \mathbf{\Phi}\mathbf{A}\mathbf{x} + \mathbf{w}, \tag{3.18}$$

where $\mathbf{\Phi}$ and $\mathbf{x}$ are unknown. We observe from (3.18) that $\mathbf{x}$ can only be estimated up to a global phase. This is because $(\Phi_1 e^{-\mathrm{j}\delta}, \mathbf{x}_1 e^{\mathrm{j}\delta})$ is a solution to (3.18) whenever $(\Phi_1, \mathbf{x}_1)$ is a solution. Hence, the goal of partially coherent CS is to estimate $\mathbf{x}$, up to a global phase, from the measurements in (3.18).

We can split the measurement model in (3.18) on a packet-by-packet basis. The measurements from the p^{th} packet correspond to rows $(p-1)M+1$ to pM of (3.18). We define the measurements acquired in the p^{th} packet as $\mathbf{y}_p$, and the CS measurement matrix associated with the p^{th} packet as $\mathbf{A}_p$. Specifically,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_P \end{bmatrix}_{MP\times N}$$

and

$$\mathbf{y}_p = e^{\mathrm{j}\phi_p}\mathbf{A}_p\mathbf{x} + \mathbf{w}_p, \tag{3.19}$$

where $\mathbf{w}_p$ is additive noise. Now our channel estimation problem is equivalent to estimating $\mathbf{x}$ from the phase perturbed measurements $\{\mathbf{y}_p\}_{p=1}^{P}$ acquired using CS matrices $\{\mathbf{A}_p\}_{p=1}^{P}$. For beam training weights $\mathbf{f}_m$, we initialize it as a unit vector, keeping the codebook as a Fourier matrix.

In summary, we establish the MISO system model and illustrate the sparse representation of the channel vector under the narrowband assumption. Analyzing the transmission protocol, we explore the structural characteristics of phase noise in transmission packets, leading to the construction of partially coherent measurements. The upcoming chapter will delve into three methods for channel vector estimation under the influence of phase noise.

Existing Methods for Sparse Channel Estimation Under Phase Noise

4

Here we will introduce the idea and implementations of *OMP* [26], *Sparse-Lift* [20] and *PC-CPR* [22] with different assumptions on phase noise. These CS-based algorithms are the benchmarks in our simulations.

4.1 Orthogonal Matching Pursuit (OMP)

4.1.1 Algorithm flow

For OMP, we consider the long-term coherent model (2.2) and ℓ_0 optimization (2.5). Although this approach does not account for phase errors, we still discuss standard OMP as it will aid the explanation of our proposed PC-MP algorithm later. There are two fundamental steps for this simple and fast method: support detection and signal estimation. For support detection, it first identifies the component with the most contribution to $\mathbf{y}$ and records its index. Then, it solves an LS-minimization problem to estimate the signal contributed by that index. After that, this contribution is subtracted from $\mathbf{y}$. After repeating the above process, $\mathbf{x}$ can be successfully recovered.

More specifically, the OMP first initializes the residual $\mathbf{r}^{(0)} = \mathbf{y}$ and support set $\lambda^{(0)} = \varnothing$. Until iterating K times, where K is the sparsity of $\mathbf{x}$, or matching the stop criteria, e.g. the error between $\mathbf{y}$ and the reconstructed measurements is less than a small constant, i.e.

$$\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leqslant \varepsilon$$

at the t^{th} loop the support set updates by adding the column index k of the sensing matrix $\mathbf{A}$ that maximizes the projection on the residual vector, i.e.

$$\begin{aligned} k^{(t)} &= \underset{k}{\operatorname{argmax}} \left| \mathbf{A}(:,k)^{\mathsf{T}} \mathbf{r}^{(t-1)} \right| \\ \lambda^{(t)} &= \lambda^{(t-1)} \cap k^{(t)}. \end{aligned}$$

Then the signal is estimated by solving a least square (LS) problem

$$\hat{\mathbf{x}} = \underset{\hat{\mathbf{x}}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{A}_{\lambda^{(t)}} \hat{\mathbf{x}}\|_2$$

as

$$\hat{\mathbf{x}}^{(t)} = \mathbf{A}^{\dagger}_{\lambda^{(t)}}\mathbf{y}.$$

The current residual is updated by subtracting the contribution of the current estimation as

$$\mathbf{r}^{(t)} = \mathbf{y} - \mathbf{A}_{\lambda^{(t)}}\hat{\mathbf{x}}^{(t)}.$$

4.1.2 Computation Complexity

Iterating K times where K is the sparsity level takes $\sum_K \mathcal{O}(T(k))$ FLOPS, where $T(k)$ is the computation complexity of k_{th} iteration.

1. Product of $\mathbf{A}(:,k)^*\mathbf{r}_{p,t-1}$ takes $\mathcal{O}(MPN)$ FLOPS. Selecting the maximum over N scalars takes $\mathcal{O}(N\log(N))$ FLOPS.
2. Pseudo inverse operation takes $\mathcal{O}((MP)^2k) + \mathcal{O}((MP)^3)$ FLOPS.
3. Update of residual takes $\mathcal{O}(MP) + \mathcal{O}(MPk^2)$ FLOPS.

4.2 Sparse-Lift

4.2.1 Algorithm Flow

In this section, we discuss how the Sparse-Lift algorithm in [20] can be applied to solve for the sparse vector in the partially coherent sparse recovery problem. Sparse-Lift can be extended to match a partially coherent model. This algorithm considers a non-coherent model with a random phase error vector. The idea in [20] is to jointly estimate the calibration errors and the sparse vector by solving for a high-dimensional lifted matrix, which is an outer product of the error vector and the sparse vector. After this sparse matrix is estimated, singular value decomposition (SVD) is employed to separate these two components. To apply Sparse-Lift, we first define the calibration error vector as

$$\mathbf{p} = (e^{\mathsf{j}\phi_1}, e^{\mathsf{j}\phi_2}, \cdots, e^{\mathsf{j}\phi_P}),$$

so we can rewrite the phase matrix (3.17) as

$$\mathbf{\Phi} = \mathrm{diag}(\mathbf{B}\mathbf{p}),$$

where

$$\begin{aligned}\mathbf{B} &= \mathbf{I}_P \otimes \mathbf{1}_M^{\mathsf{T}} \\ &= \begin{bmatrix} \mathbf{1}_M & & \\ & \ddots & \\ & & \mathbf{1}_M \end{bmatrix}_{MP\times P}.\end{aligned}$$

In this way, the measurement vector $\mathbf{y}$ in (3.18) is then

$$\mathbf{y} = \text{diag}(\mathbf{Bp})\mathbf{Ax} + \mathbf{w}, \tag{4.1}$$

and the m^{th} entry of $\mathbf{y}$ is given by

$$y[m] = \tilde{\mathbf{b}}_m^{\mathsf{T}} \mathbf{p}\mathbf{x}^{\mathsf{T}} \tilde{\mathbf{a}}_m + w[m], \tag{4.2}$$

where $\tilde{\mathbf{b}}_m^{\mathsf{T}}$ is the m^{th} row of $\mathbf{B}$ and $\tilde{\mathbf{a}}_m$ is the transpose of the m^{th} row of $\mathbf{A}$. We define

$$\mathbf{X} = \mathbf{p}\mathbf{x}^{\mathsf{T}}$$

as the lifted matrix, which is sparse as $\mathbf{x}$ is sparse. Using the identity $\mathbf{b}^{\mathsf{T}}\mathbf{X}\mathbf{a} = (\mathbf{a}^{\mathsf{T}} \otimes \mathbf{b})\text{vec}(\mathbf{X})$, we can rewrite (4.2) as

$$y[m] = (\tilde{\mathbf{a}}_m^{\mathsf{T}} \otimes \tilde{\mathbf{b}}_m^{\mathsf{T}})\text{vec}(\mathbf{X}) + w[m], \tag{4.3}$$

which is a linear measurement of the lifted vector $\text{vec}(\mathbf{X})$. Then the collection of all the measurements can be represented as

$$\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{a}}_1^{\mathsf{T}} \otimes \tilde{\mathbf{b}}_1^{\mathsf{T}} \\ \vdots \\ \tilde{\mathbf{a}}_{MP}^{\mathsf{T}} \otimes \tilde{\mathbf{b}}_{MP}^{\mathsf{T}} \end{bmatrix} \text{vec}(\mathbf{X}) + \mathbf{w}. \tag{4.4}$$

With Sparse-Lift, MP measurements of this form are used to solve for the sparse matrix $\mathbf{X} = \mathbf{p}\mathbf{x}^{\mathsf{T}}$. Then, the singular value decomposition of the estimate $\hat{\mathbf{X}} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^*$ is computed. The estimate of the sparse channel, up to a global scaling, is then $\hat{\mathbf{x}} = \overline{\mathbf{v}}_1$, where $\overline{\mathbf{v}}_1$ is the conjugate of the singular vector corresponding to the largest singular value.

4.2.2 Computation Complexity

We use OMP to address the standard sparse recovery of $\text{vec}(\mathbf{X})$. The sparsity level of $\text{vec}(\mathbf{X})$ is PK. Sparse recovery of $\text{vec}(\mathbf{X})$ takes $\sum_{PK} \mathcal{O}(T(k))$ FLOPS, where $T(k)$ is the computation complexity of k_{th} iteration.

1. The first iteration step, similar to OMP, takes $\mathcal{O}(MNP^2)$ FLOPS.
2. Selecting the maximum over PN scalars takes $\mathcal{O}(PN\log(PN))$ FLOPS.
3. Pseudo inverse operation takes $\mathcal{O}((MP)^2k) + \mathcal{O}((MP)^3)$ FLOPS.
4. Update of residual takes $\mathcal{O}(MP) + \mathcal{O}(MPk^2)$ FLOPS.

After sparse recovery, the SVD takes $\mathcal{O}(P^2N)$ FLOPS.

As we know the sparsity of $\text{vec}(\mathbf{X}) \in \mathbb{C}^{PN\times 1}$ is PK, where K is the sparsity of $\mathbf{x}$. Compared to OMP which recovers a K-sparse vector, the sparse recovery of this higher dimension tensor definitely owns higher computation complexity. From the aspect of accuracy, we expect the recovered matrix $\hat{\mathbf{X}}$ is strictly rank-1, which is hard to achieve by standard CS algorithm, leading to degradation of the principle singular value during SVD step. This fact has impacts on the performance of Sparse-Lift.

4.3 Partially Coherent Compressive Phase Retrieval (PC-CPR)

The partially coherent CS model in [22] assumes that the coherent measurements are acquired using different radio frequency chains. Although we consider a single radio frequency chain, our mathematical model is identical to the one in [22]. The algorithm in [22] is a two-stage method. In the first stage, the support set of the sparse vector is found by taking indices corresponding to the K largest values of $\mathbf{z}$, where $z[k] = \sum_{p=1}^{P} |\mathbf{A}_p(:,k)^* \mathbf{y}_p|^2/M$. Then, the sparse estimate $\hat{\mathbf{x}}$ is initialized using the eigen-decomposition-based method in [27]. In the second stage, the algorithm iteratively estimates the phase errors and the sparse signal. A hard thresholding algorithm is used to estimate the sparse signal. In t^{th} iteration, the compensation of p^{th} packet can be written as

$$\mathbf{y}_p^{(t)} = \mathbf{y}_p^{(t-1)} e^{-\mathrm{j}\hat{\phi}_p^{(t)}}, \tag{4.5}$$

where the phase noise $\hat{\phi}_p$ is estimated by taking

$$\hat{\phi}_p(t) = \underset{\hat{\phi}_p}{\text{argmin}} \|\mathbf{y}_p^{(t-1)} - e^{\mathrm{j}\hat{\phi}_p} \mathbf{A}_{p,\Lambda} \hat{\mathbf{x}}^{(t-1)}\|_2^2. \tag{4.6}$$

Stack all the packets in $\mathbf{y}^{(t)}$, and the sparse channel estimation can be formed as a standard CS recovery:

$$\hat{\mathbf{x}}^{(t)} = \underset{\hat{\mathbf{x}}}{\text{argmin}} \left\|\mathbf{y}^{(t)} - \mathbf{A}\hat{\mathbf{x}}^{(t-1)}\right\|^2 \quad \text{s. t. } \|\hat{\mathbf{x}}\|_0 \leq K. \tag{4.7}$$

Then [22] solved it by a hard thresholding method.

The above steps iterate until the stop criteria is met. In [22], the stop criteria is defined as

$$\left\|\mathbf{y}^{(t)} - \mathbf{A}\hat{\mathbf{x}}^{(t)}\right\| < \epsilon,$$

where ϵ is the termination threshold.

4.3.1 Computation Complexity

1. Initialization of $\hat{\mathbf{x}}$ takes $\mathcal{O}(2MPN + N\log(N) + 2(MP)^2 + K^2(MP)^2 + k^3 + MPK + MP + K)$ FLOPS. The complexity mainly comes from multiplication of 3 matrices as $\mathcal{O}(K^2(MP)^2)$.

2. Iterative refinement takes $\mathcal{O}(\log(1/\epsilon) \cdot T(t))$ FLOPS, where ϵ is the stopping criteria, and $T(t)$ the complexity of each refinement.

3. Each refinement, including calibration of $\mathbf{y}$ and a thresholding step, takes $\mathcal{O}(P \cdot (M^2N^2 + MN^2) + MN + KN)$.

In conclusion, the complexity of PC-CPR is about $\mathcal{O}(PM^2N^2\log(1/\epsilon))$, mainly from step 2 and 3.

In this chapter, we presented the implementation details of OMP, Sparse-Lift, and PC-CPR, encompassing their algorithmic flow and computational complexities. Additionally, we conducted a preliminary analysis of their limitations. The following chapter will introduce our proposed algorithm. To facilitate a comprehensive comparison with these methods, we will showcase simulation results in Chapter 6.

Proposed Partially Coherent Matching Pursuit

5

5.1 Partially coherent matching pursuit (PC-MP)

Our PC-MP algorithm is a greedy approach that adds one element to the estimated support set in each iteration. Then, the algorithm estimates the phase errors and the sparse signal over the estimated support through alternating minimization. This procedure is carried out until the stopping criterion is met. In this project, we use the assumption in [22] that the number of non-zero coefficients in $\mathbf{x}$ is known. There are several methods for sparsity-level exploiting.

We first discuss our support detection rule in PC-MP. Our algorithm is initialized by setting Λ_0 to an empty set and $\hat{\mathbf{x}}_0$ to a zero vector. Here, we use Λ_t to denote the estimated support set of the sparse vector, $\hat{\mathbf{x}}_t$ as the estimate of $\mathbf{x}$, and $\hat{\phi}_{p,t}$ as the phase error estimated for packet p, after t iterations. The vector of estimated phase errors is denoted by $\hat{\boldsymbol{\phi}}_t$. For the p^{th} packet, we denote $\mathbf{r}_{p,t}$ as the residue error between the observed measurements, and initialize $\mathbf{r}_{p,0} = \mathbf{y}_p$. For the prediction in the t^{th} iteration, we subtract the components contributed by current $\hat{\mathbf{x}}_t$ as

$$\mathbf{r}_{p,t} = \mathbf{y}_p - e^{\mathrm{j}\hat{\phi}_{p,t}} \mathbf{A}_p \hat{\mathbf{x}}_t. \tag{5.1}$$

In the t^{th} iteration, matching pursuit algorithms in standard CS identify the column of $\mathbf{A}_p$ that results in the largest $|\mathbf{A}_p(:,k)^* \mathbf{r}_p^{(t-1)}|$, i.e., the absolute value of the correlation with the residue. In our problem, however, we have P different residues derived from P packets. As the measurements across the P packets are non-coherent, our algorithm sums up the absolute values of the correlations and selects the index that maximizes the summation. The new element added to the support set is

$$\hat{k}_t = \underset{k \in [N] \setminus \Lambda_{t-1}}{\operatorname{argmax}} \sum_{p=1}^{P} |\mathbf{A}_p(:,k)^* \mathbf{r}_{p,t-1}|, \tag{5.2}$$

and the augmented support set is $\Lambda_t = \Lambda_{t-1} \cup \hat{k}_t$. For the special case when $P = 1$, we observe that the objective in (5.2) becomes identical to that used in matching pursuit algorithms for standard CS. In the t^{th} iteration, our approach explicitly excludes support elements in Λ_{t-1} while the OMP inherently avoids selecting such

elements. This is because the residue in our approach may not be orthogonal to the selected columns of the CS matrix, unlike the OMP.

After the support estimation step, we use an alternating minimization approach to estimate the non-zero entries of $\mathbf{x}$ over the identified support, and the phase errors. Due to the different support detection schemes from the methods in [19,22], we do not need to initialize the phase error and signal vectors in a specific way, because the signal vector converges to the accurate one as support set updates.

As we acquired different measurement vectors for the same signal vector, it is crucial to recover one $\mathbf{x}$ over all the packets. The joint estimation of the signal and the phase error vector $\phi = [e^{\mathrm{j}\phi_1}, ..., e^{\mathrm{j}\phi_P}]$ can be formed as an optimization problem by minimizing the summation of LS errors between exact $\mathbf{y}_p$ and the reconstructed measurements, i.e.

$$\hat{\mathbf{x}}_{\Lambda_t}, \hat{\boldsymbol{\phi}}_t = \underset{\mathbf{z}_{\Lambda_t}, \boldsymbol{\delta}}{\operatorname{argmin}} \sum_{p=1}^{P} \left\| \mathbf{y}_p - e^{\mathrm{j}\delta_p} \mathbf{A}_{p,\Lambda_t} \mathbf{z}_{\Lambda_t} \right\|^2, \tag{5.3}$$

where $\mathbf{A}_{p,\Lambda_t}$ is a submatrix of $\mathbf{A}_p$ obtained by retaining only those columns with indices in Λ_t. We observe that (5.3) is a non-convex problem. It is, however, a standard least squares problem in $\mathbf{z}_{\Lambda_t}$ for a fixed $\boldsymbol{\delta}$. Furthermore, a closed form solution for $\boldsymbol{\delta}$ can be obtained for a fixed $\mathbf{z}_{\Lambda_t}$. Our alternating minimization procedure leverages both of these properties.

We now provide closed-form expressions for phase recovery and signal estimation in the ℓ^{th} iteration of alternating minimization in (5.3). For a fixed $\mathbf{z}_{\Lambda_t}^{(\ell-1)}$ in (5.3), we observe that the minimization problem is separable in $\{\delta_p\}_{p=1}^{P}$. The phase recovery problem for the p^{th} packet is then

$$\begin{aligned} \hat{\phi}_{p,t}^{(\ell)} &= \underset{\delta_p}{\operatorname{argmin}} \, \| \mathbf{y}_p - e^{\mathrm{j}\delta_p} \mathbf{A}_{p,\Lambda_t} \mathbf{z}_{\Lambda_t}^{(\ell-1)} \|_2 \\ &= \underset{\delta_p}{\operatorname{argmax}} \, \mathcal{R}\{ e^{\mathrm{j}\delta_p} \mathbf{y}_p^* \mathbf{A}_{p,\Lambda_t} \mathbf{z}_{\Lambda_t}^{(\ell-1)} \} \\ &= -\text{phase}\left(\mathbf{y}_p^* \mathbf{A}_{p,\Lambda_t} \mathbf{z}_{\Lambda_t}^{(\ell-1)} \right). \end{aligned} \tag{5.4}$$

After estimating $\hat{\phi}_{p,t}^{(\ell)}$ for each p, the signal in (5.3) is estimated by solving a convex least squares problem. Let the gradient of the objective

$$\begin{aligned} &2 \textstyle\sum_{p=1}^{P} \left(\mathbf{A}_{p,\Lambda_t}^* \mathbf{A}_{p,\Lambda_t} \hat{\mathbf{x}}_{\Lambda_t}^{(\ell)} - e^{-\mathrm{j}\hat{\phi}_{p,t}^{(\ell)}} \mathbf{A}_{p,\Lambda_t}^* \mathbf{y}_p \right) \\ &= 2 \left(\textstyle\sum_{p=1}^{P} \mathbf{A}_{p,\Lambda_t}^* \mathbf{A}_{p,\Lambda_t} \right) \hat{\mathbf{x}}_{\Lambda_t}^{(\ell)} - 2 \left(\textstyle\sum_{p=1}^{P} e^{-\mathrm{j}\hat{\phi}_{p,t}^{(\ell)}} \mathbf{A}_{p,\Lambda_t}^* \mathbf{y}_p \right), \\ &= 0 \end{aligned}$$

the solution then is given by

$$\hat{\mathbf{x}}_{\Lambda_t}^{(\ell)} = \left(\sum_{p=1}^{P} \mathbf{A}_{p,\Lambda_t}^{*} \mathbf{A}_{p,\Lambda_t} \right)^{\dagger} \left(\sum_{p=1}^{P} e^{-\mathrm{j}\hat{\phi}_{p,t}^{(\ell)}} \mathbf{A}_{p,\Lambda_t}^{*} \mathbf{y}_p \right). \tag{5.5}$$

After the alternating minimization procedure converges in L iterations, our method sets $\hat{\mathbf{x}}_{\Lambda_t} = \hat{\mathbf{x}}_{\Lambda_t}^{(L)}$ and $\hat{\boldsymbol{\phi}}_t = \hat{\phi}_{p,t}^{(L)}$. The subsequent PC-MP step updates the residue according to (5.1) and then identifies the next element of the support. A summary of our PC-MP technique is provided in Algorithm 1.

Algorithm 1 Proposed PC-MP algorithm for sparse recovery

Input: Partially coherent measurements $\{\mathbf{y}_p\}_{p=1}^{P}$, CS matrices $\{\mathbf{A}_p\}_{p=1}^{P}$, sparsity level K

Initilize: $\mathbf{r}_{p,0} = \mathbf{y}_p, \hat{\mathbf{x}}_0 = \mathbf{0}, \Lambda_0 = \emptyset$

1: **for** $t = 1, 2, \ldots, K$ **do**

#Support detection:

2: $\hat{k}_t = \underset{k \in [N] \setminus \Lambda_{t-1}}{\operatorname{argmax}} \sum_{p=1}^{P} |\mathbf{A}_p(:,k)^{*} \mathbf{r}_{p,t-1}|$

3: $\Lambda_t = \Lambda_{t-1} \cup \hat{k}_t$

4: *#Signal estimation:*

5: **while** $\left| \hat{\mathbf{x}}_{\Lambda_t}^{(l)} - \hat{\mathbf{x}}_{\Lambda_t}^{(l-1)} \right| > \epsilon$ or $\ell < L$ **do**

6: $\hat{\phi}_{p,t}^{(l)} = -\operatorname{phase}\left(\mathbf{y}_p^{H} \mathbf{A}_{p,\Lambda} \hat{\mathbf{x}}_{\Lambda_t}^{(l-1)} \right) \ \forall p$

7: $\hat{\mathbf{x}}_{\Lambda_t}^{(l)} = \left[\sum_{p=1}^{P} \mathbf{A}_{p,\Lambda_t}^{*} \mathbf{A}_{p,\Lambda_t} \right]^{\dagger} \left[\sum_{p=1}^{P} e^{-\mathrm{j}\phi_{p,t}^{(l)}} \mathbf{A}_{p,\Lambda_t}^{*} \mathbf{y}_p \right]$

8: **end while**

9: $\mathbf{r}_{p,t} = \mathbf{y}_p - e^{\mathrm{j}\hat{\phi}_{p,t}^{(l)}} \mathbf{A}_{p,\Lambda_t} \hat{\mathbf{x}}_{\Lambda_t} \ \forall p$

10: **end for**

11: **Output:** $\hat{\mathbf{x}}$

5.2 Performance Guarantees

5.2.1 Definition of Mutual Coherence

In compressive sensing, the concept of mutual coherence quantifies the quality of sparse sampling and plays a crucial role in the success of sparse signal recovery algorithms. Mutual coherence measures how correlated the columns of the measurement matrix (here denoted as $\mathbf{A}$) are in terms of inner products. To achive a good recovery, we expect that for different sparse vectors $\mathbf{x}$, there would be distinct product results of $\mathbf{Ax}$. For a given measurement matrix

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N]$$

with normalized columns

$$\|\mathbf{a}_i\|_2 = 1$$

for each column $\mathbf{a}_i$, the mutual coherence is defined as the maximum absolute inner product between any two distinct columns of $\mathbf{A}$, i.e.

$$\mu(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i^{\mathsf{T}} \mathbf{a}_j|.$$

A lower mutual coherence value indicates that the columns of the matrix $\mathbf{A}$ are less correlated, making it easier to distinguish between the sparse signals during the reconstruction process. In other words, a measurement matrix with lower mutual coherence provides a higher-quality sparse sampling, allowing for more accurate recovery of sparse signals [28].

5.2.2 Mutual Coherence-Based Support Detection Guarantees

Now, we derive a mutual coherence-based guarantee to successfully identify one element of the true support set of $\mathbf{x}$ with PC-MP for real-valued CS matrices.

Proposition 1. *Consider the channel estimation problem in* (3.19) *using measurements from P packets. Assume that the columns in real-value CS matrix $\mathbf{A}_p$ are normalized as $\|\mathbf{A}_p(:,i)\|_2 = 1$. Given a constant $\beta > 0$, the support entry identified in the first iteration is correct with probability exceeding*

$$1 - \frac{1}{N^{\beta} P^{\beta} \sqrt{\pi(1+\beta)\log(NP)}} \,. \tag{5.6}$$

under the condition

$$|x_{\max}| \left(1 - \frac{(2K-1)}{P} \sum_{p=1}^{P} \mu_p\right) > 2\sigma \sqrt{2(1+\beta)\log(NP)} \,, \tag{5.7}$$

where $|x_{max}| = \max_{i \in \Lambda} \{|x_i|, x_i \in \mathbf{x}\}$, μ_p is the mutual coherence of $\mathbf{A}_p$, σ is the standard deviation of Gaussian additive noise $\mathbf{w}_p$, and K is the sparsity level of $\mathbf{x}$.

Proof. We extend the support identification guarantee in [28] to the partially coherent case. Let Λ be the true support of the sparse vector $\mathbf{x}$. Our proof verifies that when (5.7) holds,

$$\max_{i \in \Lambda} \sum_{p=1}^{P} \left|\mathbf{A}_p(:,i)^{\mathsf{T}} \mathbf{y}_p\right| > \max_{i \notin \Lambda} \sum_{p=1}^{P} \left|\mathbf{A}_p(:,i)^{\mathsf{T}} \mathbf{y}_p\right| , \tag{5.8}$$

which matches our detection rule to successfully identify one element of the support. To this end, we show that (5.8) holds under $\mathcal{D}$, where

$$\mathcal{D} = \left\{ \max_{1\le i\le N} \max_{1\le p\le P} |\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p| < \zeta \right\}, \tag{5.9}$$

where

$$\zeta = \sigma\sqrt{2(1+\beta)\log N}$$

controls the success probability of event $\mathcal{D}$. As $\left\{\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p\right\}_{i,p}$ is jointly Gaussian, with [29, theorem 1] we simplify

$$\begin{aligned}
\Pr\{\mathcal{D}\} &= \Pr\left\{ \max_{i,p} |\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p| < \zeta \right\} \\
&= \Pr\left\{ \max_{p} |\mathbf{A}_p(:,1)^{\mathsf{T}}\mathbf{w}_p| < \zeta, ..., \max_{p} |\mathbf{A}_p(:,N)^{\mathsf{T}}\mathbf{w}_p| < \zeta \right\} \\
&\ge \prod_{i=1}^{N} \Pr\left\{ \max_{p} |\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p| < \zeta \right\} \\
&\ge \prod_{i=1}^{N}\prod_{p=1}^{P} \Pr\left\{ |\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p| < \zeta \right\} \\
&= \left[1 - 2Q\left(\frac{\zeta}{\sigma}\right) \right]^{NP},
\end{aligned}$$

where

$$Q(x) = (\frac{1}{\sqrt{2\pi}}) \int_x^{\infty} e^{-z^2/2} dz$$

is the Gaussian tail probability.

Theorem 1. [29] *Collect a set of random variables* $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ *with normal distribution with zero means and arbitrary variances. For any positive numbers* $(c_1, c_2, \ldots, c_k)$*,*

$$\begin{aligned}
&\Pr\{|X_1| \le c_1, |X_2| \le c_2, \ldots, |X_k| \le c_k\} \\
&\quad \ge \Pr\{|X_1| \le c_1\} \cdot \Pr\{|X_2| \le c_2, \ldots, |X_k| \le c_k\} \\
&\quad \ge \prod_{i=1}^{k} \Pr\{|X_i| \ge c_i\}.
\end{aligned}$$

Further, since the Gaussian tail probability is bounded by

$$Q(x) \le \frac{1}{x\sqrt{2\pi}} e^{-x^2/2},$$

we obtain

$$\begin{aligned}\Pr\{\mathcal{D}\} &\geq \left(1 - \sqrt{\frac{2}{\pi}}\frac{\sigma}{\zeta}e^{\frac{-\zeta^2}{2\sigma^2}}\right)^{NP} \\ &\geq 1 - NP\sqrt{\frac{2}{\pi}}\frac{\sigma}{\zeta}e^{\frac{-\zeta^2}{2\sigma^2}}.\end{aligned}$$

The second step is according to Taylor expansion. Substituting

$$\zeta = \sigma\sqrt{2(1+\beta)\log(NP)}$$

in the above relation shows that the event $\mathcal{D}$ occurs with probability exceeding (5.6).

Once the event $\mathcal{D}$ holds for $p = 1, 2, \ldots, P$, we can express the left-hand side of (5.8) as

$$\begin{aligned}&\max_{i\notin\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{y}_p\right| \\ &\quad= \max_{i\notin\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\left(\sum_{k\in\Lambda}e^{\mathsf{j}\phi_p}\mathbf{A}_p(:,k)x_k + \mathbf{w}_p\right)\right| \\ &\quad= \max_{i\notin\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p + \sum_{k\in\Lambda}e^{\mathsf{j}\phi_p}\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{A}_p(:,k)x_k\right| \\ &\quad\leq \max_{i\notin\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p\right| + \max_{i\notin\Lambda}\sum_{p=1}^{P}\sum_{k\in\Lambda}\left|e^{\mathsf{j}\phi_p}\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{A}_p(:,k)x_k\right| \\ &\quad\leq P\zeta + K\sum_{p=1}^{P}\mu_p\left|x_{\max}\right|, \qquad (5.10)\end{aligned}$$

where the last step follows from (5.9) and the definition of $|x_{\max}|$ is

$$|x_{\max}| = \max_{i\in\Lambda}\{|x_i|, x_i \in \mathbf{x},\}.$$

For a sparsity level $K \geq 1$, using (3.19) and column normalization assumption on

$\mathbf{A}_p$, we can bound the right-hand side of (5.8) as

$$\begin{aligned}
&\max_{i\in\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{y}_p\right| \\
&\quad=\max_{i\in\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\left(\sum_{k\in\Lambda}e^{\mathrm{j}\phi_p}\mathbf{A}_p(:,k)x_k+\mathbf{w}_p\right)\right| \\
&\quad=\max_{i\in\Lambda}\sum_{p=1}^{P}\left|x_i e^{j\phi_p}+\sum_{k\in\Lambda\setminus\{i\}}e^{\mathrm{j}\phi_p}\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{A}_p(:,k)x_k+\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p\right| \\
&\quad\geq\max_{i\in\Lambda}\sum_{p=1}^{P}|x_i|-\sum_{p=1}^{P}\left|\sum_{k\in\Lambda\setminus\{i\}}e^{\mathrm{j}\phi_p}\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{A}_p(:,k)x_k+\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{w}_p\right|. \qquad (5.11)
\end{aligned}$$

Here, from the definition of $x_{\max}$ and μ_p, we get

$$\begin{aligned}
\max_{i\in\Lambda}\sum_{p=1}^{P}\left|\mathbf{A}_p(:,i)^{\mathsf{T}}\mathbf{y}_p\right|&\geq P|x_{\max}|-\sum_{p=1}^{P}\left[\sum_{k\in\Lambda/i}\mu_p|x_{max}|+\zeta\right] \\
&\geq P|x_{\max}|-P\zeta-(K-1)\sum_{p=1}^{P}\mu_p|x_{\max}|. \qquad (5.12)
\end{aligned}$$

In the first iteration of PC-MP, support identification is successful when (5.8) holds. We observe that the condition in (5.8) holds if the lower bound in (5.12) exceeds the upper bound in (5.10), which is equivalent to

$$P|x_{\max}|-2P\zeta-(2K-1)\sum_{p=1}^{P}\mu_p\,|x_{\max}|>0, \qquad (5.13)$$

which is the condition stated in (5.7). □

Our guarantees are limited to identifying only one element from the support set, not the entire support. This limitation arises due to inaccuracies in phase error estimation during each iteration, leading to residual errors. These errors might not be orthogonal to the previously selected columns, making it challenging to apply an induction-based argument. Intuitively, the result shows that support detection can only be successful under the assumption that the maximum absolute entry of $\mathbf{x}$ is "larger" than the additive noise level as described by (5.7). Furthermore, it establishes that a sufficient condition for the faithful identification of the first

element of the support is given by the inequality:

$$\frac{1}{P}\sum_{p=1}^{P}\mu_p < 1/(2K-1).$$

Additionally, it is evident that a decrease in mutual coherence makes it easier to meet the condition (5.7). Therefore, selecting compressive sensing matrices for PC-MP with the lowest average mutual coherence, i.e. $\sum_{p=1}^{P}\mu_p/P$, proves to be a wise choice. Knowing the complexity helps in allocating computational resources effectively. For instance, in real-time systems or resource-constrained environments, choosing algorithms with lower complexity can ensure optimal performance.

5.3 Computation Complexity

In this section, we are going to discuss the complexity of PC-MP in each step. Iterating K times where K is the sparsity level takes $\sum_K \mathcal{O}(T(k))$ FLOPS, where $T(k)$ is the computation complexity of k_{th} iteration.

1. Product of $\mathbf{A}_p(:,k)^*\mathbf{r}_{p,t-1}$ takes $\mathcal{O}(MN)$ FLOPS, Summing P scalars takes $\mathcal{O}(P)$ FLOPS. Selecting the maximum over N scalars takes $\mathcal{O}(N\log(N))$ FLOPS.
2. The state takes $\mathcal{O}(1)$ FLOPS.
3. Product of $\mathbf{y}_p^H\mathbf{A}_{p,\Lambda}\hat{\mathbf{x}}_{\Lambda_t}^{(l-1)}$ takes $\mathcal{O}(Mk)$ FLOPS.
4. Product of $\mathbf{A}_{p,\Lambda_t}^*\mathbf{A}_{p,\Lambda_t}$ takes $\mathcal{O}(M^2k)$ FLOPS, the summation of vectors over P packets takes $\mathcal{O}(Pk^2)$ FLOPS, and the inverse takes $\mathcal{O}(M^3)$ FLOPS.
5. Product of $e^{-\mathrm{j}\phi_{p,t}^{(l)}}\mathbf{A}_{p,\Lambda_t}^*\mathbf{y}_p$ takes $\mathcal{O}(Mk)$ FLOPS, summing of vectors over P packets takes $\mathcal{O}(Pk)$ FLOPS.

In the worst case, step 2$\sim$5 takes $\sum_L \mathcal{O}(T(\ell))$ FLOPS, where $T(\ell)$ is the computation complexity of the ℓ_{th} alternating minimization step. The main complexity comes from the matrix inverse operation, thus the complexity of PC-MP is $\mathcal{O}(KLM^3)$. From the analysis result, we can conclude that a small size of the training packet can decrease the computation complexity.

In this chapter, we systematically derived the PC-MP algorithm, examined the conditions guaranteeing successful detection of the first support, and analyzed the computational complexity of the algorithm. The subsequent chapter will present simulation results, comparing PC-MP with benchmark algorithms.

6 Simulations & Results

In this chapter, we compare the performance of our proposed method with OMP, self-calibration-based CS called *Sparse-Lift* [20], and partially coherent CS (PC-CPR) [22], considering the system model described in section Chapter 3.

6.1 System Parameters

We first generate the measurements of the multi-path channel according to (3.18) assuming partially coherent measurements, then generate data according to (3.15) with practical phase noise. The parameters are set as:

- Set a uniform linear array of size $N = 256$ and the same number of on-grid paths in the channel.
- The number of acquired spatial channel measurements in each packet is $M = 16$.
- The DoA are assumed to be uniformly distributed in $[-\pi,\ \pi)$.
- The complex gain is assumed to follow circularly symmetrix complex Gaussian distribution, i.e. $h_k \sim \mathcal{CN}(0, 1)$.
- The standard deviation of phase noise is given as $\tau = 2\pi f_c\sqrt{cT_s}$ [2], with $f_c = 60$ GHz as the carrier frequency, $c = 4.7 \times 10^{-18}$ dependent on the oscillator, and $T_s = 128$ ns as the duration time of a single pilot signal.

In our simulation, we treat the channel vector $\mathbf{x}$ as exactly sparse, by randomly activating some of the paths and normalizing the complex path gains.

6.2 Metrics Used to Evaluate Algorithms

For the performance evaluation, we use the achievable rate and normalized mean square error (NMSE) as metrics and analyze how they vary with the signal-to-noise ratio (SNR), the number of measurements, and the channel sparsity. We define the SNR as $10\log_{10}(1/\sigma^2)$.

- **Normalized Mean Squared Error**

As the measurements in the partially coherent model are perturbed in phase, the algorithms can estimate the channel only upto a global phase. We define the normalized mean squared error as

$$\text{NMSE} = \frac{\|e^{\mathrm{j}\delta}\mathbf{h}_{\text{est}} - \mathbf{h}\|_2^2]}{\mathbb{E}[\|\mathbf{h}\|_2^2]},$$

where

$$\delta = \underset{\delta}{\text{argmin}}\|e^{\mathrm{j}\delta}\mathbf{h}_{\text{est}} - \mathbf{h}\|_2^2.$$

- **Achievable Rate**

The achievable rate for the channel is defined as

$$R = \log_2\left(1 + |\mathbf{f}_{\text{est}}^*\mathbf{h}|^2/\sigma^2\right).$$

$\mathbf{f}_{\text{est}}$ is set to satisfy

$$\mathbf{f}_{\text{est}} = \underset{\mathbf{f}}{\text{argmax}}|\mathbf{f}^*\mathbf{h}|^2$$
$$\text{s.t. } \|\mathbf{f}_{\text{est}}\|_2 = 1,$$

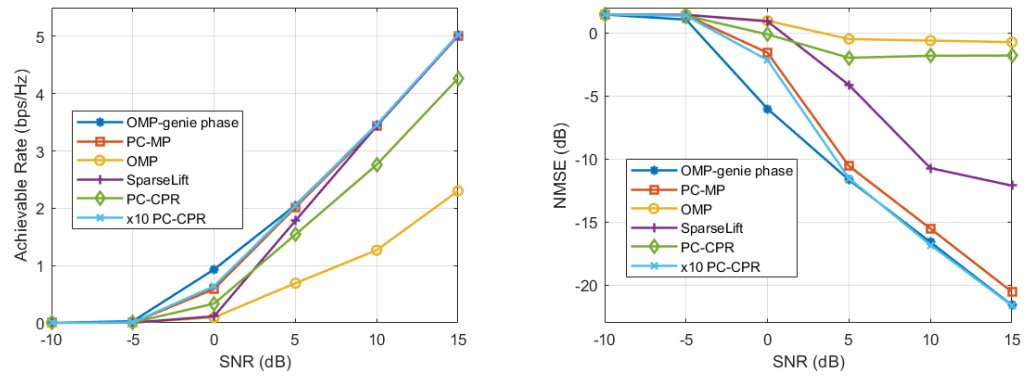
which is a unit-norm conjugate beamformer, i.e.

$$\mathbf{f}_{\text{est}} = \overline{\mathbf{h}}_{\text{est}}/\|\mathbf{h}_{\text{est}}\|_2.$$

6.3 Results and Discussion

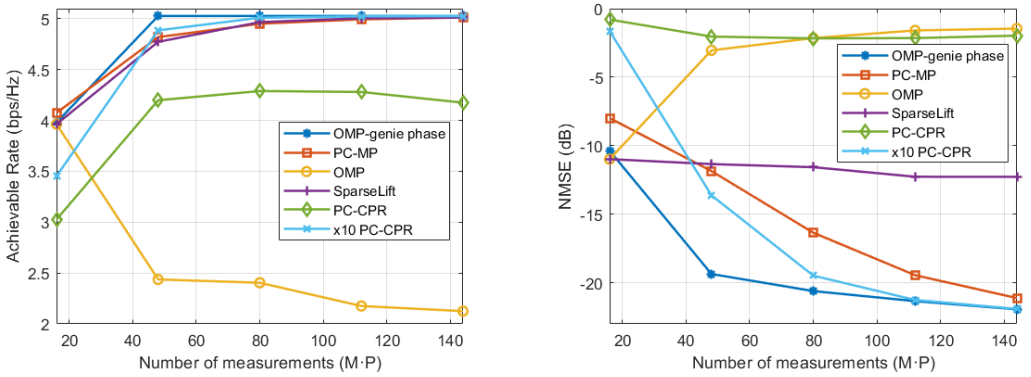
To evaluate our algorithm, we initially simulate and discuss algorithms operating on the partially coherent model, assuming that the phase noise is constant within a packet. However, as discussed in Section 1.4 and Section 3.4, practical phase noise also fluctuates within a packet. In this section, we further simulate these algorithms under a practical phase noise setting to test the robustness of the proposed approach to variations in phase noise within each packet. Subsequently, we conduct simulations in scenarios where the channel vector $\mathbf{x}$ is approximately sparse.

6.3.1 Simulations when phase noise is constant within each packet



(a) (b)

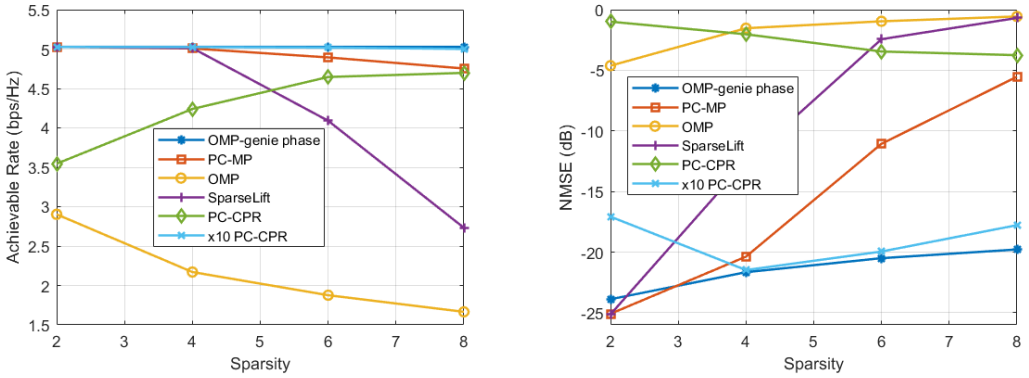
Figure 6.1: Simulation results with perfect structured phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and the total number of measurements $MP = 128$. (a) Achievable rate with SNR. (b) NMSE in the channel estimate with SNR. The number of iterations used for PC-CPR is as same as that used for PC-MP, while $\times 10$ PC-CPR uses $10\times$ more iterations than PC-MP.



(a) (b)

Figure 6.2: Simulation results with perfect structured phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and SNR = 15 dB. (a) Achievable rate with numbers of measurements. (b) NMSE in the channel estimate with numbers of measurements.

In this subsection, simulations are done with a perfectly structured phase noise matrix as (3.17). We compare the average achievable rate and NMSE versus SNR in Figure 6.1 for SNR range from -10dB to 15dB, and versus the total number



(a) (b)

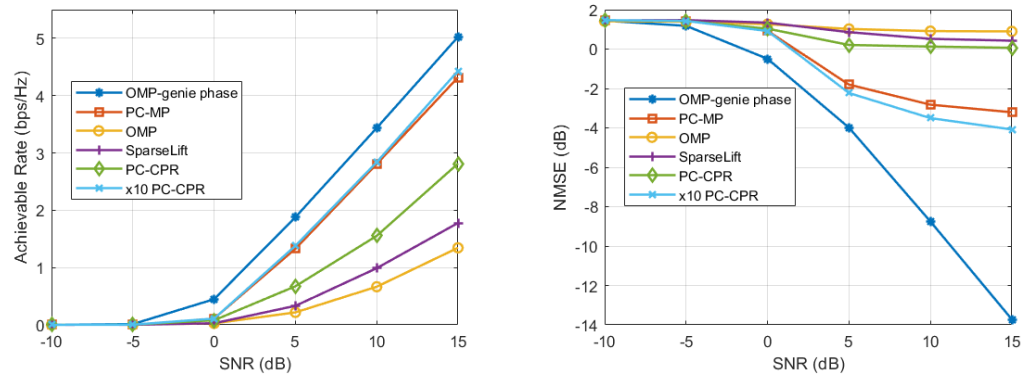
Figure 6.3: Simulation results with perfect structured phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, with SNR $= 15$ dB, and the total number of measurements $MP = 128$. (a) Achievable rate with channel sparsity. (b) NMSE in the channel estimate with channel sparsity.

of measurements in Figure 6.2 for the number of measurements range from 16 to 144, which is equivalent to 1 to 9 packets. The plots in these two figures demonstrate that improved performance is observed with increasing SNR and the number of acquired measurements across all approaches. Conversely, as depicted in Figure 6.3, when the sparsity level of the channel increases, estimating channel state information under phase noise becomes more challenging.

Specifically, our observations reveal that standard OMP performs well when provided with known phase errors (genie phase), but fails in the absence of phase information. With the same number of iterations, the proposed PC-MP consistently outperforms PC-CPR. When over $\times 10$ internal iterations can PC-CPR achieve performances comparable to OMP with genie phase. Moreover, Sparse-Lift has comparable performance to PC-MP and $\times 10$ PC-CPR in Figure 6.1(a) and Figure 6.2(a). However, the impact of increasing sparsity on Sparse-Lift is much more significant compared to other methods. This implies that under complex phase noise conditions, the sparse recovery of high-dimensional tensors is not strictly rank-1. Consequently, exploiting the original sparse vector for the principle vector becomes more challenging.

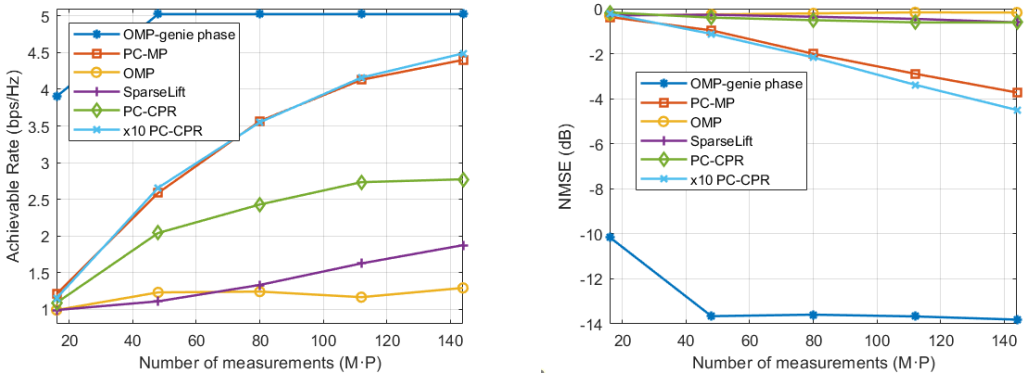
However, in practice, the phase noise does not strictly adhere to the structure defined in (3.17). The phase noise fluctuates within a packet. Therefore, we will present simulation results considering realistic phase noise conditions.

6.3.2 Simulations when phase noise varies within each pachet



(a) (b)

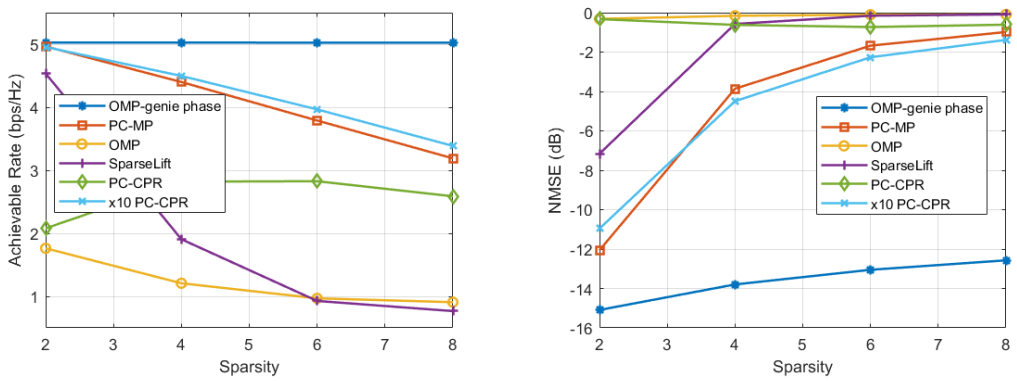
Figure 6.4: Simulation results with practical phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and the total number of measurements $MP = 128$. (a) Achievable rate with SNR. (b) NMSE in the channel estimate with SNR. The number of iterations used for PC-CPR is as same as that used for PC-MP, while $\times 10$ PC-CPR uses $10\times$ more iterations than PC-MP.



(a) (b)

Figure 6.5: Simulation results with practical phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and SNR $= 15$ dB. (a) Achievable rate with numbers of measurements. (b) NMSE in the channel estimate with numbers of measurements.

In this subsection, simulations are conducted with practical phase noise, and the comparison among the approaches is made using the same metrics outlined in Section 6.3.1. Figure 6.4, Figure 6.5, and Figure 6.6 demonstrate trends similar to



(a) (b)

Figure 6.6: Simulation results with practical phase noise that compare PC-MP against the benchmarks for $N = 256$ antennas at the TX, with SNR $= 15$ dB, and the total number of measurements $MP = 128$. (a) Achievable rate with channel sparsity. (b) NMSE in the channel estimate with channel sparsity.

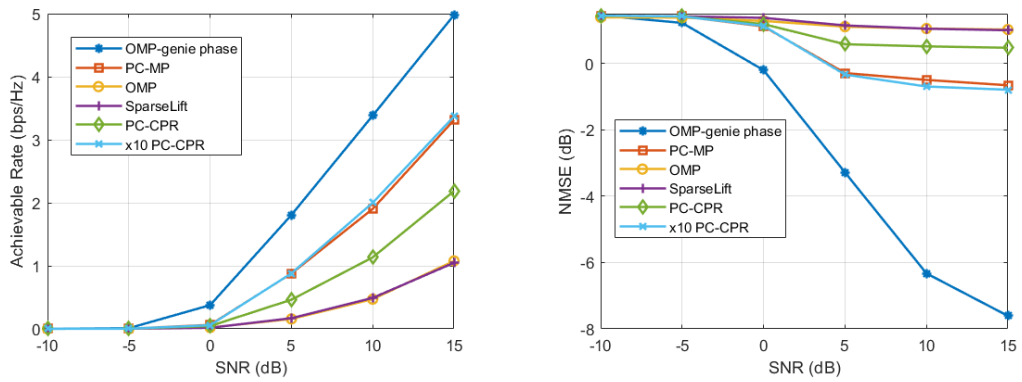
those observed in Section 6.3.1. The fluctuating nature of phase noise significantly impacts the performance of sparse recovery algorithms. PC-MP and PC-CPR exhibit greater robustness to phase noise compared to Sparse-Lift, which proves to be highly sensitive to these fluctuations. Notably, Sparse-Lift displays poor performance in this scenario, while the PC-MP algorithm consistently outperforms Sparse-Lift in terms of NMSE and achievable rate. This improvement can be attributed to our algorithm's ability to exploit the constant magnitude structure in the calibration vector, a feature not utilized by Sparse-Lift. Moreover, the introduction of fluctuating phase noise disrupts the recovery of the rank-1 structure of the joint sparse matrix in Sparse-Lift, rendering SVD-based separation ineffective.

Furthermore, we observe that the proposed PC-MP algorithm outperforms PC-CPR for the same K iterations. This is likely due to the nature of the algorithms, i.e. PC-CPR is based on hard thresholding while our approach is based on matching pursuit. Both these algorithms assume a known sparsity level of K. We would like to mention that the performance of PC-CPR for a large number of iterations ($\times 10$ PC-CPR) is close to our PC-MP method for K iterations.

6.3.3 Simulations when the channel vector is approximately sparse

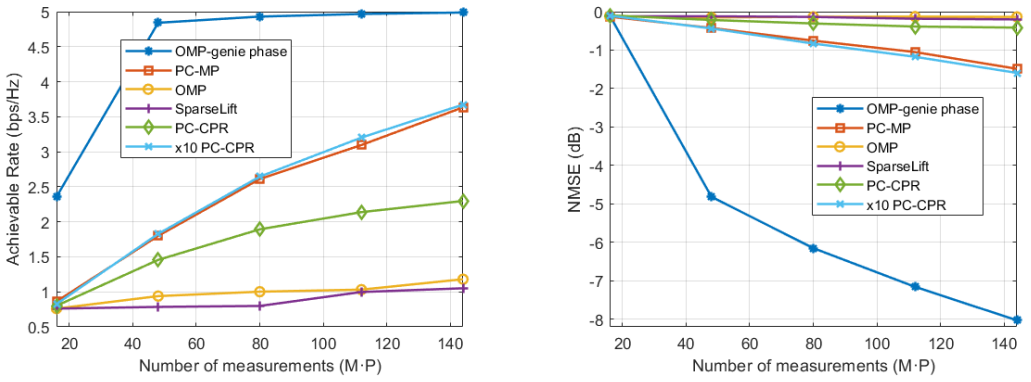
The four algorithms simulated in this project are designed under the assumption of an on-grid channel model. In practice, however, the channel is not on-grid due to the fact that the angles of departure may not exactly align with those in (3.13). The resulting mismatch introduces energy leakage, causing the channel vector $\mathbf{x}$ to

be not strictly sparse. Previous simulations were conducted assuming a perfectly sparse $\mathbf{x}$. Here, we further compare PC-MP against other methods when $\mathbf{x}$ is approximately sparse.



(a) (b)

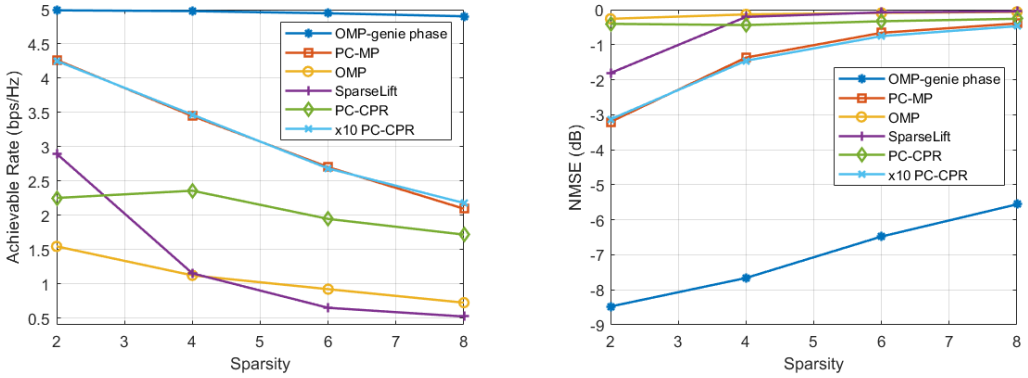
Figure 6.7: Simulation results that compare PC-MP against the benchmarks with approximated sparse channel and practical phase noise for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and the total number of measurements $MP = 128$. (a) Achievable rate with SNR. (b) NMSE in the channel estimate with SNR. The number of iterations used for PC-CPR is as same as that used for PC-MP, while $\times 10$ PC-CPR uses $10\times$ more iterations than PC-MP.



(a) (b)

Figure 6.8: Simulation results that compare PC-MP against the benchmarks with approximated sparse channel and practical phase noise for $N = 256$ antennas at the TX, $K = 4$ sparse channels, and SNR $= 15$ dB. (a) Achievable rate with numbers of measurements. (b) NMSE in the channel estimate with numbers of measurements.

Figure 6.7, Figure 6.8 and Figure 6.9 show a wider gap between OMP with genie phase and other algorithms compared to previous simulations. Notably, Sparse-



(a) (b)

Figure 6.9: Simulation results that compare PC-MP against the benchmarks with approximated sparse channel and practical phase noise for $N = 256$ antennas at the TX, with SNR $= 15$ dB, and the total number of measurements $MP = 128$. (a) Achievable rate with channel sparsity. (b) NMSE in the channel estimate with channel sparsity.

Lift performs even worse than OMP with phase noise in this scenario. This performance degradation might be attributed to the increased challenge for Sparse-Lift in maintaining the approximated rank-1 structure during the recovery of the approximately sparse matrix. Meanwhile, the trends for other algorithms remain consistent with those observed in previous results.

6.3.4 Execution Time

Table 6.1 shows the execution time of OMP, Sparse-Lift, PC-MP, $\times 10$ PC-CPR over 5000 averaging.

Methods	OMP	Sparse-Lift	PC-MP	$\times 10$ PC-CPR
Averaging operation time (ms)	0.18	8.55	5.79	23.96

Table 6.1: The execution time of PC-MP and the benchmarks for $N = 256$ antennas at the TX, with SNR $= 15$ dB, $K = 4$, and the total number of measurements $MP = 128$. Timing in MATLAB on a desktop computer with CPU i7-11800H @ 2.30GHz.

OMP stands out as a rapid algorithm with minimal execution time. Building upon the principles of matching pursuit, our proposed PC-MP method boasts lower complexity in comparison to PC-CPR. One contributing factor to this efficiency is evident in the alternating estimation step, where PC-MP consistently

reaches an analytical solution, while PC-CPR relies on a reweighted gradient descent approach, requiring more iterations for convergence. Notably, our observations indicate that Sparse-Lift demands significantly more computational time than our approach. This disparity arises from Sparse-Lift's need to solve for a high-dimensional lifted vector with NP variables, whereas our method focuses on solving for $N + P$ variables. In summary, PC-MP achieves accurate channel estimates with reduced computational complexity when compared to benchmark methods.

7 Conclusions & Future Works

7.1 Conclusions

In this project, we introduced a greedy algorithm called PC-MP for mmWave channel estimation in the presence of phase noise. Our approach makes use of the partially coherent structure in the phase perturbed measurements to perform sparse recovery and phase error estimation. PC-MP performs support detection by considering measurements from different packets and iteratively estimates the sparse vector through alternating minimization.

As discussed in Chapter 1, mmWave systems offer high data rate transmission capabilities due to their wide frequency bands. The high scattering at mmWave frequency leads to a sparse representation of the multi-path channel, making compressive channel estimation a viable approach for beam training. However, at high carrier frequencies, inherent hardware impairments such as CFO and phase noise become more significant. These impairments perturb the measurements of the wireless channel, posing challenges in beam training.

We have shown that standard CS algorithms fail to identify and estimate the sparse channel when practical phase noise is present. Previous works treated phase noise as a completely random process and focused on estimating the channel and the phase noise by formulating a CS-phase retrieval problem with phaseless measurements. However, considering the statistical distribution and the beam training protocol, a partially coherent model of measurements emerged.

Addressing the need for low complexity in fast-changing communication environments, our algorithm is built upon matching pursuit, a fast iterative process comprising two fundamental stages: support detection and signal estimation. In the support detection step, we utilize packets from different acquisitions to jointly detect one of the supports. In the signal estimation step, alternating optimization is employed to estimate the sparse channel and the phase noise. Furthermore, we derived a preliminary bound that ensures the successful identification of one element from the support set.

Through a series of simulations, comparisons among OMP, Sparse-Lift, and the proposed PC-MP have demonstrated that PC-MP outperforms the others in mmWave channel estimation. It achieves higher achievable rates and lower NMSE.

For the same number of iterations, PC-MP surpasses PC-CPR. Additionally, PC-MP exhibits lower computational complexity compared to Sparse-Lift and PC-CPR, highlighting the efficiency of this proposed algorithm.

7.2 Future Work

In our project, our focus has been on the on-grid channel model, although in real-world scenarios, the model is off-grid. Our future work will involve extending our algorithm to accommodate off-grid effects. Additionally, because of the global phase estimation, extending the guarantee for the first support detection to the entire support set is not straightforward. We still need to conduct a detailed analysis to determine how effectively PC-MP can identify the complete support of the sparse vector.

Bibliography

[1] "IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std*, pp. 1–4379, 2021.

[2] R. Zhang, B. Shim, and H. Zhao, "Downlink compressive channel estimation with phase noise in massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5534–5548, 2020.

[3] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 ghz communication for multi-gigabit-per-second Wi-Fi [invited paper]," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, 2014.

[4] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, and G. Chen, "Millimeter wave communication: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018.

[5] Z. Xiao, T. He, P. Xia, and X.-G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3380–3392, 2016.

[6] O. Abari, H. Hassanieh, M. Rodriguez, and D. Katabi, "Millimeter wave communications: From point-to-point links to agile network connections," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, HotNets '16, (New York, NY, USA), p. 169–175, Association for Computing Machinery, 2016.

[7] G. Wang, Q. Liu, R. He, F. Gao, and C. Tellambura, "Acquisition of channel state information in heterogeneous cloud radio access networks: challenges and research directions," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 100–107, 2015.

[8] I. Barhumi, G. Leus, and M. Moonen, "Optimal training design for mimo ofdm systems in mobile wireless channels," *IEEE Transactions on Signal Processing*, vol. 51, no. 6, pp. 1615–1624, 2003.

[9] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive millimeter-wave sector selection in off-the-shelf ieee 802.11ad devices," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, (New York, NY, USA), p. 414–425, Association for Computing Machinery, 2017.

[10] B. K. J. Al-Shammari, I. Hburi, H. R. Idan, and H. F. Khazaal, "An overview of mmWave communications for 5G," in *Proc. Int. Conf. Commun. Inf. Technol.*, pp. 133–139, 2021.

[11] Z. Chen, J. Tang, X. Y. Zhang, D. K. C. So, S. Jin, and K.-K. Wong, "Hybrid evolutionary-based sparse channel estimation for IRS-assisted mmWave MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 3, pp. 1586–1601, 2022.

[12] J. Zhu, D. W. K. Ng, N. Wang, R. Schober, and V. K. Bhargava, "Analysis and design of secure massive MIMO systems in the presence of hardware impairments," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 3, pp. 2001–2016, 2017.

[13] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R. W. Heath, "Frequency-domain compressive channel estimation for frequency-selective hybrid millimeter wave MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 5, pp. 2946–2960, 2018.

[14] X. Ma, F. Yang, S. Liu, J. Song, and Z. Han, "Design and optimization on training sequence for mmWave communications: A new approach for sparse channel estimation in massive MIMO," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1486–1497, 2017.

[15] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. of Sel. topics in Signal Process.*, vol. 8, no. 5, pp. 831–846, 2014.

[16] M. Wang, W. Xu, and A. Tang, "On the performance of sparse recovery via ℓ_p-minimization ($0 \leq p \leq 1$)," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7255–7278, 2011.

[17] F. Chen, L. Shen, and B. W. Suter, "Computing the proximity operator of the ℓ_p norm with $0 \leq p \leq 1$," *IET Signal Processing*, vol. 10, no. 5, pp. 557–565, 2016.

[18] H. Yan and D. Cabria, "Compressive sensing based initial beamforming training for massive MIMO millimeter-wave systems," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, pp. 620–624, 2016.

[19] K.-H. Liu, X. Li, H. Zhao, and G. Fan, "Structured phase retrieval-aided channel estimation for millimeter-wave/sub-terahertz MIMO systems," in *Proc. IEEE Veh. Technol. Conf.*, pp. 1–5, 2022.

[20] S. Ling and T. Strohmer, "Self-calibration and biconvex compressive sensing," *Inverse Problems*, vol. 31, no. 11, p. 115002, 2015.

[21] X. Li, J. Fang, H. Duan, Z. Chen, and H. Li, "Fast beam alignment for millimeter wave communications: A sparse encoding and phaseless decoding approach," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4402–4417, 2019.

[22] C. Hu, X. Wang, L. Dai, and J. Ma, "Partially coherent compressive phase retrieval for millimeter-wave massive MIMO channel estimation," *IEEE Trans. Signal Process.*, vol. 68, pp. 1673–1687, 2020.

[23] C. Wei, H. Jiang, J. Dang, L. Wu, and H. Zhang, "Accurate channel estimation for mmWave massive MIMO with partially coherent phase offsets," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2170–2174, 2022.

[24] "IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications - redline," *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016) - Redline*, pp. 1–7524, 2021.

[25] O. Kome, L. Cen, L. Hanqing, and Y. Rui, "Further details on multi-stage, multi-resolution beamforming training in 802.11 ay, doc," 2016.

[26] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory.*, vol. 57, no. 7, pp. 4680–4688, 2011.

[27] L. Zhang, G. Wang, G. B. Giannakis, and J. Chen, "Compressive phase retrieval via reweighted amplitude flow," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5029–5040, 2018.

[28] Z. Ben-Haim, Y. C. Eldar, and M. Elad, "Coherence-based performance guarantees for estimating a sparse vector under random noise," *IEEE Trans. Signal Process.*, vol. 58, pp. 5030–5043, Oct 2010.

[29] Z. Sidak, "Rectangular confidence regions for the means of multivariate normal distributions," *J. Am. Stat. Assoc.*, vol. 62, no. 318, pp. 626–633, 1967.