

ADAPTIVE BEAMFORMING AND CODING FOR MULTI-NODE WIRELESS  
NETWORKS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Dennis Osasumwen Ogbe

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. Chih-Chun Wang, 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

Dr. Michail Matthaiou

ECIT Institute, Queen's University Belfast

**Approved by:**

Dr. Dimitrios Peroulis

Head of Electrical and Computer Engineering

Gewidmet an all diejenigen, die ihre Heimat verließen um anderswo ihr Glück zu suchen.

## ACKNOWLEDGMENTS

It is an immense privilege to be able to complete a dissertation document like this one. I acknowledge with great humility that this would not have been possible without the invaluable support of a large number of people. This section is an attempt at thanking them for this support, however, no written word can accurately describe my feelings of gratitude toward the people listed on these pages.

First, I want to thank my advisors, Dr. David. J. Love and Dr. Chih-Chun Wang, for their continued and unwavering support of me and my work throughout the last six years. It is only due to their guidance and willingness to pass on their immense knowledge that I was able to start and to continue contributing to a field as vast as Electrical Engineering. Strictly speaking, without them, none of this work would have been possible.

In addition, I want to thank all members of my advisory committee: Dr. James V. Krogmeier, Dr. Michael D. Zoltowski, and Dr. Michail Matthaiou. Their continued engagement and comments have significantly improved the quality of my work. I encountered most of my committee members by taking advanced courses in Electrical Engineering at Purdue and I would like to thank them for passing on some of their vast knowledge to me.

Next, I want to thank the various funding agencies which have supported my work throughout my time at Purdue. I was supported in varying amounts by the School of Electrical and Computer Engineering at Purdue, the National Science Foundation<sup>1</sup>, DARPA, the Center for the Science of Information, and the IEEE Information Theory Society. These funds have enabled me to dig deeper and continue my work as well

---

<sup>1</sup>The work in this dissertation was supported in part by the National Science Foundation under Grant CNS-1642982, Grant CCF-1422997, and Grant CCF-1816013.

as allowed me to travel the world to present my results and engage with the broader research community.

I now want to thank all of my collaborators throughout my time at Purdue. First among them, I thank Dr. Vasanthan Raghavan for treating me like a colleague on equal footing from the first day and providing invaluable insight into the broader electrical engineering community. Next, I want to thank all of my labmates from the TASC lab at Purdue for stimulating discussions (not always about engineering), for legendary camping trips, and for long nights spent writing code and debugging the mistakes we made while doing so.

On the personal side, I would first like to thank all of the school teachers and basketball coaches that believed in me and encouraged me to become the best I could possibly be in all aspects of life. I want to thank my teachers in Munich and Nuremberg for fostering my interest in mathematics and technology early. I especially want to thank my basketball coaches from Munich, Nuremberg, and Cookeville for instilling in me the values of hard work and mental toughness.

Finally, I want to thank my family, without whom I would not be the person I am today. I want to thank my parents for raising my brother and me and enabling us to venture out into the world and chase our athletic and academic dreams. I want to thank my brother for always being with me despite our physical distance. Lastly, I want to thank my wife for going through this entire process by my side, for her unwavering support and companionship during my studies, and in advance for all of our future endeavors.

## TABLE OF CONTENTS

	Page
LIST OF TABLES . . . . .	ix
LIST OF FIGURES . . . . .	x
ABSTRACT . . . . .	xii
1 Introduction . . . . .	1
2 Noisy Beam Alignment Techniques for Reciprocal MIMO Channels . . . . .	5
2.1 Introduction . . . . .	5
2.2 System Model . . . . .	8
2.3 Power Method Using A Sequentially Estimated Channel Matrix . . . . .	11
2.3.1 Batch Least-Squares Estimator . . . . .	11
2.3.2 Optimal Sequential Least-Squares Estimator . . . . .	15
2.3.3 Suboptimal Sequential Least-Squares Estimator . . . . .	17
2.4 Summed Power Method . . . . .	18
2.5 Least-Squares Initialized Summed Power Method . . . . .	22
2.6 Numerical Studies . . . . .	23
2.6.1 Convergence Studies . . . . .	23
2.6.2 Impact of Antenna Dimensions and $k_{\text{switch}}$ . . . . .	28
2.6.3 Comparison with a Pilot-Based Channel Estimation Scheme . . . . .	29
2.7 Concluding Remarks . . . . .	32
3 Efficient Channel Estimation for Aerial Wireless Communications . . . . .	34
3.1 Introduction . . . . .	34
3.2 System Model . . . . .	38
3.3 Cramer-Rao Lower Bound . . . . .	42
3.4 Estimation Procedure . . . . .	45
3.4.1 Block-based processing . . . . .	45

	Page
3.4.2 Doppler Estimation . . . . .	48
3.4.3 Channel Estimation . . . . .	51
3.4.4 Simplified Channel Estimation using CAZAC Sequences . . . . .	52
3.5 Numerical Studies . . . . .	53
3.5.1 Comparison to theoretical optimum . . . . .	54
3.5.2 Performance under Model Mismatch . . . . .	56
3.5.3 Pulse Length and Repetition Count . . . . .	58
3.6 Concluding Remarks . . . . .	59
4 On The Optimal Delay Growth Rate of Multi-hop Line Networks . . . . .	62
4.1 Introduction . . . . .	62
4.1.1 Reliability Function . . . . .	62
4.1.2 Multi-hop Line Networks . . . . .	63
4.1.3 Reliability Function of Multi-hop Line Networks . . . . .	65
4.1.4 Our Contributions . . . . .	66
4.2 Problem Formulation . . . . .	68
4.2.1 The Multi-hop Line Network Channel Model . . . . .	68
4.2.2 Open-loop setting . . . . .	70
4.2.3 Stop-feedback setting . . . . .	72
4.2.4 Discussion #1: Decode-&-Forward is a special instance of this framework . . . . .	74
4.2.5 Discussion #2: The physical interpretation of the DAF . . . . .	76
4.3 Main Result #1: The open-loop setting . . . . .	78
4.3.1 The converse of the optimal DAF . . . . .	78
4.3.2 An achievability scheme for the setting $\ell^* = L$ . . . . .	79
4.3.3 Description of the Transmission Scheme . . . . .	80
4.3.4 DAF Analysis . . . . .	84
4.4 Main Result #2: The stop-feedback setting . . . . .	87
4.4.1 Description of the Transmission Scheme . . . . .	88

	Page
4.4.2 DAF Analysis . . . . .	101
4.5 Summary, Discussion, & Conclusion . . . . .	107
5 Conclusion . . . . .	110
A Proofs for Chapter 2 . . . . .	112
A.1 Proof of Theorem 2.3.1 . . . . .	112
A.2 Proof of Theorem 2.3.2 . . . . .	113
A.3 Proof of Theorem 2.3.3 . . . . .	114
A.4 Proof of Lemma 2.4.1 . . . . .	116
A.5 Proof of Lemma 2.4.2 . . . . .	118
A.6 Proof of Theorem 2.4.1 . . . . .	119
B Algorithms for Chapter 2 . . . . .	122
B.1 SLS Estimator (Optimal) . . . . .	122
B.2 SLS Estimator (Approximate) . . . . .	123
B.3 Summed Power Method . . . . .	123
B.4 LISP Method . . . . .	124
C Proofs for Chapter 4 . . . . .	125
C.1 Proof of Lemma 4.2.1 . . . . .	125
C.2 Proof of Lemma 4.4.2 . . . . .	128
C.3 Proof of Lemma C.2.1 . . . . .	130
C.4 Proof of Lemma C.2.2 . . . . .	132
C.5 Proof of Lemma 4.4.3 . . . . .	137
C.6 Proof of Lemma 4.4.4 . . . . .	138
REFERENCES . . . . .	141



LIST OF TABLES

Table	Page
3.1 Doppler shift to line style mapping for Figs. 3.3a)– 3.5 . . . . .	56

## LIST OF FIGURES

Figure	Page
1.1 Illustration of examples of a future heterogeneous mobile network. A mobile broadband user is served through a UAV backhaul relay. A low-latency user is served by a small cell communicating wirelessly with a fiber-equipped cell tower. In both cases, multi-hop relaying is employed. . .	2
2.1 Communication node 1 transmits data over the downlink channel $\mathbf{H}$ to node 2, while node 2 transmits data over the uplink channel $\mathbf{H}^T$ . . . . .	9
2.2 Simulation results for the i.i.d. channel model at $\rho_e = \rho_o = -10$ dB with $M_r = 4, M_t = 32$ . . . . .	25
2.3 Simulation results for the i.i.d. channel model at $\rho_e = \rho_o = 0$ dB with $M_r = 4, M_t = 32$ . . . . .	26
2.4 Simulation results for the i.i.d. channel model at $\rho_e = \rho_o = 20$ dB with $M_r = 4, M_t = 32$ . . . . .	26
2.5 Simulation results for the sparse mmWave channel model at $\rho_e = \rho_o = -10$ dB with $M_r = 4, M_t = 32$ . . . . .	27
2.6 Average value of $ \mathbf{z}^* \mathbf{H} \mathbf{f} ^2$ normalized by $\ \mathbf{H}\ _2^2$ at $k = 100$ for $\rho_e = \rho_o = -10$ dB using different channel models with $M_r = 4$ and $M_t \in 6, 8, 10, \dots, 64$ . .	30
2.7 Optimization of normalized channel gain as a function of $k_{\text{switch}}$ for different SNR regimes. $M_r = 4, M_t = 32$ . . . . .	30
2.8 Normalized channel gain with $k_{\text{max}} = 100$ for varying values of $\rho = \rho_o = \rho_e$ in the i.i.d Rayleigh fading channel case. . . . .	32
3.1 An aircraft wishes to estimate the channel taps $\mathbf{h}$ of an air-to-ground or air-to-air channel with a single dominant Doppler component. . . . .	41
3.2 The receiver drops the cyclic prefix components (red), extracts $M$ receive pulses (green, $K$ samples each), and re-orders them into the matrix of received pulses $\mathbf{Y}$ . . . . .	46
3.3 Difference between sample statistics and CRLB vs. SNR of our proposed estimator for white noise and colored noise environments. Grid search based estimator included for reference. $K = 127$ samples, $M = 5$ pulses, $L = 16$ channel taps. . . . .	57

Figure	Page
3.4 Difference between sample MSE of the channel estimate $\hat{\mathbf{h}}$ and the CRLB (3.14) vs. SNR of our proposed estimator for white noise without intra-block Doppler correction from (3.28). $K = 127$ samples, $M = 5$ pulses, $L = 16$ channel taps. . . . .	57
3.5 Increase in mean squared error relative to the bulk-only Doppler shift model when using our proposed estimator on the mismatched random Doppler shift model from (3.40). “MSE Increase” denotes the difference in mean square error performance between the matched and mismatched scenario. $K = 127$ samples, $M = 5$ pulses, $L = 16$ channel taps. Pictured for various SNR regimes. . . . .	59
3.6 Impact of sounding signal length factors on MSE for different SNRs. . . . .	60
4.1 Variations on the relay channel model. . . . .	64
4.2 Starting times, transmission duration, and encoding/decoding timepoints for an example open-loop scheme (unit: slots). . . . .	69
4.3 Relaying a message through an example 3-hop line network using the DF scheme. . . . .	75
4.4 A DAF-optimal transmission scheme for bottleneck-terminated DMC line networks. $K$ microblocks using concatenated code construction. ( $K = 3$ and $L = 3$ pictured for illustration) . . . . .	81
4.5 A DAF-optimal transmission scheme for DMC line networks with stop-feedback. Illustration for $L = 3$ hops and $K = 3$ microblocks. The bottleneck transmitter continues to send encoded microblocks after the initial $K$ microblocks are forwarded. The sequential learning phase at the bottleneck receiver lasts $K + 1$ microblock in this example. During the correction phase, the bottleneck receiver sends correction messages to the destination. . . . .	88

## ABSTRACT

Ogbe, Dennis Osasumwen PhD, Purdue University, May 2020. Adaptive Beamforming and Coding for Multi-node Wireless Networks. Major Professors: David J. Love and Chih-Chun Wang.

As wireless communications continue to permeate many aspects of human life and technology, future generations of communication networks are expected to become increasingly heterogeneous due to an explosion of the number of different types of user devices, a diverse set of available air interfaces, and a large variety of choices for the architecture of the network core. This heterogeneity, coupled with increasingly strict demands on the communication rate, latency, and fidelity demanded by a growing list of services delivered using wireless technologies, requires optimizations across the entire networking stack. Our contribution to this effort considers three key aspects of modern communication systems: First, we present a set of new techniques for multiple-input, multi-output beam alignment specifically suited for unfavorable signal-to-noise ratio regimes like the ones encountered in beamformed millimeter-wave wireless communication links. Second, we present a computationally efficient estimation algorithm for a specific class of aeronautical channels, which applies to systems designed to extend wireless coverage and communication capacity using unmanned aerial vehicles. Third, we present a new class of multi-hop relaying schemes designed to minimize communication latency with applications in the emerging domain of ultra-reliable and low-latency communications. Each of the three problem areas covered in this work is motivated by the demands of a future generation of wireless communication networks and we develop theoretical and/or numerical results outperforming the state of the art.

## 1. INTRODUCTION

Wireless communication networks have manifested themselves as a critical part of the global infrastructure and have permeated many aspects of human society and technological progress. As the demand for wireless services continues to grow exponentially, the total number of mobile wireless connections surpassed the world population at some point between the years 2018 and 2019 [1] and similar trends can be observed for the number of wireless local area networks [2].

The technologies underpinning mobile wireless networks are developed and deployed on a roughly "generational" schedule, with each generation improving on some or all aspects on its predecessors. The first generation of mobile networks, often abbreviated as 1G, provided analog voice communications for a small number of cell phone subscribers from the 1980s to the early 1990s. Second-generation (2G) networks first employed digital modulations for voice and low-rate data transmissions starting in the 1990s, with some legacy 2G systems still operational into the early 2020s. Third-generation (3G) mobile networks, first deployed in the early 2000s, provided early mobile broadband speeds and paved the way for the mass adoption of smartphones. By the time fourth-generation (4G) mobile networking technology, introduced in the 2010s, matured and penetrated the market, smartphones developed into commodity devices and mobile broadband enabled a significant fraction of the world population to consume near-real-time and high-bandwidth services such as high-quality video telephony and mobile gaming. The fifth and sixth generation (5G and 6G) of mobile networks are promising further increases in bandwidth, coverage, and the number of devices and use cases.

The increased sophistication of the user devices and the services delivered using them continues the drive the demand for higher throughput, increased coverage, and lower latencies. In an attempt to meet these ever-growing demands, the architecture

of the core networks serving the users is growing increasingly complex and heterogeneous, evolving from the traditional cellular model from 1G through 4G to more flexible and adaptive models developed as part of 5G and 6G. As an example, it is not unrealistic to expect to encounter scenarios like the one sketched in Figure 1.1 as 5G and 6G mobile networks continue to mature.

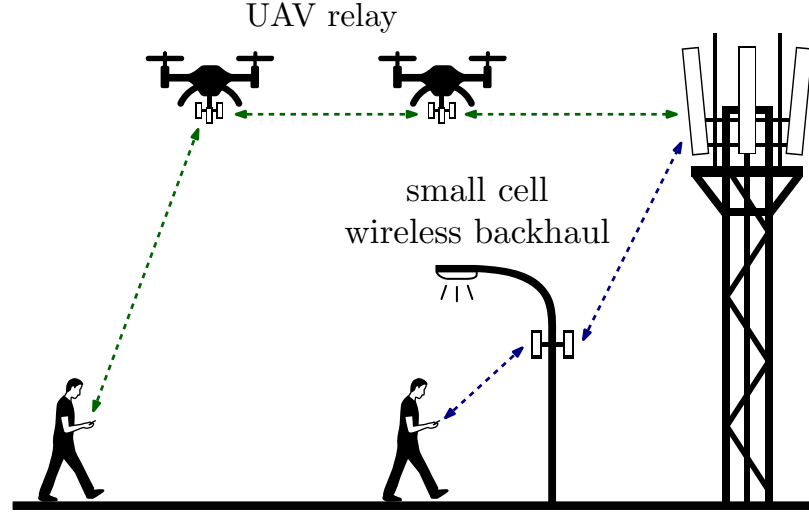


Fig. 1.1.: Illustration of examples of a future heterogeneous mobile network. A mobile broadband user is served through a UAV backhaul relay. A low-latency user is served by a small cell communicating wirelessly with a fiber-equipped cell tower. In both cases, multi-hop relaying is employed.

We highlight a few key technologies of next-generation mobile networks in Figure 1.1: Starting with late technologies of fourth-generation networks, academic and industry stakeholders distilled different fundamental "use cases" for the users of mobile networks of the fifth generation and beyond [1]. These use cases map to different requirements on the communication throughput, the latency, and the fidelity. The enhanced mobile broadband (eMBB) use case covers applications like video streaming and bulk file downloads with throughput in the 10s of Gbps. The massive machine-type communications (mMTC) use case covers applications in sensing and the Internet of Things (IoT) and envisions unprecedented devices densities. Finally, the ultra-reliable low latency (URLLC) use case covers applications which require com-

munication latencies on the order of single milliseconds. To increase coverage and capacity, many network providers are deploying so-called "small cells", which serve users while being themselves wirelessly connected to a wired base station. A straightforward evolution of this concept is to deploy flexible small cells mounted on unmanned aerial vehicles (UAVs) to either extend coverage temporarily or increase capacity in areas of peak demand. In both cases, the data between the users and their serving base station traverses multiple "hops", each with different channel characteristics. Putting it all together, we can thus envision a scenario in which a mobile broadband user accesses the network through a UAV-based multi-hop relay, while a low-latency user access the network through a small cell wireless relay, for example.

Of course, there are many more possible combinations of the aforementioned technologies, but these examples were not picked at random. Indeed, the topics covered in this dissertation all address one or more aspects of these next-generation wireless networks.

In chapter 2, we present a set of novel beam alignment techniques for multiple-input multiple-output (MIMO) time-division duplex (TDD) wireless communication links utilizing beamforming. These types of links are utilized between network users and base stations as well as backhaul links between small cells and the core network. In 5G systems, beamforming is used to compensate for the increased path loss incurred at millimeter wave (mmWave) frequencies. There exist a variety of beam alignment techniques to choose from in the literature, but the optimally-aligned beams for any channel depend on the quality of the channel state information (CSI) at each end of the link. Given the large number of antennas utilized in these systems, this CSI is difficult to obtain. In contrast, the techniques we present in chapter 2 circumvent this problem by exploiting the inherent reciprocity of wireless channels inspired by a classical subspace estimation technique called the "power method" to estimate the optimally-aligned beams.

In chapter 3, we present a novel estimation algorithm to obtain the CSI of a special set of aerial wireless channels commonly found between unmanned aerial vehicles.

One major aspect of aerial channels is the Doppler shift due to vehicle motion and its effect on the signals used to convey information. This Doppler shift can severely degrade any signals transmitted and received by UAVs, creating a need for computationally efficient signal processing schemes to estimate and compensate for it. The work in chapter 3 presents a scheme to jointly estimate the Doppler shift and the channel coefficients, again utilizing ideas from the power method subspace estimation technique.

In chapter 4, we present a novel set of transmission schemes designed to minimize communication latency over multi-hop relay channels. As we mentioned earlier, these multi-hop channels appear in a variety of places throughout the modern network. Combined with the emergence of novel low-latency use cases, a treatment of multi-hop channels from a latency perspective becomes paramount. The work in chapter 4 first lays a theoretical foundation for the analytical comparison of different transmission schemes from a latency perspective before introducing two novel transmission schemes which are designed from first principles to minimize latency.

While one common thread of this dissertation is the application of all three parts to some aspect of a modern wireless communication network, another commonality resides in the overall method of treatment of each of the subjects. In each of the subsequent chapters, we begin by identifying the need for further investigation and performance optimization of one specific aspect of the network. From this, we distill a concise theoretical model and problem statement. We then present novel algorithms or techniques to achieve the performance improvement we set to seek out in the beginning. Finally, whenever applicable, we demonstrate the viability of our approaches using numerical simulations.



## 2. NOISY BEAM ALIGNMENT TECHNIQUES FOR RECIPROCAL MIMO CHANNELS

© 2017 IEEE. Reprinted, with permission, from: 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, Oct. 2017.

### 2.1 Introduction

Advanced multi-input multi-output (MIMO) systems will be among the most important technologies to realize the ever-increasing data rate demands of 5G wireless communication networks [3, 4]. The two most promising MIMO applications<sup>1</sup>, millimeter-wave (mmWave) MIMO [5–8] and massive MIMO [9–11] rely on utilizing large beamforming gains to realize the large data rate requirements set for future 5G networks. In mmWave systems, beamforming will be used to compensate for the increased path and penetration losses in the 25–100 GHz band [12, 13], whereas massive MIMO systems will multiplex signals of different users via multi-user beamforming [14, 15] in sub-6 GHz bands.

Many recent works such as [16–19] study the information theoretic limits of beamforming with practical mmWave hardware constraints. However, the substantial gains promised by these studies can be realized only if sufficient channel state information (CSI) is available at the communication nodes. In current state-of-the-art systems, this information is acquired by the use of channel sounding sequences and feedback [20–23]. The use of a large number of antenna elements in mmWave and

---

<sup>1</sup>We use the terms massive and mmWave MIMO in the sense of the common understanding at 3GPP 5G-NR with massive MIMO typically corresponding to sub-6 GHz systems and mmWave MIMO typically corresponding to over-25 GHz systems.

massive MIMO systems will make CSI acquisition via the traditional approach impractical [24–26]. Further, in mmWave channels with a relatively small coherence period, it is not possible to *simultaneously* estimate all the elements of the channel matrix due to hardware constraints that render per-antenna sampling inefficient.

One way to circumvent this problem is to exploit the reciprocal nature<sup>2</sup> of wireless channels using time-division duplexing (TDD) systems. Channel reciprocity reduces the overall resources spent on channel sounding since CSI about the channel in one direction can be used to adapt to the channel in the reverse direction. Without readily available channel estimates, communication nodes are forced to obtain their optimal beamformer/combiner pair by sounding different beams during a beam alignment phase [27]. Furthermore, since it is desirable to minimize the usage of time and power resources of the beam alignment phase relative to actual data transmission [24–26], it is necessary to employ greedy strategies that maximize the signal-to-noise-ratio (SNR) during each time slot.

One approach to this goal is to leverage the underlying sparse structure [26, 28–31] or the directional structure [32–36] of mmWave channels via the use of low-complexity beamforming approaches. The focus of this work is on another approach that leverages greedy TDD-based beamforming. Many recent works such as [35, 37–41] have pursued this approach. The common theme that ties these works is the fact that repeated conjugation, normalization, and retransmission of an arbitrarily initialized beamforming vector through a reciprocal MIMO channel (with no noise) is akin to performing the power method<sup>3</sup> on the channel matrix.

Beam alignment algorithms based on the power method are attractive due to their simplicity and low computational complexity. However, simple implementations like the ones proposed in [35, 37, 38] are likely to perform poorly in the low-SNR regime [35]. Other approaches for finding good beams using the power method have been proposed

---

<sup>2</sup>This work assumes that the radio-frequency (RF) circuit asymmetries in the uplink and downlink have been compensated via calibration and hence does not consider these aspects.

<sup>3</sup>The power method is a result from numerical linear algebra which provides a simple algorithm to find the dominant eigenvector(s)/eigenspace of a matrix [42].

in [40] and [43]. These techniques offer improvements on the robustness and speed of convergence of the basic power method at the cost of additional complexity. The main idea behind these improved techniques is to combine previous estimates of the optimal beams with the received information during each time slot. In addition to these techniques, recent works such as [44] study the application of the more general Arnoldi iteration to the beam alignment problem. Furthermore, feedback-based beam alignment techniques for frequency-division duplexing (FDD) systems, which represent the majority of currently deployed commercial systems, have been studied in [45] and [46].

Building on [40, 43, 44], this section presents multiple novel techniques for the TDD MIMO beam alignment problem in reciprocal channels. These techniques improve upon the performance of the simple power method-based algorithms, especially in low-SNR environments which are typical of practical mmWave systems [35]. The first technique, labeled the *sequential least-squares method*, is based on constructing a least-squares estimate of the channel matrix sequentially using the previously-used sounding beams. The channel estimates at each iteration can then be used to compute the next sounding beamformer/combiner pair, which is exchanged through a feedback<sup>4</sup> link. The second technique, labeled the *summed power method*, does not require a feedback link and computes a normalized running sum of the previous beamformers, thus gaining greater robustness against noise through averaging.

The first technique achieves better performance in the high-SNR regime at the cost of additional complexity and feedback<sup>5</sup> overhead. On the other hand, the second technique achieves better performance in the low-SNR regime and yet does not need significant complexity/feedback overhead. However, this technique has deteriorating performance as the SNR increases due to continued noise averaging. To enjoy the complementary advantages of both techniques, we propose a third technique, labeled

---

<sup>4</sup>Due to the small packet overheads, the feedback link is assumed to be ideal: error-free and incurring no delay.

<sup>5</sup>Nevertheless, the feedback link in itself is not onerous given that mmWave links are expected to support Gbps rates.

the *least-squares initialized summed power method*, that switches from the first technique to the second technique after a certain number of iterations. By appropriately choosing the switching point  $k_{\text{switch}}$ , significant performance improvement can be realized in the high-SNR regime with a small increase in feedback and computational complexity. The motivation behind the third technique is that the high-SNR performance of a beam alignment algorithm critically depends on the beam initialization. By choosing this initialization from a scheme that rejects noise near-optimally, we are able to prime a low-complexity scheme and improve performance. Thus, the proposed approaches in this chapter provide useful low-complexity solutions for realizing the large beamforming gains of mmWave systems.

The rest of this chapter is organized as follows. Section 2.2 provides an overview of the system model. Sections 2.3 — 2.5 give detailed descriptions of the power methods proposed. Simulation results illustrating the advantages of the proposed techniques are presented in Section 2.6 with concluding remarks provided in Section 2.7.

**Notations:** The following notations are used in this chapter. Bold upper-case and lower-case letters (such as  $\mathbf{A}$  and  $\mathbf{a}$ ) denote matrices and column vectors, respectively. The operators  $(\cdot)^T$ ,  $\overline{(\cdot)}$ ,  $(\cdot)^*$  and  $(\cdot)^\dagger$  denote matrix transposition, element-wise complex conjugation, matrix Hermitian transposition and Moore-Penrose pseudoinverse operations, respectively.  $\|\cdot\|_2$  denotes the vector  $\ell_2$ -norm and  $\|\cdot\|_F$  denotes the Frobenius norm of a matrix.  $\mathbf{x} \sim \mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  denotes a complex Gaussian random vector with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ .  $\mathbb{C}^{n \times m}$ ,  $\mathbb{C}^n$  and  $\mathbb{E}\{\cdot\}$  stand for the space of  $n \times m$  complex matrices,  $n \times 1$  complex vectors and the expectation operator, respectively.

## 2.2 System Model

We consider a multi-antenna communication system such as the one shown in Fig. 2.1, consisting of two transceivers (communication nodes), with  $M_t$  antennas at node 1 and  $M_r$  antennas at node 2. The two nodes communicate over a channel

$\mathbf{H} \in \mathbb{C}^{M_r \times M_t}$ . We also assume that  $\mathbf{H}$  is reciprocal, i.e. the channel matrix from node 2 to node 1 (uplink) is the transpose of the channel matrix from node 1 to node 2 (downlink). For a transmission on the downlink channel, the transmit data at node 1 is precoded by a unit-norm transmit beamforming vector  $\mathbf{f} = [f_1 \ f_2 \ \dots \ f_{M_t}]^T \in \mathbb{C}^{M_t}$ , sent over the channel, and combined at node 2 with a unit-norm receive combiner  $\mathbf{z} = [z_1 \ z_2 \ \dots \ z_{M_r}]^T \in \mathbb{C}^{M_r}$ . Hence, for a data symbol  $s_o[k]$  sent on the downlink channel, we obtain the received symbol

$$r_o[k] = \sqrt{\rho_o} \mathbf{z}^* \mathbf{H} \mathbf{f} s_o[k] + n_o[k], \quad (2.1)$$

where  $\rho_o$  is the downlink SNR and  $n_o[k] \sim \mathcal{CN}(0, 1)$  is additive Gaussian noise, which we assume to be independent and identically distributed (i.i.d.) spatially as well as temporally. Similarly, for a data symbol  $s_e[k]$  sent on the uplink channel, node 1 obtains the received symbol

$$r_e[k] = \sqrt{\rho_e} \mathbf{f}^T \mathbf{H}^T \bar{\mathbf{z}} s_e[k] + n_e[k]. \quad (2.2)$$

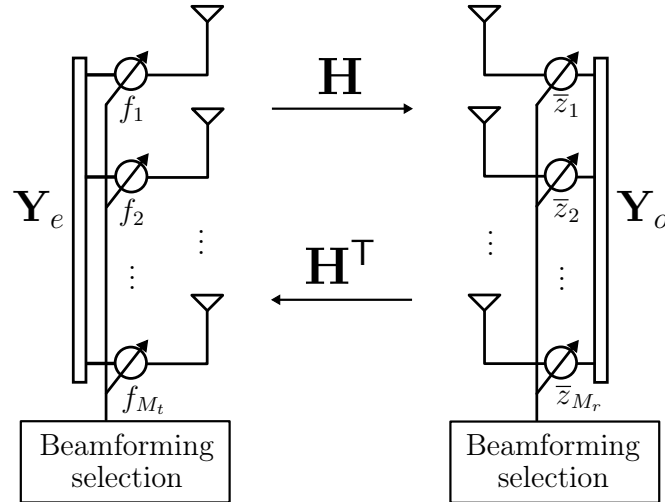


Fig. 2.1.: Communication node 1 transmits data over the downlink channel  $\mathbf{H}$  to node 2, while node 2 transmits data over the uplink channel  $\mathbf{H}^T$ .

In both (2.1) and (2.2), we denote  $|\mathbf{z}^* \mathbf{H} \mathbf{f}|^2 = |\mathbf{f}^\top \mathbf{H}^\top \bar{\mathbf{z}}|^2$  as the effective channel gain, which we want to maximize in order to achieve reliable communications and the highest possible data rates in both directions. We denote the vectors that achieve this as  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{z}_{\text{opt}}$ , respectively. It is well-known from [47, 48] that the effective channel gain is maximized when  $\mathbf{f}$  and  $\mathbf{z}$  are the right- and left-singular vectors of  $\mathbf{H}$  corresponding to the largest singular value of  $\mathbf{H}$  and that its maximum achievable value is  $\|\mathbf{H}\|_2^2 = \lambda_{\max}(\mathbf{H}^* \mathbf{H})$ . Further, we assume that neither node has knowledge of the channel. It is therefore impossible for either node to compute the estimates of  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{z}_{\text{opt}}$  using the singular value decomposition (SVD) of their channel estimate. Instead, as mentioned earlier, these estimates are obtained iteratively.

In the proposed techniques, both nodes cooperatively determine  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{z}_{\text{opt}}$  during a beam training phase by exploiting the channel's reciprocity property. To model this, our system operates on a *ping-pong* observation framework, which divides each discrete channel use into two time slots. During slot 1 (*ping*), node 1 sends a training symbol to node 2 on the downlink channel  $\mathbf{H}$ . During slot 2 (*pong*), node 2 sends a training symbol back to node 1 on the uplink channel  $\mathbf{H}^\top$ . Since the two nodes are exchanging training symbols that are known to both sides, we focus on the received signal vectors after correlating with the known training data. Hence, the observation at slot 1 (at node 2) during the  $k$ -th channel use is given as

$$\mathbf{y}_o[k] = \sqrt{\rho_o} \mathbf{H} \mathbf{f}[k] + \mathbf{n}_o[k]. \quad (2.3)$$

In (2.3), the term  $\mathbf{f}[k]$  denotes an estimate of  $\mathbf{f}_{\text{opt}}$  at training phase time-index  $k$  and  $\mathbf{n}_o[k] \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$  is a complex Gaussian noise vector of size  $M_r$ . Due to the reciprocity of the uplink and downlink channels, the observation at slot 2 (at node 1) is given as

$$\mathbf{y}_e[k] = \sqrt{\rho_e} \mathbf{H}^\top \bar{\mathbf{z}}[k] + \mathbf{n}_e[k]. \quad (2.4)$$

Similar to (2.3),  $\rho_e$  denotes the uplink SNR,  $\mathbf{z}[k]$  denotes an estimate of  $\mathbf{z}_{\text{opt}}$  at training phase time-index  $k$  and  $\mathbf{n}_e[k] \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$  is a complex Gaussian noise vector of size  $M_t$ .

## 2.3 Power Method Using A Sequentially Estimated Channel Matrix

### 2.3.1 Batch Least-Squares Estimator

In the first scheme, since the channel matrix is not known at either node, the nodes construct a least-squares estimate of  $\mathbf{H}$  before each ping-pong time slot using all of the previous estimates of  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{z}_{\text{opt}}$ . These estimates are then used to compute the next state of their beamforming vectors.

In particular, using all observations up to time slot  $k$ , we can write (2.3) and (2.4) in matrix form as

$$\mathbf{Y}_{o,k} = \sqrt{\rho_o} \mathbf{H} \mathbf{F}_k + \mathbf{N}_{o,k} \quad (2.5)$$

and

$$\mathbf{Y}_{e,k} = \sqrt{\rho_e} \mathbf{H}^T \bar{\mathbf{Z}}_k + \mathbf{N}_{e,k}. \quad (2.6)$$

In (2.5) and (2.6),  $\mathbf{F}_k = [\mathbf{f}[0] \ \mathbf{f}[1] \ \dots \ \mathbf{f}[k]]$  and  $\mathbf{Z}_k = [\mathbf{z}[0] \ \mathbf{z}[1] \ \dots \ \mathbf{z}[k]]$  contain all of the estimates of  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{z}_{\text{opt}}$  up to time-index  $k$ . Also,  $\mathbf{Y}_{o,k} = [\mathbf{y}_o[0] \ \mathbf{y}_o[1] \ \dots \ \mathbf{y}_o[k]]$  and  $\mathbf{Y}_{e,k} = [\mathbf{y}_e[0] \ \mathbf{y}_e[1] \ \dots \ \mathbf{y}_e[k]]$  contain all of the observed signal vectors, respectively. On the other hand,  $\mathbf{N}_{o,k} = [\mathbf{n}_o[0] \ \mathbf{n}_o[1] \ \dots \ \mathbf{n}_o[k]]$  and  $\mathbf{N}_{e,k} = [\mathbf{n}_e[0] \ \mathbf{n}_e[1] \ \dots \ \mathbf{n}_e[k]]$  contain all of the noise vectors, respectively.

Based on this information, node 1 constructs an estimate of the channel by solving the least-squares problem

$$\hat{\mathbf{H}}_{e,k} = \underset{\tilde{\mathbf{H}} \in \mathbb{C}^{M_r \times M_t}}{\text{argmin}} \left( \left\| \mathbf{Y}_{e,k-1}^T - \sqrt{\rho_e} \mathbf{Z}_{k-1}^* \tilde{\mathbf{H}} \right\|_F^2 \right). \quad (2.7)$$

Similarly, node 2 constructs an estimate of the channel by solving

$$\hat{\mathbf{H}}_{o,k} = \underset{\tilde{\mathbf{H}} \in \mathbb{C}^{M_r \times M_t}}{\operatorname{argmin}} \left( \left\| \mathbf{Y}_{o,k} - \sqrt{\rho_o} \tilde{\mathbf{H}} \mathbf{F}_k \right\|_F^2 \right). \quad (2.8)$$

Note that there exists an asymmetry in the time-index between (2.7) and (2.8). The solutions to these least-squares problems ( $\hat{\mathbf{H}}_{e,k}$  and  $\hat{\mathbf{H}}_{o,k}$ ) are obtained using all of the previously observed outputs and beamforming vectors and are as follows:

$$\hat{\mathbf{H}}_{e,k} = \frac{(\mathbf{Z}_{k-1}^*)^\dagger \mathbf{Y}_{e,k-1}^\top}{\sqrt{\rho_e}}, \quad (2.9)$$

$$\hat{\mathbf{H}}_{o,k} = \frac{\mathbf{Y}_{o,k} (\mathbf{F}_k)^\dagger}{\sqrt{\rho_o}}. \quad (2.10)$$

In (2.9) and (2.10), the  $(\cdot)^\dagger$  operation stands for the Moore-Penrose pseudoinverse<sup>6</sup> of the underlying matrix. Using the definitions of the pseudoinverse, we have the following simplifications:

$$\hat{\mathbf{H}}_{e,k} = \frac{1}{\sqrt{\rho_e}} \cdot \begin{cases} \mathbf{Z}_{k-1} (\mathbf{Z}_{k-1}^* \mathbf{Z}_{k-1})^{-1} \mathbf{Y}_{e,k-1}^\top & \text{if } k < M_r \\ (\mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^*)^{-1} \mathbf{Z}_{k-1} \mathbf{Y}_{e,k-1}^\top & \text{if } k \geq M_r, \end{cases} \quad (2.11)$$

$$\hat{\mathbf{H}}_{o,k} = \frac{1}{\sqrt{\rho_o}} \cdot \begin{cases} \mathbf{Y}_{o,k} (\mathbf{F}_k^* \mathbf{F}_k)^{-1} \mathbf{F}_k^* & \text{if } k < M_t \\ \mathbf{Y}_{o,k} \mathbf{F}_k^* (\mathbf{F}_k \mathbf{F}_k^*)^{-1} & \text{if } k \geq M_t. \end{cases} \quad (2.12)$$

Note that the second condition in both (2.11) and (2.12) has been separated (from the first) at the  $k = M_r$  and  $k = M_t$  cases artificially. Since the solutions in (2.9) and (2.10) use all the underlying data up to time-index  $k$ , we call this approach the *batch least-squares method*. Once  $\hat{\mathbf{H}}_{e,k}$  and  $\hat{\mathbf{H}}_{o,k}$  have been estimated, beamforming vector computation follows directly from the SVD theorem [42, 49].

---

<sup>6</sup>Note that the expressions in (2.9) and (2.10) hold even in the case when  $k < M_r$  since it can be shown that the left pseudoinverse of a “tall matrix”, i.e., a  $K_1 \times K_2$  matrix with  $K_1 > K_2$  minimizes  $\|\mathbf{A}\mathbf{C} - \mathbf{I}\|^2$ , where  $\mathbf{C}$  is optimized over all  $K_2 \times K_1$  matrices.



**Lemma 2.3.1** *Let the SVD of a matrix  $\mathbf{A}$  be denoted as  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$ . We can obtain a multiple of the  $i$ -th left-singular vector of  $\mathbf{A}$  by multiplying  $\mathbf{A}$  with its  $i$ -th right-singular vector, i.e.,  $\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i$ . Here,  $\sigma_i$  is the  $i$ -th singular value. Similarly, we can obtain a multiple of the  $i$ -th right-singular vector by multiplying  $\mathbf{A}^*$  with its  $i$ -th left-singular vector, i.e.,  $\mathbf{A}^*\mathbf{u}_i = \sigma_i\mathbf{v}_i$ .*

Applying Lemma 2.3.1, we note that node 1 can compute its  $k$ -th estimate for  $\mathbf{f}_{\text{opt}}$  as

$$\mathbf{f}[k] = \frac{\widehat{\mathbf{H}}_{e,k}^* \mathbf{z}[k-1]}{\left\| \widehat{\mathbf{H}}_{e,k}^* \mathbf{z}[k-1] \right\|_2}. \quad (2.13)$$

Similarly, applying Lemma 2.3.1, we note that node 2 obtains its  $k$ -th estimate for  $\mathbf{z}_{\text{opt}}$  as

$$\mathbf{z}[k] = \frac{\widehat{\mathbf{H}}_{o,k} \mathbf{f}[k]}{\left\| \widehat{\mathbf{H}}_{o,k} \mathbf{f}[k] \right\|_2}. \quad (2.14)$$

Some comments are in order at this stage.

1. We have the following result on error covariance matrices with the batch estimators.

**Theorem 2.3.1** *If  $k \geq \max(M_t, M_r)$ , the error covariance matrices of the columns of  $\widehat{\mathbf{H}}_{e,k}$  and  $\widehat{\mathbf{H}}_{o,k}$  under the assumption of a channel  $\mathbf{H}$  with independent and identically distributed (i.i.d.) entries are given as*

$$\mathbf{C}_{e,k} = \frac{1}{\rho_e} (\mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^*)^{-1} \quad (2.15)$$

and

$$\mathbf{C}_{o,k} = \frac{1}{\rho_o} (\mathbf{F}_k \mathbf{F}_k^*)^{-1}, \quad (2.16)$$

respectively.

For the proof, see Appendix A.1.

2. The proposed algorithm is valid for a general channel matrix  $\mathbf{H}$  and the i.i.d. assumption has been made only in the context of Theorem 2.3.1. From Theorem 2.3.1, we note that the estimation error is monotonically decreasing in the SNRs,  $\rho_e$  and  $\rho_o$ . This shows that a reasonable channel estimate can be obtained in the medium- to high-SNR regimes. Nevertheless, the low-SNR regime is typical in mmWave systems, especially with self-blocking or blocking due to other humans, vehicles, buildings, foliage, etc. [13,35]. Thus, Section 2.6 studies the performance of the different approaches proposed in this work as a function of the SNR as well as for both i.i.d. and sparse channel models.
3. While we need  $\rho_e$  and  $\rho_o$  to compute  $\hat{\mathbf{H}}_{e,k}$  and  $\hat{\mathbf{H}}_{o,k}$ , the beamformer estimates do not depend on these quantities. Therefore, a mismatched estimate of  $\rho_e$  and  $\rho_o$  is still sufficient to implement the proposed scheme.
4. The computation of  $\hat{\mathbf{H}}_{e,k}$  and  $\mathbf{f}[k]$  at node 1 requires the feedback of  $\mathbf{z}[k-1]$  from node 2. Similarly, computation of  $\hat{\mathbf{H}}_{o,k}$  and  $\mathbf{z}[k]$  at node 2 requires the feed forward of  $\mathbf{f}[k]$  from node 1. While on a first glance this feedback and feed forward sounds onerous, given the Gbps rates that mmWave systems are expected to realize, these feedback overheads can be supported on either a lower frequency control/data channel or on a mmWave control channel. This feedback/feed forward has to be specified only over a large sub-band (a component carrier, for example) or on a wideband basis, further reducing the overhead. Thus, it makes sense to not dismiss this approach as impractical and study its performance gain relative to other competing approaches. This is the subject of Section 2.6. We will also consider other lower feedback overhead approaches in Sections 2.4 and 2.5.
5. Throughout this text, we are assuming that the initial transmit beam  $\mathbf{f}[0]$  is a unit-norm complex random vector. An alternative approach which could

be considered for channels with a large line-of-sight component would be to initialize  $\mathbf{f}[0]$  with an omni-directional beam pattern that approximates equal gain in every spatial direction. Omni-directional beams have been constructed and used in [27], but are out of the scope of this work.

The batch least-squares estimators are obtained by computing the Moore-Penrose pseudoinverse. The complexity in computing these estimators in (2.11) and (2.12) is limited to the inversion of a  $\tilde{k} \times \tilde{k}$  matrix where  $\tilde{k} = \min(M_r, k)$  in the former case and  $\tilde{k} = \min(M_t, k)$  in the latter case. However, computation of the matrix to be inverted requires a multiplication count that scales with  $k$  and can hence be onerous.

### 2.3.2 Optimal Sequential Least-Squares Estimator

Following a similar approach to [50], we therefore propose a sequential algorithm that updates each previous channel estimate based on the current received signal vector. This approach minimizes computational burden as well as eliminates the need to store all of the previously received signal. Since (2.13) uses the conjugate transpose of the channel to compute a new beamformer, we use an algorithm that directly computes an estimate for  $\hat{\mathbf{H}}_{e,k}^*$  instead of  $\hat{\mathbf{H}}_{e,k}$ . This choice is made here simply to make the derivation of the sequential formulas more consistent between the two nodes. In this setup, the sequential version of (2.9) (the channel estimator update) is given as

$$\hat{\mathbf{H}}_{e,k}^* = \hat{\mathbf{H}}_{e,k-1}^* + \left( \frac{\bar{\mathbf{y}}_e[k-1]}{\sqrt{\rho_e}} - \hat{\mathbf{H}}_{e,k-1}^* \mathbf{z}[k-1] \right) \mathbf{K}_{e,k} \quad (2.17)$$

where

$$\mathbf{K}_{e,k} = \frac{\mathbf{z}^*[k-1] \mathbf{C}_{e,k-1}}{1 + \mathbf{z}^*[k-1] \mathbf{C}_{e,k-1} \mathbf{z}[k-1]} \quad (2.18)$$

and the covariance matrix update is given as

$$\mathbf{C}_{e,k} = \mathbf{C}_{e,k-1} (\mathbf{I} - \mathbf{z}[k-1]\mathbf{K}_{e,k}). \quad (2.19)$$

After obtaining  $\hat{\mathbf{H}}_{e,k}^*$ , node 1 uses (2.13) to obtain the  $k$ -th estimate for  $\mathbf{f}_{\text{opt}}$ . The value of this beamformer then needs to be fed back to node 2, where it will be used to obtain the next estimate for  $\mathbf{z}_{\text{opt}}$ .

At node 2, the same sequential algorithm is used to solve the least-squares problem, and the update expression for  $\hat{\mathbf{H}}_{o,k}$  becomes

$$\hat{\mathbf{H}}_{o,k} = \hat{\mathbf{H}}_{o,k-1} + \left( \frac{\mathbf{y}_o[k]}{\sqrt{\rho_o}} - \hat{\mathbf{H}}_{o,k-1}\mathbf{f}[k] \right) \mathbf{K}_{o,k} \quad (2.20)$$

where

$$\mathbf{K}_{o,k} = \frac{\mathbf{f}^*[k]\mathbf{C}_{o,k-1}}{1 + \mathbf{f}^*[k]\mathbf{C}_{o,k-1}\mathbf{f}[k]} \quad (2.21)$$

with the covariance matrix update

$$\mathbf{C}_{o,k} = \mathbf{C}_{o,k-1} (\mathbf{I} - \mathbf{f}[k]\mathbf{K}_{o,k}). \quad (2.22)$$

Node 2 then obtains  $\mathbf{z}[k]$  from (2.14), which in turn is fed back to node 1 to compute  $\mathbf{f}[k+1]$ .

We observe that these sequential least-squares (SLS) estimators are only equivalent to their batch estimators when the beamformer matrices  $\mathbf{F}_k$  and  $\mathbf{Z}_k$  are of full column rank. That is, for  $k \leq \text{rank}[\mathbf{H}]$ , both nodes would need to compute their channel estimates using the batch approach. Theorem 2.3.2 establishes that the sequential approach is equivalent to using the batch estimator for all  $k$ .

**Theorem 2.3.2** *The sequential least-squares estimator  $\hat{\mathbf{H}}_{o,k}^{\text{Seq}}$  is identical to the batch least-squares estimator  $\hat{\mathbf{H}}_{o,k}^{\text{Batch}}$  for  $k > r$  if  $\hat{\mathbf{H}}_{o,r}^{\text{Seq}} = \hat{\mathbf{H}}_{o,r}^{\text{Batch}}$  where  $r = \text{rank}[\mathbf{H}]$ .*

For the proof, see Appendix A.2.

Motivated by Theorem 2.3.2, we propose to initialize  $\mathbf{f}[0]$  as a complex random unit-norm vector. We then use the batch estimator from (2.9) and (2.10) for  $k \leq \text{rank}[\mathbf{H}]$  and switch to the sequential estimator for  $k > \text{rank}[\mathbf{H}]$ . Under these assumptions, the Gauss-Markov Theorem states that the least-squares estimator is the best linear unbiased estimator (BLUE) for the channel matrix  $\mathbf{H}$  [50]. The asymptotic normality property of the least-squares estimator [51] then shows how our sequential estimates for the channel matrix converge to its true value. As the channel estimate becomes more accurate with the number of iterations, steps (2.13) and (2.14) essentially perform a two-iteration power method without noise, which converges at a rate of  $(\sigma_1/\sigma_2)^2$  [42]. The description under Algorithm B.1 (see Appendix B.1) gives a succinct summary of this technique (labeled as *SLS Estimator (Optimal)*) corresponding to stopping at  $k_{\max}$  iterations, where  $k_{\max}$  is chosen appropriately.

### 2.3.3 Suboptimal Sequential Least-Squares Estimator

For large antenna dimensions as is typical in mmWave systems, it can be computationally difficult to use the batch estimator for the first  $M_t$  iterations. In this case, we initialize the sequential least-squares estimator with an arbitrary initial covariance estimate. With such a choice, the following result shows that we are guaranteed to asymptotically approach the batch least-squares estimate.

**Theorem 2.3.3** *The sequential least squares estimate  $\hat{\mathbf{H}}_{o,k}^{\text{Seq}}$ , initialized with  $\mathbf{C}_{o,0} = \alpha \mathbf{I}$  approaches the batch least-squares estimate  $\hat{\mathbf{H}}_{o,k}^{\text{Batch}}$  as  $\alpha \rightarrow \infty$ .*

For the proof, see Appendix A.3

Using Theorem 2.3.3, the alternative algorithm (labeled as *SLS Estimator (Suboptimal)*) also requires us to initialize  $\mathbf{f}[0]$  as a complex random unit-norm vector. The

nodes then transmit this vector across  $\mathbf{H}$  according to (2.3) and (2.4) and compute their initial rank-1 channel estimates and beamforming vectors as follows.

$$\hat{\mathbf{H}}_{o,0} = \frac{\mathbf{y}_o[0]\mathbf{f}^*[0]}{\sqrt{\rho_o}} \quad (2.23)$$

$$\mathbf{z}[0] = \frac{\hat{\mathbf{H}}_{o,0}\mathbf{f}[0]}{\left\|\hat{\mathbf{H}}_{o,0}\mathbf{f}[0]\right\|_2} = \frac{\mathbf{y}_o[0]}{\|\mathbf{y}_o[0]\|_2} \quad (2.24)$$

$$\hat{\mathbf{H}}_{e,1}^* = \frac{\bar{\mathbf{y}}_e[0]\mathbf{z}^*[0]}{\sqrt{\rho_e}} \quad (2.25)$$

$$\mathbf{f}[1] = \frac{\hat{\mathbf{H}}_{e,1}^*\mathbf{z}[0]}{\left\|\hat{\mathbf{H}}_{e,1}^*\mathbf{z}[0]\right\|_2}. \quad (2.26)$$

The nodes then initialize  $\mathbf{C}_{o,0} = \mathbf{C}_{e,1} = \alpha\mathbf{I}$  for an appropriately chosen  $\alpha$ . The nodes then use the sequential formulas (2.17) — (2.22) to estimate their beamformers. To conclude this section, Algorithm B.2 (see Appendix B.2) provides a brief summary of this technique corresponding to  $k_{\max}$  iterations.

## 2.4 Summed Power Method

We now propose an alternate approach, labeled the *summed power method*, to align the beams at the two nodes. The main idea behind this scheme is that both nodes calculate their next beamformers as a function of the running sum of their previously received vectors, effectively averaging out noise in the estimation process. This low-complexity approach adds only one additional vector addition per iteration at each node when compared to the simple power method [35,37]. Additionally, there is no need for a feedback link, as neither node needs to have knowledge of the other node's beamformer.

As described in Section 2.2, both nodes exchange training symbols according to (2.3) and (2.4). However, instead of simply conjugating and retransmitting their received vector as in the simple power method, both nodes obtain their next beam-

formers from a running sum of all of their previous received vectors. At each time-index  $k$ , node 1 computes its next beamformer as

$$\mathbf{f}[k+1] = \alpha_k [\bar{\mathbf{y}}_e[k] + \bar{\mathbf{y}}_e[k-1] + \cdots + \bar{\mathbf{y}}_e[0]] \quad (2.27)$$

$$= \alpha_k \mathbf{s}_e[k]. \quad (2.28)$$

Similarly, node 2 computes its next beamformer as

$$\mathbf{z}[k+1] = \beta_k [\mathbf{y}_o[k] + \mathbf{y}_o[k-1] + \cdots + \mathbf{y}_o[0]] \quad (2.29)$$

$$= \beta_k \mathbf{s}_o[k]. \quad (2.30)$$

In (2.28) and (2.30),  $\mathbf{s}_e[k]$  and  $\mathbf{s}_o[k]$  are the state vectors at each node which hold the running sum of the received vectors. The terms  $\alpha_k$  and  $\beta_k$  are normalization factors ensuring the unit-norm constraint and are given as

$$\alpha_k = \frac{1}{\|\mathbf{s}_e[k]\|_2} \quad (2.31)$$

and

$$\beta_k = \frac{1}{\|\mathbf{s}_o[k]\|_2}. \quad (2.32)$$

Algorithm B.3 (see Appendix B.3) provides an overview of the proposed technique.

For further analysis of the proposed algorithm, it is useful to define the state-space model of the combined system state:  $\mathbf{s}[k] = [\mathbf{s}_e^T[k] \ \mathbf{s}_o^T[k]]^T$ . A straightforward simplification of  $\mathbf{s}[k]$  shows that

$$\mathbf{s}[k] = \begin{bmatrix} \mathbf{s}_e[k] \\ \mathbf{s}_o[k] \end{bmatrix} \quad (2.33)$$

$$= \begin{bmatrix} \mathbf{I} & \sqrt{\rho_e} \beta_{k-1} \mathbf{H}^* \\ \sqrt{\rho_o} \alpha_{k-1} \mathbf{H} & \mathbf{I} \end{bmatrix} \mathbf{s}[k-1] + \mathbf{n}[k] \quad (2.34)$$

$$= \prod_{i=0}^{k-1} \begin{bmatrix} \mathbf{I} & \sqrt{\rho_e} \beta_{k-1-i} \mathbf{H}^* \\ \sqrt{\rho_o} \alpha_{k-1-i} \mathbf{H} & \mathbf{I} \end{bmatrix} \mathbf{s}[0] \\ + \sum_{\ell=1}^k \prod_{j=\ell}^{k-1} \begin{bmatrix} \mathbf{I} & \sqrt{\rho_e} \beta_{k-1+\ell-j} \mathbf{H}^* \\ \sqrt{\rho_o} \alpha_{k-1+\ell-j} \mathbf{H} & \mathbf{I} \end{bmatrix} \mathbf{n}[\ell] \quad (2.35)$$

where

$$\mathbf{n}[k] = \begin{bmatrix} \bar{\mathbf{n}}_e[k] \\ \mathbf{n}_o[k] \end{bmatrix}. \quad (2.36)$$

Without loss in generality, we can transform an  $M_r \times M_t$  channel matrix to an  $M \times M$  channel matrix by appending zero columns/rows where  $M = \max(M_r, M_t)$ . Thus, we restrict attention to square channel matrices. We can also assume that  $\rho_e = \rho_o = \rho$  without loss in generality to simplify the convergence studies. While establishing a convergence result under the general Rayleigh fading model appears difficult, we now establish this under certain restrictions. Nevertheless, numerical studies in Section 2.6 show that convergence of the summed power method holds true even for general channel matrix settings. These assumptions (listed as Hypotheses 1-3) are as follows:

- **Hypothesis 1:** Since convergence studies make more sense in the high-SNR regime, we assume that  $\rho \gg 1$ .



- **Hypothesis 2:** Let  $\mathbf{f}[i] = [f_{i,1}, \dots, f_{i,M}]^\top$  and  $\mathbf{z}[i] = [z_{i,1}, \dots, z_{i,M}]^\top$ . We make the assumptions that as  $k$  increases,  $\sum_{i=0}^k f_{i,n} \approx C_k$  for all  $n$  and  $\sum_{i=0}^k z_{i,m} \approx C_k$  for all  $m$ . In other words, the statistics of the beamformers remain invariant to the antenna indices at either node as  $k$  increases.
- **Hypothesis 3:** We consider real-valued, diagonal channel matrices  $\mathbf{H} = \text{Diag}([h_1, \dots, h_M])$  with diagonal elements ordered in non-increasing order. These assumptions can be viewed as restricting all the signal processing to happen within the bases corresponding to the left- and right-singular vectors of  $\mathbf{H}$ . Also, assume that  $h_1 > h_2$  implying a singular dominant eigen-mode for  $\mathbf{H}$ .

We now discuss the behavior of the summed power method as  $k$  (the number of iterations) increases under the above assumptions. Under Hypothesis 3, it can be seen that the optimal beamformers reduce to a scaled version of the first column of the  $M \times M$ -dimensional identity matrix, denoted as  $\mathbf{e}_1$ . Thus, the desired state vector is  $\mathbf{s}_{\text{opt}} = [\alpha \mathbf{e}_1^\top \ \beta \mathbf{e}_1^\top]^\top = \begin{bmatrix} \alpha & 0 & \dots & 0 & \beta & 0 & \dots & 0 \end{bmatrix}^\top$  for some  $\alpha$  and  $\beta$ . The impreciseness in the choice of  $\alpha$  and  $\beta$  is because the beamforming vector is defined only up to a point on the Grassmann manifold [20, 21, 52].

Convergence of the summed power method is equivalent to the limiting behavior/convergence of  $\mathbf{s}[k]$  from (2.33) to  $\mathbf{s}_{\text{opt}}$ . Lemma 2.4.1 provides a preliminary result needed to establish this convergence result.

**Lemma 2.4.1** *Under Hypothesis 3, the state transition matrix from (2.35) is diagonalized by*

$$\mathbf{U}_{k-1} = \begin{bmatrix} \sqrt{\frac{\beta_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} & \sqrt{\frac{\beta_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} \\ \sqrt{\frac{\alpha_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} & -\sqrt{\frac{\alpha_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} \end{bmatrix}. \quad (2.37)$$

For the proof, see Appendix A.4.

Note that  $\mathbf{U}_{k-1}$  is not unitary for general  $\alpha_{k-1}$  and  $\beta_{k-1}$ . However, we have the following additional result that simplifies  $\mathbf{U}_{k-1}$ .

**Lemma 2.4.2** *Under Hypotheses 1-3, we can assume that  $\alpha_k \approx \beta_k$  for large  $k$ . Thus, as  $k$  increases,  $\mathbf{U}_{k-1}$  converges to*

$$\tilde{\mathbf{U}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}. \quad (2.38)$$

For the proof, see Appendix A.5.

We now have the following main result.

**Theorem 2.4.1** *Under Hypotheses 1-3, we have that  $\mathbf{s}[k] \rightarrow \mathbf{s}_{\text{opt}}$  as  $k$  increases.*

For the proof, see Appendix A.6.

The results of Section 2.6 will show that these results hold for more general channel models and are not restricted to satisfaction of Hypotheses 1-3. In addition, Section 2.5 presents two modifications to the summed power method which aim to improve performance over a wider range of SNRs while maintaining low computational complexity.

## 2.5 Least-Squares Initialized Summed Power Method

We now consider a refinement that trades off the advantages of both the approaches in Sections 2.3 and 2.4 in terms of complexity, feedback and performance. The main motivation behind this approach is the observation that the performance of a beam alignment algorithm critically depends on how  $\mathbf{f}[0]$  (or  $\mathbf{z}[0]$ ) is initialized. When  $\mathbf{f}[0]$  is initialized as a complex random unit-norm vector, we rely on multiple iterations over the channel to re-align this choice towards the singular vectors of the channel. Depending on the approach used for alignment as well as the SNR on the downlink and uplink, the beam alignment algorithm could take a substantial number of iterations to improve the effective channel gain.

In this context, we note that the (sequential/batch) least-squares approach from Section 2.3 achieves good performance in the high-SNR regime by optimally estimating the channel matrix over every iteration and re-aligning the alignment problem at

every step. However, this gain comes at the cost of complexity and feedback overhead of the algorithm. On the other hand, at low-SNR, averaging over the noise results in significant performance improvement with the summed power method from Section 2.4, which is a low-complexity/feedback overhead scheme.

These observations suggest that the two approaches can be married together, which is the focus of the *least-squares initialized summed power (LISP) method*. In this method, both nodes “prime” their beamformers using either the batch/sequential least-squares method for the first  $k_{\text{switch}}$  iterations, after which they switch to the summed power method. In particular, we have the following description in Algorithm B.4 (see Appendix B.4) for the proposed technique with the sequential least-squares initialization. The switching point  $k_{\text{switch}}$  can be chosen in multiple ways. Specific choices for  $k_{\text{switch}}$  include  $\min(M_r, M_t)$ ,  $\max(M_r, M_t)$  or via some optimality studies as in Sec. 2.6.

## 2.6 Numerical Studies

In this section, we present performance comparisons of the proposed schemes obtained via Monte Carlo experiments. We first present results on the convergence properties of the different techniques under varying conditions. We then present the impact of an increase in  $M_t$  on the performance of these schemes.

### 2.6.1 Convergence Studies

We study two variants of the proposed sequential least-squares technique from Section 2.3: “SLS (Optimal)” and “SLS (Suboptimal).” The first variant computes the batch least-squares estimator for the first  $M_r$  (or  $M_t$ ) iterations before switching to the sequential version after that. The second variant relies on the result from Theorem 2.3.3 to be computationally efficient and to avoid having to compute the batch estimator. It is initialized with  $\alpha = 1000$  and uses the sequential estimator starting at the first iteration. We also study the performance of the iterative solutions based on

the summed power method from Section 2.4 and the least-squares initialized summed power method with  $k_{\text{switch}} = \max(M_r, M_t)$  from Section 2.5. These approaches are denoted as “Summed Power” and “LISP” in the plots, respectively.

In terms of performance benchmarking, we consider the one-dimensional versions of the techniques proposed in [37] and [40]. The algorithm from [37] is called *Blind Iterative MIMO Algorithm (BIMA)* by the authors and is denoted as “BIMA” in the plots here. The algorithm from [40] is called *Best Singular Mode (BSM) estimation* by the authors and is denoted as “BSM” in the plots here. The value of the design parameter  $\mu$  for the BSM algorithm from [40] is set to  $1.5k$  where  $k$  is the time-index.

In Figs. 2.2 — 2.4, we compare the performance of these six schemes at different SNR values with  $M_r = 4$  and  $M_t = 32$  (corresponding to a downlink channel matrix  $\mathbf{H}$  of dimensions  $4 \times 32$ ). The channel matrix  $\mathbf{H}$  has i.i.d. entries. In particular, Figs. 2.2a and 2.2b show the results for uplink and downlink SNR values of  $-10$  dB, whereas, Figs. 2.3a and 2.3b, and Figs. 2.4a and 2.4b provide similar plots for an SNR of  $0$  dB and  $20$  dB, respectively. These SNR values are expected to be typical of low-, medium- and high-SNR regimes, respectively.

Practical mmWave channels are expected to be sparser [31,35] than i.i.d. channels. In this context, Fig. 2.5 illustrates the performance of the same set of six schemes in a sparse MIMO channel model with  $\lambda/2$  spaced uniform linear arrays (ULAs) at both ends corresponding to  $M_r = 4$  and  $M_t = 32$ . Both downlink and uplink SNRs are assumed to be  $-10$  dB and  $f_c = 28$  GHz is used. The channel is made of  $K = 3$  dominant clusters with one path per cluster (hence the channel matrix  $\mathbf{H}$  is rank-deficient). The angles of arrival and departure are assumed to be in the azimuth plane and uniformly distributed in a  $120^\circ$  angular spread at both ends. Rayleigh fading is assumed for the path gains. Such a model is commonly used in mmWave system analysis (see [35] and references therein for details).

We study two metrics capturing the performance of these six schemes: i) the instantaneous effective channel gain  $|\mathbf{z}^*[k]\mathbf{H}\mathbf{f}[k]|^2$  at time-index  $k$ , and ii) the angle between the true singular vector  $\mathbf{f}_{\text{opt}}$  and its estimate  $\mathbf{f}[k]$ , given as

$$\phi_k = \cos^{-1}(|\mathbf{f}_{\text{opt}}^* \mathbf{f}[k]|), \quad (2.39)$$

and measured in radians. Note that  $\phi_k$  equivalently captures the chordal distance between  $\mathbf{f}_{\text{opt}}$  and  $\mathbf{f}[k]$ . In order to average results over different channel realizations, we normalize the effective channel gain by  $\|\mathbf{H}\|_2^2 = \lambda_{\max}(\mathbf{H}^* \mathbf{H})$ . Fast convergence of the algorithm is then equivalent to fast convergence of the normalized instantaneous effective channel gain to 1.

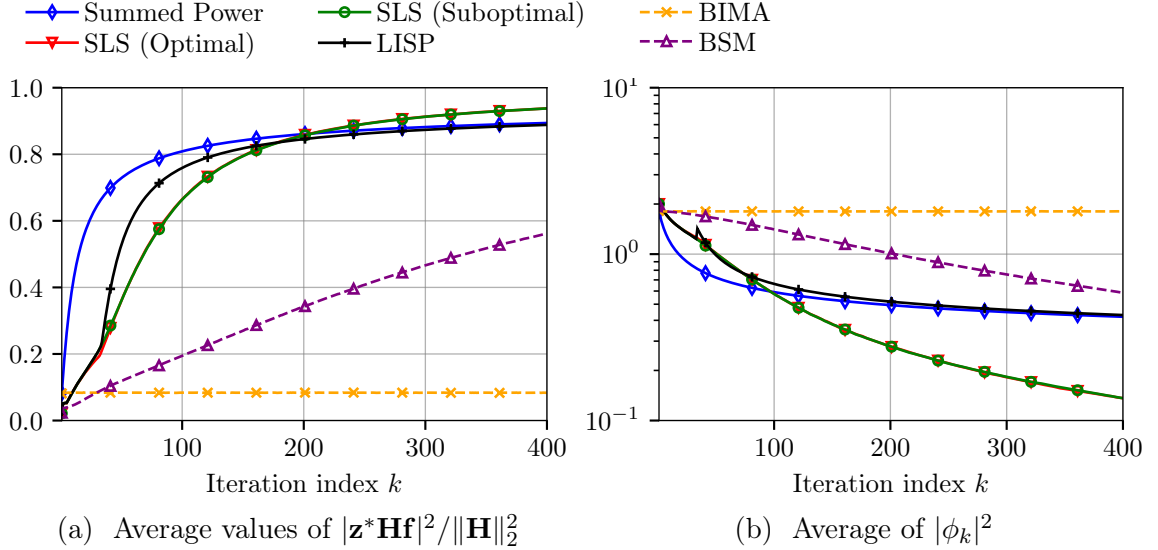


Fig. 2.2.: Simulation results for the i.i.d. channel model at  $\rho_e = \rho_o = -10$  dB with  $M_r = 4, M_t = 32$

From Figs. 2.2 — 2.4, we make the following remarks:

1. There is a minor performance gap (both in terms of gains and angles) between the optimal and suboptimal variants of the SLS estimator across all the three SNRs, even though there is a significant complexity reduction with the subop-

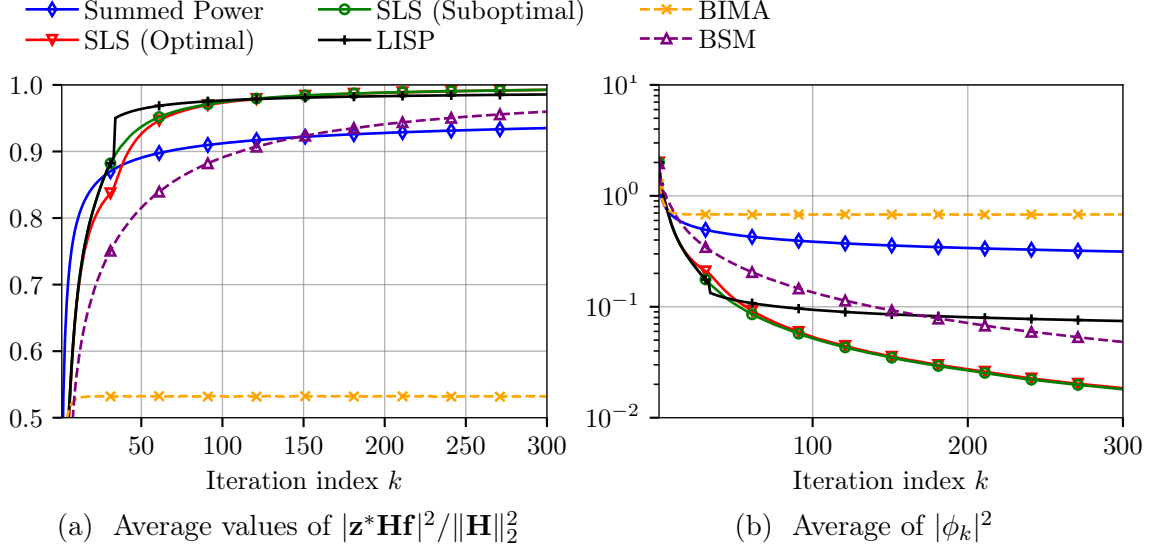


Fig. 2.3.: Simulation results for the i.i.d. channel model at  $\rho_e = \rho_o = 0$  dB with  $M_r = 4, M_t = 32$

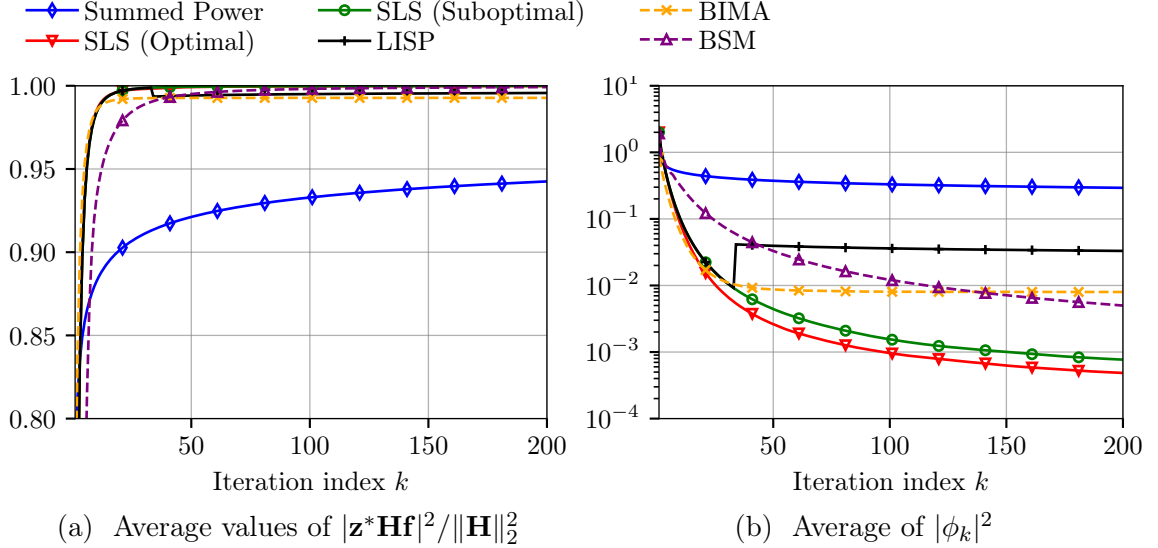


Fig. 2.4.: Simulation results for the i.i.d. channel model at  $\rho_e = \rho_o = 20$  dB with  $M_r = 4, M_t = 32$

timal variant. Thus, this study motivates the use of the suboptimal variant of the SLS estimator over the optimal variant.

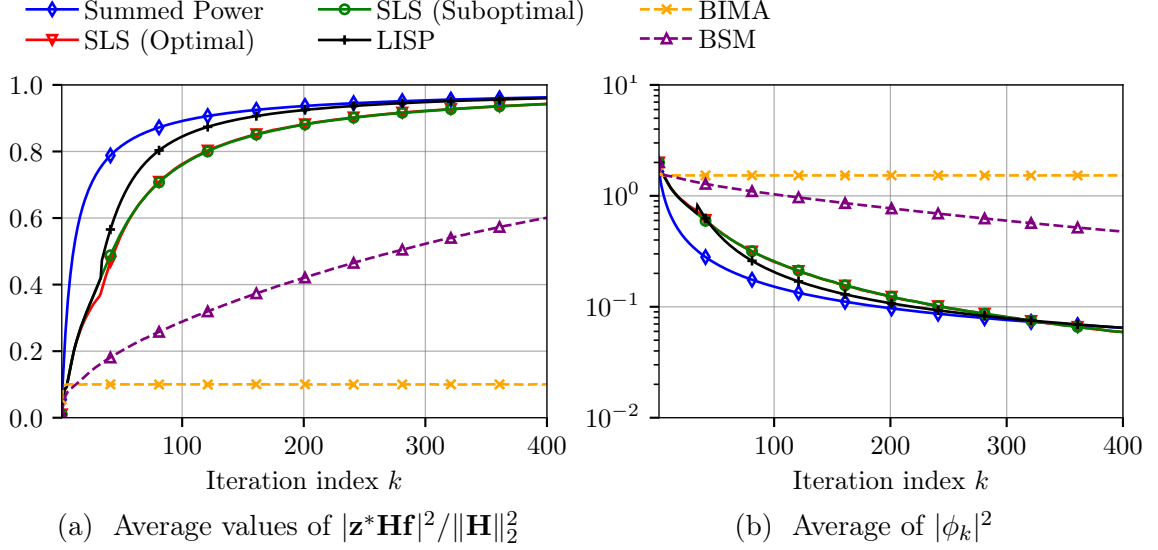


Fig. 2.5.: Simulation results for the sparse mmWave channel model at  $\rho_e = \rho_o = -10$  dB with  $M_r = 4$ ,  $M_t = 32$

2. In the low-SNR regime typical of mmWave settings, the summed power method significantly outperforms all the methods for small  $k$  values, whereas the additional channel estimation step of the SLS estimator contributes to its utility for large  $k$  values. While the method from [40] is better in performance than the one from [37], neither method produces a performance comparable to the schemes proposed in this work.
3. The performance of the schemes in [40] and [37] improve with SNR. In the high-SNR regime, both methods become comparable to the SLS estimator. However, the summed power method is significantly inferior in this regime as it cannot suppress the effect of noise from the beamformer estimates for large  $k$  values.
4. The switching between the SLS part and the summed power part means that the LISP method shows a switch in terms of performance at  $k = k_{\text{switch}} = \max(M_r, M_t) = 32$ . But more importantly, in the low-SNR regime, the LISP method approaches the performance of the summed power method for large  $k$  and in the high-SNR regime, it approaches the performance of the SLS estimator

(even for small  $k$  values) without the additional complexity overhead of these methods. Thus, this method may be a suitable low-complexity alternative to the SLS estimator in the medium- to high-SNR regime.

5. In the sparse mmWave setting with low SNR, the summed power method outperforms all the methods over all the values of  $k$  considered here. The LISP method quickly approaches the performance of the summed power method after  $k = k_{\text{switch}}$ .

Summarizing the above statements, we have the following conclusions: i) In the low-SNR regime, the summed power method is advantageous for small  $k$  and the SLS estimator is advantageous for large  $k$ . If computational complexity is an important issue for large  $k$ , the LISP method can be a useful alternative. ii) In the high-SNR regime, the LISP method or the method proposed in [37] are advantageous for all  $k$ . iii) These broad conclusions appear to be true for both i.i.d. as well as sparse mmWave channel models.

### 2.6.2 Impact of Antenna Dimensions and $k_{\text{switch}}$

Fig. 2.6 studies the impact of  $M_t$  (as  $M_t$  increases from 6 to 64) on the effective channel gain after  $k = 100$  iterations with the different beam alignment techniques. The low-SNR regime corresponding to  $\rho_e = \rho_o = -10$  dB and  $M_r = 4$  is considered in this study. Figs. 2.6a and 2.6b present the results for the i.i.d channel model and the sparse mmWave channel model introduced earlier.

This study reinforces the advantages of the summed power and least-squares initialized summed power methods relative to other methods. In particular, the performance of the summed power method remains approximately invariant in the i.i.d. case as  $M_t$  increases. On the other hand, the smaller rank of the channel matrix in the sparse case improves the fraction of power in the dominant eigen-mode, which is reflected in improving performance as  $M_t$  increases. But more importantly, the performance of all other schemes depreciate with  $M_t$  suggesting their sensitivity to



larger antenna dimensions. Nevertheless, the LISP method appears closest to the summed power method in performance at low-SNR and is also superior at high-SNRs. From these results, we conclude that the proposed beam alignment techniques and in particular, the LISP method can deliver substantial performance improvement as  $M_t$  increases with low complexity and feedback overheads making them viable candidates for practical large/massive MIMO systems.

Figs. 2.7a and 2.7b study the choice of  $k_{\text{switch}}$  to be used in the LISP method with  $M_r = 4$ ,  $M_t = 32$  and  $\rho_e = \rho_o = 0$  dB and  $\rho_e = \rho_o = -10$  dB, respectively. From Fig. 2.7a, we note that there exists an optimal  $k_{\text{switch}}$  that maximizes the effective channel gain for both the i.i.d. and sparse mmWave channel models. The optimal  $k_{\text{switch}}$  value is typically small in the case of sparse mmWave channels for both SNR settings. In fact, for  $\rho_e = \rho_o = -10$  dB, the optimal  $k_{\text{switch}}$  in the sparse setting is 1 implying that the summed power method starting at  $k = 1$  is better than a noisy initialization based on the SLS estimator. While the optimal  $k_{\text{switch}}$  can be high in the i.i.d. setting, constraining it to be a small number does not result in a significantly poorer performance relative to the optimal  $k_{\text{switch}}$  value. Thus, Figs. 2.7a and 2.7b suggest that, in the moderate- to high-SNR regime and depending on the level of richness/sparsity structure of the channel, a small  $k_{\text{switch}}$  may be a better choice than the use of summed power method ( $k_{\text{switch}} = 1$ ). Thus, an improved performance can be ensured with the LISP method at the cost of a small feedback and complexity overhead.

### 2.6.3 Comparison with a Pilot-Based Channel Estimation Scheme

We are now interested in comparing the performance of the proposed beam alignment schemes with a traditional pilot-based channel estimation scheme. In order to simplify the structure of the pilot-based scheme, we assume that the channel matrices are i.i.d. Rayleigh fading. In order to fairly compare the iterative schemes with the batch-oriented pilot-based scheme, we impose a constraint on the total energy used

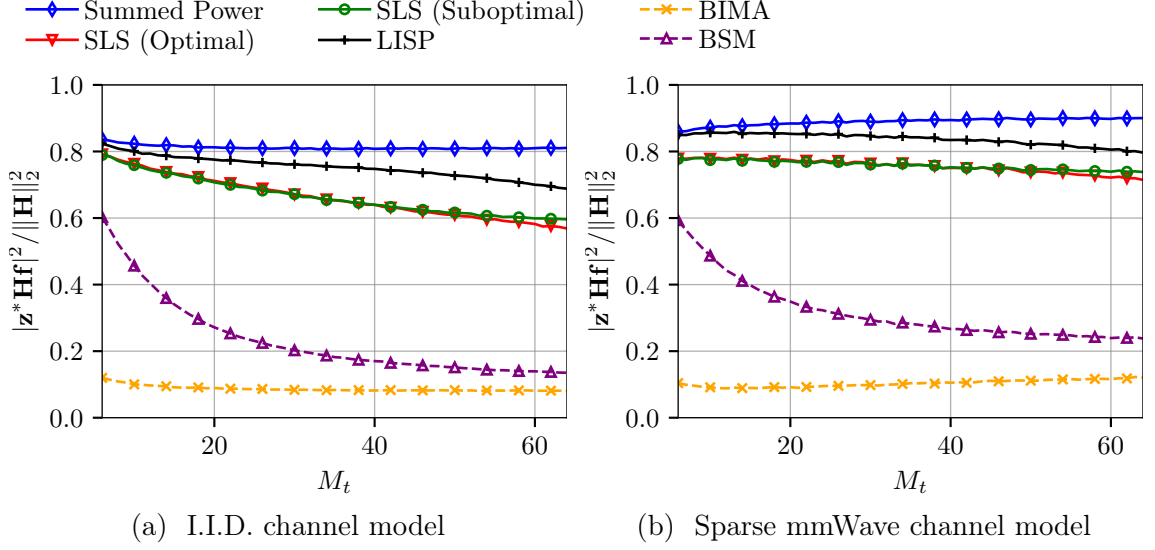


Fig. 2.6.: Average value of  $|\mathbf{z}^* \mathbf{H} \mathbf{f}|^2$  normalized by  $\|\mathbf{H}\|_2^2$  at  $k = 100$  for  $\rho_e = \rho_o = -10$  dB using different channel models with  $M_r = 4$  and  $M_t \in \{6, 8, 10, \dots, 64\}$

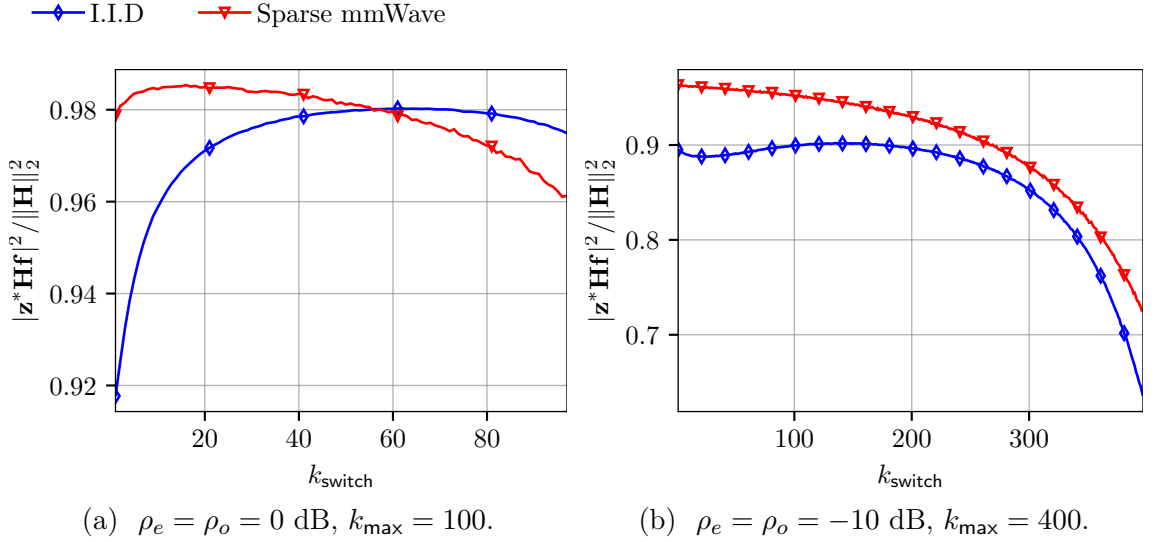


Fig. 2.7.: Optimization of normalized channel gain as a function of  $k_{\text{switch}}$  for different SNR regimes.  $M_r = 4$ ,  $M_t = 32$ .

during the beam alignment/channel estimation phase. Let  $k_{\text{max}}$  be the number of time slots allocated for this phase. With the iterative schemes considered in this work, the total energy used by nodes 1 and 2 reduces to  $\rho_o \cdot k_{\text{max}}$  and  $\rho_e \cdot k_{\text{max}}$ , respectively.

With the pilot-based scheme, it is well understood [24, 25, 53] that the quality of the channel estimate *only* depends on the energy in the training matrices (denoted as  $\mathbf{P}_o$  and  $\mathbf{P}_e$  for the downlink and uplink, respectively) as long as the number of pilot symbols exceeds the transmit antenna dimensions. Thus, we can assume that  $\mathbf{P}_o$  and  $\mathbf{P}_e$  are  $M_t \times M_t$  and  $M_r \times M_r$  scaled-unitary matrices meeting the energy constraint, respectively. With the energy scaling, we have the following system equations:

$$\mathbf{Y}_o = \sqrt{\rho_o \cdot k_{\max}/M_t} \mathbf{H} \mathbf{P}_o + \mathbf{N}_o \quad (2.40)$$

for the downlink, and

$$\mathbf{Y}_e = \sqrt{\rho_e \cdot k_{\max}/M_r} \mathbf{H}^T \mathbf{P}_e + \mathbf{N}_e \quad (2.41)$$

for the uplink.

Upon reception of  $\mathbf{Y}_o$  and  $\mathbf{Y}_e$ , each node computes a minimum mean-squared error (MMSE) channel estimate as follows:

$$\hat{\mathbf{H}}_o = \frac{\sqrt{\rho_o \cdot k_{\max}/M_t}}{1 + \rho_o \cdot k_{\max}/M_t} \cdot \mathbf{Y}_o \mathbf{P}_o^* \quad (2.42)$$

$$\hat{\mathbf{H}}_e = \frac{\sqrt{\rho_e \cdot k_{\max}/M_r}}{1 + \rho_e \cdot k_{\max}/M_r} \cdot \mathbf{Y}_e \mathbf{P}_e^*. \quad (2.43)$$

The beamformers are estimated using the SVD of the channel estimates. In our study, we use scaled discrete Fourier transform (DFT) matrices for  $\mathbf{P}_o$  and  $\mathbf{P}_e$  over the i.i.d. channel. With  $k_{\max} = 100$ , the normalized channel gain across different SNR values is plotted for the different schemes in Fig. 2.8. These results show that in addition to outperforming iterative schemes from prior works in the low-SNR regime, the proposed methods also compare favorably to the pilot-based channel estimation scheme. The pilot-based scheme requires a substantial pre-beamforming SNR (over 5-10 dB) for improved performance which may not be feasible in practical mmWave

systems. Further, it also requires a computational overhead in computing the SVD of the channel estimate.

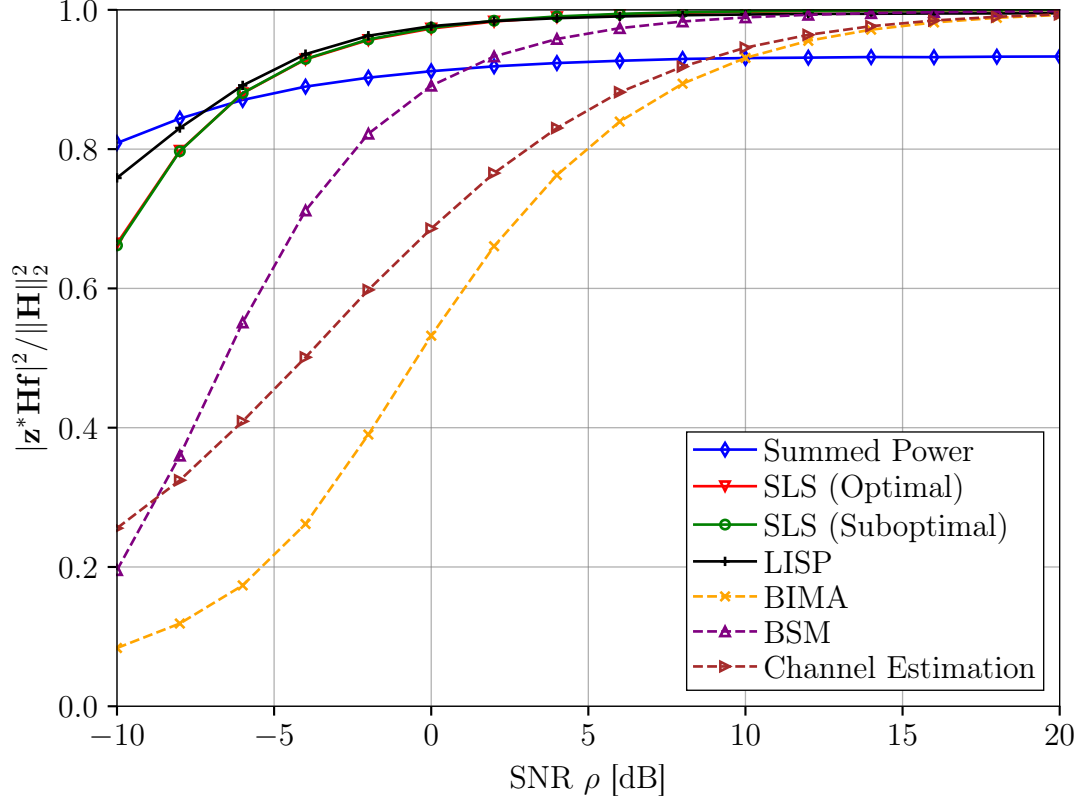


Fig. 2.8.: Normalized channel gain with  $k_{\max} = 100$  for varying values of  $\rho = \rho_o = \rho_e$  in the i.i.d Rayleigh fading channel case.

## 2.7 Concluding Remarks

This chapter studied the problem of estimating the dominant singular vectors of a MIMO channel matrix in a TDD system. Such a task is of importance in realizing the full analog beamforming gains in practical mmWave systems, typically impaired with low SNR. We presented multiple iterative approaches based on the power method to address this problem. These approaches included batch and sequential least-squares estimation, summed power method, and least-squares initialized summed power method. Numerical studies and analysis established that the proposed ap-

proaches enjoy several advantages over competing approaches from the literature. These advantages include improved convergence and/or performance (beamforming gain) at low- as well as high-SNR at a low-complexity and feedback overhead.

That said, this work has only scratched the surface of the noisy beam alignment problem. Further studies on developing an analytical/manifold optimization-based framework for the rate of convergence of the proposed algorithms as a function of the SNR, antenna dimensions, mmWave channel eigen-mode/sparsity structure, etc. are important. Such a step could also be of independent interest in problems in machine learning, principal component analysis, and linear algebra. Other problems of interest include understanding the impact of an imperfect (e.g., finite-rate, noisy, etc.) feedback link on the performance of the sequential least-squares estimation scheme, performance comparison with other directional learning approaches [35], impact of temporal variation in the channel and wideband aspects on the performance of the proposed schemes, extending the proposed analog beamforming schemes to a hybrid architectural set-up or multi-user settings [36], intuitive understanding of  $k_{\text{switch}}$  and further optimization of the beam alignment parameters given an asymmetrical antenna setting in the single-user case, etc.

### 3. EFFICIENT CHANNEL ESTIMATION FOR AERIAL WIRELESS COMMUNICATIONS

© 2019 IEEE. Reprinted, with permission, from: D. Ogbe, D. J. Love, M. Rebholz, and T. P. Bidigare, "Efficient channel estimation for aerial wireless communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 2774–2785, Dec. 2019.

#### 3.1 Introduction

The performance of modern wireless communication systems fundamentally depends on the quality of the channel estimates at either the receiver, the transmitter, or both. This statement holds of course for aerial wireless communication systems, which have recently experienced a resurgence of interest from both the academic and industrial community. Aerial communication systems, especially those based on unmanned aerial vehicle (UAV) platforms, are considered enabling technologies for future wireless networks like the consumer-oriented fifth-generation (5G) cellular networks and ad-hoc public safety networks [54]. In addition to 5G applications like UAV control and video streaming, aerial communication systems are actively being researched, developed, and deployed as part of the ongoing effort to increase the number of people connected to the Internet, lead by initiatives like Google's balloon-based "Project Loon" [55]. These efforts seek to increase the availability of low-cost Internet access by connecting a distributed set of airborne transmitters through wireless backhaul. Furthermore, apart from being considered for future communication networks, aerial wireless communication systems continue to play a significant role in other civilian applications like air traffic control as well as in a variety of military applications.

In addition to the time dispersion due to multipath propagation present in many wireless channels, the effects of vehicular motion inherent to aeronautical communication systems may induce frequency dispersion due to the Doppler effect [56]. The general class of channels exhibiting both time dispersive and frequency dispersive effects, usually referred to as *doubly dispersive* channels, is continuing to attract considerable interest from researchers and practitioners, in part due to its ubiquity in modern wireless communication systems, e.g., [57–63]. In the classical *channel estimation followed by data transmission* set-up, which we follow in this manuscript, it is thus desirable from a channel estimation perspective to obtain knowledge of the parameters governing both the time dispersion of the channel—assumed to be modeled as a discrete-time tapped delay line [64]—as well as the frequency dispersion, here assumed to be mainly caused by the Doppler effect due to vehicular motion.

The specific model we introduce and consider in this manuscript arises from our consideration of communication systems on high-velocity airborne vehicles. The general model of doubly dispersive channels where the frequency dispersion is due to Doppler assumes that the Doppler effect and thus the resulting frequency shift varies for each multipath component of the channel, resulting in a Doppler *spectrum*. In this manuscript however, we assume that for high-velocity airborne vehicles the Doppler spectrum is dominated by the *bulk* frequency shift due to the motion of the vehicle (see Section 3.2), rendering the individual shifts on each multipath component negligible in our analysis.

The model and corresponding estimation problem of a single bulk frequency shift coupled with multipath transmission mirrors the well-researched problem of estimating the frequency offset in orthogonal frequency division multiplexing (OFDM) systems. The tightly-spaced subcarriers in OFDM systems lose orthogonality in the presence of any kind of frequency offset, resulting in inter-channel interference. It has been shown that the bit error rate increases significantly if those offsets are left uncompensated [65].

The popularity of OFDM in modern communication systems has lead to the development of many different techniques to estimate and compensate for the frequency offset due to the Doppler effect and/or mismatches between the transmitter and receiver local oscillators. In general, the available frequency offset estimators can be classified into two categories. Blind techniques provide estimates of the frequency offset without the need for pilot symbols. The technique in [66] exploits the cyclostationarity inherent to OFDM waveforms to extract an estimate of the frequency offset. The authors of [67] derived a kurtosis-based estimator, which generalizes their single input, single output (SISO) results to multiple input, multiple output (MIMO) OFDM systems. The other class of OFDM frequency offset techniques includes semi-blind and non-blind estimators. The common property of all of these techniques, which relates to the results of this paper, is the reliance on some sort of redundancy in the structure of the waveform to compute the estimates. The work in [68] presented a frequency offset estimator based on the repetition of two OFDM symbols. The work in [69] uses a similar principle, keeping the redundant data within one OFDM symbol and using a second symbol for fine estimation as well as timing synchronization. The main idea behind these two techniques is that the cross-correlation of the repeated time-domain sequence of the receiver would perfectly reproduce the frequency offset in the absence of noise. The technique presented by Schmidl and Cox in [69] exploits the fact that the inverse discrete Fourier transform (IDFT) of a subcarrier allocation where every other subcarrier is set to zero produces a time-domain sequence of two repeated half-symbols. This concept was extended in [70] and generalized to produce time-domain sequences with more than two repetitions. The complexity of the Schmidl and Cox technique was further simplified in [71], reducing the number of OFDM pilot symbols needed from two to one. Furthermore, the work in [72–74] exploits the inherent redundancy of the usage of a cyclic prefix in OFDM to estimate the frequency offset. Finally, the frequency offset compensation problem was studied specifically for aeronautical channels in [75] and the references therein. The authors



assume a two-ray, dual Doppler shift model and compensate for each Doppler shift using the techniques from [72] after separating the signals of the two incoming paths.

Our contribution to the problem is a technique inspired by the various OFDM frequency offset estimators which jointly estimates the bulk Doppler shift and channel taps of an aeronautical channel. In contrast to the aforementioned OFDM-based techniques, our algorithm assumes single-carrier modulation, but could potentially be adapted to support multi-carrier modulations with a few modifications. More specifically, although our algorithm requires a certain time-domain structure combined with time-domain processing of the sounding signals, it places no restriction on the data transmission, allowing multi-carrier modulations to be used. The main idea behind our technique is to transmit a cyclically prefixed training sequence consisting of a repeated shorter training pulse. The receiver then computes an estimate of the Doppler shift using a combination of subspace estimation and matched filtering. After estimating and correcting for the effects of the Doppler shift, the receiver then computes the conditional maximum-likelihood (ML) estimate of the channel taps. We summarize the contributions of this paper as follows.

- We introduce a system model for high-velocity airborne wireless communication systems exhibiting time dispersion due to multipath and frequency dispersion due to a bulk Doppler shift caused by vehicular motion
- We derive the estimation-theoretic lower bound for estimating the channel taps as well as the Doppler shift of these channels
- We develop a pulse-repetition based channel estimator for these parameters
- We show that using constant amplitude, zero autocorrelation (CAZAC) sequences as pulses in our algorithm decreases the computational burden of our estimator
- We present numerical studies analyzing the performance of our estimator

The rest of this paper is structured as follows. Section 3.2 describes the system model considered throughout the paper. In Section 3.3 we derive the Cramer-Rao lower bound (CRLB) for the joint estimation of the Doppler shift and the channel taps using the observation model given in Section 3.2. We present and discuss the joint estimation algorithm in Section 3.4 and discuss the special case for CAZAC sequences in Section 3.4.4. Simulation results are presented and discussed in Section 3.5 and we provide some concluding remarks in Section 3.6.

**Notation:** We will use the following notation throughout this manuscript. Bold upper-case and lower-case letters (such as  $\mathbf{A}$  and  $\mathbf{a}$ ) denote matrices and column vectors, respectively. The operators  $(\cdot)^\top$ ,  $\overline{(\cdot)}$  and  $(\cdot)^*$  denote matrix transposition, element-wise complex conjugation and matrix Hermitian transposition, respectively.  $\|\cdot\|_2$  denotes the vector  $\ell_2$ -norm and  $\|\cdot\|_F$  denotes the Frobenius norm of a matrix.  $\mathcal{CN}(\mathbf{a}, \mathbf{A})$  denotes a complex Gaussian random vector with mean  $\mathbf{a}$  and covariance matrix  $\mathbf{A}$ .

### 3.2 System Model

Our system model consists of a single-antenna transmitter and receiver pair communicating over a single-input, single-output (SISO) doubly dispersive wireless communication channel. Under these assumptions, the general discrete-time complex baseband input-output model between the transmitter and the receiver can be written as

$$y[k] = \sqrt{\rho} \sum_{\ell=0}^{L-1} h_k[\ell] s[k-\ell] + n[k], \quad (3.1)$$

where we denote  $h_k[\ell]$ ,  $\ell \in \{0, \dots, L-1\}$  as the  $\ell$ -th complex channel filter tap at time index  $k$ . Furthermore,  $s[k]$ ,  $k \in \{0, \dots, L_s-1\}$  denotes the  $k$ -th sample of an arbitrary data or sounding sequence of length  $L_s$ ,  $n[k] \sim \mathcal{CN}(0, \sigma_k^2)$  is a sample of an additive Gaussian noise process, and  $\rho$  is a transmit signal-to-noise ratio (SNR) term.

Our channel model assumes an airborne platform where the Doppler shift is approximated as being of the same magnitude on each multipath component of  $h_k[\ell]$ . In general, the expression for a channel tap  $h_k[\ell]$  can be written as [56]

$$h_k[\ell] = \frac{1}{\sqrt{N_p}} \sum_{n=1}^{N_p} e^{j\theta_n} e^{j2\pi f_{d_n} k T_s} g_{\text{total}}(\ell T_s - \tau_n), \quad (3.2)$$

where  $N_p$  denotes the number of paths,  $\theta_n$  and  $\tau_n$  denote the phase shift and time delay of path  $n$ ,  $g_{\text{total}}$  denotes the convolution of the transmitter and receiver pulse shaping filters,  $T_s$  denotes the sampling period of the system, and  $f_{d_n}$  represents the Doppler shift on the  $n$ -th path from transmitter to receiver. As mentioned in Section 3.1, we make the assumption that the Doppler shift is equal across all paths, i.e.,  $f_{d_1} = f_{d_2} = \dots = f_{d_{N_p}} = f_d$ . This is a reasonable assumption when considering a ring of scatterers close to the transmitter traveling at the same velocity as the transmitter, for example the surface of an airplane. In this case, if we let  $v$  denote the relative velocity between the transmitter and receiver, we can define the Doppler shift as  $f_d = f_c v / c$ ,  $f_c$  being the carrier frequency of the communication system. Figure 3.1 illustrates two possible scenarios of our system model. In either case (air-to-air or air-to-ground), the Doppler shift creating a frequency offset at the receiver is due to non-zero relative velocities between the transmitter and the receiver. To further illustrate the near-equal Doppler shift assumption, consider the case of scattering off the airframe. In this case, the Doppler *spread*, i.e., the variance of the Doppler shifts on the different paths, will be most significantly affected by the roll, pitch, and yaw rates of the aircraft. Suppose for example scattering off the engines of a commercial two-engine airliner. The yaw rate for a "dutch roll" maneuver is 2.2 rad/second [76], resulting in a maximum Doppler frequency of  $\frac{f_c}{c} \cdot 12.4$  meters per second for an engine offset by 6 meters from the fuselage. However, realistic cruising speeds for commercial jet airliners are around 250 meters per second, resulting in a bulk Doppler shift about 20 times larger than the Doppler spread due to maneuvering. Similar arguments can be made for scattering off the ground in the air-to-ground case.

Here, the ratio of the Doppler spread due to the scatterers and the Bulk Doppler shift is bounded by the ratio of the diameter of the ring of scatterers and the distance from the transmitter to the receiver. Scatterer ring diameters in the single-kilometer range and standoff distances on the order of tens of kilometers (expected standoff ranges for tactical distributed beamforming applications, for example [77]) give bulk/scatterer ratios comparable to the aforementioned air-to-air setting. Considering, for example, a distance of 50 km and a scatterer ring diameter of 1 km, the bulk Doppler shift is approximately 50 times larger than the shifts due to scattering.

To simplify notation, we denote the sampled baseband frequency offset as  $\alpha = 2\pi T_s f_d$ . We can thus rewrite (3.2) as

$$h_k[\ell] = e^{j\alpha k} \sum_{n=1}^{N_p} \frac{1}{\sqrt{N_p}} e^{j\theta_n} g_{\text{total}}(\ell T_s - \tau_n), \quad (3.3)$$

where, if the number of paths  $N_p$  grows large, we can approximate the summation term as a circularly symmetric complex Gaussian random variable [64]. We can thus write the time-varying channel impulse response as

$$h_k[\ell] = e^{j\alpha k} h[\ell], \quad (3.4)$$

where  $h[\ell]$  can be approximated as  $\mathcal{CN}(0, 1)$ . Our doubly dispersive channel model thus consists of two separate components. We model the frequency dispersive part due to the Doppler shift with the complex exponential  $e^{j\alpha k}$ . We model the time dispersive part as a purely feed-forward tapped delay line with  $L$  taps. For the remainder of this manuscript, we will refer to these taps as finite impulse response (FIR) taps of the channel and denote them as  $h[\ell]$ ,  $\ell \in \{0, \dots, L-1\}$ . Substituting (3.4) in (3.1) yields the input-output relationship that is considered in this paper:

$$y[k] = e^{j\alpha k} \sqrt{\rho} \sum_{\ell=0}^{L-1} h[\ell] s[k-\ell] + n[k]. \quad (3.5)$$

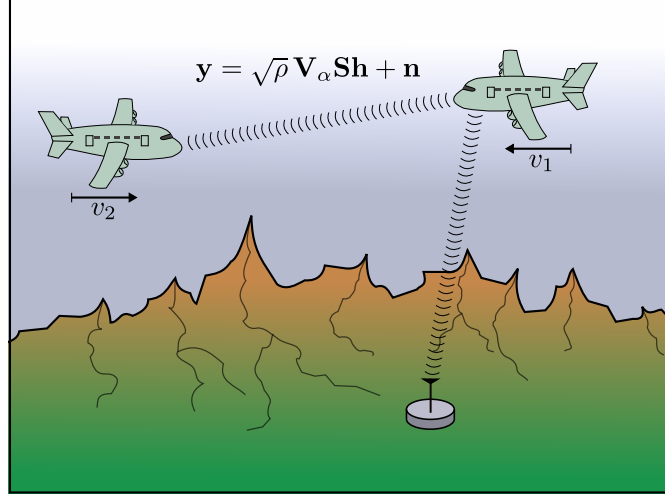


Fig. 3.1.: An aircraft wishes to estimate the channel taps  $\mathbf{h}$  of an air-to-ground or air-to-air channel with a single dominant Doppler component.

In our model, the receiver wishes to estimate a vector consisting of the  $L$  FIR taps of the channel impulse response  $h[\ell]$  using the received samples of a known training sequence  $s[k]$ . We assume that the training sequence  $s[k]$  contains  $L_s$  samples. The received signal  $y[k]$  thus consists of  $N = L_s + L - 1$  samples after the convolution with the channel impulse response. Using matrix-vector notation, and after defining the vectors  $\mathbf{s} = [s[0], \dots, s[L_s - 1]]^\top$ ,  $\mathbf{h} = [h[0], \dots, h[L - 1]]^\top$ , and  $\mathbf{y} = [y[0], \dots, y[N - 1]]^\top$ , as the training sequence, received samples, and channel impulse response vectors, respectively, the vector of received samples  $\mathbf{y}$  can be written as

$$\mathbf{y} = \sqrt{\rho} \mathbf{V}_\alpha \mathbf{S} \mathbf{h} + \mathbf{n}, \quad (3.6)$$

where  $\mathbf{V}_\alpha = \text{diag}([1 \ e^{j\alpha} \ \dots \ e^{j\alpha(N-1)}])$  represents the Doppler shift matrix,  $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{C})$  is a vector of complex Gaussian noise samples with covariance matrix  $\mathbf{C}$ , and  $\mathbf{S} \in \mathbb{C}^{N \times L}$  is the Toeplitz matrix obtained by linearly shifting the samples of the training sequence  $\mathbf{s}$  for each column.

The estimation problem that the receiver is seeking to solve is a joint Doppler/channel estimation problem, since the Doppler shift  $\alpha$  is unknown. More specifically, using our proposed technique, the receiver will use its estimate of  $\alpha$  to correct for the effects of the Doppler shift matrix  $\mathbf{V}_\alpha$  when obtaining an estimate of the vector of FIR taps  $\mathbf{h}$ . In the next section, we derive theoretical bounds on the variance of the estimators for the Doppler shift  $\alpha$  and the channel taps  $\mathbf{h}$  in this joint estimation framework.

### 3.3 Cramer-Rao Lower Bound

The derivation of the Cramer-Rao lower bounds for the joint estimator of the Doppler shift and the channel taps follows. The receiver wishes to estimate the parameter vector

$$\boldsymbol{\theta} = [\alpha \ \mathbf{h}^\top]^\top \quad (3.7)$$

from the observation given in (3.6). In order to derive the CRLB for any estimator  $\hat{\boldsymbol{\theta}}$ , we construct the Fisher information matrix  $\mathcal{I}(\boldsymbol{\theta})$ , where its elements are defined as [50]

$$\mathcal{I}_{k,\ell}(\boldsymbol{\theta}) = \left[ \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_k} \right]^* \mathbf{C}^{-1} \left[ \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_\ell} \right] \quad (3.8)$$

and  $\boldsymbol{\mu}(\boldsymbol{\theta})$  denotes the expectation of  $\boldsymbol{\theta}$ . A lower bound on the variance of the  $i$ -th element of  $\boldsymbol{\theta}$  is then given by

$$\text{var}(\hat{\theta}_i) \geq [\mathcal{I}^{-1}(\boldsymbol{\theta})]_{ii}. \quad (3.9)$$

**Lemma 3.3.1** *For  $\boldsymbol{\theta}$  as defined in (3.7) and the observation model given in (3.6), the Fisher information matrix is given by*

$$\mathcal{I}(\boldsymbol{\theta}) = \rho \begin{bmatrix} -\mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h} & -j \mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S} \\ j \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h} & \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S} \end{bmatrix} \quad (3.10)$$

where  $\mathbf{A} = \text{diag}([0 \ \cdots \ N-1])$ .

*Proof:* By inspection of (3.6), we have  $\boldsymbol{\mu}(\boldsymbol{\theta}) = \sqrt{\rho} \mathbf{V}_\alpha \mathbf{S} \mathbf{h}$ . It can then be shown that

$$\frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_1} = \sqrt{\rho} \frac{\partial \mathbf{V}_\alpha}{\partial \alpha} \mathbf{S} \mathbf{h} = j \sqrt{\rho} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h}, \quad (3.11)$$

where  $\mathbf{A} = \text{diag}([0 \ \cdots \ N-1])$ . It can further be shown that

$$\frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_2} = \sqrt{\rho} \mathbf{V}_\alpha \mathbf{S}. \quad (3.12)$$

After application of (3.8), we arrive at (3.10). ■

**Theorem 3.3.1** *The Cramer-Rao Lower Bound for any estimate of the Doppler frequency  $\alpha$ , denoted by  $\hat{\alpha}$ , is given by*

$$\begin{aligned} \text{var}(\hat{\alpha}) &\geq (1/\rho) \cdot \\ &(\mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h} \\ &+ \mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S} (\mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S})^{-1} \cdot \\ &\mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h})^{-1}. \end{aligned} \quad (3.13)$$

Furthermore, the Cramer-Rao Lower Bound for any estimate of the channel  $\mathbf{h}$ , denoted by  $\hat{\mathbf{h}}$ , is given by

$$\begin{aligned} \text{cov}(\hat{\mathbf{h}}) &\geq (1/\rho) \cdot \\ &(\mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S} \\ &+ \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h} \cdot \\ &(\mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{A} \mathbf{V}_\alpha \mathbf{S} \mathbf{h})^{-1} \cdot \\ &\mathbf{h}^* \mathbf{S}^* \mathbf{V}_\alpha^* \mathbf{A}^* \mathbf{C}^{-1} \mathbf{V}_\alpha \mathbf{S})^{-1}, \end{aligned} \quad (3.14)$$

where  $\mathbf{A} \geq \mathbf{B}$  for two compatible matrices  $\mathbf{A}$  and  $\mathbf{B}$  means  $\mathbf{A} - \mathbf{B}$  is a positive semi-definite matrix, or, equivalently, the product  $\mathbf{x}^* (\mathbf{A} - \mathbf{B}) \mathbf{x} \geq 0$  for all  $\mathbf{x} \in \mathbb{C}^n$ .

*Proof:* We invert (3.10) using the Schur complement formula [49] and apply (3.9) to arrive at (3.13) and (3.14).  $\blacksquare$

We note that (3.13) and (3.14) hold for arbitrary noise covariance matrices  $\mathbf{C}$ . In the case of white noise, the expressions for the CRLB simplify significantly, a fact shown in Corollary 3.3.1.

**Corollary 3.3.1** *If the additive noise in (3.6) is i.i.d. Gaussian distributed, i.e.,  $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ , the Cramer-Rao Lower Bounds are given by*

$$\text{var}(\hat{\alpha}) \geq \frac{1}{\rho} (\mathbf{h}^* \mathbf{S}^* \mathbf{A}^* \mathbf{A} \mathbf{S} \mathbf{h} + \mathbf{h}^* \mathbf{S}^* \mathbf{A}^* \mathbf{S} (\mathbf{S}^* \mathbf{S})^{-1} \mathbf{S}^* \mathbf{A} \mathbf{S} \mathbf{h})^{-1} \quad (3.15)$$

and

$$\text{cov}(\hat{\mathbf{h}}) \geq \frac{1}{\rho} (\mathbf{S}^* \mathbf{S} + \mathbf{S}^* \mathbf{A} \mathbf{S} \mathbf{h} (\mathbf{h}^* \mathbf{S}^* \mathbf{A}^* \mathbf{A} \mathbf{S} \mathbf{h})^{-1} \mathbf{h}^* \mathbf{S}^* \mathbf{A}^* \mathbf{S})^{-1}. \quad (3.16)$$

The expressions for the CRLB for both the Doppler and the channel estimates are crucial tools for analyzing the performance of our proposed algorithms. The simulation results in Section 3.5 will verify that our proposed techniques fall within a desirably small margin to the theoretical bounds on the estimator performance.



### 3.4 Estimation Procedure

This section provides a detailed description of our proposed block-based estimation algorithm. Subsection 3.4.1 covers the details of the block structure of the training signal, which is required in Subsection 3.4.2 to extract information about the Doppler shift at the receiver. The Doppler information is then used in Subsection 3.4.3 to compute an estimate of the channel coefficients. We close this section by providing an overview of the estimation steps in Algorithm 3.2.

#### 3.4.1 Block-based processing

Our proposed algorithm demands that the training sequence consists of  $M$  repetitions of an arbitrary training pulse  $\mathbf{x}$  and a cyclic prefix  $\mathbf{x}_{CP}$ . Specifically, in the notation of Section 3.2, the training sequence is structured as

$$\mathbf{s} = \left[ \underbrace{\mathbf{x}_{CP}^T}_{\text{cyclic prefix}} \mid \underbrace{\mathbf{x}^T \ \mathbf{x}^T \ \dots \ \mathbf{x}^T}_{M \text{ training pulses}} \right]^T, \quad (3.17)$$

where the cyclic prefix samples  $\mathbf{x}_{CP} = [x[K - L + 1] \ \dots \ x[K - 1]]^T \in \mathbb{C}^{L-1 \times 1}$  consist of a block of  $L - 1$  data symbols rotated cyclically [64]. In order to mitigate the effects of inter-symbol interference (ISI) induced by the time dispersion due to the  $L$  channel taps of  $\mathbf{h}$ , the receiver processes only the samples in the time interval  $k \in [L, KM + L - 1]$  and discards the rest. This results in a modified input/output model, which can be expressed in terms of the cyclic convolution of concatenation of  $M$  training pulses and the channel vector  $\mathbf{h}$  [64]. In addition to the cyclic prefix removal operation, the receiver also reorders the  $M$  segments of the received signal corresponding to the  $M$  transmitted training pulses into columns of a matrix of received samples denoted as  $\mathbf{Y} \in \mathbb{C}^{K \times M}$ . The process of cyclic prefix removal and sample reordering is visualized in Figure 3.2.

$$\begin{aligned}
\mathbf{y}^\top &= \left[ \begin{array}{c|c|c|c|c|c} \mathbf{y}_{CP,a}^\top & \mathbf{y}_0^\top & \mathbf{y}_1^\top & \cdots & \mathbf{y}_{M-1}^\top & \mathbf{y}_{CP,b}^\top \end{array} \right] \\
&\Downarrow \\
\mathbf{Y} &= \left[ \begin{array}{c|c|c|c} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{M-1} \end{array} \right]
\end{aligned}$$

Fig. 3.2.: The receiver drops the cyclic prefix components (red), extracts  $M$  receive pulses (green,  $K$  samples each), and re-orders them into the matrix of received pulses  $\mathbf{Y}$ .

The goal of this block processing at the receiver is to write an expression for the individual columns of  $\mathbf{Y}$  in terms of the cyclic convolution of one sequence pulse  $\mathbf{x}$  with the channel vector  $\mathbf{h}$ , which is given as

$$\mathbf{y}_\ell = \sqrt{\rho} \tilde{\mathbf{V}}_\ell \tilde{\mathbf{X}} \mathbf{h} + \mathbf{n}_\ell, \quad \ell \in 0, \dots, M-1. \quad (3.18)$$

Here, the additive noise vector  $\mathbf{n}_\ell$  is drawn from the complex Gaussian distribution with zero mean and covariance matrix  $\tilde{\mathbf{C}}$ , which is a truncated version of  $\mathbf{C}$  from (3.6). The diagonal matrix  $\tilde{\mathbf{V}}_\ell \in \mathbb{C}^{K \times K}$  accounts for the Doppler shift of the  $\ell$ -th block and is given by

$$\tilde{\mathbf{V}}_\ell = e^{j\ell K\alpha} \text{diag} \left( [e^{j(L-1)\alpha} \ e^{jL\alpha} \ \dots \ e^{j(K+L-2)\alpha}] \right). \quad (3.19)$$

Furthermore,  $\tilde{\mathbf{X}} \in \mathbb{C}^{K \times L}$  is the cyclic convolution matrix obtained from the training pulse  $\mathbf{x}$ , truncated to its first  $L$  columns. The  $K \times K$  cyclic convolution matrix

obtained from a vector  $\mathbf{x} \in \mathbb{C}^{K \times 1}$  is given by the matrix of cyclic shifts of  $\mathbf{x}$  and is defined as

$$\mathbf{X} = \begin{bmatrix} x[0] & x[K-1] & \cdots & x[1] \\ x[1] & x[0] & \cdots & x[2] \\ x[2] & x[1] & \cdots & \vdots \\ \vdots & x[2] & \cdots & x[K-2] \\ x[K-2] & \vdots & \vdots & x[K-1] \\ x[K-1] & x[K-2] & \cdots & x[0] \end{bmatrix}. \quad (3.20)$$

Upon further inspection of (3.19), we note that the effects of the Doppler shift  $\alpha$  can be separated into the Doppler shift internal to each block and the frequency offset between the blocks. Specifically, if we let  $\tilde{\mathbf{V}}_0$  represent the Doppler shift on each block ( $\tilde{\mathbf{V}}_\ell$  from (3.19) with  $\ell = 0$ ) and define the inter-block Doppler offset vector  $\mathbf{d}^*$  as

$$\mathbf{d}^* = [1 \ e^{jK\alpha} \ \cdots \ e^{j(M-1)K\alpha}], \quad (3.21)$$

we can write the expression for  $\mathbf{Y}$  as

$$\mathbf{Y} = \sqrt{\rho} \tilde{\mathbf{V}}_0 \tilde{\mathbf{X}} \mathbf{h} \mathbf{d}^* + \mathbf{N}, \quad (3.22)$$

where  $\tilde{\mathbf{X}}$  is defined as above and  $\mathbf{N} = [\mathbf{n}_0 \ \cdots \ \mathbf{n}_{M-1}]$  is the matrix of additive noise vectors. Writing the input/output model like (3.22) is a desirable step, since it lets us break the estimation algorithm into two distinct parts: The first step computes an estimate of the Doppler frequency, denoted  $\hat{\alpha}$ , which is then used to cancel out the effects of  $\mathbf{d}^*$  and  $\tilde{\mathbf{V}}_0$ , effectively turning the estimation of the channel coefficients into a straightforward linear Gaussian estimation problem.

### 3.4.2 Doppler Estimation

This estimation step directly exploits the block structure of the training sequence. We observe that at high SNRs, i.e., for large  $\rho$ , the received signal matrix  $\mathbf{Y}$  can be approximated as the rank-1 outer product of the vectors  $\sqrt{\rho}\tilde{\mathbf{V}}_0\tilde{\mathbf{X}}\mathbf{h}$  and  $\mathbf{d}^*$ . It is a well known fact that the singular value decomposition (SVD) can be used to construct low-rank approximations to matrices of any dimension [78]. In order to extract an estimate of  $\mathbf{d}^*$  from (3.22), we can therefore use the SVD of  $\mathbf{Y}$ , defined as

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_K \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_M \end{bmatrix}^*. \quad (3.23)$$

If  $\mathbf{Y}$  were truly rank one, we would have  $\sigma_2 = \sigma_3 \dots = \sigma_K = 0$  and could therefore write the SVD as the outer product  $\mathbf{Y} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^*$ . Comparing with the model given in (3.22), it becomes clear that in high SNRs,  $\mathbf{d}^*$  must be some scaled version of the vector  $\mathbf{v}_1^*$ , which is often referred to as the dominant right singular vector. This observation is the key to the Doppler estimation step, and we define the estimate of the Doppler offset vector, denoted as  $\hat{\mathbf{d}}$ , as

$$\hat{\mathbf{d}} = \sqrt{M} \mathbf{v}_1, \quad (3.24)$$

where the scale factor  $\sqrt{M}$  compensates for the fact that  $\mathbf{v}_1$  is usually obtained with unit norm. The simulation results from Section 3.5 will show that this estimator provides acceptable results even in low-SNR regimes, where  $\mathbf{Y}$  is unlikely to be of rank one due to noise.

Since the estimator for  $\mathbf{d}$  only requires the dominant right singular vector of  $\mathbf{Y}$ , computing the full SVD of  $\mathbf{Y}$  can be a waste of computational resources. Fortunately, there exist well-known iterative algorithms in numerical linear algebra to compute the dominant eigenvectors and singular vectors directly, with the simplest one being the power method [42]. The power method belongs to a larger class of general eigenvalue

or "power" iterations, which have been applied to problems in the space of multiple-input, multiple-output (MIMO) communication systems in recent works [44,79]. The basic idea behind the power method is that the repeated multiplication of a randomly selected vector  $\mathbf{x}$  with a matrix  $\mathbf{A}$  converges to a scalar multiple of the dominant eigenvector of  $\mathbf{A}$ . Normalization between the iterations of the power method produces the unit norm dominant eigenvector of  $\mathbf{A}$ . The algorithm is defined as

---

**Algorithm 3.1** Power Method

---

**Input:** A diagonalizable matrix  $\mathbf{A} \in \mathbb{C}^{n \times n}$   
 Let  $\mathbf{q}^{(0)}$  be a randomly chosen unit vector  $\in \mathbb{C}^n$   
**for**  $k = 1, 2, \dots$  **do**  
      $\mathbf{z}^{(k)} = \mathbf{A}\mathbf{q}^{(k-1)}$   
      $\mathbf{q}^{(k)} = \mathbf{z}^{(k)} / \|\mathbf{z}^{(k)}\|_2$   
**end for**

---

In Lemma 3.4.1, we briefly show the convergence of  $\mathbf{q}^{(k)}$  to the dominant eigenvector of  $\mathbf{A}$ .

**Lemma 3.4.1** *The power method given in Algorithm 3.1 converges to a scalar multiple of the dominant eigenvector of  $\mathbf{A}$ .*

*Proof:* Without loss of generality, we suppose that the eigenvalues of  $\mathbf{A}$  are ordered as

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_n|, \quad (3.25)$$

where we denote  $\lambda_1$  as the dominant eigenvalue and the corresponding eigenvector  $\mathbf{x}_1$  as the dominant eigenvector. Then, since  $\mathbf{A}$  is diagonalizable, we can write  $\mathbf{q}^{(0)}$  as linear combination of the eigenvectors of  $\mathbf{A}$ , i.e.,

$$\mathbf{q}^{(0)} = \sum_{i=1}^n q_i \mathbf{x}_i. \quad (3.26)$$

We can thus write  $\mathbf{A}^k \mathbf{q}^{(0)}$  as

$$\mathbf{A}^k \mathbf{q}^{(0)} = q_1 \lambda_1^k \left( \mathbf{x}_1 + \sum_{i=2}^n \frac{q_i}{q_1} \left( \frac{\lambda_i}{\lambda_1} \right)^k \mathbf{x}_i \right) \quad (3.27)$$

It is straightforward to show that  $\mathbf{q}^{(k)}$  is a scalar multiple of  $\mathbf{A}^k \mathbf{q}^{(0)}$ . Furthermore, since the ratio  $(\lambda_i/\lambda_1)^k$  approaches zero for all  $i \neq 1$ , all components of  $\mathbf{q}^{(k)}$  corresponding to eigenvectors other than  $\mathbf{x}_1$  vanish as  $k \rightarrow \infty$ . ■

Since the set of right singular vectors of  $\mathbf{Y}$  is equal to the set of eigenvectors of  $\mathbf{Y}^* \mathbf{Y}$ , we can use the power method to compute the dominant right singular vector  $\mathbf{v}_1$ , from which the estimator  $\hat{\mathbf{d}}$  can be obtained. Note that due to the block structure of  $\mathbf{Y}$  we can safely assume that (3.25) holds for moderate to high SNRs, since the signal component of  $\mathbf{Y}$  is rank 1 and all higher-rank components are due to additive noise. In practice, this means that a sufficiently accurate estimate of the dominant right singular vector can be obtained with a moderate number of iterations.

Recall that in order to mitigate the effects of the Doppler shift in the model from (3.22), the receiver must ideally cancel both the block-based Doppler offset due to  $\mathbf{d}^*$  as well as the intra-block Doppler shift due to  $\tilde{\mathbf{V}}_0$ . With perfect knowledge of  $\alpha$ , the receiver could simply construct the inverse of  $\tilde{\mathbf{V}}_0$  for this. However, since  $\alpha$  is not known, the receiver has to extract an estimate of it from the previously computed estimate of the Doppler offset vector  $\mathbf{d}^*$ . This estimate can be constructed using a simple correlation operation, and is given by

$$\hat{\alpha} = \arg \max_{\beta \in \mathbb{R}^+} \left| \sum_{\ell=1}^M e^{j(\ell-1)K\beta} \hat{d}_\ell \right|, \quad (3.28)$$

where  $\hat{d}_\ell$  is the  $\ell$ -th element of  $\hat{\mathbf{d}}$ . To minimize computational cost, the receiver could have a precomputed set of vectors to correlate against stored, resulting in a single matrix multiplication per estimation.

### 3.4.3 Channel Estimation

To arrive at the expression for the estimates of the  $L$  channel taps, we observe that with perfect knowledge of  $\alpha$ , the channel estimation problem would reduce to the well known linear Gaussian estimation problem, for which we could write the estimator as [50]

$$\hat{\mathbf{h}}_\alpha = \frac{1}{M\sqrt{\rho}} \left( \tilde{\mathbf{X}}^* \tilde{\mathbf{V}}_0^* \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{V}}_0 \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^* \tilde{\mathbf{V}}_0^* \tilde{\mathbf{C}}^{-1} \mathbf{Y} \mathbf{d}. \quad (3.29)$$

Since in our system setup the Doppler shift  $\alpha$  is unknown, the receiver has to resort to using the estimates of  $\tilde{\mathbf{V}}_0$  and  $\mathbf{d}$ , which are obtained as described in the previous subsection. More specifically, the receiver computes the estimate as

$$\hat{\mathbf{h}} = \frac{1}{M\sqrt{\rho}} \left( \tilde{\mathbf{X}}^* \hat{\mathbf{V}}_0^* \tilde{\mathbf{C}}^{-1} \hat{\mathbf{V}}_0 \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^* \hat{\mathbf{V}}_0^* \tilde{\mathbf{C}}^{-1} \mathbf{Y} \hat{\mathbf{d}}, \quad (3.30)$$

where  $\hat{\mathbf{d}}$  was derived in the previous section and  $\hat{\mathbf{V}}_0$  is constructed using the Doppler shift estimate  $\hat{\alpha}$  as

$$\hat{\mathbf{V}}_0 = \text{diag} \left( [e^{j(L-1)\hat{\alpha}} \ e^{jL\hat{\alpha}} \ \dots \ e^{j(K+L-2)\hat{\alpha}}] \right). \quad (3.31)$$

To conclude our discussion on the specifics of our proposed estimation algorithm, the steps outlined in 3.2 provides a short summary of the necessary steps at the receiver with our proposed technique.

---

**Algorithm 3.2** Doubly dispersive channel estimation (Summary)

---

- |   |                                   |
|---|-----------------------------------|
| <b>Input:</b> Received estimation sequence $\mathbf{y}$ | $\triangleright$ (3.6)            |
| 1. Drop cyclic prefix and reshape                       | $\triangleright$ (3.22), Fig. 3.2 |
| 2. Estimate Doppler shift vector $\mathbf{d}$           | $\triangleright$ (3.24), Alg. 3.1 |
| 3. Estimate Doppler frequency using correlation         | $\triangleright$ (3.28)           |
| 4. Compute channel estimate $\hat{\mathbf{h}}$          | $\triangleright$ (3.30)           |
-

### 3.4.4 Simplified Channel Estimation using CAZAC Sequences

Until now, the discussion of our channel estimation algorithm has remained independent of the choice of base sequence for our training pulses, denoted earlier as  $\mathbf{x}$ . While there are many potential choices of pseudo-random (PN) training sequences, we have exclusively considered the class of Zadoff-Chu sequences [80] in this work. Zadoff-Chu sequences belong to the class of constant-amplitude, zero autocorrelation (CAZAC) waveforms and have most recently found use in various applications in the LTE physical layer [81]. The  $k$ -th symbol of a Zadoff-Chu sequence of length  $K$  is given by

$$x_k = \begin{cases} e^{-j\pi uk(k+2q)/K} & \text{if } K \text{ is even} \\ e^{-j\pi uk(k+1+2q)/K} & \text{if } K \text{ is odd,} \end{cases} \quad (3.32)$$

where the parameter  $q$  is any positive integer or zero and the parameter  $u$  is some positive integer relatively prime to  $K$ . For the remainder of this text, we used the values  $q = 0$  and  $u = 1$ . Zadoff-Chu sequences have various beneficial properties, the most interesting for this application being the fact that cyclic shifts of the same sequence are orthogonal. Recall that our estimator, given in (3.30), utilizes the circular convolution matrix obtained from a training pulse, denoted as  $\tilde{\mathbf{X}}$ . If the training pulse  $\mathbf{x}$  is a Zadoff-Chu sequence, it can be shown that the circular convolution matrix  $\tilde{\mathbf{X}}$  satisfies

$$\tilde{\mathbf{X}}^* \tilde{\mathbf{X}} = \tilde{\mathbf{X}} \tilde{\mathbf{X}}^* = K \mathbf{I}, \quad (3.33)$$

which greatly simplifies the computation of (3.30), since it is easily shown that under white noise this expression simplifies to



$$\hat{\mathbf{h}} = \frac{1}{M\sqrt{\rho}} \tilde{\mathbf{X}}^* \hat{\mathbf{V}}_0^* \mathbf{Y} \hat{\mathbf{d}}. \quad (3.34)$$

The switch from (3.30) to (3.34) obviates the need to compute expensive high-dimensional matrix inverses on-the-fly in the white noise regime, resulting in a substantially decreased computational load. Given these favorable properties, the remainder of this paper will assume that every training pulse is a Zadoff-Chu sequence of appropriate length. An added benefit to using Zadoff-Chu sequences in practical systems are their constant-cross-correlation properties [80], which are leveraged in the LTE physical random access channel (PRACH) to resolve collisions using initial access. In multi-user systems employing our proposed estimation technique, different users could be assigned different root parameters  $u$  and  $q$  to minimize conflicts during channel estimation.

### 3.5 Numerical Studies

In this section, we evaluate the performance of our proposed algorithm using Monte-Carlo methods and synthetic data. Our quantities of interest for all simulations are the sample mean squared error of the channel estimates  $\hat{\mathbf{h}}$  and the sample error variance of the Doppler frequency estimate  $\hat{\alpha}$ . In all results of this section, the  $L$  channel taps were drawn independently from a complex Gaussian distribution, i.e.,  $h[\ell] \sim \mathcal{N}(0, \mathbf{I})$ ,  $\ell \in 0, \dots, L-1$ . In all simulations, the length of the cyclic prefix was held constant at  $L-1$  symbols, the minimum required length to write the input-output model using circular convolution. Furthermore, for each Monte-Carlo iteration a channel realization, noise realization, and a initial seed vector for the power method iteration was generated independently.

### 3.5.1 Comparison to theoretical optimum

Figure 3.3 studies the performance of our proposed estimator in relation to the theoretical optimum derived in Section 3.3. Specifically, in Fig. 3.3a) we examine the difference between the sample mean squared error of the channel estimate and the CRLB given in (3.14) for signal-to-noise ratios ranging from -10 dB to 20 dB. Correspondingly, Fig. 3.3b) examines the difference between the sample variance of the estimation error of  $\hat{\alpha}$  and the CRLB given in (3.13) for the same range of SNRs. We compare these performance metrics in both figures for three different cases.

1. **White Noise.** The curves labeled "White Noise" present results for uncorrelated white complex Gaussian noise, i.e.,  $\mathbf{C} = \mathbf{I}$ . As noted earlier, the training pulses consist of Zadoff-Chu sequences, with the sequence length  $K$  fixed to 127. The equation used for the channel estimation step is thus (3.34).
2. **Correlated Noise.** The curves labeled "Correlated Noise" examine the performance of our estimator for correlated noise environments. To simulate such an environment, we constructed a synthetic noise correlation matrix

$$\mathbf{C} = \begin{bmatrix} 1 & \beta & \beta^2 & \dots & \beta^N \\ \beta & 1 & \beta & \dots & \beta^{N-1} \\ \beta^2 & \beta & 1 & \dots & \beta^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^N & \beta^{N-1} & \beta^{N-2} & \dots & 1 \end{bmatrix}, \quad (3.35)$$

where the specific correlation parameter  $\beta$  used for producing the curves in Fig. 3.3 was  $\beta = 0.8$ . Due to the structure of  $\mathbf{C}$ , the channel estimation step uses (3.30), including the costly matrix inverses.

3. **Grid Search.** We provide an alternative channel and Doppler estimation algorithm as a point of reference for our proposed algorithms. This estimator,

denoted as "Grid Search" in Figure 3.3, computes an approximate maximum-likelihood estimate of the channel taps and the frequency offset  $\alpha$  by performing the minimization

$$\hat{\alpha} = \underset{\beta \in G}{\operatorname{argmin}} \quad \|\mathbf{y} - \mathbf{y}_\beta\|^2, \quad (3.36)$$

$$\hat{\mathbf{h}} = \hat{\mathbf{h}}_{\hat{\alpha}}, \quad (3.37)$$

where

$$\mathbf{y}_\beta = \sqrt{\rho} \mathbf{V}_\beta \mathbf{S} \mathbf{h}_\beta \quad (3.38)$$

$$\hat{\mathbf{h}}_\beta = \frac{1}{\sqrt{\rho}} (\mathbf{S}^* \mathbf{V}_\beta^* \mathbf{C}^{-1} \mathbf{V}_\beta \mathbf{S})^{-1} \mathbf{S}^* \mathbf{V}_\beta^* \mathbf{C}^{-1} \mathbf{y} \quad (3.39)$$

over a pre-defined set of grid points  $G$  for the observation vector  $\mathbf{y}$  (3.6). We note that in (3.36),  $\mathbf{V}_\beta = \operatorname{diag}([1 \ e^{j\beta} \ \dots \ e^{j\beta(N-1)}])$ . In order to obtain a fair comparison, the set of grid points is chosen to be equivalent to the set of points over which our proposed estimators perform the Doppler correlation step (3.28), rendering this search prohibitively expensive from a computational perspective. However, this estimator helps to provide a good baseline for the performance of our proposed algorithms due to its near-optimal performance.

We show the error curves in Figure 3.3 for three different values of the Doppler frequency, each presented using a different line style. More specifically, the mapping is described in Table 3.1. Although our studies suggest that the impact of the specific Doppler frequency on the estimation error is minimal, these three values were chosen to provide examples of realistic conditions and reasonable values for reference. For example, the aforementioned cruising speed of 250 m/s combined with a transmission at the center frequency  $f_c = 120$  MHz (reserved for aeronautical mobile radio according to [82]) results in a bulk Doppler shift of approximately 100 Hz. The values of 50 Hz and 0 Hz were studied to provide additional points for reference. In all of these studies, one training sequence pulse consisted of a Zadoff-Chu sequence of

$f_d$	Style
0 Hz	dotted ....
50 Hz	dashed ...
100 Hz	solid —

Table 3.1.: Doppler shift to line style mapping for Figs. 3.3a)– 3.5

length  $K = 127$ , while the entire training sequence consisted of  $M = 5$  pulses with a cyclic prefix. At each iteration,  $L = 16$  channel taps were generated.

The results in Figure 3.3 indicate that our proposed algorithm produces desirable results largely independent of the magnitude of the Doppler shift present, especially for medium to high SNRs. Note that due to the loss of information from performing cyclic prefix removal at the receiver, our proposed estimator never fully attains the CRLB, even for high SNRs. Our algorithm performs especially well for the white noise scenario, quickly approaching a sub-1 dB difference from the theoretical optimum and a difference to the computationally expensive minimum-distance grid search estimator of less than 0.5 dB. The constant gap for the colored noise case is explained by the fact that the estimator from (3.30) disregards any correlation across blocks that could be present in the noise samples.

The contrast between Figure 3.3a) and Figure 3.4 illustrates the importance of the intra-block Doppler correction for this algorithm. To generate the results in Figure 3.4, we disabled the intra-block Doppler correction from (3.28), resulting in  $\hat{\mathbf{V}}_0 = \mathbf{I}$  at all iterations. The severe performance penalty of only correcting for the per-block Doppler shifts is evident in the increasing loss in estimator performance as a function of increasing SNR. We note that as expected, this performance loss does not occur for the 0 Hz case.

### 3.5.2 Performance under Model Mismatch

Figure 3.5 studies the loss in performance of our proposed estimator when the bulk Doppler assumption introduced in Section 3.2 does not apply and the Doppler

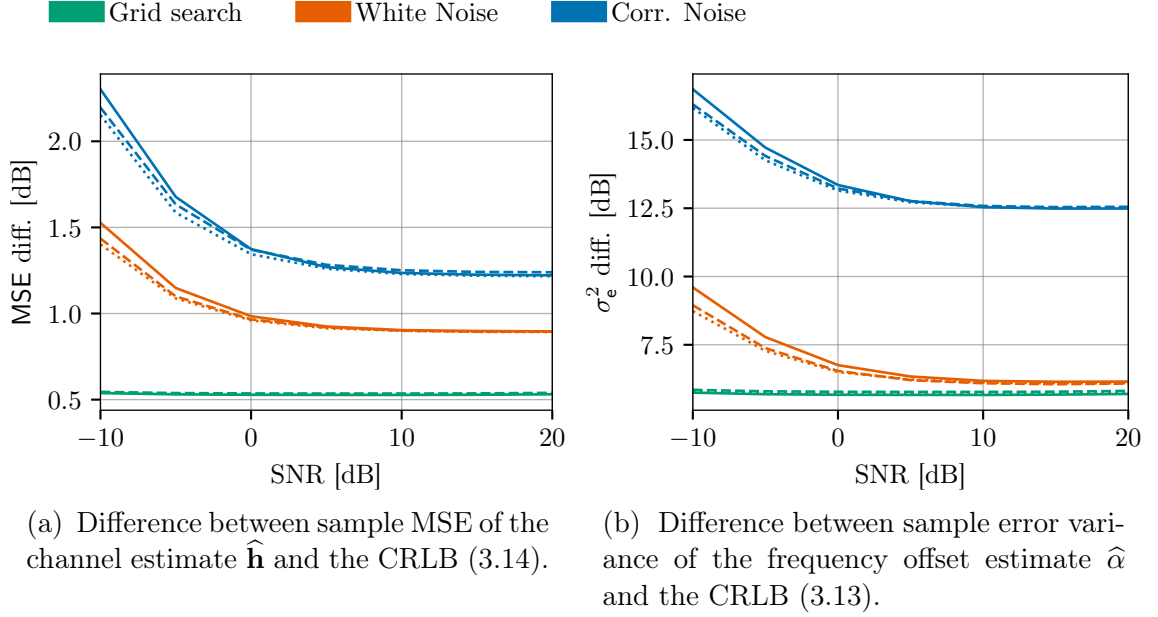


Fig. 3.3.: Difference between sample statistics and CRLB vs. SNR of our proposed estimator for white noise and colored noise environments. Grid search based estimator included for reference.  $K = 127$  samples,  $M = 5$  pulses,  $L = 16$  channel taps.

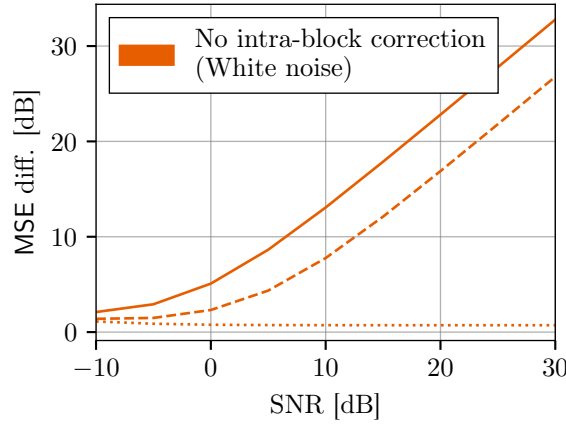


Fig. 3.4.: Difference between sample MSE of the channel estimate  $\hat{\mathbf{h}}$  and the CRLB (3.14) vs. SNR of our proposed estimator for white noise without intra-block Doppler correction from (3.28).  $K = 127$  samples,  $M = 5$  pulses,  $L = 16$  channel taps.

shift varies across the multipath components of the channel. In this case, the channel model from (3.4) becomes

$$h_k[\ell] = e^{j\alpha[\ell]k} h[\ell], \quad (3.40)$$

where  $\alpha[\ell] = 2\pi T_s f_d[\ell]$  and  $f_d[\ell]$  is sampled from a probability distribution for each channel tap. We furthermore restrict ourselves to the white noise case for this study and re-use the parameters  $K = 127$ ,  $L = 16$ , and  $M = 5$  from the previous subsection.

Figure 3.5 plots these trials for normally distributed  $f_d[\ell] \sim \mathcal{N}(\bar{f}_d, \sigma_f^2)$  for the three different mean values  $\bar{f}_d$  given in Table 3.1 as a function of the standard deviation  $\sigma_f$ . Each curve is obtained by using our proposed algorithms (designed for the bulk Doppler-only model) on the channel model from (3.40). The plot shows the difference in mean squared error between this mismatched scenario (Estimator assumes equal Doppler, but channel model generates random Doppler per multipath component) and the matched scenario from in Fig. 3.3a) (Estimator assumes equal Doppler and channel model generates equal Doppler per multipath component) for a range of SNR regimes. As expected, the performance of our estimator degrades with increasing standard deviation of the Doppler frequency. The simulation furthermore indicates that the performance in high-SNR regimes is limited by the model mismatch, whereas it is limited by the noise power in the low to medium SNR regime. This implies that our estimators remain attractive options in the low-to medium SNR regime, especially in the 0-10 dB range. We note that the choice of normally distributed Doppler shifts is applicable especially in the air-to-ground scenario with automobiles as scatterers due to the widely accepted assumption of normally distributed velocities [83–86].

### 3.5.3 Pulse Length and Repetition Count

The plot in Figure 3.6a) shows the channel estimation MSE as the length of a training pulse  $K$  increases. The results were obtained assuming white noise, i.e.,  $\beta = 0$ ,  $M = 5$  training pulse repetitions,  $L = 16$  channel taps, and transmit powers ranging from -10 dB to 30 dB. The training sequence pulses were again chosen to be Zadoff-Chu sequences with the length of one pulse varying from  $K = 29$  to  $K = 331$ . For these simulations, the Doppler shift was fixed to  $f_d = 50$  Hz. As we can see, performance gains can be achieved when increasing the pulse length, but

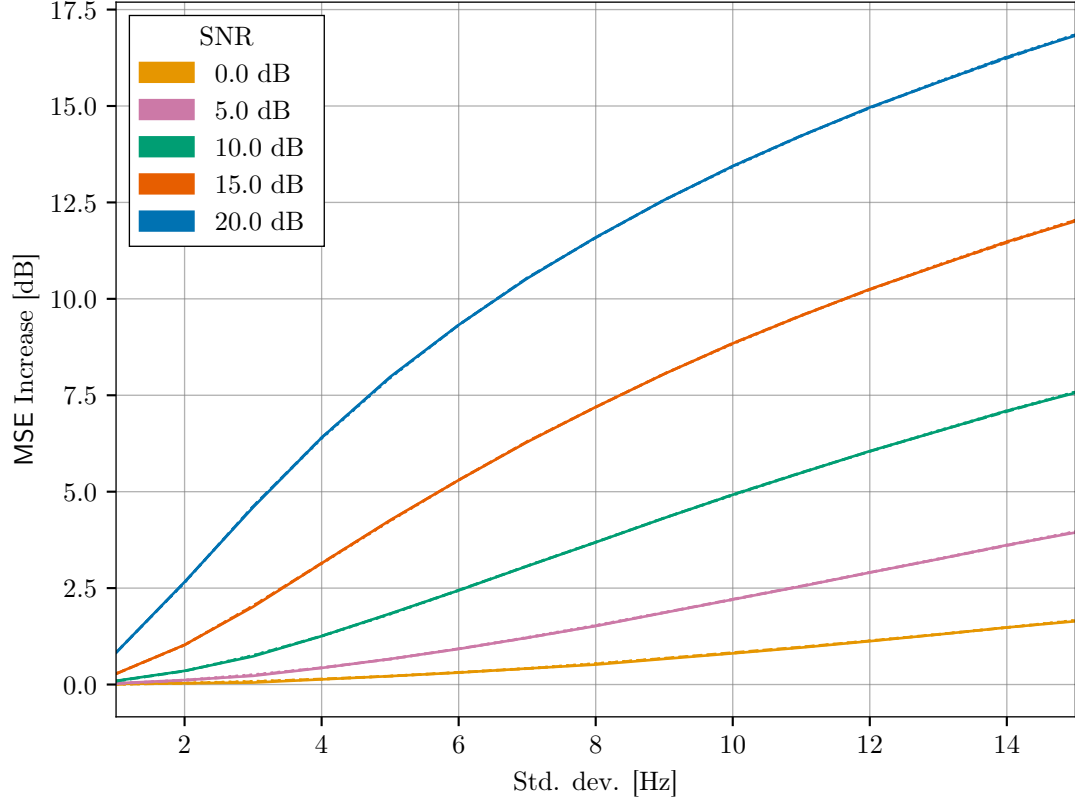


Fig. 3.5.: Increase in mean squared error relative to the bulk-only Doppler shift model when using our proposed estimator on the mismatched random Doppler shift model from (3.40). “MSE Increase” denotes the difference in mean square error performance between the matched and mismatched scenario.  $K = 127$  samples,  $M = 5$  pulses,  $L = 16$  channel taps. Pictured for various SNR regimes.

the magnitude of the performance gains decreases with the length. Figure 3.6b) gives a similar comparison, however, whereas the number of pulses  $M$  was fixed in Figure 3.6a), we now fix the length of one training pulse to  $K = 127$  symbols and vary the number of pulses  $M$  between  $M = 2$  and  $M = 20$ . In this case, the MSE behaves similar as it does with increasing pulse length.

### 3.6 Concluding Remarks

This chapter studied the problem of jointly estimating the bulk Doppler shift and the channel taps of a doubly dispersive aerial channel. Work in this area is of impor-

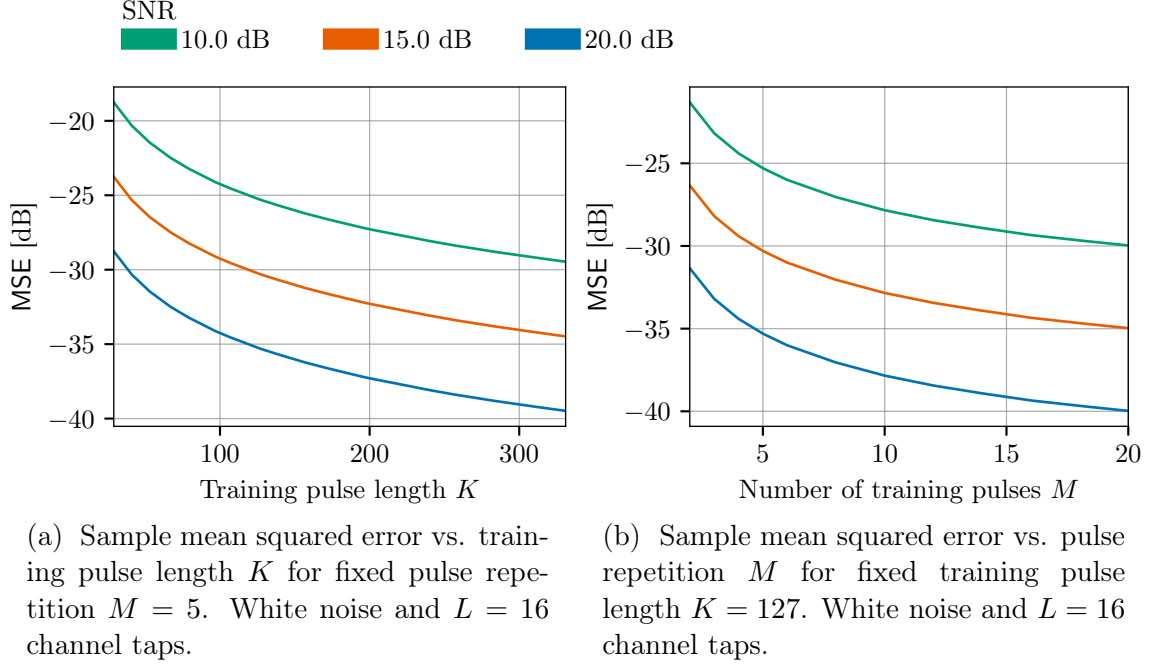


Fig. 3.6.: Impact of sounding signal length factors on MSE for different SNRs.

tance and significance because channel estimation is an essential part of any modern wireless communication system, including aeronautical systems. This chapter and other current and future work focusing on aeronautical wireless communication systems is in line with the recent surge in popularity of airborne platforms in consumer electronics, public safety networks, and defense applications. Our contribution to the field concerns a special class of channels in which the Doppler spread is dominated by the bulk Doppler shift due to vehicular motion. For this class of channels, we presented a simple channel model incorporating time and frequency dispersion before deriving the theoretical bounds of the resulting channel estimation problem. In addition to the theoretical analysis, we developed a novel channel estimation algorithm by combining traditional pulse-repetition techniques with simple and computationally efficient processing techniques. Our numerical studies indicate that our proposed techniques perform comparatively well to computationally expensive brute-force search methods.



The main contribution of this chapter lies in our combination of prior work in estimation theory and signal processing into an efficient algorithm for our model. However, aerial channels like the one considered in this chapter are still sources of many open problems. More specifically, future work in this area includes the extension of our model to multi-antenna systems and multi-user scenarios.

## 4. ON THE OPTIMAL DELAY GROWTH RATE OF MULTI-HOP LINE NETWORKS

© 2020 IEEE. Reprinted, with permission, from: D. Ogbe, C.-C. Wang, and D. J. Love, “On the optimal delay growth rate of multi-hop line networks — asymptotically optimal designs for low-latency relays,” *Submission to: IEEE Transactions on Information Theory*, May 2020.

### 4.1 Introduction

#### 4.1.1 Reliability Function

Along with capacity, the reliability of a communication channel, which describes the probability of erroneous message reception for a fixed-length message transmission, has played a significant role in the development of information theory. This emphasis on the *error rate vs. codeword length* tradeoff is particularly relevant for modern ultra-reliable ultra-low-latency (URLLC) communications [87] and massive machine-type communications (mMTC) [88] since long codeword lengths translate directly to a long *transmission delay* between the start of the transmission at the sender and the actual extraction of the messages at the receiver, even if we assume that the underlying encoding/decoding algorithms can be carried out and finished instantaneously (with infinite hardware clock rate).

The reliability function of point-to-point channels is a well-studied subject. In 1959, Shannon discovered upper and lower bounds on the error exponent of the AWGN channel [89], which spearheaded numerous follow-up works in the next decades, including but not limited to [90–98]. More advanced studies of the reliability function other than error exponent analysis have received significant attention in recent years

under the new framework of *finite-length analysis*, see e.g., [99–104]. These latter works focus on the more practical *communication rate vs. codeword length* tradeoff under a fixed error rate requirement as opposed to the more traditional *error rate vs. codeword length* tradeoff under a fixed communication rate requirement. Essentially, both the reliability function and finite-length analysis study the joint relationship between the error rate, the throughput, and the delay, and this work falls under the same umbrella as these important prior results.

#### 4.1.2 Multi-hop Line Networks

One signature trait of modern wireless communication systems is the overall densification of the network due to novel infrastructure devices such as femto- or pico-cells. With the continuing densification in 5G and beyond-5G networks, these "small cells" are increasingly not directly connected to fiber-optic networks due to cost or other site-specific constraints. Instead, they rely on wireless connections to the bigger cell(s), an architecture commonly referred to as "integrated access and backhaul" (IAB) [105, 106]. This has kindled renewed interest in classical relay channels.

The history of the relay channel dates back to the general three-terminal channels by van der Meulen [107] in the 1970s, and a comprehensive review can be found in [108]. Fig. 4.1a) describes the most general relay channel model, for which the relay channel is modeled as the joint conditional probability  $P(Y_2, Y_1 | X_1, X_2)$ , where  $Y_1$  and  $Y_2$  represent the received signals at the relay and the destination, respectively, and  $X_1$  and  $X_2$  represent the signals transmitted at the source and the relay, respectively. While the relay channel has continued to attract research interest throughout the years [109–118], the capacity of the most general relay model remains unknown.<sup>1</sup> The difficulty of characterizing the general relay channel capacity lies in the fact that with an arbitrarily given conditional distribution  $P(Y_2, Y_1 | X_1, X_2)$ , a relay may

---

<sup>1</sup>A short, non-comprehensive list of the types of relay channels for which the capacity is known is: separated relay channel (see Sec. 4.2), degraded and reversely degraded relay channel [108], semi-deterministic relay channel [109], permuting relay channel [110], deterministic relay channel [111], two-way relay channel [113], diamond channel [114], (genie-aided) non-causal relay channel [116].

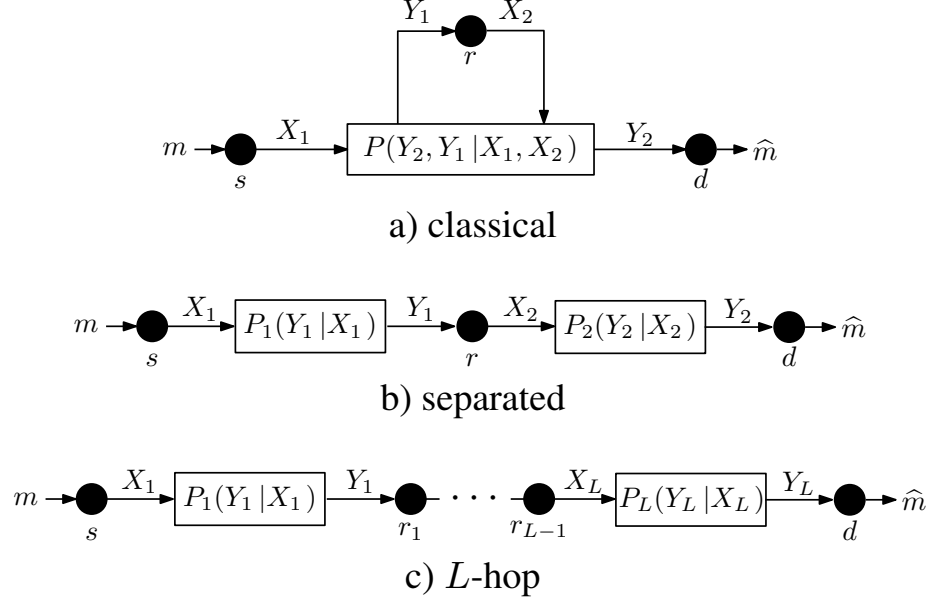


Fig. 4.1.: Variations on the relay channel model.

"assist" with the communication task between the source and the destination by judiciously selecting its transmitted value  $X_2$ . Various ideas have since been developed for the relay to better assist in the communication task, including Compress-&-Forward (CompressF) [108, 119], Compute-&-Forward [120], Noisy Network Coding [121], etc.

As challenging as the general relay channel study can be, in practice, almost all practical relay systems are assuming the so-called *separated relay channel* in Fig. 4.1b)., for which there is no direct link between the source and the destination. Specifically, the joint distribution admits a simpler form  $P(Y_2, Y_1 | X_1, X_2) = P_1(Y_1 | X_1)P_2(Y_2 | X_2)$  that separates the destination from the source. By the max-flow/min-cut theorem, the capacity of the separated relay channel is the bottleneck hop capacity:  $C = \min(C_1, C_2)$ , where  $C_1$  (resp.  $C_2$ ) is the capacity of the source-to-relay (resp. relay-to-destination) channel, and the capacity is achieved by Decode-&-Forward (DF) [108]. The separated relay channel model and its capacity analysis can be easily extended to the multi-hop line network depicted in Fig. 4.1c). While being one of the simplest communication networks, multi-hop line networks are arguably the most widely used relay model for any wireless/wireline network, for which the sole

task of the intermediate nodes is to "relay" the messages from source to destination, not to actively alter/assist the direct transmission<sup>2</sup> through  $P(Y_2, Y_1 | X_1, X_2)$ .

### 4.1.3 Reliability Function of Multi-hop Line Networks

Motivated by the renewed focus on lowering the latency of dense communication networks, this work studies the reliability function of multi-hop line networks. As is commonly adopted in the early studies of point-to-point channels, we use the error exponent as a tool to analyze the communication latency over these networks. In particular, we study the following questions:

*Question 1:* For any given multi-hop line network, does there exist an upper bound on the (largest) error exponent for any transmission scheme one can possibly design?

*Question 2:* If such a bound on the error exponent exists, can we design a new scheme from scratch that attains said bound?

Our goal of finding the optimal error exponent among *all possible schemes* separates this work from the existing results. For example, [96] derived the error exponents of the existing (partial) DF scheme, the Compress-F scheme, etc., and no new scheme was developed.<sup>3</sup> In contrast, this work is not bound by any existing solutions and aims to directly optimize the end-to-end error exponent by proposing new analytical approaches and design ideas.

---

<sup>2</sup>In wireless network terminology, if a third-party node actively alters/assists the main transmission from A to B, it is sometimes known as a *distributed multi-input, multi-output (MIMO)* solution [122] rather than being a relay design.

<sup>3</sup>As will be shown in Lemma 4.2.1 in Sec. 4.2, none of the existing block-based schemes, including DF, Compress-&-Forward, noisy network coding, etc., can attain the error exponent upper bound in Proposition 4.3.1.

#### 4.1.4 Our Contributions

##### 4.1.4.1 A new problem formulation

Any rigorous discussion of "optimality" must be accompanied by a clear definition of the set of schemes over which the optimality is defined. In this work, we formulate a general class of relay schemes for multi-hop line networks with  $L$  hops, which has the following desirable features: (i) It includes any existing relay schemes as special cases and enables fair comparison that takes into account various important techniques like Markov block coding, pipelining, and/or full-duplex capability; (ii) It is flexible for any  $L \geq 1$ ; (iii) If  $L = 1$  the class of schemes (and its definitions of rates and block-lengths) naturally coincides with the traditional definitions of block codes for point-to-point channels [89]; and (iv) If  $L = 2$ , the class of schemes (and its definitions of rates and block lengths) naturally falls back to the definitions of traditional block codes for the classical channel model [108].

With the new problem formulation, for any arbitrarily given scheme, we introduce a new metric dubbed the *delay amplification factor* (DAF), which *essentially measures the ratio of the delay of applying the scheme over the  $L$ -hop line network to the (optimal) random-coding delay over just the bottleneck hop, assuming we operate in the asymptotic regime  $R \rightarrow C$  and  $\epsilon \rightarrow 0$* . For example, over a 5-hop line network, a scheme achieving DAF=3.5 has an asymptotic delay roughly 3.5 times higher than the random-coding delay over just the bottleneck hop, and incurs only  $\frac{3.5}{5} = 70\%$  delay when compared to a uniform time-division transmission scheme.<sup>4</sup> We can then use the new DAF metric to compare the delay performance of any different schemes.

---

<sup>4</sup>A uniform time-division transmission scheme over an  $L$ -hop line network will take  $L$  times the delay experienced in the bottleneck hop for the packets to traverse from the source to destination over the  $L$  hops.

#### 4.1.4.2 An achievability scheme for the open-loop setting

It is intuitive<sup>5</sup> that the  $\text{DAF} \geq 1$  for any scheme and that for DF schemes, we have  $\text{DAF} = O(L)$  since their delay grows linearly with respect to the number of hops. To answer the questions posed in Sec. 4.1.3, we thus need to close the gap between the lower bound  $\text{DAF} \geq 1$  and the state-of-the-art  $\text{DAF} = O(L)$ . In particular, we show that by designing a new scheme from scratch, we can attain  $\text{DAF} = 1$  if the bottleneck hop is the  $L$ -th hop (the last hop) of the line network. The contribution of this finding is two-fold. Analytically, it shows that the common belief that the delay over an  $L$ -hop line network grows linearly with respect to  $L$  is not a fundamental limit, but rather an artifact of the delay-suboptimal DF schemes. Operationally, it shows that with new transmission schemes, system designers could significantly lower the end-to-end delay when compared to the de-facto industry standard DF solutions. This thus opens up new possibilities for next-generation low-latency communication standards. Also see [123] for some system-level design ideas and numerical verifications on lowering the end-to-end delay beyond what is possible in the traditional DF-based paradigm.

#### 4.1.4.3 An achievability scheme for the stop-feedback setting

Motivated heavily by the goals of designing new practical URLLC schemes, we also consider the effects of feedback from the perspective of DAF. Specifically, Contribution 2 is based on an open-loop feedback-free setting with fixed transmission duration. However, for practical wireless multi-hop communications, we almost always have (some form of) ACK feedback for each of the  $L$  hops. Even in the simpler point-to-point channel ( $L = 1$ ), the use of ACK feedback has led to significant performance improvements in the form of hybrid ARQ [124–126]. In this contribution, we thus consider the setting of *one-time stop-feedback* [100] of the system. Namely, we consider a variable-length relay scheme that adjusts its transmission duration based on a one-time, one-bit stop-feedback within the network. We show that with the

---

<sup>5</sup>We will formalize this part of discussion in Sec. 4.2.

help of the stop-feedback, we can relax the condition "bottleneck hop must be the last hop" and design a scheme with DAF=1 for *arbitrary line networks*. This finding establishes that with the one-time stop-feedback, the asymptotic delay of a multi-hop line network can always be made as small as the asymptotic delay of its bottleneck hop.<sup>6</sup>

The remainder of this paper is structured as follows. We give the channel model and all necessary definitions in Sec. 4.2.1 – 4.2.3. Sec. 4.3.1 contains the problem formulation in the form of a simple converse bound on the DAF metric. We provide some intuition and discuss the application of our analysis to the DF and Amplify-&-Forward (AF) schemes in Sec. 4.2.4. Our two DAF-optimal transmission schemes are presented in Sec. 4.3 and 4.4, respectively. Finally, we conclude the paper in Sec. 4.5.

## 4.2 Problem Formulation

### 4.2.1 The Multi-hop Line Network Channel Model

We define the stationary memoryless  $L$ -hop line network as follows, also see Fig. 4.1c). Denote the source node as  $s$ , the destination node as  $d$  and the  $L - 1$  intermediate relay nodes as  $r_\ell$  for all  $\ell = 1, \dots, L - 1$ , respectively. Consider slotted transmission for  $t = 1, 2, \dots$ . One symbol is sent in each time slot, which is sometimes called a *channel use*<sup>7</sup>. Each hop is discrete and memoryless and we denote the input and output symbols of the  $\ell$ -th hop at time slot  $t$  as  $X_\ell(t)$  and  $Y_\ell(t)$ , respectively. We denote the input alphabet, output alphabet, and conditional distribution of the  $\ell$ -th channel as  $\mathcal{X}_\ell$ ,  $\mathcal{Y}_\ell$ , and  $P_\ell(y|x)$ , respectively, i.e., for any input symbol  $X = x \in \mathcal{X}_\ell$ , the probability of observing  $Y = y \in \mathcal{Y}_\ell$  at the channel output is denoted as  $P_\ell(y|x)$ . We refer to the  $\ell$ -th channel as  $\mathcal{W}_\ell = (\mathcal{X}_\ell, \mathcal{Y}_\ell, P_\ell)$ .

<sup>6</sup>It is known that the stop-feedback can shorten the (expected) delay of variable length coding [100]. For fair comparison, we define the DAF in Sec. 4.2.3 as the ratio of the expected delay of the given scheme over the *improved* expected delay over the bottleneck hop. The discussion of DAF=1 for one-time stop-feedback is based on this new definition.

<sup>7</sup>In practice, the time for each channel use may vary from hop to hop. For simplicity, our model assumes all hops sharing a common channel use time. Our results can be easily revised to handle heterogeneous slot duration as well.



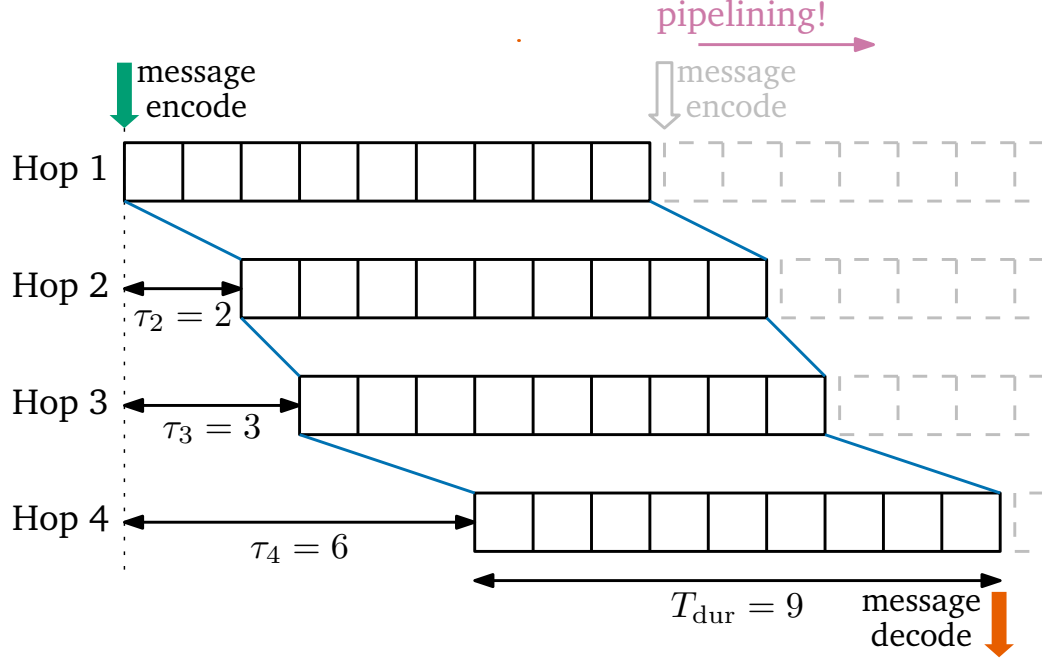


Fig. 4.2.: Starting times, transmission duration, and encoding/decoding timepoints for an example open-loop scheme (unit: slots).

Denote the capacity of the  $\ell$ -th hop by  $C_\ell$  (unit: nats/slot), which is finite since the alphabet  $\mathcal{X}_\ell$  is discrete and finite. Clearly, the capacity of the end-to-end channel is  $C = \min_\ell C_\ell$ .

**Technical Assumptions.** *Assumption 1:* All our results assume exclusively that there exists a unique hop  $\ell^*$  with the lowest capacity, i.e.,  $\exists \ell^*$  such that  $C_\ell > C_{\ell^*} \forall \ell \neq \ell^*$ . We refer to this hop as the *bottleneck hop* and assume  $C_{\ell^*} > 0$ . In practice, the corner case that there are two hops satisfying  $C_{\ell_1} = C_{\ell_2}$  with infinite precision is very unlikely. Furthermore, one can always apply some infinitesimal perturbation to break the tie if it happens. This assumption is thus not very restrictive for practical applications.

*Assumption 2:* We assume that all transmission probabilities are non-zero, i.e.,  $P_\ell(y|x) > 0 \forall x, y, \ell$ . This assumption is to ensure that the error exponent of any of the  $L$  hops is bounded away from infinity and can be relaxed in ways similar to Conditions (a) and (b) in [127, [Sec. 1].

The source  $s$  wishes to send an integer message  $m$ , drawn uniformly randomly from  $\mathcal{M} = \{1, 2, \dots, |\mathcal{M}|\}$  to destination  $d$  using a *transmission scheme*  $\Phi$ . A transmission scheme consists of the following elements depending on whether it is an open-loop setting or if stop-feedback is allowed.

#### 4.2.2 Open-loop setting

**Starting times and duration.** A sequence of  $L$  deterministic, non-decreasing time points  $\tau_1 = 0 \leq \tau_2 \leq \dots \leq \tau_L < \infty$  determines the starting times (unit: slots) of data transmission for the corresponding hops. The maximum duration<sup>8</sup> of the transmission at each node is denoted as  $T_{\text{dur}}$  (unit: slots). Fig. 4.2 gives an example for  $T_{\text{dur}} = 9$  and starting times  $\tau_1 = 0, \tau_2 = 2, \tau_3 = 3$ , and  $\tau_4 = 6$  over  $L = 4$  hops.

**Sequential encoders at the relay nodes.** We assume full-duplex relays with causal encoding. That is,

$$X_1(t) = f_t^{(1)}(m), \quad \forall t \in (\tau_1, \tau_1 + T_{\text{dur}}] \quad (4.1)$$

$$X_\ell(t) = f_t^{(\ell)}([Y_{\ell-1}]_*^{t-1}), \quad \forall \ell \geq 2, \forall t \in (\tau_\ell, \tau_\ell + T_{\text{dur}}] \quad (4.2)$$

where  $f_t^{(\ell)}$  is the encoder of the  $\ell$ -th hop at time slot  $t$ , and

$$[Y_{\ell-1}]_*^{t-1} \triangleq \{Y_{\ell-1}(\tau) : \tau \in (\tau_{\ell-1}, \min(t-1, \tau_{\ell-1} + T_{\text{dur}})]\} \quad (4.3)$$

denotes all causally received observations from the upstream hop. The definition  $Y_{\ell-1}(\tau)$  imposes that the observation at the transmitter of the  $\ell$ -th hop is always a subset of the upstream hop's "active period"  $(\tau_{\ell-1}, \tau_{\ell-1} + T_{\text{dur}}]$ .

**Block decoder at the destination.** The final block-based decoding function is given as

$$\hat{m} = g([Y_L]_*^{\tau_L + T_{\text{dur}}}). \quad (4.4)$$

---

<sup>8</sup> $T_{\text{dur}}$  is the maximum number of slots that can be used by each hop. A scheme can instruct some of its relays to use less than  $T_{\text{dur}}$  slots if desired.

The reason for defining  $\tau_1$  to  $\tau_L$  and the maximum duration  $T_{\text{dur}}$  is to accommodate pipelined transmission. That is, once the source finishes transmitting the current message at time  $\tau_1 + T_{\text{dur}} = T_{\text{dur}}$ , it can immediately inject the next message at time  $T_{\text{dur}} + 1$ , even though the current message is still being transmitted to the destination by the rest of the network. This behavior is illustrated with the dotted lines in Fig. 4.2. While this model requires full-duplex capability, it is in line with most existing information-theoretic relay models, e.g., [96, 108, 112], for which full duplex is commonly assumed. In practice, different hops may be operating over different bands, for which the half-duplex receivers of the previous hop and the half-duplex transmitters of the next hop may operate simultaneously. One may also employ the recent development of full-duplex antennas [128, 129] for in-band transmission.

The consequence of this pipelining assumption is that messages arrive at the destination once every  $T_{\text{dur}}$  slots, which is less than the total delay  $\tau_L + T_{\text{dur}}$ . This observation leads to the following classification of open-loop transmission schemes.

**Definition 4.2.1** *An  $L$ -hop open-loop transmission scheme attains a delay-throughput-error-rate tuple  $(T, R, \epsilon)$  if it satisfies*

$$T \geq \tau_L + T_{\text{dur}} \quad (4.5)$$

$$R \leq \frac{\ln(|\mathcal{M}|)}{T_{\text{dur}}} \quad (4.6)$$

$$\epsilon \geq \Pr(\hat{m} \neq m). \quad (4.7)$$

Let  $\mathcal{A}_\Phi$  denote the set of all  $(T, R, \epsilon)$ -tuples attained by scheme  $\Phi$ , i.e.,

$$\mathcal{A}_\Phi \triangleq \{(T, R, \epsilon) : \Phi \text{ attains } (T, R, \epsilon)\}. \quad (4.8)$$

**Definition 4.2.2** *The end-to-end error exponent of an open-loop scheme  $\Phi$  is defined as*

$$E_\Phi(R) \triangleq \limsup_{T \rightarrow \infty} \sup_{\epsilon: (T, R, \epsilon) \in \mathcal{A}_\Phi} \frac{-\ln(\epsilon)}{T}. \quad (4.9)$$

Denote the optimal random-coding error exponent of the  $\ell$ -th hop as  $E_{\text{rc},\ell}(R)$ . Here we note that since the random-coding error exponent usually depends on the input distribution used, we assume that in all of our use cases the optimal (i.e., maximizing the exponent) input distribution is used. Put succinctly, if  $E_{\text{rc},\ell}(R, \mathbf{Q})$  denotes the random coding error exponent of the  $\ell$ -th hop for channel symbols distributed according to  $\mathbf{Q}$ , then we always use  $E_{\text{rc},\ell}(R) \triangleq E_{\text{rc},\ell}(R, \mathbf{Q}^*(R))$  as a shorthand assuming the optimal distribution of the channel symbols for the given rate (here denoted as  $\mathbf{Q}^*(R)$ ).

Finally, Def. 4.2.3 presents the main metric investigated in this text for the open-loop setting.

**Definition 4.2.3** *The delay amplification factor (DAF) of an  $L$ -hop open-loop transmission scheme  $\Phi$  is defined as*

$$\Gamma_{\Phi} \triangleq \lim_{R \rightarrow C} \frac{E_{\text{rc},\ell^*}(R)}{E_{\Phi}(R)}. \quad (4.10)$$

### 4.2.3 Stop-feedback setting

In the stop-feedback setting, we assume the existence of a feedback link for all hops. The feedback model we assume is the *one-time stop-feedback* model [100]. In this model, we assume that there is a feedback channel from the destination to all other nodes (the source plus the relays). The feedback channel is assumed to be delay-free and error-free and can be used (by the destination) only once per message to indicate the end of a message transmission. The main motivation behind this feedback model is its application to real-world communication systems, in which control data like feedback messages are usually handled by a different layer than the bulk data transmission [100]. From this perspective, it is a more realistic model than the *per-slot* channel output feedback model commonly found in the literature [130–138], which assume that every channel output symbol is causally available at the transmitter and can be prohibitively expensive in terms of the delay and bandwidth requirements

imposed on the feedback channel(s). In contrast, the stop-feedback model is closer to real systems like hybrid ARQ where only a small amount of feedback is transmitted per message.

Technically, the stop-feedback model implies a) that the message duration  $T_{\text{dur}}$  becomes a stopping time of the filtration generated by  $[Y_L]_*^t$ , since the destination now triggers the *end-of-transmission* for each message and b) since there is only a single stop-feedback for each message, before the stop-feedback, all nodes are working simultaneously on transmitting the same message and collectively switch to the next message after receiving the end-of-transmission feedback.

As a result, we hardwire  $\tau_1 = \tau_2 = \dots = \tau_L = 0$  and modify Def. 4.2.1 for the stop-feedback setting as follows.

**Definition 4.2.4** *An  $L$ -hop stop-feedback transmission scheme attains a delay-throughput-error-rate tuple  $(T, R, \epsilon)$  if it satisfies*

$$T \geq E\{T_{\text{dur}}\} \quad (4.11)$$

$$R \leq \frac{\ln(|\mathcal{M}|)}{E\{T_{\text{dur}}\}} \quad (4.12)$$

$$\epsilon \geq \Pr(\hat{m} \neq m). \quad (4.13)$$

Namely,  $T$  in (4.11) defines the average delay,  $R$  in (4.12) defines the long-term throughput, and  $\epsilon$  in (4.13) defines the probability of error for each variable-length transmission.

The definition of the end-to-end error exponent for the stop-feedback setting remains identical to Def. 4.2.2. However, due to the availability of the stop-feedback, the bottleneck error exponent changes. Results in [100] show that for a point-to-point channel, stop-feedback improves the random-coding error exponent from  $E_{\text{rc},\ell}(R)$  to a strictly larger value

$$E_{\text{sf},\ell}(R) = (C_\ell - R)^+ \triangleq \max(C_\ell - R, 0), \quad (4.14)$$

which we use in the following new definition of the DAF.

**Definition 4.2.5** *The DAF of an  $L$ -hop variable-length stop-feedback scheme  $\Phi$  is defined as*

$$\tilde{\Gamma}_{\Phi} \triangleq \lim_{R \rightarrow C_{\ell^*}} \frac{E_{\text{sf}, \ell^*}(R)}{E_{\Phi}(R)} = \lim_{R \rightarrow C_{\ell^*}} \frac{C_{\ell^*} - R}{E_{\Phi}(R)}. \quad (4.15)$$

#### 4.2.4 Discussion #1: Decode-&-Forward is a special instance of this framework

Our first contribution is the new formulation of the  $L$ -hop line network in Sec. 4.2.1 to 4.2.3. One can easily see that when  $L = 1$ , since we always have  $\tau_1 = 0$ , the term  $T_{\text{dur}}$  is equivalent to the codeword length of block coding. The new definitions in Secs. 4.2.1 and 4.2.2 are thus identical to the traditional point-to-point channel reliability functions.

To further illustrate the flexibility of the new framework, we provide some intuition and discussion on how it includes DF as a special case. We restrict our focus to the open-loop setting since it is how DF was originally designed, but the insights gained apply to the stop-feedback variation of DF e.g., using the stop-feedback scheme in [100] as part of DF.

For illustration purposes, we assume that the DF scheme uses random block codes for each of the  $L$  hops. Specifically, in a DF scheme the source  $s$  first produces a corresponding randomly-generated codeword of block length  $t_1$  (unit: slot) and forwards it over the first channel to  $r_1$ . After the codeword is received, possibly in error, at the relay  $r_1$ ,  $r_1$  computes an estimate of the message and produces another randomly generated codeword using this estimate. Suppose that the length of  $r_1$ 's codeword is  $t_2$  (unit: slots). This codeword is then forwarded to  $r_2$ , and so forth. After  $L$  hops, the destination receives a codeword of length  $t_L$  and computes an estimate of the original message, denoted by  $\hat{m}$ . This procedure is illustrated in items

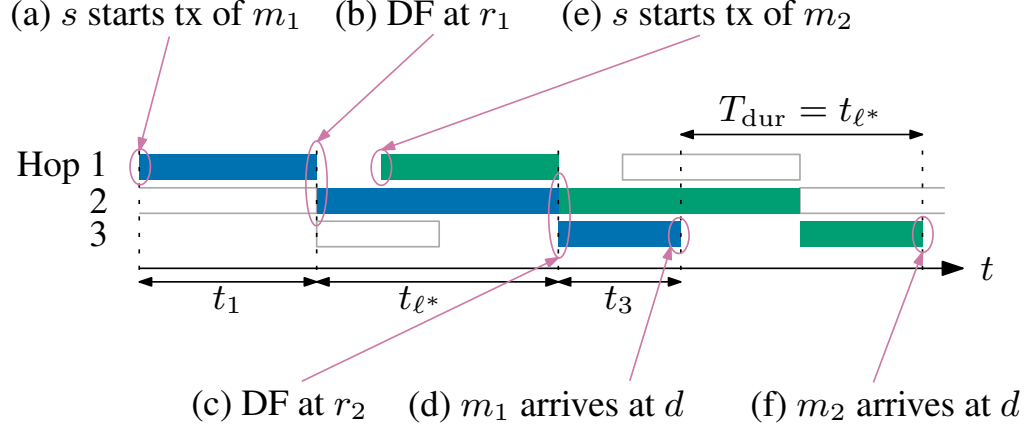


Fig. 4.3.: Relaying a message through an example 3-hop line network using the DF scheme.

(a)–(d) of Fig. 4.3. The codeword lengths  $t_1, t_2, \dots, t_L$  may vary for each hop and the end-to-end delay is thus  $T \triangleq t_1 + t_2 + \dots + t_L$ .

We now explain how this DF description fits the framework in Secs. 4.2.1 and 4.2.2. Define  $\ell^{**} \triangleq \operatorname{argmax}_{\ell} \{t_{\ell}\}$ . We set the starting times  $\{\tau_{\ell}\}$  and duration  $T_{\text{dur}}$  to be

$$\tau_{\ell} = \max \left( 0, -t_{\ell^{**}} + \sum_{j=1}^{\ell} t_j \right) \quad \forall \ell \in [1, L] \quad (4.16)$$

$$T_{\text{dur}} = \max_{\ell} t_{\ell} = t_{\ell^{**}}. \quad (4.17)$$

With the above choices, one can easily verify that (i)  $\tau_1 = 0 \leq \tau_2 \leq \dots \leq \tau_L$ ; and (ii) the *forwarding time slots* of the  $\ell$ -th hop are  $(\sum_{j=1}^{\ell-1} t_j, \sum_{j=1}^{\ell} t_j]$ , which is a subset of its active period  $(\tau_{\ell}, \tau_{\ell} + T_{\text{dur}}]$ . The DF encoders and decoders can thus be rewritten as a special instance of (4.1) to (4.4).

The achievable tuple  $(T, R, \epsilon)$  then satisfies

$$T = \sum_{\ell=1}^L t_{\ell} = \tau_L + T_{\text{dur}} \quad (4.18)$$

$$R = \frac{\ln(|\mathcal{M}|)}{t_{\ell^{**}}} = \frac{\ln(|\mathcal{M}|)}{T_{\text{dur}}} \quad (4.19)$$

$$\epsilon = \Pr(\hat{m} \neq m) = 1 - \prod_{\ell=1}^L (1 - \epsilon_{\ell}) \leq \sum_{\ell=1}^L \epsilon_{\ell} \quad (4.20)$$

where  $\epsilon_{\ell}$  is the random coding error probability of the  $\ell$ -th hop. Note that our pipelining rate definition in (4.19) (and thus (4.6)) indeed matches the commonly used definition of the achievable rate of DF and the delay definition in (4.18) also matches the commonly used definition of the delay of DF.

Following similar reasoning, one can easily show that other schemes like AF, Compress-F, Noisy Network Coding, etc. are all special instances of the new framework, and the rate and delay definitions of ours coincide with the commonly used definitions for the individual schemes. This thus provides the footing for a fair comparison between different schemes.

#### 4.2.5 Discussion #2: The physical interpretation of the DAF

We now demonstrate the relationship between the DAF and the end-to-end delay of an arbitrarily given scheme  $\Phi$ . Specifically, by (4.9), for any fixed rate  $R$  and error probability  $\epsilon$ , the end-to-end delay  $T$  of scheme  $\Phi$  is approximately

$$T(R, \epsilon) \approx -\ln(\epsilon)/E_{\Phi}(R). \quad (4.21)$$

If we apply a rate- $R$  random code over just the bottleneck hop  $\ell^*$  (while ignoring all other hops), to achieve the same error probability  $\epsilon$ , the corresponding delay is approximately

$$T_{\text{bttl}}(R, \epsilon) \approx -\ln(\epsilon)/E_{\text{rc}, \ell^*}(R). \quad (4.22)$$



The delay ratio,  $\frac{T(R,\epsilon)}{T_{\text{bttl}}(R,\epsilon)}$ , when evaluated in the asymptotic regime  $R \rightarrow C$  and  $\epsilon \rightarrow 0$  is indeed the DAF in (4.10).

The main motivation behind this DAF definition is to enable fair comparison of the end-to-end delay between different transmission schemes. In particular, since the latency performance depends on the channel distribution of all  $L$  hops, the expression of the end-to-end reliability function will inevitably be affected by the intricate interactions among all  $L$  hops (assuming general schemes, not necessarily the highly structured DF solution) and may not reveal the fundamental attributes of the proposed concepts and design choices. In contrast, the DAF captures the multiplicative increase in latency—or, effectively, the *error exponent penalty*—when communicating over  $L$  hops as opposed to communicating over the bottleneck hop alone. In proposing a metric relative to the underlying channel, we circumvent the complexity of jointly considering  $L$  heterogeneous hops and enable an even playing field for comparison.

To further demonstrate the new DAF metric, we calculate the DAF value of the DF scheme<sup>9</sup> discussed in Sec. 4.2.4.

**Lemma 4.2.1** *For the DF scheme, we have*

$$\Gamma_{\text{DF}} = \sum_{\ell=1}^L \frac{C_{\ell^*}}{C_{\ell}} \quad (4.23)$$

We note that DF represents a class of schemes that can have different choices<sup>10</sup> of the alphabet size  $|\mathcal{M}|$  and the active periods of each hop  $t_{\ell}$ , see Sec. 4.2.4. As a result, to prove Lemma 4.2.1, we have to show (i) there exists a way of choosing  $(|\mathcal{M}|, t_1, \dots, t_L)$  such that the corresponding DAF  $\leq \sum_{\ell=1}^L \frac{C_{\ell^*}}{C_{\ell}}$ ; and (ii) regardless

---

<sup>9</sup>Since we exclusively consider discrete memoryless channels, there is no concept of "amplifying" the received signal and thus there is no amplify-&-forward scheme. Additionally, DAF is defined only over capacity-achieving schemes because of the limit  $R \rightarrow C$  in (4.10) and (4.15). Since AF is in general not capacity-achieving, its DAF= $\infty$  even if we relax the models from discrete channels to continuous channels.

<sup>10</sup>In fact, the description of DF also includes how to choose the codebook for each hop. Lemma 4.2.1 and its proof in Appendix C.1 allow the class of DF to include any possible codebook choices as well. Nonetheless, due to the (near-)optimality of random codes, it is more intuitive for readers to temporarily assume random coding over each hop and focus only on the choices of  $(|\mathcal{M}|, t_1, \dots, t_L)$ .

how we choose  $(|\mathcal{M}|, t_1, \dots, t_L)$ , we always have  $\text{DAF} \geq \sum_{\ell=1}^L \frac{C_{\ell^*}}{C_{\ell}}$ . The detailed proofs of both directions are provided in Appendix C.1.

*Remark:* While the proof of Lemma 4.2.1 involves carefully applying the definitions in Secs. 4.2.1 and 4.2.2 to calculate the error exponents and its DAF value, the results are actually quite intuitive. Namely, suppose the source would like to send  $b$  bits to the destination using DF. It takes roughly  $\frac{b}{C_{\ell}}$  time slots to traverse over the  $\ell$ -th hop. The total delay is thus  $\sum_{\ell=1}^L \frac{b}{C_{\ell}}$ . If we only have the bottleneck hop, then the point-to-point channel delay is  $\frac{b}{C_{\ell^*}}$ . The ratio of total delay versus point-to-point bottleneck delay  $\sum_{\ell=1}^L \frac{C_{\ell^*}}{C_{\ell}}$  is indeed the DAF. Also see the first couple of paragraphs in Sec. 4.2.5.

For example, if  $C_1 = 5$  (nats/symbol),  $C_2 = 4$ , and  $C_3 = 3$ , the DAF of DF is  $\frac{3}{5} + \frac{3}{4} + \frac{3}{3} = 2.35$ . Namely, the end-to-end delay of DF in this 3-hop example is roughly 2.35 times the delay experienced in a point-to-point system consisting of only the third hop. One can clearly see that the DAF grows linearly with respect to  $L$ , assuming all  $\{C_{\ell}\}$  are of comparable magnitude.

### 4.3 Main Result #1: The open-loop setting

All the results in this section are based exclusively on the open loop setting in Secs. 4.2.1 and 4.2.2.

#### 4.3.1 The converse of the optimal DAF

**Proposition 4.3.1** *Regardless of the transmission scheme  $\Phi$ , we always have*

$$\Gamma_{\Phi} \geq 1.$$

**Proof** The proof is by reduction. Suppose there exists an  $L$ -hop line network  $\{P_{\ell}(y_{\ell}|x_{\ell}) : \ell \in [1, L]\}$  and a scheme  $\Phi$  such that  $\Gamma_{\Phi} < 1$ , which implies that  $E_{\Phi}(R) > E_{\text{rc}, \ell^*}(R)$  for some  $R$  sufficiently close to  $C = C_{\ell^*}$ . Recall that a scheme  $\Phi$

is determined by  $\{\tau_\ell : \ell \in [1, L]\}$ ,  $T_{\text{dur}}$ , and the encoding/decoding functions in (4.1) to (4.4). We will use the given scheme to construct a block code over the point-to-point (p2p) channel  $P_{\ell^*}(y_{\ell^*}|x_{\ell^*})$  and show that the corresponding error exponent will be strictly larger than the sphere packing bound  $E_{\text{sp},\ell^*}(R)$ , the needed contradiction.

The rest of the proof is straightforward. The block length of the p2p channel is set to  $T = \tau_L + T_{\text{dur}}$ . The encoder of the p2p channel uses the given multi-hop scheme, encodes the message, *simulates* the operations/transmissions of the first  $\ell^* - 1$  hops, and physically sends out the encoded symbols of what is supposed to be over the  $\ell^*$ -th hop over the physical p2p channel. Namely, it forfeits the first  $\tau_{\ell^*}$  time slots and only uses the interval  $(\tau_{\ell^*}, \tau_{\ell^*} + T_{\text{dur}}]$  even though the overall block length is  $\tau_L + T_{\text{dur}}$ . The receiver of the p2p channel will simulate the remaining  $L - \ell^*$  hops plus the final decoder. Per our definitions in (4.5) to (4.9), if the given multi-hop scheme achieves the error exponent  $E_\Phi(R) > E_{\text{rc},\ell^*}(R)$ , then the new p2p scheme will attain the same error exponent  $> E_{\text{rc},\ell^*}(R)$ . However, when  $R$  is sufficiently close to  $C$ , we must have  $E_{\text{rc},\ell^*}(R) = E_{\text{sp},\ell^*}(R)$ . This implies that the code surpasses the sphere packing bound. By contradiction, the proof is complete. ■

Proposition 4.3.1 effectively answers Question 1 from Sec. 4.1.3.

#### 4.3.2 An achievability scheme for the setting $\ell^* = L$

**Proposition 4.3.2** *Consider arbitrary  $L \geq 2$  and arbitrary channels  $P_\ell(y_\ell|x_\ell)$ . If the unique bottleneck hop is the last hop, i.e.,  $\ell^* = L$ , then there exists a scheme that achieves  $\Gamma_\Phi = 1$ .*

For the rest of this section, we explicitly construct such a scheme. By the lower bound from Proposition 4.3.1, such a scheme is DAF-optimal. Attaining DAF=1 also shows that the linearly growing delay DAF= $O(L)$  is an artifact of the existing DF designs, see Lemma 4.2.1, and it is possible to surpass it and attain the true fundamental limit DAF=1 with a new relay scheme in scenarios where  $\ell^* = L$ . In

Sec. 4.3.3, we describe the mechanics of the scheme in detail. We then prove that our scheme achieves the optimal DAF in Sec. 4.3.4.

### 4.3.3 Description of the Transmission Scheme

The main ideas of the scheme are as follows. Each message  $m$  is mapped to  $K$  microblocks of equal length  $\Delta$  symbols. The  $K$  microblocks are obtained using a concatenated code consisting of a single end-to-end outer code and a set of inner codes for each individual hop. After the source generates the  $K$  microblocks, each microblock is relayed through the  $L$  hops using DF in a pipelined block Markov coding fashion, but the decoders use *only* the inner codes. Since each of the  $K$  microblocks will take  $L \cdot \Delta$  time slots to traverse from the source to the destination using DF, the destination will receive all  $K$  microblocks after  $(K - 1 + L)\Delta$  slots once we assume pipelined transmission. Also see Fig. 4.4. After accumulating all  $K$  microblocks, the destination performs optimal *joint* inner/outer-code ML decoding.

We call such a scheme *transcoding* since the relay nodes do not perform full global decoding and re-encoding. Instead, the decoding and re-encoding operates on a local scale and "transforms" the signals from one inner codeword to another inner codeword along the hops. The detailed description of the transcoding scheme is provided below.

The transcoding scheme for an  $L$ -hop line network is parameterized by the number of microblocks  $K$ , the code rate  $R$  (unit: nats/symbol), and the microblock length  $\Delta$ . It consists of the following elements.

**Partitioning the time axis as microblocks.** Every  $\Delta$  time slots are grouped as a microblock. That is, the  $k$ -th micro block refers to the time slots  $t \in (k\Delta, (k+1)\Delta]$ . All the operations are aligned with the microblocks. In particular, we set the starting time instant  $\tau_\ell = (\ell - 1)\Delta$  for all  $\ell \in [1, L]$  and the duration  $T_{\text{dur}} = K\Delta$ .

**Inner/outer code architecture.**

- We define the inner code rate  $R_I = \frac{K}{K-1}R$ , for which the physical interpretation will be clear shortly after.

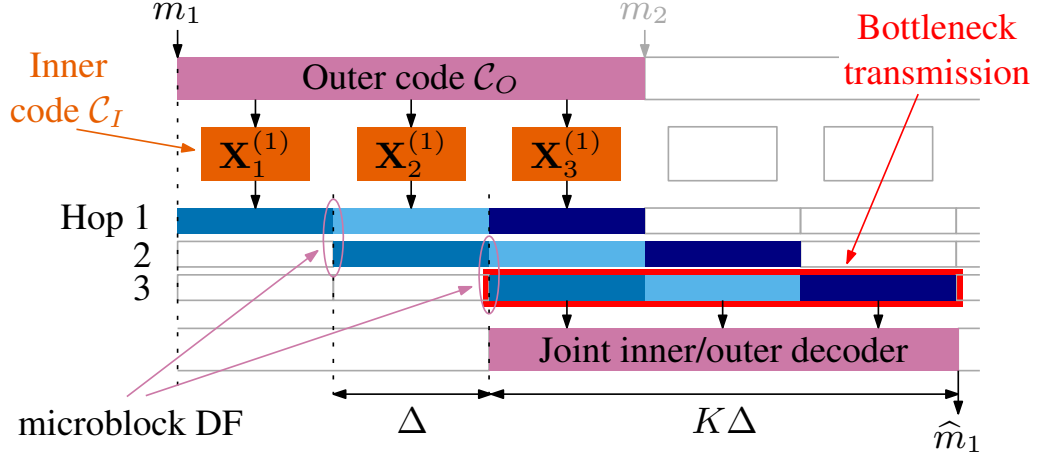


Fig. 4.4.: A DAF-optimal transmission scheme for bottleneck-terminated DMC line networks.  $K$  microblocks using concatenated code construction. ( $K = 3$  and  $L = 3$  pictured for illustration)

- The message set is  $\mathcal{M} = [1, e^{K\Delta R}] = [1, e^{(K-1)\Delta R_I}]$ .
- The outer code  $\mathcal{C}_O$  is a single parity check code over  $K$  *super symbols*, where each super symbol is chosen from  $[0, e^{\Delta R_I}]$ . Namely, any message  $m \in \mathcal{M}$  is first bijectively mapped to a  $(K-1)$ -dimensional vector  $(i_1^{[m]}, \dots, i_{K-1}^{[m]}) \in [0, e^{\Delta R_I}]^{(K-1)}$ . Then a parity super symbol  $i_K^{[m]}$  is computed from the parity check equation

$$\left( \sum_{k=1}^K i_k^{[m]} \mod e^{\Delta R_I} \right) = 0.$$

- A set of  $K \cdot L$  random inner codes

$$\mathcal{C}_I = \left\{ \{\mathbf{X}_1^{(1)}, \dots, \mathbf{X}_K^{(1)}\}, \dots, \{\mathbf{X}_1^{(L)}, \dots, \mathbf{X}_K^{(L)}\} \right\}.$$

As will be shown later,  $\mathbf{X}_k^{(\ell)}$  is the  $k$ -th codebook used by the  $\ell$ -th hop. We assume that  $\mathbf{X}_k^{(\ell)}$  is of codeword length  $\Delta$  symbols and is randomly generated by

$$\mathbf{X}_k^{(\ell)} = \left\{ \mathbf{x}_{k,i}^{(\ell)} \in (\mathcal{X}_\ell)^\Delta : i \in [0, e^{\Delta R_\ell}) \right\},$$

where each coordinate of the  $i$ -th ( $\Delta$ -dimensional) codeword  $\mathbf{x}_{k,i}^{(\ell)}$  is sampled independently and identically (i.i.d.) from the capacity-achieving input distribution  $P_\ell^*(X_\ell)$  of the  $\ell$ -th channel  $P_\ell(y_\ell|x_\ell)$ . The random codebook is generated independently for different  $k$  and  $\ell$  as well. Namely, even for the same  $\ell$ , we repeat the independent random codebook construction for different  $k_1$  and  $k_2$ .

**Encoding at the source node.** The source encoding function  $f_1 : \mathcal{M} \mapsto (\mathcal{X}_1)^{K\Delta}$  maps a message  $m$  to a block of symbols of length  $K\Delta$  by

$$f_1(m) = \left( \mathbf{x}_{1,i_1^{[m]}}^{(1)}, \mathbf{x}_{2,i_2^{[m]}}^{(1)}, \dots, \mathbf{x}_{K,i_K^{[m]}}^{(1)} \right),$$

where the  $i_k^{[m]}$  are obtained from the single-parity outer code and each microblock  $\mathbf{x}_{k,i}^{(1)}$  is drawn from  $\mathbf{X}_{k,1}$ . Since  $\tau_1 = 0$  and  $T_{\text{dur}} = K\Delta$  and all transmissions are aligned with the microblocks, the source  $s$  will transmit each inner codeword  $\mathbf{x}_{k,i_k^{[m]}}^{(1)}$  during the  $k$ -th microblock.

**Relaying through the network.** We define a set of  $K \cdot (L - 1)$  relaying functions  $f_k^{(\ell)} : (\mathcal{Y}_{\ell-1})^\Delta \mapsto (\mathcal{X}_\ell)^\Delta$  for all  $\ell \in [2, L]$  which map the channel outputs from the  $(\ell - 1)$ -th hop to the inputs of the  $\ell$ -th hop at the relays. The mapping is performed by combining the maximum-likelihood (ML) *inner code* decoder of the previous hop plus the inner code encoder of the current hop, i.e.,

$$f_k^{(\ell)}(\vec{Y}_{\ell-1}[k + \ell - 2]) = \mathbf{x}_{k,\hat{i}_k^{(\ell-1)}}^{(\ell)}, \quad (4.24)$$

where the vector  $\vec{Y}_{\ell-1}[j] \triangleq \{Y_{\ell-1}(\tau) : \tau \in ((j - 1)\Delta, j\Delta]\}$  is the  $\Delta$ -dimensional observation of the  $(\ell - 1)$ -th hop over the  $j$ -th microblock; the microblock index being

$k + \ell - 2$  is because since the starting time of the  $\ell$ -th hop is  $\tau_{\ell-1} = (\ell - 1)\Delta$ , the  $k$ -th microblock of the  $(\ell - 1)$ -th hop is occupying the  $(k + \ell - 2)$ -th microblock in the overall time axis, also see Fig. 4.4. The index

$$\hat{i}_k^{(\ell-1)} = \operatorname{argmax}_{i \in [0, e^{\Delta R_I})} P_{\ell-1} \left( \vec{Y}_{\ell-1}[k + \ell - 2] \mid \mathbf{x}_{k,i}^{(\ell-1)} \right) \quad (4.25)$$

is the optimal ML inner code decoder over the just received microblock, for which we slightly abuse the notation  $P_{\ell-1}(\cdot|\cdot)$  by letting

$$P_{\ell-1} \left( \vec{Y}_{\ell-1}[k + \ell - 2] \mid \mathbf{x}_{k,i}^{(\ell-1)} \right) \triangleq \Pr \left( \vec{Y}_{\ell-1}[k + \ell - 2] \mid \vec{X}_{\ell-1}[k + \ell - 2] = \mathbf{x}_{k,i}^{(\ell-1)} \right). \quad (4.26)$$

After decoding in (4.25) and re-encoding in (4.24), the  $k$ -th microblock of the  $\ell$ -th hop will be transmitted in the  $(k + \ell - 1)$ -th microblock in the overall time axis (recalling that the starting time  $\tau_\ell = (\ell - 1)\Delta$ ).

**Decoding at the destination.** We define a decoding function  $g : (\mathcal{Y}_L)^{K\Delta} \mapsto \mathcal{M}$  which maps an observation of  $K$  microblocks at the destination to an estimate of the message  $\hat{m}$  using the ML joint inner/outer decoder, i.e.,

$$\hat{m} = \operatorname{argmax}_{m \in \mathcal{M}} P_L \left( \left[ \vec{Y}_L \right]_1^K \mid \mathbf{c}_L^{[m]} \right), \quad (4.27)$$

where  $\left[ \vec{Y}_L \right]_1^K = \left\{ \vec{Y}_L[j + L - 1] : j \in [1, K] \right\}$  denotes the  $K$  microblocks received through the  $L$ -th hop;

$$\mathbf{c}_L^{[m]} = \left( \mathbf{x}_{1,i_1^{[m]}}^{(L)}, \dots, \mathbf{x}_{K,i_K^{[m]}}^{(L)} \right) \quad (4.28)$$

is the concatenation of the outer code with the  $K$  inner codebooks  $\mathbf{X}_1^{(L)}$  through  $\mathbf{X}_K^{(L)}$  designed for the  $L$ -th hop; and  $P_L(\mathbf{y} \mid \mathbf{c})$  is the conditional probability (likelihood) of receiving  $\mathbf{y} \in (\mathcal{Y}_L)^{K\Delta}$  at the destination given that  $\mathbf{c} \in (\mathcal{X}_L)^{K\Delta}$  was transmitted over the last hop, i.e., we slightly abuse the notation in a way similar to (4.26).

The error probability is defined as  $\epsilon = \Pr(m \neq \hat{m})$ . In the next subsection, we demonstrate how to choose the parameters  $(K, R, \Delta)$  of the transcoding scheme to attain  $\Gamma_\Phi = 1$ .

#### 4.3.4 DAF Analysis

Recall that any transcoding scheme is defined by the tuple  $(R, K, \Delta)$ . As a result, its error exponent is determined by the pair  $R, K$  since in (4.9) we let  $T \rightarrow \infty$  and thus the third coordinate  $\Delta \rightarrow \infty$ . As a result, we use the notation  $E_\Phi(R, K)$  for the error exponent of the transcoding scheme  $\Phi(R, K, \Delta)$ . We then have

**Lemma 4.3.1** *Assume  $\ell^* = L$ . For any  $R < C = C_{\ell^*}$  that is sufficiently close to  $C$ , we have*

$$E_\Phi(R, K^*(R)) = \frac{K^*(R)}{K^*(R) + L - 1} E_{\text{rc}, L}(R), \quad (4.29)$$

where  $K^*(R)$  is the smallest  $K$  such that  $R_I = \frac{K}{K-1}R < C$ . The closed-form expression of  $K^*(R)$  is

$$K^*(R) \triangleq \left\lfloor \frac{C}{C-R} \right\rfloor + 1 \quad (4.30)$$

One can easily see that when letting  $R \rightarrow C$ , the transcoding scheme  $\Phi(R, K^*(R))$  attains DAF=1 in Proposition 4.3.2 once we apply the definition in (4.10). The remainder of this section proves the above lemma.

#### Probability of Error

Recall that  $m$  was the selected message at the source. The corresponding tuple of microblock messages, generated by the source, is denoted as  $\mathbf{i}^{[m]} = (i_1^{[m]}, i_2^{[m]}, \dots, i_{K^*}^{[m]})$ . During the transmission, the estimate of the  $k$ -th microblock message at the receiver of the  $\ell$ -th hop is denoted as  $\hat{i}_k^{(\ell)}$ . Define  $\mathcal{A}$  as the event that there exists at least one pair  $(k, \ell)$  satisfying simultaneously  $k \in [1, K^*], \ell \in [1, L-1], \hat{i}_k^{(\ell)} \neq i_k^{[m]}$ .



Recall that  $\left[\vec{Y}_L\right]_1^K$  denotes the received symbols at the destination. Using the union bound, we can then bound the probability of message error as

$$\begin{aligned}\epsilon &\triangleq \Pr(\hat{m} \neq m) \\ &\leq \Pr(\mathcal{A}) + \Pr\left(g\left(\left[\vec{Y}_L\right]_1^K\right) \neq m \mid \mathcal{A}^c\right).\end{aligned}\quad (4.31)$$

The intuition behind (4.31) is that since we transmit the message over  $L$  hops, there are two types of errors. The first type is the error caused by performing DF using only the inner codes during the first  $(L - 1)$  hops, and the second type is the error of the joint inner/outer decoder at the destination.

We proceed by bounding the two terms in (4.31) individually. For the first term, we use the individual random coding error exponents for the first  $L - 1$  hops and obtain the bound

$$\begin{aligned}\Pr(\mathcal{A}) &\leq \sum_{k=1}^{K^*} \sum_{\ell=1}^{L-1} e^{-\Delta E_{\text{rc},\ell}(R_I)} \\ &\leq K^* \sum_{\ell=1}^{L-1} e^{-\Delta E_{\text{rc},\ell}(C_L)}\end{aligned}\quad (4.32)$$

$$\leq K^*(L - 1)e^{-K^* \Delta E_{\text{rc},L}(R)},\quad (4.33)$$

where (4.32) follows from  $R_I < C_L$  due to our choice of  $K^*(R)$  in (4.30). (4.33) follows from the fact that when  $R$  is sufficiently close to  $C = C_L$ , we have

$$E_{\text{rc},L}(R) < \frac{\min_{l \in [1, L-1]} E_{\text{rc},l}(C_L)}{K^*}.\quad (4.34)$$

The reason is that  $E_{\text{rc},L}(R) = O((C - R)^2)$  is a quadratic function of  $(C - R)$  when  $R$  is sufficiently close to  $C$  [139, Ex. 5.23], but  $K^*(R)$  (or simply  $K^*$  as shorthand) is approximately  $\frac{C}{C-R}$ .

For the second term, we use the results in [140], which show that if the outer code is a Maximum Distance Separable (MDS) code, then the ML joint inner/outer decoder of any concatenated code over a point-to-point channel (assuming the outer

codeword length  $K$  is fixed and the inner codeword length  $\Delta$  grows to infinity) can achieve the same error exponent as non-concatenated random coding over the same channel. Since our parity-check outer code in (4.24) is MDS and we condition on the event  $\mathcal{A}^c$ , the concatenated coding is essentially applied only to the  $L$ -th hop and [140] immediately implies

$$\Pr(\hat{m} \neq m | \mathcal{A}^c) \leq e^{-K^* \Delta E_{rc,L}(R) + \varsigma(\Delta)}, \quad (4.35)$$

where  $\varsigma(\Delta)$  signifies a term which diminishes as  $\Delta \rightarrow \infty$ .

Combining (4.33) and (4.35), we then get

$$\epsilon \leq (K^*(L-1) + 1) e^{-K^* \Delta E_{rc,L}(R) + \varsigma(\Delta)}. \quad (4.36)$$

### End-to-end Latency

The length of each microblock is  $\Delta$ . There are  $K^*$  total microblocks per message. Each microblock needs to traverse  $L$  hops. In total, the latency of one message is given as

$$T = (K^* + L - 1)\Delta. \quad (4.37)$$

### Error Exponent and DAF

Now, combining the results about the error probability and delay, we can derive the end-to-end error exponent of the proposed transmission scheme:

$$\begin{aligned} E_{\Phi}(R) &= \lim_{T \rightarrow \infty} \frac{-\ln(\epsilon)}{T} \\ &= \lim_{\Delta \rightarrow \infty} \frac{-\ln[(K^*(L-1) + 1) e^{-K^* \Delta E_{rc,L}(R) + \varsigma(\Delta)}]}{(K^* + L - 1)\Delta} \\ &= \frac{K^*}{K^* + L - 1} E_{rc,L}(R). \end{aligned} \quad (4.38)$$

The proof of Lemma 4.3.1 is complete.

We now note that even if the bottleneck hop is not the last hop, i.e.,  $\ell^* \neq L$ , we can iteratively apply our proposed scheme to achieve the following DAF value.

**Corollary 4.3.1** *For the case of  $\ell^* \neq L$ , define  $\ell_0^* = 0$  and iteratively define  $\ell_i^* = \arg \min_{\ell \in (\ell_{i-1}^*, L]} C_\ell$  for  $i = 1, 2, 3, \dots$  until  $\ell_i^* = L$ . Suppose there are  $I$  such  $\ell_i^*$ , i.e.,  $\ell_I^* = L$ . Also assume the minimum is unique when computing each  $\ell_i^*$ . Then we can construct a scheme  $\Phi$  such that*

$$\Gamma_\Phi = \sum_{i=1}^I \frac{C_{\ell_i^*}}{C_{\ell_i^*}}. \quad (4.39)$$

In Corollary 4.3.1, one can easily see that our scheme outperforms the DAF of DF in all scenarios except the ones for which the capacity of each hop rises in ascending order  $C_1 \leq C_2 \leq \dots \leq C_L$ . Corollary 4.3.1 can be proved by combining the DF principle and the transcoding scheme from Proposition 4.3.2.

#### 4.4 Main Result #2: The stop-feedback setting

In this section, we present a transmission scheme which achieves DAF=1 in the stop-feedback setting. It is worth pointing out that DAF is defined as the asymptotic delay ratio between a given scheme over  $L$  hops and the random coding scheme over only the bottleneck hop. For the stop-feedback setting, there is no guaranteed optimality for random codes, which is in contrast with the open-loop setting for which random codes are known to be error-exponent optimal when  $R \rightarrow C$ . As a result, unlike Proposition 4.3.1 for the open-loop setting, our investigation has not established the converse  $\text{DAF} \geq 1$  for the stop-feedback setting. That said, random coding currently achieves the highest error exponent out of all existing stop-feedback solutions and we thus use it as the benchmark in our DAF definition.

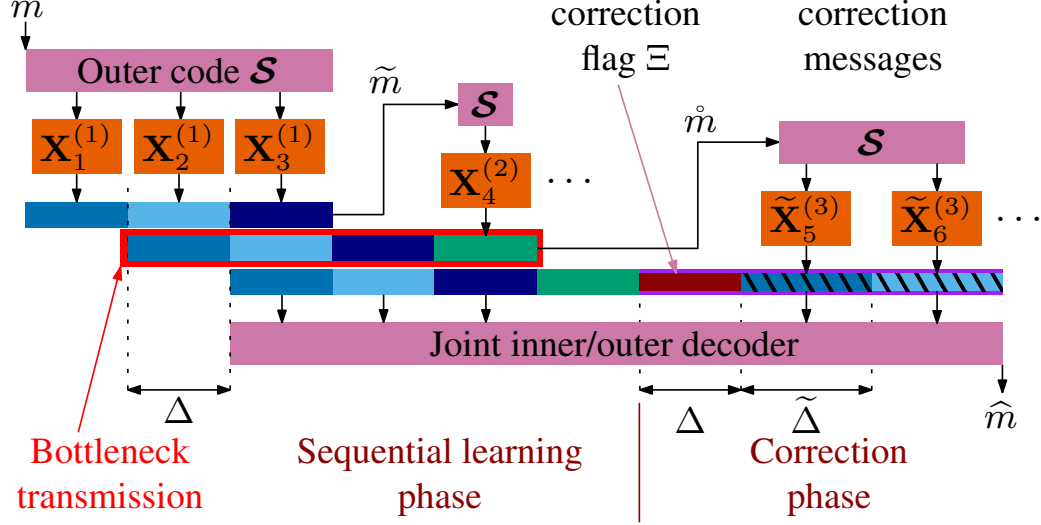


Fig. 4.5.: A DAF-optimal transmission scheme for DMC line networks with stop-feedback. Illustration for  $L = 3$  hops and  $K = 3$  microblocks. The bottleneck transmitter continues to send encoded microblocks after the initial  $K$  microblocks are forwarded. The sequential learning phase at the bottleneck receiver lasts  $K + 1$  microblock in this example. During the correction phase, the bottleneck receiver sends correction messages to the destination.

**Proposition 4.4.1** *Consider arbitrary  $L \geq 2$  and arbitrary channels  $P_\ell(y_\ell|x_\ell)$ . In the stop-feedback setting described in Section 4.2.3, there exists a scheme that achieves  $\tilde{\Gamma}_\Phi = 1$ .*

We proceed according to the same recipe as in Sec. 4.3 and first describe the scheme in detail in Sec. 4.4.1 before proving in Sec. 4.4.2 that it achieves DAF=1 regardless of whether  $\ell^* = L$  or not.

#### 4.4.1 Description of the Transmission Scheme

Recall that we refer to the hop with the minimum capacity as the *bottleneck hop*  $\ell^*$ . Our scheme operates regardless whether  $\ell^* = L$  or not. To describe the operation of each relay node, we need some additional terminology. We refer to the transmitter and receiver of the bottleneck hop as the *bottleneck transmitter* and *bottleneck receiver*, respectively. All relays between the source and the bottleneck transmitter are referred

to as the *pre-bottleneck relays* while all relays between the bottleneck receiver and the destination are referred to as the *post-bottleneck relays*.

The proposed scheme is a variable-length scheme with termination, i.e., all of the nodes in the network work on the same message until the destination signals an *end-of-transmission* (EOT) message over the feedback link. This feedback message then instantly propagates through the entire network and triggers the transmission of the next message.

In the subsequent discussion, we will use Fig. 4.5 as illustration, which shows an example transmission over an  $L = 3$  hop line network where the bottleneck hop is the second hop, i.e.,  $\ell^* = 2$ . Similar to the scheme from Sec. 4.3, this scheme utilizes a concatenated code consisting of an outer code mapping the message to a set of microblock messages and a set of inner codes mapping microblock messages to microblock codewords. Comparing to the fixed-length scheme in Section 4.3, that scheme uses its outer code to generate a fixed number of microblock messages, the outer code from this variable-length scheme generates a *variable-length sequence* of microblock messages for each message. However, as illustrated in Fig. 4.5, the source node and all the transmitters of the pre-bottleneck hops generate and transmit only the first  $K$  microblocks of the infinite-length sequence generated by the outer code. The additional microblock messages are used/sent only by the bottleneck transmitter and all the transmitters of the post-bottleneck hops until the destination declares EOT.

As will be elaborated further, among all the nodes, the bottleneck transmitter and the bottleneck receiver play the most important roles in this scheme. The bottleneck transmitter is the first node that operates indefinitely until receiving the EOT (all pre-bottleneck hops send only  $K$  microblocks) and the bottleneck receiver is the node that consumes the sequential microblock transmission over the bottleneck until it can compute an estimate of the source message with high confidence.<sup>11</sup>

---

<sup>11</sup>In this discussion, we use the term *high confidence* by its intuitive definition. The concept of high confidence will be made rigorous once we provide the complete proof in Sec. 4.4.2.

The superior (asymptotic) delay performance of our scheme lies in the unique design of the bottleneck receiver. For comparison, in a traditional DF scheme, only when the bottleneck receiver has high confidence of the source message, does it start to re-encode and forward the message to its downstream node(s). However, obtaining a high-confidence estimate requires accumulating a lot of received symbols, which incurs long delay. The main idea of our new scheme is for the bottleneck receiver to start forwarding even before it has deduced a high-confidence estimate, thus the reduced delay. The cost of this early forwarding is that at the time when the bottleneck receiver finally has a high-confidence estimate, it has already forwarded many "low-confidence" microblocks and some of them may be in error, which leads to higher error rate.

To address this drawback, we note that when the bottleneck receiver finally has enough observations to deduce a high-confidence estimate of the source message, it also has compiled the records of all low-confidence microblock messages that have been forwarded to downstream nodes prematurely (before having the high-confidence source message). As a result, we let the bottleneck receiver use the high-confidence estimate of the source message to recompute the set of high-confidence microblocks that *should have been forwarded* to its downstream nodes if it could go back in time. Then it compares the low-confidence microblocks that have actually been forwarded against the high-confidence microblocks that should have been forwarded. If there is any mismatch, the bottleneck receiver *forwards additional microblocks to correct the ones it now knows were relayed in error*.

To realize the above idea, we design two separate phases of the message transmission at the bottleneck receiver. In the *sequential learning phase*, the bottleneck receiver aims to obtain the high-confidence message estimate from its observations. In the *correction phase*, the bottleneck receiver aims to correct any mistake(s) due to premature forwarding. Also see the illustration in Fig. 4.5. Note that the lengths of both phases are random (and correlated), where the duration of the first phase depends on the channel realization and the duration of the second phase depends on

how many previously transmitted microblocks were in error. Since only the bottleneck receiver knows when the first phase ends and the second phase starts, it has to "instruct" the downstream nodes how to correctly interpret the traffic. To that end, we thus insert a *correction flag* in the forward traffic. The correction flag  $\Xi$  can take values in  $\{0, 1, \dots, K\}$ . The value  $\Xi = 0$  indicates that there is no correction needed and  $\Xi = j$  indicates that we will send an additional  $j$  microblocks that are meant to replace  $j$  of the previously transmitted microblocks. We limit the maximum number of replacements to be  $K$ , the reason of which will be made clear in the detailed analysis.

The post-bottleneck hops mainly perform DF to help relay the initial microblocks (during the sequential learning phase) and the correction microblocks (during the correction phase) to the destination. After the destination receives the correction flag  $\Xi$ , it terminates the transmission of the message for the entire network using the stop-feedback if  $\Xi = 0$ . Or, if  $\Xi = j \geq 1$ , it terminates the transmission after receiving  $j$  additional microblocks.

Based on the above high-level description, we now give a concise definition of the codes used in our proposed scheme. Specifically, the scheme has four deterministic parameters  $(K, \Delta, R, \alpha)$ , where  $\Delta$  is the microblock length and  $\alpha \in (0, 1)$  is a tuning parameter that will be used during the construction. The deterministic parameter  $K$  is the *target number of microblocks* used in the transmission since the actual number of microblocks is random.  $R$  is directly related to the message alphabet size via  $R = \frac{\log(|\mathcal{M}|)}{K\Delta}$  and can be interpreted as the *target end-to-end throughput*. Again, the actual throughput depends on the expected duration of the variable-length scheme, see (4.12), and will be analyzed later.

Since we always operate within the capacity, we are exclusively interested in the set of parameter values satisfying  $R < C = C_{\ell^*}$ . For given values of  $(K, \Delta, R, \alpha)$ , we compute the following constants:

$$R_I \triangleq R + \frac{\alpha + 1}{2}(C_{\ell^*} - R) \quad (4.40)$$

$$K_{\max} = K e^{K\Delta\alpha(C_{\ell^*} - R)} \quad (4.41)$$

$$\eta = e^{-K\Delta\alpha(C_{\ell^*} - R)}. \quad (4.42)$$

Intuitively,  $R_I$  is the inner code rate;  $K_{\max}$  is the upper bound on the length of our variable-length scheme which keeps the scheme from running indefinitely; and  $\eta$  is the *target error rate* of the given scheme.

**Outer code at the source.** The scheme uses a *sequential random permutation outer code (SRPOC)*  $\mathcal{S}$ . Given any  $(K, \Delta, R, R_I)$  tuple, the SRPOC is a rateless code consisting of

1. The message set  $\mathcal{M} = [1, e^{K\Delta R}]$ .
2. A finite sequence of permutations  $\{\pi_k : k \in [1, K_{\max}]\}$ .
  - A permutation  $\pi$  on  $\mathcal{M}$  is a bijective mapping from  $\mathcal{M}$  to  $\mathcal{M}$ . In total, there are  $|\mathcal{M}|!$  different permutations.
  - Each  $\pi_k$  in the finite sequence of permutations  $\pi_k : \mathcal{M} \mapsto \mathcal{M}$  for  $k \in [1, K_{\max}]$  is drawn independently and uniformly randomly from the set of all  $|\mathcal{M}|!$  possible permutations.
3. A finite sequence of encoding functions  $\{f_k\}$ .
  - It is best to interpret  $k$  as the *microblock index*.
  - For any  $k \in [1, K_{\max}]$  and the randomly chosen permutation  $\pi_k$ , the  $k$ -th encoding function  $f_k : \mathcal{M} \mapsto [1, e^{\Delta R_I}]$  maps a message  $m$  to an outer-code symbol  $i \in [1, e^{\Delta R_I}]$ .



- Specifically, given any  $m \in \mathcal{M}$ , the output  $i_k^{[m]}$  is the unique integer satisfying

$$(i_k^{[m]} - 1) \cdot \frac{|\mathcal{M}|}{b} < \pi_k(m) \leq i_k^{[m]} \cdot \frac{|\mathcal{M}|}{b}. \quad (4.43)$$

where  $b \triangleq e^{\Delta R_I}$ .

- This mechanism partitions the message set  $\mathcal{M}$  into buckets of  $b \triangleq e^{\Delta R_I}$  messages and selects the corresponding bucket index  $i_k^{[m]}$  for a given message  $m \in \mathcal{M}$  after applying the permutation  $\pi_k$ . This is closely related to the random binning technique, though described in a deterministic fashion. Furthermore, the bins/buckets are re-labeled independently for each  $k$ .

**Inner codes.** We use the standard random block code construction extensively. Namely, the coordinate values of each codeword are chosen i.i.d. according to the capacity-achieving input distribution. For mathematical rigor, we never reuse any codebook and all codewords/codebooks are generated independently. We now describe how many inner codebooks are used in our scheme.

1. For each of the pre-bottleneck hops  $\ell \in [1, \ell^*)$  and for all  $k \in [1, K]$ , we have a random inner codebook  $\mathbf{X}_k^{(\ell)}$  for which the codeword length is  $\Delta$  (unit: symbol) and the total number of codewords in each codebook is  $e^{\Delta R_I}$ . In total, there are  $K(\ell^* - 1)$  of them and these codebooks are used for microblock-based DF in the pre-bottleneck hops.
2. For the bottleneck hop, we have a finite number of inner codebooks  $\mathbf{X}_k^{(\ell^*)}$  for all  $k \in [1, K_{\max}]$  for which the codeword length is  $\Delta$  (unit: symbol) and the total number of codewords in each codebook is  $e^{\Delta R_I}$ . The only difference between the bottleneck hop and the pre-bottleneck hops is that the bottleneck hop needs to support variable-length encoding and thus we need to prepare a larger number of codebooks  $K_{\max} \geq K$ .

3. For each of the post-bottleneck hops  $\ell \in [\ell^* + 1, L]$ , we have *two* finite sequences of codebooks and we denote them by  $\{\mathring{\mathbf{X}}_k^{(\ell)} : k \in [1, K_{\max}]\}$  and  $\{\tilde{\mathbf{X}}_k^{(\ell)} : k \in [1, K]\}$ , respectively. The constructions of  $\mathring{\mathbf{X}}_k^{(\ell)}$  and  $\tilde{\mathbf{X}}_k^{(\ell)}$  have the following subtle but important differences.

- Each  $\mathring{\mathbf{X}}_k^{(\ell)}$  codebook has codeword length  $\Delta$  but the total number of codewords is  $e^{\Delta R_I} + K + 1$ . Namely, compared to the pre-bottleneck codebooks,  $\mathring{\mathbf{X}}_k^{(\ell)}$  has  $K + 1$  additional codewords which are used to represent the correction flag values  $\Xi = 0$  to  $\Xi = K$  discussed previously.
- Each  $\tilde{\mathbf{X}}_k^{(\ell)}$  codebook has codeword length  $\tilde{\Delta} = \Delta + \frac{\log(K)}{R_I}$  and the total number of codewords is  $e^{\tilde{\Delta} R_I} = K e^{\Delta R_I}$ . Namely, the effective rate of the codebook  $\tilde{\mathbf{X}}_k^{(\ell)}$  is still  $R_I$  (units: nats/symbol), but we slightly elongate the codeword length to accommodate a factor<sup>12</sup> of  $K$  more codewords than the pre-bottleneck codebooks  $\mathbf{X}_k^{(\ell)}$ . The new codebook  $\tilde{\mathbf{X}}_k^{(\ell)}$  is used to carry the *correction microblocks*. More specifically, to correct a microblock, one not only has to specify the new value of the previously incorrect microblock but also which of the previous microblocks is being corrected (i.e., the microblock index). Thus, we expand the number of codewords by a factor of  $K$  so that we can specify, out of the  $K$  previously sent microblocks, which one needs to be corrected by which value. More details can be found shortly after.

We are now ready to describe how to assemble the outer and inner codes for the final scheme.

**Encoding at the source node.** The source first maps  $m$  to a sequence of  $K$  microblock messages  $\{i_k^{[m]}\}_{k=1}^K$  using the outer code  $\mathcal{S}$ . Then it transmits  $\mathbf{x}_{k, i_k^{[m]}}^{(1)}$ , one codeword for each microblock, sequentially for  $k \in [1, K]$ . In the end, the source

---

<sup>12</sup>For comparison,  $\mathring{\mathbf{X}}_k^{(\ell)}$  has an additional  $K + 1$  codewords when compared to  $\mathbf{X}_k^{(\ell)}$  and  $\tilde{\mathbf{X}}_k^{(\ell)}$  has  $K$  times more codewords than  $\mathbf{X}_k^{(\ell)}$ .

transmits  $K\Delta$  symbols over the first hop. After the end of its transmission, the source idles<sup>13</sup>.

**Relaying at the pre-bottleneck nodes.** The transmitters of the pre-bottleneck hops ( $2 \leq \ell < \ell^*$ ) perform microblock DF using the inner-code decoders and encoders. More specifically, the transmitter of the  $\ell$ -th hop decodes the message of the  $k$ -th microblock using the codebook  $\mathbf{X}_k^{(\ell-1)}$  into an estimate  $\hat{i}_k^{(\ell-1)}$  and forwards the re-encoded codeword  $\mathbf{x}_{k, \hat{i}_k^{(\ell-1)}}^{(\ell)} \in \mathbf{X}_k^{(\ell)}$  to the next node. After forwarding  $K$  microblocks, each of the pre-bottleneck nodes enter the idle state.

**Operation of the bottleneck transmitter.** While relaying the first  $K$  microblocks, the bottleneck transmitter mirrors the behavior of the transmitters of the pre-bottleneck hops. However, after forwarding  $K$  microblocks through the bottleneck hop, the operation of this transmitter starts to differ.

After receiving the  $K$ -th microblock, the bottleneck transmitter checks whether its past  $K$  estimates of the microblock messages  $\{\hat{i}_k\}_{k=1}^K$  correspond to a unique codeword  $\tilde{m}$  from the message set. Namely, using the SRPOC outer code, the bottleneck transmitter computes

$$\tilde{\mathcal{M}}_{\ell^*-1} \triangleq \left\{ m : \hat{i}_k^{(\ell^*-1)} = i_k^{[m]} \ \forall k \in [1, K] \right\}. \quad (4.44)$$

If the set contains a single message, i.e.,  $\tilde{\mathcal{M}}_{\ell^*-1} = \{\tilde{m}\}$ , the bottleneck transmitter effectively subsumes the role of the source and continues to use SRPOC outer code to transmit microblocks of the estimated message  $\tilde{m}$  until the stop-feedback from the destination or until it has reached the maximum number of transmissions  $k = K_{\max}$ . That is, it sends  $\mathbf{x}_{k, \hat{i}_k^{[\tilde{m}]}}^{(\ell^*)} \in \mathbf{X}_k^{(\ell^*)}$  for  $k \in [K+1, K_{\max}]$  until EOT.

If  $|\tilde{\mathcal{M}}_{\ell^*-1}| \neq 1$ , i.e., either no such  $\tilde{m}$  can be found or multiple  $\tilde{m}$  correspond to the past decisions, the transmitter declares an error event<sup>14</sup> and enters the idle state.

<sup>13</sup>Due to the lack of a special "idle" symbol, in the subsequent discussion, we assume that a node in the idle state arbitrarily picks one symbol from its channel input alphabet and transmits it repeatedly until the next message is to be transmitted.

<sup>14</sup>Note that the transmitter does not need to send any "error flags" to the downstream nodes and only needs to idle. The error-event declaration is only used for analysis purposes.

This mechanism is illustrated in Fig. 4.5, where the bottleneck transmitter uses its message estimate  $\tilde{m}$  to produce additional microblocks.

**Operation of the bottleneck receiver.** As we briefly mentioned earlier, the operation of the bottleneck receiver during one message transmission consists of two distinct phases: the sequential learning phase and the correction phase. We describe both phases separately below.

1) *The sequential learning phase.* During the sequential learning phase, the bottleneck receiver computes a running log-likelihood ratio (LLR) for each of the messages in the message set  $\mathcal{M}$ . When any of the ratios exceeds a certain threshold, it switches to the correction phase. More precisely, recall that  $\vec{Y}_{\ell^*}[k]$  denotes the  $\Delta$ -dimensional received signals of the  $k$ -th microblock and let  $\left[\vec{Y}_{\ell^*}\right]_1^k = \left\{\vec{Y}_{\ell^*}[j + \ell^* - 1] : j \in [1, k]\right\}$  denote the vector of all observed channel outputs at the bottleneck receiver after the receiving the  $k$ -th microblock. Then, for every message  $m \in \mathcal{M} = [1, e^{K\Delta R}]$ , the bottleneck receiver computes the LLR

$$Z_m(k) = \ln \left( \frac{\Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m\right)}{\sum_{m' \neq m} \Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m'\right)} \right), \quad (4.45)$$

where the conditional probability functions are computed by

$$\Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m\right) \triangleq \Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| \mathbf{c}_{\ell^*}^{[m]}\right) \quad (4.46)$$

$$= \prod_{i=1}^{k\Delta} P_{\ell^*}(y_i | c_i) \quad (4.47)$$

and

$$\mathbf{c}_{\ell^*}^{[m]} = \left( \mathbf{x}_{1, i_1^{[m]}}^{(\ell^*)}, \mathbf{x}_{2, i_2^{[m]}}^{(\ell^*)}, \dots \right). \quad (4.48)$$

That is, the conditional probability function is based on the assumption that the sequence of transmitted channel symbols (over the bottleneck hop) is produced directly

by the message  $m$  using the joint outer/inner encoder. Note that this assumption is false since the symbols transmitted over the  $\ell^*$ -th hop are based either on the inner codeword estimate  $\hat{i}_k^{(\ell^*-1)}$  of the previous hop (for those  $k \in [1, K]$ ) without the help of the outer codebook, or on the re-encoded versions based on  $\tilde{m}$  (for those  $k \in [K + 1, K_{\max}]$ ), cf. the discussion around (4.44). Hence (4.46) should be viewed as a way of computing the "score" of each message  $m$ , not the actual likelihood of receiving  $m$ .

Using this running LLR value for each  $k$ , the bottleneck receiver performs what is essentially a sequential probability ratio test [141] to determine the end of the phase and an estimate of the current message. In other words, the sequential learning phase ends whenever there exists one  $m$  such that

$$Z_m(k) > \ln \left( \frac{1}{\eta} \right), \quad (4.49)$$

where  $\eta$  is the target error probability parameter fixed earlier. Since  $\log \left( \frac{1}{\eta} \right) > 0$  per our choice of  $\eta$ , by (4.45) we can have at most one  $m$  satisfying (4.49). Namely, if there exists any  $m$  satisfying (4.49), such  $m$  must be unique.

Additionally, to limit the maximum latency of our scheme, we further restrict the length of the sequential learning phase to be at most  $K_{\max}$ . Namely, if the sequential learning phase has not properly ended by the  $K_{\max}$ -th microblock (i.e., there was no such  $m$  satisfying (4.49)), we declare "sequential learning failure" and forcefully terminate the sequential learning phase.

In addition to performing the described probability ratio test, during the sequential learning phase, the bottleneck receiver, (i.e., the transmitter of the  $(\ell + 1)$ -th hop) continues to relay microblocks using the familiar microblock DF scheme. However, it uses the codebooks  $\mathring{\mathbf{X}}_k^{(\ell+1)}$  defined earlier, which have the same microblock length  $\Delta$  but contain  $e^{\Delta R_I} + K + 1$  codewords. The additional  $K + 1$  codewords represent the value of the correction flag  $\Xi$  and are not used at all during the sequential learning phase.

Other than using a slightly modified codebook, the microblock DF operation for the  $(\ell^* + 1)$ -th hop (performed by the bottleneck receiver) mirrors the operation of the pre-bottleneck hops: The  $k$ -th microblock is mapped to an estimate of the microblock message  $\hat{i}_k^{(\ell^*)}$  using ML decoding and then used to generate the microblock codeword  $\hat{\mathbf{x}}_{k, \hat{i}_k^{(\ell^*)}}^{(\ell^*+1)} \in \hat{\mathbf{X}}_k^{(\ell^*+1)}$ , which is forwarded to the next node.

2) *The correction phase.* The description herein assumes that the sequential learning phase is properly terminated, i.e., there exists a unique  $\hat{m}$  satisfying (4.49). The operations under the forced termination scenario  $k = K_{\max}$  are inconsequential since they are lumped under the "error event". During the correction phase, the bottleneck receiver transmits up to  $K$  additional microblocks that aim to correct the first  $K$  microblocks it transmitted. For this, it must first compile the set of microblocks which were forwarded in error. Let  $\tilde{k}$  denote the microblock index for which the sequential learning phase ends and let  $\hat{m}$  denote the message which triggered the end of the sequential learning phase. Next, let  $\mathbf{i}^{[\hat{m}]}$  denote the corresponding sequence of microblock messages generated by  $\hat{m}$  for the first  $K$  microblocks, i.e.,

$$\mathbf{i}^{[\hat{m}]} = \left\{ i_1^{[\hat{m}]}, \dots, i_K^{[\hat{m}]} \right\}. \quad (4.50)$$

Since the bottleneck receiver observes the transmission over the  $\ell^*$ -th hop, we let  $\hat{\mathbf{i}}^{(\ell^*)}$  denote the sequence of microblock messages that were decoded (using only the inner code decoder of the  $\ell^*$ -th hop) and forwarded to the  $(\ell^* + 1)$ -th hop during the sequential learning phase, i.e.,

$$\hat{\mathbf{i}}^{(\ell^*)} = \left\{ \hat{i}_1^{(\ell^*)}, \dots, \hat{i}_K^{(\ell^*)} \right\}. \quad (4.51)$$

We call  $\hat{m}$  the *high-confidence estimate* since it is based on the sequential log-likelihood ratio test that jointly considers the inner and outer codes while (4.51) is of (relatively) low confidence due to the use of only the inner decoders.

Finally, denote the set of microblock messages forwarded *in error* as

$$\mathcal{K}_{\text{err}} = \left\{ k \in [1, K] : \hat{i}_k^{(\ell^*)} \neq i_k^{[m]} \right\}. \quad (4.52)$$

The transmission of the correction microblocks during the correction phase starts immediately after the  $\tilde{k}$ -th microblock is received and  $\mathcal{K}_{\text{err}}$  is computed. As described earlier, to correct a single microblock, the bottleneck receiver needs to send (i) a correction flag  $\Xi = |\mathcal{K}_{\text{err}}|$  that instructs the downstream nodes to expect  $\mathcal{K}_{\text{err}}$  additional correction microblocks; (ii) the new/corrected content of the microblock message; and (iii) the microblock index which is being corrected. As a result, after  $\tilde{k}$  microblock transmissions from  $\mathring{\mathbf{X}}_k^{\ell^*+1} : k \in [1, \tilde{k}]$ , the next microblock  $k = \tilde{k} + 1$  will still be chosen from the inner codebook  $\mathring{\mathbf{X}}_{\tilde{k}+1}^{(\ell^*+1)}$ , but we transmit one of the additional  $(K + 1)$  codewords of  $\mathring{\mathbf{X}}_{\tilde{k}+1}^{(\ell^*+1)}$  that signifies the value of  $\Xi \in [0, K]$ . Then, for purposes (ii) and (iii), we use the different codebooks  $\tilde{\mathbf{X}}_k^{(\ell+1)}$ , which have  $K$  times the number of codewords of the regular codebooks. Specifically, after transmitting the  $\Xi$  value, the bottleneck receiver picks one microblock index  $j$  from  $\mathcal{K}_{\text{err}}$  that has not been "corrected" and maps the index/payload pair  $(j, i_j^{[m]})$  to the corresponding codeword from  $\tilde{\mathbf{X}}_k^{(\ell+1)}$ . It then repeats this process until all  $k \in \mathcal{K}_{\text{err}}$  have been "corrected". Effectively, the correction phase consumes  $\Delta + |\mathcal{K}_{\text{err}}|\tilde{\Delta}$  slots, where  $\Delta$  of which are used to transmit the flag  $\Xi$  and  $|\mathcal{K}_{\text{err}}|\tilde{\Delta}$  of which are used to correct all microblocks in  $\mathcal{K}_{\text{err}}$ .

Once all microblocks in  $\mathcal{K}_{\text{err}}$  have been corrected, the bottleneck receiver remains idle. This mechanism is illustrated in Fig. 4.5, where two correction microblocks are displayed.

**Relaying at the post-bottleneck nodes.** The operation of the post-bottleneck nodes mostly mirrors that of the pre-bottleneck nodes. That is, DF is performed on all microblocks. The main difference is, as described, how to coordinate the use of the two codebooks of different sizes. Specifically, the post-bottleneck receivers initially perform ML decoding for every  $\Delta$  received channel symbols based on the length- $\Delta$  inner decoders of  $\mathring{\mathbf{X}}_k^{(\ell)}$  for each  $k$ . However, once its ML decoder outputs a

correction flag  $\Xi \in \{0, \dots, K\}$ , say it decodes  $\Xi = j$ , the post-bottleneck receivers then anticipate the next  $j$  microblocks to be based on the length- $\tilde{\Delta}$  inner codebooks  $\tilde{\mathbf{X}}_k^{(\ell)}$  and thus use the inner codebooks  $\tilde{\mathbf{X}}_k^{(\ell)}$  to perform DF (instead of the pre- $\Xi$  codebooks  $\mathring{\mathbf{X}}_k^{(\ell)}$ ).

**Decoding at the destination.** The destination operates in the same way as the other post-bottleneck receivers with some minor additions. That is, after its ML decoder  $\mathring{\mathbf{X}}_k^{(L)}$  outputs a correction flag  $\Xi = j$ , the destination starts anticipating the final  $j$  correction microblocks and will decode them based on the length- $\tilde{\Delta}$  inner decoder  $\tilde{\mathbf{X}}_k^{(L)}$ . After the final  $j$  correction microblocks, it sends the stop-feedback signal through the feedback channel.

To decode the original message  $m$ , the destination takes the very first  $K$  microblock messages it previously decoded using the inner code ML decoders, and then replaces a subset of those estimates by the final  $j$  correction microblocks received after receiving  $\Xi = j$ . Let  $\{\check{i}_k \in [1, e^{\Delta R_I}] : k \in [1, K]\}$  denote the resulting  $K$  microblock messages after correction/replacement of  $\Xi = j$  of them. The destination uses the SRPOC outer code  $\mathcal{S}$  to compute the *compatible set*

$$\hat{\mathcal{M}} \triangleq \left\{ m \in \mathcal{M} : \check{i}_k = i_k^{[m]} \right\}. \quad (4.53)$$

The destination then proceeds as follows. If  $\hat{\mathcal{M}}$  contains exactly one message  $\hat{m}$ , then output such  $\hat{m}$ . If  $\hat{\mathcal{M}}$  contains zero or multiple messages, then declare error. In either case, the destination stops the transmission of the entire network after the final  $j$  microblocks after decoding  $\Xi = j$ .

It is possible that the destination's ML decoder  $\mathring{\mathbf{X}}_k^{(L)}$  never outputs a correction flag even after receiving  $K_{\max}$  microblocks. In this case, the destination also sends out the one-time stop-feedback. Namely, the destination aborts the transmission of the entire network once the waiting time for the correction flag exceeds the upper limit  $K_{\max}$  (unit: microblocks).



#### 4.4.2 DAF Analysis

Recall that our proposed scheme is defined by the four deterministic parameters  $(K, \Delta, R, \alpha)$ . Analogous to the analysis in Section 4.3.4, the second coordinate  $\Delta \rightarrow \infty$  since we let  $T \rightarrow \infty$  in (4.9). The error exponent of our proposed scheme is thus determined by the tuple  $(K, R, \alpha)$  and we use the notation  $E_\Phi(K, R, \alpha)$ . We then have

**Lemma 4.4.1** *For any  $R < C = C_{\ell^*}$  that is sufficiently close to  $C$ , we have*

$$E_\Phi(K^*(R, \alpha), R, \alpha) = \frac{K^*(R, \alpha)}{K^*(R, \alpha) + L} \cdot \alpha(C_{\ell^*} - R), \quad (4.54)$$

where  $K^*(R, \alpha)$  is the largest integer  $K$  satisfying

$$\alpha(C_{\ell^*} - R) < \frac{\min_{\ell \neq \ell^*} E_{\text{rc}, \ell}(C_{\ell^*})}{K} \quad (4.55)$$

where  $E_{\text{rc}, \ell}(R)$  is the open-loop random coding error exponent (even though this lemma is analyzing a stop-feedback scheme).

For any fixed  $\alpha \in (0, 1)$ , when  $R \rightarrow C = C_{\ell^*}$ , we have  $K^*(R, \alpha) \rightarrow \infty$ . By Lemma 4.4.1 and the definition of the DAF in (4.15), the proposed scheme thus achieves  $\tilde{\Gamma} = \frac{1}{\alpha}$ . Finally, letting  $\alpha \rightarrow 1$  lets the proposed scheme attain DAF=1. The remainder of this section proves the above lemma.

#### Probability of Error

We first introduce some definitions. We denote the message sent by the source as  $m_0$ .

**Definition 4.4.1** *We say that the SRPOC outer code is reversible (with respect to  $m_0$ ) if there exists no other  $m \in \mathcal{M} \setminus \{m_0\}$  such that*

$$(i_1^{[m]}, i_2^{[m]}, \dots, i_K^{[m]}) = (i_1^{[m_0]}, i_2^{[m_0]}, \dots, i_K^{[m_0]}). \quad (4.56)$$

Namely, by observing the first  $K$  coded outer code messages  $(i_1^{[m_0]}, i_2^{[m_0]}, \dots, i_K^{[m_0]})$ , one can uniquely recover the original sent message  $m_0$ .

Recall that  $\tilde{k}$  is defined as the time then the sequential probability test ends (i.e., when (4.49) holds for some  $m$  or when we reach the time limit  $K_{\max}$ ). We can further formalize this discussion by the following definition.

**Definition 4.4.2** *Define  $\tilde{k}_m$  as the microblock index at which the LLR for a specific message  $m$  first crosses the threshold in (4.49), assuming we run the sequential probability test indefinitely without the maximum constraint. As a result, we can rewrite  $\tilde{k}$  as*

$$\tilde{k} \triangleq \min \left( K_{\max}, \min_{m \in \mathcal{M}} \tilde{k}_m \right). \quad (4.57)$$

We then note that the following conditions are sufficient for an error-free message transmission.

- C1: There are no microblock decoding errors at the pre-bottleneck relays.
- C2: The encoding function of the SRPOC  $\mathcal{S}$  is reversible. When both C1 and C2 hold, the bottleneck transmitter can correctly decode the message  $m_0$  after  $K$  microblocks, successfully subsume the role of the source, and continue to transmit the correct microblocks across the bottleneck hop.
- C3: The message decision at the end of the sequential learning phase at the bottleneck receiver is within the hard time limit (i.e.,  $\min_m \tilde{k}_m \leq K_{\max}$ ) and is correct (i.e.,  $\hat{m} = m$ ).
- C4: There is no microblock decoding error over all the post-bottleneck hops.

When C1 to C4 all hold, the destination can recover the first  $K$  microblock messages generated at the source correctly either during the sequential learning phase or during the correction phase. Note that since C2 ensures reversibility of the outer SRPOC, when C1 to C4 hold, the destination can recover the message correctly.

We can distill a set of useful error events from these observations. First, let  $N_{\text{pre}}$  and  $N_{\text{post}}$  denote the total number of microblock errors over all pre- and post-bottleneck hops, respectively. For example,

$$N_{\text{pre}} = \sum_{\ell=1}^{\ell^*-1} \sum_{k=1}^K \mathbf{1}_{\{\hat{i}_k^{(\ell-1)} \neq \hat{i}_k^{(\ell)}\}}(k), \quad (4.58)$$

where (i) the indicator function  $\mathbf{1}_{\mathcal{A}}(x)$  is defined as

$$\mathbf{1}_{\mathcal{A}}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{A} \\ 0 & \text{if } x \notin \mathcal{A} \end{cases} \quad (4.59)$$

(ii) we recall that  $\hat{i}_k^{(\ell-1)}$  is the estimate when applying the ML inner code decoder over the upstream hop (the  $(\ell-1)$ -th hop); and (iii) we slightly abuse the notation by setting  $\hat{i}_k^{(0)} = i_k^{[m_0]}$  as the  $k$ -th microblock message since the source node does not have an upstream hop. The expression for  $N_{\text{post}}$  is more complicated since (i) the active duration of the post-bottleneck hops is a random number, which is in contrast with the pre-bottleneck hops that are active for only  $K$  microblocks; and (ii) two different codebooks  $\mathring{\mathbf{X}}_k^{(\ell)}$  and  $\widetilde{\mathbf{X}}_k^{(\ell)}$  are used in the post-bottleneck hops. Since our proof does not rely on the exact form of  $N_{\text{post}}$ , we omit its closed-form expression.

We then define the following (error) events:

$$\mathcal{A}_1 = \{N_{\text{pre}} > 0\} \quad (4.60)$$

$$\mathcal{A}_2 = \{\text{the SRPOC is not reversible}\} \quad (4.61)$$

$$\mathcal{A}_3 = \left\{ \min_m \tilde{k}_m > K_{\text{max}} \right\} \quad (4.62)$$

$$\mathcal{A}_4 = \left\{ \text{the output of the sequential LLR test } \hat{m} \neq m_0 \right\} \quad (4.63)$$

$$\mathcal{A}_5 = \{N_{\text{post}} > 0\} \quad (4.64)$$

By mapping the above events to the previous discussion of sufficient conditions C1 to C4, the error probability of our scheme must satisfy

$$\begin{aligned}
\epsilon &\triangleq \Pr(\hat{m} \neq m_0) \\
&\leq \Pr\left(\bigcup_{i=1}^5 \mathcal{A}_i\right) \\
&\leq \Pr(\mathcal{A}_1) + \Pr(\mathcal{A}_2) + \Pr(\mathcal{A}_3|\mathcal{A}_1^c\mathcal{A}_2^c) \\
&\quad + \Pr(\mathcal{A}_4|\mathcal{A}_1^c\mathcal{A}_2^c\mathcal{A}_3^c) + \Pr(\mathcal{A}_5|\mathcal{A}_1^c\mathcal{A}_2^c), \tag{4.65}
\end{aligned}$$

for which we use the set equation  $\bigcup_{i=1}^5 \mathcal{A}_i = \mathcal{A}_1 \cup \mathcal{A}_2 \cup (\mathcal{A}_3 \setminus \mathcal{A}_1\mathcal{A}_2) \cup (\mathcal{A}_4 \setminus \mathcal{A}_1\mathcal{A}_2\mathcal{A}_3) \cup (\mathcal{A}_5 \setminus \mathcal{A}_1\mathcal{A}_2)$ .

To continue the analysis, we must bound each of the five terms in (4.65) separately. We begin with  $\mathcal{A}_1$ . Using the individual random coding error exponents for the pre-bottleneck hops, we can bound

$$\Pr(\mathcal{A}_1) \leq K \sum_{\ell=1}^{\ell^*-1} e^{-\Delta E_{rc,\ell}(R_I)}. \tag{4.66}$$

From the code construction of the SRPOC  $\mathcal{S}$ , we know that the probability of the event  $\mathcal{A}_2$  satisfies the following union bound.

$$\Pr(\mathcal{A}_2) \leq \sum_{m' \neq m} \prod_{k=1}^K \Pr\left(i_k^{[m']} = i_k^{[m]}\right) \tag{4.67}$$

$$= \sum_{m' \neq m} \prod_{k=1}^K \left( \frac{\frac{|\mathcal{M}|}{e^{\Delta R_I}} - 1}{|\mathcal{M}| - 1} \right) \tag{4.68}$$

$$= (|\mathcal{M}| - 1) \left( \frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1} \right)^K \tag{4.69}$$

$$\leq (|\mathcal{M}| - 1)e^{-K\Delta R_I} \tag{4.70}$$

$$\leq e^{-K\Delta(R_I - R)}, \tag{4.71}$$

where (4.68) is due to the uniform random choices of the permutation  $\pi_k$  of  $\mathcal{S}$ .

To continue, the following lemma provides a bound on the third term of (4.65).

**Lemma 4.4.2** *The third term of (4.65) satisfies*

$$\Pr(\mathcal{A}_3 | \mathcal{A}_1^c \mathcal{A}_2^c) \leq (1 + \varsigma_1(K)) e^{-K\Delta\alpha(C_{\ell^*} - R)}, \quad (4.72)$$

where the term  $\varsigma_1(K) \rightarrow 0$  as  $K \rightarrow \infty$ .

For the proof refer to Appendix C.2.

For the fourth term of (4.65), we note that at the moment<sup>15</sup> that the sequential decision rule from (4.49) reaches the threshold  $\ln 1/\eta$ , the conditional error probability is upper bounded by  $\eta$ . Since we chose the target error probability  $\eta$  by (4.42), we have

$$\Pr(\mathcal{A}_4 | \mathcal{A}_1^c \mathcal{A}_2^c \mathcal{A}_3^c) \leq \eta = e^{-K\Delta\alpha(C_{\ell^*} - R)}. \quad (4.73)$$

Next, the following lemma provides a bound on the final term of (4.65).

**Lemma 4.4.3** *The fifth term of (4.65) satisfies*

$$\Pr(\mathcal{A}_5 | \mathcal{A}_1^c \mathcal{A}_2^c) \leq K(1 + \varsigma_2(K)) \sum_{\ell=\ell^*+1}^L e^{-\Delta E_{\text{rc},\ell}(R_I)}, \quad (4.74)$$

where the term  $\varsigma_2(K) \rightarrow 0$  as  $K \rightarrow \infty$ .

For the proof refer to Appendix C.5.

We now make the following observations.

1. Due to (4.55) and (4.66), we have

$$\Pr(\mathcal{A}_1) \leq e^{-K\Delta\alpha(C_{\ell^*} - R)} \quad (4.75)$$

---

<sup>15</sup>One needs to carefully argue whether there is a non-zero probability that the sequential value in (4.49) never reaches the threshold, i.e.,  $\Pr(\min_m \tilde{k}_m = \infty) > 0$ . That is why we upper bound  $\Pr(\mathcal{A}_3 | \mathcal{A}_1^c \mathcal{A}_2^c)$  separately from  $\Pr(\mathcal{A}_4 | \mathcal{A}_1^c \mathcal{A}_2^c \mathcal{A}_3^c)$ .

and

$$\Pr(\mathcal{A}_5 | \mathcal{A}_1^c \mathcal{A}_2^c) \leq e^{-K\Delta\alpha(C_{\ell^*} - R)} \quad (4.76)$$

for sufficiently large  $\Delta$ .

2. Due to (4.40) and (4.71), we have

$$\Pr(\mathcal{A}_2) \leq e^{-K\Delta\alpha(C_{\ell^*} - R)} \quad (4.77)$$

for sufficiently large  $\Delta$ .

In other words, with the selected parameters, all of the terms in (4.65) decay exponentially with the rate  $K\Delta\alpha(C_{\ell^*} - R)$ . Combining all terms, we can thus write

$$\epsilon \leq (5 + \varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*} - R)} \quad (4.78)$$

for sufficiently large  $\Delta$ .

### End-to-end Latency

We next must derive a bound on the *expected* end-to-end delay  $T$  for one message. For this, we will first bound the expectation of the *total duration of message reception* at the destination, denoted by  $D$ . This time, measured from the time slot in which the first microblock arrives until the end of the last microblock, does not include the amount of time it takes to relay all microblocks from the source to the destination. As a result,  $D$  and  $T$  are directly related by  $T = D + (L-1)\Delta$ , where the incremental term accounts for the time it takes for the first microblock to be relayed to the destination. We then have

**Lemma 4.4.4** *The expected total duration of message reception satisfies*

$$E\{D\} \leq K(1 + \varsigma_3(K, \Delta))\Delta \quad (4.79)$$

where  $\varsigma_3(K, \Delta) \rightarrow 0$  if we let  $\Delta \rightarrow \infty$  and then  $K \rightarrow \infty$  in this order.

For the proof refer to Appendix C.6.

Using this result, we can thus write

$$E\{T\} < [K(1 + \varsigma_3(K, \Delta)) + (L - 1)] \Delta. \quad (4.80)$$

### Error Exponent and DAF

Using the results of the above analysis, we can bound the error exponent as

$$\begin{aligned} E_\Phi(R) &= \lim_{\Delta \rightarrow \infty} \frac{-\ln(\epsilon)}{E\{T\}} \\ &\geq \lim_{\Delta \rightarrow \infty} \frac{-\ln[(5 + \varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*} - R)}]}{[K(1 + \varsigma_3(K, \Delta)) + (L - 1)] \Delta} \\ &= \frac{K \cdot \alpha(C_{\ell^*} - R)}{K(1 + \lim_{\Delta \rightarrow \infty} \varsigma_3(K, \Delta)) + (L - 1)}. \end{aligned} \quad (4.81)$$

Since  $\lim_{K \rightarrow \infty} \lim_{\Delta \rightarrow \infty} \varsigma_3(K, \Delta) = 0$  and since  $K^*(R) \rightarrow \infty$  when  $R \rightarrow C$ , by choosing a sufficiently large  $R \rightarrow C$  satisfying  $\lim_{\Delta \rightarrow \infty} \varsigma_3(K, \Delta) \leq 1$ , we have proven the statement of Lemma 4.4.1.

## 4.5 Summary, Discussion, & Conclusion

In the beginning of this article, we made the case for approaching the analysis of multi-hop line networks from a low-latency perspective. On the one hand, this idea was motivated by the increased densification of modern communication networks and the resulting increased number of hops a message must traverse on its way from the originating source to its destination. On the other hand, as the demand for real-time and mission-critical services grows, designers of multi-hop communication systems find themselves under increased pressure of lowering the end-to-end latency of their networks.

One of the main goals of this work was to introduce a new metric for fair comparison between the asymptotic delay performance of different transmission schemes. To

this end, we introduced and discussed the motivation behind the delay amplification factor in Section 4.2. This simple metric captures the error exponent penalty a system suffers when communicating over  $L$  hops as compared to communicating over the bottleneck hop, which provides a firm lower bound on the latency. We then showed how the de-facto standard transmission scheme, consisting of block codes and decoding and re-encoding at each relay, is strictly sub-optimal due to its linear growth in delay. This came at no surprise, since the DF strategy was not designed with latency in mind.

We then proceeded to introduce two novel transmission schemes designed with DAF optimality in mind. For the open-loop feedback free setting, we showed that by judiciously combining (i) the concept of block Markov coding for relay channels, (ii) concatenated coding for point-to-point channels, and (iii) the ML joint inner/outer decoder, we can achieve the optimal DAF in bottleneck-terminated DMC line networks. This channel model is applicable to real-world scenarios in which the last hop is of particular low quality, for example, line networks composed of microwave links delivering a real-time service to a cellular device limited by a low-quality connection to and from its base station.

For the one-time stop feedback setting, the scheme presented in Section 4.4 lifted this restriction on the bottleneck position and can provably achieve  $\text{DAF}=1$  for any line network. The design judiciously combines several new components: (i) block Markov coding for microblocks; (ii) use of a rateless code as the outer code; (iii) sequential probability ratio test at the bottleneck receiver based on the ML joint inner/outer decoder; (iv) the bottleneck receiver sending correction packets; (v) the carefully chosen inner and outer code rates and other system parameters to balance the error probabilities within the line network; (vi) carefully incorporating the network-layer concepts of the *timer*  $K_{\max}$  and the correction flag  $\Xi$  into the physical-layer code design. Since the one-time stop feedback is commonly used in real-world systems, the proposed scheme could have broad impact on low-latency communication systems.



Finally, the results of this paper are intended to provide a theoretical background and insight for designers of low-latency communication networks. In this regard, one of the key conclusions gained from this work is the fact that carefully designed transmission schemes utilizing microblocks and concatenated codes can be used to comply to the stringent low-latency requirements of modern networks in ways that simple, more intuitive schemes like Decode-&-Forward cannot.

## 5. CONCLUSION

This dissertation presented three topics related to the areas of beamforming and channel coding of multi-node wireless networks. In chapter 2 we presented a set of novel beam alignment techniques for beamformed MIMO channels. We provided theoretical insights about the convergence of our algorithms and illustrated their performance using numerical simulations. In chapter 3 we presented a novel channel and Doppler shift estimation algorithm for aerial communication links. Here, we again developed a theoretical foundation before demonstrating the viability of our approach using numerical simulations. Finally, in chapter 4 we presented a novel analysis framework and transmission schemes for low-latency communication over multi-hop relay channels. This line of work will aid in enabling future low-latency real-time control and sensing applications.

Innovations like the transmission schemes and signal processing techniques discussed in this dissertation will continue to push the technological boundaries enabling network operators worldwide to deliver critical services and infrastructure to their subscribers. Our networks are thus in a continuous cycle of evolution. As our technological capabilities improve, the demand for wireless systems increases. As this demand increases, engineers will continue to improve our networks' underlying technologies.

Of course, the topics of this dissertation represent merely a tiny sliver of the ongoing innovation in this vast field. In this sense, our work is truly "standing on the shoulder of giants". In this spirit, we would like to leave this work with the reminder that wireless is and will be for the foreseeable future the most popular and most accessible means of communication and connection. It is our belief that connecting people across the globe provides a net benefit to the entirety of humanity. As re-

searchers and engineers, we thus must continue to push forward and drive progress and innovation in this fundamental technology.

## APPENDICES

## A. PROOFS FOR CHAPTER 2

### A.1 Proof of Theorem 2.3.1

The derivation of (2.16) mirrors (2.15) and thus it suffices to establish (2.15). Transposing (2.6) at time  $k - 1$ , we get

$$\mathbf{Y}_{e,k-1}^\top = \sqrt{\rho_e} \mathbf{Z}_{k-1}^* \mathbf{H} + \mathbf{N}_{e,k-1}^\top. \quad (\text{A.1})$$

Since the columns of  $\mathbf{H}$  are i.i.d. complex Gaussian random vectors, we focus on the first column without loss in generality. Let this first column of  $\mathbf{H}$  be denoted as  $\mathbf{h}_1$  and let its estimator be  $\hat{\mathbf{h}}_1$ . With  $\tilde{\mathbf{y}}$  denoting the first column of  $\mathbf{Y}_{e,k-1}^\top$ , we have

$$\tilde{\mathbf{y}} = \sqrt{\rho_e} \mathbf{Z}_{k-1}^* \mathbf{h}_1 + \tilde{\mathbf{n}}, \quad (\text{A.2})$$

where  $\tilde{\mathbf{n}}$  is the first column of  $\mathbf{N}_{e,k-1}^\top$  with i.i.d. complex Gaussian entries. The estimator of  $\mathbf{h}_1$  is given as

$$\hat{\mathbf{h}}_1 = \frac{(\mathbf{Z}_{k-1}^*)^\dagger \tilde{\mathbf{y}}}{\sqrt{\rho_e}} \quad (\text{A.3})$$

with error covariance matrix  $\mathbf{C}_{e,k}$ , defined as  $\mathbf{C}_{e,k} \triangleq E\left\{\left(\mathbf{h}_1 - \hat{\mathbf{h}}_1\right)\left(\mathbf{h}_1 - \hat{\mathbf{h}}_1\right)^*\right\}$ . It can be seen that

$$\mathbf{h}_1 - \hat{\mathbf{h}}_1 = \mathbf{h}_1 - \frac{(\mathbf{Z}_{k-1}^*)^\dagger}{\sqrt{\rho_e}} (\sqrt{\rho_e} \mathbf{Z}_{k-1}^* \mathbf{h}_1 + \tilde{\mathbf{n}}) \quad (\text{A.4})$$

$$= -\frac{(\mathbf{Z}_{k-1}^*)^\dagger}{\sqrt{\rho_e}} \tilde{\mathbf{n}} \quad (\text{A.5})$$

and

$$\mathbf{C}_{e,k} = \frac{(\mathbf{Z}_{k-1}^*)^\dagger}{\sqrt{\rho_e}} E\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^*\} \frac{(\mathbf{Z}_{k-1})^\dagger}{\sqrt{\rho_e}} \quad (\text{A.6})$$

$$= \frac{1}{\rho_e} (\mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^*)^{-1} \mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^* (\mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^*)^{-1} \quad (\text{A.7})$$

$$= \frac{1}{\rho_e} (\mathbf{Z}_{k-1} \mathbf{Z}_{k-1}^*)^{-1}. \quad (\text{A.8})$$

Note that the above equation holds only under the i.i.d.  $\mathbf{H}$  assumption and for  $k \geq M_r$ . ■

## A.2 Proof of Theorem 2.3.2

Without loss in generality, we can assume that  $\rho_e = \rho_o = 1$ . From (2.10), we have

$$\hat{\mathbf{H}}_{o,k}^{\text{Batch}} = \mathbf{Y}_{o,k} \mathbf{F}_k^* \mathbf{C}_{o,k} \quad (\text{A.9})$$

$$= \mathbf{Y}_{o,k} \mathbf{F}_k^* (\mathbf{F}_k \mathbf{F}_k^*)^{-1} \quad (\text{A.10})$$

$$= \begin{bmatrix} \mathbf{Y}_{o,k-1} & \mathbf{y}_o[k] \end{bmatrix} \begin{bmatrix} \mathbf{F}_{k-1}^* \\ \mathbf{f}^*[k] \end{bmatrix} \left( \begin{bmatrix} \mathbf{F}_{k-1} & \mathbf{f}[k] \end{bmatrix} \begin{bmatrix} \mathbf{F}_{k-1}^* \\ \mathbf{f}^*[k] \end{bmatrix} \right)^{-1} \quad (\text{A.11})$$

$$= \begin{bmatrix} \mathbf{Y}_{o,k-1} \mathbf{F}_{k-1}^* + \mathbf{y}_o[k] \mathbf{f}^*[k] \end{bmatrix} (\mathbf{F}_{k-1} \mathbf{F}_{k-1}^* + \mathbf{f}[k] \mathbf{f}^*[k])^{-1}. \quad (\text{A.12})$$

Substituting (2.16) and applying the Woodbury matrix identity [49] to the second term, we get

$$\mathbf{C}_{o,k} = \mathbf{C}_{o,k-1} - \frac{\mathbf{C}_{o,k-1} \mathbf{f}[k] \mathbf{f}^*[k] \mathbf{C}_{o,k-1}}{1 + \mathbf{f}^*[k] \mathbf{C}_{o,k-1} \mathbf{f}[k]}. \quad (\text{A.13})$$

We now let

$$\mathbf{K}_{o,k} = \frac{\mathbf{f}^*[k] \mathbf{C}_{o,k-1}}{1 + \mathbf{f}^*[k] \mathbf{C}_{o,k-1} \mathbf{f}[k]} \quad (\text{A.14})$$

and write

$$\begin{aligned}\widehat{\mathbf{H}}_{o,k}^{\text{Batch}} &= \mathbf{Y}_{o,k-1} \mathbf{F}_{k-1}^* \mathbf{C}_{o,k-1} - \mathbf{Y}_{o,k-1} \mathbf{F}_{k-1}^* \mathbf{C}_{o,k-1} \mathbf{f}[k] \mathbf{K}_{o,k} \\ &\quad + \mathbf{y}_o[k] \mathbf{f}^*[k] \mathbf{C}_{o,k-1} - \mathbf{y}_o[k] \mathbf{f}^*[k] \mathbf{C}_{o,k-1} \mathbf{f}[k] \mathbf{K}_{o,k}.\end{aligned}\tag{A.15}$$

Now, since

$$\mathbf{y}_o[k] \mathbf{f}^*[k] \mathbf{C}_{o,k-1} = \mathbf{y}_o[k] (1 + \mathbf{f}^*[k] \mathbf{C}_{o,k-1} \mathbf{f}[k]) \mathbf{K}_{o,k},\tag{A.16}$$

we get

$$\widehat{\mathbf{H}}_{o,k}^{\text{Batch}} = \widehat{\mathbf{H}}_{o,k-1} (\mathbf{I} - \mathbf{f}[k] \mathbf{K}_{o,k}) + \mathbf{y}_o[k] \mathbf{K}_{o,k}\tag{A.17}$$

$$= \widehat{\mathbf{H}}_{o,k-1} + \left( \mathbf{y}_o[k] - \widehat{\mathbf{H}}_{o,k-1} \mathbf{f}[k] \right) \mathbf{K}_{o,k}\tag{A.18}$$

$$= \widehat{\mathbf{H}}_{o,k}^{\text{Seq}}.\tag{A.19}$$

■

### A.3 Proof of Theorem 2.3.3

Along the same lines of the proof of Theorem 2.3.2, let us assume that  $\rho_e = \rho_o = 1$ . Suppose that node 2 has access to  $M_t$  previous observations at time slot  $k = 0$ , indexed from  $k = -(M_t - 1)$  to  $k = 0$ . Using this data, node 2 could thus compute the batch estimate at time slot  $k = 0$ , given as

$$\widehat{\mathbf{H}}_{o,0}^{\text{Batch}} = \mathbf{Y}_{o,0} \mathbf{F}_0^\dagger = \mathbf{Y}_{o,0} \mathbf{F}_0^* (\mathbf{F}_0 \mathbf{F}_0^*)^{-1},\tag{A.20}$$

where

$$\mathbf{F}_0 = \begin{bmatrix} \mathbf{f}[-(M_t - 1)] & \mathbf{f}[-M_t] & \dots & \mathbf{f}[0] \end{bmatrix}\tag{A.21}$$

and

$$\mathbf{Y}_{o,0} = \begin{bmatrix} \mathbf{y}_o[-(M_t - 1)] & \mathbf{y}_o[-M_t] & \dots & \mathbf{y}_o[0] \end{bmatrix}. \quad (\text{A.22})$$

Using Theorem 2.3.1, we note that the covariance matrix of each column of this estimated matrix is given as

$$\mathbf{C}_{o,0} = (\mathbf{F}_0 \mathbf{F}_0^*)^{-1}. \quad (\text{A.23})$$

Applying the result of Theorem 2.3.2, we note that for any  $k > 0$ , a sequential least-squares estimator would be identical to the batch estimator using all of the data from  $k = -(M_t - 1)$  up to  $k$ . We can thus write

$$\hat{\mathbf{H}}_{o,k}^{\text{Seq}} = \hat{\mathbf{H}}_{o,k}^{\text{Batch}} = \mathbf{Y}_{o,k} \mathbf{F}_k^\dagger \quad (\text{A.24})$$

$$= \left( \sum_{n=-(M_t-1)}^k \mathbf{y}_o[n] \mathbf{f}^*[n] \right) \left( \sum_{n=-(M_t-1)}^k \mathbf{f}[n] \mathbf{f}^*[n] \right)^{-1}, \quad (\text{A.25})$$

where we have rewritten  $\hat{\mathbf{H}}_{o,k}^{\text{Batch}}$  in terms of individual vector outer products. After separating the hypothetical data from  $k = -(M_t - 1)$  to  $k = 0$  from the data starting at  $k = 1$ , we have for the sequential estimator

$$\begin{aligned} \hat{\mathbf{H}}_{o,k}^{\text{Seq}} &= \left( \sum_{n=-(M_t-1)}^0 \mathbf{y}_o[n] \mathbf{f}^*[n] + \sum_{n=1}^k \mathbf{y}_o[n] \mathbf{f}^*[n] \right) \\ &\quad \cdot \left( \sum_{n=-(M_t-1)}^0 \mathbf{f}[n] \mathbf{f}^*[n] + \sum_{n=1}^k \mathbf{f}[n] \mathbf{f}^*[n] \right)^{-1} \end{aligned} \quad (\text{A.26})$$

$$= (\mathbf{Y}_{o,0} \mathbf{F}_0^* + \mathbf{Y}_{o,k} \mathbf{F}_k^*) (\mathbf{F}_0 \mathbf{F}_0^* + \mathbf{F}_k \mathbf{F}_k^*)^{-1} \quad (\text{A.27})$$

$$= \left( \hat{\mathbf{H}}_{o,0} \mathbf{C}_{o,0}^{-1} + \mathbf{Y}_{o,k} \mathbf{F}_k^* \right) (\mathbf{C}_{o,0}^{-1} + \mathbf{F}_k \mathbf{F}_k^*)^{-1}. \quad (\text{A.28})$$



Upon further inspection of (A.26), we observe that for any  $k > M_t$ , the product  $\mathbf{F}_k \mathbf{F}_k^*$  is invertible and we can let  $\mathbf{C}_{o,0}^{-1}$  arbitrarily approach the matrix of all zeros. This can be accomplished by choosing  $\mathbf{C}_{o,0} = \alpha \mathbf{I}$ . If  $\mathbf{C}_{o,0}$  is indeed chosen this way, (A.26) loses its dependence on the previous data from the supposition and we can start the sequential iteration at  $k = 1$ . For sufficiently large  $\alpha$ , the sequential estimator will approach the batch estimator for any  $k > M_t$ . ■

#### A.4 Proof of Lemma 2.4.1

Under Hypothesis 3, the state transition matrix for a real, diagonal channel matrix  $\mathbf{H}$  is given as

$$\mathbf{S}_{k-1} = \begin{bmatrix} \mathbf{I} & \sqrt{\rho} \beta_{k-1} \cdot \mathbf{H} \\ \sqrt{\rho} \alpha_{k-1} \cdot \mathbf{H} & \mathbf{I} \end{bmatrix} \quad (\text{A.29})$$

since  $\mathbf{H} = \mathbf{H}^* = \text{diag}([h_1, \dots, h_M])$ . Note that the size of  $\mathbf{S}_{k-1}$  is  $2M \times 2M$ . The characteristic equation of  $\mathbf{S}_{k-1}$  is given as

$$\chi(\mathbf{S}_{k-1}, \lambda) = \det(\mathbf{S}_{k-1} - \lambda \mathbf{I}) \quad (\text{A.30})$$

$$= \det \left( \begin{bmatrix} \mathbf{I} - \lambda \mathbf{I} & \sqrt{\rho} \beta_{k-1} \cdot \mathbf{H} \\ \sqrt{\rho} \alpha_{k-1} \cdot \mathbf{H} & \mathbf{I} - \lambda \mathbf{I} \end{bmatrix} \right). \quad (\text{A.31})$$

By using the Schur complement lemma [49], this equation can be written as

$$\begin{aligned} \chi(\mathbf{S}_{k-1}, \lambda) &= \det(\mathbf{I} - \lambda \mathbf{I}) \cdot \\ &\det(\mathbf{I} - \lambda \mathbf{I} - \rho \alpha_{k-1} \beta_{k-1} \cdot \mathbf{H} (\mathbf{I} - \lambda \mathbf{I})^{-1} \mathbf{H}). \end{aligned} \quad (\text{A.32})$$

Since all of the matrices involved are diagonal, we can write the determinants as the product of the diagonal elements, resulting in

$$\chi(\mathbf{S}_{k-1}, \lambda) = (1 - \lambda)^{2M} \prod_{i=1}^{2M} \left\{ (1 - \lambda) - \frac{\rho \alpha_{k-1} \beta_{k-1} \cdot h_i^2}{1 - \lambda} \right\} \quad (\text{A.33})$$

$$= \prod_{i=1}^{2M} \{ (1 - \lambda)^2 - \rho \alpha_{k-1} \beta_{k-1} \cdot h_i^2 \}, \quad (\text{A.34})$$

which has  $2M$  roots (denoted as  $\lambda_1, \dots, \lambda_{2M}$ ) of the form  $1 \pm \sqrt{\rho \alpha_{k-1} \beta_{k-1}} \cdot h_i$  for  $i = 1, \dots, M$ . We can thus write the eigenvalue matrix  $\mathbf{\Lambda}_{k-1}$  as

$$\mathbf{\Lambda}_{k-1} = \text{diag} \left( \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{2M} \end{bmatrix} \right) \quad (\text{A.35})$$

$$= \text{diag} \left( \begin{bmatrix} 1 + \sqrt{\rho \alpha_{k-1} \beta_{k-1}} \cdot h_1 \\ \vdots \\ 1 + \sqrt{\rho \alpha_{k-1} \beta_{k-1}} \cdot h_M \\ 1 - \sqrt{\rho \alpha_{k-1} \beta_{k-1}} \cdot h_1 \\ \vdots \\ 1 - \sqrt{\rho \alpha_{k-1} \beta_{k-1}} \cdot h_M \end{bmatrix} \right). \quad (\text{A.36})$$

Solving for the  $2M$  eigenvectors (i.e. solving  $\mathbf{S}_{k-1} \mathbf{u}_i = \lambda_i \mathbf{u}_i$  for  $i = 1, \dots, 2M$ ) and normalizing each column to unit-norm finally results in the following eigenvector matrix:

$$\mathbf{U}_{k-1} = \begin{bmatrix} \sqrt{\frac{\beta_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} & \sqrt{\frac{\beta_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} \\ \sqrt{\frac{\alpha_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} & -\sqrt{\frac{\alpha_{k-1}}{\alpha_{k-1} + \beta_{k-1}}} \mathbf{I} \end{bmatrix}. \quad (\text{A.37})$$

Note that  $\mathbf{U}_{k-1}$  is not unitary in general and  $\mathbf{S}_{k-1}$  can be written as  $\mathbf{S}_{k-1} = \mathbf{U}_{k-1} \cdot \mathbf{\Lambda}_{k-1} \cdot (\mathbf{U}_{k-1})^{-1}$ . ■

### A.5 Proof of Lemma 2.4.2

Let  $\mathbf{H} = \{\mathbf{H}_{m,n}\}$ ,  $\mathbf{f}[i] = [f_{i,1}, \dots, f_{i,M}]^\top$  and  $\mathbf{z}[i] = [z_{i,1}, \dots, z_{i,M}]^\top$ . Also, let  $\mathbf{n}_e[i] = [\mathbf{n}_{e,1}[i], \dots, \mathbf{n}_{e,M}[i]]^\top$  and  $\mathbf{n}_o[i] = [\mathbf{n}_{o,1}[i], \dots, \mathbf{n}_{o,M}[i]]^\top$ . Now observe that

$$\frac{1}{\alpha_k^2} = \|\mathbf{s}_e[k]\|_2^2 = \left\| \sum_{i=0}^k \bar{\mathbf{y}}_e[i] \right\|_2^2 \quad (\text{A.38})$$

$$= \sum_{n=1}^M \left| \sqrt{\rho} \sum_{m=1}^M \bar{\mathbf{H}}_{m,n} \sum_{i=0}^k z_{i,m} + \sum_{i=0}^k \bar{\mathbf{n}}_{e,n}[i] \right|^2. \quad (\text{A.39})$$

Similarly, we have

$$\frac{1}{\beta_k^2} = \sum_{m=1}^M \left| \sqrt{\rho} \sum_{n=1}^M \mathbf{H}_{m,n} \sum_{i=0}^k f_{i,n} + \sum_{i=0}^k \mathbf{n}_{o,m}[i] \right|^2. \quad (\text{A.40})$$

From (A.38) and (A.40), we have the following simplifications.

$$\frac{1}{\rho \cdot \alpha_k^2} \stackrel{(a)}{\approx} \sum_{n=1}^M \left| \sum_{m=1}^M \bar{\mathbf{H}}_{m,n} \sum_{i=0}^k z_{i,m} \right|^2 \quad (\text{A.41})$$

$$\stackrel{(b)}{\approx} |\mathbf{C}_k|^2 \cdot \sum_{n=1}^M \left| \sum_{m=1}^M \bar{\mathbf{H}}_{m,n} \right|^2 \quad (\text{A.42})$$

$$\stackrel{(c)}{=} |\mathbf{C}_k|^2 \cdot \text{Tr}(\mathbf{H}\mathbf{H}^*) \quad (\text{A.43})$$

where (a), (b) and (c) follow from Hypotheses 1-3, respectively. Similarly, we have

$$\frac{1}{\rho \cdot \beta_k^2} \approx |\mathbf{C}_k|^2 \cdot \text{Tr}(\mathbf{H}\mathbf{H}^*). \quad (\text{A.44})$$

Thus, when Hypotheses 1-3 hold,  $\alpha_k \approx \beta_k$  as  $k$  increases. ■

### A.6 Proof of Theorem 2.4.1

When Hypotheses 1-3 hold, from Lemma 2.4.2, we have that  $\alpha_k \approx \beta_k$  and  $\mathbf{U}_{k-1} \approx \tilde{\mathbf{U}}$ . Thus, the state-space model in (2.35) can be written as

$$\begin{aligned} \mathbf{s}[k] = & \tilde{\mathbf{U}} \begin{bmatrix} \prod_{i=0}^{k-1} (\mathbf{I} + \sqrt{\rho} \alpha_i \mathbf{H}) & \mathbf{0} \\ \mathbf{0} & \prod_{i=0}^{k-1} (\mathbf{I} - \sqrt{\rho} \alpha_i \mathbf{H}) \end{bmatrix} \tilde{\mathbf{U}}^* \mathbf{s}[0] \\ & + \tilde{\mathbf{U}} \sum_{\ell=1}^k \begin{bmatrix} \prod_{j=\ell}^{k-1} (\mathbf{I} + \sqrt{\rho} \alpha_j \mathbf{H}) & \mathbf{0} \\ \mathbf{0} & \prod_{j=\ell}^{k-1} (\mathbf{I} - \sqrt{\rho} \alpha_j \mathbf{H}) \end{bmatrix} \tilde{\mathbf{U}}^* \mathbf{n}[\ell]. \end{aligned} \quad (\text{A.45})$$

Let us now consider the  $2M \times 2M$ -dimensional diagonal matrix  $\tilde{\mathbf{\Lambda}}_{k-1,0}$ .

$$\tilde{\mathbf{\Lambda}}_{k-1,0} = \begin{bmatrix} \prod_{i=0}^{k-1} (\mathbf{I} + \sqrt{\rho} \alpha_i \mathbf{H}) & \mathbf{0} \\ \mathbf{0} & \prod_{i=0}^{k-1} (\mathbf{I} - \sqrt{\rho} \alpha_i \mathbf{H}) \end{bmatrix}. \quad (\text{A.46})$$

From Hypothesis 3, since the diagonal entries of  $\mathbf{H}$  are arranged in non-increasing order and  $h_1 > h_2$ , we have

$$\frac{\prod_{i=0}^{k-1} (1 + \sqrt{\rho} \alpha_i h_1)}{\prod_{i=0}^{k-1} (1 + \sqrt{\rho} \alpha_i h_\ell)} \approx \left( \frac{h_1}{h_\ell} \right)^k \rightarrow \infty \text{ as } k \rightarrow \infty \quad (\text{A.47})$$

for  $\ell = 2, \dots, M$ . Similarly, we have

$$\frac{\prod_{i=0}^{k-1} (1 - \sqrt{\rho} \alpha_i h_1)}{\prod_{i=0}^{k-1} (1 - \sqrt{\rho} \alpha_i h_\ell)} \approx \left( \frac{h_1}{h_\ell} \right)^k \rightarrow \infty \text{ as } k \rightarrow \infty \quad (\text{A.48})$$

for  $\ell = 2, \dots, M$  and

$$\frac{\prod_{i=0}^{k-1} (1 + \sqrt{\rho} \alpha_i h_1)}{\prod_{i=0}^{k-1} (1 - \sqrt{\rho} \alpha_i h_1)} \approx (-1)^k \text{ as } k \rightarrow \infty. \quad (\text{A.49})$$

Thus, the diagonal entries of  $\tilde{\mathbf{\Lambda}}_{k-1,0}$  are dominated by (as  $k$  increases) the first entry, which is denoted as

$$\lambda_{1,0} \approx (\sqrt{\rho}h_1)^k \cdot \prod_{i=0}^k \alpha_i, \quad (\text{A.50})$$

and the  $(M+1)$ -th entry, which is denoted as

$$\lambda_{M+1,0} \approx (-\sqrt{\rho}h_1)^k \cdot \prod_{i=0}^k \alpha_i. \quad (\text{A.51})$$

Similarly, we can consider the diagonal matrices  $\tilde{\mathbf{\Lambda}}_{k-1,\ell}$  for  $\ell = 1, \dots, k-1$ :

$$\tilde{\mathbf{\Lambda}}_{k-1,\ell} = \begin{bmatrix} \prod_{i=\ell}^{k-1} (\mathbf{I} + \sqrt{\rho}\alpha_i \mathbf{H}) & \mathbf{0} \\ \mathbf{0} & \prod_{i=\ell}^{k-1} (\mathbf{I} - \sqrt{\rho}\alpha_i \mathbf{H}) \end{bmatrix}. \quad (\text{A.52})$$

Following the same logic as before, these matrices are also dominated by the first entry, which is denoted as

$$\lambda_{1,\ell} \approx (\sqrt{\rho}h_1)^{k-\ell} \cdot \prod_{i=\ell}^{k-1} \alpha_i, \quad (\text{A.53})$$

and the  $(M+1)$ th entry, which is denoted as

$$\lambda_{M+1,\ell} \approx (-\sqrt{\rho}h_1)^{k-\ell} \cdot \prod_{i=\ell}^{k-1} \alpha_i. \quad (\text{A.54})$$

With  $\mathbf{s}[0] = [s_1(0), \dots, s_{2M}(0)]^\top$ ,  $\mathbf{s}[k] = [s_1(k), \dots, s_{2M}(k)]^\top$ , and  $\mathbf{n}[\ell] = [n_1(\ell), \dots, n_{2M}(\ell)]^\top$ , it is straightforward to see that as  $k$  increases and for  $\rho \gg 1$ , we have

$$\frac{s_1(k)}{(\sqrt{\rho}h_1)^k \cdot \prod_{i=0}^{k-1} \alpha_i} \rightarrow \begin{cases} s_1(0) & \text{if } k \text{ is even} \\ s_{M+1}(0) & \text{if } k \text{ is odd} \end{cases} \quad (\text{A.55})$$

$$\frac{s_{M+1}(k)}{(\sqrt{\rho}h_1)^k \cdot \prod_{i=0}^{k-1} \alpha_i} \rightarrow \begin{cases} s_{M+1}(0) & \text{if } k \text{ is even} \\ s_1(0) & \text{if } k \text{ is odd.} \end{cases} \quad (\text{A.56})$$

And we also have  $\frac{s_\ell(k)}{(\sqrt{\rho}h_1)^k \cdot \prod_{i=0}^{k-1} \alpha_i} \rightarrow 0$  for all  $\ell \in \{2, \dots, M, M+2, \dots, 2M\}$ . Thus,  $\mathbf{s}[k] \rightarrow \mathbf{s}_{\text{opt}}$  as  $k$  increases. ■

## B. ALGORITHMS FOR CHAPTER 2

### B.1 SLS Estimator (Optimal)

---

**Algorithm B.1** SLS Estimator (Optimal)

---

Initialize  $\mathbf{f}[0]$  as a complex random unit-norm vector.

**for all**  $k = 1, \dots, k_{\max}$  **do**

Node 2 receives  $\mathbf{y}_o[k-1]$  as in (2.3) and gets  $\mathbf{f}[k-1]$  from Node 1

**if**  $k \leq \text{rank}[\mathbf{H}]$  **then**

Node 2 estimates  $\hat{\mathbf{H}}_{o,k-1}$  as in (2.12)

**else if**  $k > \text{rank}[\mathbf{H}]$  **then**

Node 2 estimates  $\hat{\mathbf{H}}_{o,k-1}$  as in (2.20)

**end if**

Node 2 computes  $\mathbf{z}[k-1]$  as in (2.14)

Node 1 receives  $\mathbf{y}_e[k-1]$  as in (2.4) and gets  $\mathbf{z}[k-1]$  from Node 2

**if**  $k \leq \text{rank}[\mathbf{H}]$  **then**

Node 1 estimates  $\hat{\mathbf{H}}_{e,k}$  as in (2.11)

**else if**  $k > \text{rank}[\mathbf{H}]$  **then**

Node 1 estimates  $\hat{\mathbf{H}}_{e,k}$  as in (2.17)

**end if**

Node 1 computes  $\mathbf{f}[k]$  as in (2.13)

**end for**

---

## B.2 SLS Estimator (Approximate)

---

**Algorithm B.2** SLS Estimator (Suboptimal)

---

Initialize  $\mathbf{f}[0]$  as a complex random unit-norm vector and obtain  $\hat{\mathbf{H}}_{o,0}$ ,  $\mathbf{z}[0]$ ,  $\hat{\mathbf{H}}_{e,1}$  and  $\mathbf{f}[1]$  as in (2.23)-(2.26).  
Initialize  $\mathbf{C}_{o,0} = \mathbf{C}_{e,1} = \alpha \mathbf{I}$  for an appropriate  $\alpha$ .  
**for all**  $k = 1, \dots, k_{\max}$  **do**  
    Node 2 receives  $\mathbf{y}_o[k]$  as in (2.3) and gets  $\mathbf{f}[k]$  from Node 1  
    Node 2 estimates  $\hat{\mathbf{H}}_{o,k}$  as in (2.20)  
    Node 2 computes  $\mathbf{z}[k]$  as in (2.14)  
  
    Node 1 receives  $\mathbf{y}_e[k]$  as in (2.4) and gets  $\mathbf{z}[k]$  from Node 2  
    Node 1 estimates  $\hat{\mathbf{H}}_{e,k+1}$  as in (2.17)  
    Node 1 computes  $\mathbf{f}[k+1]$  as in (2.13)  
**end for**

---

## B.3 Summed Power Method

---

**Algorithm B.3** Summed Power Method

---

Initialize  $\mathbf{f}[0]$  and  $\mathbf{z}[0]$  as complex random unit-norm vectors.  
**for all**  $k = 1, \dots, k_{\max}$  **do**  
    Node 2 receives  $\mathbf{y}_o[k-1]$  as in (2.3)  
    Node 2 computes  $\mathbf{z}[k]$  as in (2.29)  
  
    Node 1 receives  $\mathbf{y}_e[k-1]$  as in (2.4)  
    Node 1 computes  $\mathbf{f}[k]$  as in (2.27)  
**end for**

---



## B.4 LISP Method

---

**Algorithm B.4** Least-squares Initialized Summed Power Method

---

Initialize  $\mathbf{f}[0]$  as a complex random unit-norm vector and obtain  $\hat{\mathbf{H}}_{o,0}$ ,  $\mathbf{z}[0]$ ,  $\hat{\mathbf{H}}_{e,1}$  and  $\mathbf{f}[1]$  as in (2.23)-(2.26).  
Initialize  $\mathbf{C}_{o,0} = \mathbf{C}_{e,1} = \alpha \mathbf{I}$  for an appropriate  $\alpha$ .  
**for all**  $k = 1, \dots, k_{\max}$  **do**  
    **if**  $k \leq k_{\text{switch}}$  **then**  
        Node 2 receives  $\mathbf{y}_o[k]$  as in (2.3) and gets  $\mathbf{f}[k]$  from Node 1  
        Node 2 estimates  $\hat{\mathbf{H}}_{o,k}$  as in (2.20)  
        Node 2 computes  $\mathbf{z}[k]$  as in (2.14)  
    **else if**  $k > k_{\text{switch}}$  **then**  
        Node 2 receives  $\mathbf{y}_o[k-1]$  as in (2.3)  
        Node 2 computes  $\mathbf{z}[k]$  as in (2.29)  
    **end if**  
    **if**  $k \leq k_{\text{switch}} - 1$  **then**  
        Node 1 receives  $\mathbf{y}_e[k]$  as in (2.4) and gets  $\mathbf{z}[k]$  from Node 2  
        Node 1 estimates  $\hat{\mathbf{H}}_{e,k+1}$  as in (2.17)  
        Node 1 computes  $\mathbf{f}[k+1]$  as in (2.13)  
    **else if**  $k > k_{\text{switch}} - 1$  **then**  
        Node 1 receives  $\mathbf{y}_e[k]$  as in (2.4)  
        Node 1 computes  $\mathbf{f}[k+1]$  as in (2.27)  
    **end if**  
**end for**

---

## C. PROOFS FOR CHAPTER 4

### C.1 Proof of Lemma 4.2.1

The proof in this section is reproduced from [142]. We first prove  $\Gamma_{\text{DF}} \leq \sum_{\ell} \frac{C_{\ell^*}}{C_{\ell}}$ . To this end, we will construct explicitly the  $T_{\text{dur}}$ ,  $\tau_1$  to  $\tau_L$  and analyze its performance, assuming random coding for each hop.

For any arbitrarily given  $\delta > 0$ , we define  $\delta_{\ell^*} = \delta$  and for any  $\ell \neq \ell^*$ , we define

$$\delta_{\ell} \triangleq \inf \left\{ x > 0 : \frac{E_{\text{rc},\ell}(C_{\ell} - x)}{C_{\ell} - x} \geq 2 \cdot \frac{E_{\text{rc},\ell^*}(C_{\ell^*} - \delta)}{C_{\ell^*} - \delta} \right\} \quad (\text{C.1})$$

It is well known that any random coding error exponent must satisfy  $E_{\text{rc},\ell}(C_{\ell} - x) \rightarrow 0$  when  $x \rightarrow 0$  (including  $\ell = \ell^*$ ). Therefore, we have  $\delta_{\ell} \rightarrow 0$  for all  $\ell$  when  $\delta \rightarrow 0$ . We then choose  $t_{\ell}$ , the active duration of the  $\ell$ -th hop, as

$$t_{\ell} = \frac{\ln(|\mathcal{M}|)}{C_{\ell} - \delta_{\ell}}, \quad \forall \ell \in [1, L]. \quad (\text{C.2})$$

With random coding on each hop, the description of the DF scheme is thus complete, once we fix the values of  $|\mathcal{M}|$  and  $\delta$ . We now analyze the DAF value under this construction.

Since  $\delta_\ell \rightarrow 0$  if  $\delta \rightarrow 0$ , we have  $\ell^{**} = \operatorname{argmax}_\ell \{t_\ell\} = \ell^* = \operatorname{argmin}_\ell \{C_\ell\}$  when  $\delta$  is sufficiently small. By the discussion in Sec. 4.2.4,  $T_{\text{dur}} = t_{\ell^{**}} = t_{\ell^*}$ . The tuple  $(T, R, \epsilon)$  of this DF scheme thus satisfies

$$T = \sum_{\ell=1}^L t_\ell = \ln(|\mathcal{M}|) \cdot \sum_{\ell=1}^L \frac{1}{C_\ell - \delta_\ell} \quad (\text{C.3})$$

$$R = \frac{\ln(|\mathcal{M}|)}{T_{\text{dur}}} = \frac{\ln(|\mathcal{M}|)}{t_{\ell^*}} = C_{\ell^*} - \delta_{\ell^*} = C - \delta \quad (\text{C.4})$$

$$\epsilon \leq \sum_{\ell=1}^L \epsilon_\ell \leq \sum_{\ell=1}^L \exp \left\{ -t_\ell E_{\text{rc},\ell} \left( \frac{\ln(|\mathcal{M}|)}{t_\ell} \right) \right\} \quad (\text{C.5})$$

$$= \sum_{\ell=1}^L \exp \left\{ -\frac{\ln(|\mathcal{M}|)}{C_\ell - \delta_\ell} E_{\text{rc},\ell} (C_\ell - \delta_\ell) \right\} \quad (\text{C.6})$$

$$\leq \exp \left\{ -\frac{\ln(|\mathcal{M}|)}{C_{\ell^*} - \delta_{\ell^*}} E_{\text{rc},\ell^*} (C_{\ell^*} - \delta_{\ell^*}) \right\} \\ + (L-1) \exp \left\{ -\frac{\ln(|\mathcal{M}|)}{C_{\ell^*} - \delta_{\ell^*}} 2 \cdot E_{\text{rc},\ell^*} (C_{\ell^*} - \delta_{\ell^*}) \right\} \quad (\text{C.7})$$

where (C.3) follows from the construction of  $t_\ell$ ; (C.4) follows from  $t_{\ell^{**}} = t_{\ell^*}$ ; (C.5) follows from the random coding reliability function; and (C.7) follows from the construction of  $\delta_\ell$  in (C.1).

Letting  $|\mathcal{M}| \rightarrow \infty$ , by (4.9) we have

$$E_{\text{DF}}(R) = E_{\text{DF}}(C_{\ell^*} - \delta_{\ell^*}) \\ \geq \frac{\frac{1}{C_{\ell^*} - \delta_{\ell^*}}}{\sum_{\ell=1}^L \frac{1}{C_\ell - \delta_\ell}} \cdot E_{\text{rc},\ell^*} (C_{\ell^*} - \delta_{\ell^*}). \quad (\text{C.8})$$

Finally, letting  $\delta \rightarrow 0$ , by (4.10) and (C.8), we have

$$\Gamma_{\text{DF}} \leq \lim_{\delta \rightarrow 0} \frac{\sum_{\ell=1}^L \frac{1}{C_\ell - \delta_\ell}}{\frac{1}{C_{\ell^*} - \delta_{\ell^*}}} \\ = \sum_{\ell=1}^L \frac{C_{\ell^*}}{C_\ell}. \quad (\text{C.9})$$

Namely, the above DF construction attains  $\text{DAF} \leq \sum_{\ell=1}^L \frac{C_{\ell^*}}{C_\ell}$ .

We now prove that regardless how one chooses  $(|\mathcal{M}|, t_1, \dots, t_L)$ , the DAF is always lower bounded by  $\sum_{\ell=1}^L \frac{C_{\ell}^*}{C_{\ell}}$ . Suppose a given choice of  $(|\mathcal{M}|, t_1, \dots, t_L)$  attains the tuple  $(T, R, \epsilon)$ . Assuming sufficiently small  $\epsilon$ , we must have the following inequalities.

$$t_{\ell} \geq \frac{\ln(|\mathcal{M}|)}{C_{\ell}}, \quad \forall \ell \in [1, L] \quad (\text{C.10})$$

$$T = \sum_{\ell=1}^L t_{\ell} \geq \sum_{\ell=1}^L \frac{\ln(|\mathcal{M}|)}{C_{\ell}} \quad (\text{C.11})$$

$$R \triangleq \frac{\ln(|\mathcal{M}|)}{T_{\text{dur}}} = \frac{\ln(|\mathcal{M}|)}{\max_{\ell} t_{\ell}} \leq \frac{\ln(|\mathcal{M}|)}{t_{\ell^*}} \leq C \quad (\text{C.12})$$

$$\epsilon \geq \epsilon_{\ell^*} \geq \exp \left\{ -t_{\ell^*} E_{\text{sp}, \ell^*} \left( \frac{\ln(|\mathcal{M}|)}{t_{\ell^*}} \right) - o(t_{\ell^*}) \right\} \quad (\text{C.13})$$

$$\geq \exp \{ -t_{\ell^*} E_{\text{sp}, \ell^*}(R) - o(t_{\ell^*}) \} \quad (\text{C.14})$$

where we have (C.10) since in order to achieve small  $\epsilon$ , the coding rate per hop  $\frac{\ln(|\mathcal{M}|)}{t_{\ell}}$  must be less than the capacity  $C_{\ell}$ ; (C.11) follows from (C.10); (C.12) follows from the definition of  $T_{\text{dur}}$ ; (C.13) follows from the fact that the end-to-end error rate is lower bounded by the error rate of the  $\ell^*$ -th hop, which is lower bounded later by the sphere packing bound; and (C.14) follows from (C.12) and  $E_{\text{sp}, \ell^*}(\cdot)$  being non-increasing.

By (C.14) and (C.11), we have

$$\begin{aligned} \frac{-\ln(\epsilon)}{T} &\leq \frac{t_{\ell^*} E_{\text{rc}, \ell^*}(R) + o(t_{\ell^*})}{\sum_{\ell=1}^L \frac{\ln(|\mathcal{M}|)}{C_{\ell}}} \iff \\ \frac{E_{\text{rc}, \ell^*}(R)}{\frac{-\ln(\epsilon)}{T}} &\geq \frac{\sum_{\ell=1}^L \frac{\ln(|\mathcal{M}|)}{C_{\ell}}}{t_{\ell^*} + o(t_{\ell^*})} \end{aligned} \quad (\text{C.15})$$

where we use the fact that  $E_{\text{sp}, \ell^*}(R) = E_{\text{rc}, \ell^*}(R)$  when  $R$  is sufficiently close to  $C$ . We note that by (C.12), we always have  $\frac{\ln(|\mathcal{M}|)}{C} \leq t_{\ell^*} \leq \frac{\ln(|\mathcal{M}|)}{R}$ . By letting  $|\mathcal{M}| \rightarrow \infty$  and then  $R \rightarrow C$ , the left-hand side of (C.15) becomes the DAF and the  $t_{\ell^*}$  in the right-hand side becomes  $\frac{\ln(|\mathcal{M}|)}{C_{\ell^*}}$ . We have thus proven  $\text{DAF} \geq \sum_{\ell=1}^L \frac{C_{\ell}^*}{C_{\ell}}$  for any possible choices of  $(|\mathcal{M}|, t_1, \dots, t_L)$  and any possible codebook for each hop.

## C.2 Proof of Lemma 4.4.2

Let  $\check{k} = \min_m \tilde{k}_m$ , and note that

$$\Pr\left(\mathcal{A}_3 \middle| \mathcal{A}_1^c \mathcal{A}_2^c\right) = \Pr\left(\check{k} \geq K_{\max} \middle| \mathcal{A}_1^c \mathcal{A}_2^c\right). \quad (\text{C.16})$$

For notational simplicity, we denote the expectation conditioned on the event  $\mathcal{A}_1^c \mathcal{A}_2^c$  as  $E_*\left\{ \cdot \right\} \triangleq E\left\{ \cdot \middle| \mathcal{A}_1^c \mathcal{A}_2^c \right\}$ . Using Markov's inequality, we then have

$$\Pr\left(\check{k} \geq K_{\max} \middle| \mathcal{A}_1^c \mathcal{A}_2^c\right) \leq \frac{E_*\left\{ \check{k} \right\}}{K_{\max}}. \quad (\text{C.17})$$

To bound  $E_*\left\{ \check{k} \right\}$ , we first note that the definition of  $Z_m(k)$  in (4.45) holds only for the range of  $k \geq 1$ . To facilitate our discussion, we define

$$Z_m(0) = -K\Delta R + \ln \frac{1}{1 - e^{-K\Delta R}} > -K\Delta R$$

. Namely, before we receive any observations (i.e.,  $k = 0$ ), the LLR value is computed using the uniform prior  $P(m) = e^{-K\Delta R}$  over  $\mathcal{M} = [1, e^{K\Delta R}]$ . We then introduce the following Lemmas.

**Lemma C.2.1** *For any fixed  $\Delta$ , there exists a constant  $B$  such that*

$$|Z_m(k+1) - Z_m(k)| < \Delta \cdot B$$

*with probability one regardless of the values of  $m \in \mathcal{M}$  and  $k \geq 0$ .*

For the proof of this Lemma, see Appendix C.3.

**Lemma C.2.2** *For sufficiently large but fixed  $\Delta$  and assuming  $m_0$  is the transmitted message, the random process*

$$Z_{m_0}(k) - k\Delta (R + \alpha(C_{\ell^*} - R)) \quad (\text{C.18})$$

is a submartingale with respect to the time index  $k \geq 0$ .

For the proof of this Lemma, see Appendix C.4.

We then notice that

$$\begin{aligned}
& \Pr\left(Z_m(j) < \ln(1/\eta)\right) \\
&= \Pr\left(Z_m(j) < K\Delta\alpha(C_{\ell^*} - R)\right) \\
&= \Pr\left(Z_m(j) - j\Delta(R + \alpha(C_{\ell^*} - R)) - Z_m(0) < (K - j)\Delta(R + \alpha(C_{\ell^*} - R))\right) \\
&\leq c_1 \leq e^{-jc_2}
\end{aligned} \tag{C.19}$$

for some positive constants  $c_1, c_2 > 0$ , where (C.19) is by applying Lemma C.2.1 and Azuma's inequality to the submartingale  $Z_m(j) - j\Delta(R + \alpha(C_{\ell^*} - R))$  from Lemma C.2.2. We then have

$$E_*\{\tilde{k}_{m_0}\} = \sum_{j=0}^{\infty} \Pr(\tilde{k}_m > j) \tag{C.20}$$

$$\leq \sum_{j=1}^{\infty} \Pr(Z_m(j) < \ln(1/\eta)) < \infty. \tag{C.21}$$

Namely, the stopping time  $\tilde{k}_{m_0}$  has finite expectation. With bounded  $E_*\{\tilde{k}_{m_0}\}$  and the globally bounded variation established in Lemma C.2.1, we can then apply Doob's optional stopping theorem and get

$$\begin{aligned}
E_*\left\{Z_m(\tilde{k}_{m_0}) - \tilde{k}_{m_0}\Delta(R + \alpha(C_{\ell^*} - R))\right\} &\geq E_*\{Z_m(0)\} \\
&\geq -K\Delta R
\end{aligned}$$

which implies

$$E_*\{\tilde{k}_{m_0}\} \leq \frac{E_*\{Z_m(\tilde{k}_{m_0})\} + K\Delta R}{\Delta(R + \alpha(C_{\ell^*} - R))}. \tag{C.22}$$

We then note that

$$E_* \left\{ Z_{m_0}(\tilde{k}_{m_0}) \right\} \leq E_* \left\{ Z_{m_0}(\tilde{k}_{m_0} - 1) \right\} + \Delta \cdot B \quad (\text{C.23})$$

$$\leq K\Delta\alpha(C_{\ell^*} - R) + \Delta \cdot B, \quad (\text{C.24})$$

where the first inequality is due to Lemma C.2.1 and the second inequality is because at time  $k = \tilde{k}_{m_0} - 1$ , the term  $Z_{m_0}(k)$  has not hit the threshold  $K\Delta\alpha(C_{\ell^*} - R)$  yet.

We then note that since  $\check{k} \leq \tilde{k}_{m_0}$ , we have  $E_* \left\{ \check{k} \right\} \leq E_* \left\{ \tilde{k}_{m_0} \right\}$ . Then, after combining (C.22) and (C.24), we get

$$E_* \left\{ \check{k} \right\} \leq K + \frac{B}{R + \alpha(C_{\ell^*} - R)}. \quad (\text{C.25})$$

Finally, after combining (C.25) and (C.17), we get

$$\Pr \left( \mathcal{A}_3 \mid \mathcal{A}_1^c \mathcal{A}_2^c \right) \leq (1 + \varsigma_1(K)) e^{-K\Delta\alpha(C_{\ell^*} - R)}, \quad (\text{C.26})$$

where the term  $\varsigma_1(K) = B/[K(R + \alpha(C_{\ell^*} - R))]$  goes to zero as  $K \rightarrow \infty$ . ■

### C.3 Proof of Lemma C.2.1

Recall (4.45) and note that we can write

$$Z_m(k+1) = \ln \left( \frac{\Pr \left( \left[ \vec{Y}_{\ell^*} \right]_1^k \mid m \right) \cdot \Pr \left( \vec{Y}_{\ell^*}[k + \ell^*] \mid \mathbf{x}_{k+1, i_{k+1}^{[m]}}^{(\ell^*)} \right)}{\sum_{m' \neq m} \Pr \left( \left[ \vec{Y}_{\ell^*} \right]_1^k \mid m' \right) \cdot \Pr \left( \vec{Y}_{\ell^*}[k + \ell^*] \mid \mathbf{x}_{k+1, i_{k+1}^{[m']}}^{(\ell^*)} \right)} \right), \quad (\text{C.27})$$

where we recall that  $\vec{Y}_{\ell^*}[k + \ell^*]$  denotes the channel output symbols of the  $(k+1)$ -th microblock at the bottleneck receiver and we slightly abuse notation for

$$\Pr \left( \vec{Y}_{\ell^*}[k + \ell^*] \mid \mathbf{x}_{k+1, i_{k+1}^{[m]}}^{(\ell^*)} \right) = \Pr \left( \vec{Y}_{\ell^*}[k + \ell^*] \mid \vec{X}_{\ell^*}[k + \ell^*] = \mathbf{x}_{k+1, i_{k+1}^{[m]}}^{(\ell^*)} \right). \quad (\text{C.28})$$

Next, we define

$$p_{\min} \triangleq \min_{X \in \mathcal{X}_{\ell^*}, Y \in \mathcal{Y}_{\ell^*}} P_{\ell^*}(Y|X) \quad (\text{C.29})$$

and

$$p_{\max} \triangleq \max_{X \in \mathcal{X}_{\ell^*}, Y \in \mathcal{Y}_{\ell^*}} P_{\ell^*}(Y|X). \quad (\text{C.30})$$

Since  $\mathcal{X}_{\ell^*}$  and  $\mathcal{Y}_{\ell^*}$  are finite and  $P_{\ell}(y|x) > 0$  for all  $(\ell \in [1, L], x \in \mathcal{X}_{\ell}, y \in \mathcal{Y}_{\ell})$ , we know that  $p_{\min}$  and  $p_{\max}$  exist and are  $> 0$ .

We then notice that for any fixed  $\Delta$ , for any choice of  $m$ , and for any micro-block index  $k$ , we have

$$(p_{\min})^{\Delta} \leq \Pr\left(\vec{Y}_{\ell^*}[k + \ell^*] \middle| \mathbf{x}_{k+1, i_{k+1}^{[m]}}^{(\ell^*)}\right) \leq (p_{\max})^{\Delta}. \quad (\text{C.31})$$

We thus have

$$\begin{aligned} Z_m(k+1) &\leq \ln \left( \left( \frac{p_{\max}}{p_{\min}} \right)^{\Delta} \cdot \frac{\Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m\right)}{\sum_{m' \neq m} \Pr\left(\left[\vec{Y}_{\ell^*}\right]_1^k \middle| m'\right)} \right) \\ &= \Delta \ln \left( \frac{p_{\max}}{p_{\min}} \right) + Z_m(k) \end{aligned} \quad (\text{C.32})$$

and similarly

$$Z_m(k+1) \geq \Delta \ln \left( \frac{p_{\min}}{p_{\max}} \right) + Z_m(k). \quad (\text{C.33})$$

Applying the bounds (C.32) and (C.33) to the absolute difference  $|Z_m(k+1) - Z_m(k)|$ , we get

$$|Z_m(k+1) - Z_m(k)| \leq \Delta \cdot \left| \ln \frac{p_{\max}}{p_{\min}} \right| \triangleq \Delta \cdot B, \quad (\text{C.34})$$



which completes the proof. ■

#### C.4 Proof of Lemma C.2.2

We first define the filtration  $\mathcal{F}_k$  on which the submartingale is defined. Let  $\mathbf{Y}^k \triangleq \left[ \vec{Y}_{\ell^*} \right]_1^k$  denote the history of all observed channel outputs at the bottleneck receiver after the  $k$ -th microblock, let  $[\pi]^k$  denote all permutations  $\pi_1$  to  $\pi_k$  for the outer code, and let  $[\mathbf{X}^{(\ell^*)}]^k$  denote all existing inner codebooks until time  $k$ . The filtration  $\mathcal{F}_k$  is then generated by the tuple  $(\mathbf{Y}^k, [\pi]^k, [\mathbf{X}^{(\ell^*)}]^k)$ . To prove Lemma C.2.2, we thus have to prove (i)  $E\{|Z_{m_0}(k)|\} < \infty$  and (ii)

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathcal{F}_k \right\} \geq \Delta (R + \alpha(C_{\ell^*} - R)). \quad (\text{C.35})$$

The finite expectation can be quickly proven by iteratively applying Lemma C.2.1. To prove the second statement, let  $\mathbf{y}_{k+1} \triangleq \vec{Y}_{\ell^*}[k + \ell^*]$  denote the channel outputs of the bottleneck hop corresponding to only the  $(k+1)$ -th microblock. Without loss of generality, we assume that the transmitted message  $m_0$  results in  $i_{k+1}^{[m_0]} = 1$  based on the randomly chosen permutation  $\pi_{k+1}$  from (4.43). This can be achieved by renaming whatever the output  $i_{k+1}^{[m_0]}$  is as the first symbol. For ease of exposition, for the remainder of this section, we drop the explicit mention of the  $(\ell^*)$ -th hop when discussing the microblock codewords, i.e., we will set  $\mathbf{x}_{k,i}^{(\ell^*)} \triangleq \mathbf{x}_{k,i}$  for the rest of this section. We continue by writing

$$Z_{m_0}(k+1) = \ln \left( \frac{\Pr(\mathbf{Y}^k | m_0) \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1, i_{k+1}^{[m]}})} \right). \quad (\text{C.36})$$

The difference between (C.36) and (4.45) thus becomes

$$Z_{m_0}(k+1) - Z_{m_0}(k) = \ln \left( \frac{\left( \sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \right) \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1, i_{k+1}^{[m]}})} \right). \quad (\text{C.37})$$

Since our scheme is based on a concatenated inner/outer code construction, the corresponding analysis, as will be seen, is much more involved than the single random code construction. Specifically, we notice that the above difference depends on the realizations of the following sets of random variables.

- (a) With the inner/outer code construction, any message  $m \in \mathcal{M}$  will be encoded as  $i_{k+1}^{[m]} \in [1, e^{\Delta R_I}]$  based on the randomly chosen permutation  $\pi_{k+1}$ , see the definition in (4.43). The first random variable to consider is thus the random outer code permutation  $\pi_{k+1}$ .
- (b) For any outer code message  $i \in [1, e^{\Delta R_I}]$ , the corresponding inner codeword  $\mathbf{x}_{k,i}$  is chosen randomly. The second set of random variables is the random choices of  $\mathbf{x}_{k,i}$  for all  $i \in [2, e^{\Delta R_I}]$ . Namely, the choices of inner codewords that are not selected by the actual transmitted message  $m_0$ . (Recall that we assume  $i_{k+1}^{[m_0]} = 1$ .)
- (c) The  $(k+1)$ -th microblock codeword  $\mathbf{x}_{k+1,1}$ , i.e., the codeword choice of the transmitted outer code message  $i_{k+1}^{[m_0]}$ .
- (d) The channel output symbols corresponding to the  $(k+1)$ -th microblock  $\mathbf{y}_{k+1}$  when the input codeword is  $\mathbf{x}_{k+1,1}$ .

In the following, we take a sequence of conditional expectations until we reach the desired conditional expectation in (C.49).

Our first step is to take the expectation over the randomness in (b) while conditioning on (a), (c), and (d). Specifically, we note that  $\ln \left( \frac{a}{x} \right)$  is a convex function of  $x$

and apply Jensen's inequality to (C.37) while conditioning on (a), (c), and (d). This leads to the bound

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathbf{x}_{k+1,1}, \pi_{k+1}, \mathbf{y}_{k+1}, \mathcal{F}_k \right\} \\ \geq \ln \left( \frac{\left( \sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \right) \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) E_* \left\{ \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1, i_{k+1}^{[m]}}) \middle| \mathbf{x}_{k+1,1}, \pi_{k+1}, \mathbf{y}_{k+1} \right\}} \right), \quad (\text{C.38})$$

which uses the fact that given  $\mathbf{x}_{k+1,1}$ ,  $\pi_{k+1}$ ,  $\mathbf{y}_{k+1}$ , and  $\mathcal{F}_k$ , the terms  $\Pr(\mathbf{Y}^k | m)$  and  $\Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})$  become deterministic.

To continue, we define the set

$$\mathcal{M}_{k+1}^{[m_0]} = \left\{ m \in \mathcal{M} : i_{k+1}^{[m]} = i_{k+1}^{[m_0]} \right\}, \quad (\text{C.39})$$

which contains all outer code messages  $m$  that result in the same microblock message  $i_{k+1}^{[m]}$  for the  $(k+1)$ -th microblock as that of the transmitted message  $m_0$ . It is clear that  $\mathcal{M}_{k+1}^{[m_0]}$  is a function of the permutation  $\pi_{k+1}$ .

Now, continuing from (C.38), we observe the following. First, for all messages  $m \in \mathcal{M}_{k+1}^{[m_0]}$ , since the microblock messages are equal, the microblock codewords must be equal too and thus

$$E_* \left\{ \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1, i_{k+1}^{[m]}}) \middle| \mathbf{x}_{k+1,1}, \pi_{k+1}, \mathbf{y}_{k+1} \right\} = \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1}) \quad (\text{C.40})$$

for all  $m \in \mathcal{M}_{k+1}^{[m_0]}$ . Second, for all messages  $m \notin \mathcal{M}_{k+1}^{[m_0]}$ , taking the expectation over the distribution of the microblock codewords gives the marginal distribution, and

$$E_* \left\{ \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1, i_{k+1}^{[m]}}) \middle| \mathbf{x}_{k+1,1}, \pi_{k+1}, \mathbf{y}_{k+1} \right\} = \Pr(\mathbf{y}_{k+1}) \quad (\text{C.41})$$

for all  $m \notin \mathcal{M}_{k+1}^{[m_0]}$ . Using these observations, (C.38) can be rewritten in the equivalent form

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathbf{x}_{k+1,1}, \pi_{k+1}, \mathbf{y}_{k+1}, \mathcal{F}_k \right\} \geq \ln \left( \frac{\left( \sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \right) \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \left( \mathbf{1}_{\mathcal{M}_{k+1}^{[m_0]} \setminus \{m_0\}}(m) \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1}) + \mathbf{1}_{\mathcal{M} \setminus \mathcal{M}_{k+1}^{[m_0]}}(m) \Pr(\mathbf{y}_{k+1}) \right)} \right), \quad (\text{C.42})$$

where the indicator function  $\mathbf{1}_{\mathcal{A}}(x)$  is defined as in (4.59).

By further averaging over  $\pi_{k+1}$  and applying Jensen's inequality again, we obtain the bound

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathbf{x}_{k+1,1}, \mathbf{y}_{k+1}, \mathcal{F}_k \right\} \geq \ln \left( \frac{\left( \sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \right) \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\sum_{m \neq m_0} \Pr(\mathbf{Y}^k | m) \left( p_1 \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1}) + p_2 \cdot \Pr(\mathbf{y}_{k+1}) \right)} \right), \quad (\text{C.43})$$

where  $p_1 \triangleq \Pr(m \in \mathcal{M}_{k+1}^{[m_0]} \setminus \{m_0\})$  and  $p_2 \triangleq \Pr(m \notin \mathcal{M}_{k+1}^{[m_0]})$ . This simplifies to

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathbf{x}_{k+1,1}, \mathbf{y}_{k+1}, \mathcal{F}_k \right\} \geq \ln \left( \frac{\Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1})}{\frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1} \cdot \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1}) + \frac{|\mathcal{M}| - |\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}| - 1} \cdot \Pr(\mathbf{y}_{k+1})} \right) \quad (\text{C.44})$$

by noting that regardless of the value of  $m$ , we always have  $p_1 = \frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1}$  and  $p_2 = \frac{|\mathcal{M}| - |\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}| - 1}$  due to the uniform random permutation of  $\pi_{k+1}$ .

To further bound (C.44), we notice that by the asymptotic equipartition property of  $\mathbf{y}_{k+1}$  and  $\mathbf{x}_{k+1,1}$ , there exists a sufficiently large  $\Delta$  for any  $\epsilon > 0$  such that the event

$$\Pr(\mathbf{y}_{k+1}) > e^{-\Delta(H(Y)+\epsilon)} \text{ and} \quad (\text{C.45})$$

$$e^{-\Delta(H(Y|X)+\epsilon)} < \Pr(\mathbf{y}_{k+1} | \mathbf{x}_{k+1,1}) < e^{-\Delta(H(Y|X)-\epsilon)} \quad (\text{C.46})$$

has probability  $\geq 1 - \epsilon$ . Using this property and Lemma C.2.1, we obtain

$$\begin{aligned} & E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathcal{F}_k \right\} \\ & \geq \epsilon \Delta B + (1 - \epsilon) \ln \left( \frac{e^{-\Delta(H(Y|X)+\epsilon)}}{\frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1} e^{-\Delta(H(Y|X)-\epsilon)} + \frac{|\mathcal{M}| - |\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}| - 1} e^{-\Delta(H(Y)-\epsilon)}} \right) \end{aligned} \quad (\text{C.47})$$

By upper bounding  $\frac{|\mathcal{M}|e^{-\Delta R_I} - 1}{|\mathcal{M}| - 1} \leq e^{-\Delta R_I}$  and  $\frac{|\mathcal{M}| - |\mathcal{M}|e^{-\Delta R_I}}{|\mathcal{M}| - 1} \leq 1$  in the denominator of (C.47), we have

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathcal{F}_k \right\} \geq \epsilon \cdot \Delta \cdot B + (1 - \epsilon) \ln \left( \frac{1}{e^{-\Delta(R_I - 2\epsilon)} + e^{-\Delta(C_{\ell^*} - 2\epsilon)}} \right). \quad (\text{C.48})$$

Now, since our choice of  $R_I$  in (4.40) satisfies  $R_I < C_{\ell^*}$  and  $\epsilon$  can be made arbitrarily small when a sufficiently large  $\Delta$  is used, we can rewrite (C.48) as

$$E_* \left\{ Z_{m_0}(k+1) - Z_{m_0}(k) \middle| \mathcal{F}_k \right\} \geq \Delta R_I + \varsigma(\Delta) \quad (\text{C.49})$$

where  $\varsigma(\Delta) \rightarrow 0$  for sufficiently large  $\Delta$ . Finally, the choice of  $R_I$  from (4.40) always satisfies  $R_I > R + \alpha(C_{\ell^*} - R)$ . Using a sufficiently large  $\Delta$ , (C.49) implies (C.35), finishing the proof.  $\blacksquare$

### C.5 Proof of Lemma 4.4.3

In this section, for notational simplicity, we denote the expectation conditioned on the event  $\mathcal{A}_1^c \mathcal{A}_2^c$  as  $E_* \left\{ \cdot \right\} \triangleq E \left\{ \cdot \mid \mathcal{A}_1^c \mathcal{A}_2^c \right\}$ .

We now bound the probability of the event  $\mathcal{A}_5 \mid \mathcal{A}_1^c \mathcal{A}_2^c$ . We first notice that the bottleneck receiver will transmit  $\check{k} = \min_m \tilde{k}_m$  microblocks of size  $\Delta$  during the sequential learning phase, 1 microblock of size  $\Delta$  for the correction flag, and at most  $K$  additional microblocks of size  $\tilde{\Delta}$ . We then observe that whether a microblock in any of the post-bottleneck hops is in error is determined by the post-bottleneck hop channel realizations and is thus independent of the number of microblocks transmitted by the bottleneck receiver. As a result, by Wald's lemma, we can employ the same union bound argument as in (4.66) using the expected number of microblock transmissions and obtain

$$\Pr(\mathcal{A}_5 \mid \mathcal{A}_1^c \mathcal{A}_2^c) \leq \left( E_* \left\{ \check{k} \right\} + 1 \right) \cdot \mathbb{E}_\Delta + K \cdot \mathbb{E}_{\tilde{\Delta}} \quad (\text{C.50})$$

where

$$\mathbb{E}_\Delta = \sum_{\ell=\ell^*+1}^L e^{-\Delta E_{rc,\ell}(R_I)} \quad (\text{C.51})$$

and  $\mathbb{E}_{\tilde{\Delta}}$  is defined similarly by replacing  $\Delta$  with  $\tilde{\Delta}$ . Due to (C.25), we know  $E_* \left\{ \check{k} \right\} \leq K(1 + \varsigma_1(K))$ .

Now we note that we can re-write (C.50) as

$$\Pr(\mathcal{A}_5 \mid \mathcal{A}_1^c \mathcal{A}_2^c) \leq \sum_{\ell=\ell^*+1}^L \left( E_* \left\{ \check{k} \right\} + 1 \right) e^{-\Delta E_{rc,\ell}(R_I)} + K e^{-\tilde{\Delta} E_{rc,\ell}(R_I)}. \quad (\text{C.52})$$

Next, since  $\tilde{\Delta} = \frac{\ln(K)}{R_I} + \Delta$ , we have for any values of  $\Delta$  and any hop  $\ell$

$$\begin{aligned} K e^{-\tilde{\Delta} E_{rc,\ell}(R_I)} &= K e^{-\frac{\ln(K)}{R_I} E_{rc,\ell}(R_I)} e^{-\Delta E_{rc,\ell}(R_I)} \\ &\leq K e^{-\frac{\ln(K)}{R_I} E_{rc,\ell^*}(R_I)} e^{-\Delta E_{rc,\ell}(R_I)}, \end{aligned} \quad (\text{C.53})$$

meaning we can re-write (C.52) as

$$\begin{aligned} \Pr(\mathcal{A}_5 | \mathcal{A}_1^c \mathcal{A}_2^c) &\leq \left( E_* \left\{ \tilde{k} \right\} + 1 + K e^{-\frac{\ln(K)}{R_I} E_{rc, \ell^*}(R_I)} \right) \cdot \sum_{\ell=\ell^*+1}^L e^{-\Delta E_{rc, \ell}(R_I)} \\ &\leq K(1 + \varsigma_2(K)) \cdot \sum_{\ell=\ell^*+1}^L e^{-\Delta E_{rc, \ell}(R_I)}, \end{aligned} \quad (\text{C.54})$$

where  $\varsigma_2(K) \rightarrow 0$  as  $K \rightarrow \infty$ . This completes the proof.  $\blacksquare$

## C.6 Proof of Lemma 4.4.4

In this section, for notational simplicity, we denote the expectation conditioned on the event  $\mathcal{A}_1^c \mathcal{A}_2^c$  as  $E_* \left\{ \cdot \right\} \triangleq E \left\{ \cdot \mid \mathcal{A}_1^c \mathcal{A}_2^c \right\}$ . We start with the observation that we can write the expected duration as

$$E\{D\} = E\{D \mid \mathcal{A}_1 \cup \mathcal{A}_2\} \Pr(\mathcal{A}_1 \cup \mathcal{A}_2) + E_*\{D\} \Pr(\mathcal{A}_1^c \mathcal{A}_2^c). \quad (\text{C.55})$$

For the first term in (C.55), we note that the maximum possible duration of  $D$  (unit: slots) is given by

$$D_{\max} = (K_{\max} + 1)\Delta + K\tilde{\Delta}. \quad (\text{C.56})$$

We can thus bound this term by

$$E\{D \mid \mathcal{A}_1 \cup \mathcal{A}_2\} \Pr(\mathcal{A}_1 \cup \mathcal{A}_2) \leq D_{\max} (\Pr(\mathcal{A}_1) + \Pr(\mathcal{A}_2)), \quad (\text{C.57})$$

where we note that due to (4.66) and (4.71), the upper bound decays exponentially as a function of  $\Delta$ .

For the second term in (C.55), define  $D_{\ell^*}$  as the combined duration of the learning and correction phase of the bottleneck receiver. We can then bound  $E_*\{D\}$  as

$$E_*\{D\} \leq E_*\{(D - D_{\ell^*})^+\} + E_*\{D_{\ell^*}\} \quad (\text{C.58})$$

and note that  $(D - D_{\ell^*})^+ > 0$  implies that the event  $\mathcal{A}_5$  must be true, since it is only possible for the bottleneck receiver's duration to be longer than the destination's if any of the post-bottleneck transmissions are in error, which destroys the synchronization between the bottleneck receiver and the destination. In addition, it is also trivially true that  $(D - D_{\ell^*})^+ \leq D_{\max}$ . We thus have

$$E_*\{(D - D_{\ell^*})^+\} \leq D_{\max}P(\mathcal{A}_5|\mathcal{A}_1^c\mathcal{A}_2^c). \quad (\text{C.59})$$

We note that by (4.41),  $D_{\max}$  grows exponentially at a rate  $e^{K\Delta\alpha(C_{\ell^*}-R)}$  and by (4.74),  $\Pr(\mathcal{A}_5|\mathcal{A}_1^c\mathcal{A}_2^c)$  decays exponentially fast in  $\Delta$ . Combining these results, we have

$$\begin{aligned} E\{D\} &\leq E_*\{D_{\ell^*}\}P(\mathcal{A}_1^c\mathcal{A}_2^c) + \varsigma(\Delta) \\ &\leq E_*\{D_{\ell^*}\} + \varsigma(\Delta) \\ &\leq \left(1 + E_*\{\check{k}\}\right)\Delta + E_*\{|\mathcal{K}_{\text{err}}|\}\tilde{\Delta} + \varsigma(\Delta), \end{aligned} \quad (\text{C.60})$$

where  $\varsigma(\Delta)$  signifies a term which diminishes as  $\Delta \rightarrow \infty$ . The term  $E_*\{\check{k}\}$  is upper bounded in (C.25). We now upper bound  $E_*\{|\mathcal{K}_{\text{err}}|\}$ .

First, define  $K_{\text{true}}$  as the number of erroneous bucket indices at the bottleneck receiver assuming that it knows the true  $m$ . We then have

$$E_*\{|\mathcal{K}_{\text{err}}|\} \leq E_*\{(|\mathcal{K}_{\text{err}}| - K_{\text{true}})^+\} + E_*\{K_{\text{true}}\} \quad (\text{C.61})$$

Note that  $(|\mathcal{K}_{\text{err}}| - K_{\text{true}})^+ > 0$  implies that the sequential learning phase did not provide the correct estimate, i.e., the sequentially learned  $\hat{m} \neq m_0$ , the actual transmitted message. This implies that either  $\mathcal{A}_3$  or  $\mathcal{A}_4$  must be true. Combined with the fact that  $(|\mathcal{K}_{\text{err}}| - K_{\text{true}})^+ \leq K$ , we have

$$\begin{aligned} E_*\{(|\mathcal{K}_{\text{err}}| - K_{\text{true}})^+\} &\leq K \cdot \Pr\left(\mathcal{A}_3 \cup \mathcal{A}_4 \mid \mathcal{A}_1^c\mathcal{A}_2^c\right) \\ &\leq K(2 + \varsigma_1(K))e^{-K\Delta\alpha(C_{\ell^*}-R)}, \end{aligned} \quad (\text{C.62})$$



where (C.62) is due to (4.72) and (4.73). Furthermore, since  $K_{\text{true}}$  is computed based on the actual transmitted message  $m_0$ , each of the first  $K$  microblocks that is in error (i.e., contributing to  $K_{\text{true}}$ ) is an independent event with probability  $\leq e^{-\Delta E_{\text{rc}, \ell^*}(R_I)}$ . By the union bound, we have

$$E_*\{K_{\text{true}}\} < K \cdot e^{-\Delta E_{\text{rc}, \ell^*}(R_I)}. \quad (\text{C.63})$$

Combining these results, we get

$$E_*\{|\mathcal{K}_{\text{err}}|\} \leq K \left( (2 + \varsigma_1(K)) e^{-K\Delta\alpha(C_{\ell^*} - R)} + e^{-\Delta E_{\text{rc}, \ell^*}(R_I)} \right). \quad (\text{C.64})$$

Finally, combining (C.64), (C.60), and (C.25).

$$\begin{aligned} E\{D\} &\leq [1 + K(1 + \varsigma_1(K))] \Delta + K \left( (2 + \varsigma_1(K)) e^{-K\Delta\alpha(C_{\ell^*} - R)} + e^{-\Delta E_{\text{rc}, \ell^*}(R_I)} \right) \\ &\quad \cdot \left( \Delta + \frac{\ln(K)}{R_I} \right) + \varsigma(\Delta) \\ &\leq K (1 + \varsigma_3(K, \Delta)) \Delta, \end{aligned} \quad (\text{C.65})$$

where  $\varsigma_3(K, \Delta) \rightarrow 0$  if we let  $\Delta \rightarrow \infty$  and then  $K \rightarrow \infty$  in this order. ■

## REFERENCES

## REFERENCES

- [1] 5G Americas, “Global 5G: Implications of a transformational technology,” Tech. Rep., Sep. 2019. [Online]. Available: <https://www.5gamericas.org/global-5g-implications-of-a-transformational-technology/>
- [2] Cisco, “Cisco annual internet report,” Tech. Rep., Mar. 2020. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. Zhang, “What will 5G be?” *IEEE Journ. Sel. Areas in Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [4] F. Boccardi, R. W. Heath, Jr., A. Lozano, T. L. Marzetta, and P. Popovski, “Five disruptive technology directions for 5G,” *IEEE Commun. Magaz.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
- [5] Z. Pi and F. Khan, “An introduction to millimeter wave mobile broadband systems,” *IEEE Commun. Magaz.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.
- [6] 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.
- [7] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-wave cellular wireless networks: Potentials and challenges,” *Proc. IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014.
- [8] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, “Millimeter-wave massive MIMO: The next wireless revolution?” *IEEE Commun. Magaz.*, vol. 52, no. 9, pp. 56–62, Sep. 2014.
- [9] T. L. Marzetta, “Non-cooperative cellular wireless with unlimited numbers of base station antennas,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.
- [10] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, “Energy and spectral efficiency of very large multiuser MIMO systems,” *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.
- [11] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, “Scaling up MIMO: Opportunities and challenges with very large arrays,” *IEEE Sig. Proc. Magaz.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

- [12] Aalto University, AT&T, BUPT, CMCC, Ericsson, Huawei, Intel, KT Corporation, Nokia, NTT DOCOMO, NYU, Qualcomm, Samsung, U. Bristol, and USC, "White paper on "5G channel model for bands up to 100 GHz"," *v2.3*, Oct. 2016.
- [13] 3GPP, "Study on channel model for frequency spectrum above 6 GHz," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.900, Sep. 2016, version 14.1.0.
- [14] D. Gesbert, M. Kountouris, J. R. W. Heath, C.-B. Chae, and T. Salzer, "Shifting the MIMO paradigm: From single user to multiuser communications," *IEEE Sig. Proc. Magaz.*, vol. 24, no. 5, pp. 36–46, Oct. 2007.
- [15] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, "An introduction to the multi-user MIMO downlink," *IEEE Commun. Magaz.*, vol. 42, no. 10, pp. 60–67, Oct. 2004.
- [16] V. Venkateswaran and A.-J. van der Veen, "Analog beamforming in MIMO communications with phase shift networks and online channel estimation," *IEEE Trans. Sig. Proc.*, vol. 58, no. 8, pp. 4131–4143, Aug. 2010.
- [17] E. Torkildson, U. Madhow, and M. Rodwell, "Indoor millimeter wave MIMO: Feasibility and performance," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4150–4160, Dec. 2011.
- [18] J. Brady, N. Behdad, and A. M. Sayeed, "Beamspace MIMO for millimeter-wave communications: System architecture, modeling, analysis and measurements," *IEEE Trans. Ant. Propag.*, vol. 61, no. 7, pp. 3814–3827, Jul. 2013.
- [19] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5g cellular communications: Theoretical feasibility and prototype results," *IEEE Commun. Magaz.*, vol. 52, no. 2, pp. 106–113, Feb. 2014.
- [20] D. J. Love, R. W. Heath, Jr., and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2735–2747, Oct. 2003.
- [21] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, "On beamforming with finite rate feedback in multiple-antenna systems," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2562–2579, Oct. 2003.
- [22] F. Boccardi, B. Clerckx, A. Ghosh, E. Hardouin, G. Jöngren, K. Kusume, E. Onggosanusi, and Y. Tang, "Multiple antenna techniques in LTE-Advanced," *IEEE Commun. Magaz.*, vol. 50, no. 2, pp. 114–121, Mar. 2012.
- [23] C. Lim, T. Yoo, B. Clerckx, B. Lee, and B. Shim, "Recent trend of multiuser MIMO in LTE-Advanced," *IEEE Commun. Magaz.*, vol. 51, no. 3, pp. 127–135, Mar. 2013.
- [24] M. Medard, "The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel," *IEEE Trans. Inform. Theory*, vol. 46, no. 3, pp. 935–946, May 2000.

- [25] B. Hassibi and B. Hochwald, "How much training is needed in a multiple antenna wireless link?" *IEEE Trans. Inform. Theory*, vol. 49, no. 4, pp. 951–964, Apr. 2003.
- [26] G. Hariharan, V. Raghavan, and A. M. Sayeed, "Capacity of sparse wideband channels with partial channel feedback," *Trans. Emerging Telecommun. Technologies*, vol. 19, no. 4, pp. 475–493, Jun. 2008.
- [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 Trans. Commun.*, vol. 61, no. 10, pp. 4391–4403, Oct. 2013.
- [28] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, Jr., "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, Mar. 2014.
- [29] P. Schniter and A. M. Sayeed, "Channel estimation and precoder design for millimeter-wave communications: The sparse way," *Proc. IEEE Asilomar Conf. Signals, Systems and Computers, Pacific Grove, CA*, pp. 273–277, Nov. 2014.
- [30] D. Ramasamy, S. Venkateswaran, and U. Madhow, "Compressive tracking with 1000-element arrays: A framework for multi-gbps mm-wave cellular downlinks," *Proc. Annual Allerton Conf. Commun., Control and Computing, Allerton, IL*, pp. 690–697, Oct. 2012.
- [31] V. Raghavan and A. M. Sayeed, "MIMO capacity scaling and saturation in correlated environments," *Proc. IEEE Intern. Conf. Commun.*, vol. 5, pp. 3006–3010, May 2003.
- [32] A. Alkhateeb, O. E. Ayach, G. Leus, and R. W. Heath, Jr., "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journ. Sel. Topics in Sig. Proc.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.
- [33] 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 Commun. Magaz.*, vol. 52, no. 12, pp. 110–121, Dec. 2014.
- [34] A. Adhikary, E. A. Safadi, M. K. Samimi, R. Wang, G. Caire, T. S. Rappaport, and A. F. Molisch, "Joint spatial division and multiplexing for mm-Wave channels," *IEEE Journ. Sel. Areas in Commun.*, vol. 32, no. 6, pp. 1239–1255, Jun. 2014.
- [35] V. Raghavan, J. Cezanne, S. Subramanian, A. Sampath, and O. H. Koymen, "Beamforming tradeoffs for initial UE discovery in millimeter-wave MIMO systems," *IEEE Journ. Sel. Topics in Sig. Proc.*, vol. 10, no. 3, pp. 543–559, Apr. 2016.
- [36] V. Raghavan, S. Subramanian, J. Cezanne, A. Sampath, O. H. Koymen, and J. Li, "Directional hybrid precoding in millimeter-wave MIMO systems," *Proc. IEEE Global Telecommun. Conf., Washington, DC*, pp. 1–7, Dec. 2016.
- [37] T. Dahl, N. Christophersen, and D. Gesbert, "Blind MIMO eigenmode transmission based on the algebraic power method," *IEEE Trans. Sig. Proc.*, vol. 52, no. 9, pp. 2424–2431, Sep. 2004.

- [38] Y. Tang, B. Vucetic, and Y. Li, "An iterative singular vectors estimation scheme for beamforming transmission and detection in MIMO systems," *IEEE Commun. Letters*, vol. 9, no. 6, pp. 505–507, Jun. 2005.
- [39] S. Mandelli and M. Magarini, "Blind iterative singular vectors estimation and adaptive spatial loading in a reciprocal MIMO channel," *Proc. IEEE Wireless Commun. and Netwk. Conf., Istanbul, Turkey*, pp. 1036–1041, Apr. 2014.
- [40] S. Gazor and K. AlSuhaili, "Communications over the best singular mode of a reciprocal MIMO channel," *IEEE Trans. Commun.*, vol. 58, no. 7, pp. 1993–2001, Jul. 2010.
- [41] T. Dahl, S. S. Pereira, N. Christophersen, and D. Gesbert, "Intrinsic subspace convergence in TDD MIMO communication," *IEEE Trans. Sig. Proc.*, vol. 55, no. 6, pp. 2676–2687, Jun. 2007.
- [42] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 4th ed. Baltimore: Johns Hopkins University Press, Dec. 2012.
- [43] R. Prasad, B. N. Bharath, and C. R. Murthy, "Joint data detection and dominant singular mode estimation in time varying reciprocal MIMO systems," *Proc. IEEE Intern. Conf. on Acoust., Speech and Sig. Proc., Prague, Czech Rep.*, pp. 3240–3243, May 2011.
- [44] H. Ghauch, T. Kim, M. Skoglund, and M. Bengtsson, "Subspace estimation and decomposition in large millimeter-wave MIMO systems," *IEEE Journ. Sel. Topics in Sig. Proc.*, vol. 10, no. 3, pp. 528–542, Apr. 2016.
- [45] J. Choi, D. J. Love, and P. Bidigare, "Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory," *IEEE Journ. Sel. Topics in Sig. Proc.*, vol. 8, no. 5, pp. 802–814, Oct. 2014.
- [46] S. Noh, M. D. Zoltowski, and D. J. Love, "Training sequence design for feedback assisted hybrid beamforming in massive MIMO systems," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 187–200, Jan. 2016.
- [47] D. J. Love and R. W. Heath, Jr., "Equal gain transmission in multiple-input multiple-output wireless systems," *IEEE Trans. Commun.*, vol. 51, no. 7, pp. 1102–1110, Jul. 2003.
- [48] C.-H. Tse, K.-W. Yip, and T.-S. Ng, "Performance tradeoffs between maximum ratio transmission and switched-transmit diversity," *Proc. IEEE Intern. Symp. Pers. Indoor and Mob. Radio Commun., London, UK*, vol. 2, pp. 1485–1489, Sep. 2000.
- [49] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge: Cambridge University Press, Oct. 2012.
- [50] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, 1st ed. Englewood Cliffs, N.J: Prentice Hall, Apr. 1993.
- [51] F. Eicker, "Asymptotic normality and consistency of the least squares estimators for families of linear regressions," *Ann. Math. Stat.*, vol. 34, no. 2, pp. 447–456, Jun. 1963.

- [52] J. J. Choi, V. Raghavan, and D. J. Love, "Limited feedback design for the spatially correlated multi-antenna broadcast channel," *Proc. IEEE Global Telecommun. Conf., Atlanta, GA*, pp. 3481–3486, Dec. 2013.
- [53] V. Raghavan, G. Hariharan, and A. M. Sayeed, "Capacity of sparse multipath channels in the ultra-wideband regime," *IEEE Journ. Sel. Topics Sig. Proc.*, vol. 1, no. 3, pp. 357–371, Oct. 2007.
- [54] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5G and resilient public safety networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 36–42, Jan. 2018.
- [55] Google, "Project Loon." [Online]. Available: <https://www.google.com/loon/>
- [56] E. Haas, "Aeronautical channel modeling," *IEEE Trans. Veh. Technol.*, vol. 51, no. 2, pp. 254–264, Mar. 2002.
- [57] G. Matz and F. Hlawatsch, "Fundamentals of time-varying communication channels," in *Wireless Communications Over Rapidly Time-Varying Channels*, F. Hlawatsch and G. Matz, Eds. Oxford: Academic Press, 2011, pp. 1 – 63.
- [58] G. Matz, H. Bolcskei, and F. Hlawatsch, "Time-frequency foundations of communications: Concepts and tools," *IEEE Signal Processing Mag.*, vol. 30, no. 6, pp. 87–96, 2013.
- [59] G. Matz, "On non-WSSUS wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2465–2478, 2005.
- [60] P. Schniter, "Low-complexity equalization of OFDM in doubly selective channels," *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 1002–1011, 2004.
- [61] A. Kannu and P. Schniter, "Design and analysis of MMSE pilot-aided cyclic-prefixed block transmissions for doubly selective channels," *IEEE Trans. Signal Processing*, vol. 56, no. 3, pp. 1148–1160, 2008.
- [62] G. Leus, S. Zhou, and G. B. Giannakis, "Orthogonal multiple access over time- and frequency-selective channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 8, pp. 1942–1950, 2003.
- [63] X. Ma, G. B. Giannakis, and S. Ohno, "Optimal training for block transmissions over doubly selective wireless fading channels," *IEEE Trans. Signal Processing*, vol. 51, no. 5, pp. 1351–1366, 2003.
- [64] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, UK ; New York: Cambridge University Press, Jul. 2005.
- [65] T. Pollet, M. V. Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 191–193, Feb. 1995.
- [66] H. Bolcskei, "Blind estimation of symbol timing and carrier frequency offset in wireless OFDM systems," *IEEE Trans. Commun.*, vol. 49, no. 6, pp. 988–999, Jun. 2001.

- [67] Y. Yao and G. B. Giannakis, "Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 173–183, Jan. 2005.
- [68] P. H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2908–2914, Oct. 1994.
- [69] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.
- [70] M. Morelli and U. Mengali, "An improved frequency offset estimator for OFDM applications," in *Communication Theory Mini-Conference, 1999*, Jun. 1999, pp. 106–109.
- [71] D. Huang and K. B. Letaief, "Carrier frequency offset estimation for OFDM systems using null subcarriers," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 813–823, May 2006.
- [72] J. J. v. d. Beek, M. Sandell, and P. O. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Trans. Signal Processing*, vol. 45, no. 7, pp. 1800–1805, Jul. 1997.
- [73] N. Lashkarian and S. Kiaei, "Class of cyclic-based estimators for frequency-offset estimation of OFDM systems," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2139–2149, Dec. 2000.
- [74] M.-H. Hsieh and C.-H. Wei, "A low-complexity frame synchronization and frequency offset compensation scheme for OFDM systems over fading channels," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1596–1609, Sep. 1999.
- [75] M. Erturk, J. Haque, W. Moreno, and H. Arslan, "Doppler mitigation in OFDM-based aeronautical communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 1, pp. 120–129, Jan. 2014.
- [76] D. R. Ellis and E. Seckel, "Flying qualities of small general aviation airplanes. part 1. the influence of dutch-roll frequency, dutch-roll damping, and dihedral effect," Princeton University, Tech. Rep., 1969.
- [77] D. Scherber, P. Bidigare, R. O'Donnell, M. Rebholz, M. Oyarzun, C. Obrenovich, W. Kulp, D. Chang, and D. R. B. III, "Coherent distributed techniques for tactical radio networks: Enabling long range communications with reduced size, weight, power and cost," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, Nov. 2013, pp. 655–660.
- [78] G. Strang, *Linear Algebra and Its Applications, 4th Edition*, 4th ed. Belmont, CA: Cengage Learning, Jul. 2005.
- [79] 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, Oct. 2017.
- [80] D. Chu, "Polyphase codes with good periodic correlation properties (Corresp.)," *IEEE Trans. Inform. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.



- [81] 3GPP, “Evolved universal terrestrial radio access (E-UTRA); physical channels and modulation,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.211, Jan. 2016, version 13.0.0.
- [82] Federal Communications Commission, Office of Engineering and Technology, “FCC Online Table of Frequency Allocations,” Jun. 2018.
- [83] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, “Analytical model for connectivity in vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3341–3356, Nov. 2008.
- [84] Z. Zhang, G. Mao, and B. D. O. Anderson, “On the information propagation process in mobile vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 5, pp. 2314–2325, Jun. 2011.
- [85] —, “Stochastic characterization of information propagation process in vehicular ad hoc networks,” *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 1, pp. 122–135, Feb. 2014.
- [86] R. Nagel, “The effect of vehicular distance distributions and mobility on vanet communications,” in *2010 IEEE Intelligent Vehicles Symposium*, Jun. 2010, pp. 1190–1194.
- [87] M. Bennis, M. Debbah, and H. V. Poor, “Ultrareliable and low-latency wireless communication: Tail, risk, and scale,” *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
- [88] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, “Massive machine-type communications in 5g: physical and mac-layer solutions,” *IEEE Communications Magazine*, vol. 54, no. 9, pp. 59–65, Sep. 2016.
- [89] C. E. Shannon, “Probability of error for optimal codes in a Gaussian channel,” *The Bell System Technical Journal*, vol. 38, no. 3, pp. 611–656, May 1959.
- [90] R. G. Gallager, “A simple derivation of the coding theorem and some applications,” *IEEE Trans. Inform. Theory*, vol. 11, no. 1, pp. 3–18, 1965.
- [91] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, “Lower bounds to error probability for coding on discrete memoryless channels. II,” *Information and Control*, vol. 10, no. 5, pp. 522–552, 1967.
- [92] G. D. Forney, “Exponential error bounds for erasure, list, and decision feedback schemes,” *IEEE Transactions on Information Theory*, vol. 14, no. 2, pp. 206–220, Mar. 1968.
- [93] A. E. Ashikhmin, A. Barg, and S. N. Litsyn, “A new upper bound on the reliability function of the Gaussian channel,” *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 1945–1961, Sep. 2000.
- [94] S. Shamai and I. Sason, “Variations on the gallager bounds, connections, and applications,” *IEEE Transactions on Information Theory*, vol. 48, no. 12, pp. 3029–3051, Dec. 2002.

- [95] A. Barg and G. D. Forney, "Random codes: minimum distances and error exponents," *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2568–2573, Sep. 2002.
- [96] V. Y. F. Tan, "On the reliability function of the discrete memoryless relay channel," *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1550–1573, Apr. 2015.
- [97] D. Cao and V. Y. F. Tan, "Exact error and erasure exponents for the asymmetric broadcast channel," *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 865–885, Feb. 2020.
- [98] N. Merhav, "A lagrange–dual lower bound to the error exponent of the typical random code," *IEEE Transactions on Information Theory*, pp. 1–1, 2019.
- [99] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
- [100] —, "Feedback in the non-asymptotic regime," *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.
- [101] V. Kostina and S. Verdú, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3309–3338, 2012.
- [102] V. Y. F. Tan and O. Kosut, "On the dispersions of three network information theory problems," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 881–903, 2014.
- [103] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy, "Quasi-static multiple-antenna fading channels at finite blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4232–4265, 2014.
- [104] Y. Hu, J. Gross, and A. Schmeink, "On the capacity of relaying with finite blocklength," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1790–1794, 2016.
- [105] O. Teyeb, A. Muhammad, G. Mildh, E. Dahlman, F. Barac, and B. Makki, "Integrated access backhauled networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.
- [106] Y. Li, E. Pateromichelakis, N. Vucic, J. Luo, W. Xu, and G. Caire, "Radio resource management considerations for 5G millimeter wave backhaul and access networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 86–92, 2017.
- [107] E. C. V. D. Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971.
- [108] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sep. 1979.
- [109] A. E. Gamal and M. Aref, "The capacity of the semideterministic relay channel (corresp.)," *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 536–536, 1982.

- [110] K. Kobayashi, “Combinatorial structure and capacity of the permuting relay channel,” *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 813–826, 1987.
- [111] P. Vanroose and E. C. van der Meulen, “Uniquely decodable codes for deterministic relay channels,” *IEEE Transactions on Information Theory*, vol. 38, no. 4, pp. 1203–1212, 1992.
- [112] G. Kramer, M. Gastpar, and P. Gupta, “Cooperative strategies and capacity theorems for relay networks,” *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.
- [113] W. Nam, S. Chung, and Y. H. Lee, “Capacity of the Gaussian two-way relay channel to within  $\frac{1}{2}$  bit,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5488–5494, 2010.
- [114] W. Kang and S. Ulukus, “Capacity of a class of diamond channels,” *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4955–4960, 2011.
- [115] D. Gunduz, A. Yener, A. Goldsmith, and H. V. Poor, “The multiway relay channel,” *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 51–63, 2013.
- [116] L. Wang and M. Naghshvar, “On the capacity of the noncausal relay channel,” *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3554–3564, 2017.
- [117] S. H. Lim, K. T. Kim, and Y. Kim, “Distributed decode-forward for relay networks,” *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4103–4118, 2017.
- [118] X. Wu, L. P. Barnes, and A. Özgür, ““The Capacity of the Relay Channel”: Solution to Cover’s problem in the Gaussian case,” *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 255–275, 2019.
- [119] S.-H. Lee and S.-Y. Chung, “When is compress-and-forward optimal?” in *Information Theory and Applications Workshop (ITA)*, 2010. IEEE, 2010, pp. 1–3.
- [120] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
- [121] S. H. Lim, Y. Kim, A. E. Gamal, and S. Chung, “Noisy network coding,” *IEEE Trans. Inform. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.
- [122] U. Madhow, D. R. Brown, S. Dasgupta, and R. Mudumbai, “Distributed massive MIMO: Algorithms, architectures and concept systems,” in *2014 Information Theory and Applications Workshop (ITA)*, 2014, pp. 1–7.
- [123] C.-C. Wang, D. J. Love, and D. Ogbe, “Transcoding: A new strategy for relay channels,” in *55th Annual Allerton Conference on Communication, Control and Computing*, Oct. 2017.

- [124] J. M. Wozencraft and M. Horstein, "Coding for two-way channels," Massachusetts Institute of Technology, Research Laboratory of Electronics, Tech. Rep., 1961.
- [125] K. R. Narayanan and G. L. Stuber, "A novel ARQ technique using the turbo coding principle," *IEEE Communications Letters*, vol. 1, no. 2, pp. 49–51, 1997.
- [126] S. Sesia, G. Caire, and G. Vivier, "Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311–1321, 2004.
- [127] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels. I," *Information and Control*, vol. 10, no. 1, pp. 65 – 103, 1967.
- [128] Z. Zhang, K. Long, A. V. Vasilakos, and L. Hanzo, "Full-duplex wireless communications: Challenges, solutions, and future research directions," *Proceedings of the IEEE*, vol. 104, no. 7, pp. 1369–1409, 2016.
- [129] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 9, pp. 1637–1652, 2014.
- [130] C.-H. Chang and C.-C. Wang, "A new capacity-approaching scheme for general 1-to-K broadcast packet erasure channels with ACK/NACK," *to appear in: IEEE Transactions on Information Theory*, 2020.
- [131] W. Kuo and C.-C. Wang, "Robust and optimal opportunistic scheduling for downlink two-flow network coding with varying channel quality and rate adaptation," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 465–479, 2017.
- [132] J. Han and C.-C. Wang, "General capacity region for the fully connected three-node packet erasure network," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5503–5523, 2016.
- [133] Z. Ahmad, Z. Chance, D. J. Love, and C.-C. Wang, "Concatenated coding using linear schemes for Gaussian broadcast channels with noisy channel output feedback," *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 4576–4590, 2015.
- [134] C.-C. Wang and J. Han, "The capacity region of two-receiver multiple-input broadcast packet erasure channels with channel output feedback," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5597–5626, 2014.
- [135] C.-C. Wang, "On the capacity of 1-to-  $k$  broadcast packet erasure channels with channel output feedback," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 931–956, 2012.
- [136] M. Agrawal, D. J. Love, and V. Balakrishnan, "Communicating over filter-and-forward relay networks with channel output feedback," *IEEE Transactions on Signal Processing*, vol. 64, no. 5, pp. 1117–1131, 2016.
- [137] M. Agrawal, Z. Chance, D. J. Love, and V. Balakrishnan, "Using channel output feedback to increase throughput in hybrid-arq," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6465–6480, 2012.

- [138] Z. Chance and D. J. Love, "Concatenated coding for the AWGN channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6633–6649, 2011.
- [139] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, Inc., 1968.
- [140] C. Thommesen, "Error-correcting capabilities of concatenated codes with MDS outer codes on memoryless channels with maximum-likelihood decoding," *IEEE Trans. Inform. Theory*, vol. 33, no. 5, pp. 632–640, Sep. 1987.
- [141] A. Wald, "Sequential tests of statistical hypotheses," *The annals of mathematical statistics*, vol. 16, no. 2, pp. 117–186, 1945.
- [142] C.-C. Wang, "DAF of decode-&-forward," Personal Communication, Mar. 2020.