

MULTIPLE ANTENNA SIGNAL PROCESSING TECHNIQUES FOR  
MILLIMETER WAVE COMMUNICATIONS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Matthew B. Booth

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF DISSERTATION APPROVAL**

Dr. David J. Love, Co-chair

School of Electrical and Computer Engineering

Dr. Nicolò Michelusi, Co-chair

School of Electrical and Computer Engineering

Dr. James V. Krogmeier

School of Electrical and Computer Engineering

Dr. Michael D. Zoltowski

School of Electrical and Computer Engineering

**Approved by:**

Dr. Dimitrios Peroulis

Head of the School of Electrical and Computer Engineering

To my wife and children.

## ACKNOWLEDGMENTS

Without the help of many people, I would not be at this point in my educational pursuits. Above all of them, though, I thank my Heavenly Father for blessing me with the opportunity to further my education and for helping me throughout the process. I am grateful over the past few years for the reminders God gave me to refocus my perspective on what is eternally significant - faith, family, relationships - and not sacrifice them for anything else. My wife and children are always a buoy when my confidence starts to sink. I cannot express how much I love them and how grateful I am for the sacrifices they made for me to pursue my goals.

The members of the TASC lab made challenging times easier. I learned much through their kind, patient explanations and thoroughly enjoyed the many discussions we had on all sorts of topics. I must give a special thanks to Eric Ruzomberka for his continued willingness to answer questions and to be a sounding board, even when I was not physically present at Purdue. My gratitude goes to my advisors, Prof. David J. Love and Prof. Nicolò Michelusi, for the time and help they have given me. Additionally, I would like to thank the rest of my committee, Prof. James V. Krogmeier and Prof. Michael D. Zoltowski, for their advice and assistance.

Finally, I would also like to acknowledge the support I received through the Purdue Military Research Institute Fellowship and from National Science Foundation Grant CNS-1642982.

## PREFACE

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
ABBREVIATIONS . . . . .	xi
ABSTRACT . . . . .	xiii
1 INTRODUCTION . . . . .	1
1.1 Multi-Armed Bandit Beam Alignment and Tracking for Mobile Mil- limeter Wave Communications . . . . .	3
1.1.1 Multi-Armed Bandits . . . . .	3
1.1.2 Beam Alignment . . . . .	4
1.2 Quantized MIMO Mutual Information Selection . . . . .	6
1.2.1 Antenna Subset Selection . . . . .	6
1.2.2 Related Selection Problems . . . . .	7
1.3 Notation . . . . .	8
2 MULTI-ARMED BANDIT BEAM ALIGNMENT AND TRACKING FOR MOBILE MILLIMETER WAVE COMMUNICATIONS . . . . .	10
2.1 System Overview . . . . .	12
2.2 KSBL-LTS Beam Alignment . . . . .	15
2.2.1 Linear Thompson Sampling (LTS) . . . . .	15
2.2.2 Sparse Bayesian Learning (SBL) and Kalman Filters . . . . .	19
2.3 Numerical Results . . . . .	27
2.3.1 Performance Metrics . . . . .	27
2.3.2 Normalized Beamforming Gain Results . . . . .	29
2.3.3 Example Angle Tracking Results . . . . .	32
2.4 Conclusion . . . . .	34

	Page
3 QUANTIZED MIMO MUTUAL INFORMATION SELECTION . . . . .	35
3.1 Signal Model . . . . .	38
3.2 Quantization . . . . .	39
3.2.1 Additive Quantization Noise Model (AQNM) . . . . .	39
3.2.2 Quantization Gains for Gaussian Distributed Inputs . . . . .	43
3.2.3 Quantized Signal Covariance Using AQNM . . . . .	46
3.3 Problem Setup . . . . .	48
3.4 Mutual Information Selection - Gaussian Approximation . . . . .	52
3.4.1 Joint Mutual Information . . . . .	52
3.4.2 Greedy Algorithm . . . . .	54
3.4.3 Generalized System Model for Quantized MIMO Antenna Selection	58
3.4.4 Numerical Results . . . . .	62
4 CONCLUSION . . . . .	74
REFERENCES . . . . .	76
VITA . . . . .	84

## LIST OF TABLES

Table	Page
1.1 Matrix of SBL algorithms in the literature showing novelty of KSBL-LTS algorithm described in Chapter 2. . . . .	5
2.1 Beam alignment simulation assumptions and parameters . . . . .	27
3.1 Uniform quantization interval lengths and error variances for different values of $B_m$ [97] . . . . .	45
3.2 Non-uniform quantization error variances (actual [97] and estimated) for different values of $B_m$ . . . . .	45
3.3 Mutual information selection simulation parameters . . . . .	63

## LIST OF FIGURES

Figure	Page
2.1 Algorithm timeslot structure. The time step indexes each timeslot, and each timeslot is divided into three phases. Although the figure shows equal times for each of the phases, $T_{tr}$ , $T_{fb}$ , and $T_d$ in general are not equal but must meet the constraint $T_{tr} + T_{fb} + T_d = T_{ts} \leq \min\{T_c, T_b\}$ . . . . .	15
2.2 KSBL-LTS, random beamforming, greedy and omni-directional training $G_{BF}$ vs. time using the channel model from Section 2.1 and $\rho = 0$ dB. All results are for a mobility parameter of $p = 0.01$ with the exception of the LTS ( $p = 0.1$ ) and Omni ( $p = 0.1$ ) curves. These additional results show the respective algorithms' sensitivities to mobility changes for at least $t \leq 300$ channel uses. . . . .	29
2.3 Omni-directional training by turning on one beam at a time (Beam) and by turning on one antenna at a time (Antenna). Both results have similar performance, except cycling through antennas creates an oscillating $G_{BF}$ . . . . .	31
2.4 Example from QuaDRiGa of 100 randomly placed, single antenna receivers (blue) moving in random directions along 15m linear tracks (black lines). The transmitter (red) employs an antenna array. Beamforming gain results from the receivers are averaged together and presented in Fig. 2.5. . . . .	32
2.5 KSBL-LTS and omni-directional training $G_{BF}$ vs. time using the QuaDRiGa channel model. . . . .	33
2.6 Example of KSBL-LTS, greedy, and omni-directional training angle tracking performance over $t = 300$ channel uses for $N_P = 1$ and $p = 0.1$ . In general, random beamforming tracking performance is very noisy and is not shown. . . . .	34
3.1 Analog domain mutual information selection. Analog switch selects a fixed number of ADCs <i>before</i> quantization. . . . .	36
3.2 Digital domain mutual information selection. Bits are generated using one ADC per antenna and then selected <i>after</i> quantization. . . . .	36

Figure	Page
3.3 Receiver block diagram of receiver signal flow and bit subset selection after quantization. The parallel ADC outputs a total of $J$ bits in vector $\mathbf{b}$ . The application requires the output data rate be no greater than some threshold $T$ . After subselection the receiver outputs the $L$ -bit vector $\mathbf{q}$ , where $L \leq T < J$ . . . . .	49
3.4 Average capacity vs. number of selected antennas at various SNRs for unquantized (FAS, MIBM) and quantized (QAFAS, QMIBM) MIMO antenna selection algorithms. For fair comparisons with unquantized algorithms, we only use the equal $B_m = 5$ case. We include the optimal quantized MIMO antenna selection results for $M = 16$ receive antennas. . . . .	64
3.5 Average capacity vs. SNR for the i.i.d. channel at various output data rates $T$ for equal (dash-dot) and random (dot-dot) $B_m$ . (a)-(c) show average capacity when $M = 16$ for optimal selection, modified QAFAS, and QMIBM, respectively, while (d) shows average capacity when $M = 128$ for modified QAFAS. Note the higher average capacity when using random $B_m$ versus equal $B_m$ at lower SNRs and output data rates. . . . .	66
3.6 Average capacity vs. SNR for the mmWave channel at various output data rates $T$ for equal (dash-dot) and random (dot-dot) $B_m$ . (a)-(c) show average capacity when $M = 16$ for optimal selection, modified QAFAS, and QMIBM, respectively, while (d) shows average capacity when $M = 128$ for modified QAFAS. . . . .	67
3.7 Difference between equal $B_m$ and random $B_m$ vs. SNR at various values of $T$ when using optimal selection (a)-(b) and the modified QAFAS algorithm (c)-(d) for $M = 16$ and both channel models. . . . .	68
3.8 Difference between random $B_m$ and equal $B_m$ vs. SNR at various values of $T$ when using the modified QAFAS algorithm. . . . .	69
3.9 Average capacity vs. output data rate $T$ for equal $B_m$ and random $B_m$ at various SNRs for $M = 16$ . . . . .	69
3.10 Average capacity vs. output data rate $T$ for equal $B_m$ and random $B_m$ at various SNRs for $M = 128$ . . . . .	70
3.11 BER curves comparing performance of equal $B_m$ and random $B_m$ for various output data rate thresholds $T$ . . . . .	72
3.12 BER curves comparing performance of equal $B_m$ and random $B_m$ for $T = T_{max}/2$ . . . . .	73

## ABBREVIATIONS

5G	fifth generation
ADC	analog-to-digital converter
AoA	angle of arrival
AoD	angle of departure
AQNM	additive quantization noise model
AWGN	additive white Gaussian noise
BER	bit error rate
CSI	channel state information
DFT	Discrete Fourier Transform
EM	Expectation Maximization
FAS	Fast Antenna Selection
IB	information bottleneck
i.i.d.	independent and identically distributed
KFS	Kalman filtering and smoothing
KSBL	Kalman filter SBL
LTS	Linear Thompson sampling
MAB	multi-armed bandit
MIBM	mutual information-based method
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
MMSE	minimum mean square error
mmWave	millimeter wave
MSE	mean squared-error
OFDM	orthogonal frequency division multiplexing

PDF	probability density function
QAFAS	Quantization Aware Fast Antenna Selection (FAS)
QAM	quadrature amplitude modulation
QMIBM	quantized mutual information-based method (MIBM)
QuaDRiGa	QUasi Deterministic RadIo channel GenerAtor
RF	radio frequency
SBL	sparse Bayesian learning
SNR	signal-to-noise ratio
SVD	singular value decomposition
ULA	uniform linear array

## ABSTRACT

Booth, Matthew B. Ph.D., Purdue University, May 2020. Multiple Antenna Signal Processing Techniques for Millimeter Wave Communications. Major Professors: David J. Love and Nicolò Michelusi.

Mobile devices operating at millimeter wave (mmWave) frequencies are expected to comprise an integral part of fifth generation (5G) communication systems to meet increasing data rate demands. Massive multiple-input multiple-output (MIMO) and advanced signal processing techniques are required to overcome the harsh propagation environment in this spectrum. We focus on two aspects of mmWave communication systems.

First, the large number of antennas creates a challenge in aligning and tracking highly directional, narrow beams. Algorithms which rapidly adapt to the changing mobile environment are required. We propose a novel beam alignment and tracking algorithm for time-varying, sparse mmWave channels using multi-armed bandit beam selection. We show our algorithm has a more rapid initial beam alignment compared to other beam selection policies and, for dynamic channel support, long-term beam-forming gain commensurate to omni-directional channel training. Simulation results are accomplished using idealized and realistic mmWave channel models.

Second, massive MIMO systems can generate potentially prohibitive amounts of data due to the large numbers of antennas. With modern parallel, low-rate analog-to-digital converters (ADCs), the bottleneck is often not in the quantization of the received signals but, rather, in the processing of the digitized bits. Therefore, we develop an adaptive algorithm for down-selecting the digital output data to meet some required output data rate threshold while simultaneously maximizing the information between the transmitted signal and the selected output.

## 1. INTRODUCTION

The exploding demand for wireless bandwidth is driving research into increased use of higher carrier frequencies in wireless systems [1–3]. Although in this work we consider commercial cellular systems, these challenges are not unique to terrestrial, commercial communications. Consider, for example, an imagery satellite passing over a ground station or a military unmanned aerial vehicle enabling communications in a remote location. Using multiple-input multiple-output (MIMO) systems with large antenna arrays at these higher frequencies helps obtain the link margin necessary. Regardless of the application, there exists a ubiquitous push for higher bandwidths and more efficient spectrum use which necessitates adaptive signal processing techniques.

Wireless networks operating between 30 to 300 GHz, known as the millimeter wave (mmWave) spectrum, offer greatly increased bandwidths compared to current cellular frequencies.<sup>1</sup> As such, mmWave devices are expected to comprise an integral part of fifth generation (5G) communication systems to meet increasing data rate demands [4]. However, the mmWave spectrum is a harsh propagation environment where signals experience high propagation losses and are susceptible to severe blocking due to high penetration losses and poor diffraction characteristics [5]. To overcome the propagation loss, practical mmWave systems use massive MIMO with large antenna arrays and narrow, highly directional beams [2]. Along with the sizeable bandwidth, the very large number of transmit and receive antennas creates additional challenges, such as channel estimation and complex analog and digital signal processing. Economically powering these large arrays is also a significant problem.

As a consequence of directional beams and signal blocking, mmWave signals exhibit very limited multipath, quickly changing channel coefficients, and sparse channel

---

<sup>1</sup>Although in this work we focus on cellular systems due to the unfolding of 5G networks, the conclusions generally apply to other MIMO wireless communication systems.

characteristics. Precise alignment between the transmit and receive beams is critical to establish and maintain a communication link. Moreover, mobile mmWave channels are dynamic, meaning the sparse channel support evolves over time. Given the challenges of the mmWave channel, rapid beam training and tracking methods are crucial.

With a large number of antennas, high-speed arrays of analog-to-digital converters (ADCs) digitize the received signal, and scenarios can occur where more output information bits are generated than desired or can be used, i.e., an excessive output data rate. For example, receiver power usage is related to the number of quantization bits. Thus, a power threshold due to system requirements could constrain the number or resolution of the ADCs. An additional example can be found in wireless networks comprised of differing capacity links. To pass data from a high rate link to a lower rate link, some bits must be discarded. When this happens, we wish to intelligently select and send the bits which contribute the most to the end-to-end mutual information. While the application layer can determine which bits to accept and which to reject, the physical layer at the receiver can also alleviate this bottleneck. We can avoid uncontrolled information loss by characterizing how to maximize the mutual information between the input and quantized output given a constraint on the output data rate.

In this dissertation, we focus on two aspects of mobile mmWave communication systems. First, we propose a novel beam alignment and tracking algorithm using multi-armed bandits (MABs) from the area of reinforcement learning. We show our algorithm has rapid initial beam alignment followed by high beam tracking performance in both static and dynamic channel support scenarios. Second, we design algorithms to maximize the mutual information when performing antenna selection to meet some output data rate constraint. In particular, we consider greedy selection algorithms for varying numbers of quantization bits at the antenna outputs.

## 1.1 Multi-Armed Bandit Beam Alignment and Tracking for Mobile Millimeter Wave Communications

### 1.1.1 Multi-Armed Bandits

The MAB name originates from the colloquial term for slot machine levers, and the problem is essentially equivalent to that of a gambler attempting to maximize the total expected payoff from sequentially choosing which slot machine arm (a single-armed bandit) to play from among a set of arms. A trade-off exists between taking actions to *exploit* arms with high expected rewards or to *explore* previously untried arms that may result in even greater rewards. Classic MAB algorithms seek to optimally balance the *exploration* and *exploitation* such that the cumulative rewards are maximized. With diverse applications across many disciplines, MABs enjoy a rich body of literature, and detailed descriptions of the classic MAB problem and its many variants can be found in [6, 7].

In wireless communications, MABs have been used in areas such as power management [8], wideband channel estimation [9, 10], and cognitive radio [11–15]. In [16] MAB algorithms select so-called optimal adaptive antenna states in an orthogonal frequency division multiplexing (OFDM) single-user, MIMO system from the IEEE 802.11n standard. In [17] the mmWave beam alignment problem is formulated as a partially observable Markov decision process (POMDP) for a block-fading channel model, where the process associates each channel block with a state. Both [18, 19] recommend using MABs for training in a wireless communication system. Finally, [20, 21] both use an indirect type of MAB known as the upper confidence bound (UCB) to evaluate transmit-receive beam pairs, returning a reward if the optimal beam pair is exploited and zero otherwise. Neither [20, 21] make use of the sparsity of the mmWave channel.

### 1.1.2 Beam Alignment

In this work, we use the fact that the structure and basic objectives of the beam alignment problem closely parallel those of the MAB problem and can be directly applied to the unknown channel. In beam alignment, the channel is a random vector, and the objective is to quickly determine the beamformers which provide the maximum receive channel signal-to-noise ratio (SNR). We *explore* a set of feasible beamformers through beam training, or by sounding the channel subspace with different beams to more precisely align the transmit and receive beams and improve the receive SNR. We *exploit* a beamformer when we use it for data transmission. After the transmitter sends a pilot signal using the selected beam, the receiver obtains the effective receive channel output and sends it as feedback to the transmitter. This effective receive channel output is the MAB “reward”. In keeping with the general MAB setting, the transmitter does not observe the channel directly. Rather, it only receives the feedback. Linear MAB algorithms use this fact to minimize the *cumulative regret*, or the sum loss relative to the optimal reward at each exploitation.

In Chapter 2 we develop an online Kalman filter sparse Bayesian learning (SBL) algorithm with a Bayesian MAB beam selection policy. We make use of the mmWave channel sparsity by using SBL, a Bayesian machine learning framework for finding sparse representations of signals [22,23]. Although SBL was not initially developed for the time-varying case, the authors in [24], recognizing the Bayesian relation between SBL and Kalman filters, use exact Bayesian inference to derive a recursive algorithm for processing time-varying input data in batches. This algorithm is called Kalman filter SBL (KSBL) and is used in [24] for channel estimation and tracking of OFDM systems. In [25] KSBL is converted to an online algorithm. Parallel to these results, [26] combines static SBL and a Bayesian MAB known as Linear Thompson sampling (LTS) to design a novel sparse MAB algorithm.

As shown in Table 1.1, the contribution we describe in Chapter 2 falls in the intersection of dynamic SBL and LTS MAB action selection. Our algorithm tracks the

Table 1.1.  
Matrix of SBL algorithms in the literature showing novelty of KSBL-LTS algorithm described in Chapter 2.

	No MAB action selection	<b>LTS MAB action selection</b>
Static SBL	[22, 23]	[26]
<b>Dynamic SBL</b>	[24, 25]	<b>Chapter 2</b>

non-zero channel coefficients and rapidly reacts to changes in channel support. Since we modify KSBL by how actions are selected, we compare our algorithm’s normalized beamforming gain performance to that of KSBL using random and greedy selection policies, as well as to omni-directional training. Additionally, we compare KSBL and omni-directional training normalized beamforming gain performance when using a more realistic mmWave channel model generated from the QUasi Deterministic RadIo channel GenerAtor (QuaDRiGa) software package. Finally, we define a tracking measure which we use to compare the ability of KSBL-LTS to find and closely maintain alignment between the transmitter and receiver.

Several beam alignment algorithms which exploit the channel sparsity under static or slowly-changing mmWave channel conditions have been proposed (e.g., [27–32]). These solutions must, essentially, “restart” every time the channel support changes, making them unsuitable for beam tracking in highly mobile systems. Beam tracking algorithms designed for the mobile mmWave scenario are proposed in [33–35]. The authors in [33, 34] assume an *a priori* initial channel estimate and track the channel using an extended Kalman filter for its relative low complexity. However, neither proposal takes advantage of the sparse structure of the mmWave channel as does our approach. In [35], the authors use sequential Monte Carlo sampling to track the sparse channel support; we differ in our use of SBL to determine and track the channel support.

## 1.2 Quantized MIMO Mutual Information Selection

Millimeter wave systems must utilize large, directive antenna arrays, e.g., massive MIMO, in order to overcome the propagation losses inherent at high frequencies. Taking advantage of the wide bandwidths available at these frequencies necessitates many high speed ADCs in parallel. It is well-known that high-speed, high-resolution ADCs are power hungry, not to mention expensive. Thus, since ADC power use increases exponentially with the number of resolution bits, several recent works have proposed parallel implementations of low-resolution ADCs, e.g., [36–40]. Furthermore, regardless of the ADC resolution, as the number of antennas increases so does the potential for exceeding an output data rate constraint due to processing capabilities or link capacities. How these challenges, namely, power consumption and excessive output data rates, are addressed is critical to the performance of mmWave systems.

### 1.2.1 Antenna Subset Selection

We can divide the literature of proposed solutions to these challenges into two main categories: 1) the aforementioned low-resolution ADCs, and 2) antenna subset selection. While the majority of these works focus on minimizing the cost and power consumption under certain constraints, it should be noted that their approaches can also apply to restricting the output data rate. Recent works investigating the performance of low-resolution ADCs include [36–41]. Processing of the signal can occur before the quantization (analog) or completely after quantization (fully digital). Either way, the power used in analog-to-digital conversion increases exponentially with the number of resolution bits [39]. Thus, configurations of antenna arrays with low-resolution ADCs in parallel can significantly reduce the power consumption. The performance loss due to coarse quantization is mitigated by a large number of receive antennas [37].

The second approach, more related to our work, is to only select a subset of antennas for operation. Several earlier works focused on maximizing the mutual information

of unquantized MIMO systems through antenna subset selection, e.g., [42–48]. Since an exhaustive search of all possible combinations of selected antennas is computationally prohibitive, other approaches select antennas based on those returning the largest Frobenius norm of the channel matrix [42], convex optimization [43], the largest post-processing SNR [44], highest energy efficiency [45], the correlation between different channel vectors [48], and the greatest capacity or mutual information [46–48]. More recently, the authors in [37, 49, 50] investigated the capacity limits of MIMO receivers under a variety of architectures, to include multiple antenna selection and sign quantization. Most similar to our work is the recent work in [51]. There, the authors propose a quantized version of the MIMO antenna selection algorithm in [46], under the assumption that all of the quantizers have the same resolution. We differ in that we allow differing numbers of quantization bits, and in our results we show this improves the average capacity over equal numbers of quantization bits.

### 1.2.2 Related Selection Problems

For completeness, we also note that the antenna selection problem is by no means unique to wireless communication systems. For instance, for radar systems the authors in [52] select transmitter-receiver pairs based on the optimization of a spatial correlation coefficient; in [53] the authors formulate the selection as a knapsack problem using the Cramer-Rao bound as the performance metric; and in [54] the authors jointly optimize antenna selection and power allocation using second-order cone programming by convex relaxation.

The antenna selection problem can also be viewed through the lens of machine learning. The bit vector at the output of the ADC is similar to a full set of features extracted from noisy observations, in our case the received signal vector, from which we must classify (estimate) the transmitted vector. In machine learning high numbers of features results in a greater computational load. To combat this, feature selection uses some criteria to select a subset of the features such that the estimation accuracy

is diminished as little as possible. Several machine learning algorithms use mutual information for feature selection, e.g., [55–57], and we refer the interested reader to [58] for a review.

Another related area is the body of work done on the information bottleneck (IB) (cf. [59–61]). Our output bit vector prior to selection must be compressed based on the data rate constraint. Clearly, sending all of the bits would generate the greatest mutual information, so removing some of the bits creates a “bottleneck” on the mutual information.. We briefly summarize some of the IB results for the case of a Gaussian input and a Gaussian channel without a fixed compression rate, as studied in [62–64]. The authors in [62] determine an optimal solution to the Gaussian IB problem. In [63], the authors build on [62] to develop the optimal rate-information trade-off for the scalar, real random variable case, such that the compression rate can be fixed at a desired level and the relevant information maximized. This is extended to the case of real vectors in [64], and it is shown that the information-rate function for the vector case is the sum of information-rate functions of scalar Gaussian channels each with an individual SNR. In practice, each sub-channel includes all sub-channels with positive rate allocations and, for sparse channels, is equal to the number of paths in the channel.

### 1.3 Notation

We use the following notations in this dissertation. Bold capital letters denote matrices, bold lowercase letters denote column vectors, and lower case letters not bolded denote scalars.  $\mathbf{A}^\top$ ,  $\mathbf{A}^H$ ,  $\|\mathbf{a}\|$ ,  $|\mathbf{a}|$ ,  $\text{Tr}(\mathbf{A})$ ,  $\text{vec}(\mathbf{A})$ ,  $\det(\mathbf{A})$ , and  $\text{card}(\mathcal{A})$  denote, respectively, transpose of  $\mathbf{A}$ , conjugate transpose of  $\mathbf{A}$ ,  $\ell_2$ -norm of  $\mathbf{a}$ , absolute value of  $\mathbf{a}$ , trace of  $\mathbf{A}$ , matrix vectorization of  $\mathbf{A}$ , determinant of  $\mathbf{A}$  and cardinality of the set  $\mathcal{A}$ .  $\mathcal{CN}(\mathbf{a}, \mathbf{A})$  denotes a complex Gaussian random vector with mean  $\mathbf{a}$  and covariance matrix  $\mathbf{A}$ . The real and imaginary parts of a number are given by  $\Re\{\cdot\}$  and  $\Im\{\cdot\}$ , respectively. We denote the real domain by  $\mathbb{R}$  and complex domain by  $\mathbb{C}$ . We

denote by  $H(a)$  the entropy of random variable  $a$ , by  $h(a)$  the differential entropy of continuous random variable  $a$ , and by  $I(a; b)$  the mutual information between random variables  $a$  and  $b$ . The probability density (mass) function of a random variable  $a$  is denoted by  $f(a)$  ( $p(a)$ ); the probability of discrete random variable  $a$  being equal to a value  $a'$  is denoted by  $\Pr(a = a')$ . In Chapter 2,  $\log$  is the natural logarithm, while in Chapter 3 it is the base-2 logarithm.

## 2. MULTI-ARMED BANDIT BEAM ALIGNMENT AND TRACKING FOR MOBILE MILLIMETER WAVE COMMUNICATIONS

Millimeter wave (mmWave) communication systems are expected to employ large antenna arrays with beamforming to overcome the propagation loss inherent at high frequencies. Mobility increases the challenge of maintaining alignment of highly directional, narrow beams. Efficient approaches to rapid beam alignment and tracking are critical, and many proposals, such as [65,66], have been made. In [65], the authors point out the fact that mobile scatterers can cause significant, unpredictable angle of departure (AoD) and angle of arrival (AoA) changes even for static users in mmWave systems, and they propose a beamforming algorithm based on compressed sensing. In [66], the authors use pairs of auxiliary beams to track angle variations and steer the data beams based on the angle evolution.

Recently, Kalman filter-based solutions to the beam tracking problem were proposed in [33–35]. The algorithms in [33,34] require external initialization with accurate channel information after which extended Kalman filters track the channel. In [35], sequential Monte Carlo sampling is used to track the sparse channel support, and a Kalman filter estimates the channel separately.

In this chapter, we propose a novel beam alignment and tracking algorithm for time-varying mmWave channels using multi-armed bandit (MAB) beam selection. Differing from [33,34], we exploit the channel sparsity through sparse Bayesian learning (SBL) [22]. Also, our algorithm does both initial beam alignment and subsequent tracking. In contrast to [35], the Kalman filter in our algorithm jointly tracks the sparse support and time-varying channel coefficients.

Sparse Bayesian learning is a machine learning framework for determining an unknown, static sparse vector [22]. The offline *time-varying* SBL was initially developed

in [24] by merging SBL with a Kalman filter to account for temporal correlation with a common support; [25] extends this algorithm to an online version using sawtooth lag smoothing. In [26], *static* SBL is integrated with a linear Bayesian MAB called Linear Thompson sampling (LTS) to combine sparse learning with smart exploration of feasible actions in an offline setting. We advance this progression by integrating the online Kalman filter SBL (KSBL) algorithm with LTS.

Prior work on mmWave beam alignment using MAB algorithms is found in [20,21,67,68], each of which employs *frequentist*-based MABs, whereas our MAB is *Bayesian*-based. Furthermore, [20,21,67] each use MABs with *indirect*, binary feedback, while we use the time-varying mmWave channel estimates as *direct* feedback. In [20], coarse and fine levels of position-aided beam alignment use different MAB variants. On the other hand, our algorithm utilizes a single, common MAB and is designed to flexibly work with arbitrary architectures and beamformer codebooks. The authors in [67] frame the beam alignment problem as a distributed architecture between the transmitter and receiver using adversarial MABs which make no assumption of an underlying channel distribution. Perhaps the beam alignment proposal with a MAB most similar to ours is in [68], where the authors base their algorithm on a general *contextual* MAB and use the receiver’s direction of arrival as the *context*, or side-information, provided to the transmitter to initialize the beam alignment learning algorithm. Our algorithm may be considered a linear contextual MAB, where our contexts would be a given beamformer codebook. The quality of a selected beam in [68] is measured by the amount of data received, while in our algorithm we use the received channel output.

The contributions of this chapter are as follows: (a) development of a KSBL algorithm with an LTS beam selection policy, or KSBL-LTS, and (b) numerical comparison of the beamforming gain performance of KSBL-LTS to that of omni-directional training and KSBL using random beamforming and greedy beam selection. Of the policies considered, we demonstrate that KSBL-LTS has the largest beamforming

gain over the initial rounds of training and data transmission, i.e., rapid initial beam alignment.

## 2.1 System Overview

We assume a transmitter equipped with  $N_{tx}$  antennas for beamforming to a single antenna receiver. We use a narrowband multiple-input single-output (MISO) model for clarity in explaining our approach in this work. However, extensions to the more representative MIMO model can be made by using an equivalent approach from the receiver to the transmitter using combiners. The outdoor mmWave channel is characterized by a small number  $N_P \ll N_{tx}$  of dominant propagation paths due to limited scattering [69]. We consider a channel model given by

$$\mathbf{h}_t^{(g)} = \sqrt{N_{tx}} \sum_{p=1}^{N_P} \beta_p \mathbf{u}(\theta_p),$$

where  $\beta_p \sim \mathcal{CN}(0, \sigma_\beta^2)$  is the complex path gain,  $\theta_p$  is the spatial angle of the  $p$ -th path, and superscript  $(g)$  indicates the *geometric* channel model. The array steering vector for a uniform linear array (ULA) is given by

$$\mathbf{u}(\theta) = \frac{1}{\sqrt{N_{tx}}} [1, e^{-j2\pi\theta}, \dots, e^{-j2\pi(N_{tx}-1)\theta}]^T \in \mathbb{C}^{N_{tx}}. \quad (2.1)$$

The spatial angle  $\theta$  is defined as  $\theta \triangleq \frac{d}{\lambda} \sin(\phi)$ , where  $d$  is the antenna spacing,  $\lambda$  is the signal wavelength, and  $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  is the physical angle in the azimuth plane. Throughout this chapter, we assume  $d = \frac{\lambda}{2}$ .

After correlation with a training sequence, the *geometric* channel input-output relationship at channel use  $t$  is given by

$$y_t = \sqrt{\rho} \left( \mathbf{h}_t^{(g)} \right)^H \mathbf{f}_t^{(g)} + z_t, \quad (2.2)$$

where  $y_t \in \mathbb{C}$  is the received post-processed signal,  $\rho$  is the pre-beamforming signal-to-noise ratio (SNR),  $z_t \sim \mathcal{CN}(0, 1)$  is i.i.d. noise, and  $\mathbf{f}_t^{(g)} \in \mathcal{F}^{(g)}$  is the geometric beamformer. The finite geometric beamforming set  $\mathcal{F}^{(g)} \subset \mathbb{C}^{N_{tx}}$  is arbitrary, allowing for flexible, potentially proprietary, codebooks and hardware configurations. Given  $N_P$  is small, we can infer the channel sparsity over the range of  $\theta$ , but  $\mathbf{h}_t^{(g)}$  itself is *not* sparse.

We explicitly model the mmWave channel as an  $N_P$ -sparse vector by converting  $\mathbf{h}_t^{(g)}$  to an equivalent *virtual* channel vector  $\mathbf{h}_t$  [70]. In the virtual channel model when  $d = \frac{\lambda}{2}$ , we can uniformly divide the range of  $\theta$  into  $N \geq N_{tx}$  discrete spatial angles

$$\theta_i \in \mathcal{V}(N) = \left\{ \frac{i-1}{N} - \frac{1}{2} : i = 1, \dots, N \right\}. \quad (2.3)$$

The vectors  $\mathbf{h}_t^{(g)}$  and  $\mathbf{h}_t$  are related by the linear transformation

$$\mathbf{h}_t = \mathbf{U}^H \mathbf{h}_t^{(g)}, \quad (2.4)$$

where  $\mathbf{U} = [\mathbf{u}(\theta_1), \dots, \mathbf{u}(\theta_N)] \in \mathbb{C}^{N_{tx} \times N}$  and  $\mathbf{h}_t \in \mathbb{C}^N$  is a vector of virtual fading gains corresponding to each  $\theta_i \in \mathcal{V}(N)$ . Note that when  $N > N_{tx}$  we use a more general dictionary than just the set of basis functions of the Discrete Fourier Transform (DFT); this is the so-called *extended virtual representation* [70]. The virtual beamforming vector is obtained from  $\mathbf{f}_t^{(g)}$  by  $\mathbf{f}_t \in \mathcal{F} = \{\mathbf{U}^H (\mathbf{U}\mathbf{U}^H)^{-1} \mathbf{f}_t^{(g)}; \mathbf{f}_t^{(g)} \in \mathcal{F}^{(g)}\}$ . Then the *virtual* input-output relationship can be written as

$$\begin{aligned} y_t &= \sqrt{\rho} \left( \mathbf{h}_t^{(g)} \right)^H \mathbf{U}\mathbf{U}^H (\mathbf{U}\mathbf{U}^H)^{-1} \mathbf{f}_t^{(g)} + z_t \\ &= \sqrt{\rho} \mathbf{h}_t^H \mathbf{f}_t + z_t. \end{aligned} \quad (2.5)$$

The non-zero channel elements of  $\mathbf{h}_t$  are designated by the support  $\mathcal{S}_t$ . Mobility causes both  $\mathbf{h}_t$  and  $\mathcal{S}_t$  to evolve temporally. Since in practice channels are temporally

and spatially correlated, we model this correlation using a Gauss-Markov process. Let the path gains in  $\mathbf{h}_t$  evolve from block to block according to a Gauss-Markov process,

$$\mathbf{h}_t = \epsilon \mathbf{h}_{t-1} + \sqrt{1 - \epsilon^2} \mathbf{g}_t, \quad (2.6)$$

where  $\epsilon \in [0, 1]$  is the correlation coefficient. We assume  $\mathbf{h}_1 \sim \mathcal{CN}(\mathbf{0}, \mathbf{A}_1^{-1})$  and  $\mathbf{g}_t \sim \mathcal{CN}(\mathbf{0}, \mathbf{A}_t^{-1})$ , where  $\mathbf{A}_t^{-1} = \text{diag}(1/\alpha_t)$  is a hyperparameter matrix with  $\alpha_t = [\alpha_{t,1}, \dots, \alpha_{t,N}]$  on the diagonal to enforce channel sparsity [22].  $\mathbf{g}_t$  is i.i.d. and independent of  $\mathbf{h}_1$ , i.e.,  $\mathbb{E}[\mathbf{h}_1 \mathbf{g}_t^H] = \mathbf{0}_{N_{tx}}$ , and  $\mathbf{g}_t, \mathbf{h}_t$  are independent of  $z_t$  for all  $t$ . In addition to varying the path gains, at channel use  $t$  let the elements of  $\mathcal{S}_t$  independently evolve as follows. For  $i \in \mathcal{S}_t$ , at the next channel use  $t + 1$ , with probability  $p$ ,  $i \notin \mathcal{S}_{t+1}$  with either  $i - 1 \in \mathcal{S}_{t+1}$  or  $i + 1 \in \mathcal{S}_{t+1}$  equally likely, or, with probability  $1 - p$ ,  $i \in \mathcal{S}_{t+1}$ . Elements of  $\mathcal{S}_t$  may combine and subsequently separate over time. We assume  $p$  is small to maintain practical mobility settings where the channel support varies *slowly* with time.

As we ultimately consider the general case of an evolving channel, to simplify the notation we delineate an entire timeslot of  $T_{ts}$  seconds with a time step index  $t$  during which  $\mathbf{h}_t$  essentially remains constant. We assume  $T_{ts}$  is less than the channel and beam coherence times  $T_c$  and  $T_b$ . In other words, the timeslots can be considered similar to very brief blocks with duration less than  $\min\{T_c, T_b\}$ . The timeslot structure is shown in Fig. 2.1. Within each timeslot are three divisions: 1) the training phase of duration  $T_{tr}$ , 2) the feedback/update period of duration  $T_{fb}$ , and 3) the data transmission phase of duration  $T_d$ . An initialization phase, which we designate by time step  $t = 1$ , is required for the KSBL algorithm to avoid an ill-conditioned estimate of the channel covariance matrix. We conduct  $N_P$  training phases using random linear combinations of beams from  $\mathcal{F}$ . Upon receiving feedback  $y_{N_P}$ , the estimated channel mean and covariance are updated, as will be described in the subsequent section, using Lines 4 and 5 of Algorithm 1.

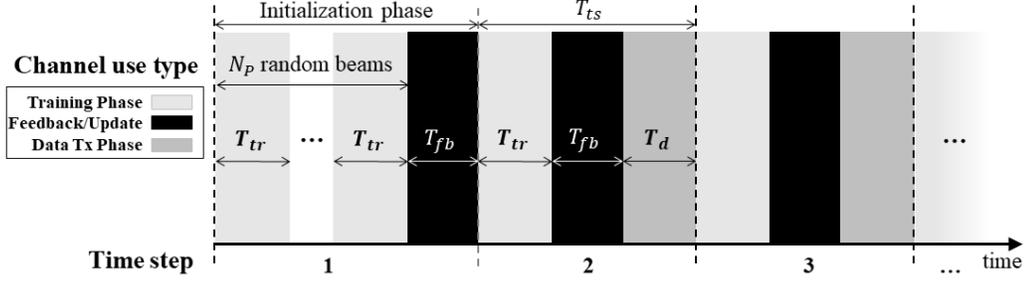


Fig. 2.1. Algorithm timeslot structure. The time step indexes each timeslot, and each timeslot is divided into three phases. Although the figure shows equal times for each of the phases,  $T_{tr}$ ,  $T_{fb}$ , and  $T_d$  in general are not equal but must meet the constraint  $T_{tr} + T_{fb} + T_d = T_{ts} \leq \min\{T_c, T_b\}$ .

After the transmitter sounds with  $\mathbf{f}_t$  from  $\mathcal{F}$  in the transmission phase, the receiver sends only  $y_t$  to the transmitter as feedback. Given the expectation of very reliable, low latency control channels in 5G systems [4], we assume error- and delay-free feedback. Using the training history  $\mathcal{H}_t = \{(\mathbf{f}_\tau, y_\tau) : \tau = 1, \dots, t\}$  as input, the algorithm updates the channel statistics. Data transmission follows after each round of training, feedback and update. After the data transmission, the transmitter selects a beamformer for use during the training phase of the next time step. Given oracle knowledge of  $\mathbf{h}_t$ , the optimal strategy is to select the beamformer  $\mathbf{f}_t^{opt} = \arg \max_{\mathbf{f} \in \mathcal{F}} |\mathbf{h}_t^H \mathbf{f}|$ , which maximizes the receive SNR,  $\Gamma_t = |\mathbf{h}_t^H \mathbf{f}_t|^2$ . We explain the algorithm steps in greater detail in the following section.

## 2.2 KSBL-LTS Beam Alignment

### 2.2.1 Linear Thompson Sampling (LTS)

Conventionally, channel estimation for beam alignment often employs one of the following beam selection policies:

- *Random beamforming.* Beams are randomly generated, as commonly done in compressed channel sensing [71]. We use random beamforming as a policy

within the KSBL algorithm for comparison with the proposed selection policy. As random beamforming continually probes the channel space with random beams regardless of estimated channel statistics, it is comparable to *pure exploration* in decision theory.

- *Greedy beam selection.* Given the channel estimate  $\hat{\mathbf{h}}_{t-1}$ , this policy selects beams which maximize the expected receive SNR, or  $\mathbf{f}_t = \arg \max_{\mathbf{f} \in \mathcal{F}} \left| \hat{\mathbf{h}}_{t-1}^H \mathbf{f} \right|^2$ . This policy always selects beams which maximize performance based on the current channel knowledge and is comparable to *pure exploitation* in decision theory. Similar to the random beamforming policy, greedy beam selection is not a stand alone algorithm but, rather, is used within the KSBL algorithm for comparison with the proposed selection policy.
- *Omni-directional training.* A purely deterministic stand alone algorithm which cycles beam selection through a set of orthogonal beams. It is a version of *exhaustive search*. In this beam training method we let  $N = N_{tx}$  such that the transform matrix  $\mathbf{U}^{(g)}$  is the square  $N_{tx} \times N_{tx}$  DFT matrix. We only use geometric channel vectors  $\mathbf{h}_t^{(g)}$ . The beamforming vectors consist of the columns of  $\mathbf{U}^{(g)}$ , or  $\mathcal{F}^{(g)} = \{\mathbf{u}(\theta_n) : \theta_n \in \mathcal{V}(N_{tx}), n = 1, \dots, N_{tx}\}$ , where  $\theta_n$  is defined in (2.3). At the  $t$ -th channel use, column  $n$  of the DFT matrix is selected for the training beamformer  $\mathbf{f}_t^{(g)}$ , where  $n$  is found by

$$n = \begin{cases} t \bmod N_{tx}, & \text{when } t \bmod N_{tx} \neq 0 \\ N_{tx}, & \text{when } t \bmod N_{tx} = 0 \end{cases}$$

and mod is the modulus operation. Thus, we continuously cycle through all  $N_{tx}$  beams when selecting a DFT training beam at each new channel use.

For a fair comparison with the KSBL-LTS algorithm, we use a Kalman filter while cycling through the DFT training beams. This enables us to incorporate both knowledge of the channel evolution from (2.6) and of the received signal

samples from (2.2). Note that since the omni-directional training algorithm uses the geometric channel representation, (2.6) is rewritten as

$$\mathbf{h}_t^{(g)} = \epsilon \mathbf{h}_{t-1}^{(g)} + \sqrt{1 - \epsilon^2} \mathbf{g}_t^{(g)},$$

where  $\mathbf{h}_1^{(g)}, \mathbf{g}_t^{(g)} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q})$ . The covariance matrix  $\mathbf{Q}$  is derived from converting  $\mathbf{h}_1^{(g)}, \mathbf{g}_t$  to their geometric representations. For example, applying the inverse relationship of (2.4) to  $\mathbf{g}_t^{(g)}$ , we get

$$\mathbf{g}_t^{(g)} = (\mathbf{U}\mathbf{U}^H)^{-1} \mathbf{U} \mathbf{g}_t, \quad (2.7)$$

and  $\mathbb{E} \left[ \mathbf{g}_t^{(g)} \left( \mathbf{g}_t^{(g)} \right)^H \right] = (\mathbf{U}\mathbf{U}^H)^{-1} \mathbf{U} \mathbf{A}_t^{-1} \mathbf{U}^H (\mathbf{U}\mathbf{U}^H)^{-1}$ . However, the omni-directional training algorithm does not have knowledge of the initial channel support and cannot explicitly track channel support changes or time-varying covariance matrices. Therefore, we assume  $\mathbf{A}_t^{-1} = \mathbf{I}_N, \forall t$ , which is equal to the KSBL-LTS initialization assumption we will make for  $\mathbf{A}_t^{-1}$  in Section 2.2.2. Thus,  $\mathbf{Q} = (\mathbf{U}\mathbf{U}^H)^{-1}$ .

The omni-directional training Kalman filter is then described by (2.8)-(2.12) for  $s = 1, \dots, T$ . We initialize the algorithm with  $\widehat{\mathbf{h}}_{0|0}^{(g)} = \mathbf{0}$  and  $\Sigma_{0|0}^{(g)} = \mathbf{Q}$ .

$$\widehat{\mathbf{h}}_{s|s-1}^{(g)} = \epsilon \widehat{\mathbf{h}}_{s-1|s-1}^{(g)} \quad (2.8)$$

$$\Sigma_{s|s-1}^{(g)} = \epsilon^2 \Sigma_{s-1|s-1}^{(g)} + (1 - \epsilon^2) \mathbf{Q} \quad (2.9)$$

$$\mathbf{K}_s^{(g)} = \sqrt{\rho} \Sigma_{s|s-1}^{(g)} \mathbf{f}_s^{(g)} \left( 1 + \rho (\mathbf{f}_s^{(g)})^H \Sigma_{s|s-1}^{(g)} \mathbf{f}_s^{(g)} \right)^{-1} \quad (2.10)$$

$$\widehat{\mathbf{h}}_{s|s}^{(g)} = \widehat{\mathbf{h}}_{s|s-1}^{(g)} + \mathbf{K}_s^{(g)} \left( y_s^H - \sqrt{\rho} (\mathbf{f}_s^{(g)})^H \widehat{\mathbf{h}}_{s|s-1}^{(g)} \right) \quad (2.11)$$

$$\Sigma_{s|s}^{(g)} = (\mathbf{I}_{N_{tx}} - \sqrt{\rho} \mathbf{K}_s^{(g)} (\mathbf{f}_s^{(g)})^H) \Sigma_{s|s-1}^{(g)} \quad (2.12)$$

At each channel use we use a greedy policy with the current channel estimate  $\widehat{\mathbf{h}}_{t|t}^{(g)}$  to select the beam used in the data transmission phase. In summary, we

select a single DFT training beam for each training phase, get feedback and update the channel statistics using a Kalman filter, and greedily determine the beam for data transmission before advancing to the next training beam in the cycle.

This omni-directional training approach is distinctly different from sounding the channel by turning on one transmit antenna in an array at a time. Omni-directional training turns on one beam at a time and sequentially points the beam in different, discretized angular directions. All of the angular directions must be explored to obtain the most accurate channel estimate. To form such a beam, every antenna element is used, as seen from the array steering vector equation in (2.1).

On the other hand, training by transmitting on a single element of an array at a time only returns channel information between the selected transmit antenna and the receiver antenna. We must sound with each of the transmit antennas in order to obtain an accurate channel estimate. This is accomplished in largely the same way as is done in omni-directional training, except we replace the DFT matrix with the identity matrix  $\mathbf{I}_{N_{tx}}$ . Then, instead of selecting  $\mathbf{f}_t^{(g)}$  from the columns of the DFT matrix, we successively select  $\mathbf{f}_t^{(g)}$  from the columns of  $\mathbf{I}_{N_{tx}}$ . The performances for both approaches are compared in Fig. 2.3 in Section 2.3.2.

Our algorithm *does not use any of these approaches*. Instead, we employ a MAB method for online decision-making called Linear Thompson sampling (LTS) that *balances the exploration and exploitation trade-off* [72]. With LTS we draw a random sample  $\tilde{\mathbf{h}}_t \sim \mathcal{CN}(\hat{\mathbf{h}}_{t-1}, \mathbf{\Sigma}_{t-1})$  from  $p(\mathbf{h}|\mathcal{H}_{t-1})$ . We then select a training beamformer that maximizes the reward given  $\tilde{\mathbf{h}}_t$ , or

$$\mathbf{f}_t = \arg \max_{\mathbf{f} \in \mathcal{F}} \left| \tilde{\mathbf{h}}_t^H \mathbf{f} \right|.$$

Basing our beamformer selection on a random sample of the channel posterior enables exploration of  $\mathcal{F}$  by exploiting the covariance  $\mathbf{\Sigma}_{t-1}$  of the channel estimate. Larger

(smaller) values of  $\|\boldsymbol{\Sigma}_{t-1}\|_F$  imply more (less) exploration around  $\hat{\mathbf{h}}_{t-1}$ . In this way, LTS builds *intelligent exploration* into the MAB beam selection policy we use during training.

Before describing the algorithm details in the next section, we need to make some clarifying statements. First, the training channel outputs and beamformers up to time  $t$  are collected, respectively, into vector  $\mathbf{y}_t = [y_1, \dots, y_t]^\top$  and  $t \times N$  matrix  $\mathbf{F}_t = [\mathbf{f}_1, \dots, \mathbf{f}_t]^\top$ . Second, *data transmission always employs a greedy beamformer selection policy*, regardless of the selection policy used during the training. Finally, LTS by itself *does not induce sparsity in the channel*. This is accomplished through the SBL part of the algorithm.

### 2.2.2 Sparse Bayesian Learning (SBL) and Kalman Filters

Sparse Bayesian learning uses hierarchical Bayesian inference to estimate a parameterized prior based on the data. We assume a complex Gaussian prior on each channel vector element  $h_{t,i}, i = \{1, \dots, N\}$  with parametric form  $p(h_{t,i}; \alpha_{t,i}) \sim \mathcal{CN}(0, \alpha_{t,i}^{-1})$ , where  $\alpha_{t,i}$  is the hyperparameter controlling the variance of  $h_{t,i}$ . Each element  $i$  of  $\mathbf{h}_t$  and  $\boldsymbol{\alpha}_t$  corresponds to  $\theta_i$ . If  $\alpha_{t,i} \rightarrow \infty$ ,  $h_{t,i}$  is effectively ‘turned off’ for that  $\theta_i$ . Thus,  $\boldsymbol{\alpha}_t$  relates to the support  $\mathcal{S}_t$  of  $\mathbf{h}_t$ , which contains the indices of the paths with nonzero gains. For element indices  $i \notin \mathcal{S}_t$ ,  $\alpha_{t,i} = \infty$ . We induce channel sparseness by assigning a hyperprior for  $\boldsymbol{\alpha}_t$  that ensures the majority of the  $\alpha_i$ ’s are very large.

After sounding, we update the hyperparameter estimates  $\hat{\boldsymbol{\alpha}}_t$  using a SBL algorithm which we describe shortly. Updates to the elements of  $\hat{\boldsymbol{\alpha}}_t$  determine the corresponding estimated channel weights in  $\hat{\mathbf{h}}_t$  and the diagonal values of  $\boldsymbol{\Sigma}_t$ . In turn,  $\boldsymbol{\Sigma}_t$  affects the distribution of the next channel realization used by the LTS beam selection policy. For every  $i \notin \mathcal{S}$ , the algorithm attempts to cause  $\hat{\alpha}_{t,i} \rightarrow \infty$  such that the corresponding  $\hat{h}_{t,i} \rightarrow 0$ , i.e., the expected multipath channel component at spatial angle  $\theta_i$  is negligible. Most  $\hat{\alpha}_{t,i} \rightarrow \infty$  as the algorithm progresses, reflecting the channel sparseness.

We estimate the channel statistics and hyperparameters using a modified Kalman filter. Without modification, the general Kalman filter is not optimal for sparse channel estimation. In [24] an offline version of Kalman filtering and smoothing is developed which incorporates SBL for sparse channel estimation. The authors in [24] integrate SBL with a Kalman filter (KSBL) using a common hyperparameter vector. They assume the time-varying channels are so-called ‘group approximately-sparse’, meaning the large-scale parameters, and by consequence the hyperparameters, vary slowly enough so as to be considered constant for the duration of their offline estimation.

While we also assume a slowly changing channel support, we do not assume  $\boldsymbol{\alpha}_t$  is constant. Rather, we adapt an *online* version of KSBL as described in [25] for our beam alignment and tracking problem. In this version of KSBL, a small time lag of  $\Delta$  channel uses is introduced over which the algorithm estimates the channel and slowly time-varying hyperparameters. However, for ease of exposition, in the following description of the KSBL algorithm *we assume  $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}$  is constant for a time interval of  $T$  channel uses* and, by extension,  $\mathbf{A}_t = \mathbf{A}$  for  $t = 1, \dots, T$ . After explaining the KSBL algorithm, we then describe the incorporation of the interval  $\Delta$  into our algorithm.

We now summarize the steps of the KSBL algorithm; we refer the reader to [24] for additional details. After  $T$  transmissions and feedback the joint probability density function (PDF) of the received feedback and the temporally correlated channels, parameterized by  $\boldsymbol{\alpha}$ , can be written as

$$p(\mathbf{y}_T, \mathbf{h}_1, \dots, \mathbf{h}_T; \boldsymbol{\alpha}) = \prod_{t=1}^T p(y_t | \mathbf{h}_t) p(\mathbf{h}_t | \mathbf{h}_{t-1}; \boldsymbol{\alpha}), \quad (2.13)$$

where, by convention, we assume  $p(\mathbf{h}_1|\mathbf{h}_0; \boldsymbol{\alpha}) = p(\mathbf{h}_1; \boldsymbol{\alpha})$  and  $\mathbf{h}_1 \sim \mathcal{CN}(\mathbf{0}, \mathbf{A})$ . Taking  $-\log p(\mathbf{y}_T, \mathbf{h}_1, \dots, \mathbf{h}_T; \boldsymbol{\alpha})$  and neglecting terms constant with respect to  $\mathbf{h}$  and  $\boldsymbol{\alpha}$ , we define the cost function  $\mathcal{L}_T$  as

$$\begin{aligned} \mathcal{L}_T \triangleq & \sum_{t=1}^T |y_t - \sqrt{\rho} \mathbf{h}_t^H \mathbf{f}_t|^2 - T \log \det(\mathbf{A}) \\ & + \sum_{t=2}^T \frac{(\mathbf{h}_t - \epsilon \mathbf{h}_{t-1})^H \mathbf{A} (\mathbf{h}_t - \epsilon \mathbf{h}_{t-1})}{(1 - \epsilon^2)} + \mathbf{h}_1^H \mathbf{A} \mathbf{h}_1. \end{aligned} \quad (2.14)$$

We jointly estimate the time-varying channel and hyperparameter vector by

$$\hat{\mathbf{h}}_1, \dots, \hat{\mathbf{h}}_T, \hat{\boldsymbol{\alpha}} = \arg \min_{\mathbf{h}_1, \dots, \mathbf{h}_T, \boldsymbol{\alpha}} \mathcal{L}_T. \quad (2.15)$$

A closed-form solution to the optimization problem in (2.15) does not exist, and we can only obtain the incomplete data from observations  $\mathbf{y}_T = [y_1, \dots, y_T]^T$ . Therefore, letting  $\mathbf{h}_1, \dots, \mathbf{h}_T$  be hidden variables, we use the Expectation Maximization (EM) algorithm to maximize the expectation of the complete-data log likelihood with respect to the posterior distribution. The EM steps are the following:

$$\text{E-step: } \mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)}) = \mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} \mathcal{L}_T \quad (2.16)$$

$$\text{M-step: } \boldsymbol{\alpha}^{(k+1)} = \arg \min_{\boldsymbol{\alpha}} \mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)}) \quad (2.17)$$

Note that we found (2.14) using a negative logarithm, so minimizing the expectation with respect to  $\boldsymbol{\alpha}$  accomplishes the M-step.

The E-step finds the marginal log-likelihood of the observations. To do so, we use the current hyperparameter values  $\boldsymbol{\alpha}^{(k)}$ , where  $k$  is the EM algorithm iteration index, to compute the posterior mean and covariance of the channel over the  $T$  channel uses. This can be accomplished recursively using the Kalman filtering and smoothing (KFS) equations. We use the notation  $\mathbf{a}_{t|s}$  to denote the estimate (channel mean or covariance) of  $\mathbf{a}$  at time  $t$  given the training history  $\mathcal{H}_s$  (beamformers and channel outputs) through time  $s$ , where  $t$  may be greater than, less than or equal to  $s$ . The

channel estimate  $\widehat{\mathbf{h}}_{t|T}$  depends on past and future channel observations. However, the Kalman filter only uses past observations. Therefore, we require backward recursions using the Kalman smoother to ensure the observations up to the  $T$ -th channel use are included in  $\widehat{\mathbf{h}}_{t|T}$  for  $1 \leq t < T$ . The KFS equations for the  $k$ -th iteration are given below, where we define  $\mathbf{J}_{r-1} \triangleq \epsilon \boldsymbol{\Sigma}_{r-1|r-1} \boldsymbol{\Sigma}_{r|r-1}^{-1}$  and  $\mathbf{K}_s$  as the Kalman gain. We initialize the recursions with  $\widehat{\mathbf{h}}_{0|0} = \mathbf{0}$  and  $\boldsymbol{\Sigma}_{0|0} = (\mathbf{A}^{(0)})^{-1}$ , where we arbitrarily set  $\mathbf{A}^{(0)} = \mathbf{I}_N$ . Then the update proceeds as follows:

**for**  $s = 1, \dots, T$  **do**

Prediction:

$$\widehat{\mathbf{h}}_{s|s-1} = \epsilon \widehat{\mathbf{h}}_{s-1|s-1} \quad (2.18)$$

$$\boldsymbol{\Sigma}_{s|s-1} = \epsilon^2 \boldsymbol{\Sigma}_{s-1|s-1} + (1 - \epsilon^2) (\mathbf{A}^{(k)})^{-1} \quad (2.19)$$

Filtering:

$$\mathbf{K}_s = \sqrt{\rho} \boldsymbol{\Sigma}_{s|s-1} \mathbf{f}_s \left( 1 + \rho \mathbf{f}_s^H \boldsymbol{\Sigma}_{s|s-1} \mathbf{f}_s \right)^{-1} \quad (2.20)$$

$$\widehat{\mathbf{h}}_{s|s} = \widehat{\mathbf{h}}_{s|s-1} + \mathbf{K}_s \left( y_s^H - \sqrt{\rho} \mathbf{f}_s^H \widehat{\mathbf{h}}_{s|s-1} \right) \quad (2.21)$$

$$\boldsymbol{\Sigma}_{s|s} = (\mathbf{I}_N - \sqrt{\rho} \mathbf{K}_s \mathbf{f}_s^H) \boldsymbol{\Sigma}_{s|s-1} \quad (2.22)$$

**end**

**for**  $r = T, T - 1, \dots, 2$  **do**

Smoothing:

$$\widehat{\mathbf{h}}_{r-1|T} = \widehat{\mathbf{h}}_{r-1|r-1} + \mathbf{J}_{r-1} (\widehat{\mathbf{h}}_{r|T} - \widehat{\mathbf{h}}_{r|r-1}) \quad (2.23)$$

$$\boldsymbol{\Sigma}_{r-1|T} = \boldsymbol{\Sigma}_{r-1|r-1} + \mathbf{J}_{r-1} (\boldsymbol{\Sigma}_{r|T} - \boldsymbol{\Sigma}_{r|r-1}) \mathbf{J}_{r-1}^H \quad (2.24)$$

**end**

In the prediction and filtering equations in (2.18)-(2.22), we use forward recursions to estimate the channel statistics  $\widehat{\mathbf{h}}_{t|t}$  and  $\boldsymbol{\Sigma}_{t|t}$  using the correlation coefficient  $\epsilon$  from the Gauss-Markov model in (2.6) and the training history  $\mathcal{H}_{t-1}$ . The Kalman filter estimates these statistics using a weighted average of the previous channel estimate

$\widehat{\mathbf{h}}_{t|t-1}$  and the most recently received feedback  $y_t$ . The weights, contained in the Kalman gain matrix  $\mathbf{K}_s$ , are updated at each time step using the estimated channel covariance  $\boldsymbol{\Sigma}_{t|t-1}$ . The smoothing equations in (2.23)-(2.24) go backward from time step  $T$  to recursively improve the accuracy of the previous channel estimates.

Given  $\widehat{\mathbf{h}}_{t|T}$  and  $\boldsymbol{\Sigma}_{t|T}$ ,  $1 \leq t \leq T$ , from the KFS equations,  $\mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)})$  in (2.16) can be simplified as

$$\begin{aligned} \mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)}) &= T \log \det(\mathbf{A}^{-1}) + \frac{1}{1 - \epsilon^2} \sum_{t=2}^T \mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} \left[ (\mathbf{h}_t - \epsilon \mathbf{h}_{t-1})^H \mathbf{A} (\mathbf{h}_t - \epsilon \mathbf{h}_{t-1}) \right] \\ &\quad + \mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} \left[ \mathbf{h}_1^H \mathbf{A} \mathbf{h}_1 \right], \end{aligned} \quad (2.25)$$

where we have dropped the first term from the right-hand side of (2.14) due to it being constant with respect to  $\boldsymbol{\alpha}$ . We further simplify  $\mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)})$  by computing the following expectations in terms of variables calculated from the KFS equations:

$$\mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} [\mathbf{h}_r] \triangleq \widehat{\mathbf{h}}_{r|T}, \quad (2.26)$$

$$\mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} [\mathbf{h}_r \mathbf{h}_r^H] \triangleq \boldsymbol{\Sigma}_{r|T} + \widehat{\mathbf{h}}_{r|T} \widehat{\mathbf{h}}_{r|T}^H, \quad (2.27)$$

for  $r = 1, \dots, T$ , and

$$\mathbb{E}_{\mathbf{h}_1, \dots, \mathbf{h}_T | \mathbf{y}_T; \boldsymbol{\alpha}^{(k)}} [\mathbf{h}_r \mathbf{h}_{r-1}^H] \triangleq \boldsymbol{\Sigma}_{r, r-1|T} + \widehat{\mathbf{h}}_{r|T} \widehat{\mathbf{h}}_{r-1|T}^H, \quad (2.28)$$

for  $r = T, T-1, \dots, 2$ . Equations (2.26) and (2.27) are the posterior mean and covariance, respectively, and are obtained from (2.18)-(2.24). We have from [73] that (2.28) can be written as

$$\boldsymbol{\Sigma}_{r-1, r-2|T} = \boldsymbol{\Sigma}_{r-1|r-1} \mathbf{J}_{r-2}^H + \mathbf{J}_{r-1}^H (\boldsymbol{\Sigma}_{r, r-1|T} - \epsilon \boldsymbol{\Sigma}_{r-1|r-1}) \mathbf{J}_{r-2}. \quad (2.29)$$

---

**Algorithm 1:** KSBL-LTS Algorithm
 

---

- 1 **Initialization:**
  - 2 Generate  $N_P$  beamformers and observe channel outputs  $y_{N_P}$ ;
  - 3 Set  $\hat{\mathbf{A}}_{N_P} = \mathbf{I}_N$  and calculate  $\boldsymbol{\Sigma}_{N_P} = \left( \hat{\mathbf{A}}_{N_P} + \rho \mathbf{F}_{N_P}^H \mathbf{F}_{N_P} \right)^{-1}$ ;
  - 4 Calculate  $\hat{\mathbf{h}}_{N_P} = \sqrt{\rho} \boldsymbol{\Sigma}_{N_P} \mathbf{F}_{N_P}^H \mathbf{y}_{N_P}$ ;
  - 5 **for**  $t = N_P + 1, N_P + 2, \dots$  **do**
  - 6     **Training:**
  - 7     Get sample  $\tilde{\mathbf{h}}_t$  from distribution  $\mathcal{CN} \left( \hat{\mathbf{h}}_{t-1}, \boldsymbol{\Sigma}_{t-1} \right)$ ;
  - 8     Select beamformer  $\mathbf{f}_t = \arg \max_{\mathbf{f} \in \mathcal{F}} \left| \tilde{\mathbf{h}}_t^H \mathbf{f} \right|$ ;
  - 9     Observe channel output  $y_t$ ;
  - 10    Update  $\{\hat{\mathbf{h}}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \hat{\mathbf{A}}_{t-1}\} \rightarrow \{\hat{\mathbf{h}}_t, \boldsymbol{\Sigma}_t, \hat{\mathbf{A}}_t\}$  by inputting  $\{\mathbf{y}_{t-(\Delta+1):t}, \mathbf{F}_{t-(\Delta+1):t}\}$  to Algorithm 2;
  - 11    **Data Transmission:**
  - 12    Transmit with  $\mathbf{f}_t^{(data)} = \arg \max_{\mathbf{f} \in \mathcal{F}} \left| \hat{\mathbf{h}}_t^H \mathbf{f} \right|$ ;
  - 13    Increment  $t$ ;
  - 14 **end**
- 

We initialize  $\boldsymbol{\Sigma}_{r-1, r-2|T}$  by  $\boldsymbol{\Sigma}_{T, T-1|T} = \epsilon \left( \mathbf{I}_N - \sqrt{\rho} \mathbf{K}_T \mathbf{f}_T^H \right) \boldsymbol{\Sigma}_{T-1|T-1}$ . Defining

$$\mathbf{M}_{r|T} \triangleq \boldsymbol{\Sigma}_{r|T} + \hat{\mathbf{h}}_{r|T} \hat{\mathbf{h}}_{r|T}^H + \epsilon^2 \left( \boldsymbol{\Sigma}_{r-1|T} + \hat{\mathbf{h}}_{r-1|T} \hat{\mathbf{h}}_{r-1|T}^H \right) - 2\epsilon \text{Re} \left\{ \boldsymbol{\Sigma}_{r, r-1|T} + \hat{\mathbf{h}}_{r|T} \hat{\mathbf{h}}_{r-1|T}^H \right\},$$

and

$$\mathbf{M}_{1|T} \triangleq \boldsymbol{\Sigma}_{1|T} + \hat{\mathbf{h}}_{1|T} \hat{\mathbf{h}}_{1|T}^H, \quad (2.30)$$

we obtain the final simplification of the E-step as

$$\mathcal{Q}(\boldsymbol{\alpha} | \boldsymbol{\alpha}^{(k)}) = T \log |\mathbf{A}^{-1}| + \text{Tr}(\mathbf{A} \mathbf{M}_{1|T}) + \frac{1}{1 - \epsilon^2} \sum_{r=2}^T \text{Tr}(\mathbf{A} \mathbf{M}_{r|T}). \quad (2.31)$$

The M-step finds the hyperparameters  $\boldsymbol{\alpha}$  which minimize  $\mathcal{Q}(\boldsymbol{\alpha}|\boldsymbol{\alpha}^{(k)})$ . Setting the derivative w.r.t  $\alpha^{(k+1)}(i)$  of (2.31) equal to zero, we get the following update equation for the  $i$ -th element,  $i = 1, \dots, N$ , of the hyperparameter vector:

$$\alpha^{(k+1)}(i) = T \left( \sum_{r=2}^T \frac{M_{r|T}(i, i)}{(1 - \epsilon^2)} + M_{1|T}(i, i) \right)^{-1}.$$

We iterate the EM algorithm until the difference in hyperparameter updates converges to some threshold  $\epsilon$ .

The KSBL algorithm in [24] is an offline only algorithm and does not enable efficient beam alignment and tracking. Therefore, we adopt the online KSBL approach from [25]. In this approach, we estimate over  $\Delta$  channel uses instead of all  $T$  channel uses. We accomplish KFS over  $t - (\Delta + 1), t - \Delta, \dots, t$  and output channel estimates  $\hat{\mathbf{h}}_{t-(\Delta+1):t}$ .<sup>1</sup> Then the interval shifts by one and repeats. In the preceding description of the KSBL algorithm, we assumed a constant hyperparameter vector  $\boldsymbol{\alpha}$ . Now we modify this assumption to make  $\boldsymbol{\alpha}_t$  only constant over the lag  $\Delta$ . Since the interval shifts by one at each time step, we revert back to using a subscript  $t$ . The parameter  $\Delta$  should be adjusted to meet system requirements. Upon meeting its stopping condition, Algorithm 2 returns  $\hat{\mathbf{h}}_{t|t}$ ,  $\boldsymbol{\Sigma}_{t|t}$ , and  $\hat{\boldsymbol{\alpha}}_t$ . For data transmission, our algorithm uses  $\hat{\mathbf{h}}_{t|t}$ . The KFS equations for the  $k$ -th iteration of the online version of KSBL are summarized in Algorithm 2.

As described, however, the KSBL algorithm contains no mechanism for balancing exploration and exploitation in beamformer selection. We incorporate this by constructing an LTS “wrapper” for the Kalman filter. Using the estimated channel distribution  $\mathcal{CN}(\hat{\mathbf{h}}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ , LTS obtains a sample channel realization  $\tilde{\mathbf{h}}_t$ . It then selects the beamformer  $\mathbf{f}_t = \arg \max_{\mathbf{f} \in \mathcal{F}} |\tilde{\mathbf{h}}_t^H \mathbf{f}|$  to sound the channel on. After transmission and feedback, we input the observed channel output  $y_t$  into the KFS update equations in Algorithm 2 to re-estimate the channel statistics. We use the newly estimated channel vector to greedily select the best beamformer for the data trans-

<sup>1</sup>The +1 in  $(\Delta + 1)$  is purely for accounting purposes as in [25].

---

**Algorithm 2:** KSBL Algorithm
 

---

**Input:**  $\mathbf{y}_{t-(\Delta+1):t}$ ,  $\mathbf{F}_{t-(\Delta+1):t}$ ,  $k_{max}$ ,  $\varepsilon$   
 1 Set difference = 1,  $k = 0$ ,  $\boldsymbol{\alpha}^{(0)} = \mathbf{I}_N$ ;  
 2 **while** (difference  $> \varepsilon$  and  $k < k_{max}$ ) **do**  
 3     **E-step:** Set  $\boldsymbol{\Sigma}_{0|0} = (\mathbf{A}^{(k)})^{-1}$ ;  $\hat{\mathbf{h}}_{0|0} = \mathbf{0}$  ;  
 4     **for**  $s = t - (\Delta + 1), t - \Delta, \dots, t$  **do**  
 5         Prediction:  $\hat{\mathbf{h}}_{s|s-1} = \epsilon \hat{\mathbf{h}}_{s-1|s-1}$  ;  
 6          $\boldsymbol{\Sigma}_{s|s-1} = \epsilon^2 \boldsymbol{\Sigma}_{s-1|s-1} + (1 - \epsilon^2)(\mathbf{A}^{(k)})^{-1}$  ;  
 7         Filtering:  $\mathbf{K}_s = \sqrt{\rho} \boldsymbol{\Sigma}_{s|s-1} \mathbf{f}_s (1 + \rho \mathbf{f}_s^H \boldsymbol{\Sigma}_{s|s-1} \mathbf{f}_s)^{-1}$  ;  
 8          $\hat{\mathbf{h}}_{s|s} = \hat{\mathbf{h}}_{s|s-1} + \mathbf{K}_s (\mathbf{y}_s^H - \sqrt{\rho} \mathbf{f}_s^H \hat{\mathbf{h}}_{s|s-1})$  ;  
 9          $\boldsymbol{\Sigma}_{s|s} = (\mathbf{I}_N - \sqrt{\rho} \mathbf{K}_s \mathbf{f}_s^H) \boldsymbol{\Sigma}_{s|s-1}$  ;  
 10     **end**  
 11     Smoothing: Set  $\boldsymbol{\Sigma}_{t,t-1|t} = \epsilon (\mathbf{I}_N - \sqrt{\rho} \mathbf{K}_t \mathbf{f}_t^H) \boldsymbol{\Sigma}_{t-1|t-1}$ ;  
 12     **for**  $r = t, t-1, \dots, t - (\Delta + 1)$  **do**  
 13          $\mathbf{J}_{r-1} = \epsilon \boldsymbol{\Sigma}_{r-1|r-1} \boldsymbol{\Sigma}_{r|r-1}^{-1}$  and  $\mathbf{J}_{r-2} = \epsilon \boldsymbol{\Sigma}_{r-2|r-2} \boldsymbol{\Sigma}_{r-1|r-2}^{-1}$ ;  
 14          $\hat{\mathbf{h}}_{r-1|t} = \hat{\mathbf{h}}_{r-1|r-1} + \mathbf{J}_{r-1} (\hat{\mathbf{h}}_{r|t} - \hat{\mathbf{h}}_{r|r-1})$ ;  
 15          $\boldsymbol{\Sigma}_{r-1|t} = \boldsymbol{\Sigma}_{r-1|r-1} + \mathbf{J}_{r-1} (\boldsymbol{\Sigma}_{r|t} - \boldsymbol{\Sigma}_{r|r-1}) \mathbf{J}_{r-1}^H$ ;  
 16          $\boldsymbol{\Sigma}_{r-1,r-2|t} = \boldsymbol{\Sigma}_{r-1|r-1} \mathbf{J}_{r-2}^H + \mathbf{J}_{r-1}^H (\boldsymbol{\Sigma}_{r,r-1|t} - \epsilon \boldsymbol{\Sigma}_{r-1|r-1}) \mathbf{J}_{r-2}$ ;  
 17          $\mathbf{M}_{r|t} =$   
            $\boldsymbol{\Sigma}_{r|t} + \hat{\mathbf{h}}_{r|t} \hat{\mathbf{h}}_{r|t}^H + \epsilon^2 (\boldsymbol{\Sigma}_{r-1|t} + \hat{\mathbf{h}}_{r-1|t} \hat{\mathbf{h}}_{r-1|t}^H) - 2\epsilon \text{Re} \left\{ \boldsymbol{\Sigma}_{r,r-1|t} + \hat{\mathbf{h}}_{r|t} \hat{\mathbf{h}}_{r-1|t}^H \right\}$ ;  
 18     **end**  
 19     **M-step:**  $\mathbf{M}_{(t-\Delta)|t} \triangleq \boldsymbol{\Sigma}_{(t-\Delta)|t} + \hat{\mathbf{h}}_{(t-\Delta)|t} \hat{\mathbf{h}}_{(t-\Delta)|t}^H$  ;  
 20      $\alpha^{(k+1)}(i) = t \left( \sum_{r=2}^t \frac{M_{r|t}(i,i)}{(1-\epsilon^2)} + M_{(t-\Delta)|t}(i,i) \right)^{-1}$  for  $i = 1, \dots, N$  ;  
 21     Compute difference  $\triangleq \left\| \frac{1}{\boldsymbol{\alpha}^{(k+1)}} - \frac{1}{\boldsymbol{\alpha}^{(k)}} \right\|^2$ ,  $k \leftarrow k + 1$  ;  
 22 **end**  
**Output:**  $\hat{\mathbf{h}}_{t|t}$ ;  $\boldsymbol{\Sigma}_{t|t}$ ;  $\hat{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}^{(k)}$

---

mission phase at the end of the timeslot. We summarize KSBL-LTS in Algorithms 1 and 2.

Table 2.1.  
Beam alignment simulation assumptions and parameters

System Parameters	Simulation Assumptions
Number of transmit antennas $N_t$	64
Antenna spacing $d$	$\lambda/2$
SNR $\rho$ (dB)	0
Virtual channel dimension $N$	128
<b>Channel Model from Section 2.1</b>	
Sparsity (virtual domain) $N_P$	4
Channel correlation coefficient $\epsilon$	0.9931
Mobility parameter $p$	{0.01, 0.1}
Number of Monte Carlo iterations	1500
<b>Channel Model from QuaDRiGa</b>	
Carrier frequency (GHz)	28
Transmit element power (dBm)	15
Receiver track length (m)	15
Receiver velocity (m/s)	30
Number of clusters	3
Number of Monte Carlo iterations	100
Additional parameters as given for UMi scenario in [74]	

## 2.3 Numerical Results

### 2.3.1 Performance Metrics

To evaluate performance, we use the normalized beamforming gain, defined as

$$G_{BF}(t) = \frac{|\mathbf{h}^H \mathbf{f}_t^{(data)}|^2}{|\mathbf{h}^H \mathbf{f}_t^{opt}|^2}. \quad (2.32)$$

This metric conveys the accuracy of the greedily-selected beam in the data transmission phase relative to the optimal beam  $\mathbf{f}_t^{opt} \in \mathcal{F}$  and is a function of the channel estimate quality and the beam selection policy used for training. We compare performance results of the KSBL beam alignment algorithm for LTS, random beamforming, and greedy beam selection policies with the normalized beamforming gain performance of omni-directional training.

We provide beamforming gain simulated results using: a) the channel model described in Section 2.1, and b) the channel from QuaDRiGa [75]. Parameters for both simulations are listed in Table 2.1. All beamforming gain dynamic support results are for a mobility parameter of  $p = 0.01$ , with LTS and omni-directional training each having an additional curve for  $p = 0.1$  to show the respective algorithm’s sensitivity to mobility changes. Assuming a channel coherence time of 10 ms,  $p = \{0.01, 0.1\}$  represents a range of mobile speeds  $\leq 90$  km/h. While any arbitrary  $\mathcal{F}$  may be used, in the simulations we use the DFT codebook [76]. All beam selection policies except for omni-directional training use the KSBL algorithm and require a training-only, initialization phase of  $N_P$  sounding vectors to avoid singular covariance matrix estimates, as shown in Lines 2-4 of Algorithm 1.

In addition to the normalized beamforming gain, we capture the tracking performance of the algorithm. For these results, we use  $N_P = 1$  for clarity in describing the algorithm’s performance and  $p = 0.1$  since it is the more challenging path evolution case. Given  $N_P = 1$ , we are able to define the “angle” of  $\mathbf{h}_t$  as the virtual spatial angle  $\theta_{t,i_{max}}$  corresponding to the virtual path gain  $h_{t,i_{max}}$ , where  $i_{max} \in \{1, \dots, N\}$  is the virtual sector at which the magnitude of  $\mathbf{h}_t$  is maximum, or

$$i_{max} = \arg \max_{i \in \{1, \dots, N\}} |h_{t,i}|.$$

The estimated spatial angle of  $\widehat{\mathbf{h}}_t$  for a given beam selection policy is determined similarly. Before plotting the angle tracking performance, we convert all angles to physical angles using  $\phi = \sin^{-1}(\theta)$ . In this work we only provide an example realization plot of the angle tracking performance. We note, however, that the normalized beamforming gain and angle tracking are related in that more accurate channel estimates result in better performance for both. Thus, we expect beam selection policies and algorithms with higher average normalized beamforming gains to also exhibit closer tracking of the true channel on average.

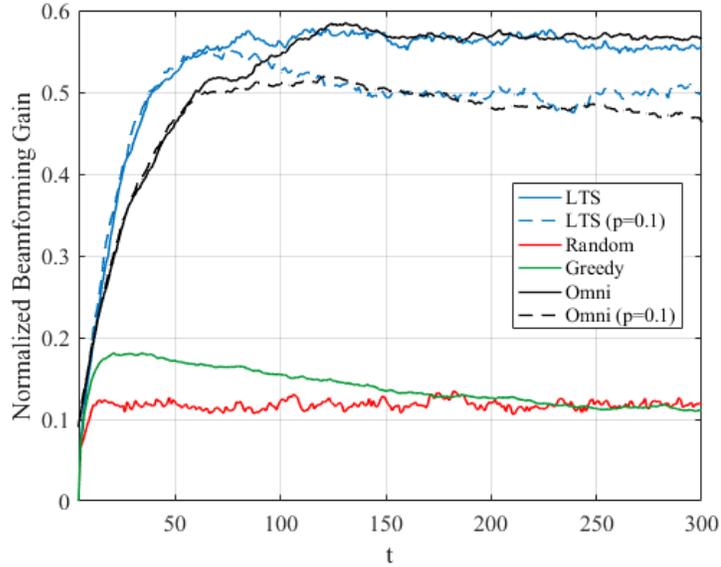


Fig. 2.2. KSBL-LTS, random beamforming, greedy and omni-directional training  $G_{BF}$  vs. time using the channel model from Section 2.1 and  $\rho = 0$  dB. All results are for a mobility parameter of  $p = 0.01$  with the exception of the LTS ( $p = 0.1$ ) and Omni ( $p = 0.1$ ) curves. These additional results show the respective algorithms' sensitivities to mobility changes for at least  $t \leq 300$  channel uses.

### 2.3.2 Normalized Beamforming Gain Results

#### Defined Channel Model

The initial beam alignment rate of KSBL-LTS is more rapid than omni-directional training. We observe in Fig. 2.2 for the channel model defined in Section 2.1 that in the data transmission phase KSBL-LTS achieves greater  $G_{BF}$  than omni-directional training for at least  $t \leq 100$ . Omni-directional training has slightly greater  $G_{BF}$  in the long-term, but this comes at the cost of slower initial beam alignment. The steady-state difference in  $G_{BF}$  between LTS and omni-directional training is only about 0.01. Random beamforming and greedy beam selection both have a much lower average  $G_{BF}$  over time.

We compare the steady-state sensitivities of LTS and omni-directional training to mobility changes. While not shown in Fig. 2.2, these steady-state behaviors hold for  $t > 300$ . LTS reaches a steady-state performance for  $p = 0.01$  of  $G_{BF} \approx 0.55$  and for  $p = 0.1$  of  $G_{BF} \approx 0.5$ . Omni-directional training reaches a steady state performance for  $p = 0.01$  of  $G_{BF} \approx 0.565$  and for  $p = 0.1$  of  $G_{BF} \approx 0.47$ . Comparing the two approaches, LTS is on average less sensitive to mobility changes, with a  $G_{BF}$  decrease of about 0.05 versus a decrease of about 0.095 for omni-directional training.

The policies using the KSBL algorithm, namely, the LTS, random, and greedy policies, all utilize an initialization phase which accounts for part of the beginning rapid  $G_{BF}$  rate. However, it must be noted that, although all three policies have similar initialization phases, LTS still has a greater  $G_{BF}$  for all  $t$  than random and greedy. This can be attributed to the guided exploration characteristic of LTS, which enables the KSBL algorithm to better track the estimate of the dynamic channel. On the other hand, the  $G_{BF}$  of random beamforming increases steadily until about  $t = 11$  after which it maintains a noisy, low level of performance around  $G_{BF} = 0.11$ . The  $G_{BF}$  of greedy beam selection also increases steadily until about channel use  $t = 20$  and maintains a low performance at  $G_{BF} \approx 0.18$  through about  $t = 40$ . After channel use  $t = 40$ , however, greedy  $G_{BF}$  gradually declines due to the greedy policy's lack exploration after converging to an estimate.

Omni-directional training does not use the KSBL algorithm and, thus, does not require the same type of initialization phase. However, the omni-directional training performance does depend on how the omni-directional training is implemented. As described in Section 2.2.1, omni-directional training can be done by cycling through individual beams at quantized angles or through individual antenna elements of the array. The normalized beamforming gain performance of both approaches over 5,000 Monte Carlo iterations is compared in Fig. 2.3. We observe that the beam approach provides steadier performance, while the antenna approach oscillates at a period of  $N_{tx} = 64$  channel uses due to the channel variation.

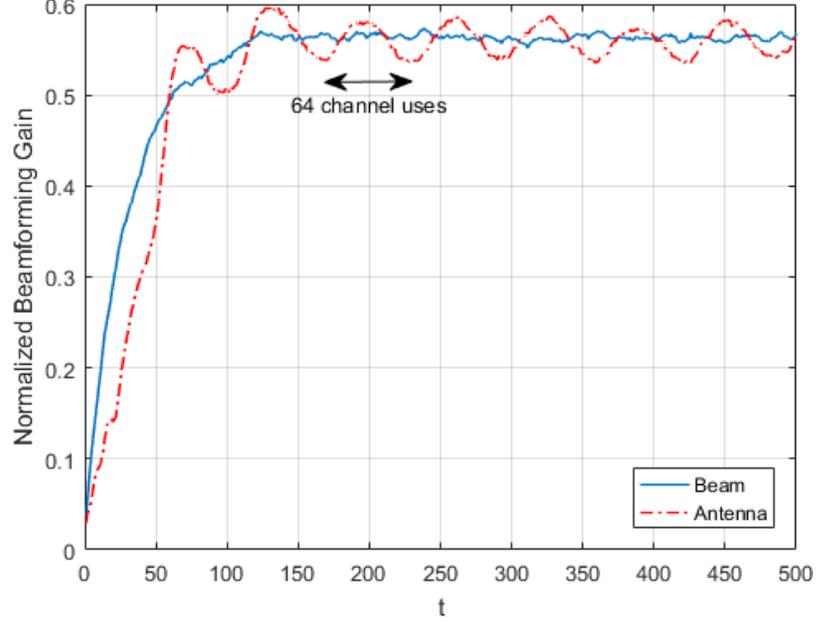


Fig. 2.3. Omni-directional training by turning on one beam at a time (Beam) and by turning on one antenna at a time (Antenna). Both results have similar performance, except cycling through antennas creates an oscillating  $G_{BF}$ .

### QuaDRiGa Channel Model

From the channel coefficients outputted by QuaDRiGa for the parameters in Table 2.1 we generate geometric channel vectors  $\mathbf{h}_t^{(g)}$  and virtual channel vectors  $\mathbf{h}_t$ . Using these channel vectors, we accomplish beam alignment by employing KSBL-LTS and omni-directional training. The  $G_{BF}$  results are averaged over independent realizations of randomly-placed receivers equipped with omni-directional antennas moving along linear tracks, as displayed in Fig. 2.4. Due to the complexity of the QuaDRiGa channel model, simulating large numbers of independent receivers is very computationally intensive, and we limited our simulation to 100 receivers. Results for the  $G_{BF}$  using the QuaDRiGa channel model are displayed in Fig. 2.5. We note that the curves are noisy due to the small number of independent realizations. As can be seen, the  $G_{BF}$  of both KSBL-LTS and omni-directional training using the

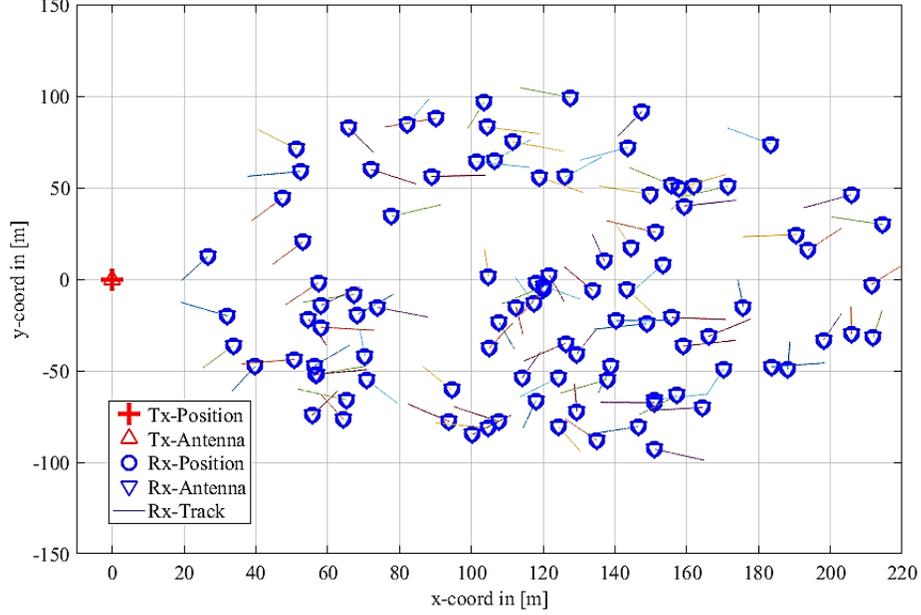


Fig. 2.4. Example from QuaDRiGa of 100 randomly placed, single antenna receivers (blue) moving in random directions along 15m linear tracks (black lines). The transmitter (red) employs an antenna array. Beamforming gain results from the receivers are averaged together and presented in Fig. 2.5.

QuaDRiGa channel model follow the same overall trend from the channel model defined in Section II. That is, KSBL-LTS has a more rapid initial rate of performance than omni-directional training, while omni-directional training has slightly better long-term average  $G_{BF}$  performance.

### 2.3.3 Example Angle Tracking Results

In Fig. 2.6 we show an example of the angle tracking performance of the various beam selection policies relative to the true channel angle for  $N_P = 1$  and  $p = 0.1$  over 300 channel uses of the channel model from Section 2.1. Random beamforming has very noisy angle tracking performance and is not included so as not to clutter the plot. In this single simulation realization, KSBL-LTS “locks on” to the true channel angle (i.e., accurately estimates the true channel) at channel use  $t = 56$ , has

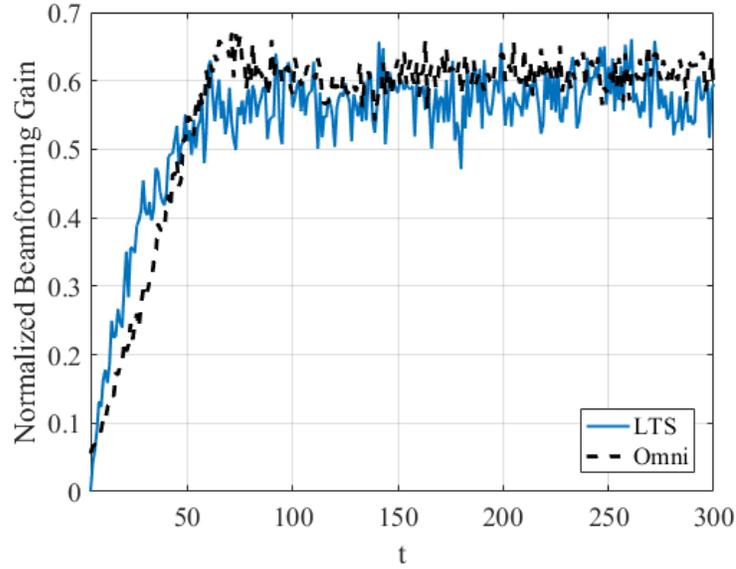


Fig. 2.5. KSBL-LTS and omni-directional training  $G_{BF}$  vs. time using the QuaDRiGa channel model.

a widely varying estimate between channel uses  $t = 116 - 132$  due to exploration in its training phase, and then “locks on” to the true channel angle again. Omni-directional training actually estimates the true channel angle very quickly at  $t = 9$  in this realization. However, at  $t = 74$  it jumps to the incorrect greedy channel angle estimate before returning to the true channel angle at  $t = 138$ . An additional period of incorrect channel angle estimates occurs between channel uses  $t = 202$  and  $t = 232$ . Greedy beam selection never finds the true channel angle and exhibits very limited exploration for the duration of the simulation despite the temporal evolution of the true mobile channel.

Throughout the simulation KSBL-LTS continues to explore during its training phase, typically leading to much quicker corrections to its channel estimation. For example, the incorrect channel angle estimates between channel uses  $t = 116 - 132$  last a much shorter period than the incorrect omni-directional channel estimates between channel uses  $t = 74 - 138$ . Overall, KSBL-LTS is generally able to respond more rapidly to a dynamic, evolving channel than can omni-directional training.

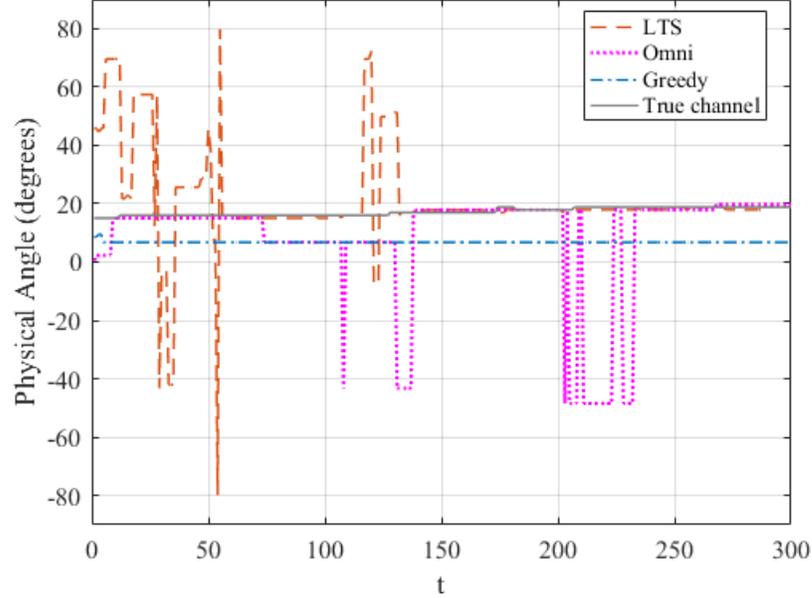


Fig. 2.6. Example of KSBL-LTS, greedy, and omni-directional training angle tracking performance over  $t = 300$  channel uses for  $N_P = 1$  and  $p = 0.1$ . In general, random beamforming tracking performance is very noisy and is not shown.

## 2.4 Conclusion

In this chapter we proposed a novel beam alignment and tracking algorithm for sparse, time-varying mmWave channels using a SBL Kalman filter with Bayesian MAB beam selection. We numerically compared the normalized beamforming gain performance of KSBL-LTS with that of other beam selection policies using theoretical and real-world channel models. Furthermore, we discussed an example realization of the angle tracking performances of the policies, especially as it pertained to the exploration evident in the KSBL-LTS and omni-directional training algorithms. Significantly, KSBL-LTS has a faster learning rate than omni-directional training for mmWave channels with a slowly time-varying channel support.

### 3. QUANTIZED MIMO MUTUAL INFORMATION SELECTION

Millimeter wave systems must utilize large, directive antenna arrays, e.g., massive multiple-input multiple-output (MIMO), in order to overcome the propagation losses inherent at high frequencies. As these arrays become larger, however, energy consumption becomes a critical issue, and the system performance must be balanced with the power use. Furthermore, the potential exists for an output data rate greater than the application layer can accommodate, in which case selectively throttling the number of bits per sample can be accomplished in the physical layer or in the application layer; we focus on the physical layer in this work. This rate constraint, whether due to power or data requirements, creates a bottleneck at the output of the receiver. For best performance, the receiver should reduce the number of output bits by only passing on the bits which provide the most information about the transmitted signal. We term this process to be mutual information selection.

Mutual information selection can be accomplished through two different methods: 1) in the analog domain through antenna subset selection using a switch as shown in Fig. 3.1, or 2) in the digital domain through subset selection of bits after quantization as shown in Fig. 3.2. While we describe both methods in greater detail below, we note that the analysis and results of both methods can be very similar. Thus, in this work we focus entirely on the second method illustrated in Fig. 3.2.

1. The first method uses an analog switch to select a subset of antennas *prior to* quantization. This enables the receiver to fix the number of required radio frequency (RF) and quantizer chains to some number  $\widetilde{M}$ , determined by the rate constraint. Denoting the total number of antennas by  $M$ , for  $\widetilde{M} < M$  the cost and power consumption of the receiver are reduced. However, the receiver

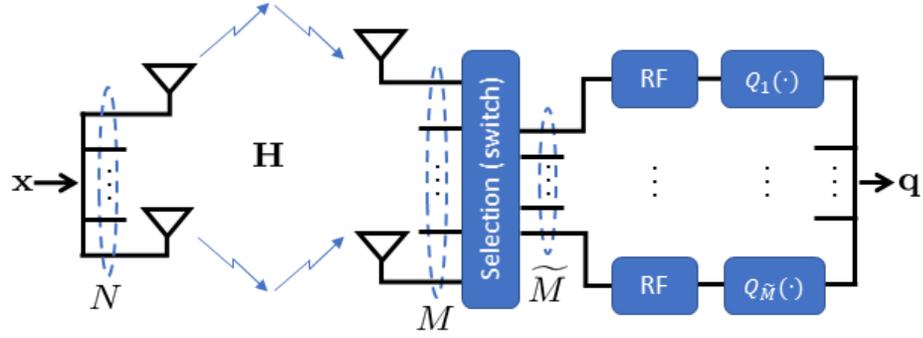


Fig. 3.1. Analog domain mutual information selection. Analog switch selects a fixed number of ADCs *before* quantization.

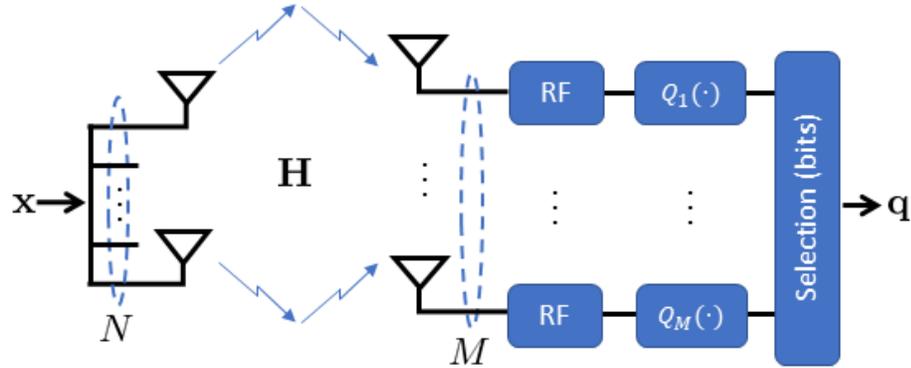


Fig. 3.2. Digital domain mutual information selection. Bits are generated using one ADC per antenna and then selected *after* quantization.

requires either delay in or storage of the signal prior to quantization and/or a dedicated selection training phase in order to determine which antennas to turn on to maximize the mutual information between the transmitted signal  $\mathbf{x}$  and the receiver output  $\mathbf{q}$ . This first method has been the focus of several previous papers in the literature for the case of *unquantized* MIMO (e.g., [46, 48, 62, 63]).

2. In the second method, we assume each antenna has a dedicated RF chain and quantizer. We note that although this assumption ignores hybrid architectures with several antennas per an RF chain, such as used in [77, 78], extensions can be made to incorporate these more complex designs. The mutual information

selection is digital, taking place *after* quantization, and is based on the quantized signals. Delay or storage of the quantized signal is assumed acceptable. Using this method allows for real-time quantization of the incoming signal. Literature in the case of *quantized* MIMO includes [49, 77–79].

Note that hybrids of the two methods are also possible. For example, we can use feedback from the digital bit subset selection to set the analog, antenna selection switch.

We comment on the apparent similarity between our categorization of mutual information selection into analog and digital domain mutual information selection methods and the architectures used by Rini *et al.* in [49]. There, the authors compare different MIMO architectures by varying antenna selection and parallel sign quantization techniques. Architecture (c) in [49] corresponds to multiple antenna selection and multilevel quantization, where potentially more than one sign quantizer is used with a single receive antenna output. Rini *et al.* term this to implicitly be a digital domain approach. Architecture (d) corresponds to linear combining and multilevel quantization and is termed to be in the analog domain. In our approaches, we only consider antenna selection and do not consider linear combining. Also, while understanding that multilevel quantization can be accomplished using multiple sign quantizers in parallel, we simply consider the number of bits used in the quantization and not the specific architecture.

In this chapter we formulate mutual information selection as a quantized MIMO antenna subset selection problem. The signal model is provided in Section 3.1. Then, in Section 3.2 we describe the quantization and the modeling of varying levels of quantization bits across the overall analog-to-digital converter (ADC). In Section 3.3 we detail the problem, which is summarized as follows. Given a rate constraint, we determine a linear selection matrix  $\mathbf{G}$ , based on the quantized received signal and channel  $\mathbf{H}$ , which returns a subset of the total bits by only choosing the outputs corresponding to specific antennas. Of all possible selection matrices, we seek ones that result in the greatest mutual information between  $\mathbf{x}$  and  $\mathbf{q}$ . We address this

problem in Section 3.4 with an optimal joint mutual information selection and by developing a greedy subselection algorithm. Additionally, we show how previously developed *unquantized* MIMO antenna subselection algorithms can be adapted for *quantized* MIMO using our signal model. Finally, in Section 3.4.4 we present and discuss numerical results from our algorithms.

### 3.1 Signal Model

Consider a MIMO system equipped with  $N$  transmit antennas and  $M$  receive antennas. At the  $m$ -th receive antenna the received signal is given by

$$y_m = \sqrt{\frac{\rho}{N}} \mathbf{h}_m^H \mathbf{x} + z_m, \quad m = 1, \dots, M \quad (3.1)$$

where  $\rho/N$  is the per-antenna, average transmit signal-to-noise ratio (SNR),  $\mathbf{h}_m \in \mathbb{C}^N$  is the channel vector between the transmitter and the  $m$ -th receive antenna, and  $z_m$  is the noise at the  $m$ -th receive antenna with assumed distribution  $\mathcal{CN}(0, \sigma_{z,m}^2)$  where  $\sigma_{z,m}^2 = 1$ . To simplify the characterization of the overall mutual information between the transmitted signal vector  $\mathbf{x} \in \mathbb{C}^N$  and the quantized output, we approximate  $\mathbf{x}$  as Gaussian-distributed, as also done in, for example, [80–82]. Specifically,  $\mathbf{x}$  is a zero mean, circularly symmetric complex Gaussian with covariance  $\mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{C}_x$  and satisfying the energy constraint  $\text{Tr}(\mathbf{C}_x) = N$ .

The overall received signal vector after RF processing is given by the input-output relationship

$$\mathbf{y} = \sqrt{\frac{\rho}{N}} \mathbf{H} \mathbf{x} + \mathbf{z}, \quad (3.2)$$

where  $\mathbf{y} = [y_1, \dots, y_M]^T \in \mathbb{C}^M$ ,  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_M]^H$ , and  $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}_M, \mathbf{I}_M)$ . Furthermore,  $\mathbf{z}$  is independent of  $\mathbf{x}$ , and  $\mathbb{E}[\mathbf{z}\mathbf{z}^H] = \mathbf{I}_M$ , i.e.,  $z_m, m = 1, \dots, M$  are spatially independent across the receive antennas. We assume channel state information (CSI) is available at the receiver. This assumption is motivated by the fact that even when quantization is very coarse (1-4 bits), it has been shown that reliable estimation is

still possible for coherence times as low as  $T_c = 1$  ms [83], well below the  $T_c = 10$  ms coherence time used in [84–86].

### 3.2 Quantization

The signal at each receive antenna  $y_m$  is connected to an ADC pair represented by the operator  $Q_m(\cdot)$ ,  $m = 1, \dots, M$ , where  $Q_m(\cdot)$  is a scalar quantizer function separately applied to the real and imaginary parts of  $y_m$ . Each ADC of the  $m$ -th ADC pair has  $B_m$  bits for quantizing each of the real and imaginary parts of  $y_m$ , where  $B_m$  is a function of the antenna index  $m$ . The quantization maps  $y_m$  to quantization points such that the quantizer output  $\hat{y}_m$  is selected as  $\hat{y}_m = \mathbb{E}[y_m | \hat{y}_m]$ . For each quantizer  $Q_m(\cdot)$ , we assume the quantization regions are non-overlapping and span  $\mathbb{C}$ . Across the receive array a parallel configuration of ADCs, defined by  $\mathcal{Q}(\cdot) = [Q_1(\cdot), \dots, Q_M(\cdot)]^\top$ , performs element-wise quantization of the received signal vector  $\mathbf{y}$ . The vector of quantized outputs is given by  $\hat{\mathbf{y}} = \mathcal{Q}(\mathbf{y})$ .

Quantization is inherently nonlinear, which complicates analysis. In Section 3.2.1 we describe the additive quantization noise model (AQNM), which we use to convert the quantization into a more analytically friendly linear operation for our analysis. This model is comprised of multiplying the input signal by a linear quantization gain and then adding independent quantization noise. The quantization gain depends on the quantizer input distribution, the scalar quantizer design and the number of quantization bits  $B_m$ . In Section 3.2.2 we explain how to obtain the values of these quantization gains. In Section 3.2.3 we derive the covariance of the received, quantized signal in terms of the quantization gains, channel vectors, and channel noise.

#### 3.2.1 Additive Quantization Noise Model (AQNM)

The nonlinearities inherent in quantization make exact analysis of mutual information selection difficult. A more analytically tractable model is obtained by using

the Bussgang theorem [87]. For Gaussian quantizer inputs, the Bussgang theorem allows us to model the non-linear quantization operation  $\hat{\mathbf{y}} = \mathcal{Q}(\mathbf{y})$  in a linear form:

$$\hat{\mathbf{y}} = \mathbf{V}_\alpha \mathbf{y} + \mathbf{w}. \quad (3.3)$$

This model is the AQNM described in [82] and has been used in several other works, e.g., [37, 51, 77–79, 88–90]. We define the quantization gain matrix  $\mathbf{V}_\alpha$  from the linear minimum mean square error (MMSE) estimation of  $\hat{\mathbf{y}}$  from  $\mathbf{y}$  such that  $\mathbb{E}[\mathbf{w}\mathbf{y}^H] = \mathbf{0}_M$  [82, 89]. Thus, we have

$$\mathbf{V}_\alpha = \mathbf{C}_{\hat{\mathbf{y}}\mathbf{y}} \mathbf{C}_\mathbf{y}^{-1}, \quad (3.4)$$

where  $\mathbf{C}_{\hat{\mathbf{y}}\mathbf{y}} = \mathbb{E}[\hat{\mathbf{y}}\hat{\mathbf{y}}^H]$  is the covariance between the quantizer input and output signals and  $\mathbf{C}_\mathbf{y} = \mathbb{E}[\mathbf{y}\mathbf{y}^H]$  is the covariance of the quantizer input signal. The quantization noise vector  $\mathbf{w}$  is zero mean with covariance matrix

$$\begin{aligned} \mathbf{C}_\mathbf{w} &= \mathbb{E}[\mathbf{w}\mathbf{w}^H] = \mathbb{E}[(\hat{\mathbf{y}} - \mathbf{V}_\alpha \mathbf{y})(\hat{\mathbf{y}} - \mathbf{V}_\alpha \mathbf{y})^H] \\ &= \mathbf{C}_{\hat{\mathbf{y}}} - \mathbf{C}_{\hat{\mathbf{y}}\mathbf{y}} \mathbf{C}_\mathbf{y}^{-1} \mathbf{C}_{\mathbf{y}\hat{\mathbf{y}}}, \end{aligned} \quad (3.5)$$

where  $\mathbf{C}_{\hat{\mathbf{y}}} = \mathbb{E}[\hat{\mathbf{y}}\hat{\mathbf{y}}^H]$ . We lower bound the mutual information in our analysis by assuming  $\mathbf{w}$  has the worst-case noise of Gaussian distributed [80, 91]. Any deviation by the noise from a Gaussian distribution improves the mutual information.

We aim to define the quantization variables  $\mathbf{V}_\alpha$  and  $\mathbf{w}$  in terms of  $\mathbf{C}_\mathbf{y}$  and the quantization loss. The quantization loss at the  $m$ -th quantizer is defined as

$$\beta_m \triangleq \frac{MSE_m}{\sigma_{y,m}^2}, \quad (3.6)$$

where  $MSE_m = \mathbb{E}[|e_m|^2]$  is the quantization mean squared-error (MSE),  $e_m = \hat{y}_m - y_m$  is the quantization error, and  $\sigma_{y,m}^2 = \mathbb{E}[|y_m|^2]$  is the variance of  $y_m$ . As we will demonstrate in Section 3.2.2, the type of quantizer, number of quantization bits, and input signal distribution determine the value of  $\beta_m$ . We consider symmetric, MMSE uniform quantizers and complex Gaussian inputs with uncorrelated real and

imaginary parts. With these conditions in place, for  $m, n \in \{1, \dots, M\}$ , we have the following correlations [82, 90]:

$$\mathbb{E}[y_m e_m^*] = -\beta_m \sigma_{y,m}^2 \quad (3.7)$$

$$\mathbb{E}[e_m e_m^*] = \beta_m \sigma_{y,m}^2 \quad (3.8)$$

Furthermore, we can obtain two additional correlations using the fact that  $e_m$  conditioned on  $y_m$  is statistically independent of all of the other random variables [92]. The first is [82, 90]

$$\begin{aligned} \mathbb{E}[y_m e_n^*] &= \mathbb{E}_{y_n} [\mathbb{E}[y_m | y_n] \mathbb{E}[e_n^* | y_n]] \\ &\stackrel{(a)}{=} \mathbb{E}_{y_n} \left[ \mathbb{E}[y_m y_n^*] \mathbb{E}[|y_n|^2]^{-1} y_n \mathbb{E}[e_n^* | y_n] \right] \\ &= \mathbb{E}[y_m y_n^*] \mathbb{E}[|y_n|^2]^{-1} \mathbb{E}[y_n e_n^*] \\ &\stackrel{(b)}{=} -\beta_n \mathbb{E}[y_m y_n^*], \forall m \neq n, \end{aligned} \quad (3.9)$$

where, since  $\mathbf{y}$  is a vector of jointly Gaussian random variables, (a) uses the linear estimator  $\mathbb{E}[y_m y_n^*] \mathbb{E}[|y_n|^2]^{-1} y_n$  corresponding to the Bayesian estimator  $\mathbb{E}[y_m | y_n]$  and (b) follows from (3.7). The final correlation we require is [82, 90]

$$\begin{aligned} \mathbb{E}[e_m e_n^*] &= \mathbb{E}_{y_n} [\mathbb{E}[e_m | y_n] \mathbb{E}[e_n^* | y_n]] \\ &\stackrel{(a)}{=} \mathbb{E}_{y_n} \left[ \mathbb{E}[e_m y_n^*] \mathbb{E}[|y_n|^2]^{-1} y_n \mathbb{E}[e_n^* | y_n] \right] \\ &= (\mathbb{E}[y_n e_m^*])^* \mathbb{E}[|y_n|^2]^{-1} \mathbb{E}[y_n e_n^*] \\ &\stackrel{(b)}{=} -\beta_m (\mathbb{E}[y_n y_m^*])^* \mathbb{E}[|y_n|^2]^{-1} (-\beta_n \sigma_{y,n}^2) \\ &\stackrel{(c)}{=} \beta_m \beta_n \mathbb{E}[y_m y_n^*], \forall m \neq n, \end{aligned} \quad (3.10)$$

where (a) uses the linear estimator of the Bayesian estimator  $\mathbb{E}[e_m | y_n]$ , (b) follows from (3.7) and (3.9), and (c) uses the fact that  $(\mathbb{E}[y_n y_m^*])^* = \mathbb{E}[y_m y_n^*]$ .

In what follows, we use the steps from [82, 90] to write the quantization gain  $\mathbf{V}_\alpha$  and noise covariance  $\mathbf{C}_w$  in (3.4) and (3.5), respectively, in terms of  $\mathbf{C}_y$  and

$\mathbf{V}_\beta \triangleq \text{diag}(\beta_1, \dots, \beta_M)$ . Note that  $\mathbf{V}_\beta$  contains only real values on the diagonal. From (3.7) and (3.9) we obtain

$$\mathbf{C}_{\mathbf{y}\mathbf{e}} = \mathbb{E} [\mathbf{y}\mathbf{e}^H] = -\mathbf{C}_\mathbf{y}\mathbf{V}_\beta, \quad (3.11)$$

where the quantization error vector  $\mathbf{e}$  relates the quantization input and output through  $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{e}$ . The result of (3.11) leads to

$$\begin{aligned} \mathbf{C}_{\mathbf{y}\hat{\mathbf{y}}} &= \mathbb{E} [\mathbf{y}(\mathbf{y} + \mathbf{e})^H] \\ &= \mathbf{C}_\mathbf{y} + \mathbf{C}_{\mathbf{y}\mathbf{e}} \\ &= \mathbf{C}_\mathbf{y} (\mathbf{I}_M - \mathbf{V}_\beta). \end{aligned} \quad (3.12)$$

Similarly,  $\mathbf{C}_{\hat{\mathbf{y}}\mathbf{y}} = \mathbf{C}_{\mathbf{y}\hat{\mathbf{y}}}^H = (\mathbf{I}_M - \mathbf{V}_\beta) \mathbf{C}_\mathbf{y}$ . Thus,

$$\mathbf{V}_\alpha = (\mathbf{I}_M - \mathbf{V}_\beta) \mathbf{C}_\mathbf{y} \mathbf{C}_\mathbf{y}^{-1} = (\mathbf{I}_M - \mathbf{V}_\beta), \quad (3.13)$$

and we define  $\alpha_m \triangleq 1 - \beta_m$ .

To simplify  $\mathbf{C}_\mathbf{w}$ , we require  $\mathbf{C}_{\hat{\mathbf{y}}} = \mathbb{E} [(\mathbf{y} + \mathbf{e})(\mathbf{y} + \mathbf{e})^H] = \mathbf{C}_\mathbf{y} + \mathbf{C}_{\mathbf{y}\mathbf{e}} + \mathbf{C}_{\mathbf{y}\mathbf{e}}^H + \mathbf{C}_\mathbf{e}$ . Using (3.8) and (3.10) we obtain

$$\begin{aligned} \mathbf{C}_\mathbf{e} &= \mathbb{E} [\mathbf{e}\mathbf{e}^H] \\ &= \mathbf{V}_\beta \text{diag}(\mathbf{C}_\mathbf{y}) + \mathbf{V}_\beta \text{offdiag}(\mathbf{C}_\mathbf{y})\mathbf{V}_\beta \\ &= \mathbf{V}_\beta \mathbf{C}_\mathbf{y} - \mathbf{V}_\beta \text{offdiag}(\mathbf{C}_\mathbf{y}) + \mathbf{V}_\beta \text{offdiag}(\mathbf{C}_\mathbf{y})\mathbf{V}_\beta \\ &= \mathbf{V}_\beta \mathbf{C}_\mathbf{y} - \mathbf{V}_\beta \text{offdiag}(\mathbf{C}_\mathbf{y}) (\mathbf{I}_M - \mathbf{V}_\beta) \\ &= \mathbf{V}_\beta \mathbf{C}_\mathbf{y} - \mathbf{V}_\beta \text{offdiag}(\mathbf{C}_\mathbf{y})\mathbf{V}_\alpha. \end{aligned} \quad (3.14)$$

With the results of (3.11) and (3.14), we have

$$\begin{aligned}
\mathbf{C}_{\hat{\mathbf{y}}} &= \mathbf{C}_{\mathbf{y}} - \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\beta} - \mathbf{V}_{\beta}\mathbf{C}_{\mathbf{y}} + \mathbf{V}_{\beta}\mathbf{C}_{\mathbf{y}} - \mathbf{V}_{\beta} \text{offdiag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{C}_{\mathbf{y}}(\mathbf{I}_M - \mathbf{V}_{\beta}) - \mathbf{V}_{\beta} \text{offdiag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} - \mathbf{V}_{\beta} \text{offdiag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} - \mathbf{V}_{\beta}\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} - (\mathbf{I}_M - \mathbf{V}_{\alpha})\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} - \mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\alpha}\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} \\
&= \mathbf{V}_{\alpha}\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha}.
\end{aligned} \tag{3.15}$$

Now we can plug in the results of (3.12) and (3.15) to simplify  $\mathbf{C}_{\mathbf{w}}$  in (3.5) as

$$\begin{aligned}
\mathbf{C}_{\mathbf{w}} &= \mathbf{C}_{\hat{\mathbf{y}}} - \mathbf{C}_{\hat{\mathbf{y}}\mathbf{y}}\mathbf{C}_{\mathbf{y}}^{-1}\mathbf{C}_{\mathbf{y}\hat{\mathbf{y}}} \\
&= \mathbf{V}_{\alpha}\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} + \mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}})\mathbf{V}_{\alpha} - \mathbf{V}_{\alpha}\mathbf{C}_{\mathbf{y}}\mathbf{C}_{\mathbf{y}}^{-1}\mathbf{C}_{\mathbf{y}}\mathbf{V}_{\alpha} \\
&= \mathbf{V}_{\alpha}\mathbf{V}_{\beta} \text{diag}(\mathbf{C}_{\mathbf{y}}).
\end{aligned} \tag{3.16}$$

With the assumptions given in this section,  $\mathbf{C}_{\mathbf{w}}$  is a diagonal matrix, as in [37, 51, 77–79, 81, 82, 93]. Note that [81] shows this simplifying approximation is fairly accurate, especially for low to medium SNRs.

### 3.2.2 Quantization Gains for Gaussian Distributed Inputs

In the previous section, quantization is modeled as a linear operation in (3.3) requiring only the input signal covariance  $\mathbf{C}_{\mathbf{y}}$  and quantization gains  $\mathbf{V}_{\alpha}$  (or, equivalently, the quantization losses  $\mathbf{V}_{\beta}$ ) to characterize the quantizer output. In this section, we describe the quantization gains for a uniform scalar quantizer as a function of the number of quantization bits  $B_m$  assuming Gaussian distributed input signals. We first provide a closed-form expression of a uniform scalar quantizer for a generic  $B_m$ . Then we explain why we use numerically derived quantization gains with a uniform scalar quantizer. We also note that, as an alternative, there exists a

simple, closed-form expression for the quantization gains of a *non-uniform* quantizer as a function of  $B_m$ . This expression can be used, if desired, to lower bound the uniform quantization error variance. However, we opt to simply use the numerically calculated values for a uniform quantizer.

For a uniform scalar quantizer with a finite set of quantization points equally spaced in a bounded interval  $[-A, A]$  and symmetric about the origin, we can determine a closed-form expression for the quantization point locations given a generic number of bits. Considering only the real domain (indicated by the superscript  $R$ ), the  $m$ -th ADC uses  $Q_m^R(\cdot)$  to quantize the real input into discrete values from a set  $\mathcal{A}$  of  $K_m^R = 2^{B_m}$  quantization points indexed by  $k = 0, \dots, K_m^R - 1$ , or  $\mathcal{A} = \{\hat{y}_{m,0}^R, \dots, \hat{y}_{m,K_m^R-1}^R\}$ , where  $\hat{y}_{m,k}^R$  is determined by

$$\hat{y}_{m,k}^R = \frac{(2k + 1 - K_m^R)A}{K_m^R}, \forall k = 0, \dots, K_m^R - 1. \quad (3.17)$$

The quantization interval length is given by  $\Delta_m = \frac{2A}{K_m^R}$ . As an example, say we have  $B_m = 2$  such that  $K_m^R = 4$ . Then, since  $\hat{y}_{m,k}^R, k = 0, \dots, 3$  are equally spaced between  $-A$  and  $A$ , the set of quantization points is  $\mathcal{A} = \{-\frac{3A}{4}, -\frac{A}{4}, \frac{A}{4}, \frac{3A}{4}\}$ . The quantization of the imaginary part of  $y_m$  follows an identical process.

Given a uniform scalar quantizer such as we just described, we minimize the MSE by adjusting  $\Delta_m$  based on  $K_m$  and the input quantization signal probability density function (PDF). When the input has a finite support, this optimization is straightforward and can result in a closed-form, exact characterization of  $MSE_m$  in terms of  $B_m$ . However, when the input has infinite support, such as in the case of the Gaussian distribution we consider in this work, the optimization becomes more complicated and even closed-form approximations of  $MSE_m$  in terms of  $B_m$  are difficult. Closed-form formulas approximating  $MSE_m$  in terms of  $B_m$  for *uniform* scalar quantizers with Gaussian inputs can be found in [94–96]. However, these approximation formulas are of sufficient length and complication such that they do not lend themselves for use in straightforward analysis. Therefore, we turn to numerically derived results. The

Table 3.1.

Uniform quantization interval lengths and error variances for different values of  $B_m$  [97]

$B_m$	1	2	3	4	5
$\Delta_m$	1.596	0.9957	0.5860	0.3352	0.1881
$\beta_m$	0.3634	0.1188	0.03744	0.01154	0.003490

Table 3.2.

Non-uniform quantization error variances (actual [97] and estimated) for different values of  $B_m$

$B_m$	1	2	3	4	5
$\beta_m$	0.3634	0.1175	0.03454	0.009497	0.002499
Est. $\beta_m$	0.6802	0.1700	0.04251	0.01063	0.002657

first numerical calculations of the MSE-optimal  $\Delta_m$  for a standard Gaussian input with  $\sigma_{y,m}^2 = 1$  were accomplished by Max in [97] for  $K_m^R = 1$  to 36. We reproduce these values for  $K_m^R = \{2, 4, 8, 16, 32\}$  in Table 3.1. Note that when  $\sigma_{y,m}^2 = 1$ , then  $\beta_m = MSE_m$ .

Alternatively, there does exist a simple, closed-form approximation relating  $\beta_m$  and  $B_m$  for MSE-optimal *non-uniform* scalar quantization of Gaussian random variables. In the non-uniform formula,  $\beta_m$  is approximated as [81, 82]

$$\beta_m \approx \frac{\pi\sqrt{3}}{2} 2^{-2B_m}. \quad (3.18)$$

Recall that  $B_m$  is the number of bits for both the real and imaginary part of  $y_m$ . The approximation in (3.18) holds for  $B_m \geq 4$ . We show this in Table 3.2 for the actual values of  $\beta_m$  numerically determined in [97] compared with the estimated values of  $\beta_m$  from (3.18). Asymptotically, i.e., at high resolution,  $\beta_m$  is equivalent for the uniform and non-uniform cases [96]. As non-uniform quantization minimum MSE for  $B_m \geq 4$  is guaranteed to be less than that of uniform quantization, (3.18) could be used as a lower bound to the uniform quantization error variance. However, as we consider

$B_m \geq 1$ , in our simulations we use the numerically calculated values of  $\beta_m$  for a given  $B_m$  shown in Table 3.1 for uniform scalar quantization.

### 3.2.3 Quantized Signal Covariance Using AQNM

Using the results of the previous two sections, we can determine the covariance of the received, quantized signal  $\mathbf{C}_{\hat{\mathbf{y}}}$  in terms of the quantization gains  $\mathbf{V}_\alpha$ , channel  $\mathbf{H}$ , and channel noise  $\mathbf{C}_{\mathbf{z}} = \mathbf{I}_M$ . We first sum the quantization noise  $\mathbf{w}$  from (3.3) with  $\mathbf{z}$  scaled by the quantization gain to form a total noise term as shown:

$$\mathbf{z}_{tot} \triangleq \mathbf{V}_\alpha \mathbf{z} + \mathbf{w}. \quad (3.19)$$

Scaling and combining the covariances, we obtain  $\mathbf{z}_{tot} \sim \mathcal{CN}(\mathbf{0}_M, \mathbf{C}_{\mathbf{z}_{tot}})$ , where

$$\begin{aligned} \mathbf{C}_{\mathbf{z}_{tot}} &= \mathbf{V}_\alpha \mathbf{C}_{\mathbf{z}} \mathbf{V}_\alpha^\top + \mathbf{C}_{\mathbf{w}} \\ &= \mathbf{V}_\alpha (\mathbf{V}_\alpha + \mathbf{V}_\beta \text{diag}(\mathbf{C}_{\mathbf{y}})) \\ &= \begin{bmatrix} \sigma_{z_{tot},1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{z_{tot},M}^2 \end{bmatrix}, \end{aligned} \quad (3.20)$$

and

$$\sigma_{z_{tot},m}^2 = \alpha_m^2 + \alpha_m \beta_m \left( \frac{\rho}{N} \mathbf{h}_m^\mathbf{H} \mathbf{C}_{\mathbf{x}} \mathbf{h}_m + 1 \right), \quad m = \{1, \dots, M\}. \quad (3.21)$$

Note that since  $\mathbf{V}_\alpha$  is a diagonal matrix  $\mathbf{V}_\alpha = \mathbf{V}_\alpha^\top$ . With  $\mathbf{z}_{tot}$  thus defined, we can rewrite the system model as

$$\hat{\mathbf{y}} = \sqrt{\frac{\rho}{N}} \mathbf{V}_\alpha \mathbf{H} \mathbf{x} + \mathbf{z}_{tot}. \quad (3.22)$$

Given  $\mathbf{x}$  and  $\mathbf{z}$  are both marginally Gaussian and independent, the received signal  $\mathbf{y}$  is also Gaussian with distribution  $\mathbf{y} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C}_{\mathbf{y}})$  where  $\mathbf{C}_{\mathbf{y}} = \frac{\rho}{N} \mathbf{H} \mathbf{C}_{\mathbf{x}} \mathbf{H}^\mathbf{H} + \mathbf{I}_M$ .

Therefore, the covariance of the quantized, received signal  $\hat{\mathbf{y}}$  given  $\mathbf{H}$  can be written as

$$\begin{aligned}\mathbf{C}_{\hat{\mathbf{y}}} &= \mathbf{V}_\alpha \mathbf{C}_y \mathbf{V}_\alpha^\top + \mathbf{C}_w \\ &= \frac{\rho}{N} \mathbf{V}_\alpha \mathbf{H} \mathbf{C}_x \mathbf{H}^\mathbf{H} \mathbf{V}_\alpha^\top + \mathbf{C}_{z_{\text{tot}}} \\ &= \frac{\rho}{N} \mathbf{V}_\alpha \mathbf{H} \mathbf{C}_x \mathbf{H}^\mathbf{H} \mathbf{V}_\alpha^\top + \mathbf{V}_\alpha \mathbf{V}_\alpha^\top + \mathbf{V}_\alpha \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^\mathbf{H} + \mathbf{I}_M \right),\end{aligned}\tag{3.23}$$

where the total noise term covariance  $\mathbf{C}_{z_{\text{tot}}}$  is defined in (3.20). We can see how the number of quantization bits  $B_m$  is used in our signal model by considering just the quantizer output variances  $\sigma_{\hat{y},m}^2$  on the diagonal of  $\mathbf{C}_{\hat{\mathbf{y}}}$ . Specifically, for  $m = 1, \dots, M$  and a given  $\mathbf{h}_m$ ,

$$\begin{aligned}\sigma_{\hat{y},m}^2 &= \alpha_m^2 \left( \frac{\rho}{N} \mathbf{h}_m^\mathbf{H} \mathbf{C}_x \mathbf{h}_m + 1 \right) + \alpha_m \beta_m \left( \frac{\rho}{N} \mathbf{h}_m^\mathbf{H} \mathbf{C}_x \mathbf{h}_m + 1 \right) \\ &= \alpha_m (\alpha_m + \beta_m) \left( \frac{\rho}{N} \mathbf{h}_m^\mathbf{H} \mathbf{C}_x \mathbf{h}_m + 1 \right) \\ &= (1 - \beta_m) \left( \frac{\rho}{N} \mathbf{h}_m^\mathbf{H} \mathbf{C}_x \mathbf{h}_m + 1 \right),\end{aligned}\tag{3.24}$$

where we have used the fact  $\alpha_m + \beta_m = 1$ . Thus, since  $\beta_m$  can be found in Table 3.1 for  $m = 1, \dots, 5$ , we can model varying numbers of quantization bits  $B_m$  at each receiver branch. The off-diagonal terms of  $\mathbf{C}_{\hat{\mathbf{y}}}$  are similarly related to  $B_m$ .

We finish this section by defining a few variables which are used in subsequent sections. A one-to-one mapping encodes  $\hat{y}_m$  into a  $2B_m$ -bit binary vector  $\mathbf{b}_m$ , written as

$$\mathbf{b}_m = [b_{m,1}, \dots, b_{m,2B_m}]^\top,$$

where  $\{b_{m,1}, \dots, b_{m,2B_m}\} \in \{0, 1\}$  represent individual bits. Thus,  $\hat{y}_m$  represents the quantized decimal value of  $y_m$ , while  $\mathbf{b}_m$  represents the quantized binary value of  $y_m$ . The total number of quantization bits across all quantizers is denoted by  $J =$

$\sum_{m=1}^M (2B_m)$ , and the binary quantization output of the received vector  $\mathbf{y}$  is written as

$$\begin{aligned} \mathbf{b} = \mathcal{Q}_{bin}(\mathbf{y}) &= \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix} \\ &= \left[ b_{1,1}, \dots, b_{1,2B_1}, \dots, b_{M,1}, \dots, b_{M,2B_M} \right]^\top. \end{aligned} \quad (3.25)$$

An illustration of the signal flow through the receiver is shown in Fig. 3.3 in Section 3.3.

### 3.3 Problem Setup

In the scenario we consider, that is, massive MIMO with prohibitively large data rates at the output of the receiver, we must constrain the receiver output data rate so that only up to  $T$  bits per sample are processed, where  $T < J$  is some threshold determined by the application using the receiver. We assume the reduction in bits from the quantization stage to the receiver output is accomplished by simple subselection. This means a subset of bits from the quantization are forwarded on to the application and the rest are thrown away; no other bit operations are performed. This is illustrated in Fig. 3.3 and explained in detail in the next paragraph.

The ADC outputs the  $J$ -bit vector  $\mathbf{b}$ . We assume the channel matrix  $\mathbf{H}$  is known at the receiver. As a generalization of subselection, we consider a linear transformation matrix  $\mathbf{G}$  on bits  $\mathbf{b}$ , where  $\mathbf{G}$  is determined based on  $\mathbf{H}$ . We denote the subselected output data rate by  $L$ , where  $L \leq T$  bits. Thus,  $\mathbf{G}$  is a  $L \times J$  matrix and selects  $L$  bits from  $\mathbf{b}$  for inclusion in a bit vector  $\mathbf{q}$  at the final receiver output. Let  $\mathcal{L}$  be the set of  $L \times J$  matrices modified from the identity matrix  $\mathbf{I}_J$  by removing  $J - L$  rows  $r_j \in \{j : j = 1, \dots, J\}$ . Then, for  $\mathbf{G} \in \mathcal{L}$ , the relationship between  $\mathbf{b}$  and  $\mathbf{q}$  is written as

$$\mathbf{q} = \mathbf{G}\mathbf{b}. \quad (3.26)$$

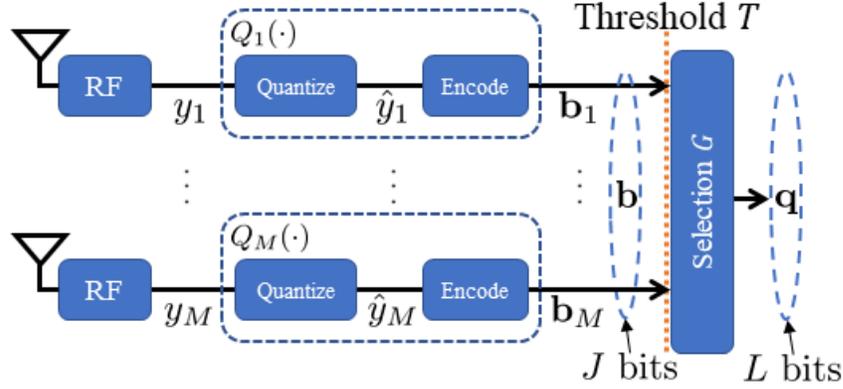


Fig. 3.3. Receiver block diagram of receiver signal flow and bit subset selection after quantization. The parallel ADC outputs a total of  $J$  bits in vector  $\mathbf{b}$ . The application requires the output data rate be no greater than some threshold  $T$ . After subselection the receiver outputs the  $L$ -bit vector  $\mathbf{q}$ , where  $L \leq T < J$ .

The problem we address is to determine  $\mathbf{G}$  for a given  $\mathbf{H}$  such that the mutual information between  $\mathbf{x}$  and  $\mathbf{q}$  is maximized under the constraint  $T$ , or

$$\mathbf{G} = \arg \max_{\substack{\mathbf{G}' \in \mathcal{L} \\ \text{s.t. } L \leq T}} I(\mathbf{x}; \mathbf{q} | \mathbf{H}). \quad (3.27)$$

The threshold  $T$  forces a down selection from  $J$  bits in  $\mathbf{b}$  to  $L$  bits in  $\mathbf{q}$ . This creates a “bottleneck” on the maximum  $I(\mathbf{x}; \mathbf{q} | \mathbf{H})$  possible since  $T < J$  and fewer bits implies less information. Let the indices of quantizers from which bits are selected be given by  $i_v, v = 1, \dots, \widetilde{M}$ , where  $i_v \in \{1, \dots, M\}$  and  $\widetilde{M} \leq M$  is the number of quantizers from which bits are selected. Therefore, we can write total number of selected bits as  $L = \sum_{v=1}^{\widetilde{M}} B_{i_v}$ , where  $L \leq T$ . Once the quantizer indices are chosen, the output of each selected quantizer may be further refined. In general each quantizer output  $\mathbf{b}_m$  can have anywhere from zero up to  $B_m$  bits selected from it. The number of bits selected from each  $\mathbf{b}_m$  need not be equivalent. We denote by  $\mathbf{q}_v, v = 1, \dots, \widetilde{M}$

the vector of bits selected from  $\mathbf{b}_{i_v}$ , and the combined sub-selected receiver output is given by

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_{\tilde{M}} \end{bmatrix}, \quad (3.28)$$

where the number of bits in  $\mathbf{q}_v$  ranges from one to  $B_m$  and may or may not be equal in each  $\mathbf{q}_v$ , subject to the constraint  $T$ .

We use mutual information to determine which bits from  $\mathbf{b}$  to forward on and which to throw away. We do so by selecting  $\mathbf{G}$  under the output rate constraint  $T$  such that  $I(\mathbf{x}; \mathbf{q} | \mathbf{H})$  is maximized. Two methods of bit selection from  $\mathbf{b}$  can be considered:

1. *Bit Subselection*: The most general method for determining  $\mathbf{G}$  is to allow selection of anywhere from one to  $L \leq T$  individual bits from the overall ADC output bit vector  $\mathbf{b}$  regardless from which quantizer the bits originate. Thus, from each receive antenna's quantized output  $\mathbf{b}_m$ , zero to  $B_m$  bits may be selected and may vary from quantizer to quantizer. Due to the selection of zero to  $B_m$  bits from a given quantizer, this method is significantly more complicated, and we leave its consideration to future research.
2. *Receive Antenna Subselection*: In this method we set  $\mathbf{G}$  based on  $\mathbf{H}$  such that exclusively all or none of the bits from an antenna's quantized signal  $\mathbf{b}_m$  are included in  $\mathbf{q}$ . Equivalently, we can define a  $\tilde{M} \times M$  antenna subselection matrix  $\tilde{\mathbf{G}} \in \tilde{\mathcal{L}}$ , where  $\tilde{\mathcal{L}}$  is the set of  $\tilde{M} \times M$  identity matrices modified similar to the previously defined set  $\mathcal{L}$ . In our notation, we differentiate between variables related to bit subselection, e.g.,  $\mathbf{G}$ , and variables related to antenna subselection by adding a tilde to the variable. For example, in addition to  $\tilde{M}$ ,  $\tilde{\mathbf{G}}$ , and  $\tilde{\mathcal{L}}$ , we denote the  $L \times N$  matrix of selected channel vectors by

$$\tilde{\mathbf{H}} = \tilde{\mathbf{G}}\mathbf{H}, \quad (3.29)$$

and the subselected vector of quantized values by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{G}}\hat{\mathbf{y}}. \quad (3.30)$$

The selected indices are given by

$$\begin{bmatrix} i_1 \\ \vdots \\ i_{\tilde{M}} \end{bmatrix} = \tilde{\mathbf{G}} \begin{bmatrix} 1 \\ \vdots \\ M \end{bmatrix}. \quad (3.31)$$

Whether we use  $\mathbf{G}$  or  $\tilde{\mathbf{G}}$ ,  $\mathbf{q}$  consists of  $\tilde{M}$  selected bit vectors and  $L = \sum_{v=1}^{\tilde{M}} B_{i_v}$  total bits, where  $i_v \in \mathcal{M} = \{1, \dots, M\}$  and, again,  $L \leq T$ . Noting  $\mathbf{q}_v = \mathbf{b}_{i_v}$ ,  $v = 1 \dots, \tilde{M}$  and rewriting (3.28), we have

$$\mathbf{q} = \text{vec} \left( \tilde{\mathbf{G}} \begin{bmatrix} \mathbf{b}_1^\top \\ \vdots \\ \mathbf{b}_M^\top \end{bmatrix} \right) = \begin{bmatrix} \mathbf{b}_{i_1} \\ \vdots \\ \mathbf{b}_{i_{\tilde{M}}} \end{bmatrix}. \quad (3.32)$$

At first glance, this method may seem equivalent to the many antenna selection approaches in the literature, e.g., [46–48]. However, due to the varying numbers of bits at each quantizer, a mutual information selection algorithm not only must select the quantized outputs returning the most mutual information about the transmitted signal but must also account for how many bits are used to quantize each antenna’s received signal. Our mutual information selection problem relates antenna selection with the bit allocation problem, similar to [79].

We are only considering subselection antenna by antenna (versus by individual bits as discussed in Section 3.3). The quantized subselected decimal values are given by  $\tilde{\mathbf{y}} = [\hat{y}_{i_1}, \dots, \hat{y}_{i_{\tilde{M}}}]$ , and there exists a one-to-one mapping between  $\tilde{\mathbf{y}}$  and  $\mathbf{q}$ . When

stating the subselected results in terms of Gaussian distributions we use  $\tilde{\mathbf{y}}$  instead of the subselected bit vectors  $\mathbf{q}$ .

In the what follows, we discuss mutual information selection algorithms under the assumption of a Gaussian distributed input signal and derive expressions for the mutual information using joint and greedy selection algorithms.

### 3.4 Mutual Information Selection - Gaussian Approximation

#### 3.4.1 Joint Mutual Information

Having defined the system model and setup the problem, we now turn to finding the mutual information between the transmitted vector  $\mathbf{x}$  and the subselected bit vector  $\mathbf{q}$ . We represent the subselection through multiplication by the matrix  $\tilde{\mathbf{G}} \in \tilde{\mathcal{L}}$ , where  $\tilde{\mathcal{L}}$  is the set of  $\tilde{M} \times M$  matrices modified from the identity matrix as described in Section 3.3. The optimal solution finds the matrix  $\tilde{\mathbf{G}}^{opt} \in \tilde{\mathcal{L}}$  which returns the maximum joint mutual information  $I(\mathbf{x}; \mathbf{q} | \mathbf{H})$  (or, equivalently,  $I(\mathbf{x}; \tilde{\mathbf{y}} | \mathbf{H})$ ) under some overall output rate constraint  $T$ .

After quantization we can use (3.22) and write the mutual information between  $\mathbf{x}$  and  $\hat{\mathbf{y}}$  as follows:

$$\begin{aligned} I(\mathbf{x}; \hat{\mathbf{y}} | \mathbf{H}) &= h(\hat{\mathbf{y}} | \mathbf{H}) - h(\hat{\mathbf{y}} | \mathbf{x}, \mathbf{H}) \\ &= h(\hat{\mathbf{y}} | \mathbf{H}) - h(\mathbf{z}_{tot} | \mathbf{H}) \\ &= \log((\pi e)^M \det(\mathbf{C}_{\hat{\mathbf{y}}})) - \log((\pi e)^M \det(\mathbf{C}_{\mathbf{z}_{tot}})). \end{aligned} \tag{3.33}$$

We can write  $\mathbf{C}_{\mathbf{z}_{tot}}$  as

$$\begin{aligned} \mathbf{C}_{\mathbf{z}_{tot}} &= \mathbf{V}_\alpha \mathbf{V}_\alpha^\top + \mathbf{V}_\alpha \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^\mathbf{H} + \mathbf{I}_M \right) \\ &= \mathbf{V}_\alpha \left( \mathbf{V}_\alpha + \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^\mathbf{H} + \mathbf{I}_M \right) \right) \\ &= \mathbf{V}_\alpha \mathbf{D}_M, \end{aligned} \tag{3.34}$$

where

$$\begin{aligned}
\mathbf{D}_M &= \mathbf{V}_\alpha + \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^H + \mathbf{I}_M \right) \\
&\stackrel{(a)}{=} \mathbf{V}_\alpha + \mathbf{V}_\beta + \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^H \right) \\
&\stackrel{(b)}{=} \mathbf{I}_M + \mathbf{V}_\beta \text{diag} \left( \frac{\rho}{N} \mathbf{H} \mathbf{C}_x \mathbf{H}^H \right).
\end{aligned} \tag{3.35}$$

In (a)–(b), respectively, we used the facts that  $\mathbf{V}_\beta$  is diagonal and that  $\mathbf{V}_\alpha = \mathbf{I}_M - \mathbf{V}_\beta$ . Recalling from (3.23) that  $\mathbf{C}_{\hat{\mathbf{y}}} = \frac{\rho}{N} \mathbf{V}_\alpha \mathbf{H} \mathbf{C}_x \mathbf{H}^H \mathbf{V}_\alpha^T + \mathbf{C}_{\mathbf{z}_{\text{tot}}}$ , the mutual information becomes

$$\begin{aligned}
I(\mathbf{x}; \hat{\mathbf{y}} \mid \mathbf{H}) &= \frac{\log \det (\mathbf{C}_{\hat{\mathbf{y}}})}{\log \det (\mathbf{C}_{\mathbf{z}_{\text{tot}}})} \\
&\stackrel{(a)}{=} \log \det \left( \left( \frac{\rho}{N} \mathbf{V}_\alpha \mathbf{H} \mathbf{C}_x \mathbf{H}^H \mathbf{V}_\alpha^T + \mathbf{C}_{\mathbf{z}_{\text{tot}}} \right) \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-1} \right) \\
&\stackrel{(b)}{=} \log \det \left( \frac{\rho}{N} \mathbf{V}_\alpha \mathbf{D}_M^{-1} \mathbf{H} \mathbf{C}_x \mathbf{H}^H + \mathbf{I}_M \right),
\end{aligned} \tag{3.36}$$

where in (a) we use the fact that  $\frac{\det \mathbf{A}}{\det \mathbf{B}} = \det \mathbf{A} \mathbf{B}^{-1}$  and in (b) the fact that  $\mathbf{C}_{\mathbf{z}_{\text{tot}}} = \mathbf{V}_\alpha \mathbf{D}_M$  is diagonal.

We incorporate antenna subselection into the equation using  $\tilde{\mathbf{G}}$ . Recall  $\tilde{\mathbf{G}}$  is the identity matrix modified by removing the unselected rows  $\{1, \dots, M\} \setminus \{i_1, \dots, i_{\tilde{M}}\}$ . Using the tilde notation introduced in the previous section, we define the following  $\tilde{M} \times \tilde{M}$  matrices:

$$\begin{aligned}
\tilde{\mathbf{V}}_{\alpha_{\tilde{M}}} &\triangleq \text{diag} \left( \tilde{\mathbf{G}}[\alpha_1, \dots, \alpha_M]^T \right) \\
\tilde{\mathbf{V}}_{\beta_{\tilde{M}}} &\triangleq \text{diag} \left( \tilde{\mathbf{G}}[\beta_1, \dots, \beta_M]^T \right) \\
\tilde{\mathbf{D}}_{\tilde{M}} &\triangleq \mathbf{I}_{\tilde{M}} + \tilde{\mathbf{V}}_{\beta_{\tilde{M}}} \text{diag} \left( \frac{\rho}{N} \tilde{\mathbf{H}} \mathbf{C}_x \tilde{\mathbf{H}}^H \right).
\end{aligned}$$

Recall  $\tilde{\mathbf{H}} = \tilde{\mathbf{G}} \mathbf{H}$  is an  $\tilde{M} \times N$  matrix. Now we can write the subselected mutual information as

$$I(\mathbf{x}; \tilde{\mathbf{y}} \mid \mathbf{H}) = \log \det \left( \mathbf{I}_{\tilde{M}} + \frac{\rho}{N} \tilde{\mathbf{V}}_{\alpha_{\tilde{M}}} \tilde{\mathbf{D}}_{\tilde{M}}^{-1} \tilde{\mathbf{H}} \mathbf{C}_x \tilde{\mathbf{H}}^H \right). \tag{3.37}$$

The optimal antenna subselection matrix  $\tilde{\mathbf{G}}^{opt}$  for the joint mutual information selection is found by

$$\tilde{\mathbf{G}}^{opt} = \arg \max_{\substack{\tilde{\mathbf{G}} \in \tilde{\mathcal{L}} \\ \text{s.t. } L \leq T}} I(\mathbf{x}; \tilde{\mathbf{y}} \mid \mathbf{H}), \quad (3.38)$$

where  $L = \sum_{v=1}^{\tilde{M}} B_{i_v}$ . Equivalently, we can write this expression as a joint selection of antenna indices, under the same constraints, by

$$\{i_1, \dots, i_{\tilde{M}}\} = \arg \max_{\{1, \dots, M\}} I(\mathbf{x}; \tilde{\mathbf{y}} \mid \mathbf{H}).$$

With Gaussian approximations and antenna selection, this problem formulation appears similar to many antenna selection algorithms in the literature. However, as noted in Section 3.3, due to the quantization and the potential for varying numbers of bits at each quantizer, joint mutual information selection must not only account for the most informative channel vectors but also for how many bits each quantizer uses. Unfortunately, exhaustive search is the only method known to the authors for finding the optimal solution. An exhaustive search results in  $\binom{M}{\tilde{M}}$  possible solutions and is infeasible for large  $M$ . This motivates the need for sub-optimal approaches, such as the greedy algorithm, which we consider next.

### 3.4.2 Greedy Algorithm

In a greedy algorithm we sequentially choose the quantized outputs which maximize the mutual information at each selection. This approach is sub-optimal for not considering the correlations between antennas. As we will show, instead of using the full covariance matrices employed in the joint selection expression, the greedy approach only uses the variance terms from the diagonals of the covariance matrices. However, greedy algorithms are straightforward and computationally efficient.

In our approach, we begin with  $i_1$ , then  $i_2$ , and so on to  $i_{\widetilde{M}}$ , and we successively perform antenna subselection based on

$$i_v = \arg \max_{i_v \in \{1, \dots, M\} \setminus \{i_1, \dots, i_{v-1}\}} I(\mathbf{x}; [\hat{y}_{i_1}, \dots, \hat{y}_{i_v}] | \mathbf{H}). \quad (3.39)$$

We want to write this selection equation in terms of the channel, the transmitted signal covariance, the number of quantization bits, and the noise variance (recall  $\sigma_{z,m}^2 = 1$ ). To do so, we first use the chain rule for information [80], which allows us to write the total mutual information after  $\widetilde{M}$  selections as

$$I(\mathbf{x}; [\hat{y}_{i_1}, \dots, \hat{y}_{i_{\widetilde{M}}}] | \mathbf{H}) = \sum_{v=1}^{\widetilde{M}} I(\mathbf{x}; \hat{y}_{i_v} | \hat{y}_{i_1}, \dots, \hat{y}_{i_{v-1}}, \mathbf{H}). \quad (3.40)$$

Combining the previous two equations and explicitly writing out the successive selection equations, we have

$$\begin{aligned} i_1 &= \arg \max_{i \in \{1, \dots, M\}} I(\mathbf{x}; \hat{y}_i | \mathbf{H}) \\ i_2 &= \arg \max_{i \in \{1, \dots, M\} \setminus \{i_1\}} \{I(\mathbf{x}; \hat{y}_{i_1} | \mathbf{H}) + I(\mathbf{x}; \hat{y}_i | \hat{y}_{i_1}, \mathbf{H})\} \\ &\vdots \\ i_v &= \arg \max_{i \in \{1, \dots, M\} \setminus \{i_1, \dots, i_{v-1}\}} \{I(\mathbf{x}; \hat{y}_{i_1} | \mathbf{H}) + I(\mathbf{x}; \hat{y}_{i_2} | \hat{y}_{i_1}, \mathbf{H}) + \dots \\ &\quad + I(\mathbf{x}; \hat{y}_i | [\hat{y}_{i_1}, \dots, \hat{y}_{i_{v-1}}], \mathbf{H})\}, \end{aligned} \quad (3.41)$$

for  $v = 1, \dots, \widetilde{M}$ .

We clearly see that each succeeding selection incorporates the mutual information of the preceding selections. However, these past values are constant with respect to

the current selection. Thus, the selection equations in (3.41) can be simplified as follows:

$$\begin{aligned}
i_1 &= \arg \max_{i \in \{1, \dots, M\}} I(\mathbf{x}; \hat{y}_i \mid \mathbf{H}) \\
i_2 &= \arg \max_{i \in \{1, \dots, M\} \setminus \{i_1\}} I(\mathbf{x}; \hat{y}_i \mid \hat{y}_{i_1}, \mathbf{H}) \\
&\vdots \\
i_v &= \arg \max_{i \in \{1, \dots, M\} \setminus \{i_1, \dots, i_{v-1}\}} I(\mathbf{x}; \hat{y}_i \mid [\hat{y}_{i_1}, \dots, \hat{y}_{i_{v-1}}], \mathbf{H}).
\end{aligned} \tag{3.42}$$

Let us examine the selection of the first index. Following an approach similar to that used for the joint selection mutual information expression in (3.36), we obtain

$$I(\mathbf{x}; \hat{y}_{i_1} \mid \mathbf{H}) = \log \left| 1 + \frac{\rho}{N} \frac{\alpha_{i_1}}{d_{i_1}} \mathbf{h}_{i_1}^H \mathbf{C}_x \mathbf{h}_{i_1} \right|,$$

where  $d_{i_v}$  is the located on the  $i_v$ -th position of the diagonal on  $D_M$  in (3.35), or

$$d_{i_v} = 1 + \frac{\rho}{N} \beta_{i_v} \mathbf{h}_{i_v}^H \mathbf{C}_x \mathbf{h}_{i_v}.$$

The first index selected is the one which maximizes  $I(\mathbf{x}; \hat{y}_{i_1} \mid \mathbf{H})$ . This result shows the first greedy antenna selection is a function of the channel and the quantization effects, through the quantization gain  $\alpha_{i_v}$  and normalized quantization error variance  $\beta_{i_v}$ . As will be shown, this result also holds for subsequent selections.

To obtain selection equations for  $v > 1$ , we generalize an approach taken in [51]. In [51] the authors apply quantization to the Fast Antenna Selection (FAS) algorithm developed in [46]. They dub the quantized version of FAS as Quantization Aware FAS (QAFAS). However, the authors in [51] assume the same number of quantization bits at each quantizer, i.e.,  $B_m = B, \forall m \in \{1, \dots, M\}$ , in their analysis. On the other hand, our proposed algorithm, which we call *modified QAFAS*, is more general in that

we allow for varying numbers of bits  $B_m$  across all of the  $M$  quantizers. If we select  $v \leq M$  antennas, the joint mutual information from (3.36) can be written as

$$\begin{aligned} I(\mathbf{x}; [\hat{y}_{i_1}, \dots, \hat{y}_{i_v}] | \mathbf{H}) &= \log \det \left( \mathbf{I}_N + \frac{\rho}{N} \tilde{\mathbf{H}}_v^H (\tilde{\mathbf{V}}_{\alpha_v} \tilde{\mathbf{D}}_v^{-1}) \mathbf{C}_x \tilde{\mathbf{H}}_v \right) \\ &= \log \det \left( \mathbf{I}_N + \frac{\rho}{N} \left( \tilde{\mathbf{H}}_{v-1}^H (\tilde{\mathbf{V}}_{\alpha_{v-1}} \tilde{\mathbf{D}}_{v-1}^{-1}) \mathbf{C}_x \tilde{\mathbf{H}}_{v-1} + \frac{\alpha_{i_v}}{d_{i_v}} \mathbf{h}_{i_v} \mathbf{h}_{i_v}^H \right) \right), \end{aligned} \quad (3.43)$$

where for  $v$  selected antennas  $\tilde{\mathbf{H}}_v$  is the  $v \times N$  matrix of channel vectors,  $\tilde{\mathbf{V}}_{\alpha_v}$  is the  $v \times v$  diagonal matrix of quantization gains, and  $\tilde{\mathbf{D}}_v$  is the  $v \times v$  diagonal matrix with  $d_{i_1}, \dots, d_{i_v}$  on the diagonal.

Using the matrix determinant lemma, we can write the mutual information at the  $v$ -th antenna selection as

$$I(\mathbf{x}; [\hat{y}_{i_1}, \dots, \hat{y}_{i_v}] | \mathbf{H}) = I(\mathbf{x}; [\hat{y}_{i_1}, \dots, \hat{y}_{i_{v-1}}] | \mathbf{H}) + \log \det \left( 1 + \frac{\rho}{N} \frac{\alpha_{i_v}}{d_{i_v}} c_{i_v} \right), \quad (3.44)$$

where we define

$$c_{i_v} = \mathbf{h}_{i_v}^H \left( \mathbf{I}_N + \frac{\rho}{N} \tilde{\mathbf{H}}_{v-1}^H (\tilde{\mathbf{V}}_{\alpha_{v-1}} \tilde{\mathbf{D}}_{v-1}^{-1}) \mathbf{C}_x \tilde{\mathbf{H}}_{v-1} \right)^{-1} \mathbf{h}_{i_v}.$$

The last term of (3.44) corresponds to mutual information being maximized in the  $v$ -th selection equation in (3.42), or

$$I(\mathbf{x}; \hat{y}_{i_v} | [\hat{y}_{i_1}, \dots, \hat{y}_{i_{v-1}}], \mathbf{H}) = \log \det \left( 1 + \frac{\rho}{N} \frac{\alpha_{i_v}}{d_{i_v}} c_{i_v} \right). \quad (3.45)$$

Thus, we can reduce the selection to the following objective function:

$$i_v = \arg \max_{i \in \{1, \dots, M\} \setminus \{i_1, \dots, i_{v-1}\}} \frac{\alpha_i}{d_i} c_i. \quad (3.46)$$

We comment that in [51] the authors assume all quantizers have the same number of quantization bits and, therefore, they need only maximize over  $c_i/d_i$ , thereby omit-

ting the quantization effect of  $\alpha_i$ . Our more general objective function, on the other hand, assumes varying numbers of quantization bits at the quantizers and, thus, must include  $\alpha_i$  in the numerator. From (3.46), we see the optimization is a function of the channel vectors and the quantization resolution.

A formula for efficiently updating  $c_i$  can be found using the matrix inversion lemma, as shown in [51]. After all  $\widetilde{M}$  antennas are selected, the greedy selection matrix  $\widetilde{\mathbf{G}}_g$  is found by removing rows  $\{1, \dots, M\} \setminus \{i_1, \dots, i_{\widetilde{M}}\}$  from an identity matrix  $\mathbf{I}_M$ . Each  $\hat{y}_{i_v}$  has binary representation  $\mathbf{b}_{i_v}$ . All  $B_{i_v}$  bits associated with each  $\mathbf{b}_{i_v}, v = 1, \dots, \widetilde{M}$  are included in the output  $L$ -bit vector  $\mathbf{q}$  (recall that  $L = \sum_{v=1}^{\widetilde{M}} B_{i_v}$ ).

This greedy approach assumes the system model using the combined error term in (3.22). Furthermore, it is, for the most part, derived specifically from the unquantized MIMO antenna selection algorithm in [46] and is not generally applicable to converting other unquantized MIMO antenna selection algorithms to their quantized counterparts. In the next subsection we address how we can adapt our system model in (3.22) to be generally applicable to converting most *unquantized* MIMO antenna selection algorithms to their *quantized* MIMO versions.

### 3.4.3 Generalized System Model for Quantized MIMO Antenna Selection

Several antenna subset selection algorithms using greedy approaches exist in the literature for *unquantized* MIMO systems (see, for example, [46–48, 98]). We can convert an *unquantized* MIMO selection algorithm to *quantized* MIMO by re-deriving the entire algorithm to include quantization, for instance, by incorporating the AQNM. This is similar to what was done by the authors in [51] by converting the algorithm in [46] to the quantized MIMO case. They incorporated quantization using the AQNM and derived new selection equations based on the addition of quantization gains and a quantization error term. In the previous section, we generalize the approach in [51] by varying the number of bits utilized at each quantizer, i.e., we use  $\alpha_i$  instead of just  $\alpha$  in the algorithm.

In this section, however, instead of re-deriving the entire algorithm we develop an approach which is more generally applicable to converting unquantized to quantized MIMO antenna selection algorithms. We develop a modified system model which, when used, causes the conversion to occur. We convert the quantization effects from being a combination of gains and noise terms to only being scale factors on the channel vectors. We do this by whitening our *quantized* MIMO system model in (3.22) and derive an effective channel  $\mathbf{H}'$ . Our new system model is more generally applicable for converting unquantized MIMO antenna selection algorithms to quantized versions. We demonstrate this by using our model to convert the unquantized mutual information-based method (MIBM) in [48] to a quantized MIBM (QMIBM).

We now derive the effective channel by factoring out the effects of the different quantization error variances on the various channel vectors using channel whitening. Let  $\mathbf{y}' \triangleq \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \hat{\mathbf{y}}$ . Moreover, we rewrite (3.21) as

$$\begin{aligned}
\sigma_{z_{\text{tot}},m}^2 &= \alpha_m^2 + \alpha_m \beta_m \left( \frac{\rho}{N} \mathbf{h}_m^H \mathbf{C}_{\mathbf{x}} \mathbf{h}_m + 1 \right) \\
&= \alpha_m (\alpha_m + \beta_m) + \alpha_m \beta_m \frac{\rho}{N} \mathbf{h}_m^H \mathbf{C}_{\mathbf{x}} \mathbf{h}_m \\
&= \alpha_m \left( 1 + \beta_m \frac{\rho}{N} \mathbf{h}_m^H \mathbf{C}_{\mathbf{x}} \mathbf{h}_m \right) \\
&= \alpha_m d_m.
\end{aligned} \tag{3.47}$$

Then,

$$\begin{aligned}
\mathbf{y}' &= \sqrt{\frac{\rho}{N}} \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{V}_{\alpha} \mathbf{H} \mathbf{x} + \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{z}_{\text{tot}} \\
&= \sqrt{\frac{\rho}{N}} \mathbf{H}' \mathbf{x} + \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{z}_{\text{tot}},
\end{aligned} \tag{3.48}$$

where

$$\mathbf{H}' = \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{V}_{\alpha} \mathbf{H} = \begin{bmatrix} \sqrt{\alpha_1/d_1} \mathbf{h}_1^H \\ \vdots \\ \sqrt{\alpha_M/d_M} \mathbf{h}_M^H \end{bmatrix},$$

and the error term  $\mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{z}_{\text{tot}}$  is whitened noise, as shown in the following:

$$\begin{aligned}
\mathbb{E} \left[ \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{z}_{\text{tot}} \mathbf{z}_{\text{tot}}^H \left( \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \right)^H \right] &= \mathbb{E} \left[ \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} (\mathbf{V}_\alpha \mathbf{z} + \mathbf{w}) (\mathbf{V}_\alpha \mathbf{z}^H + \mathbf{w}^H) \left( \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \right)^H \right] \\
&\stackrel{(a)}{=} \mathbb{E} \left[ \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} (\mathbf{V}_\alpha \mathbf{z} \mathbf{z}^H \mathbf{V}_\alpha + \mathbf{w} \mathbf{w}^H) \left( \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \right)^H \right] \\
&= \mathbb{E} \left[ \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} (\mathbf{V}_\alpha^2 + \mathbf{C}_w) \left( \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \right)^H \right] \\
&= \mathbb{E} \left[ \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \mathbf{C}_{\mathbf{z}_{\text{tot}}} \left( \mathbf{C}_{\mathbf{z}_{\text{tot}}}^{-\frac{1}{2}} \right)^H \right] \\
&= \mathbf{I}_M,
\end{aligned}$$

where (a) is due to the fact that the quantization error  $\mathbf{w}$  is uncorrelated with  $\mathbf{y}$  [82, 93].

With the channel thus scaled, we can use (3.48) in the modified antenna subselection algorithm QMIBM. In [48], the authors take a top-down approach to antenna selection by beginning with the full  $\mathbf{H}$  and removing rows one-by-one based on the mutual information until the required dimension is achieved. The QMIBM algorithm selects antennas based on the pairwise mutual information between the quantized outputs  $y'_{i_v}$  and  $y'_{i_u}$  for  $i_v, i_u \in \mathcal{M} = \{1, \dots, M\}$ . If receive antenna outputs contain nearly identical information (high pairwise mutual information), then clearly we do not need both, and  $\tilde{\mathbf{G}}$  should select the one with the highest effective channel vector norm squared, i.e.,  $\frac{\alpha_{i_v}}{d_{i_v}} \|\mathbf{h}_{i_v}\|^2 \stackrel{?}{\geq} \frac{\alpha_{i_u}}{d_{i_u}} \|\mathbf{h}_{i_u}\|^2$ . Following [48], we use a normalized mutual information metric, which is defined as

$$I_0(y'_{i_v}, y'_{i_u}) = \frac{I(y'_{i_v}, y'_{i_u})}{I_{UB}(y'_{i_v}, y'_{i_u})}, \quad (3.49)$$

where

$$I(y'_{i_v}, y'_{i_u}) = \log \frac{\left( \|\mathbf{h}_{i_v}\|^2 \frac{\rho \alpha_{i_v}}{N d_{i_v}} + 1 \right) \left( \|\mathbf{h}_{i_u}\|^2 \frac{\rho \alpha_{i_u}}{N d_{i_u}} + 1 \right)}{\left( \|\mathbf{h}_{i_v}\|^2 \frac{\rho \alpha_{i_v}}{N d_{i_v}} + 1 \right) \left( \|\mathbf{h}_{i_u}\|^2 \frac{\rho \alpha_{i_u}}{N d_{i_u}} + 1 \right) - |\langle \mathbf{h}_{i_v}, \mathbf{h}_{i_u} \rangle|^2 \frac{\rho^2 \alpha_{i_v} \alpha_{i_u}}{N^2 d_{i_v} d_{i_u}}}, \quad (3.50)$$

---

**Algorithm 3:** Quantized MIBM Algorithm
 

---

```

1 Initialization:
2 Set  $L = J$ ;
3 For all  $i_v$  and  $i_u$ ,  $i_v > i_u$ , in  $\mathcal{M}$ , calculate  $I_0(y'_{i_v}, y'_{i_u})$ ;
4 while  $L > T$  do
5   Choose the  $i_v$  and  $i_u$  which return the largest  $I_0(y'_{i_v}, y'_{i_u})$ ;
6   if  $\frac{\alpha_{i_v}}{d_{i_v}} \|\mathbf{h}_{i_v}\|^2 \geq \frac{\alpha_{i_u}}{d_{i_u}} \|\mathbf{h}_{i_u}\|^2$  then
7     Remove  $\mathbf{h}_{i_u}$ ;
8     Delete  $i_u$  from  $\mathcal{M}$ ;
9     Set  $L = L - B_{i_u}$ ;
10  else
11    Remove  $\mathbf{h}_{i_v}$ ;
12    Delete  $i_v$  from  $\mathcal{M}$ ;
13    Set  $L = L - B_{i_v}$ ;
14  end
15 end

```

---

and  $I_{UB}(y'_{i_v}, y'_{i_u})$  is the mutual information upper bound given as

$$I_{UB}(y'_{i_v}, y'_{i_u}) = \min \left\{ \log \left( \|\mathbf{h}_{i_v}\|^2 \frac{\rho \alpha_{i_v}}{N d_{i_v}} + 1 \right), \log \left( \|\mathbf{h}_{i_u}\|^2 \frac{\rho \alpha_{i_u}}{N d_{i_u}} + 1 \right) \right\}. \quad (3.51)$$

The derivation for this upper bound on the quantized mutual information is similar to the unquantized one provided in [48].

Different from [48], we assume a constraint on the number of bits selected. Note the dependence of the mutual information on the channel vectors scaled by the quantization term  $\alpha_m/d_m$  and, thus, the number of bits per quantizer. Therefore, we start with the total number of bits before subselection  $J$  and track the number of bits after each down selection. We continue until the total number of selected bits  $L$  is less than or equal to the rate constraint  $T$ . The QMIBM algorithm proceeds as shown in Algorithm 3.

### 3.4.4 Numerical Results

In this section we verify the performance of the proposed modified QAFAS and QMIBM algorithms. We first show average capacity results for both algorithms under various conditions using Monte Carlo experiments. Next we evaluate the bit error rate (BER) of our algorithms in a practical system using quadrature amplitude modulation (QAM) to encode/decode the bits. Throughout this section we use two channel models:

1. *i.i.d. channel model.* A channel matrix  $\mathbf{H}$  with independent and identically distributed (i.i.d.) entries distributed as  $h_{ij} \sim \mathcal{CN}(0, 1)$ .
2. *mmWave channel model.* We assume a narrowband geometric millimeter wave (mmWave) channel model given by

$$\mathbf{H} = \sqrt{\frac{NM}{N_p}} \cdot \sum_{n=1}^{N_p} \alpha_n \mathbf{u}_n \mathbf{v}_n^H, \quad (3.52)$$

where  $\alpha_n \sim \mathcal{CN}(0, 1)$  denotes the complex path gain,  $\mathbf{u}_n$  denotes the  $M \times 1$  receive array steering vector, and  $\mathbf{v}_n$  denotes the  $N \times 1$  transmit array steering vector. The constant  $\sqrt{\frac{NM}{N_p}}$  normalizes the channel power. Assuming uniform linear arrays (ULAs) with  $\lambda/2$  element spacing where  $\lambda$  is the wavelength, the receive and transmit array steering vectors corresponding to angle of arrival (AoA)  $\phi_{R,n}$  and angle of departure (AoD)  $\phi_{T,n}$ , respectively, are given as

$$\mathbf{u}_n = \frac{1}{\sqrt{M}} [1, e^{-j\pi \sin(\phi_{R,n})}, \dots, e^{-j(M-1)\pi \sin(\phi_{R,n})}]^T \quad (3.53)$$

$$\mathbf{v}_n = \frac{1}{\sqrt{N}} [1, e^{-j\pi \sin(\phi_{T,n})}, \dots, e^{-j(N-1)\pi \sin(\phi_{T,n})}]^T. \quad (3.54)$$

We assume the AoAs and AoDs have a uniformly random distribution in an angular spread of  $180^\circ$  in the azimuth plane.

Additional simulation parameters are given in Table 3.3.

Table 3.3.  
Mutual information selection simulation parameters

Transmit antennas, $N$	8
Total receive antennas, $M$	$\{16, 64, 128\}$
MmWave paths, $N_p$	2
SNRs	$\{-10, -5, 0, 5, 10, 15, 20\}$ dB
Max number of quantization bits, $B_m$	5 bits

### General Performance Evaluation

We obtain numerical results in this section by Monte Carlo simulation of 10,000 independent channel realizations and, when using random  $B_m$ , quantization bit assignments. In addition to modified QAFAS and QMIBM results, for  $M = 16$  receive antennas we also show the optimal quantized MIMO antenna selection results obtained by exhaustive search. For  $M > 16$  the optimal selection is computationally prohibitive and is not shown.

Our first set of results in Fig. 3.4 verifies the performance of our proposed quantized MIMO selection algorithms. We compare the average capacity of the modified QAFAS and QMIBM to that of their corresponding *unquantized* versions, respectively, FAS and MIBM, as function of the number of antennas selected. As the unquantized algorithms are equivalent to infinite quantization, we are unable to use the rate constraint  $T$  as the independent variable on the horizontal axis. Therefore, for fair comparisons between the quantized and unquantized algorithms, *we only compare the unquantized results with equal  $B_m$  quantized results*, since we can easily attribute the equal  $B_m$  case to a specific number of selected antennas. In the case of  $M = 16$  receive antennas, we also include the optimal selection results. In every instance, the average capacity of a quantized algorithm is less than or equal to that of its corresponding unquantized algorithm. This is expected due to the quantization error. We only display results from the medium to high SNR regime, since at lower SNRs the average capacities for all of the algorithms are essentially equal. However, as can be seen for both the mmWave and i.i.d. channels, in the high SNR regime

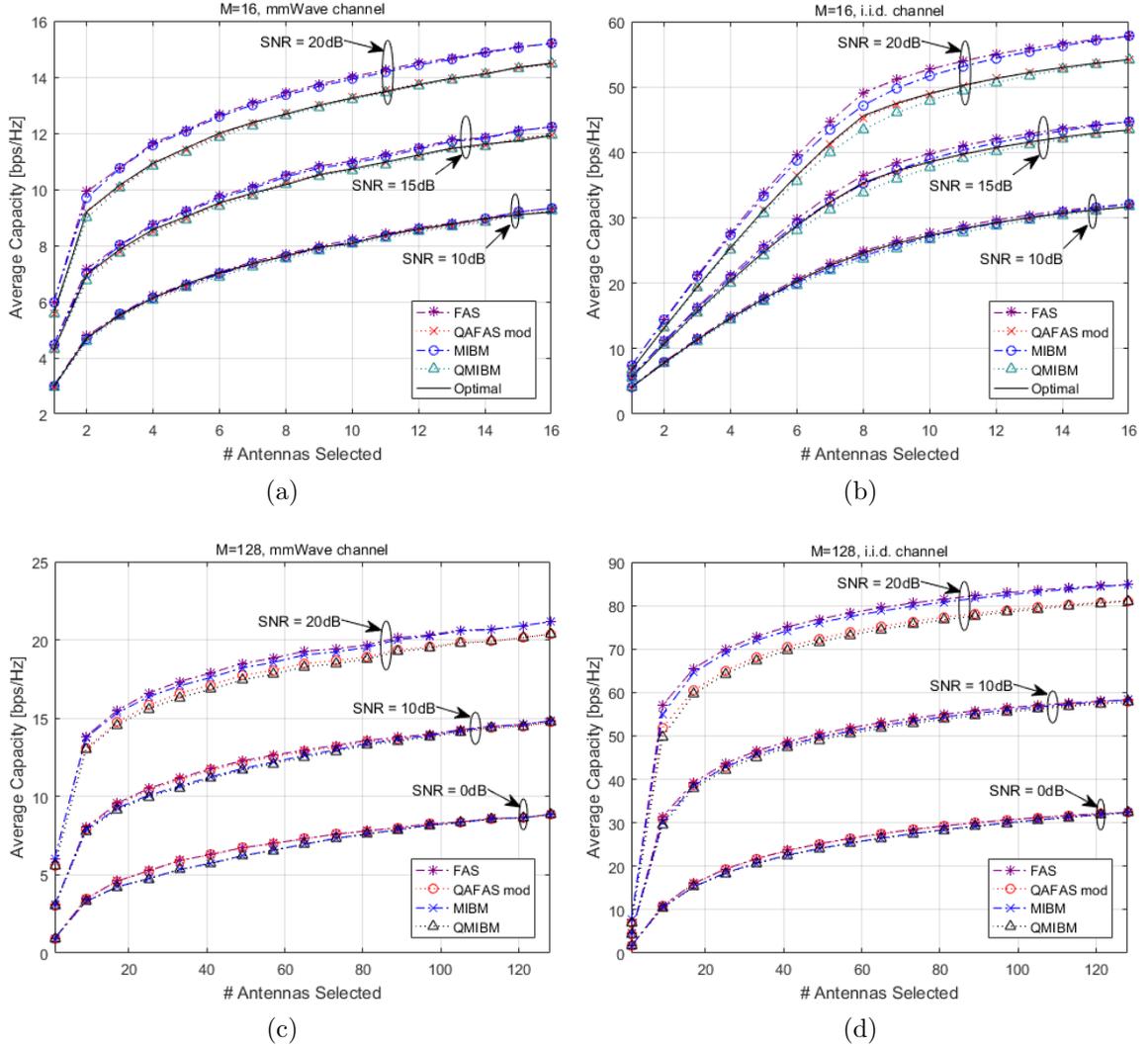


Fig. 3.4. Average capacity vs. number of selected antennas at various SNRs for unquantized (FAS, MIBM) and quantized (QAFAS, QMIBM) MIMO antenna selection algorithms. For fair comparisons with unquantized algorithms, we only use the equal  $B_m = 5$  case. We include the optimal quantized MIMO antenna selection results for  $M = 16$  receive antennas.

the quantization error dominants, and there exists greater separation between the unquantized and quantized algorithms.

In Figures 3.5 and 3.6 for the i.i.d. and mmWave channels, respectively, we show the average capacity as a function of SNR for the various output data rate constraints

$T$  shown. As discussed in Section 3.4.2, our algorithms allow for varying numbers of bits at the quantizers. Thus, our modified QAFAS algorithm generalizes the QAFAS algorithm developed in [51]. These results show how varying  $B_m$  can improve the average capacity under certain parameters. The dash-dot line denotes results obtained when randomly assigning  $B_m$  to quantizers. We use a uniform distribution between 1 and 5 bits. The dotted line denotes results obtained when all quantizers have the same number of bits (as in the QAFAS algorithm in [51]), which in this case we set to  $B_m = 5$  bits. For a clearer comparison between the random  $B_m$  and equal  $B_m$  average capacity performances, in Figures 3.7 and 3.8 we plot the difference between the equal  $B_m$  and random  $B_m$  results for optimal selection and modified QAFAS at  $M = 16$  and for modified QAFAS at  $M = 128$  using both channel models.

We plot the average capacity as a function of the output data rate constraint  $T$  in Figures 3.9 and 3.10 for  $M = 16$  and  $M = 128$ , respectively, at several SNRs. As is evident in Fig. 3.4, the modified QAFAS and QMIBM algorithms are nearly equivalent, with the modified QAFAS having a slightly higher average capacity than QMIBM. Therefore, in Figures 3.9 and 3.10 we only plot the modified QAFAS results; for  $M = 16$  in Fig. 3.9 we also include the optimal selection results.

We make the following observations and comments from Figures 3.5 - 3.10:

1. The i.i.d. and mmWave channels present similar trends, despite the average capacity for mmWave channels being scaled down relative to that of the i.i.d. channels due to the sparsity of the mmWave channels.
2. For equal values of low output data rate thresholds  $T$  and at low SNRs, the random  $B_m$  average capacity is generally greater than the corresponding equal  $B_m$  average capacity. This is especially noticeable in the difference plots in Figures 3.7 and 3.8. When the additive white Gaussian noise (AWGN) dominates the quantization error (low SNR regime), assigning more quantization bits to the higher capacity sub-channels and fewer quantization bits to the lower capacity sub-channels results in a greater overall capacity for the MIMO channel.

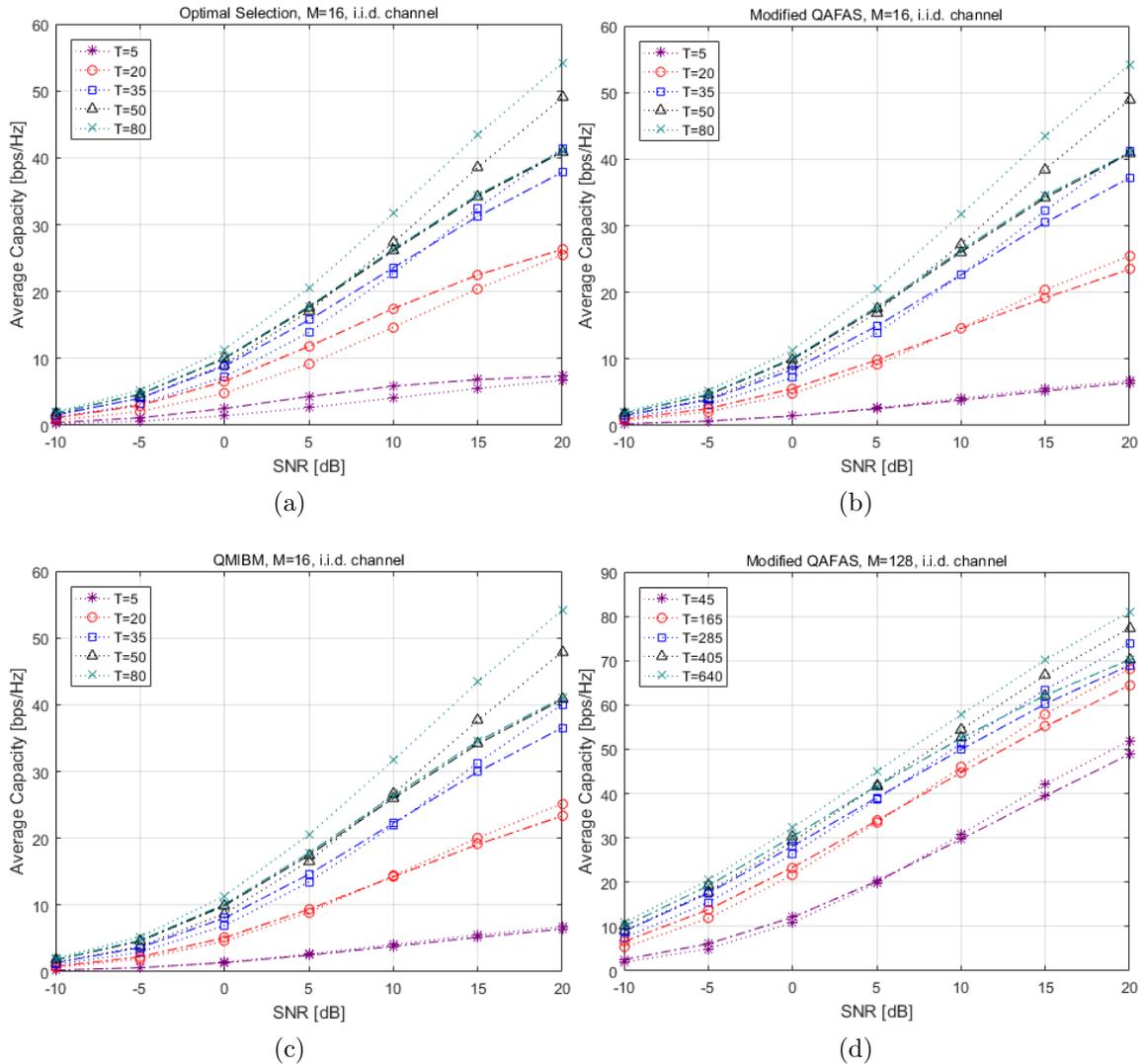


Fig. 3.5. Average capacity vs. SNR for the i.i.d. channel at various output data rates  $T$  for equal (dash-dot) and random (dot-dot)  $B_m$ . (a)-(c) show average capacity when  $M = 16$  for optimal selection, modified QAFAS, and QMIBM, respectively, while (d) shows average capacity when  $M = 128$  for modified QAFAS. Note the higher average capacity when using random  $B_m$  versus equal  $B_m$  at lower SNRs and output data rates.

In the high SNR regime, the quantization error dominates, and higher average numbers of quantization bits returns greater average capacity. In general, vary-

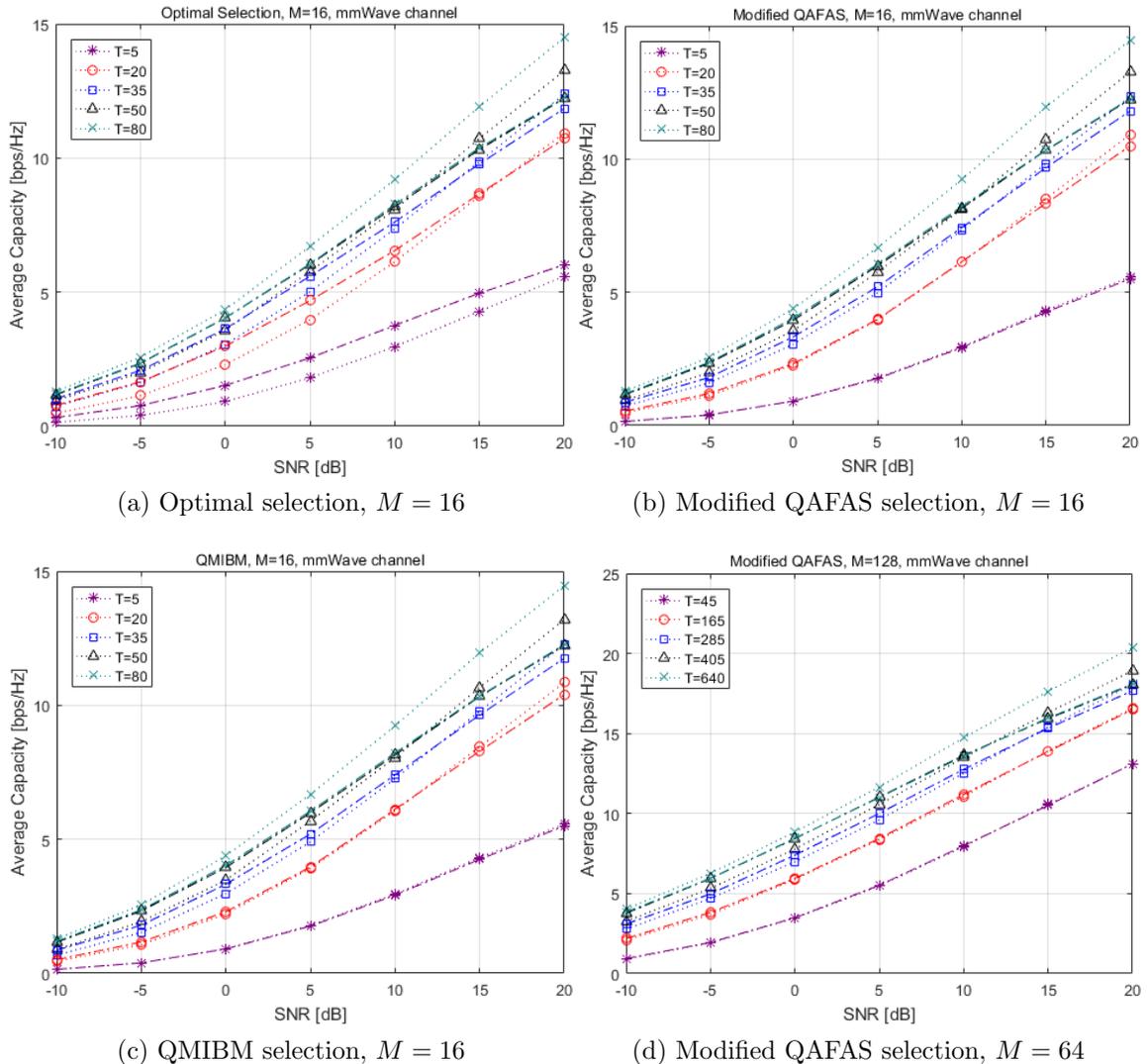


Fig. 3.6. Average capacity vs. SNR for the mmWave channel at various output data rates  $T$  for equal (dash-dot) and random (dot-dot)  $B_m$ . (a)-(c) show average capacity when  $M = 16$  for optimal selection, modified QAFAS, and QMIBM, respectively, while (d) shows average capacity when  $M = 128$  for modified QAFAS.

ing the number of quantization bits - even randomly, as in our results - improves the overall average capacity in the low SNR regime.

3. For the optimal selection algorithm, as the rate constraint  $T$  increases, the SNR at which the random  $B_m$  average capacity is greater than the equal  $B_m$  average

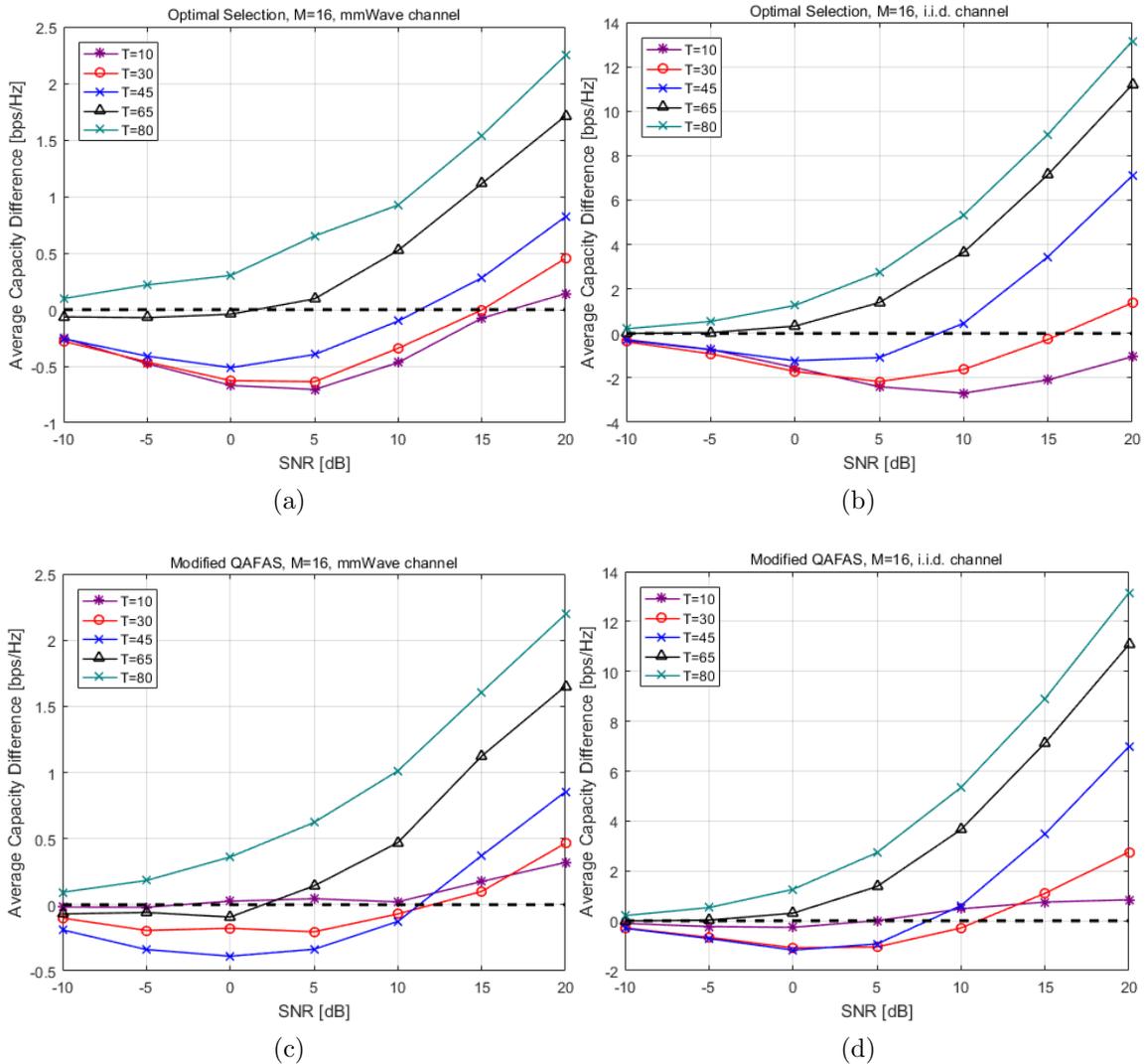


Fig. 3.7. Difference between equal  $B_m$  and random  $B_m$  vs. SNR at various values of  $T$  when using optimal selection (a)-(b) and the modified QAFAS algorithm (c)-(d) for  $M = 16$  and both channel models.

capacity decreases. In our simulations for the random  $B_m$  we assume  $B_m$  is assigned according to a random uniform distribution between 1 and 5 bits. Thus, on average a randomly assigned quantizer has 3 bits - 2 bits less than the 5 bits assigned to each quantizer when using equal  $B_m$  - and, as a result, the advantage of the random  $B_m$  diminishes as  $T$  increases.

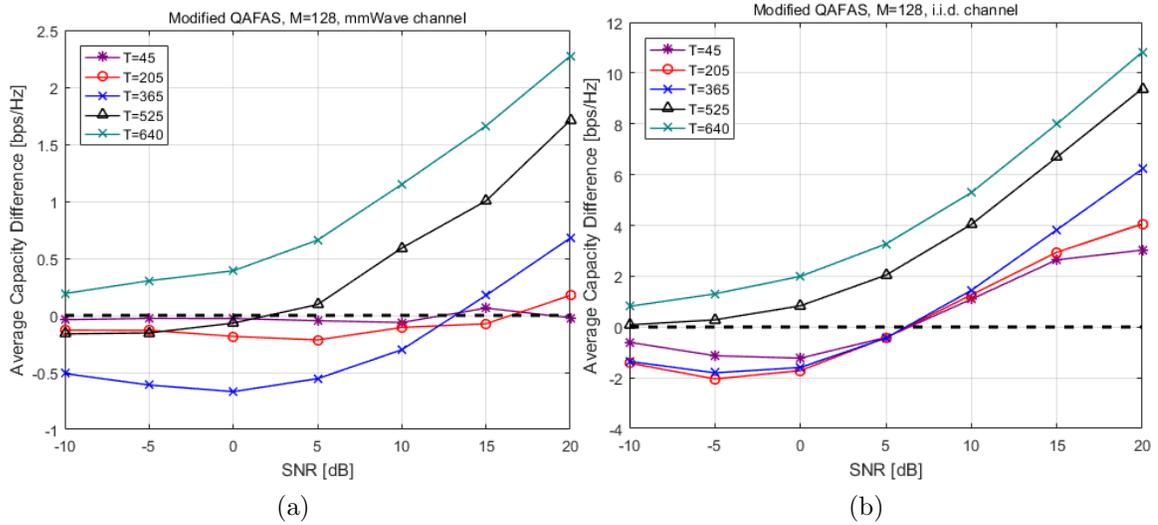


Fig. 3.8. Difference between random  $B_m$  and equal  $B_m$  vs. SNR at various values of  $T$  when using the modified QAFAS algorithm.

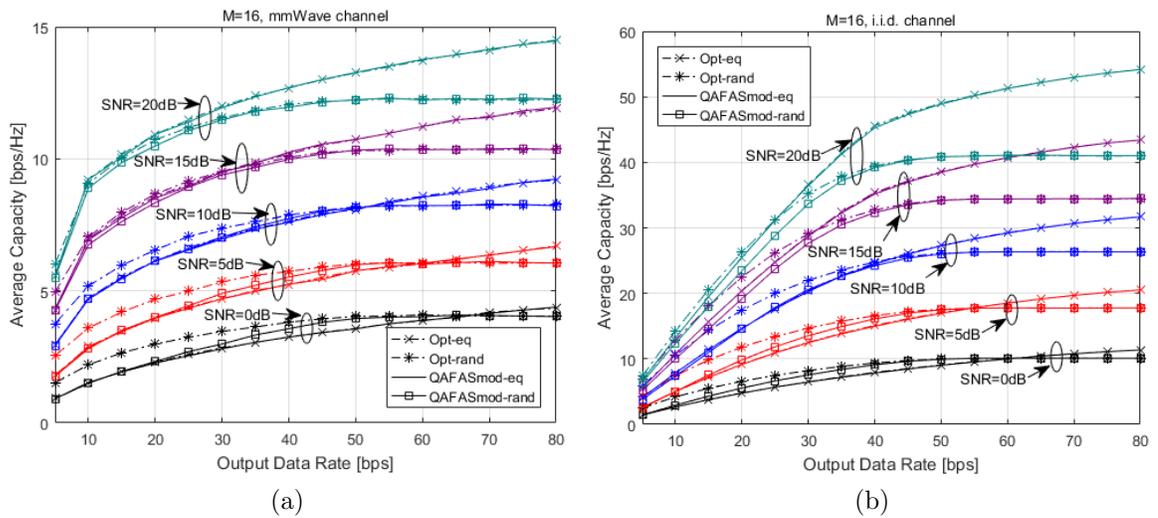


Fig. 3.9. Average capacity vs. output data rate  $T$  for equal  $B_m$  and random  $B_m$  at various SNRs for  $M = 16$ .

This outcome is accentuated in our sub-optimal selection algorithms, as seen for modified QAFAS in Figures 3.7 and 3.8. (Note that while we do not show the QMIBM results in these figures to improve readability, our simulations confirm nearly equivalent results for the QMIBM algorithm.) The difference

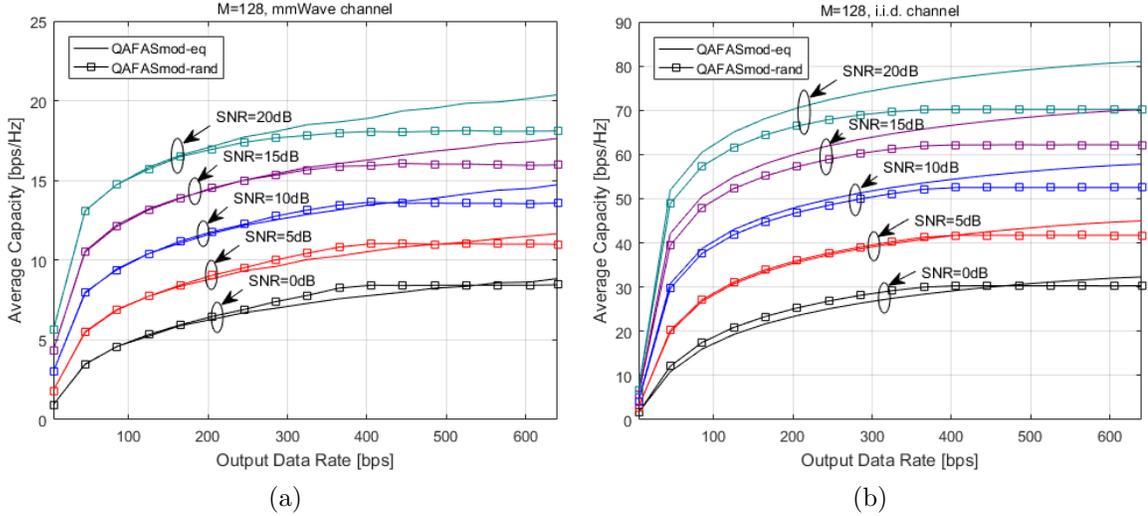


Fig. 3.10. Average capacity vs. output data rate  $T$  for equal  $B_m$  and random  $B_m$  at various SNRs for  $M = 128$ .

between the equal  $B_m$  and random  $B_m$  average capacities becomes more negative as  $T$  moves towards 50% of the maximum output data rate threshold, or  $T_{max} = M \max_m(B_m)$ . However, as  $T$  increases above  $T_{max}/2$  the average capacity difference moves in a positive direction. Thus, the average capacity upper bound when using random  $B_m$  is achieved at  $T_{UB} = M \text{mean}(B_m)$ , as we would intuitively expect. Figures 3.9 and 3.10 also illustrate this result.

### Practical QAM Evaluation

To evaluate the accuracy of our proposed approximations and to provide a proof-of-concept example, we implement our modified QAFAS antenna selection algorithm using a QAM constellation of  $\Lambda$  symbols and  $k = \log_2 \Lambda$  bits per symbol. We compare the BER performance of the modified QAFAS with random  $B_m$  to that of the unmodified QAFAS from [51] with equal  $B_m$ . While not presented, the BER performance of QMIBM is similar to that of the modified QAFAS. Our BER simulations are for  $10^6$  bits or  $10^3$  errors, whichever occurs first. Bit error rate curves are gen-

erated for  $\Lambda = \{16, 256\}$  symbols and  $M = \{16, 64\}$  receive antennas over SNRs  $\rho = \{-10, -5, \dots, 15, 20\}$  dB. All other parameters are as given in the previous section.

A symbol  $s$  is randomly selected for signaling from an  $\Lambda$ -QAM constellation with  $\mathbb{E}[s] = 0$  and  $\mathbb{E}[|s|^2] = 1$ . We generate an independent channel realization  $\mathbf{H}$  and perform antenna selection, assuming knowledge of the channel and the number of quantization bits  $B_m$  at the  $m$ -th receive chain quantizer. The sub-selected channel matrix is denoted by  $\mathbf{H}_s$ . In order to focus our results on the performance of the modified QAFAS with varying  $B_m$ , we simply use the singular value decomposition (SVD) of  $\mathbf{H}_s$  to obtain the optimal beamforming and combining vectors,  $\mathbf{f}_{opt}$  and  $\mathbf{g}_{opt}$ , respectively, where

$$\mathbf{f}_{opt} = \mathbf{v}_1, \mathbf{g}_{opt} = \frac{\mathbf{H}_s \mathbf{v}_1}{\|\mathbf{H}_s \mathbf{v}_1\|}, \quad (3.55)$$

and  $\mathbf{v}_1$  represents the dominant unit-norm right singular vector of  $\mathbf{H}_s$ . Recall that under the AQNM the quantized signal is given as (from (3.22), but repeated here for convenience)

$$\hat{\mathbf{y}} = \sqrt{\frac{\rho}{N}} \mathbf{V}_\alpha \mathbf{H} \mathbf{x} + \mathbf{z}_{tot}.$$

Using the linear combiner  $\mathbf{g}_{opt}$  with nearest-neighbor decoding, we obtain an estimate of the transmitted symbol as

$$\hat{s} = \mathbf{g}_{opt}^H \hat{\mathbf{y}} = \sqrt{\frac{\rho}{N}} \mathbf{g}_{opt}^H \mathbf{V}_\alpha \mathbf{H}_s \mathbf{f}_{opt} s + \mathbf{g}_{opt}^H \mathbf{z}_{tot}. \quad (3.56)$$

We first consider BER curves generated for a  $\Lambda = 16$ -QAM constellation and  $M = 16$  antennas under both channel models. As expected based on our previous observations, the equal  $B_m$  curves have a BER greater than or equal to the random  $B_m$  BER curves. On average, the largest difference between equal and random  $B_m$  is when  $T = T_{max}/2$ . Therefore, in our next BER plots we only include results at this threshold.

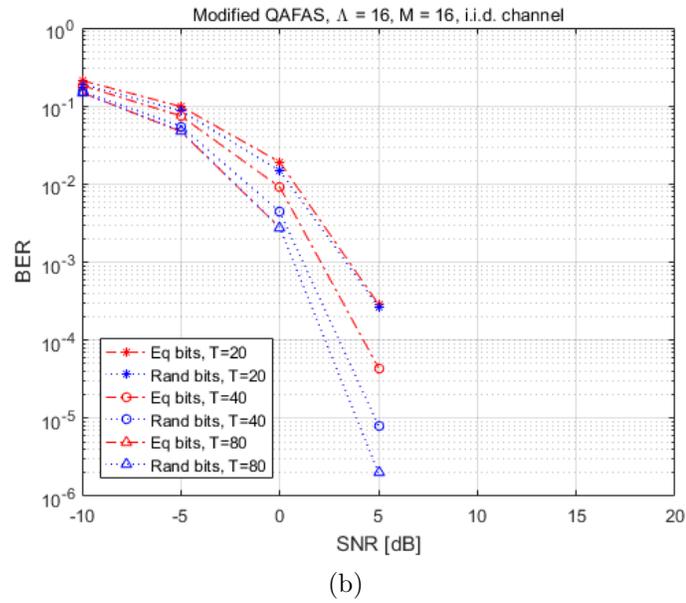
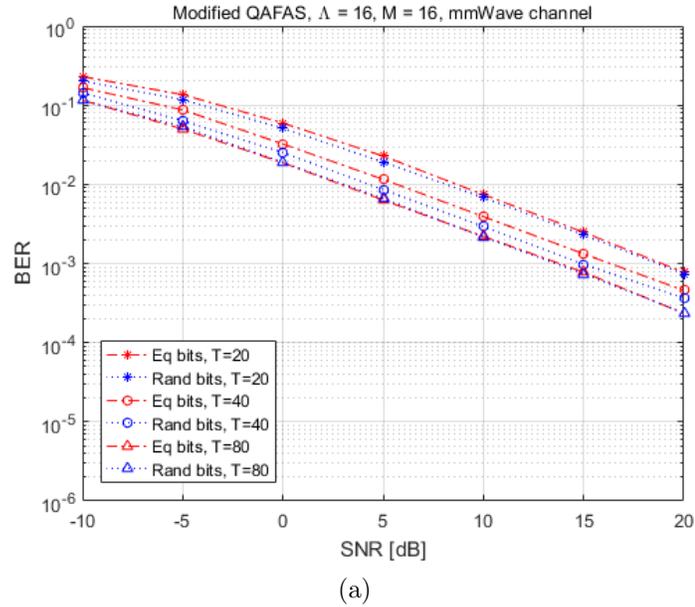


Fig. 3.11. BER curves comparing performance of equal  $B_m$  and random  $B_m$  for various output data rate thresholds  $T$ .

In Fig. 3.12 we show pairs of equal and random  $B_m$  BER curves for different configurations of  $\Lambda$  and  $M$  for both channel models. Clearly, the BER decreases when  $M$  increases. Conversely, when  $\Lambda$  increases, the BER also increases. Both of these behaviors are as expected for a MIMO system. In each of the configurations,

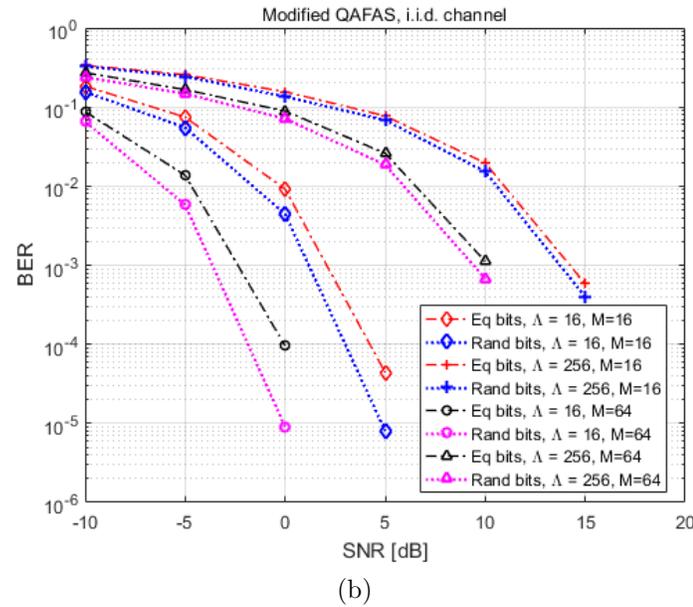
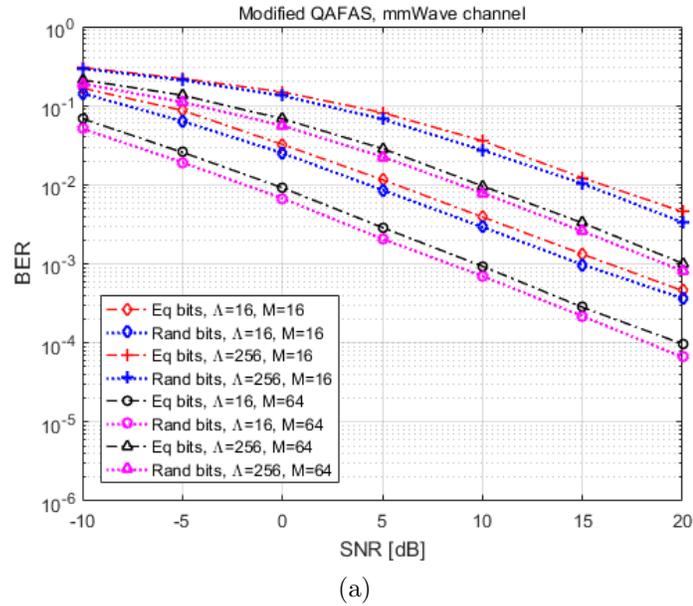


Fig. 3.12. BER curves comparing performance of equal  $B_m$  and random  $B_m$  for  $T = T_{max}/2$ .

the random  $B_m$  BER is less than the equal  $B_m$  BER, although at low SNR and large  $\Lambda$  the two performances are similar due to the AWGN and the constellation symbol proximity to its neighboring symbols.

## 4. CONCLUSION

In this dissertation, we investigated adaptive signal processing techniques for high frequency, multiple-input multiple-output (MIMO) wireless communication systems. Although our approaches can be applied elsewhere, we focused on the mobile millimeter wave (mmWave) communication system scenario. Specifically, we addressed two problems encountered by mobile MIMO systems at mmWaves: 1) transmitter/receiver beam alignment and 2) mutual information selection via antenna subset selection.

In Chapter 2, we proposed a novel beam alignment and tracking algorithm for sparse, time-varying mmWave channels using a sparse Bayesian learning (SBL) Kalman filter (KSBL) with a multi-armed bandit (MAB) beam selection known as Linear Thompson sampling (LTS). We showed KSBL-LTS has rapid initial beam alignment compared to other beam selection policies. Significantly, KSBL-LTS performed nearly as well as omni-directional training for mmWave channels with a dynamic support. This result held when using an ideal narrowband, MIMO channel model and when using a more realistic mmWave channel model known as QUasi Deterministic RadIo channel GenerAtor (QuaDRiGa). This work has been published in IEEE Communication Letters [99].

In Chapter 3, we developed algorithms for maximizing the mutual information between the input and output of a quantized MIMO system when constrained by the output data rate. Specifically, we used the additive quantization noise model (AQNM) to model varying numbers of quantization bits in the receive signal processing chains. Given an output rate constraint, we developed a mutual information-based expression for the optimal joint selection of quantized MIMO antenna outputs. We determined algorithms for finding a linear selection matrix using a general greedy selection algorithm that allowed for varying numbers of quantization bits at each receive antenna.

We demonstrated how our signal model can be used in general to convert unquantized MIMO antenna selection algorithms to quantized MIMO antenna selection algorithms. Finally, we showed several results verifying the performance of our algorithms and approach. Significantly, under low signal-to-noise ratio (SNR) and low output data rate thresholds the uniformly random determination of numbers of quantization bits has a higher average capacity than if the number of quantization bits were all equal.

## REFERENCES

## REFERENCES

- [1] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-wave cellular wireless networks: Potentials and challenges,” *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.
- [2] Z. Pi and F. Khan, “An introduction to millimeter-wave mobile broadband systems,” *IEEE Communications Magazine*, vol. 49, no. 6, pp. 101–107, 6 2011.
- [3] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, “Millimeter wave mobile communications for 5G cellular: It will work!” *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [4] “3rd Generation Partnership Project (3GPP), Release 15,” Tech. Rep. Release 15, 2018. [Online]. Available: <http://www.3gpp.org>
- [5] J. G. Andrews, T. Bai, M. Kulkarni, A. Alkhateeb, A. Gupta, and R. W. Heath Jr., “Modeling and Analyzing Millimeter Wave Cellular Systems,” *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 403–430, 2017.
- [6] A. O. Hero III, D. A. Castafion, D. Cochran, and K. D. Kastella, Eds., *Foundations and Applications of Sensor Management*. Springer, 2008.
- [7] S. Bubeck and N. Cesa-Bianchi, *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*, M. Jordan, Ed. Boston: now Publishers Inc., 2012, vol. 5, no 1. [Online]. Available: [www.nowpublishers.com](http://www.nowpublishers.com)
- [8] A. Mukherjee and A. Hottinen, “Learning Algorithms For Energy Efficient MIMO Antenna Subset Selection : Multi-Armed Bandit Framework,” in *2012 Proceedings of the 20th European Signal Processing Conference*, no. Eusipco, 2012, pp. 659–663.
- [9] S. Jing, L. Zheng, and M. Medard, “On Training with Feedback in Wideband Channels,” *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1607–1614, 2008.
- [10] M. Agarwal and M. L. Honig, “Wideband Fading Channel Capacity With Training and Partial Feedback,” *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4865–4873, 2010.
- [11] Y. Gai, B. Krishnamachari, and R. Jain, “Learning multiuser channel allocations in cognitive radio networks: a combinatorial multi-armed bandit formulation,” *2010 IEEE Symposium on New Frontiers in Dynamic Spectrum, DySPAN 2010*, 2010.
- [12] —, “Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

- [13] K. Liu and Q. Zhao, "Distributed Learning in Multi-Armed Bandit," *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5667–5681, 2010.
- [14] L. Lai, H. Jiang, and H. V. Poor, "Medium access in cognitive radio networks: A competitive multi-armed bandit framework," in *Signals, Systems and Computers, 2008 42nd Asilomar Conference on*, 2008, pp. 98–102.
- [15] T. V. Nguyen, H. Shin, T. Q. S. Quek, and M. Z. Win, "Sensing and Probing Cardinalities for Active Cognitive Radios," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1833–1848, 2012.
- [16] N. Gulati and K. R. Dandekar, "Learning state selection for reconfigurable antennas: A multi-armed bandit approach," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 3, pp. 1027–1038, 2014.
- [17] J. Seo, Y. Sung, G. Lee, and D. Kim, "Training Beam Sequence Design for Millimeter-Wave MIMO Systems: A POMDP Framework," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1228–1242, 2016.
- [18] J.-y. Audibert and S. Bubeck, "Best Arm Identification in Multi-Armed Bandits," in *23th Conference on Learning Theory*, Haifa, Israel, 2010.
- [19] S. Shahrampour, M. Noshad, and V. Tarokh, "On Sequential Elimination Algorithms for Best-Arm Identification in Multi-Armed Bandits," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4281–4292, 2017.
- [20] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Online Learning for Position-Aided Millimeter Wave Beam Training," *IEEE Access*, vol. 7, pp. 30 507–30 526, 2019.
- [21] M. Hashemi, A. Sabharwal, C. E. Koksal, and N. B. Shroff, "Efficient Beam Alignment in Millimeter Wave Systems Using Contextual Bandits," in *IEEE INFOCOM*, 2018.
- [22] M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *J. of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
- [23] D. P. Wipf and B. D. Rao, "Sparse Bayesian Learning for Basis Selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.
- [24] R. Prasad, C. R. Murthy, and B. D. Rao, "Joint Approximately Sparse Channel Estimation and Data Detection in OFDM Systems Using Sparse Bayesian Learning," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3591–3603, 2014.
- [25] G. Joseph, C. R. Murthy, R. Prasad, and B. D. Rao, "Online recovery of temporally correlated sparse signals using multiple measurement vectors," in *IEEE GLOBECOM*, San Diego, CA, USA, 2015.
- [26] D. Gilton and R. Willett, "Sparse linear contextual bandits via relevance vector machines," in *Int'l Conf. on Sampling Theory & Applications (SampTA)*, 2017, pp. 518–522.
- [27] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, 2013.

- [28] A. Alkhateeb, Y.-H. Nam, M. S. Rahman, J. Zhang, and R. W. Heath Jr., “Initial Beam Association in Millimeter Wave Cellular Systems: Analysis and Design Insights,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2807–2821, 2016. [Online]. Available: <http://arxiv.org/abs/1602.06598>
- [29] C. Jeong, J. Park, and H. Yu, “Random access in millimeter-wave beamforming cellular networks: Issues and approaches,” *IEEE Communications Magazine*, vol. 53, no. 1, pp. 180–185, 2015.
- [30] D. Ogbe, D. J. Love, and V. Raghavan, “Noisy Beam Alignment Techniques for Reciprocal MIMO Channels,” *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5092–5107, 2017.
- [31] V. Raghavan, J. Cezanne, S. Subramanian, A. Sampath, and O. Koymen, “Beamforming Tradeoffs for Initial UE Discovery in Millimeter-Wave MIMO Systems,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 543–559, 2016.
- [32] S. Sun, T. S. Rappaport, R. W. Heath Jr., A. Nix, and S. Rangan, “MIMO for millimeter-wave wireless communications: Beamforming, spatial multiplexing, or both?” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 110–121, 2014.
- [33] V. Va, H. Vikalo, and R. W. Heath Jr., “Beam Tracking for Mobile Millimeter Wave Communication Systems,” in *IEEE GlobalSIP*, Washington, DC, USA, 2016, pp. 743–747.
- [34] S. Jayaprakasam, X. Ma, J. W. Choi, and S. Kim, “Robust Beam-Tracking for mmWave Mobile Communications,” *IEEE Communications Letters*, vol. 21, no. 12, pp. 2654–2657, 2017.
- [35] J. H. Yoo, J. Bae, S. H. Lim, S. Kim, J. W. Choi, and B. Shim, “Sampling-based tracking of time-varying channels for millimeter wave-band communications,” in *IEEE ICC*, 2017.
- [36] J. Singh, O. Dabeer, and U. Madhow, “On the limits of communication with low-precision analog-to-digital conversion at the receiver,” *IEEE Transactions on Communications*, vol. 57, no. 12, pp. 3629–3639, 2009.
- [37] J. Mo and R. W. Heath, “Capacity Analysis of One-Bit Quantized MIMO Systems with Transmitter Channel State Information,” *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5498–5512, 2015.
- [38] J. Mo, P. Schniter, and R. W. Heath, “Channel Estimation in Broadband Millimeter Wave MIMO Systems with Few-Bit ADCs,” *IEEE Transactions on Signal Processing*, vol. 66, no. 5, pp. 1141–1154, 2018.
- [39] D. S. Palguna, D. J. Love, T. A. Thomas, and A. Ghosh, “Millimeter Wave Receiver Design Using Low Precision Quantization and Parallel  $\Delta\Sigma$  Architecture,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6556–6569, 2016.
- [40] C. Mollen, J. Choi, E. G. Larsson, and R. W. Heath, “Uplink Performance of Wideband Massive MIMO with One-Bit ADCs,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 87–100, 2017.

- [41] G. Zeitler, A. C. Singer, and G. Kramer, "Low-precision A/D conversion for maximum information rate in channels with memory," *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2511–2521, 2012.
- [42] S. Sandhu, R. U. Nabart, D. A. Gore, and A. Paulraj, "Near-Optimal Selection of Transmit Antennas for a MIMO Channel Based on Shannon Capacity," in *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2000, pp. 567–571.
- [43] A. Dua, K. Medepalli, and A. J. Paulraj, "Receive antenna selection in MIMO systems using convex optimization," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2353–2357, 2006.
- [44] R. W. Heath Jr., S. Sandhu, and A. J. Paulraj, "Antenna Selection for Spatial Multiplexing Systems with Linear Receivers," *IEEE Communications Letters*, vol. 5, no. 4, pp. 142–144, 2001.
- [45] C. Jiang and L. J. Cimini, "Antenna selection for energy-efficient MIMO transmission," *IEEE Wireless Communications Letters*, vol. 1, no. 6, pp. 577–580, 2012.
- [46] M. Gharavi-Alkhansari and A. B. Gershman, "Fast antenna subset selection in MIMO systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 339–347, 2004.
- [47] A. Gorokhov, D. A. Gore, and A. J. Paulraj, "Receive Antenna Selection for MIMO Spatial Multiplexing : Theory and Algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2796–2807, 2003.
- [48] A. F. Molisch, M. Z. Win, Y.-S. Choi, and J. H. Winters, "Capacity of MIMO systems with antenna selection," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1759–1772, 2005.
- [49] S. Rini, L. Barletta, Y. C. Eldar, and E. Erkip, "A general framework for MIMO receivers with low-resolution quantization," in *IEEE Information Theory Workshop (ITW)*. Kaohsiung, Taiwan: IEEE, 2017, pp. 599–603.
- [50] A. Khalili, S. Rini, L. Barletta, E. Erkip, and Y. C. Eldar, "On MIMO Channel Capacity with Output Quantization Constraints," in *IEEE International Symposium on Information Theory (ISIT)*. Vail, CO, USA: IEEE, 2018, pp. 1355–1359.
- [51] J. Choi, J. Sung, N. Prasad, X.-F. Qi, B. L. Evans, and A. Gatherer, "Base Station Antenna Selection for Low-Resolution ADC Systems," *IEEE Transactions on Communications*, vol. 6778, no. c, pp. 1–1, 2019.
- [52] H. Nosrati, E. Aboutanios, and D. B. Smith, "Receiver-transmitter pair selection in MIMO phased array radar," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3206–3210, 2017.
- [53] H. Godrich, A. P. Petropulu, and H. V. Poor, "Sensor selection in distributed multiple-radar architectures for localization: A knapsack problem formulation," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 247–260, 2012.

- [54] B. Ma, H. Chen, B. Sun, and H. Xiao, "A joint scheme of antenna selection and power allocation for localization in MIMO radar sensor networks," *International Conference on Signal Processing Proceedings, ICSP*, vol. 2015-Janua, no. October, pp. 2226–2229, 2014.
- [55] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.
- [56] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143–159, 2002.
- [57] M. Tesmer and P. A. Estévez, "AMIFS: adaptive feature selection by using mutual information," in *IEEE International Joint Conference on Neural Networks*, Budapest, 2004, pp. 303–308.
- [58] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Computing and Applications*, vol. 24, no. 1, pp. 175–186, 2014.
- [59] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, 1999, pp. 368–377.
- [60] D. J. Strouse and D. J. Schwab, "The Deterministic Information Bottleneck," *Neural Computation*, vol. 29, no. 6, pp. 1611–1630, 2017.
- [61] A. Homri, M. Peleg, and S. Shamai Shitz, "Oblivious fronthaul-constrained relay for a Gaussian channel," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5112–5123, 2018.
- [62] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, "Information bottleneck for Gaussian variables," *Journal of Machine Learning Research*, vol. 6, no. 1, pp. 165–188, 2005.
- [63] A. Winkelbauer and G. Matz, "Rate-information-optimal Gaussian channel output compression," in *48th Annual Conference on Information Sciences and Systems*. IEEE, 2014.
- [64] A. Winkelbauer, S. Farthofer, and G. Matz, "The rate-information trade-off for Gaussian vector channels," in *IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 2849–2853.
- [65] V. Boljanovic, H. Yan, and D. Cabric, "Tracking Sparse mmWave Channel under Time Varying Multipath Scatterers," in *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1274–1279.
- [66] D. Zhu, J. Choi, Q. Cheng, W. Xiao, and R. W. Heath, "High-Resolution Angle Tracking for Mobile With Antenna Array Calibration," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7173–7189, 2018.
- [67] I. Chafaa, E. V. Belmega, and M. Debbah, "Adversarial Multi-armed Bandit for mmWave Beam Alignment with One-Bit Feedback," in *12th EAI Int'l Conf. on Perf. Eval. Methodologies & Tools*. ACM, 2019, pp. 23–30.

- [68] G. H. Sim, S. Klos, A. Asadi, A. Klein, and M. Hollick, "An Online Context-Aware Machine Learning Algorithm for 5G mmWave Vehicular Communications," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2487–2500, 2018.
- [69] M. R. Akdeniz, Y. Liu, K. Mathew, S. Sun, and S. Rangan, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, 2014.
- [70] R. W. Heath Jr., N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, 2016.
- [71] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [72] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A Tutorial on Thompson Sampling," in *Foundations and Trends® in Machine Learning*, 2018, vol. 11, pp. 1–96.
- [73] Z. Ghahramani and G. E. Hinton, "Parameter Estimation for Linear Dynamical Systems," University of Toronto, Tech. Rep. CRG-TR-96-2, 1996.
- [74] "3rd Generation Partnership Project (3GPP), Release 14," Tech. Rep. Release 14, 2017. [Online]. Available: <http://www.3gpp.org>
- [75] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.
- [76] S. Noh, M. D. Zoltowski, and D. J. Love, "Multi-Resolution Codebook and Adaptive Beamforming Sequence Design for Millimeter Wave Beam Alignment," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5689–5701, 2017.
- [77] J. Mo, A. Alkhateeb, S. Abu-Surra, and R. W. Heath, "Hybrid Architectures with Few-Bit ADC Receivers: Achievable Rates and Energy-Rate Tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2274–2287, 2017.
- [78] J. Zhang, L. Dai, Z. He, S. Jin, and X. Li, "Performance analysis of mixed-ADC massive MIMO systems over rician fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1327–1338, 2017.
- [79] J. Choi, B. L. Evans, and A. Gatherer, "Resolution-Adaptive Hybrid MIMO Architectures for Millimeter Wave Communications," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6201–6216, 2017.
- [80] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.
- [81] O. Orhan, E. Erkip, and S. Rangan, "Low Power Analog-to-Digital Conversion in Millimeter Wave Systems : Impact of Resolution and Bandwidth on Performance," in *2015 Information Theory and Applications Workshop (ITA)*. IEEE, 2015, pp. 191–198.

- [82] A. Mezghani and J. A. Nossek, "Capacity Lower Bound of MIMO Channels with Output Quantization and Correlated Noise," *IEEE International Symposium on Information Theory*, no. 3, pp. 1732–1736, 2012.
- [83] A. Mezghani, F. Antreich, and J. A. Nossek, "Multiple parameter estimation with quantized channel output," *2010 International ITG Workshop on Smart Antennas, WSA 2010*, no. Wsa, pp. 143–150, 2010.
- [84] M. Hussain and N. Michelusi, "Energy-Efficient Interactive Beam-Alignment for Millimeter-Wave Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 838–851, 2019.
- [85] C. N. Barati, S. A. Hosseini, M. Mezzavilla, T. Korakis, S. S. Panwar, S. Rangan, and M. Zorzi, "Initial Access in Millimeter Wave Cellular Systems," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 7926–7940, 2016.
- [86] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath Jr., "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, 2014.
- [87] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted gaussian signals," Research Laboratory of Electronics, Massachusetts Institute of Technology, Tech. Rep. 216, 1952.
- [88] K. Roth, H. Pirzadeh, A. L. Swindlehurst, and J. A. Nossek, "A Comparison of Hybrid Beamforming and Digital Beamforming With Low-Resolution ADCs for Multiple Users and Imperfect CSI," *IEEE Journal on Selected Topics in Signal Processing*, vol. 12, no. 3, pp. 484–498, 2018.
- [89] S. Jacobsson, G. Durisi, M. Coldrey, T. Goldstein, and C. Studer, "Quantized Precoding for Massive MU-MIMO," *IEEE Transactions on Communications*, vol. 65, no. 11, pp. 4670–4684, 2017.
- [90] Q. Bai, A. Mezghani, and J. A. Nossek, "On the Optimization of ADC Resolution in Multi-antenna Systems," in *The Tenth International Symposium on Wireless Communication Systems*, Offenbach, Germany, 2013, pp. 424–428.
- [91] S. N. Diggavi and T. M. Cover, "The worst case additive noise under a covariance constraint," *IEEE Trans. on Info. Theory*, vol. 47, no. 7, pp. 3072–3081, 2001.
- [92] A. Mezghani and A. L. Swindlehurst, "Blind estimation of sparse broadband massive MIMO channels with ideal and one-bit ADCs," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2972–2983, 2018.
- [93] A. K. Fletcher, S. Rangan, V. K. Goyal, and K. Ramchandran, "Robust predictive quantization: Analysis and design via convex optimization," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 618–632, 2007.
- [94] N. Al-Dhahir and J. M. Cioffi, "On the uniform ADC bit precision and clip level computation for a Gaussian signal," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 434–438, 1996.
- [95] G. M. Roe, "Quantizing for Minimum Distortion (Corresp.)," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 384–385, 1964.

- [96] D. Hui and D. L. Neuhoff, "Asymptotic analysis of optimal fixed-rate uniform scalar quantization," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 957–977, 2001.
- [97] J. Max, "Quantizing for Minimum Distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, 1960.
- [98] A. Gorokhov, "Antenna selection algorithms for MEA transmission systems," in *IEEE International Conference on Acoustics Speech and Signal Processing*. Orlando, FL, USA: IEEE, 2002.
- [99] M. B. Booth, V. Suresh, N. Michelusi, and D. J. Love, "Multi-Armed Bandit Beam Alignment and Tracking for Mobile Millimeter Wave Communications," *IEEE Communications Letters*, vol. 23, no. 7, pp. 1–1, 2019.

VITA

## VITA

Matthew B. Booth received a BSEE from the University of Kansas in 2008 and a MSEE in Digital Communications at the Air Force Institute of Technology (AFIT) in 2013. His professional experience includes Research Assistant at Information and Telecommunication Technology Center (ITTC), University of Kansas; Project Manager at Wright-Patterson AFB; Assistant Professor of Electrical and Computer Engineering (ECE) at the United States Air Force Academy; and Deputy Space Communications Mission Lead at Air Force Research Laboratory, Space Vehicles Directorate. His research interests include wireless communications and signal processing.