BINARY FEEDBACK IN COMMUNICATION SYSTEMS: BEAM ALIGNMENT, ADVERSARIES AND ENCODING

by

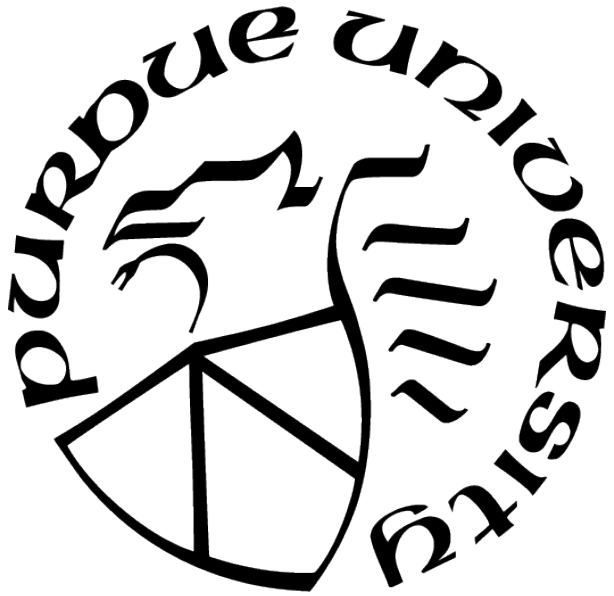
Vinayak Suresh

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. David J. Love, Chair

School of Electrical and Computer Engineering

Dr. Chih-Chun Wang

School of Electrical and Computer Engineering

Dr. James V. Krogmeier

School of Electrical and Computer Engineering

Dr. Michael D. Zoltowski

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

In loving memory of Jeykar *thatha*, Chandra *paati* and Swaminathan *thatha*

To my parents Suresh and Bama, and my twin sister Vinodha, for their many sacrifices and unconditional love

கேடில் விழுச்செல்வம் கல்வி ஒருவற்கு மாடல்ல மற்றை யவை.

Knowledge is the only form of wealth that is indestructible. - Tirukkural, 3rd century BC

ज्ञानं परमं ध्येयम् |

The supreme goal of life is to seek further knowledge.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. David J. Love for his constant support and encouragement throughout the time I have spent at Purdue. Anytime that I sit in his class or have a technical discussion with him, I am awed by his deep understanding and quick thinking. His ability to look through the *noise* and get to the heart of a problem has greatly inspired me, which I have tried to emulate both in my thinking and in writing. Graduate school also becomes a little less daunting when you know you have an advisor who wants the best for you and your future.

I would also like to thank all of my esteemed committee members - Dr. Chih-Chun Wang, Dr. James V. Krogmeier and Dr. Michael D. Zoltowski. Their candid feedback and insightful suggestions have helped me improve this thesis greatly. I owe a special debt of gratitude to Dr. Wang for his involvement in my work on adversarial channels. Answering his subtle questions has greatly improved and further cemented my own understanding of many mathematical intricacies.

In addition, I would like to thank the various funding agencies that have supported me in my Ph.D. journey, specifically, the School of Electrical and Computer Engineering at Purdue, NOKIA and the National Science Foundation [1].

I am immensely fortunate to have had two rewarding summer internships with great mentors, where I learned how theory meets the real world. My thanks to Dr. Amitava Ghosh and Dr. Jie Chen for hosting me as an intern at Nokia Bell Labs in summer 2018. My thanks also to Dr. Jianzhong (Charlie) Zhang, Dr. Ahmed Ibrahim and Dr. Yeqing Hu for their mentorship during my time as an intern at Samsung Research America in summer 2019. I am grateful also to Dr. Hemalatha Thiagarajan from NIT Trichy who inspired in me love for mathematics.

Next, I would like to thank all my colleagues at TASC lab for their encouragement and many intellectually stimulating conversations and debates. In particular, I would like to thank my collaborator Eric Ruzomberka for his involvement and for giving me the confidence

[1]↑The work in this thesis was supported in part by the National Science Foundation under grants CNS1642982, CCF1816013, CCF2008527 and EEC1941529.

to tackle difficult roadblocks. Together, we have worked on a beautiful problem of which I will always remain proud.

It is now time to thank my friends and family without whom life would be without meaning. 'Hum Homies' - Saikiran, Pia and Zubin, thanks for loving me as your brother and always standing by me. Thanks to my oldest friend Chirag, for always being there for me and supporting me. Thanks to 'Poker Buddies' - Sidharth (Oola), Sudharshan, Shreeman, Sarang, Kushagra, Smitan, Rick, Vivek and Rohit for the beautiful memories during my undergrad at NIT Trichy. Thanks Henry, Jing, Tomohiro, Dennis, Rashika, Sriram (Mongoose), Monika, Varun, Rhea, Mohit, Yash, Akansha, Parag, Radhika, Chandnee, Sukshita, Udit, Aastha, Reena, Priya, Advait, Sneha and Easwara for your wonderful friendship. Special thanks to Pinky aunty, Peru uncle, Spencer, Pranita and Priyanka - for their generous love and for treating me as their own family member.

Finally, thanks to my family back home, mummy (Bama Suresh), papa (P. J. Suresh) and Munni (Vinodha Suresh), for always loving me unconditionally despite my faults and supporting me in all my endeavours. Thanks also to my extended family, cousins, uncles and aunts for their support and best wishes.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The availability of feedback from the receiver to the transmitter in a communication system can play a significant role. In this dissertation, our focus is specifically on binary or one-bit feedback. First, we study the problem of successive beam alignment for millimeter-wave channels where the receiver sends back only one-bit of information per beam sounding. The sparse nature of the channel allows us to interpret channel sounding as a form of questioning. By posing the alignment problem as a questioning strategy, we describe adaptive (closed-loop) and non-adaptive (open-loop) channel sounding techniques which are robust to erroneous feedback signals caused by noisy quantization. In the second part, we tightly characterize the capacity for two binary stochastic-adversarial mixed noise channels. Specifically, the transmitter (Alice) intends to convey a message to the receiver (Bob) over a binary symmetric channel (BSC) or a binary erasure channel (BEC) in the presence of an adversary (Calvin) who injects additional noise at the channel's input subject to a budget constraint. Calvin is online or causal in that at any point during the transmission, he can infer the bits being sent by Alice and those being received by Bob via a feedback link. Finally in the third part, we study the applicability of binary feedback for encoding and propose the framework of linearly adapting block feedback codes. We also prove a new result for Reed-Muller (RM) codes to demonstrate how an uncoded system can mimic a RM code under this framework, against remarkably large feedback delays.

1. INTRODUCTION

In this dissertation, we study three distinct problems in communication systems where the presence of *binary* feedback plays a significant role.

1.1 Binary Feedback and Beam Alignment for Millimeter-Wave Channels

We begin in Chapter 2, where we study the problem of beam alignment for millimeter-wave (mmWave) communication. With the advent of 5G and beyond, this is an important and timely problem as mmWave technology is an essential component of most solutions that address the coverage and throughput demands of next generations wireless networks. Fully harnessing the benefits of mmWave requires high resolution channel state information (CSI) feedback in FDD systems. This however creates the problem of excessive overhead due to a large CSI payload. Motivated by this, we consider a model where the receiver sends back to the transmitter only *one bit* of feedback information about the optimal beam per channel sounding. The system model is illustrated in Fig 1.1.

It turns out that mmWave channels are generally sparse with a few dominant paths. We demonstrate that the highly directional nature of the millimeter wave channel and the binary feedback setup allows us to interpret channel sounding as a *questioning strategy*. The sounding beams correspond to *questions* (about the channel) while the 0/1 feedback bits correspond to yes/no *answers*. By exploiting this connection, we develop novel adaptive (closed-loop) and non-adaptive (open-loop) beam alignment algorithms for single dominant path channels. Here, by adaptive we mean that the beams are designed 'on-the-fly' as a function of bits received at the transmitter, while non-adaptive refers to the case where all the beams to be sounded are designed ahead of time. We also develop algorithms for multi-path channels and evaluate their performance via simulations.

1.2 Binary Feedback and Adversaries

In Chapter 3, we study the fundamental problem of characterizing capacity when communicating against a certain jammer or adversary. Different from past work, the 'main channel'

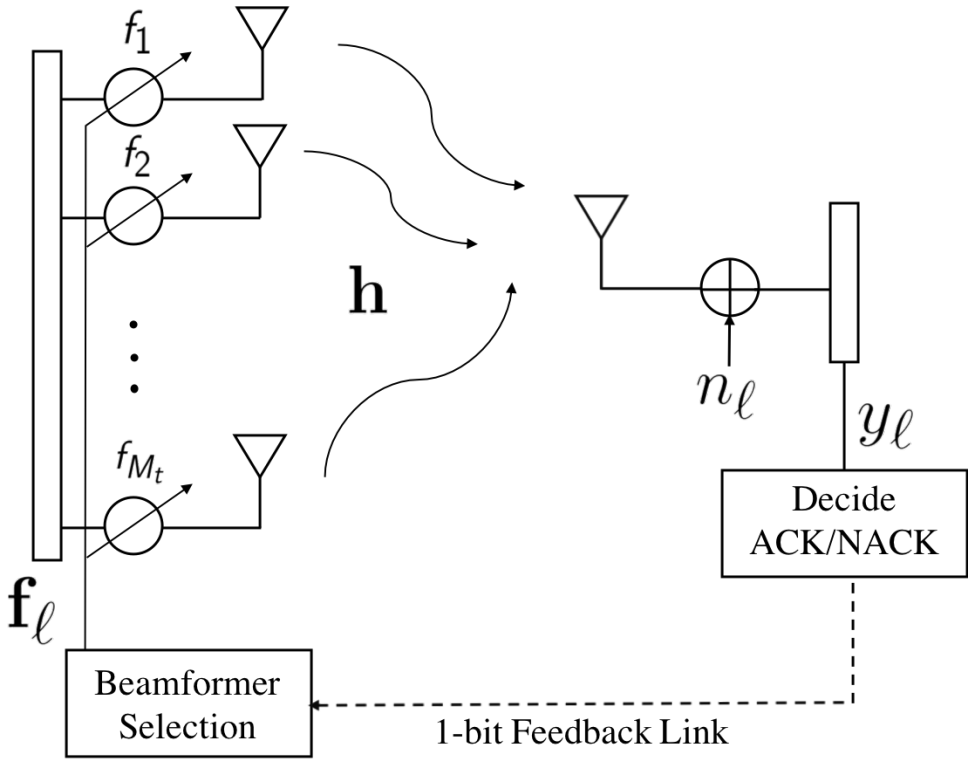


Figure 1.1. System model for a millimeter-wave MISO system with 1-bit feedback considered in chapter 2.

between the communicating parties if the adversary were removed is not perfect, instead, it is modeled as being stochastic. This is motivated by the fact that naturally occurring channels in the real world are rarely ever perfect, and often the noise process affecting the transmission can be accurately modeled as a stochastic process. Specifically, we study two binary channel models with both adversarial and random noise sources, depicted in Fig. 1.2.

Alice wishes to communicate a message reliably to Bob over a binary erasure channel (BEC(q)) or a binary symmetric channel (BSC(q)) in the presence of Calvin, who can introduce additional noise at the channel's input by erasing or flipping bits. Calvin assumes the role of an *online* jammer or adversary who has the ability to spy on *both* terminals *causally* in real time. His ability to access Bob's reception is referred to as *feedback snooping*. If $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ are the transmitted and the received codewords respectively, Calvin at any instant k knows $(x_1, x_2, \cdots, x_k)$ as well as $(y_1, y_2, \cdots, y_{k-1})$. He may not exceed a given budget constraint of pn erasures or flips but can otherwise freely

corrupt parts of the transmission. We prove in closed-form the maximum rate achievable against such a Calvin. Our converse proof involves showing an explicit attack strategy that Calvin can employ. This (optimal) attack relies crucially on the presence of binary feedback available to Calvin. To prove achievability, we resort to a random coding argument.

Interestingly, our results show that in the case of bit-flips, for any $q \in [0, 1/2)$, there is a threshold p_q such that when $p < p_q$, Calvin can do no better than inject bit-flips in an i.i.d. manner. In other words, an adversary who is weak enough is no better than an i.i.d. memoryless noise source.

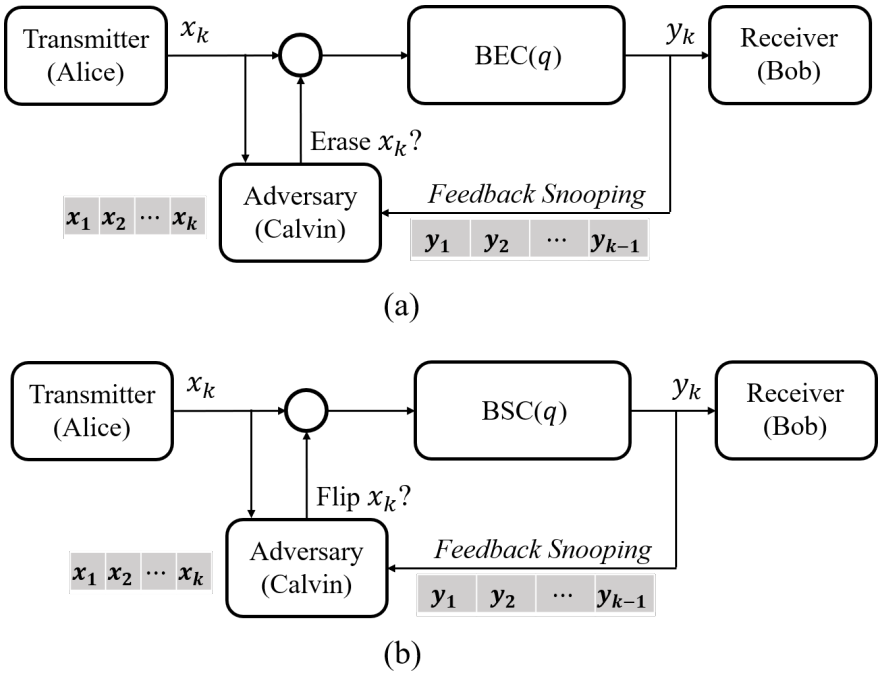


Figure 1.2. Channel models studied in chapter 3 - (a) BEC(q)-ADV(p)-FS and (b) BSC(q)-ADV(p)-FS. Calvin is constrained such that he may only inject up to pn erasures or flips.

Next, we consider an extension to these models by introducing *transmitter feedback* where Alice also has causal access to Bob's reception. Thus, Alice can potentially employ better *closed-loop* encoding strategies to overcome Calvin and achieve higher rates. Here, we give a complete capacity characterization for the case of erasures. In this case, Calvin's side information proves to be completely useless for all values of $p \in [0, 1]$ and $q \in [0, 1]$, and he can do no better than mimic an i.i.d. memoryless noise source. Finally for the case of

bit-flips with transmitter feedback, we provide partial results ending with a conjecture on the true capacity expression.

1.3 Encoding with Binary Feedback and Reed-Muller Codes

In chapter 4, we switch gears and investigate the applicability of binary feedback for encoding. Specifically, we propose *linearly adapting block feedback codes* where the feedback information is linearly processed and combined with an open loop codeword. This is partly inspired by the successes of linear feedback schemes for channels with real inputs and outputs such as the AWGN channel and its variants. We show that linear processing of noise realizations obtained causally at the transmitter (*linear nose shaping*) when XORed with an open loop codeword effectively turns one code into another. A random coding type result is proved showing that uncoded transmission combined with random linear noise shaping is capacity achieving. We also show that linear noise shaping in a closed-loop setting is intimately related to transmission of parity bits in an open-loop setting.

Finally, feedback encoding strategies are described for limited feedback - a) infrequent compressed feedback and b) delayed feedback. We prove a previously unknown property for the powerful class of Reed-Muller (RM) codes, specifically relating to their systematic generator matrix forms. It is shown that on account of this property, an uncoded system can be turned to mimic a Reed-Muller code even when the delay in the feedback link is remarkably large. Many illustrative examples are provided throughout the chapter to explain applications of our results.

2. SINGLE-BIT MILLIMETER WAVE BEAM ALIGNMENT USING ERROR CONTROL SOUNDING STRATEGIES

2.1 Introduction

It is estimated that by 2021, there will be up to 1.5 billion wireless devices with cellular connections [1]. The current efforts for 5G standardization have thus proposed for use of frequencies in the 20 to 100 GHz range commonly referred to as millimeter wave (mmWave) frequencies [2]. The millimeter wave spectrum affords extremely wide channel bandwidths (up to 1 GHz) which would provide the necessary capacity increase and enable high data rates as envisioned in 5G.

Communication at mmWave frequencies suffers from higher isotropic path loss, attenuation due to rain, reduced diffraction around obstacles, sparse scattering, and sensitivity to blockages [3]–[5]. It is thus necessary to use a large number of antennas to synthesize highly directional beams with high beamforming gain. Spatial localization of energy will require selection of a high-dimensional beamformer at the transmitter.

Transparent beam sounding is an important feature in LTE, specifically enabling advanced beamforming and coordinated multipoint [6]. One way this is accomplished is by precoding the pilot or the reference signal the same way as the accompanying data. The beamforming operation at the transmitter is then open to implementation and remains oblivious to the UE.

This work deals with the problem of feedback-assisted selection of beams for communication between two nodes operating in the mmWave band. The goal is to pick a good beam to maximize the desired performance metric while minimizing the time and resources to do so. It is well known from MIMO theory that the optimal beam achieving maximum

data throughput is a function of the channel realization. Maximum spectral efficiency is obtained when the beam picked is aligned to the channel subspace [7]. Unfortunately, the current channel realization (CSI) is not available to the transmitter apriori. The receiver must provide some auxiliary channel information in the feedback to help the transmitter ascertain the optimal beam. In legacy cellular systems, a known pilot or reference signal is sent out from each antenna element [8]. The receiver then estimates the individual per-antenna gains and conveys it back to the transmitter. Due to a large number of antennas in a millimeter wave MIMO system however, the multi-dimensional channel vectors impose a large communication overhead rendering per-atenna sampling inefficient.

Several codebook-based techniques that allow CSI acquisition without explicit channel vector estimation for mmWave use have been proposed in the literature [9]–[14]. The transmitter is equipped with a finite codebook of beamformers. In exhaustive sampling, each beam in the codebook is sounded once and the receiver feeds back the index of the best beam after all beams have been sounded. In hierarchical sampling, the transmitter and receiver jointly determine the best beam pair of a relatively coarse resolution which is further refined in successive stages. This involves the receiver feeding a locally optimal index back to the transmitter to ascertain the beams for subsequent channel sounding. This is usually accomplished by designing hierarchical subcodebooks containing beams of varying resolution.

Another popular approach is the so-called *compressive beam alignment* approach, where the channel entries are compressed into a few linear measurements using random beamforming vectors and fed back to the transmitter. In [15], [16] for example, the sounding or sensing beams are generated by applying quantized random i.i.d. phase shifts across antenna elements. The path gains and angles of departure in the downlink are then estimated by exploiting the spatial sparsity of the millimeter wave channel. Many other beam alignment strategies leveraging tools from compressed sensing have been studied extensively (see for eg. [17], [18]). These approaches typically require phase coherence between measurements meaning that the receiver needs to report both the signal magnitude and phase information in the feedback link. Alignment with magnitude only measurements is explored in [19].

In this work, we consider a model where the receiver conveys only *one bit* of information per channel sounding about the optimal beam. This ACK/NACK type of feedback can be a function of the decoded channel sounding sequence or determined via simple thresholding. This model enables transparent beam sounding - the UE does not see the actual number of transmit antennas and is not required to be informed of the beamformers used at the transmitter. Rough beam alignment using *very* low-resolution feedback (such as ARQ) may prove to be important for standardization. The system setup is shown in Figure 2.1.

Our contributions can be summarized as :

1. We demonstrate that the highly directional nature of the millimeter wave channel allows us to interpret channel sounding as a questioning strategy. The sounding beams correspond to *questions* (about the channel) while the feedback bits correspond to *answers*.

2. We investigate both adaptive (or closed-loop) and non-adaptive (or open-loop) beam alignment algorithms in this framework. In the non-adaptive algorithm, all beams to be sounded are pre-selected and are designed corresponding to a chosen error control code.

3. In the adaptive case, where the beams are selected 'on the fly', we show that the beam alignment problem ties closely with Ulam's problem well known in computer science literature [20]–[22]. We formulate new sounding signals by exploiting this connection. We then study the sounding time gap between adaptive and non-adaptive beam alignment techniques via simulations.

4. The questioning interpretation of channel sounding is also useful when multiple paths are present. Using tools developed in group testing [23]–[26], we design new sounding signals that enable the transmitter to identify the dominant channel directions. The beam for communication is then selected by training only on these directions.

5. The quantization of channel state information in a real system is noisy. This could be due to beam imperfections, fading, channel noise or interference. The alignment algorithms we propose are designed to be resilient to noisy quantization.

A preliminary version of this work was presented in 2018 [27]. The authors in [28] built upon our work to propose a non-adaptive resource-efficient design only for the case of one path. An open-loop channel estimation technique inspired by linear block codes in a different setting was described in [29]. In [30], the authors develop new techniques for quantitative group testing, and note that this may be useful for simultaneous sensing of multiple users when the number of users within a sounding beam is available as feedback to the transmitter.

The rest of the chapter is organized as follows. In Section 2.2, we explain the system model and state the problem we wish to solve. Section 2.3 and 2.4 discuss open-loop and closed-loop beam alignment algorithms respectively when a single path dominates. Section 2.5 deals with the case of multipaths. Simulation studies are presented in Section 2.6. Finally, concluding remarks are presented in Section 2.7.

Notation: All vectors unless stated are column vectors and their ℓ^2-norm is represented by $\|.\|$. $\mathcal{CN}(m, \sigma^2)$ represents a circularly symmetric complex Gaussian random variable with mean m and variance σ^2. $\mathbf{a}^*$ denotes the conjugate transpose of the vector $\mathbf{a}$. The hamming distance between two binary vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathcal{H}(\mathbf{x}, \mathbf{y})$. $\mathbb{1}(.)$ denotes the indicator function. $\lceil\ \rceil$ denotes the ceiling function while $\lfloor\ \rfloor$ is the floor function. The set of complex numbers is denoted by $\mathbb{C}$ and the set of natural numbers by $\mathbb{N}$. $GF(q)$ is the finite field with q elements where q is some power of a prime.

2.2 System Model

Consider a multiple-input single-output (MISO) millimeter wave communication system with M_t antennas at the transmitter. The receiver is assumed to have a single omni-directional antenna. The methods discussed are applicable to a multi-antenna receiver, but this is beyond the scope of the present article. The system setup is shown in Figure 2.1. A number of different beamforming architectures have been proposed in literature (see for eg. [17], [31], [32]). Sounding schemes proposed in this work can be adapted to any given architecture.

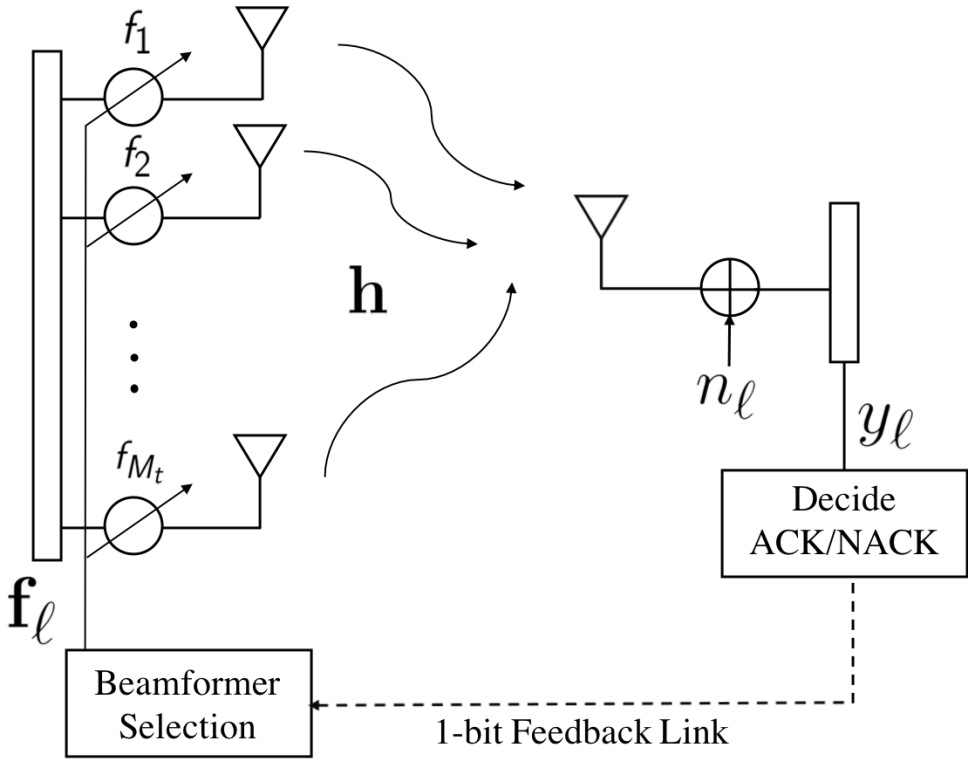


Figure 2.1. System Model for a Millimeter wave MISO system with 1-bit feedback considered in this paper.

To accomplish beam alignment, we assume that the transmitter sends a training sequence or reference signal. The receiver (or user) then processes this known signal. After processing, we model the symbol received by the receiver on the ℓ^{th} sounding interval as

$$y_\ell = \sqrt{M_t}\mathbf{h}^*\mathbf{f}_\ell + n_\ell \tag{2.1}$$

where $\mathbf{f}_\ell \in \mathbb{C}^{M_t}$ is the beamforming vector picked by the transmitter in the ℓ^{th} sounding, $\mathbf{h} \in \mathbb{C}^{M_t}$ describes the channel and $n_\ell \sim \mathcal{CN}(0, 1/\rho)$ is the noise term. ρ is the post processed SNR after the channel sounding sequence is match filtered. To restrict the total power at the transmitter, $\mathbf{f}_\ell$ is constrained to be unit norm.

The millimeter wave channel is characterized by large coherent bandwidths and a sparse scattering environment. We thus adopt a ray-based narrow-band channel model [9]. The

multi-path propagation delays are suppressed and the channel $\mathbf{h} \in \mathbb{C}^{M_t \times 1}$ with p paths is modeled as

$$\mathbf{h} = \sum_{i=1}^{p} \alpha_i \mathbf{a}(\theta_i) \tag{2.2}$$

where θ_i corresponds to the angle of departure (AoD) for the i^{th} path, $\alpha_i \sim \mathcal{CN}(0,1)$ is its complex channel gain, and $\mathbf{a}(\theta_i) \in \mathbb{C}^{M_t \times 1}$ represents the beam steering vector for direction θ_i in the transmitter's array manifold. Due to its highly directional nature, the mmWave channel has only a few dominant paths. The channel model in (2.2) with a single dominant path reduces to $\mathbf{h} = \alpha \mathbf{a}(\theta_0)$. For pedagogical reasons, we will assume a uniform linear array (ULA) at the transmitter. Extensions to planar arrays is possible. For a ULA, the unit norm beam steering vector is

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{M_t}} \left[1 \; e^{j2\pi\beta \sin\theta} \; e^{j2\pi(2)\beta \sin\theta} \cdots \; e^{j2\pi(M_t-1)\beta \sin\theta}\right]^T \tag{2.3}$$

where β is the ratio of inter-antenna spacing to wavelength.

Suppose that the desired coverage area is $\mathcal{I} = [\theta_{min}, \theta_{max}]$. The transmitter chooses an appropriate collection of T mutually disjoint intervals or bins that covers $\mathcal{I}$, labeled 1 through to T. One possible choice is uniform partitioning where all intervals are picked to have equal length. A non-uniform partitioning may be used if the base station has some prior knowledge of where the user is located. For example, it may choose finer intervals in regions where the user is more likely to be to achieve greater beam directionality. For the purpose of beam alignment, the transmitter is equipped with a beam set denoted as $\mathcal{C} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots \mathbf{a}_T\}$, where each beam $\mathbf{a}_i$ is designed to span the interval with label i. To test if the user is in a region $S \subset \{1, 2, 3, \cdots T\}$, the transmitter chooses the beamformer $\mathbf{f}$ as a normalized linear combination of beams in S according to

$$\mathbf{f}_S = \frac{\sum_{i \in S} \mathbf{a}_i}{\| \sum_{i \in S} \mathbf{a}_i \|}. \tag{2.4}$$

As noted previously, we consider a model where the receiver provides only *one bit* of information per channel sounding about the best beam. The received symbol y_ℓ is quantized to a single bit r_ℓ and fed back to the transmitter according to some rule

$$r_\ell = \Gamma_\ell(y_\ell). \tag{2.5}$$

One possible choice is to set $r_\ell = \mathbb{1}(|y_\ell|^2 > \gamma_\ell)$ where $\mathbb{1}(.)$ is the standard indicator function and γ_ℓ is the chosen threshold on the ℓ^{th} channel sounding.

The channel-normalized beamforming gain corresponding to a beamforming vector $\mathbf{f}$ is defined to be

$$A(\mathbf{f}) = \frac{|\mathbf{h}^*\mathbf{f}|^2}{\|\mathbf{h}\|^2}. \tag{2.6}$$

After receiving N bits of feedback over the N sounding intervals denoted $\{r_j\}_{j=1}^N$, the transmitter wishes to select

$$\mathbf{f}_{opt} = \arg\max A(\mathbf{f}). \tag{2.7}$$

In the case of one dominant path for example, if all bits r_ℓ were reliable and no prior information was available, the optimal strategy for the transmitter is to use a simple binary search like algorithm by successively refining the search beam width by half until the user is located to the desired beam resolution. However due to noisy quantization, some of the bits received in the feedback could be inconsistent with the receiver's actual location. We then say that these bits are in error or are erroneous. This is depicted in Fig. 2.2.

Since the feedback link carries only a single bit per feedback interval, channel sounding can be interpreted as a *questioning strategy* with yes/no answers (ACK or NACK). However, quantization error can cause some of the yes/no answers to be incorrect, which equates to lies in questioning. The general problem of searching over a finite set under different error models is a well studied problem in theoretical computer science [20]–[22], [33]. The actual number of erroneous bits in a real system is a random quantity. We thus follow a worst-case design philosophy in that our algorithms have guaranteed resilience against a given maximum number of errors.

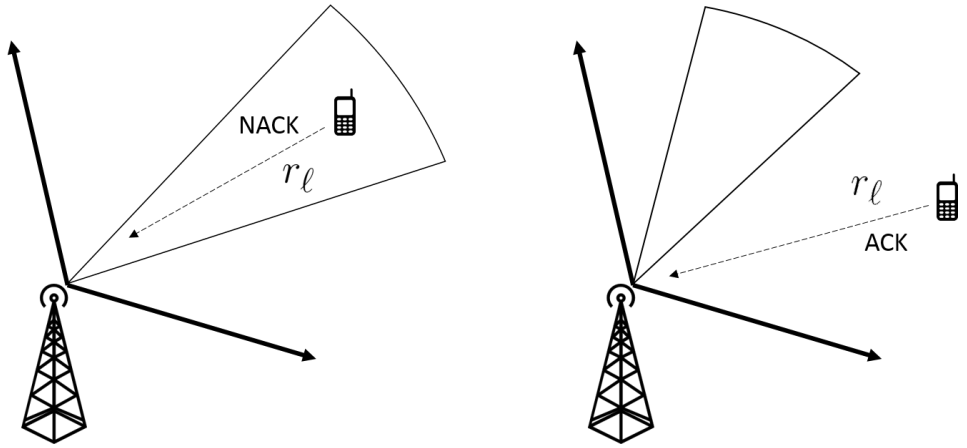


Figure 2.2. Down-link noise, beam imperfections and other factors can cause the ACK/NACK feedback to be in error.

The beam $\mathbf{f}_\ell \in \mathbb{C}^{M_t \times 1}$ picked in the ℓ^{th} sounding can be described mathematically as

$$\mathbf{f}_\ell = \mathcal{F}(\mathcal{C}, r_1, \cdots, r_{\ell-1}) \qquad \ell = 1, 2, \cdots, N, \tag{2.8}$$

a function of the beam set $\mathcal{C}$ and the previously received feedback bits $\{r_i\}_{i=1}^{\ell-1}$. This type of beam selection is referred to as closed-loop or adaptive channel sounding. The base station keeps track of bits received in the feedback to select subsequent beams in an 'online' manner.

An alternative approach with far less complexity is to sound beams agnostic to the received bits. Mathematically,

$$\mathbf{f}_\ell = \mathcal{F}(\mathcal{C}) \qquad \ell = 1, 2, \cdots, N. \tag{2.9}$$

In other words, $\mathbf{f}_\ell$ is not a function of $\{r_i\}_{i=1}^{\ell-1}$. All beamforming codewords $\{\mathbf{f}_\ell\}_{\ell=1}^{N}$ are designed 'offline' before the sounding process even begins. We refer to such a strategy as open-loop or non-adaptive channel sounding. Sections 2.3 and 2.4 discuss open-loop and closed-loop channel sounding techniques respectively for the case of one dominant path. Channel sounding for multi-path is dealt with in Section 2.5.

It is clear from Fig. 2.2 that an erroneous bit r_ℓ received in the feedback corresponds to a 'lie' in the questioning interpretation. In closed-loop channel sounding, since $\mathbf{f}_\ell$ is selected as

a function of $\{r_i\}_{i=1}^{\ell-1}$, an erroneous bit will change the subsequent beams that are sounded. On the other hand, in open-loop channel sounding, the effect of erroneous bits is felt only in post processing. For beamforming vector $\mathbf{f}$, its normalized spatial pattern as a function of the physical angle $\theta \in [-90°, 90°]$ is characterized as

$$G_{\mathbf{f}}(\theta) = |\mathbf{a}(\theta)^* \mathbf{f}|^2. \tag{2.10}$$

2.3 Open-Loop Channel Sounding

2.3.1 Algorithm Description

This section describes non-adaptive techniques for beam alignment when only one path dominates. Without loss of generality, assume $T = 2^K$. We label each of the 2^K bins in the coverage area with K bits according to a one-to-one mapping

$$\varphi : \{1, 2, 3, \cdots, T\} \longmapsto \{0, 1\}^K, \tag{2.11}$$

One choice of φ in (2.11) is simply to use the standard decimal to binary representation. This is illustrated for the case of $T = 8$ in Fig. 2.3. Let $\mathbf{m}_j$ denote the vector label for bin j and $\mathbf{m}_{dp}$ be the label that corresponds to the dominant path. This is to say the optimal beam from the codebook maximizing (2.6) is $\mathbf{a}_{\varphi^{-1}(\mathbf{m}_{dp})}$.

Notice in Fig. 2.3 that the first bit of the vector label for each of the bins in the second half of the search region is 1. In other words, a beam sounded to search the second half of the desired coverage area say $\mathbf{f}_1$ can be mapped to a question of the form : "Is the first bit of $\mathbf{m}_{dp}$ equal to 1?". After $\mathbf{f}_1$ is sounded, the receiver has access to the symbol y_1 according to (2.1) which is quantized to a single bit r_1 and fed back to the transmitter. Then, $r_1 = 0$ corresponds to a no answer while $r_1 = 1$ corresponds to a yes answer. Due to the fact that quantization is noisy, some of the r_i, $i = 1, 2, \cdots, N$, could be inconsistent with the direction of the dominant path. Hence, K *questions*, one for each bit, are not enough.

Suppose we assume that no more than L bits received at the transmitter are erroneous. It is then clear that picking the optimal beam is equivalent to determining K information

bits (of the label $\mathbf{m}_{dp}$) in the presence of up to L errors. This is the classical problem of error control coding.

Let $\mathbf{G} = [\mathbf{g}_1 \mathbf{g}_2 \cdots \mathbf{g}_N]$ be the generator matrix of a linear (N, K, d_{min}) block code over $GF(2)$ where N is the code length, K is the code's dimension and d_{min} is the minimum distance between codewords. Denote by S_ℓ the bins to be sounded in the ℓ^{th} sounding. Under the non-adaptive strategy, we select S_ℓ according to

$$S_\ell = \{j : \mathbf{m}_j \mathbf{g}_\ell = 1\}, \tag{2.12}$$

where the corresponding beam $\mathbf{f}_\ell$ is given by (2.4). The transmitter thus picks the sounding signals in accordance with the columns of the generator. Only those regions whose labels when XORed corresponding to the generator's ℓ^{th} column return 1 are picked on the ℓ^{th} channel sounding. Fig. 2.3 illustrates the shapes of the beams to be sounded for a $(6, 3, 3)$ code. The first 3 columns of the generator form the identity matrix and translate to bit-by-bit questions, one for each bit. Since the 4^{th} column is $[1\ 1\ 0]^T$, $\mathbf{f}_4$ is chosen as a linear combination of beams corresponding to labels $(010), (011), (100)$ and (101). The related *question* is, "Is the XOR of the first two bits in $\mathbf{m}_{dp}$ equal to 1?"

From (2.12), if all bits $\{r_i\}_{i=1}^N$ are consistent with the direction of the dominant path, we would have

$$r_\ell = \mathbf{I}(\varphi^{-1}(\mathbf{m}_{dp}) \in S_\ell) = \mathbf{m}_{dp} \mathbf{g}_\ell. \tag{2.13}$$

After N beams are sounded then, the received bit vector $\mathbf{r} = (r_1, r_2, \cdots, r_N)$ can be interpreted as a distorted version of the codeword $\mathbf{m}_{dp}\mathbf{G}$, corrupted due to the noisy quantization.

At the end of channel sounding, standard decoding methods from coding theory literature are applied to decode the received codeword $\mathbf{r}$ into the binary label $\hat{\mathbf{m}}$ for the dominant path. The transmitter selects

$$\mathbf{f}_{sel} = \mathbf{a}_{\varphi^{-1}(\hat{\mathbf{m}})} / \|\mathbf{a}_{\varphi^{-1}(\hat{\mathbf{m}})}\|. \tag{2.14}$$

Since a linear code with $d_{min} \geq 2L+1$ is resilient against up to L errors [34], the code in Fig. 2.3 can determine the optimal beam even if one of the six bits received in the feedback is in

error. It should be noted that one can also use non-linear codes for designing the sounding scheme.

2.3.2 Analysis

The generator matrix $\mathbf{G}$ for any (N, K, d_{min}) linear block code can be converted into what is called the systematic form by Gaussian elimination [34], meaning it has the form

$$\mathbf{G} = [\mathbf{I}_K | \, \mathbf{P}] \tag{2.15}$$

where $\mathbf{G}$ is of size $K \times N$ and $\mathbf{I}_K$ is the $K \times K$ identity matrix. This implies that the first K beams sounded at the transmitter in open-loop channel sounding can always be made to correspond to K individual bit level questions. We shall see that a parallel observation also holds in the case of closed-loop channel sounding.

One natural goal for beam alignment is to minimize the total sounding time and correspondingly the feedback overhead N. Any bounds known for open-loop codes are useful to characterize the trade-offs between N, the desired beam resolution (relates to K) and the desired degree of error correction L. Given K and L, we look for a $(N, K, 2L+1)$ code with the smallest codeword length N possible. A lower bound on N is due to the celebrated sphere-packing bound

$$2^K \left(\sum_{j=0}^{L} \binom{N}{j} \right) \leq 2^N. \tag{2.16}$$

The minimum sounding time with an open-loop beam alignment algorithm is the smallest N for which (2.16) holds.

The well-known Singleton bound [34] states that for any arbitrary block code, $d_{min} \leq N - K + 1$. Thus, if N and K are fixed, the maximum number of erroneous feedback bits L that are guaranteed to be corrected satisfies $L \leq \frac{N-K}{2}$. Codes meeting this bound are called Maximum Distance Separable (MDS) codes. These have been used extensively in data storage systems due to their excellent error correction capabilities. We can leverage them for robust open-loop beam alignment.

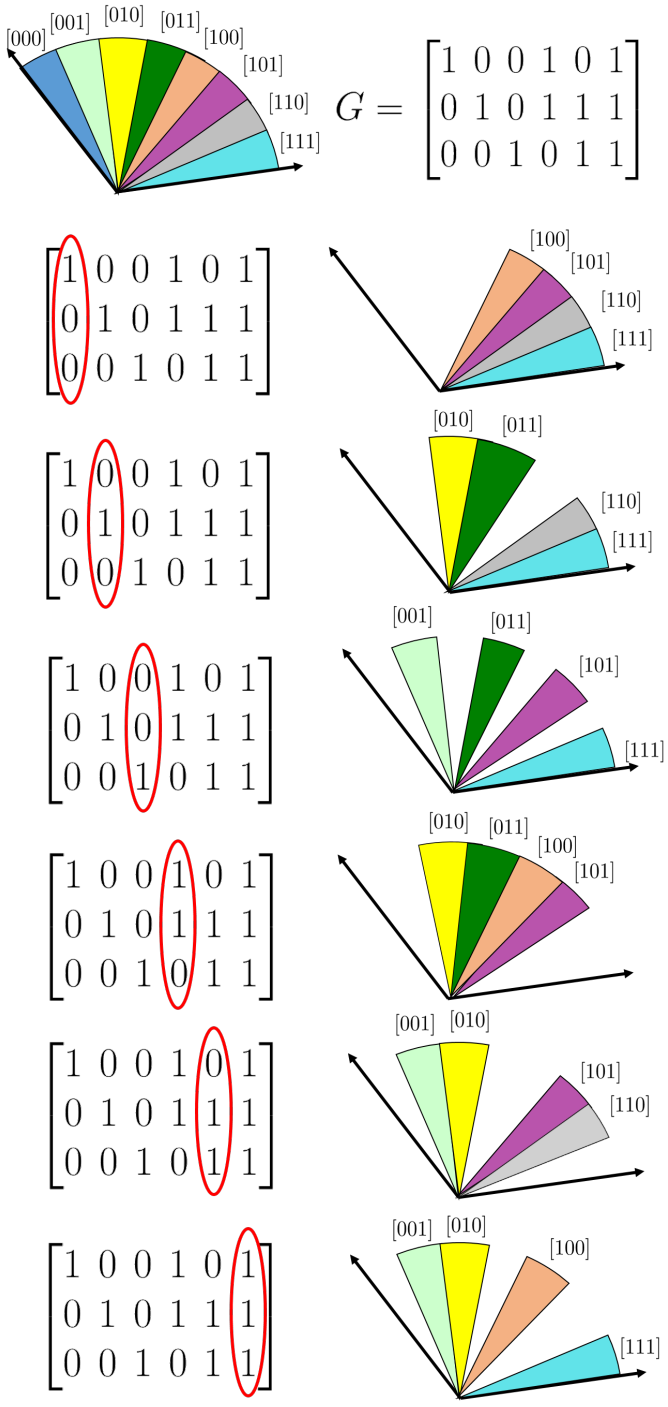


Figure 2.3. Assume $T = 2^3 = 8$, $L = 1$, N=6. The generator matrix used for the (6,3,3) code is G. The regions to be sounded are shown and correspond to the columns of the generator.

2.4 Closed-Loop Channel Sounding

In this section, we describe selection of beams as a function of previously received feedback bits as in (2.8).

2.4.1 Preliminaries

The famous mathematician S.M Ulam in his autobiography *'Adventures of a Mathematician'* [35] posed the following question: What is the minimal number of yes/no questions that one needs to determine an unknown number between one and a million if at most one or two of the answers may be lies? The generalization of this problem to distinguish between T numbers with at most L lies has since been extensively studied in the computer science literature [20]–[22] and is popularly called *Ulam's problem*.

> *Someone thinks of a number between one and one million (which is just less than* 2^{20}*). Another person is allowed to ask up to twenty questions, to each of which the first person is supposed to answer only yes or no. Obviously the number can be guessed by asking first: Is the number in the first half-million? and then again reduce the reservoir of numbers in the next question by one-half, and so on. Finally the number is obtained in less than* $\log_2(1000000)$*. Now suppose one were allowed to lie once or twice, then how many questions would one need to get the right answer?*

We can think of an adaptive channel sounding strategy for the case of one dominant path as Ulam's game between the communicating nodes. The unknown number corresponds to the bin index containing the dominant path. The channel sounding relates to questioning and the erroneously received bits relate to lies during questioning. As before, we have a total of T bins covering the region of interest $[\theta_{min}, \theta_{max}]$ and assume that no more than L bits received at the transmitter are in error.

Berlekamp was the first to develop an analytical framework to analyze Ulam's problem [36] which we outline in 2.4.2. The general idea is to assign a negative vote to bins that disagree with the received bit in a given sounding iteration. As more beams are subsequently

sounded, the 'incorrect' bins hopefully accumulate enough votes and are eventually discarded until only the correct one remains.

Suppose that the transmitter sounds a region $S \subset \{1, 2, 3, \cdots T\}$. If it receives an ACK, the bins in S^c are each assigned a negative vote and if a NACK is received, the bins in S are each assigned a negative vote. Denote A_i to be the collection of bin numbers that have received i negative votes so far. In other words, A_i contains bin numbers with a disagreement tally of i. Since we assume a maximum of L erroneous bits, any bins receiving more than L negative votes can be discarded.

The transmitter's knowledge at any point in the sounding process can thus be summarized by a collection of $L + 1$ disjoint sets $\{A_0, A_1, \cdots A_L\}$. An example of how these sets evolve as the channel sounding progresses is shown in Fig. 2.4

2.4.2 Berlekamp's Analysis

Since the sets change only by assignment of negative votes, it is enough to work with their cardinalities. Let x_j denote the cardinality of A_j and define the *n-state* to be the integer sequence $\underline{x} = (x_0, x_1, \cdots x_L) \in \mathbb{N}^{L+1}$. The integer n refers to the number of times that the transmitter is allowed to sound the channel from that point onward. With a total budget of N sounding signals then, the initial state is the N-state and the final state is the 0-state.

On sounding beams with labels in the set $S \subset \{1, 2, 3, \cdots T\}$, we define $U_i = S \cap A_i$ representing channel sounding as the vector $\underline{u} = (u_0, u_1, \cdots u_L)$, where $u_i = |U_i|$. This is equivalent to partitioning each set A_i into disjoint subsets U_i and V_i of sizes u_i and v_i and testing if the user is in the region $\bigcup_{j=0}^{L} U_j$. The initial state (the N-state) is $(T, 0, 0, \cdots 0) \in \mathbb{N}^{L+1}$.

In this formulation, the goal is to devise a strategy with minimal sounding time so that at the end of channel sounding, only one of the sets A_j is non-empty. What one would like is for the final 0-state to look like one of the following : $(1, 0, 0, \cdots, 0)$, $(0, 1, 0, 0, \cdots, 0)$, $(0, 0, 1, 0, \cdots, 0)$, $\cdots$, $(0, 0, 0, 0, \cdots, 1)$. Note also that not all sets can come up empty as that would imply more than L lies have occurred, a violation of the rules.

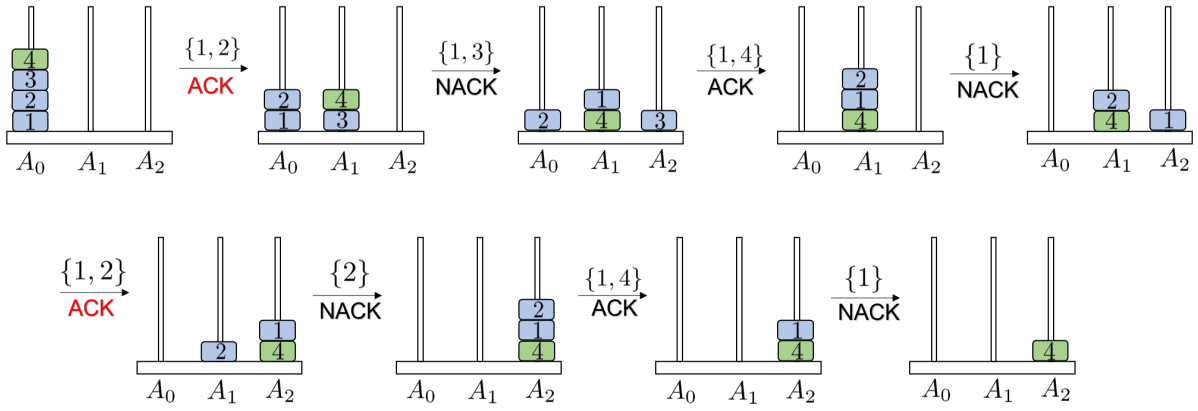


Figure 2.4. An example tracking the complete evolution of states for $K = 2$ and $L = 2$ assuming that the correct bin is the one labelled 4. The query regions are shown within braces at each step. The answers that are erroneous lies are colored red. The question at each step was selected by solving the integer program in (2.23) exactly. At the end of $N = 8$ questions, only the correct bin remains.

Berlekamp introduced the concept of "volume" for states. The volume of a n-state $\underline{x} = (x_0, x_1, \cdots x_L)$ is defined to be [36]

$$V_n(\underline{x}) = \sum_{i=0}^{L} x_i \sum_{j=0}^{L-i} \binom{n}{j}.$$

This definition is intuitively the total number of ways in which lies could possibly be distributed; for each of the x_i elements in the set A_i, there are $\binom{n}{j}$ arrangements of j erroneous feedback bits in the n remaining probes, where j takes any value from 0 to $L-i$. Berlekamp proved the following theorems:

Theorem 2.4.1. *[36] (Conservation of Volume) Let $\underline{x}$ be any non trivial n-state, and let $\underline{y}$ and $\underline{z}$ be the $(n-1)$-states that result from it following a probe that corresponds to ACK and NACK respectively. We then have*

$$V_n(\underline{x}) = V_{n-1}(\underline{y}) + V_{n-1}(\underline{z}).$$

The above theorem is a simple consequence of Pascal's combinatorial identity and the definition of volume. In words it states, no matter the question selected, volumes of the resulting ACK-state and the NACK-state add up to the volume of the state at which the question was asked.

Theorem 2.4.2. *[36] (Volume Bound) If the current n-state $\underline{x}$ is such that n sounding signals are sufficient to determine the dominant path, then $V_n(\underline{x}) \leq 2^n$.*

The initial state is $(T, 0, 0, \cdots 0) \in \mathbb{N}^{L+1}$ with volume $T\left(\sum_{j=0}^{L} \binom{N}{j}\right)$ by definition. If the transmitter has a sounding strategy that determines the bin number corresponding to the dominant path with no more than N sounding signals, the volume bound implies

$$\left(\sum_{j=0}^{L} \binom{N}{j}\right) \leq \frac{2^N}{T}. \tag{2.17}$$

Thus, (2.17) gives the a lower bound on the minimum sounding time to adaptively determine the optimal beam, no matter how the lies are distributed.

2.4.3 Adaptive Selection Of Sounding Signals

An examination of (2.17) reveals that this bound is identical to the sphere packing bound for open-loop codes in (2.16). For certain values of N, L and T, perfect error correcting codes exist that meet this bound exactly and *adaptation provably offers no benefits*. But such codes are extremely limited since it is known that any non-trivial perfect code over a finite field has the same code parameters as the Hamming code or the Golay code [34].

The Volume Bound together with the conservation of volume reveals the optimal regions to sound at any stage of the beam alignment algorithm. Suppose that the transmitter in the $(N - \ell)$ state picks $\mathbf{f}_\ell$ to sound beams with labels in a set $S_\ell \subset \{1, 2, 3, \cdots T\}$ according to (2.4). Theorems 1 and 2 imply that S_ℓ must be selected so that the resulting states that correspond to ACK and NACK have nearly the same volume. This idea of splitting into equal halves is

If in a current n-state $(x_0, x_1, \cdots x_L)$, all x_j's are even, then the selection $\underline{u} = \left(\frac{x_0}{2}, \frac{x_1}{2}, \cdots \frac{x_L}{2}\right)$ results in the two $(n-1)$-states corresponding to ACK and NACK having equal volume, for

any n. Thus any sounding strategy that is optimal begins with the same sounding signals, which is to pick half the number of elements in each of the A_i's successively as long as all terms in the state sequence are even. This observation is in parallel to (2.15). The first batch of optimal sounding signals for both adaptive and non-adaptive algorithms are simply individual bit-level questions. A simple induction argument with Pascal's identity gives the following Lemma.

Lemma 1. *Suppose that the initial state is* $(T, 0, 0, \cdots 0) \in \mathbb{N}^{L+1}$. *The resulting state after* q *beams are optimally sounded is*

$$\left(\frac{T}{2^q}\binom{q}{0}, \frac{T}{2^q}\binom{q}{1}, \cdots, \frac{T}{2^q}\binom{q}{L}\right) \tag{2.18}$$

as long as 2^q *divides* T. *If* $T = 2^K$, *the resulting state after* K *beams are optimally sounded is*

$$\left(\binom{K}{0}, \binom{K}{1}, \cdots, \binom{K}{L}\right). \tag{2.19}$$

Proof. We use an induction argument. The proposition is clearly true for $q = 1$. Since 2 divides T, the optimal beam is that which sounds half of the desired coverage region. The resulting state after the first beam is sounded is then $\left(\frac{T}{2}, \frac{T}{2}, \cdots, 0\right)$. Suppose that the proposition holds for $q = k - 1$ and that 2^k divides T. The current state is then $\left(\frac{T}{2^{k-1}}\binom{k-1}{0}, \frac{T}{2^{k-1}}\binom{k-1}{1}, \cdots, \frac{T}{2^{k-1}}\binom{k-1}{L}\right)$. The optimal beam to sound at this stage is to pick half the bins in each set. The resulting state by assigning votes and by Pascal's identity is $\left(\frac{T}{2^k}\binom{k}{0}, \frac{T}{2^k}\binom{k}{1}, \cdots, \frac{T}{2^k}\binom{k}{L}\right)$. □

The difficulty then in designing an optimal strategy is that eventually, some terms in the resulting states will be odd. It is clear that the leading terms in a state contribute most to the volume. Hence, even a unit difference between the respective leading terms of the Yes-state and the No-state will cause a large difference in their volumes. To compensate for this difference, the rest of the terms will have to be split unevenly.

A dumb strategy is to set $u_j = \left\lfloor \frac{x_j}{2} \right\rfloor$ for each $j = 0, 1, \cdots, L$ until a stage where each set $A_j, j = 0, 1, \cdots, L$ has at most one bin. Using a result from [33], we obtain an upper bound on the sounding time under this strategy.

Lemma 2. *[33] Denote $x_i(q)$ as the number of bins at level i after q probes. If we use the strategy described above, then $\forall q \geq 0$ and $j \leq q$*

$$\sum_{i=0}^{j} \left(x_i(q) - \frac{T}{2^q} \binom{q}{i} \right) \leq j + 1 \tag{2.20}$$

Theorem 2.4.3. *Denote the minimum sounding time given by the volume bound to be N_{vol}. The sounding time with the dumb strategy is no more than $N_{vol} + L^2 + L + 1$.*

Proof. After q beams are sounded, Lemma 4 implies

$$\sum_{i=0}^{L} x_i(q) \leq \sum_{i=0}^{L} \frac{T}{2^q} \binom{q}{i} + L + 1. \tag{2.21}$$

Choose the smallest $q = q_{min}$ such that

$$\sum_{i=0}^{L} \frac{T}{2^q} \binom{q}{i} < 1 \tag{2.22}$$

An inspection with equation (2.17) reveals either $q_{min} = N_{vol}$ or $q_{min} = N_{vol} + 1$. (the latter holds when (2.17) is satisfied with equality) Thus after q_{min} probes, sets A_0 through A_L collectively contain at most $L + 1$ bins. Since each probe will push at least one bin ahead, the claim holds. □

While this strategy is extremely simple, a penalty of $L^2 + L + 1$ may not be tolerable especially when L is large. We instead look for a direct attack. If $\underline{x} = \underline{u} + \underline{v}$ where $\underline{u}$ is the probe that reduces the n-state $\underline{x}$ to either the $(n-1)$-yes-state $\underline{y}$ or the $(n-1)$-no-state $\underline{z}$, we see that

$$\left| V_{n-1}(\underline{y}) - V_{n-1}(\underline{z}) \right| = \left| \sum_{j=0}^{L} \binom{n-1}{L-j} (2u_j - x_j) \right|. \tag{2.23}$$

The optimal choice of $\underline{u} = (u_0, u_1, \cdots u_L)$ is then to minimize (2.23). Fig. 2.4 demonstrates the complete sequence of states for $K = 2$ and $L = 2$ where $\underline{u}$ is always selected optimally.

The optimization problem of minimizing (2.23) is an integer linear program and known to be NP-hard. A commonly used technique to handle these type of problems is to first relax the integer constraints and solve the corresponding linear program. The solutions obtained

are then rounded to integers using methods like branch and bound or cutting planes. A comprehensive discussion of different solution techniques can be found in [37]. Alternately, the authors in[38] suggest a greedy-like approach to minimize (2.23) by accumulating one term at a time. This is summarized as Algorithm 1. The function ChooseU $(A_0, A_1, \cdots A_L, n)$ chooses the region $[U_0, U_1, \cdots U_L]$ to test, given the current n-state.

Algorithm 1 ChooseU $(A_0, A_1, \cdots A_L, n)$ [38]

$p, q \leftarrow 0$ ▷ Initialise
for $i \leftarrow 0$ to L **do**
    $\Delta \leftarrow \left| \left(p + \binom{n-1}{L-i} u_i \right) - \left(q + \binom{n-1}{L-i} (x_i - u_i) \right) \right|$
    Choose $U_i \subseteq A_i$to minimise Δ
    $p \leftarrow p + \binom{n-1}{L-i} u_i$
    $q \leftarrow q + \binom{n-1}{L-i} (x_i - u_i)$
end for
return $S = \bigcup_{j=0}^{L} U_j$

We are now ready to describe the adaptive channel sounding strategy. The sounding time budget N is fixed before. The transmitter maintains $\{A_0, A_1, \cdots A_L, n\}$ for an appropriately chosen value of L. The role that parameter L plays is that any bin receiving more than L negative votes is discarded during the sounding process. On testing a region and receiving an ACK/NACK, the sets are updated by assignment of votes. Having received bits r_1 through r_j in the feedback link, the transmitter picks $S_{j+1} =$ ChooseU $(A_0, A_1, \cdots A_L, N - j)$. Thus in sounding interval $j + 1$, transmitter sounds the beam

$$\mathbf{f}_{j+1} = \frac{\sum_{i \in S_{j+1}} \mathbf{a}_i}{\| \sum_{i \in S_{j+1}} \mathbf{a}_i \|}. \tag{2.24}$$

The most likely angular region corresponding to the path angle in the channel is the one with the least number of negative votes. Thus at the end of channel sounding, the transmitter picks A_j with the smallest index j such that it is non-empty. If A_j contains only one bin number $p \in \{1, 2, 3, \cdots T\}$, we set $\mathbf{f}_{sel} = \mathbf{a}_p$. If not, we set $\mathbf{f}_{sel} = \mathbf{a}_q$ where q is randomly chosen from A_j.

In Table 1, we compare the (worst case) sounding times for adaptive and non-adaptive beam alignment. Non-adaptive alignment is implemented with the shortest length code

Table 2.1. Sounding time Comparison for adaptive vs non-adaptive Alignment

K\L	1	2	3	4	5	6
1	3	5	7	9	11	13
2	5	8	11	14	17	20
3	6	9(10)	12(13)	15(17)	18(20)	21(24)
4	7	10(11)	13(14)	16(19)	19(22)	22(26)
5	9	12(13)	15	18(20)	21(23)	24(27)
6	10	13(14)	16(17)	19(22)	22(25)	25(29)

known for the given parameters. A table of best known codes is in [39]. The adaptive alignment is implemented via the greedy sub-optimal Algorithm 1. Table cells with a single entry are cases where the total sounding time for both algorithms coincide meaning that adaption does not provide any benefit. As a general trend, the gap between sounding time for the two techniques gets larger as either K or L increase.

2.5 Channel Sounding for Multipath

We now consider the effect of multipaths between the transmitter receiver pair. The channel model is

$$\mathbf{h} = \sum_{i=1}^{p} \alpha_i \mathbf{a}(\theta_i) \tag{2.25}$$

with parameters as defined in (2.2). As before, the desired coverage area is covered by T angular regions or bins. The beam alignment algorithm is split into two phases. In the first phase, the channel is sounded N times and the N bits received in the feedback are post-processed to identify the set $B \subset \{1, 2, \cdots, T\}$ of bin indices that correspond to path directions that are active. See Fig. 2.5 for an example. In the second phase, the transmitter sounds beams along directions specified in B to determine the optimal beam. For pedagogical reasons, we assume in this section that all of the beams in the beamset $\mathcal{C}$ are idealized beams whose gain patterns are constant in their intended support and zero elsewhere (see Fig. 2.8).

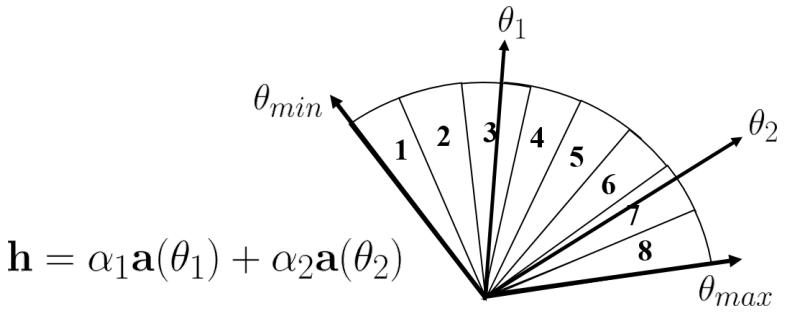


Figure 2.5. The coverage area is split into $T = 8$ regions and the beamset $\mathcal{C} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_8\}$. In this example, there are two paths that correspond to directions θ_1 and θ_2 respectively. We thus have $B = \{3, 7\}$ assuming that each beam $\mathbf{a}_i$ directs energy only in the sector labelled i.

2.5.1 Preliminaries

The beam alignment algorithms described for the single path case can no longer be applied to the multi-path scenario in a straightforward manner. In the case of open-loop beam alignment, each bin was assigned a message label $\mathbf{m}_j \in \{0,1\}^K$, $j = 1, 2, \cdots, T$. A suitable code with the generator matrix $\mathbf{G}$ was then chosen for beam selection. Suppose that there are two strong paths that correspond to bins with labels say $\mathbf{m}_1$ and $\mathbf{m}_2$. Even with perfect feedback (which means no bits in the feedback are inconsistent), the transmitter after channel sounding receives $\mathbf{r} = (r_1, r_2, \cdots, r_N)^T$ given by

$$\mathbf{r} = \mathbf{m}_1\mathbf{G} \vee \mathbf{m}_2\mathbf{G}, \tag{2.26}$$

where $\vee$ is the component-wise logical OR operation. With access to only $\mathbf{r}$ at the end of channel sounding, the identities of individual paths are completely lost and they cannot be resolved without imposing some additional constraint on $\mathbf{G}$.

In the adaptive algorithm, negative votes were assigned to individual regions with the goal of picking out one dominant path. Suppose that there are two dominant paths and we decompose $\{1, 2, \cdots, T\} = S_1 \cup S_2$ where S_1 and S_2 each contain one path. For example in Fig. 2.5, we could take $S_1 = \{1, 2, 3\}$ and $S_2 = \{4, 5, 6, 7, 8\}$. If the transmitter now sounds either $\mathbf{f}_{S_1}$ or $\mathbf{f}_{S_2}$ as in (2.4), all bins in one of these sets are assigned a negative vote. Thus, the notion of a disagreement tally is obscured.

We instead model channel sounding with one-bit of feedback as a *noisy group testing* problem with d defectives [24]–[26]. Group testing was originally introduced during World War II to detect the presence or absence of syphilitic antigen in a blood sample from a large population of samples in as few tests as possible [23]. Blood samples containing the antigen are called positives or defectives and are far fewer in number compared to the total population size. The main idea is to test groups of samples (called *pools*) together rather than test each one individually. A group testing algorithm is adaptive if the successive pools to be tested depend on the the outcomes of previously tested pools. However, much of research in this area is focused on non adaptive algorithms where all pools to be tested are decided beforehand. This is primarily since adaptive algorithms are sequential by nature and hence incur high latency, while the tests in a non-adaptive algorithm can be implemented in parallel when used for applications like blood testing. We consider here a noisy variant of the problem where the outcome of a test may be erroneous.

All vectors we deal with in this section have 0-1 entries. We say that a vector $\mathbf{p} = [p_1, p_2, \cdots, p_N]^t$ *contains* vector $\mathbf{q} = [q_1, q_2, \cdots, q_N]^t$ denoted $\mathbf{q} \preceq \mathbf{p}$, if $q_i \leq p_i$ for all $i = 1, 2, \cdots, N$. We associate channel sounding to a binary $N \times T$ test matrix $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_T\}$ with a 1 in position (i, j) if j^{th} bin is sounded on the i^{th} channel sounding and 0 otherwise. In other words, the i^{th} row of $\mathbf{Z}$ completely specifies the bins that are sounded in the i^{th} channel sounding interval. The corresponding beamformer $\mathbf{f}_i$ is selected as

$$\mathbf{f}_i = \frac{\sum_{j=1}^{T} \mathbf{a}_j \mathbb{1}(\mathbf{z}_j(i) = 1)}{\| \sum_{j=1}^{T} \mathbf{a}_j \mathbb{1}(\mathbf{z}_j(i) = 1) \|}. \tag{2.27}$$

Here, $\mathbf{z}_j(i)$ refers to the i^{th} entry in the column vector $\mathbf{z}_j$. An example of the shapes of beams to be sounded is illustrated in Fig. 2.6. We can also think of the columns of $\mathbf{Z}$ as the individual binary vector labels or *codewords* assigned to each bin.

2.5.2 Perfect Feedback

First, suppose that all of the bits $\{r_i\}_{i=1}^{N}$ are consistent with the actual path directions. In other words, none of the bits received at the transmitter are incorrect. By definition, the rows of $\mathbf{Z}$ indicate the regions where energy is directed in a particular sounding. Thus, if

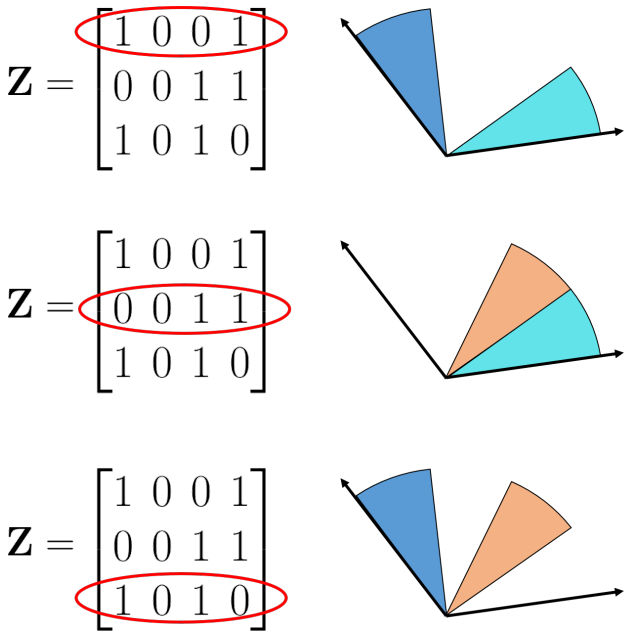


Figure 2.6. Shapes of beams to be sounded for given design matrix $\mathbf{Z}$.

the bins that correspond to the p path directions have labels $\mathbf{z}_{i_1}, \mathbf{z}_{i_2}, \cdots, \mathbf{z}_{i_p}$ (columns of $\mathbf{Z}$), we would have

$$\mathbf{r} = \bigvee_{j=1}^{p} \mathbf{z}_{i_j} \tag{2.28}$$

where $\bigvee$ represents the component-wise logical OR operation of column vectors. In practice, we may not know p exactly and instead assume $p \leq d$. (2.28) reveals how $\mathbf{Z}$ should be designed. In principle, all we need is the component-wise logical OR of every d or less columns of $\mathbf{Z}$ to be a unique vector to determine which path directions are active. However, even if we had such a $\mathbf{Z}$, determining the set B would involve an exhaustive search over a total of $\sum_{i=1}^{d} \binom{T}{i}$ possibilities. This would make the scheme impractical from an implementation perspective. To overcome this, it is common in the group testing literature to enforce the following additional structure on $\mathbf{Z}$ [24].

Def: *A matrix $\boldsymbol{Z}$ is said to be d-disjunct if the component-wise logical OR of any d or less columns does not contain any other column.*

A 1-disjunct matrix is one where no column is contained in another. An example of a 2-disjunct matrix is illustrated in Fig. 2.7. If $\mathbf{Z}$ were d-disjunct, it suffices to iterate over each of the T columns once and check the ones that are contained in the received bit

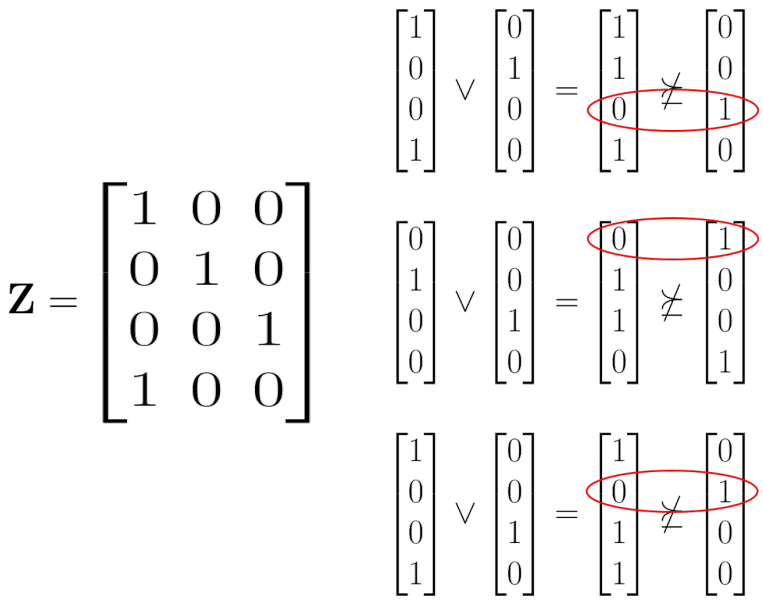


Figure 2.7. An example of a 2-disjunct matrix $\mathbf{Z}$. Clearly, no individual column contains any other column. Further, the logical OR of any 2 columns does not contain the third due to the violations that are marked in red.

vector $\mathbf{r}$ to decide which bins correspond to active path directions. This greatly reduces the implementation complexity and involves only T vector comparisons. Mathematically,

$$B = \{k \mid \mathbf{z}_k \preceq \mathbf{r}\}. \tag{2.29}$$

2.5.3 Imperfect Feedback

As noted before, the quantization of the received symbol at the receiver is noisy. As a result, the transmitter only has access to a corrupted version of $\mathbf{r}$ in (2.28), say $\mathbf{r}$. If $\mathbf{n}$ is a 0-1 noise sequence, 1 indicating an erroneously received bit, we have

$$\mathbf{r'} = \left(\bigvee_{j=1}^{p} \mathbf{z}_{i_j} \right) \oplus \mathbf{n} = \mathbf{r} \oplus \mathbf{n} \tag{2.30}$$

where $\oplus$ is the addition operation over $GF(2)$. Even a single error in the received bit vector $\mathbf{r}$ can cause a failure if we used only a d-disjunct matrix. To be resilient against up to e errors, the logical OR of one set of d (or less) columns should be at least $2e + 1$ away in

hamming distance from that of another set of d (or less) columns. We thus need special kind of d-disjunct matrices.

Def: *A d^e-disjunct matrix $\boldsymbol{Z}$ is a d-disjunct matrix with the following property: given any $d+1$ columns with one designated, there are at least $e+1$ rows with a 1-entry in the designated column and a 0-entry in the others.*

Theorem 2.5.1. *A d^e-disjunct binary matrix $\boldsymbol{Z}$ can identify up to d multipaths correctly against up to $\lfloor \frac{e}{2} \rfloor$ erroneous feedback bits.*

Proof. Let $\mathbf{P} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|\mathbf{P}|}\}$ and $\mathbf{Q} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{|\mathbf{Q}|}\}$ each be a set of d or less columns from $\mathbf{Z}$ i.e. $|\mathbf{P}|, |\mathbf{Q}| \leq d$. Denote component-wise OR of columns in $\mathbf{P}$ and $\mathbf{Q}$ as

$$\tilde{\mathbf{p}} = \bigvee_{i=1}^{|\mathbf{P}|} \mathbf{x}_i \qquad \text{and} \qquad \tilde{\mathbf{q}} = \bigvee_{j=1}^{|\mathbf{Q}|} \mathbf{y}_j \tag{2.31}$$

Choose a column column $\mathbf{c} \in \mathbf{Q} \setminus \mathbf{P}$. Since $\mathbf{Z}$ is d^e-disjunct, $\mathbf{c}$ contains a 1-entry in $e+1$ rows where all columns $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|\mathbf{P}|}$ have 0-entries. Thus, $\mathcal{H}(\tilde{\mathbf{p}}, \tilde{\mathbf{q}}) \geq e+1$.

□

The property of d^e-disjunctness in addition to providing error tolerance also allows for a very simple decoding strategy whose runtime is linear in the dimensions of the test matrix (Section 2.5.4). However, deterministic construction of such matrices is a non-trivial endeavor and a subject of intensive research. We refer the interested reader to [24], [25], [40] for specific design methodologies and a general overview on group testing literature.

In a parallel line of thought, many authors have advocated for random constructions of test matrices. One particularly simple construction is where each individual entry in the test matrix is sampled i.i.d. from a Bernoulli distribution and has been shown to perform well in practice [41]. In our simulations, we consider both kinds of designs and compare their performances under different decoding algorithms. It was shown in [42] that optimal adaptive measurements provide no gain over Bernoulli i.i.d designs in the asymptotic regime for either perfect or noisy feedback, provided that the number of defectives d grows slowly with the number of items T. We will thus only consider non-adaptive designs for the multi-path case.

2.5.4 Recovering Path Directions

Suppose that $\mathbf{Z}$ is d^e-disjunct. The matrix $\mathbf{Z}$ with binary entries completely describes the sounding signals at any stage of the algorithm. It is designed to identify up to certain number of paths with the desired error tolerance. Specifically, it can provably identify up to d paths and correct up to $\lfloor \frac{e}{2} \rfloor$ erroneously received bits. As before, the transmitter is equipped with a beam set $\mathcal{C}$. Sounding progresses according to design matrix $\mathbf{Z}$ and the one bit ACK/NACK responses are collected in a vector $\mathbf{r}$. The active path directions then need to be inferred by decoding $\mathbf{r}$ described in algorithm 2. A straightforward proof of correctness is in [43]. Note that its time complexity scales as $O(NT)$ and it returns a set B of bin indices that correspond to path directions. Since one does not know the number of errors that can occur apriori, we shall also consider other decoding methods in our simulations.

Algorithm 2 Decoding Multipath($\mathbf{Z}, \mathbf{r}$)

$B \leftarrow \emptyset$ ▷ $\mathbf{Z}$ is d^e-disjunct
for $i \leftarrow 1$ to T **do** ▷ size of $\mathbf{Z}$ is $N \times T$
  $C(\mathbf{z}_i, \mathbf{r}) \leftarrow \text{card}\left(\{j \mid \mathbf{z}_i(j) = 1 \text{ and } r_j = 0\}\right)$
  if $C(\mathbf{z}_i, \mathbf{r}) \leq \lfloor \frac{e}{2} \rfloor$ **then**
    $B \leftarrow B \cup \{i\}$
  end if
end for
return B

2.5.5 Selection Of Beamformer

For the case of multipath, we allow the transmitter to choose its beam for communication as a linear combination of a small subset of beams from the beamset. Suppose that the beamset $\mathcal{C} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{M_t}\}$ consists of M_t orthogonal beams. Having recovered the set $B = \{i_i, i_2, \cdots, i_m\}$ with a group testing approach, the beamformer $\mathbf{f}$ chosen at the transmitter is a linear combination of beams $\mathbf{a}_{i_1}$ through $\mathbf{a}_{i_m}$. The transmitter estimates the complex weights to be applied to these beams before they are summed by training only on these beams. Due to channel sparsity, $|B| \ll |\mathcal{C}|$. As a concrete example, suppose $B = \{2, 5\}$.

By sending out training beacons on $\mathbf{a}_2$ and $\mathbf{a}_5$, the transmitter obtains a good estimate of $\gamma_2 = \mathbf{a}_2^*\mathbf{h}$ and $\gamma_5 = \mathbf{a}_5^*\mathbf{h}$. The beamformer is then selected to be $\mathbf{f}_{sel} = \frac{\gamma_2\mathbf{a}_2+\gamma_5\mathbf{a}_5}{\|\gamma_2\mathbf{a}_2+\gamma_5\mathbf{a}_5\|}$.

2.6 Simulation Results

We present simulation studies to validate our proposed algorithms. It is convenient to define the spatial frequency variable $\psi = 2\pi\beta\sin(\theta) = \pi\sin(\theta)$, assuming $\beta = \frac{1}{2}$. We consider full 180° beamforming ($\theta_{min} = -90°$, $\theta_{max} = 90°$) and the beam set at the transmitter $\mathcal{C} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{M_t}\}$ has the form

$$\mathbf{a}_i = \frac{1}{\sqrt{M_t}} e^{-j\frac{(M_t-1)}{2}\psi_i} \left[1 \; e^{j\psi_i} \cdots \; e^{j(M_t-1)\psi_i}\right]^T \tag{2.32}$$

where $\psi_i = -\pi + (2i-1)\frac{\pi}{M_t}, i = 1, \cdots, M_t$. The centers ψ_i of the beams are spaced equally in the $\psi \in [-\pi, \pi]$ domain so that the resulting beams are orthonormal i.e. $\mathbf{a}_i^*\mathbf{a}_j = 0$ when $i \neq j$. While similar in form to the beam steering vector, the additional phase shift term $e^{-j\frac{(M_t-1)}{2}\psi_i}$ applied to the beams ensures that when the individual harmonics are summed together in (2.4), the resulting beam patterns have nearly flat gains in the regions of interest and low side-lobes in others. In comparison, beams from a simple DFT-type codebook work poorly due to the nulls that occur on summing of harmonics. This is illustrated in Fig. 2.8. We also define idealized beams to simplify the detector design. Let $\tilde{\mathcal{C}} = \{\tilde{\mathbf{a}}_1, \cdots, \tilde{\mathbf{a}}_{M_t}\}$ be the set of ideal beams where beam $\tilde{\mathbf{a}}_i$ has a normalized spatial pattern given by

$$G_{\tilde{\mathbf{a}}_i}(\theta) = \begin{cases} 1 & \theta \in \mathcal{G}_i \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$

where $\mathcal{G}_i = \left\{\theta : \pi\sin(\theta) \in \left[-\pi + \frac{2\pi}{M_t}(i-1), -\pi + \frac{2\pi}{M_t}i\right]\right\}$.

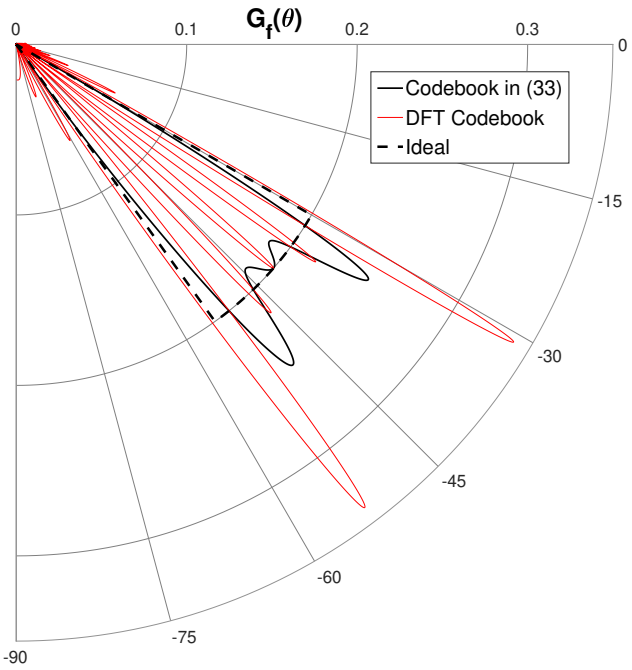


Figure 2.8. Consider $M_t = 32, \theta_{min} = -90°, \theta_{max} = 90°$. Spatial pattern $G_{\mathbf{f}}(\theta)$ of a beamformer designed to sound the region $S = \{4, 5, 6, 7, 8\}$ in (2.4) with DFT type beams vs the codebook in (2.32). Notice the maximum gain of an ideal beam is $\frac{1}{|S|} = 0.2$.

2.6.1 Single Path Channel

Consider the channel with one dominant path $\mathbf{h} = \alpha \mathbf{a}(\theta_0)$, where $\theta_0 \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ and $\alpha \sim \mathcal{CN}(0, 1)$. The optimal beam then achieves a normalized gain of $\max_i |\mathbf{a}(\theta_0)^* \mathbf{a}_i|^2$. On average, the best achievable beamforming gain for the codebook in (2.32) is equal to

$$A_{max} = \mathbb{E}_{\theta_0}\left[\max_i |\mathbf{a}(\theta_0)^H \mathbf{a}_i|^2\right] = \mathbb{E}_{\theta_0}\left[\max_i \frac{\sin^2\left(\frac{M_t(\psi_i - \pi\sin(\theta))}{2}\right)}{\sin^2\left(\frac{(\psi_i - \pi\sin(\theta))}{2}\right)}\right]. \tag{2.34}$$

For $M_t = 32$ antennas for example, the performance limit is numerically evaluated to be $A_{max} \approx 75.4\%$.

Since the detector outputs either an ACK or a NACK, we formulate the detector design problem as a hypothesis test assuming idealized beams. For practicality of our schemes, we consider a simple threshold detector of the form $r_\ell = \mathbf{I}(|y_\ell| > \gamma)$ where the decision threshold γ is chosen independent of the beamformer $\mathbf{f}_\ell$ selected and sounding iteration ℓ.

On receiving the complex symbol y_ℓ, the beam detection problem from (2.1) then reduces to the hypothesis test

$$\begin{cases} \mathcal{H}_0 : |y_\ell| \sim Rician\left(|\alpha|\sqrt{2}, \frac{1}{2\rho}\right) \\ \mathcal{H}_1 : |y_\ell| \sim Rayleigh\left(\frac{1}{2\rho}\right) \end{cases} \tag{2.35}$$

where we have assumed that the transmitter always sounds half the number of bins. In other words, in (2.4) we always have that $|S| = \frac{T}{2}$. Threshold rule γ is then selected based on the ROC curves. If an estimate of the fading gain $|\alpha|$ is not available at the receiver, the hypothesis test is formulated as

$$\begin{cases} \mathcal{H}_0 : |y_\ell| \sim Rayleigh\left(1 + \frac{1}{2\rho}\right) \\ \mathcal{H}_1 : |y_\ell| \sim Rayleigh\left(\frac{1}{2\rho}\right) \end{cases} . \tag{2.36}$$

In Fig. 2.9, the transmitter is equipped with $M_t = 32$ antennas and the given sounding time budget is $N = 16$. The feedback link is assumed to be perfectly noiseless and the errors are only due to beam detection errors.

The scheme labelled 'Bit-by-Bit' is a simple non-adaptive scheme chosen as the baseline where the first $N = 16$ columns of the matrix $[\mathbf{I}_5|\mathbf{I}_5|\cdots\mathbf{I}_5]$ are set as the generator. This yields a $(16, 5, 3)$ code and corresponds to asking questions per bit, and cycling through them repeatedly. The coded non-adaptive scheme is based on the best known open-loop code for given K and N which is the $(16, 5, 8)$ code constructed using MAGMA. A minimum distance of 8 ensures that up to 3 errors can always be corrected. The adaptive sounding scheme is implemented based on Algorithm 1 and we set the maximum number of lies parameter at $L = 3$. Thus any region receiving more than 3 negative votes is discarded during the sounding process. The natural performance metric is the channel normalized beamforming gain defined to be

$$A_{BF} = \frac{|\mathbf{h}^* \mathbf{f}_{sel}|^2}{\|\mathbf{h}\|^2} \tag{2.37}$$

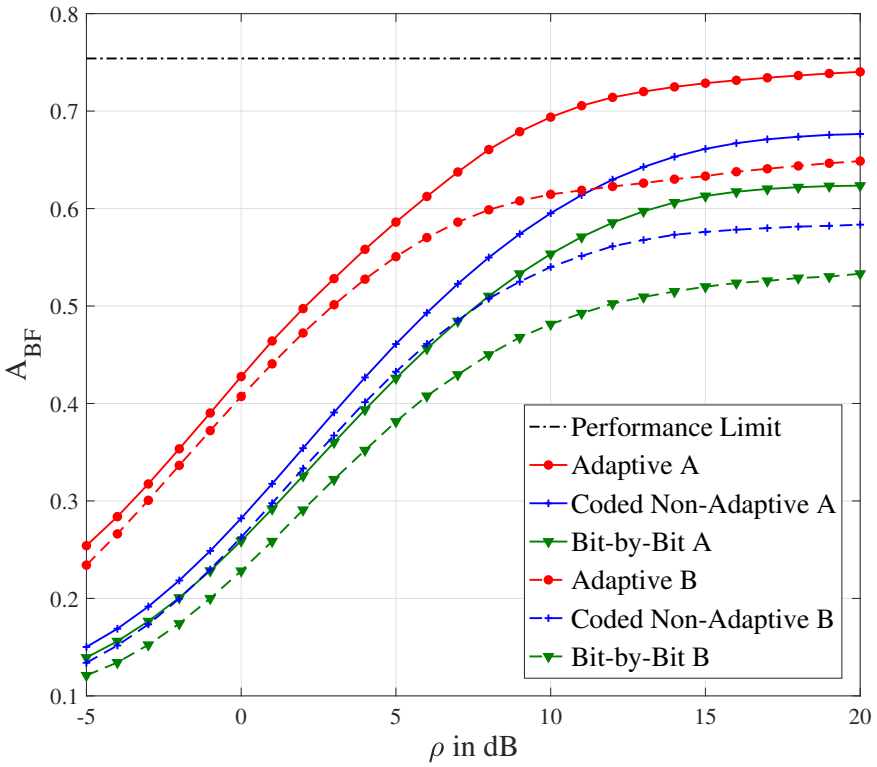


Figure 2.9. Expected channel-normalized beamforming gain for single path millimeter wave channel as a function of sounding SNR ρ. The parameters are fixed at $M_t = 32, N = 16, K = 5$ and full 180° beamforming is considered. The feedback link is assumed to be perfectly noiseless.

where $\mathbf{f}_{sel}$ is the beam selected by the transmitter at the end of the sounding process. Two sets of curves are shown for comparison, 'A' is where the detection threshold is selected according to (2.35) and 'B' where it is chosen according to (2.36).

We repeat the same set of simulations for the case where the feedback link is noisy, specifically a binary symmetric channel with probability of error $p = 0.05$. The results are indicated in Fig. 2.10. We also compare our proposed techniques to two other schemes. One is where the transmitter exhaustively sweeps over a codebook containing $T = 16$ orthonormal beams that approximately span the horizon $[-90°, 90°]$. The other is uncoded beam sounding wherein the transmitter sounds $K = 5$ beams, one for each bit, with their powers adjusted

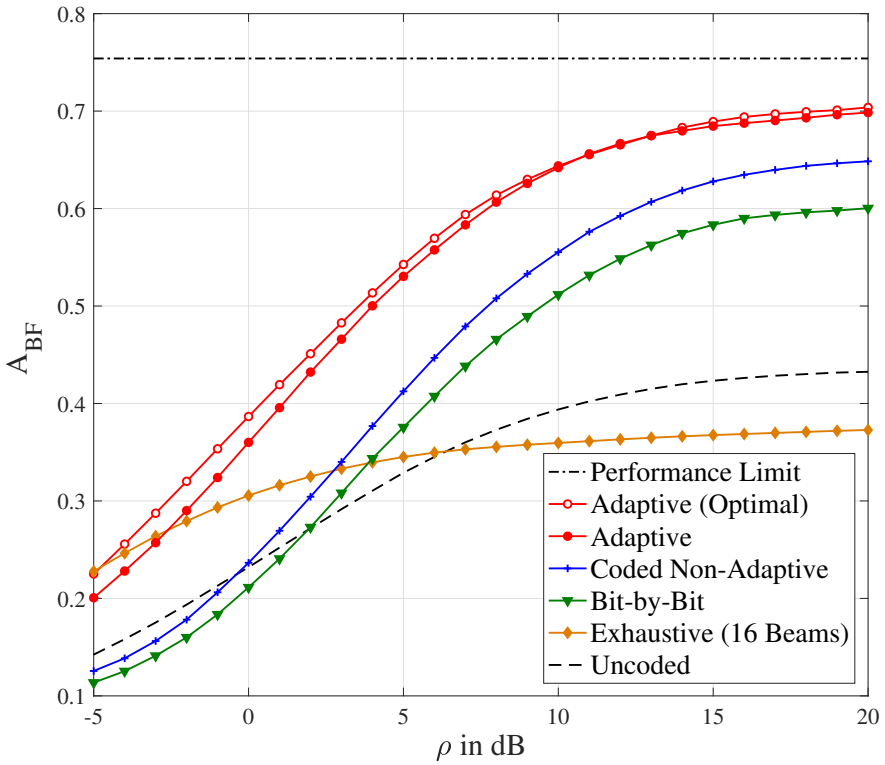


Figure 2.10. Performance when the feedback link is a binary symmetric channel with error probability 5%. Detection threshold was designed based on (2.35).

so that the total power budget across all schemes being compared is the same. We can draw the following conclusions from these simulation studies:

1. For adaptive sounding, optimal beam selection by solving the integer program in (2.23) each time performs only slightly better than greedy selection outlined in Algorithm 1. The corresponding performance curve is labelled as 'Adaptive (Optimal)' in Fig. 2.10.

2. Adaptive sounding outperforms the best non-adaptive coded strategy by 3-4 dB and the bit-by-bit scheme by up to 5 dB.

3. The proposed coded sounding schemes generally maintain good performance even when the feedback link is noisy as seen in Fig. 2.10.

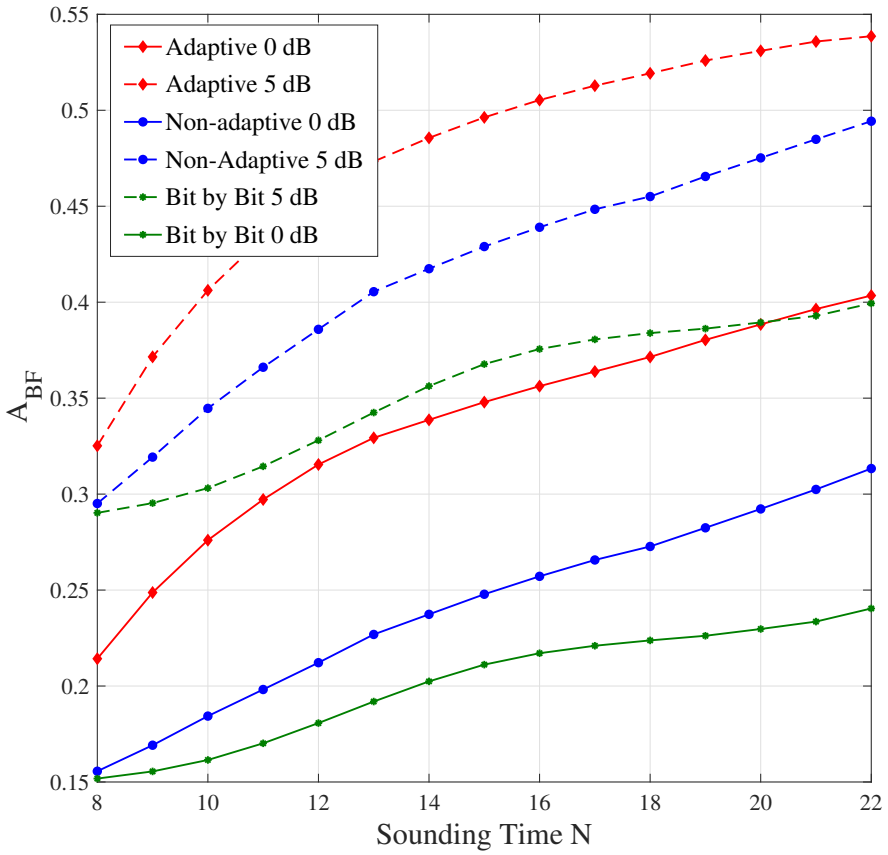


Figure 2.11. Expected channel-normalized beamforming gain for single path millimeter wave channel as a function of sounding time N. The parameters are fixed at $M_t = 32, K = 5$ and full 180° beamforming is considered. Two sets of curves, one at sounding SNR $\rho_1 = 0$ dB and the other at $\rho_2 = 5$ dB are shown.

Next, we study the performance of our algorithms as a function of the sounding time budget N. Fig. 2.12 shows two sets of performance curves for sounding SNR's fixed at $\rho_1 = 0$ dB and $\rho_2 = 5$ dB. For each N, the non-adaptive scheme was implemented by choosing the known length N code with the largest minimum distance. The adaptive scheme is based on Algorithm 1 with the threshold for all curves chosen based on (2.36). For the case of $\rho_1 = 0$ dB for example, adaptive channel sounding can achieve the same expected beamforming gain as non-adaptive schemes with just half the sounding time budget.

2.6.2 Multi-Path Channel

In this section, we consider a channel model with two paths $\mathbf{h} = (\alpha_1 \mathbf{a}(\theta_1) + \alpha_2 \mathbf{a}(\theta_2))$, where $\theta_1, \theta_2 \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ and $\alpha_1, \alpha_2 \sim \mathcal{CN}(0,1)$. We assume that the transmitter is equipped with $M_t = 512$ antennas and the sounding time is fixed at $N = 63$. The region $[-90°, 90°]$ is thus split into 512 bins, and the sounding signals for the multi-path scenario correspond to a suitably chosen group testing matrix. Similar to the single-path scenario, the detection threshold at the receiver is held constant throughout the sounding process for a given sounding SNR. At the end of channel sounding, the transmitter comes up with an estimate B of bin indices corresponding to the dominant path directions. It then performs training on beams specified by B to choose a beamformer as discussed in Section 2.5.5.

We consider two different non-adaptive designs of the sounding matrix $\mathbf{Z}$. The first design $\mathbf{Z}_1$ is a randomly generated 0-1 matrix whose each entry is i.i.d. Ber(p), where $p = 1-(2^{-\frac{1}{2}})$ is selected based on the analysis in [44]. The second test matrix $\mathbf{Z}_2$ is a carefully constructed deterministic 2^3-disjunct design based on "matrix-containment" construction techniques using the GAP software package [45]. In the notation of [46], the 63×512 submatrix of $M_2(6,4,1)$ is set as $\mathbf{Z}_2$.

Suppose that the received bit vector is **r'** in (2.30). We consider the following practical decoding strategies. The first two have a combinatorial flavour while the third is based on assuming a noise model and probabilistic.

- Noisy Combinatorial Orthogonal Matching Pursuit (NCOMP) algorithm [47]: For each column $\mathbf{z}_i$ of $\mathbf{Z}$, we define the metric $m_i \triangleq \frac{\left|\{j \mid \mathbf{z}_i(j)=1 \text{ and } r_j=1\}\right|}{\mathcal{H}(\mathbf{z}_i,\mathbf{0})}$. The indices corresponding to the two largest values of this metric are returned where ties are broken arbitrarily.

- Direct Decoding (DD) : We use Algorithm 2 for decoding for increasing values of e until the algorithm returns two indices. This is essentially picking out the two columns with the lowest values of $C(\mathbf{z}_i, \mathbf{r})$.

- (Modified) Separate Decoding of Items (SDI) algorithm: This type of decoder was first described in the Russian literature [44] and recently studied extensively in [48].

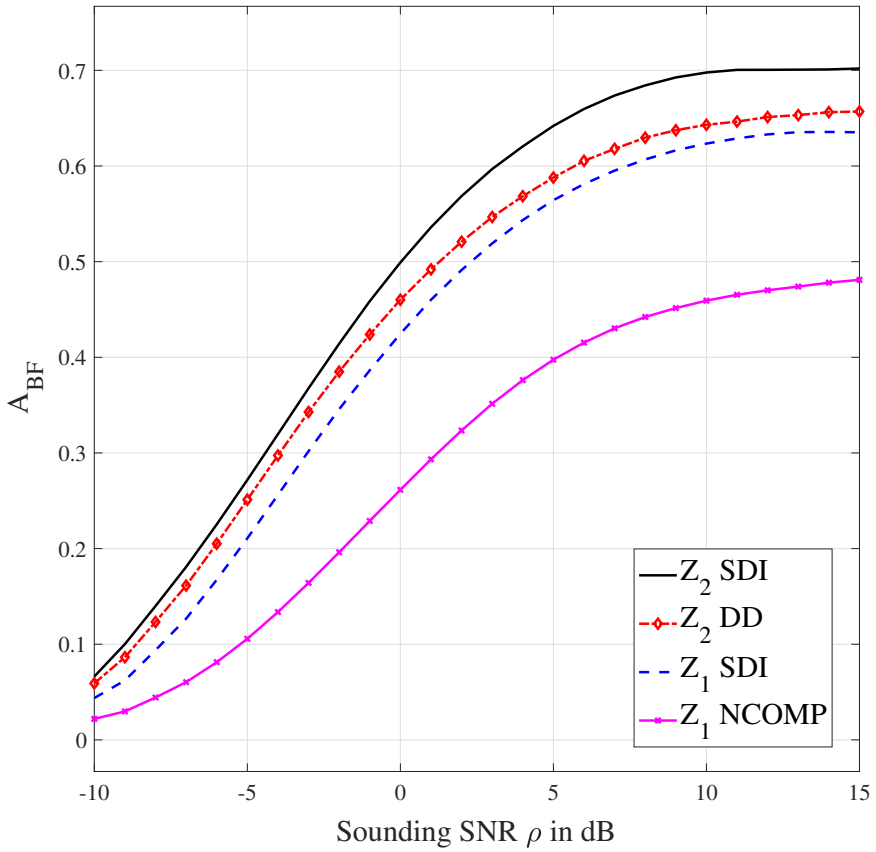


Figure 2.12. Expected channel-normalized beamforming gain for a 2-path millimeter wave channel. The parameters are fixed at $M_t = 512, N = 63$ and full 180° beamforming is considered. $\mathbf{Z}_1$ is a Bernoulli i.i.d. design while $\mathbf{Z}_2$ is a deterministic 2^3-disjunct matrix.

We assume that a bit $\mathbf{r}_i$ in (2.28) is flipped with some error probability q which we estimate via Monte-Carlo simulations. This is the so-called symmetric additive noise model. Following [48], the decoder involves computing for each column $\mathbf{z}_i$

$$\phi_i = \sum_{i=1}^{n} \ln \frac{f_1(\mathbf{r}_i, \mathbf{z}_i(j))}{f_2(\mathbf{r}_i)} \tag{2.38}$$

where $f_1(0,0) = pq + (1-p)(1-q), f_1(1,0) = 1 - f_1(0,0), f_1(0,1) = q, f_1(1,1) = 1 - f_1(0,1), f_2(0) = (1-q)(2p-p^2) + q(1-p)^2$ and $f_2(1) = 1 - f_2(0)$. The bin indices corresponding to the two largest values of ϕ_i are then returned by the algorithm. We

caution the reader that due to the final sorting step, the algorithm used here is not strictly separate decoding as defined in [48].

Fig. 2.11 shows the performance of the proposed techniques as a function of transmit SNR ρ. As one would expect, the carefully constructed deterministic design beats a random i.i.d. design for any decoding algorithm. In case of a random Bernoulli design, decoding with SDI beats NCOMP significantly which is in agreement with the simulations reported in [48]. For the case of a deterministic design, SDI provides a slight improvement over DD as it captures the probability of feedback bits being in error to make better decoding decisions. A simple Bernoulli i.i.d design is only worse by about 1.5 dB than the hard to construct deterministic design when SDI is used.

2.7 Conclusions

In this work, we studied the problem of one-bit feedback-assisted beam alignment in millimeter wave networks. By interpreting the beamforming problem as one of searching in a finite set, we investigated adaptive and non-adaptive channel sounding strategies that were designed to be robust to noisy quantization. The open-loop technique is based on standard block codes while the closed-loop technique corresponds to playing Ulam's game against a liar. We showed that it is also possible to identify multi-paths by leveraging tools from group testing.

New beam adaption techniques can potentially be formulated by exploring other error models studied in the literature such as one where the ACK/NACK is erroneous with a certain error probability. Future work includes extending the proposed techniques to the case where there are restrictions on the beams that can be sounded. Questioning strategies studied traditionally in computer science may prove to be useful for other feedback based problems in communications and signal processing.

3. THE CAPACITY OF BINARY STOCHASTIC-ADVERSARIAL CHANNELS: ONLINE ADVERSARIES WITH FEEDBACK SNOOPING

V. Suresh, E. Ruzomberka and D. J. Love, "Stochastic-Adversarial Channels: Online Adversaries With Feedback Snooping," *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021.

3.1 Introduction

Consider the following situation depicted in Fig. 3.1 - Alice wishes to communicate a message reliably to Bob over a binary erasure channel (BEC(q)) or a binary symmetric channel (BSC(q)) in the presence of Calvin, who can introduce additional noise at the channel's input by erasing or flipping bits. Calvin assumes the role of an *online* jammer or adversary who has the ability to spy on *both* terminals in real time. He may not exceed a given budget constraint but can otherwise freely corrupt parts of the transmission. Here, his budget is specified as a fraction of the codeword length (pn erasures or flips where n is the codeword length). What is the largest rate at which reliable communication is possible (i.e. *channel capacity*) in this setting? Answering this question is the central goal of this work.

Many of the channel models in information theory are broadly of two kinds. On one side are *stochastic* models whose behavior is characterized by a probability law and errors get injected independent of the communication scheme. Here, it is sufficient to deal with *average-case* errors. On the other extreme are *adversarial* models where one must deal with the *worst-case* errors. As expected, the latter often behave much differently from the former. A growing need for reliable communication over untrusted networks has sparked a renewed interest in the study of adversarial models.

In the case of adversarial channels, the capacity generally depends strongly on what the adversary knows. An *oblivious* adversary [49]–[53] is one who possibly knows the coding scheme agreed upon by Alice and Bob but has no knowledge of the transmitted codeword. In

complete contrast is the *omniscient* adversary [54]–[56] who non-causally knows the entire length-n codeword chosen by Alice for transmission. An intermediate model also considered in this paper is that of an *online* or *causal* adversary [57]–[60] wherein at any point during the transmission, the adversary has access to part of the codeword that is transmitted thus far, i.e., if $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is the codeword transmitted, Calvin at each time k knows $(x_1, x_2, \cdots, x_k)$. Another interesting set of models are the *delayed* adversary model [61], [62] and the *look-ahead* adversary model [57] where Calvin at each time k knows $(x_1, x_2, \cdots, x_{k+dn})$, where d is the delay ($d > 0$) or the look-ahead ($d < 0$) parameter. Different from these is also the *myopic* adversary model [63], [64] where Calvin knows only a noisy version of the transmitted codeword.

Along with the adversary's side information, another important criterion that affects the capacity is whether Alice and Bob have any shared randomness between them that is unknown to Calvin. In most cases, it turns out that the adversary in these settings is no worse than an i.i.d. memory-less noise source [57], [65]–[67]. Therefore in this paper, we do not allow any shared randomness between the terminals. However, we will allow Alice to employ *stochastic encoding* or randomized encoding using private random coins that are shared neither to Bob nor Calvin.

Without Calvin's presence, i.e. when $p = 0$, our models reduce to the classical BEC(q) or the BSC(q). When there is no random channel present, i.e., $q = 0$, the only source of noise is adversarial for which a complete capacity characterization is known [57]–[59]. Our models differ from the ones considered previously in two ways:

- **Mixing of random and adversarial noise:** We position this work as a study of communication limits against an adversary. However, much of the previous work in this area only considers the adversary's effect. If the adversary were absent, the channel between Alice and Bob is assumed to be perfect, which is impractical. A common approach is to model channels in nature as being stochastic. One such example is that of the additive white Gaussian noise (AWGN) channel which is known to accurately characterize many wireless channels. Therefore, in our models, we consider the effect of both i.i.d. stochastic noise and adversarial noise together.

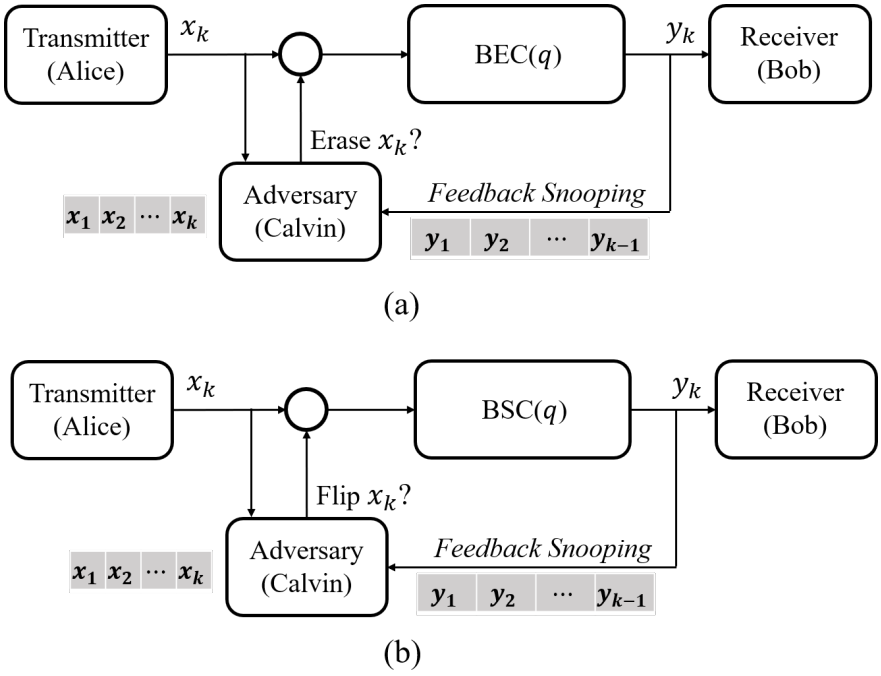


Figure 3.1. Channel models considered in this work - (a) BEC(q)-ADV(p)-FS and (b) BSC(q)-ADV(p)-FS. Calvin is constrained such that he may only inject up to pn erasures or flips.

From Fig. 3.1, the noise in the received word is affected by the random channel BEC/BSC as well as the actions of Calvin who is erasing/flipping bits. For example in the erasure case, a bit not erased by Calvin can be erased by the BEC. Similarly, in the bit-flip case, a bit flipped by Calvin may be "unflipped" by the BSC. Conceptually, we think of the stochastic channel as the main channel through which Alice and Bob communicate, and Calvin as a malicious entity who attempts to disrupt the transmission. Since we only deal with binary channels, we refer to our models as *binary stochastic-adversarial channels*. Extensions to real channels such as the AWGN channel are left for future investigation.

- **Feedback to adversary** - In our setting, we will allow Calvin access to Bob's reception through *feedback snooping*. This becomes important due to the presence of the stochastic channel that also influences the bits received at Bob. Note that feedback snooping is unnecessary when $q = 0$.

Finally, we note that our models are in fact special cases of the more general framework of arbitrarily varying channels (AVCs) [49], [68]. However, known results for AVCs do not directly imply the results of this paper and therefore we do not pursue this connection.

Our contributions can be summarized as under:

- We provide a complete characterization of capacity in the case of erasures for arbitrary budget parameter $p \in [0, 1]$ and erasure probability $q \in [0, 1]$. Our result implies that the presence of the random channel BEC(q) in addition to adversarial erasures scales the capacity expression of the $q = 0$ case by a multiplicative factor.

- We provide a complete capacity characterization in the case of bit-flips for arbitrary budget parameter $p \in [0, 1]$ and erasure probability $q \in [0, 1/2]$. Here, we show that for every $q \in [0, 1/2)$, there is a threshold $p_q > 0$ s.t. when $p < p_q$, Calvin can do no better than making flip decisions in an i.i.d. manner. In other words, a weak enough adversary is no worse than an i.i.d. memory-less noise source.

- We also consider an extension to our models where Alice has causal access to Bob's reception (*transmitter feedback*) allowing encoding to be *closed-loop*. In this scenario, we tightly characterize the capacity for erasures and provide partial results for bit-flips.

A preliminary version of this work was presented at the 2021 IEEE International Symposium on Information Theory [69]. An extended version of the ISIT conference paper with proofs is available at [70]. In [69], [70], while the capacity for the erasure model was completely characterized, only upper and lower bounds were given for the harder bit-flip model. In this work, we close this gap and show that our converse sketched in [69], [70] is in fact tight.

The rest of the paper is organized as follows. Section 3.2 formally defines the channel models and the capacity characterization problem. In Section 3.3, we state our main capacity results. Converse proofs are provided in Section 3.4 and the proofs for achievability are provided in Section 3.5. Transmitted feedback is considered in Section 3.6. Finally, conclusions and future research directions are given in Section 3.7.

3.2 Preliminaries

3.2.1 Channel Models

The channel models are depicted in Fig. 3.1. We first describe the model for the case of erasures. Alice (the transmitter) attempts to convey a message to Bob (the receiver) over a BEC(q), in the presence of a p-limited causal adversary (Calvin) where the terms will be clarified shortly. The input and output alphabets are $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, 1, \Lambda\}$, respectively, where Λ denotes an erasure symbol. Encoding is done over n channel uses, and the size of the message set at the transmitter is 2^{nR}. We allow stochastic encoding and assume the presence of private or local randomness available only to Alice for this purpose. Denote $x_k \in \mathcal{X}$ to be the bit selected by the transmitter at channel use k. At time k, the adversary makes a decision on whether to erase x_k based on his side-information to be specified shortly. If Calvin erases x_k, the received symbol at time k at the receiver is an erasure, i.e., $y_k = \Lambda$. If Calvin decides not to erase x_k, then

$$y_k = \begin{cases} \Lambda & \text{with prob. } q \\ x_k & \text{with prob. } 1 - q \end{cases},$$

i.e., x_k is erased with probability q. We now specify the side-information available to Calvin:

- **Knowledge of transmission Scheme:** Calvin has knowledge of the transmission scheme agreed upon by Alice and Bob. In the case of deterministic encoding, Calvin knows the mapping between the set of messages and the codewords while in the case of stochastic encoding, he knows the codeword distribution selected for each message.

- **Transmitter Snooping:** Calvin has causal access to symbols being transmitted by Alice, i.e., at each channel use k, $1 \leq k \leq n$, Calvin knows $(x_1, x_2, \cdots, x_k) \in \mathcal{X}^k$.

- **Feedback Snooping:** Calvin has the capability to spy into Bob's reception through a noise-free *strictly* causal feedback link as shown in Fig. 3.1. At each channel use k, $1 \leq k \leq n$, Calvin knows $(y_1, y_2, \cdots, y_{k-1}) \in \mathcal{Y}^{k-1}$.

Thus, Calvin's decision on whether or not to erase x_k is a function of the encoding rule, $(x_1, x_2, \cdots, x_k) \in \mathcal{X}^k$ and $(y_1, y_2, \cdots, y_{k-1}) \in \mathcal{Y}^{k-1}$. A power constraint is further imposed by enforcing Calvin to be p-limited, meaning that he can erase at most a constant fraction p of the bits, i.e., if $\mathbf{a} \in \{0, \Lambda\}^n$ denotes the positions where Calvin decides to erase symbols from $(x_1, x_2, \cdots, x_n)$, we must have $weight(\mathbf{a}) \leq pn$. We refer to this model as *the BEC causal adversarial channel with feedback snooping* (or BEC(q)-ADV(p)-FS). Note that the BEC block in Fig. 3.1(a) is slightly different from the classical BEC. If Calvin erases x_k to an erasure symbol Λ, we have $y_k = \Lambda$, where Λ does not carry any information.

We also consider a related and more interesting model (Fig. 3.1(b)) where Calvin can attempt to flip up to pn bits and the stochastic channel is a BSC(q) instead of a BEC(q). The input and output alphabets are revised to $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, 1\}$. At time k, Calvin produces $a_k \in \mathcal{A} = \{0, 1\}$ based on his side information which is the same as was for erasures, i.e., at time k, he knows $(x_1, x_2, \cdots, x_k)$, the codebook or the codeword distribution, and $(y_1, y_2, \cdots, y_{k-1})$. The received symbol at time k at the receiver is

$$y_k = \begin{cases} x_k \oplus a_k \oplus 1 & \text{with prob. } q \\ x_k \oplus a_k & \text{with prob. } 1 - q \end{cases},$$

where $\oplus$ denotes mod-2 addition and $q \in [0, 1/2]$. Hence, $\mathbf{a} \in \{0, 1\}^n$ denotes the positions where Calvin injects bit-flips and the constraint on the adversary can be expressed as $weight(\mathbf{a}) \leq pn$. Note that a flip-attempt of Calvin can now be undone by the BSC. This happens exactly at positions where both Calvin and the BSC inject errors. This is in contrast to the case of erasures where a bit erased by Calvin remains erased. This model is referred to as *the BSC causal adversarial channel with feedback snooping* (or BSC(q)-ADV(p)-FS).

Our aim is to characterize the capacity of these channels, i.e., the largest value of R such that Alice can reliably convey one out of 2^{nR} possible messages to Bob. The capacities of the BEC(q)-ADV(p)-FS channel and the BSC(q)-ADV(p)-FS channel are denoted by $C^E(p, q)$ and $C(p, q)$ respectively. Precise definitions to follow.

Notation and Definitions: In this work, we only consider fixed length encoding. The blocklength is denoted by n. The transmitted message is denoted by the random variable

(r.v.) $\mathbf{U}$ chosen uniformly from the message set $\mathcal{U} = \{1, 2, 3, \cdots, 2^{nR}\}$. We denote by $\mathcal{C}(n, R)$ a code of rate R and block-length n. A deterministic code consists of a fixed encoder map $\Phi_d : \mathcal{U} \to \mathcal{X}^n$ and a decoder map $\Gamma_d : \mathcal{Y}^n \to \mathcal{U}$, where each message is associated to a unique codeword. In case of stochastic encoding, a codeword $\mathbf{x}$ is selected for a message u according to a chosen conditional distribution $\tilde{\Phi}(.|u)$ defined on $\mathcal{X}^n$. A stochastic code is fully specified by defining all conditional distributions $\left\{\tilde{\Phi}(.|u)\right\}_{u\in\mathcal{U}}$ and a decoder $\Gamma : \mathcal{Y}^n \to \mathcal{U}$. Without loss of generality, we assume in proving converse results that no two distinct messages map to the same codeword. The (maximum) probability of error is then

$$P_e = \max_{u\in\mathcal{U}} \max_{\mathrm{ADV}(p)} \sum_{\mathbf{y}} \sum_{\mathbf{x}} \mathbb{P}(\mathbf{y}|\mathbf{x})\tilde{\Phi}(\mathbf{x}|u)\mathbb{1}(\Gamma(\mathbf{y}) \neq u) \tag{3.1}$$

where $\mathbb{1}(.)$ denotes the indicator function and $\mathrm{ADV}(p)$ denotes a feasible strategy chosen by Calvin. Note that $P(\mathbf{y}|\mathbf{x})$ in (3.1) is a function of both the stochastic channel and the chosen adversarial strategy.

When proving achievability results, we consider for simplicity the following alternate form of a stochastic code: Alice is endowed with a set $\mathcal{S}$ of private secrets or keys and the stochastic code is defined by a deterministic map $\Phi : \mathcal{U} \times \mathcal{S} \to \mathcal{X}^n$. For a given message $u \in \mathcal{U}$, the codeword $\Phi(u, s)$ is selected by uniformly picking a secret $s \in \mathcal{S}$. As discussed in [58], this definition does not change the capacity. In this case, the (maximum) probability of error from (3.1) is revised to

$$P_e = \max_{u\in\mathcal{U}} \max_{\mathrm{ADV}(p)} \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} \sum_{\mathbf{y}} \mathbb{P}(\mathbf{y}|\Phi(u, s))\mathbb{1}(\Gamma(\mathbf{y}) \neq u). \tag{3.2}$$

Note in (3.2) that the probability of decoding error is averaged over all possible secrets available to Alice for encoding. We say that $R > 0$ is achievable if for every $\epsilon > 0$, there is a sequence of rate $R - \epsilon$ codes of increasing block-lengths $\{\mathcal{C}(n, R - \epsilon)\}_{n\geq 1}$ such that for any $\delta > 0$, there is an N so that $P_e(\mathcal{C}(n, R - \epsilon)) < \delta$ for any $n > N$. The capacity is defined to be the supremum of all achievable rates.

For $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, we let $\mathbf{x}_i^j = (x_i, x_{i+1}, \cdots, x_j)$. Denote by $d_H(\mathbf{x}, \mathbf{y})$ the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$. Let $\mathrm{Ber}(q)$ denote a Bernoulli r.v. with success probability q.

For $x, y \in [0, 1/2]$, we define $x \star y = x(1-y) + y(1-x)$. The following lemma will prove useful for the bit-flip model.

Lemma 3. *Let $x, y \in [0, 1/2]$. Then, $x \star y = 1/2$ iff either $x = 1/2$ or $y = 1/2$ (or both).*

Proof. If either $x = 1/2$ or $y = 1/2$ (or both) it is easy to see that $x \star y = 1/2$. Suppose that $x, y \in [0, 1/2)$. Note then that

$$x \star y = x(1-y) + y(1-x) = x(1-2y) + y < \frac{1}{2}(1-2y) + y = \frac{1}{2}.$$

□

Remark. The cascade of channels BSC(x) and BSC(y) is equivalent to the BSC with bit-flip probability $x \star y$. This implies the well-known result that it is possible to communicate at a non-zero rate through a cascade of two BSC's unless at least one is completely noisy.

3.2.2 Simple Converse Bounds - The i.i.d. Attack

In this section, we describe a simple adversarial attack for Calvin wherein no side information is required. Calvin simply simulates an i.i.d. noise source as follows:

- For the BEC(q)-ADV(p)-FS channel, Calvin erases each bit x_i independently with probability (approximately) p. Since the combination of this attack with the $BEC(q)$ is the $BEC(s)$ with error probability $s = p + q - pq$, the capacity is bounded as

$$C^E(p, q) \leq (1-p)(1-q).$$

- For the BSC(q)-ADV(p)-FS channel, Calvin flips each bit x_i independently with probability (approximately) p. Since the combination of this attack with the $BSC(q)$ is the $BSC(p \star q)$, the capacity is bounded as

$$C(p, q) \leq 1 - h_2(p \star q). \tag{3.3}$$

The proofs rely on the fact that under the above attacks, the power constraint is respected with high probability.

As we show in this work, for the BEC(q)-ADV(p)-FS channel, the i.i.d. erasure attack is *always* sub-optimal. In contrast however, for the BSC(q)-ADV(p)-FS channel, there are regimes where the i.i.d. bit-flip attack is optimal. Here, the side information available to Calvin as described in 3.2.1 proves to be of no benefit, and Calvin is no worse than an i.i.d. Bernoulli noise source.

3.2.3 Effective number of erasures or flips

By the Chernoff bound, the BEC(q)/BSC(q) when acting alone induces about qn erasures/flips. Here, we also have Calvin who can introduce up to pn additional erasures/flips. However, since Calvin is causal, his error pattern and the error pattern induced by the random channel may have several overlapping error injections. The total number of errors will thus be much less than $pn + qn$.

Let $\delta > 0$ be a small arbitrary constant. We present two lemmas showing that, under any strategy employed by Calvin, we have the following:

- For the BEC(q)-ADV(p)-FS channel, the total effective number of erasures injected on to the received codeword due to actions of both Calvin and the BEC(q) w.h.p. does not exceed $(p + q - pq + \delta)n$.
- For the BSC(q)-ADV(p)-FS channel, the total effective number of flips injected on to the received codeword due to actions of both Calvin and the BSC(q) w.h.p. does not exceed $(p \star q + \delta)n$.

The above is also true for a Calvin simulating a Ber(p) noise source injecting i.i.d. random erasures/flips. However, while the number of flips is approximately the same in the two cases, their positions may indeed be very different.

Lemma 4. *Let $X_1, X_2, \cdots, X_n$ be i.i.d. Ber(q) indicator random variables representing the erasure sequence injected by the BEC(q). Let Calvin's erasure injections be represented by the inidicator random variables $Y_1, Y_2, \cdots, Y_n$ where for each j, Y_j is Bernoulli distributed*

with success probability that is possibly a function of $X_1, X_2, \cdots, X_{j-1}, Y_1, Y_2, \cdots, Y_{j-1}$*, such that the random variable* $\sum_j Y_j$ *is almost surely less than or equal to* pn *(adversary power constraint). For* $\delta > 0$*, defining the event*

$$E = \left\{ \sum_{j=1}^{n} \mathcal{I}(X_j = 1 \text{ or } Y_j = 1) \leq (p + q - pq)n + \delta n \right\},$$

we have $P(E) \geq 1 - 2^{-\Omega(\delta^2 n)}$.

Proof. Let $Z_j = \mathcal{I}(X_j = 1 \text{ or } Y_j = 1)$. Define

$$P_j = Z_j - \mathbb{E}(Z_j \mid X_1, X_2, \cdots, X_{j-1}, Y_1, Y_2, \cdots, Y_{j-1})$$

and

$$S_k = \sum_{j=1}^{k} P_j, \quad k = 1, 2, \cdots, n.$$

Clearly, S_j is a martingale because

$$\mathbb{E}(S_{j+1} \mid X_1, \cdots, X_j, Y_1, \cdots, Y_j) = S_j.$$

We now apply Azuma's inequality. Note that $|S_j - S_{j-1}| = |P_j| \leq 1$ holds almost surely. Thus, by Azuma's inequality,

$$Pr\left(|S_n| \geq \delta n\right) \leq 2 \exp\left(-\frac{\delta^2 n}{2}\right). \tag{3.4}$$

Re-examining S_n, we have

$$S_n = \sum_{j=1}^{n} Z_j - \sum_{j=1}^{n} \mathbb{E}(Z_j | X_1, \cdots, X_{j-1}, Y_1, \cdots, Y_{j-1}),$$

where, it is easy to see that

$$\sum_{j=1}^{n} \mathbb{E}(Z_j | X_1, \cdots, X_{j-1}, Y_1, \cdots, Y_{j-1}) = \mathbb{E}\left(\sum_{k=1}^{n} Z_k\right).$$

We have also

$$\begin{aligned}
\mathbb{E}\left(\sum_{k=1}^{n} Z_k\right) &= \sum_{k=1}^{n} P(X_j = 1 \text{ or } Y_j = 1) \\
&= \sum_{k=1}^{n} 1 - P(X_k = 0, Y_k = 0) \\
&\overset{(a)}{=} n - (1-q)\sum_{k=1}^{n} P(Y_k = 0) \\
&= qn + (1-q)\mathbb{E}\left(\sum_{k=1}^{n} \mathcal{I}(Y_k = 1)\right) \\
&\overset{(b)}{\leq} (p + q - pq)n,
\end{aligned} \tag{3.5}$$

where (a) holds due to the independence of X_j and Y_j, and (b) is a consequence of the power constraint on Calvin. The required result then follows from (3.4) and (3.5). □

Lemma 5. *Let $X_1, X_2, \cdots, X_n$ be i.i.d. Ber(q) random variables representing the error sequence injected by the BSC(q). Let Calvin's error injections be represented by the random variables $Y_1, Y_2, \cdots, Y_n$ where for each j, Y_j is Bernoulli distributed with success probability that is possibly a function of $X_1, X_2, \cdots, X_{j-1}, Y_1, Y_2, \cdots, Y_{j-1}$, such that the random variable $\sum_j Y_j$ is almost surely less than or equal to pn (adversary power constraint). For $\delta > 0$, defining the event*

$$E = \left\{\sum_{j=1}^{n}(X_j \oplus Y_j) \leq (p \star q)n + \delta n\right\},$$

we have $P(E) \geq 1 - 2^{-\Omega(\delta^2 n)}$.

Proof. Let $Z_j = X_j \oplus Y_j$. Proceeding exactly as in the proof of Lemma 4, define

$$P_j = Z_j - \mathbb{E}(Z_j \mid X_1, X_2, \cdots, X_{j-1}, Y_1, Y_2, \cdots, Y_{j-1})$$

and

$$S_k = \sum_{j=1}^{k} P_j, \quad k = 1, 2, \cdots, n.$$

Again, S_j is a martingale because

$$\mathbb{E}(S_{j+1} \mid X_1, \cdots, X_j, Y_1, \cdots, Y_j) = S_j.$$

We have $|S_j - S_{j-1}| = |P_j| \leq 1$ and thus, by Azuma's inequality,

$$Pr\left(|S_n| \geq \delta n\right) \leq 2\exp\left(-\frac{\delta^2 n}{2}\right), \tag{3.6}$$

where

$$S_n = \sum_{j=1}^{n} Z_j - \mathbb{E}\left(\sum_{j=1}^{n} Z_j\right).$$

Here,

$$\begin{aligned}
\mathbb{E}\left(\sum_{k=1}^{n} Z_k\right) &= \mathbb{E}\left(\sum_{k=1}^{n}(X_k \oplus Y_k)\right) \\
&= \sum_{k=1}^{n} P(X_k \oplus Y_k = 1) \\
&= \sum_{k=1}^{n} \left(qP(Y_k = 0) + (1-q)\mathbb{P}(Y_k = 1)\right) \\
&= qn + (1-2q)\mathbb{E}\left(\sum_{k=1}^{n} Y_k\right) \\
&\leq (p \star q)n,
\end{aligned}$$

and therefore the result follows from (3.6). □

3.3 Main Results

3.3.1 Results for Erasures

The capacity characterization for the BEC(q)-ADV(p)-FS channel is given by the following theorem.

Theorem 3.3.1. *The capacity $C^E(p,q)$ of BEC(q)-ADV(p)-FS is given by*

$$C^E(p,q) = \begin{cases} (1-2p)(1-q) & \text{for } 0 \leq p \leq \frac{1}{2},\ 0 \leq q \leq 1 \\ 0 & \text{otherwise} \end{cases}. \tag{3.7}$$

When there is no BEC, i.e., when $q = 0$, our model reduces to the one studied in [57], [58]. Our result implies that in the setting where both causal adversarial erasures and random erasures are present, the capacity expression is scaled by a factor of $1 - q$.

3.3.2 Results for Bit-flips

The capacity characterization for the BSC(q)-ADV(p)-FS channel is given by the following theorem.

Theorem 3.3.2. *For $p \in [0, 1/4]$ and $q \in [0, 1/2]$, the capacity $C(p, q)$ of BSC(q)-ADV(p)-FS is*

$$C(p,q) = \min_{\bar{p}:\bar{p}\in\mathcal{P}} \alpha(p,\bar{p}) \left(1 - h_2\left(\frac{\bar{p}}{\alpha(p,\bar{p})} \star q\right)\right). \tag{3.8}$$

$$\alpha(p,\bar{p}) = 1 - 4(p - \bar{p}) \quad , \quad \mathcal{P} = \{\bar{p} : 0 \le \bar{p} \le p\}$$

If $p \ge 1/4$, we have $C(p, q) = 0$.

When $q = 0$, i.e., there is no BSC, the channel model reduces to that considered in [58], and the capacity expression (3.8) matches with the result proved in [58]. As we will show shortly, the solution $C(p, q)$ to the optimization problem in (3.8) has the following form:

- For a fixed $q \in [0, 1/2)$, $C(p, q) > 0$ for all $p \in [0, 1/4)$. Thus, the addition of the BSC stochastic channel does not change the support over p for which the a positive rate is achievable.

- For a fixed $q \in [0, 1/2)$, there is a $p_q \in (0, 1/4)$ such that for $0 \le p \le p_q$, $C(p, q)$ is convex and equal to $1 - h_2(p \star q)$. This implies that when $0 \le p \le p_q$, the i.i.d. bit-flip attack strategy in Section 3.2.2 is optimal for the adversary. In this regime, the knowledge of the encoding scheme or the ability to spy on Alice or Bob buys Calvin no benefit.

- It can be shown that the value of p_q is the unique solution (different from $1/2$) of the equation

$$4 + (1 + 2q)\log_2(p_q \star q) + (3 - 2q)\log_2(1 - p_q \star q) = 0. \tag{3.9}$$

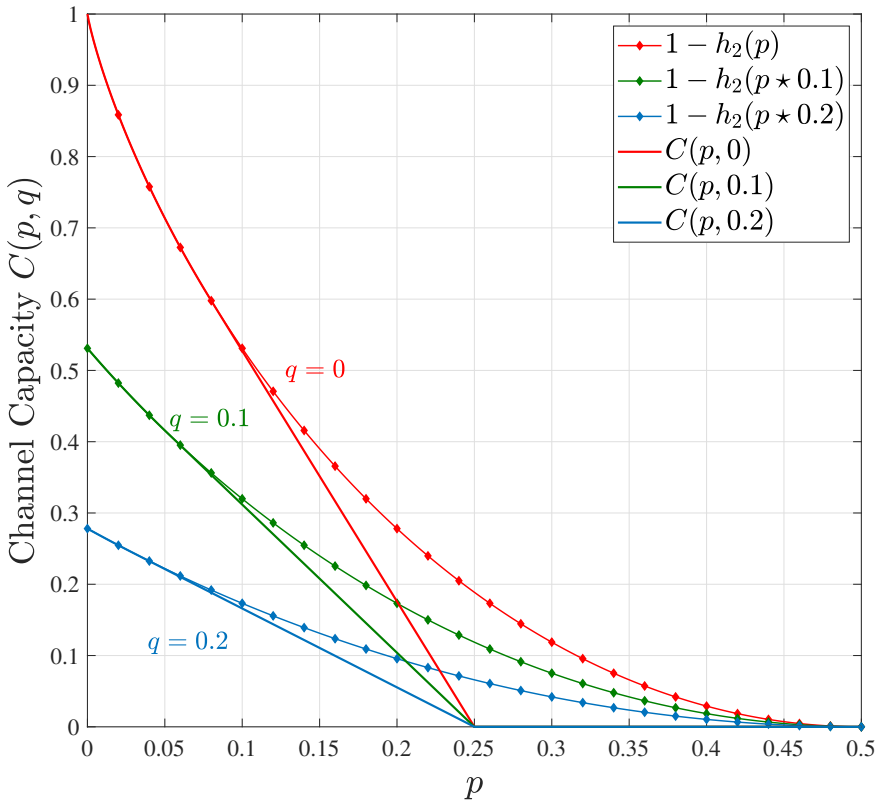


Figure 3.2. The capacity $C(p,q)$ of BSC(q)-ADV(p)-FS as a function of p. The cut-off value of p beyond which $C(p,q)=0$ is $p=1/4$ independent of q.

- It can also be seen that as $q \searrow 0$,

$$p_q \nearrow p_0 = \frac{1}{6}\left(5 - \frac{4}{\sqrt[3]{19-3\sqrt{33}}} - \sqrt[3]{19-3\sqrt{33}}\right),$$

  and as $q \nearrow 1/2$, $p_q \searrow 0$. The regime over which a simple i.i.d. adversarial attack is optimal (i.e. $p \in [0, p_q]$) shrinks as the BSC gets noisier.

- For a fixed $q \in [0, 1/2)$, $C(p,q)$ for $p_q \leq p \leq 1/4$ is a decreasing linear function in p that intersects the p-axis at $p = 1/4$. Furthermore, $C(p,q)$, $p_q \leq p \leq 1/4$ is in fact the *tangent* to the curve $1 - h_2(p \star q)$ at $p = p_q$. The optimal attack for Calvin in this regime relies on his snooping abilities, and is described in section 3.4.2.

In summary, for $q \in [0, 1/2)$, we have

$$C(p,q) = \begin{cases} 1 - h_2(p \star q) & 0 \le p \le p_q \\ \frac{1-4p}{1-4p_q}\left(1 - h_2\left(p_q \star q\right)\right) & p_q \le p \le 1/4 \\ 0 & p \ge 1/4 \end{cases}$$

where p_q is implicitly given by (3.9). In Fig. 3.2, we plot $C(p,q)$ as a function of p for various values of q, specifically, $q = 0.0, 0.1, 0.2$.

Analytical form of $C(p,q)$: Fix a $q \in [0, 1/2)$. The optimization problem (3.8) in Theorem 3.3.2 is

$$\min_{0 \le x \le p} f(x) \tag{3.10}$$

where

$$f(x) = (1 - 4p + 4x)\left(1 - h_2\left(\frac{x}{1 - 4p + 4x} \star q\right)\right).$$

When $p = 1/4$, $f(x) = 0$ at $x = 0$ and hence $C(p,q) = 0$ when $p = 1/4$. Differentiating the objective function in (3.10),

$$\frac{d}{dx}\left((1 - 4p + 4x)\left(1 - h_2\left(\frac{x}{1 - 4p + 4x} \star q\right)\right)\right) = 0$$

we get,

$$4 + (2q+1)\log_2\left(\frac{x(1+2q) + q(1-4p)}{1 - 4p + 4x}\right) + (3-2q)\log_2\left(\frac{1 - 4p + 4x - x(1+2q) - q(1-4p)}{1 - 4p + 4x}\right) = 0.$$

Solution x^* has the form $x^* = \frac{1-4p}{\alpha - 3}$ where α satisfies

$$4 + (1+2q)\log_2\left(\frac{1 - q(1-\alpha)}{1+\alpha}\right) + (3-2q)\log_2\left(\frac{\alpha + q(1-\alpha)}{1+\alpha}\right) = 0.$$

Since $0 \le x \le p$, we must have $\frac{1-4p}{\alpha-3} \le p \implies p \ge \frac{1}{1+\alpha} = p_q$. Thus, for $p \in [p_q, 1/4]$, the minimizer in (3.10) is $x^* = \frac{(1-4p)p_q}{1-4p_q}$ where p_q satisfies

$$4 + (1+2q)\log_2\left(p_q \star q\right) + (3-2q)\log_2\left(1 - p_q \star q\right) = 0, \tag{3.11}$$

and the capacity expression becomes

$$\begin{aligned} C(p,q) &= \frac{1-4p}{1-4p_q}\left(1-h_2\left(\frac{\frac{p_q}{1-4p_q}}{1+\frac{p_q}{1-4p_q}}\star q\right)\right) \\ &= \frac{1-4p}{1-4p_q}\left(1-h_2(p_q\star q)\right) \end{aligned}.$$

Thus, $C(p,q)$, $p_q \le p \le 1/4$ is a straight line that intersects the p-axis at $p=1/4$. For $p \in [0,p_q]$, the minimizer in (3.10) is $x^*=p$ and the capacity expression is $C(p,q)=1-h_2(p\star q)$.

Next we show that, $C(p,q)$, $p_q \le p \le 1/4$ is in fact the tangent to the curve $1-h_2(p\star q)$ at $p=p_q$. Consider the line $L(p)$ that is tangent to $1-h_2(p\star q)$ and passes through $(1/4,0)$. Its equation can be written as $L(p)=C(1-4p)$ where C is a constant. Suppose that $L(x)$ intersects $1-h_2(p\star q)$ at $p=\tilde{p_q}$. To complete the proof, it suffices to show that $\tilde{p_q}=p_q$ i.e. $\tilde{p_q}$ satisfies (3.11). Since $L(p)$ is the tangent to $1-h_2(p,q)$ at $p=\tilde{p_q}$, we have

$$\frac{d}{dp}L(p)\Big|_{p=\tilde{p_q}} = \frac{d}{dp}(1-h_2(p\star q))\Big|_{p=\tilde{p_q}}$$

which gives

$$-4C = (1-2q)\log_2\left(\frac{\tilde{p_q}\star q}{1-\tilde{p_q}\star q}\right). \tag{3.12}$$

We also have

$$L(\tilde{p_q}) = 1-h_2(\tilde{p_q}\star q) = C(1-4\tilde{p_q}). \tag{3.13}$$

Eliminating the constant C from (3.12) and (3.13), $\tilde{p_q}$ satisfies the equation

$$(1-2q)\log_2\left(\frac{\tilde{p_q}\star q}{1-\tilde{p_q}\star q}\right) = -4\left(\frac{1-h_2(\tilde{p_q}\star q)}{1-4\tilde{p_q}}\right).$$

Rearranging the terms,

$$((1-2q)(4\tilde{p_q}-1)-(\tilde{p_q}\star q))\log_2(\tilde{p_q}\star q) - ((1-2q)(4\tilde{p_q}-1)-(1-(\tilde{p_q}\star q)))\log_2(1-\tilde{p_q}\star q) = 4$$

which simplifies to

$$4 + (1 + 2q) \log_2 (\tilde{p_q} \star q) + (3 - 2q) \log_2 (1 - \tilde{p_q} \star q) = 0$$

which is the same as (3.11). Hence, $p_q = \tilde{p_q}$ and the claim holds.

3.4 Converse Proofs

To prove the converse, we demonstrate an attack strategy for Calvin in each of our models under which no rate larger than the claimed capacity expression is achievable. These attacks are inspired by, but different from, the attacks in [58], [59], [71] which only work when the erasure or the bit-flip probability $q = 0$. Specifically, our modified attacks rely crucially on Calvin's ability to snoop.

We shall denote the transmitted and the received codewords as $\mathbf{x}$ and $\mathbf{y}$ respectively. The (possibly stochastic) encoder and the decoder being used by Alice and Bob are denoted as $\Phi(.|.)$ and $\Gamma(.)$ Let $\mathbf{x}_L = (x_1, x_2, \cdots, x_\ell)$ and $\mathbf{x}_R = (x_{\ell+1}, \cdots, x_n)$, where ℓ is selected suitably for each model. Similarly, let $\mathbf{y}_L = (y_1, y_2, \cdots, y_\ell)$ and $\mathbf{y}_R = (y_{\ell+1}, \cdots, y_n)$.

3.4.1 Converse for BEC(q)-ADV(p)-FS

Our proof is based on a *wait and snoop, then push* attack. Suppose Alice attempts to communicate at a rate $R = C^E(p, q) + \epsilon = (1 - 2p)(1 - q) + \epsilon$. We will show that for sufficiently large block-length n, the probability of decoding error under the proposed attack is lower bounded by a constant that is only a function of ϵ (and independent of n). The two phases of the attack are:

- **Wait and Snoop**: Calvin waits and does not induce any erasures for the first $\ell = n\frac{R-\frac{\epsilon}{2}}{1-q}$ channel uses. Instead, Calvin simply snoops into Bob's reception to determine the erased/unerased bits and their positions. At the end of this phase, Bob receives $\mathbf{y}_L = (y_1, y_2, \cdots, y_\ell)$ containing some erased and some unerased bits. Note that the erasures in this phase occur purely due to the BEC(q) channel. Let $\{i_j\}_{j=1}^m$ be the

indices of symbols in $\mathbf{y}_L$ that remain unerased. Here, m is a random quantity in accordance to the distribution of erasures from the BEC(q).

- **Push**: Calvin forms the set $\mathcal{B}_{\mathbf{y}_L}$ of codewords consistent with $\mathbf{y}_L$ as

$$\mathcal{B}_{\mathbf{y}_L} = \{\mathbf{v} \in \mathcal{X}^n : \exists \tilde{u} \in \mathcal{U} \text{ s.t. } \Phi(\mathbf{v}|\tilde{u}) > 0 \text{ and } v_{i_k} = x_{i_k}\ k = 1, 2, \cdots, m\}, \tag{3.14}$$

where $\Phi(.|u)$ is the distribution of codewords when message u is to be transmitted. In other words, $\mathcal{B}_{\mathbf{y}_L}$ consists of all possible codewords that align with $\mathbf{y}_L$ at the positions that are unerased. Calvin then samples a codeword $\mathbf{x}'$ from $\mathcal{B}_{\mathbf{y}_L}$ according to the distribution $\mathbf{x}' \sim P_{\mathbf{X}|\mathbf{Y}_L=\mathbf{y}_L}(.|\mathbf{y}_L)$. In the push phase then, Calvin simply erases bit x_i, $i = \ell+1, \ell+2, \cdots n$ whenever $x_i \neq x_i$, until his budget of pn erasures runs out. If codewords $\mathbf{x}$ and $\mathbf{x}'$ correspond to distinct messages u and u and we have that $d(\mathbf{x}_R, \mathbf{x}'_R) < pn$, there is no way for Bob to distinguish between messages u and u and a decoding error occurs with probability at least 1/2. This is illustrated in Fig. 3.3. We shall argue that this indeed occurs with a positive probability independent of n.

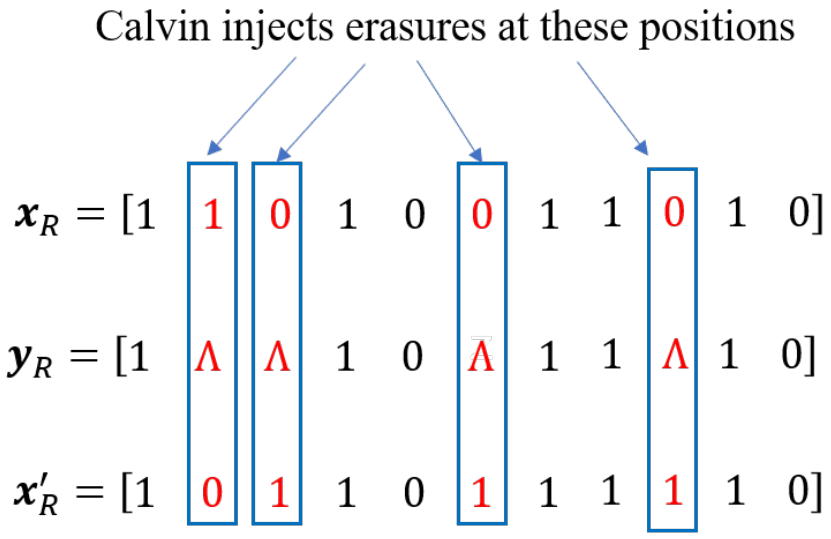


Figure 3.3. In the push phase, if $\mathbf{x}_R$ and $\mathbf{x}'_R$ are sufficiently close (within distance pn), Calvin can make Bob completely uncertain whether the transmitted codeword was $\mathbf{x}$ or $\mathbf{x}'$.

Note that while the presence of the BEC(q) lowers the target rate, Calvin adds no erasures for approximately $n(1-2p)$ channel uses which from [57], [59] is optimal when there is no BEC(q). The main difference in attack when $q \neq 0$ is that even though Calvin knows the entire prefix of the transmitted codeword $\mathbf{x}_L = (x_1, x_2, \cdots, x_\ell)$, he forms his set in (3.14) based only on the unerased bits. Thanks to feedback snooping, Calvin exploits the additional equivocation induced by the BEC(q) in the wait and snoop phase to pick a codeword that is sufficiently close to the transmitted codeword, and which corresponds to a message different from one that Alice chose. While we give Calvin full causal access to Bob's reception, an alternate model where Calvin is allowed *one-time block feedback* is sufficient - he would add no erasures for ℓ channel uses, retrieve through feedback the entire block $\mathbf{y}_L$ and then 'push'.

The proof steps are similar to section A from [59] except that we account for the presence of the BEC(q) in our claims. Define the set $A_0 = \left\{\mathbf{y}_L : H(\mathbf{U} \mid \mathbf{Y}_L = \mathbf{y}_L) > \frac{n\epsilon}{4}\right\}$ and the event $E_1 = \{\mathbf{Y}_L \in A_0\}$. We have the following lemma.

Lemma 6. $P(E_1) \geq \frac{\epsilon}{4}$.

Proof. Since $\mathbf{U} \to \mathbf{X}_L \to \mathbf{Y}_L$ is a Markov chain, by the data processing inequality, we have

$$I(\mathbf{U}; \mathbf{Y}_L) \leq I(\mathbf{X}_L, \mathbf{Y}_L) = \ell(1-q) = n(R - \epsilon/2).$$

The above holds since Calvin adds no erasures in the wait and snoop phase and the channel between $\mathbf{X}_L$ and $\mathbf{Y}_L$ is a BEC(q). Now, since $H(\mathbf{U}) = nR$, we have

$$H(\mathbf{U}|\mathbf{Y}_L) = \mathbb{E}_{\mathbf{Y}_L} H(\mathbf{U}|\mathbf{Y}_L = \mathbf{y}_L) = H(\mathbf{U}) - I(\mathbf{U}; \mathbf{Y}_L) \geq n\epsilon/2.$$

By Markov's inequality then,

$$P\left(nR - H(\mathbf{U}|\mathbf{Y}_L = \mathbf{y}_L) > nR - n\epsilon/4\right) \leq 1 - \frac{\epsilon/4}{R - \epsilon/4}$$

which gives as desired,

$$P(E_1) = P\left(H(\mathbf{U} \mid \mathbf{Y}_L = \mathbf{y}_L) > \frac{n\epsilon}{4}\right) \geq \frac{\epsilon}{4}.$$

□

Now let E_2 be the event $\{\mathbf{U} \neq \mathbf{U}'\}$ and E_3 be the event $\{d(\mathbf{X}_R, \mathbf{X}'_R) < pn\}$. First, we show the following.

Lemma 7. *For* $\mathbf{y}_L \in A_0$,

$$P(E_2, E_3 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \epsilon^{\mathcal{O}(1/\epsilon)} \tag{3.15}$$

Proof. Consider sampling $t = \frac{9}{\epsilon}$ codewords $\mathcal{C}_t = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(t)}\right\}$ from the set $\mathcal{B}_{\mathbf{y}_L}$ where each codeword is sampled independently according to the conditional distribution $P_{\mathbf{X}|\mathbf{Y}_L=\mathbf{y}_L}(.|\mathbf{y}_L)$. Let the messages corresponding to the codewords be $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_t$ and let E_4 be the event that $\{\mathbf{U}_1, \mathbf{U}_2, \cdots \mathbf{U}_t$ are all distinct$\}$. We have from [59, A.2, Proposition 1] that for $\mathbf{y}_L \in A_0$ and for sufficiently large block length n,

$$P(E_4 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \left(\frac{\epsilon}{5}\right)^{t-1}. \tag{3.16}$$

Now, the average Hamming distance between the suffixes of codewords in $\mathcal{C}_t$ is defined as

$$d_{avg}(\mathcal{C}_t) = \frac{1}{t(t-1)} \sum_{i \neq j} d_H\left(\mathbf{X}_R^{(i)}, \mathbf{X}_R^{(j)}\right).$$

Conditioning on E_4, Plotkin's bound dictates that

$$\begin{aligned} d_{avg}(\mathcal{C}_t) \leq \frac{1}{2}\frac{t}{t-1}(n-\ell) &= n\frac{t}{t-1}\left(p - \frac{\epsilon}{4(1-q)}\right) \\ &\leq n\frac{\frac{9}{\epsilon}}{\frac{9}{\epsilon}-1}\left(p - \frac{\epsilon}{4}\right) \leq np - n\frac{\epsilon}{8}. \end{aligned}$$

Thus for $\mathbf{y}_L \in A_0$, we have

$$\mathbb{E}(d_{avg}(\mathcal{C}_t) \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) \leq np - n\epsilon/8.$$

Now, since all of the $\mathbf{X}^{(i)}$'s are picked independently, all pairs $(\mathbf{X}^{(i)}, \mathbf{X}^{(j)})$ have identical distribution. Thus,

$$\mathbb{E}(d_{avg}(\mathcal{C}_t) \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) = \mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_4, \mathbf{Y}_L = \mathbf{y}_L)$$

and also

$$\mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) = \mathbb{E}(d_H(\mathbf{X}_R, \mathbf{X}_R') \mid E_4, \mathbf{Y}_L = \mathbf{y}_L).$$

Thus, we have

$$\mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) \leq np - n\epsilon/8$$

and by Markov's inequality

$$P(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) > np \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) \leq 1 - \frac{\epsilon}{8p}. \tag{3.17}$$

We have also,

$$P(E_2, E_3 \mid \mathbf{Y}_L = \mathbf{y}_L) = P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq pn, \mathbf{U}_1 \neq \mathbf{U}_2 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq$$
$$P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq pn, E_4 \mid \mathbf{Y}_L = \mathbf{y}_L).$$

where the last inequality holds because event E_4 is a subset of the event $\{U_1 \neq U_2\}$. We have also,

$$P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq pn, E_4 \mid \mathbf{Y}_L = \mathbf{y}_L) = P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq pn \mid E_4, \mathbf{Y}_L = \mathbf{y}_L) P(E_4 \mid \mathbf{Y}_L = \mathbf{y}_L).$$

From (3.16) and (3.17), finally we get,

$$P(E_2, E_3 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \frac{\epsilon}{8p} \left(\frac{\epsilon}{5}\right)^{\frac{9}{\epsilon}-1} = \epsilon^{\mathcal{O}(1/\epsilon)}$$

as we set out to prove. □

Recall that E_2 is the event that the message $\mathbf{U}'$ picked by the adversary is different from the one transmitted and E_3 is the event that the corresponding codewords $\mathbf{X}_R$ and $\mathbf{X}_R'$ are

close enough so that Calvin's push phase succeeds and Bob is completely uncertain whether the message transmitted was $\mathbf{U}$ or $\mathbf{U}'$. Hence when E_2 and E_3 occur, the probability of decoding error is at least $1/2$. To finish the proof, we need only show a lower bound on $P(E_2, E_3)$. We have,

$$\begin{aligned}
P(E_2, E_3) &\geq P(E_2, E_3, E_1) \\
&= \sum_{\mathbf{y}_L \in A_0} P(E_2, E_3 \mid \mathbf{Y}_L = \mathbf{y}_L) P(\mathbf{Y}_L = \mathbf{y}_L) \\
&\geq \frac{\epsilon}{8p} \left(\frac{\epsilon}{5}\right)^{\frac{9}{\epsilon}-1} \sum_{\mathbf{y}_L \in A_0} P(\mathbf{Y}_L = \mathbf{y}_L) \\
&= \frac{\epsilon}{8p} \left(\frac{\epsilon}{5}\right)^{\frac{9}{\epsilon}-1} P(E_1) \\
&\geq \frac{\epsilon}{4} \frac{\epsilon}{8p} \left(\frac{\epsilon}{5}\right)^{\frac{9}{\epsilon}-1},
\end{aligned}$$

a lower bound that is independent of n, hence completing the proof.

3.4.2 Converse for BSC(q)-ADV(p)-FS

Fix a $\bar{p} \in [0, p]$. Suppose that for some $\epsilon > 0$, the transmitter attempts to communicate at a rate of $R = \alpha(p, \bar{p}) \left(1 - h_2\left(\frac{\bar{p}}{\alpha(p,\bar{p})} \star q\right)\right) + \epsilon$. We show that for sufficiently large n, under the proposed attack strategy for Calvin, the probability of decoding error in (3.1) is lower bounded by $\epsilon^{O(1/\epsilon)}$, a quantity *independent* of n. Since the same argument works for any $\bar{p}$, the result in theorem 3.3.2 holds.

Our proof is based on a *babble and snoop, then push* attack that consists of the following two phases:

- **Babble and Snoop**: For the first $\ell = (\alpha(p, \bar{p}) + \epsilon/2)n$ channel uses, Calvin injects random bit-flips and monitors Bob's reception - at channel use i, $1 \leq i \leq \ell$, he flips bit x_i with probability $\bar{p}n/\ell$. At the end of this phase, Calvin knows $\mathbf{x}_L$ and $\mathbf{y}_L$.

- **Push**: Calvin samples a codeword $\mathbf{x}'$ (corresponding to message u) according to the conditional distribution $P_{\mathbf{X}|\mathbf{Y}_L=\mathbf{y}_L}(.|\mathbf{y}_L)$. His goal is to confuse the receiver between $\mathbf{x}$ and $\mathbf{x}'$. At positions where $\mathbf{x}_R$ and $\mathbf{x}'_R$ agree, he does nothing. Positions j where $\mathbf{x}_R$

and $\mathbf{x}'_R$ disagree, he flips x_j with probability 1/2. This is illustrated in Fig. This way, the Bob cannot distinguish between $\mathbf{x}$ and $\mathbf{x}'$ (even with the BSC(q)) due to the fact that $p(\mathbf{y}_R|\mathbf{x}_R) = p(\mathbf{y}_R|\mathbf{x}'_R)$. The proof relies on showing that with a small probability independent of n, u, u are distinct and $\mathbf{x}_R$, $\mathbf{x}'_R$ are sufficiently close.

Note that Calvin requires knowledge of $\mathbf{Y}_L$, i.e., the symbols received by Bob during the first phase of the attack. Intuitively, the presence of the BSC(q) introduces additional equivocation at the receiver which Calvin is able to exploit to cause a reduction in rate. Here also, *one-time block feedback* (of entire block $\mathbf{y}_L$) after the first ℓ channel uses is sufficient for the attack to succeed.

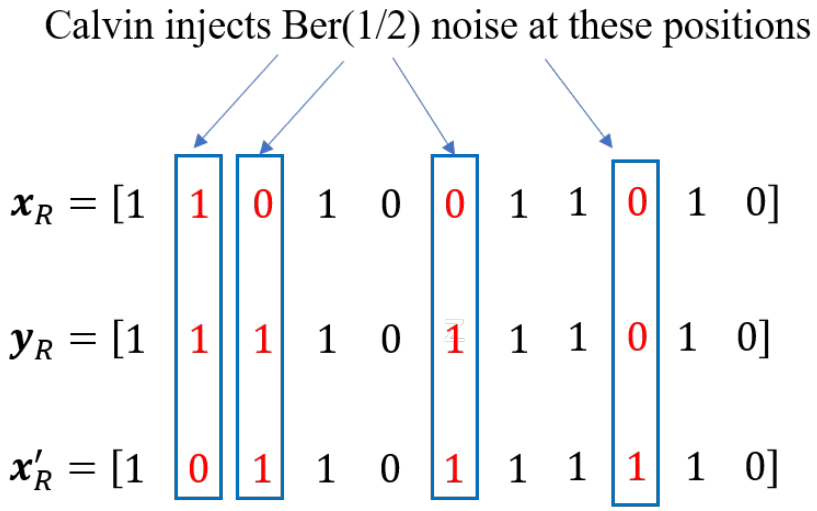


Figure 3.4. In the push phase, if $\mathbf{x}_R$ and $\mathbf{x}'_R$ are sufficiently close, Calvin can make Bob completely uncertain whether the transmitted codeword was $\mathbf{x}$ or $\mathbf{x}'$ by injecting Ber(1/2) noise at positions where $\mathbf{x}_R$ differs from $\mathbf{x}'_R$.

In the babble and snoop phase, by the Chernoff bound, Calvin uses at most $\bar{p}n + \epsilon n/64$ flips with probability at least $1 - e^{-\Omega(\epsilon^2 n)}$. Let this be denoted as event E_1. Conditioned on E_1, Calvin's remaining budget in the push phase is atleast $(p - \bar{p})n - \epsilon n/64$. Define the set

$$A_0 = \left\{ \mathbf{y}_L : H(\mathbf{U} \mid \mathbf{Y}_L = \mathbf{y}_L) > \frac{n\epsilon}{4} \right\}.$$

Defining the event $E_2 = \{\mathbf{Y}_L \in A_0\}$, we have the following lemma.

Lemma 8. $P(E_2) \geq \epsilon/4$.

Proof. The proof closely follows claim 4 in [58]. Note that $\mathbf{U} \to \mathbf{X}_L \to \mathbf{Y}_L$ is a markov chain and hence, by the data processing inequality and Calvin's actions in the babble phase,

$$I(\mathbf{U}; \mathbf{Y}_L) \leq I(\mathbf{X}_L; \mathbf{Y}_L) = \ell \left(1 - h_2\left(\frac{\bar{p}n}{\ell} \star q\right)\right).$$

This is because the channel between $\mathbf{X}_L$ and $\mathbf{Y}_L$ is now a cascade of $BSC(\bar{p}n/\ell)$ and $BSC(q)$. Noting that $\ell = (\alpha + \epsilon/2)n$,

$$I(\mathbf{U}; \mathbf{Y}_L) \leq n(\alpha + \epsilon/2)\left(1 - h_2\left(\frac{\bar{p}}{\alpha + \epsilon/2} \star q\right)\right).$$

Since $I(\mathbf{U}, \mathbf{Y}_L) = H(\mathbf{U}) - H(\mathbf{U}|\mathbf{Y}_L)$ and $H(\mathbf{U}) = nR = n\alpha\left(1 - h_2\left(\frac{\bar{p}}{\alpha} \star q\right)\right) + n\epsilon$, we get,

$$H(\mathbf{U}|\mathbf{Y}_L) \geq \frac{n\epsilon}{2} + n\left((\alpha + \epsilon/2)h_2\left(\frac{\bar{p}}{\alpha + \epsilon/2} \star q\right) - \alpha h_2\left(\frac{\bar{p}}{\alpha} \star q\right)\right).$$

Now, the function $f(x) = xh_2\left(\frac{\bar{p}}{x} \star q\right)$ is increasing in x, for any fixed $q \in (0, 1/2)$. To see this, note that

$$\frac{df}{dx} = h_2\left(\frac{\bar{p}}{x} \star q\right) + (2q - 1)\frac{\bar{p}}{x}\log_2\left(\frac{1 - \frac{\bar{p}}{x} \star q}{\frac{\bar{p}}{x} \star q}\right) > 0$$

since $\frac{\bar{p}}{x} \star q < 1/2$ and $\log_2\left(\frac{1-y}{y}\right) = \frac{d}{dy}h_2(y) > 0$ for $y \in (0, 1/2)$. Hence, we have $H(\mathbf{U}|\mathbf{Y}_L) = \mathbb{E}_{\mathbf{Y}_L} H(\mathbf{U}|\mathbf{Y}_L = \mathbf{y}_L) \geq n\epsilon/2$. Finally, by Markov's inequality,

$$P\left(nR - H(\mathbf{U}|\mathbf{Y}_L = \mathbf{y}_L) > nR - n\epsilon/4\right) \leq 1 - \frac{\epsilon/4}{R - \epsilon/4}$$

which gives as desired,

$$P\left(H(\mathbf{U} \mid \mathbf{Y}_L = \mathbf{y}_L) > \frac{n\epsilon}{4}\right) \geq \frac{\epsilon}{4}.$$

□

Next, define the events $E_3 = \{\mathbf{U} \neq \mathbf{U}'\}$ and $E_4 = \{d_H(\mathbf{X}_R, \mathbf{X}'_R) \leq 2(p - \bar{p})n - \epsilon n/8\}$. E_3 is the event that the message picked by the adversary to confuse Bob in the push phase is different from the one transmitted. Similarly, event E_4 ensures that Calvin's remaining flips

are enough to carry his push attack. Using techniques from section A.2 of [59] and claim 6 in [58], we can now show the following.

Lemma 9. *For* $\mathbf{y}_L \in A_0$,

$$P(E_3, E_4 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \frac{\epsilon}{48}\left(\frac{\epsilon}{5}\right)^{\frac{12}{\epsilon}-1} = \epsilon^{\mathcal{O}(1/\epsilon)}. \tag{3.18}$$

Proof. Consider sampling $t = \frac{12}{\epsilon}$ codewords $\mathcal{C}_t = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(t)}\right\}$, each codeword sampled according to the conditional distribution $P_{\mathbf{X}|\mathbf{Y}_L=\mathbf{y}_L}(.|\mathbf{y}_L)$. Let the messages corresponding to the codewords be $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_t$ and let E_5 be the event that $\{\mathbf{U}_1, \mathbf{U}_2, \cdots \mathbf{U}_t$ are all distinct$\}$ i.e. all of the codewords are distinct. We have from proposition 1, section A.2 from [59] that for $\mathbf{y}_L \in A_0$, for sufficiently large block length n,

$$P(E_5 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \left(\frac{\epsilon}{5}\right)^{t-1}.$$

The average Hamming distance between the suffixes of codewords in $\mathcal{C}_t$ is defined as

$$d_{avg}(\mathcal{C}_t) = \frac{1}{t(t-1)}\sum_{i\neq j} d_H\left(\mathbf{X}_R^{(i)}, \mathbf{X}_R^{(j)}\right).$$

Recall that $\ell = (1 - 4(p - \bar{p}) + \epsilon/2)n$. Conditioning on E_5, by Plotkin's bound we have

$$d_{avg}(\mathcal{C}_t) \leq \frac{1}{2}\frac{t}{t-1}(n - \ell) \leq 2(p - \bar{p})n - \epsilon n/6.$$

Thus for $\mathbf{y}_L \in A_0$, we have

$$\mathbb{E}(d_{avg}(\mathcal{C}_t) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) \leq 2(p - \bar{p})n - \epsilon n/6.$$

Now, since all of the $\mathbf{X}^{(i)}$'s are picked independently, all pairs $(\mathbf{X}^{(i)}, \mathbf{X}^{(j)})$ have identical distribution. Thus,

$$\mathbb{E}(d_{avg}(\mathcal{C}_t) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) = \mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L)$$

and also

$$\mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) = \mathbb{E}(d_H(\mathbf{X}_R, \mathbf{X}'_R) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L).$$

Thus, we have

$$\mathbb{E}(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) \leq 2(p-\bar{p})n - \epsilon n/6.$$

and by Markov's inequality

$$P(d_H(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) > 2(p-\bar{p})n - \epsilon n/8 \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) \leq \frac{2(p-\bar{p})n - \epsilon n/6}{2(p-\bar{p})n - \epsilon n/8} = 1 - \frac{\epsilon}{48(p-\bar{p}) - 3\epsilon} \leq 1 - \frac{\epsilon}{48}. \quad (3.19)$$

Thus,

$$\begin{aligned} P(E_3, E_4 \mid \mathbf{Y}_L = \mathbf{y}_L) &= P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq 2(p-\bar{p})n - \epsilon n/8, \mathbf{U}_1 \neq \mathbf{U}_2 \mid \mathbf{Y}_L = \mathbf{y}_L) \\ &\geq P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq 2(p-\bar{p})n - \epsilon n/8, E_5 \mid \mathbf{Y}_L = \mathbf{y}_L), \end{aligned}$$

where the last inequality holds because event E_5 is a subset of the event $\{U_1 \neq U_2\}$. We then have,

$$\begin{aligned} P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq 2(p-\bar{p})n - \epsilon n/8, E_5 \mid \mathbf{Y}_L = \mathbf{y}_L) = \\ P(d(\mathbf{X}_R^{(1)}, \mathbf{X}_R^{(2)}) \leq 2(p-\bar{p})n - \epsilon n/8 \mid E_5, \mathbf{Y}_L = \mathbf{y}_L) P(E_5 \mid \mathbf{Y}_L = \mathbf{y}_L). \end{aligned}$$

From (3.19), when E_1 occurs i.e. $\mathbf{y}_L \in A_0$, we get,

$$P(E_3, E_4 \mid \mathbf{Y}_L = \mathbf{y}_L) \geq \frac{\epsilon}{48} \left(\frac{\epsilon}{5}\right)^{\frac{12}{\epsilon}-1} = \epsilon^{\mathcal{O}(1/\epsilon)}$$

as we set out to prove. □

Now, in the push phase, Calvin injects $Ber(1/2)$ noise at $d_H(\mathbf{X}_R, \mathbf{X}_R)$ positions. Conditioned on E_1, Calvin has at least a budget of $(p-\bar{p})n - \epsilon n/64$ bit-flips that remain. If $\mathbf{a}_R$ is the error vector chosen by Calvin in the push phase, conditioned on E_3 and E_4 we have $\mathbb{E}(d_H(\mathbf{a}_R, \mathbf{0})) = (p-\bar{p})n - \epsilon n/16$. Further by the Chernoff bound, with probability at least

$1 - 2^{-\Omega(\epsilon^2 n)}$, the distance $d_H(\mathbf{a}_R, \mathbf{0})$ is within $3\epsilon n/64$ of its expected value. Let this event be E_5. Since $\mathbb{E}(d_H(\mathbf{a}_R, \mathbf{0})) + 3\epsilon n/64 = (p - \bar{p})n - \epsilon n/64$, the power constraint is respected w.h.p..

When events E_1, E_3, E_4, E_5 occur, the probability of decoding error is clearly at least $1/2$ since the receiver cannot distinguish between $\mathbf{x}$ and $\mathbf{x}'$. Since $P(E_1) \geq 1 - e^{-\Omega(\epsilon^2 n)}$ and $P(E_5) \geq 1 - e^{-\Omega(\epsilon^2 n)}$, the bound in (3.18) together with the bound $P(E_2) \geq \epsilon/4$ implies for sufficiently large n, the maximum probability of error in (3.1) is at least of the order $\epsilon^{O(1/\epsilon)}$, a quantity independent of n and the proof is complete.

3.5 Achievability Proofs

To prove achievability, we resort to a random coding argument. We consider a distribution over an ensemble of stochastic codes, and show that with positive probability, a code drawn randomly from the ensemble enables reliable communication between Alice and Bob. This then implies the existence of a specific (stochastic) code achieving capacity.

For both channel models BEC(p)-ADV(p)-FS and BSC(p)-ADV(p)-FS, we shall use the random code ensemble from [57] with a reduced rate as given in theorems 3.3.1 and 3.3.2 respectively. However, note that compared to the $q = 0$ case, the decoding procedure will need to be modified greatly to deal with compounded adversarial and random errors.

Random Code Distribution: Alice is endowed with a set of private keys or secrets for encoding, $\mathcal{S} = \{1, 2, \cdots, 2^{nS}\}$. The encoding procedure is carried out in chunks, each of size $n\theta$ where $\theta < 1$ is a quantization parameter. The values for S and θ are set specific to the channel model later. Let Γ be the uniform distribution over stochastic codes $\mathcal{C} : \mathcal{U} \times \mathcal{S} \to \mathcal{X}^{n\theta}$. Then each chunk i, $1 \leq i \leq \frac{1}{\theta}$, is associated to a stochastic code $\mathcal{C}_i$ drawn independently from the distribution Γ.

Encoding Procedure: For message $u \in \mathcal{U}$ and keys $s_1, s_2, \cdots, s_{\frac{1}{\theta}}$, the codeword $\mathbf{x}$ selected for transmission is

$$\mathbf{x} = \mathcal{C}_1(u, s_1) \circ \mathcal{C}_2(u, s_2) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u, s_{\frac{1}{\theta}}),$$

where $\circ$ represents the concatenation operator. We refer to codeword $\mathcal{C}_i(u, s_i)$ as the i^{th} sub-codeword or the i^{th} chunk and the code $\mathcal{C}_i$ as the i^{th} sub-code. Each secret or key s_i for encoding with $\mathcal{C}_i$ is chosen uniformly randomly from $\mathcal{S}$.

Decoding Procedure: The decoding procedure is specific to each of the channel models and is described later.

Define the set $\mathcal{T} = \{n\theta, 2n\theta, \cdots, n - n\theta\}$ containing indices of the chunk ends. For some $t \in \mathcal{T}$ where $t = kn\theta$, we refer to $\mathcal{C}_1 \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_k$ as the left mega sub-code w.r.t. t and $\mathcal{C}_{k+1} \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_{\frac{1}{\theta}}$ as the right mega sub-code w.r.t. t. Accordingly, the concatenation of the first k sub-codewords is be referred to as the left mega sub-codeword w.r.t. t, and that of the last $\frac{1}{\theta} - k$ sub-codewords is referred to as right mega sub-codeword w.r.t. t. We shall also denote the key sequences used to encode the left and the right mega-subcodewords as $s_{left} = (s_1, s_2, \cdots, s_k)$ and $s_{right} = (s_{k+1}, s_{k+2}, \cdots, s_{\frac{1}{\theta}})$.

3.5.1 Achievability for BEC(q)-ADV(p)-FS

Fix $\epsilon > 0$ and let the rate be $R = (1 - 2p)(1 - q) - \epsilon = (1 - 2p - \epsilon)(1 - q)$, where $\epsilon = \epsilon/(1 - q)$. We show that R is achievable for the BEC(q)-ADV(p)-FS model, i.e., there exists a stochastic code from the ensemble for which our decoder (to be described shortly) succeeds w.h.p. in finding the transmitted message u^*. We set initially $\theta = \frac{\epsilon}{4}$ and $S = \frac{\theta^3}{8}$.

Our decoder construction is a modification of the one described in [57] which we first review. While reviewing, we provide key insights into how the decoding might fail once a BEC is added. Following the review, we use our insights to modify the decoder in order to account for the additional random noise when $q > 0$.

Decoding when $q = 0$ [57]: Let $\mathbf{y}$ denote the entire n-symbol channel output received by Bob. For some $t^* = k^* n\theta$ to be defined shortly, Bob partitions $\mathbf{y}$ into 2 strings: $\mathbf{y}_1^{t^*} = (y_1, \ldots, y_{t^*})$ and $\mathbf{y}_{t^*+1}^n = (y_{t^*+1}, \ldots, n)$. Decoding occurs in two sequential phases.

- **List Decoding:** In the first phase, Bob performs list decoding on $\mathbf{y}_1^{t^*}$ to create a list of messages $\mathcal{L}$ that are consistent with Bob's reception $\mathbf{y}_1^{t^*}$. Here, a message u

is consistent with $\mathbf{y}_1^{t^*}$ iff some codeword corresponding to u agrees with $\mathbf{y}_1^{t^*}$ (on the unerased positions). Thus,

$$\mathcal{L} = \{u \in \mathcal{U} : \exists\ (s_1, \cdots, s_{k^*}) \in \mathcal{S}^{k^*} s.t. \mathcal{C}_1(u, s_1) \circ \cdots \mathcal{C}_k(u, s_k) \text{ and } \mathbf{y}_1^{t^*} \text{agree}\}.$$

- **Unique Decoding (List Refinement):** In the second phase, he refines the list by removing all messages in $\mathcal{L}$ that are not consistent with $\mathbf{y}_{t^*+1}^n$.

If exactly one message, say $\hat{u}$, remains in $\mathcal{L}$ after refinement, the decoder outputs $\hat{u}$. If the refined list does not contain exactly one message, a decoding error is declared. Decoding is successful if $\hat{u} = u^*$, the true message transmitted by Alice.

Here, t^* is chosen as a function of the number of (purely adversarial since $q = 0$) erasures $\lambda_{t^*}^a$ observed in $\mathbf{y}$ up until time t^*. Specifically, Bob chooses t^* as the smallest integer that satisfies the so-called *list-decoding condition*

$$\lambda_{t^*}^a \leq t^*(1-\theta) - ((1-2p) - \epsilon)n \tag{3.20}$$

and the *energy bounding condition*

$$np - \lambda_{t^*}^a \leq \frac{(n-t^*)(1-\theta)}{2}. \tag{3.21}$$

Condition (3.20) ensures the size of $\mathcal{L}$ is small (at most a constant) while condition (3.21) ensures the fraction of erasures that occur in $\mathbf{y}_{t^*+1}^n$ is small enough to perform list refinement.

Problems in this construction arise when $q > 0$. If the decoder assumes that all erasures that he sees are adversarial and performs decoding by selecting t^* according to (3.20) and (3.21), the maximum rate that can be achieved is $C^E(p + q - pq, 0) = C^E(p, q) - q$ which is strictly less than capacity. Therefore, simply counting erasures without knowing (or estimating) their source is no longer a viable strategy when $q > 0$. To circumvent this issue, we modify conditions (3.20) and (3.21) appropriately.

Modified choice of t^*: Let λ_t denote the number of erasures observed by Bob up until time t, which includes contributions both from Calvin and the BEC(q). An example clarifying the definitions of λ_t and λ_t^a is illustrated in Fig. 3.5.

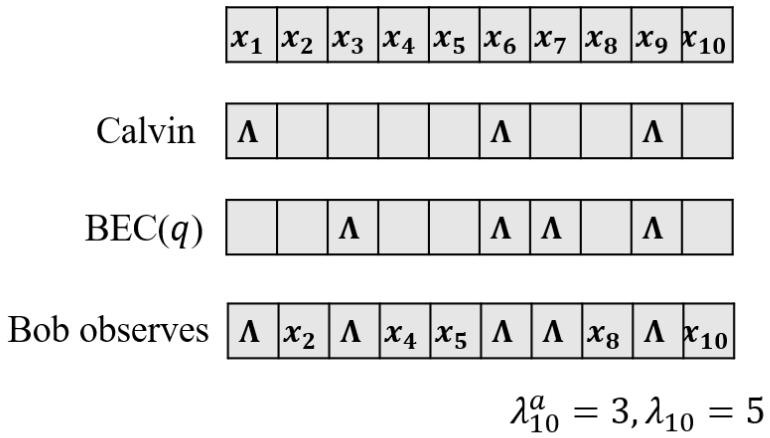


Figure 3.5. In this example, Calvin causes an erasure at indices 1, 6 and 9 while the BEC(q) causes an erasure at indices 3 6, 7 and 9.

Then, Bob chooses t^* as the smallest integer that satisfies the modified list-decoding condition

$$\lambda_{t^*} - qt^* \leq t^*(1-q)(1-\theta) - Rn \tag{3.22}$$

and the modified list refinement condition

$$np(1-q) - (\lambda_{t^*} - qt^*) \leq \frac{(n-t^*)(1-q)(1-\theta)}{2}. \tag{3.23}$$

Note that by the Chernoff bound, if Calvin adds $\lambda_{t^*}^a$ erasures up until t^*, the total number of erasures λ_{t^*} that Bob observes is approximately $\lambda_{t^*} \approx \lambda_{t^*}^a + q(t^* - \lambda_{t^*}^a)$. On making this substitution we see that t^* satisfying (3.22) and (3.23) is nearly the same as that satisfying (3.20) and (3.21) i.e. it is sufficient to choose t^* only as a function of pure adversarial erasures. However, since Bob has no way of knowing this, he works with the quantity $\lambda_{t^*} - qt^*$. Note that since qt^* is an estimate of the number of erasures added by the BEC(q), we can interpret $\lambda_{t^*} - qt^*$ to be an estimate of the number of adversarial erasures that *do not* coincide with random erasures.

Having selected t^*, Bob can then finish decoding using the two-phase decoding process described previously to successfully recover w.h.p. the transmitted message. We now prove this.

Analysis: We begin by reviewing some simple results that are useful in showing that decoding succeeds w.h.p.. Recall that λ_t^a denotes the number of erasures added by Calvin up until time t. By the Chernoff bound we have then, for $\delta > 0$ to be set later, the total number of erasures that Bob observes at time $t = kn\theta$ satisfies

$$\lambda_t \in [\lambda_t^a + (t - \lambda_t^a)(q - \delta), \lambda_t^a + (t - \lambda_t^a)(q + \delta)]$$

with probability at least $P_\delta = 1 - 2^{-\Omega(\delta^2 n)}$. Thus, $\hat{\lambda}_t = \lambda_t - qt$, Bob's estimate of the number of adversarial erasures that do not coincide with BEC(q) erasures satisfies w.h.p.

$$\hat{\lambda}_t \in [\lambda_t^a(1 - q + \delta) - \delta t, \lambda_t^a(1 - q - \delta) + \delta t]\,.$$

Recall that with the modified list-decoding and list refinement conditions, Bob selects the smallest value of $t^* \in \mathcal{T}$ satisfying

$$\lambda_{t^*} - qt^* \leq t^*(1 - q)(1 - \theta) - Rn \tag{3.24}$$

and

$$np(1 - q) - (\lambda_{t^*} - qt^*) \leq \frac{(n - t^*)(1 - q)(1 - \theta)}{2}. \tag{3.25}$$

From the preceding discussion, w.h.p. we have that

$$\lambda_{t^*} - qt^* = \hat{\lambda}_{t^*} \in [\lambda_{t^*}^a(1 - q + \delta) - \delta t^*, \lambda_{t^*}^a(1 - q - \delta) + \delta t^*]. \tag{3.26}$$

Let $\mathcal{Z} = [\lambda_{t^*}^a(1 - q + \delta) - \delta t^*, \lambda_{t^*}^a(1 - q - \delta) + \delta t^*]$. By a similar analysis as in [57, Claim B.3], we show that when $\delta > 0$ is small enough, a $t^* \in \mathcal{T}$ exists that satisfies both (3.24) and (3.25), for any realization of $\hat{\lambda}_{t^*} \in \mathcal{Z}$.

Lemma 10. *Let*

$$\delta < \frac{1}{4}\frac{(1-q)\theta^2(1-\theta)}{1+2\theta-\theta^2}.$$

Then, for any erasure pattern selected by Calvin, there exists a $t^* \in \mathcal{T}$ such that both of the following conditions hold with probability at least $1 - 2^{-\Omega(\delta^2 n)}$:

$$\lambda_{t^*} - qt^* \leq t^*(1-q)(1-\theta) - Rn, \text{ and}$$

$$np(1-q) - (\lambda_{t^*} - qt^*) \leq \frac{(n-t^*)(1-q)(1-\theta)}{2}.$$

Proof. As previously noted, we have that with probability at least P_δ, $\lambda_{t^*} - qt^* = \hat{\lambda}_{t^*} \in [\lambda^a_{t^*}(1-q) - \delta(t^* - \lambda^a_{t^*}), \lambda^a_{t^*}(1-q) + \delta(t^* - \lambda^a_{t^*})]$.

To prove this claim, it suffices to show that a small enough $\delta > 0$ can be set so that a t^* satisfying both conditions exists at both the extremes $\hat{\lambda}_{t^*} = \lambda^a_{t^*}(1-q) - \delta(t^* - \lambda^a_{t^*})$ and $\hat{\lambda}_{t^*} = \lambda^a_{t^*}(1-q) + \delta(t^* - \lambda^a_{t^*})$.

In the first case, we need to prove existence of t^* such that

$$n(1-2p-\epsilon) + \lambda^a_{t^*}\left(1 + \frac{\delta}{1-q}\right) + \theta t^* \leq t^*\left(1 + \frac{\delta}{1-q}\right) \tag{3.27}$$

and

$$np - \lambda^a_{t^*} \leq \frac{(n-t^*)(1-\theta)}{2} - \frac{\delta}{1-q}(t^* - \lambda^a_{t^*}). \tag{3.28}$$

First, choose a $t^* \leq n - n\theta$ in $\mathcal{T}$ such that

$$t^* \geq n(1-2p-\epsilon) + \lambda^a_{t^*}\left(1 + \frac{\delta}{1-q}\right) + \left(\theta - \frac{\delta}{1-q}\right)(n - n\theta).$$

This ensures that (3.27) holds. Rearranging (3.28), we also require

$$t^* \leq \frac{n\left(1 - \frac{2p}{1-\theta}\right) + \frac{2\lambda^a_{t^*}}{1-\theta}\left(1 + \frac{\delta}{1-q}\right)}{1 + \frac{2\delta}{(1-q)(1-\theta)}}.$$

Hence, to prove existence of t^* simultaneously satisfying both required conditions, it is sufficient to show that

$$\left(\frac{n\left(1-\frac{2p}{1-\theta}\right)+\frac{2\lambda^a_{t^*}}{1-\theta}\left(1+\frac{\delta}{1-q}\right)}{1+\frac{2\delta}{(1-q)(1-\theta)}}\right)-\left(n(1-2p-\epsilon)+\lambda^a_{t^*}\left(1+\frac{\delta}{1-q}\right)+\left(\theta-\frac{\delta}{1-q}\right)(n-n\theta)\right)\geq n\theta.$$

Multiplying by $1+\frac{2\delta}{(1-q)(1-\theta)}$, and simplifying, the coefficient of $\lambda^a_{t^*}$ in the above inequality becomes

$$\left(1+\frac{\delta}{1-q}\right)\left(\frac{2}{1-\theta}-1-\frac{2\delta}{(1-q)(1-\theta)}\right)$$

which is positive when $\delta<\frac{1}{2}(1+\theta)(1-q)$. For such a choice of δ, it is sufficient to show

$$\frac{\left(1-\frac{2p}{1-\theta}\right)}{1+\frac{2\delta}{(1-q)(1-\theta)}}-(1-2p-\epsilon)-\left(\theta-\frac{\delta}{1-q}\right)(1-\theta)\geq\theta.$$

Simplifying further, it is enough to show that

$$p\leq\frac{1}{2}\left(\frac{1-\theta}{\theta}\right)\left(\epsilon-2\theta+\theta^2+\frac{\delta}{1-q}\left[1-\theta-\frac{2}{1-\theta}\right]\right). \tag{3.29}$$

Since $\epsilon=4\theta$, choosing

$$\delta<\min\left\{\frac{(\theta^2-\theta^3)(1-q)}{1+2\theta-\theta^2},\frac{1}{2}(1+\theta)(1-q)\right\},$$

we will have $\left(\frac{1-\theta}{\theta}\right)\left(\epsilon-2\theta+\theta^2+\frac{\delta}{1-q}\left[1-\theta-\frac{2}{1-\theta}\right]\right)>1$ so that (3.29) always holds for any $p\in[0,1/2)$ and we are done.

In the second case, we need to prove existence of t^* such that

$$n(1-2p-\epsilon)+\lambda^a_{t^*}\left(1-\frac{\delta}{1-q}\right)+\theta t^*\leq t^*\left(1-\frac{\delta}{1-q}\right) \tag{3.30}$$

and

$$np-\lambda^a_{t^*}\leq\frac{(n-t^*)(1-\theta)}{2}+\frac{\delta}{1-q}(t^*-\lambda^a_{t^*}). \tag{3.31}$$

Proceeding like earlier, choose a $t^* \leq n - n\theta$ in $\mathcal{T}$ such that

$$t^* \geq n(1-2p-\epsilon) + \lambda^a_{t^*}\left(1 - \frac{\delta}{1-q}\right) + \left(\theta + \frac{\delta}{1-q}\right)(n - n\theta),$$

ensuring (3.30) holds. For (3.31) to hold, we need

$$t^* \leq \frac{n\left(1 - \frac{2p}{1-\theta}\right) + \frac{2\lambda^a_{t^*}}{1-\theta}\left(1 - \frac{\delta}{1-q}\right)}{1 - \frac{2\delta}{(1-q)(1-\theta)}}.$$

Since the denominator $1 - \frac{2\delta}{(1-q)(1-\theta)} > 0$ for $\delta < \frac{1}{2}(1-\theta)(1-q)$, we will require that

$$n\left(1 - \frac{2p}{1-\theta}\right) + \frac{2\lambda^a_{t^*}}{1-\theta}\left(1 - \frac{\delta}{1-q}\right) - \left(n(1-2p-\epsilon) + \lambda^a_{t^*}\left(1 - \frac{\delta}{1-q}\right) + \left(\theta + \frac{\delta}{1-q}\right)(n - n\theta)\right) > n\theta.$$

Now, the coefficient of $\lambda^a_{t^*}$ in the above expression is $\frac{1+\theta}{1-\theta}\left(1 - \frac{\delta}{1-q}\right)$, which is always positive. Thus, we only need

$$p \leq \frac{1}{2}\left(\frac{1-\theta}{\theta}\right)\left(\epsilon - 2\theta + \theta^2 - \frac{\delta}{1-q}(1-\theta)\right). \tag{3.32}$$

Proceeding exactly like before, choosing $\delta < (1-q)\frac{\theta^2}{1-\theta}$, inequality (3.32) always holds.

Backtracking the proof steps, if we choose

$$\delta = \frac{1}{4}\frac{(1-q)\theta^2(1-\theta)}{1 + 2\theta - \theta^2},$$

all of the required conditions are satisfied and the proof of this lemma is complete.

□

Calvin's Unused Budget: We now prove an upper bound on the number of adversarial erasures that Calvin is left with to add on to the right mega sub-codeword. Since the total budget is pn, the remaining number erasures is $pn - \lambda^a_{t^*}$. From (3.25) and (3.26), for any $\hat{\lambda}_{t^*} \in \mathcal{Z}$, we have

$$pn - \lambda^a_{t^*} \leq \frac{(n - t^*)(1-\theta)}{2} + \frac{\delta(t - \lambda^a_{t^*})}{1-q}.$$

Since we are proving an achievability result and θ is representative of the back-off from the capacity expression, we can choose θ as small as we would like. Choosing θ suffciently small so that for instance $\delta = \frac{1}{4}\frac{(1-q)\theta^2(1-\theta)}{1+2\theta-\theta^2} \leq \frac{1}{16}(1-q)\theta^2$, we have

$$pn - \lambda^a_{t^*} \leq (n - t^*)\left(\frac{1}{2} - \frac{7\theta}{16}\right). \tag{3.33}$$

List Decoding: We show that with probability at least $\left(1 - \frac{1}{n}\right)$ over the code design, the size of the list of messages $\mathcal{L}$ obtained by Bob in the list-decoding phase is at most a constant, specifically, $|\mathcal{L}| < C/\epsilon$ for some constant C.

Lemma 11. *(Modified from [57, Claims B.5-B.7]) Suppose $t^* \in \mathcal{T}$ where $t^* = k^* n\theta$ satisfies (3.24), i.e.,*

$$\lambda_{t^*} - qt^* \leq t^*(1-q)(1-\theta) - Rn.$$

Then, for sufficiently large n, with probability at least $\left(1 - \frac{1}{n}\right)$ over the code design, the left mega sub-code $\mathcal{C}_1 \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_{k^*}$ is list decodable with list size L for λ_{t^*} erasures where $L = O\left(\frac{1}{\epsilon}\right)$.

Proof. Rearranging (3.24), we have

$$t^* - \lambda t^* - nR \geq \theta(1-q)t^*. \tag{3.34}$$

The proof follows exactly the analysis in [57, Claims B.5-B.7]. The only additional step is to verify if the bound

$$1 - \frac{\lambda_{t^*}}{t^*} - \frac{nR}{t^*} - \frac{S}{\theta} \geq \frac{\theta}{2}$$

holds. This is indeed the case as we have

$$\begin{aligned} 1 - \frac{\lambda_{t^*}}{t^*} - \frac{nR}{t^*} - \frac{S}{\theta} - \frac{\theta}{2} &\overset{(a)}{=} \frac{1}{t^*}\left(1 - \lambda_{t^*} - nR\right) - \frac{\theta^2}{8} - \frac{\theta}{2} \\ &\overset{(b)}{\geq} \theta(1-q) - \frac{\theta^2}{8} - \frac{\theta}{2} \\ &\overset{(c)}{\geq} 0, \end{aligned}$$

where (a) follows from the substitution $S = \frac{\theta^3}{8}$, (b) follows from (3.34) and (c) holds by choosing θ sufficiently small. □

List Refinement: For some chunk end $t \in \mathcal{T}$ where $t = kn\theta$, $\mathbf{y}_1^t = (y_1, y_2, \cdots, y_t)$ and $\mathbf{y}_{t+1}^n = (y_{t+1}, \cdots, y_n)$ are the left mega received word and the right mega received word w.r.t. t respectively. Consider the list of messages $\mathcal{L}$ obtained by Bob by list-decoding the left mega received word $\mathbf{y}_1^t$. Let u^* be the true message chosen by Alice for transmission and let $\mathcal{L}(u^*)$ be the set of all possible right mega sub-codewords w.r.t t for each message in $\mathcal{L} \setminus \{u^*\}$ i.e.

$$\mathcal{L}(u^*) = \{\mathcal{C}_{k+1}(u, s_{k+1}) \circ \mathcal{C}_{k+2}(u, s_{k+2}) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u, s_{\frac{1}{\theta}}) : u \in \mathcal{L}, u \neq u^*, (s_{k+1}, \cdots, s_{1/\theta}) \in \mathcal{S}^{\frac{1}{\theta}-k}\}.$$

For notational convenience, also enumerate $\mathcal{L}(u^*)$ containing codewords of length $(n-t)$ as $\mathcal{L}(u^*) = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{|\mathcal{L}(u^*)|}\}$. The right mega-subcodeword for the true message is

$$\mathbf{x}_{t+1}^n(u^*, s_{right}) = \mathcal{C}_{k+1}(u^*, s_{k+1}) \circ \mathcal{C}_{k+2}(u^*, s_{k+2}) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u^*, s_{\frac{1}{\theta}})$$

which we emphasize is a function of the specific realization of $s_{right=}(s_{k+1}, \cdots, s_{\frac{1}{\theta}})$ during encoding.

We would like our code design to satisfy the following distance condition

$$d_H\left(\mathbf{x}_{t+1}^n(u^*, s_{right}), \mathbf{w}_j\right) \geq (n-t)\left(\frac{1}{2} - \frac{3\theta}{8}\right) \quad \forall \mathbf{w}_j \in \mathcal{L}(u^*). \tag{3.35}$$

Equation (3.35) is a key property that guarantees successful decoding. It ensures that the right mega sub-codeword for the transmitted message is sufficiently far in Hamming distance from the right mega sub words for any of the other messages in list $\mathcal{L}$. We show that (3.35) indeed occurs w.h.p., for almost all possible sequence of secrets s_{right}.

Lemma 12. *(Modified from [57, Claims B.11-B.14]) For sufficiently large n, with probability at least $1 - 2^{-n}$, a code drawn from the random ensemble satisfies the following property : for every chunk end $t \in \mathcal{T}$, for every message u^*, and every list $\mathcal{L}$ of size at most $O(1/\epsilon)$, we have that (3.35) holds for at least a $(1 - 2^{-nS/4})$ portion of all possible secret sequences s_{right}.*

Proof. Given a sequence of secrets $s_{right} = (s_{k+1}, \cdots, s_{\frac{1}{\theta}})$, message u^* and list $\mathcal{L}$, we first show that (3.35) holds w.h.p.. Let radius $r = \left(\frac{1}{2} - \frac{3\theta}{8}\right)$. We surround each word $\mathbf{w}_j \in \mathcal{L}(u^*)$ with a Hamming ball of radius r and the union of all the balls is the so called forbidden region.

For (3.35) to hold, we must have that $\mathbf{x}_{t+1}^n(u^*, s_{right})$ is outside all these balls, i.e. outside the forbidden region. Due to the code construction, $\mathbf{x}_{t+1}^n(u^*, s_{right})$ is uniformly distributed over all possible binary vectors of length $(n-t)$ and thus it is enough to bound the size of the forbidden region. If the size of the list $\mathcal{L}$ is L, the size of $\mathcal{L}(u^*)$ is at most $L.2^{nS\left(\frac{1}{\theta} - \frac{t}{n\theta}\right)}$. Hence the number of codewords in the forbidden region is at most

$$L.2^{nS\left(\frac{1}{\theta} - \frac{t}{n\theta}\right)} \sum_{j=0}^{r} \binom{n-t}{j} < 2^{(n-t)\left(\frac{\log_2 L}{n-t} + \frac{S}{\theta} + h_2\left(\frac{1}{2} - \frac{3\theta}{8}\right)\right)}.$$

From the Taylor expansion of function $h_2(x)$ in a neighborhood of $1/2$, we have

$$\begin{aligned} h_2\left(\frac{1}{2} - \frac{3\theta}{8}\right) &< 1 - \frac{1}{2\ln(2)}\left(1 - 2\left(\frac{1}{2} - \frac{3\theta}{8}\right)\right)^2 \\ &= 1 - \frac{9\theta^2}{32\ln(2)} \end{aligned}$$

Let $\eta = \frac{\theta^2}{4}$. For sufficiently large n, we have

$$\left(\frac{\log_2 L}{n-t} + \frac{S}{\theta} + h_2\left(\frac{1}{2} - \frac{3\theta}{8}\right)\right) < \left(\frac{\log_2 L}{n-t} + \frac{S}{\theta} + \left(1 - \frac{9\theta^2}{32\ln(2)}\right)\right) < 1 - \eta.$$

Hence, the total number of codewords in the forbidden region is at most $2^{(n-t)(1-\eta)}$ and we have

$$P\Big(\mathbf{x}_{t+1}^n(u^*, s_{right}) \text{ is outside the forbidden region}\Big) > \frac{2^{(n-t)} - 2^{-(n-t)(1-\eta)}}{2^{n-t}} = 1 - 2^{-(n-t)\eta}.$$

From here on, the rest of the steps in the proof follow claims B.12-B.14 in [57]. □

Success of Unique Decoding: From the preceding discussion, there exists a code in our random ensemble that satisfies the following simultaneously:

- For t^* satisfying (3.24) and (3.25), the size of the list $\mathcal{L}$ obtained by Bob during list decoding is at most C/ϵ for some constant C. Further, the transmitted message u^* is inside list $\mathcal{L}$.

- For almost all possible realizations of secret sequences s_{right} (at least a fraction $1 - 2^{-nS/4}$ of them), the right mega codeword corresponding to message u^* denoted $\mathbf{x}_{t^*+1}^n(u^*, s_{right})$, is at least $(n - t^*)\left(\frac{1}{2} - \frac{3\theta}{8}\right)$ away from any codeword in the set $\mathcal{L}(u^*)$.

Recall from (3.33) that with probability at least $1-2^{-\Omega(\delta^2 n)}$, Calvin has at most $pn-\lambda_{t^*}^a \leq (n - t^*)\left(\frac{1}{2} - \frac{7\theta}{16}\right)$ erasures that remain.

Consider any arbitrary codeword $\mathbf{w}_j \in \mathcal{L}(u^*)$ that is associated with message $u \neq u^*$. Let $\mathcal{I}^c$ be the set of indices where $\mathbf{w}_j$ and $\mathbf{x}_{t^*+1}^n(u^*, s_{right})$ disagree. The only way that Bob is unable to distinguish between $\mathbf{w}_j$ and $\mathbf{x}_{t^*+1}^n(u^*, s_{right})$ and hence makes a decoding error of at least 1/2 is when indices $\mathcal{I}^c$ in $\mathbf{y}_{t^*+1}^n$ are all erased due to Calvin and the BEC(q). In other words, if $\mathcal{J}$ is the set of indices of erasures in $\mathbf{y}_{t^*+1}^n$, we must have $\mathcal{J} \supset \mathcal{I}^c$. An example is illustrated in Fig. 3.6.

Now clearly, if Calvin wishes to confuse Bob between $\mathbf{w}_j$ and $\mathbf{x}_{t^*+1}^n(u^*, s_{right})$, his best strategy is to add all erasures at positions $\mathcal{I}^c$. However, this still leaves at least $(n - t^*)\left(\frac{1}{2} - \frac{3\theta}{8}\right) - (n-t^*)\left(\frac{1}{2} - \frac{7\theta}{16}\right) = (n-t^*)\frac{\theta}{16}$ positions where $\mathbf{w}_j$ and $\mathbf{x}_{t+1}^n(u^*, s_{right})$ disagree but no adversarial erasures are added. For Bob to be confused between $\mathbf{w}_j$ and $\mathbf{x}_{t+1}^n(u^*, s_{right})$, the BEC($q$) must erase *all of the* $(n - t^*)\frac{\theta}{16}$ bits that Calvin could not erase. However, this event occurs with probability $q^{(n-t^*)\frac{\theta}{16}} \leq 2^{-n\Omega(\theta^2)}$. Thus, the probability of the error event that Bob cannot distinguish between $\mathbf{y}_{t^*+1}^n$ and $\mathbf{w}_j$ is exponentially small.

Repeating the same argument for any $\mathbf{w}_j \in \mathcal{L}(u^*)$, we have that a decoding error occurs with exponentially small probability. Thus, Bob succeeds in determining the transmitted message u^* and the proof is complete.

3.5.2 Achievability for BSC(q)-ADV(p)-FS

Let $\epsilon > 0$ such that $p = p + \frac{\epsilon^2}{16} < \frac{1}{4}$. We also set $\theta = \frac{\epsilon^2(1-4p)}{4}$, $S = \frac{\theta^3}{8}$. To show that $C(p, q)$ in Theorem (3.3.2) is the capacity, we let the rate be $R = C(p, q) - \epsilon$ and prove that

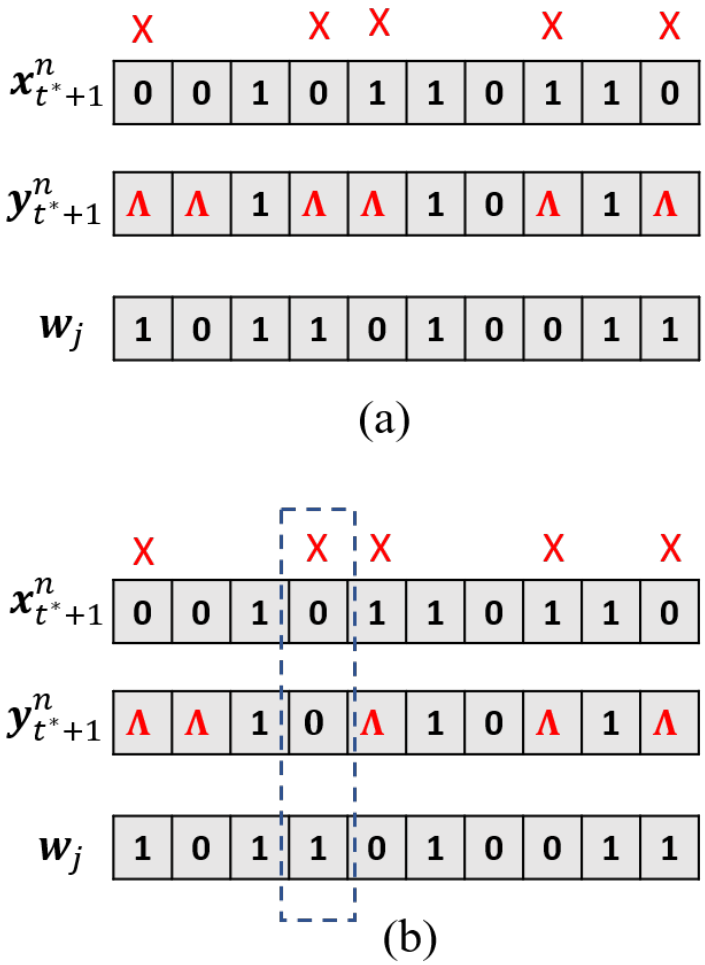


Figure 3.6. In (a), the set of indices in Bob's observation $\mathbf{y}^n_{t^*+1}$ where $\mathbf{x}^n_{t^*+1}$ and $\mathbf{w}_j$ differ are all erased. Therefore, Bob cannot determine if Alice transmitted $\mathbf{x}^n_{t^*+1}$ or $\mathbf{w}_j$. In (b), successful reception of even one bit where $\mathbf{x}^n_{t^*+1}$ and $\mathbf{w}_j$ disagree allows Bob to disambiguate between $\mathbf{x}^n_{t^*+1}$ and $\mathbf{w}_j$.

for any $\delta > 0$ and every sufficiently large block length n, a randomly sampled stochastic code $\mathcal{C}$ with rate R satisfies $P_e(\mathcal{C}) < \delta$ with a positive probability.

As before, let $\mathbf{x}$ denote the transmitted codeword. The received codeword can be written as $\mathbf{y} = \mathbf{x} \oplus \mathbf{e} \oplus \mathbf{z}$ where $\mathbf{e} = (e_1, e_2, \cdots, e_n)$ is the adversarial error vector added by Calvin and $\mathbf{z} = (z_1, z_2, \cdots, z_n)$ is the error vector produced by the BSC(q). In accordance to the power constraint, we have $d_H(\mathbf{e}, \mathbf{0}) \leq pn$. Note that positions i where $e_i = z_i = 1$, symbols x_i remain unflipped.

Before we describe the decoding process, we define certain useful quantities. For a chunk end $t \in \mathcal{T}$, let p_t be the normalized number of bit-flip attempts used up by Calvin up until time t i.e.

$$p_t = \frac{weight\{(e_1, e_2, \cdots, e_t)\}}{t}.$$

Note that since p_t only captures adversarial error injections, the word received up until time t may have more or less effective bit-flips than tp_t. For the purposes of decoding, Bob maintains a reference $\hat{p}_t$ which is approximately defined as follows: $\hat{p}_t = \frac{n}{t}\left(p - \frac{1}{4}\right) + \frac{1}{4}$ for $t \geq n(1-4p)$ and $\hat{p}_t = 0$ for $t < n(1-4p)$. It can be seen that $\hat{p}_t$ in increasing in $t \in [n(1-4p), n]$ reaching $\hat{p}_n = p$ as is expected. For a rigorous analysis, certain twiddle terms need to be added to this definition as is explained later in the analysis. We shall refer to p_t as the true trajectory and $\hat{p}_t$ as the reference trajectory for adversarial bit-flip attempts.

Decoding: The overall decoding process is iterative potentially involving several decoding attempts. For some chunk end $t \in \mathcal{T}$ where $t = kn\theta$, $\mathbf{y}_1^t = (y_1, y_2, \cdots, y_t)$ and $\mathbf{y}_{t+1}^n = (y_{t+1}, \cdots, y_n)$ are the left mega received word and the right mega received word w.r.t. t. Similarly, $\mathbf{x}_1^t = (x_1, x_2, \cdots, x_t)$ and $\mathbf{x}_{t+1}^n = (x_{t+1}, \cdots, x_n)$ are the left mega transmitted codeword and the right mega transmitted codeword w.r.t. t. A decoding attempt w.r.t t consists of two phases - a list-decoding phase followed by a unique decoding phase.

List decoding: In the list decoding phase, Bob identifies the set of messages for whom there is at least one associated codeword whose left mega sub-codeword w.r.t. t is within Hamming distance $t(\hat{p}_t \star q + \delta_1)$ from $\mathbf{y}_1^t$, where $\delta_1 = \frac{\epsilon^2}{256}$ is a small constant. In other words, Bob performs list-decoding on the left mega sub-code w.r.t. t, i.e. $\mathcal{C}_1 \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_k$, with a list-decoding radius equal to $r_{list} = t(\hat{p}_t \star q + \delta_1)$. Let the list of messages obtained in this phase be denoted by $\mathcal{L}$. We have,

$$\mathcal{L} = \{u \in \mathcal{U} : \exists\ s_{left} = (s_1, \cdots, s_k) \in \mathcal{S}^k\ s.t. d_H\left(\mathcal{C}_1(u, s_1) \circ \cdots \mathcal{C}_k(u, s_k), \mathbf{y}_2\right) \leq t(\hat{p}_t \star q + \delta_1)\}.$$

Unique decoding: In the unique decoding phase, Bob forms the set $\mathcal{A}$ of all possible right mega sub-codewords w.r.t. t (one for each possible sequence of secrets $s_{k+1}, s_{k+2}, \cdots, s_{1/\theta}$) for each message u in the list $\mathcal{L}$, i.e.,

$$\mathcal{A} = \{\mathcal{C}_{k+1}(u, s_{k+1}) \circ \mathcal{C}_{k+2}(u, s_2) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u, s_{\frac{1}{\theta}}) : u \in \mathcal{L}, (s_{k+1}, \cdots, s_{1/\theta}) \in \mathcal{S}^{\frac{1}{\theta}-k}\}.$$

He then considers Hamming balls of radius

$$r_{unique} = (n - t^*)\left(\frac{1}{4} \star q + \frac{\theta(q - 1/2)}{10}\right),$$

each centered at a right mega sub-codeword from $\mathcal{A}$.

- If $\mathbf{y}_{t+1}^n$ lies within *exactly* one of the balls, the decoder outputs the message u corresponding to its center, i.e., $\Gamma(\mathbf{y}) = u$.
- If $\mathbf{y}_{t+1}^n$ lies in more than one ball, a decoding error is declared.
- If $\mathbf{y}_{t+1}^n$ lies outside all the balls, Bob picks the next chunk end in $\mathcal{T}$ and re-attempts decoding.

As we will show, depending on the adversary's attack strategy and the noise due to the BSC, there is a value of $t = t^*$ for which the decoding-attempt successfully recovers the transmitted message. However, Bob does know this value apriori. Bob begins by first identifying the smallest value of $t \geq n(1-4p)$ that coincides with a chunk end in $\mathcal{T}$, say $t_0 \in \mathcal{T}$, and performs a decoding attempt w.r.t t_0. Clearly, $t_0 = \min\{t : t \geq n(1-4p), t \in \mathcal{T}\} = \left\lceil \frac{1-4p}{\theta} \right\rceil n\theta$. If no message is returned, he re-attempts decoding with the next chunk end, $t = t_0 + n\theta$, and so on, each time picking a chunk end from the set $\mathcal{T} = \{t_0, t_0 + n\theta, \cdots, n - n\theta\}$ until a message is returned. At any point in the decoding process, if $\mathbf{y}_{t+1}^n$ during unique decoding lies in more than one ball, a decoding error is declared and decoding terminates. If all decoding attempts fail to return a message having reached the end of the codeword, again a decoding error is declared.

Analysis: We begin our analysis with the following useful lemma.

Lemma 13. *Let $p, q \in [0, 1/2)$ and γ be a small positive constant such that $\gamma(1-2q) < 1/16$ and $p + \gamma < 1/2$. Then, we have the inequality*

$$h_2\left((p+\gamma) \star q\right) < h_2(p \star q) + 2\sqrt{\gamma}, \tag{3.36}$$

where $h_2(x \star y) = x(1-y) + y(1-x)$.

Proof. We have

$$\begin{aligned}
h_2\left((p+\gamma) \star q\right) &= h_2\left((p+\gamma)(1-q) + q(1-p-\gamma)\right) \\
&= h_2\left((p \star q) + \gamma(1-2q)\right) \\
&\overset{(a)}{<} h_2(p \star q) + 2\gamma(1-2q)\log_2\left(\frac{1}{2\gamma - 4\gamma q}\right) \\
&\overset{(b)}{<} h_2(p \star q) + 2\sqrt{\gamma(1-2q)} \\
&\overset{(c)}{\leq} h_2(p \star q) + 2\sqrt{\gamma},
\end{aligned}$$

where (a) follows from the inequality $h_2(a+b) < h_2(a) + 2b\log_2\left(\frac{1}{b}\right)$ (see for example [57, Lemma A.5] for a proof), (b) follows from the fact that $x\log_2\left(\frac{1}{x}\right) < \sqrt{x}$ when $x < \frac{1}{16}$ and (c) is true because $(1-2q) \in (0, 1]$. □

Reference trajectory $\hat{p}_t$: We now give an exact definition of $\hat{p}_t$, the reference trajectory for adversarial bit-flip attempts. It suffices to use the same $\hat{p}_t$ as defined in [57] where no BSC was present ($q = 0$) i.e. the decoder sets $\hat{p}_t$ independent of q.

Definition 3.5.1. *(Definition of $\hat{p}_t$) Let $t \in \mathcal{T}$ be some chunk-end and recall $p = p + \frac{\epsilon^2}{16}$. Define,*

$$\bar{p}_t = p - \frac{(n-t)}{4n}.$$

For $t < n(1-4p)$, $\hat{p}_t = 0$. For $t \geq n(1-4p)$, $\hat{p}_t$ is defined to be

$$\hat{p}_t = \frac{\bar{p}_t}{\alpha(p, \bar{p}_t)} + \frac{\epsilon^2}{16\alpha^2(p, \bar{p}_t)},$$

where

$$\alpha(p, \bar{p}_t) = 1 - 4(p - \bar{p}_t) = \frac{t}{n}.$$

In the following lemma, we prove that $\hat{p}_t$ satisfies two key technical conditions, the so-called *list decoding condition* given by (3.37), and the *energy bounding condition* given by (3.38).

Lemma 14. *(Modified from [57, Claim A.6]) For any $t \in \mathcal{T}$ such that $t \geq n(1-4p)$, the reference trajectory $\hat{p}_t$ satisfies*

$$t\left(1 - h_2\left(\hat{p}_t \star q\right)\right) - \frac{n\epsilon}{2} \geq nR \tag{3.37}$$

and

$$pn - t\hat{p}_t \leq (n-t)\left(\frac{1}{4} - \frac{\epsilon^2}{16}\right). \tag{3.38}$$

Proof. Note that (3.38) follows directly from [57, Claim A.6] as it does not involve q. We only need to verify that (3.37) holds. Diving (3.37) by n and noting that $\alpha(p, \bar{p}_t) = t/n$, we need to show that

$$\alpha(p, \bar{p}_t)\left(1 - h_2\left(\hat{p}_t \star q\right)\right) - \frac{\epsilon}{2} \geq R.$$

Substituting in the value of $\hat{p}_t$, we have

$$\begin{aligned}
&\alpha(p, \bar{p}_t)\left(1 - h_2\left(\left(\frac{\bar{p}_t}{\alpha(p, \bar{p}_t)} + \frac{\epsilon^2}{16\alpha^2(p, \bar{p}_t)}\right) \star q\right)\right) - \frac{\epsilon}{2} \\
&\overset{(a)}{\geq} \alpha(p, \bar{p}_t)\left(1 - h_2\left(\frac{\bar{p}_t}{\alpha(p, \bar{p}_t)} \star q\right) - 2\sqrt{\frac{\epsilon^2}{16\alpha^2(p, \bar{p}_t)}}\right) - \frac{\epsilon}{2} \\
&= \alpha(p, \bar{p}_t)\left(1 - h_2\left(\frac{\bar{p}_t}{\alpha(p, \bar{p}_t)} \star q\right)\right) - \epsilon \\
&\geq \min_{\bar{p}_t \in [0,p]} \alpha(p, \bar{p}_t)\left(1 - h_2\left(\frac{\bar{p}_t}{\alpha(p, \bar{p}_t)} \star q\right)\right) - \epsilon \\
&= C(p, q) - \epsilon \\
&= R,
\end{aligned}$$

proving the result, where inequality (a) follows from (3.36) proven in Lemma 13. □

Correct Decoding Point t^*: From [57, Section A.3], for any trajectory p_t chosen by Calvin, Bob's reference trajectory $\hat{p}_t$ intersects p_t at some point before the second to last chunk end. Further, there is a $t^* \in \mathcal{T} = \{t_0, t_0 + n\theta, \cdots, n - n\theta\}$ such that

$$\forall t \in \{t_0, t_0 + n\theta, \cdots, t^* - n\theta\}, \quad p_t > \hat{p}_t, \tag{3.39}$$

$$p_{t^*} \leq \hat{p}_{t^*}, \tag{3.40}$$

and

$$\forall t \in \{t_0, t_0 + n\theta, \cdots, t^*\}, \quad pn - tp_t \leq (n-t)\left(\frac{1}{4} - \frac{\epsilon^2}{16}\right). \tag{3.41}$$

As we will argue later, t^* defined above turns out to be the correct decoding point, where the two phase decoding attempt succeeds in finding the true message.

Code Properties: We now show that a code drawn at random from our ensemble satisfies with a positive probability two key properties.

a) **Property I - List Decoding Property:** This property will be used to prove that the size of the list obtained by Bob in a decoding attempt is at most a constant $O(1/\epsilon)$. We state it as the following lemma.

Lemma 15. *(Modified from [57, Claims A.15-A.16]) Suppose $t \in \mathcal{T} = \{t_0, t_0 + n\theta, \cdots, n - n\theta\}$ where $t = kn\theta$ satisfies (3.37), i.e.*

$$t\left(1 - h_2\left(\hat{p}_t \star q\right)\right) - \frac{n\epsilon}{2} \geq nR.$$

Then, for sufficiently large n, with probability at least $\left(1 - \frac{1}{np}\right)$ over the code design, the left mega sub-code $\mathcal{C}_1 \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_k$ is list decodable with radius $r = t\left(\hat{p}_t \star q + \frac{\epsilon^2}{256}\right)$ and list size $L = O\left(\frac{1}{\epsilon}\right)$.

Proof. The proof follows exactly the analysis in [57, Claims A.15-A.16]. The only additional step is to verify the bound

$$1 - h_2\left(\hat{p}_t \star q + \frac{\epsilon^2}{256}\right) - \frac{nR}{t} - \frac{nS}{t\theta} \geq \frac{\epsilon}{4}.$$

Since $\theta = \frac{\epsilon^2(1-4p)}{4}$, $S = \frac{\theta^3}{8}$, from Lemma 13 and given that (3.37) is true, we have

$$\begin{aligned} 1 - h_2\left(\hat{p}_t \star q + \frac{\epsilon^2}{256}\right) - \frac{nR}{t} - \frac{nS}{t\theta} &\geq \frac{n\epsilon}{2t} - \frac{n\theta^2}{8t} - 2\sqrt{\frac{\epsilon^2}{256}} \\ &\geq \frac{n}{2}\left(\frac{3\epsilon}{8} - \frac{\theta^2}{8}\right) \\ &\geq \frac{\epsilon}{4} \end{aligned}$$

as desired. □

a) **Property II - Minimum Distance Property:** Now, for a decoding attempt at $t \in \mathcal{T}$, consider the list of messages $\mathcal{L}$ obtained by Bob in the list-decoding phase. Let u^* be the true message chosen by Alice for transmission and let $\mathcal{L}(u^*)$ be the set of all possible right mega sub-codewords w.r.t t for each message in $\mathcal{L} \setminus \{u^*\}$ i.e.

$$\mathcal{L}(u^*) = \{\mathcal{C}_{k+1}(u, s_{k+1}) \circ \mathcal{C}_{k+2}(u, s_{k+2}) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u, s_{\frac{1}{\theta}}) : u \in \mathcal{L}, u \neq u^*, (s_{k+1}, \cdots, s_{1/\theta}) \in \mathcal{S}^{\frac{1}{\theta}-k}\}.$$

For notational convenience, also enumerate $\mathcal{L}(u^*)$ containing codewords of length $(n-t)$ as $\mathcal{L}(u^*) = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{|\mathcal{L}(u^*)|}\}$. The right mega-subcodeword for the true message is

$$\mathbf{x}_{t+1}^n(u^*, s_{right}) = \mathcal{C}_{k+1}(u^*, s_{k+1}) \circ \mathcal{C}_{k+2}(u^*, s_{k+2}) \circ \cdots \mathcal{C}_{\frac{1}{\theta}}(u^*, s_{\frac{1}{\theta}})$$

which we emphasize is a function of the specific realization of $s_{right=}(s_{k+1}, \cdots, s_{\frac{1}{\theta}})$ during encoding.

We would like our code to satisfy the following distance condition

$$d_H\left(\mathbf{x}_{t+1}^n(u^*, s_{right}), \mathbf{w}_j\right) \geq (n-t)\left(\frac{1}{2} - \frac{\theta}{2}\right) \quad \forall \mathbf{w}_j \in \mathcal{L}(u^*). \tag{3.42}$$

Equation (3.42) is a key property that guarantees successful decoding. It ensures that the right mega sub-codeword for the transmitted message is sufficiently far in Hamming distance from the right mega sub words for any of the other messages in list $\mathcal{L}$. From [57, Claims A.20-A.23], (3.35) indeed occurs w.h.p., for almost all possible sequence of secrets s_{right}. We state this as the following lemma.

Lemma 16. *([57, Claims A.20-A.23]) For sufficiently large n, with probability at least $1-2^{-n}$, a code drawn from the random ensemble satisfies the following property : for every chunk end $t \in \mathcal{T}$, for every message u^*, and every list $\mathcal{L}$ of size at most $O(1/\epsilon)$, we have that (3.42) holds for at least a $(1-2^{-nS/4})$ portion of all possible secret sequences s_{right}.*

Success of Decoding Procedure: We are now ready to argue that the iterative decoding process succeeds in finding the true message with high probability. Suppose we fix a stochastic code $\mathcal{C} = \mathcal{C}_1 \circ \mathcal{C}_2 \circ \cdots \mathcal{C}_{1/\theta}$ for which both the list decoding property and the minimum distance property are satisfied, which we can do thanks to Lemmas 15 and 16. We will show that $t = t^*$ as defined by (3.39), (3.40) and (3.41) is in fact the correct decoding point i.e. at t^*, the list $\mathcal{L}$ obtained in the list decoding phase contains the true message which is then returned in the unique decoding phase.

Success of list decoding: When $t = t^*$, we have $\hat{p}_{t*} \geq p_{t^*}$. Thus, the number of adversarial bit-flip attempts injected onto $\mathbf{y}_1^{t^*}$, the left mega received word w.r.t. t^* is at most $t^*\hat{p}_{t^*}$. From Lemma 5 then, we have that $d_H(\mathbf{x}_1^{t^*}, \mathbf{y}_1^{t^*}) \leq t^* \left(\hat{p}_{t^*} \star q + \frac{\epsilon^2}{256}\right)$ with probability at least $1-2^{-\Omega(\epsilon^4 n)}$. Since the list-decoding radius is selected to be $r_{list} = t^* \left(\hat{p}_{t^*} \star q + \frac{\epsilon^2}{256}\right)$, the transmitted message is indeed in the list $\mathcal{L}$ with high probability as required.

Also note that when $t < t^*$, i.e., for $t \in \{t_0, t_0 + n\theta, \cdots, t^* - n\theta\}$, we have by the definition of t^* that $p_t > \hat{p}_t$. By a similar argument as in Lemma 5 then, $\mathbf{y}_1^t$, the left mega received word w.r.t. t, lies w.h.p. outside the Hammming ball $\mathcal{B}(\mathbf{x}_1^t, r_{list})$. In other words, when $t < t^*$, the transmitted message u^* is w.h.p. not in the list $\mathcal{L}$ obtained by Bob.

Success of unique decoding: For $t_0 \leq t \leq t^*$, our code for almost all key sequences s_{right} satisfies

$$d_H\left(\mathbf{x}_{t+1}^n(u^*, s_{right}), \mathbf{w}_j\right) \geq (n-t)\left(\frac{1}{2} - \frac{\theta}{2}\right) \ \forall \mathbf{w}_j \in \mathcal{L}(u^*), \tag{3.43}$$

where recall that $\mathbf{w}_j$'s are the right-mega subcodewords corresponding to messages in $\mathcal{L}$ excluding u^*. Further, we also have that Calvin has at most $(n-t)\left(\frac{1}{4} - \frac{\epsilon^2}{16}\right)$ bit-flip attempts left to inject onto $\mathbf{x}_{t+1}^n$.

Recall that Bob considers Hamming balls of radius $r_{unique} = (n-t)\left(\frac{1}{4} \star q + \frac{\theta(q-1/2)}{10}\right)$ that are each centered at right-mega subcodewords in $\mathcal{L}$.

When $t_0 \leq t < t^*$, the true message $u^* \notin \mathcal{L}$ while at $t = t^*$ we have that $u^* \in \mathcal{L}$. Fortunately at $t = t^*$, thanks to Lemma 5, we have that for any error pattern induced by Calvin,

$$d(\mathbf{x}_{t^*+1}^n, \mathbf{y}_{t^*+1}^n) \leq (n - t^*) \left(\frac{1}{4} \star q + \frac{\theta(q - 1/2)}{10} \right) = r_{unique}$$

with probability at least $1 - 2^{-\Omega(\theta^2 n)}$, i.e. $\mathbf{y}_{t^*+1}^n$ is indeed w.h.p.inside the Hamming ball $\mathcal{B}(\mathbf{x}_{t^*+1}^n, r_{unique})$.

Next, consider any $t_0 \leq t \leq t^*$ and $\mathbf{w}_j$ from the set $\mathcal{L}(u^*)$. We argue that as required, no matter what Calvin does, $\mathbf{y}_{t+1}^n$ is outside $\mathcal{B}(\mathbf{w}_j, r)$. Let $\mathcal{I}$ be the set of indices where $\mathbf{w}_j$ and $\mathbf{x}_{t+1}^n(u^*, s_{right})$ agree and $\mathcal{I}^c$ be the set of indices where they disagree. For a vector $\mathbf{v}$, let $(\mathbf{v})_{\mathcal{I}}$ denote $\mathbf{v}$ restricted to indices from $\mathcal{I}$. We have that

$$d_H(\mathbf{x}_{t+1}^n, \mathbf{y}_{t+1}^n) = d_H\left((\mathbf{x}_{t+1}^n)_{\mathcal{I}}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}}\right) + d_H\left((\mathbf{x}_{t+1}^n)_{\mathcal{I}^c}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}^c}\right)$$

and

$$d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n) = d_H\left((\mathbf{w}_j)_{\mathcal{I}}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}}\right) + d_H\left((\mathbf{w}_j)_{\mathcal{I}^c}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}^c}\right) \tag{3.44}$$

Bob decodes $\mathbf{y}_{t+1}^n$ incorrectly to $\mathbf{w}_j$ when $d_H(\mathbf{x}_{t+1}^n, \mathbf{y}_{t+1}^n) > r$ and $d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n) \leq r$. Calvin's desire is to inject his remaining bit-flip attempts in such a way that $\mathbf{y}_{t+1}^n$ is as far away as possible from $\mathbf{x}_{t+1}^n$, and at the same time, as close as possible to $\mathbf{w}_j$. Clearly, the best strategy is then to only inject bit-flip attempts onto $(\mathbf{x}_{t+1}^n)_{\mathcal{I}^c}$.

Then, since $(\mathbf{x}_{t+1}^n)_{\mathcal{I}}$ only suffers corruption due to the BSC(q), by the Chernoff bound we have

$$d_H\left((\mathbf{x}_{t+1}^n)_{\mathcal{I}}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}}\right) \geq |\mathcal{I}|\,(q - \eta_1) \tag{3.45}$$

with probability at least $(1 - 2^{-\Omega(\eta_1^2 n)})$. By Lemma 5 for $\mathcal{I}^c$, we also have

$$d_H\left((\mathbf{x}_{t+1}^n)_{\mathcal{I}^c}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}^c}\right) \leq |\mathcal{I}^c| \left(\left(\frac{(n - t^*)\left(\frac{1}{4} - \frac{\epsilon^2}{16}\right)}{|\mathcal{I}^c|} \right) \star q + \eta_2 \right) \tag{3.46}$$

with probability at least $(1-2^{-\Omega(\eta_2^2 n)})$. By definition of $\mathcal{I}$ and $\mathcal{I}^c$, (3.45) and (3.46) then imply that

$$d_H\left((\mathbf{w}_j)_{\mathcal{I}}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}}\right) \geq |\mathcal{I}|\,(q-\eta_1) \tag{3.47}$$

and

$$d_H\left((\mathbf{w}_j)_{\mathcal{I}^c}, (\mathbf{y}_{t+1}^n)_{\mathcal{I}^c}\right) \geq |\mathcal{I}^c|\left(1-\left(\frac{(n-t)\left(\frac{1}{4}-\frac{\epsilon^2}{16}\right)}{|\mathcal{I}^c|}\right)\star q-\eta_2\right). \tag{3.48}$$

Now, from lemma 3, since $\left(\frac{(n-t)\left(\frac{1}{4}-\frac{\epsilon^2}{16}\right)}{|\mathcal{I}^c|}\right) < \frac{1}{2}$, there exists a constant $\delta_1 > 0$ that is only of ϵ, q and p such that

$$\left(\frac{(n-t)\left(\frac{1}{4}-\frac{\epsilon^2}{16}\right)}{|\mathcal{I}^c|}\right)\star q = \left(\frac{1}{2}-\delta_1\right)$$

We have then from (3.44)

$$d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n) \geq |\mathcal{I}|\,(q-\eta_1) + |\mathcal{I}^c|\left(1-\left(\frac{1}{2}-\delta_1\right)-\eta_2\right).$$

We wish to show a lower bound on $d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n)$. The worst case occurs when (3.43) holds with equality, i.e. $|\mathcal{I}^c| = (n-t)\left(\frac{1}{2}-\frac{\theta}{2}\right)$. Making the choice $\eta_2 = \delta_1$, we have

$$\begin{aligned} d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n) &\geq |\mathcal{I}|\,(q-\eta_1) + \frac{|\mathcal{I}^c|}{2} \\ &= (n-t)\left(\left(\frac{1+\theta}{2}\right)(q-\eta_1) + \left(\frac{1-\theta}{4}\right)\right) \\ &\overset{(a)}{\geq} (n-t)\left(\frac{1}{4}\star q + \eta_1\right) \end{aligned}$$

where (a) follows by choosing $\eta_1 \leq \frac{2\theta(q-1/2)}{3+\theta}$. Choosing, $\eta_2 = \frac{\theta(q-1/2)}{3+\theta}$, we have that w.h.p.

$$d_H(\mathbf{w}_j, \mathbf{y}_{t+1}^n) \geq (n-t)\left(\frac{1}{4}\star q + \frac{\theta(q-1/2)}{3+\theta}\right) \overset{(a)}{>} r_{unique},$$

where (a) follows by choosing ϵ (hence θ) small enough so that $\frac{\theta}{\theta+3} > \frac{\theta}{5}$. The same argument can be repeated considering any $\mathbf{w}_j \in \mathcal{L}(u^*)$.

Summarising, we have shown that w.h.p.

- when $t_0 \leq t < t^*$, the transmitted message u^* is not in the list obtained by Bob, and we have that $d(\mathbf{y}_{t+1}^n, \mathbf{w}_j) > r_{unique}$ for all $\mathbf{w}_j \in \mathcal{L}(u^*)$.
- when $t = t^*$, the transmitted message u^* is indeed in the list obtained by Bob. Further we have $d(\mathbf{y}_{t+1}^n, \mathbf{x}_{t+1}^n) \leq r_{unique}$ and $d(\mathbf{y}_{t+1}^n, \mathbf{w}_j) > r_{unique}$ for all $\mathbf{w}_j \in \mathcal{L}(u^*)$.

Thus, the iterative decoding procedure used by Bob succeeds in finding the true message u^*.

3.6 Capacity with Transmitter Feedback

Suppose now that Alice in addition to Calvin has access to Bob's reception perfectly through a separate causal feedback link (see Fig. 3.7). This allows Alice to employ *closed-loop* encoding strategies where the input x_k at time k is possibly a function of both the message and Bob's reception thus far $(y_1, y_2, \cdots, y_{k-1})$, i.e.,

$$\mathbf{X}_k \sim f_k(\mathbf{U}, \mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_{k-1}) \quad k = 1, 2, \cdots, n \tag{3.49}$$

where for each k, f_k is either deterministic or, more generally, a probabilistic map defining a conditional distribution $P_{\mathbf{X}|\mathbf{U},\mathbf{Y}_1,\mathbf{Y}_2,\cdots,\mathbf{Y}_{k-1}}$ over $\mathcal{X}$. Calvin is assumed to be causal as before. He does not know the message but knows the closed-loop encoding (possibly stochastic) maps $\{f_k\}_{k=1}^n$ used by Alice. Let the capacities in this case be denoted as $C_f^E(p, q)$ and $C_f(p, q)$ respectively.

3.6.1 BEC(q)-ADV(p)-FS with Transmitter Feedback

Theorem 3.6.1. *The capacity $C_f^E(p, q)$ of BEC(q)-ADV(p)-FS with causal feedback to the transmitter is*

$$C_f^E(p, q) = (1 - p)(1 - q) \quad \forall\ 0 \leq p \leq 1, 0 \leq q \leq 1. \tag{3.50}$$

Remark. If Calvin were to simply erase each symbol with probability p, the rate is limited to[1] $(1-p)(1-q)$ which matches with the expression in (3.50). This implies that *the optimal*

[1]↑For a vanilla DMC such as the BEC, the capacity is the same under deterministic and stochastic encoding [49].

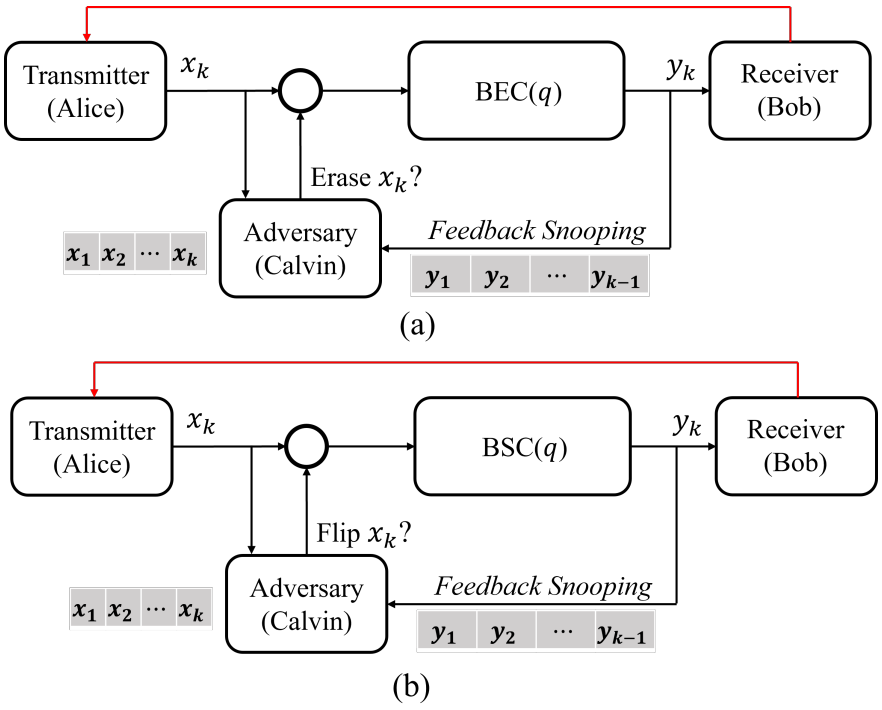


Figure 3.7. Channel models with transmitter feedback

attack for the adversary is to simply cause i.i.d. erasures. The knowledge of the (closed-loop) encoding scheme or the ability to snoop into Bob's reception does not buy Calvin any benefit.

Proof. First, we show the converse. Fix $\epsilon > 0$. Calvin simply erases each symbol with probability $p - \frac{\epsilon}{1-q}$. By the Chernoff bound, the probability that Calvin will run out of his budget of pn erasures is at most $1 - 2^{-\Omega(\epsilon^2 n)}$. The combined effect of the adversary and the BEC(q) then is a BEC with erasure probability $s = p + q - pq - \epsilon$. Since causal feedback does not increase capacity for a BEC, $C_f^E(p, q) \leq 1 - s = (1 - p)(1 - q) + \epsilon$.

The achievability scheme is essentially an ARQ scheme - transmit each of the k bits in the message repeatedly until it is successfully received. If e_Λ is the total number of erasures (a random quantity) that occur due to both the actions of Calvin and the BEC(q), Alice needs $n = k + e_\Lambda$ channel uses for this scheme to succeed. From Lemma 4 however, we have that $P\left(e_\Lambda > ((p + q - pq) + \epsilon)n\right)$ is at most $1-2^{-\Omega(\epsilon^2 n)}$ and hence, $C_f^E(p, q) \geq (1-p)(1-q)-\epsilon$. □

3.6.2 BSC(q)-ADV(p)-FS with Transmitter Feedback : A Conjecture

For the BSC(q)-ADV(p)-FS channel with causal transmitter feedback, we prove a simple achievable rate and conjecture on the true capacity expression. Proving stronger converse and achievability results is left for future work.

First, consider the simpler case where there is no BSC, i.e. $q = 0$, and Calvin knows the message Alice wants to transmit. Then, a tight capacity characterization $\tilde{C}_f(p)$ for this scenario is implied by the work of Berlekamp [36] and Zigangirov [72]:

$$\tilde{C}_f(p) = \begin{cases} 1 - h_2(p) & 0 \leq p \leq p_f \\ R_0(1-3p) & p_f \leq p \leq \frac{1}{3} \\ 0 & \frac{1}{3} \leq p \leq \frac{1}{2} \end{cases}, \tag{3.51}$$

where $p_f = \frac{1}{3+\sqrt{5}}$ and $R_0 = \log_2\left(\frac{1+\sqrt{5}}{2}\right)$. Interestingly, analogous to our result in theorem (3.3.2), the capacity curve in (3.51) has two parts. The first is a convex part that is equal to the random i.i.d. noise capacity $1 - h_2(p)$ when $p < p_f$. For $p \geq p_f$, the curve is a tangent to the function $1 - h_2(p)$ with abscissa at $p = \frac{1}{3}$. It can be shown that (3.51) admits a form that is very similar to our result in Theorem 3.3.2 where $q = 0$, and 4 is instead replaced with a 3, i.e.,

$$\tilde{C}_f(p) = \min_{x \in [0,p]} (1 - 3(p-x)) \left(1 - h_2\left(\frac{x}{1-3(p-x)}\right)\right). \tag{3.52}$$

Now, since an adversary who knows Alice's message is stronger than one who does not, using (3.51) and the result in theorem 5, we obtain the following simple lower bound to $C_f(p, q)$.

Lemma 17. *The capacity $C_f(p, q)$ of BSC(q)-ADV(p)-FS channel with causal transmitter feedback is lower bounded as $C_f(p, q) \geq \tilde{C}_f(p \star q)$.*

Remark. Lemma 17 implies that when $p = \frac{1}{4}$ and q is small enough, it is possible to achieve a non-zero rate when Alice has feedback, i.e. $C_f\left(\frac{1}{4}, q\right) > 0$. Contrast this to the fact that without transmitter feedback, $C\left(\frac{1}{4}, q\right) = 0$ for all values of q.

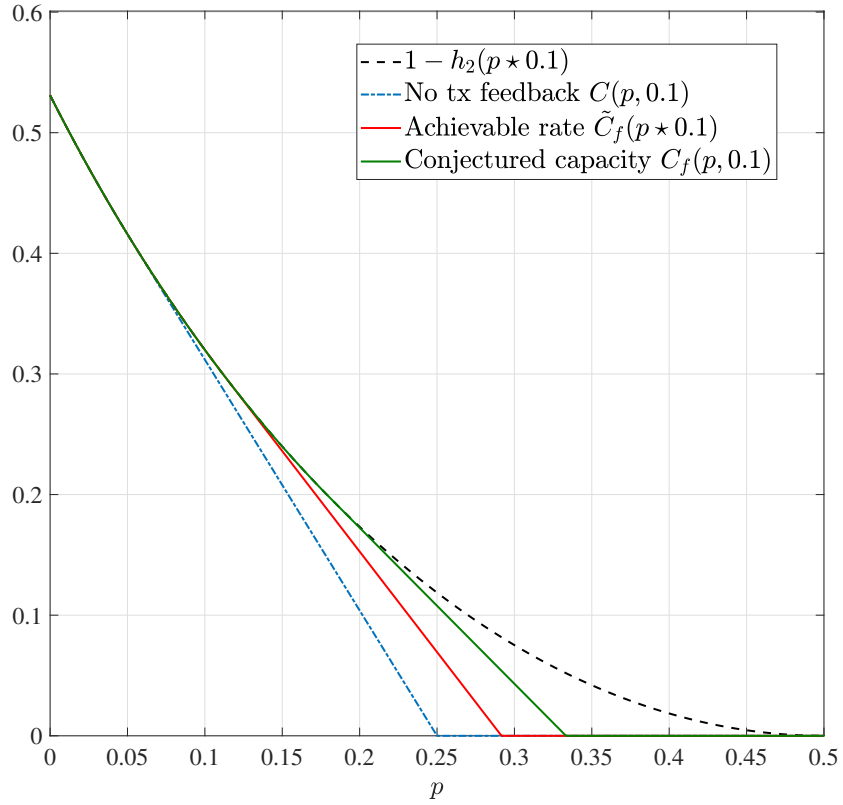


Figure 3.8. Capacity of BSC(q)-ADV(p)-FS when $q = 0.1$. Here, the presence of transmitter feedback provably increases the capacity for all values of p.

Conjecture: We conjecture that the true capacity of BSC(q)-ADV(p)-FS with feedback to the transmitter is given by

$$C_f(p, q) = \min_{\bar{p}:\bar{p}\in\mathcal{P}} \alpha_f(p, \bar{p}) \left(1 - h_2\left(\frac{\bar{p}}{\alpha_f(p, \bar{p})} \star q\right)\right), \tag{3.53}$$

where

$$\alpha_f(p, \bar{p}) = 1 - 3(p - \bar{p}) \quad , \quad \mathcal{P} = \{\bar{p} : 0 \leq \bar{p} \leq p\}.$$

An example for $q = 0.1$ is plotted in Fig. 3.8.

3.7 Concluding Remarks

In this work, we studied the problem of communicating over a stochastic channel (BEC/BSC) in the presence of a powerful adversary who can spy on both communicating terminals and inject further erasures/bit-flips at the input of the channel. We then gave a complete capacity characterization for both models. Our work implies that in certain cases, an adversary that is weak enough is no better than an i.i.d. memory-less noise source. We also considered extensions to our models by introducing transmitter feedback. Future work includes proving our conjecture in (3.53). Another interesting direction is to characterize capacity in the case where the adversary has no feedback snooping.

4. LINEAR BLOCK FEEDBACK ENCODING AND A NOVEL SYSTEMATIC REPRESENTATION FOR REED-MULLER CODES

V. Suresh and D. J. Love, "A Novel Systematic Representation of Reed-Muller Codes with an Application to Linear Block Feedback Encoding," *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 677-682.

4.1 Introduction

The problem of coding with feedback dates back to Shannon who showed that even perfect causal feedback does not increase the capacity of discrete memoryless channels (DMCs). However, a myriad of works since then have shown that feedback can indeed be useful in simplifying the communication strategy or improving the error exponent i.e. the rate at which the probability of error decays with block-length. The most famous example of this is for the point to point AWGN channel described in a beautiful seminal work by Schalkwijk and Kailath (the S-K scheme) [73], [74], later extended to other scenarios [75], [76]. The general idea is to send an *uncoded* signal in the first channel use and then perform linear processing of the feedback signal (mainly through linear encoding of the error realizations or current estimate error at the receiver) in subsequent channel uses. Remarkably, not only does the S-K scheme achieve any rate up to capacity but does so with a probability of error that decaying doubly exponentially with block length. Thus, linear feedback processing enables a dramatic reduction in complexity while simultaneously improving reliability. Since then, linear schemes have found applicability for a host of other problems (see for eg. [77]–[79]).

The situation in the case of discrete channels is less dramatic. It is known that for symmetric DMC's with perfect causal feedback, *fixed-length* block codes do not improve either the capacity or the error exponent [80]. Hence, beginning with the seminal work of Burnashev [81], *variable-length* block coding with feedback which he showed indeed improves the error exponent, has received much attention both from a theoretical and practical interest [82]–[85].

In this work, we deal with binary codes and revisit the problem of *fixed-length* block coding with feedback over a binary symmetric channel (BSC) and provide new original perspectives.

1. Recall that the capacity of $BSC(p)$ where p is the probability of bit-flip is $C(p) = 1 - h(p)$. In the spirit of the S-K scheme, we ask the following question -

   Q 1. *Can uncoded transmission followed by linear feedback processing achieve capacity over a $BSC(p)$?*

   Consider the following simple strategy - the transmitter begins with an uncoded transmission of the message bits $\mathbf{m} = (m_0, m_1, \cdots, m_{K-1})$. The received bits $\mathbf{y} = (y_0, y_1, \cdots, y_{K-1})$ are relayed to the transmitter by means of a feedback channel. Knowing $\mathbf{m}$, the transmitter ascertains the error bits $\mathbf{e} = (e_0, e_1, \cdots, e_{K-1})$ that occured in the initial transmission. At this point, the initial error bits $\mathbf{e} = (e_0, e_1, \cdots, e_{K-1})$ can be thought of as the "new message" to be communicated to the receiver for successful decoding. This can be achieved by linear processing of feedback signals $e_0, e_1, \cdots, e_{K-1}$ in accordance to some capacity achieving open-loop code. While this strategy is natural, it suffers from a rate-loss. In particular, the best rate achievable by this strategy over a $BSC(p)$ is $\frac{C(p)}{1+C(p)}$ which is strictly less than $C(p)$. Can we do better? In this work, we will show this is indeed the case and answer Q1 in the affirmative.

2. Suppose that a given system is equipped with a relatively weak open-loop code operating at some rate. We show that we can enhance the error resilience of this system by augmenting it with a simple feedback and linear processing setup. In fact, it is shown that a weak code can effectively be strengthened to be as good as any desired code by introducing a *linear noise-shaping* component during encoding. A random coding type result shows that a random i.i.d. noise-shaping matrix is capacity-achieving. Finally, we also show that the transmission of parity bits in an open loop code and noise-shaped bits in a closed loop setting are quite intimately related.

3. We next consider feedback encoding under feedback limitations namely, (i) infrequent compressed feedback and (ii) delayed feedback. In the first case, the receiver sends

linearly compressed versions of the bits received thus far, in only a few channel uses. In the case of delayed feedback, there is a delay between the time that the receiver sends a feedback symbol and when it is received at the transmitter. We characterize a class of codes that can be emulated by noise-shaping for both cases and provide many illustrative examples.

4. Reed-Muller codes are a powerful class of codes with excellent performance that are conjectured to be capacity-achieving. We prove a new result that shows that Reed-Muller codes admit a special systematic representation. While this result is of independent interest, we use it to demonstrate that uncoded transmission combined with feedback processing can be made to mimic a Reed-Muller code against remarkably large feedback delays.

4.2 System Model and Problem Statement

4.2.1 Input-Output Expressions and Assumptions

Consider a memoryless binary symmetric channel (BSC) with an input-output for each channel use i of

$$y_i = c_i \oplus e_i \tag{4.1}$$

where $y_i \in \{0,1\}$ is the received bit, $c_i \in \{0,1\}$ is the transmitted bit, $e_i \in \{0,1\}$ is memoryless additive noise with $P(e_i = 1) = 1 - P(e_i = 0) = p$, $\oplus$ is mod-2 addition. Traditional error control coding analysis is *open-loop*, meaning that each c_i is generated independently of e_j for all j (i.e., the transmitted signal is independent of all past, current, and future noise realizations). In practice, this means that the transmitter does not have any side information about the received signal during encoding.

We consider the problem of *closed-loop* error control coding and assume that the transmitter has causal access to the received signal. This means that prior to channel use i the transmitter has perfect knowledge of $\{y_j\}_{j=0}^{i-1}$. Because the transmitter perfectly knows $\{c_j\}_{j=0}^{i-1}$, this side information is equivalent to the transmitter having knowledge of the past

noise realizations $\{e_j\}_{j=0}^{i-1}$ prior to channel use i. This assumption is later suitably modified in Section 4.6 where compressed and delayed feedback are considered.

The transmitter must encode a message $m \in \{1, \ldots, 2^K\}$ to convey to the receiver. The transmitter encodes using a function

$$c_i = \phi_{closed,i}\left(m, \{e_j\}_{j=0}^{i-1}\right). \tag{4.2}$$

We focus on block coding. where m is encoded only over N channel uses to generate the codeword $\mathbf{c} = [c_0\ c_1\ \cdots\ c_{N-1}]$. In this paper, we investigate the design of encoding functions.

4.2.2 Review of Open-Loop Coding

In open-loop error control coding, the transmitter sends the message m over N channel uses by constructing a vector $\mathbf{c}$ according to $\mathbf{c} = \phi_{open}(m)$ where ϕ_{open} is an encoding function and N is the blocklength. This can be thought of as using a codebook $\mathcal{C} = \{\phi_{open}(1), \phi_{open}(2), \ldots, \phi_{open}(2^K)\}$, which is known to both the transmitter and receiver. The receiver then decodes using the vector input-output expression

$$\mathbf{y} = \mathbf{c} \oplus \mathbf{e} \tag{4.3}$$

where $\mathbf{y} = [y_0\ y_1\ \cdots\ y_{N-1}]$, $\mathbf{e} = [e_0\ e_1\ \cdots\ e_{N-1}]$. Because the noise is independent and identically distributed (i.i.d.) Bernoulli, the minimum probability of error receiver decodes to the vector from codebook $\mathcal{C}$ that is nearest to $\mathbf{y}$ in Hamming distance.

Most of the popular error control codes are linear block codes. These use the fact that the message m can be alternatively written as a K-bit vector $\mathbf{m} \in \{0,1\}^K$. A linear block code utilizes an encoding function

$$\mathbf{c} = \phi_{open}(m) = \mathbf{mG} \tag{4.4}$$

where $\mathbf{G} \in \{0,1\}^{K \times N}$ is the generator matrix of the code. The codewords together form a linear vector space, the row space of the generator.

The subspace structure of a linear block code means that the receiver in can be efficiently implemented using syndrome decoding. The syndrome decoder uses the parity check matrix $\mathbf{H} \in \{0,1\}^{(N-K)\times N}$ satisfying $\mathbf{c}\mathbf{H}^T = 0$ for all $\mathbf{c} \in \mathcal{C}$. Given the received bit vector $\mathbf{y}$, the receiver computes a syndrome $\mathbf{s} = \mathbf{y}\mathbf{H}^T$. The receiver then decodes an error pattern

$$\mathbf{e}_{rec} = \arg\min_{\mathbf{e}:\mathbf{e}\mathbf{H}^T=\mathbf{s}} d(\mathbf{e}, 0),$$

and concludes $\hat{m} = \phi_{open}^{-1}(\mathbf{y} \oplus \mathbf{e}_{rec})$, where $d(\cdot,\cdot)$ denotes Hamming distance.

4.3 Linear Feedback Encoding

Inspired by the successes of linear schemes, we investigate the use of *linear block feedback codes*. The encoding function has the form

$$c_i = \phi_{closed,i}\left(\mathbf{m}, \{e_j\}_{j=0}^{i-1}\right) = \sum_{j=0}^{K-1} m_j g_{j,i} \oplus \sum_{\ell=0}^{i-1} e_\ell f_{\ell,i}, \tag{4.5}$$

where $\mathbf{m} = [m_0, m_1, \cdots, m_{K-1}]$ is the message string and $\{g_{j,i}\}_{j,i}$, $\{f_{\ell,i}\}_{\ell,i} \in \{0,1\}$. In particular, expressing (4.5) in vector form

$$\mathbf{c} = \underbrace{\mathbf{m}\mathbf{G}}_{\text{open-loop component}} \oplus \underbrace{\mathbf{e}\mathbf{F}}_{\text{noise-shaping}}, \tag{4.6}$$

$\mathbf{G}$ is the $K \times N$ open-loop generator matrix and $\mathbf{F}$ is a $N \times N$ binary matrix that represents feedback encoding in the form of *linear noise-shaping*. Since c_i cannot possibly depend on future errors $\{e_j\}_{j=i}^{N-1}$, we must have $f_{i,j} = 0\ \forall i \leq j$. In other words, causality enforces $\mathbf{F}$ to be strictly upper-triangular. Using (4.6), the receiver observes

$$\mathbf{y} = \mathbf{c} \oplus \mathbf{e} = \mathbf{m}\mathbf{G} \oplus \mathbf{e}\,(\mathbf{I} \oplus \mathbf{F})\,. \tag{4.7}$$

Eq.(4.7) indicates transmission of an open-loop codeword over a channel with a special kind of correlated Bernoulli noise. Since $\mathbf{F}$ is strictly upper-triangular, $(\mathbf{I} \oplus \mathbf{F})$ has linearly

independent columns and is full rank over $GF(2)$. The maximum-likelihood (ML) decoder is seen to be

$$\begin{aligned}\mathbf{m}^* &= \arg\max_{\mathbf{m}} Pr(\mathbf{y}|\mathbf{m}) \\ &= \arg\max_{\mathbf{m}} Pr(\mathbf{e}(\mathbf{I}\oplus\mathbf{F}) = \mathbf{y} - \mathbf{m}\mathbf{G}) \\ &= \arg\max_{\mathbf{m}} Pr(\mathbf{e} = \mathbf{y}(\mathbf{I}\oplus\mathbf{F})^{-1} - \mathbf{m}\mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1}) \\ &= \arg\min_{\mathbf{m}} d(\mathbf{y}(\mathbf{I}\oplus\mathbf{F})^{-1}, \mathbf{m}\mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1}).\end{aligned}$$

where the last equality follows since the elements of $\mathbf{e}$ are i.i.d. $Ber(p)$. Thus, the optimal detector constitutes "noise-whitening" followed by a minimum-distance decoder.

What is the effect of noise-shaping on the error detection and correction capabilities of the system? Let $\mathbf{H}$ denote a parity-check corresponding to $\mathbf{G}$, and suppose that the receiver calculates the syndrome [86] to be

$$\mathbf{s} = \mathbf{y}\mathbf{H}^T \overset{(a)}{=} \mathbf{e}\widetilde{\mathbf{H}}^T \tag{4.8}$$

where (a) follows from setting $\widetilde{\mathbf{H}} = \mathbf{H}\,(\mathbf{I}\oplus\mathbf{F})^T$ and noting that $\mathbf{G}\mathbf{H}^T = \mathbf{0}$. From (4.8), it is clear that the use of feedback allows us to detect and correct error patterns as robustly as an open-loop code with the parity check matrix $\widetilde{\mathbf{H}}$.

This is also evident from the noise-whitening interpretation. The receiver whitens the noise sequence by multiplying the received symbols $\mathbf{y}$ by $(\mathbf{I}\oplus\mathbf{F})^{-1}$ and we have

$$\tilde{\mathbf{y}} = \mathbf{y}(\mathbf{I}\oplus\mathbf{F})^{-1} = \mathbf{m}\mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1}\oplus\mathbf{e}. \tag{4.9}$$

The syndrome can be calculated using $\widetilde{\mathbf{H}}$ to be $\tilde{\mathbf{y}}\widetilde{\mathbf{H}}^T = \mathbf{e}\widetilde{\mathbf{H}}^T$ as in (4.8). Since $\mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1}\widetilde{\mathbf{H}}^T = \mathbf{G}\mathbf{H}^T = \mathbf{0}$, $\widetilde{\mathbf{G}} = \mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1}$ and $\widetilde{\mathbf{H}} = \mathbf{H}\,(\mathbf{I}\oplus\mathbf{F})^T$ can be thought of as the effective open-loop generator and open-loop parity-check matrices respectively that we hope is an improvement when $\mathbf{G}$ or $\mathbf{H}$ is relatively weak.

Our development so far suggests that possibly, a weak open-loop code can effectively be enhanced by a judicious choice of feedback encoding $\mathbf{F}$. But, how far can we go? What is the

class of linear codes (through $\widetilde{\mathbf{G}}$ or $\widetilde{\mathbf{H}}$) that can be emulated in this way? This is resolved in section 4.5. But first, we consider random feedback encoding in Section 4.4.

4.4 Random Noise-Shaping is Capacity-Achieving

In reference to Q1, consider an uncoded system i.e. with $\mathbf{G} = [\mathbf{I}_K \ \mathbf{0}_{K\times(N-K)}]$. Our task is to design a noise-shaping rule $\mathbf{F}$ in (4.6) that hopefully yields good performance. We show that a randomly chosen feedback-encoding matrix is capacity achieving.

Theorem 4.4.1. *Let* $\mathbf{G} = \begin{bmatrix}\mathbf{I}_K \ \mathbf{0}_{K\times(N-K)}\end{bmatrix}$ *be an uncoded system and consider the ensemble of random linear block feedback codes generated from by a noise-shaping matrix of the form*

$$\mathbf{F}_{N\times N} = \begin{bmatrix} 0 & f_{0,1} & \cdots & f_{0,N-2} & f_{0,N-1} \\ 0 & 0 & f_{1,2} & \cdots & f_{1,N-1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & f_{N-2,N-1} \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \tag{4.10}$$

where each mutable entry $f_{i,j}$, $i = 0, 1, \cdots, N-1$, $j = 1, 2, \cdots, N-1$, $i < j$ *in* $\mathbf{F}$ *is i.i.d. Ber* $\left(\frac{1}{2}\right)$, *and* $K = \lfloor NR \rfloor$ *for some fixed rate* R. *At all rates below capacity for a BSC*(p), *i.e., for* $R < C(p) = 1 - h(p)$, *the average probability of error for this ensemble decays exponentially with block-length* N.

As shown earlier, the resulting block feedback code mimics in performance an open-loop code with the generator

$$\widetilde{\mathbf{G}} = \mathbf{G}(\mathbf{I} \oplus \mathbf{F})^{-1}. \tag{4.11}$$

Hence, partition

$$(\mathbf{I} \oplus \mathbf{F}) = \left[\begin{array}{c:c} \mathbf{J}^{(1)}_{K\times K} & \mathbf{J}^{(2)}_{K\times(N-K)} \\ \hdashline \mathbf{0}_{(N-K)\times K} & \mathbf{J}^{(3)}_{(N-K)\times(N-K)} \end{array}\right]. \tag{4.12}$$

Here, $\mathbf{J}^{(1)}$ and $\mathbf{J}^{(3)}$ are square invertible upper triangular matrices of dimensions K and $(N-K)$, with off-diagonal entries distributed as i.i.d. $Ber\left(\frac{1}{2}\right)$. $\mathbf{J}^{(2)}$ is a $K \times (N-K)$ binary matrix with each entry i.i.d. $Ber\left(\frac{1}{2}\right)$. It is straightforward to compute

$$(\mathbf{I} \oplus \mathbf{F})^{-1} = \begin{bmatrix} \mathbf{J}^{(1)^{-1}} & \mathbf{J}^{(1)^{-1}}\mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}} \\ \mathbf{0} & \mathbf{J}^{(3)^{-1}} \end{bmatrix},$$

which gives $\widetilde{\mathbf{G}} = \mathbf{J}^{(1)^{-1}}[\mathbf{I} \;\; \mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}]$. Since pre-multiplication of a generator matrix by an invertible transformation leaves the row-space and hence the set of codewords unaltered, $\mathbf{J}^{(1)^{-1}}$ can be ignored. The following lemma gives the distribution of $\mathbf{J}^{(3)^{-1}}$.

Lemma 18. *[87] Let $\mathbf{R}$ be a square invertible upper-triangular matrix over $GF(2)$ with the off-diagonal entries i.i.d. $Ber(\frac{1}{2})$. Then, $\mathbf{R}^{-1}$ is square invertible upper-triangular with its off-diagonal entries distributed as i.i.d. $Ber(\frac{1}{2})$.*

Proof. We have

$$\mathbf{R} = \begin{bmatrix} 1 & r_1 & \cdots & r_{m-2} & r_{m-1} \\ 0 & 1 & r_m & \cdots & r_{2m-3} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & r_{\frac{m(m-1)}{2}} \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}$$

where r_i are distributed i.i.d. $Ber(\frac{1}{2})$. Its inverse over $GF(2)$ has the form

$$\mathbf{R}^{-1} = \begin{bmatrix} 1 & s_1 & \cdots & s_{m-2} & s_{m-1} \\ 0 & 1 & s_m & \cdots & s_{2m-3} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & s_{\frac{m(m-1)}{2}} \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}.$$

We are interested in characterizing the distribution of $\{s_j\}$. Since a necessary and sufficient condition for an upper triangular matrix over $GF(2)$ to be invertible is for the diagonal to have all ones, the (non-linear) map that takes $\{r_j\}$ to $\{s_j\}$ is one-to-one and onto. Thus,

each realization of $\mathbf{R}$ being equally likely implies that each realization of $\mathbf{R}^{-1}$ or that of the q-tuple $(s_1, s_2, \cdots, s_q)$ is equally likely with probability $\frac{1}{2^q}$, where $q = \frac{m(m-1)}{2}$. The marginal distribution of s_k for some $1 \leq k \leq q$ can now be easily calculated by summing out the rest $q-1$ random variables as

$$Pr(s_k = i) = \sum_{s_1, s_2, \cdots, s_{k-1}, s_{k+1}, \cdots, s_q} \frac{1}{2^q} = \frac{2^{q-1}}{2^q} = \frac{1}{2} \tag{4.13}$$

for $i = 0, 1$.

Similar to (4.13), calculation of the joint distribution of an arbitrary sub-collection of the random variables say $(s_{i_1}, s_{i_2}, \cdots, s_{i_b})$ involves summing 2^{q-b} terms each equal to $\frac{1}{2^q}$, yielding a uniform distribution which is equal to the product of the marginals involved. Hence, $\{s_i\}$ are indeed distributed i.i.d. $Ber(\frac{1}{2})$.

□

Lemma 19. *The random matrix* $\mathbf{Q} = \mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}$ *is a* $K \times N$ *matrix over* $GF(2)$ *with each entry distributed i.i.d.* $Ber(\frac{1}{2})$.

Proof. From Lemma 18, $\mathbf{J}^{(3)^{-1}}$ is a random-matrix

$$\mathbf{J}^{(3)^{-1}} = \begin{bmatrix} 1 & s_1 & \cdots & s_{m-2} & s_{m-1} \\ 0 & 1 & s_m & \cdots & s_{2m-3} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & s_{\frac{m(m-1)}{2}} \\ 0 & 0 & \ldots & 0 & 1 \end{bmatrix}$$

with $\{s_i\}$ distributed i.i.d. $Ber(\frac{1}{2})$ and

$$\mathbf{J}^{(2)} = \begin{bmatrix} f_{0,K} & f_{0,K+1} & \cdots & f_{0,N-1} \\ f_{1,K} & f_{1,K+1} & \cdots & f_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ f_{K-1,K} & f_{K-1,K+1} & \cdots & f_{K-1,N-1} \end{bmatrix}$$

where $\{f_{i,j}\}, 0 \leq i \leq K-1, K \leq j \leq N-1$ are distributed i.i.d. $Ber(\frac{1}{2})$. Since each $f_{i,j} \sim Ber(\frac{1}{2})$, the marginal distribution of each element of $\mathbf{Q}$ is $Ber(\frac{1}{2})$. To see independence, consider calculating the joint distribution of an arbitrary sub-collection of elements from $\mathbf{Q}$. For a fixed realization $\mathbf{J}^{(3)^{-1}}$, it can be verified that the conditional joint distribution is uniform and since every realization of $\mathbf{J}^{(3)^{-1}}$ is equally likely, the overall joint distribution is indeed uniform and the claim holds. □

From the above discussion, it suffices to prove the theorem for the ensemble of random systematic linear codes generated by $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{Q}]$ where entries of $\mathbf{Q}$ are distributed i.i.d. $Ber(\frac{1}{2})$. This is essentially a known result [88] hence proving theorem 4.4.1.

4.5 Design of Encoding Function

4.5.1 Strengthen a Weak Code

We briefly recall some well-known facts about binary linear codes. Consider a code with generator matrix $\mathbf{G}$. First, the generator matrix representation is not unique - $\mathbf{G}$ and $\mathbf{KG}$ both generate the same code for any invertible matrix $\mathbf{K}$. This is because pre-multiplication by $\mathbf{K}$ can be decomposed into a sequence of elementary row operations none of which alter the row space, and hence the set of codewords. Second, applying a permutation on the coordinate positions of the codewords yields a different linear code but with the same error-correction capabilities. Such codes are referred to as being *equivalent* in literature [86].

Suppose now that a system is equipped with a relatively weak (N, K) code with the generator $\mathbf{G}$. Without loss of generality, we assume that the $K \times K$ submatrix of $\mathbf{G}$ comprising of its first K columns is full rank. From the above discussion then, we can let $\mathbf{G}$ be of the form $\mathbf{G} = [\mathbf{I}_K\ \mathbf{P}_{K\times(N-K)}]$. To improve performance, we employ linear feedback encoding as in (4.6) by means of a noise-shaping matrix $\mathbf{F}$. The main result is now stated as follows:

Theorem 4.5.1. *By means of noise-shaping,* $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$ *can be strengthened to have performance of that of any desired linear code.*

Proof. As developed in Section 4.3, the resulting linear block feedback code has performance equivalent to the open-loop linear code with generator $\widetilde{\mathbf{G}} = \mathbf{G}(\mathbf{I} \oplus \mathbf{F})^{-1}$. We have the general structure

$$(\mathbf{I} \oplus \mathbf{F}) = \left[\begin{array}{c:c} \mathbf{J}^{(1)}_{K\times K} & \mathbf{J}^{(2)}_{K\times(N-K)} \\ \hdashline \mathbf{0}_{(N-K)\times K} & \mathbf{J}^{(3)}_{(N-K)\times(N-K)} \end{array}\right] \tag{4.14}$$

where $\mathbf{J}^{(1)}$, $\mathbf{J}^{(3)}$ are both invertible and upper-triangular. We then have,

$$\widetilde{\mathbf{G}} = \mathbf{G}(\mathbf{I} \oplus \mathbf{F})^{-1} = \left[\mathbf{J}^{(1)^{-1}} \quad \left(\mathbf{P} \oplus \mathbf{J}^{(1)^{-1}}\mathbf{J}^{(2)}\right)\mathbf{J}^{(3)^{-1}}\right]$$

Pre-multiplying by the invertible transform $\mathbf{J}^{(1)}$, we have

$$\widetilde{\mathbf{G}} = \left[\mathbf{I} \quad \left(\mathbf{J}^{(1)}\mathbf{P} \oplus \mathbf{J}^{(2)}\right)\mathbf{J}^{(3)^{-1}}\right]. \tag{4.15}$$

Choosing $\mathbf{J}^{(1)} = \mathbf{I}_K$, $\mathbf{J}^{(3)} = \mathbf{I}_{(N-K)}$ gives $\widetilde{\mathbf{G}} = [\mathbf{I}\ (\mathbf{P} \oplus \mathbf{J}^{(2)})]$. We can strengthen $\mathbf{G}$ to any desired linear code by choosing a suitable $\mathbf{J}^{(2)}$. Suppose that the desired linear code has generator $\mathbf{G}_0 = [\mathbf{I}\ \mathbf{Q}]$. For example, one could choose $\mathbf{G}_0$ to be the systematic generator of a capacity-achieving code. Then, set $\mathbf{J}^{(2)} = \mathbf{P} \oplus \mathbf{Q}$, i.e.

$$\mathbf{F} = \left[\begin{array}{c:c} \mathbf{0}_{K\times K} & \mathbf{Q} \oplus \mathbf{P} \\ \hdashline \mathbf{0}_{(N-K)\times K} & \mathbf{0}_{(N-K)\times(N-K)} \end{array}\right]. \tag{4.16}$$

For this specific choice, $\mathbf{F}^2 = \mathbf{0}_{N\times N}$ and $(\mathbf{I}\oplus\mathbf{F})^{-1} = (\mathbf{I}\oplus\mathbf{F})$. Hence we have $\widetilde{\mathbf{G}} = \mathbf{G}(\mathbf{I}\oplus\mathbf{F})^{-1} = \mathbf{G}(\mathbf{I} \oplus \mathbf{F}) = [\mathbf{I}\ \mathbf{Q}] = \mathbf{G}_0$ and we are done. □

Remark. Note from the choice of $\mathbf{F}$ in (4.16), the codewords of the linear block feedback code in (4.6) have a rather special structure

$$\begin{aligned}\mathbf{c} &= \mathbf{m}[\mathbf{I}\ \mathbf{P}] \oplus \mathbf{e}\left[\begin{array}{c:c} \mathbf{0} & \mathbf{P} \oplus \mathbf{Q} \\ \hdashline \mathbf{0} & \mathbf{0} \end{array}\right] \\ &= \mathbf{m}[\mathbf{I}\ \mathbf{P}] \oplus [\mathbf{0}\ \mathbf{e}_s(\mathbf{P} \oplus \mathbf{Q})]\end{aligned}$$

where $\mathbf{e}_s = [e_0, e_1, \cdots, e_{K-1}]$ is noise vector that corrupts the systematic message bits. The resulting code involves linear encoding of only the initial error bits. We lose nothing in performance by ignoring error bits e_K through e_{N-1} and hence, *feedback can be shut off for non-systematic bits.*

4.5.2 Noise-Shaping vs Parity bits

Consider an uncoded system, i.e. $\mathbf{G} = [\mathbf{I}\ \mathbf{0}]$. Suppose we wish to enhance it to a code with generator $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{Q}]$. As discussed in theorem 4.5.1, we set

$$\mathbf{F} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{Q} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right]. \tag{4.17}$$

From Eq. 4.6 then, the codewords are given by

$$\begin{aligned} \mathbf{c} &= \mathbf{m}[\mathbf{I}_K\ \mathbf{0}] \oplus \mathbf{e} \left[\begin{array}{c|c} \mathbf{0} & \mathbf{Q} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] \\ &= [\ \underbrace{m_0\ m_1\ \cdots\ m_{K-1}}_{\text{Systematic Tx}}\ |\ \underbrace{[e_0\ e_1\ \cdots\ e_{K-1}]\mathbf{Q}}_{\text{Noise-Shaping}}\ ] \end{aligned} \tag{4.18}$$

where e_i is the noise bit that corrupts message bit m_i, for $i = 0, 1, \cdots, K-1$. *Systematic message bit transmission followed by noise-shaping of only the initial error bits is thus sufficient to mimic any arbitrary open-loop code.* Choosing $\widetilde{\mathbf{G}}$ to be capacity-achieving settles Q1 with a positive answer.

It is interesting to compare the codewords $\mathbf{c}$ in (4.18) to the codewords $\mathbf{c}$ in the case where the system was directly using the open-loop code $\widetilde{\mathbf{G}}$ -

$$\begin{aligned} \mathbf{c} &= \mathbf{m}\widetilde{\mathbf{G}} \\ &= [\ \underbrace{m_0\ m_1\ \cdots\ m_{K-1}}_{\text{Systematic Tx}}\ |\ \underbrace{[m_0\ m_1\ \cdots\ m_{K-1}]\mathbf{Q}}_{\text{Tx of Parity Bits}}\ ]. \end{aligned}$$

Analogous to transmission of parity bits in an open-loop code, linear feedback encoding can be thought of as noise-shaping of errors that occur in the systematic transmission, according to the *same rule as encapsulated in* $\mathbf{Q}$.

4.6 Encoding under Feedback Limitations

4.6.1 Compressed Feedback

Feedback resources in a real system are generally limited making full causal feedback unrealistic. Motivated by this, we consider the scenario where the feedback symbols are linearly compressed before transmission. The receiver sends causal linear combinations of bits received thus far, in a few channel uses. An example is illustrated in Fig. 4.1 where symbols y_0, $(y_0 \oplus y_1)$ and $(y_1 \oplus y_3)$ are fed back after channel uses 1, 3 and 4 respectively.

First, consider upgrading an uncoded system. Recall from Section 4.5.2, an uncoded transmission of K message bits followed by $(N-K)$ bits $\mathbf{e}_s\mathbf{Q}$ where $\mathbf{e}_s = [e_0, e_1, \cdots, e_{K-1}]$ is the vector of initial errors and $\mathbf{Q}$ is some $K \times (N-K)$ binary matrix, emulates the code with generator $\widetilde{\mathbf{G}} = [\mathbf{I}_K \ \mathbf{Q}]$. We have already seen that K bits of uncompressed feedback (i.e. $y_0, y_1, y_2, \cdots, y_{K-1}$) can turn the uncoded system to any code. When $K > (N-K)$, i.e. the target code rate is larger than 0.5, we can do better. It is indeed sufficient for the receiver to send the $(N-K)$ bits $\mathbf{y}_s\mathbf{Q}$ instead of the K bits $\mathbf{y}_s$, where $\mathbf{y}_s = [y_0, y_1, \cdots, y_{K-1}]$ is the vector of bits received during uncoded transmission. We thus conclude, $b = \min(K, N-K)$ bits of feedback (possibly after compression) per code-block suffice to enhance an uncoded system by shaping feedback signals to any desired code.

Example 4.6.1. *To transform an uncoded system to a* $(4, 3)$ *code with*

$$\widetilde{\mathbf{G}} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

it is sufficient for the receiver to send $b = 1$ *bit* $(y_0 \oplus y_1 \oplus y_2)$ *after the third channel use.*

The next situation to consider is when $b < \min(K, N-K)$. Which are the codes that we can emulate?

Theorem 4.6.1. *With b-bits of (possibly compressed) feedback, any choice of feedback encoding effectively turns an uncoded system to a code with* $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{P}]$ *where* $rank(\mathbf{P}) \leq b$. *Conversely, for any given* $\mathbf{P}$ *with* $rank(\mathbf{P}) \leq b$, *compression of* $\mathbf{y}_s$ *to b bits of feedback is sufficient to emulate* $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{P}]$.

Proof. First, the converse is easy to see. If $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_b$ are column vectors that form a basis for the column span of $\mathbf{P}$, it is sufficient for the receiver to send bits $\mathbf{y}_s\mathbf{p}_j$, $j = 1, 2, \cdots, b$.

For the forward direction, we introduce some notation. Let $\mathbf{W}$ be a binary compression matrix of dimension $N \times N$ with only b non-zero columns. $\mathbf{W}$ describes how the received bits $\mathbf{y}$ are compressed. The indices of the non-zero columns corresponds to the channel uses where feedback is sent. Due to causality, note that $\mathbf{W}$ is upper triangular. An example of $\mathbf{W}$ is illustrated in Fig. 4.1. Hence, the encoding rule at the transmitter becomes

$$\mathbf{c} = \mathbf{mG} \oplus \mathbf{eWF}$$

where $\mathbf{m}$ is the $1 \times K$ message vector, $\mathbf{G}$ is the open-loop $K \times N$ generator matrix and the $N \times N$ matrix $\mathbf{F}$ describes how the transmitter shapes the bits received in the feedback for encoding. The effective generator matrix is then $\widetilde{\mathbf{G}} = \mathbf{G}\left(\mathbf{I} \oplus \mathbf{WF}\right)^{-1}$. Since $\mathbf{G}$ is uncoded, denoting

$$(\mathbf{I} \oplus \mathbf{WF}) = \left[\begin{array}{c:c} \mathbf{J}^{(1)}_{K\times K} & \mathbf{J}^{(2)}_{K\times(N-K)} \\ \hdashline \mathbf{0}_{(N-K)\times K} & \mathbf{J}^{(3)}_{(N-K)\times(N-K)} \end{array}\right], \tag{4.19}$$

the effective generator matrix is $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}]$. We have $rank(\mathbf{W}) \leq b$ since it has at most b columns that are non-zero. This implies that $\mathbf{J}^{(2)}$ and hence the matrix $\mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}$ has rank at most b completing the proof. Note that the result holds for any choice of $\mathbf{W}$ or $\mathbf{F}$ i.e. for any sort of feedback encoding. □

The next theorem gives the best minimum distance distance achievable and some necessary conditions that need to be met in order to achieve it.

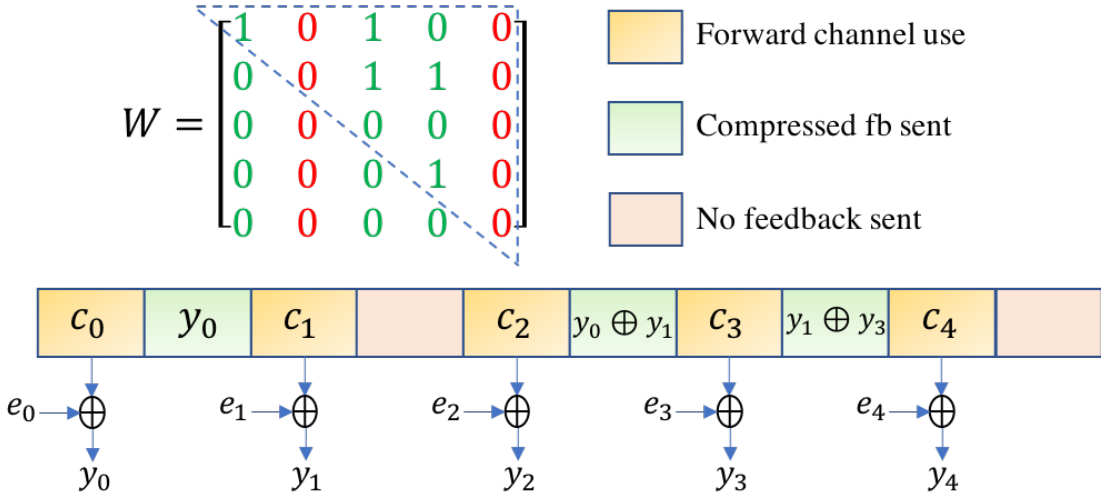


Figure 4.1. An example of feedback compression matrix $\mathbf{W}$ as defined in the proof of Theorem 4.6.1 with $N = 5$ and $b = 3$. The result holds irrespective of the choice of $\mathbf{W}$ or $\mathbf{F}$.

Theorem 4.6.2. *For a (N, K) linear code $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$ with $rank(\mathbf{P}) \leq b$ where $b < \min(K, N-K)$, we have the bound*

$$d_{min} \leq b + 1. \tag{4.20}$$

Furthermore, when $d_{min} = b + 1$ for $b > 1$, then

(i) $rank(\mathbf{P}) = b$

(ii) $K = b + 1 = d_{min}$

(iii) $N \geq 2b + 2$ *i.e. rate* $\frac{K}{N} \leq \frac{1}{2}$.

Proof. The parity check matrix is $\mathbf{H} = [\mathbf{P}^T\ \mathbf{I}]$. Recall the following property [86] of linear codes

Property 1. $d_{min} = d$ iff every $d-1$ columns from $\mathbf{H}$ are linearly independent and some d columns are linearly dependent.

Since $rank(\mathbf{P}) \leq b$, any $b+1$ columns from $\mathbf{P}^T$ are linearly dependent and we can conclude

$$d_{min} \leq b + 1.$$

For the second part, since $d_{min} = b+1$, every b columns from $\mathbf{P}^T$ (or b rows from $\mathbf{P}$) are linearly independent giving $rank(\mathbf{P}) \geq b$. Since it is also given that $rank(\mathbf{P}) \leq b$, we have $rank(\mathbf{P}) = b = rank(\mathbf{P}^T)$. Let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_b$ be a set of b linearly independent columns from $\mathbf{P}^T$. Every other column in $\mathbf{P}^T$ is some linear combination of these. Note $b < \min(K, N-K)$ implies $K \geq b+1$. We now show that $K > b+1$ is impossible. This can be seen in two steps:

1. First, suppose $K = b+1$ and the $(b+1)^{th}$ column is $\mathbf{v}_{b+1} = \sum_{i \in \mathcal{I}} \mathbf{v}_i$ for some index set $\mathcal{I} \subseteq \{1, 2, 3, \cdots, b\}$. Then, we must have $\mathcal{I} = \{1, 2, 3, \cdots, b\}$. This is because if $k \notin \mathcal{I}$, then $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{k-1}, \mathbf{v}_{k+1}, \cdots, \mathbf{v}_b, \mathbf{v}_{b+1}$ are b columns that are linearly dependent, violating property 1.

2. Suppose $K = b+2$, then from above we must have $\mathbf{v}_{b+1} = \mathbf{v}_{b+2} = \sum_i \mathbf{v}_i$. However then, $\mathbf{v}_{b+1}, \mathbf{v}_{b+2}$ are linearly dependent. The same argument extends for $K \geq b+2$.

Thus, when $d_{min} = b+1$ for some $b > 1$, we have $rank(\mathbf{P}) = b = K-1$. We also have $(N-K) > b$ implying $N \geq 2b+2$ or $\frac{K}{N} \leq \frac{1}{2}$. □

Finally, suppose that the original system is not uncoded but equipped with some weak code $\mathbf{G} = [\mathbf{I}\ \mathbf{P}]$. Recall from (4.19) and (4.15) that with feedback encoding using b bits of feedback, we can emulate a generator

$$\widetilde{\mathbf{G}} = \left[\mathbf{I} \quad \left(\mathbf{J}^{(1)}\mathbf{P} \oplus \mathbf{J}^{(2)}\right)\mathbf{J}^{(3)^{-1}}\right]. \tag{4.21}$$

We can choose $\mathbf{W}$ and $\mathbf{F}$ such that $\mathbf{J}^{(1)} = \mathbf{I}_K$, $\mathbf{J}^{(3)} = \mathbf{I}_{N-K}$ and $\mathbf{J}^{(2)}$ is any desired $K \times (N-K)$ binary matrix with $rank\left(\mathbf{J}^{(2)}\right) = b$. Then, (4.21) becomes

$$\widetilde{\mathbf{G}} = \left[\mathbf{I} \quad \mathbf{P} \oplus \mathbf{J}^{(2)}\right].$$

In sum, with b of compressed feedback, the parity component of a given code can be *perturbed* by a matrix of rank b.

We end this section with an example of how a simple repeat-transmission scheme is enhanced with one bit of compressed feedback.

Example 4.6.2. *Consider the* $(12, 4)$ *repetition code that has generator*

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and $d_{min} = 3$. *We show that* $b = 1$ *bit of compressed feedback can enhance the minimum distance to* $d_{min} = 5$, *hence granting guaranteed resilience against* 2 *errors as opposed to* 1. *Having received* $\mathbf{y}_s = [y_0, y_1, y_2, y_3]$, *the receiver sends the compressed version* $d = (y_0 \oplus y_1 \oplus y_2 \oplus y_3)$. *The transmitter ascertains* $d = (e_0 \oplus e_1 \oplus e_2 \oplus e_3)$ *and the closed-loop codeword is*

$$\mathbf{c} = (m_0, m_1, m_2, m_3, m_0, m_1, m_2, m_3, m_0 \oplus d, m_1 \oplus d, m_2 \oplus d, m_3 \oplus d).$$

The new effective generator $\widetilde{\mathbf{G}}$ *with* $d_{min} = 5$ *is*

$$\widetilde{\mathbf{G}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

obtained by a rank $b = 1$ *perturbation of* $\mathbf{G}$.

4.6.2 Delayed Feedback

Now suppose that there is a delay of Δ channel uses between the time that feedback is sent and when it can be used for encoding at the transmitter. The encoding function changes to

$$c_i = \phi_{closed,i}\left(\mathbf{m}, \{e_j\}_{j=0}^{i-1-\Delta}\right) = \sum_{j=0}^{K-1} m_j g_{j,i} \oplus \sum_{\ell=0}^{i-1-\Delta} e_\ell f_{\ell,i}.$$

In vector form, this is expressed as

$$\mathbf{c} = \underbrace{\mathbf{mG}}_{\text{open-loop component}} \oplus \underbrace{\mathbf{eF}}_{\text{noise-shaping}}$$

where the noise-shaping matrix $\mathbf{F}$ is strictly upper triangular additionally with Δ all-zero diagonals above the main diagonal, i.e., $f_{i,j} = 0 \ \forall i \leq j + k$ for $k = 0, 1, \cdots, \Delta$.

Let $\mathcal{S}^{(\Delta)}$ be the set of all (N, K) open-loop systematic linear codes induced by a generator matrix of the form

$$\mathbf{G} = [\mathbf{I}_K \ \mathbf{P}^{(\Delta)}] \tag{4.22}$$

where $\mathbf{P}^{(\Delta)}$ is a $K \times (N - K)$ binary matrix with the property that it has Δ consecutive all-zero sub-diagonals beginning at the lower left end. For instance,

$$\mathbf{P}^{(3)} = \begin{bmatrix} \times & \times & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \times & \cdots & \times \\ 0 & 0 & \times & \times & \cdots & \times \\ 0 & 0 & 0 & \times & \cdots & \times \end{bmatrix}.$$

In other words, the parity component $\mathbf{P}^{(\Delta)}$ consists of an 'all-zero triangle' of size Δ embedded from the lower left. Note that the largest such triangle that can be fit into the $K \times (N - K)$ matrix $\mathbf{P}^{(\Delta)}$ has size $\Delta^\star = \min(K, N - K)$ where K and N are the code dimension and code length respectively. The class of linear codes that can be emulated from an uncoded system, when there is feedback delay is characterized by the following theorem.

Theorem 4.6.3. *Suppose that we start with an uncoded system with* $\mathbf{G} = \left[\mathbf{I}_K \ \mathbf{0}_{K\times(N-K)}\right]$. *For any choice of linear block feedback encoding with complete causal feedback and* Δ *units of feedback delay, the new code obtained is effectively equivalent to some code in* $\mathcal{S}^{(\Delta)}$. *Conversely, every code in the set* $\mathcal{S}^{(\Delta)}$ *can be emulated with suitable encoding.*

Proof. Let $\mathbf{F}$ be the noise-shaping matrix chosen for encoding. Denoting

$$(\mathbf{I} \oplus \mathbf{F}) = \left[\begin{array}{c:c} \mathbf{J}^{(1)}_{K\times K} & \mathbf{J}^{(2)}_{K\times(N-K)} \\ \hdashline \mathbf{0}_{(N-K)\times K} & \mathbf{J}^{(3)}_{(N-K)\times(N-K)} \end{array}\right], \tag{4.23}$$

the effective generator matrix is $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}]$. Since $\mathbf{F}$ has Δ all-zero diagonals above its main diagonal and $\mathbf{J}^{(3)^{-1}}$ is upper-triangular, the matrix $\mathbf{Q} = \mathbf{J}^{(2)}\mathbf{J}^{(3)^{-1}}$ has Δ all-zero sub-diagonals beginning at the lower left corner. In other words, $\widetilde{\mathbf{G}}$ essentially is of the form $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{P}^{(\Delta)}]$ and the first part of the theorem is proved. To see the converse, consider emulating an arbitrary code from $\mathcal{S}^{(\Delta)}$ with generator $\widetilde{\mathbf{G}} = [\mathbf{I}\ \mathbf{P}_1^{(\Delta)}]$, and note that a valid choice of feedback encoding is to simply set

$$\mathbf{F} = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{P}_1^{(\Delta)} \\ \hdashline \mathbf{0} & \mathbf{0} \end{array}\right].$$

□

From Theorem 4.6.3, an uncoded system can be transformed to a code of the form (4.22) (and no better) when there is a delay of Δ units in the feedback link. A central question is thus whether there exists a good code or a good class of codes that has a generator of the form (4.22) with a possibly large Δ. *The larger the value of Δ, the larger is the feedback delay that can be tolerated.*

We demonstrate in the next section that Reed-Muller (RM) codes prove to be a good candidate in that they have excellent performance and admit very large values of Δ. To the best of authors' knowledge, this type of result for RM codes is new and has never been shown before. *Thus, a RM code can be emulated from an uncoded system by means of linear noise-shaping against a feedback delay that is nearly as large as $\Delta^\star = \min(K, N-K)$.*

4.7 A Novel Systematic Representation for Reed-Muller Codes

Reed-Muller (RM) codes are some of the oldest and well-studied code families that remain relevant even today. They have found application in many research areas such as cryptogra-

phy, distributed computing, theory of randomness (e.g., see [89] and the references therein). The invention of polar codes [90] has rekindled intensive research into RM codes due to their close relationship. Polar codes are known to be provably capacity achieving for binary-input symmetric discrete memoryless channels (DMCs). Recent developments have shown that RM codes achieve capacity on the Binary Erasure Channel [91] and are also long believed to achieve capacity over the Binary Symmetric Channel, which is strongly supported by simulations [92], [93]. Table 1 in [94] provides the best known capacity results for RM codes to date. Systematic RM codes have been considered from an encoding perspective in [95] and decoding perspective in [96]. A comprehensive survey on RM codes, their applications and connections to other research problems can be found in [97].

We wish prove that RM codes admit a systematic generator matrix of the form given in (4.22),

$$\mathbf{G} = [\mathbf{I}_K \ \mathbf{P}^{(\Delta)}]. \tag{4.24}$$

Denote the vector space of all binary m-tuples as V_m. A boolean function in m variables $f(v_1, v_2, \cdots, v_m)$ is a mapping from V_m to $\{0, 1\}$. By fixing an ordering on $\{(v_1, v_2, \cdots, v_m) \in V_m\}$, we can uniquely associate to function f a binary vector $\mathbf{f}$ of length 2^m whose components are the result of evaluating f at all possible ordered input combinations.

Definition. The r-th order Reed-Muller (RM) code of length $N = 2^m$ denoted $\mathcal{R}(r, m)$ is a linear code that consists of vectors associated to all boolean polynomials f of degree less than equal to r in m variables. The dimension of $\mathcal{R}(r, m)$ is $K(r, m) = \sum_{j=0}^{r} \binom{m}{j}$ and minimum distance is $d_{min}(r, m) = 2^{m-r}$ [86].

4.7.1 A formula for $\Delta(r, m)$

In this section, we prove that the code $\mathcal{R}(r, m)$ admits a generator of the form (4.24) with $\Delta = \Delta(r, m)$ given by

$$\Delta(r, m) = \begin{cases} \sum_{j=0}^{r} \binom{m}{j} - \sum_{j=0}^{r} \binom{2j}{j} & 0 \leq r \leq \lfloor \frac{m}{2} \rfloor \\ \Delta(m - r - 1, m) & \lfloor \frac{m}{2} \rfloor < r < m \end{cases}. \tag{4.25}$$

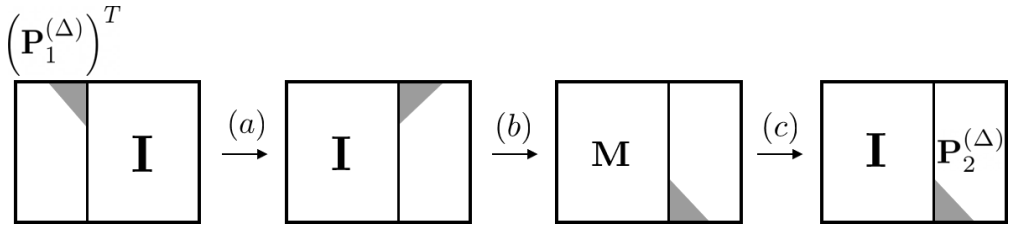


Figure 4.2. Sequence of operations in the proof of Lemma 20.

We being by showing that if a RM code admits a certain value of Δ, then so does its dual.

Lemma 20. *Suppose that for (r, m) with $r \leq \lfloor \frac{m}{2} \rfloor$, the Reed-Muller code $\mathcal{R}(r, m)$ admits a systematic form $[\mathbf{I}\ \mathbf{P}_1^{(\Delta)}]$. Then, its dual $\mathcal{R}(m - r - 1, m)$ admits a form $[\mathbf{I}\ \mathbf{P}_2^{(\Delta)}]$, with the same Δ.*

Proof. Since $\mathcal{R}(m - r - 1, m)$ is the dual code to $\mathcal{R}(r, m)$, a valid generator matrix is $\left[\left(\mathbf{P}_1^{(\Delta)}\right)^T\ \mathbf{I}\right]$. The rest of the steps are pictorially represented in Fig. 4.2 where the triangle of zeros is shown in grey. The steps are

(a) Permute columns as shown.

(b) Apply a row permutation matrix $\mathbf{M}$ on the left to rearrange rows to obtain the form shown.

(c) Apply column permutation matrix $\mathbf{M}^T$ on the right to only the first block of columns to obtain the identity matrix for this block.

□

We are now ready to state the main theorem.

Theorem 4.7.1. *For $0 \leq r < m$, $\mathcal{R}(r, m)$, the r-th order Reed-Muller code of length 2^m admits a systematic generator matrix of the form*

$$\mathbf{G}_{(r,m)} = [\mathbf{I}\ \mathbf{P}^{(\Delta(r,m))}] \tag{4.26}$$

where $\Delta(r, m)$ is given by (4.25).

Remark. Note that $\mathcal{R}(m, m)$ consists of all binary 2^m-tuples and its only systematic representation is simply $\mathbf{I}_{2^m}$, i.e., $\Delta(m, m) = 0$.

Remark. When $\lfloor \frac{m}{2} \rfloor < r < m$, we have $\Delta(r, m) = \Delta(m-r-1, m)$ which agrees with Lemma 20 since codes $\mathcal{R}(r, m)$ and $\mathcal{R}(m-r-1, m)$ are dual of one another.

Proof. We use an induction argument. Let $\mathcal{P}(r, m)$ be the proposition that $\mathcal{R}(r, m)$ admits a generator of the form given in (4.26). For the base case we prove $\mathcal{P}(0, m)$ and $\mathcal{P}(1, m)$. In the induction step, assuming $\mathcal{P}(r+1, m)$ and $\mathcal{P}(r, m)$ to be true, we prove $P(r+1, m+1)$ to conclude the proof.

Base Case: For $r = 0$, the generator matrix for $\mathcal{R}(0, m)$ is simply

$$\mathbf{G}_{(0,m)} = [1\ 1\ 1\ \cdots 1]$$

meaning that $\Delta(0, m) = 0\ \forall m$. This agrees with (4.25), where $\binom{0}{0}$ is understood to be 1. For $r = 1$, we need to show that $\Delta(1, m) = (m+1) - (1+2) = m - 2$. This is most easily seen by induction. For the base case, $\mathcal{R}(1, 3)$ has the generator matrix

$$\mathbf{G}_{(1,3)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The row operation to transform the above to a systematic form is to add rows $2, 3, \cdots, m+1$ to row 1 which gives

$$\mathbf{G}_{(1,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Hence, $\Delta(1, 3) = 1 = 3 - 2$. Now, suppose that the hypothesis is true for $\mathcal{R}(1, m)$. It is known that [86]

$$\mathcal{R}(1, m+1) = \{(\mathbf{u}, \mathbf{1} + \mathbf{u}), \mathbf{u} \in \mathcal{R}(1, m)\}$$

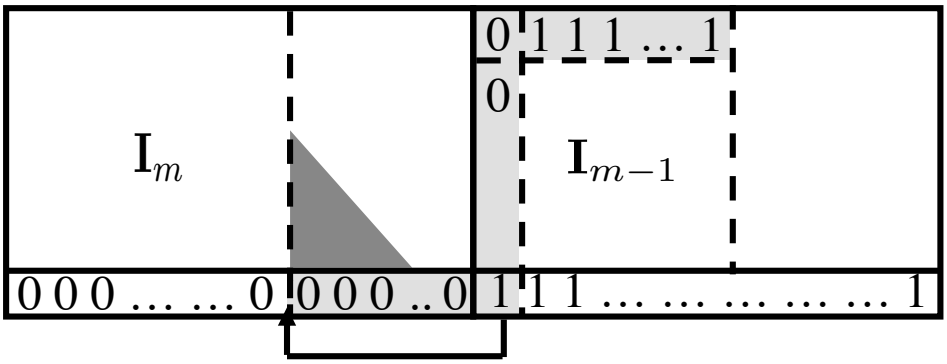


Figure 4.3. Proof of base case proposition $\mathcal{P}(1, m)$ of Theorem 4.7.1.

where $\mathbf{1}$ is the all-one codeword. Hence, the generator for $\mathcal{R}(1, m+1)$ after row operations and column permutations can be put in a form shown in Fig. 4.3. The 'triangle' of zeros for $\mathcal{R}(1, m)$ is of size $m-2$ and shown shaded. Finally, the column vector $[0, 0, \cdots, 1]^T$ is adjoined to $\mathbf{I}_m$ to obtain a systematic form for $\mathcal{R}(1, m+1)$. When $2^m - (m+1) > m-2$ which indeed holds for $\forall\ m \geq 3$, we see that the size of the triangle is guaranteed to increase by 1, i.e., $\Delta(1, m+1) = \Delta(1, m) + 1 = m - 1 = (m+1) - 2$ proving that $\mathcal{P}(1, m)$ is true.

Induction Step: Assume that $\mathcal{P}(r+1, m)$ and $\mathcal{P}(r, m)$ are true. By the well-known Plotkin $(\mathbf{u}, \mathbf{u}+\mathbf{v})$ construction, a valid generator for $\mathcal{R}(r+1, m+1)$ is

$$\mathbf{G}_{(r+1,m+1)} = \begin{bmatrix} \mathbf{G}_{(r+1,m)} & \mathbf{G}_{(r+1,m)} \\ \mathbf{0} & \mathbf{G}_{(r,m)} \end{bmatrix}.$$

It is clear that $\mathbf{G}_{(r+1,m+1)}$ can be put into a form shown in Fig. 4.4.(a) by row operations on the top and bottom block, followed by suitable column permutations. The next steps to obtain a systematic form are illustrated in Fig. 4.4:

I: Suitable row operations are done to zero out the matrix marked with a crosshatch pattern shown in 4.4.(a).

II: The newly obtained block $\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$ is then moved as shown in 4.4.(b) resulting in a systematic form shown in 4.4.(c).

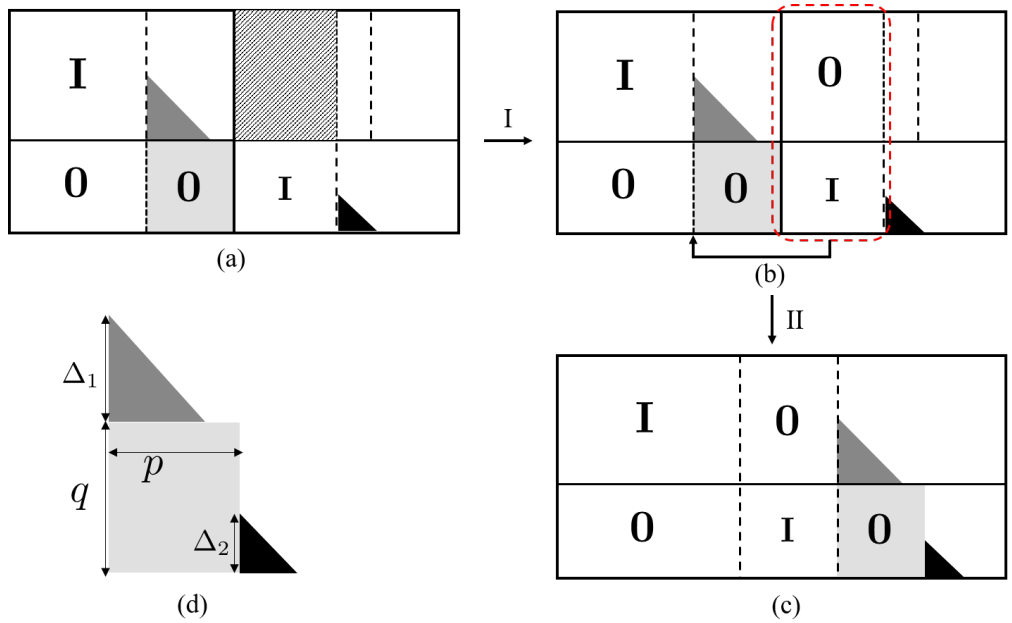


Figure 4.4. Steps in the proof of Theorem 4.7.1.

The systematic form obtained admits a 'triangle' of zeros in its parity matrix component as illustrated in Fig. 4.4.(d). The guaranteed number of consecutive all-zero sub-diagonals, beginning at the lower left end can be seen to be

$$\Delta(r+1, m+1) = \min\{\Delta_1 + q, \Delta_2 + p\}, \tag{4.27}$$

where $\Delta_1 = \Delta(r+1, m)$, $\Delta_2 = \Delta(r, m)$ and

$$p = \sum_{j=r+2}^{m} \binom{m}{j} \ , \ q = \sum_{j=0}^{r} \binom{m}{j}.$$

Our goal is to show that the expression in (4.27) matches with the hypothesis (4.25) for all $0 \leq r < m$. The proof will extensively use the well-known Pascal's identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \tag{4.28}$$

and the symmetry property of the binomial coefficients $\binom{n}{k} = \binom{n}{n-k}$ which are recalled here for ease of exposition.

Case 1: $r+1 \leq \lfloor \frac{m}{2} \rfloor$

The above assumption implies that $r \leq \lfloor \frac{m}{2} \rfloor$ and $r+1 \leq \lfloor \frac{m+1}{2} \rfloor$. From (4.25) then, we have

$$\Delta_1 = \sum_{j=0}^{r+1} \binom{m}{j} - \sum_{j=0}^{r+1} \binom{2j}{j}\ ,\ \Delta_2 = \sum_{j=0}^{r} \binom{m}{j} - \sum_{j=0}^{r} \binom{2j}{j}.$$

Note that

$$(p+\Delta_2) - (q+\Delta_1) = \sum_{j=r+2}^{m} \binom{m}{j} - \sum_{j=0}^{r+1} \binom{m}{j} + \binom{2r+2}{r+1} \overset{(i)}{\geq} 0,$$

where (i) is proved in the following Lemma.

Lemma 21. *For $\ell \leq \lfloor \frac{m}{2} \rfloor$, we have that*

$$\sum_{j=\ell+1}^{m} \binom{m}{j} \geq \sum_{j=0}^{\ell} \binom{m}{j} - \binom{2\ell}{\ell}. \tag{4.29}$$

Proof. We have,

$$\sum_{j=\ell+1}^{m} \binom{m}{j} - \sum_{j=0}^{\ell} \binom{m}{j} = \sum_{j=0}^{m-\ell-1} \binom{m}{j} - \sum_{j=0}^{\ell} \binom{m}{j}.$$

For $\ell \leq \lfloor \frac{m-1}{2} \rfloor$, $m-\ell-1 \geq \ell$ meaning that (4.29) holds. For odd m, $\lfloor \frac{m-1}{2} \rfloor = \lfloor \frac{m}{2} \rfloor$ and there is nothing left to prove. When m is even, say $m=2q$, the only case left to prove is for $\ell = \lfloor \frac{m}{2} \rfloor = q$. This holds trivially since both the LHS and RHS in (4.29) simplify to $\sum_{j=q+1}^{2q} \binom{2q}{j}$. □

From (4.27) and (4.28) then,

$$\Delta(r+1, m+1) = q + \Delta_1 = \sum_{j=0}^{r+1} \binom{m+1}{j} - \sum_{j=0}^{r+1} \binom{2j}{j},$$

settling case 1.

Case 2: $r > \lfloor \frac{m}{2} \rfloor$

In this case, we have $r+1 > \lfloor \frac{m}{2} \rfloor$ and $r+1 > \lfloor \frac{m+1}{2} \rfloor$. From (4.25),

$$\begin{aligned} \Delta_1 &= \sum_{j=0}^{m-r-2} \binom{m}{j} - \sum_{j=0}^{m-r-2} \binom{2j}{j} \\ \Delta_2 &= \sum_{j=0}^{m-r-1} \binom{m}{j} - \sum_{j=0}^{m-r-1} \binom{2j}{j} \end{aligned}.$$

Here, we have $(p+\Delta_2) - (q+\Delta_1) = -\sum_{j=m-r}^{r} \binom{m}{j} - \binom{2(m-r-1)}{m-r-1} < 0$. From (4.27) then,

$$\begin{aligned} \Delta(r+1, m+1) &= p + \Delta_2 \\ &= \sum_{j=0}^{m-r-1} \binom{m+1}{j} - \sum_{j=0}^{m-r-1} \binom{2j}{j}, \end{aligned}$$

which is indeed the form in (4.25) and case 2 is settled.

Case 3: $r \leq \lfloor \frac{m}{2} \rfloor$, $r+1 > \lfloor \frac{m}{2} \rfloor$ and $m = 2s$ even.

The above assumptions simplify to $r = s$. We also have $r+1 = s+1 > s = \lfloor \frac{m+1}{2} \rfloor$. From (4.25),

$$\Delta_1 = \sum_{j=0}^{s-2} \binom{2s}{j} - \sum_{j=0}^{s-2} \binom{2j}{j} \ , \ \Delta_2 = \sum_{j=0}^{s} \binom{2s}{j} - \sum_{j=0}^{s} \binom{2j}{j}.$$

We have $(p+\Delta_2) - (q+\Delta_1) = -\binom{2s}{s} - \binom{2(s-1)}{s-1} < 0$. Thus, from (4.27),

$$\Delta(r+1, m+1) = p + \Delta_2 = \sum_{j=0}^{s-1} \binom{2s+1}{j} - \sum_{j=0}^{s-1} \binom{2j}{j}$$

which agrees with the hypothesis.

Case 4: $r \leq \lfloor \frac{m}{2} \rfloor$, $r+1 > \lfloor \frac{m}{2} \rfloor$ and $m = 2t+1$ odd.

The assumptions imply $r = t$, $r+1 = \lfloor \frac{m+1}{2} \rfloor$ and

$$\begin{aligned} \Delta_1 &= \sum_{j=0}^{t-1} \binom{2t+1}{j} - \sum_{j=0}^{t-1} \binom{2j}{j} \\ \Delta_2 &= \sum_{j=0}^{t} \binom{2t+1}{j} - \sum_{j=0}^{t} \binom{2j}{j} \end{aligned}.$$

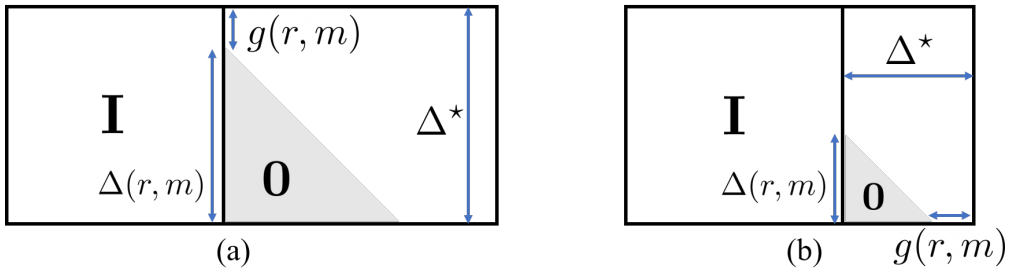


Figure 4.5. A systematic form for RM codes where the parity component has a large triangle of zeros. Also shown is the gap $g(r,m)$ from Δ^* (4.31) for code rates (a) $\gamma(r,m) < 0.5$ and (b) $\gamma(r,m) > 0.5$. For long RM codes, $\frac{g(r,m)}{\Delta^\star} \approx 0$.

We have $(p+\Delta_2)-(q+\Delta_1) = -\binom{2t}{t} < 0$. Hence,

$$\Delta(r+1,m+1) = p+\Delta_2 = \sum_{j=0}^{t+1}\binom{2t+2}{j} - \sum_{j=0}^{t+1}\binom{2j}{j},$$

which is the desired form, hence settling all cases and completing the induction. □

4.7.2 Asymptotic scaling of $\Delta(r,m)$

In this section, we study how $\Delta(r,m)$ behaves asymptotically. Denote the coding rate by

$$\gamma(r,m) = \frac{K(r,m)}{2^m} = \frac{\sum_{i=0}^{r}\binom{m}{i}}{2^m}. \tag{4.30}$$

The implication of Theorem 4.7.1 is illustrated in Fig. 4.5. RM codes admit a systematic generator matrix where one can almost fit an all-zero triangle of size

$$\Delta^\star = \min\left(K(r,m), 2^m - K(r,m)\right)$$

in the parity component except that there is a gap $g(r,m)$ from $\Delta^\star$ given by

$$g(r,m) = \begin{cases} \sum_{j=0}^{r}\binom{2j}{j} & \gamma(r,m) \le 0.5 \\ \sum_{j=0}^{m-r-1}\binom{2j}{j} & \gamma(r,m) > 0.5 \end{cases}. \tag{4.31}$$

We show that for long RM codes of constant rate, $\frac{g(r,m)}{\Delta^\star} \approx 0$. Note that $\gamma(r,m)$ in (4.30) can be interpreted to be the probability that a random binary m-tuple has Hamming weight at most r, i.e.,

$$\gamma(r,m) = Pr(X_1 + X_2 + \cdots X_m \leq r) \tag{4.32}$$

where $\{X_j\}$ are i.i.d. $Ber\left(\frac{1}{2}\right)$. Then, by the central limit theorem, long Reed-muller codes (i.e., $m \to \infty$) of constant rate $0 < \alpha < 1$ can be obtained by letting r to scale with m as

$$r = \frac{m}{2} + \frac{\sqrt{m}}{2}\Phi^{-1}(\alpha) \tag{4.33}$$

where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt \tag{4.34}$$

is the standard Gaussian CDF.

Theorem 4.7.2. *1. For long RM codes $\mathcal{R}(r,m)$ of constant-rate $\alpha < 0.5$ with r scaling as (4.33), we have*

$$\lim_{m\to\infty} \frac{\Delta(r,m)}{K(r,m)} = \lim_{m\to\infty} \frac{\Delta(r,m)}{\Delta^\star} = 1. \tag{4.35}$$

2. For long RM codes $\mathcal{R}(r,m)$ of constant-rate $\alpha > 0.5$ with r scaling as (4.33), we have

$$\lim_{m\to\infty} \frac{\Delta(r,m)}{2^m - K(r,m)} = \lim_{m\to\infty} \frac{\Delta(r,m)}{\Delta^\star} = 1. \tag{4.36}$$

Proof. For $0 < \alpha < 0.5$, (4.33) becomes $r = \frac{m}{2} - \beta\frac{\sqrt{m}}{2}$ where $\beta = -\Phi^{-1}(\alpha) > 0$ and (4.35) is

$$\lim_{m\to\infty} \frac{\Delta(r,m)}{K(r,m)} = 1 - \lim_{m\to\infty} \frac{\sum_{j=0}^{r} \binom{2j}{j}}{K(r,m)}.$$

Since $\frac{K(r,m)}{2^m} \to \alpha$, all that remains to be shown is that $\frac{\sum_{j=0}^{r} \binom{2j}{j}}{2^m} \to 0$. To see this, simply note

$$\frac{\sum_{j=0}^{r} \binom{2j}{j}}{2^m} < \frac{\sum_{j=0}^{r} 4^j}{2^m} < \frac{4}{3} 2^{2r-m} \to 0$$

where the first inequality follows from the identity $4^n = (1+1)^{2n} = \sum_k \binom{2n}{k}$.

When $\alpha > 0.5$, we have $r = \frac{m}{2} + \beta\frac{\sqrt{m}}{2}$ where $\beta = \Phi^{-1}(\alpha) > 0$ and we get

$$\lim_{m\to\infty} \frac{\Delta(r,m)}{2^m - K(r,m)} = \lim_{m\to\infty} 1 - \frac{\sum_{j=0}^{m-r-1} \binom{2j}{j}}{K(m-r-1,m)} = 1,$$

for the same reasons as above, hence proving (4.36). □

Theorem 4.7.2 thus implies that asymptotically for constant-rate RM codes, the gap $g(r,m)$ relative to $\Delta^\star$ vanishes and $\Delta(r,m) \approx \Delta^\star$.

4.8 Concluding Remarks

In this work, we investigated how the presence of binary feedback can sometimes be useful in designing good encoding schemes. We introduced the framework of linearly adapting block feedback codes and showed that weak codes can be transformed into strong ones, even when feedback limitations exist. We then proved a novel result for RM codes, showing that they admit a systematic generator matrix whose parity component has a rather large number of contiguous all-zero sub-diagonals. Our result implies that RM codes can be emulated from an uncoded system against very large feedback delays. An interesting open question is whether for finite r and m, $\mathcal{R}(r,m)$ admits a systematic form (4.24) with a Δ larger than what is proved in Theorem 4.7.1.

5. SUMMARY

In this dissertation, we looked at the the role of binary feedback in communication systems through the lens of three different problems. In chapter 2, binary feedback was cleverly leveraged to propose novel solutions for the beam alignment problem in millimeter wave systems. In chapter 3, we characterized the capacity for certain types of channel models where both stochastic and adversarial sources of noise were present simultaneously. Here, binary feedback appeared in the form of an adversary with receiver snooping abilities, and as causal receiver observation feedback to the transmitter. Finally, in chapter 4, we studied the applicability of binary feedback for encoding, and proved a new result for the important family of Reed-Muller (RM) codes.

REFERENCES

[1] Ericsson, "Mobility Report November 2017," [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2017.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014, ISSN: 0733-8716. DOI: 10.1109/JSAC.2014.2328098.

[3] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, Mar. 2014, ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2299397.

[4] 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 Communications Magazine*, vol. 52, no. 2, pp. 106–113, Feb. 2014, ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6736750.

[5] Qualcomm, "Mobilizing 5G NR Millimeter Wave: Network Coverage Simulation Studies for Global Cities," Oct. 2017. [Online]. Available: https://www.qualcomm.com/media/documents/files/white-paper-5g-nr-millimeter-wave-network-coverage-simulation.pdf.

[6] Y. Nam, Y. Akimoto, Y. Kim, M. Lee, K. Bhattad, and A. Ekpenyong, "Evolution of reference signals for LTE-advanced systems," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 132–138, Feb. 2012, ISSN: 0163-6804. DOI: 10.1109/MCOM.2012.6146492.

[7] T. K. Y. Lo, "Maximum ratio transmission," in *1999 IEEE International Conference on Communications (Cat. No. 99CH36311)*, vol. 2, Jun. 1999, 1310–1314 vol.2. DOI: 10.1109/ICC.1999.765552.

[8] M. Kobayashi, N. Jindal, and G. Caire, "Training and feedback optimization for multiuser MIMO downlink," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2228–2240, Aug. 2011, ISSN: 0090-6778. DOI: 10.1109/TCOMM.2011.051711.090752.

[9] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, "Millimeter wave beamforming for wireless backhaul and access in small cell networks," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, Oct. 2013, ISSN: 0090-6778. DOI: 10.1109/TCOMM.2013.090513.120848.

[10] J. Wang, Z. Lan, C.-w. Pyo, T. Baykas, C.-s. Sum, M. A. Rahman, J. Gao, R. Funada, F. Kojima, H. Harada, and S. Kato, "Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009, ISSN: 0733-8716. DOI: 10.1109/JSAC.2009.091009.

[11] Z. Xiao, T. He, P. Xia, and X. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3380–3392, May 2016, ISSN: 1536-1276. DOI: 10.1109/TWC.2016.2520930.

[12] J. Singh and S. Ramakrishna, "On the feasibility of codebook-based beamforming in millimeter wave systems with multiple antenna arrays," *IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 2670–2683, May 2015, ISSN: 1536-1276. DOI: 10.1109/TWC.2015.2390637.

[13] J. Song, J. Choi, and D. J. Love, "Common codebook millimeter wave beam design: Designing beams for both sounding and communication with uniform planar arrays," *IEEE Transactions on Communications*, vol. 65, no. 4, pp. 1859–1872, Apr. 2017, ISSN: 0090-6778. DOI: 10.1109/TCOMM.2017.2665497.

[14] J. Song, J. Choi, S. G. Larew, D. J. Love, T. A. Thomas, and A. A. Ghosh, "Adaptive millimeter wave beam alignment for dual-polarized MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6283–6296, Nov. 2015, ISSN: 1536-1276. DOI: 10.1109/TWC.2015.2452263.

[15] C. Tsai and A. Wu, "Structured random compressed channel sensing for millimeter-wave large-scale antenna systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 19, pp. 5096–5110, Oct. 2018, ISSN: 1053-587X. DOI: 10.1109/TSP.2018.2860545.

[16] Z. Marzi, D. Ramasamy, and U. Madhow, "Compressive channel estimation and tracking for large arrays in mm-wave picocells," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 514–527, Apr. 2016, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2016.2520899.

[17] A. Alkhateeb, O. E. Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831–846, Oct. 2014, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2014.2334278.

[18] A. Alkhateeb, G. Leus, and R. W. Heath, "Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?" In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 2909–2913. DOI: 10.1109/ICASSP.2015.7178503.

[19] M. E. Rasekh, Z. Marzi, Y. Zhu, U. Madhow, and H. Zheng, "Noncoherent mmwave path tracking," in *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '17, Sonoma, CA, USA: ACM, 2017, pp. 13–18, ISBN: 978-1-4503-4907-9. DOI: 10.1145/3032970.3032974.

[20] A. Pelc, "Solution of Ulam's problem on searching with a lie," *J. Comb. Theory Ser. A*, vol. 44, no. 1, pp. 129–140, Jan. 1987, ISSN: 0097-3165.

[21] A. Pelc, "Searching games with errors fifty years of coping with liars," *Theoretical Computer Science*, vol. 270, no. 1, pp. 71–109, 2002, ISSN: 0304-3975.

[22] M. Aigner, "Searching with lies," *Journal of Combinatorial Theory, Series A*, vol. 74, no. 1, pp. 43–56, 1996.

[23] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, 1943. DOI: 10.1214/aoms/1177731363.

[24] D. Z. Du and F. K. Hwang, *Combinatorial group testing and its applications*. World Scientific, 1999, ISBN: 978-981-02-4107-0.

[25] D. Z. Du and F. K. Hwang, *Pooling Designs and Nonadaptive Group Testing*. World Scientific, 2006, ISBN: 978-981-256-822-9.

[26] A. G Dyachkov and V. Rykov, "Survey of superimposed code theory.," vol. 12, pp. 229–242, Jan. 1983.

[27] V. Suresh and D. J. Love, "Error control sounding strategies for millimeter wave beam alignment," in *2018 Information Theory and Applications Workshop (ITA)*, Feb. 2018, pp. 1–6. DOI: 10.1109/ITA.2018.8503221.

[28] M. Hussain and N. Michelusi, "Coded energy-efficient beam-alignment for millimeter-wave networks," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2018, pp. 407–412.

[29] Y. Shabara, C. E. Koksal, and E. Ekici, "Linear block coding for efficient beam discovery in millimeter wave communication networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Apr. 2018, pp. 2285–2293. DOI: 10.1109/INFOCOM.2018.8486302.

[30] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Sparse Graph Codes for Non-adaptive Quantitative Group Testing," *arXiv e-prints*, Jan. 2019. arXiv: 1901.07635 [cs.IT].

[31] S. Han, C. l. I, Z. Xu, and C. Rowell, "Large-scale antenna systems with hybrid analog and digital beamforming for millimeter wave 5G," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 186–194, Jan. 2015, ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7010533.

[32] R. Méndez-Rial, C. Rusu, N. González-Prelcic, A. Alkhateeb, and R. W. Heath, "Hybrid MIMO architectures for millimeter wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016. DOI: 10.1109/ACCESS.2015.2514261.

[33] J. A. Aslam and A. Dhagat, "Searching in the presence of linearly bounded errors," in *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, ser. STOC '91, 1991, pp. 486–493, ISBN: 0-89791-397-3.

[34] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Elsevier, 1977, ISBN: 0924-6509.

[35] S. M. Ulam, *Adventures of a Mathematician*. 1976.

[36] E. R. Berlekamp, "Block coding with noiseless feedback," PhD thesis.

[37] D. Bertsimas and R. Weismantel, *Optimization Over Integers*. Dynamic Ideas, 2005, ISBN: 9780975914625.

[38] E. L. Lawler and S. Sarkissian, "An algorithm for "Ulam's Game" and its application to error correcting codes.," vol. 56, pp. 89–93, Oct. 1995.

[39] M. Grassl, *"Code tables: Bounds on the parameters of various types of codes"*. [Online]. Available: http://www.codetables.de/.

[40] H. Q. Ngo and D.-Z. Du, "A survey on combinatorial group testing algorithms with applications to DNA library screening," *Discrete mathematical problems with medical applications*, vol. 55, pp. 171–182, 2000.

[41] M. Aldridge, "The capacity of bernoulli nonadaptive group testing," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7142–7148, Nov. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2017.2748564.

[42] J. Scarlett and V. Cevher, "Limits on support recovery with probabilistic models: An information-theoretic framework," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 593–620, Jan. 2017, ISSN: 0018-9448. DOI: 10.1109/TIT.2016.2606605.

[43] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Journal of Combinatorial Optimization*, vol. 15, no. 1, pp. 49–59, Jan. 2008, ISSN: 1573-2886. DOI: 10.1007/s10878-007-9083-3.

[44] M. B. Malyutov and P. S. Mateev, "Planning of screening experiments for a non-symmetric response function," *Matematicheskie Zametki*, vol. 27, no. 1, pp. 109–127, 1980.

[45] *GAP – Groups, Algorithms, and Programming, Version 4.10.0*, The GAP Group, 2018.

[46] A. D'yachkov, F. Hwang, A. Macula, P. Vilenkin, and C.-w. Weng, "A construction of pooling designs with some happy surprises," vol. 12, pp. 1129–36, Nov. 2005.

[47] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 3019–3035, May 2014, ISSN: 0018-9448. DOI: 10.1109/TIT.2014.2310477.

[48] J. Scarlett and V. Cevher, "Near-optimal noisy group testing via separate decoding of items," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 5, pp. 902–915, Oct. 2018, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2018.2844818.

[49] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2148–2177, 1998. DOI: 10.1109/18.720535.

[50] M. Langberg, "Oblivious communication channels and their capacity," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 424–429, 2008. DOI: 10.1109/TIT.2007.911217.

[51] V. Guruswami and A. Smith, "Codes for computationally simple channels: Explicit constructions with optimal rate," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 2010, pp. 723–732. DOI: 10.1109/FOCS.2010.74.

[52] I. Csiszar and P. Narayan, "Arbitrarily varying channels with constrained inputs and states," *IEEE Transactions on Information Theory*, vol. 34, no. 1, pp. 27–34, 1988. DOI: 10.1109/18.2598.

[53] I. Csiszar and P. Narayan, "The capacity of the arbitrarily varying channel revisited: Positivity, constraints," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 181–193, 1988. DOI: 10.1109/18.2627.

[54] E. N. Gilbert, "A comparison of signalling alphabets," *The Bell System Technical Journal*, vol. 31, no. 3, pp. 504–522, 1952. DOI: 10.1002/j.1538-7305.1952.tb01393.x.

[55] R. R. Varshamov, "Estimate of the number of signals in error correcting codes," *Docklady Akad. Nauk, SSSR*, vol. 117, pp. 739–741, 1957.

[56] R. McEliece, E. Rodemich, H. Rumsey, and L. Welch, "New upper bounds on the rate of a code via the delsarte-macwilliams inequalities," *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 157–166, 1977. DOI: 10.1109/TIT.1977.1055688.

[57] Z. Chen, S. Jaggi, and M. Langberg, "A characterization of the capacity of online (causal) binary channels," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 287–296.

[58] B. K. Dey, S. Jaggi, M. Langberg, and A. D. Sarwate, "Upper bounds on the capacity of binary channels with causal adversaries," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3753–3763, 2013.

[59] R. Bassily and A. Smith, "Causal erasure channels," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, SIAM, 2014, pp. 1844–1857.

[60] B. K. Dey, S. Jaggi, and M. Langberg, "Codes against online adversaries: Large alphabets," *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3304–3316, 2013. DOI: 10.1109/TIT.2013.2245717.

[61] B. K. Dey, S. Jaggi, M. Langberg, and A. D. Sarwate, "Coding against delayed adversaries," in *2010 IEEE International Symposium on Information Theory*, 2010, pp. 285–289.

[62] B. K. Dey, S. Jaggi, M. Langberg, and A. D. Sarwate, "A bit of delay is sufficient and stochastic encoding is necessary to overcome online adversarial erasures," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 880–884.

[63] B. K. Dey, S. Jaggi, M. Langberg, A. D. Sarwate, and C. Wang, "The interplay of causality and myopia in adversarial channel models," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 1002–1006.

[64] B. K. Dey, S. Jaggi, and M. Langberg, "Sufficiently myopic adversaries are blind," *IEEE Transactions on Information Theory*, vol. 65, no. 9, pp. 5718–5736, 2019.

[65] M. Langberg, "Private codes or succinct random codes that are (almost) perfect," in *45th Annual IEEE Symposium on Foundations of Computer Science*, 2004, pp. 325–334. DOI: 10.1109/FOCS.2004.51.

[66] A. Smith, "Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07, New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 395–404, ISBN: 9780898716245.

[67] S. Bhattacharya, A. J. Budkuley, and S. Jaggi, "Shared randomness in arbitrarily varying channels," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 627–631. DOI: 10.1109/ISIT.2019.8849801.

[68] I. Csiszar and J. Körner, *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[69] V. Suresh, E. Ruzomberka, and D. J. Love, "Stochastic-adversarial channels : Online adversaries with feedback snooping," in *2021 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2021.

[70] V. Suresh, E. Ruzomberka, and D. J. Love, *Stochastic-adversarial channels : Online adversaries with feedback snooping*, 2021. arXiv: 2104.07194 [cs.IT]. [Online]. Available: https://arxiv.org/abs/2104.07194.

[71] M. Langberg, S. Jaggi, and B. K. Dey, "Binary causal-adversary channels," in *2009 IEEE International Symposium on Information Theory*, IEEE, 2009, pp. 2723–2727.

[72] K. Zigangirov, "On the number of correctable errors for transmission over a binary symmetrical channel with feedback," *Problemy Peredachi Informatsii*, vol. 12, no. 2, pp. 3–19, 1976.

[73] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback–i: No bandwidth constraint," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 172–182, Apr. 1966. DOI: 10.1109/TIT.1966.1053879.

[74] J. Schalkwijk, "A coding scheme for additive noise channels with feedback–ii: Band-limited signals," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 183–189, Apr. 1966. DOI: 10.1109/TIT.1966.1053880.

[75] S. Butman, "A general formulation of linear feedback communication systems with solutions," *IEEE Transactions on Information Theory*, vol. 15, no. 3, pp. 392–400, May 1969. DOI: 10.1109/TIT.1969.1054302.

[76] 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, Oct. 2011. DOI: 10.1109/TIT.2011.2165796.

[77] E. Ardestanizadeh, M. Wigger, Y. Kim, and T. Javidi, "Linear-feedback sum-capacity for gaussian multiple access channels," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 224–236, Jan. 2012, ISSN: 1557-9654. DOI: 10.1109/TIT.2011.2169307.

[78] S. Belhadj Amor, Y. Steinberg, and M. Wigger, "MIMO MAC-BC duality with linear-feedback coding schemes," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 5976–5998, Nov. 2015, ISSN: 1557-9654. DOI: 10.1109/TIT.2015.2473838.

[79] Young-Han Kim, "Feedback capacity of the first-order moving average gaussian channel," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 3063–3079, Jul. 2006, ISSN: 1557-9654. DOI: 10.1109/TIT.2006.876217.

[80] R. Dobrushin, "Asymptotic bound on the error probability for message transmission over a memoryless channel with feedback," *Probl. Kibern.*, vol. 8, pp. 161–168, 1962.

[81] M. V. Burnasev, "Information transmission over discrete channels with feedback. Random transmission time.," Russian, *Probl. Peredachi Inf.*, vol. 12, no. 4, pp. 10–30, 1976, ISSN: 0555-2923.

[82] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Feedback in the non-asymptotic regime," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011, ISSN: 1557-9654. DOI: 10.1109/TIT.2011.2158476.

[83] B. Nakiboğlu and R. G. Gallager, "Error exponents for variable-length block codes with feedback and cost constraints," *IEEE Transactions on Information Theory*, vol. 54, no. 3, pp. 945–963, Mar. 2008, ISSN: 1557-9654. DOI: 10.1109/TIT.2007.915913.

[84] K. Vakilinia, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary LDPC examples," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2245–2257, Jun. 2016, ISSN: 1558-0857. DOI: 10.1109/TCOMM.2016.2538770.

[85] A. R. Williamson, T. Chen, and R. D. Wesel, "Variable-length convolutional coding for short blocklengths with decision feedback," *IEEE Transactions on Communications*, vol. 63, no. 7, pp. 2389–2403, Jul. 2015, ISSN: 1558-0857. DOI: 10.1109/TCOMM.2015.2429583.

[86] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977, vol. 16.

[87] [Online]. Available: https://math.stackexchange.com/q/3526270.

[88] R. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.

[89] E. Abbe, A. Shpilka, and A. Wigderson, "Reed–Muller codes for random erasures and errors," *IEEE Transactions on Information Theory*, vol. 61, no. 10, pp. 5229–5252, 2015.

[90] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[91] S. Kudekar, S. Kumar, M. Mondelli, H. D. Pfister, E. Şaşoğlu, and R. L. Urbanke, "Reed–Muller codes achieve capacity on erasure channels," *IEEE Transactions on information theory*, vol. 63, no. 7, pp. 4298–4316, 2017.

[92] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, 2008.

[93] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "From polar to Reed-Muller codes: A technique to improve the finite-length performance," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3084–3091, 2014.

[94] O. Sberlo and A. Shpilka, "On the performance of Reed-Muller codes with respect to random errors and erasures," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2020, pp. 1357–1376.

[95] E. Arikan, "Systematic polar coding," *IEEE communications letters*, vol. 15, no. 8, pp. 860–862, 2011.

[96] P. Hauck, M. Huber, J. Bertram, D. Brauchle, and S. Ziesche, "Efficient majority-logic decoding of short-length Reed-Muller codes at information positions," *IEEE transactions on communications*, vol. 61, no. 3, pp. 930–938, 2013.

[97] E. Abbe, A. Shpilka, and M. Ye, "Reed-Muller codes: Theory and algorithms," *arXiv preprint arXiv:2002.03317*, 2020.